Scientific Research and Essays Vol. 6(27), pp. 5696-5707, 16 November, 2011
DOI: 10.5897/SRE11.487
Available online at http://www.academicjournals.org/SRE
ISSN 1992-2248 ©2011 Academic Journals

Full Length Research Paper

A novel shape optimization method using knot insertion algorithm in B-spline and its application to transonic airfoil design

P. A. Sherar[1], C. P. Thompson[1], B. Xu[2]* and B. Zhong[3]

[1]Department of Process and Systems Engineering, School of Engineering, Cranfield University, Cranfield, MK43 0AL, UK.
[2]PetroChina Pipeline Research and Development Center No. 51, Jinguang RD, Langfang City, 065000, P.R.China.
[3]Department of Aerospace Sciences, School of Engineering, Cranfield University, Cranfield, MK43 0AL, UK.

Accepted 22 August, 2011

A new method using the cubic B-spline curves with nominal uniform knot set to parameterize the geometry is proposed to deal with shape optimization problems. In the method, the control points of the B-spline curves are set to be the design variables in the optimization scheme. A knot insertion algorithm has been introduced in order to keep the geometry unchanged whilst increasing the number of control points at the final optimization stage. The super-reduced idea and the mesh refinement are also employed to deal with the equality constraint and speed up the optimization process. The method is applied to two problems. The first is a 2-dimensional Poisson problem, and the second is an airfoil design problem. In both applications, the results show that the new method is much more efficient when compared with the traditional methods. In the airfoil design problem, the drag of the airfoil has been reduced significantly with much less function calls.

Key words: Shape optimization, B-spline, knot-insertion, control points.

INTRODUCTION

Shape optimization problems or shape design problems can be regarded as a problem of finding a shape which can achieve a given performance while satisfying some constraints. There are two kinds of methods in optimization, one is gradient-based method, and the other is global method, such as the method based on the genetic algorithm (Holst and Pulliam, 2001; Wah and Chen, 2000).

The global method is aimed to find the global optimum. Despite this advantage, the computational cost of this kind of method is usually very expensive because the numbers of the function calls are usually huge. The gradient-based method has the efficiency of finding a local optimum within a finite number of iterations.

With gradient-based method, the gradient information is necessary to minimize the objective function. In shape design problems, to obtain the gradient of the objective function with respect to the design variables, one usually needs the gradient of the state variables with respect to design variables, named sensitivity. Some well-known methods to compute the sensitivity are the finite difference method (Lee and Eyi, 1991; Lee and Eyi, 1993), the complex variable method (Anderson and Nielsen, 2001), automatic differentiation (Bischof et al., 1992; Bischof et al., 1997; Rostaining et al., 1993), sensitivity analysis (Borggaard and Burns, 1994; Burkardt and Gunzburger, 1995), and the adjoint method (Jameson, 1988; Jameson, 1995; Jacob, 1982; Pironneau, 1973; Pironneau, 1974; Ta'asan, 1995).

Among these methods to compute the sensitivity, the adjoint method is particularly useful in the shape optimization where there is usually a large design space. In this method, the computational cost of computing the sensitivity is independent of the number of design

*Corresponding author. E-mail: kjxubo@petrochina.com.cn.

variable while dependent of numbers of constraints. Another good method sharing the same feature is the reverse mode of automatic differentiation.

In this paper, a new method using the cubic B-spline curves with nominal uniform knot set to parameterize the geometry is proposed. The new optimization method is aimed at achieving quadratic convergence of Broyden–Fletcher–Goldfarb–Shanno (BFGS)-based method and superior efficiency as shown in the later content. This would accelerate the optimization process and achieve the better efficiency while kept the accuracy unchanged. With the idea of using knot insertion and the B-spline parameterization, the initial number of control points can be decreased which potentially decrease the number of optimization iterations. Then the knot insertion algorithm is introduced in order to keep the geometry unchanged whilst increasing the number of control points at the late-stage of optimization. The super-reduced idea and the mesh refinement are also employed to convert the constrained optimization problem into an unconstraint optimization problem and speed up the optimization process. Finally the optimum shape is found by the new method.

One of the aims of the method proposed in this paper is to improve the efficiency and enlarge the search space at the final optimization stage in order to achieve a better value of the objective function efficiently.

THE DESCRIPTION OF THE DEVELOPED METHOD

To solve a general shape optimization problem, an optimisation scheme is necessary. The new method, a BFGS-based method, is used because of its quadratic convergence and better efficiency. The selection of the search direction and the update of approximate Hessian matrix follow the same rule as BFGS method (Fletcher, 1987) do:

$$\mathbf{S}_k=-H(\mathbf{x}_k)^{-1}\mathbf{g}(\mathbf{x}_k) \quad (1)$$

$$\mathbf{H_{k+1}} = \mathbf{H_k} + \frac{\mathbf{q_k q_k^T}}{\mathbf{q_k^T p_k}} - \frac{\mathbf{H_k p_k p_k^T H_k}}{\mathbf{p_k^T H_k p_k}} \quad (2)$$

where $\mathbf{p}_k=\mathbf{x}_{k+1}-\mathbf{x}_k$, $\mathbf{q}_k=\mathbf{g}(\mathbf{x}_{k+1})-\mathbf{g}(\mathbf{x}_k)$, and $\mathbf{x}_k$,$\mathbf{g}_k$ denotes the point and the gradient at the k-th iteration, respectively.

Using any gradient-based methods requires user to supply the design variables. In our case, B-spline is chosen to parameterize the shape, therefore the objective function becomes a function of the control points of the B-spline curves, denotes by $F(\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2, \ldots,\boldsymbol{\alpha}_n)$, where $\boldsymbol{\alpha}=(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2,\ldots,\boldsymbol{\alpha}_n)$ is a set of control points to describe the geometry boundary of spatial domain ($\partial\boldsymbol{\Omega}$).

The number of control points, *n* for the initial shape can be chosen as less as possible. The method first uses the control points of B-splines as design variables to generate the shape. Based on the shape, the value of the objective function and the gradient are computed. The optimizer will repeat this process until a certain criteria is reached. The criteria is chosen as:

$$\left\|\mathbf{F_{k+1}} - \mathbf{F_k}\right\| < \eta$$

where η is a sufficient small value to demonstrate whether the value is close to the optimum value. Once this criterion is met, the knot insertion algorithm and the mesh refinement start. The knot insertion algorithm (Farin, 2002) will insert numbers of knots to the current knot set and increase the corresponding numbers of control points while the geometry is kept unchanged. In that case, the numbers of control points are increased from *n* to *m* where *m>n*. Suppose $\alpha = (\alpha_1, \alpha_2, \ldots, \alpha_n)$ is the control points which defines the geometry before knot insertion, while $\hat{\alpha} = (\hat{\alpha}_1, \hat{\alpha}_2, \ldots, \hat{\alpha}_m)$ is the control pints which defines the geometry after knot insertion. Then, the following equation is obtained:

$$\partial\Omega_1(\alpha) = \partial\Omega_2(\hat{\alpha}) \quad m>n \quad (3)$$

The knot insertion algorithm does not change optimization because the objective function is a function of the geometry which is kept unchanged.

$$\mathbf{F}(\partial\Omega_1) = \mathbf{F}(\partial\Omega_2) \quad (4)$$

Introducing knot insertion algorithm is the core idea of the new method. Besides this, the method includes the mesh refinement, super-reduced idea. These will be discussed in the following subsection. The procedure of using the new method will be described thereafter. Finally the advantages of the method will be addressed.

Knot insertion algorithm

A knot insertion algorithm (Farin, 2002) can allow user to insert a knot into a B-spline curve without changing its shape. After knot insertion, the B-spline evaluation ($\mathbf{p}(u)$) function changes from

$$\mathbf{p(u)} = \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{d_i N_i, k(u)} \text{ , with knot set } \mathbf{(u_i)_{i=1}^{n+k}} \quad (5)$$

To

$$\sum_{\mathbf{i=1}}^{\mathbf{n+1}} \mathbf{d_i^1 N_i, k(u)} \text{ , with knot set } \mathbf{(u_i^1)_{i=1}^{n+1+k}} \quad (6)$$

where *k* is the order, *u* is the parameter value, $\mathbf{d}_i$ is the *i*-th control points for the B-spline curve before knot insertion and $\mathbf{d}_i^1$ the *i*-th control points for the B-spline curve after knot insertion.The second knot set differs from the first one in having an additional knot inserted somewhere in the interval $[u_k, u_{n+1}]$.

If a knot û is to be inserted into the knot set, coinciding with the knot u_{p+1} which may already have multiplicity *s* (if this does not occur in the knot sequence then *s* =0), and the new knot set is denoted as:

$$u_i^1 = \begin{cases} u_i & i \le p \\ \hat{u} = u_{p+1} & i = p+1 \\ u_{i-1} & i \ge p+2 \end{cases} \quad (7)$$

then the number of control point has to increase form *n* to *n+1* as the number of the previous knot set increases from *n+k* to *n+k+1*.

Since B-spline has a local modification property, only some of the control points need to change by a linear interpolation of the previous control points; that is:

$$\mathbf{d}_i^1 = \alpha_i^1 \mathbf{d}_i + (1-\alpha_i^1)\mathbf{d}_{i-1} \quad i=p+k+s+2,\ p-k+s+3,\ \ldots,\ p \qquad (8)$$

where:

$$\alpha_i^1 = \left(\hat{\mu} - \mu_i^1\right) / \left(\mu_{i+k}^1 - \mu_i^1\right) = \left(\hat{\mu} - \mu_i\right) / \left(\mu_{i+k-1} - \mu_i\right) .$$

This is the core idea of the knot insertion algorithm. The details of the algorithm can be found in [8]. By repeating the knot insertion algorithm, the control points can be increased as required.

The knot insertion algorithm allows the user add additional control points, i.e. design variables in optimization, to the B-spline curve without changing the geometry. Therefore, it is possible for the user to start the optimization iteration a few control points, and end with a large number of design variables to represent the geometry.

The purpose of using knot insertion algorithm is to decrease the numbers of optimization iterations while obtaining a better value of the objective function.

In shape optimization, how to represent the geometry is an important issue. Hicks, Henne, Vanderplaats (Hicks and Henne, 1977; Hicks and Vanderplaats, 1977) use shape function to parameterize shape changes. Jameson employs the coordinates of every surface point as design variables, which makes a large design space. This approach removes the geometric model from the optimization loop; however, it may lead to discontinuities on gradient which can be eliminated by a smoothing technique. An example of choosing B-splines to parameterize the geometry is given by Anderson and Venkatakrishnan (1997). Once the shape is parameterized, the next issue is to discretize the geometry as shown in the next subsection.

Mesh refinement

The purpose of using mesh refinement is to make the optimizer do most of iterations on the coarse grid and less on the refined grid. To refine the mesh, there are basically two choices: namely local refinement, and global refinement. Since the global refinement may introduce inefficiency, local refinement is used in this paper.

Now the questions are how coarse the initial mesh and how refine the final mesh should be. To answer the first question, let us recall the Taylor's expansion:

$$\mathbf{F}(\mathbf{x}_k + \beta_k \mathbf{s}_k) \approx \mathbf{F}(\mathbf{x}_k) + \beta_k [\mathbf{g}]^T \mathbf{s}_k + \frac{1}{2}\beta_k^2 [\mathbf{s}_k]^T \mathbf{H}\mathbf{s}_k \qquad (9)$$

Where β_k, a positive real number, is the step size in the search direction s_k, *g and H* denotes the analytic gradient and analytic Hessian matrix respectively, $\mathbf{s}_k = -H(\mathbf{x}_k)^{-1}\mathbf{g}(\mathbf{x}_k)$.

In order to obtain a descent in the value of objective function, the following condition: $[\mathbf{g}]^T \cdot \mathbf{s}_k < 0$ can be adopted which is a sufficient condition, that is, if:

$$[\mathbf{g}]^T \cdot H(\mathbf{x}_k)^{-1}\mathbf{g} <= 0 \text{ where } \mathbf{g}^h = \mathbf{g}(\mathbf{x}_k) \qquad (10)$$

Is satisfied, then it is guaranteed that the optimizer will be able to find a smaller than value than $F(\mathbf{x}_k)$.

In the above formula, $\mathbf{g}^h$ represents the approximated gradient in a mesh where the mesh size is *h*, and $H(\mathbf{x}_k)$ is an approximated Hessian matrix by BFGS update, which is a positive definitely matrix and the initial approximation for such a matrix is usually the unit matrix.

To answer the second question, one can use the error analysis to find out how the error (mainly discretization error) influences the value of *F*, or use the consist gradient theory (Mohammadi and Pironneau, 2001; Polak, 1984). In this paper, the error analysis is used.

Super-reduced idea for computing gradients

Shape optimization problem is usually a PDE-constrained optimization problem in CFD area with or without other constraints. For a shape optimization with one equality constraint,

$$\min_{\boldsymbol{\alpha}} \mathbf{F}(\alpha, \Phi(\alpha)) \quad \text{subject to } \boldsymbol{c}(\boldsymbol{\alpha})=0 \qquad (11)$$

where $\boldsymbol{\alpha}=[\alpha_1, \alpha_2, \ldots, \alpha_n]^T$ is a column vector which consists of *n* numbers of design variables, $\Phi(\boldsymbol{\alpha})$ represents the partial differential equation, and $\boldsymbol{c}(\boldsymbol{\alpha})$ is the constraint.

It is obvious that solving an unconstrained optimization problem is much easier than solving a corresponding constrained optimization problem. The super-reduced idea, see (Xie, 2002), is introduced to convert the constrained optimization problem into an unconstrained optimization problem. The idea is to solve the PDE constraint firstly in each optimization in order to reduce the aforementioned optimization system. The second step is to solve the other constraint to super reduce the system.

In the new method, a design variable, say $\boldsymbol{\alpha}_i$ is chosen, and this design variable, that is:

$$\boldsymbol{\alpha}_i = E(\boldsymbol{\alpha}_1, \boldsymbol{\alpha}_2, \ldots, \boldsymbol{\alpha}_{i-1}, \boldsymbol{\alpha}_{i+1}, \ldots, \boldsymbol{\alpha}_n) \qquad (12)$$

where $\boldsymbol{\alpha}_i$ is a dependent design variable and all other design variables are independent.

Therefore, the constrained optimization problem (Equation 11) becomes the following unconstrained optimization problem:

$$\min_{\boldsymbol{\alpha}} \mathbf{F}(\alpha_1, \alpha_2, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_n)$$
$$\boldsymbol{\alpha} = [\alpha_1, \alpha_2, \ldots, \alpha_{i-1}, \alpha_{i+1}, \ldots, \alpha_n]^T \qquad (13)$$

provided that the partial differential equation is solved in each optimization iteration.

The solution of the above system is equivalent to the solution of the Lagrange system stated below:

$$\min L(\alpha_1, \alpha_2, \ldots, \alpha_n, \lambda_1) = \min \mathbf{F}(\alpha_1, \alpha_2, \ldots, \alpha_n) - \lambda_1 c(\alpha_1, \alpha_2, \ldots, \alpha_n) \qquad (14)$$

as long as the selected dependent variable is uniquely defined by other design variables. The mathematic proof for this argument is simple and given in Xu (2007).

The procedure for the developed optimization method

The procedure of the new method can be summarized as follows:

(1) Start from the initial design point $\mathbf{x}_0$, the inverse matrix of approximated Hessian matrix, B_0.

(2) Evaluate the objective function, and evaluate/update the gradient.

(3) Set the search direction $\boldsymbol{s}_k = -B_k \boldsymbol{g}_k$, Check the magnitude of the search direction $\boldsymbol{s}_k$ to avoid a potentially self-intersected shape. Use the line search method to find the step size β and keep the constraint constant all the times.

(4) Update the design point $\mathbf{x}_k \longrightarrow \mathbf{x}_{k+1}$. Evaluate the gradient at the new point $\mathbf{x}_{k+1}$.
(5) Decide whether the mesh refinement and the knot insertion algorithm should be employed.
(6) If the knot insertion algorithm is required, produce the new design point X_k, compute the value of the objective function and the gradient.
(7) Go back to step 2 if the termination criteria are not satisfied, otherwise output the result.

The advantages of the developed optimization method

The advantages of the developed optimization method are as follows:

Using this method, an optimum shape with a good value of the objective function can be found efficiently. Generally using less design variables, the optimizer will also require less iteration to complete. However the value of the objective function obtained may not be very good, particularly in the case where the value of the objective function is sensitive to the shape change. On the other hand, using a large number of design variables, the optimizer may require a large number of iterations to find an optimum shape. The new method can find a good value of the objective function with fewer initial design variables and less iteration.

The new method shares the quadratic convergence feature of the BFGS method. Furthermore, its efficiency for finding an optimum shape is better than the BFGS method as shown later.

It makes the BFGS based methods possible for problems with a large number of design variables. Generally the BFGS method is not a good choice in this case because of its computational cost on the matrix operation. The new method will limit most of the matrix operations to those of small dimension.

Application to 2-dimensional Poisson problem

Problem description

For the Poisson equation stated below

$$\nabla^2\Phi = -1.0 \text{ in } \Omega \tag{15}$$

with the boundary condition

Φ=0 on $\partial\Omega$,

the purpose is to find an optimum shape to maximise the maximum value in the domain (Ω) :

$$\max_{\partial\Omega} \max_{\Omega} \Phi \tag{16}$$

subject to a geometric constraint: area of the domain, A(Ω) is a constant.
After reorganizing, the objective function can be chosen as:

$$\mathbf{F} = \max_{\Omega} \Phi \tag{17}$$

Technique requirements

To solve this problem, the items below are necessary:

Representation of the geometry

Two cubic B-spline curves with nominal uniform knot set are chosen to parameterize the geometry. Therefore the objective function becomes a function of the control points and the control points become the design variables in this optimization scheme.

Mesh generator

A mesh generator can produce a mesh file for input geometry. The BAMG is chosen as the mesh generator (INRIA, BAMG v0.68, http://www-rocq1.inria.fr/gamma/cdrom/www/bamg/eng.htm, latest access time 20/05/2005).

Poisson solver

A PDE solver to solve the Poisson equation is needed. The solver in use is CRANDNS, which is a parallelized FORTRAN code (Becker and Thompson, 2005).

Optimization scheme

Geometry representation

Two B-spline curves are chosen to parameterize the upper geometry and the lower geometry respectively. In the paper, the distance between the first and the last control point is fixed since otherwise, it is quite possible to get an open shape or self-intersected shape. The expression for each of the two B-splines curves is given by the equation below:

$$\mathbf{y} = \sum_{i=1}^{n} \mathbf{y}_i \mathbf{N}_{i,k}(\mathbf{u}) \quad \mathbf{0} \le \mathbf{u} \le \mathbf{1} \tag{18}$$

where y_i is the i-th control point in the B-spline curve and the B-spline basis function is defined as:

$N_{i,1}(u)$=1 if $\mathbf{u}_i \le \mathbf{u} \le \mathbf{u}_{i+1}$

$N_{i,1}(u)$=0 otherwise

$$\mathbf{N}_{i,k}(\mathbf{u}) = \frac{\mathbf{u}-\mathbf{u}_i}{\mathbf{u}_{i+k-1}-\mathbf{u}_i}\mathbf{N}_{i,k-1}(\mathbf{u}) + \frac{\mathbf{u}_{i+k}-\mathbf{u}}{\mathbf{u}_{i+k}-\mathbf{u}_{i+1}}\mathbf{N}_{i+1,k-1}(\mathbf{u}) \tag{19}$$

Suppose the knot set below is chosen:

$$\underbrace{(0,0,\ldots,0}_{k}, \underbrace{\frac{1}{n-k+1}, \frac{2}{n-k+1}, \frac{3}{n-k+1} \ldots, \frac{n-k}{n-k+1}}_{n-k}, \underbrace{1,1,\ldots,1)}_{k}$$

which is a nominal uniform knot set. In this case, the area of the shape is given by the equation below:

$$\begin{aligned} \mathbf{A}(\Gamma) &= \mathbf{A}(\Gamma^U) - \mathbf{A}(\Gamma^L) \\ \mathbf{A}(\Gamma^U) &= \int_0^1 (\mathbf{y}^U(\mathbf{x}))\mathbf{dx} \\ \mathbf{A}(\Gamma^L) &= \int_0^1 (\mathbf{y}^L(\mathbf{x}))\mathbf{dx} \end{aligned} \tag{20}$$

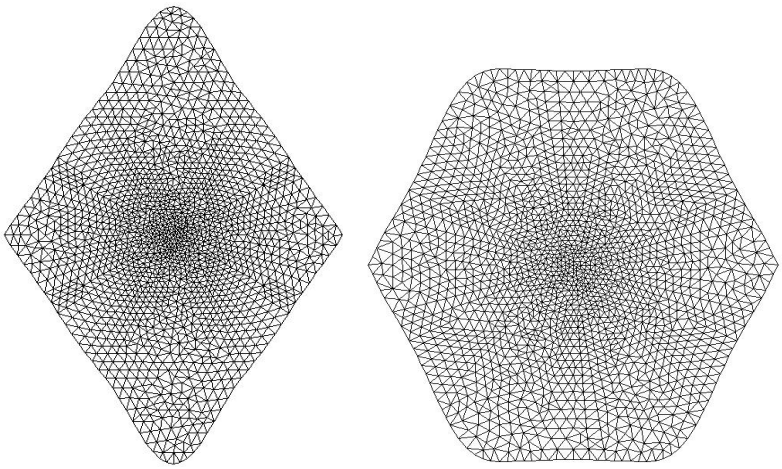


Figure 1. Initial shapes.

provided that the upper geometry Γ^U and the lower geometry Γ^L are not crossed. In the above equations, y^U, y^L denote the upper curve and the lower curve respectively, A^U, A^L denote the area of the upper part and the lower part, respectively.

The derivative of the area with respect to the control points in the upper curve is given by the equation below:

$$\frac{\partial A}{\partial y_i} = \int_0^1 Ni,k(u)du \quad (21)$$

If a cubic B-spline curve with nominal uniform knot set is used, the gradient of the area with respect to the control points in the upper B-spline curve is given by

$$\frac{\partial A}{\partial y_i} = \frac{u_{i+4} - u_i}{4} \quad i=1,2,\ldots,n \quad (22)$$

For the control points in the lower B-spline curve, the partial direviative is given by:

$$\frac{\partial A}{\partial y_i} = -\frac{u_{i+4} - u_i}{4} \quad i=1,2,\ldots,n \quad (23)$$

and the area of the whole domain is stated as below:

$$A = \sum_{i=1}^{n} (y_i \frac{u_{i+4} - u_i}{4}) - \sum_{j=1}^{m} (y_j \frac{u_{j+4} - u_j}{4}) \quad (24)$$

where n, m is the number of control points in the upper curve, and lower curve, respectively.

The finite difference method to compute the super-reduced gradient

The geometry $\partial\Omega$ is a function of design variables, i.e., control points of the B-spline curves (specifically $\partial\Omega$ $(y_1,y_2,\ldots,y_n)$ are the control points of the B-spline curves). The area of the domain needs to be fixed, which is obviously a linear constraint, see Equation 24. Therefore a clever choice is to move one of the control points in order to keep the area constant rather than deal with the linear constrained optimization problem. Now the problem is redefined as:

$$F=F(y_1,y_2,\ldots,y_{n-1}) \quad (25)$$

with y_n eliminated in each optimization iteration.

The super-reduced gradient is computed by using the following finite difference method so that the constraint is kept unchanged:

$$\frac{\partial F}{\partial y_i} \approx \frac{F(y_1,y_2,\ldots,y_{i-1},y_i+\varepsilon,y_{i+1},\ldots,y_n+\delta)}{\varepsilon} - \frac{F(y_1,y_2,y_3\ldots,y_n}{\varepsilon} \quad i=1,2,\ldots,n-1 \quad (26)$$

$$\frac{\partial F}{\partial y_n} = 0$$

where δ is given by solving the following constraints:

$A(\Omega(y_1,y_2,\ldots,y_{i-1},y_i+\varepsilon,y_{i+1},\ldots,y_n+\delta))=k$ k is a constant

RESULTS

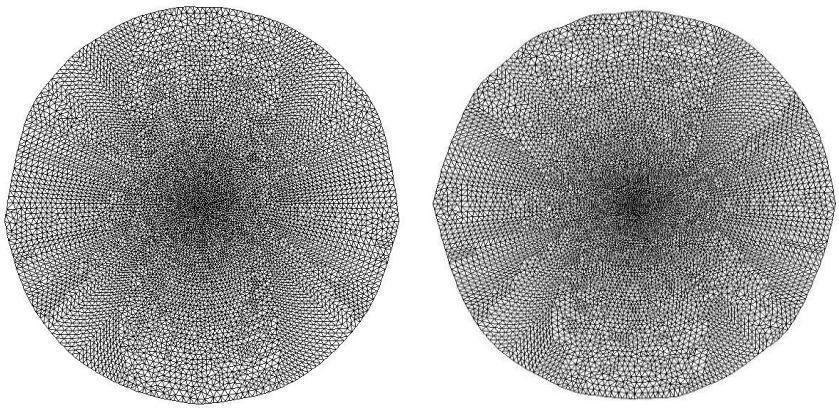
The developed method has been used for varying initial shapes, varying accuracy level placed on the magnitude of gradient vector, different number of control points, and different objective functions. In all cases tested, the solution converges, the optimum shape is found.

Some optimum shapes are obtained without using the mesh refinement. One of them is hereby presented. The optimizer starts from the intial shape 1 (Figure 1.a).

After 103 optimization iterations, it finds the optimum shape (Figure 4) and the optimal value of the

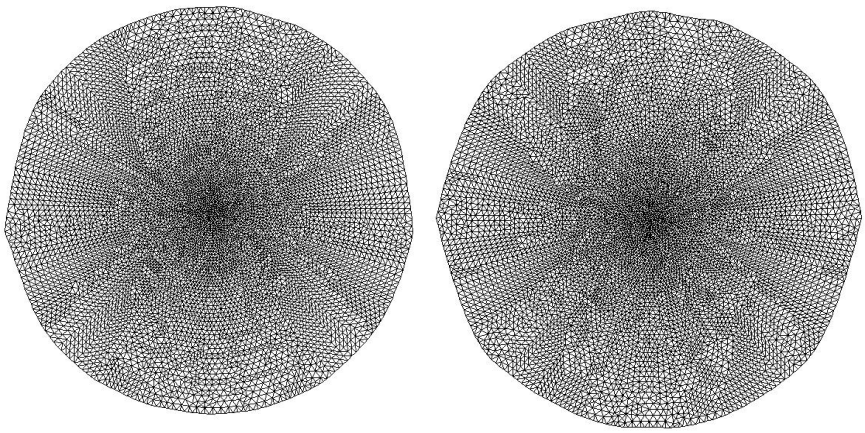
Table 1. Results of shape optimization problem using mesh refinement.

Type	Starting guess	Tolerance	Its	Value of F	Final shape	CPU (h)
Without KI	Figure 1a	2.5e-3	94	-0.062323	Figure 2a	2.7266
With KI	Figure 1a	2.5e-3	37	-0.062323	Figure 3a	0.82251
Without KI	Figure 1b	3.5e-3	48	-0.062280	Figure 2b	1.3596
With KI	Figure 1b	3.5e-3	18	-0.062302	Figure 3b	0.53511
Without KI	4-star shape	4.5e-3	62	-0.062110	N/S	1.4697
With KI	4-star shape	4.5e-3	33	-0.062072	N/S	0.70719



(a) (b)

Figure 2. Optimum shapes produced without using the knot insertion algorithm.



(a) (b)

Figure 3. Optimum shapes produced with the new method.

objective function which is -0.062202.

All the data in Table 1 are produced with mesh refinement integrated. In Table 1, it denotes the iteration numbers, KI is an abbreviation of knot insertion, N/S means not shown.

For the different initial shapes as shown in Table 1 and pictures, the new method starts from 22 control points, and the initial mesh has about 3300 triangles. When the knot insertion algorithm is required, additional knots are uniformly inserted into the existing knot set, and the

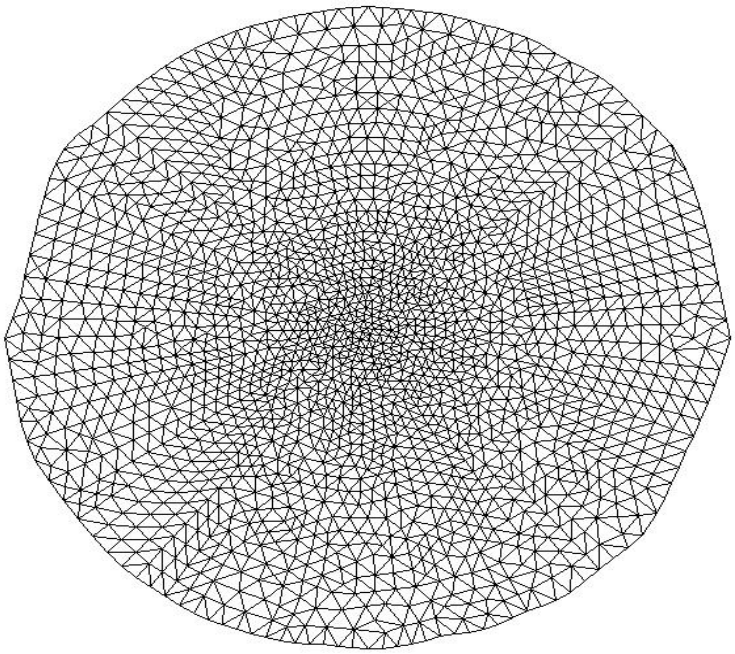


Figure 4. Optimum shape for initial shape 1.a without using mesh refinement.

number of control points becomes 40. When the mesh refinement starts, a fine mesh is produced. It has about 13,000 triangles.

The idea of the method is presented previously. There are different ways to fulfil the idea. For example, one can locally insert knot to achieve the requirements on the curvature instead of inserting knots uniformly.

As shown in Table 1, with the knot insertion algorithm, the number of iterations has been decreased in various cases. In the tested cases, the current method saves CPU time by about 69.8, 60.6, and 51.9%, respectively.

Comparing with BFGS

The BFGS method is regarded as one of the best methods in optimization, hence, it is chosen for comparison with our newly developed method. The BFGS method used in the paper has been enhanced with the following modules: super-reduced idea, the geometry model, and the mesh refinement in order to deal with the constrained shape optimization problems. The convergence history comparison between the new method and the BFGS method is shown Figures 5 and 6 where the initial shapes and the way of mesh refinement used are exactly the same in both methods.

In Figure 6, the value of the objective function in the 7-th iteration actually increases instead of decreases. This is because the shape is going to be self-intersected, and the algorithm produces a new point to replace it.

Application for airfoil design

The purpose of this application is to find optimum shapes to minimize the drag of transonic airfoils subject to some constraints of design conditions. The constraints are constraint for keeping the lift coefficient constant and constraint for keeping the maximum thickness constant;

The geometry representation

In this application, it follows Zhong and Qiao's work (Zhong and Qiao, 1994). In their work, the shape of the airfoil is formed by a base shape plus shape change where the shape change is parameterized by the coefficients times corresponding shape functions. Therefore, the shape of the airfoil is defined by the following equations.

$$\mathbf{Y}(\Gamma_{\mathbf{U}}) = \mathbf{Y}(\Gamma_{\mathbf{ini}}) + \sum_{\mathbf{i=1}}^{\mathbf{n}} \mathbf{a_i f_i(X)} \tag{27}$$

$$\mathbf{Y}(\Gamma_{\mathbf{L}}) = \mathbf{Y}(\Gamma_{\mathbf{ini}}) + \sum_{\mathbf{i=1}}^{\mathbf{m}} \mathbf{b_i g_i(X)} \tag{28}$$

Where $Y(\Gamma_U)$, $Y(\Gamma_L)$ represent the y coordinates of the upper geometry and the lower geometry of the airfoil respectively, $Y(\Gamma_{ini})$ is the y coordinates of the base airfoil,

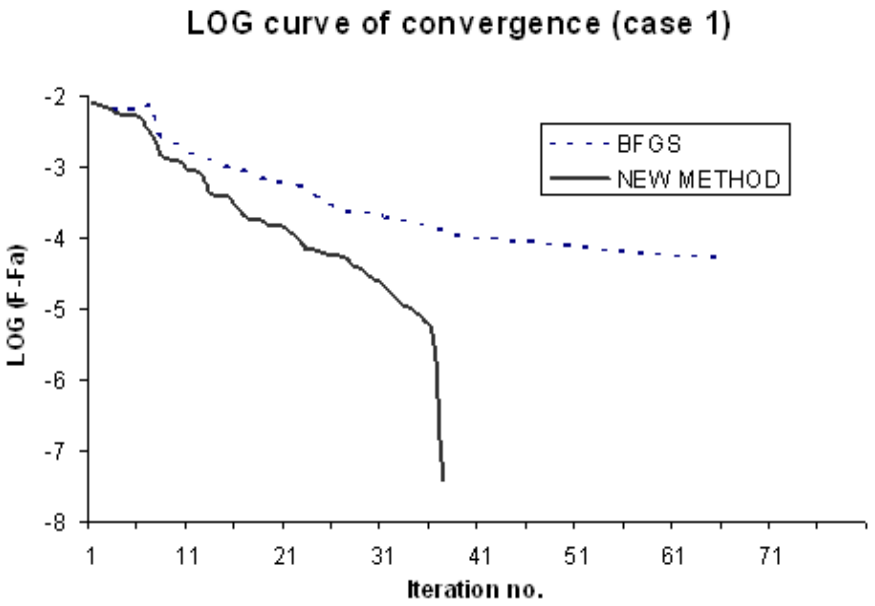


Figure 5. Performance comparison with BFGS method (without using mesh refinement).

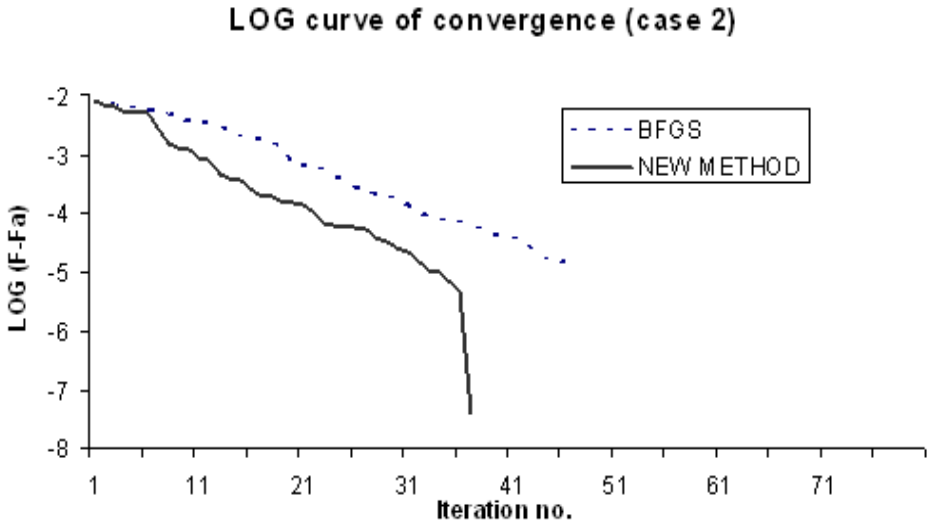


Figure 6. Performance comparison with BFGS method (using mesh refinement).

or initial airfoil, $f_i(X)$, $g_i(X)$ are shape functions.

The shape functions are chosen based on Hicks and Vanderplaats's work (Hicks and Vanderplaats, 1977), and given by the following equations:

$$\mathbf{f_i(X) = \frac{X^{n_i}(1-X)}{e^{m_i \times X}}} \qquad i=1 \qquad (29)$$

$$\mathbf{f_i(X) = \sin^2(\pi X^{n_i})} \qquad i=2,3,4,5 \qquad (30)$$

in the design case of RAE2822.In the above equations, for i=1, m_1=20,n_1=0.25 is selected, and it's peak occurs at about 1.2% chord position. For i=2,3,4,5, n_i=log 0.5/log X_i isused to place the peak at X_i, which occurs at 0.2,0.4,0.6.0.8 respectively. In the design case of NACA 0012, eight shape functions are chosen to represent the shape change, which will be described in more details later.

For the new method, the shape changes are parameterized by cubic B-spline with a nominal uniform

Table 2. Performance comparison for the two methods in design case of RAE2822

Optimization method	Cd	Initial Cd	NFC	CPUPFC
EXTREM	0.0087	0.0116	599	0.22246
New Method	0.0077	0.0116	177	0.22418

knot set. The geometry of the airfoil is given by the following equation.

$$\mathbf{Y}(\Gamma_U) = \mathbf{Y}(\Gamma_{ini}) + \sum_{i=1}^{n} \mathbf{P_i N_i(X)} \quad (31)$$

$$\mathbf{Y}(\Gamma_L) = \mathbf{Y}(\Gamma_{ini}) + \sum_{i=1}^{m} \mathbf{Q_i N_i(X)} \quad (32)$$

where P_i, Q_i are the control points of the B-spline and $N_i(X)$ are the base functions of the B-spline. In these 2-dimensional design cases, P_i, Q_i can be regarded as the value of the *y* co-ordinates of the control points and the value of the *x* co-ordinates of the control points is the parameter.

The flow solver and the optimization algorithm

The transonic airfoil analysis code used in this paper is the NPUFOIL program developed by Qiao (Zhong and Qiao, 1994). The program is based on a combination of the full potential equation solver and the boundary layer flow analysis. The flow and the aerodynamic characteristics are computed via viscous/in viscid iteration.

In Zhong's paper (Zhong and Qiao, 1994), the direct search EXTREM (Lee and Eyi, 1991) optimization method developed by Jacob is used. EXTREM method is also employed in this paper and the results obtained are compared with those achieved by using the new optimization algorithm presented in previous content in two different design cases in the following sections. One design case is based on using RAE 2822 as the initial airfoil, the other uses NACA 0012.

RAE2822

In this design case, the initial airfoil RAE2822 is chosen. The purpose is to get an optimum shape under the design conditions: Mach number M_a=0.75, and Reynolds number Re=2.0×10^6 while the lift coefficient Cl is kept to be 0.7.

By using the EXTREM method, the optimum solution of the shape coefficients is:

a_1=0.39521 a_2=-0.26958 a_3=-0.030086 a_4=0.23385 a_5=0.20672

For the new method, then the optimum solution of the control points is obtained as follows:

P1=-0.29105 P2=-0.42379 P3=-0.48731 P4=-0.36079 P5=-0.32293 P6=0.36791

P7=0.30644 P8=0.091182 P9=0.081497.

The performance comparison for the two methods is given in Table 2:

From Table 2, we can see that using the EXTREM method, the drag coefficient is reduced by 25%, whereas using the new method, the drag coefficient is reduced by about 34%. Moreover, the new method uses only 177 function calls while the EXTREM method requires 599 function calls. The total CPU time used is reduced by around 70%.

The comparison of the optimum shape and pressure distribution are shown in Figures 7 and 8, respectively. Figure 7 demonstrates the change of the geometry from the initial to the optimal and Figure 8 shows that the strength of the shock is very much weakened which results in a significant reduction in the shock wave drag. Hence the total drag is also reduced.

NACA0012

In this design case, the symmetric airfoil NACA0012 is chosen as the initial geometry. The lift coefficient Cl=0.5, Mach number M_a=0.77, and the Reynolds number Re=6.5×10^6 are kept unchanged. When the EXTREM method is employed, 8 shape functions are used to parameterize the geometry whilst in the new optimization algorithm, five free control points are used to parameterize the geometry and at the late stage nine free control points is used to generate a desired geometry. The shape functions chosen are given by the following equations

$$\mathbf{f_i(X) = \sin^4(\pi^{n_i})} \quad i=1, 2, 3, 4, 5 \quad (33)$$

$$\mathbf{f_i(X) = \sin^4(\pi(1-X)^{n_i})} \quad i=6, 7, 8 \quad (34)$$

In the above equations, for i=1, 2, 3, 4, 5 n_i=log 0.5/log X_i isused to place the peak at X_i, while n_i are 0.06, 0.13, 0.2, 0.4, 0.6 respectively. For i=6, 7, 8, assigning $n_6=n_3$, $n_7=n_2$, $n_8=n_1$ makes the peak occur at 0.8, 0.87, 0.94.

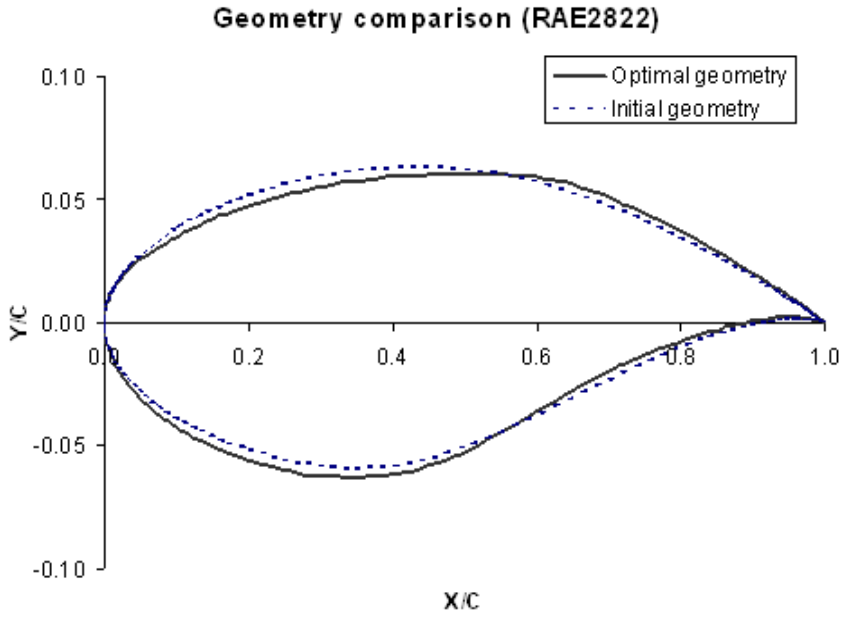


Figure 7. Initial and optimal geometry for RAE 2822.

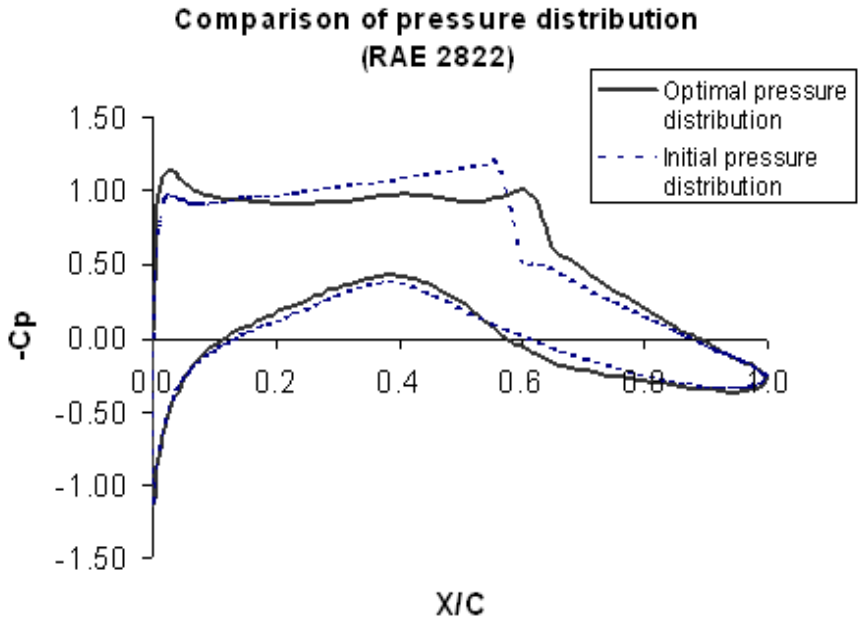


Figure 8. Initial and optimal pressure distribution for RAE 2822.

By using the EXTREM method, the optimum solution of the shape coefficients is obtained as follows:

a_1=-0.045544 a_2=-0.21081 a_3=-0.10340 a_4=0.16878 a_5=0.42756

a_6=1.6056 a_7=0.74775 a_8=0.79883

For the new method, the geometry is first parameterized by five free control points, and finally it reaches the optimum solution which has nine free control points and the drag is reduced from 0.0182 to 0.0070.The optimum solution of the control point is:

P1=-0.571028 P2=-0.70975 P3=-0.41511 P4=0.13373 P5=0.78084 P6=1.3277

P7=0.98959 P8=0.56642 P9=0.13606

The performance comparison for the two methods is given by Table 3:

In Table 3, NFC denotes number of function calls;

Table 3. Performance comparison for the two methods in design case of NACA0012.

Optimization method	Cd	Iteration number	NFC	CPUPFC
EXTREM	0.0070	N/A	408	0.22246
New Method	0.0070	20	233	0.22418

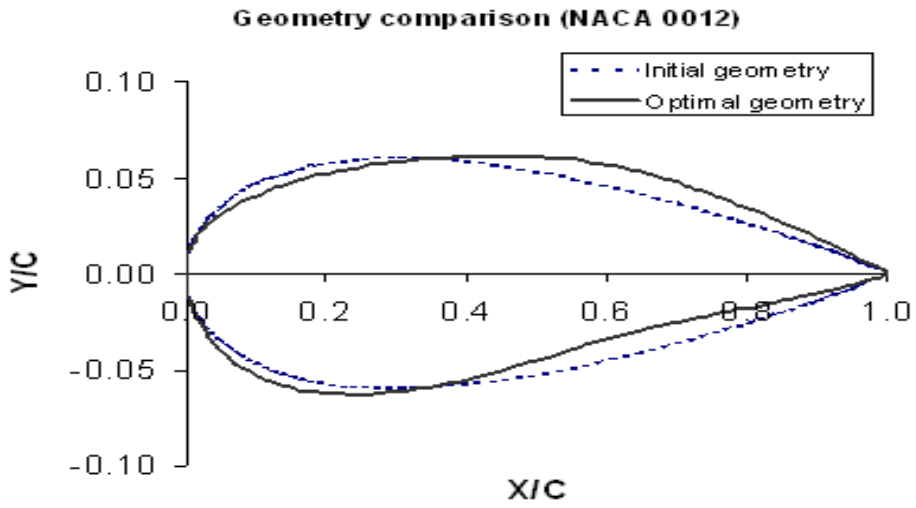


Figure 9. Initial and optimal geometry for NACA 0012.

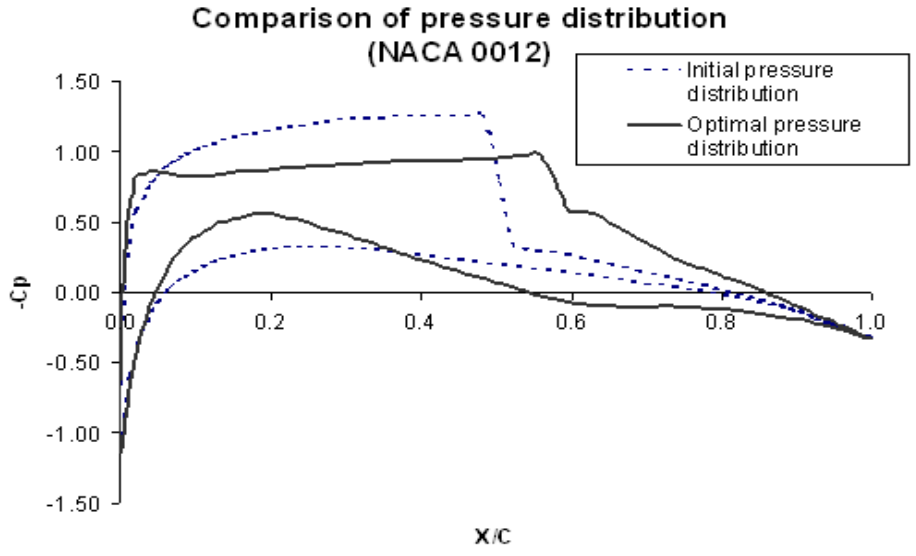


Figure10. Initial and optimal pressure distribution for NACA 0012.

CPUPFC refers to the CPU time per function call.

From Table 3, we can see that using the EXTREM method and the new method, the drag coefficient is reduced by about 62%. The same low drag coefficient is also achieved by using the new method. However using the new method only requires 233 function calls and it saves about 43 % total CPU time compared with the EXTREM method.

The comparison of the optimum shape and pressure distribution are shown in Figures 9 and 10, respectively.

Figure 10 shows that the shock strength is very much weakened which results in a significant reduction in the wave drag and consequently of the total drag of the airfoil.

NOMENCLATURE

Ω, spatial domain; $\partial\Omega$, boundary of spatial domain; **F**, the objective function;
F_k, the value of the objective function at the k-th iteration; x_k, the point at the k-th iteration; **g,** the gradient of the objective function with respect to design variables; s_k, the search direction at the k-th iteration; Φ, state variable; **H**, Hessian matrix; H^{-1} **B**, the inverse matrix of Hessian matrix; d_i, the i-th control points in B-spline; y_i, the y-coordinate of the i-th control points in B-spline curve; u_i, the i-th knot in the knot set of B-spline; α, the set of design variables before knot insertion, vector; $\bar{\alpha}$ the set of design variables after knot insertion, vector; **BFGS**, Broyden–Fletcher–Goldfarb–Shanno.

REFERENCES

Anderson WK, Nielsen EJ (2001). Sensitivity analysis for Navier-Stokes equations on unstructured meshes using complex variables, AIAA J., 39(1): 56-63.

Anderson WK, Venkatakrishnan V (1997). Aerodynamic design optimisation on unstructured grids with a continuous adjoint formulation, AIAA Report 97-0643.

Becker D, Thompson CP (2005). A novel parallel PDE solver for unstructured grids, 5th International Conference on Large-scale Scientific Computation

Bischof C, Carle A, Gorliss G, Griewank A and Hovland P (1992). Generating derivative codes from fortran programs, Sci. Program., 1(1): 11-29

Bischof C, Roh L, Mauer-Oats AJ, ADIC (1997). An extensible automatic differentiation tool ANSI-C, Software Prac. Exp., 27(12): 1427-1456.

Borggaard J, Burns J (1994). A sensitivity equation approach to shape optimisation in fluid flows, ICASE Report, pp. 94-8.

Burkardt J, Gunzburger M (1995). Sensitivity discrepenancy for geometric parameters, ASME CFD Design Optim., 232: 9-15.

Farin GE (2002). Curves and surfaces for CAGD: a practical guide 5th, ISBN 1558607374, San Diego: Academic Press

Fletcher R (1987). Practical methods of optimisation second edition, ISBN 0-471-91547-5, Wiley

Hicks RM, Henne PA (1977). Wing design by numerical optimization, AIAA 77-1247, AIAA Aircraft System and Technology Conference, Seattle, Wash., Aug. 22-24.

Hicks RM, Vanderplaats GN (1977), Application of numerical optimization to the design of supercritical airfoils without drag-creep}, SAE p. 770440.

Holst TL, Pulliam H (2001). Aerodynamic shape optimisation using a real-number-encoded genetic algorithm, AIAA 2001-2473.

INRIA, BAMG v0.68, http://www-rocq1.inria.fr/gamma/cdrom/www/bamg/eng.htm, latest access time 20/05/2005.

Jameson A (1988). Aerodynamic design via control theory, J. sci. comput., 3(3): 233-260.

Jameson A (1995). Optimum aerodynamic design using CFD and control theory, AIAA, pp. 95-1729.

Jacob HG (1982). Rechnergestutzte optimierung statischer and dynamischer system, ISBN: 9783540116417, Springer, Verlag.

Lee KD, Eyi S (1991). Transonic airfoil design by constrained optimisation, AIAA 91-3287, AIAA 9th Applied Aerodynamics Conference, Baltimore, MD

Lee KD, Eyi S (1993). Transonic airfoil design by constrained optimisation, J. aircraft., 30(6).

Mohammadi B, Pironneau O (2001). Applied shape optimisation for fluids, ISBN 0198507437, Oxford: Clarendon Press

Pironneau O (1973). On optimum profiles in Stokes flow, J. fluid mech., 59(1): 117-128

Pironneau O (1974). On optimum design in fluid mechanics, J. fluid mech., 64: 97-110.

Polak E (1984). Optimization: algorithms and consistent approximations, Springer, New York

Rostaining N, Dalmas S, Galligo A (1993). Automatic differentiation in Odyssee, Tellus, 45a(5): 558-568.

Ta'asan S (1995). Pseudo-time methods for constrained optimisation problems governed by PDE, ICASE pp. 95-32.

Wah BW, Chen YX (2000). Constrained genetic algorithms and their applications in nonlinear constrained optimisation, pp 286-293, 12th International Conference on Tools with Artificial Intelligence, November.

constrained optimisation, 12th International Conference on Tools with Artificial Intelligence, November

Xie L (2002). Gradient-based optimum aerodynamic design using adjoint-methods, PhD thesis of Virginia Polytechinc Institute and State University

Xu B (2007). A novel method with knot insertion algorithm of B-spline in shape optimization problems, PhD thesis

Zhong B, Qiao Z (1994). Multi-objective optimization design of transonic airfoils, ICAS-94-2.1.

A novel shape optimization method using knot insertion algorithm in B-spline and its application to transonic airfoil design

Sherar, P. A.

2011-11-16T00:00:00Z

Sherar PA, Thompson CP, Xu B, Zhong B. (2011) A novel shape optimization method using knot insertion algorithm in B-spline and its application to transonic airfoil design. Scientific Research and Essays, Volume 6, Issue 27, pp. 5696-5707

http://dx.doi.org/10.5897/SRE11.487