ADJOINT OPTIMISATION OF INTERNAL TURBINE COOLING CHANNEL USING NURBS-BASED AUTOMATIC AND ADAPTIVE PARAMETRISATION METHOD

Rejish Jesudasan*, Xingchen Zhang, Dr. Jens-Dominik Mueller
School of Engineering and Material Science
Queen Mary University of London
Mile End Rd, London E1 4NS
Email: r.jesudasan@qmul.ac.uk

ABSTRACT
A well-formulated design space parametrisation is the key to the success of design optimisation. Most parametrisation methods require manual set-up which typically results in a restricted design space and impedes the generation of superior designs which may be found outside this restricted envelope. In this work, we adopt a NURBS-based automatic and adaptive parametrisation approach where the optimisation begins in a coarser design space and adapts to finer parametrisation during the optimisation. Our approach takes CAD descriptions as input and to alter the shape perturbs the control points of the NURBS patches that form the boundary representation. Driven by adjoint sensitivity information the control net is adaptively enriched using knot insertion. The sensitivity-driven parametrisation method is applied here to reduce the pressure loss of a U-bend passage of a turbine blade serpentine cooling channel.

1 INTRODUCTION
Advancements in Computational Fluid Dynamics (CFD) in combination with powerful computers have enabled the designers to create various design tools based on numerical optimisation algorithms. As a result, CFD is now no longer used only for computing the flow for a given complex configuration. By combining CFD with optimisation algorithms it is now possible to modify the shape and increase the performance of a component in an iterative manner [1–5]. This shortens the design cycle of a component and enables to adapt to rapidly changing market demands. However, many different designs need to be evaluated and compared in the design loop, which requires numerical optimisation loops that systematically explore the design space.

Stochastic optimisation techniques such as Evolutionary or Genetic Algorithms have been successfully applied to various design and shape optimisation applications. These methods are well suited to handle multi-modal, non-convex or objective functions with noise [6–8]. However, the major disadvantage of evolutionary-based optimisation is that they are slow to converge and an optimal solution may require $O(1000) - O(10000)$ function evaluations for a fine design space with $O(100)$ design variables. Hence these methods are suitable only to handle coarser design spaces with only few design variables. A comprehensive review of recently introduced deterministic based optimisation methods can be found in [9]. These algorithms shall not be discussed further in here.

On the other hand, gradient-based methods are very efficient if the objective function is differentiable. As opposed to stochastic methods, gradient-based approaches can handle a large number of design variables and converge to optimum far more effectively. However, these algorithms need the computation of the gradient of the objective function with respect to each design variable to determine the direction of the design improvement in the design space [2, 10, 11]. Of particular interest is the adjoint method which can compute a large number of sensitivities at a cost that is essentially independent of the number of design parameters [12, 13]. Over the last decades adjoint methods have been successfully applied to optimise various turbomachinery [14–16] and aerodynamic components [17–19].

*Address all correspondence to this author
Along with the efficient computation of the design sensitivity, a well-formulated design space parametrisation is needed to obtain an optimal solution in the design space. More often an optimal solution is strongly influenced by the geometry representation and design variables considered in the design process. The definition of the design space should not be artificially constrained by the relative size of the design variables. A small number of design variables places the burden on the user to establish an appropriate design space while a larger number of design variables places the burden on the optimisation algorithm to accommodate the problems with higher degrees of freedom.

A wide range of parametrisation approaches are available and most shape parametrisation methods require manual setup and the design space of all reachable shapes is fixed before the optimisation [20]. This traditional static parametrisation approach restricts the design space and impedes the generation of superior designs outside this envelope. Furthermore, a designer needs to be aware that the final design is likely to be suboptimal and often leads to redesign using additional design variables to further enhance the performance measure of the design. However, this potential pitfall can be overcome by using a very large number of design variables.

Adjoint methods do not penalise the size of the design space, hence we can consider a very larger design space that guarantees to incorporate all the possible shape modes in the optimisation. In the node-based parametrisation method, all the mesh nodes on the design surface are considered as the degrees of freedom which offers the richest design space that CFD discretisation can express [21, 22]. No additional user input is needed and the parametrisation is directly extracted from the CFD mesh used for the simulation. However, the presence of high-frequency oscillatory shape modes are not sufficiently resolved by the CFD solver and hence remain poorly damped. Therefore, additional regularisation of design space is necessary, and implicit [23] as well as explicit [24] smoothing methods have been proposed, both of them requiring to tune a smoothing coefficient that controls the design space.

Alternatively one can reduce the design space richness by constructing auxiliary hexahedral volume spline grids around the surface grids which allow to define a smooth deformation field controlled by displacement of the spline control points, an approach termed as free form deformation [25]. While easy to set up for simple geometries such as wings or turbine blades, these methods involve substantial setup effort for complex geometries. Generally, manual construction of the design space typically results in a restricted design space that may not contain the relevant shape modes for the optimisation.

The major drawback of both above-mentioned methods is that the optimised geometry exists only as a deformed surface mesh or deformation field which is usually not interchangeable between disciplines. Therefore manual translation to CAD is required which is not straightforward and any approximation will incur the loss of optimality.

On the other hand, geometry-based parametrisation methods are fully autonomous and independent of the computational mesh used for the simulation. The conventional CAD-based approach considers engineering parameters such as camber, thickness along with skew and stagger angle distribution along the blade height to parametrise turbo-machinery blades [26]. More often these engineering parameters are drawn from the designers’ expertise and knowledge about how the geometry effects on the flow. This low-dimensional CAD-based parametrisation impedes the design space during the design process and different CAD parameters may lead to different optimal shapes.

Additionally, commercial CAD systems do not offer derivatives of surface displacements with respect to design parameters which are needed in the chain rule to compute the overall sensitivity of the objective function. The only available option is to use the finite difference method to calculate CAD-sensitivities which may lead to topological changes in the CAD geometry, or to possibly erroneous derivatives if the finite-difference is computed using an intermediate projection surface [27]. One could consider here the open source CAD system Open Cascade Technology (OCC). Access to the source code enabled the use of automatic differentiation software ADOL-C, which facilitated gradient calculations [28]. However, this would still incur the limitations of a manually defined design space.

As an alternative, we adopt an automatic and adaptive parametrisation approach where the optimisation begins in a coarser design space and adapts to finer parametrisation during optimisation. Our NURBS-based approach [14, 17, 29] takes CAD descriptions as input and perturbs the control points of the NURBS patches from boundary representation (BRep) to alter the shape. To maintain geometric continuity such as G0 (Continuity), or G1 (Tangency) between patches, constraint equations and their Jacobians are numerically evaluated at test-points shared along the patch interfaces and the design space is the kernel of this Jacobian which is evaluated using a Singular Value Decomposition. We then automatically introduce more control points using knot insertion only when necessary to further improve the design. In a progressive parametrisation approach, the user can enrich the design space through a predetermined sequence of uniformly distributed control points globally on the NURBS patches [30]. An alternative to global refinement is the adaptive refinement by analogy to adaptive meshing where only most important design variables which are all having the high potential for the design improvement are added thus reducing the overall dimension of the search space.

In this work, the adaptive refinement is driven by node-based sensitivity information, therefore we refine the control point distribution only in the region where larger smoothed node-based gradient modes remain when the optimiser has converged the objective function. As a consequence, our approach replaces user in the design loop as design variables are automatically created.
and bounded for complex turbomachinery test cases. Finally, the modified CAD description of the surface is exported as a STEP file for further processing and manufacturing. The aim of this paper is to apply NURBS-based automatic and adaptive parametrisation method to reduce the pressure loss of a U-bend passage of a turbine blade serpentine cooling channel.

This paper is organised as follows: Sec. 2 explains the discrete adjoint formulation. Mathematical background of the NURBS-based parametrisation, sensitivity based adaptation criteria and the methodology used for continuity constraints employed are described in Sec 3. The optimisation problem formulation, baseline geometry and the mesh are described in Sec. 4. Section 5 analyses the optimisation results obtained for 3D-integrated turbine cooling channel using both static and adaptive parametrisation methods and demonstrates the effectiveness of the methodology. Conclusions are presented in Section 6.

2 DISCRETE ADJOINT FORMULATION

Generally, in aerodynamic shape optimisation the function of interest that we are minimising depends not only on the design variables but also on the physical state of the system. Due to these dependencies, the cost function can be written as,

\[ J = J(\alpha_n, U_k(\alpha_n)) \]  

(1)

where \( \alpha_n \) represents the vector of design variables for \( n = 1, ..., N_d \) and \( U_k \) is the vector of state variables for \( i = 1, ..., N_t \).

For a given vector \( \alpha_n \), the solution of the governing equations of the system yields a state vector \( U_k \), thus establishing the dependence of the state vectors on the design variables. Generally, these steady state governing equations are non-linear and this system of equations are solved using an iterative method by driving the residuals \( R_k \) to zero which arises from the discretisation of the conservation equation. Hence, the governing equations are denoted as,

\[ R_k(\alpha_n, U_k(\alpha_n)) = 0 \]  

(2)

where \( R_k \) represents all the governing equations of the system, (e.g.) the steady-state Navier-Stokes equations.

In a gradient-based optimisation, sensitivity of the objective function with respect to design variables are computed by applying chain rule to Eq. (1). It can be written as,

\[ \frac{dJ}{d\alpha_n} = \frac{\partial J}{\partial \alpha_n} + \frac{\partial J}{\partial U_k} \frac{dU_k}{d\alpha_n} \]  

(3)

In Eq. (3), the term \( \frac{dU_k}{d\alpha_n} \) represents the change of the state variable \( U_k \) with respect to \( \alpha_n \) which is called perturbation field. Therefore to evaluate the total sensitivity of the cost function, one needs to evaluate the perturbation flow field for each design variables.

Linearising the non-linear discrete governing equations yields,

\[ \frac{\partial R_k}{\partial \alpha_n} + \frac{\partial R_k}{\partial U_k} \frac{dU_k}{d\alpha_n} = 0 \]  

(4)

This could be solved iteratively to compute the perturbation field \( \frac{dU_k}{d\alpha_n} \) with respect to each design variable.

\[ \frac{\partial R_k}{\partial U_k} \frac{dU_k}{d\alpha_n} = - \frac{\partial R_k}{\partial \alpha_n} \]  

(5)

which can be written in short form as

\[ Au = f \]  

(6)

where \( A \) represents Jacobian and \( f = -\frac{\partial R_k}{\partial \alpha_n} \) is the source term which is the negative partial derivative of the residual with respect to the design variables and \( u = \frac{dU_k}{d\alpha_n} \).

Eliminating \( u \) from Eq. (3), the total sensitivity equation becomes,

\[ \frac{dJ}{d\alpha_n} = \frac{\partial J}{\partial \alpha_n} - \frac{\partial J}{\partial U_k} \left( \frac{\partial R_k}{\partial U_k} \right)^{-1} \frac{\partial R_k}{\partial \alpha_n} \]  

(7)

\[ \frac{dJ}{d\alpha_n} = \frac{\partial J}{\partial \alpha_n} + g^T A^{-1} f \]  

(8)

where \( g^T = \frac{\partial R_k}{\partial U_k} \).

We first need to solve the Eq.(6) for \( \frac{dU_k}{d\alpha_n} \) and then substitute the result in expression (7) for the computation of total sensitivity for each design variable \( \alpha_n \). In CFD-based shape optimisation, the cost of solving the linear system of equations is comparable to the cost of solving the non-linear system, so there are no computational savings from using the direct tangent linear sensitivity approach.

However, this is the starting point of discrete version of the
adjoint-based approach. By transposing and rearranging Eq. (7), we can observe that for computing the adjoint sensitivity \( \left( \frac{dJ}{d\alpha_n} \right)^T \), the auxiliary vector \( \psi \) can be obtained by solving the linear system of adjoint equations (see Eq. 11) generated by the second and third terms in the right hand side of the total sensitivity Eq. (7).

\[
\left( \frac{dJ}{d\alpha_n} \right)^T = \left( \frac{\partial J}{\partial \alpha_n} + s^T A^{-1} f \right)^T \tag{9}
\]

\[
\left( \frac{dJ}{d\alpha_n} \right)^T = \left( \frac{\partial J}{\partial \alpha_n} \right)^T + f^T A^{-T} g \tag{10}
\]

The term \( A^{-T} g \) is solved for the adjoint variable \( \psi \)

\[ A^T \psi = g \tag{11} \]

which modifies the adjoint sensitivity equation to,

\[
\left( \frac{dJ}{d\alpha_n} \right)^T = \left( \frac{\partial J}{\partial \alpha_n} \right)^T + f^T \psi \tag{12}
\]

By transposing back, the adjoint sensitivity equation becomes

\[
\frac{dJ}{d\alpha_n} = \frac{\partial J}{\partial \alpha_n} + \psi^T f \tag{13}
\]

By comparing the two equations (8) and (13), we can identify that the adjoint solution \( \psi \) depends only on the Jacobian \( A^T \), i.e. depends only on the state vectors and the choice of objective function and it is independent of the number of design variables. The only additional cost for each design variable is computation of \( f = \frac{\partial R}{\partial \alpha_n} \) and \( \frac{\partial J}{\partial \alpha_n} \) which is inexpensive and the cost of solving the adjoint equations is similar to that of full solve of the primal equations. Therefore after solving the governing equations, an adjoint system of equations needs to be solved once for each objective and it is valid for all the design variables. However, if there is more than a single objective, all additional objectives require an additional adjoint solve.

### 3 CAD-BASED PARAMETRISATION

We use CAD-based parametrisation using boundary representation (BRep) given in the STEP standard as a collection of Non-Uniform Rational B-Spline (NURBS) surface patches. In general, NURBS patches are represented by the finite set of points in the space called control points and each of them is allowed to move in all directions, hence each control point representing 3 degrees of freedom (DoF). Mathematically, NURBS can be represented as [31],

\[
X_i(s,t) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_{i,j}(s,t) P_{i,j} \tag{14}
\]

where \( P_{i,j} \) is the coordinate vector of the control points, \( i \) and \( j \) represents row and column index respectively, \( s \) and \( t \) are the parametric variables of the surface mesh point and \( B_{i,j} \) is the rational basis functions which are the influence weights of each control point. The rational basis functions are defined as [31],

\[
B_{i,j}(s,t) = \frac{N_{i,p}(s)N_{j,q}(t)w_{i,j}}{\sum_{k=0}^{p} \sum_{l=0}^{q} N_{i,p}(s)N_{j,q}(t)w_{k,l}} \tag{15}
\]

where \( N_{i,p}(s) \) and \( N_{j,q}(t) \) are the \( p^{th} \) and \( q^{th} \) degree basis functions defined on the knot vector \( S \) and \( T \).

\[
S = \{0, \ldots, 0, s_{p+1}, \ldots, s_i, \ldots, s_{k_p-p-1}, 1, \ldots, 1 \} \tag{16}
\]

\[
T = \{0, \ldots, 0, t_{q+1}, \ldots, t_i, \ldots, t_{k_q-q-1}, 1, \ldots, 1 \} \tag{17}
\]

\( N_{i,p}(s) \) and \( N_{j,q}(t) \) are given by the following expressions,

\[
N_{i,0}(s) = \begin{cases} 
1 & \text{if } s_i \leq s < s_{i+1} \\
0 & \text{otherwise} 
\end{cases} \tag{18}
\]

\[
N_{i,k}(s) = \frac{(s-s_i)}{s_{i+k}-s_i}N_{i,k-1}(s) + \frac{s_{i+k+1}-s}{s_{i+k+1}-s_{i+1}}N_{i+1,k-1}(s) \tag{19}
\]

Relationship between the total number of knots, total number of control points and the degree of each parameter direction is given by,
where Eqn. (20) and (21) corresponds to parameter direction $s$ and $t$ respectively.

3.1 Adaptive refinement

NURBS is a generalization of B-spline and has several fundamental geometric algorithms such as degree elevation, degree reduction, knot insertion and knot removal. Among these degree elevation and knot insertion can be used to refine the design space. However, degree elevation modifies the surface globally during the deformation. Hence this approach is not considered in here. The knot insertion is the process of adding a new knot vector into an existing knot vector sequence without modifying the shape of the surface.

3.1.1 Refinement along s-direction After adding a new knot $r$ into an existing knot vector sequence $S$, from Eqn. (20) the value of $k_s$ increased by one and by fixing the degree of the surface $p$, number of control points increases by one. Consequently the resultant control net $P_{i,j}^s$ is obtained by doing a knot insertion $3$ on each of the $m + 1$ columns of control points.

$$X_i(s,t) = \sum_{i=0}^{n+1} \sum_{j=0}^{m} B_{i,j}^s(s,t)P_{i,j}^t$$

3.1.2 Refinement along t-direction Similarly, a new knot $r$ must be inserted on each of the $n + 1$ rows of control points to obtain the resultant control net $P_{i,j}^t$

$$X_i(s,t) = \sum_{i=0}^{n+1} \sum_{j=0}^{m+1} B_{i,j}^t(s,t)P_{i,j}^t$$

3.1.3 Refinement along both st-direction Refinement can also be performed in each of the parameter direction by adding both $s$ and $t$ knot in a corresponding existing knot vector and the resultant dimension of the control net $P_{i,j}^{st}$ is $0 \leq i \leq n + 1$, $0 \leq j \leq m + 1$

$$X_i(s,t) = \sum_{i=0}^{n+1} \sum_{j=0}^{m+1} B_{i,j}^{st}(s,t)P_{i,j}^{st}$$

where $B_{i,j}^s$, $B_{i,j}^t$ and $B_{i,j}^{st}$ are the updated basis functions corresponds to knot insertion in $s$, $t$ and $st$ direction respectively. Figure 1 shows quadratic $\times$ cubic degree NURBS patch with its control points distribution defined on $S = \{0.0, 0.0, 0.0, 0.0, 0.25, 0.5, 0.75, 1.0, 1.0, 1.0, 1.0\}$ and $T = \{0.0, 0.0, 0.0, 0.0, 0.0, 0.33, 0.66, 1.0, 1.0, 1.0, 1.0\}$. Figure 2 shows the insertion of both knots $s = 0.1$ and $t = 0.1$ at the same time. It is important to note that knot insertion is really change the vector space basis and the NURBS surface is not changed geometrically.

$\text{FIGURE 1: INITIAL NURBS PATCH WITH ITS CONTROL POINTS DISTRIBUTION}$

$\text{FIGURE 2: EFFECT OF KNOT INSERTION AT } s = 0.1 , t = 0.1$
However, knot multiplicity changes the smoothness of the basis function and affects the continuity of the NURBS surface during the deformation. Therefore additional care has been taken to avoid the insertion of repeating knots into the knot vector.

### 3.1.4 Refinement Trigger
In a progressive parametrisation approach, the user can terminate the optimisation and manually enrich the design space using knot insertion. However, to avoid user in the design loop, in this work refinement is triggered in a periodic manner based on the pre-chosen fixed number of design iteration. This determines when to terminate current design space level and refine the design space before continuing the optimisation.

While this trigger could be further improved by considering the convergence history of the objective function value for a given optimisation problem and will be considered in our future work. Based on our experience for this test case, we triggered the refinement at every 10 design iteration.

### 3.1.5 Sensitivity-Driven Adaptation Criteria
The number of design variables and their distribution on the design surface dictates the range of reachable shape modes during the optimisation and hence affects the best achievable design in the design space. In CAD-based parametrisation approach, sensitivities of the objective function with respect to surface grid nodes are projected onto the design variables which controls the deformation of the CAD surface. This projection filters out high-frequency shape modes generated by the CFD mesh and more often the optimisation process is driven by handling only low-frequency shape modes in the design space.

However, the user can enrich the design space using high-frequency shape modes in a progressive manner through a predetermined sequence of uniformly distributed control points globally on the NURBS patches. This does not imply that optimiser can navigate to the optimal design more easily because ineffective parameters are also added in the design space which often leads to ineffective navigation in the design space. An alternative to global refinement is the adaptive refinement where only the most important design variables are added thus maintaining the overall dimension of the search space.

Therefore, a suitable adaptation criterion is needed to precisely enrich the control point distribution locally on the design surface. Generally, high-frequency shape modes are visible only in the surface grid level. For example in Fig. 3 a single high-frequency shape mode is shown by perturbing a node in the CFD mesh but is actually richer than what the CFD discretisaion can resolve. Therefore, in the node-based parametrisation method additional surface regularisation method is normally employed to damp out the high-frequency oscillatory shape modes in the design space. In Fig. 4 the effect of implicit smoothing on the high-frequency shape deformation mode is shown.

\[
\frac{g_i^{k+1} - \beta}{2} (g_{i-1}^{k+1} - 2g_i^{k+1} + g_{i+1}^{k+1}) = g_i^k
\]

where \(g_i^k\) is the gradient of the \(i\)-th node at \(k\)-th smoothing iteration and \(\beta\) is the smoothing parameter.

\[
R_C = \left| g_i^{k+1} \right| = \left| \frac{\partial J}{\partial X_i} \right|
\]

### 3.2 Geometric Constrains
Generally, multiple NURBS patches are used to represent complex aerodynamic shapes. In order to employ multiple neighbouring NURBS patches as the design surfaces in the shape optimisation process, one should guarantee that the geometric continuity between the movable patches maintained during the design process. Based on the smoothness requirements following geometric constrains are need to be maintained among the adjacent patches in the design process \(G_0\) (no gaps), \(G_1\) (tangency) and \(G_2\) (curvature).

In general, one can easily maintain the geometric constrains
between fixed and movable patches by directly imposing constraints on the control points. For example, to maintain \( G_0 \) continuity both patches should retain a common edge and this can be achieved by fixing the control points on an interface (i.e.) control points on an interface between fixed and movable patch should not move at all during the design process. Similarly, \( G_1 \) and \( G_2 \) can be achieved by fixing the additional second and third rows of the control points. However, in between movable patches geometric continuities could be maintained by exploiting convex hull property of splines [31] and more often restricts the design space offered by the NURBS-based parametrisation.

Alternatively, in our approach we impose geometric constraints at a number of linearly distributed test points on the common edges of the adjacent patches. However, this does not require that adjacent patches should maintain same number and control points distribution. The only requirement is that the test points need to be distributed only in pairs with one on each adjacent NURBS patch. With this approach, we can achieve geometric constraints like \( G_0 \), \( G_1 \) and \( G_2 \) between the adjacent movable patches without reducing the design space offered by NURBS-based parametrisation. Previously, the test point approach was successfully tested and implemented in aerodynamic shape optimisation using automotive [17] and turbomachinery [14] test cases. The required number of test points could be determined a-priori by considering each non-zero knot interval and more details can be found in [29].

In this work, only \( G_0 \) constraints are need to be maintained between the movable adjacent patches. Therefore details about \( G_0 \) constraints are only discussed further. More details about \( G_1 \) constraints handling and the constraint recovery can be found in [17, 29].

For \( G_0 \) constraints to be maintained at the interface between two adjacent patches, the linear constraint Eqn.(27) needs to be satisfied at all the test points evaluated at a common edge.

\[
G_0 = X_{i,L} - X_{i,R} = 0 \quad (27)
\]

Therefore, after each design update design perturbations provided by the optimiser should retain \( G_0 \) continuity. This requires that the change in constraints provided through Eqn. (28) should remain zero during the optimisation.

\[
G_0^n - G_0^{n+1} = 0 \quad (28)
\]

Where \( G_0^n \) and \( G_0^{n+1} \) represents the constraint values at design iteration \( n \) and \( n+1 \) respectively.

By linearising the Eqn. (28) and assembling the Jacobian for each of the constraint equation in a constraint matrix \( C \) the allowable design perturbations for each control points can be obtained by solving the linear system of Eqn.(30).

\[
C \delta P = 0 \quad (30)
\]

With \( G_0 \) constraint, the constraint matrix \( C \) has \( M_C \) rows where \( M_C \) corresponds to total number of constrain equations and \( 3 \times N \) columns. Using a projected gradient approach, the design space has to lie in the null space of \( C \) and the design modes \( \alpha \) are the \( N \) basis vector of the null space and determined using Singular Value Decomposition (SVD).

\[
C = U \Sigma V^T \quad (31)
\]

where \( U \) is the \( M_c \times M_c \) unitary matrix, \( \Sigma \) is an \( M_C \times N \) diagonal matrix with positive real numbers on the diagonal and \( V^T \) represents \( N \times N \) unitary matrix. The number of non-zero diagonal entries in \( \Sigma \) determines the rank of the constraint matrix, \( C \) and the last \( (4N - r) \) columns of the matrix \( V \) span the null space of \( C \) and it is denoted as \( \text{Ker}(C) \).
3.3 CAD Sensitivities

In addition to the allowable control point displacements, we also need to compute the sensitivity of the geometry surface with respect to the control points termed as shape derivatives \( \frac{\partial X}{\partial \alpha} \). Robinson et al [27] calculate the shape derivatives using finite difference method which is inexact and subjected to truncation error. Hence, we have chosen to implement a basic light weight CAD kernel based on NURBS and the shape derivatives and Jacobian of the constraints have been calculated by differentiating the CAD kernel using source transformation AD tool Tapenade [32] in a forward mode. The final CAD-sensitivities needed for the gradient-based shape optimisation process can be written as,

\[
\frac{dX_i}{d\alpha} = \frac{\partial X_i}{\partial dP} \frac{\partial dP}{\partial d\alpha} = \frac{\partial X_i}{\partial dP} \text{Ker}(C) \quad (32)
\]

3.4 Surface Mesh Mapping

The surface extractor of our NSPCC CAD kernel reads a surface description file in STEP format and recovers the patch information using NURBS source implementation. However, the NURBS representation and discrete surface grids are totally independent. More often NURBS parametric values are not provided by the grid generator. Hence, the assignment of the corresponding parametric coordinates \((s,t)\) to each surface grid needs to be obtained through the projection of surface grid points onto the original CAD geometry which is stored as a collection of NURBS patches. In literature the process of mapping from geometric space to parametric space \((\mathbb{R}_{x,y,z} \rightarrow \mathbb{R}_{s,t})\) is termed as the point inversion problem.

Surface grid point \(S\) is considered to be on the NURBS patch \(X\), if the minimum distance is less than the specified tolerance. Therefore, one can use an approach based on distance minimisation and with an initial guess \((s_0, t_0)\), the equation to be solved is that the vector pointing from \(S\) to a point on the surface is orthogonal to the tangent vector at that point. This can be mathematically described as,

\[
f(s,t) = (X(s,t) - S) \cdot X_s = 0 \quad (33)
\]
\[
g(s,t) = (X(s,t) - S) \cdot X_t = 0 \quad (34)
\]

where \(X_s\) and \(X_t\) are the derivatives of the surface point with respect to \(s\) and \(t\) respectively.

The roots of the Eqns. (33) and (34) can be calculated using Newton-Raphson method,

\[
\begin{bmatrix} s_{i+1} \\ t_{i+1} \end{bmatrix} = \begin{bmatrix} s_i \\ t_i \end{bmatrix} - J_i^{-1} k_i \quad (35)
\]

where \(k_i = \begin{bmatrix} f(s,t) \\ g(s,t) \end{bmatrix}, J_i = \begin{bmatrix} f_s & f_t \\ g_s & g_t \end{bmatrix}\) and all the elements in the Jacobian matrix \(J_i\) are evaluated at \((s_i, t_i)\) by differentiating the Eqns. (33) and (34) in forward mode using AD tool Tapenade.

4 CASE DESCRIPTION

4.1 Objective Function

The goal of the optimisation process is to reduce the pressure loss of an internal serpentine cooling channel of a gas turbine blades connected by U-bend passage. This passage turns the cooling fluid 180 degrees and of crucial importance since they represent the region of high-pressure loss. As a consequence, design improvements can be obtained by minimizing the mass averaged total pressure loss between the inlet and outlet of the U-bend passage and it is defined as \(J\):

\[
J = \frac{\int_{\text{inlet}} P_{\text{inlet}} \vec{u} \cdot \vec{n} dS + \int_{\text{outlet}} P_{\text{outlet}} \vec{u} \cdot \vec{n} dS}{\int_{\text{inlet}} \vec{u} \cdot \vec{n} dS} \quad (36)
\]

4.2 Flow and Adjoint solver

In this work, the objective function value and flow sensitivities are calculated using in-house RANS based serial compressible flow and discrete adjoint solver named STAMPS [33, 34]. The primal flow solver of STAMPS uses a standard node-centred, edge-based compressible finite volume discretisation using MUSCL type reconstruction of primitive variables using second order accuracy with stable implicit JT-KIRK scheme [35]. The standard Spalart-Allmaras turbulence model was used for the primal flow field along with AUSM scheme for the convective fluxes. The Reynolds number is 15,000 based on the hydraulic diameter \((D_h = 0.075m)\) of the U-bend and the Mach number of 0.1 allows using an incompressible assumption. The U-bend geometry is indeed a challenging turbulent test case and involves strong secondary flow motion. Hence, it is computationally expensive to converge both primal and adjoint fields to sufficient level. Therefore, pressure scaling method suggested by Robert et al. [36] for low Mach number flows is employed to accelerate the convergence behaviour. For a given maximum velocity \(u_{\text{max}}\), density \(\rho\), and Mach number M, the required pressure \(p\) can be calculated as follows,

\[
p = \frac{\rho u_{\text{max}}^2}{\gamma M^2} \quad (37)
\]
ROE based convective flux was tested with Mach number 0.3 without pressure scaling but did not give sufficient convergence for the flow and adjoint solver. This type of pressure scaling ensures good convergence for low Mach number flows and especially suitable for internal flows where pressure drop across the domain is of interest. The adjoint solver in STAMPS is derived from the flow solver using the automatic differentiation (AD) tool Tapenade. The time stepping of the adjoint equations is based on a fixed-point method using the same assembly steps as the primal. In this work, optimisation using both the static and adaptive parametrisation methods are performed using only single core in Apocrita which is an HPC Cluster running Centos 7 Linux.

4.3 Initial Geometry and Computational Set-up

The geometry of the U-bend pipe consists of three parts: inlet, outlet, and walls where the inlet and outlet legs are composed of 10 B-spline patches and 180-degree bend part is composed of 4 NURBS patches directly extracted from the STEP file. The initial geometry with its control point distribution on the design surface is shown in the Fig. 5. The baseline geometry used in this present study having only \( G_0 \) continuity at the turn. Therefore, optimum geometry (See Section 5) obtained using both the static and adaptive parametrisation method exhibits \( G_0 \) continuity at this position.

For design improvement, all the NURBS patches are allowed to move during the design process while the B-spline patches correspond to inlet and outlet legs are remain fixed. To modify the shape, each control point of the NURBS patches is allowed to move in all directions. Hence each contributes 3 degrees of freedom (DoF). In addition to that, control points shared between the fixed and deformable patches are also kept fixed to maintain \( G_0 \) continuity between the fixed-free patch situation.

The total number of control points correspond to each NURBS patch are shown in the Table 1. The mesh used in the optimisation is a pure hex mesh containing around 165k cells and having an average \( y^+ \) value of 1. A preliminary grid convergence study has been performed using three different grid levels regarding the global volume mesh density: a coarse (85k), a medium (165k) and a fine (300k) mesh.

The difference between the objective function values for the medium and fine mesh level is 0.3% which is acceptable for the purpose of the present study. Hence, the medium mesh level with around 165k cells is used for the optimisation. In Fig. 6 a typical 3D view of the computational grid used in the present study is shown.

4.4 Optimiser

In the present work, the direction of the design improvement is controlled by a method of steepest descent, in which perturbation \( (\delta \alpha_i) \) to each design variable is defined as,

\[
\delta \alpha_i = -\gamma \frac{dJ}{d\alpha_i}
\]  

(38)

where \( J \) is the objective function and \( \alpha_i \) is the vector of design variables. The second term in the Eqn. (38) represents the total sensitivity of the objective function with respect to the design variables. \( \gamma \) represents the step size and it is given by Armijo and Wolf based line search condition. A one-shot approach is used to simultaneously converge flow, adjoint and de-
TABLE 1: NUMBER OF CONTROL POINTS ON THE INITIAL GEOMETRY

<table>
<thead>
<tr>
<th>NURBS</th>
<th>No. of control points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top panel</td>
<td>56</td>
</tr>
<tr>
<td>Bottom panel</td>
<td>56</td>
</tr>
<tr>
<td>Inner Bend</td>
<td>56</td>
</tr>
<tr>
<td>Outer Bend</td>
<td>56</td>
</tr>
</tbody>
</table>

5 RESULTS AND DISCUSSION

This section shows the results from both static and adaptive based parametrisation method under the constraint of geometrical $G_0$ continuity.

5.1 Flow Field of the Initial Geometry

The 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. Figure 8 shows the velocity magnitude at the middle plane of the initial geometry. The main feature of flow through a U-bend is the presence of pressure gradient normal to the stream line which provides the centripetal force required to turn the flow around the bend. This results in very low static pressure at the inner
bend, hence a strong adverse pressure gradient as the flow exits the bend and consequently a large flow separation as shown in Fig. 8. The fully developed flow profile disrupts the balance between pressure gradient and centripetal force. As a result of inertial effects, fluid at the center of the bend moves towards the outer wall at the mid-plane and comes back towards the inner wall near the top and bottom walls. This creates a strong secondary flow field which is shown in the Fig. 9. A similar pattern was found in both experimental [38] and numerical simulation [39]. The recirculation region near the inner wall of the exit channel reduces the effective cross-sectional area and accelerates the flow towards the outer wall of the U-bend which enhances losses due to diffusion. These diffusive losses significantly contribute to the total pressure loss of the coolant flow. Therefore an effective design should be able to reduce the effects of the secondary flows near the inner wall region of the U-bend.

5.2 Static vs Adaptive Parametrisation

In the present study, optimisation starts with the coarser design space and refined periodically after each 10 design iteration using knot insertion in both the parameter direction. The refinement criteria is driven by the smoothed node-based adjoint sensitivity information and the control nets of all the four design NURBS patches are refined only in the region where the gradient modes are high.

During the optimisation, the design space has been refined twice. Figures 10 and 11 shows the convergence history of the objective function value for the static and adaptive based parametrisation methods respectively. The degrees of freedom (DoF) corresponding to each design space levels are shown in the Tab. 2. As shown in the Fig. 10, an optimal solution is very difficult to obtain with the conventional static parametrisation approach where the fixed design space can able to converge only to a suboptimal solution with only 15.54% reduction in the total pressure loss. This is the expected result because, with the coarser design space, the optimisation is often driven only by the low-frequency shape modes.

On the other hand, the optimisation using the refined design space is able to reduce the objective function value further and has the potential to obtain the optimal solution outside the fixed envelope offered by the static design space. However, no appropriate improvement can be made with the third design space level. This might be due to the fact that the chosen fixed number of iteration as the refinement trigger is not suitable for this test case to further refine the design space as this exhibits highly non-

### TABLE 2: DEGREES OF FREEDOM FOR EACH DESIGN SPACE LEVEL

<table>
<thead>
<tr>
<th>Design space</th>
<th>No. of Control Points</th>
<th>DoF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Static</td>
<td>224</td>
<td>576</td>
</tr>
<tr>
<td>Level-2</td>
<td>300</td>
<td>780</td>
</tr>
<tr>
<td>Level-3</td>
<td>384</td>
<td>1008</td>
</tr>
</tbody>
</table>
smooth objective function convergence as shown in the Fig. 10. Optimisation terminates after 26th design iteration because the mesh becomes invalid due to the presence of negative volume. Although not shown here the optimisation process could be able to continue by re-meshing the updated step file.

Figure 12 shows the influence of the U-bend geometry change on the velocity magnitude corresponding to static and adaptive parametrisation methods. In both cases, the optimiser has managed to alter the shape of the inner wall to reduce the effects of the secondary flows near the bend exit. The geometry obtained using static parametrisation approach is purely suboptimal and reveals that the coarser design space is less effective at improving the objective function values. As shown in the Fig. 11 optimisation using the refined design space further reduced the objective function value up to 21.1%. The baseline geometry used in this present study having only \( G_0 \) continuity at the turn. Therefore, optimum geometry obtained using both the parametrisation method exhibits \( G_0 \) continuity at this position.

5.3 Flowfield of the Optimised Geometry

When compared with the baseline configuration, both the optimised geometry obtained with a static parametrisation, as well as the one with adaptive parametrisation, suppress the flow separation near the inner wall and reduces the pressure drop significantly, as shown in Figs. 12a and 12b. The reason for the design improvement is twofold. Firstly both approaches increase the inner radius, which 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 optimised geometries. 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. Furthermore, reduces velocities reduce the wall shear stress.

However, the optimised geometry obtained using the adaptive parametrisation method shows better performance improvement than the geometry obtained using static parametrisation method. This is due to the fact that the design space evolved automatically during the optimisation both in terms of number of design variables and their distribution. Because of this, the refined parametrisation can produce superior designs which are outside the restricted envelope offered by the static parametrisation. Figure 13a and 13b show the velocity magnitude comparison of the optimal geometry obtained using static and adaptive parametrisation method.

Fig. 13c and 13d show the secondary vortex structure at the 90° angle of the bend. It can be seen that flow separation is milder along the inner wall using the adaptive design space (See Fig.13c and 13d). This is due to the fact that the cross-sectional area increases at 90° turn, defusing the flow and 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 Fig. 13e and 13f effect of variation of cross-sectional area at the exit channel on the flow field are shown. It can be seen that the outer
FIGURE 13: COMPARISON OF OPTIMAL GEOMETRY OBTAINED USING STATIC (LEFT) AND ADAPTIVE (RIGHT) PARAMETRISATION
corners have become rounded for the geometry obtained using adaptive design space which reduces the wetted area and hence reduces wall shear stress at the outer radius. In addition to that, the optimised geometry exhibits lower acceleration of the flow at the outer wall of the U-bend. This further reduces the adverse pressure gradient along the flow direction which additionally reduces the irreversible diffusion loss in the straight exit section.

Figure 14 shows streamlines to compare the secondary flows. The baseline geometry (Fig. 14a) shows a large area of reversed flow along the inner wall right at the exit of the turn. The optimal shape obtained using static parametrisation, Fig. 14b shows a relatively small area of separation right at the 90° turn which is also visible in the CS-view of Fig. 13c. The existence of a small separation in the optimal design indicates that further improvement may be possible. However, the design obtained using adaptive design space (See Fig. 14c and 13d) further reduces the flow separation throughout the turning and exhibits superior performance than the design obtained using static design space. In this work, optimisation performed using single grid with single core and the computational cost for primal and adjoint computation in the first design iteration is shown in the Table 3.

6 CONCLUSIONS
The CAD-based parametrisation method of Xu et al. [17] which uses a BRep-based parametrisation has been extended to include adaptive design space enrichment. The adaptive parametrisation method has been used to reduce the pressure loss
of the internal turbine cooling channel U-bend. Optimisation begins in a coarser design space focusing on low-frequency shape modes and then automatically refining the parametrisation to include high-frequency shape modes only in the regions where significant high-frequency surface sensitivities are detected. Design space enrichment is performed by inserting knots in both parameter directions of the NURBS patches. The parametrisation 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 the design variables in the design space by using a static parametrisation. The optimised geometry obtained using adaptive parametrisation outperforms the optimal solution obtained using standard static parametrisation approach. Geometric continuity between the NURBS patches is imposed as constraints throughout the optimisation. Finally, the optimised geometry is available in the standard CAD format hence easily interchangeable between disciplines for multi-disciplinary optimisation and manufacturing.

7 ACKNOWLEDGEMENTS

The authors would like to thank the Industrial Optimal Design using Adjoint CFD (IODA) project (http://ioda.sems.qmul.ac.uk), funded by the European Union HORIZON 2020 Framework Programme for Research and Innovation under Grant Agreement No. 642959.

REFERENCES


