ARTICLE TEMPLATE

Geometric continuity constraints of automatically derived parametrisations in CAD-based shape optimisation

Jens-Dominik Müller, Xingchen Zhang, Siamak Akbarzadeh, and Yang Wang

School of Engineering and Materials Science, Queen Mary University of London, E1 4NS, London, UK

ARTICLE HISTORY
Compiled September 23, 2019

ABSTRACT
CAD geometries are most often exchanged between analysis tools using NURBS patches to represent the boundary (BRep). We present a method where the control points of the BRep are used to automatically derive parametrisations suitable for shape optimisation with gradient-based methods. Particular focus is on ensuring geometric continuity between the NURBS patches, which is achieved through formulation of discrete constraints. Design variables then arise from formulating an orthogonal basis to the remaining design space using a singular value decomposition (SVD). The manuscript presents the extension of earlier work on B-spline surfaces to full NURBS surfaces and investigates the effect of the cut-off threshold of the SVD on the optimisation results. To enable routine automatic use, an estimation of the effective rank is proposed which allows to automatically determine the suitable cut-off threshold. The effectiveness of the algorithm is demonstrated for the minimisation of total pressure loss over a section of an automotive climate duct, and a U-bend cooling channel.

KEYWORDS
continuity constraints; nullspace; singular value decomposition; CAD; NURBS; shape optimisation

1. Introduction

In many engineering applications, further improvements in performance or reductions in environmental impact can only be achieved by considering large design spaces with very many design variables. Gradient-based methods have shown to have a convergence rate to the optimum that is near-independent of the number of design variables, hence are suitable for applications with large design spaces. The state of the art to compute gradients for expensive computational models such as CFD is the adjoint approach which computes the entire gradient vector of a single objective function with respect to all design variables in a single computation. Adjoint CFD is now well established with all major code vendors offering adjoint CFD codes, as well as a number of open-source offerings.

The choice of parametrisation determines the design space and hence affects the outcome of the optimisation. The use of adjoint methods removes obstacles arising from

The work was carried out while Dr. Zhang was a PhD student at QMUL, he his now a Post-Doctoral fellow at Shanghai Jiao Tong University.

the computational cost of gradient computation and allows to consider a much wider range of parametrisation approaches, many have been presented and compared in the literature (Samareh 2001; Mousavi, Castonguay, and Nadarajah 2007; Masters et al. 2017), and their advantages and disadvantages are briefly reviewed in the following paragraph.

Node-based methods use the displacement of surface nodes of a given mesh and provide the richest design space the Computational Fluid Dynamics (CFD) mesh can express. This design space includes unwanted highly oscillatory modes which are in general not adequately damped by the CFD solver, which needs to be addressed by regularisation or smoothing of the gradients or displacements (Jameson and Vassberg 2000; Jaworski and Müller 2008; Schmidt and Schulz 2009). Lattice-based methods where suitable shape modes are defined on skeleton grids are popular in aeronautical design, such as Hicks-Henne bumps for aerofoils (Hicks and Henne 1978) or stacks of B-spline curves to define turbo-machinery blade shapes (Shahpar 2011). While they can give appropriate design freedom, they are cumbersome to extend beyond topologically rectangular planforms. Free-form deformation (FFD) with Radial Basis Function (RBF) (Jakobsson and Amoignon 2007) is often used for the parametrisation of arbitrary shapes, however the global nature of the basis functions makes it computationally expensive to ensure water tightness of the geometry at joins with non-deformable parts. FFD with volume splines (Samareh 2004) or does allow to define a deformation field on specific volumes, but the setup of suitable deformation volumes and appropriate constraints can become onerous for complex geometry.

All the aforementioned methods suffer from two issues. Firstly, imposition of complex constraints such as profile thickness or leading edge curvature has either not yet been demonstrated or is not possible. Secondly, and more importantly, the optimal shape is produced as a deformed mesh, but not in CAD format as typically required for multi-physics analysis or manufacturing. Reconstruction of the optimal shape through some CAD parametrisation will typically incur approximations and impair optimality. To overcome this bottleneck, CAD-based methods are needed which update a CAD model within the optimisation loop. Two major approaches can be distinguished in CAD-based shape optimisation. When used in gradient-based optimisation, in both approaches a significant challenge is to compute the shape gradients, i.e. the derivatives of a surface coordinate with respect to the CAD parameter.

'Explicit' CAD parametrisations use the parameters of the geometric construction in the CAD feature tree as design variables (Fudge, Zingg, and Haimes 2005; Robinson et al. 2012; Brock et al. 2012; Banovic et al. 2017). While this closely parallels the manual design change and can employ traditional engineering parameters, the design space is typically too sparse and cannot express some important modes. Moreover, explicit parametrisations are typically limited to a specific shape such as a 3-D blade.

'Implicit' CAD parametrisations work with the boundary representation (BRep) of the geometry which is commonly represented as a set of B-Spline or NURBS surface patches in the STEP or IGES format. The surface displacement is linear in the position of the B-spline control points, and if the deformation is expressed on a single patch with clamped edges, the computation of the shape derivatives is trivial and easy to use in optimisation (Fudge, Zingg, and Haimes 2005; Martín et al. 2012; Andrés et al. 2010).

CAD models of complex geometry, however, describe the surface with a net of patches, and constraints need to be imposed on the control point movement in order to retain geometric continuity across patch interfaces such as water-tightness, tangency or curvature. The NURBS-based Parametrisation with Complex Constraints (NSPCC) method of Xu et al. (2013) is able to ensure patch continuity under design deformation.

The approach has been successfully demonstrated on automotive ducts (Xu, Jahn, and Müller 2013), turbomachinery blades (Xu et al. 2015), aircraft wing-belly fairings (Mykhaskiv et al. 2017), and automotive wing mirrors (Myhkaskiv 2019).

This paper further develops and investigates the NSPCC approach in a number of aspects. Firstly, the shape deformation modes arising from applying SVD to the constraint matrix are analysed to gain insight into the preconditioning effect that NSPCC exhibits for very rich design spaces. Secondly, the influence of the SVD cut-off threshold σ_T on the design space and ultimately the convergence of the optimisation and the obtained reduction in objective is investigated.

The paper is organised as follows: Section 2 introduces the adjoint method, Section 3 NURBS surfaces. Section 4 presents the NSPCC approach and discusses details such as the deformation modes, cut-off σ_T and number of test points. Constraint recovery is discussed in Section 5. The optimisation framework and optimisation results are shown in Sections 6, 7 and 8. Finally, conclusions are given in Section 9.

2. The adjoint approach

Gradient-based optimisations require computation of the derivative of the objective such as drag or pressure drop with respect to the design variables. This can be achieved using finite differences, complex variable methods or tangent linearisation, however all of these approaches incur the cost of an additional CFD solve for each design variable. The adjoint approach (Pironneau 1974; Jameson 1988; Giles and Pierce 2000) on the other hand computes the entire gradient vector for one objective in a single computation of similar cost as the flow simulation. In this work, we use the discrete adjoint method (Giles et al. 2003), as a) it provides a derivative that is consistent with the discretised primal, and b) as it can be produced with Automatic Differentiation software tools. The discretised steady-state flow equations can be written as

$$\mathbf{R}(\mathbf{U}(\mathbf{x}_V), \mathbf{x}_V) = 0,$$

where $\mathbf{R}$ is the conservative residual of the flow equations, $\mathbf{U}$ is the state and $\mathbf{x}_V$ The adjoint equations can be written as

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

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

The adjoint approach allows to compute the sensitivities of J with respect to all N design variables $\boldsymbol{\alpha}$ in a single computation of comparable cost to the flow, so

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

The direct dependence of J on $\boldsymbol{\alpha}$, e.g. if 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 $\frac{\partial J}{\partial \mathbf{R}} \frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}}$

can be separated as

$$\frac{\partial J}{\partial \mathbf{R}}\frac{\partial \mathbf{R}}{\partial \boldsymbol{\alpha}} = \frac{\partial J}{\partial \mathbf{R}}\frac{\partial \mathbf{R}}{\partial \mathbf{x}_V}\frac{\partial \mathbf{x}_V}{\partial \mathbf{x}_B}\frac{\partial \mathbf{x}_B}{\partial \boldsymbol{\alpha}} = \mathbf{v}^T \frac{\partial \mathbf{R}}{\partial \mathbf{x}_V}\frac{\partial \mathbf{x}_V}{\partial \mathbf{x}_B}\frac{\partial \mathbf{x}_B}{\partial \boldsymbol{\alpha}}. \tag{1}$$

where $\mathbf{x}_V$ are the volume grid and $\mathbf{x}_B$ the surface grid coordinates. The term $\frac{\partial \mathbf{R}}{\partial \mathbf{x}_V}$ on the right hand side is computed by differentiating the flux and metrics computation of the flow solver w.r.t. the mesh coordinates. The second term $\frac{\partial \mathbf{x}_V}{\partial \mathbf{x}_B}$ arises from differentiating the volume mesh relaxation. The final term $\frac{\partial \mathbf{x}_B}{\partial \boldsymbol{\alpha}}$ requires a differentiation of the parametrisation.

As presented, Eq. 1 suggests to compute the geometric derivatives in forward mode, i.e. perturbing a design variable and cascading this through the computational chain in the same direction as the primal geometry. This approach would require to perform the geometric computation for each design variable. As an alternative, we could transpose Eq. 1 and compute the derivative in reverse or adjoint mode, which incurs only a single geometric computation for the entire gradient vector. Forward-mode differentiation is arguably much simpler, and as the computational cost of the geometric perturbation is negligibly small compared to the flow simulation (Xu, Jahn, and Müller 2013), we have used the forward approach in this work.

3. NURBS surface patches

NURBS are widely used to describe geometries. A NURBS patch is defined as (Piegl and Tiller 2012):

$$S(u,v) = \frac{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)\omega_{i,j}\mathbf{P}_{i,j}}{\sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)\omega_{i,j}} \qquad 0 \le u,v \le 1, \tag{2}$$

where $\mathbf{P}_{i,j}$ are the control point coordinates, $\omega_{i,j}$ the corresponding weights, $N_{i,p}(u)$ and $N_{j,q}(v)$ the p-th and q-th degree B-spline 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:

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

$$N_{i,k}(u) = \frac{(u-u_i)}{u_{i+k}-u_i}N_{i,k-1}(u) + \frac{(u_{i+k+1}-u)}{u_{i+k+1}-u_{i+1}}N_{i+1,k-1}(u). \tag{3}$$

4. NURBS-based parametrisation with complex constraints

4.1. *Testpoint approach for constraint imposition*

The BRep of a complex geometry arising from a CAD construction is typically represented with a number of NURBS surface patches that abut or intersect. The interfaces between surfaces require at least G_0 continuity (watertightness), but for immersed surfaces typically G_1 (tangency) or in case of wing and blade profiles G_2 (curvature) to avoid pressure spikes.

NSPCC is a novel approach to discretely impose geometric constraints on Boundary representations (BRep) which naturally provides a parametrisation with orthogonal modes (Xu, Jahn, and Müller 2013; Xu et al. 2015; Zhang, Jesudasan, and Müller 2019). NSPCC evaluates discrete constraint equations in sets of test points that are distributed along patch interfaces as illustrated in Fig. 1. B-splines are polynomials and given a sufficient number of test points, geometric continuity to machine precision can be guaranteed. NURBS surfaces on the other hand are rational, however still possess regularity. Hence a sufficient number of test points can ensure that the error in constraint satisfaction is below a required geometric tolerance. Estimation formulae for the test point number are given in Sec. 4.4.

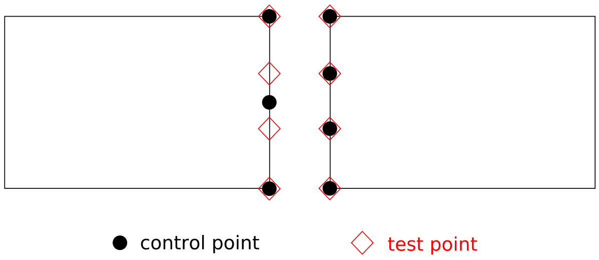


Figure 1. NURBS control points and test points along an interface shared between two NURBS patches.

G_0 continuity can be formulated as

$$G_0 = \mathbf{X}_{S,L} - \mathbf{X}_{S,R} = 0, \tag{4}$$

where $\mathbf{X}_{S,L}$ and $\mathbf{X}_{S,R}$ are the test point coordinates on the left and right patches of the interface. The G_1 continuity constraint is formulated as

$$G_1 = \mathbf{n}_L \times \mathbf{n}_R = 0, \tag{5}$$

where $\mathbf{n}_L$ and $\mathbf{n}_R$ are the unit normal vectors of the tangent plane at the test points on either side of the patch interface,

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

where u and v are the parametric coordinates of the surface as introduced in equation (2).

The test point framework is not limited to continuity constraints but supports the formulation of arbitrary constraints at test points. E.g. Xu et al. (2015) formulate thickness constraints by requiring a minimal distance between test points on opposite blade surfaces. Mykhaskiv et al (2018) constrain surface curvature in test points to enforce a minimal trailing edge radius of a turbomachinery blade.

Let G denote the set of constraint equations at all test points. Assuming a feasible design iterate n, i.e. $G^n = 0$, any design change needs to remain feasible, $G^{n+1} = 0$. Linearising the change in feasibility results in

$$G^{n+1} - G^n = \sum_{i=1}^{N} \frac{\partial G}{\partial \mathbf{P}_i} \delta \mathbf{P}_i = \mathbf{C} \delta \mathbf{P} = 0, \tag{7}$$

where δP_i is the displacement of the homogeneous coordinate of control point P_i. To maintain the continuity constraints. The matrix $\mathbf{C}$ is termed *constraint Jacobian.* It has M_C rows and in the general case of NURBS $4 \times N$ columns, where M_C is the number of constraint equations, N is the number of NURBS control points. To retain feasibility, any control point update δP_i needs to reside in the nullspace of $\mathbf{C}$.

The NSPCC approach uses a source-code implementation of NURBS patches which can then be differentiated using Algorithmic Differentiation (AD) to compute the elements in the constraint Jacobian $\mathbf{C}$. The AD tool Tapenade (Hascoët and Pascual 2004) is applied to obtain entries in $\mathbf{C}$.

4.2. *Computing nullspace of the constraint matrix using SVD*

The SVD factorises the constraint Jacobian $\mathbf{C}$ as

$$\mathbf{C} = \mathbf{U} \mathbf{\Sigma} \mathbf{V}^T, \tag{8}$$

where $\mathbf{U}$ is a $M_C \times M_C$ unitary matrix and $\mathbf{\Sigma}$ is a $M_C \times 4N$ non-negative diagonal matrix. $\mathbf{V}$ is a $4N \times 4N$ unitary matrix, its last $(4N - r)$ columns span the nullspace of $\mathbf{C}$ (Strang 2006), here r is the rank of $\mathbf{C}$. To allow for finite precision arithmetic, a cut-off threshold σ_T is defined below which singular values are considered zero and which hence determines the numerical rank r' and hence the numerical nullspace $\ker'(\mathbf{C})$ (Foster and Davis 2013).

NSPCC uses the SVD basis vectors of the nullspace as a basis for the design space, any feasible perturbations of control points can be expressed as their linear combinations,

$$\delta \mathbf{P} = \sum_{k=1}^{4N-r'} v_{k+r'} \delta \alpha_k = \mathbf{V}_N \delta \boldsymbol{\alpha}, \tag{9}$$

where $\delta \alpha_k, k = 1, 2, ..., 4N - r'$ are the perturbations of design parameters, and $v_{r'+1}, v_{r'+2}, ..., v_{4N}$ are the column vectors of $\mathbf{V}$ that form a basis for the nullspace. Note that the $\mathbf{V}$ is a unitary matrix, which means all deformation modes v_i are orthogonal to each other.

The choice of SVD cut-off σ_T determines the rank and hence the number of design variables and their mode shapes. The effect of varying σ_T is examined in Sec. 4.4.

An update of $\delta\mathbf{P}$ results in a surface perturbation

$$\Delta\mathbf{S} = \sum_{i=0}^{n}\sum_{j=0}^{m} N_{i,p}(u)N_{j,q}(v)\delta\mathbf{P}_{i,j}. \tag{10}$$

and a new NURBS surface

$$\mathbf{S}^{n+1} = \mathbf{S}^{n} + \Delta\mathbf{S}. \tag{11}$$

The gradient of the objective function w.r.t. the design variables $\boldsymbol{\alpha}$ is

$$\frac{dJ}{d\boldsymbol{\alpha}} = \frac{dJ}{d\mathbf{X}_s}\frac{\partial\mathbf{X}_s}{\partial\mathbf{P}}\frac{\partial\mathbf{P}}{\delta\boldsymbol{\alpha}} = \frac{dJ}{d\mathbf{X}_s}\frac{\partial\mathbf{X}_s}{\partial\mathbf{P}}\mathbf{V}_N. \tag{12}$$

Sample deformation modes for an automotive duct example are shown in Sec. 7.3.

4.3. *Required number of test points*

In the case of abutting (as opposed to intersecting) B-spline patches, the representation in the parametric coordinate along the patch edge is a polynomial of order N_q. The number of required test point pairs for a B-spline curve could be determined a-priori by considering each non-zero knot-interval. Given N_q+1 distinct test points within the knot vector interval that supports a section of the curve, the polynomial can be matched exactly. 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

$$N_k = N_p + N_q \tag{13}$$

The number of non-zero knot intervals is

$$N_i = (N_k - 1) - 2(N_q - 1) - N_M \tag{14}$$

where N_M is the number of zero knot intervals because of internal multiplicities. From Eq. (13) and Eq. (14), the number of non-zero knot intervals becomes

$$N_i = N_p - N_q + 1 - N_M \tag{15}$$

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

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

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

$$M_{T,E} \geq \max(M_L,\, M_R). \tag{16}$$

The SVD filters out redundant constraints, it is hence not penalising to imposing an excessive number of test points. To allow for non-regular knot-intervals, and avoid a detailed analysis of how mismatched knot intervals either side align, we use

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

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

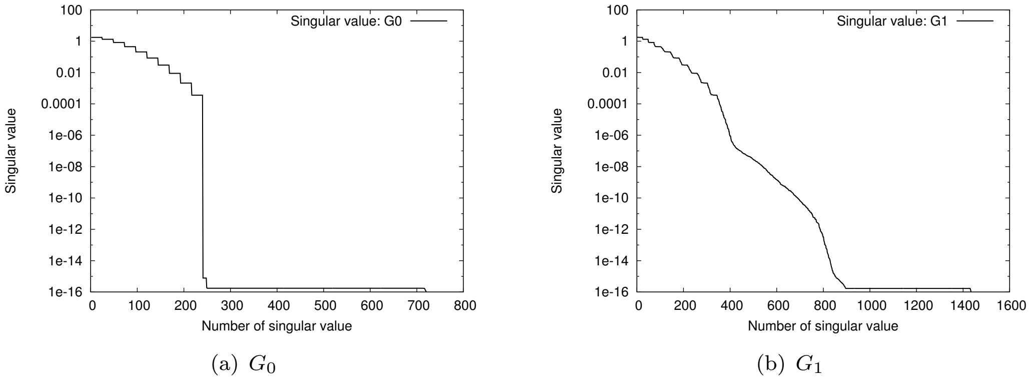
$$M_{T,E} \geq f_T(N_q{+}1)(\max(N_p|_L,\, N_p|_R) - N_q + 1 - N_M). \tag{17}$$

The exactness argument employed for B-Splines does not carry over straightforwardly to intersecting patches and/or 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. A study of the effect of varying the parameter f_T can be found in (Zhang et al. 2016)

4.4. *Effect of cut-off value and effective rank*

Equation (9) indicates the number of design parameters in $\boldsymbol{\alpha}$ is $(4N - r')$. For a specific case, N is a constant, so the number of design parameters is dependent on r' and hence the choice of σ_T.

In the case of G_0 continuity, a distinct drop-off in singular values is typically observed as shown in Fig. 2(a). A suitable choice of the cut-off σ_T is then any value in the range of rapid drop of singular value, with very little effect on the numerical rank r'.



(a) G_0

(b) G_1

Figure 2. Singular values when imposing G_0 (left) and G_1 (right) continuity constraints.

The drop-off in singular values is much more gradual for G_1 and higher continuity requirements, see Fig 2(b). In this case the value of the numerical rank r' will depend on the choice of σ_T. Choosing a higher σ_T will result in modes that have a larger effect on feasibility to be included in the design space. While this enlarges the design space and may lead to lower values of the objective, feasibility may be impaired. In turn, raising σ_T will require that an effective algorithm for constraint recovery can be formulated, see Sec. 5.

Even for cases where a clear drop off in σ can be identified, the scaling of the constraint equations in G affects the scaling of the SVD, choice of σ_T is case dependent. To arrive at a robust computation of σ_T, we propose to determine the numerical rank from $\mathbf{C}\mathbf{C}^T$ or $\mathbf{C}^T\mathbf{C}$ instead of the constraint matrix $\mathbf{C}$. In infinite precision arithmetic, both $\mathbf{C}\mathbf{C}^T$ and $\mathbf{C}^T\mathbf{C}$ share the same rank with $\mathbf{C}$, but $\mathbf{C}^T\mathbf{C}$ provides a more stable

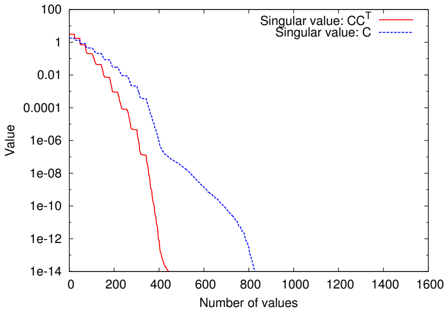


Figure 3. The comparison of singular values of $\mathbf{CC}^T$ and $\mathbf{C}$.

and effective rank for finite-precision arithmetic (Strang 2006). The effective rank is determined as per following procedures in this study:

(1) Compute $\mathbf{C}^* = \mathbf{CC}^T$,
(2) Calculate the singular values σ^* of $\mathbf{C}^*$,
(3) Estimate the cut-off value σ_T as

$$\sigma_T^* = \max(size(\mathbf{C}^*))\sigma_1^*\epsilon, \tag{18}$$

where σ_1^* is the largest singular value of $\mathbf{C}^*$, and ϵ is the machine precision.

Figure 3 presents a comparison between the singular values of $\mathbf{C}^*$ and $\mathbf{C}$ for an S-bend air duct optimisation case (see Section 7). It indicates clearly that the singular values of $\mathbf{C}^*$ drop to machine zero much more rapidly compared to those of $\mathbf{C}$. σ_T^* is around 1.31×10^{-5} in this case, using double precision arithmetic with $\epsilon = 2 \cdot 10^{-16}$.

5. Continuity recovery step

We can distinguish three situation where a design iterate is not feasible, i.e. constraints are violated: a) the initial design is not feasible, achieving feasibility is the first design step; b) the constraint is non-linear, as e.g. is the case for G_1, G_2, and violation arises from a finite-size step in the feasible direction; c) finite-precision arithmetic and/or a choice of a high cut-off σ_T includes modes in the design space which have a small but non-zero effect on feasibility. Fig. 4 demonstrates b) and c).

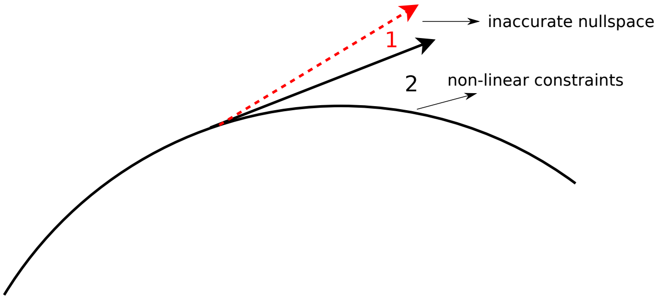


Figure 4. Violation geometric constraints due to nonlinearity and nullspace inaccuracy.

Given a constraint violation of δG, a linear approximation of the required change in control points δP is

$$\frac{\partial G}{\partial P}\delta P_r = \mathbf{C}\delta P_r = -\delta G, \tag{19}$$

which does not have a unique solution. However, the SVD provides us with a basis for the row space in $v_1..v_{r'}$ and the null-space $v_{r'+1}..v_{4N}$. In the row-space, the solution to (19) is unique and produces an update that is orthogonal to the null-space, $\delta P_r \notin \ker(\mathbf{C})$, hence does not affect the design variables to linear approximation.

Given a constraint violation of δG, the correction in row space is

$$\mathbf{V}_R\delta\beta_R + \delta G = 0, \tag{20}$$

where $\mathbf{V}_R$ contains the row space basis vectors $v_1..v_{r'}$.

Except for G_0, (19) is non-linear, thus an iterative method is required to solve it. The iterative recovery steps will continue until the deviation value from exact G_1 is below the chosen threshold value. If design steps are small, it is sufficient to retain the SVD and the basis V_R for the recovery iterations. In practice, G_0 (watertightness) is a constraint that has to be satisfied to machine precision, e.g. to ensure robust meshing, while deviations in G_1, G_2 and other geometric constraints may be acceptable up to the magnitude of manufacturing tolerances. In our implementation we therefore solve (20) first for δG_o, then for the remaining constraints. Demonstration of recovery steps are presented in Section 7.5.

6. Optimisation framework

The geometric elements presented in the preceding sections, namely design space definition, gradient computation, design (tangent) and normal (recovery) steps, can be integrated effectively into the optimisation loop as shown in Fig. 5. A number of advantages arise from this:

CAD output Integrating the CAD description into the design loop enables to start with a CAD geometry, and produce an optimal CAD geometry at convergence. A valid CAD model is produced at any stage, either for inspection and computational steering, or available upon premature termination of the loop.

Minimal user input A design parametrisation through the NURBS control points arises automatically from the initial CAD input, in our experience that typically provides a design space that is sufficiently rich.

Complex constraints A wide variety of constraints can be imposed using the test point framework. As the design space is derived from the constraint Jacobian using SVD, imposing redundant constraints does not impair the convergence of the optimiser.

Smoothness The NURBS surface description is by construction smooth, hence requires no regularisation that could impair optimality.

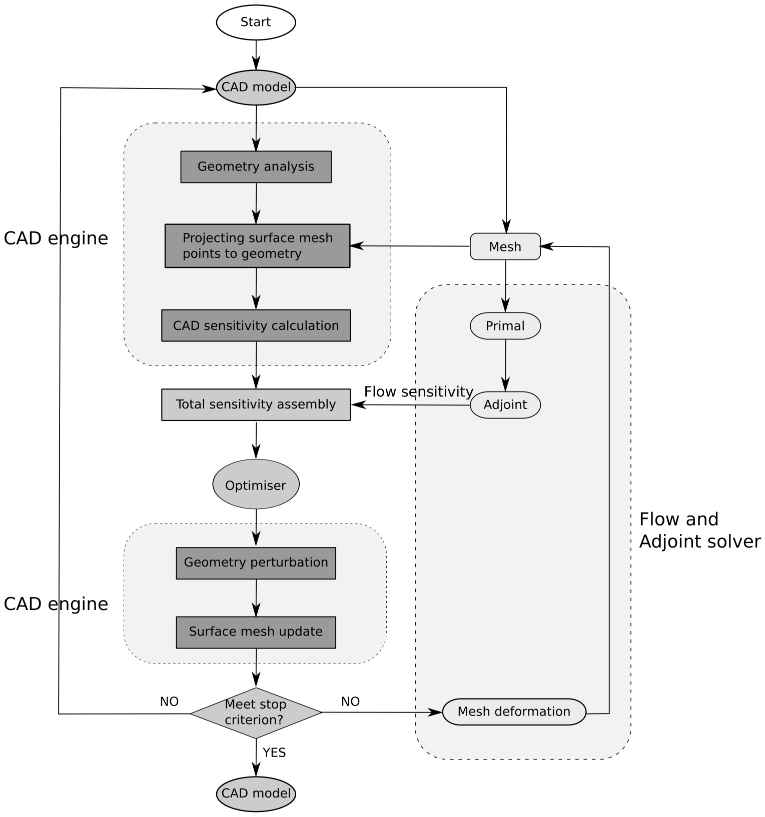


Figure 5. The CAD-based shape optimisation framework.

7. Optimisation testcase: S-bend duct

7.1. *Case description and parametrisation*

An automotive S-bend climatisation duct is optimised to minimise pressure loss. The Reynolds number is 300, based on the height of the duct. The Inlet velocity is 0.1 m/s, with zero back pressure and non-slip walls imposed as boundary conditions. The objective function is the mass-averaged total pressure loss between inlet and outlet,

$$J = \frac{\int_{in} p_{tot}(\mathbf{u}\cdot\mathbf{n})d\mathbf{S} - \int_{out} p_{tot}(\mathbf{u}\cdot\mathbf{n})d\mathbf{S}}{\int_{in}(\mathbf{u}\cdot\mathbf{n})d\mathbf{S}}, \tag{21}$$

where p_{tot} is the total pressure, $\mathbf{u}$ is the velocity vector, $\mathbf{n}$ is the normal direction and $\mathbf{S}$ is the cross sectional area.

7.2. *CAD parametrisation*

The S-Bend benchmark testcase was provided by Volkswagen for the About Flow and IODA benchmarks[1]. It is part of an automotive climate duct, the objective is to minimise total pressure losses. The baseline shape (Fig. 6) consists of 30 surfaces, with inlet and outlet legs being fixed, while the eight patches forming the middle, cranked section are allowed to move. The moveable section consists of 4 wider and 4 narrower patches, with 96 (16×6) and 64 (16×4) control points, respectively, resulting in 640 control points on design surfaces in total.

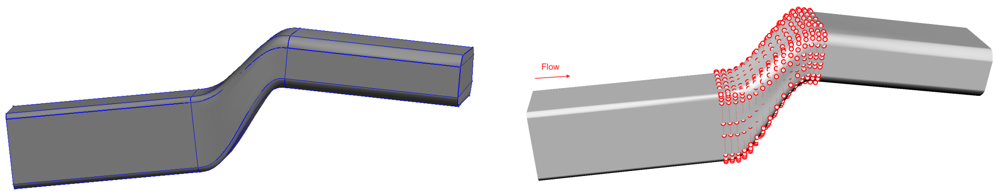


Figure 6. BRep of the S-bend air duct. Left: patch topology. Right: control points of the moveable middle cranked section.

The test point approach is used to impose G_1 continuity across the interfaces of the 8 deformable surfaces. 30 pairs of test points are imposed to each common edge based on the estimate of (17).

G_2 continuity is imposed between moveable and non-moveable surfaces by fixing the first two rows of control points on free patches adjacent to these interfaces, fixing a total of 80 control points.

In this B-spline case, the weights of control points are not perturbed. Each of the moveable 480 control point can move in (x, y, z) directions, resulting in 1449 degrees of freedom (DoF) in total. Performing the SVD with a cut-off value is 10^{-6} results in a numerical rank of $r' = 215$, and a size of the design space of $3N - r' = 1225$.

[1] http://aboutflow.sems.qmul.ac.uk/events/munich2016/benchmark/

7.3. *Deformation modes arising from the SVD*

It is instructive to analyse the structure of the design space. Deformation modes with singular values close to machine precision can be considered to be in the exact nullspace. Fig. 7 shows the magnitude of the normal surface displacement associated with modes α_{1223} and α_{1224}. The modes are contained within a single patch, hence do not affect the constraint equations.

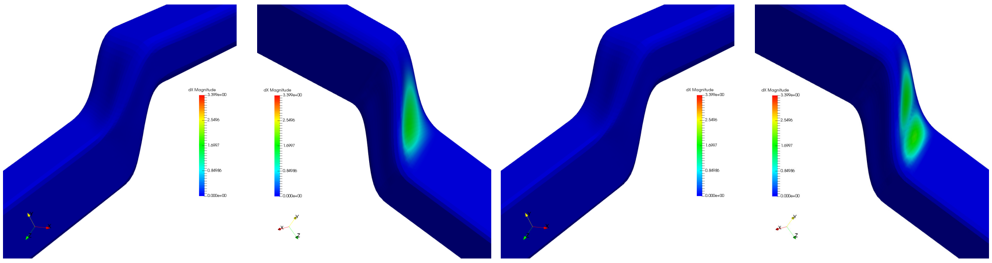


Figure 7. Example deformation modes α_{1223} (left) and α_{1224} (right), which are in the exact nullspace.

Fig. 8 shows the first two deformation modes below the cut-off, α_1 and α_2. Both modes straddle the interfaces either side of the narrow moveable patches, hence strongly affect the G_1 constraint equations along either interface with the wider moveable patches. By construction, the design modes are ordered by their effect on the constraint equations, but not e.g. by their effect on the objective. However, all modes are orthogonal by the unitary nature of $\mathbf{V}$. The convergence rate of NSPCC to the optimum is hence not affected by the number of design variables, i.e. the choice of σ_T.

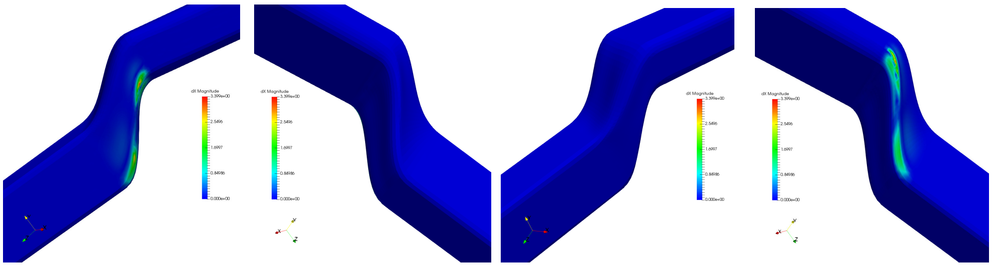


Figure 8. Deformation modes α_1 (left) and α_2 (right), just below the SVD cut-off σ_T.

7.4. *Solvers and parameters*

The in-house CFD and adjoint code GPDE (Jones, Christakopoulos, and Müller 2011) is used GPDE is an incompressible, viscous, steady flow solver on unstructured grids. A discrete adjoint solver is used to obtain the sensitivity of objective function w.r.t. surface nodes, $\frac{\partial J}{\partial \mathbf{x}_S}$. Source-transformation AD with Tapenade (Hascoët and Pascual 2004) is applied to the pressure-correction loop to derive a discrete adjoint. The differentiated routines are then assembled in a hand-written driver code to improve performance, exploiting the fixed-point nature of the primal. The mesh has 41,044 hexahedra and 43,848 nodes as shown in Fig. 9.

A simple steepest descent method optimisation algorithm with Armijo line search is used.

7.5. *Effect of cut-off value on optimisation results*

The choice of SVD cut-off σ_T plays an important role in controlling the size of the design space in cases with the more gradual decay of singular values observed with constraints other than G_0. A range of cutoff values $2 \cdot 10^{-10} \leq \sigma_T \leq \cdot 10^{-4}$ is used, resulting in numerical ranks of $490 \leq r' \leq 341$ as shown in Tab. 1. The effective SVD threshold computed with (18) is $\sigma_T = 9.9 \cdot 10^{-7}$, resulting in a rank of $r' = 374$, hence essentially identical to $\sigma_T = \cdot 10^{-6}$ with a rank of $r' = 375$, and not shown separately in the table.

Table 1. Cut-off values, corresponding numerical rank, and final G_1 deviation.

Cut-off value	Rank	G_1 deviation value
2×10^{-10}	490	8.81×10^{-5}
10^{-7}	387	1.15×10^{-6}
10^{-6}	375	1.17×10^{-6}
10^{-5}	359	2.27×10^{-6}
10^{-4}	341	1.68×10^{-5}

The recovery step is crucial in enabling the use of higher cut-off thresholds σ_T.

Continuity recovery steps (see Sec. 5) are applied at each design iteration with a threshold of $\cdot 10^{-5}$, as suggested by Xu et al. (2013). Without recomputing the nullspace all mid-value choices of σ_T manged to satisfy the constraints, while the most restrictive $\sigma_T = 2 \times 10^{-10}$ and most permissive $\sigma_T = 2 \times 10^{-4}$ did not.

Figure 10 shows the deviation values from exact G_1 with and without recovery steps. These figures indicate clearly that the G_1 deviation value are negligible after applying recovery steps, compared to those without recovery steps. Specifically, in the present optimisation case, two recovery steps in every iteration can bring the G_1 deviation value below the chosen threshold 10^{-5}.

Convergence histories of the objective function corresponding to different cut-off values are given in Fig. 11. A more stringent cutoff of $\sigma_T = 2 \times 10^{-10}$ allows to reduce the objective by only 20%, exhibiting the slowest convergence rate. Choices of $10^{-7} \leq \sigma_T \leq 10^{-4}$ all show much better reduction of the objective with 22.3% and faster convergence.

Figure 12 shows cross sections at the middle of the cranked section for the different cut-off values. Consistent with the behaviour of the objective shown in Fig. 11, a low cut-off ($\sigma_T = 2 \times 10^{-10}$) provides a restricted design space, leading to a limited deformation and consequently a less effective design with higher objective. Results are relatively independent of the choice of cut-off in the range of $10^{-7} \leq \sigma_T \leq 10^{-4}$,

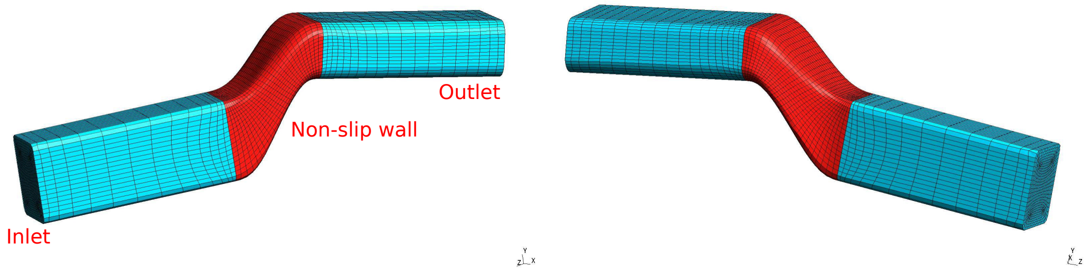


Figure 9. Hexahedral mesh of S-bend air duct.

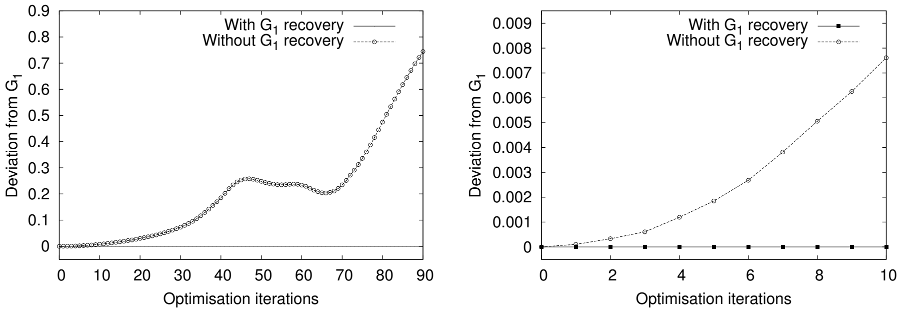


Figure 10. Deviation from exact G_1 continuity with and without recovery steps. Left: during 90 iterations. Right: details in the first 10 iterations.

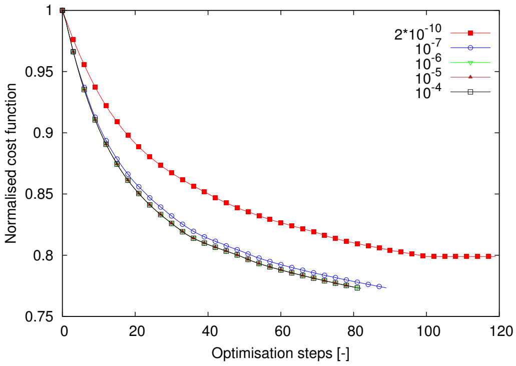
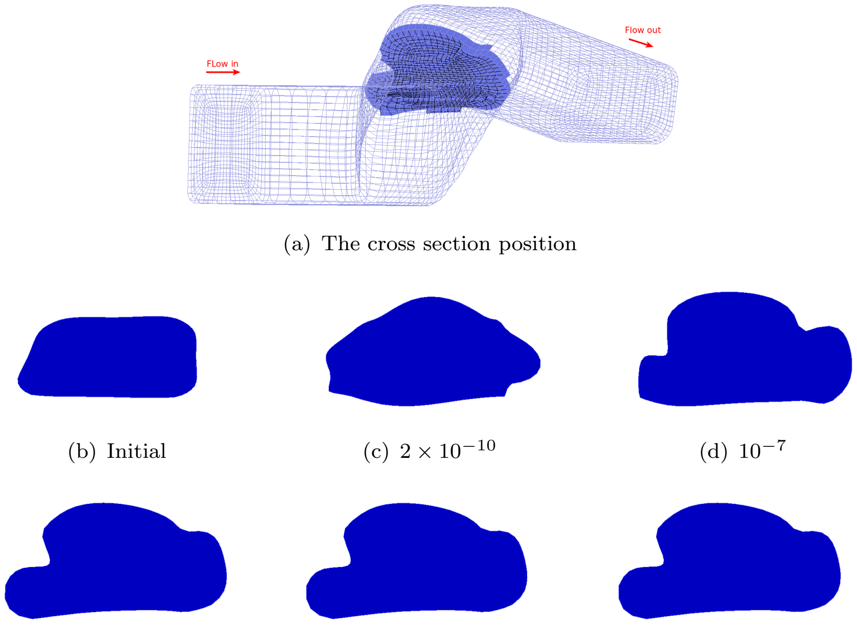


Figure 11. The cost function convergence history when using different cut-off value

hence also including the robust threshold computation of (18), with very similar cross-sectional shapes, convergence and objective.



(a) The cross section position

(b) Initial

(c) 2×10^{-10}

(d) 10^{-7}

(e) 10^{-6}

(f) 10^{-5}

(g) 10^{-4}

Figure 12. Cross section shapes of S-bend before (b) and after (c-g) optimisation

Based on the comparison of cost function convergence histories, cross section shapes and the G_1 deviation values, it can be seen that values of $10^{-7} \leq \sigma_T \leq \cdot 10^{-5}$, offer the best choice among all chosen cut-off values. The value provided by the robust estimation of (18) is in the middle of this interval. Larger cut-off values do not offer additional relevant design modes which would enable a lower objective value, but may compromise constraint recovery. Smaller cut-off values adversely affect the richness of the design space with impaired optimality.

7.5.1. Analysis of S-Bend optimisation results

The geometry before and after optimisation are illustrated in Fig. 13, where a large deformation can be observed on the design surfaces, especially the squeezed-in sides. Constraints were only imposed on continuity between patches and the smoothness of the deformations can clearly be observed. Manufacturing constraints have not been imposed in this case, but could be included in the test-point framework with radius constraints or by imposing G1 constraints between patches at mold split to ensure that the part can extracted from the mold.

To have a better understanding on the result we plot the contour plots of velocity magnitude at different cross sections, as shown in Fig. 14 and 15. The streamlines of the flow are given in Fig. 17. It can be seen that the initial flow field has a strong secondary flow, and the cross-sectional cuts are highly non-uniform in terms of velocity magnitude. This kind of secondary flow in bent ducts is known as Dean vortices (Berger, Talbot, and Yao 1983), which will increase the pressure loss. Figure 15 also indicates that there is a large flow separation at the bottom of the outlet leg, which will additionally gives rise to pressure drop.

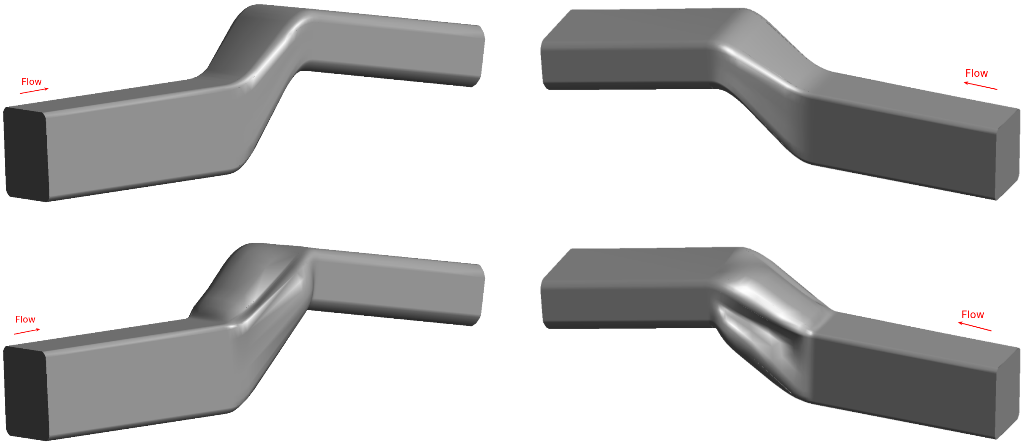


Figure 13. Comparison between the initial (top) and optimised (bottom) shape.

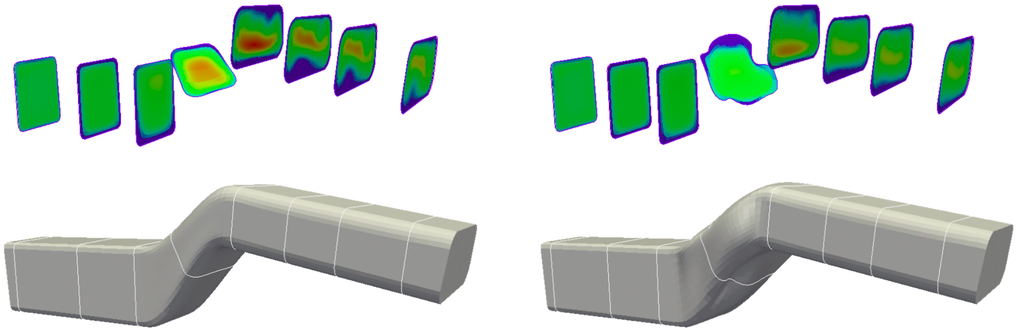


Figure 14. Contour plots of velocity magnitude for the initial (left) and optimised (right) S-bend duct at different cross sections along the flow direction.

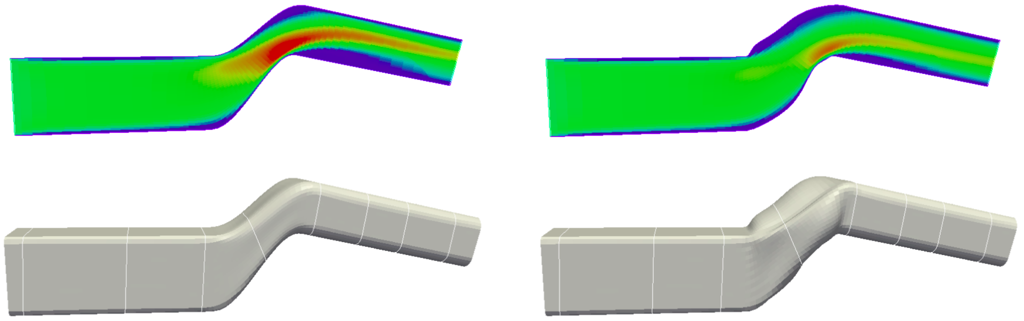


Figure 15. Contour plots of velocity magnitude for the initial (left) and optimised (right) S-bend duct at a cross section parallel to the inlet flow direction.

Two major mechanisms to reduce pressure loss can be identified. Firstly, the cross-sectional area in the bend increases, which in turn reduces the pressure gradient normal to the streamline required to turn the flow, as clearly visible with the lower pressure on the outer radius of the first bend in Fig. 16. This results in increased pressure at the inner bend, a reduced adverse pressure gradient exiting the bend, and hence a reduced separation. As a result, lower total pressure loss is achieved after optimisation, which is confirmed by the pressure distributions along the bend shown in Fig. 16.

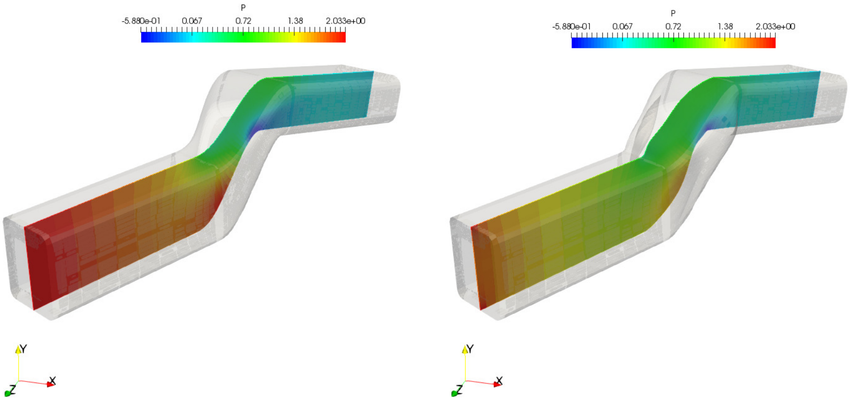


Figure 16. Pressure field distribution before (left) and after (right) optimisation.

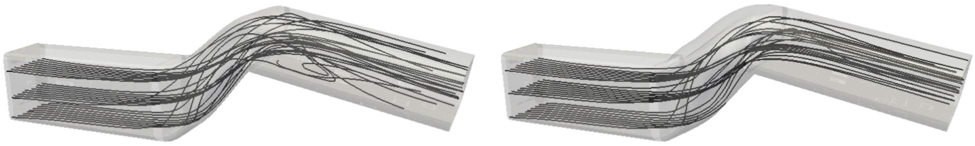
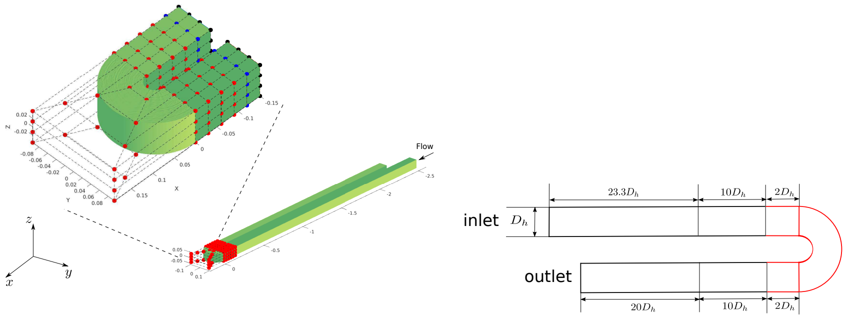


Figure 17. The comparison of streamlines before (left) and after (right) the optimisation.

Secondly, the strongly deformed cross-sectional shape increases wetted area and hence skin friction, but strongly inhibits the formation of secondary flow Dean vortices. The hollowed sides in the optimal shape resembles strake-like shape, as illustrated in Fig. 13. As a consequence of both loss-reducing mechanisms, the streamlines of the optimised shape shown in Fig. 17 exhibit much reduced separation and secondary motion.

8. Optimisation test case: U-bend cooling channel

To demonstrate the effect of including the NURBS weights in the design space, NSPCC is applied to the optimisation of a turbo-machinery U-bend cooling channel (Verstraete et al. 2013). The geometry and main dimensions of this U-bend are shown in Fig. 18.



(a) 3-D view of the U-bend cooling channel and control points of design surfaces.

(b) Main dimensions of the U-Bend.

Figure 18. Parametrisation of the U-bend cooling channel.

The baseline geometry is comprised of a circular bend with straight inlet and outlet legs with square cross-sections. The inlet leg is long enough to guarantee a fully developed flow at the entrance of the circular bend. The U-bend geometry consists of 22 NURBS surfaces in total. Only the bend, which is composed of 8 rectangular and 4 circular surfaces (shown in red in Fig. 18), is deformable. Each of the initially planar surfaces has 24 (6×4) control points, each of the circular surfaces has 16 (4×4) control points. As a result, there are in total 256 control points for the bend. G_2 continuity to the fixed patches is imposed by freezing the first row of control points (shown in black) and restricting the movement of the second row (blue) in the planes of the fixed walls. This results in 192 control points that can move in three and 32 control points can move in two directions. Wtih NURBS weights included in the design space, this resulting in a total of 864 degrees of freedom before G_0, G_1 constraints between moveable patches are imposed.

8.1. *Solver and parameters*

The design objective for this benchmark case is to reduce the total pressure loss in the bend, the flow has Reynolds number: 15000, based on the hydraulic diameter. The in-house solver STAMPS (Müller, Mykhaskiv, and Hückelheim 2018) is employed to solve the flow equations and provide the sensitivity of cost function w.r.t. the mesh coordinates. STAMPS uses a compressible formulation, the testcase is run at Ma=0.1. Turbulence is modelled with the Spalart-Allmaras (SA) model, heat transfer is neglected in this study. The mesh is regular with 167K nodes and 177K hexahedra, boundary layer refinement supports low-Re wall treatment with $y^+ \approx 1$.

Table 2. Computational cost of the NSPCC optimisation workflow, U-Bend with fixed NURBS weights.

Item	Run time (min)	%
Adjoint	2375.15	62.92%
Primal	1194.01	31.63%
NSPCC	153.69	4.07%
Vol. Mesh deformation	28.05	0.74%
Flow sensitivity assembly	24.19	0.64%
Total	**3775.09**	**100%**

Tab. 2 breaks down the computational cost for a typical run. Even though only forward mode differentiation is used for the geometric kernel, its computational time is only about 4% of the total CPU time.

8.2. *Optimisation results*

The BFGS algorithm is chosen as optimiser, which builds up an approximation of the inverse Hessian for a given parametrisation with a fixed set of shape modes. NSPCC on the other hand computes a new design space at every evaluation of the constraint matrix **C** and its SVD. To use BFGS with NSPCC, we hence chose to freeze the design space for a number of iterations, 4 for this testcase, and then re-evaluate **C** and restart BFGS. Although the restart limits the accuracy of the Hessian approximation, BFGS shows improved convergence rate and much lower objectives compared to Steepest Descent (Fig. 19).

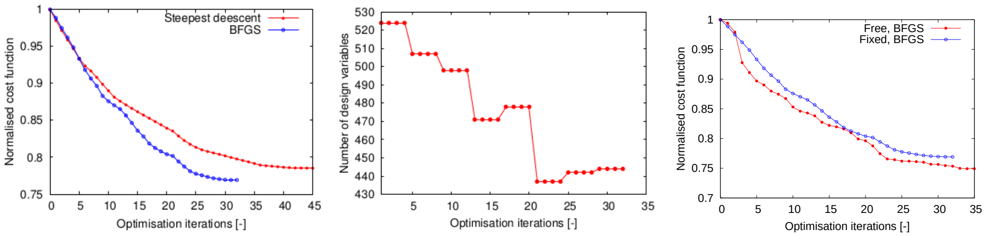
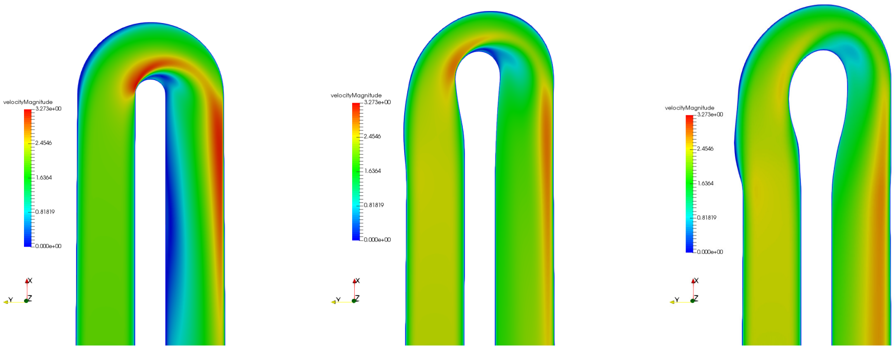


Figure 19. Left; Convergence of objective with Steepest Descent and BFGS with fixed NURBS weights, Middle: size of design space with BFGS, Right: convergence of objective with BFGS, comparison of fixed and free NURBS weights.

Including the NURBS weights as design parameters results in further improvements: with free weights the total pressure loss is reduced by around 25% after 35 iterations, while fixed weights result in a 23.1% reduction.



(a) Initial

(b) Optimised, fixed weights

(c) Optimised, free weights

Figure 20. Shape optimisation of the U-bend cooling channel.

Figure 20 presents the comparison of velocity magnitude in the middle plane before and after optimisation using a parametrisation which fixes the NURBS weights, and comparing this to a case that includes the NURBS weights in the design space. In both variants, the optimal shape significantly reduces the pressure loss in the channel using three major mechanisms. The optimisation with free weights demonstrates three mechanisms which contribute to the reduction in loss. The same mechanisms can be identified in the case with fixed weights, but to a lesser extent.

The major mechanism is the enlargement of the inner bend radius as clearly shown in Fig. 20. The radial pressure gradient of a turning streamline is proportional to v^2/r, furthermore the low pressure at the inner bend of the initial geometry results in a high velocity as visible in the left part of Fig. 21. Enlarging the inner radius results in a dramatic reduction of flow velocity and pressure gradient at the inner bend, see right part of Fig. 21.

The increased minimal pressure in the bend reduces the adverse pressure gradient the flow experiences at the exit of the bend. The flow separation is further reduced by a much more gradual opening of the channel cross section, compared to the case of fixed weights (Fig. 20, middle) or the initial geometry (Fig. 20, left).

The final mechanism is the deformation of the cross-sectional shape of the apex of the bend from a square into a rectangular cross-section which is elongated in the direction of the turning axis. The reduction of the channel width in the radial direction results in a decimation of Dean vortices, and hence a reduction in secondary flow losses.

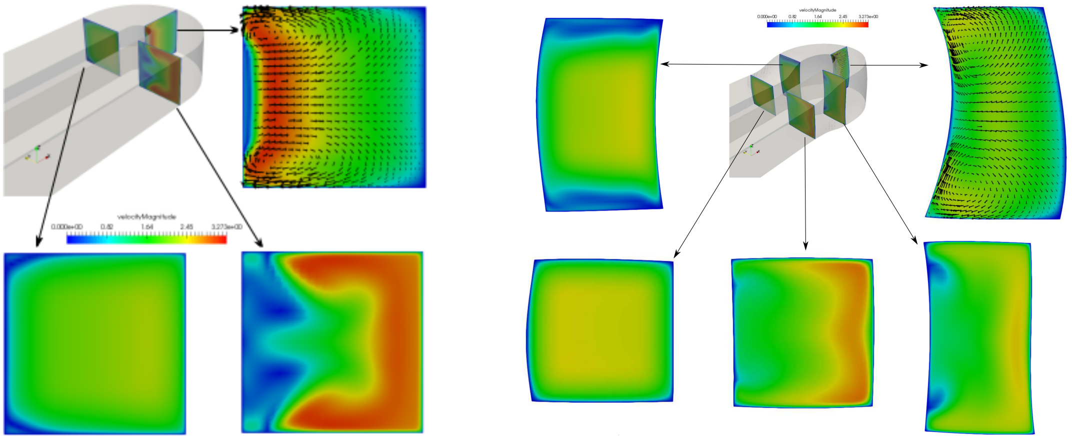


Figure 21. Velocity magnitude at different locations of the optimised geometry, left: initial geometry, right: optimised including NURBS weights.

It can be observed that the inclusion of the NURBS weights in the design space leads to an improvement in flow behaviour. Fig. 19 demonstrates that the case with fixed weights shows the same changes as the case with free weights, but to a reduced extent, which explains that the fixed-weight case achieves only 23.1% reduction in pressure loss, compared to 25% for the free weight case.

9. Conclusion

The NUBRS-based parametrisation approach for CAD-based shape optimisation, NSPCC, has been presented and further investigated and developed. NSPCC derives the design space automatically from a boundary representation (BRep) of the geometry. The design space is derived from the SVD of the Jacobian of a set of discrete constraint equations at sets of testpoints.

The paper investigated a number of parameters that need to be determined for the application of the method. A formula for estimating the required number of testpoints at patch interfaces has been given. It is based on the order of B-spline degree and the number of control points on either surface, hence can be evaluated straightforwardly. A robust method for estimating a suitable cut-off for the singular value decomposition has been presented. The method does not require to manually determine any scalings, and is hence robustly and automatically applicable as shown in an industrial testcase. The resulting design modes from the SVD have been evaluated and analysed, explaining the good preconditioning behaviour of the approach. A methodology to achieve or recover feasibility of geometric constraints has been shown. It exploits the row-space which is a free by-product of computing the SVD for the design space. The NSPCC approach produces its own design space automatically, including number and character of design modes. The paper has presented a successful extension to use quasi-Newton approaches with the NSPCC approach by retaining the SVD for a number of design steps.

The developments have been demonstrated on two industrial duct cases, a laminar automotive S-bend air duct and a turbulent U-bend turbo-machinery cooling channel. In both cases NSPCC worked fully automatically from the BRep and achieved reductions in pressure loss by 22-25%. Analysis of the flow in the optimised shape confirms that the adjoint sensitivity calculation enables an accurate trade-off between competing mechanisms such as primary wall friction and secondary flow losses.

The NSPCC parametrisation is able to effectively express a very rich design space. Although each case comprises hundreds of modes, by construction they are orthogonal and good convergence in 35 design iterations can be achieved. The inclusion of the NURBS weights as design variables, and the use of quasi-Newton optimisers has been shown to lead to superior results.

Acknowledgement

The authors would like to thank the funding from China Scholarship Council (No. 201306230097, No. 201206280018) and Queen Mary University of London for this research. The third author acknowledges the support from the European Union's Seventh Framework Programme (FP7/2007-2013) under REA grant agreement no 371006.

References

Andrés, E., P. Bitrián, M. Martin, and M. Widhalm. 2010. "Preliminary comparison between two CAD-based aerodynamic shape optimization approaches using adjoint methods for fast gradient computation." In *2nd International Conference on Engineering Optimization*, EngOpt 2010, Lisbon, Portugal, Sep.

Banovic, M., O. Mykhaskiv, S. Auriemma, A. Walther, H. Legrand, and J.-D. Müller. 2017. "Automatic Differentiation of the Open CASCADE Technology CAD System and its coupling with an Adjoint CFD Solver." *Optimization Methods and Software* 33 (4-6): 813–28.

Berger, SA, L Talbot, and LS Yao. 1983. "Flow in Curved Pipes." *Annual review of fluid mechanics* 15 (1): 461–512.

Brock, W.E., C. Burdyshaw, S.L. Karman, V.C. Betro, B. Hilbert, W.K. Anderson, and R. Haimes. 2012. "Adjoint-based Design Optimization Using CAD Parameterization Through CAPRI." In *50th AIAA Aerosp. Sc. Meeting*, .

Foster, L.V., and T.A. Davis. 2013. "Algorithm 933: Reliable Calculation of Numerical Rank, Null Space Bases, Pseudoinverse Solutions, and Basic Solutions Using SuitesparseQR." *ACM Transactions on Mathematical Software (TOMS)* 40 (1): 7.

Fudge, D.M., D.W. Zingg, and R. Haimes. 2005. "A CAD-free and a CAD-based geometry control system for aerodynamic shape optimization." *AIAA CP 05-0451* .

Giles, M. B., M. C. Duta, J.-D Müller, and N. A. Pierce. 2003. "Algorithm Developments for Discrete Adjoint Methods." *AIAA Journal* 41 (2): 198–205.

Giles, M. B., and N. A. Pierce. 2000. "An Introduction to the Adjoint Approach to Design." *Flow, Turb. Comb.* 65: 393–415. Also Oxford University Computing Laboratory NA report 00/04.

Hascoët, L., and V. Pascual. 2004. *TAPENADE 2.1 user's guide.* Technical Report 0300. INRIA. http://www.inria.fr/rrrt/rt-0300.html.

Hicks, R.M., and P.A. Henne. 1978. "Wing design by numerical optimization." *Journal of Aircraft* 15: 407–412.

Jakobsson, S., and O. Amoignon. 2007. "Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization." *Computers and Fluids* 36 (6): 1119 – 1136. http://www.sciencedirect.com/science/article/pii/S0045793006001320.

Jameson, A. 1988. "Aerodynamic Design via Control Theory." *Journal of Scientific Computing* 3: 233–260.

Jameson, A., and A. Vassberg. 2000. "Studies of alternative numerical optimization methods applied to the brachistochrone problem." *Computational Fluid Dynamics Journal* 9 (3).

Jaworski, A., and J.-D. Müller. 2008. “Toward modular multigrid design optimisation.” In *Lecture Notes in Computational Science and Engineering*, edited by C. Bischof and J. Utke, Vol. 64, ”New York, NY, USA”, 281–291. Springer.

Jones, D., F. Christakopoulos, and J.-D. Müller. 2011. “Preparation and Assembly of Adjoint CFD codes.” *Computers and Fluids* 46 (1): 282–286.

Martín, M. J., E. Andrés, M. Widhalm, P. Bitrián, and C. Lozano. 2012. “Non-uniform rational B-splines-based aerodynamic shape design optimization with the DLR TAU code.” *Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering* 226 (10): 1225–1242.

Masters, D.A., N.J. Taylor, T. Rendall, C.B. Allen, and D.J. Poole. 2017. “A geometric comparison of aerofoil shape parameterisation methods.” *AIAA Journal* 55 (5): 1575–1589.

Mousavi, A., P. Castonguay, and S. Nadarajah. 2007. “Survey of shape parameterization techniques and its effect on three-dimensional aerodynamic shape optimization.” In *18th AIAA CFD conference*, AIAA CP–2007–3837.

Müller, J.-D., O. Mykhaskiv, and J. Hückelheim. 2018. “STAMPS: a Finite-Volume Solver Framework for Adjoint Codes Derived with Source-Transformation AD.” *2018 Multidisciplinary Analysis and Optimization Conference* AIAA-CP 2928-2018.

Myhkaskiv, O. 2019. “Optimal shape design with automatically differentiated CAD parametrisations.” PhD diss., Queen Mary University of London.

Mykhaskiv, O., M. Banovic, S. Auriemma, A. Walther, and J.-D. Mueller. 2018. “NURBS-based and Parametric-based shape optimisation with differentiated CAD kernel.” *Computer-Aided Design and Applications* 1–11.

Mykhaskiv, O., P. Mohanamuraly, J.-D. Müller, S. Xu, and S. Timme. 2017. “CAD-based shape optimisation of the NASA CRM wing-body intersection using differentiated CAD-kernel.” In *35th AIAA Applied Aerodynamics Conference*, .

Piegl, Les, and Wayne Tiller. 2012. *The NURBS Book.* Springer Science & Business Media.

Pironneau, O. 1974. “On Optimum Design in Fluid Mechanics.” *Journal of Fluid Mechanics* 64: 97–110.

Robinson, T.T., C.G. Armstrong, H.S. Chua, C. Othmer, and T. Grahs. 2012. “Optimizing Parameterized CAD Geometries Using Sensitivities Based on Adjoint Functions.” *Computer-Aided Design & Applications* 9 (3): 253–268.

Samareh, J.A. 2001. “Survey of Shape Parameterization Techniques for High-Fidelity Multidisciplinary Shape Optimization.” *AIAA Journal* 39 (5): 877–884.

Samareh, J.A. 2004. “Aerodynamic shape optimization based on free-form deformation.” *AIAA* paper 4630: 1–13. DOI:10.2514/6.2004-4630.

Schmidt, S., and V. Schulz. 2009. “Impulse Response Approximations of Discrete Shape Hessians with Application in CFD.” *SIAM Journal on Control and Optimization* 48 (4): 2562–80.

Shahpar, Shahrokh. 2011. “Challenges to overcome for routine usage of automatic optimisation in the propulsion industry.” *The Aeronautical Journal* 115 (1172): 615–25.

Strang, G. 2006. *Linear Algebra and Its Applications.* Cengage Learning.

Verstraete, Tom, Filippo Coletti, Jérémy Bulle, Timothée Vanderwielen, and Tony Arts. 2013. “Optimization of a U-Bend for Minimal Pressure Loss in Internal Cooling Channels Part I: Numerical Method.” *Journal of Turbomachinery* 135 (5): 051015.

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

Xu, Shenren, Wolfram Jahn, and Jens-Dominik Müller. 2013. “CAD-based shape optimisation with CFD using a discrete adjoint.” *International Journal for Numerical Methods in Fluids* 74 (3): 153–68.

Zhang, Xingchen, Rejish Jesudasan, and Jens-Dominik Müller. 2019. “Adjoint-Based Aerodynamic Optimisation of Wing Shape Using Non-uniform Rational B-Splines.” In *Computational Methods in Applied Sciences*, Vol. 49, 143–158. Springer.

Zhang, Xingchen, Yang Wang, Mateusz Gugala, and Jens-Dominik Müller. 2016. “Geometric

Continuity Constraints for Adjacent NURBS Patches in Shape Optimisation." In *proceedings of VII European Congress on Computational Methods in Applied Sciences and Engineering (ECCOMAS Congress 2016)*, .