ECCOMAS Congress 2016
VII European Congress on Computational Methods in Applied Sciences and Engineering
M. Papadrakakis, V. Papadopoulos, G. Stefanou, V. Plevris (eds.)
Crete Island, Greece, 5–10 June 2016

GEOMETRIC CONTINUITY CONSTRAINTS FOR ADJACENT NURBS PATCHES IN SHAPE OPTIMISATION

Xingchen Zhang, Yang Wang, Mateusz Gugala and Jens-Dominik Müller

Queen Mary University of London
Mile End Road, E1 4NS, UK
e-mail: {xingchen.zhang, yang.wang, m.gugala, j.mueller}@qmul.ac.uk

Keywords: shape optimisation, adjoint methods, automatic differentiation, geometrical continuity, constraints, NURBS, CAD.

Abstract. *The automatic shape parametrisation method 'NURBS-based parametrisation with complex constraints (NSPCC)' which uses the control points of multiple B-spline patches as design variables, is extended here to include Non-Uniform Rational B-Splines (NURBS) patches. The necessary modifications to the continuity constraints across patch interfaces are presented and the number of required test points to evaluate the constraints is estimated. Numerical tests to demonstrate the developments are presented.*

1 INTRODUCTION

In shape optimisation problems, several important issues should be considered carefully. A crucial aspect is the choice of parametrisation method as it determines the design space. A wide range of parametrisations have been presented in the literature [1].

Node-based methods use the displacement of the nodes of the surface grid provide the richest design space the CFD surface mesh can express. This space is richer than the CFD discretisation can typically see as high-frequency oscillations can be expressed which are not adequately resolved by the CFD. This is addressed by regularisation or smoothing of the gradients or displacements [2, 3]. Lattice-based methods where suitable shape modes are defined on skeleton grids such as Hicks-Henne bumps for aerofoils [4] or stacks of B-spline curves to define turbomachinery blade shapes [5] are popular in aeronautical design. While they can give appropriate design freedom, they are typically limited to topologically rectangular planforms. Free-form deformations (FFD) with volume splines [6] or Radial Basis Function (RBF) [7] are often used for more general design approaches but are cumbersome to define if interfaces between design (deformable) and fixed surfaces need to be respected.

All of the aforementioned methods suffer from the fact that the optimal shape is produced as a deformed mesh, further processing or manufacturing will then require a manual step to capture the relevant modes in a CAD model. This step will typically incur approximations and impair the optimal shape.

At the other end of the spectrum one could consider using parametrisation defined in CAD to define the design space. While this produces the optimal design in CAD, it is typically very time-consuming and complex to define a design space that is rich enough to contain the relevant shape modes. Furthermore, commercial CAD systems are black-box and the relevant shape derivatives for gradient-based optimisation need to be approximated using finite-differences [8], resulting in high run-times for rich design spaces and issues with robustness under topological changes. The companion paper [9] presents work of the group to overcome some of these issues by applying exact algorithmic differentiation to an open-source CAD system.

As an alternative, we here pursue a vendor-neutral CAD approach which does not rely on a parametrisation defined in a CAD system. Using an initial net of surface patches from a CAD Boundary Representation (BRep), the NSPCC approach [10] derives a rich parametrisation in an automatic fashion by considering all control points of these patches as design variables. The unique feature of NSPCC is that it is not limited to single patches or clamped interfaces between patches, but formulates constraints across patch interfaces to guarantee a user-selected level of geometric continuity such as water-tightness ($G0$), tangency ($G1$) and curvature ($G2$). The approach exploits the polynomial nature of B-Splines by evaluating the value and derivative of the constraint function at a small number of test points. Using a projected gradient method, the design space is then projected into the null space of the constraint matrix, while the range space is used to recover a non-linear constraint using a normal step. The method has been successfully demonstrated on a range of configurations and is currently extended to include shape constraints such as thickness and radius [11].

This paper further develops the approach of Xu et al. [10] in two aspects, a) the approach will be extended from B-Splines to NURBS b) an approximation for the number of required test points is proposed. Finally, a S-bend air duct test case using NURBS is presented.

The remaining parts of this paper are structured as follows: Sec. 2 provides the background knowledge of this study, Sec. 3 describes the methods used to describe and maintain the geometrical continuity. Sec. 4 presents number the estimation of the number required test points.

Conclusions and future research directions are presented in Sec. 5.

2 Geometry parametrisation for CFD shape optimisation

Optimisation of shapes immersed in fluids using CFD typically uses gradient-based algorithms as they converge to the optimum in many fewer design iterations compared to stochastic or surrogate modelling approaches. The gradients $\frac{\partial J}{\partial \alpha}$ of the objective function J with respect to a design variable α induced in each control volume by a design change is best computed using the adjoint method [12, 13, 14]. Writing the flow equations as

$$R(U(x_V), x_V) = 0,$$

where R is the conservative residual of the flow equations, U is the state and x_V are the volume grid coordinates, the adjoint equations can be written as

$$\mathbf{A}^T v = g$$

where $\mathbf{A}^T$ is the transpose of the system Jacobian $\mathbf{A} = \frac{\partial R}{\partial U}$, $v = \frac{\partial J}{\partial R}$ is the adjoint solution and $g = \frac{\partial J}{\partial U}$ only depends on the objective function J, but not on the design variables α.

The adjoint approach allows to compute the sensitivities of J with respect to all N design variables α in a single computation of comparable cost to the flow,

$$\frac{\mathrm{d}J}{\mathrm{d}\alpha} = \frac{\partial J}{\partial \alpha} + \frac{\partial J}{\partial R}\frac{\partial R}{\partial \alpha} = \frac{\partial J}{\partial \alpha} + v^T f$$

The direct dependence of J on α, e.g. if the J is defined on the design surface, is straightforward to compute and omitted for simplicity in the following.

Using the chain rule of calculus, the differentiation of the remaining term $\frac{\partial J}{\partial R}\frac{\partial R}{\partial \alpha}$ can be separated into contributions from the perturbations in volume grid coordinates x_V, which in turn are linked to perturbations of the surface grid coordinates x_B through a mesh deformation algorithm such as Laplacian smoothing.

$$\frac{\partial J}{\partial R}\frac{\partial R}{\partial \alpha} = \frac{\partial J}{\partial R}\frac{\partial R}{\partial x_V}\frac{\partial x_V}{\partial x_B}\frac{\partial x_B}{\partial \alpha} = v^T \frac{\partial R}{\partial x_V}\frac{\partial x_V}{\partial x_B}\frac{\partial x_B}{\partial \alpha}.$$

The first term on the right hand side $\partial R/\partial x_V$ is computed by differentiating the flux and metrics computation of the flow solver w.r.t. the mesh coordinates. The second term $\partial x_V/\partial x_B$ arises from differentiating the volume mesh smoothing. The final term in the chain rule is the sensitivity of the surface coordinates w.r.t. the changes in design variables, $\partial x_B/\partial \alpha$, which requires a differentiation of the parametrisation.

The NSPCC approach uses a source-code implementation of NURBS patches which can then be differentiated using AD, in our work we employ Automatic Differentiation Software tools (AD) [15]. While the implementation in Xu et al.'s work [10] used the coordinates of the control points $\mathbf{P}_{ij}$ as design variables, with each control point having three degrees of freedom (DoF) to move, here we extend the approach to also include the weights ω_{ij} as design variable, leading to 4 DoF per control point.

2.1 NURBS surface patches

Non-Uniform Rational B-splines (NURBS) are widely used to describe geometries. A NURBS patch is a 3D surface defined as [16]:

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)\omega_{i,j}P_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)\omega_{i,j}} \quad 0 \le u,v \le 1 \tag{1}$$

where $P_{i,j}$ are control points, $\omega_{i,j}$ are weights, $N_{i,p}(u)$ and $N_{j,q}(v)$ are p-th and q-th degree bspline basis functions defined in the following knot vectors:

$$\{\underbrace{0,\ldots,0}_{p+1}, u_{p+1},\ldots,u_i,\ldots,u_{r-p-1}, \underbrace{1,\ldots,1}_{p+1}\}$$
$$\{\underbrace{0,\ldots,0}_{q+1}, v_{q+1},\ldots,v_j,\ldots,v_{s-q-1}, \underbrace{1,\ldots,1}_{q+1}\}$$

where $r = n+p+1$ and $s = m+q+1$. $N_{i,p}(u)$ and $N_{j,q}(v)$ are given by the following expression:

$$N_{i,0}(u) = \begin{cases} 1 & if \quad u_i \le u < u_{i+1} \\ 0 & otherwise \end{cases}$$

$$N_{i,k}(u) = \frac{(u-u_i)}{u_{i+k}-u_i}N_{i,k-1}(u) + \frac{(u_{i+k+1}-u)}{u_{i+k+1}-u_{i+1}}N_{i+1,k-1}(u) \tag{2}$$

Written in homogeneous form, a NURBS surface can be expressed as

$$S^{\omega}(u,v) = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)P^{\omega}_{i,j} \qquad 0 \le u,v \le 1, \tag{3}$$

where $P^{\omega}_{i,j} = (\omega_{i,j}x_{i,j}, \omega_{i,j}y_{i,j}, \omega_{i,j}z_{i,j}, \omega_{i,j})$. Written in this form, most of algorithms formulated for B-spline surfaces can straightforwardly be applied to NURBS. We shall hence omit the superscript ω and use P for the homogeneous control point coordinate in the following.

3 NSPCC algorithm using NURBS surfaces

3.1 Geometric continuity

In shape optimisation problems, the congruence between patches usually requires at least $G0$ (adjacent surfaces touch or intersect to be watertight). Smoothness of the geometry is typically a very important aspect in the CFD design, the changes in tangency and curvature often have a very strong influence on the pressure field. $G2$ continuity (continuous curvature) would be required to not induce pressure spikes, however this seems difficult to implement with standard methods and is often not imposed. While our methodology could also be used to impose $G2$, in this study $G0$ and $G1$ continuity are considered.

In the NSPCC approach the constraint between patches is evaluated numerically at the same position in all touching or intersecting patches. Pairs of test points are evenly distributed along the common edges as shown in Fig. 1. $G0$ continuity can then be written as:

$$G0 = X_{s,L} - X_{s,R} = 0 \tag{4}$$

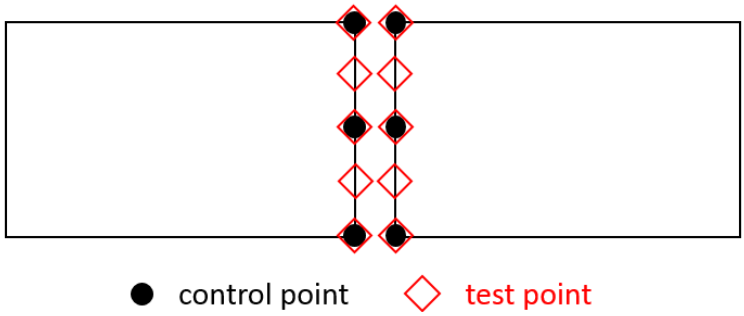

Figure 1: Two patches sharing one common edge.

where $X_{s,L}$ is the test point on the left patch, and $X_{s,R}$ is the test point on the right patch sharing an edge. $G1$ continuity can be written as:

$$G1 = \frac{\mathbf{n}_L \cdot \mathbf{n}_R}{||\mathbf{n}_L||||\mathbf{n}_R||} = 1 \tag{5}$$

where $\mathbf{n}_L$ and $\mathbf{n}_R$ are the unit normal vectors of the tangent plane at the test points either side of the patch interface. The unit normal vector is expressed as:

$$\mathbf{n} = \frac{(\frac{\partial X_s}{\partial u} \times \frac{\partial X_s}{\partial v})}{\left\| \frac{\partial X_s}{\partial u} \times \frac{\partial X_s}{\partial v} \right\|} \tag{6}$$

where u and v are the parametric coordinates of the surface as introduced in equation (1).

3.2 Computation of the constraint matrix using singular value decomposition (SVD)

Linearising the difference in continuity constraint values G between two design iterations n and $n+1$ one obtains

$$G^{n+1} = G^n + \sum_{i=1}^{N} \frac{\partial G}{\partial P_i} \delta P_i, \tag{7}$$

where G comprises the constraints of all required levels of continuity, δP_i is the displacement of the homogeneous coordinate of control point P_i. Requiring $G^{n+1} = G^n$ one obtains a linearised condition for zero change in constraints, hence

$$\sum_{i=1}^{N} \frac{\partial G}{\partial P_i} \delta P_i = \mathbf{C}\delta P = 0. \tag{8}$$

The matrix $\mathbf{C}$ has M_C rows and $4 \times N$ columns in total, corresponding to the total number of M_C constraint equations in the M test points and the N NURBS control points. The elements of the matrix $\mathbf{C}$ are calculated using Automatic Differentiation (AD) of the STEP file interpreter. In the typical situation of all patches having a much higher number of control points in each direction compared to the order of the spline, the majority of the control points do not affect the continuity of the patch borders and the matrix $\mathbf{C}$ is sparse.

It can be seen from equation (8) that the displacements of the control points δP have to lie in the null space of $\mathbf{C}$ to satisfy the continuity constraints in a linearised sense. To obtain the

space of displacement vectors of control points, we need to compute the null space of $\mathbf{C}$. This is performed by means of a SVD from the LAPACK library [17],

$$\mathbf{C} = \mathbf{U}\Sigma\mathbf{V}^T, \tag{9}$$

where $\mathbf{U}$ is an $M_C \times M_C$ unitary matrix, Σ is an $M_C \times 4N$ diagonal matrix with non-negative real numbers on the diagonal, and the $4N \times 4N$ unitary matrix $\mathbf{V}^T$ denotes the transpose of $\mathbf{V}$. The rank of the matrix, r, is the number of the non-zero diagonal entries in Σ. The last $(4N - r)$ columns of the matrix $\mathbf{V}$ span the nullspace of $\mathbf{C}$, denoted by $\mathrm{Ker}(\mathbf{C})$.

3.3 Required number of test points

The polynomial of order N_q on the parametric coordinate of the patch edge, along a section of a B-spline curve can be matched uniquely if there are N_q+1 distinct test points within the knot vector interval that supports this section of the curve. In practice it is algorithmically cumbersome to determine the appropriate number as the support intervals overlap.

The number of required test point pairs for a B-spline curve can be determined a-priori by considering each non-zero knot-interval. To match the polynomial of order N_q exactly we need $N_q + 1$ test point pairs in the interval. The relationship between the number of knots N_k, the number of control points N_p and the order of the spline N_q is

$$N_k = N_p + N_q \tag{10}$$

The number of non-zero knot intervals is

$$N_i = (N_k - 1) - 2(N_q - 1) - N_M \tag{11}$$

where N_M is the number of zero knot intervals because of internal multiplicities. From Eq. (10) and Eq. (11), the number of non-zero knot intervals becomes

$$N_i = N_p - N_q - N_M + 1 \tag{12}$$

In each interval we then need N_q+1 test points to fit the polynomial exactly, for the left side of a patch edge with N_p control points, we hence need M_L test points

$$M_L \geq (N_q+1)(N_p - N_q - N_M + 1),$$

similarly for the right side M_R. Assuming a regular spacing of knots, the number of required test points $M_{T,E}$ along edge E then becomes

$$M_{T,E} \geq \max(M_L, M_R). \tag{13}$$

However, imposing too many test points does not pose a problem as the SVD filters out linearly dependent constraint equations. To allow for non-regular knot-intervals we use

$$M_{T,E} \geq f_T \max(M_L, M_R)$$

with an inflation factor f_T chosen around $1.2 \leq f_T \leq 1.5$. In the typical case of equal polynomial orders $N_q|_L = N_q|_R$ this becomes

$$M_{T,E} \geq f_T(N_q+1)(\max(N_p|_L, N_p|_R) - N_q - N_M + 1). \tag{14}$$

The exactness argument employed for B-Splines does not carry over straightforwardly to the non-rational functions in NURBS. This could be accommodated by increasing the factor f_T, however in practice we have not found this to be necessary.

Due to finite-precision arithmetic and ill-conditioning of the constraints, the singular values do not drop off to zero sharply after all non-singular modes, but show a rather gradual decrease. In practice the authors have found the NSPCC method applied to B-Splines to be very insensitive against setting the cut-off threshold. The constraint equations G are non-linear and for finite-size steps the design iteration introduces a constraint violation, requiring a recovery step in the range of $\mathbf{C}$. An exact representation of the feasible space is hence not essential.

4 Results

4.1 Discrete incompressible flow and adjoint solver

In this work, our in-house code GPDE [18] is used. GPDE is an incompressible, viscous, steady flow solver on unstructured grids. The spatial discretisation of the flow solver (primal) is based on a finite volume approach with 2nd order schemes. The SIMPLE-type time-discretisation [19] is used to decouple velocity and pressure computation.

A discrete adjoint solver is used to obtain the sensitivity of the objective function w.r.t. changes in surface node coordinates, $\frac{\partial J}{\partial x_S}$. The adjoint solver is derived from the flow solver via automatic differentiation with the AD tool Tapenade [15]. AD is used to produce the relevant derivative source code for fluxes and source terms. The differentiated routines are then assembled in a hand-written driver code to improve performance, exploiting the fixed-point nature of the primal. In the fixed point iteration of GPDE, stabilisation strategies such as CFL-ramping, skewness correction, semi-coupled methods are implemented to stabilise both primal and discrete adjoint solvers [20].

4.2 The number of test points

Two test cases are utilised in this study to investigate the number of test points required. The first one is a half cylinder case as shown in Fig. 2 and the other one is the ONERA M6 case shown in Fig. 3.

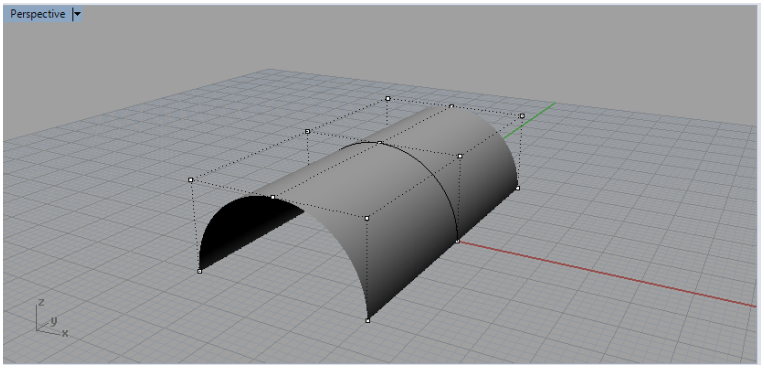

Figure 2: The 3D model of a half cylinder and its control points.

For both cases, all of parameters about the geometry such as the control points, the knot vectors are read from the STEP file. STEP file is a standard file to exchange data among different systems [21]. In addition, a random perturbation to the geometry is given in these two cases.

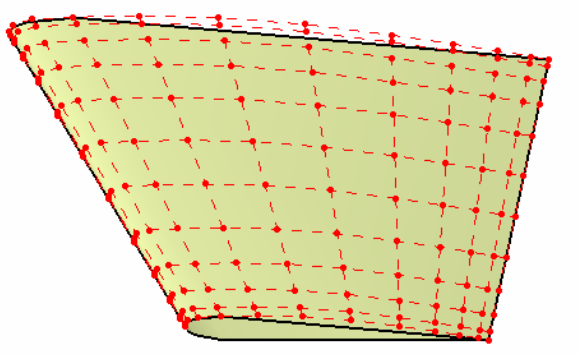

Figure 3: The 3D model of ONERA M6 wing which consists of two patches, and the control points of one patch.

4.2.1 Half cylinder

The half cylinder case used in this study is created using Rhino 5.0 [1] This half cylinder consists of two NURBS surfaces sharing one common edge. The number of control points on each surface along the common edge is 5, and the degree of surface along the common edge is 2, hence order 3. The knot vector along the common edge is

$$\{0,0,0,0.5,0.5,1.0,1.0,1.0\},$$

hence the number of non-zero knot vector interval is 2, matching the estimate of $N_p - N_q - N_M + 1$ in Eq. (12). According to Eq. (14), the number of test points should be:

$$M \geq (3+1)2 = 8.$$

Fig. 4 shows the number of singular values found vs. the number of test points. To useful numerical precision, the number of non-singular values no longer increases beyond 10 test points., which means that 10 test points are enough to ensure the continuity constraints. This matches well with the result of equation (14).

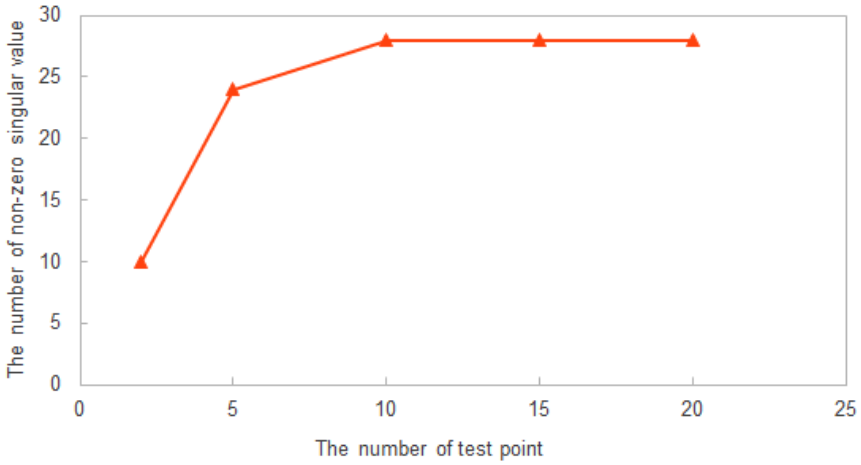

Figure 4: The number of non-zero singular values when the number of test points changes for half cylinder case.

[1] https://www.rhino3d.com/

4.2.2 ONERA M6 wing

The M6 wing case used in this study is created using CATIA V5. It consists of two B-spline patches and there are two common edges as shown in Fig. 3. In this study, B-splines are treated as NURBS by setting weight 1.0 to every control point. There are 12 control points along each common edge and the degree of surface along the common edge is 5, hence order 6. The knot vector along the common edge is

$$\{0,0,0,0,0,0,0.12,0.27,0.41,0.56,0.70,0.85,1.0,1.0,1.0,1.0,1.0,1.0\}\ ,$$

hence the number of non-zero knot vector intervals is 7, matching the estimate of $N_p - N_q - N_M + 1$ in Eq. (12).

According to equation (14), the number of test points should be:

$$M \geq (6+1)7 = 49.$$

In this study, the relationship between the number of non-zero singular value and the number of test points is shown in Fig. 5. It indicates that the number of singular values keep constant after the number of test points reaches 60 which means that 60 test points are enough.

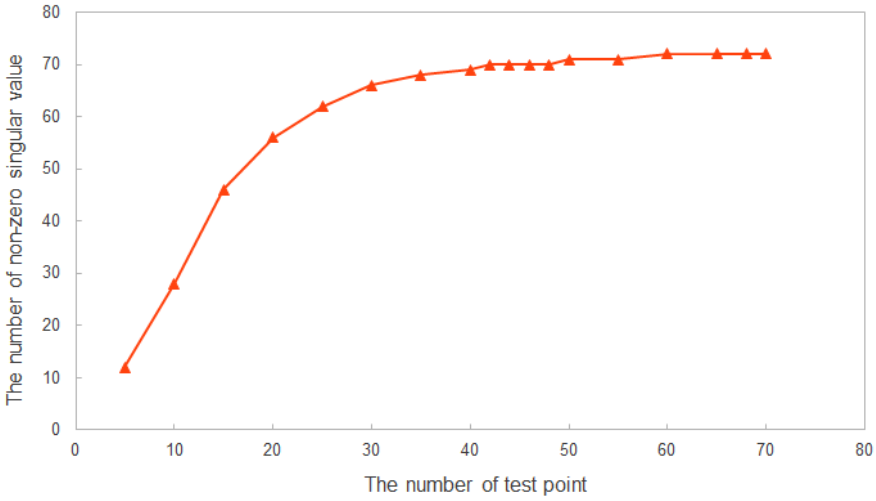

Figure 5: The number of non-zero singular values when the number of test points changes for M6 wing case.

4.3 Optimisation results

In this paper, we apply the NURBS-based parametrisation method to a S-bend air duct case. The original shape of this S-bend is shown in Fig. 6. There are 30 surfaces in this geometry, only the 8 surfaces in the middle section are design surfaces. 400 control points on these surfaces are allowed to move and change weights, thus there are 1600 DoF in total. $G1$ continuity is imposed across the interfaces of these 8 surfaces.

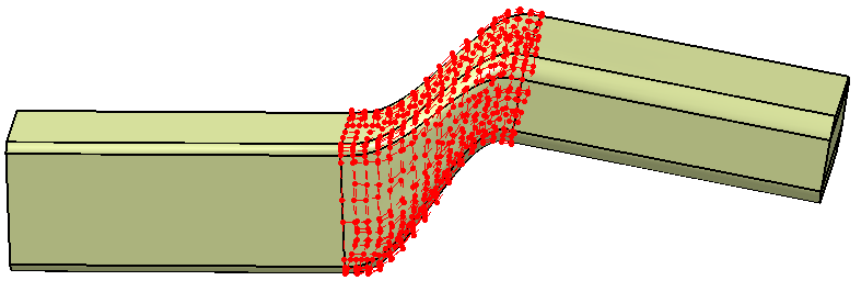

Figure 6: The S-bend air duct and the control points of middle section.

The cost function is the mass-averaged total pressure loss between inlet and outlet defined as

$$J = \frac{\int_{inlet} p_{total}(\mathbf{u}\cdot\mathbf{n})d\mathbf{S} - \int_{outlet} p_{total}(\mathbf{u}\cdot\mathbf{n})d\mathbf{S}}{\int_{inlet}(\mathbf{u}\cdot\mathbf{n})d\mathbf{S}} \tag{15}$$

The flow solver GPDE is used with steepest descent with fixed step width as optimiser. The cost function and gradient history during the 62 iterations are shown in Fig. 7 and Fig. 8, respectively. From Fig. 7 it can be seen that, the cost function is reduced by about 9.4% during the optimisation.

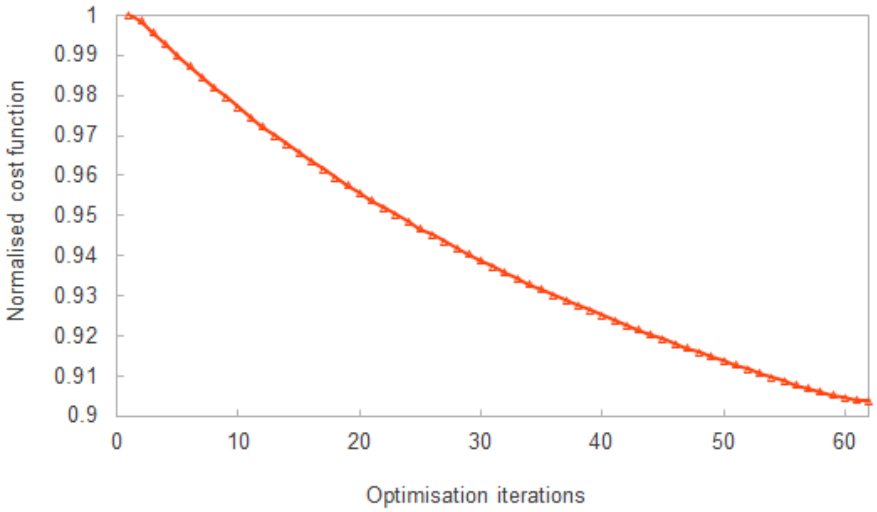

Figure 7: The cost function history.

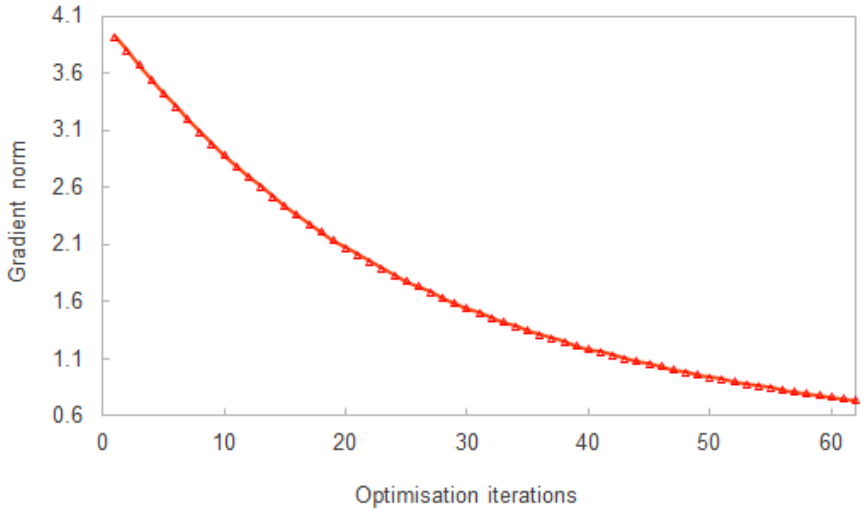

Figure 8: The gradient history.

The optimised shape is shown in Fig. 9. Checks with the CAD software, confirm that the $G1$ continuity across the interfaces in the middle section is satisfied. In addition, the maximum $G1$ deviation value during the optimisation is 2.3×10^{-10}. This means after the extension to NURBS, the geometric continuity can be ensured across different patches.

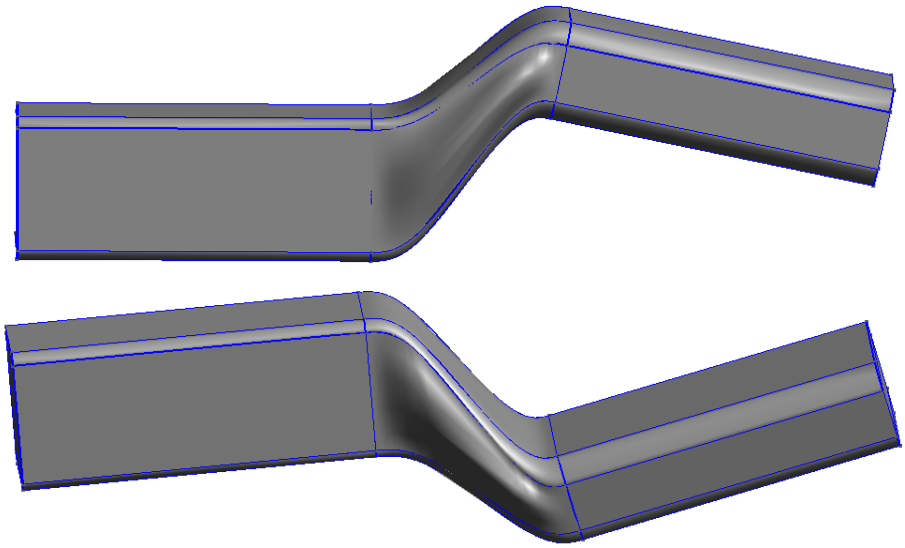

Figure 9: The optimised S-bend air duct.

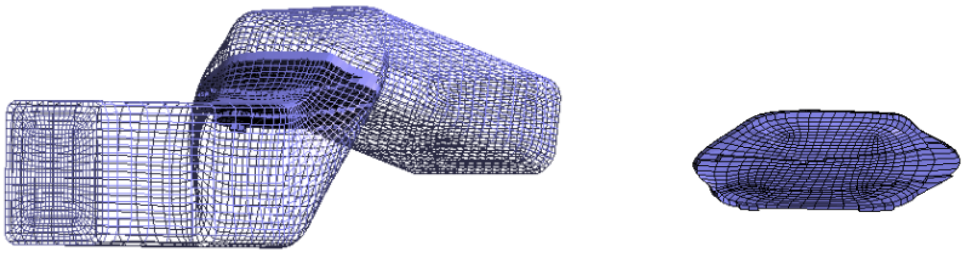

Figure 10: The cross section of optimised S-bend air duct.

Fig. 10 shows the cross section of the cranked part of this S-bend. It can be seen that the side surfaces are pushed out after optimisation offering more cross-sectional area to the flow which through reduction the velocity gradients at the wall reduces skin friction and ultimately total pressure loss. A more significant effect is the reduction in secondary flow motion as visible in Fig. 11, showing the streamlines for the initial and deformed shape after 62 optimisation iterations. The optimised shape shows diminished secondary flow, which results in lower total pressure loss.

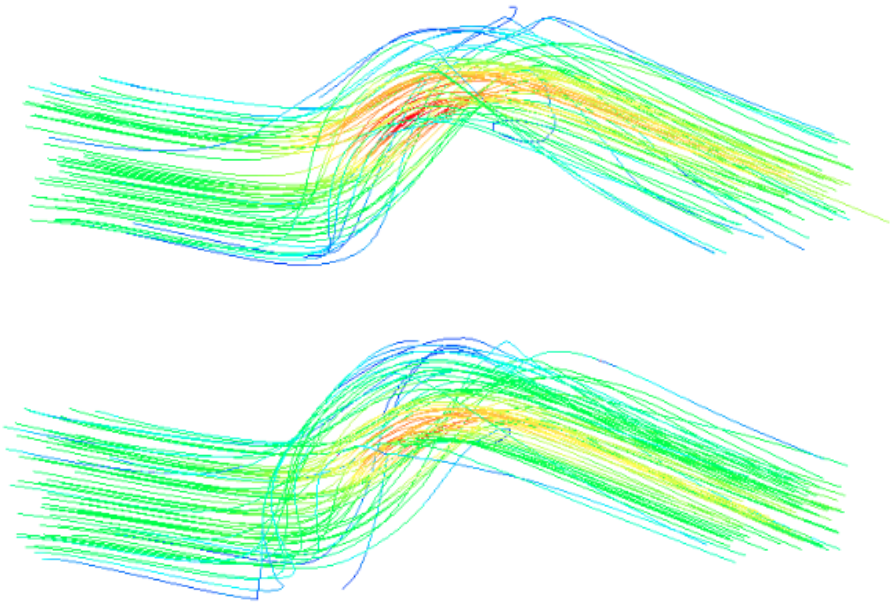

Figure 11: Streamline plots for initial (top) and optimised ducts (bottom).

5 CONCLUSIONS

The NSPCC method of Xu et al. [10] has been extended to include NURBS patches. The test-point approach projecting the design space of control point movements has been extended to include weight changes. An estimate of the required number of test points has been presented, numerical tests confirm the validity of the estimate.

An S-bend air duct optimisation case is presented to demonstrate the proposed method works for shape optimisation problem. The NURBS-based parametrisation using a typical density

of control points is able to develop a very rich but smooth geometry perturbation with $G1$ geometric continuity. Using a simple optimiser the objective function was reduced by over 9%.

6 ACKNOWLEDGEMENTS

The authors would like to thank the funding from China Scholarship Council and Queen Mary, University of London for this research (No. 201306230097). The discussion with colleagues from EC projects About Flow (No. 317006) and IODA (No. 642959) are also very appreciated.

REFERENCES

[1] J.A. Samareh, A survey of shape parameterization techniques. *NASA Conference Publication*, Citeseer, 1999.

[2] A. Jameson, A. Vassberg, Studies of alternative numerical optimization methods applied to the brachistochrone problem. *CFDJ*, **9**(3), 2000.

[3] A. Jaworski and J.-D. Müller, Toward modular multigrid design optimisation. *Lecture Notes in Computational Science and Engineering, Vol. 64*. Springer, 2008.

[4] R.M. Hicks, Wing Design by Numerical Optimization. *Journal of Aircraft*, **15**(7), 407-412, 1978.

[5] S. Shahpar, Challenges to overcome for routine usage of automatic optimisation in the propulsion industry. *The Aeronautical Journal*, **115**(1172), 612-25, 2011.

[6] W. Song, X. Yang, Free-form deformation with weighted T-spline. *The Visual Computer*, **21**(3), 139-151, 2005.

[7] S. Jakobsson, O. Amoignon, Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization. *Computers & Fluids*, **36**(6), 1119-1136, 2007.

[8] T.T. Robinson, C.G. Armstrong, H.S. Chua, C. Othmer, T. Grahs, Sensitivity-based optimization of parameterised CAD geometries. *8th World Congress on Structural and Multidisciplinary Optimisation*, Lisbon, 2009.

[9] S. Auriemma, M. Banovic, O. Mykhaskiv, H. Legrand, J.D. Müller, A. Walther, Optimisation of a u-bend using CAD-based adjoint method with differentiated CAD kernel. *ECCOMAS Congress 2016*, Crete Island, Greece, 5-10 June 2016.

[10] S. Xu, W. Jahn, J.-D. Müller, CAD-based shape optimisation with CFD using a discrete adjoint. *IJNMF*, **74**(3), 153-68, 2013.

[11] S. Xu, D. Radford, M. Meyer, J.-D. Müller, CAD-based adjoint shape optimisation of a one-stage turbine with geometric constraints. *ASME Turbo Expo 2015*, June, 2015. GT2015-42237.

[12] O. Pironneau, On Optimum Design in Fluid Mechanics. *JFM*, **64**, 97-100, 1974.

[13] A. Jameson, Aerodynamic Design via Control Theory. *JSC*, **3**, 233-260, 1988.

[14] M.B. Giles, N.A. Pierce, An introduction to the adjoint approach to design. *Flow, turbulence and combustion*, **65**(3-4), 393-415, 2000.

[15] L. Hascoët, V. Pascual, TAPENADE 2.1 user's guide. *Technical Report*, INRIA, 2004.

[16] L. Piegl, W. Tiller, *The NURBS Book. 2nd*, New York: Springer, 1997.

[17] E. Anderson, Z. Bai, C. Bischof, et al. LAPACK Users' guide[M], Siam, 1999.

[18] D. Jones, J.D. Müller, F. Christakopoulos, Preparation and assembly of discrete adjoint CFD codes. *Computers & Fluids*, **46**(1). 282-286, 2011.

[19] J.H. Ferziger, M. Peric, Computational methods for fluid dynamics[M], Springer Science & Business Media, 2012.

[20] Y. Wang, S. Akbarzadeh, J.D. Müller, Stabilisation of Discrete Adjoint Solvers for Incompressible Flow. *22nd AIAA Computational Fluid Dynamics Conference*, 2749, 2015

[21] M.J. Pratt, Introduction to ISO 10303 - the STEP Standard for Product Data Exchange. *Journal of Computing and Information Science in Engineering*, **1**(1), 102-103, 2001.