ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

TOWARDS AN OUTPUT-BASED RE-MESHING FOR TURBOMACHINERY APPLICATIONS

Mateusz Gugala[1], Marcus Meyer[2] and Jens-Dominik Müller[3]

[1] Queen Mary, University of London The School of Engineering and Material Science
Mile End Road, London, E1 4NS, UK,
e-mail: *m.gugala@qmul.ac.uk

[2] Rolls-Royce Deutschland Ltd & Co KG
Eschenweg 11, 15827 Blankenfelde-Mahlow,
e-mail: Marcus.Meyer@rolls-royce.com

[3] Queen Mary, University of London The School of Engineering and Material Science
Mile End Road, London, E1 4NS, UK,
e-mail: j.mueller@qmul.ac.uk

Keywords: Mesh adaptation, Adjoint method, Multi-grid, Output-based sensor, Truncation error, Re-meshing, Turbomachinery.

Abstract. *A methodology for estimating the truncation error between a fine computational grid and topologically inconsistent coarse grid from a multi-grid sequence is presented. This sensor is then weighted with the adjoint solution to obtain a goal-based refinement sensor which provides local scales for a re-meshing procedure. The truncation-based sensor, as well as an output-based indicator, were implemented in the proprietary Rolls-Royce code Hydra. The output-based re-meshing refinement is first applied to the simple cube case with an inviscid flow. The truncation-based sensor and the output-based sensor are then evaluated for a turbine stator case as a first step towards the turbomachinery application of the re-meshing refinement methodology. The results are presented and discussed.*

1 INTRODUCTION

The adjoint method [1] is well-established as the most efficient method for an aerodynamic shape optimisation with CFD. Adjoint sensitivities can be also used to obtain a robust adaptation sensor for an objective function which drives a solution-adaptive mesh refinement process that computes a more accurate objective function at the lower computational cost compared to error-estimators without adjoint weighting or compared to heuristic sensors. The adjoint-weighted truncation error known also as an output-based indicator gained popularity as an effective driver for an adaptation process - see e.g. [2, 3, 4]. Through the adjoint weighting, the mesh adaptation is very effectively targeted to those areas of the computational domain where the objective function is highly sensitive to mesh resolution and the local error estimate is large. That is the key advantage of an output-based indicator as compared to other approaches such as e.g. gradient/Hessian-based sensors or pure truncation-error-based sensors. While the latter at least attempts to estimate the actual errors, both of these methods apply the refinement to all errors, regardless of whether they are relevant to the computation of the objective function or not.

Assuming that the adjoint solver is available, the remaining challenge in obtaining output-based sensor lies in the estimation of the truncation error. A popular approach in the literature has been presented by Venditti and Darmofal [5]. The computational grid is locally refined preserving its original topology and the current solution interpolated to the refined grid is used to estimate the truncation error on the computational grid. Further refinement and application of this approach can be found in [2]. Even though no full solution is computed on the refined mesh, some work as higher-order interpolation or a few iterations have to be done on the refined grid and the computational cost is not negligible.

As an alternative, we could consider using the difference between finest computational grid and a coarsened grid to estimate the local error. In this variant the computational effort is significantly reduced and in case of fully coarsened geometric multi-grid methods can even be at no additional effort since the coarse grid is computed anyways. On the other hand, the error estimate is then only valid for the coarse grid, and care has to be taken to make the extrapolation to the fine grid valid.

Using geometric multi-grid levels for truncation error estimation has been proposed by Fraysse and Ponsin [4]. In this approach, the truncation error is estimated on the finest grid level using a topologically consistent coarser mesh. In our approach, we explore topologically inconsistent fine and coarse grids as arising from unstructured grid coarsening [6].

The proposed output-based sensor estimation methodology described in section 2 was implemented in the Rolls-Royce proprietary CFD code Hydra. A small number of explicit smoothing iterations (see appendix A) is applied to the obtained refinement indicator in order to regularise unwanted high-frequency modes arising mainly from the topological inconsistency between meshes used for truncation error estimation. The obtained sensor fields can either be used for hierarchical refinement, leading to topologically consistent grids, or be used in as local sizing fields in a re-meshing procedure as is the case in this work. Obtained fields are used to drive re-meshing process using BoxerMesh[1] [7]. The application of the output-based re-meshing refinement to the simple cube case with inviscid flow is presented in section 4.1. The preliminary results i.e. truncation sensor and output sensor plots are presented for a turbine stator case and discussed in section 4.2.

[1] http://www.cambridgeflowsolutions.com/en/products/boxer-mesh/

2 Description of Methods

2.1 Basic Quantities and Output Error

The discretisation error δU_h is defined as the difference between the exact solution U of the continuous PDE system $R(U) = 0$ and its discrete approximation U_h, where the latter is resulting from the discrete PDE system solve $R_h(U_h) = 0$. The notation $\cdot|_h$ indicates that the exact quantity is either evaluated at the discrete space with characteristic size h, or interpolated onto this space, not to be confused with the discrete quantity U_h denoted only with subscript h. More formally it can be written using a symbolic interpolation operator $U|_h = \mathcal{I}^h U$. However, to keep the derivations more transparent the following notation is used in this section.

$$\delta U_h = U|_h - U_h \tag{1}$$

The truncation error δR_h can obtained using Taylor series expansion (2) which also shows the relation between the discretisation error and the truncation error.

$$R(U)|_h - R_h(U|_h) = \left(\frac{\partial R}{\partial U}\right)_h \delta U_h + \dots \tag{2}$$

An output error δJ_h due to the inexact solution can be similarly expanded using Taylor series:

$$J_h(U|_h) - J_h(U_h) = \left(\frac{\partial J}{\partial U}\right)_h \delta U_h + \dots \tag{3}$$

Replacing δU_h in (3) with the discretisation error derived from (2) and skipping higher order terms, one can arrive at the formula (4):

$$\delta J_h \approx \left(\frac{\partial J}{\partial U}\right)_h \left(\frac{\partial R}{\partial U}\right)_h^{-1} \delta R_h = v_h^T \, \delta R_h \tag{4}$$

The adjoint variable v for the objective function J, e.g. lift, drag, translates the truncation error δR into the error in cost function δJ. In this manner, the information on how the truncation errors in each control volume contribute to the error in the cost function is obtained. Summation of all contributions together gives a scalar variable i.e. the output correction, which can be used to increase cost function estimation accuracy. The discrete adjoint approach is used in order to obtain adjoint variable ν_h. Tapenade[2] algorithmic differentiation tool is used for flow solver differentiation. The adjoint system is presented in equation (5). Using discrete adjoint allow obtaining exact gradients that correspond to the cost function evaluated on the discrete space J_h.

$$\left(\frac{\partial R}{\partial U}\right)_h v_h^T = \left(\frac{\partial J}{\partial U}\right)_h \tag{5}$$

Assuming that an adjoint solution is available, the remaining task is to get a good estimate of the truncation error.

2.2 Truncation Error Estimation

The truncation error TE_H is the difference between a mathematical model (PDE, denoted $R(U) = 0$) and its discrete approximation $R_H(U|_H)$, or in other words, it is the error due

[2]AD tool developed at Inria http://www-sop.inria.fr/tropics/

to the truncation of the continuous model. First, the exact system $R(U)$ is expanded using Taylor series around a point in space H where the higher order components are denoted TE_H - equation (6). The truncation error can be evaluated using discrete operator R_H and an exact solution U, where the former is evaluated at $|_H$, the discrete space H - equation (8). The operator $R_H(U|_H)$ is a finite-volume residual which is an integral quantity over each control volume. The truncation error is defined for PDE so the mentioned residual has to be divided by the volume. In this work the TE_H^Ω indicates the undivided finite-volume residual scaled with the volume, and $TE_H = TE_H^\Omega/\Omega$ denotes the truncation error in PDE - as per its definition.

$$R(U) = R_H(U|_H) + TE_H^\Omega \tag{6}$$

$$TE_H^\Omega = R(U) - R_H(U|_H), \;\; R(U) = 0 \tag{7}$$

$$TE_H^\Omega = -R_H(U|_H), \; TE_H = TE_H^\Omega/\Omega_H \tag{8}$$

The truncation error calculated using formula (8) will be called exact truncation error and it is applicable only when the exact solution U is known. In practical applications the truncation error has to be estimated. For this purpose the exact solution U is replaced with its approximate i.e. discrete solution U_h resulting from the discrete system solve $R_h(U_h) = 0$.

$$TE_H = -R_H(\mathcal{I}_h^H U_h) \tag{9}$$

In order to get the truncation error estimate on the finest mesh the prolongation operator is used:

$$TE_h = \mathcal{I}_H^h TE_H \tag{10}$$

The list of simplifications that lead to errors in the estimated TE using presented methodology are following:

- The fine grid solution is used as an approximation of the exact one.
- In practice, a finite precision interpolation operators are used.
- It is assumed that the truncation error estimated on space H can be used as an approximation of truncation errors on space h which should be valid as long as both meshes lie within the asymptotic convergence range.

As mentioned in the last point, both meshes, i.e. coarse (H) and fine (h), should lie within the asymptotic convergence range of solution U in order to get a good error estimate. However in practice, it was shown that even for cases which do not meet this criterion the refinement driven by an adaptation sensor based on the method described can be effective, see e.g. Fidkowski and Darmofal [2].

Within the context of geometric multi-grid solvers, inter-grid transfer operators between fine and coarse grids are available in the solver, and the error estimation can be implemented with a very low effort. The complete procedure used for truncation error estimation is presented in algorithm 1.

Algorithm 1 Truncation Error Estimation Using Geometric Multi-grid

1: Solve discrete system for mesh (h)	U_h		
2: Restrict solution from fine (h) to coarse (H) grid	U_H^h	$\leftarrow$	$\mathcal{I}_h^H U_h$
3: Estimate truncation error on the coarse grid	TE_H	$\leftarrow$	$R_H(U_H^h)/\Omega_H$
4: Prolong TE_H to fine grid, add remaining (h) residual	TE_h	$\leftarrow$	$R_h(U_h)/\Omega_h - \mathcal{I}_H^h TE_H$

2.3 Adaptation Sensors

Two adaptation sensors are used for the re-meshing algorithm: a) a truncation-error-based sensor TS and b) an output-error-based indicator OS. The variable N_{eq} denotes the number of equations to solve (e.g. 6 for the RANS with Spalart-Allmaras turbulence model). The truncation error sensor TS is a sum of the absolute values of the truncation error from each equation as shown in (11). The sensor is multiplied with the characteristic size of each cell i.e. $h_i = \Omega^{1/3}$ which plays a role of scaling factor in order to prevent infinite refinement in the regions where the errors decrease at a very low rate or diverge (e.g. at shocks) see e.g. Fraysse [8].

$$TS_{h,i} = h_i \sum_{j=1}^{N_{eq}} |TE_{h,i,j}| \tag{11}$$

The output sensor (OS) is the sum of the undivided local truncation errors weighted by the adjoint solution as shown in (12). It hence is the absolute value of the sum of the contributions of each control volume to the error in the objective function.

$$OS_{h,i} = \left| \sum_{j=1}^{N_{eq}} OE_{h,i,j} \right| = \left| \sum_{j=1}^{N_{eq}} v_{h,i,j}\, TE_{h,i,j}^{\Omega} \right| \tag{12}$$

Five explicit smoothing iterations are applied to obtained sensors in order to regularise unwanted high-frequency modes - section A.

3 Re-meshing Strategy

The re-meshing approach using BoxerMesh[3] [7] and the output-based sensor defined in equation (12) is used to drive the adaptation process. The mesher uses octree cut-cell algorithm to create an initial mesh respecting defined areas of refinement. Created octree mesh is then fitted to the geometry defined by the user and in the final step the boundary layer is extruded. An example cross-section of the stator blade mesh is presented in figure 1.

[3] http://www.cambridgeflowsolutions.com/en/products/boxer-mesh/

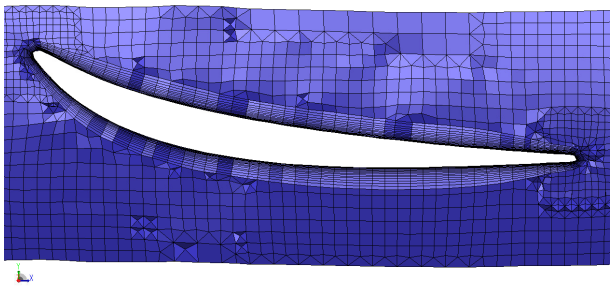


Figure 1: *Cross-section of the stator mesh generated in Boxer.*

The procedure for single re-meshing step is as follows:

1. Obtain flow solution (U_h).
2. Estimate truncation error (TE_h) as presented in algorithm 1.
3. Obtain adjoint solution (ν_h).
4. Evaluate output-based sensor (OS_h) - eqn. (12).
5. Perform 5 explicit smoothing iterations (see appendix A) on obtained sensor (OS_h) to damp unwanted high-frequency modes.
6. Use Paraview to extract mesh region for refinement.
    - Use the 'Threshold' option to mark region for refinement, figure 2.
    - Extract surface and output an STL file.
7. Import surface to Boxer and specify new refinement region for octree mesher.
8. Generate new mesh and re-run the case.

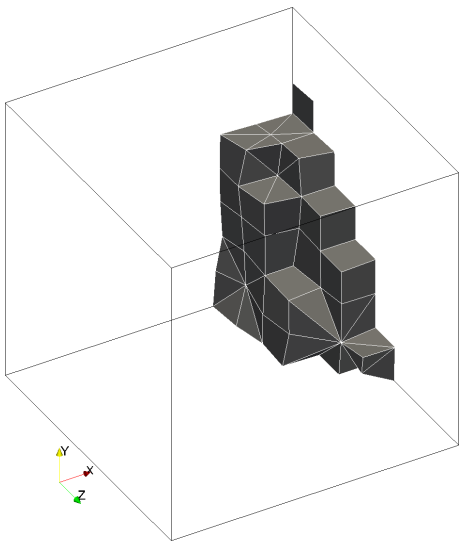


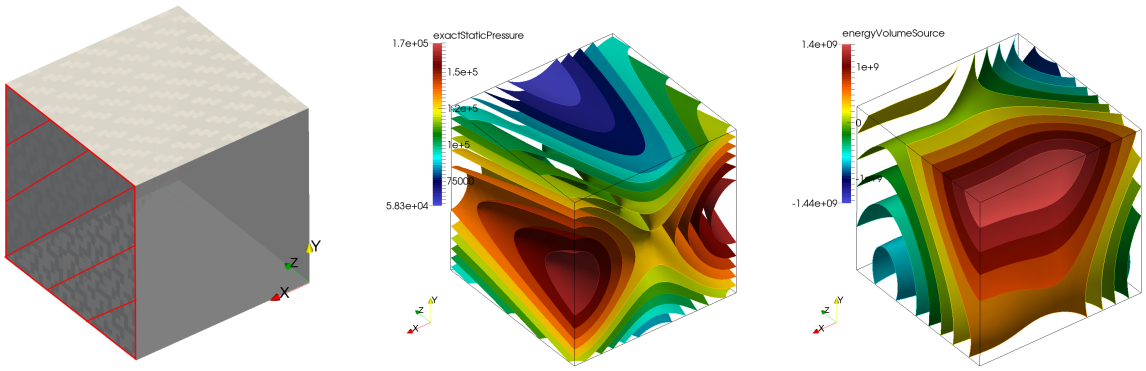
Figure 2: *Cells forming region for refinement obtained using 'Threshold' feature in Paraview.*

Currently steps (5-8) requires manual operations from the user.

4 Application

4.1 Cube with 3D Manufactured Solution

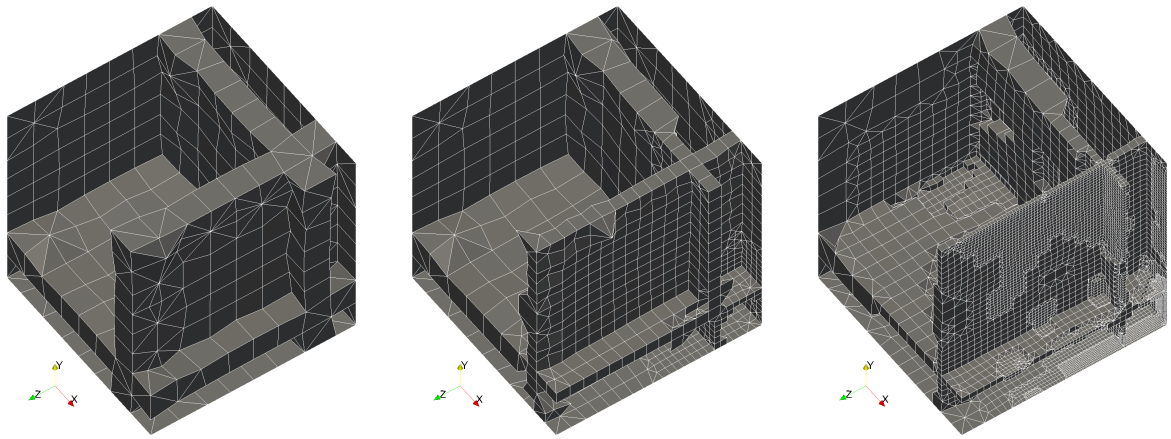
As the first output-based re-meshing example the cube case with 3D manufactured solution by Roy [9] is used. Although the case is physically meaningless it is challenging for the solver as it uses highly nonlinear manufactured solution. It is a compressible, supersonic Euler flow where the example pressure field and corresponding manufactured source term are presented on figure 3. The objective function is a drag force integrated over on one of the cube sides marked in figure 3a.



(a) *Patch for objective evaluation* (b) *pressure (p)* (c) *energy src* (f_e)

Figure 3: The 3D supersonic manufactured solution.

The initial grid was generated using Boxer and is of mixed cell type. The coarse grid for truncation error estimation was generated using internal Rolls-Royce edge-collapsing tool. The re-meshing was performed according to the procedure described in section 3. Total two re-meshing steps were applied and the resulting refined meshes are presented on figure 4. The complex and non-intuitive refinement structures can be easily noticed.



(a) *Initial mesh* (b) *Re-mesh step 1* (c) *Re-mesh step 2*

Figure 4: *Re-meshing process driven by the output-error-based sensor.*

Figure 5 shows the comparison of the achieved error in objective function for the uniformly refined set of grids - 3x3x3 to 129x129x129, regular hex where each refinement stage was achieved by halving the coarser grid edges, and the output-based re-meshed grids. The improved convergence slope for the later approach is obtained - between h^3-h^4. The summary table 1 shows that the same objective function accuracy can be obtained for almost an order of magnitude lower mesh size when the output-based sensor is used, even though a rather crude re-meshing methodology and the topologically inconsistent meshes for truncation error estimation were used. The topological inconsistency between grids is what distinguishes truncation error estimation methodology used in this work as compared to nested grids used by Ponsin and Fraysse [4] as well as Venditti or Fidkowski [2, 3].

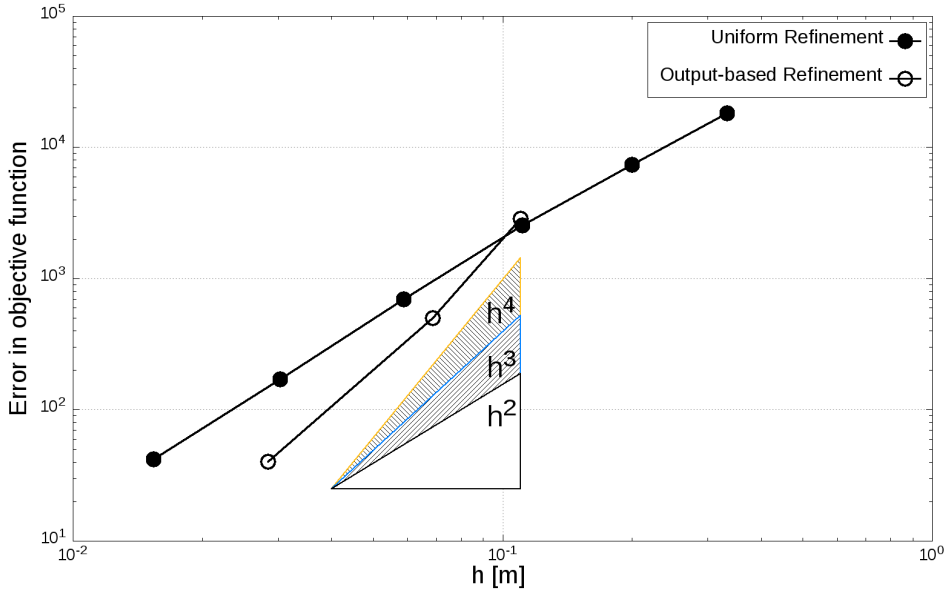


Figure 5: *Re-meshed vs uniformly refined regular mesh - error convergence comparison.*

Re-meshing stage	$\delta L/\widetilde{L}$ [%]	N^{OS}_{DoF}	N^{U}_{DoF}	N^{OS}_{DoF} / N^{U}_{DoF}
0	2.11	754	660	0.87
1	0.37	3082	12100	3.9
2	0.03	43349	335000	7.7

Table 1: Quantitative comparison of achieved objective accuracy between re-meshed grids using output-based sensor and uniformly refined regular hex meshes. DoF - degrees of freedom, N^{OS} - DoF for output-based refinement, N^{U} - DoF for corresponding uniformly refined grid.

4.2 Preliminary Application Results for the TurboLab Stator[4] Case from TU Berlin

The turbine stator from TurboLab at TU Berlin is used The boundary conditions are 42 degrees of swirl angle at the inlet, and outlet static pressure adjusted to keep the mass flow

[4]http://aboutflow.sems.qmul.ac.uk/events/munich2016/benchmark/testcase3/

rate at 9.0 kg/s. The objective function is total pressure loss weighted by mass flow. The flow solution is presented in Figure 6 which shows static pressure on the hub and blade whereas velocity profiles are presented in axial and radial sections.

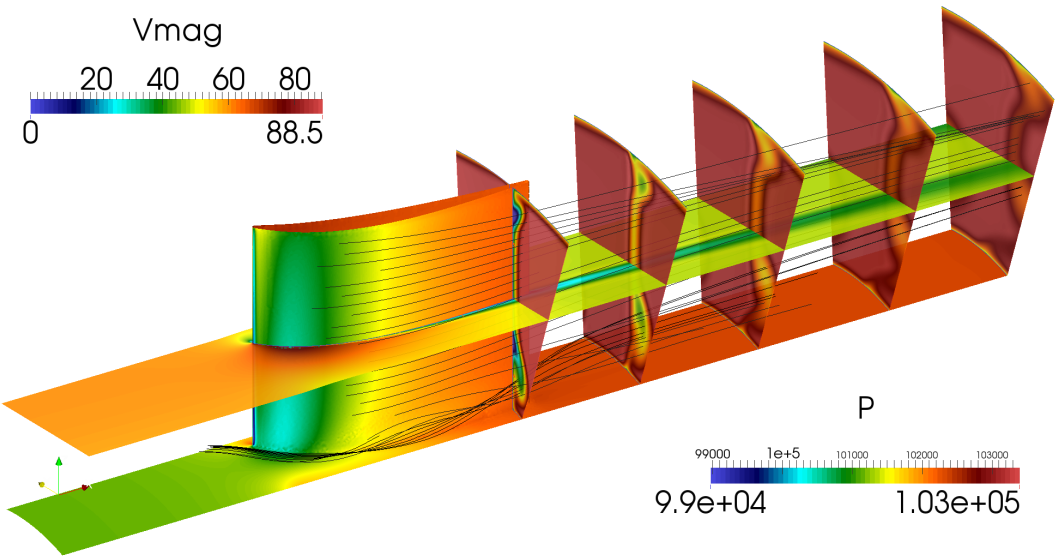


Figure 6: *Flow around the stator blade.*

Figure 7 shows the iso-volume that encloses regions within the domain with the truncation sensor value (11) above a user specified threshold.

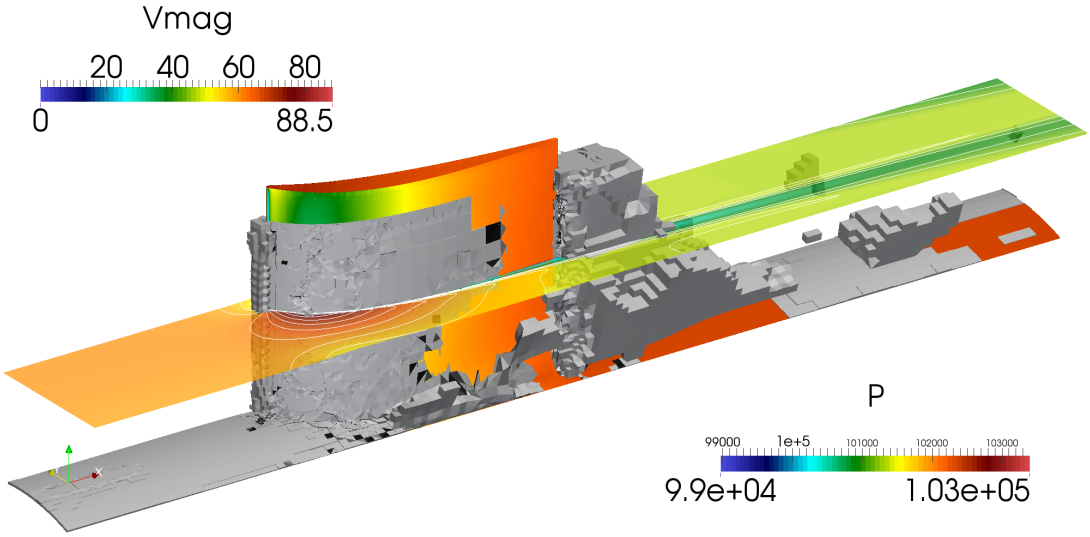


Figure 7: *Iso-volume of truncation sensor.*

Similarly, figure 8 presents iso-volumes for the output sensor (12). The iso-fields were clipped to hide the casing and make the picture more transparent. The fields differ significantly. The truncation sensor targets mainly leading and trailing edges and regions close to the blade. The output-sensor however marks mainly coarse regions of mesh at the proximity of the

inlet and regions in the wake. Both sensors capture the region of a spinning horseshoe vortex between the hub and the blade suction side - visible in figure 6.

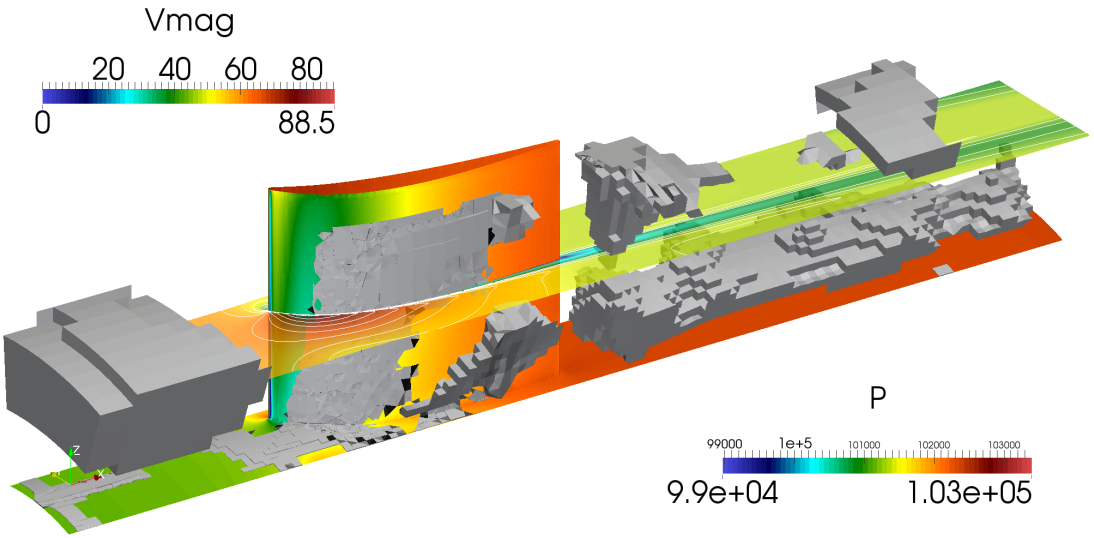


Figure 8: *Iso-volume of output sensor.*

At the moment, only the sensor fields were generated. Application of the re-meshing to the stator case is an ongoing work.

5 Summary

The geometric multi-grid method with topologically inconsistent fine and coarse grids was used for truncation error estimation. The methodology was successfully applied to the simple cube case with a highly nonlinear manufactured solution. Roughly an order of magnitude grid size reduction was obtained for the re-meshed grid as compared to uniformly refined case keeping the same cost function accuracy. The preliminary results for the turbine stator case were presented and the obtained truncation sensor and output sensor patterns compared and discussed. The re-meshing refinement application of the obtained sensor fields to the presented turbine stator case is and ongoing work.

Acknowledgement

This research utilised Queen Mary's MidPlus computational facilities, supported by QMUL Research-IT and funded by EPSRC grant EP/K000128/1.

The authors gratefully acknowledge the permission of Rolls-Royce Deutschland to publish this paper. This work has been conducted within the AboutFlow project[5], funded by the European Union FP7 Programme for research, technological development and demonstration under Grant Agreement No. 317006.

The author gratefully acknowledges and thank B. Diskin and F. Fraysse for valuable comments on author's questions (electronic correspondence) regarding their work and publications.

[5]http://aboutflow.sems.qmul.ac.uk

REFERENCES

[1] Pironneau, O. On optimum design in fluid mechanics. **64**, 97–110 (1974).

[2] Fidkowski, K. J. and Darmofal, D. L. Review of output-based error estimation and mesh adaptation in computational fluid dynamics. **49**(4), 673–694 (2011).

[3] Darmofal, D. and Venditti, D. Anisotropic grid adaptation for functional outputs: application to two-dimensional viscous flows. *JCP* **187**, 22–46 (2003).

[4] Ponsin, J., Fraysse, F., Gómez, M., and Cordero-Gracia, M. An adjoint-truncation error based approach for goal-oriented mesh adaptation. *Aerospace Science and Technology* **41**, 229–240 (2015).

[5] Venditti, D. A. and Darmofal, D. L. Grid adaptation for functional outputs: application to two-dimensional inviscid flows. *Journal of Computational Physics* **176**(1), 40–69 (2002).

[6] Moinier, P., Müller, J.-D., and Giles, M. B. Edge-based multigrid schemes and preconditioning for hybrid grids. **40**(10), 1954–60 (2002).

[7] Demargne, A., Evans, R., Tiller, P., and Dawes, W. N. Practical and reliable mesh generation for complex, real-world geometries. In *52nd AIAA Aerospace Sciences Meeting-AIAA Science and Technology Forum and Exposition, SciTech 2014*. American Institute of Aeronautics and Astronautics Inc., (2014).

[8] Fraysse, F., Valero, E., and Ponsin, J. Comparison of mesh adaptation using the adjoint methodology and truncation error estimates. *AIAA journal* **50**(9), 1920–1932 (2012).

[9] Roy, C. J., Smith, T. M., and Ober, C. C. Verification of a compressible cfd code using the method of manufactured solutions. *energy* **2**, s2 (2002).

[10] Desbrun, M., Meyer, M., Schröder, P., and Barr, A. H. Implicit fairing of irregular meshes using diffusion and curvature flow. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, 317–324. ACM Press/Addison-Wesley Publishing Co., (1999).

Appendix

A Weighted Explicit Laplacian Smoothing

The Laplacian equation has the form presented in equations 13. It can be easily discretised using e.g. and explicit time stepping and an edge-based data structure as presented in equation (14). The constant $\beta = 1.0$ was used which is related to the maximum allowed time step, constrained by the stability condition of an explicit scheme. The edge-length-weighting is applied in order to prevent distortions due to the grid size, see e.g. Desbrun [10].

$$\frac{\partial \phi}{\partial t} = \Delta \phi \tag{13}$$

$$\phi_i^{n+1} = \phi_i^n + \beta \left(\sum_{j=1}^{m} \frac{\phi_j^n - \phi_i^n}{l_{ij}} \right) \Bigg/ \sum_{j=1}^{m} l_{ij} \tag{14}$$

Figure 9 shows the eigenvalues of explicit smoothing system matrix for a simple 1D case. Each eigenvalue corresponds to the eigenvector i.e. shape mode. The magnitude of the eigenvalue tells how much given shape mode will be damped when the smoothing is performed. The graph confirms that the explicit smoothing is very effective in filtering the high-frequency modes while having a minor influence on the the low-frequency modes.

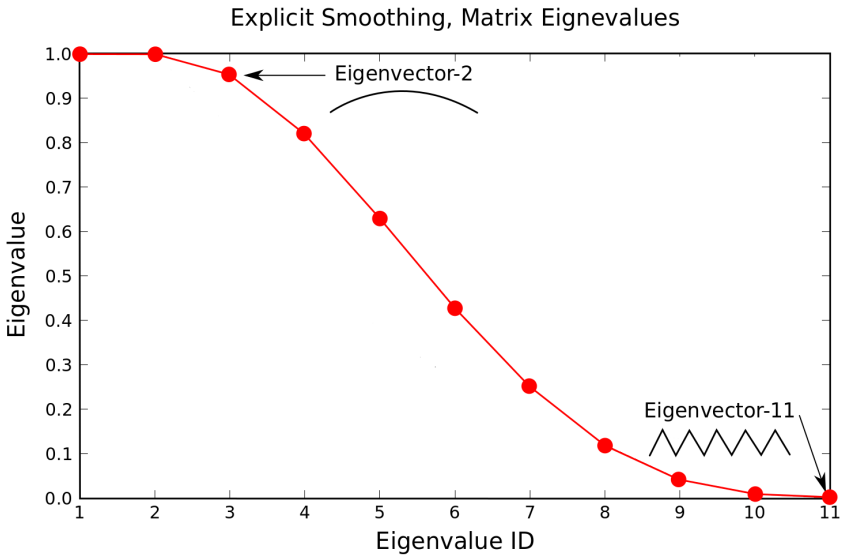


Figure 9: *Eigenvalues of explicit smoothing system matrix for simple 1D case.*