6th European Conference on Computational Mechanics (ECCM 6)
7th European Conference on Computational Fluid Dynamics (ECFD 7)
11-15 June 2018, Glasgow, UK

AN OPTIMIZATION BASED APPROACH TO MULTI-BLOCK STRUCTURED GRID GENERATION

Marcel Sauer

German Aerospace Center (DLR), Institute of Propulsion Technology, Linder Höhe,
51147 Cologne, Germany, E-mail: Marcel.Sauer@dlr.de

Key words: Grid Generation, Optimization, CFD

Abstract.

In this paper, a novel approach suited to the specific requirements of multi-block structured grid generation within design optimizations for turbomachinery components is presented. In contrast to traditional methods like elliptical and algebraic grid generation, the presented approach optimizes grid quality criteria such as cell expansion ratios, inner cell angles, and grid line curvatures. These criteria are evaluated approximately on a B-spline based abstract representation of the multi-block structured grid, which drastically reduces the degrees of freedom. The 2D mathematical formulation of the abstraction layer and the optimization procedure are presented. Automatically generated grids for 2D blade-to-blade slices of a compressor, a turbine and a tandem configuration are shown as examples.

1 INTRODUCTION

Grid generation is a necessary step for all computational fluid dynamics (CFD) calculations. It has a big impact on the convergence speed and quality of the physical results. Especially within a design optimization process (e.g. [1]) grid generation is crucial. The geometries may vary greatly which often leads to failures in the grid generation step or results in poor quality grids leading to non-convergence or doubtful CFD results. Multi-block structured meshes, compared with unstructured grids, need to follow a complex topology and are more difficult to create but provide faster CFD calculations and are more accurate in terms of gradient estimations.

Within design optimizations for turbomachinery components, often a fixed multi-block topology can be used. The presented approach aims to generate high quality grids in a stable manner without further user interaction, while the geometry changes. While elliptical grid generation [2] uses partial differential equations, this approach optimizes different criteria such as stretching ratios, inner cell angles, and curvatures of grid lines.

In order to limit the computational effort, a 2$^{\text{nd}}$ degree B-spline based abstraction layer is introduced. To this end, the B-spline formulation is extended to handle a multi-block structured grid which represents the topology. This abstraction layer is used for the parametrization within the optimization and allows an estimation of the criteria to be optimized.

While this representation drastically reduces the degrees of freedom and allows for a coupled optimization of all parameters, it is, due to the strong convex hull property of B-splines [3], not capable of representing geometric borders in the discretization directly. The remaining deltas are reduced by the usage of an automatic refinement of the control grid and eliminated by a final projection onto the geometry border.

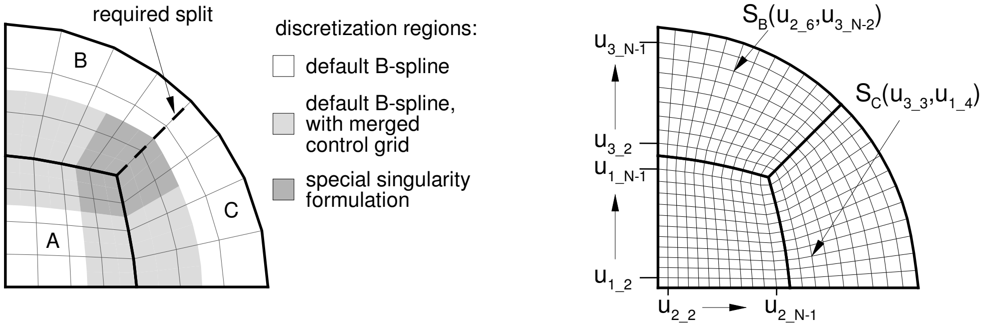
As a prerequisite, a coarse initial mesh representing the desired topology and containing boundary conditions is required. The boundary conditions, defined on nodes of the coarse mesh, are used to include restrictions for nodes to be placed on curves and surfaces or to ensure orthogonality and prescribed distances in the final grid. For each node, the user can prioritize the grid quality criteria arbitrarily and adjust the overall influence.

The mathematical formulation of the method will be presented and the optimization procedure will be described for two grid dimensions. Results of the method's application will be shown for 2D blade-to-blade slices of a compressor, a turbine and a tandem configuration.

2 B-SPLINE BASED ABSTRACTION

In order to reduce the degrees of freedom for the grid optimization problem a B-spline abstraction layer is introduced. A definition of B-spline curves and surfaces with their properties can be found in [3]. The abstraction layer consists of a coarse control grid (see Figure 1a) and a set of surface parameters (u_{*_*}-values in Figure 1b). These surface parameters will be called u-vectors in this paper. Within the discretization step, this abstraction layer is used to generate the full resolution grid. For one block, the generated nodes are surface points for combinations of u-vector entries assigned to the index directions.

The abstraction layer uses the 2nd degree B-spline formulation with equally spaced knot vectors. These B-splines follow its control grid closely and allow to estimate inner cell angles and grid line curvatures directly from it. The resulting grid, where mesh lines



(a) control grid (b) discetization

Figure 1: Abstraction layer and its discretization

follow curves on the surface with one parameter kept constant, will be smoother than the control grid itself. Another advantage is that 2^{nd} degree B-splines need only at least three points in each direction allowing for very coarse control grids.

While the smoothing effect, which can be directly derived from the strong convex hull property of B-splines, has a positive effect on inner cell nodes, it leads to a displacement when representing geometry borders. Here the discretization does only approximate the geometry although the relevant nodes of the control grid are placed onto the geometry.

A standard B-spline is defined with a single control block. In order to represent a block structured topology a merged control grid is used to discretize points near a block-to-block border, but not next to a singularity, where other than four grid lines lead to a node in the interior of the grid. Within the merged control grid, the discretization follows the standard definition where the block border will be represented by one grid line. Near singularities a special treatment is required, which is presented in Section 2.2. The regions for the different treatment within the discretization are visualized in Figure 1a. The terms "near" and "next to" refer to regions influenced by a singularity or an edge connecting two blocks. With the first point being on the edge or a singularity its influence vanishes at the center between the second and third point. Going further into the grid the surface definitions with a single block control grid and a merged control grid coincide with each other.

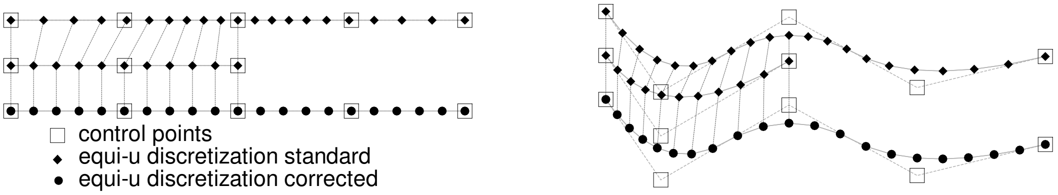
The u-vectors are shared over edges. For example the u_3-vector in Figure 1 is used for the blocks B and C. In order to limit the complexity for the u-vector propagation and for the near singularity discretization the topology is preprocessed in a way, that each edge of a block has either no connectivity or exactly one full connectivity to another block. In Figure 1a a necessary split in is shown by the dashed line that divides one of the initial block into the blocks B and C.

Sharing the u-vectors over edges leads to a linear scaling of u-vector parameters according to the grid resolution while the control grid is independent of the resolution. With a native parametrization each grid point would have its degrees of freedom leading to a quadratic (in 2D) or cubic (in 3D) scaling of the parameters with the grid resolution. For the compressor and the turbine examples shown later less then 1,000 parameters are required for the initial abstract representation. Estimations for a 3D application are less than 6,000 parameters. With the presented resolution the native approach would use about 30,000 parameters in 2D and 3,500,000 parameters in 3D.

2.1 Length correction

The abstraction layer requires a similar behaviour of the B-splines independent of the number of control points. In the case of three control nodes, an equidistant u-distribution in combination with an equally spaced control grid on a straight line leads to an equidistant discretization. But if the number of control nodes is increased, the behaviour changes, leading to a shrinking region near the start, an equidistant region in the center and an expanding region near the end. The different characteristics of the standard B-spline formulation are illustrated in Figure 2 by the top and the middle curve.

To compensate this behaviour equation (2) is used if more than three control points are


□ control points
◆ equi-u discretization standard
● equi-u discretization corrected

(a) Equidistant control grid

(b) Non-equidistant curved control grid

Figure 2: Length correction illustration

present to transfer u-vector entries to the corrected B-spline parameter u_{BS} dependent on the knot vector entries k. The effect of the correction can be seen in Figure 2 comparing the middle and bottom curve.

$$\Delta k = k_2 - k_1 = k_n - k_{n-1} \tag{1}$$

$$u_{BS}(u) = \begin{cases} k_3 - \sqrt{-2 \cdot \Delta k \cdot u + k_3^2 - k_2^2} & \text{for } u < k_2 \\ u & \text{for } k_2 \leq u \leq k_{n-1} \\ k_{n-2} + \sqrt{2 \cdot \Delta k \cdot u + k_{n-2}^2 - k_{n-1}^2} & \text{for } u > k_{n-1} \end{cases} \tag{2}$$

2.2 Discretization near singularities

B-splines are not able to represent singularities. Therefore a special discretization method has been developed to handle the regions near a singularity. A singularity is defined as a corner of a block that has neighbours over its edges but there is no explicit neighbour over the corner. This is visualized in Figure 3, on the left.

The discretization within one block near a singularity (the grey region of block A in Figure 3) is based on different surface definitions which use control grids that take into account neighbouring points of other blocks (B and C). Within the influence range of a singularity, these surface definitions are blended by quadratic functions to construct a smooth interface to the standard B-spline region and achieve a desired grid near a singularity. The surface definitions hereby differ from the length-corrected standard version to further limit the influence of these control points which are more then one index away from the singularity. This is done by double inserting the points of one edge in the control

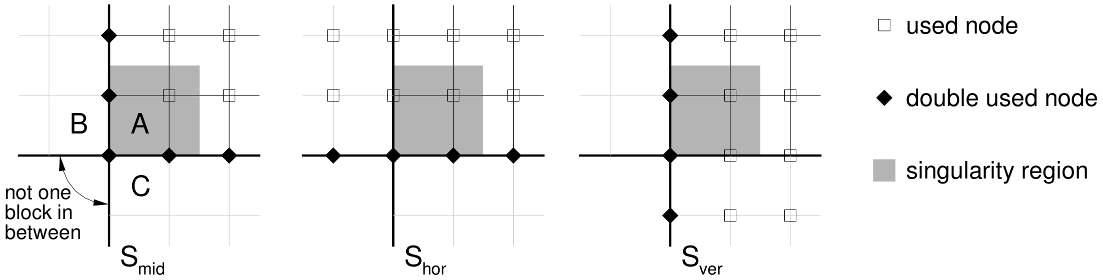


Figure 3: Surface definitions for near singularity discretization

grid and falling back to a 1st degree B-spline definition as soon as the third control point away from the singularity loses its influence.

The surface definitions used are visualized through their control grids in Figure 3. S_{hor} uses horizontal neighbouring points and has a degree loss towards the lower edge. This surface definition fits the standard definition at the upper border of the near singularity range. Accordingly, S_{ver} is constructed by switching the index directions. The surface S_{mid} uses only control nodes of the current block itself and has a degree loss at the left and the lower edge. This formulation represents the desired discretization at the singularity point.

For the following calculations u is defined as the horizontal surface parameter in the interval $[0, 1]$ with 0 at the singularity and 1 at the end of the influence region of the singularity, and v as the vertical parameter accordingly.

In order to create the final discretization, blending operations are applied u-wise, v-wise and based on relative diagonal distance $d(u, v)$. It is defined as the distance from point u, v to the main diagonal relative to the orthogonal intersection point with the definition range.

$$f_d(u,v) = \begin{cases} \frac{1}{2}\left(2-(1-d(u,v))^2\right) & \text{for } v > u \\ 1-\frac{1}{2}\left(2-(1-d(u,v))^2\right) & \text{for } v \le u \end{cases} \tag{3}$$

$$f_{mid}(u,v) = (1-u)^2 \cdot (1-f_d(u,v)) + (1-v)^2 \cdot f_d(u,v) \tag{4}$$

$$f_{hor}(u,v) = (1-(1-v)^2) \cdot f_d(u,v) \tag{5}$$

$$f_{ver}(u,v) = (1-(1-u)^2) \cdot (1-f_d(u,v)) \tag{6}$$

$$S(u,v) = f_{mid}(u,v) \cdot S_{\text{mid}}(u,v) + f_{ver}(u,v) \cdot S_{\text{ver}}(u,v) + f_{hor}(u,v) \cdot S_{\text{hor}}(u,v) \tag{7}$$

The points in the near singularity region of Figure 1b were calculated by equation (7).

3 GRID OPTIMIZATION

3.1 Initial grid and boundary conditions

An initial not folding control grid must be provided as the starting point for the optimization. The initial grids that are used within this work are mostly constructed by placing edge nodes equidistantly along straight lines and applying the transfinite interpolation (see [4]). The u-vectors are initialized equally spaced when possible. To complete the topology, direct and periodic connectivities have to be defined.

For each node, the displacement constraints must be known. In the presented examples the motion is constraint to be fixed, on a curve, circumferential, or on a surface of revolution.

With one control point on the geometry and next point following the geometry normal at the first point, an orthogonality constraint can be prescribed for this index direction.

First distances can be included by restrictions directly. For an approximate distance setup the second control point in the desired direction must keep its distance to the first one. Additionally the value for the second grid line in the corresponding u-vector must be fixed depending on the desired first distance and the distance of the control nodes. The distance of the control nodes should be chosen to be at least twice the desired distance. To enforce the exact distance the next two control nodes must keep their distance and be placed equally spaced along a line. The first distance then can be chosen to be as large as 1.5 times the distance between the first and the second control node.

Other options like prescribing a desired stretching and modifying the overall or a criterion specific influence can be set locally.

3.2 Refinement considerations

As described in Section 2 the abstraction layer is not able to discretize the full resolution grid directly on the geometry. To reduce the discrepancy between the discretization and the geometry it is possible to define refinements for index directions of different blocks. These refinement settings are, as the u-vectors, shared over edges. When using a refinement, the resolution of the control grid will be increased in the given index direction. Until unlocked the newly created intermediate points are defined as a linear combination of neighbouring points. The refinement is not applied near singularities.

When unlocked, the motion constraints for the refinement points are extracted from the neighbouring points and deltas are propagated into the grid. As 2^{nd} degree B-splines follow their control polygon closely the discretization error decreases, allowing for a simple projection algorithm to be used at the final projection.

3.3 Optimization parameters and targets

The optimization is defined by a parameter vector and an objective to be minimized. The parameter vector consists of the degrees of freedom for each point in the control grid and the free values of the u-vectors.

The objective to be minimized defined by equation (8):

$$c_{sqsum} = \sum (w_a \cdot c_a)^2 + \sum (w_c \cdot c_c)^2 + \sum (w_{sa} \cdot c_{sa})^2 + \sum (w_{sd} \cdot c_{sd})^2 \tag{8}$$

The weighting factors w are the product of a general weight of influencing control grid nodes, a criterion specific weight, and a multiplicity factor. The multiplicity factor is used to upscale the contribution to the objective function based on the resolution difference between the control grid and the final grid. The criteria c_a, c_c, c_{sa} and c_{sd} will be defined in the following.

An inner cell angle criterion is calculated by:

$$c_a = \arccos\left(\frac{(P_2 - P_1) \cdot (P_3 - P_2)}{|P_2 - P_1| \cdot |P_2 - P_1|}\right) - \alpha_{\text{target}} \tag{9}$$

Where $P_2 - P_1$ and $P_3 - P_2$ are two edge vectors of a cell within the control grid. The arccos is calculated in degrees and the target angle α defaults to 90 degrees. This value may differ near singularities.

A curvature criterion is defined by:

$$c_c = \frac{\left|\frac{P_2 - P_1}{|P_2 - P_1|} - \frac{P_3 - P_2}{|P_3 - P_2|}\right|}{\min(|P_2 - P_1|, |P_3 - P_2|)} \cdot L_{\text{ref}} \tag{10}$$

The points P_1, P_2, and P_3 are three consecutive nodes of the control mesh following one grid line. The reference length is needed to achieve a scale independence and is set dependent on the input geometry.

The control grid stretching ratio criterion is used to distribute the control grid evenly:

$$c_{sa} = \frac{\max\left(|P_2 - P_1|, |P_3 - P_2|\right)}{\min\left(|P_2 - P_1|, |P_3 - P_2|\right)} - 1 \tag{11}$$

As for curvature errors P_1, P_2, and P_3 are three consecutive nodes of the control grid following one grid line. To ensure a good quality at singularities, the weight of this criterion is set to prioritize the stretching ratio across an edge that leads to a singularity. Additionally, the stretching ratios of the near edge distance to the next distance further into the block is set to shrink the edge near distance to a smaller distance than the inner one. Hereby the refinement is taken into account. This effect can be seen in Figure 4b.

For the stretching ratios occurring in the final grid c_{sd}, a similar formulation as for the control grid streching ratios is used with the option to prescribe the expansion ratio for an index direction. This criterion is calculated on estimated grid curves defined by the usage of 2$^{\text{nd}}$ degree B-spline curves on each grid line in the control grid. These curves take into account the refinement settings and the modifications as discussed in section 2. The points P_1, P_2, and P_3 are defined as curve points by three consecutive values of the corresponding u-vector. As these curves only take into account one line of the control grid there will be a discrepancy between the estimated points and the discretized points in the interior of the grid. This discrepancy has advantages considering geometry normal stretching ratio estimations as they do not include the discretization error of the surface.

3.4 Optimization scheme

The Levenberg-Marquardt algorithm [5], a gradient-based optimization method for minimizing non-linear least squares problems, is used. The parameter vector and the objective is defined as described in Section 3.3. By the usage of the chain rule all partial derivatives of the objective function regarding the parameters are calculated analytically. Within this optimization scheme all parameters are optimized simultaneously. This is required due to the global influence of the u-vector entries. The system of linear equations is preconditioned and solved with the conjugate gradient algorithm [6]. If the resulting control grid does fold the iteration is rated as not improving.

3.5 Grid generation procedure

Coarse optimization Within the coarse optimization, refinement points stay linearly dependent on the neighbours. All control points and free u-vector entries are optimized. After this optimization the coarse control points, which will be part of the refined control grid, will remain fixed.

Edge refinement optimizations A set of optimizations will be created for each refinement setting. One optimization targets one refinement section in one index direction through the whole topology. Hereby all coarse control nodes are fixed and only the refinement nodes in the targeted section are optimized. In order to make these optimizations independent from each other all tangents pointing into a refinement section are estimated in advance and the u-vectors are fixed.

Surface refinement The edge refinement leads to higher resolution edges. As a regular grid is required for the discretization, inner points for coarse cells of the control grid, where both index directions have been refined, must be set. This is done by a transfinite interpolation using the refined edges. To further improve the grid quality this should only be the initialization for a following surface refinement optimization.

u-vector optimization The u-vectors were fixed to make the intermediate optimizations independent. As the refinement points are shifted, the u-vectors should be optimized once again. All nodes of the control grid are fixed and the only parameters are the free u-vector entries and the only criteria are the estimated stretching ratios of the final grid.

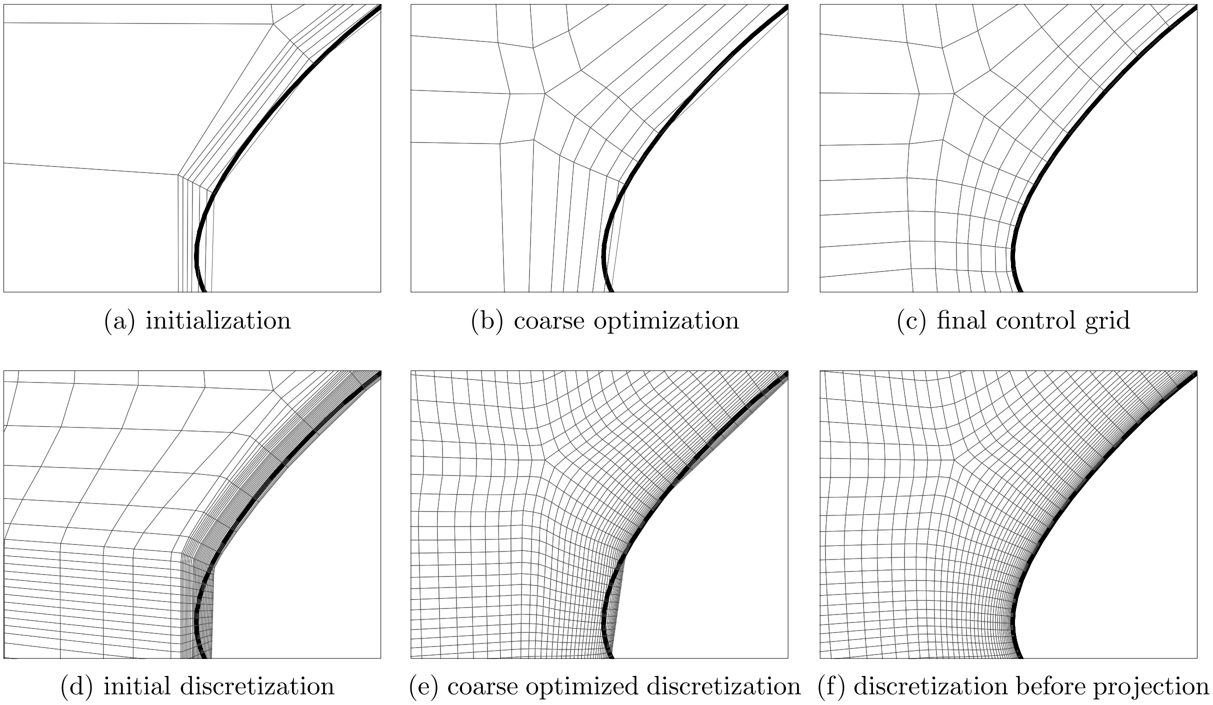
In case of stability issues with the coarse optimization, it may help to do an initial u-vector optimization at the beginning of the procedure.

Discretization and final projection In this step the full resolution grid is generated by the discretization of the abstraction layer. Due to the refinement settings and optimizations the resulting discrepancy between the geometry and the discretization is low. All points that should lie on the geometry are projected and the required shift decays within one block. For the output, the projected grid is merged to represent the original input topology.

Figure 4 shows some intermediate steps. With 4a being the initialization, the coarse optimization results in 4b. In Picture 4c the final control grid with a refined resolution at the geometry can be seen.

4 RESULTS

The presented procedure has been tested with five compressor and three turbine geometries at three spanwise positions each. Additionally, the result for a compressor tandem configuration with a more complex topology is shown. For all test cases a common set of weights for the different grid quality criteria was used. The resolution and refinement of the control grid and the target resolution are input parameters to the process. Furthermore the first distances tangential and orthogonal at the leading and trailing edge of the blades are prescribed. As general settings, the weight of the grid criteria have been increased near the airfoils and orthogonal to the airfoils an expanding grid spacing is preferred.



(a) initialization (b) coarse optimization (c) final control grid

(d) initial discretization (e) coarse optimized discretization (f) discretization before projection

Figure 4: Optimization steps for turbine test case near leading edge

For a generic turbine blade row figure 4 shows intermediate representations during the optimization. Hereby Figures 4d and 4e are only visualizations and not used within the procedure. A complete view of the generated S1 grid section can be seen in Figure 5.

In Figure 6 the first rotor of the transonic compressor rig of the Darmstadt University of Technology [7] was used. As a convergence test the presented method has been applied to 21 spanwise positions independently and the result has been merged to create a 3D grid. All input parameters were held constant, only the spanwise position changed. The spanwise curves following some prominent points in the topology and show a smooth characteristic. While the stagger angle and the spatial block distribution varies, the delta between two consecutive slices remains small. Only a little kink is present on the pressure side of the blade surface.

Figure 7 shows a grid generated for a DLR compressor tandem configuration [8, 9]. The method handles a more challenging topology here and leads to high quality fully 1-to-1 connected grid.

5 LIMITATIONS

The method needs a proper initialization with a non-folding control grid. For each topology a process of generating a initial grid, stable enough to handle geometry variations within a design optimization, is required.

As a pure gradient based single target optimization and by only allowing non-folding grids there is no guarantee for a convergence or finding the global minimum. The applica-

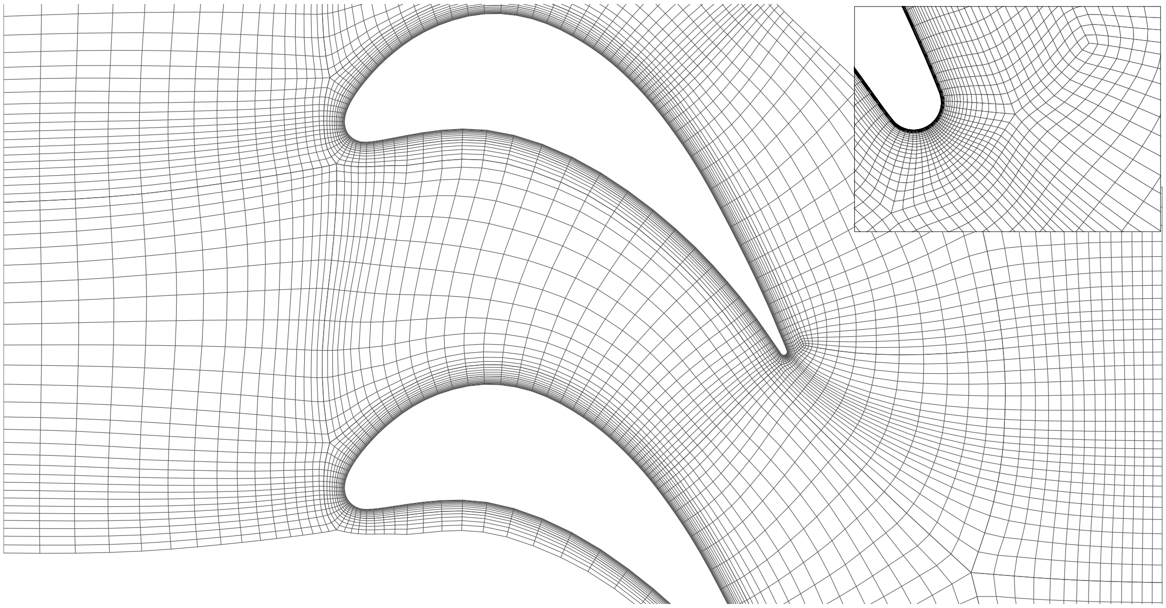


Figure 5: Optimization result for turbine test case, every 2$^{\text{nd}}$ grid line shown, with full resolution closeup

tion of the method failed to converge for only one of 25 test case which could be prevented by using a prior optimization of the u-vectors.

Depending on the topology the required splitting for the abstraction layer may lead to problems. Generally the resolution of the discretization must be higher than the resolution of the control grid. For example it is not possible to handle a topology where two lines in parallel lead to a singularity each.

6 CONCLUSIONS AND FUTURE WORK

A new approach to multi-block structured grid generation has been presented. It targets the creation of high quality grids for CFD calculations within design optimizations where a fixed topology can be used. The method is build around an abstraction layer based on the 2$^{\text{nd}}$ degree B-spline formulation to reduce the degrees of freedom for the grid generation drastically. On this abstraction layer a single objective gradient-based optimization is performed to improve different approximately calculated grid quality criteria.

The method has been described in its mathematical formulation and its application. For the set of test geometries the method produces high quality grids without parameter adjusting needed.

As the results in 2D are promising we will work on the extension to 3D with the aim to validate the procedure within a design optimization.

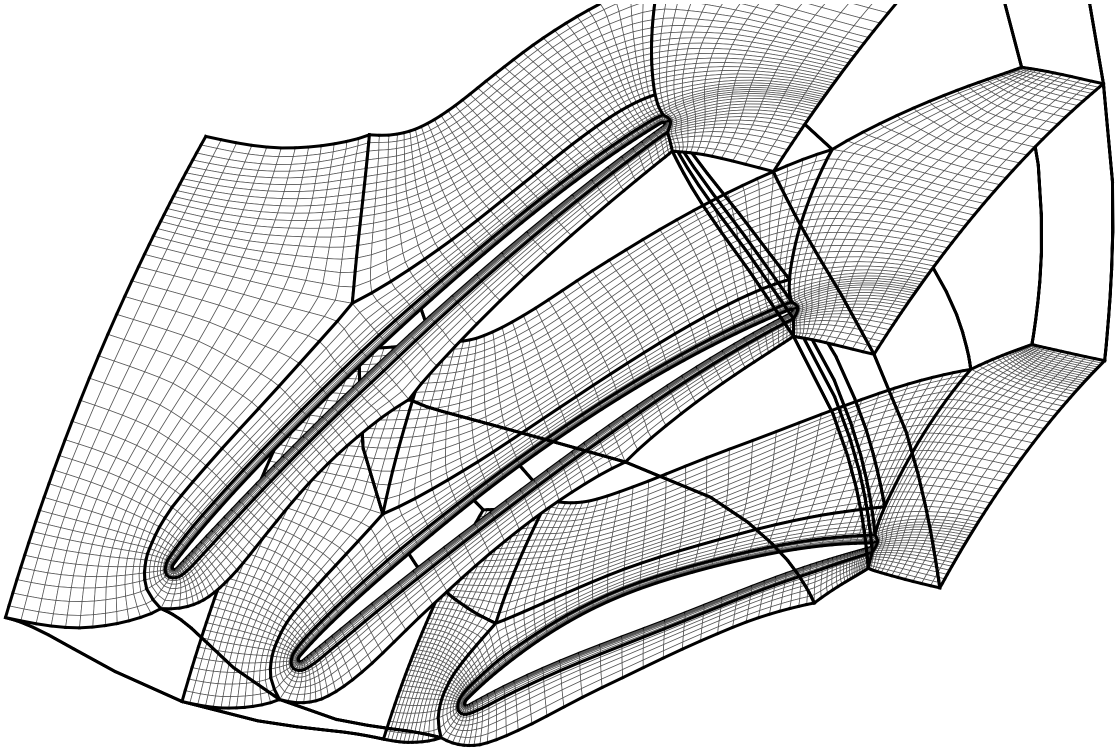


Figure 6: Optimization result for compressor test case - 3 of 21 independently created S1 slices shown with every 2^{nd} grid line

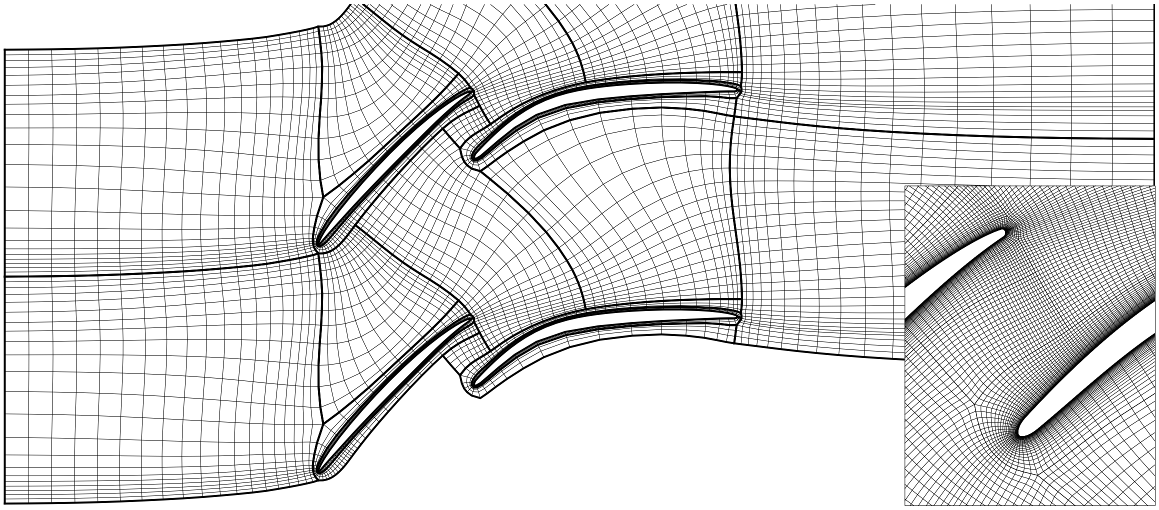


Figure 7: Optimization result for tandem test case, every 4^{th} grid line shown, with full resolution closeup

7 NOMENCLATURE

Symbols	
u, v	B-spline parameter, u for curves, u,v for surfaces
u_{i_j}	j^{th} B-spline parameter from the i^{th} u-vector in the abstraction layer
$k_1, k_2, \ldots, k_{n-1}, k_n$	strictly monotonously rising distinct B-spline knot vector entries
S	surface definition
P	vector $[x, y, z]^T$ representing a point
c	value of a local grid criterion
w	weight assigned to a grid criterion

REFERENCES

[1] Aulich, M., Voss, C. and Raitor, T., *Optimization Strategies demonstrated on a Transonic Centrifugal Compressor*. ISROMAC 15, 2014.

[2] Zhang, Y., Jia, Y., Wang, S., and Chan, H.C., *Boundary treatment for 2D elliptic mesh generation in complex geometries*. Journal of Computational Physics, 2008.

[3] Piegl, L., and Tiller, W., *The NURBS Book, 2nd ed.*. Springer, 1995.

[4] Gordon, W.N., and Hall, C.A., *Construction of curvilinear coordinate systems and application to mesh generation*. International Journal for Numerical Methods in Engineering, Vol. 7, 1973.

[5] Marquardt, D.W. *An algorithm for least-squares estimation of nonlinear parameters*. Journal of the Society for Industrial and Applied Mathematics, 1963.

[6] Hestenes, M.R., and Stiefel, E., *Methods of conjugate gradients for solving linear systems*. Journal of Research of the National Bureau of Standards, Vol. 49, 1952.

[7] Schulze, G., Blaha, C., Hennecke, D., and Henne, J., *The performance of a new axial single stage transonic compressor*. Journal of Turbomaschinery, Vol. 2006, 2006.

[8] Hergt, A., and Siller, U., *About Transonic Compressor Tandem Design: A Principle Study*. ASME Turbo Expo, GT2015-42115, 2015.

[9] Hergt, A., Grund, S., Klinner, J., Steinert, W., Beversdorff, M. and Siller, U., *Some Aspects of Transonic Compressor Tandem Design*. ASME Turbo Expo, GT2018-75132, 2018 (to be published).