

# An Optimization Method Based On B-spline Shape Functions & the Knot Insertion Algorithm

P.A. Sherar, C.P. Thompson, B. Xu, B. Zhong \*

*Abstract*—A new method is presented to deal with shape optimization problems. In this method, the geometry is parameterized by B-spline shape functions with the control points of the B-spline curves becoming the design variables in the optimization scheme. The core idea of the method presented is to introduce the knot insertion algorithm which can keep the geometry unchanged whilst increasing the number of control points. Besides this core idea, the super-reduced method and mesh refinement are also employed. The super-reduced method is used to get rid of the equality constraints. Hence a constrained optimization problem is converted into an unconstrained optimization problem with the constraints kept unchanged. We apply the method to two applications; first a problem involving Poisson's equation, and second, an application of the method to an airfoil design problem based on the airfoil NACA 0012. In both of these applications, the new method is shown to be efficient compared with 'standard' methods. In particular, in the airfoil problem we obtain a CPU saving time of about 42% compared with the EX-TREM method. *Keywords:* B-spline, knot insertion algorithm, mesh refinement, optimization

## 1 Introduction

Shape optimization problems can be regarded as a problem of finding a shape which can achieve a given performance while satisfying some constraints. Two kinds of method are used in optimization; gradient-based methods and global methods. The gradient-based method is efficient in finding a local optimum while the global method has the advantage of finding a global optimum and does not require the objective function to be differentiable.

With the 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, called the sensitivity. Some

well-known methods to compute the sensitivity are the finite difference method [16,17], the complex variable method [2], automatic differentiation [4,5], sensitivity analysis [6,7], and the adjoint method [12,13,14].

In shape optimization geometry representation is an important issue. Hicks, Henne, Vanderplaats [10,11] use shape functions to parameterize shape changes. Another approach is to use the coordinates of every surface point as design variables. The approach removes the geometric model from the optimization loop, however, it may lead to discontinuities in the gradient which can be eliminated by a smoothing technique. An example of the use of B-spline control points to parameterize the geometry is given by Anderson and Venkatakrishnan [1].

In this paper, B-splines are chosen as shape functions to parameterize the changing geometry. The knot insertion algorithm has been incorporated into the shape optimization. Using this technique, the initial number of control points can be decreased. We then use the BFGS-based gradient method to find the optimum shape.

## 2 Method Description

In this method, the geometry is parameterized by B-spline shape functions with nominal uniform knot set. Therefore, the control points of the B-spline curves become the design variables in the optimization scheme. The knot insertion algorithm is introduced to keep the geometry unchanged whilst increasing the number of control points. The initial number of control points can be set to a small number, potentially decreasing the number of optimization iterations. To achieve a suitable accuracy the final number of control points can be very large for the value of the objective function. In the approach taken here the knot insertion algorithm kicks in when the solution is close to the optimum solution. After that, a large number of design variables will be produced as specified by the user. Since the knot insertion algorithm does not change the geometry, the optimization problem will be kept the same. Beside this core idea, the super-reduced method and mesh refinement are also incorporated. The super-reduced idea [19] is employed to convert the constrained problem into

---

\*Authors are in alphabetical order. Address: School of Engineering, Cranfield University, Cranfield MK43 0AL, UK. Email: p.sherar;chris.thompson;b.xu.2002;b.zhong@cranfield.ac.uk

an unconstrained one (where the constraints are kept unchanged all the time). Using this method the gradient becomes a super-reduced gradient. Mesh refinement is extensively used in shape optimization. This approach is very useful when the exact gradient is difficult to obtain. In doing so, most of iterations will be done on the coarse grid and with relatively few on the refined grid, thus making the method more efficient.

In the new method, the selection of the search direction and the update of the approximate Hessian matrix follows the same rule as BFGS method [9] does:

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

$$H_{k+1} = H_k + \frac{\mathbf{q}_k \mathbf{q}_k^T}{\mathbf{q}_k^T \mathbf{p}_k} - \frac{H_k \mathbf{p}_k \mathbf{p}_k^T H_k}{\mathbf{p}_k^T H_k \mathbf{p}_k}$$

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.

### 2.1 Knot insertion algorithm

The knot insertion algorithm [8] allows one 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

$$\mathbf{p}(u) = \sum_{i=1}^n \mathbf{d}_i N_{i,k}(u), \text{ with knot set } (u_i)_{i=1}^{n+k}$$

to

$$\sum_{i=1}^{n+1} \mathbf{d}_i^1 N_{i,k}(u), \text{ with knot set } (u_i^1)_{i=1}^{n+1+k}.$$

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

If a knot  $\hat{u}$  is to be inserted to the knot set coinciding with the knot  $u_{p+1}$  say, which already has multiplicity  $s$ , and the new knot set is denoted as

$$u_i^1 = \begin{cases} u_i & i \leq p \\ \hat{u} = u_{p+1} & i = p + 1, \\ u_{i-1} & i \geq p + 2 \end{cases}$$

then the number of control point increases from  $n$  to  $n + 1$ , the knot set number increasing from  $n + k$  to  $n + k + 1$ .

Since B-splines have a local modification property,

only some of the control points will change. These new control points are obtained by linear interpolation of the previous control points according to the formula:

$$\mathbf{d}_i^1 = \alpha_i^1 \mathbf{d}_i + (1 - \alpha_i^1) \mathbf{d}_{i-1}$$

where  $\alpha_i^1 = (\hat{u} - u_i^1)/(u_{i+k}^1 - u_i^1) = (\hat{u} - u_i)/(u_{i+k-1} - u_i)$  and  $i = p - k + s + 2, p - k + s + 3, \dots, p$ . More details of this algorithm can be found in [8]. By repeating the knot insertion algorithm many times the number of control points can be made to increase as required.

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

The purpose in using the knot insertion algorithm is to decrease the number of optimization iterations whilst obtaining a better value for the objective function.

### 3 Application to a Poisson's problem

We take the Poisson equation,

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

with the boundary condition

$$\Phi = 0 \quad \text{on } \partial\Omega$$

and find the optimum shape maximising the maximum value in the domain ( $\Omega$ ):

$$\max_{\partial\Omega} \max_{\Omega} \Phi,$$

subject to a geometric constraint, which is taken here to be the area of the domain,  $A(\Omega)$  is a constant. After reorganizing, the objective function can be chosen as:

$$F = - \max_{\Omega} \Phi$$

Two cubic B-spline curves with nominal uniform knot sets are chosen to parameterize the geometry. After reorganization, the area of the whole domain is given as:

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

where  $y_i, y_j$  are the control points in the upper curve, and lower curve respectively,  $n, m$  are the number of control points in the upper curve, and lower curve respectively, and  $u_i$  are the knots.

The gradient used in the optimization is computed using the finite difference method with the super-reduced method, given by the following equations:

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

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

where  $\delta$  is given by solving the following constraints:

$$A(\Omega (y_1, y_2, \dots, y_{i-1}, y_i + \epsilon, y_{i+1}, \dots, y_n + \delta)) = k$$

and  $k$  is a constant.

### 3.1 Results

With the new method, the optimizer starts from the initial shape (see Figure 1), which has about 3300 triangles, and the number of control points for this initial geometry is 22. After 37 optimization iterations it finds the optimum shape (see Figure 2), which has about 13,000 number of triangles because mesh refinement has been employed. This optimum shape has 40 control points because the knot insertion algorithm has been incorporated.

The value of the objective function at the optimum geometry is  $-0.062333$ .

The software used to solve the Poisson equation is that developed by Becker and Thompson [3].

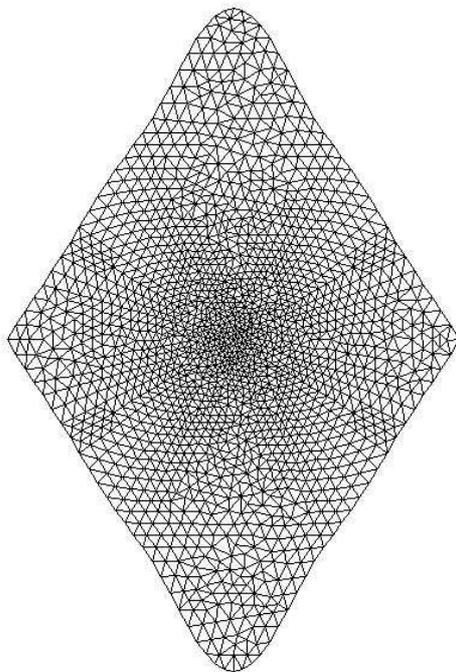


Figure 1: Initial shape

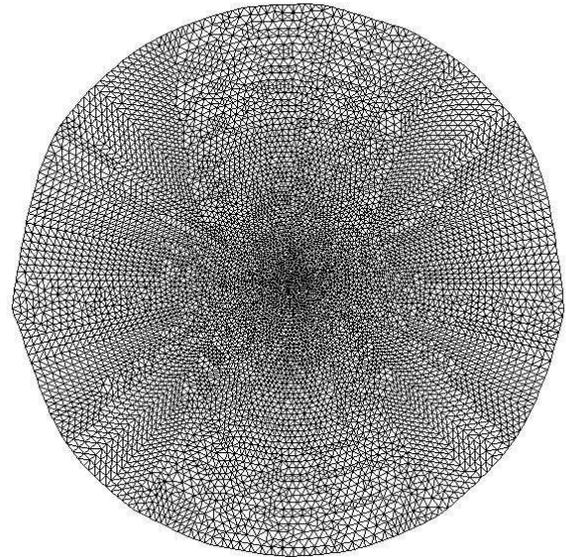


Figure 2: optimum shape produced with the new method

The BFGS method used in the paper has been added the following enhancements: super-reduction, the geometry model, an algorithm to deal with self-intersection, and mesh refinement in order to deal with the constrained shape optimization problem. When the BFGS method is used, we starts from the same initial shape and with the same mesh. The optimizer takes 44 iterations to find the optimum shape. In terms of number of iterations, the new methods decreases the iteration counts by about 20%. The performance comparison of these two methods are shown in Figure 3.

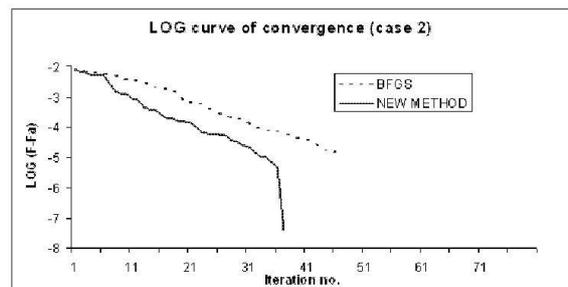


Figure 3: Convergence performance comparison with BFGS method

In the above graph, in iteration 7, the value of the objective function actually increases instead of decreasing because the shape is going to self-intersect, and the algorithm produces replacement. We discuss the algorithm to deal with self-intersecting shapes in [18].

The discretization error has also been investigated.

Below is a table that reflects the influence of discretization error on the PDE solution:

number of nodes	error in solution	rate
814	0.0002161	
3169	5.318e-05	4.06

Table 1: Discretization error analysis for the solution of Poisson's equation

The above table shows that when the number of triangles is increased to about 4 times, then the discretization error will decrease to about 1/4, in agreement with theory.

#### 4 Application for airfoil design

The purpose of this application is to find optimum shapes to minimize the drag of transonic airfoils while keeping the lift coefficient constant and keeping the maximum thickness of the airfoil constant. In this application, we follow Zhong and Qiao's work [20] using the B-spline method to parameterize the geometry. The shape modifications are parameterized by cubic B-splines with a nominal uniform knot set. The geometry of the airfoil is given by the following equation.

$$Y(\Gamma) = Y(\Gamma_{ini}) + \sum_{i=1}^n P_i N_i(X),$$

where  $Y(\Gamma)$ ,  $Y(\Gamma_{ini})$ , represent the  $y$  coordinate of the design geometry and initial geometry respectively,  $P_i$  are the control points of the B-spline, and  $N_i(X)$  are the B-spline basis functions.

The transonic airfoil analysis code used in this paper is the NPUFOIL program, see [20]. 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/inviscid iteration. In this design case, the symmetric airfoil NACA0012 is chosen as the initial geometry. The lift coefficient  $C_L = 0.5$ , Mach number  $M_a = 0.77$ , and the Reynolds number  $Re = 6.5 \times 10^6$  are kept unchanged. For comparison, both the new method and the EXTREM [15] method are used in this design case. For the new method, the geometry is first parameterized by 5 free control points, and finally it reaches the optimum solution which has 9 free control points and the drag is reduced from 0.0182 to 0.0070. The optimum solution of the control point is:

$$\begin{aligned} P_1 &= -0.571028 & P_2 &= -0.70975 \\ P_3 &= -0.41511 & P_4 &= 0.13373 \\ P_5 &= 0.78084 & P_6 &= 1.3277 \\ P_7 &= 0.98959 & P_8 &= 0.56642 \\ P_9 &= 0.13606 \end{aligned}$$

The drag coefficient has been decreased by 62% (from 0.0182 to 0.0070) for both methods, but the EXTREM method uses 408 function calls while the new method uses only 233. The CPU time per function call for these methods, (CPU/FC) is nearly the same. The performance comparison table for the two methods is given below:

Methods	Cd	iterations	FC	CPU/FC
EXTREM	0.0070	N/A	408	0.22246
New Method	0.0070	20	233	0.22418

Table 2: Performance comparison of the new method and the EXTREM method

Fig. 4 shows the comparison of pressure distribution in design case of NACA 0012, and Fig. 5 shows the comparison of the drag coefficient for initial and optimal geometry.

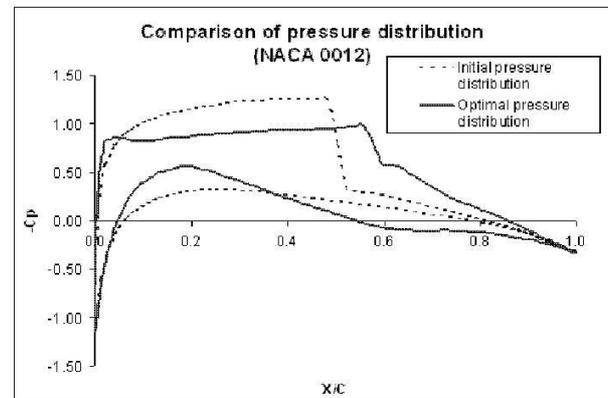


Figure 4: Initial and optimal pressure distribution for NACA 0012

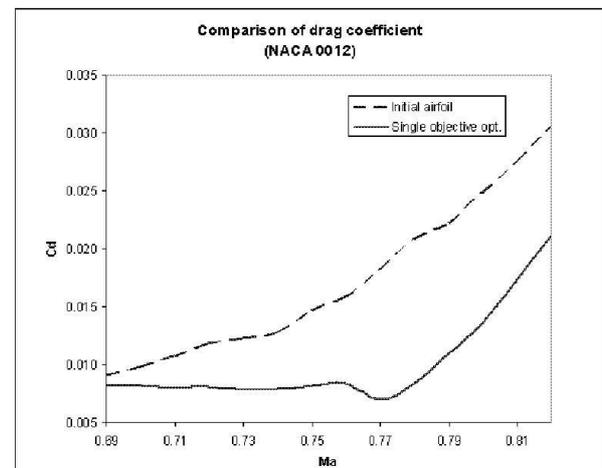


Figure 5: The comparison of drag coefficients for initial airfoil and optimal airfoil (NACA 0012)

## 5 Conclusions

In this paper, a new efficient method for shape optimization using B-spline shape functions combined with knot insertion algorithm and the super-reduced method is given. This method has good performance for both mathematical problems and airfoil design applications. In the application to a Poisson's equation problem, the new method uses only 37 iterations while the BFGS method uses 44 iterations.

In the airfoil design case, the drag coefficient has been reduced significantly, by about 62% with only 233 function calls, saving about 42% CPU time compared to the EXTREM method. Using this technique researchers will be able to accelerate their optimization as long as B-spline shape functions are used to parameterize the geometry.

In the future we will extend the range of application to include 3-dimensional problems.

## References

- [1] Anderson, W.K., Venkatakrisnan, V., "Aerodynamic design optimisation on unstructured grids with a continuous adjoint formulation," *AIAA Report* 97-0643, 1997.
- [2] Anderson, W.K., Nielsen, E.J., "Sensitivity analysis for Navier-Stokes equations on unstructured meshes using complex variables," *AIAA Journal* V39, N1, pp 56-63, 2001.
- [3] Becker, D., Thompson, C.P., "A novel, parallel PDE solver for unstructured grids," *5th International Conference on Large-scale Scientific Computation*, 2005.
- [4] Bischof, C., Carle, A., Gorliss, G., Griewank, A., Hovland, P., ADIFOR, "Generating derivative codes from FORTRAN programs", *Scientific Programming*, V1, N1, pp 11-29, 1992.
- [5] Bischof, C., Roh, L., Mauer-Oats, A.J., ADIC, "An extensible automatic differentiation tool ANSI-C", *Software Practice and Experience*, V27, N12, pp 1427-1456, 1997.
- [6] Borggaard, J., Burns, J., "A sensitivity equation approach to shape optimisation in fluid flows", *ICASE Report* No.94-8, 1994.
- [7] Burkardt, J., Gunzburger, M., "Sensitivity discrepancy for geometric parameters", *ASME CFD for Design and Optimisation*, V232, pp 9-15, 1995.
- [8] Farin, G.E., *Curves and surfaces for CAGD: a practical guide* 5th edition, San Diego: Academic Press, 2002.
- [9] Fletcher, R., *Practical methods of optimisation* second edition, Wiley, 1987
- [10] Hicks, R.M., Henne, P.A., "Wing design by numerical optimization", AIAA 77-1247, *AIAA Aircraft System and Technology Conference*, Seattle, Wash., 1977.
- [11] Hicks, R.M., Vanderplaats, G.N., "Application of numerical optimization to the design of supercritical airfoils without drag-creep", *SAE paper* 770440, 1977.
- [12] Iollo, A., Kuruvila, G., Ta'asan, S., "Pseudo-time method for optimal shape design using the Euler equations", *ICASE Report* No. 93-78, 1993.
- [13] Jameson, A., "Aerodynamic design via control theory", *Journal of scientific computing*, V3, N3, 1988.
- [14] Jameson, A., "Optimum aerodynamic design using CFD and control theory", *AIAA Paper* 95-1729, 1995.
- [15] Jacob, H.G., *Rechnergestutzte optimierung statischer and dynamischer system*, Springer, Verlag, 1982.
- [16] Lee, K.D., Eyi, S., "Transonic airfoil design by constrained optimisation", AIAA 91-3287, *AIAA 9th Applied Aerodynamics Conference*, Baltimore, MD, 1991.
- [17] Lee, K.D., Eyi, S., "Transonic airfoil design by constrained optimisation", *Journal of aircraft*, V30, N6, 1993.
- [18] Sherar, P.A., Thompson, C.P., Xu, B., Zhong, B., "A Novel Shape Optimization Method Using Knot Insertion Algorithm in B-Spline and Its Application to Transonic Airfoil Design", *Computer methods in Applied Mechanics and Engineering*, [in preparation] 2007.
- [19] Xie, L., *Gradient-based optimum aerodynamic design using adjoint-methods*, PhD thesis of Virginia Polytechnic Institute and State University, 2002
- [20] Zhong, B., Qiao, Z., "Multiobjective optimization design of transonic airfoils", *ICAS-94-2.1.1*, 1994.