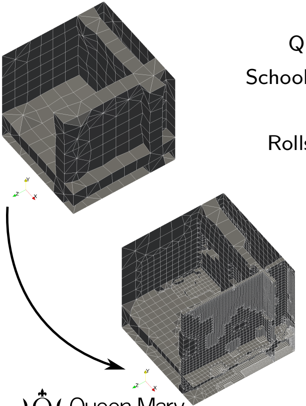
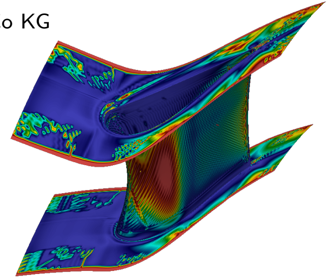
Towards An Output-based Re-meshing for Turbomachinery Applications.

Mateusz Gugała, Marcus Meyer, Jens-Dominik Müller

Queen Mary University of London,
School of Engineering and Material Science
&
Rolls-Royce Deutschland Ltd & Co KG



June, 2016



Queen Mary University of London

ABOUTflow

Contents

Background, goals and objectives

- Part of AboutFlow project
  Adjoint-based optimisation for unsteady and industrial flows.
- This work is an outcome of 3-month secondment at RR.
- Goals and objectives:
  - Implement truncation error and output error estimation procedures in Hydra (RR proprietary code). Use topologically inconsistent geometric multi-grid meshes.
  - Verify implementation using method of manufactured solution (MMS).
  - Perform re-meshing refinement driven by estimated output-based sensor.

Background, goals and objectives

Hydra is a vertex-centred finite volume flow and adjoint solver, using edge-based data structure. Other relevant info for this work:

- 2^{nd} order accurate spatial discretisation (verified using MMS).
- Semi-automatically generated **discrete adjoint solver** (*Tapenade*[1]),
- **Geometric multi-grid** solver - reused for truncation error estimation. Multi-grid meshes created using edge-collapsing algorithm.

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

Mesh adaptation - intro

- Error estimation
  - discretisation error (δU) - error in the solution
  - **truncation error** (δR) - imbalance in equations
  - output (objective function) error (δL)
- Adaptation indicator evaluation (scalar, metric)
  - feature-based: uses some form of the discretisation error
  - truncation-based
  - **output-based** - adjoint-weighted truncation error
- Adaptation method/algorithm:
  - **re-meshing**: in this work, use BoxerMesh to re-generate grid respecting regions of the computational domain marked for refinement
  - r-refinement: relocate mesh nodes, keep mesh size constant
  - h-refinement: refine mesh by subdividing cells/edges
  - p-refinement: changing order of discretisation polynomial

Truncation error estimation - intro

Broadly speaking, the truncation error (TE, δR) is the difference between a mathematical model (PDE) and its discrete approximation. TE is commonly estimated using two grids approach, a coarse grid denoted (H) and a fine grid (h).

- In the work of e.g. Venditti and Darmofal [1] [2] [3], Fidkowski [4], TE estimation requires construction of a finer embedded grid h in order to estimate TE on mesh H. The coarse mesh H is used for adaptation.
- As an alternative, for CFD solvers equipped with the **geometric multi-grid**, the truncation error can be estimated in a cheap way without much additional implementation effort. In this approach coarse grid H is used to estimate the error on the fine mesh h. The fine mesh h is used for adaptation. Similar method was presented by Fraysse [5] or Ponsin [6].

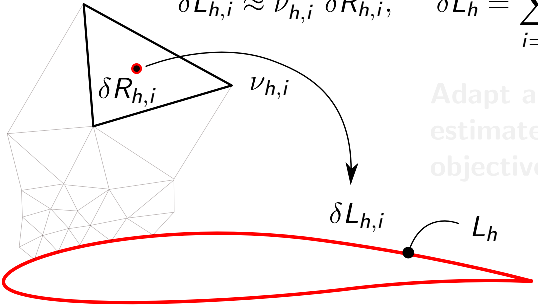
Output error and adjoint variable - intro

Exact objective function (e.g. lift or drag):

$$\widetilde{L} = L_h + \delta L_h$$

Error in objective function for discrete space witch characteristic size (h) can be estimated as follows (derivation in the conference paper):

$$\delta L_{h,i} \approx \nu_{h,i}^T \, \delta R_{h,i}, \qquad \delta L_h = \sum_{i=1}^{N} \delta L_{h,i}$$



Adapt a cell only if the estimated error influences the objective function of interest.

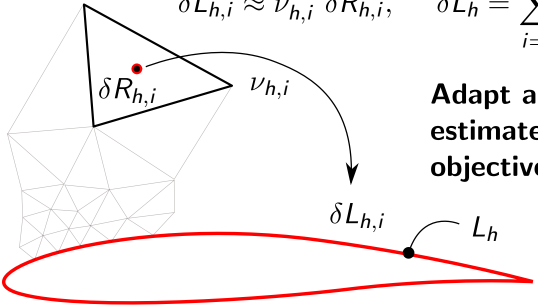
Output error and adjoint variable - intro

Exact objective function (e.g. lift or drag):

$$\widetilde{L} = L_h + \delta L_h$$

Error in objective function for discrete space witch characteristic size (h) can be estimated as follows (derivation in the conference paper):

$$\delta L_{h,i} \approx \nu_{h,i}^T \, \delta R_{h,i}, \qquad \delta L_h = \sum_{i=1}^{N} \delta L_{h,i}$$

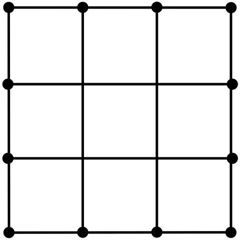


Adapt a cell only if the estimated error influences the objective function of interest.

Error estimation

Truncation error estimation with multi-grid - procedure

Fine mesh (h)

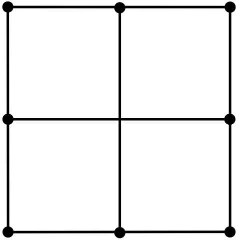


$$U_h \xrightarrow{\mathcal{I}_h^H} U_H^h$$

$$R_h^H \xleftarrow{\mathcal{I}_H^h} R_H(U_H^h)$$

$$\delta R_h = R_h(U_h) - R_h^H$$

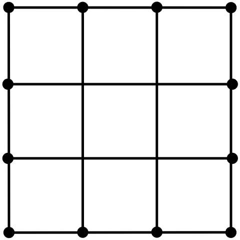
Coarse mesh (H)



1. Solve primal, $R_h(U_h) = 0$
2. Restrict primal solution to the coarse mesh, $U_H^h = \mathcal{I}_h^H U_h$
3. Calculate residual on the coarse mesh, $R_H(U_H^h)$
4. Prolong residual to the fine mesh, $R_h^H = \mathcal{I}_H^h R_H(U_H^h)$
5. Calculate the truncation error (Note: $R_h(U_h)$ is a remaining fine grid residual, at convergence $R_h(U_h) = 0$).

Truncation error estimation with multi-grid - procedure

Fine mesh (h)

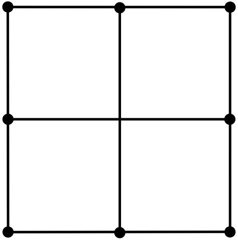


$$U_h \xrightarrow{\mathcal{I}_h^H} U_H^h$$

$$R_h^H \xleftarrow{\mathcal{I}_H^h} R_H(U_H^h)$$

$$\delta R_h = R_h(U_h) - R_h^H$$

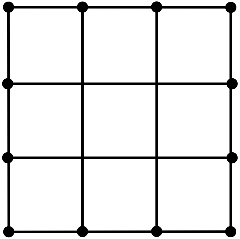
Coarse mesh (H)



1. Solve primal, $R_h(U_h) = 0$
2. Restrict primal solution to the coarse mesh, $U_H^h = \mathcal{I}_h^H U_h$
3. Calculate residual on the coarse mesh, $R_H(U_H^h)$
4. Prolong residual to the fine mesh, $R_h^H = \mathcal{I}_H^h R_H(U_H^h)$
5. Calculate the truncation error (Note: $R_h(U_h)$ is a remaining fine grid residual, at convergence $R_h(U_h) = 0$).

Truncation error estimation with multi-grid - procedure

Fine mesh (h)

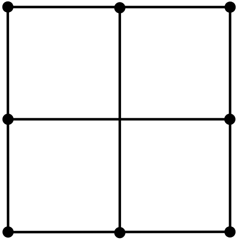


$$U_h \xrightarrow{\mathcal{I}_h^H} U_H^h$$

$$R_h^H \xleftarrow{\mathcal{I}_H^h} R_H(U_H^h)$$

$$\delta R_h = R_h(U_h) - R_h^H$$

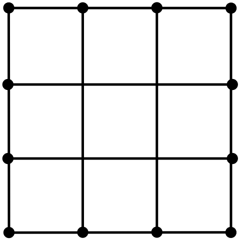
Coarse mesh (H)



1. Solve primal, $R_h(U_h) = 0$
2. Restrict primal solution to the coarse mesh, $U_H^h = \mathcal{I}_h^H U_h$
3. Calculate residual on the coarse mesh, $R_H(U_H^h)$
4. Prolong residual to the fine mesh, $R_h^H = \mathcal{I}_H^h R_H(U_H^h)$
5. Calculate the truncation error (Note: $R_h(U_h)$ is a remaining fine grid residual, at convergence $R_h(U_h) = 0$).

Truncation error estimation with multi-grid - procedure

Fine mesh (h)

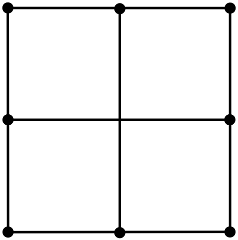


$$U_h \xrightarrow{\mathcal{I}_h^H} U_H^h$$

$$R_h^H \xleftarrow{\mathcal{I}_H^h} R_H(U_H^h)$$

$$\delta R_h = R_h(U_h) - R_h^H$$

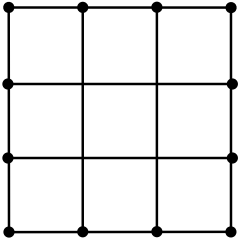
Coarse mesh (H)



1. Solve primal, $R_h(U_h) = 0$
2. Restrict primal solution to the coarse mesh, $U_H^h = \mathcal{I}_h^H U_h$
3. Calculate residual on the coarse mesh, $R_H(U_H^h)$
4. Prolong residual to the fine mesh, $R_h^H = \mathcal{I}_H^h R_H(U_H^h)$
5. Calculate the truncation error (Note: $R_h(U_h)$ is a remaining fine grid residual, at convergence $R_h(U_h) = 0$).

Truncation error estimation with multi-grid - procedure

Fine mesh (h)

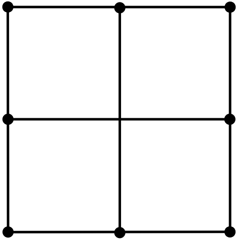


$$U_h \xrightarrow{\mathcal{I}_h^H} U_H^h$$

$$R_h^H \xleftarrow{\mathcal{I}_H^h} R_H(U_H^h)$$

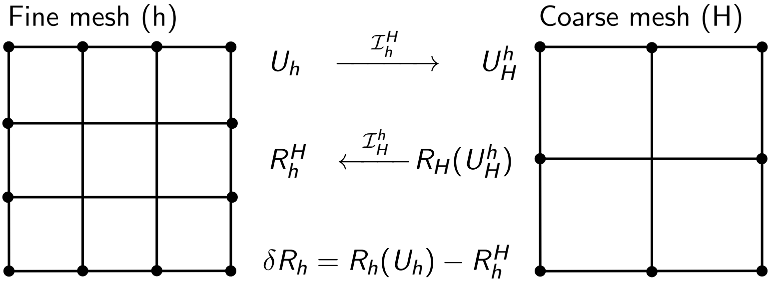
$$\delta R_h = R_h(U_h) - R_h^H$$

Coarse mesh (H)



1. Solve primal, $R_h(U_h) = 0$
2. Restrict primal solution to the coarse mesh, $U_H^h = \mathcal{I}_h^H U_h$
3. Calculate residual on the coarse mesh, $R_H(U_H^h)$
4. Prolong residual to the fine mesh, $R_h^H = \mathcal{I}_H^h R_H(U_H^h)$
5. Calculate the truncation error (Note: $R_h(U_h)$ is a remaining fine grid residual, at convergence $R_h(U_h) = 0$).

Truncation error estimation with multi-grid - procedure



1. Solve primal, $R_h(U_h) = 0$
2. Restrict primal solution to the coarse mesh, $U_H^h = \mathcal{I}_h^H U_h$
3. Calculate residual on the coarse mesh, $R_H(U_H^h)$
4. Prolong residual to the fine mesh, $R_h^H = \mathcal{I}_H^h R_H(U_H^h)$
5. Calculate the truncation error (Note: $R_h(U_h)$ is a remaining fine grid residual, at convergence $R_h(U_h) = 0$).

Adaptation sensor/indicator

Output error estimation:

$$\delta L_h \approx v_h^T \Big|_{U_h} \delta R_h \tag{1}$$

Solve adjoint on fine mesh:

$$\frac{\partial R}{\partial U}^T \Bigg|_{U_h} v_h = \frac{\partial L}{\partial U}^T \Bigg|_{U_h} \tag{2}$$

Adaptation indicator (adjoint-weighted truncation error):

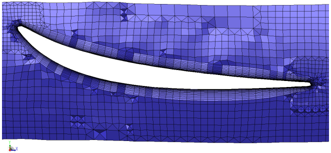
$$OS_h = \left| \sum_{i_{eqn}=1}^{N_{eqn}} v_{i_{eqn},h}^T |_{U_h} \delta R_{i_{eqn},h} \right| \tag{3}$$

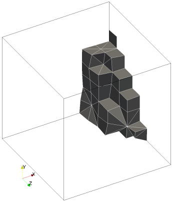
Re-meshing

Basic information

- BoxerMesh[2] [7] with volume refinement functionality is used for re-meshing. The mesher uses octree cut-cell algorithm to create an initial mesh which is then then fitted to the geometry. In the last step a boundary layer is extruded.
- Volume refinement regions are obtained using Paraview. A region is created by group of cells with an adaptation sensor above a specified threshold

Typical Boxer mesh - blade section





Region of cube domain marked for refinement

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

Procedure

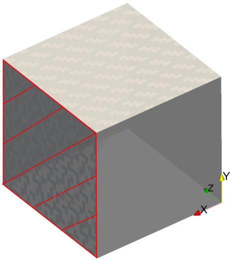
The procedure for single re-meshing step is as follows:

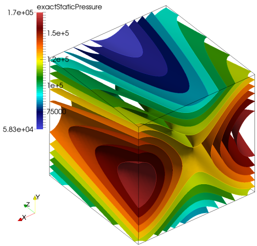
1. Obtain flow solution (U_h).
2. Estimate truncation error (δR_h).
3. Obtain adjoint solution (ν_h).
4. Evaluate output-based sensor (OS_h).
5. Perform 5 explicit smoothing iterations on obtained sensor (OS_h) to damp unwanted high-frequency modes.
6. Use Paraview to extract mesh region for refinement.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

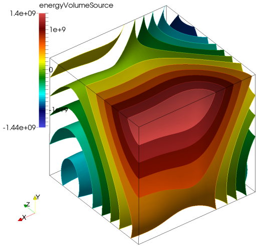
Results

Cube test case

- The simple 3D case with a complex manufactured solution is used for testing - see figures below.
- Compressible, supersonic Euler flow by Roy [8] is used - mix of sine and cosine functions.
- Objective function (L): drag evaluated on the surface marked red on the figure below.



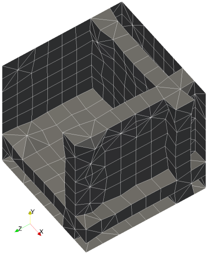




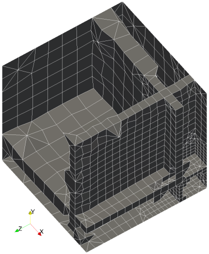
Mesh changes

- Initial mesh: 754 nodes, mixed cell type.
- Two re-meshing steps are performed using an output-based sensor as described in procedure from previous section.
- A non-intuitive and complex refinement structure is obtained.

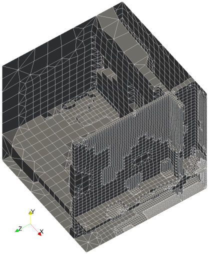
Initial mesh



Re-meshed, step 1



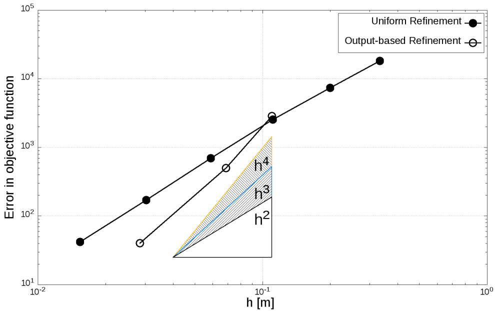
Re-meshed, step 2



Note: coarse grid used for truncation error estimation is not topologically consistent with fine mesh.

Error reduction

- The graph compares drag error reduction wrt characteristic mesh size (h) for uniformly refined vs re-meshed grid.
- Error for a uniformly refined mesh converges with a 2^{nd}O slope - as expected for 2^{nd}O discretisation, whereas for a refined grid the error converges with more than an order of magnitude higher slope i.e. between h^3 and h^4.



Quantitative comparison

Almost and order of magnitude mesh size reduction is obtained for re-meshed grid without affecting cost function accuracy as compared to a uniformly refined case.

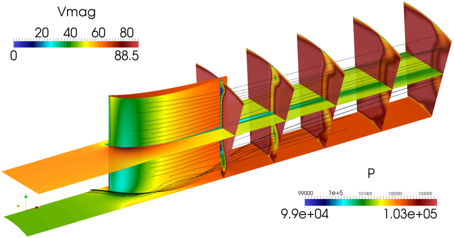
Step	$\delta L/\widetilde{L}$ [%]	N_{DoF}^{OS}	N_{DoF}^{U}	N_{DoF}^{OS} / N_{DoF}^{U}
0	2.11	754	660	0.87
1	0.37	3082	12100	3.9
2	0.03	43349	335000	7.7

Table: 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.

Note: Uniform refinement was performed for regular all hex mesh, whereas the adaptation was performed on the mixed cell grid type.

TurboLab Stator[3] Case from TU Berlin

- The boundary conditions are 42 degrees of swirl angle at the inlet, and outlet static pressure adjusted to keep the mass flow rate at 9.0 kg/s.
- Objective function: pressure loss weighted by mass flow.
- The figure shows static pressure contours on the hub and blade whereas velocity profiles are presented in axial and radial sections. A very strong horseshoe vortex between hub and blade is visualised using streamlines.

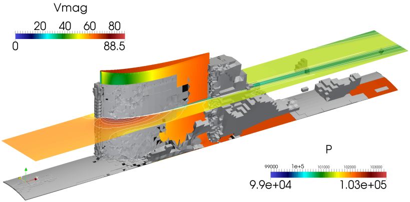


[3] http://aboutflow.sems.qmul.ac.uk/events/munich2016/benchmark/testcase3/

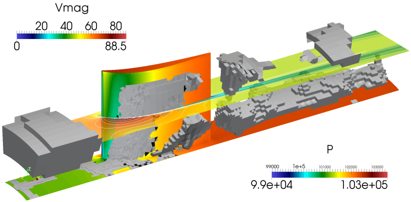
TU Stator

- Comparison of truncation-error-based and output-error-based sensors is presented on the figures (iso-volume).
- Truncation sensor targets for refinement mainly regions of the domain near leading and trailing edges.
- Output sensor marks large cells at the inlet as well as regions of the domain where the strong horseshoe vortex is generated.

Truncation-based sensor



Output-based sensor



Summary

Conclusion / Future work

- The truncation error estimation methodology using topologically inconsistent fine and coarse meshes was presented. Estimated truncation error was weighted with an adjoint solution to obtain a robust output-based adaptation sensor.
- Re-meshing using Boxer and output-based sensor field was successfully applied to the simple cube test case showing almost an order of magnitude cost function error reduction as compared to the uniformly refined grid.
- More application examples are required including a more realistic turbulent cases e.g. turbine stator in order to investigate how useful is the methodology in practice.
- The key challenge for viscous flows is related to the treatment of boundary layer when performing re-meshing with Boxer.

References

Venditti, D. A. and Darmofal, D. L.
A multilevel error estimation and grid adaptive strategy for improving the accuracy of integral outputs.
AIAA-CP 99-3292 **0** (1999).
AIAA 14th CFD Conference, Norfolk, Va.

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).

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

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).

Fraysse, F., Rubio, G., and Valero, E.
Mesh adaptation driven by truncation error estimates.
0 (2011).

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).

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).

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).

Industrial partners:

University partners:

National Technical University of Athens

Acknowledgement

The authors gratefully acknowledge the permission of Rolls-Royce Deutschland to publish this paper. This project has received funding from the European Union's Seventh Framework Programme for research, technological development and demonstration under grant agreement no [317006].

Research funded by the European Commission