Distance field driven mean camber line extraction algorithm

Kumar, S. Vijai; Vuik, Cornelis

Publication date
2020

Document Version
Final published version

Citation (APA)
Kumar, S. V., & Vuik, C. (2020). *Distance field driven mean camber line extraction algorithm*. (Reports of the Delft Institute of Applied Mathematics; Vol. 20-03). Delft University of Technology.

DELFT UNIVERSITY OF TECHNOLOGY

REPORT 20-3

Distance field driven mean camber line extraction algorithm

S Vijai Kumar, Cornelis Vuik

ISSN 1389-6520

Reports of the Delft Institute of Applied Mathematics

Delft 2020

Abstract

Distance fields finds a lot of applications recently in computational geometry. We propose an algorithm to extract the mean camber line of an airfoil using distance fields. This method does not require computing any geometric approximations inside the airfoil geometry. It also does not need the airfoil coordinates to be evenly distributed on the top and bottom. All the computations are done on a simple Cartesian mesh and can be easily coupled with any existing computational fluid dynamics codes. We show the efficiency of the algorithm on a variety of airfoil geometries. We demonstrate the robustness and flexibility of the algorithm for a wide range of airfoil geometries.

1 Introduction

Airfoil selection is one of the most preliminary decisions made in the initial design process of aerodynamic machineries such as aircraft, race cars, wind turbines, and gas turbine engines. Traditionally an Aerodynamicist designs airfoil contour using techniques such as the thin airfoil theory/panel methods[1]. Wind tunnel test results complement the theoretical results[2]. It is easy to find formulae for computing parameters such as mean camber line for existing airfoils such as NACA or Eppler series airfoils[3][4]. However, manually computing the mean camber line of an arbitrary airfoil requires computing a Voronoi diagram to pack a series of circles inside the airfoil. One must join the loci of these points for obtaining the mean camber line[5]. There have been very notable improvements done on computing airfoil coordinates at specified chord location for given camber[6]; however, the present investigation deals with the inverse problem of computing the mean camber line for any given airfoil distribution.

There has been an acute increase in the indirect/stochastic approaches provided for different components of aircraft design and engineering design. Recent studies from Sekar et al. show the ability of machine learning algorithms in airfoil selection[7]. Similar techniques have been applied for computing different performance characteristics of airfoils[8][9].

The field of computational geometry has also seen an increase in indirect approaches for several different geometry problems. The most notable work relevant to the present investigation is the usage of the heat conduction equation for computing geodesic distances [10]. The heat equation is approximated on surfaces as an alternative to traditional geodesic distance approaches. The present investigation focuses on the problem of computing mean camber lines in airfoils without requiring computation of any geometric predicates. We focus on the usage of distance fields for the extraction of mean camber lines. We compute the distance field inside the airfoil geometry which is approximated on a Cartesian mesh and perform a simple gradient computation to extract the mean camber line of the airfoil. The proposed algorithm has other applications beyond aerodynamic design. It can also be useful in areas such as mesh generation where the mean camber line can give helpful information about the curvature flow of the airfoil. The algorithm outlined in the present investigation can be handy for aerodynamic practitioners in reverse engineering mean camber lines for any random airfoil contour. The algorithm does not impose any specific requirement on the uniformity of the point distribution. It does not require any distinction between the top and bottom of the airfoil either. The present investigation is primarily intended as a reverse engineering tool for acquiring information about unknown airfoil countours.

2 Methodology

The algorithm is built on top of distance fields. Distance fields computed using the level set method or the heat method is quite popular in various domains of computational fluid dynamics where different components are moving inside a computational mesh. We make use of the fast marching method[11] for computing the distance field in any arbitrary domain. The Eikonal wave equation is solved in an arbitrary domain using finite difference method[12]. The distance field is considered to be a general scalar field. Gradient computation performed on a scalar field yields the mean camber line of an airfoil. The algorithm can be generalized to extract the medial axis of arbitrary domains. The medial axis acts as a base decomposition for applications such as mesh generation[12]. However, the paper focuses on the extraction of mean camber lines from different airfoil contours.

2.1 Theory

The Level Set method is an extension to curvature shortening flow[13] where a curve is assigned an artificial velocity and is diffused down to a singularity. The primary advantage of level set computation is the ability to compute a so-called distance field inside and outside a geometry. The distance obtained is a signed distance where the positive and negative values distinguish the inside and outside of a closed curve. It advances curvature fronts radially outward along the normal direction of the curvature in a computational domain and diffuses radially inward in the opposite direction. There are numerous applications to distance fields in the field of computational geometry, computational fluid dynamics, robotics amongst other fields. Sethian[11] made use of a stationary Eikonal wave equation for computing the distance field. The Fast marching level set method used for computing distance field d in the present investigation is based on Sethian's[11] work. It is computed by solving the Eikonal wave equation.

$$F|\nabla\phi| = 1 + \epsilon\nabla^2\phi \tag{1}$$

where F is the velocity and ϕ describes the arrival time of the wave fronts. By setting ϵ to zero in equation 1, it becomes the boundary value formulation for the "front tracking problem" where F is always greater than zero and equal to $\dfrac{d\phi}{dx}$. The Equation 1 can be solved using an upwind finite difference scheme on a two-dimensional Cartesian mesh. A more detailed explanation of the finite difference discretization for the fast marching method can be seen in reference[11] and [12]. This particular case of the distance field can also be computed with diffusion equation[14] / heat equation[10].

2.1.1 Distance field computation using Fast Marching Method

The fast marching method developed by Sethian is an excellent algorithm for computing distance fields. We project a closed polygon (closed as a result of concave hull approximation) onto a Cartesian mesh. The cells that represent the boundary of the polygon are assigned as the zero level set. We assign an artificial velocity to the zero level set (which is the boundary of the polygon), and the Eikonal wave equation is solved inside the domain using the fast marching method. The result of the fast marching method is a distance field inside the airfoil. The sign of the distance field helps distinguish the inside of the airfoil geometry from outside.

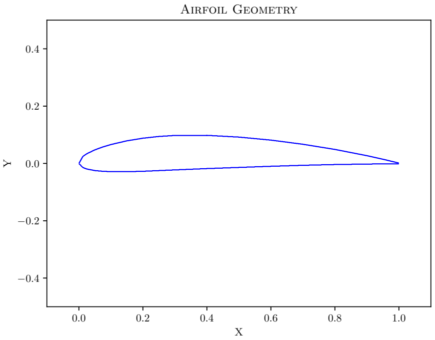


Figure 1: Input airfoil geometry

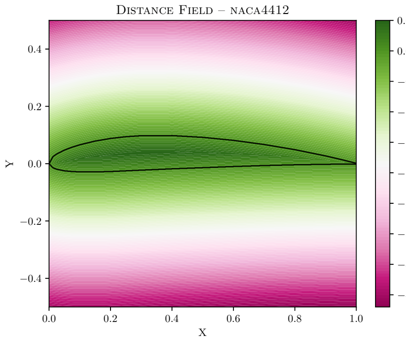


Figure 2: Distance field computed using the fast marching method

2.1.2 Distance field computation using Heat Equation / Diffusion Equation

A newer approach for approximating distance field is with the help of the Diffusion equation/heat equation. An airfoil represented in a Cartesian mesh are created using the same technique as above. Since the algorithms for computing the mean camber line of the airfoil only requires the distance field inside the airfoil. This problem reduces to a simple heat conduction problem. The boundary nodes forming the airfoil geometry are assigned a very high temperature and a null value is assigned everywhere else. The heat conduction equation is then solved inside the airfoil using the finite difference method. A scalar field that represents the temperature inside the airfoil resembles the distance field produced inside the airfoil using the fast marching method. The heat conduction simulation is run for a few time steps until the high temperature at the boundary reaches the interior of the airfoil.

$$\frac{\partial T}{\partial t} = k(\frac{\partial^2 T}{\partial x^2} + \frac{\partial^2 T}{\partial y^2}) \tag{2}$$

A detailed investigation on heat conduction based distance field amongst other applications has been conducted by Crane et al[10] which provides numerous applications in aerodynamics beyond the mean camber line extraction.

2.2 Algorithm

The algorithm for extracting the mean camber line of an airfoil is based on the distance field generated inside the airfoil. The algorithm itself is rotationally invariant; however, the computed mean camber line can be transformed with simple affine transformations. A variant of the algorithm also addresses airfoils that do not have zero angle of attack. Once computed, the mean camber line can be transformed to any axis. In the case of reverse engineering, input geometry can have a non zero angle of attack. Simple geometric transformations help compute the angle of attack of an airfoil concerning both axes.

Algorithm 1: Algorithm to extract mean camber lines in airfoils (Airfoil parallel to global X or Y axis)

```
Result: Vector of line segments
Read airfoil geometry as line segments or a vector of points;
Compute the concave hull of the airfoil coordinates;
Create a Cartesian mesh of a size greater than the axis-aligned bounding box of
  the airfoil geometry;
forall Edges of airfoil do
    Find intersecting cells in the Cartesian mesh and mark them as boundary cells;
end
Set the intersecting cells in the Cartesian mesh as the zeroth level set ($\phi(x, t = 0)$
  (in case of fast marching method);
Compute the distance field inside the airfoil geometry using fast marching level set
  method (or) heat method;
Compute the gradient along X / Y direction (depending on which axis is parallel
  to the airfoil);
The result is obtained as a mask on the Cartesian mesh where the cells forming
  the mean camber line are marked to be True;
The marked cells can be grouped into sets of edges to form the mean camber line
  of the airfoil;
```

If the airfoil is parallel to the X / Y axis, then the straightforward form of the mean camber line computation 1 can be followed. Once the airfoil geometry is projected to a Cartesian mesh, the distance field is computed inside the airfoil. The gradient of the distance field along the chord-wise direction would yield the mean camber line of the airfoil.

There are scenarios where airfoils are not parallel to either axis. In such cases, an oriented bounding box of the airfoil can be used to find the angle made by the chord-wise axis of the airfoil with respect to the X / Y-axis. The airfoil can be temporarily rotated towards one of the axes so that it is reasonably parallel. The listing below describes a modified version of algorithm 1 for computing the mean camber line of airfoils oriented arbitrarily.

Algorithm 2: Algorithm to extract mean camber lines in airfoils (Airfoil oriented arbitrarily)

```
Result: Vector of line segments
Read airfoil geometry as line segments or a vector of points;
Compute the concave hull of the airfoil geometry;
Compute the oriented bounding box of the airfoil;
Get the orientation angle of the bounding box (obb) with respect to X and Y axis
  (Let this angle be $\theta$);
if X axis forms a minimal angle with the oriented bounding box then
    Perform a rotation of $-\theta$ with respect to X axis;
end
else
    Perform a rotation of $-\theta$ with respect to Y axis;
end
Repeat the steps 3 to 11 from algorithm 1;
Rotate the airfoil back to its original position by rotating it to $\theta$ with respect to X
  or Y axis;
```

The present investigation is restricted to mean camber line extraction in airfoils. We do not intend to extract a more generic medial axis transform.

3 Different components of Algorithm

3.1 Input data and Mesh generation

The input data for the algorithm is a set of two dimensional planar points that represent the contour of the airfoil. The algorithm is tested on a wide variety of airfoil geometries available in public domain. These may also be coming from other sources such as scanned data from which airfoil coordinates could be extracted using image processing algorithms. Distance field computation using fast marching method does not require any distinction from inside and outside of the airfoil since the computations are performed on a watertight polygon computed as a concave hull of the airfoil coordinates. In a Cartesian mesh, the cells which intersect with edges of the airfoil are marked as zero level set for the fast marching method computation. This is always guaranteed to be a closed path since the computation of the concave hull guarantees a closed polygon.

3.2 Distance field computation

The Fast marching method based distance field computation is exactly the same as the algorithm employed by Xia, H. and Tucker, P.G.[12] and is explained in detail in the appendix of their paper. However the heat method for computing distance field requires a few additional steps. The advantage of fast marching level set method is that it does not require any prior distinction between inside and outside of the airfoil geometry. In many geometry / CFD based applications, the level set method is simply used to distinguish inside of a geometry from outside. A simple ray shooting approach can be used to distinguish the inside of the airfoil from outside and then heat equation can be solved inside the airfoil using finite difference method. The scalar field can be obtained using either explicit or implicit finite difference method.

Algorithm 3: Appromixated distance field like scalar field using heat equation (explicit FDM)

```
Result: Approximated distance field
Set a very high temperature value $T$ of 1 as initial value on the cells marked as the boundary cells in the input Cartesian mesh representing the airfoil geometry;
Set the temperature $T$ to be a zero value elsewhere in the mesh;
Since the temperature is obtained using an explicit finite difference scheme, stability needs to be enforced by choosing appropriate values for k and $\Delta T$;
Choose a k value of 1 and a CFL number (or) $\sigma$ of 0.25;
The value of $\Delta T$ can be computed using the following relation
    $\Delta T = \frac{\sigma * h^2}{k}$    (3)
while The temperature of every cell has not reached a value greater than or equal to 0.1 do
    forall Cells in the interior of the airfoil do
        Compute the temperature using finite difference method as follows;
        $T_{i,j}^{t+1} = T_{i,j}^{t} + \Delta T k (\frac{T_{i,j-1}^{t} + T_{i-1,j}^{t} - 4T_{i,j}^{t} + T_{i+1,j}^{t} + T_{i,j+1}^{t}}{h^2})$    (4)
    end
end
```

3.3 Derivative of distance field

The derivative of a distance field can be computed with the help of central finite difference method. Since the distance field computed using either the fast marching method or the heat method is a simple scalar field d. The problem of computing the derivative of the distance field reduces to computing the derivative of any arbitrary scalar field. The derivative in x or y direction can be computed as follows.

$$\begin{aligned} \frac{\partial d}{\partial x} &= \frac{d_{i+1,j} - d_{i-1,j}}{2h} \\ \frac{\partial d}{\partial y} &= \frac{d_{i,j+1} - d_{i,j-1}}{2h} \end{aligned} \tag{5}$$

The airfoil orientation decides the direction of the derivative. In most cases, the chord line is parallel to the x axis, hence derivative with respect to x would give the necessary derivative of the distance field for mean camber line computation.

3.4 Mean camber line extraction

The derivative of the distance field is computed in a Cartesian mesh and it does not have any geometric representation. The conversion of the distance field derivative into a mean camber line can be performed with a simple segmentation approach. The centroids of all the cells in the Cartesian mesh are computed. The Centroidal points are connected to each other based on the connectivity from their parent cells. All the newly formed edges are assigned a scalar value, which is the average value of the computed distance field derivative. Now a chaining algorithm can be used to form edge chains. A half-edge data structure[15] can be beneficial here. Given an edge, its left and right edge are checked for the derivative value, and if its a non zero value, then that edge is added to the current chain. This is repeated until no other edges are present to the left and right with the same derivative value. This approach, however, only works for mean camber line extraction and does not work in applications like multiblock decomposition for mesh generation. For such applications, a morphological thinning operation[16] is required.

3.5 Cleanup of output

In some cases, there are multiple segmented edge groups after mean camber line extraction stage. This could be due to some noise from derivative computation. The longest edge group (or) chain can be kept back and additional edge groups (or) chains can simply be ignored. The longest edge group (or) chain describes the mean camber line of the airfoil.

4 Theoretical results

Theoretical results can explain some of the rudimentary details in the algorithm better than numerical results. Some of the geometric arguments related to the algorithm, such as the ability of the algorithm to perform well without a clear distinction between top and bottom of the airfoil and the algorithm's capability in providing right mean camber line without a uniform coordinate distribution everywhere in the airfoil.

Statement 1. *A planar airfoil geometry defined by a set of discrete points does not require a uniform coordinate distribution everywhere.*

Proof. Let S_1 represent a set of finer (or semi uniform) points $(P_1, P_2, P_3, ...P_n)$ representing the airfoil geometry. Let S_2 represent a sparser set of points from S_1. Let their mean curvature value be represented by ϵ_1 and ϵ_2 respectively. The two sets S_1 and S_2 are the same as long as the deviation in mean curvature in S_2 from S_1 is within a reasonable threshold ($\|\epsilon_1 - \epsilon_2\| \leq \epsilon_{threshold}$). If this curvature threshold is set to a small value, then the sets S_1 and S_2 are essentially the same.

□

Lemma 1. *Hence, the concave polygon formed by S_1 and S_2 are essentially the same.*

Statement 2. *Distance-field based mean camber line computation does not require any distinction between the top and bottom of the airfoil.*

Proof. If statement 1 is satisfied, then the concave polygon of the planar airfoil geometry represented by a discrete set of points accurately represents the contour of the airfoil geometry. All the further computations are performed on a Cartesian mesh generated from this watertight polygon. Any boundary condition applied for distance field computation (such as defining a zero level set) is defined as a constant value for the entire polygon. Since there is no difference in the boundary condition for the top and bottom of the airfoil, there is no requirement to distinguish the top and bottom of the airfoil for distance field computation. □

Statement 3. *Derivative of a distance field along the chordwise direction produces a mean camber line of an airfoil.*

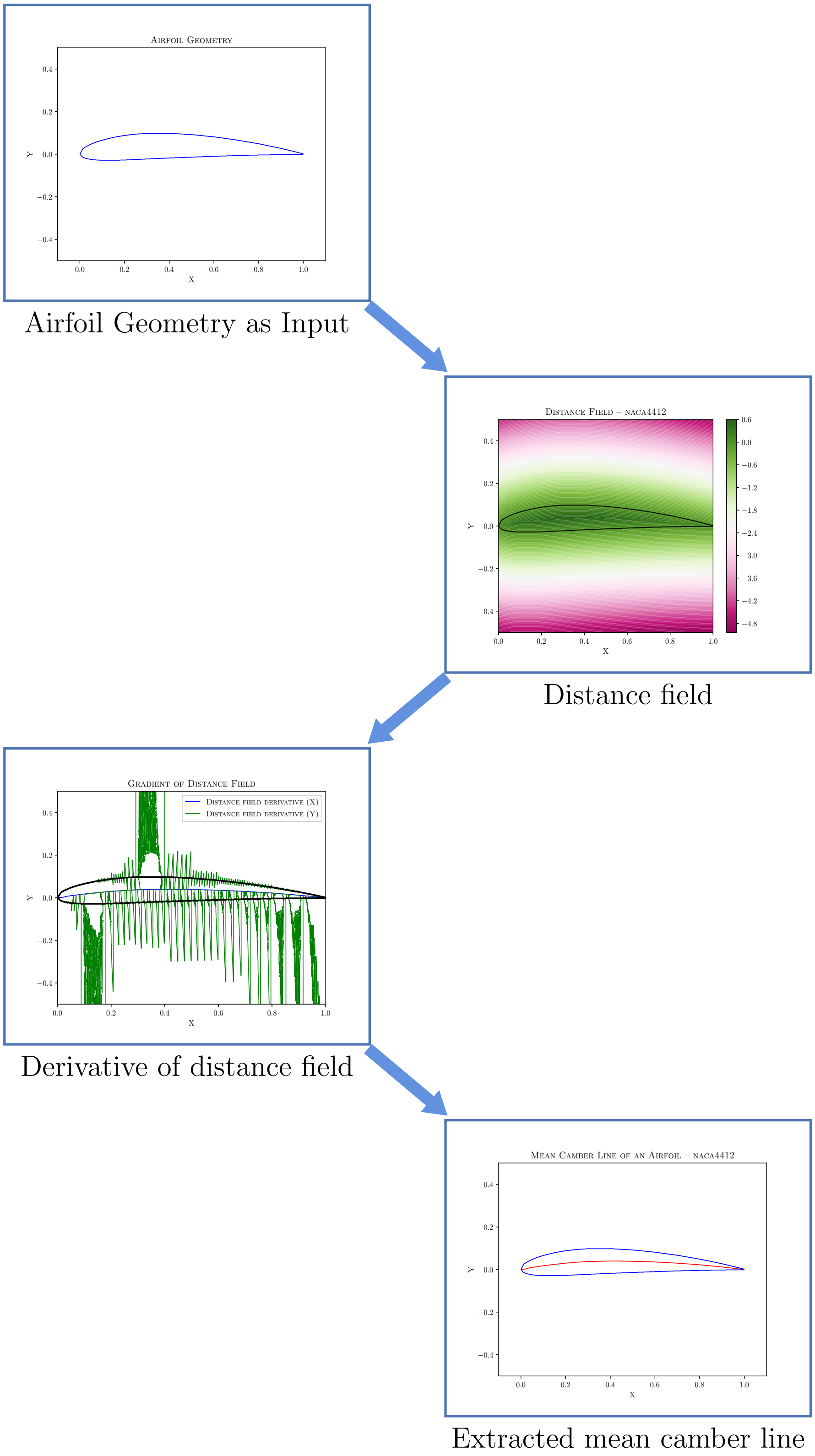
Proof. This argument can be proven easily from the geometric meaning of the derivative. The distance field produced by the fast marching method (or) heat method converges to a singularity at the core of the airfoil geometry. The slope between the various level sets starting from zero level-set to singularity would be maximum along the chordwise direction and be nearly negligible everywhere else inside the airfoil. The cells of the Cartesian mesh, which has a positive derivative value (in magnitude) inside the airfoil along the chordwise direction, would yield the mean camber line. □

5 Numerical Experiments

5.1 Analysis of results

Figure 3 shows an overview of different components of the algorithm. The following steps in the algorithm are important and common to all variants. Numerical experiments were performed for a variety of known airfoil geometries, and the mean camber line is shown in the subsequent sections.

- Reading the geometry and projecting it onto a Cartesian mesh.
- Computing the distance field.
- Computing the gradient of the distance field.
- Extracting the mean camber line from the gradient.



Airfoil Geometry as Input

Distance field

Derivative of distance field

Extracted mean camber line

Figure 3: Different steps of the algorithm

The algorithm has been tested on several airfoil geometries available in the literature, and their mean camber line distribution is shown in Figure 4. It is notable that in most cases, the algorithm is capable of computing the mean camber line directly connecting the leading and trailing edge of the airfoil. Different airfoil contours in Figure 4 are colored in blue and their respective mean camber lines colored in red. The mean camber lines were compared with the ones from existing approaches[17] and are shown to agree. If required, interpolation/curve fitting could be performed to generate splines out of the extracted mean camber lines. Techniques such as arc-length parameterization can be used to parameterize the mean-camber line for topology optimization and other design optimization purposes.

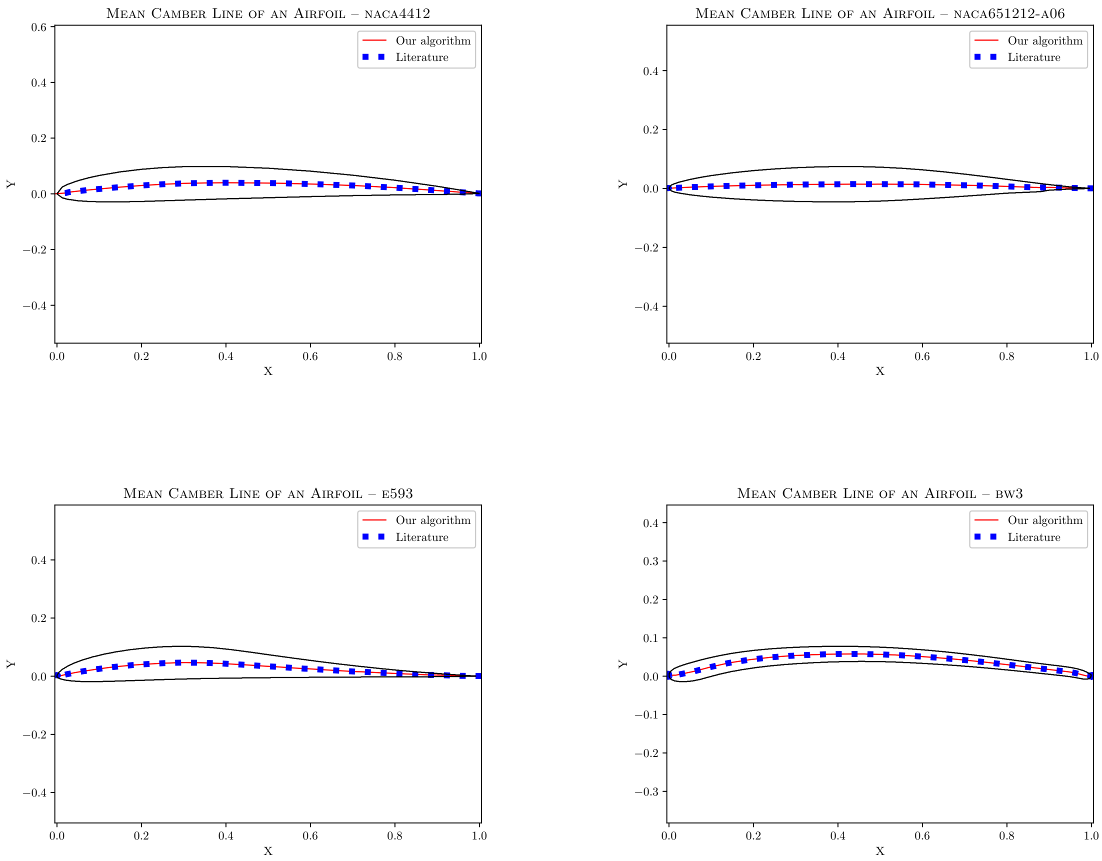


Figure 4: Mean camber line computed on different airfoils and their comparison against existing approaches

5.2 Implementation details

The code for the present investigation is entirely implemented in Python. The code does not have any parallelism. The computation of the distance field and derivative are almost instant and do not require any substantial computational resources. The airfoil geometries used for the analysis of the algorithms were obtained from the public domain.

5.3 Extension of the algorithm for mesh generation applications

In this section, we show how the algorithm can be generalized for different volumetric decomposition purposes, such as block decomposition for structured multiblock grid generation. Similar investigations have been performed using the fast marching method[12]. However, solving the level set equation is not strictly required, as mentioned in the paper, and any code to solve the diffusion equation can yield a so-called biased medial axis[12]. The heat method based algorithm can be very useful here provided a Cartesian mesh can be generated in advance. The algorithm for medial axis extraction only computes the derivative of the distance field in either X or Y direction. A derivative of the distance field in both X and Y direction is required for decomposing a closed polygon into smaller blocks. The algorithm does not produce a purely quadrilateral decomposition. However, topological modifiers can be used for generating a purely quadrilateral decomposition with fewest singularities.

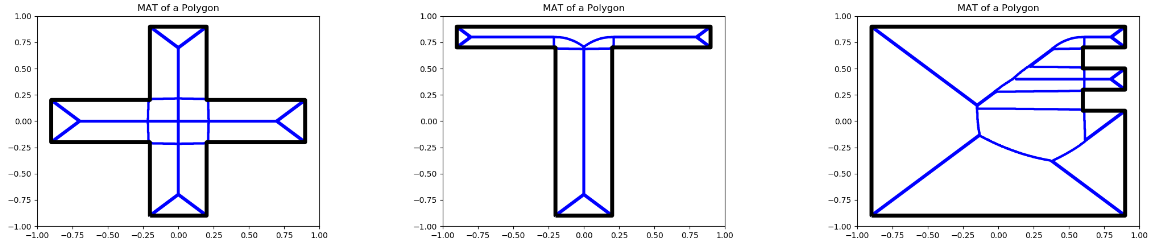


Figure 5: Different polygons and its block decompositions

6 Conclusion

The paper summarizes an algorithm for computing the mean camber line in airfoils. The primary advantage of the algorithms outlined in the paper is that they are fully automatic. Since the algorithms perform computation on a mesh, the mesh created for CFD / FEA simulation can be reused, and the computation can be performed in conjunction with the numerical simulation. Parameterization techniques could be coupled with the algorithm for airfoil/wing optimization purposes. This approach is useful for reverse engineering existing airfoil contours and for quick prototyping of new airfoil contours. It does not require that the airfoil coordinates be evenly distributed on the top and bottom. The computation can be performed instantly and does not require any information about the local curvature of the airfoil. The algorithms proposed in the present investigation can be easily extended to three-dimensional geometries, such as wings and turbine blades. The proposed algorithms in the present investigation would help train neural network-based models, which are gaining much popularity in the field of inverse design. Such models trained with the results from the proposed algorithms in the present investigation could be used to choose an airfoil with a specific amount of camber. Shape similarity algorithms use shape indicators such as skeletons and Reeb graphs of geometries. The proposed algorithm would also serve as a useful shape indicator in the selection of similar airfoil geometries.

References

[1] I.H. Abbott and A.E. von Doenhoff. *Theory of Wing Sections: Including a Summary of Airfoil Data*. Dover Books on Aeronautical Engineering. Dover Publications, 2012.

[2] J.B. Barlow, W.H. Rae, and A. Pope. *Low-Speed Wind Tunnel Testing*. Aerospace engineering, mechanical engineering. Wiley, 1999.

[3] J.D. ANDERSON and J.D. Anderson. *Fundamentals of Aerodynamics*. McGraw-Hill series in aeronautical and aerospace engineering. McGraw-Hill, 2010.

[4] *Aircraft Performance & Design*. McGraw-Hill Education (India) Pvt Limited, 2010.

[5] Wen-long Li, He Xie, Qi-dong Li, Li-ping Zhou, and Zhou-ping Yin. Section curve reconstruction and mean-camber curve extraction of a point-sampled blade surface. *PLOS ONE*, 9(12):1–30, 12 2015.

[6] Ralph Carmichael. *Algorithm for calculating coordinates of cambered NACA airfoils at specified chord locations*. 2001.

[7] Vinothkumar Sekar, Mengqi Zhang, Chang Shu, and Boo Cheong Khoo. Inverse design of airfoil using a deep convolutional neural network. *AIAA Journal*, 57(3):993–1003, 2019.

[8] Yao Zhang, Woong Je Sung, and Dimitri N. Mavris. Application of convolutional neural network to predict airfoil lift coefficient. In *2018 AIAA/ASCE/AHS/ASC Structures, Structural Dynamics, and Materials Conference*, 2018.

[9] Emre Yilmaz and Brian German. A convolutional neural network approach to training predictors for airfoil performance. In *18th AIAA/ISSMO Multidisciplinary Analysis and Optimization Conference*, pages 232–2323, 2017.

[10] Keenan Crane, Clarisse Weischedel, and Max Wardetzky. The heat method for distance computation. *Commun. ACM*, 60(11):90–99, October 2017.

[11] J A Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[12] Hao Xia and Paul G. Tucker. Fast equal and biased distance fields for medial axis transform with meshing in mind. *Applied Mathematical Modelling*, 35(12):5804 – 5819, 2011.

[13] R. Kimmel and G. Sapiro. Shortening three-dimensional curves via two-dimensional flows. *Computers & Mathematics with Applications*, 29(3):49 – 62, 1995.

[14] Haixia DU, Terry YOO, and Hong QIN. Pde-based medial axis extraction and shape manipulation of arbitrary meshes. *Journal of Systems Science and Complexity*, 21(4):609, Nov 2008.

[15] Leif Kobbelt, Stefan Bischoff, Mario Botsch, and Stefan Steinberg. Openmesh: A generic and efficient polygon mesh data structure. 2002.

[16] Zicheng Guo and Richard W. Hall. Parallel thinning with two-subiteration algorithms. *Commun. ACM*, 32(3):359–373, March 1989.

[17] Airfoil Tools. Airfoil tools. `http://airfoiltools.com/`, 2020. [Online; accessed 28-June-2020].