

# The Technique of the Immersed Boundary Method: Application to Solving Shape Optimization Problem

Ling Rao<sup>1</sup>, Hongquan Chen<sup>2</sup>

<sup>1</sup>Department of Mathematics, Nanjing University of Science and Technology, Nanjing, China

<sup>2</sup>College of Aerospace Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, China

Email: lingrao@sina.com.cn, hqchenam@nuaa.edu.cn

**How to cite this paper:** Rao, L. and Chen, H.Q. (2017) The Technique of the Immersed Boundary Method: Application to Solving Shape Optimization Problem. *Journal of Applied Mathematics and Physics*, 5, 329-340.

<https://doi.org/10.4236/jamp.2017.52030>

**Received:** January 22, 2017

**Accepted:** February 12, 2017

**Published:** February 15, 2017

---

## Abstract

We present a numerical method based on genetic algorithm combined with a fictitious domain method for a shape optimization problem governed by an elliptic equation with Dirichlet boundary condition. The technique of the immersed boundary method is incorporated into the framework of the fictitious domain method for solving the state equations. Contrary to the conventional methods, our method does not make use of the finite element discretization with obstacle fitted meshes. It conduces to overcoming difficulties arising from re-meshing operations in the optimization process. The method can lead to a reduction in computational effort and is easily programmable. It is applied to a shape reconstruction problem in the airfoil design. Numerical experiments demonstrate the efficiency of the proposed approach.

## Keywords

Immersed Boundary Method, Genetic Algorithm, Bezier Curve, Elliptic Dirichlet Boundary Value Problem, Shape Optimization Problem

---

## 1. Introduction

The aim of shape optimization is to find a shape  $\Omega$  such that the structure represented by  $\Omega$  behaves in an appropriate way. Usually this goal is realized by the minimization of a suitable cost functional  $J$  over an admissible family  $\mathcal{O}$  of domains, in which all possible candidates are included. Schematically, shape optimization problem can be expressed as follows:

$$[P] \begin{cases} \text{find } \Omega^* \in \mathcal{O} \text{ such that} \\ J(\Omega^*, u(\Omega^*)) = \min_{\Omega \in \mathcal{O}} J(\Omega, u(\Omega)) \end{cases}$$

where  $u(\Omega)$  is the solution of a state equation. In this paper, the state equation is formulated by an elliptic equation with Dirichlet boundary condition. Shape optimization problems have been extensively studied from the mathematical point of view [1]. Our paper deals with their practical aspects. The practical shape optimization problem in this paper is a reconstruction problem, in which the target is to find the shape of airfoil when the pressure distribution on it is given.

Our optimization is performed by using genetic algorithm (GA) [2] [3]. GA is very popular at present for its simplicity and ability to handle large scale problems. It is a global optimization method, and can be used to solve non-smooth optimization problems. Because GA is based on cost functional evaluations. The state equations need to be repeatedly solved on different domain changing during shape optimization computations.

The numerical solution of partial differential equations describing the fluid flow is usually done by performing spatial and temporal discretization. The spatial representation must take into account for the boundaries of the computational domain and then make use of a discrete representation via meshing. Solving the flow around objects with complex shapes may involve extensive meshing work that has to be repeated each time a change in the geometry is needed. Important benefit would be reached if we are able to solve the flow without the need of generating a mesh that fits the shape of the immersed objects. Most flow solvers are based on body-conforming grids (*i.e.* the external boundary and surfaces of immersed bodies are represented by the mesh faces), but there is an increased interest in solution algorithms for non-body-conforming grids. Such methods are presented under a variety of names: immersed boundary (IB), immersed interface, embedded mesh, fictitious domain, all having in common the fact that the spatial discretization is done over a single domain containing both fluid and solid regions and where mesh points are not necessarily located on the fluid-solid interface.

We use the Lagrange multiplier-based fictitious domain method [4] [5] for solving the state equation in the shape optimization problem. The fictitious domain method based approach in the shape optimization problem can be found in [6] [7]. Furthermore in order to avoid costly and sometimes extremely difficult meshing work on body-fitted geometries, we incorporate the technique of the IB method into the framework of the fictitious domain method.

The IB method was initially introduced by Peskin [8] [9] for finite differences applied to fluid-structure interactions. The method received particular attention in recent years [10] [11] [12]. Peskin's IB method was developed for the computer simulation of general problems involving a transient incompressible viscous fluid containing an immersed elastic interface, which may have time-dependent geometry or elastic properties, or both. The IB method is at the same time a mathematical formulation and a numerical scheme. The mathematical formulation is based on the use of Eulerian variables to describe the dynamic of fluid and of Lagrange variables along the moving structure. The force exerted by the struc-

ture on the fluid is taken into account by means of a Dirac delta function constructed according to certain principles. The main idea is to use a regular Eulerian mesh for the fluid dynamics simulation, coupled with a Lagrangian representation of the immersed boundary. The advantage of this method is that the fluid domain can have a simple shape, so that structured grids can be used. The Lagrangian mesh is independent of the Eulerian mesh. The interaction between the fluid and the immersed boundary is modeled by using a well-chosen discrete approximation to the Dirac delta function.

In our approach, all domains  $\Omega \in \mathcal{O}$  are embedded into a fictitious domain  $\hat{\Omega}$  with a simple shape (e.g. a box). It is easy to construct a triangularization of such a domain. The triangularization is fixed, *i.e.*, it does not change during an optimization computation. All computations for solving the state problem are performed in  $\hat{\Omega}$  with the same stiffness matrix, which also does not change and consequently can be computed and stored for ever. The method has the advantage of avoiding the need for re-meshing procedures in the optimization process. The necessary information on the geometry is encoded in an additional variable, contributing only to the right hand side of the state equations. We show that the resulting model can be very efficient for optimization computations. Numerical experiments demonstrate the efficiency of the proposed approach.

The paper is organized as follows. In Section 2, the state equation is presented by an elliptic equation with Dirichlet boundary condition. We describe its algorithm based on boundary Lagrangian fictitious domain method. We incorporate the technique of the immersed boundary method into the framework of the fictitious domain method. In Section 3, we consider the airfoil design for the two-dimension incompressible inviscid uniform flows. We describe the mathematical formulation of a shape optimization problem. In Section 4, we do numerical experiments to show that our proposed method is feasible and effective for solving the shape optimization problem.

## 2. Fictitious Domain Based Approach

On a domain  $\Omega \subset R^2$ , assume problem  $[P]$  with  $u$  being the solution of the following elliptic equation with Dirichlet boundary condition:

$$\begin{aligned} \alpha u - \nu \Delta u &= f && \text{in } \Omega, \\ u &= g_0 && \text{on } \gamma_r, \\ u &= g_1 && \text{on } \Gamma \end{aligned} \quad (1)$$

where domain  $\Omega$  is the exterior of an obstacle  $B$  with boundary  $\gamma_r$  (see **Figure 1**),  $\partial\Omega = \Gamma \cup \gamma_r$ , and  $\Gamma$  is the boundary of a rectangle. The  $\gamma_r$  is defined by the design variable  $r$ ,  $r \in U$ , where  $U$  is the set of admissible design parameters.

We require that the set  $U$  is compact. Coefficient  $\rho, \nu \geq 0$ ,  $g_0 \in H^{\frac{1}{2}}(\gamma_r)$ ,  $g_1 \in H^{\frac{1}{2}}(\Gamma)$ ,  $f \in L^2(\Omega)$ . For simplicity,  $\gamma_r$  is sometimes denoted by  $\gamma$ . Since the solution  $u(x)$ ,  $x \in \Omega$ , depends on parameter  $r$ , it is also denoted by  $u(x, r)$ ,  $x \in \Omega$ ,  $r \in U$  or simply  $u(r)$ ,  $r \in U$ .

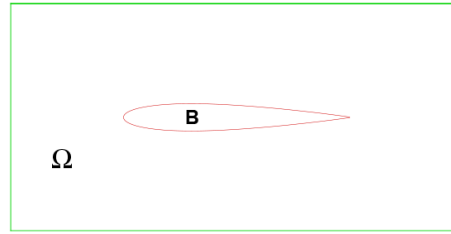


Figure 1. A domain around an obstacle.

According to the boundary Lagrangian fictitious domain method [5], the Dirichlet boundary condition on  $\gamma$  is enforced by using Lagrangian multiplier on the boundary. We can consider the problem in the extended rectangular domain  $\hat{\Omega} : \hat{\Omega} = \bar{B} \cup \bar{\Omega}$ . Problem (1) is equivalent to the following variational problem:

Find  $\bar{u} \in \bar{V}_g, \lambda \in H^{-\frac{1}{2}}(\gamma)$ , such that

$$a_{\hat{\Omega}}(\bar{u}, z) = \int_{\hat{\Omega}} \bar{f} z \, dx + \int_{\gamma} \lambda z \, d\gamma \quad \forall z \in \bar{V}_0, \tag{2}$$

$$\int_{\gamma} \mu (\bar{u} - g_0) \, d\gamma = 0 \quad \forall \mu \in H^{-\frac{1}{2}}(\gamma) \tag{3}$$

where  $\bar{V}_g = \{z \mid z \in H^1(\hat{\Omega}), z = g_1 \text{ on } \Gamma\}$ ,  $\bar{V}_0 = \{z \mid z \in H^1(\hat{\Omega}), z = 0 \text{ on } \Gamma\}$ ,  $a_{\hat{\Omega}}(\bar{u}, z) = \int_{\hat{\Omega}} (\alpha \bar{u} z + \nu \nabla \bar{u} \cdot \nabla z) \, dx$ . The  $\bar{u}$  and  $\bar{f}$  are the extensions of  $u$  and  $f$  in  $\hat{\Omega}$ , respectively, and  $\bar{u}|_{\Omega} = u, \bar{f}|_{\Omega} = f$ .

The problem (2)-(3) can be solved by GCG iterative method. We describe the algorithm presented by [13] as follows:

**Algorithm 0:**

**Step 0:** Initialization.

- Set initial Lagrangian multipliers:  $\lambda^0 \in L^2(\gamma)$  and a number  $\varepsilon \geq 0$  small enough for the convergence criterion.
- Find  $\bar{u}^0 \in \bar{V}_g$  by

$$a_{\hat{\Omega}}(\bar{u}^0, z) = \int_{\hat{\Omega}} \bar{f} z \, dx + \int_{\gamma} \lambda^0 z \, d\gamma, \quad \forall z \in \bar{V}_0. \tag{4}$$

- Calculate  $g^0 \in L^2(\gamma)$  by

$$\int_{\gamma} g^0 \mu \, d\gamma = \int_{\gamma} \mu (\bar{u}^0 - g_0) \, d\gamma, \quad \forall \mu \in L^2(\gamma).$$

- Set the initial descent direction  $w^0 = g^0$ .  
To obtain  $\lambda^{n+1}, \bar{u}^{n+1}, g^{n+1}$  and  $w^{n+1}$  from  $\lambda^n, \bar{u}^n, g^n$  and  $w^n$ , one proceeds as follows:

**Step 1:** Find descent direction.

- Solve  $\bar{u}^n$  by

$$a_{\hat{\Omega}}(\bar{u}^n, z) = \int_{\gamma} w^n z \, d\gamma, \quad \forall z \in \bar{V}_0. \tag{5}$$

- Calculate  $\rho_n$  by

$$\rho_n = \int_{\gamma} |g^n|^2 \, d\gamma / \int_{\gamma} \bar{u}^n w^n \, d\gamma. \tag{6}$$

- Find the new solution by

$$\bar{u}^{n+1} = \bar{u}^n - \rho_n \bar{u}^n,$$

$$\lambda^{n+1} = \lambda^n - \rho_n w^n.$$

- Calculate the new gradient  $g^{n+1} \in L^2(\gamma)$  by

$$\int_{\gamma} g^{n+1} \mu d\gamma = \int_{\gamma} g^n \mu d\gamma - \rho_n \int_{\gamma} \bar{u}^n \mu d\gamma, \quad \forall \mu \in L^2(\gamma).$$

**Step 2:** Construct convergence criterion and update descent direction.

If  $\int_{\gamma} |g^{n+1}|^2 d\gamma / \int_{\gamma} |g^n|^2 d\gamma \leq \varepsilon$ , then take the solution being  $\lambda = \lambda^{n+1}$ ,  $\bar{u} = \bar{u}^{n+1}$ . Otherwise

$$\gamma_n = \int_{\gamma} |g^{n+1}|^2 d\gamma / \int_{\gamma} |g^n|^2 d\gamma,$$

$$w^{n+1} = g^{n+1} + \gamma_n w^n.$$

Set  $n = n + 1$  and return to **Step 1**.

It can be seen from the above algorithm that we need calculate elliptic variational problems (4) and (5). Conventionally we solve them by the finite element method. In the computation procedure of the finite element discretizations, the mesh of the extended domain is constructed from a rectangular triangulated mesh by locally fitting this mesh to the irregular obstacle boundary. The conventional finite element discretizations result the problem in the solution of huge algebraic system of equations and will meet the trouble of computing the boundary integrals. In order to avoid these difficulties and solve more efficiently the extended problem, we will incorporate the technique of the immersed boundary method into the framework of the fictitious domain method. The main idea is to use Dirac delta function to improve the computation procedure of the discretizations. We describe the algorithm presented by [13] as follows.

We construct a regular Eulerian mesh on  $\hat{\Omega}$

$$\hat{\Omega}_k = \{x_{ij} | x_{ij} = (x_0 + ih, y_0 + jh), 0 \leq i, j \leq I\}$$

where  $h$  is the mesh width (for convenience, kept the same both in  $x$ - and in  $y$ -directions). Assume the configuration of the simple closed curve  $\gamma$  is given in a parametric form  $X(s)$ ,  $0 \leq s \leq L$ . The discretization of the boundary  $\gamma$  employs a Lagrangian mesh, represented as a finite collection of Lagrangian points  $X_k$ ,  $0 \leq k \leq J$ , apart from each other by a distance  $\Delta s$ , usually taken as being  $h/2$ .

Let  $\delta(\cdot)$  be a Dirac delta function. In the following calculation procedure,  $\delta$  is approximated by the distribution function  $\delta_h$ . The choice here is given by the product (see [10])

$$\delta_h(\mathbf{x}) = d_h(x) d_h(y)$$

where

$$d_h(z) = \begin{cases} \frac{0.25}{h} \left[ 1 + \cos\left(\frac{\pi z}{2h}\right) \right], & |z| \leq 2h, \\ 0, & |z| > 2h. \end{cases}$$

Using the above Dirac delta function we can transfer the weak forms of the

partial differential equations (4) and (5) to the strong forms and then solve them by Fast Poisson Solvers such as the fast Fourier transform or cyclic reduction. In mathematical view, we need the following Lemma (see [14]).

**Lemma 1.** *Assume that the simple closed curve  $\gamma$ , the configuration of which is given in a parametric form  $X(s)$ ,  $0 \leq s \leq L$  is Lipschitz continuous,  $f \in L^2(0, L)$ . Then  $F$  defined by*

$$F(x) = \int_0^L f(s) \delta(x - X(s)) ds \quad \forall x \in \Omega$$

is a distribution function belonging to  $H^{-1}(\Omega)$  defined as follows: for all  $v \in H_0^1(\Omega)$

$${}_{H^{-1}(\Omega)} \langle F, v \rangle_{H_0^1(\Omega)} = \int_0^L f(s) v(X(s)) ds$$

By Lemma 1, we can write the right hand in (5) as following form:

$$\int_{\gamma} \omega^n z d\gamma = {}_{H^{-1}(\hat{\Omega})} \langle W^n, z \rangle_{H_0^1(\hat{\Omega})}$$

where

$$W^n(\mathbf{x}) = \int_0^L \omega^n(s) \delta(\mathbf{x} - X(s)) ds \quad \forall \mathbf{x} \in \hat{\Omega}. \tag{7}$$

That is,  $\omega^n$  calculated over the Lagrangian points are distributed over the Eulerian points by (7). Thus we can write (5) in the strong form below:

$$\alpha \bar{u}^n - \nu \Delta \bar{u}^n = W^n \text{ in } \hat{\Omega}. \tag{8}$$

In the same way, (4) also can be written in the strong form below:

$$\alpha \bar{u}_0 - \nu \Delta \bar{u}_0 = \bar{f} + R^0 \text{ in } \hat{\Omega}, \tag{9}$$

where

$$R^0(\mathbf{x}) = \int_0^L \lambda^0(s) \delta(\mathbf{x} - X(s)) ds \quad \forall \mathbf{x} \in \hat{\Omega}. \tag{10}$$

Note that (8) and (9) are defined in the rectangular domain  $\hat{\Omega}$ . We can solve their discrete forms by Fast Poisson Equation Solvers such as the fast Fourier transfer or cyclic reduction on Cartesian mesh  $\hat{\Omega}_k$ . In order to calculate  $\int_{\gamma} \bar{u}^n w^n d\gamma$  in (6), we calculate  $\bar{u}^n$  over Lagrangian points  $X_k$ ,  $1 \leq k \leq N$  by  $\bar{u}^n$  over neighboring Eulerian points obtained by (8). We use the following formula due to the property of  $\delta(\cdot)$  (see [10]),

$$\bar{u}^n(X(s)) = \int_{\hat{\Omega}} \bar{u}^n(\mathbf{x}) \delta(\mathbf{x} - X(s)) d\mathbf{x}. \tag{11}$$

The discrete form of (7) is

$$W^n(\mathbf{x}_{ij}) = \sum_k \omega_k^n \delta_h(\mathbf{x}_{ij} - X_k) \Delta s \quad \forall \mathbf{x}_{ij} \in \hat{\Omega}_h. \tag{12}$$

The discrete form of (10) is

$$R^0(\mathbf{x}_{ij}) = \sum_k \lambda_k^0 \delta_h(\mathbf{x}_{ij} - X_k) \Delta s \quad \forall \mathbf{x}_{ij} \in \hat{\Omega}_h. \tag{13}$$

The discrete form of (11) is

$$\bar{u}_k^n = \sum_{ij} \bar{u}_{ij}^n \delta_h(\mathbf{x}_{ij} - X_k) h^2 \quad \forall 1 \leq k \leq N. \tag{14}$$

Based on the above analysis, we have the discrete algorithm of GCG for solving (2)-(3) as follows:

**Algorithm 1:**

**Step 0:** Initialization.

- 1) Distributed  $\lambda^0$  over the Lagrangian points to the Eulerian points by (13).
- 2) Calculate (9) for  $\bar{u}^0$  over the neighboring Eulerian points of the boundary  $\gamma$ .
- 3) Calculate  $\bar{u}^0$  over Lagrangian points by  $\bar{u}^0$  over neighboring Eulerian points by using

$$\bar{u}_k^0 = \sum_{ij} \bar{u}_{ij}^0 \delta_h(\mathbf{x}_{ij} - X_k) h^2 \quad \forall 1 \leq k \leq N. \tag{15}$$

- 4) Calculate  $g^0$  over Lagrangian points by

$$g_k^0 = \bar{u}_k^0 - g_{0k} \quad 1 \leq k \leq N.$$

- 5) Set the initial descent direction  $w_k^0 = g_k^0, 1 \leq n \leq N$ .

To obtain  $\lambda^{n+1}, \bar{u}^{n+1}, g^{n+1}$  and  $w^{n+1}$  from  $\lambda^n, \bar{u}^n, g^n$  and  $w^n$ , one proceeds as follows:

**Step 1:** Find descent direction.

- 1) Distributed  $\omega^n$  over the Lagrangian points to the Eulerian points by (12).
- 2) Solve (8) for  $\bar{u}^n$  over the neighboring Eulerian points of the boundary  $\gamma$ .
- 3) Calculate  $\bar{u}^n$  over Lagrangian points by  $\bar{u}^n$  over neighboring Eulerian points by using (14).
- 4) Calculate  $\rho_n$  by

$$\rho_n = \frac{\sum_k |g_k^n|^2 \Delta s}{\sum_k \bar{u}_k^n w_k^n \Delta s}.$$

- 5) Let

$$\lambda_k^{n+1} = \lambda_k^n - \rho_n w_k^n, \quad 1 \leq n \leq N.$$

- 6) Calculate the new gradient  $g^{n+1}$  by

$$g_k^{n+1} = g_k^n - \rho_k^n \bar{u}_k^n, \quad 1 \leq k \leq N.$$

**Step 2:** Construct convergence criterion and update descent direction. For given  $\varepsilon \geq 0$  small enough, if  $\sum_k |g_k^{n+1}|^2 \Delta s / \sum_k |g_k^0|^2 \Delta s \leq \varepsilon$ , then take  $\lambda = \lambda^{n+1}$  over the Lagrangian points, and calculate its correspondent solution  $\bar{u}$  by using

$$\alpha \bar{u} - \nu \Delta \bar{u} = \bar{f} + R, \quad \text{in } \Omega$$

where

$$R(\mathbf{x}_{ij}) = \sum_k \lambda_k \delta_h(\mathbf{x}_{ij} - X_k) \Delta s \quad \forall x_{ij} \in \Omega_h.$$

Otherwise

$$\gamma_n = \sum_k |g_k^{n+1}|^2 / \sum_k |g_k^n|^2,$$

$$w_k^{n+1} = g_k^{n+1} + \gamma_n w_k^n, \quad 1 \leq k \leq N.$$

Set  $n = n + 1$  and return to **Step 1**.

It can be seen from steps 4, 5, and 6 in **Step 1** that the calculations are done over the Lagrangian points and we need solve (8) only for  $\bar{u}^n$  over the neighboring Eulerian points of the boundary  $\gamma$  when we use FFT. Thus the above approach can lead to a significant reduction in computational effort and memory requirement. It need not make use of the finite element discretizations. The main advantage of the proposed method is that in the optimization process Eulerian mesh on  $\hat{\Omega}$  is fixed unlike in many of the other approaches such as obstacle fitted meshes and solution procedure is often efficient. The proposed method has a simple structure and is easily programmable.

### 3. Shape Optimization Problem

Our shape optimization problem is a shape reconstruction problem. We consider the airfoil design for the two-dimension incompressible inviscid uniform flows. We want to find the shape of an airfoil  $\gamma_0$  when the pressure coefficient  $C_p^0$  on  $\gamma_0$  is known. Below we will give the formulation of this problem.

Suppose the uniform flow around an arbitrary airfoil  $B$  is from the left toward the right and of velocity profile  $U_\infty = 1$ . For the discretization computation, this exterior problem is truncated by introducing an artificial boundary  $\Gamma$  which is a rectangle. The domain  $\Omega$  is the rectangle excluding the airfoil  $B$  with boundary  $\gamma$ , whose leading edge is in origin (see **Figure 1**). By formulating this problem for stream function  $u$ , the corresponding partial differential equation with Dirichlet boundary condition is:

$$\begin{aligned} -\Delta u &= 0 & \text{in } \Omega, \\ u &= 0 & \text{on } \gamma, \\ u &= y & \text{on } \Gamma. \end{aligned} \tag{16}$$

In order to solve the shape optimization problem numerically, the boundary  $\gamma$  must be parametrized using a finite number of design parameters. Let the vector of design parameter  $\mathbf{r} = (a_1, a_2, \dots, a_m) \in R^m$  define the boundary curve  $\gamma$ . One possible way to perform it is to use the Bezier curves; see [15], for example. These Bezier curves are just polynomial functions expressed in the Bernstein polynomial basis. The design parameters (or variables) are given by the nodal coordinates of the control points defining the Bezier curves. Since the design parameter  $\mathbf{r}$  defining uniquely  $\gamma$ , in the following, the cost functional is assumed to be of the form  $J(u, \mathbf{r})$ . We discretize the shape of airfoil  $\gamma$  using one Bezier curve for upper part of airfoil and another Bezier curve for lower part. The  $x$ -coordinates of the control points are evenly spaced between 0 and 1. The control points on leading edge and on the trailing edge are fixed and the other control points are moving in the  $y$ -direction. The  $y$ -coordinates of these control points except the control points on the leading edge and on the trailing edge are chosen to be the design parameters  $\mathbf{r}$ . We have seven design variables in the symmetric case and fourteen design variables in the unsymmetric



case. As a symmetric example, **Figure 2** illustrates the Bezier curve for the upper side of the NACA0012 profile. Nine circles in the figure are Bezier control points.

For given  $\gamma$  corresponding design parameters  $\mathbf{r}$ , pressure coefficient distribution on  $\gamma$  is:

$$C_p(u(x, y), \mathbf{r}) = 1 - \frac{\left(\frac{\partial u}{\partial x}(x, y)\right)^2 + \left(\frac{\partial u}{\partial y}(x, y)\right)^2}{U_\infty^2}, \quad (x, y) \in \gamma,$$

where  $u$  solves (16).

Now, the reconstruction problem reads: find the design parameter  $\mathbf{r}_0$  corresponding  $\gamma_0$  when the pressure coefficient  $C_p^0$  on  $\gamma_0$  is known. We have chosen to formulate our reconstruction problem as a minimization problem, where we minimize an objective function  $J$  measuring the difference between  $C_p(u(\mathbf{x}), \mathbf{r})$  and the desired pressure coefficient  $C_p^0$ . We define objective function to be

$$J(u, \mathbf{r}) = \int_{\mathcal{O}} |C_p(u(\mathbf{x}), \mathbf{r}) - C_p^0(x)| dx. \quad (17)$$

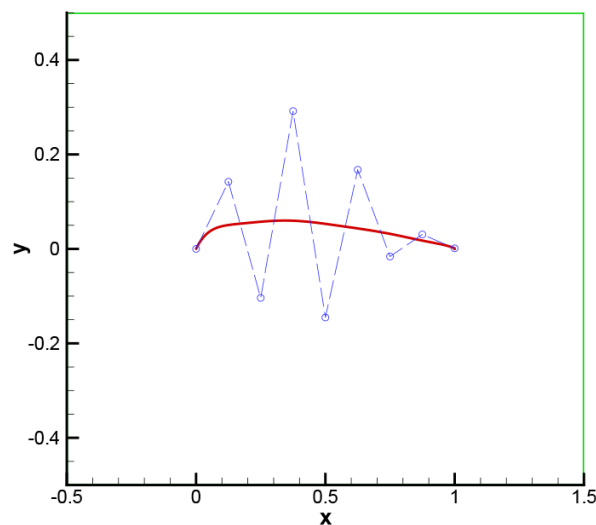
The reconstruction problem in a minimization problem form reads:

$$\begin{aligned} &\text{Find } \mathbf{r}_0 \in U \quad \text{such that} \\ &J(u_0, \mathbf{r}_0) \leq J(u, \mathbf{r}) \quad \forall \mathbf{r} \in U, \end{aligned} \quad (18)$$

where  $U$  is the set of admissible design parameters. We require that  $U$  is compact and  $\mathbf{r} \in U$ , where the design parameter  $\mathbf{r}$  corresponds the boundary  $\gamma$ .

## 4. Numerical Experiments

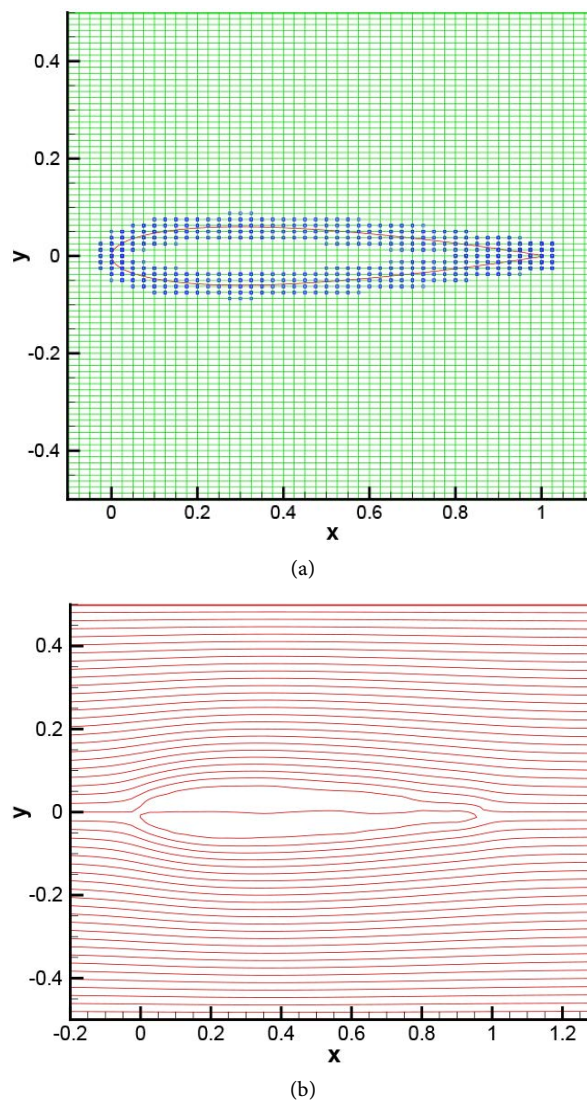
First we do numerical experiments to demonstrate the efficiency of the algorithm in section 2. We use the Algorithm 1 to solve station Equation (16) which simulate the two-dimension incompressible inviscid uniform flows around an arbitrary given airfoil B.



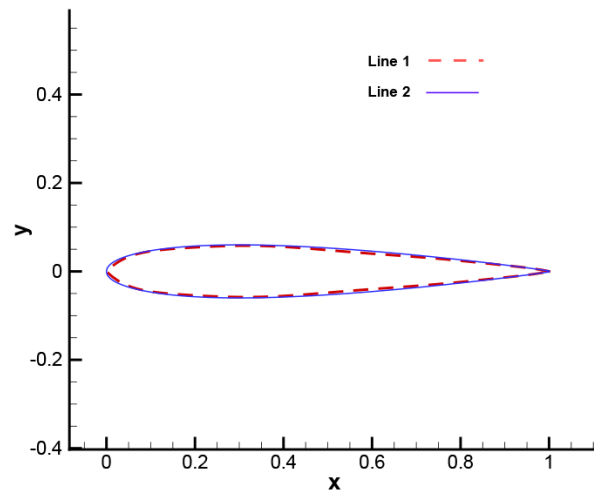
**Figure 2.** The Bezier curve for the upper side of the NACA0012 profile.

We take the fictitious domain  $\bar{\hat{\Omega}} = \bar{B} \cup \bar{\Omega} = [-0.5, 1.5] \times [-0.5, 0.5]$ . The test problem has been solved with rectangular Eulerian mesh  $80 \times 80$  nodes. **Figure 3(a)** shows the Lagrangian grids and Eulerian grids which have been used to calculate solutions for (16). In **Figure 3(b)**, the plots in solid lines represent streamline obtained by the algorithm based on the immersed boundary.

Next we solve the reconstruction problem (18). We take the NACA0012 airfoil as  $\gamma_0$ . As we stated in section 3, the shape of airfoil  $\gamma$  is discretized using one Bezier curve for upper part of airfoil and another Bezier curve for lower part. Since the airfoil is symmetric during the optimization, we can take seven design variables in the optimization process. In order to keep the design acceptable, we have added box constrains for design variables. Hence, there is a lower limit and upper limit for all variables. We require that they satisfy the constraints  $-0.5 \leq a_k \leq 0.5$ ,  $k = 1, 2, \dots, 7$ . The set of admissible designs  $U$  contains all



**Figure 3.** (a) Zoom view of Eulerian mesh, Lagrangian mesh and the used neighboring Eulerian points of the boundary; (b) Streamline visualization: numerical solution is displayed in solid lines.



**Figure 4.** The target design and the design obtained as the result of optimization.

domains which are possible to obtain by using this parametrization and constraints.

We run genetic algorithm (GA) to find seven design variables  $\mathbf{r}_0$  in (18). The major structure of this GA (see for example [6]) is as follows: 1) Evaluation of fitness functions, 2) Roulette wheel selection, 3) Crossover, 4) Mutation. And we used the following parameters: Population size 20, Crossover probability 0.3, Mutation probability 0.2. The above steps were repeated until the stopping criterion was satisfied. In our case the algorithm ended after a constant number of evaluations of the fitness function. The algorithm is based on function evaluations which are implemented by repeatedly solving the state Equations (16). After 2400 times of function evaluations we reached the result of optimization problem (18):

$$\mathbf{r}_0 = (1.266e-1, -7.284e-2, 2.572e-1, -1.261e-1, 1.384e-1, -7.238e-3, 2.788e-2).$$

In **Figure 4**, Line 2 is the target design and Line 1 is the final design obtained as the result of optimization. The euclidean distance between the two lines is  $0.5e-2$ . The computations were run on a personal computer with Intel core CPU @ 2.30 GHz and 2.0 GB RAM. One optimization takes about 5 minutes of CPU time.

## Conclusion

The results of numerical experiments show that our proposed method is feasible and effective for the optimal shape design problem.

## References

- [1] Haslinger, J. and Neittaanmäki, P. (1996) Finite Element Approximation for Optimal Shape, Material and Topology Design. Wiley, New York.
- [2] Goldberg, D.E. and Richardson, J. (1987) Genetic Algorithms with Sharing for Mul-

- timodal Function Optimization. *Genetic Algorithms and their Applications. Proceedings of the Second International Conference on Genetic Algorithms*, 41-49.
- [3] Goldberg, D.E. (1989) *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley.
- [4] Saulev, V.K. (1963) On the Solution of Some Boundary Value Problems on High Performance Computers by Fictitious Domain Method. *Siberian Mathematical Journal*, **4**, 912-925. (In Russian)
- [5] Glowinski, R., Pan, T.W. and Periaux, J. (1994) A Fictitious Domain Method for Dirichlet Problem and Applications. *Computer Methods in Applied Mechanics and Engineering*, **111**, 283-303. [https://doi.org/10.1016/0045-7825\(94\)90135-X](https://doi.org/10.1016/0045-7825(94)90135-X)
- [6] Haslinger, J. and Jedelský, D. (1996) Genetic Algorithms and Fictitious Domain Based Approaches in Shape Optimization. *Structural Optimization*, **12**, 267-264. <https://doi.org/10.1007/BF01197366>
- [7] Raino, A.E., *et al.* (2000) A Moving Mesh Fictitious Domain Approach for Shape Optimization Problems. *Mathematical Modelling and Numerical Analysis*, **34**, 32-34.
- [8] Peskin, C.S. (1977) Numerical Analysis of Blood Flow in the Heart. *Journal of Computational Physics*, **25**, 220-252. [https://doi.org/10.1016/0021-9991\(77\)90100-0](https://doi.org/10.1016/0021-9991(77)90100-0)
- [9] Peskin, C.S. (2002) The Immersed Boundary Method. *Acta Numerica*, **11**, 479-517. <https://doi.org/10.1017/S0962492902000077>
- [10] Enriquez-Remigio, S.A. and Roma, A.M. (2005) Incompressible Flows in Elastic Domains: An Immersed Boundary Method Approach. *Applied Mathematical Modelling*, **29**, 35-54. <https://doi.org/10.1016/j.apm.2004.07.007>
- [11] Lima, A.L.F., Silva, E., *et al.* (2003) Numerical Simulation of Two-Dimensional Flows over a Circular Cylinder Using the Immersed Boundary Method. *Journal of Computational Physics*, **189**, 351-370. [https://doi.org/10.1016/S0021-9991\(03\)00214-6](https://doi.org/10.1016/S0021-9991(03)00214-6)
- [12] Deng, J., Shao, X.M. and Ren, A.L. (2006) A New Modification of the Immersed-Boundary Method for Simulating Flows with Complex Moving Boundaries. *Int. J. Numer. Meth. Fluids*, **52**, 1195-1213. <https://doi.org/10.1002/flid.1237>
- [13] Chen, H. and Rao, L. (2009) The Technique of the Immersed Boundary Method: Applications to the Numerical Solution of Incompressible Flows and Wave Scattering. *Modern Physics Letters B*, **23**, 437-440. <https://doi.org/10.1142/S021798490901859X>
- [14] Daniele, B. and Lucia, G. (2003) A Finite Element Approach for the Immersed Boundary Method. *Computers and Structures*, **81**, 491-501. [https://doi.org/10.1016/S0045-7949\(02\)00404-2](https://doi.org/10.1016/S0045-7949(02)00404-2)
- [15] Farin, G. (1990) *Curves and Surfaces for Computer Aided Geometric Design—A Practical Guide*. 2nd Edition, Academic Press, Boston.

**Submit or recommend next manuscript to SCIRP and we will provide best service for you:**

Accepting pre-submission inquiries through Email, Facebook, LinkedIn, Twitter, etc.

A wide selection of journals (inclusive of 9 subjects, more than 200 journals)

Providing 24-hour high-quality service

User-friendly online submission system

Fair and swift peer-review system

Efficient typesetting and proofreading procedure

Display of the result of downloads and visits, as well as the number of cited articles

Maximum dissemination of your research work

Submit your manuscript at: <http://papersubmission.scirp.org/>

Or contact [jamp@scirp.org](mailto:jamp@scirp.org)