Multiblock Structured Mesh Generation for Turbomachinery Flows

Zaib Ali and Paul G. Tucker

Whittle Laboratory, Department of Engineering, University of Cambridge,
1 JJ Thomson Ave, Cambridge, CB3 0DY, England
{za234,pgt23}@cam.ac.uk

Abstract. Multiblock structured meshes offer better computational efficiency than the unstructured meshes which, on the other hand, are more flexible for complex geometries. The multiblock structured meshing could be more useful if the partitioning of the domain can be performed automatically. In this paper, we consider various automated blocking approaches for some turbomachinery zones. We then assess these methods by employing adjoint based error estimation which shows that medial axis based methods perform better than the other approaches like Cartesian fitting. A hybrid approach is also demonstrated for the cases where the existing automatic blocking techniques might not be useful. New blocking templates can also be generated using the techniques applied in the work presented here.

1 Introduction

1.1 Quad/Hexahedral Mesh Generation

Good quality mesh is critical to the quality and convergence of the flow simulations. There is a trade-off between the mesh quality, ease of generation, solver requirements and parallel mesh generation when choosing among the structured and unstructured mesh types. Structured meshes offer higher numerical accuracy, less cell count than the unstructured meshes and also present relatively easier implementation of higher order numerical schemes. Unstructured meshes on the other hand offer more flexibility for meshing complex 3D domains. Various approaches to quad and hexahedral mesh generation can be found in the literature, encompassing both structured and unstructured methods. These include mapping and multi-blocking (5), sweeping (1), plastering (paving in 2D) (2), whisker weaving (23), overset mesh methods (21; 4), quadtree and octree methods (20) and approaches like Q-Morph and H-Morph (14). The virtues and drawbacks of these methods are discussed elsewhere, see for example, (9; 13).

J. Sarrate & M. Staten (eds.), *Proceedings of the 22nd International Meshing Roundtable*,
DOI: 10.1007/978-3-319-02335-9_10, © Springer International Publishing Switzerland 2013

Multiblock structured mesh generation is among the most widely used meshing techniques. Multiblocking is essentially a two-stage process. In the first stage, a suitable blocking topology is generated which divides the complex domain into simple sub-domains. The resulting blocks are then meshed by various mapping methods. While this offers an efficient way to mesh the complex domains, blocking the domain itself is very time consuming job and requires considerable human intervention to produce the mesh. For relatively common and simple domains standard blocking layouts like H, O and C topologies are used but with the increase in geometric complexity, more manual effort is required. To overcome this bottleneck, various strategies to automate the blocking process have also been put forward.

Approaches towards automatic blocking are presented, for example, in (10) and (19). The medial axis transform (MAT) based algorithms also offer an efficient approach to the automatic blocking. Medial axis, first proposed by Blum (3), can be defined as the set of lines and curves (surfaces in 3D) produced from the midpoint of a maximal inscribed disc (sphere in 3D) as it rolls around the domain. Methods like those presented by Armstrong and collaborators (22; 16; 15) use the constrained Delaunay triangulation of points distributed on the boundary to get the MAT. An alternative has been presented by Rigby (17), called the 'TopMaker' approach, which makes use of medial vertices and parts of medial axis to block the domain. Additional rules are then defined to produce a good quality mesh. Distance field based methods provide an efficient and robust alternative to pure geometric based MAT approaches as extension to 3D is simpler using differential equations than pure geometric approaches. One such method called d-MAT approach is presented in (27). The hyperbolic-natured Eikonal, level set, equation is used to calculate the distance field (25). Medial axis point clouds are then extracted from the Laplacian or Hessian determinant of the distance field. A thinning algorithm is the used for thinning the point clouds into curves and surfaces. Such a hybrid approach thus avoids complexity of pure geometric approach and provides more accuracy than the pure image thinning algorithms. A further enhancement to this approach is a biased MAT approach proposed in (28) which adds more flexibility and thus improves the quality of the resulting mesh.

Malcevic (11) presents another automated blocking strategy based on a Cartesian fitting method. While preserving the geometric topology definition, a forward geometry simplification is performed followed by fitting the model into a Cartesian framework. The next step is blocking the domain after which the blocked model is mapped back on to the original geometry. Further operations such as removing singularities by J-grid wrapping are performed to enhance the mesh quality. This technique has been applied for meshing the end-wall cavities found in turbomachinery. This technique is simple and but has only been demonstrated for 2D cases so far. The method sometimes produces some unnecessary mesh clustering across the block interfaces.

1.2 Adjoint Error Estimation

As compared to the feature based error estimators (18) (which generally focus the local residual errors), the adjoint based error indicators (6; 8; 26) estimate the global error made in predicting the integral quantity of interest also called the output functional. Examples of such functionals for turbomachinery applications include the mass flow through a compressor blade row, total heat flux and pressure loss across a high pressure turbine blade, average noise level of the aircraft near the population etc. This methodology which is a *a posteriori* error estimation identifies the regions which are more sensitive to the output functional. This error estimate can derive the mesh refinement or can be indicator for improving the mesh quality (for example by smoothing, altering block structure) in the areas where the error is large. This will help generating specially tuned grids for accurate output functional calculation.

The work presented here applies the d-MAT, TopMaker and the Cartesian fitting methods approaches reviewed above along with manually generated block topologies to some turbomachinery geometries. We then apply the adjoint error estimation procedure to get a comparative view of the meshes of similar cell count produced through various multiblock topologies. A hybrid approach combining the distance field contours and the Cartesian fitting for mesh generation is also demonstrated.

2 Adjoint Error Indicators

2.1 Discrete Adjoint Analysis

After the primal flow solution is available, the discrete adjoint equations are solved to get the adjoint variables. This section presents the discrete adjoint analysis. A detailed derivation can be found in (8). Let Q, the flow variables at discrete set of points with coordinates X, be the solution of system of steady non-linear equations

$$R(Q, X, \alpha) = 0 \tag{1}$$

where R is the discrete residual vector and α is a set of design variables. Also consider an objective function $J(Q, \alpha)$ which one wishes to optimize. The sensitivity of this objective function to a set of design variables can be expressed in the following form

$$\frac{dJ}{d\alpha} = \frac{\partial J}{\partial Q}\frac{\partial Q}{\partial \alpha} + \frac{\partial J}{\partial \alpha} \tag{2}$$

The adjoint variables ν can be defined as the effect of the flow residual on the objective function:

$$\nu = \frac{\partial J}{\partial R} \tag{3}$$

Using Equations (1) and (3), Equation (2) can be written as

$$\frac{dJ}{d\alpha} = -\nu^T \frac{\partial R}{\partial \alpha} + \frac{\partial J}{\partial \alpha} \tag{4}$$

Equation (4) can be re-written to give the following equation for adjoint variables

$$\left(\frac{\partial R}{\partial Q}\right)^T \nu = \left(\frac{\partial J}{\partial Q}\right)^T \tag{5}$$

This set of linear equation can be solved to give the adjoint flow variables in a manner similar to the primary flow solution.

2.2 Adjoint Error Analysis

We apply the adjoint error estimation procedure of Venditti and Darmofal (26) in this work. A detailed description of this method is presented in the Appendix. The global error in the objective function $J(Q)$ can be related to the local residual error with the adjoint variables working as the weighting function (see Appendix). This relation is described by the equation

$$J\left(Q\right) - J_h\left(Q_h^H\right) = -(\nu_h^H)^T R_h\left(Q_h^H\right) \tag{6}$$

2.3 Flow and Adjoint Solvers

The primal flow solver used in this work is a node-based, second order accurate, and unstructured Reynolds-Averaged Navier-Stokes (RANS) code using the finite volume method (12). It is fully parallel and uses multigrid solution with pseudo time marching accelerated by Jacobi preconditioning. The steady adjoint solver is based upon the discrete adjoint approach (7). The gradients are evaluated using the automatic differentiation. First the non-linear primal flow version is linearized and then the adjoint version is obtained by transposing the linearized equations.

3 Results

In this section the automatic blocking methods are applied to various 2D turbomachinery geometries. This includes two turbine seal cavities and an engine intake geometry with cross winds in a tunnel. In addition to the automated blocking methods, the results from the manually created block topologies are also included for the sake of comparison. These hand-crafted block topologies were generated by the members within the research group. The objective function chosen for all the cases is the total pressure loss (as percentage of total inlet pressure). We then apply the adjoint error estimation procedure to these domains. The error is computed with respect to a fine mesh which is two

step uniform refinement of the original coarse mesh. A hybrid approach using the wall distance field and the Cartesian fitting method is also demonstrated for the 2D jet-wing-flap and tandem airfoils cases.

In the implementation of the d-MAT approach (27), curve fitting is used to generate the medial axis from the point cloud instead of the pixel thinning. In addition to that, three additional rules described in (24) have been used. The TopMaker (17) and the Cartesian fitting (11) techniques have been manually applied for these cases with the aim to automate the procedure in the future.

3.1 Endwall Cavities

Labyrinth Seal

The first geometry considered is a labyrinth seal found in turbine endwalls to avoid the flow leakage from the main gas flow path. The geometry, block topologies and the meshes generated by various methods are shown in the Figure 1. The flow direction is from left to right. The d-MAT approach produces blocks resulting in a mesh which is aligned with the flow direction specially around the seal teeth. The TopMaker blocking is similar to the d-MAT but it produces a number of small blocks around the round sections. The Cartesian fitting mesh has more orthogonality than the other two approaches. However the effect of cell clustering to resolve the boundary layer spreads out into the main flow path in order to retain the mesh quality across the block boundaries. Interestingly, the hand-crafted blocking looks similar to the one generated by the Cartesian fitting method.

The axial velocity contours of the primal flow and its adjoint counterpart are shown the Figure 2. The flow is modelled with an axial Reynolds number of 10,000. Regions of flow acceleration at the seal tooth tips and flow recirculation at the bottom of the steps are generated in the primal flow field. The adjoint flow field shows the sensitivity of the objective function to the residuals in the axial velocity. One important aspect of the adjoint flow field is that it propagates information backwards and hence the flow direction is opposite to that of the primal flow. Seen in the context of design optimization, the positive values of adjoint variables indicate that the geometry change here should be such that it increases the fluxes subjected to the design constraints and vice versa.

The error made in computing the objective function (total pressure loss in this case) is estimated using the method described in the Section 2. The overall error can then be obtained by summing individual error contributions weighted by the square of cell areas. If E_j is the estimated error over a cell of area A_j, Then the total error TE over the whole domain for N number of cells is given by

$$TE = \sum_{j=0}^{N} |E_j| A_j^2 \tag{7}$$

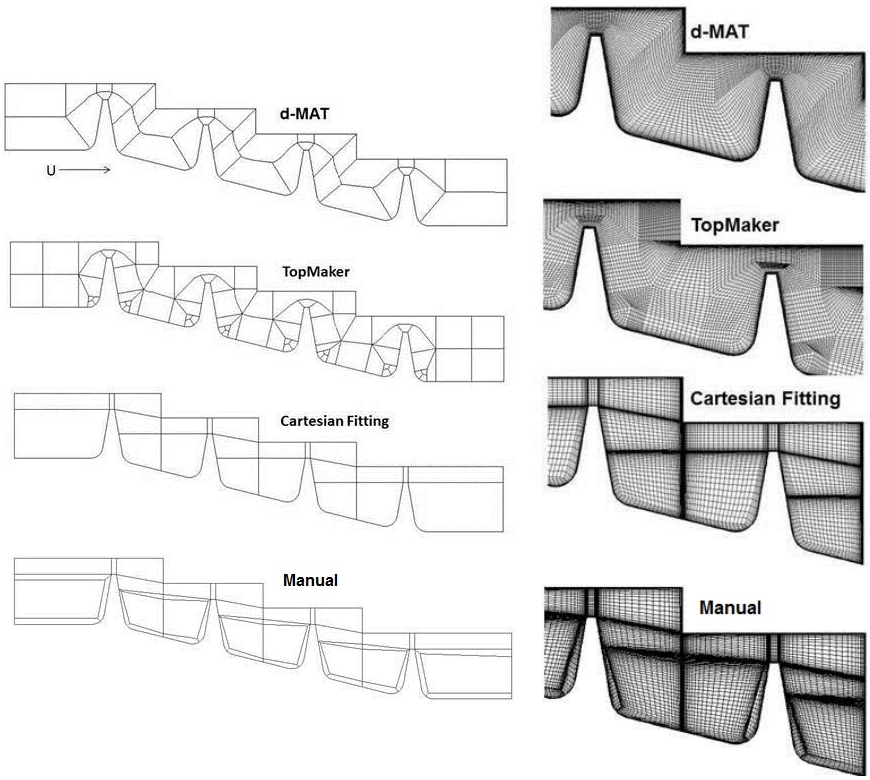

Fig. 1 The block topologies and resulting meshes for labyrinth seal using various approaches

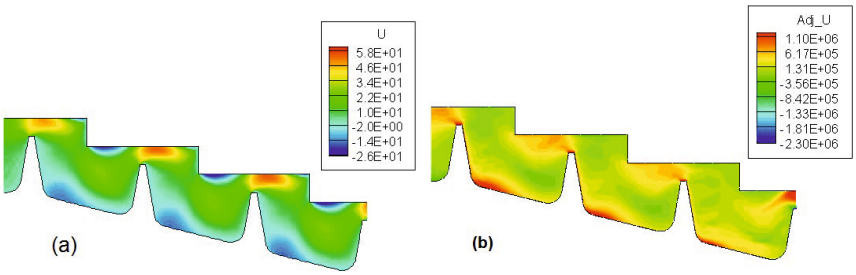

Fig. 2 The contours of (a) axial velocity (b) adjoint counterpart of the axial velocity

A comparison of the error estimates can then be made by normalizing the total error with the d-MAT error TE_{dMAT} such that the normalized value TE_N is given by

$$TE_N = TE/TE_{dMAT} \tag{8}$$

Table 1 Labyrinth seal: Coarse mesh size and the total normalized error

Blocking Type	N	TE_N
d-MAT	59,720	1
TopMaker	59,078	1.13
Cartesian fitting	65,620	1.60
Manual blocking	67,104	1.27

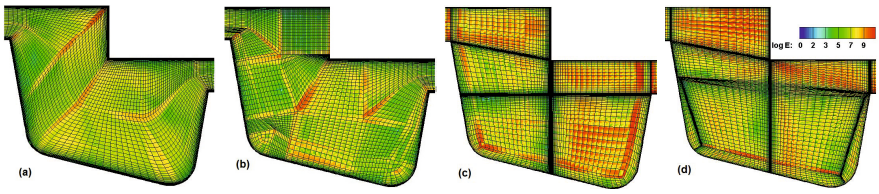

Fig. 3 The error maps for the 4th step of the seal: (a) d-MAT (b) TopMaker (c) Cartesian fitting (d) Manual

Table 1 shows the coarse mesh cell count N and the normalized error estimates TE_N for the four block topologies. The results indicate that medial axis based meshes provide more accurate value of the objective function than the Cartesian fitting and the manual block topologies. This is because, for the similar cell count, the medial axis based approaches generate the meshes which have a more uniform cell size distribution in the main flow path than the Cartesian fitting approach and have better alignment with the flow. Figure 3 highlights this point where the enlarged view of fourth seal step is shown. The Cartesian fitting mesh is clustered in the main flow path in order to keep the mesh sizes consistent between the blocks with boundary layer refinement and their neighbours. This constraint effects the overall uniformity of the mesh size producing coarser mesh in the areas away from the block boundaries, thus contributing to the error. Similar trend can be seen in the manually generated mesh. The alignment of mesh with the flow direction is demonstrated the Figure 4 where the streamlines are drawn against the various meshes for two zones of the seal step. As can be seen, the medial axis based meshes provide better alignment with the flow. In addition to the mesh density, the error indicator also marks the areas in meshes where the mesh quality (for example smoothing and skewness) needs to be improved.

Rim Seal

In this section we consider another seal cavity. The rim seal geometry, the block topologies and the meshes generated by applying the d-MAT, the Cartesian fitting and the hand-crafted blocking are shown in the Figure 5. The

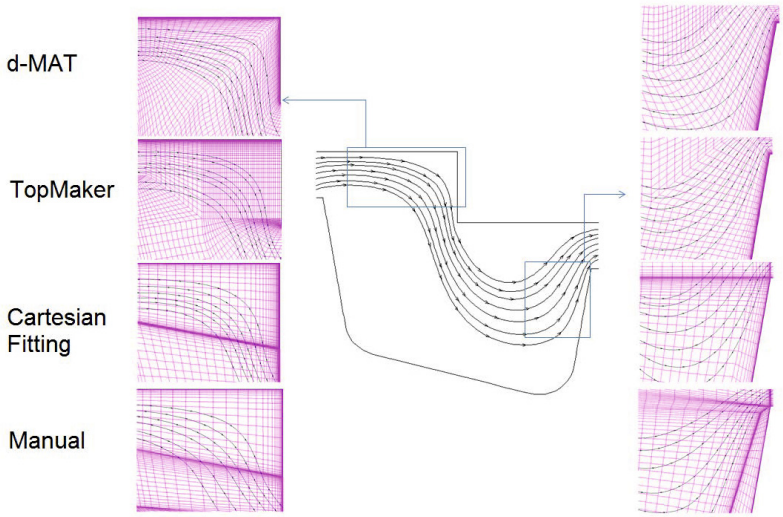

Fig. 4 Streamlines against the mesh for two zones in the fourth step of labyrinth seal

manual blocking in this case produces more uniform mesh and less number of blocks than the rest of the methods. The contours of the flow velocity and and its adjoint counterpart are shown in the Figure 6.

Table 2 compares the coarse mesh cell count N and the normalized error estimates TE_N for various approaches. As depicted by the numbers in the table, the manual and the d-MAT meshes produce more accurate functional estimates than the Cartesian fitting methods. Figures 7 and 8 shows the error maps and the mesh alignment with the flow for various approaches. Manual generated mesh performs well here due it smoothness and better flow alignment. The d-MAT mesh is more skewed and less aligned with the flow as compared to the manual mesh. The Cartesian meshes struggles to achieve uniform cell size distribution and the alignment with the flow thus resulting in the higher error.

Table 2 Rim seal: Coarse mesh size and the total normalized error

Blocking Type	N	TE_N
d-MAT	4,872	1
Cartesian fitting	4,461	2.69
Manual blocking	4,934	0.70

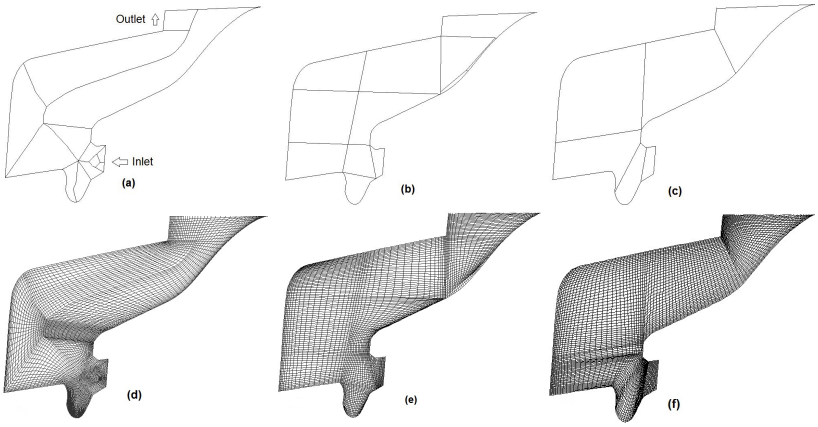

Fig. 5 Rim seal (a) d-MAT blocking (b) Cartesian fitting blocking (c) Manual blocking (d) d-MAT mesh (e) Cartesian fitting mesh (f) Manual mesh

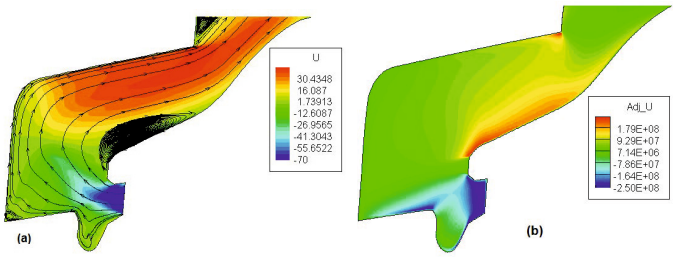

Fig. 6 Rim seal: Contours of (a) axial velocity (b) adjoint counterpart of axial velocity

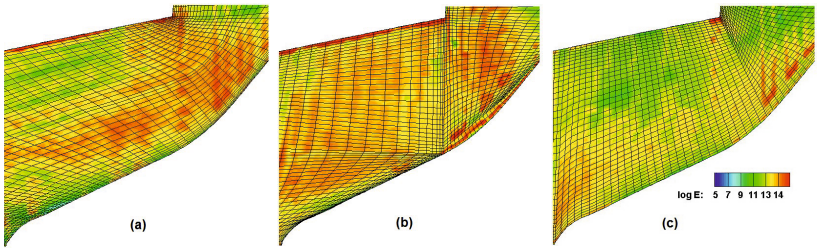

Fig. 7 The error maps for the rim seal: (a) d-MAT (b) Cartesian fitting (c) Manual

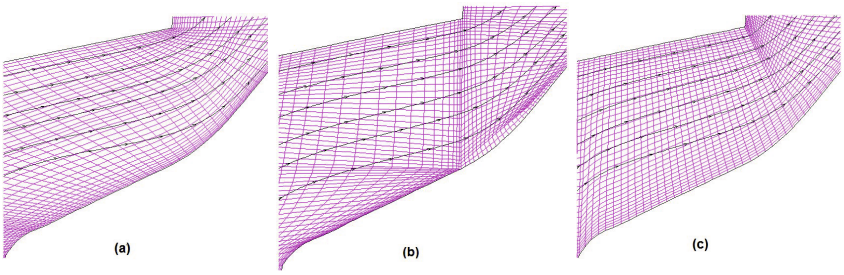

Fig. 8 Streamlines against the mesh: enlarged view for rimseal: (a) d-MAT (b) Cartesian fitting, (c) Manual

3.2 Engine Intake

The third geometry considered here is an engine intake in a tunnel with cross winds. The geometry, block topologies and the meshes using the d-MAT, Cartesian fitting and the manual blocking are shown in the Figure 9. The d-MAT mesh wraps around the intake lip geometry nicely while the Cartesian fitting mesh retains its typical nature along with mesh clustering spreading out away from the boundary layer. The flow and adjoint flow field for the axial velocity are shown in the Figure 10.

Table 3 compares the cell count and the normalized estimated error for the various methods. The d-MAT approach outperforms the other approaches due to same reasons described for the labyrinth seal case. The Cartesian fitting approach achieves the similar accuracy when the cell count is increased by approximately 40% for this case. This difference in cell count for the same accuracy becomes significant for complex 3D geometries and where LES and DNS type meshes are desired. The estimated error contours around the intake lip are shown in the Figure 11.

It can be concluded from the three cases considered in this section that the medial axis based approaches perform better than the Cartesian fitting approach due to more uniform cell size distribution and better flow alignment. We have also observed that the manually generated block topologies yield better results in some cases while not performing well in the others. Seen from this perspective, the automated approaches can always guide a CFD practitioner to optimally block the domain in addition to being less time consuming. Thus blocking templates can be generated for different parts of, for example, an aeroengine. The other factors which might influence the choice of a specific blocking approach are the objective function, ease of blocking especially in 3D and the number of blocks it produces for running simulations on parallel machines.

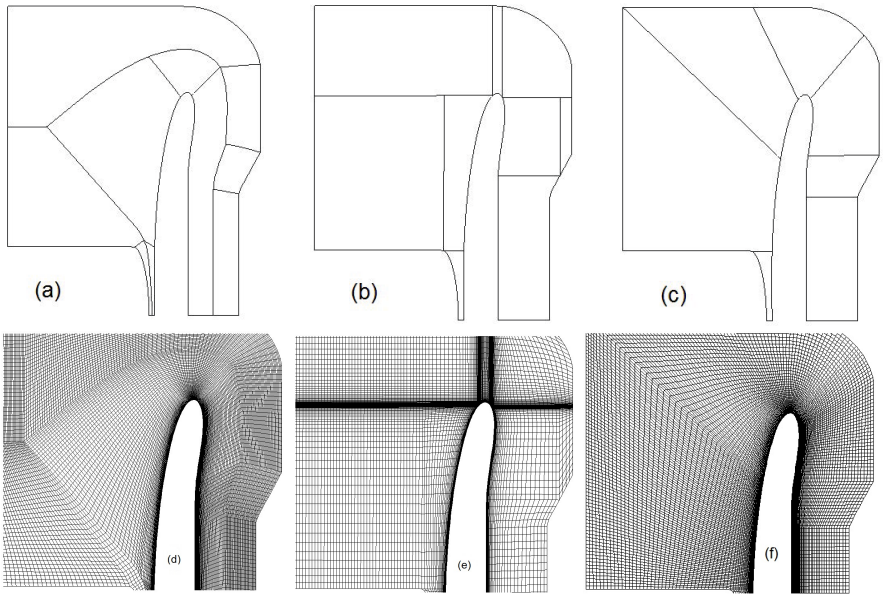

Fig. 9 Engine Intake geometry (a) d-MAT blocking (b) d-MAT mesh (c) Cartesian fitting blocking (d) Cartesian fitting mesh (e) Manual blocking (f) Manual mesh

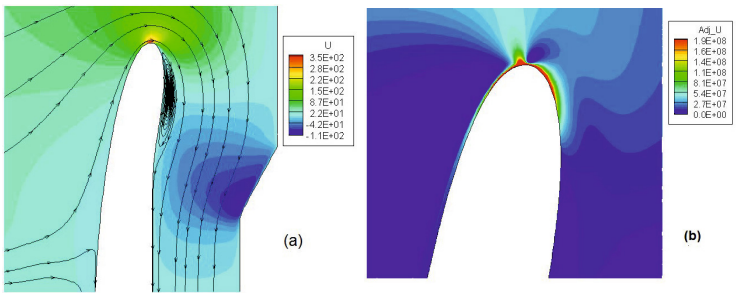

Fig. 10 Engine Intake: Contours of (a) axial velocity (b) adjoint counterpart of the axial velocity around the intake lip

Table 3 Engine Intake: Coarse mesh size and the total normalized error

Blocking Type	N	TE_N
d-MAT	26,564	1
Cartesian fitting	27,536	3.69
Cartesian fitting	37,442	0.96
Manual blocking	26,712	1.28

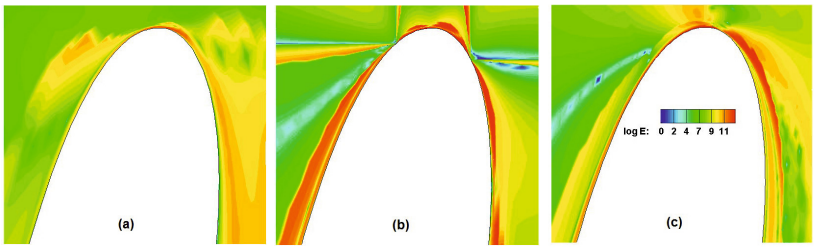

Fig. 11 The error contours near the intake lip: (a) d-MAT (b) Cartesian fitting (c) Manual

3.3 Hybrid d-Cartesian Fitting Blocking

The three blocking approaches demonstrated in the previous sections may not always provide a valid block topology. In that case some new blocking rules or hybrid approaches might be desired. An example for such an occurrence is shown in the Figure 12(a) which is a 2D jet-wing-flap geometry. Figure 12(a) also shows the medial axis generated with the d-MAT approach which does not produce a valid blocking topology.

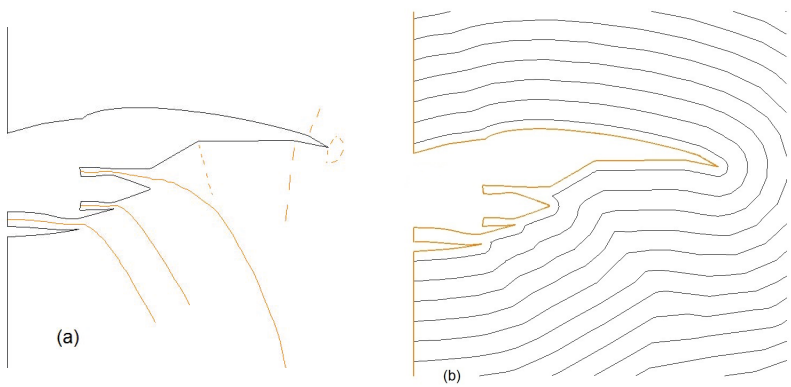

Fig. 12 2D Jet-wing-flap (a) geometry with d-MAT blocking (b) distance field contours

The distance field contours around the jet-wing-flap shown in 12(b) might be used in this case for blocking. Currently the distance field contour selection is arbitrary with the aim to automate the procedure in the future. The area between the jet-wing-flap boundary and the distance field wrap can be meshed using the d-MAT approach while the Cartesian fitting method is applied in the far field. This hybrid approach produces a good quality block topology. The resulting blocking and the mesh are shown in the Figure 13.

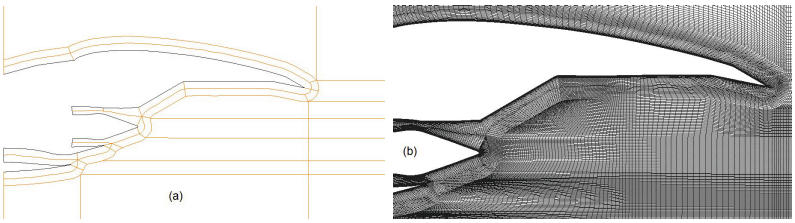

Fig. 13 Jet-wing-flap: (a) Blocking using distance field contour and the Cartesian fitting for tandem airfoils (b) resulting mesh

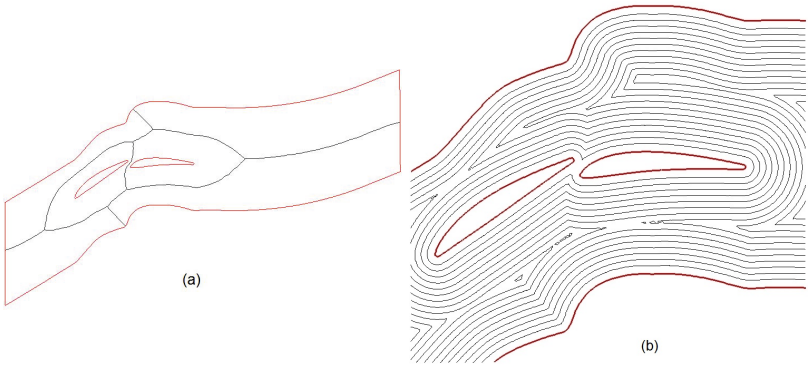

Fig. 14 Tandem airfoils: (a) geometry with d-MAT blocking (b) distance field contours

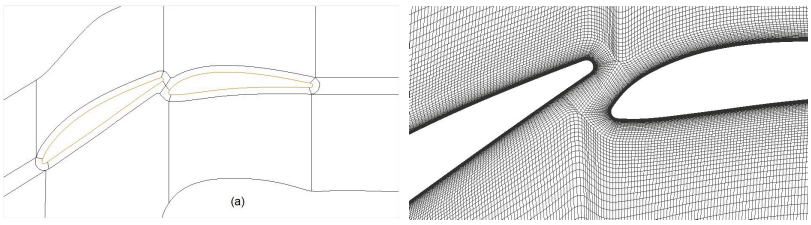

Fig. 15 Tandem airfoils: (a) Blocking using distance field contour and the Cartesian fitting for tandem airfoils (b) resulting mesh

The second example of the hybrid approach is 2D flow domain around tandem airfoils shown in 14(a). With the d-MAT blocking shown, it is difficult to mesh the region in between and around the airfoil. Again, the distance field contours around the airfoils might be used to form an O-type block around them. The distance field is shown in the Figure 14(b). Once this block is generated around the airfoils, the Cartesian fitting can then be applied in

the far field and in the interior of the O-type block to produce a good quality blocking. The resulting blocking and the mesh around the airfoils are shown in the Figure 15. This hybrid approach can be used to create standard block topology templates.

4 Conclusions

Three automated blocking techniques have been considered for meshing various turbomachinery geometries. Adjoint based error analysis shows that medial axis based approaches yield better output functional estimates due to more uniform mesh size distribution and flow alignment. The error estimation can also be used to improve the mesh both in terms of quality and quantity. A hybrid approach is also demonstrated for the cases where previous techniques might not perform well. Such approaches can also be employed to produce standard templates for various geometric configurations. The future work includes extension of the analysis to three-dimensional cases and further development and automation of the hybrid approach.

Appendix

Adjoint Error Estimation

Consider a computational domain Ω and let Ω_H and Ω_h are the coarse mesh and the fine mesh discretizations of this domain respectively. H and h $(H > h)$ here represent the mesh length scales associated with a particular discretization such as finite difference or finite volume. The coarse mesh has a mesh density that is able to capture the basic feature of the flow but might not yield the desired level of accuracy. The fine mesh on the other hand is a systematic uniform refinement of the coarse mesh which can provide the accuracy required but is computationally expensive. The discretization of the governing equations on the coarse and fine mesh yield the residual vectors which can be denoted by $R_H(Q_H)$ and $R_h(Q_h)$ respectively where Q is the solution of system of the governing flow equations. Let $J(Q)$ be the objective function which one wants to estimate. The estimation of this objective function on the coarse and fine mesh are denoted by $J_H(Q_H)$ and $J_h(Q_h)$.

An estimate of $J_h(Q_h)$ on the fine grid, without solving on the fine grid, can be made by a Taylor's series expansion of $J_h(Q_h)$ about the solution on the coarse grid

$$J_h(Q_h) = J_h\left(Q_h^H\right) + \left.\frac{\partial J_h}{\partial Q_h}\right|_{Q_h^H}\left(Q_h - Q_h^H\right) + ... \tag{9}$$

Also the expansion of $R_h(Q_h)$ about the coarse mesh yields

$$R_h(Q_h) = R_h\left(Q_h^H\right) + \left.\frac{\partial R_h}{\partial Q_h}\right|_{Q_h^H}\left(Q_h - Q_h^H\right) + ... \tag{10}$$

The vector $\left(Q_h^H\right)$ in the above equation is the coarse mesh solution estimated on the fine mesh with a suitably defined prolongation operator. The vector $\left.\frac{\partial J_h}{\partial Q_h}\right|_{Q_h^H}$ in Equation (9) represents the linear sensitivity of the fine mesh function with respect to $\left(Q_h^H\right)$. Moreover, the vector $\left.\frac{\partial R_h}{\partial Q_h}\right|_{Q_h^H}$ in Equation (10) is the fine mesh Jacobian evaluated using the projected coarse mesh solution. $\left(Q_h^H\right)$ can be evaluated by

$$Q_h^H = I_h^H Q_H \tag{11}$$

where I_h^H represents an appropriate prolongation operator. For example the coarse grid solution can be reconstructed on the fine mesh using linear or higher order interpolation. Assuming the well-posedness, the Equation (10) can be inverted. Also knowing that $R_h(Q_h) = 0$ for steady state problem, Equation (10) yields

$$\left(Q_h - Q_h^H\right) \approx -\left\{\left.\frac{\partial R_h}{\partial Q_h}\right|_{Q_h^H}\right\}^{-1} R_h\left(Q_h^H\right) \tag{12}$$

From Equations (12) and (9) we get

$$J_h(Q_h) \approx J_h\left(Q_h^H\right) - (\nu_h|_{Q_h^H})^T R_h\left(Q_h^H\right) \tag{13}$$

where $\nu_h|_{Q_h^H}$ is the discrete adjoint solution vector estimated at the fine mesh using Q_h^H . This adjoint solution vector satisfies the equation

$$\left\{\left.\frac{\partial R_h}{\partial Q_h}\right|_{Q_h^H}\right\}^T \nu_h|_{Q_h^H} = \left\{\left.\frac{\partial J_h}{\partial Q_h}\right|_{Q_h^H}\right\}^T \tag{14}$$

Equation (13) requires the evaluation of the term $(\nu_h|_{Q_h^H})^T$ on the fine grid. To avoid this evaluation, $(\nu_h|_{Q_h^H})^T$ can also be estimated thorough coarse grid adjoint interpolation onto the fine grid via some projection operator

$$\nu_h^H = I_h^H \nu_H \tag{15}$$

where the coarse mesh adjoint solution ν_H is the solution of following adjoint equations on the coarse mesh

$$\left\{\frac{\partial R_H}{\partial Q_H}\right\}^T \nu_H = \left\{\frac{\partial J_H}{\partial Q_H}\right\}^T \tag{16}$$

The final estimate of the objective function $J(Q)$ is then given by

$$J(Q) = J_h\left(Q_h^H\right) - (\nu_h^H)^T R_h\left(Q_h^H\right) \tag{17}$$

The second expression on the right hand side of the above equation is called the error correction term. It can be seen that the error in the objective function is related to the local residual error through the adjoint variables which act as weight function. Equation (13) can be disintegrated into the following form

$$J_h\left(Q_h^H\right) - J_h\left(Q_h\right) \approx \left(\nu_h^H\right)^T R_h\left(Q_h^H\right) + (\nu_h|_{Q_h^H} - \nu_h^H)^T R_h\left(Q_h^H\right) \tag{18}$$

The first term on the right hand side of Equation (18) is the main computable error estimate while the second term is the error in this computable error estimate. This first term on the right hand side shows that the adjoint variables directly relates the error in the given functional to the local residual errors. Stating in another way, the adjoint solution act as a weight function to the local residual error and gives the effect of the residual error on the output functional.

Acknowledgements. The authors would like to thank Dr. James Tyacke for his useful comments and suggestions. The authors are also thankful to the Higher Education Commission, Pakistan for funding this work.

References

[1] Blacker, T.: The Cooper tool. In: 5th International Meshing Roundtable, SAND 95-2130, Sandia National Laboratories. Citeseer (1996)
[2] Blacker, T., Myers, R.J.: Seams and wedges in plastering: A 3D hexahedral mesh generation algorithm. Engineering with Computers 2, 83–93 (1993)
[3] Blum, H.: A transformation for extracting new descriptions of shape. In: Models for the Perception of Speech and Visual Form, pp. 362–380 (1967)
[4] Chesshire, G., Henshaw, W.D.: Composite overlapping meshes for the solution of partial differential equations. Journal of Computational Physics 90(1), 1–64 (1990)
[5] Dannenhoffer, J.: A block-structuring technique for general geometries. In: 29th AIAA Aerospace Sciences Meeting, Reno, Nevada (1991)
[6] Fidkowski, K., Darmofal, D.: Review of output-based error estimation and mesh adaptation in computational fluid dynamics. AIAA Journal 49(4), 673–694 (2011)
[7] Giles, M.B., Duta, M.C., Müller, J.-D., Pierce, N.A.: Algorithm developments for discrete adjoint methods. AIAA Journal 41(2), 198–205 (2003)
[8] Giles, M.B., Pierce, N.A.: Adjoint error correction for integral outputs. In: Error Estimation and Adaptive Discretization Methods in Computational Fluid Dynamics. Lecture Notes in Computational Science and Engineering, vol. 25. Springer (2002)

[9] Henshaw, W.D.: Automatic grid generation. Acta Numerica, 1–66 (2002)
[10] Kolšek, T., Šubelj, M., Duhovnik, J.: Generation of block-structured grids in complex computational domains using templates. Finite Elements in Analysis and Design 39(12), 1139–1154 (2003)
[11] Malcevic, I.: Automated blocking for structured CFD gridding with an application to turbomachinery secondary flows. In: 20th AIAA Computational Fluid Dynamics Conference, Honolulu, Hawaii, AIAA Paper No. AIAA 2011-3049 (2011)
[12] Moinier, P.: Algorithm developments for an unstructured viscous flow solver. PhD thesis, Oxford University (1999)
[13] Owen, S.J.: A survey of unstructured mesh generation technology. In: 7th International Meshing Roundtable, vol. 3 (1998)
[14] Owen, S.J., Saigal, S.: H-morph: An indirect approach to advancing front hex meshing. International Journal for Numerical Methods in Engineering 49(1-2), 289–312 (2000)
[15] Price, M.A., Armstrong, C.G.: Hexahedral mesh generation by medial surface subdivision: Part II. solids with flat and concave edges. International Journal for Numerical Methods in Engineering 40(1), 111–136 (1997)
[16] Price, M.A., Armstrong, C.G., Sabin, M.A.: Hexahedral mesh generation by medial surface subdivision: Part I. solids with convex edges. International Journal for Numerical Methods in Engineering 38(19), 3335–3359 (1995)
[17] Rigby, D.L.: Topmaker: A technique for automatic multi-block topology generation using the medial axis. NASA/CR, 213044 (2004)
[18] Roy, C.J.: Review of discretization error estimators in scientific computing. AIAA Paper, 126: 2010 (2010)
[19] Shaw, J.A., Weatherill, N.P.: Automatic topology generation for multiblock grids. Applied Mathematics and Computation 52(2), 355–388 (1992)
[20] Shephard, M.S., Georges, M.K.: Automatic three-dimensional mesh generation by the finite octree technique. International Journal for Numerical methods in Engineering 32(4), 709–749 (1991)
[21] Steger, J.L., Dougherty, F.C., Benek, J.A.: A Chimera grid scheme. In: Advances in Grid Generation: Presented at Applied Mechanics, Bioengineering, and Fluids Engineering Conference, Houston, Texas, June 20-22, vol. 5, p. 59. American Society of Mechanical Engineers (1983)
[22] Tam, T.K.H., Armstrong, C.G.: 2D finite element mesh generation by medial axis subdivision. Advances in Engineering Software and Workstations 13(5), 313–324 (1991)
[23] Tautges, T.J., Blacker, T., Mitchell, S.A.: The whisker weaving algorithm: A connectivity-based method for constructing all-hexahedral finite element meshes. International Journal for Numerical Methods in Engineering 39(19), 3327–3350 (1996)
[24] Tucker, P.G.: On the unstructured solution of Eikonal and Hamilton-Jacobi equations and mesh generation (unpublished)
[25] Tucker, P.G.: Differential equation-based wall distance computation for DES and RANS. Journal of Computational Physics 190(1), 229–248 (2003)

[26] Venditti, D.A., Darmofal, D.L.: Grid adaptation for functional outputs: Application to two-dimensional inviscid flows. Journal of Computational Physics 176(1), 40–69 (2002)
[27] Xia, H., Tucker, P.G.: Finite volume distance field and its application to medial axis transforms. International Journal for Numerical Methods in Engineering 82(1), 114–134 (2010)
[28] Xia, H., Tucker, P.G.: Fast equal and biased distance fields for medial axis transform with meshing in mind. Applied Mathematical Modelling 35(12), 5804–5819 (2011)