14th World Congress on Computational Mechanics (WCCM)
ECCOMAS Congress 2020)
Virtual Congress: 11-–15 January 2021
F. Chinesta, R. Abgrall, O. Allix and M. Kaliske (Eds)

CAD-BASED ADAPTIVE SHAPE PARAMETERISATION FOR AERODYNAMIC SHAPE OPTIMISATION

REJISH JESUDASAN[1] AND JENS-DOMINIK MÜLLER[1]

[1] PhD student
SEMS, Queen Mary University of London
rejishmar30@gmail.com.

[2] Reader in Computational Fluid Dynamics and Optimisation
SEMS, Queen Mary University of London
j.mueller@qmul.ac.uk.

Key words: Shape optimisation, adjoint method, CAD-based parameterisation

Abstract. Non-Uniform Rational B-Splines (NURBS) have become the industrial standard to represent and exchange a CAD geometry between CAD/CAE systems. CAD-based shape parameterisation uses parameters of a CAD model to modify the shape which allows to integrate a CAD model into the design loop. However, feature-trees of typical commercial CAD systems are not open and obtaining exact derivatives for gradient-based optimisation methods is not possible. Using the CAD-based NSPCC approach a designer can deform multiple NURBS patches in the design loop without violating geometric and/or thickness constraints. The NSPCC approach takes CAD descriptions as input and perturbs the control points of the NURBS boundary representation to modify the shape.

In this work, an adaptive NSPCC method is proposed where the optimisation begins with a coarser design space and adapts to finer parametrisation during the design process where more shape control is needed. The refinement sensor is based on a comparison of smoothed node-based sensitivity compared to its projection onto the shape modes of the current parametrisation. Both static and adaptive parametrisation methods are coupled in the adjoint-based shape optimisation process to reduce the total pressure loss of a turbine blade internal cooling channel.

The discrete adjoint flow solver STAMPS is used to compute the flow fields and their derivatives w.r.t. surface node displacements. The shape derivatives for gradient-based optimisation are obtained by application of reverse mode AD to the NSPCC CAD kernel. Since a CAD model is kept inside the design loop, the resulting optimal shape is directly available in CAD for further analysis or manufacturing. Based on the analysis regarding quality of the optima and rate of convergence of the design process adaptive NSPCC method outperforms static NSPCC approach.

1 Introduction

Industrial work flow relies heavily on CAD models and requires all disciplines to share the same geometry in the design process, which reduces the CAD integration effort to carry out production. However, there are some issues that prevent the use of CAD models in a gradient-based shape optimisation process. These issues are:

1. Difficulties of computing sensitivity of the geometric shape to the design variables which is an essential measure for gradient based shape optimisation.
2. Poor shape parametrisation that cannot capture important shape modes obtained from an adjoint CFD solver.
3. Lack of functionalities for handling geometrical and manufacturing constraints.

One available option is to use finite differences [24] to calculate CAD sensitivities. This approach suffers a potential lack of accuracy in regions of high curvature. On the other hand, if the source code is available as in the case of open-source CAD engine OpenCASCADE, CAD sensitivities can be obtained by differentiating a complete CAD kernel [4]. As an alternative, the authors developed a lightweight CAD kernel 'NURBS-based Parameterisation with Complex Constraints (NSPCC)'. The implementation was done in Fortran to support the application of source-transformation AD using Tapenade[9], which results in extremely efficient derivative code.

Choice of shape parameterisation is crucial which determines the set of shape modes that can be captured during the design process [26]. Since the best achievable design belongs to this set, shape parameterisation method influence the final solution and the rate of convergence of the optimisation process. Most shape parameterisation methods require manual setup [26, 32]. Setting up auxiliary grids for lattice-based methods, such as e.g. auxiliary grids with Hicks-Henne bumps on airfoils [11] or stacked spline curves for turbomachinery blades [28], involve substantial effort and are difficult to extend to complex geometries. To reduce complexities, the designer often ends up with choosing small number of design variables and the design space that capture all the possible shape modes remain fixed throughout the optimisation. This traditional static parameterisation approach restricts the generation of superior designs outside the fixed envelope and the final solution highly depends on how the designer parameterise the baseline shape. To obtain superior designs, the designer should terminate the design process and reparameterise the shape manually in a periodic manner.

On the other hand, one can consider the displacement of every surface grid node as a design variable [15, 12, 5], called as the node-based method. This method offers richest design space for shape optimisation. However this design space is very rich and contains unwanted oscillatory modes which needs to be filtered by the use of surface regularisation method. Additional surface regularisation is necessary, and implicit [14] as well as explicit [15] Sobolev smoothing methods have been proposed. However, both of them require a smoothing coefficient and/or number of smoothing iterations to control the amount of smoothing applied on the shape.

Hradil et al. [13] proposed an adaptive parameterisation method based on FFD approach and Masters et al [18] presented an adaptive subdivision surfaces for shape optimisation. The results also show design acceleration however optimised geometry is not available in CAD format for further analysis and manufacturing. Agarwal et al. [1] presented a CAD-based adaptive parameterisation method by adding multiple CAD features to a parametric CAD model. Results showed that the inserted CAD features are not good enough to capture superior designs hence it affects the rate of convergence and leads to a sub-optimal solution. This strongly justifies the need for adding more design variables that can capture important shape modes for design improvement.

NURBS have become the de-facto industry standard for data exchange between CAD systems and offers local shape modification property. The NSPCC method derives parametrisation directly from the boundary representation (BRep) of a CAD-model [21]. The BRep, in the typical standardised STEP format, represents the shape in a number of NURBS patches. NSPCC approach uses control points of the NURBS patches to deform geometry in the design process. Since the shape parametrisation is derived from the BRep of a CAD model, construction history of a CAD model is not required hence only subset of functionalities can be added. This reduces the computational effort required to parameterising the existing CAD models. In this work, NSPCC method is extended to handle adaptive design space in the shape optimisation using knot insertion algorithm.

The important contribution of NSPCC to CAD-based parametrisation based on BRep is the formulation of geometric constraints, e.g. $G_0 - G_2$ continuity at NURBS patch interfaces [21] or box, radius and thickness constraints [22]. In the NSPCC approach, surface node sensitivities are projected onto the control points of the design surface. In general, the Boundary Representation (BRep) of a baseline shape exported by a CAD system often depends on the local curvature of a shape. For example, region with a large curvature may contain large density of control points and a flat region may contain sparse control points. If a control net is too coarse then the important gradient modes may not be projected effectively which restricts the generation of superior designs outside this fixed envelope [16]. On the other hand, finer distribution of control points may leads to inefficient navigation in the design space and may converge to local minima. Therefore, the optimal distribution of control points suitable for shape optimisation is difficult to obtain a-prior to the shape optimisation process.

The focus of the present work is three-fold.

1. First, an adaptive NSPCC parameterisation method using knot insertion algorithm is proposed to refine the control net distributions without modifying the geometry. The adaptive refinement is driven by node-based sensitivity information, therefore the control points are added only in the region where larger design improvement can be achieved when the optimiser has reached sufficient convergence. As a consequence the adaptive NSPCC replaces user in the design loop as design variables are automatically created based on the adaptive refinement.

2. Second, the entire design chain is reverse differentiated using source transformation Algorithmic Differentiation tool which is essential for handling large number of design variable in the design process.

3. Third aim is to demonstrate the efficiency of the proposed parameterisation method in an industrial design chain by minimising the mass averaged total pressure loss of the VKI U-Bend benchmark case using one-shot optimisation strategy

The paper is organised as follows. Brief details about the NSPCC method are presented in Sec. 2. The proposed adaptive NSPCC method is presented in Sec. 3. Discrete adjoint formulation to compute CFD sensitivity and shape sensitivity are presented in Sec. 4. The aerodynamic shape optimisation of the VKI U-Bend geometry including flow solver validation, mesh convergence study, CAD sensitivity verification are discussed in Sec. 5. An optimisation framework developed using the proposed adaptive

NSPCC method is presented in Sec. 6. In Sec. 7, influence of shape parametrisation is presented with analysis of the results of the optimisation and the effectiveness of the methodology. Conclusions and future work are presented in Section 8.

2 NURBS-Based Parametrisation with Complex Constraints (NSPCC)

2.1 Non-Uniform Rational B-Splines (NURBS)

The BRep, in the typical standardised STEP format, represents the geometry as a collection of NURBS patches. In the BRep, surfaces and curves are represented by NURBS which is a current industrial standard in CAD systems. NURBS patches are bivariate parametric surface defined by piecewise rational functions with degree p and q in the parameter direction s and t respectively. Mathematically it is written as [23],

$$\mathbf{S}(s,t) = \frac{\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\mathcal{B}_{i,p}(s)\mathcal{B}_{j,q}(t)\mathbf{P}_{i,j}w_{i,j}}{\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\mathcal{B}_{i,p}(s)\mathcal{B}_{j,q}(t)w_{i,j}}, \tag{1}$$

where $\mathbf{P}_{i,j}$ are the control points which form a bidirectional control polygon with n and m are the total number of control points along each parameter direction s and t respectively. $\mathcal{B}_{i,p}(s), \mathcal{B}_{j,q}(t)$ are the B-spline basis functions defined on the knot vectors $S = \{\bar{s}_0, \dots, \bar{s}_r\}$ and $T = \{\bar{t}_0, \dots, \bar{t}_k\}$ with r and k are the number of knots along each parameter direction respectively. For the current work, we use a non-periodic or clamped type of knot vectors as given in the Eqn. 2 and Eqn. 3 in which the first and last knots have multiplicity of $p+1$ and $q+1$ in S and T knot vectors respectively.

$$S = \{\underbrace{a,\dots,a}_{p+1},\bar{s}_{p+1},\dots\dots,\bar{s}_{r-p-1},\underbrace{b\dots,b}_{p+1}\} \tag{2}$$

$$T = \{\underbrace{\bar{a},\dots,\bar{a}}_{q+1},\bar{t}_{q+1},\dots\dots,\bar{t}_{s-q-1},\underbrace{\bar{b}\dots,\bar{b}}_{q+1}\} \tag{3}$$

2.2 Deformation with Geometric Constraints

The NSPCC approach [30, 31] employs control points of a conformal patch topology to deform shape in the design process. Hence, a finite control point displacement $\mathbf{P}_{i,j}$ on or near a patch interface affects the surface continuities such as G_0 (no gaps), G_1 (tangency) and G_2 (curvature) which is clearly visible in Figure 1.

(a) Original NURBS

(b) Perturbed NURBS

Figure 1: Shape deformation of a NURBS patch with its control net

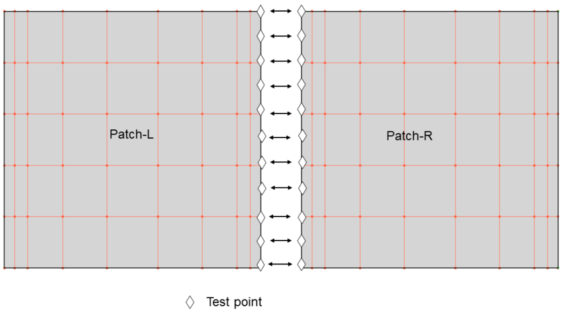


Figure 2: Test points along a common edge and corresponding control net of adjacent NURBS patches

To maintain patch continuity, constraints are need to be imposed on a common edge. This can be achieved by the use of test points approach in which constraint equations are evaluated discretely at each pair of test points deployed linearly along a common edge of the adjacent patches. Typical distribution of test points at a common edge and corresponding control net of adjacent patches are shown in Fig. 2. To ensure G_0 and G_1 continuity along an edge, the following constraint equations G_0 (Eqn. 4) and G_1 (Eqn. 5) are evaluated discretely at each pair of test points:

$$G_0 = X_{p,L} - X_{p,R} = 0 \tag{4}$$

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

where $X_{p,L}$ and $X_{p,R}$ are the positional coordinates, $\vec{n}_L$ and $\vec{n}_R$ are the unit normal tangent plane, suffix L and R corresponds to Patch-L and Patch-R respectively. Then we need to assemble Jacobian for each of the constraint equation and after linearization we obtain the following linear system of equations [21],

$$\mathbf{C}\delta\mathbf{P} = 0. \tag{6}$$

The matrix $\mathbf{C}$ is called constraint matrix and $\delta\mathbf{P}_i$ denotes the displacement of x, y, z coordinates of N control points and in vector form both terms are written as,

$$\mathbf{C} = \begin{bmatrix} \frac{\partial G_{0,1}}{\partial \mathbf{P}_1} & \frac{\partial G_{0,1}}{\partial \mathbf{P}_2} & \cdots & \frac{\partial G_{0,1}}{\partial \mathbf{P}_N} \\ \vdots & \ddots & \vdots & \\ \frac{\partial G_{0,M_0}}{\partial \mathbf{P}_1} & \frac{\partial G_{0,M_0}}{\partial \mathbf{P}_2} & \cdots & \frac{\partial G_{0,M_0}}{\partial \mathbf{P}_N} \\ \cdots & \cdots & \cdots & \\ \frac{\partial G_{1,1}}{\partial \mathbf{P}_1} & \frac{\partial G_{1,1}}{\partial \mathbf{P}_2} & \cdots & \frac{\partial G_{1,1}}{\partial \mathbf{P}_N} \\ \vdots & \ddots & \vdots & \\ \frac{\partial G_{1,M_1}}{\partial \mathbf{P}_1} & \frac{\partial G_{1,M_1}}{\partial \mathbf{P}_2} & \cdots & \frac{\partial G_{1,M_1}}{\partial \mathbf{P}_N} \end{bmatrix} = \begin{bmatrix} \mathbf{G}_0^j \\ \cdots \\ \mathbf{G}_1^j \end{bmatrix}, \delta\mathbf{P} = \begin{bmatrix} \delta\mathbf{P}_1 \\ \delta\mathbf{P}_2 \\ \vdots \\ \delta\mathbf{P}_N \end{bmatrix} \tag{7}$$

where M_0 and M_1 correspond to the total number of G_0 and G_1 constraint equations respectively and j is the edge index. By assembling the matrix $\mathbf{C}$ as given in the Eqn. 7 we can impose different continuity constraints to different edges. The matrix has a total of M_c rows where M_c corresponds to total number of constraint equations with N columns which is the total number of control points.

2.3 Design space

Using a projected gradient approach, the control point perturbations (design space) has to lie in the null space of $\mathbf{C}$ and the design modes are the N basis vector of the null space and determined using Singular Value Decomposition (SVD).

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

where $\mathbf{U}$ is the $M_c \times M_c$ unitary matrix, Σ is an $M_c \times N$ diagonal matrix with positive real numbers on the diagonal σ_i called singular values of $\mathbf{C}$ and $\mathbf{V}^T$ represents a $N \times N$ unitary matrix. The number of non-zero singular values in Σ determines the theoretical rank r of the constraint matrix, $\mathbf{C}$ and last $(N-r)$ columns of the matrix $\mathbf{V}$ span the exact null space of $\mathbf{C}$. With the presence of non-linear constraints singular values show a gradual decrease rather than sudden drop to zero. Therefore in NSPCC a cut-off threshold frequency value σ_C is used to determine the rank of the matrix $\mathbf{C}$ which is denoted as numerical rank r' and the corresponding nullspace associated with the numerical rank r' is termed as numerical nullspace and it is denoted as Ker($\mathbf{C}$). Each column in Ker($\mathbf{C}$) gives a deformation mode that satisfies the constraints and orthogonal to each other. Therefore the resultant control point perturbations are computed as the linear combination of the columns of numerical nullspace which offers richest design space for the shape optimisation and it is written as,

$$\delta\mathbf{P} = \sum_{k=1}^{N-r'} v_{k+r'}\delta\alpha_k = \mathrm{Ker}(\mathbf{C})\delta\alpha. \tag{9}$$

where $\delta\alpha_k$ with $k = 1,2,....N-r'$ are the perturbations to design variables and v_{k+r} are the columns of the numerical nullspace. Hence the size of the design space is $N-r'$ which is determined by the total number of control points and the threshold frequency σ_C.

3 Adaptive NSPCC

The adaptive NSPCC method progressively adds more control points on the design surface using knot insertion algorithm. One can insert a knot in either s or t parameter direction or in both parameter direction. Each knot insertion adds a row or column of control points and also modifies the neighboring control points. Therefore it is important to determine when to terminate the current optimisation and which knot value to insert. Details of the knot insertion algorithm can be found in [23]. The adaptive refinement consists of two steps which includes Refinement trigger and Refine.

3.1 Refinement trigger

Refinement trigger monitors the design progress and determines when to terminate the current design space level. This step has to be performed automatically without designer in the loop. In this work, the refinement trigger proposed by Anderson et al [3] is used which triggers the refinement when the rate of convergence of the objective function with respect to the design iteration falls below some fraction of the maximum attained in the current design level,

$$\frac{\triangle J_i}{max(\triangle J_d)} < \varepsilon_d \tag{10}$$

where $\triangle J_i = J_{i-1} - J_i$, d is the design space level and $\varepsilon_d = 0.1$ is a cutoff parameter.

3.2 Refine

This step computes the suitable knot value $\tilde{s}$ and $\tilde{t}$ to insert in both s and t parameter direction respectively. Zingg et al. [8] used knot insertion algorithm to refine B-spline volumes in shape optimisation and proposed to insert a new knot arbitrarily in each knot span of a knot vector. Martins et al. [10] also used knot insertion to refine FFD frame and insert a new knot in the middle of each knot span. This may be suitable for simple geometries however for a complex geometry with large number of patches inserting a knot in each knot span without any additional measure may leads to the addition of unnecessary control points on the design surface. Therefore, a suitable adaptation criterion is needed to precisely enrich the control point distribution locally on the design surface.

Once the refinement trigger terminates the design process, magnitude of the adjoint node sensitivities $(\mathcal{G} = \frac{\partial J}{\partial X_s})$ are used to identify the region of interest to refine control points. To raise the regularity of computed gradients, an additional explicit weighted Laplacian smoothing method is used. Thus the smoothed gradient $(\bar{\mathcal{G}})$ is obtained as,

$$\bar{\mathcal{G}}^{y+1} = \mathcal{G}^{y} + \beta\mathcal{U}(\mathcal{G}) \tag{11}$$

y is the number of smoothing iteration, β is the smoothing parameter and $\mathcal{U}$ is the umbrella operator. Then the surface mesh points in each patch with large magnitude of the smoothed adjoint sensitivities are identified. Then three different samples of mesh points (z_1, z_2, z_3) are selected in total of n_m number of surface mesh points in a patch $(z_1 = \frac{n_m}{2}, z_2 = \frac{n_m}{4}, x_3 = \frac{n_m}{8})$. Then the knot value to be inserted in a patch is calculated as the average value of parametric coordinates s and t of the selected mesh points as given in Eqn. 12 and 13. As a result, three different sets of new knot values for each patch are obtained. They are, $(s_1^{\kappa}, t_1^{\kappa}), (s_2^{\kappa}, t_2^{\kappa}), (s_3^{\kappa}, t_3^{\kappa})$.

$$\tilde{s}_1^{\kappa} = \sum_{i=0}^{z_1} \frac{s_i^{\kappa}}{z_1}, \quad \tilde{s}_2^{\kappa} = \sum_{i=0}^{z_2} \frac{s_i^{\kappa}}{z_2}, \quad \tilde{s}_3^{\kappa} = \sum_{i=0}^{z_3} \frac{s_i^{\kappa}}{z_3} \tag{12}$$

$$\tilde{t}_1^{\kappa} = \sum_{i=0}^{z_1} \frac{t_i^{\kappa}}{z_1}, \quad \tilde{t}_2^{\kappa} = \sum_{i=0}^{z_2} \frac{t_i^{\kappa}}{z_2}, \quad \tilde{t}_3^{\kappa} = \sum_{i=0}^{z_3} \frac{t_i^{\kappa}}{z_3} \tag{13}$$

κ is the patch ID, z_1, z_2, z_3 are the number of selected mesh points used in the average and n_m represents the total number of mesh points in a patch. Knot insertion with these set of knot values creates three refined control nets without modifying the geometry. Note that there exists an infinite number of possibilities, however in this work only three refined control nets are obtained at the end of each optimisation level. To determine which updated control net to choose for the next design level, the adaptation metric proposed by Martins et al. [10] is used.

$$AM = \frac{1}{2}\sum_{i=1}^{N}\left(\frac{\partial J}{\partial \mathbf{P}_i} - \sum_{j=1}^{N_g}\frac{\partial G_j}{\partial \mathbf{P}_i}\right)^2 \tag{14}$$

where N represents the total number of control points $\mathbf{P}_i$, N_g is the total number of geometric constrains including G_0 and G_1, $\frac{\partial G_j}{\partial \mathbf{P}_i}$ represents the Jacobian of each constraint equation j with respect to each control point P_i. This metric is evaluated for each control net and choose the one which has larger adaptation metric value. Note that gradient smoothing is performed only to determine the high sensitivity region of interest and for optimisation the original node sensitivity $(\mathcal{G})$ is projected onto the NURBS control points.

4 Computation of Adjoint Sensitivities

4.1 CFD solver

In this work, a RANS-based compressible flow and discrete adjoint solver STAMPS [20] is used to compute flow and adjoint fields. The primal solver of STAMPS uses a standard node-centered, edge-based compressible finite volume discretisation using MUSCL type reconstruction of primitive variables with second order accuracy and stable implicit JT-KIRK scheme. The viscous source terms are obtained using an edge-corrected Green-Gauss formula. Turbulence modeling is performed with Spalart-Allmaras RANS model with AUSM scheme for the convective fluxes. In general, steady state discrete system of governing equations can be written as,

$$R(U,\alpha) = 0, \tag{15}$$

where R is the residual, U is the state variables, α is a set of design variables.

4.2 Adjoint solver

In a gradient-based design loop, one must compute the sensitivity of the objective function (J) with respect to design variables of interest α. The scalar objective function of interest considered in an aerodynamic shape optimisation (e.g. lift, drag, etc.) not only depends on the design variables α, but also on the physical state of the system U, input volume mesh X_v and surface mesh X_s which intern depends on design variables α. This can be written as a series of nested function,

$$J = J(U(X_v(X_s(\alpha)))). \tag{16}$$

With a CFD solver, the discrete adjoint equation can be written as,

$$\frac{\partial R}{\partial U}^T \psi = \frac{\partial J}{\partial U}^T \tag{17}$$

where ψ is the adjoint variable. Once the flow and adjoint fields are computed, the adjoint sensitivity of objective function with respect to design variables can be obtained using [20, 30],

$$\frac{dJ}{d\alpha}^T = \frac{\partial J}{\partial \alpha}^T + \frac{\partial X_s}{\partial \alpha}^T f_S^T \psi^T. \tag{18}$$

where $f_S = \frac{\partial R}{\partial X_s}$ and $\frac{\partial X_s}{\partial \alpha}$ is the shape sensitivity term.

4.3 NSPCC

With a CAD model in the design loop and using NSPCC approach, the adjoint CAD sensitivity term can be written as,

$$\frac{\partial X_s}{\partial \alpha}^T = \frac{\partial \mathbf{P}}{\partial \alpha}^T \frac{\partial X_s}{\partial \mathbf{P}}^T. \tag{19}$$

With geometric constraints, numerical nullspace as given in Eqn. 9 needs to be included in the chain rule. Therefore Eqn. 18 can be written as,

$$\overline{\alpha} = \frac{dJ}{d\alpha}^T = \frac{\partial J}{\partial \alpha}^T + \left(\mathrm{Ker}(\mathbf{C})\right)^T \left(\frac{\partial X_s}{\partial \mathbf{P}}\right)^T \overline{X_s}. \tag{20}$$

where $\overline{X}_s$ is the CFD sensitivity which represents the adjoint sensitivity of the objective function with respect to surface mesh coordinates $\frac{\partial J}{\partial X_s}^T$. In this work, the entire implementation including both CFD and CAD modules was written in Fortran90 to support the application of source-transformation AD using Tapenade. To compute the shape sensitivities, in-house light-weight CAD kernel NSPCC is differentiated using adjoint mode. Differentiated source code modules are assembled with non or hand-differentiated code to optimise memory and runtime. Equation 20 shows how seeding the reverse differentiation of the NURBS kernel with $\frac{\partial J}{\partial X_s}$ computes $\frac{\partial J}{\partial \mathbf{P}}$ at a cost that is independent of the size of the control net density.

5 Aerodynamic Shape Optimisation of Turbine Blade Cooling Channel

5.1 VKI U-Bend Geometry

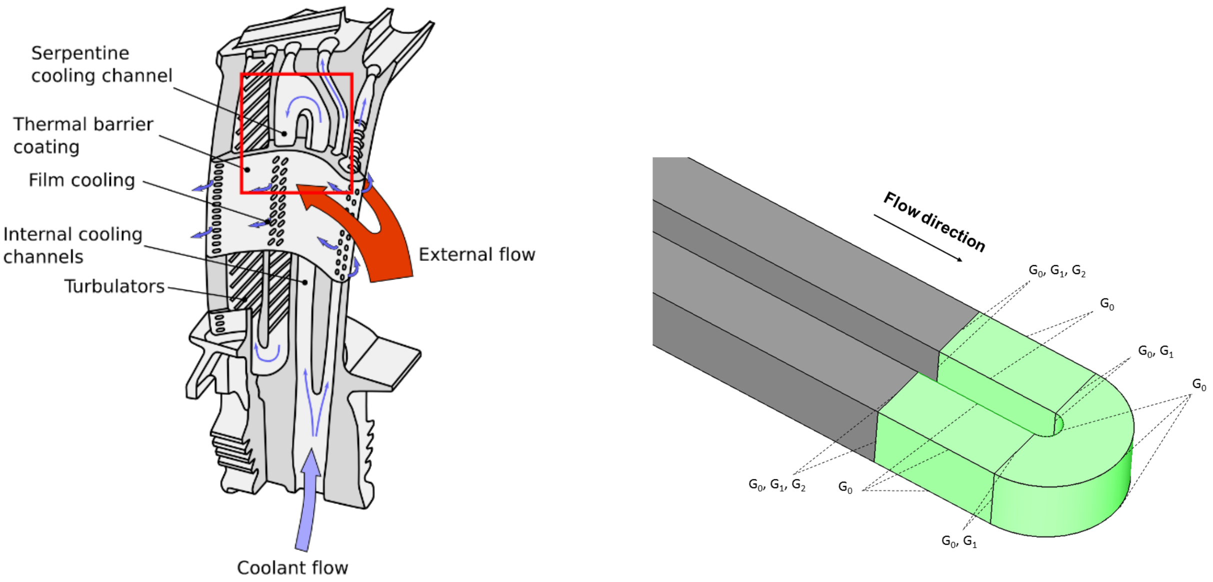
The proposed adaptive NSPCC method is used to minimise the mass-averaged total pressure loss of the VKI U-Bend geometry. The objective function is defined as J,

$$\underset{\alpha}{\text{minimize}} \;\; J = \frac{\int_{inlet} P_{tot}\vec{u}\cdot\vec{n}dS - \int_{outlet} P_{tot}\vec{u}\cdot\vec{n}dS}{\int_{inlet} \vec{u}\cdot\vec{n}dS} \tag{21}$$

where α represents the design variables, P_{tot} is the total pressure, u is the velocity vector, n is the surface normal direction and S is the cross section area. Figure 3b shows the U-bend geometry with design surfaces are highlighted in green. It consists of a two squared cross section ducts with a hydraulic diameter of $D = 0.075m$ and both ducts are connected by the half-circular U-bend. This is a typical 180° bend duct used to circulate cooling air inside the turbine blade (See Fig. 3a). This U-Bend passage turns the cooling fluid 180° and of crucial importance since they represent the region of high-pressure loss. Boundary conditions considered in this work for validation and subsequent optimisation are similar to the standard benchmark conditions specified in the About Flow project [29]. The Reynolds number considered is 43830 and the Mach number of 0.1 allows using an incompressible assumption.

5.2 Shape Parameterisation

The U-Bend region has 12 patches, of which 8 are rectangular and 4 are half-circular surfaces. To test the performance of the proposed adaptive-NSPCC method, each patch is parameterised with a coarse distribution of control points. In this work, each patch is parameterised using cubic NURBS curve along the streamwise direction with 6 control points and cubic rational Bezier curve is used along the radial direction with 4 control points (perpendicular to streamwise direction). In total this parameterisation contains 288 control points with each patch defined using 6×4 control net. Figure 4a shows the baseline U-Bend region with its control points distribution. This parametrisation exhibits global shape control along the radial direction, this can be clearly seen in Fig. 4b which shows the effect of perturbing a control point on the design surface.



(a)

(b)

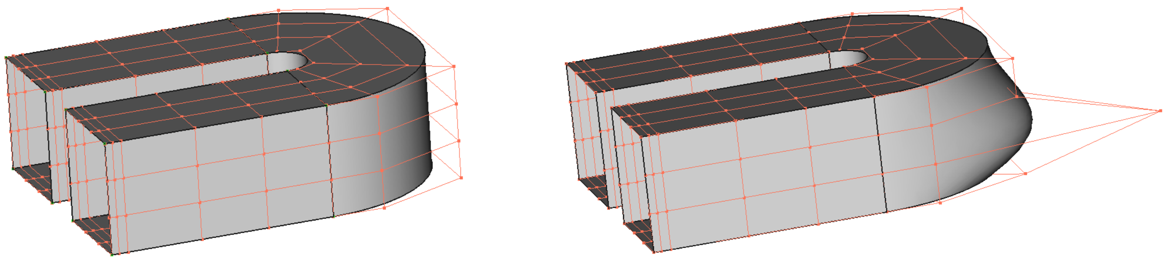
Figure 3: (a) Internal U-bend passage for turbine blade cooling channel [25], (b) U-bend geometry with design surfaces are highlighted in green

5.3 Grid Convergence Study

Based on the suggestions given by McHale et al. [19] four grid levels have been generated to perform a detailed grid convergence study: a coarse (C), a medium (M), a fine (F) and an extra fine (XF) mesh with each level containing $50k$, $125k$, $260k$ and $500k$ nodes respectively. Computational grids are created using Ansys Mesher with the coarse grid (C) having Y^+ value of 3 and remaining all grids having Y^+ value of 1. The structure of all the grid levels at the inlet and outlet of the U-bend geometry is shown in Fig. 5.

In this work, the objective function value and velocity profiles at three different locations are compared with all the grid levels to make sure the solution is independent of the mesh resolution. These velocity profiles including both streamwise as well as radial directions are taken at the 90° turn region (location A) and others (location B and C) at the exit of the channel are shown in Fig. 6a. Figure 6b shows the variation of the normalized objective function value with all the grid levels.

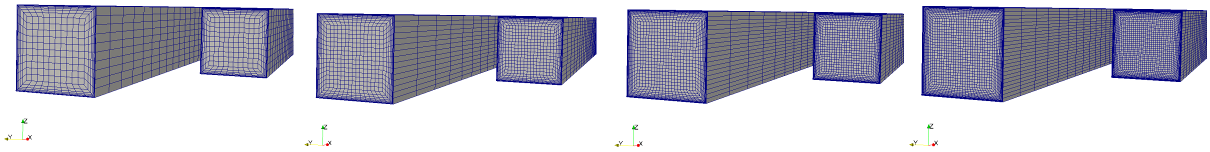
Figure 7 shows a comparison between the normalized streamwise velocity profiles at the location A, B and C (along the vertical lines). The positive and negative z/D_h indicates the top and bottom surfaces respectively from the center of the U-bend. Figure 8 shows a comparison between the normalized radial velocity profiles at the location A, B and C (along the horizontal lines). From the comparison, it is found that both streamwise and radial velocity profiles from the fine (F) and extra-fine meshes are closely matched and hence solution became independent of the mesh resolution. Figure 6b shows that the fine mesh results in a difference of less than 0.1% for the objective function value with the extra fine (XF) mesh.This difference is found acceptable for the present study [19] and hence the fine mesh (F) is used



(a) Baseline U-Bend

(b) Deformed U-Bend

Figure 4: (a) Coarse control net distribution of baseline U-Bend region. (b) Effect of perturbing a control point.



(a) Coarse (C)

(b) Medium (M)

(c) Fine (F)

(d) Extra Fine (XF)

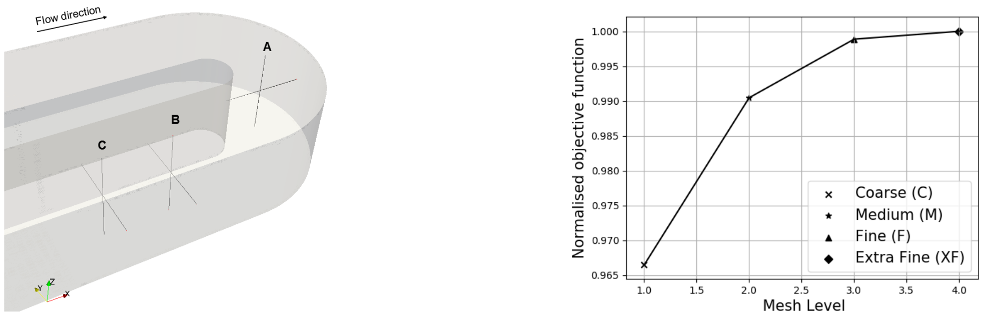
Figure 5: The inlet and outlet of the structured hexahedral meshes of all levels

for the optimisation purpose.

5.4 CFD solver validation

The numerical results obtained using STAMPS are compared with the computational and experimental ones performed by Coletti et al. [6, 2] for the same Reynolds number $Re = 43830$. In their experimental work, the inlet leg with $23.3D_h$ long is used to guarantee a fully developed flow at the location of the circular bend. Based on the suggestion given in test case description [29] the inlet leg of $10D_h$ with respect to the center of the U-Bend region is used in the present numerical study to reduce the computational cost. For the validation of our present model, the simulation is performed using fine mesh (F) containing total $260k$ nodes, as the grid-independence study showed that this grid offered sufficient resolution, see Fig. 6b. The comparison of the simulated velocity field and experimental results by Coletti et al.[6], the Large Eddy Simulation results by Alessi et al. [2] and the STAMPS results is shown in Fig. 9 for the symmetric mid plane (Z/D_h). The STAMPS solution captures the large flow separation region right after the turn. However height and length of the separation region are underestimated. Similar behavior is also noted in the RANS-based results of Alessi et al. [2].

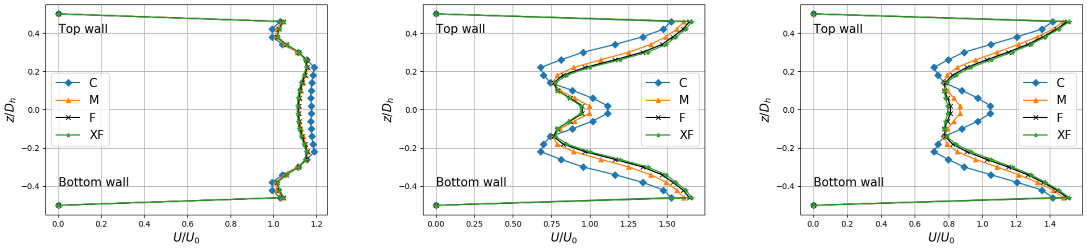
Figure 10 shows a good match between STAMPS and LES for the counter-rotating Dean vortices at the 90° turn region, only small discrepancies near the inner wall separation region can be observed. The analysis of the STAMPS solution computed on the fine (F) mesh is in good agreement with the published experimental and numerical results. Hence this configuration has been used for the optimisation studies.



(a) Locations used for velocity profiles

(b) Grid convergence of objective function

Figure 6: Grid convergence study.



(a) at *A*

(b) at *B*

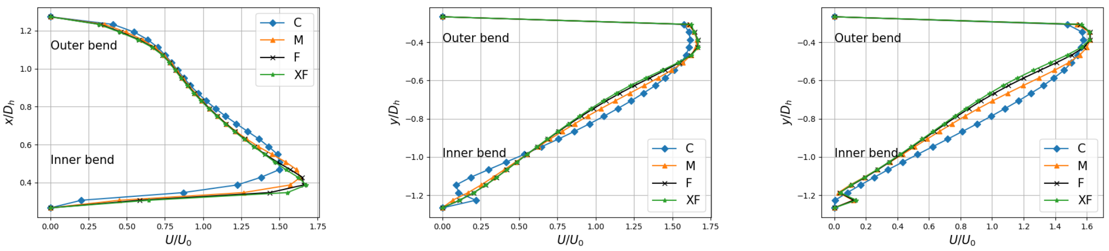
(c) at *C*

Figure 7: Streamwise velocity profiles along z axis at A, B and C.

5.5 Gradient Verification of Differentiated NSPCC

Since major focus of this work is on the CAD-based parametrisation, the verification of the gradient computation by Automatic Differentiation is presented only for the NSPCC CAD kernel, solver validation can be found in [20]. Sensitivity verification is performed by perturbing the BRep of the baseline U-bend geometry shown in Fig 4a. All the required derivatives of geometric operators, including surface sensitivities with respect to parameters s and t for point inversion (Eqn. 1), entries of the constraint matrix **C** (Eqn. 7) and shape sensitivities (Eqn. 20) are computed using derivative code produced by the source transformation AD tool Tapenade [9].

To verify the AD derivatives we compare to the values obtained with the complex step method [27]. The method is very easy to use in Fortran which allows very simple conversion from all double precision variables to double precision complex variables. However additional care has been taken as discussed in [7, 17] when handling intrinsic functions such as `abs` and conditional branches `IF..THEN..ELSE`. Figure 11 shows the convergence of truncation error for seven surface points using complex step derivative method.



(a) at *A*

(b) at *B*

(c) at *C*

Figure 8: Radial velocity profiles along *y* axis at *A*, *B* and *C*.

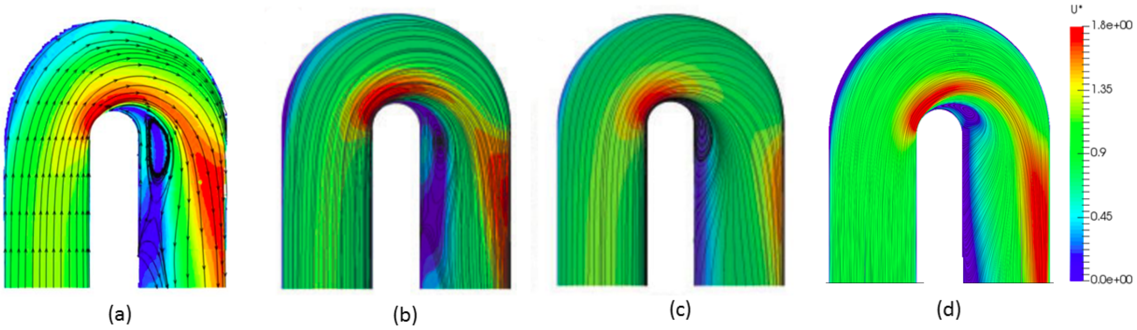


Figure 9: Comparison of normalized velocity field (U^*) along streamwise direction between experimental and simulation taken at mid plane. (a) Experimental [6] (b) LES simulation [2] (c) RANS simulation [2] (d) RANS-(STAMPS)

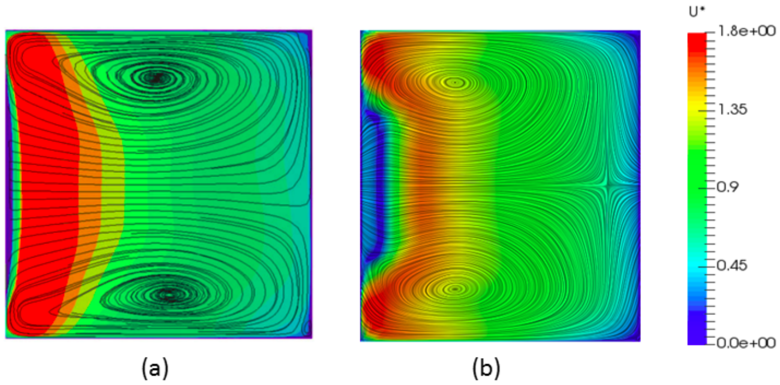


Figure 10: Comparison of normalized velocity field at 90° turn region. (a) LES [2] (b) (b) STAMPS

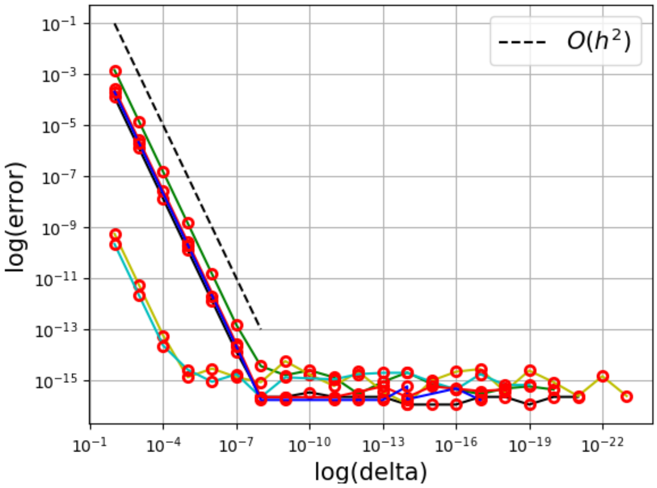


Figure 11: Relative error for seven surface points using complex step derivatives. Error: $\varepsilon = \frac{|FD-AD|}{|AD|}$.

6 Optimisation Framework using NSPCC

We developed a standalone Fortran-based CAD kernel dedicated to shape optimisation of arbitrary geometries using adaptive NSPCC method. NSPCC provides a direct interface to a CAD model's boundary representation (BRep) and derives parametrisation directly from the STEP file. Therefore CAD model's parametric design information as given in the form of 'Feature Tree' is not required. This framework is generic and can be used to setup any industrial test case for shape optimisation. In this work, a Python API has been developed to integrate CFD and CAD modules with the Python-Scipy library to drive the design process. The developed CAD-system consists of six major components. They are:

1. CAD parser: This is a python-based CAD reader and writer which is used to read and write BRep information from the STEP file and builds the initial geometry object. This object includes both topological and geometrical entities of a CAD model.

2. Mesh mapper: This module reads the computational mesh available in GMSH format and creates surface mesh objects which includes surface grid connectivity information, volume ID and surface ID. Mesh mapper also computes parametric coordinates (s,t) for each surface mesh point (X_s) using point inversion algorithm and assign a unique patch ID that it belongs to.

3. Geometry Module: This module evaluate a number of geometric quantities of interest, such as G_0, G_1, leading and trailing edge radius, curvature and thickness to handle user-defined geometric and/or manufacturing constraints. In this work, G_0 (Eqn. 4) and G_1 (Eqn. 5) constraints are used. Constraint equations are differentiated in forward mode using AD tool Tapenade. Constraint matrix $\mathbf{C}$ as shown in Eqn. 7 is then assembled and the null space $\mathrm{Ker}(\mathbf{C})$ is computed using a Singular Value Decomposition (Eqn. 8).

4. Sensitivity module: This module is used to compute CAD sensitivities in both tangent and adjoint mode. These CAD sensitivities are combined with CFD sensitivities to compute total gradients as given in Eqn. 20 and then passed it to an optimiser to obtain perturbations to design variables

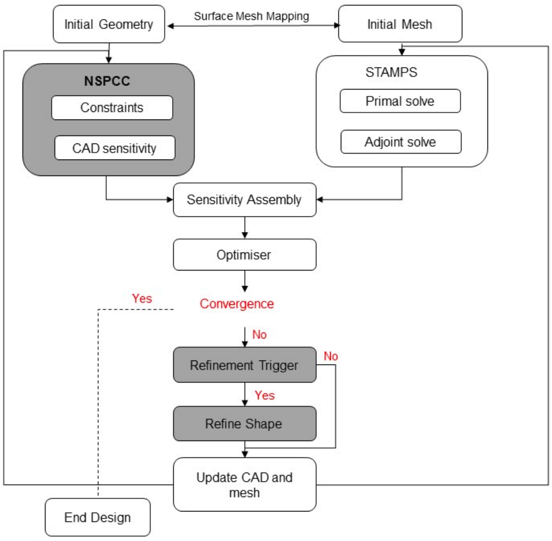


Figure 12: Shape optimisation work flow with the adaptive NSPCC parametrisation method

($\delta\alpha$). CFD sensitivities can be obtained from any adjoint solver. In this work, STAMPS a discrete adjoint solver is used to obtain flow fields (Eqn. 15), objective function value (Eqn. 21) and adjoint fields (Eqn. 17) as well as the CFD sensitivities (Eqn.18).

5. Adaptation module: Once the refinement trigger terminates the design process (Eqn. 10). This module performs gradient smoothing (Eqn. 11) and compute suitable knot values to insert in both parameter direction of all the NURBS patches (Eqn. 12 and Eqn. 12) and then perform knot insertion to obtain three refined control nets. Finally, this module evaluate the adaptation metric for each control net and choose the control net which has the highest adaptation metric value (Eqn. 14).
6. Deformation module: This module computes perturbations to control points $\delta\mathbf{P}$ as given in Eqn. 9. This ensures that geometric constraints are satisfied automatically. Similarly, a user can choose different constraints such as G_2, box, radius and thickness constraints based on the requirements. An additional constraint recovery step is also required for non-linear constraints. Details can be found in [21]. This module makes the required modification to the initial NURBS patches and compute displacements to surface grid points δX_s using the parametric coordinates computed from step 2.

After perturbing the geometry and corresponding surface mesh, we can use any mesh deformation algorithm to propagate surface grid displacements δX_s smoothly into the volume mesh δX_v. In this work, Inverse Distance Weighting (IDW) method which is also available in STAMPS deformation module is used to deform surrounding volume mesh. Figure 12 shows the work flow of a shape optimisation process with the adaptive NSPCC parametrisation method.

7 Results and Discussion

Both static and adaptive parameterisation methods are coupled in the adjoint-based shape optimisation process to reduce the total pressure loss of a turbine blade internal cooling channel. Optimisation results

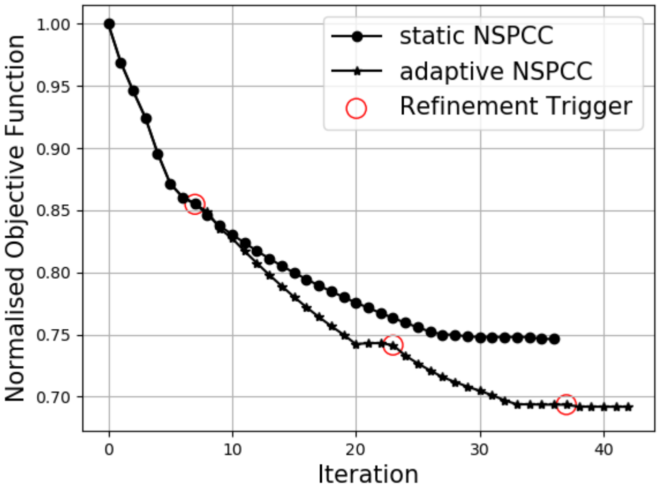


Figure 13: Convergence of objective function: static vs adaptive NSPCC

obtained using both parameterisation method are compared. Particular attention is given to rate of design convergence, quality of the optima and their sensitivity to the choice of control points distribution on the design surface.

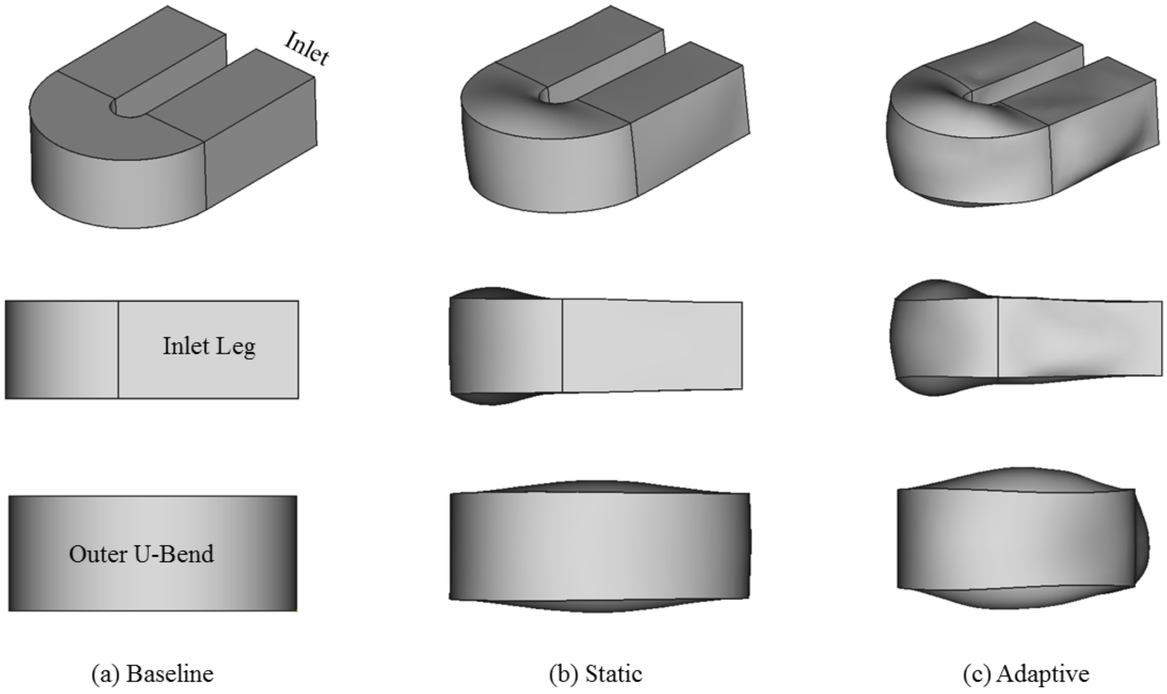
7.1 Shape optimisation

The U-Bend region is parameterised with a coarse distribution of control points, each patch is parameterised with 6×4 control net, in total it has 288 control points. First three layers of control points between the fixed and deformable patches are placed close to the inlet and exit throat and are fixed to maintain G_2 continuity. Adaptive parameterisation begins with this coarser control net and adapt to a finer parameterisation during the design process where more shape control is needed. On the other hand, in the static NSPCC method, control net remain fixed throughout the optimisation. In both case, BFGS optimiser available in the Python-Scipy library is used to drive the design process.

Figure 13 compares the convergence history of the objective function between static and adaptive NSPCC method. The size of the design space is given by $N - r'$ in which N is the total number of control points allowed to deform in the design process and r' is the numerical rank of the constraint matrix (Eqn. 9). Value of r' depends on the chosen threshold frequency value σ_C, for this test case the chosen value is 10^{-10} which results 442 number of design variables at the end of the optimisation. On the other hand, in the adaptive case, design space has been refined three times. Table 1 shows the comparison of number of design variables and the performance improvement achieved between static and adaptive NSPCC method. Shape optimisation using the refined design space is able to reduce the objective function value further and has the potential to accelerate the design convergence and capture important shape modes outside the fixed envelope offered by the static design space. Figure 14 shows the comparison of optimum geometries obtained using static and adaptive NSPCC method.

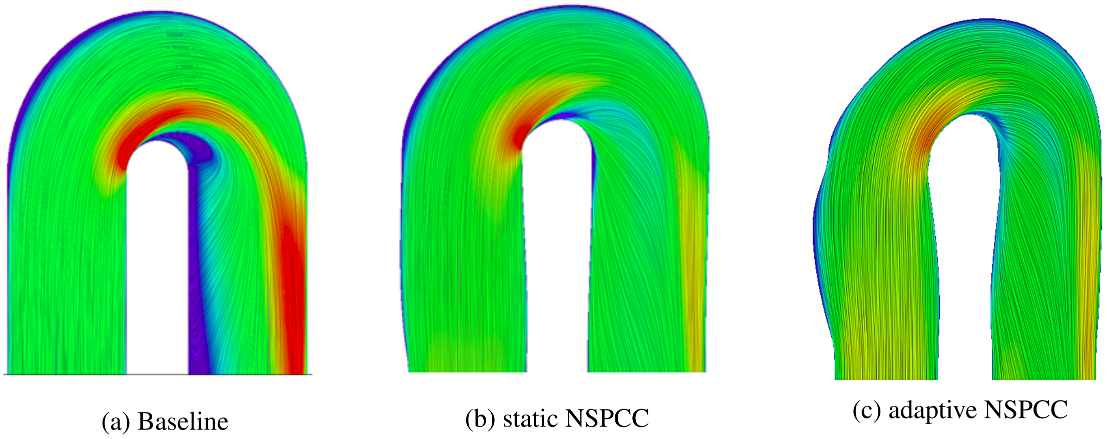
Level	Control net dimension	Number of control points (N)	Size of the design space	Percentage drop in total pressure loss
Static	4 * 6	288	442	-25.34%
Adaptive NSPCC				
Iter 1-7	4 * 6	288	574	-14.51%
Iter 8-23	5 * 7	420	736	-25.68%
Iter 24-37	6 * 8	576	1026	-30.63%
Iter 42	7 * 9	756	1402	-30.8%

Table 1: Optimisation results: static vs adaptive NSPCC



(a) Baseline (b) Static (c) Adaptive

Figure 14: Comparison of optimised geometries: Static vs Adaptive NSPCC



(a) Baseline

(b) static NSPCC

(c) adaptive NSPCC

Figure 15: Comparison of normalized velocity magnitude between experimental and CFD. Cross-sectional slice taken at the mid-symmetry plane

7.2 Flowfield of the optimised geometry

Pressure losses in a serpentine cooling channel are caused by the effect of both wall friction and momentum exchanges due to the change in the direction of the flow. Figures 15a and 16a shows the normalised velocity magnitude of the baseline geometry. When compared with the baseline configuration, optimised geometries obtained using both the parameterisation methods suppress the flow separation near the inner wall of the exit channel hence reduces wall shear stress significantly. Figure 15 shows the comparison of velocity magnitude taken at mid symmetry plane between baseline and optimised geometries. The reason for the design improvement is threefold. Firstly, all levels of parameterisation altered the radius of the inner U-Bend, this can be clearly seen in Fig 15. For incompressible and irrotational flow the velocity gradient normal to the streamline is proportional to the curvature of the streamline. Hence the optimised geometries with enhanced radius of curvature reduces the required radial pressure gradient and hence the streamwise adverse pressure gradient, resulting in a smaller separation zone.

Secondly, the duct section is considerably enlarged for both the optimised geometries, this can be clearly seen in Fig. 16 which compares the CS shape taken at the 90° turn region, both the optimum geometries exhibit larger CS area than the baseline geometry. Hence reducing the velocity in the bend which, similar to the radius increase, reduces the required centripetal forces, hence the required radial pressure gradient, hence the separation zone. When compared with the static NSPCC, the optimised shape obtained using adaptive NSPCC shows larger cross-sectional area which further reduces the pressure gradient normal to the stream line. Lower normal pressure gradient at the turn generates weaker secondary vortex which significantly reduces the associated diffusion loss. In addition, weaker secondary vortex reduces velocities near the outer wall of the exit channel which reduce the wall shear stress further.

Finally, a third contribution is obtained by the formation of strake like shape along the vertical direction of the inner U-Bend such as widely used on airplanes and pipelines. At the inner part of the U-Bend, a low velocity region was observed which is shown in the Fig. 16. The size of this region is more for baseline geometry than others (Fig. 16a). From Fig. 16c it is interesting to note that inner U-Bend region of the adaptive shape captures strong convex or hump-like shape mode along vertical direction however the optimised shape obtained using static NSPCC doesn't capture this superior design (Fig. 16b). This strake like shape mode formed at the center of the inner U-Bend that splits the counter-rotating vortices hence re-energies and reduces the low velocity region in the optimum geometries.

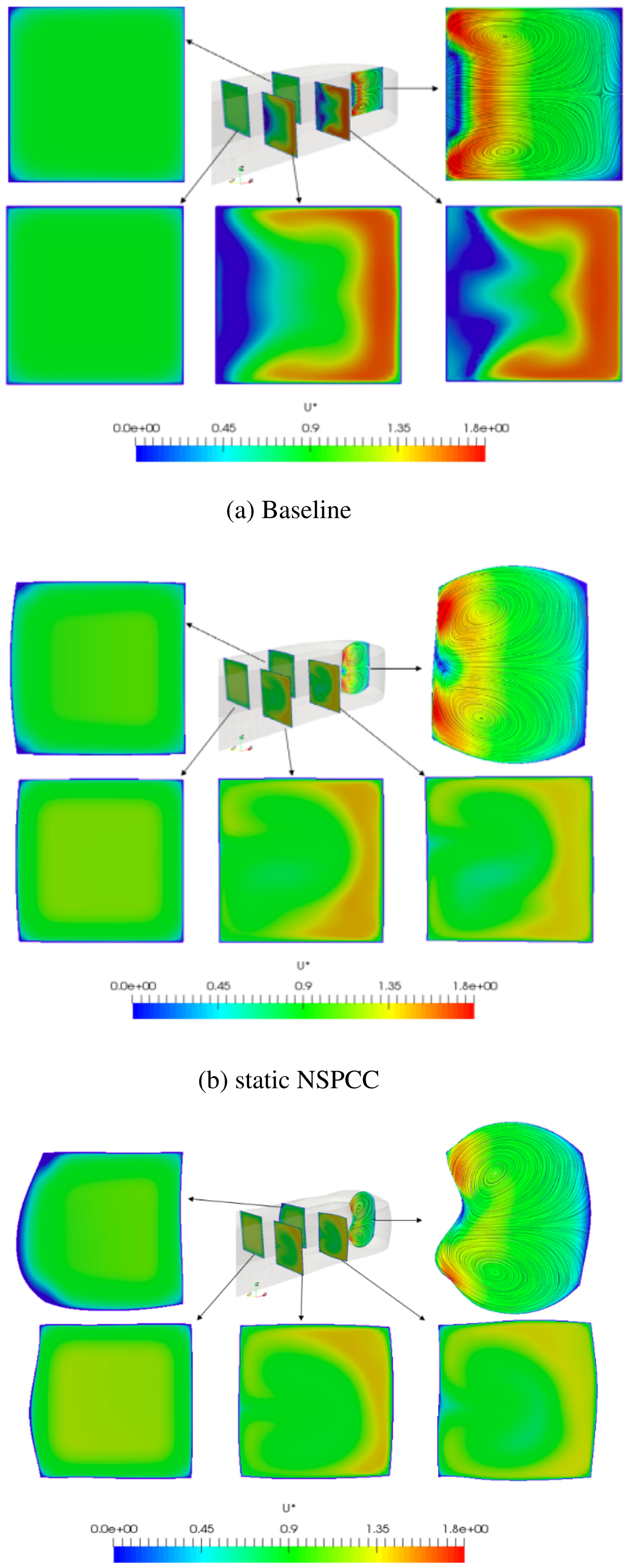
8 Conclusion

In this present work, CAD-based NSPCC approach is extended to include adaptive design space for shape optimisation by refining the control points locally on the design surface using knot insertion algorithm. The proposed adaptive parameterisation method has been used to reduce pressure loss of the VKI U-Bend. Optimisation begins with a coarser net and then automatically refine the control points distribution in the regions where significant node-based sensitivities are detected. Design space enrichment is performed by inserting knots in both parameter directions of the NURBS patches. The refinement procedure is fully automatic and minimal user input is required to setup the design space for the optimisation. This approach is both efficient and complete by eliminating the arbitrary trade-off between the dimension and distribution of control points on the design surface. The optimised geometry obtained using adaptive parameterisation outperforms static parameterisation.

In this work, CFD sensitivities are computed by using in-house flow and discrete adjoint solver named STAMPS. Previously, NSPCC CAD kernel has been differentiated in forward mode [30, 31, 16] where computational costs for computing CAD sensitivity is proportional to the number of design variables. In this present work, the NSPCC CAD kernel is differentiated in reverse mode, therefore the entire design chain in now differentiated using source transformation AD tool in reverse mode which computes adjoint sensitivities in an efficient manner.

NSPCC approach has many advantages. They are:

- User set-up: Shape parametrisation is derived from the BRep of a CAD model, hence construction history of a CAD model is not required, only subset of functionalities can be added. This reduces the computational effort required to parameterising existing CAD models.
- Orthogonal shape modes: NSPCC offers rich design space and uses the SVD basis vectors of the nullspace as a basis for the design space, hence resultant shape modes are orthogonal to each other.
- Adaptive design space: Using knot insertion algorithm, control net refinement can be performed where more shape control is needed. Hence the resultant shape is unaltered during the refinement process.
- Efficient constraint handling: Both geometric and manufacturing constraints are handled simultaneously via test point approach.
- Exact shape sensitivities: NSPCC CAD kernel has been differentiated using source transformation AD tool to compute required gradients. Shape sensitivities are available in both tangent and adjoint mode.
- CAD model: Preserves CAD geometry in the design loop hence more suitable to handle multidisciplinary optimisation with minimal effort. Currently work in progress to couple NSPCC CAD kernel with conjugate heat transfer and structural analysis.



(a) Baseline

(b) static NSPCC

(c) adaptive NSPCC

Figure 16: Comparison of secondary flow structure between baseline and optimised geometries. Cross-sectional slices are taken at different locations along the length of the channel.

REFERENCES

[1] Dheeraj Agarwal, Trevor T Robinson, Cecil G Armstrong, and Christos Kapellos. Enhancing cad-based shape optimisation by automatically updating the cad model's parameterisation. *Structural and Multidisciplinary Optimization*, pages 1–16, 2018.

[2] Giacomo Alessi, Tom Verstraete, Lilla Koloszar, and Jeronimus Petrus Antonius Johannes van Beeck. Comparison of large eddy simulation and reynolds-averaged navier–stokes evaluations with experimental tests on u-bend duct geometry. *Proceedings of the Institution of Mechanical Engineers, Part A: Journal of Power and Energy*, 234(3):315–322, 2020.

[3] George R Anderson and Michael J Aftosmis. Adaptive shape parameterization for aerodynamic design. *Nat. Aeronaut. Space Admin., Ames Res. Center, Moffett Field, CA, USA, NAS Tech. Rep. NAS-2015-02*, 2015.

[4] Mladen Banović, Orest Mykhaskiv, Salvatore Auriemma, Andrea Walther, Herve Legrand, and Jens-Dominik Müller. Algorithmic differentiation of the open cascade technology cad kernel and its coupling with an adjoint cfd solver. *Optimization Methods and Software*, 33(4-6):813–828, 2018.

[5] Faidon Christakopoulos. *Sensitivity computation and shape optimisation in aerodynamics using the adjoint methodology and Automatic Differentiation.* PhD thesis, Queen Mary University of London, 2012.

[6] Filippo Coletti, Tom Verstraete, Jérémy Bulle, Timothée Van der Wielen, Nicolas Van den Berge, and Tony Arts. Optimization of a u-bend for minimal pressure loss in internal cooling channels part ii: Experimental validation. *Journal of Turbomachinery*, 135(5):051016, 2013.

[7] P. Cusdin. *Automatic sensitivity code for Computational Fluid Dynamics*. PhD thesis, School of Aeronautical Engineering, Queen's University Belfast, 2005.

[8] Xiaocong Han and David W Zingg. An adaptive geometry parameterization for aerodynamic shape optimization. *Optimization and Engineering*, 15(1):69–91, 2014.

[9] Laurent Hascoet and Valérie Pascual. The tapenade automatic differentiation tool: Principles, model, and specification. *ACM Transactions on Mathematical Software (TOMS)*, 39(3):20, 2013.

[10] Xiaolong He, Jichao Li, Charles A Mader, Anil Yildirim, and Joaquim RRA Martins. Robust aerodynamic shape optimization—from a circle to an airfoil. *Aerospace Science and Technology*, 87:48–61, 2019.

[11] Raymond M Hicks and Preston A Henne. Wing design by numerical optimization. *Journal of Aircraft*, 15(7):407–412, 1978.

[12] Majid Hojjat. *Node-based parameterization for shape optimal design*. PhD thesis, Technische Universität München, 2015.

[13] Jiri Hradil. *Adaptive Parameterisation for Aerodynamic Shape Optimisation in Aeronautical Applications*. PhD thesis, Brno University of Technology, 2015.

[14] Antony Jameson. Aerodynamic shape optimization using the adjoint method. *Lectures at the Von Karman Institute, Brussels*, 2003.

[15] Armen Jaworski and Jens-Dominik Müller. Toward modular multigrid design optimisation. In *Advances in Automatic Differentiation*, pages 281–291. Springer, 2008.

[16] Rejish Jesudasan, Xingchen Zhang, and Jens-Dominik Mueller. Adjoint optimisation of internal turbine cooling channel using nurbs-based automatic and adaptive parametrisation method. In *ASME 2017 Gas Turbine India Conference*, pages V001T02A009–V001T02A009. American Society of Mechanical Engineers, 2017.

[17] Joaquim RRA Martins, Peter Sturdza, and Juan J Alonso. The complex-step derivative approximation. *ACM Transactions on Mathematical Software (TOMS)*, 29(3):245–262, 2003.

[18] Dominic A Masters, Nigel J Taylor, T Rendall, and Christian B Allen. A locally adaptive subdivision parameterisation scheme for aerodynamic shape optimisation. In *34th AIAA Applied Aerodynamics Conference*, page 3866, 2016.

[19] MP McHale, JR Friedman, and J Karian. Standard for verification and validation in computational fluid dynamics and heat transfer. *The American Society of Mechanical Engineers, ASME V&V*, pages 20–2009, 2009.

[20] Jens-Dominik Mueller, Jan Hueckelheim, and Orest Mykhaskiv. Stamps: a finite-volume solver framework for adjoint codes derived with source-transformation ad. In *2018 Multidisciplinary Analysis and Optimization Conference*, page 2928, 2018.

[21] Jens-Dominik Müller, Xingchen Zhang, Siamak Akbarzadeh, and Yang Wang. Geometric continuity constraints of automatically derived parametrisations in CAD-based shape optimisation. *International Journal of Computational Fluid Dynamics*, pages 1–17, 2019.

[22] Orest Mykhaskiv, Mladen Banović, Salvatore Auriemma, Pavanakumar Mohanamuraly, Andrea Walther, Herve Legrand, and Jens-Dominik Müller. NURBS-based and parametric-based shape optimization with differentiated CAD kernel. *Computer-Aided Design and Applications*, 15(6):916–926, 2018.

[23] Les Piegl and Wayne Tiller. *The NURBS book*. Springer Science & Business Media, 2012.

[24] Trevor T Robinson, Cecil G Armstrong, Hung Soon Chua, Carsten Othmer, and Thorsten Grahs. Optimizing parameterized cad geometries using sensitivities based on adjoint functions. *Computer-Aided Design and Applications*, 9(3):253–268, 2012.

[25] plc Rolls Royce. *The jet engine*. John Wiley & Sons, 2015.

[26] Jamshid A Samareh. Survey of shape parameterization techniques for high-fidelity multidisciplinary shape optimization. *AIAA journal*, 39(5):877–884, 2001.

[27] W. Squire and G. Trapp. Using complex variables to estimate derivatives of real functions. *SIAM-Review*, 10(1):110–112, 1998.

[28] Tom Verstraete. Cado: a computer aided design and optimization tool for turbomachinery applications. In *2nd Int. Conf. on Engineering Optimization, Lisbon, Portugal, September*, pages 6–9, 2010.

[29] Tom Verstraete. The VKI U-Bend optimization test case. Technical report, The von Karman Institute for Fluid Dynamics, 2016.

[30] Shenren Xu, Wolfram Jahn, and Jens-Dominik Müller. Cad-based shape optimisation with cfd using a discrete adjoint. *International Journal for Numerical Methods in Fluids*, 74(3):153–168, 2014.

[31] Shenren Xu, David Radford, Marcus Meyer, and Jens-Dominik Müller. Cad-based adjoint shape optimisation of a one-stage turbine with geometric constraints. In *ASME Turbo Expo 2015: Turbine Technical Conference and Exposition*, pages V02CT45A006–V02CT45A006. American Society of Mechanical Engineers, 2015.

[32] Wataru Yamazaki, Sylvain Mouton, and Gerald Carrier. Geometry parameterization and computational mesh deformation by physics-based direct manipulation approaches. *AIAA journal*, 48(8):1817–1832, 2010.