Mesh Generation for Aircraft Engines based on the Medial Axis

J. Barner[‡], F. Buchegger[†,‡], D. Großmann[‡], B. Jüttler[†]

[†]Institute of Applied Geometry

Johannes Kepler University, Linz, Austria

[‡]MTU Aero Engines GmbH

Munich, Germany

Cambridge, Oct 9, 2014

MTU Aero Engines

MTU Aero Engines - Germany's leading manufacturer in the engine industry

- engages in the development, manufacture, marketing and support of commercial and military aircraft engines
- 7900 employees worldwide

Motivation

Before the actual manufacturing of parts for aircraft engines MTU runs through an iterative process of

- designing,
- analyzing and
- redesigning.

For the analysis step a suitable quad-mesh has to be created. Its creation is very costly and time-consuming.

Motivation

Before the actual manufacturing of parts for aircraft engines MTU runs through an iterative process of

- designing,
- analyzing and
- redesigning.

For the analysis step a suitable quad-mesh has to be created. Its creation is very costly and time-consuming.

Commercial mesh generators do not provide enough flexibility to deal with newer, more sophisticated geometrical designs.

Motivation

We propose a framework for mesh generation that consists of two parts:

- analysis of the topology:
  - split domain into four-sided blocks based on the medial axis
- optimization of the blocks:
  - fill the blocks with B-Spline surfaces
  - optimize the whole multipatch construct with respect to a set of functionals

Motivation

We propose a framework for mesh generation that consists of two parts:

- analysis of the topology:
  - split domain into four-sided blocks based on the medial axis
- optimization of the blocks:
  - fill the blocks with B-Spline surfaces
  - optimize the whole multipatch construct with respect to a set of functionals

We can combine these two parts or use them independently.

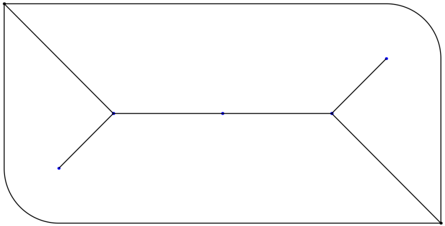
Related Work - Blocking and Mesh Generation

- Blocking methods based on the medial axis
  - Tam and Armstrong 1991.
  - Rigby 2004.
  - . . .
- Mesh generation
  - Liseikin 2010.
  - . . .

Outline

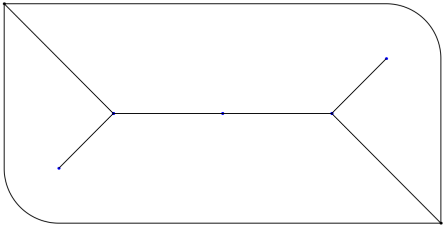
Medial axis

- **Definition:** The medial axis MA(()Ω) of a domain Ω is defined as the (closure of the) set of all points in Ω, that have at least two closest points on ∂Ω.



Medial axis

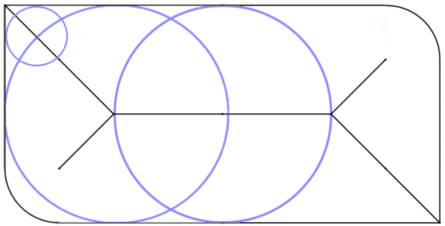
- **Definition:** The medial axis MA(()Ω) of a domain Ω is defined as the (closure of the) set of all points in Ω, that have at least two closest points on ∂Ω.



- **Equivalently:** The medial axis MA(Ω) is simply the set of centers of all maximal inscribed disks in Ω.

Medial axis

- **Definition:** The medial axis $MA(\Omega)$ of a domain Ω is defined as the (closure of the) set of all points in Ω, that have at least two closest points on $\partial\Omega$.



- **Equivalently:** The medial axis $MA(\Omega)$ is simply the set of centers of all maximal inscribed disks in Ω.

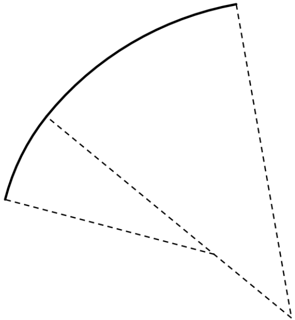
Computation of the medial axis

- Aichholzer et al. 09 proposed an algorithm for the computation of the medial axis of a planar domain.
  - **Step 1:** Approximation of the boundary by biarcs.
  - **Step 2:** Applying a divide & conquer algorithm to the approximated shape.
- We will use a slightly modified, improved version of this algorithm, details can be found in Buchegger et al. 14.

Approximating the Boundary

First the boundary is split at the stationary points of the curvature, then the pieces are approximated using spiral biarcs (Meek & Walton 08).

- This scheme does not introduce new local curvature extrema.
- An adaptive bisection method allows to reduce the error to a given threshold.



Biarc: two G^1 connected arcs

Divide & Conquer

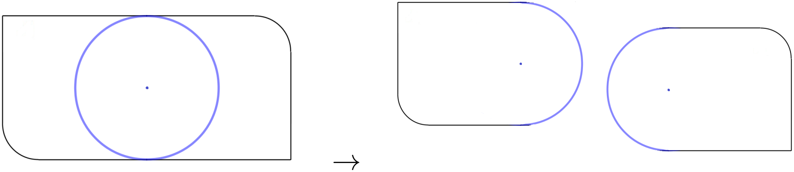
- Algorithm is based on the Domain Decomposition Lemma (Choi et al. 97).
- **Idea** of lemma & algorithm:
  - The given domain is split into subdomains.
  - The union of the medial axes of the subdomains is the medial axis of the whole domain.
  - Recursive segmentation of subdomains until all subdomains are base cases.

Divide step

- A maximal inscribed disk has to be found.
- This is done by
  - randomly selecting a curve from the boundary and
  - finding the maximal disk, which is tangent to this curve at an endpoint (instead of the midpoint as suggested in Aichholzer et al. 09).
- The number of newly introduced boundary curves is reduced by one in each step compared to Aichholzer et al. 09.

Divide step

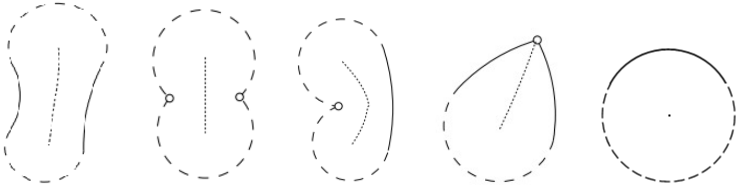
- The boundary is broken up into new subdomains at the tangent points of the maximal disk.
  - The new subdomains are closed with parts of the maximal disk inserted as artificial arcs.



- The new subdomains are again subject of the divide step.

Base cases

- We find more maximal disks and therefore we split the proposed base cases from Aichholzer et al. 09 even further.

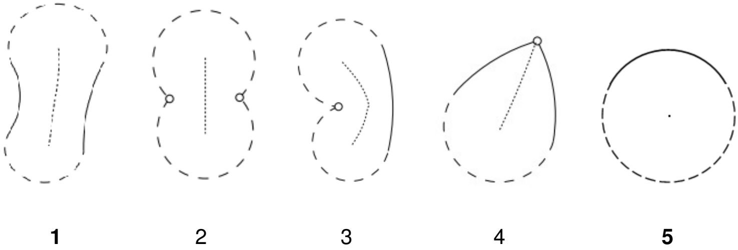


1 2 3 4 5

- Reduction of number of base cases from thirteen to five compared to Aichholzer et al. 09.

Base cases

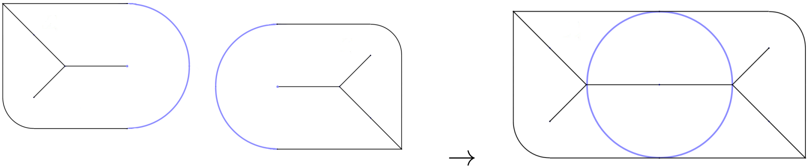
- We find more maximal disks and therefore we split the proposed base cases from Aichholzer et al. 09 even further.



- If the boundary is at least G^1-smooth, then the number is even reduced to two.

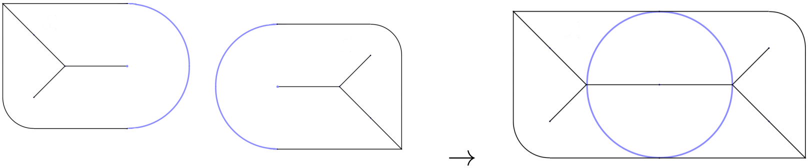
Conquer step

- The medial axes of the base cases/subdomains are simply glued together.



Conquer step

- The medial axes of the base cases/subdomains are simply glued together.



- A possible numerical error in the medial axis computation of one base case does not influence the computations of other base cases.

Computation of the medial axis

- The runtime is $\mathcal{O}(n \log n)$ under the assumption that the graph diameter of the medial axis is $\Theta(n)$ (Aichholzer et al. 09).
- The resulting medial axis is given back as edge/node structure.
- By using the concept of generalized domains (Aichholzer et al. 10), the algorithm can also deal with multiple connected domains.

Blocking

Based on the medial axis we can block the domain by applying these three steps:

- simplification of the medial axis
- insertion of normals on the boundary curves
- removing of the redundant edges

Insertion of Normals

- find vertices with more than two edges
- insert lines perpendicular to the boundary

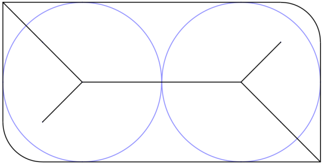


Figure: Medial axis

Insertion of Normals

- find vertices with more than two edges
- insert lines perpendicular to the boundary

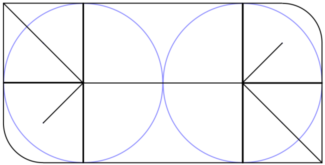


Figure: Inserted normals

Removing redundant edges

- visit vertices, which have edges to the boundary
- remove vertices dependent on the angles

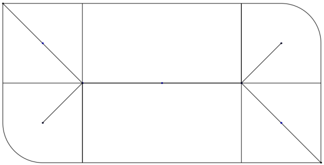


Figure: Blocking with redundant edges

Removing redundant edges

- visit vertices, which have edges to the boundary
- remove vertices dependent on the angles

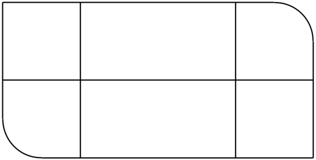


Figure: Blocked domain

Removing redundant edges

- visit vertices, which have edges to the boundary
- remove vertices dependent on the angles

Figure: Blocked domain

This process divides the shape into triangular and quadrilateral domains.

Removing redundant edges

- visit vertices, which have edges to the boundary
- remove vertices dependent on the angles

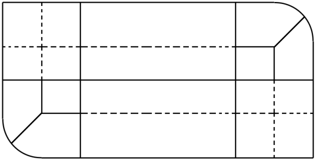


Figure: Blocked domain

This process divides the shape into triangular and quadrilateral domains.
Each triangular domain can be split in three four-sided domains.

Example: Hub

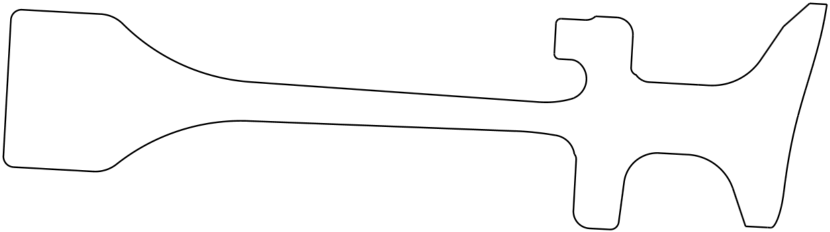


Figure: Domain

Example: Hub

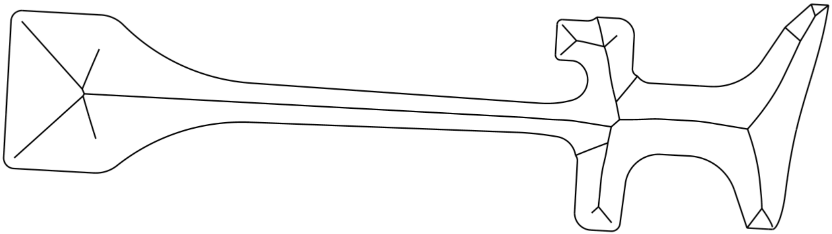


Figure: Domain with medial axis

Example: Hub

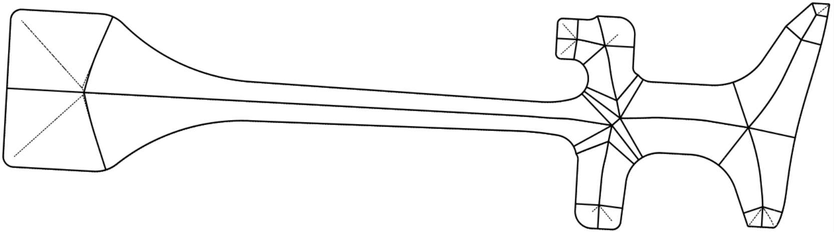


Figure: Domain, separated in blocks

Example: Hub

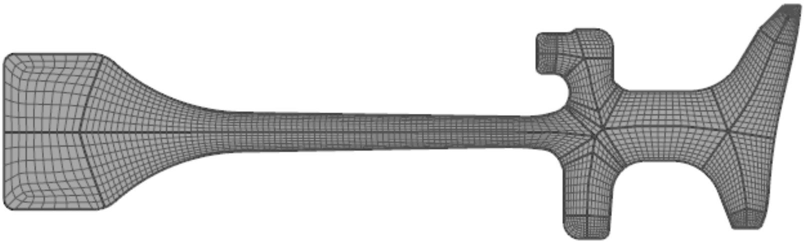


Figure: Resulting mesh

Total calculation time: 1.9 sec.

Example: Hub

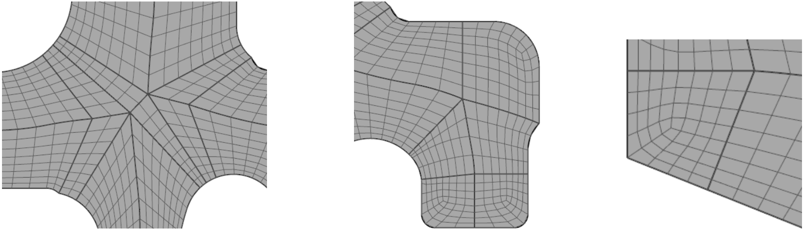


Figure: Details of the mesh

Example: Part with holes

By using the concept of generalized domains from Aichholzer et al. (2010) also multiply connected domains can be processed.

- First maximal disks that connect each of the holes with the boundary are inserted.
- The boundary curve of each hole is inserted in the outer boundary curves between the two parts of the circle bounding the maximal disks.
- We end up with one boundary, including the inner holes, so we can treat the shape as if it was simply connected.

Example: Part with holes

By using the concept of generalized domains from Aichholzer et al. (2010) also multiply connected domains can be processed.

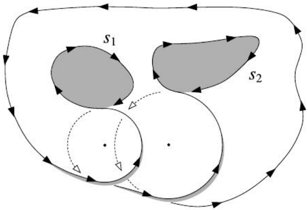


Figure: Generalized domains

Example: Part with holes

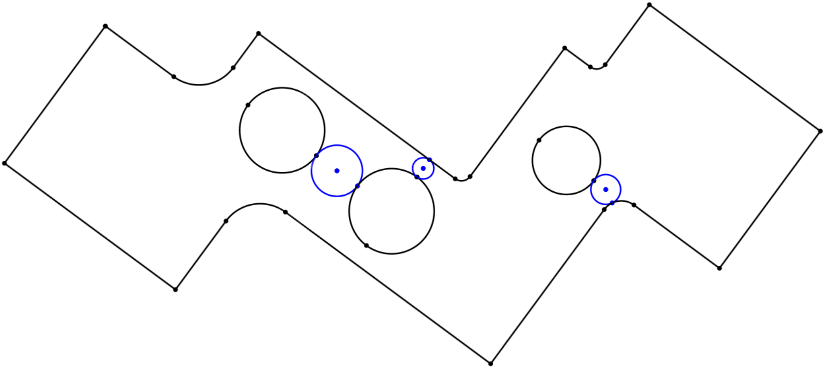


Figure: Domain

Example: Part with holes

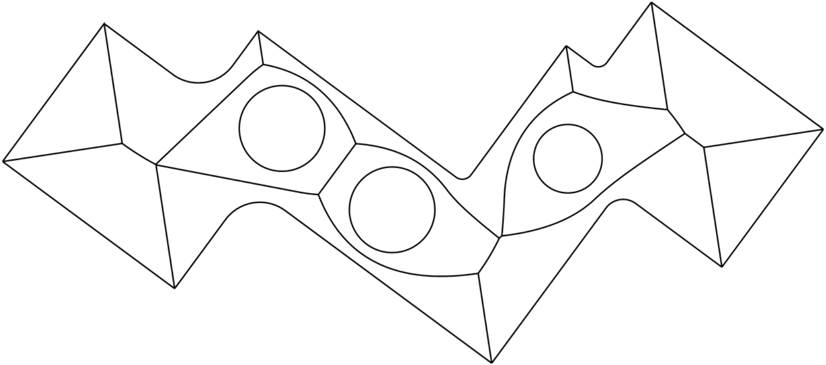


Figure: Domain with medial axis

Example: Part with holes

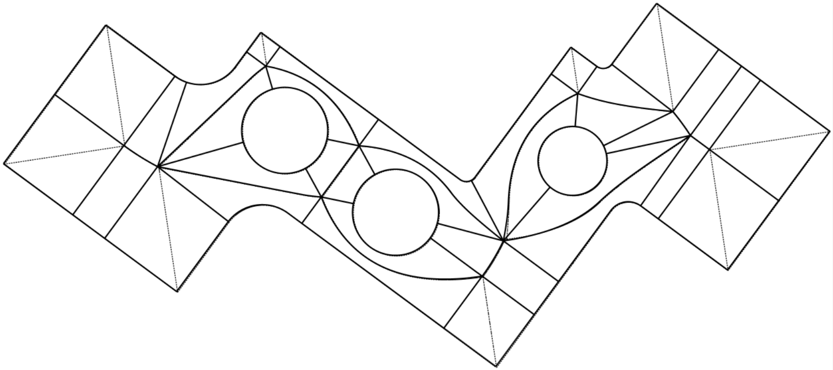


Figure: Domain, separated in blocks

Example: Part with holes

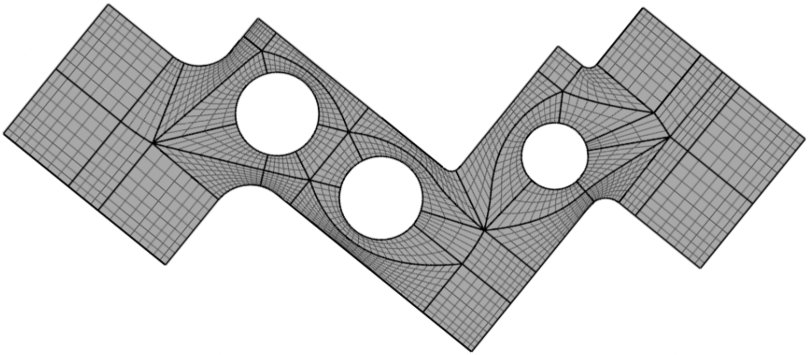
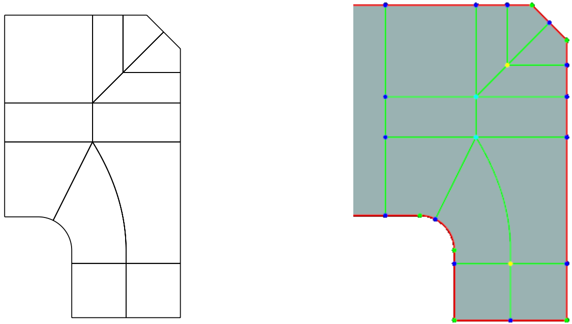


Figure: Resulting mesh

Total calculation time: 0.6 sec.

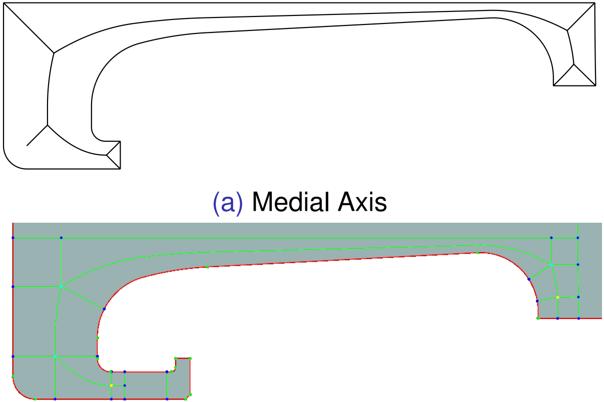
Example: Engine parts



(a) Medial Axis (b) Blocking

Figure: Engine part 1

Example: Engine parts

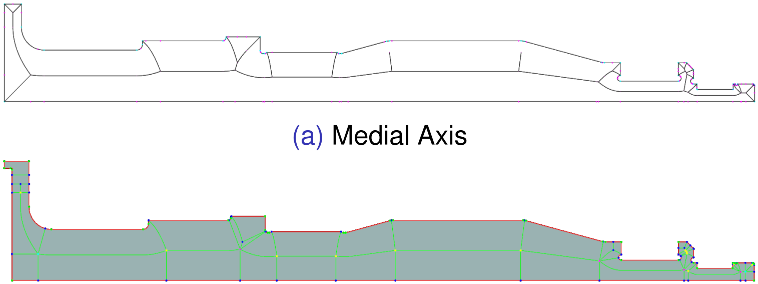


(a) Medial Axis

(b) Blocking

Figure: Engine part 2

Example: Engine parts



(a) Medial Axis

(b) Blocking

Figure: Engine part 3

Limitations of this method

- Shapes which change a lot in width.
- Shapes which have maximal disks with a lot of tangent points.
- Concave corners.

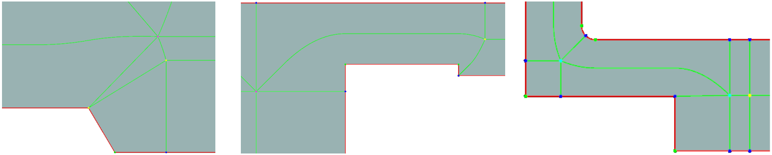


Figure: Possible effects of concave corners

Outline

General Process

General process to achieve a parameterization of a given domain:

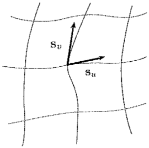
- A multipatch-topology has to be chosen for the domain (e.g. constructed by step 1).
- The patches are filled with B-Spline surfaces (with Coons patches). They are connected with $\mathcal{C}^0$ smoothness.
- Using a standard Gauss Newton method we optimize the surfaces w.r.t. a set of functionals (Großmann, 2012).
- The mesh can be easily obtained by discretizing the optimized surfaces.

Functionals on the surfaces

For optimization on the surface we use following functionals (Liseikin, 2010):

- length functional:

$$\mathcal{Q}_l = \iint_\Omega \|s_u\|^2 + \|s_v\|^2 \, \mathrm{d}u\mathrm{d}v$$



- uniformity functional:

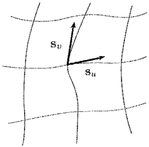
$$\mathcal{Q}_u = \iint_\Omega \|s_{uu}\|^2 + \|s_{uv}\|^2 + \|s_{vv}\|^2 \, \mathrm{d}u\mathrm{d}v$$

Functionals on the surfaces

For optimization on the surface we use following functionals (Liseikin, 2010):

- orthogonality functional:

$$\mathcal{Q}_o = \iint_\Omega (s_u s_v)^2 \,\mathrm{d}u\mathrm{d}v$$



- area-squared functional:

$$\mathcal{Q}_a = \iint_\Omega (s_u \times s_v)^2 \,\mathrm{d}u\mathrm{d}v$$

Example: Air Passage

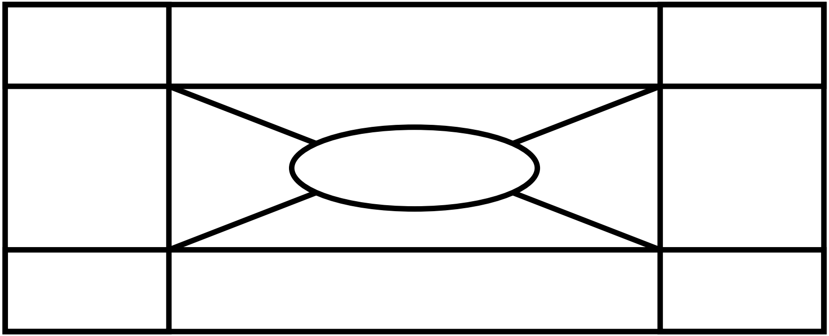


Figure: Topology

Example: Air Passage

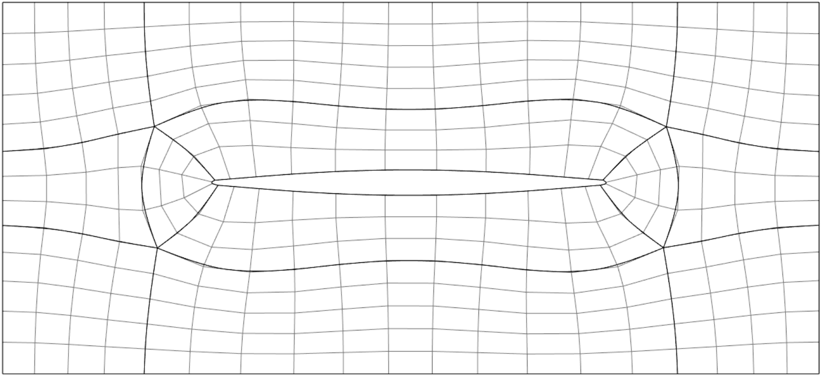


Figure: Control net

Example: Air Passage

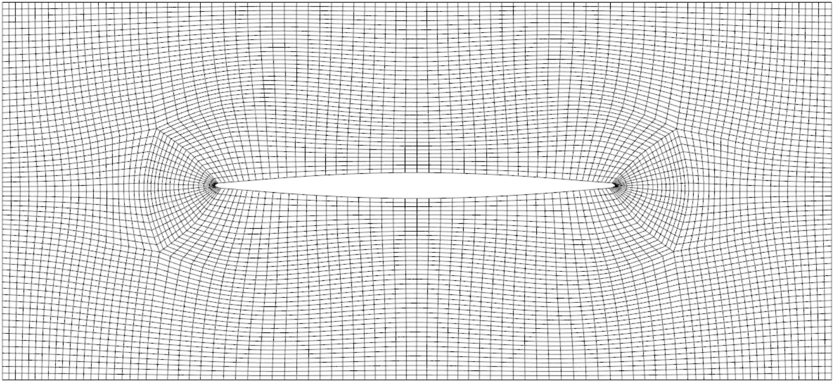


Figure: Mesh

Example: Air Passage - Multipassage

For a more efficient design it can be neccessary to look at more than one blade in the domain.

- This is a problem for the current process.
- In our framework this is easily possible.

Example: Air Passage - Multipassage

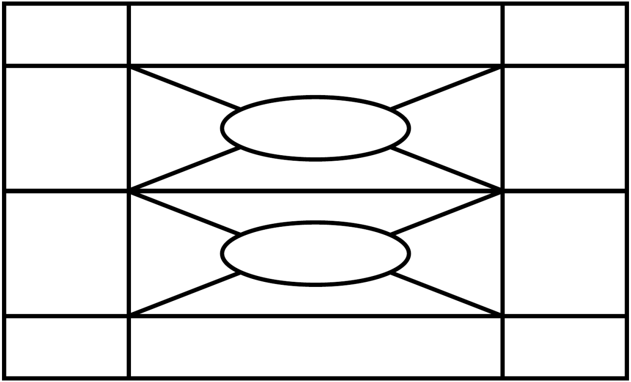


Figure: Topology

Example: Air Passage - Multipassage

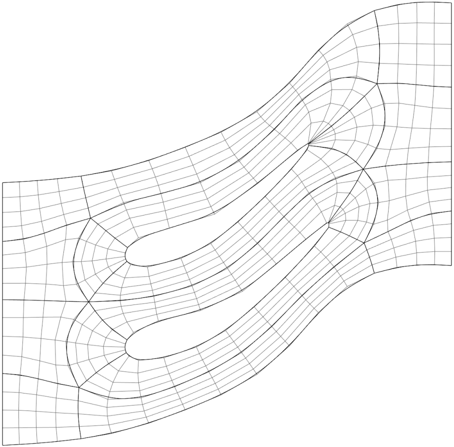


Figure: Control net

Example: Air Passage - Multipassage

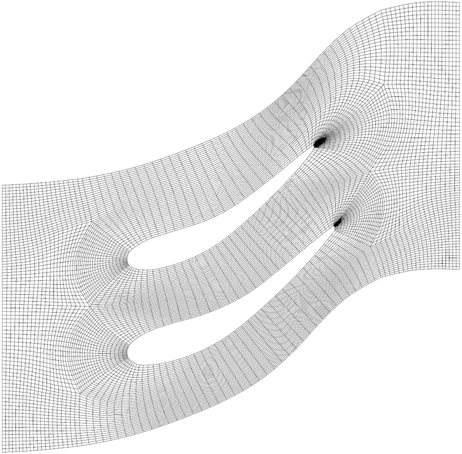


Figure: Mesh

Outline

Conclusion

We have presented a framework for generating quad meshes given a multiply connected domain by

- creating a multipatch-topology using the medial axis,
- filling the patches with B-Spline surfaces,
- optimizing the surfaces w.r.t. a set of functionals and
- obtaining the quad mesh by discretizing the surfaces.

The steps can be carried out automatically and the resulting mesh has nice properties.

Outlook

- connecting the patches with increased smoothness, having implicite smoothness in the parameterization
- provide local adaptivity
- going to 3D
- ...

References I

Aichholzer, O., Aigner, W., Aurenhammer, F., Hackl, T., Jüttler, B., Pilgerstorfer, E., and Rabl, M. (2010).
Divide-and-conquer algorithms for Voronoi diagrams revisited.
Comput. Geom. Theory Appl., 43:688–699.

Aichholzer, O., Aigner, W., Aurenhammer, F., Hackl, T., Jüttler, B., and Rabl, M. (2009).
Medial axis computation for planar free-form shapes.
Computer-Aided Design, 41:339–349.

Choi, H. I., Choi, S. W., Moon, H. P., and Wee, N.-S. (1997).
New algorithm for medial axis transform of plane domain.
Graphical Models and Image Processing, 59(6):463 – 483.

References II

Liseikin, V. D. (2010).
Grid Generation Methods.
Scientific Computation. Springer.

Meek, D. and Walton, D. (1999).
Spiral arc spline approximation to a planar spiral.
Journal of Computational and Applied Mathematics, 107(1):21 – 30.

Rigby, D. (2003).
Topmaker: A technique for automatic multi-block topology generation using the medial axis.
In *ASME/JSME 2003 4th Joint Fluids Summer Engineering Conference*, pages 1991–1997. American Society of Mechanical Engineers.

Tam, T. and Armstrong, C. (1991).
2D finite element mesh generation by medial axis subdivision.
Advances in Engineering Software and Workstations, 13(5/6):313 – 324.