Proceedings of the ASME Turbo Expo 2018: Turbomachinery Technical Conference and Exposition
GT2018
July 11-15, 2018, Oslo, Norway

GT2018-76862

DIGITAL GEOMETRY AND MORPHING TO SUPPORT ANALYSIS AND DESIGN

N. Meah, M. Hunt, R. Evans, T. Racz, J. Verdicchio, A. Kudryavtsev
Cambridge Flow Solutions Ltd, The Bradfield Centre, Cambridge Science Park, Cambridge, UK

Bill Dawes
Whittle Laboratory
Department of Engineering
University of Cambridge
Cambridge, UK

ABSTRACT

This paper describes the application of geometry morphing, integrated with meshing and flow simulation, to the topological optimisation of gas turbine film cooling holes.

Using a Genetic Algorithm to manage the digitally represented geometry a wide range of novel cooling hole shapes can be generated and useful improvements in film cooling effectiveness are observed. The simulations suggest that modified vortical flow structures are responsible for improved coolant distribution and coverage at hole exit.

INTRODUCTION

When viewed from a historical perspective engineering product design takes place in cycles: an innovative idea revolutionises a product – then the new design is refined in an evolutionary way out on to its asymptote. Turbomachinery is no exception to this: Figure 1 shows the historical evolution of gas turbine film cooling (taken from the excellent and extensive review by Bunker [1]). The cycles of *revolution* and *evolution* are clear.

In terms of the actual design process, the evolutionary phase is straightforward – parameterise the design and then use CFD and experiment to progressively refine the geometry. The opportunity for competitive advantage here is the speed of that process – which at least in part depends on the efficiency of the parameterisation.

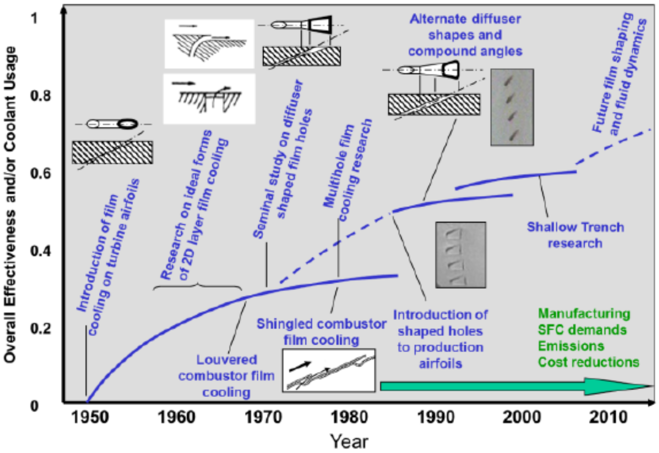


Figure 1. Historical development of gas turbine film cooling (from Bunker [1])

The revolutionary phase is more problematic – a new idea is needed. Traditionally this new idea comes from experienced engineers – or perhaps inexperienced research students – from human imagination. Until a new idea pops up the development of the field is effectively stalled. The area of turbine film cooling is especially challenging as there is ample opportunity for a new design to be topologically different, perhaps radically so, from previous designs.

This need to broaden the design process to include topological optimisation – and for this to be automated somehow – is well recognised and of course not unique to turbomachinery. In structural engineering topological optimisation is becoming widely accepted, see for example Yamada *et al* [2], and there is increasing activity on the fluid side, especially in the worlds of automotive (for example Hopf [3]) and heat exchangers (see Matsumori *et al* [4]). Recent work in the area of turbine cooling includes Pietropaoli *et al* [5] and Iseler *et al* [6]. The key challenge in topological optimisation is representing and editing and managing the geometry.

From a mid-term to long-term perspective there are new technologies already in place that could change our entire approach to geometry: Additive Manufacturing (AM)/3D printing and Artificial Intelligence (AI) based on Artificial Neural Networks (ANN). AM/3D printing only needs an STL description (tessellated surface) to manufacture parts; AI/ANN could capture knowledge without any formal a priori parametrization needing to be imposed. AM enables great freedom to explore new design spaces; this is already receiving much attention in turbomachinery design. Recent examples in the area of turbine cooling are Bunker [1], Ferster *et a*l [7] and Stimpson *et al* [8].

The heart of a simulation system is geometry and the role of the mesh is to deliver this geometry to simulation – CFD, FEA, etc. To support this, we have developed a Digital Geometry solid modelling kernel within our BOXER software system based on Distance Fields managed by Level-Set technology (Dawes *et al* [9,10,11]). The key advantages of this kernel are: its ability to support topology-free geometry transformations; to support arbitrarily complex geometry; and the ability to scale the geometry and its manipulation across parallel compute resource. We are developing a range of technologies to edit and manage geometry taking full advantage of the benefits of our geometry kernel.

In this paper we discuss morphing geometry to support the design of novel turbine film cooling geometries.

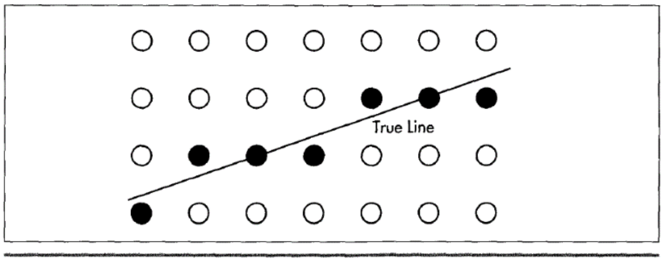
NOMENCLATURE

ϕ – level set identity function
C_d – drag coefficient
f – signed distance field
fr – friction factor
v_n – normal speed of moving front [m/s]
M – similarity metric
N – surface normal
Nu – Nusselt number
Nu^* – Normalised Nusselt number
η – cooling efficiency
$\dot{m}_{coolant}$ – Mass flow rate [kg/ms]
$T_{cross\ flow,inlet}$ – Temperature at the cross-flow inlet [K]
$T_{plenum,inlet}$ – Temperature at the plenum inlet [K]

2. METHODOLOGY

2.1 DIGITAL GEOMETRY MODEL

The famous Bresenham line algorithm was developed in the early 1960’s as a way of representing a line via discrete pixels, “rasterisation”, on the newly emerging Cathode Ray Tube terminals. As Figure 2 illustrates (from nondot.org [12]), the closest pixels to the line are illuminated.



Approximating a true line from a pixel array.

Figure 2. The Bresenham line algorithm (from nondot.org [12])

This is essentially the core idea in digital photography – a picture – in 3D this becomes *geometry*.

Our BOXER software is built on Digital Geometry using generalised 3D versions of the fundamental Bresenham algorithm [9]; illustrated in Figure 3. This consists of an integer representation of geometry down to a chosen length scale – voxels which determine “spatial occupancy”: either occupied, vacant or cut. This is combined with a local scalar Distance Field managed through Level-Set technology – to represent sub-voxel scale geometry. The upper image is part of a ship “rasterised” into voxels; the lower image shows a sketch of contours of Distance Field within a voxel – the blue dots are voxel vertices labelled with the closest distance to the geometry (solid red line); the example grey dot is used as part of the construction of the body-fitted layer mesh.

Digital Geometry offers a number of advantages: the geometry can be distributed onto any compute cluster, including the Cloud - enabling true parallel scalability; geometry editing, and management is supported in a very general, topology-independent way; and finally, geometries of arbitrary complexity can easily be dealt with.

Looking ahead, geometry will need to be available throughout the simulation process chain to support solution adaptive mesh refinement, Fluid Structure Interaction, and automated design optimization. The simulation sizes will be in the Billions of mesh cells, supporting conjugate analysis, and the process chain will have to be end-to-end parallel with no serial bottlenecks.

Hence the geometry modelling itself must be capable of being implemented and scale in parallel – this is trivial for our Digital Geometry kernel but very difficult to imagine with a kernel based on traditional NURBS/BREP constructs.

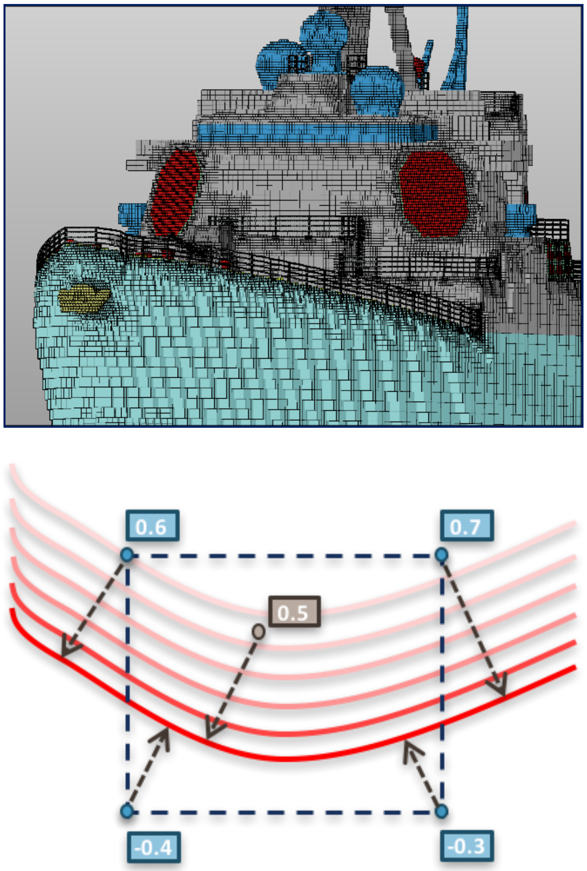


Figure 3. The Digital Geometry Kernel in BOXER; on the top the 3D voxel image; below, the Distance Field storing sub-voxel scale geometry information

An engineer presented the idea for a "filmless camera" to Kodak executives in 1975 but was laughed out of the room (see Telegraph [13]. In 2012 Kodak declared bankruptcy, having failed to adapt to the digital world. Leaving behind analogue geometry and meshing and moving on to the digital world was referred to by Chawner *et al* [14] as a potential “Kodak moment”.

2.2 LEVEL-SET MORPHING

We compute the distance field, ϕ, after capturing the geometry as voxels on an octree mesh; a process referred as *rasterization* illustrated in Figure 4. Each voxel has associated with it the distance to the nearest point on the body, known as the signed distance field $f(\boldsymbol{x},t)$. The surface of the geometry corresponds to $f(\boldsymbol{x},t) = 0$, such that:

$$\phi(\boldsymbol{x},t=0) = f \qquad (1)$$

with the convention: $f < 0$ outside the geometry, and $f > 0$ inside.

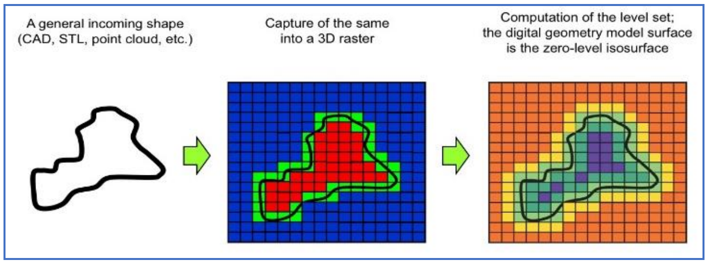

A general incoming shape
(CAD, STL, point cloud, etc.)
Capture of the same
into a 3D raster
Computation of the level set;
the digital geometry model surface
is the zero-level isosurface

Figure 4. Illustration of the Level Set geometry model

It is easy to show that there is an associated evolution equation (a very good overview is provided by Osher *et al* [15]) for ϕ which reads:

$$\frac{\partial\phi}{\partial t} = v_n|\nabla\phi| \qquad (2)$$

where the *speed function* v_n is the normal speed of the zero-distance level (the body surface).

The key concept for parameterisation, geometry and shape editing is to manage the distance field with *ϕ=0*. Consequently, *geometry edits* are just changes to the 3D scalar field, defined somehow/anyhow via the function v_n. There are very many possibilities and we have explored only a few to date.

The subject of this paper is *morphing* between different body shapes. This supports a simple idea: start from two known good designs; use this method to explore, in a very simple high-level single-parameter sense, some other intermediate designs which share features of both inputs. The morphing process between two digital models is described by Equation (2) with an appropriate definition of v_n.

Breen and Whitaker [16] used the signed distance function to define a metric that quantifies the similarity between two solid models from which they developed equations governing the morphing process. These latter equations were then coupled with the volumetric representation of the surfaces, resulting in an expression for v_n. Their very attractive method is summarised in the following paragraphs.

For a solid model T morphing to B, with T being an intermediate shape (being A initially), V_A, V_B, V_T correspond to the volumes and S_A, S_B, S_T their surfaces. The similarity between a shape T and B can be quantified by the amount of V_T

that is shared with V_B. Using a volume integral, the similarity metric reads:

$$M = \int_{V_T} (f_T(\boldsymbol{x}) - f_B(x))d\boldsymbol{x} \qquad (3)$$

The two volumes are identical when M is null, that is when V_T contains the positive values of f.

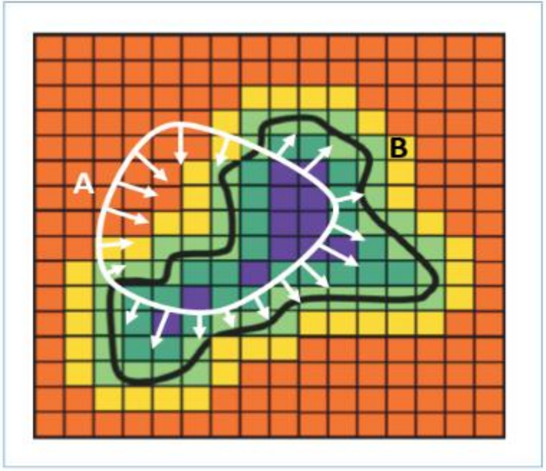


Figure 5. Illustration of the morphing process

An equation that describes the surface motion for each point on S_T can be obtained [16] using the Distance Fields of the two bodies and a “hill climbing” strategy to minimise M:

$$\frac{\partial \phi}{\partial t} = (f_T(\boldsymbol{x}) - f_B(\boldsymbol{x}))|\nabla \phi| \qquad (4)$$

$|\nabla \phi|$ corresponds to the surface normal. This equation states that at each time step, each point on the surface S_T moves in the direction of the surface normal with a magnitude proportional to the “overlap” between the two bodies with the points further away from S_B travelling faster. If V_T and V_B overlap then some segments of S_T will contract ($f < 0$), while the other ones will expand ($f > 0$) and only the points on S_B are not moving since $f = 0$. When S_T corresponds to S_B, M is null, hence the morphing process is completed. This process is sketched in Figure 5.

In Figure 6 the morphing process is illustrated for a practical case: a turbine blade with internal cooling passages showing a wide variety of intermediate geometries. Complex geometries can be morphed as a whole, (as illustrated in Figure 6), or for other cases the morphed shapes can be embedded within more complex system. For example, focussing on the design of pin-fins within a fixed, surrounding geometry, e.g. the blade internal cooling passage.

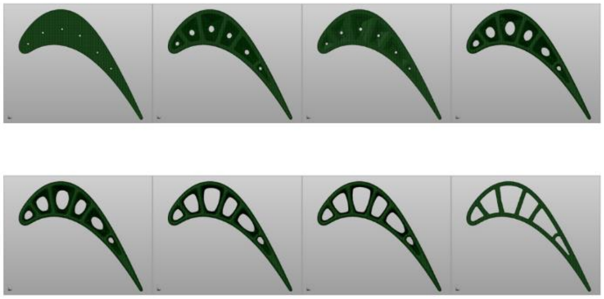


Figure 6. Morph between a turbine blade without internal cooling to one with

Positions of each geometry relative to each other can be varied and this yields different intermediate shapes. The point is illustrated in Figure 7 where a sphere is morphed to a cone. In both cases, the sphere intersects with the cone, however, in (II), the sphere has been translated to the right. It is seen that not a single intermediate shape between (I) and (II) are similar. A similar trend is to be expected if the cone had been rotated.

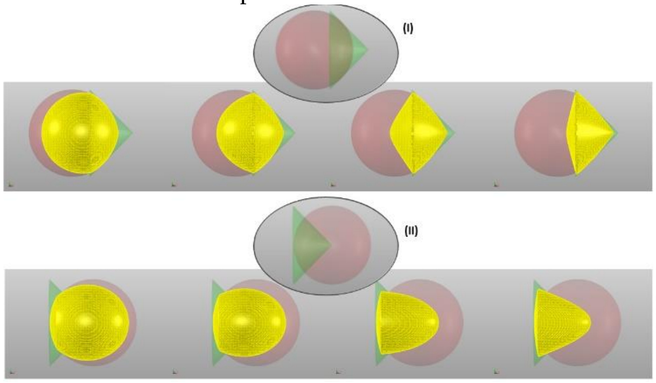


Figure 7. Morph between a sphere to a cone for two different relative positions of the sphere

2.3 PRACTICAL IMPLEMENTATION

To make real-world engineering use of this ability to morph geometries the approach has to be implemented in practical software which can not only manage the morph but also generate appropriate meshes for simulations. And all of this needs to be scriptable so that an automated workflow can be built.

It is important to maintain the integrity of the geometry as it morphs such that each intermediate geometry represents a plausible geometry in its own right. The physics-based nature of the morphing process is a big advantage over other aproaches which simply interpolate between two geometries.

This is supplemented by re-initialising the Distance Field for intermediate stages of the morph. Since ϕ is cast as a signed distance function f can be re-initialised to the current level set $\phi = 0$ by solving the following on the background unstructured mesh [17]:

$$|\nabla\phi| = 1 \quad (5)$$

This in practice consists in the re-rasterisation of the parametric geometry. The function ϕ itself is approximated on the octree with conforming finite element shape functions to provide higher accuracy in calculation of function derivatives [18]. The octree cells and the approximation are distributed between several CPUs to be processed in parallel. The default resolution used of the octree cells for this process corresponds to one-hundredth of a bounding box encompassing the initial and target shapes. Testing on numerous cases revealed that this value constitutes a good trade-off between accuracy and cost.

The method can be summarised as follows. First, the initial and target geometries are rasterised. Next, the signed distance function f is initialised with the zero-level being the target Digital Geometry model. Equation (4) is then solved until the distance fields between the two geometries is similar, with the periodic re-initialisation of f. Finally, the transformation from a volumetric representation to a parametric one is achieved by the classic Marching Cubes algorithm [19].

2.4 FLOW AND MORPH COUPLING

It is intended that the current morphing plus the meshing can be coupled with FEA, CFD, MBD solvers, and potentially with full automation (providing scripting tools are available).

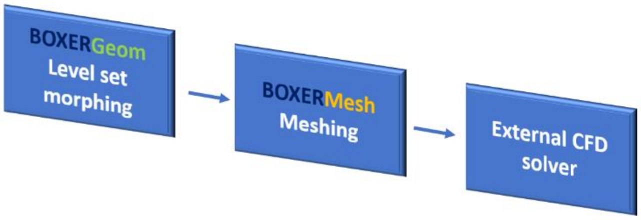


Figure 8. Coupling strategy to make a morph-mesh-solve workflow

As BOXER can execute sequences of commands using its inbuilt scripting commands (based on Lua), this enables the automation of more complex operations, such as repeated two-shape morphing. Primarily, and more fundamentally, the script enables the automation of the level set morphing operation, and the mesh generation. Extension of the morph-and-mesh automation to include a flow solver by using python scripting, results effectively in a morph-and-solve workflow tool with minimal user intervention (see Figure 8).

The automated workflow reads as follows. Upon specifying the positioning of the two shapes, the level set morphing is executed once, with all the intermediate shapes available only at the end of the operation. Subsequently, the meshes are generated. The script proceeds with the CFD calculations, only when all the meshes are available (possible improvements include managing the operations simultaneously with the available resources). For the present analysis, the operations were executed on a desktop computer and a HPC cluster (details are provided for each application). The flow solver Fluent™ (version 18.1) is used throughout the applications.

Applications of a morph-and-solve coupling are manifold. Naturally, the shape of a geometry has a strong influence on its physical property i.e. increase of surface is correlated with higher heat transfer, whereas a teardrop-like shape is associated with low drag coefficient. A two-shape level set morphing could be used in bi-optimisation problem e.g. a user wants to design a shape which has a heat transfer as high as possible from shape A, while having a drag coefficient as low as possible from shape B. Or, the objective function can be a user's appreciation on the design: an automotive designer seeking to find the car which is the best compromise between an aesthetic design (shape A), and the aerodynamic performance of another design (shape *B*).

MULTIPLE SHAPES MORPHING

In some instances, several geometries might stand out, either because they optimise each a different objective function, or a set of topologically different geometries optimise the same objective function. Ferster *et a*l [7] investigated different shapes of cooling pins for gas turbine blades applications. They noted that additive manufactured triangles (with two different orientations) and cylinders were found to feature increased heat exchange over the conventional pin fins used.

The two-shape morphing could be used as a building block to create a rich design space, where shapes would share the geometrical properties of the parent. For instance, in the above-mentioned example although the cylinder and the face facing triangle are geometrically different, they both displayed similar performances (increased improvement was observed for the triangle with the point facing the flow). A design space could be populated with newly-generated shapes combining the "key" features of the parent shapes which accounts for increased heat transfer.

Naturally the design space complexifies as the number of parent shapes increase, herein we restrict the discussion to three shapes. The possibilities for creating and discovering new geometries increase considerably already with three shapes (referred to as A, B, C in Figure 9). Morphing between each geometry results in three morphing paths (represented with the solid lines). One shape for each of these paths (D, E, F) can morph with numerous others (dashed lines).

For example, if 10 intermediate shapes are saved for the first three morphing paths, the total number of morphing operations would result to 340 for the intermediate shapes on these paths. The possibilities could increase further if the shapes on the next three morphing paths are considered (H, G, I), which would rapidly become impractical.

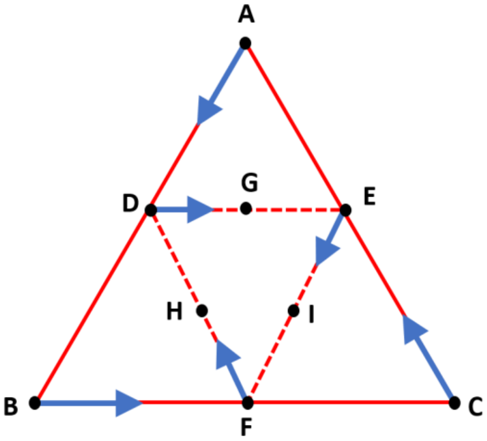


Figure 9. Design space for three shapes

However, when coupled with a flow solver the whole space does not necessarily need to be explored. Driven by the concept of *micro-Genetic Algorithm*, a powerful, yet simple, optimisation technique can be devised, which enables the automated workflow to populate the design space progressively. Following the specification of the shapes A, B, C the method reads:

1. Executes a level set morph operation for A to B, B to C, and C to A. Subsequently, the meshes are generated, and the objective function for each morphing path is obtained.
2. For each objective function, a shape corresponding to the global extrema is saved, referred as D, E, F (first generation). In the case for example the best shape between $A \rightarrow B, C \rightarrow A$ is A, other shapes might be selected to prevent premature convergence.
3. Step 1 and 2 are repeated for the three new shapes until the variance of the objective function decreases under a prescribed threshold.
4. When diversity is lost (low variance) the best shape can be morphed with other shapes for further exploration of the design space, either manually by careful choices of the source/target shapes, or automatically.

This recursive algorithm is computationally efficient as the data associated with few shapes needs to be saved only, and, the total number of morphing operations for 10 intermediate shapes saved for each morph path amounts to 6 after the second generation, and 9 with the third. The algorithm was adopted here due to the rapidity at which it yields solutions and its robustness since the results do not vary significantly between each run. As any genetic algorithm the termination criterion is not easy to define, here the exploration of the space by selecting shapes far in the design space can be pursued from the best one to bring diversity in the population, hence increasing the chance to reveal better candidates.

The algorithm has been presented for the optimisation of one objective function, however, it could be extended to account for more. One feasible way to achieve, for two objective functions: is following step 4), the best shape (according to the objective function 1) is initially morphed with the shape (among the available ones) that optimises the objective function 2, and subsequently step 4) can be repeated for further exploration of the space.

A common intersection where all the shapes overlap with each other dictates how much the shapes can change from their original design. The point is illustrated in the Figure 10 where three shapes (triangle, square, circle) are overlapping. The minimum volume that exists between the overlaps (highlighted in red) would correspond to the smallest possible intermediate shape, while the largest one would correspond to the square in this case. Expansion of this confined design space, could be carried out by adding “mutations” which could take the form of transformations, since moving the positions of the shapes alters the area/volume of overlap (see again Figure10), hence changes the morphing patterns, and limiting shapes. However, we have not investigated the concept further here.

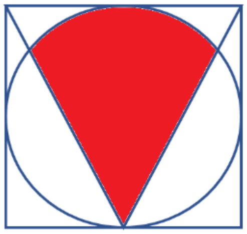


Figure 10. Overlapping area between three shapes: sphere, square, circle.

3. APPLICATIONS

BASIC SHAPES

The two shapes coupling is first illustrated in Figure 11: given a sphere (source) and a cube (target) the operations are executed as per the automated workflow described earlier. Since the rectangular domain is made of an inlet, outlet, and walls otherwise, a flow simulation of the two end shapes was first carried out to evaluate the size of the domain, by ensuring that the flow structures were not impacting the walls. Layers were added to the meshes for each shape to accurately resolve the boundary layers.

The Reynolds number is ~3.10^4, the turbulent flow is modelled with $k-\omega$. The intermediate shapes are computed every 5 iterations; however, one could choose an adaptive increment which depends on how the objective function evolves to minimise expensive flow solves.

Prior to running the script, a mesh convergence study was performed for a sphere, and with a mesh sized for $Y^{+}\sim 1$ a drag coefficient of $C_d \sim 0.65$ was found (NASA reports ~0.5 at $Re\sim 10^4$). The discrepancy is attributed to the flow model not fully capturing the vortex shedding. Notwithstanding this, the turbulence model captures the change of the drag coefficient with respect to the change in the shapes, which is sufficient for the present study. Once the drag coefficient has converged with a minimum of 300 iterations to prevent premature convergence, the mesh of the next intermediate shape is generated.

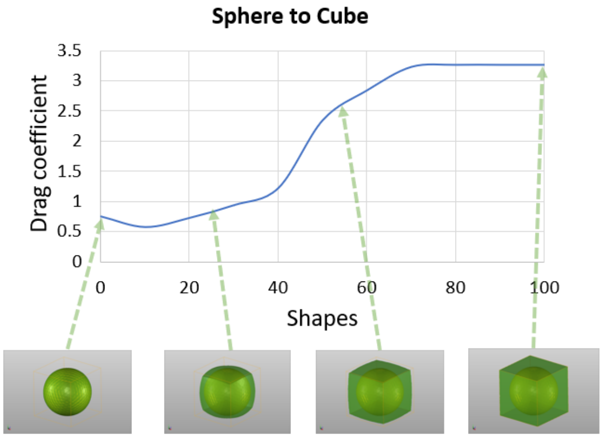


Figure 11. Drag coefficient for a sphere morphing to a cube

It is worth emphasizing that the shape minimising the drag coefficient is not quite a sphere (according to RANS anyway) but one that strongly resembles one with features of a cube. Design of even these simple shapes with traditional geometry editing tools based on BREP/NURBS CAD is an arduous task even to the expert user. Level set morphing technique provides an effortless way to access a unique design space.

The computational cost for each operation is indicated in Table 1. The resolution of the octree cells corresponds to the default one (one-hundredth of a bounding box encompassing the two geometries) which result in a level set morphing taking approximately the same time as the mesh generation. It is reminded that BOXER mesh generates the mesh following an operation of rasterisation fully controlled by the user, which differs from the one generated for the morphing operation. As for the tessellated geometries obtained, the quality of the mesh depends on the resolution of the digital model (see [10] for more details).

Full Level set morphing	Meshing (BOXER)	Flow solver (Fluent)
Rasterisation with default resolution: ~15s x 3 *morphing:* ~20s	*Mesh cell count:* ~ 0.18 million cells	*Turbulence model:* RANS $k-\omega$
Total: 1min5s	**Total: 1min20s**	**Total: ~50s**

Table 1. Computational cost for each operation for the sphere morphing to a cube with a desktop computer with 12 cores (Intel® Xeon® CPU E5-2630 v3, 2.40GHz, and memory of 15.2 GB).

Next, the optimisation method is applied for three shapes: a cylinder, cone and a cube. Despite the simplicity of the case, practical applications could be envisaged in fluidized bed reactors for the design of catalyst particles starting from the more standardised ones (and similarly for external ballistics). The shapes have been chosen for their relatively high drag coefficient to facilitate the search of better shapes in the design space (see Figure 12 with the direction of the flow indicated with the red arrow).

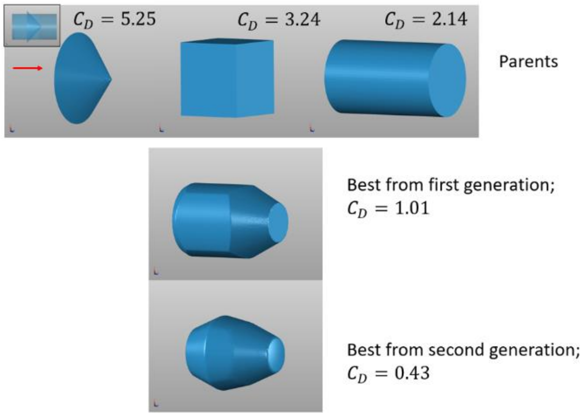


Figure 12. Parent shapes, and best ones obtained from the optimisation technique

For the present optimisation calculation, we save 25 geometries (increment of 5). The setup is the same as the one described for the cube and sphere, except that the size of the domain was decided by a flow simulation of the cylinder (bulkiest shape).

The rest of the operations were carried out without user's interventions as per the algorithm discussed previously.

In Figure 13 the drag coefficient is plotted for each morphing path: the solid line corresponds to the shapes obtained between the parents, the dash lines to the ones obtained between the best of the first generation, and the dotted lines correspond to the ones obtained for the best of the second generation. In the legend each letter refers to a morphing path. For clarity, the best shapes from the first generation are indicated by a letter followed by the number of the shape. For example, $A10 - B10\ /D$ reads: the morphing path from the shape which minimises best the drag coefficient between the cylinder and the cone ($A10$) to the one between the cylinder and the cube ($B10$). In this case, both shapes correspond to the 10th shape in the morph trajectory. The morphing path is referred to as $D.$ The shapes that minimise best the drag coefficient from the first generation are closer to a cylinder. Overall, the best shape is obtained during the second generation, and exhibits features of the three parent shapes with a closer similitude to a cylinder.

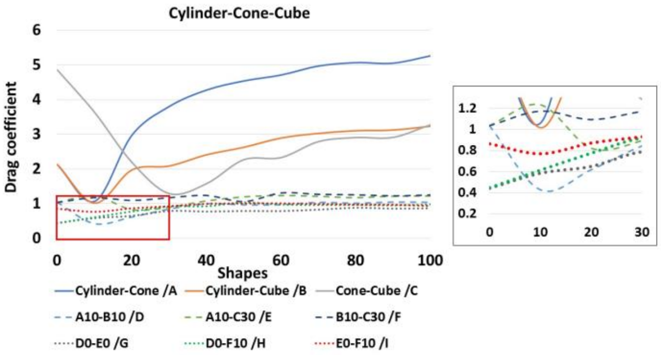


Figure 13. Drag coefficient for three shapes morphing: cube, cone, cylinder

The mean, and the variance of the objective function is plotted in Figure 14 where it is seen that the diversity decreases abruptly from the shapes of the second generation, together with a decrease in the mean, indicating that the algorithm leads to convergence. When the variance decreases under a threshold (2%), further exploration of the space was carried out by morphing the best shape with ones from previous generations, up to 5 times (with each time ensuring that the simulations were proceeding until the variance was under the threshold). However, in this case, the best shape was obtained from the second generation with a drag coefficient $\sim 200\%$ better than the best of the parent generation.

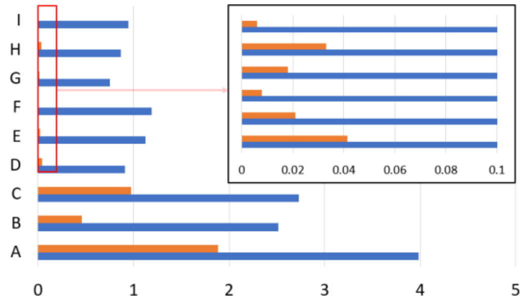


Figure 14. Mean (in blue) and Variance (orange) for the objective functions in Figure 13

GAS TURBINE FILM COOLING HOLES

Next, the level set morphing is applied to the classic case of a film cooling hole on a turbine blade surface. Low temperature air is injected through the cooling holes on a blade surface to form a protective layer between the blade surface and the hot gas medium. The interaction between this film cooling jet with the main flow, at various blowing ratios, leads to a variety of flow structures and cooling efficiencies. This is a very well published field of research (see again the excellent review by Bunker [1]); the objective here is not really to produce an innovative design but to show the potential of the present morph-solve workflow to allow rich and interesting new design spaces to be created and explored.

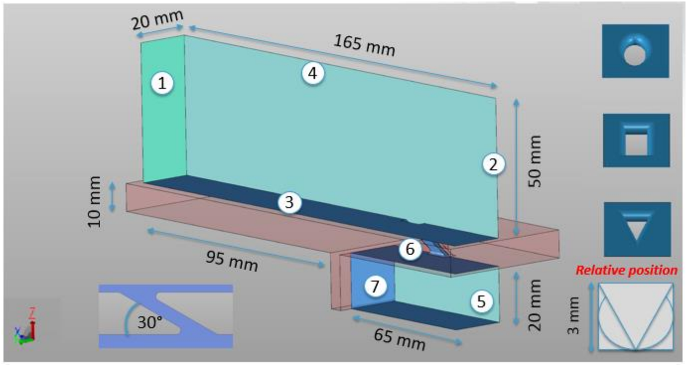


Figure 15. Dimensions of the domain, with the part of the geometry which is morphed (red), and 1 – Pressure outlet, 2 - Cross flow inlet (stagnation inlet), 3 – Test section (adiabatic), 5 – Plenum inlet, 6 – Film cooling hole (adiabatic walls, half model)

Figure 15 shows the domain – plenum, cooling hole and test plate. Three different "parent" shapes of cooling holes are investigated, square, circular, and triangular. They are morphed as part of this domain shaded **red** in the Figure. The choice of shapes is motivated by the rich design space they yield (parameters defined below). The relative position of the holes to each other is illustrated in the bottom right of the Figure. Half of the domain has been simulated, with a symmetry boundary condition at the hole centre plane.

A sample sequence of a morph for the square → cylindrical → triangular is illustrated in Figure 16.

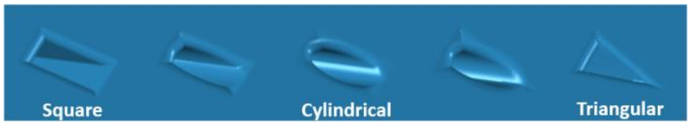


Figure 16. Rectangular cooling hole morphing to a cylindrical then to a triangular one.

Simulations were run with hot primary flow as this is a more realistic situation in a turbine; the boundary conditions are summarised in Table 2 below. A mesh of the square cooling hole is shown in Figure 17, where it can be seen that in addition to face refinement, volume refinements were used to capture more accurately the region where the air with different temperatures interacts, and for better representation of the subtle variation in geometry. Additionally, layers were added to capture more accurately the boundary layers, resulting in resolution to about $Y^{+} \sim 1$ on the test section. The overall cell count is typically ~ 1.2 million cells, varying slightly between each shape.

$\boldsymbol{T_{cross\ flow,inlet}}$	1600 $\boldsymbol{K}$
$\boldsymbol{T_{plenum,inlet}}$	700 $\boldsymbol{K}$
$\boldsymbol{p_{t,cross\ flow,inlet}}$	13.128 **bar**
$\boldsymbol{p_{t,plenum\ inlet}}$	13.44 **bar**
$\boldsymbol{Re}$ (main stream)	10^4
$\boldsymbol{Turb_{int}}$	1%
$\boldsymbol{Turb_{plen}}$	1%

Table 2. Boundary conditions

Similar to the previous analysis, the Fluent™ flow solver was used in RANS mode with the turbulent model $k-\omega$ SST employed, with a turbulent intensity of 1%, both for the cross-flow inlet, and the plenum inlet. It is known that RANS models generally provide poor prediction of lateral coolant mixing and distribution – and probably LES is really needed going forward but is too expensive for the current purpose (see for example the work done by Carnevale [20]). Nevertheless, for better prediction of heat transfer the $k-\omega$ model is recommended [21] and the SST variant has the added benefit of capturing better the separating flow and reduces the sensitivities to inlet free stream turbulence properties.

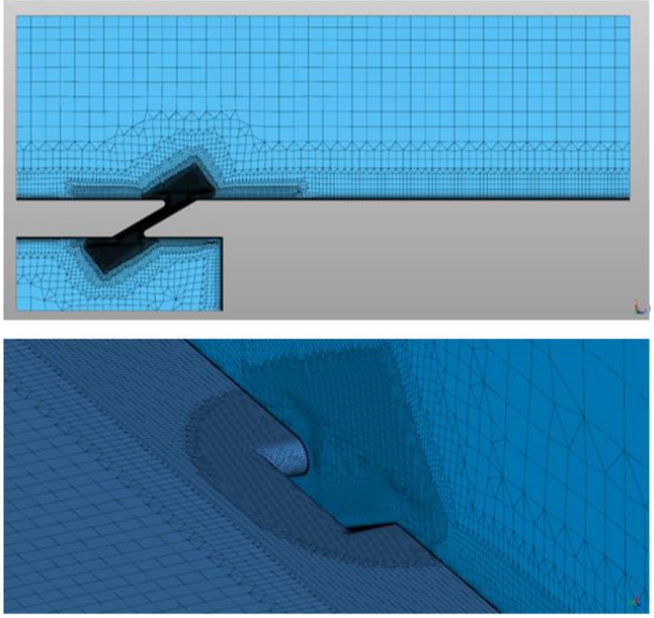


Figure 17. Mesh used for the rectangular cooling hole, (top: isosurface of the full system, and bottom: zoom on the hole, with an isosurface at the hole centre plane)

The computation costs are indicated in Table 3. The more expensive level set operation is due to the more refined resolution needed to capture the cooling holes which are much smaller than the actual geometry morphed (see Figure 15). Once the meshes are computed, they are automatically sent to a cluster where Fluent™ is executed. Each case was run up to a maximum of 1000 iteration, however, monitoring the residual for each case (using the scripting tool) identified the cases that did not reach convergence. For these cases, the number of iterations were doubled.

Full Level set morphing (1x12 cores)	Meshing (BOXER) (1x12 cores)	Flow solver (Fluent) (8x12 cores)
Rasterisation with one-twentieth of the default resolution: ~6mins x 3 *morphing:* ~30mins	*Mesh cell count* ~ 1.2 million cells.	*Turbulence model:* RANS $k-\omega$ SST
Total: 48min	**Total: 25min**	**Total: 12min**

Table 3. Computational cost for each operation for the cylindrical cooling hole morphing to a triangular one. The CFD is solved on a HPC cluster, with 8 nodes and 12 cores: Intel® Xeon® CPU E5-2620, 2.00GHz, with a memory 256 GB for one node, and 65 GB (see table 3 for the other operations).

The cooling efficiency η and the mass flow rate $\dot{m}_{coolant}$ define the figures of merit in this case. The cooling efficiency was defined as:

$$\eta = \frac{T_{cross\ flow,inlet} - T_w}{T_{cross\ flow,inlet} - T_{plenum,inlet}} \qquad (6)$$

The cooling efficiency averaged over the test surface area is plotted vs the average mass flow rate in Figure 18 for all the shapes generated.

It is seen that the parent shapes are far apart in the design space, passing, as they do, quite different mass flows for the same pressure drop. Numerous shapes maximising better the cooling efficiency arise following the morphing between the parent shapes, in particular the ones resulting from the morph between the square→ circle, where the best shape is referred to as S1. A Pareto front represented as a dot-dashed line demarcates the shapes from the parent, and first generation with the ones from the next generations. It is observed that further shapes with better cooling efficiency emerge during the second generation, with the shape S2, which has an added advantage of having a lower mass flow rate.

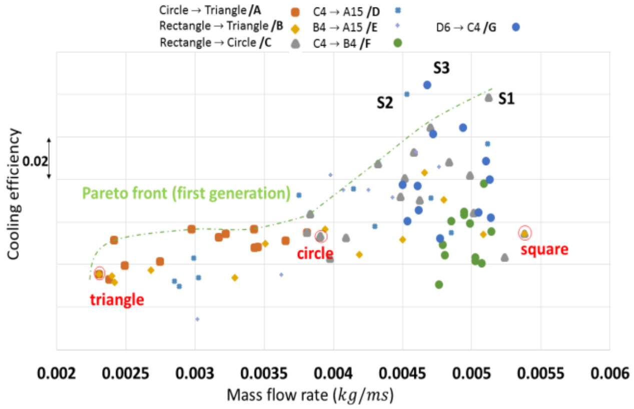


Figure 18. Average cooling efficiency vs average mass flow rate for all the shapes generated

Due to the relative expensive objective functions, further exploration of the space has not been automated in the present analysis, instead the shapes that optimise best the cooling efficiency, namely S1 (first generation) and S2 (second one), have themselves been morphed. Their morph results in a range of shapes with very different properties, among which one is maximising further the cooling efficiency (S3).

Finally, Figure 19 shows contour plots of the cooling efficiency for the square shape and for the shape **S3** which delivers a significant improvement in performance at very similar mass flow. Despite the relatively minor differences in shape the behaviour of the film cooling differs greatly between the two shapes. It is observed that in addition to impeding hot gas from flowing underneath the cooling film, the newly generated shape allows the generation of anti-counter rotating vortices which promote the lateral spread of the cooling air.

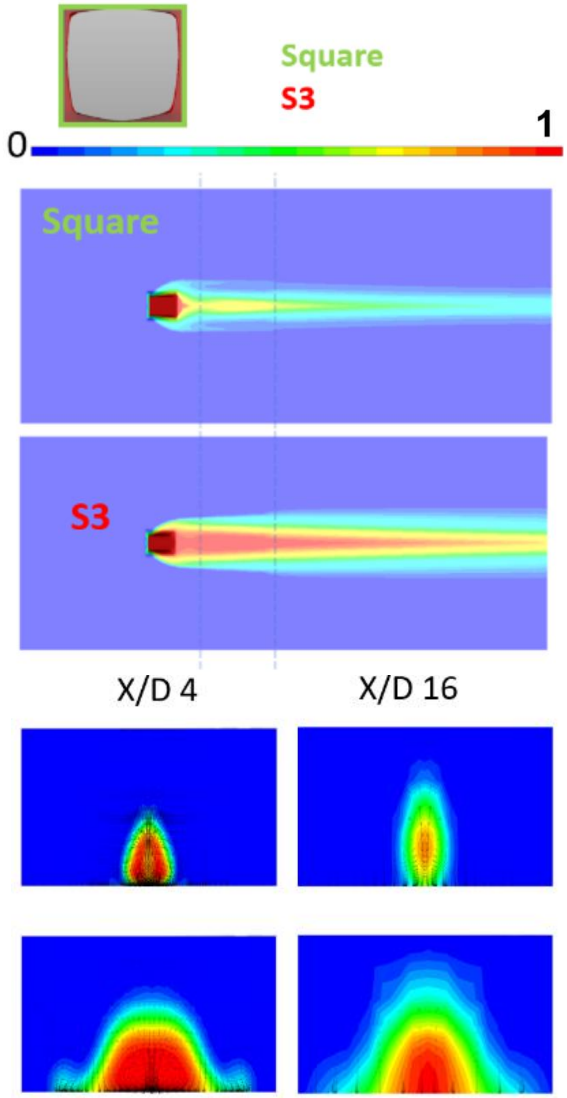


Figure 19. Contour plots of the cooling efficiency for the square shape and the shape optimising best the cooling efficiency (S3).

4. CONCLUSIONS

In this paper we have attempted to address the issue of automating the generation of *revolutions* in design by coupling the morphing of geometries, managed by a Genetic Algorithm, with meshing and flow simulation. The whole activity is enabled by a novel Digital Geometry solid modelling kernel embedded within the software.

We illustrated the potential of the new approach with two simple examples. The first was the drag of various bodies; the second was the efficiency of gas turbine film cooling. In both cases interesting and unexpected shapes emerged from the rich design space and demonstrated, at least from the point of view of RANS simulation, improved functional performance.

For the cooling hole, although the geometries are not far from traditional shapes, relatively subtle changes in geometry, were discovered automatically by the process, and which appeared able to successfully modify the vortical structures emerging from the hole to improve both coverage and cooling efficiency.

Finally, the response of the turbulent flow field to the relatively subtle changes in the design are probably not well captured by RANS and the analysis, on at least candidate improved designs, would benefit from improved modelling – even LES. In terms of manufacturing the geometries presented, even the ones with the subtlest changes could be successfully Additively Manufactured – and this is already happening in industry.

ACKNOWLEDGMENTS

We are very pleased to acknowledge partial financial support from Innovate UK via the GHandI, GEMinIDS and AuGMENT Consortia and also to our Development Partners. The authors are grateful to Cambridge Flow Solutions Ltd. For permission to publish this paper.

REFERENCES

[1] Bunker RS "Evolution of turbine cooling", ASME Paper GT2017-63205, Charlotte NC, June 2017.

[2] Yamada T, Izui K, Nishiwaki S & Takezawa A "A topology optimisation method based on the Level Set method incorporating a fictitious interface energy" Computational Methods Applied Mechanical Engineering, 2010.

[3] Hopf A "Finding the perfect flow-port development using CFD topology optimisation and adjoint solver" STAR Global Conference, Prague 2016.

[13] http://www.telegraph.co.uk/technology/2016/05/13/ biggest-mistakes-in-tech-history/

[14] Chawner JR *et al* "The Path to and State of Geometry and Meshing in 2030: Summary" AIAA 2015-3409.

[15] Osher S & Sethian JA "Fronts propagating with curvature dependent speed: Algorithms based on Hamilton-Jacobi formulation" Journal of Computational Physics 1988; 79(12):12-49.

[4] Matsumori T & Kondoh T "Topology optimisation for fluid-thermal interaction problems under constant input power" Struct. Multidisc. Optim., 47: 571-581, 2013.

[5] Pietropaoli M, Ahlfeld R, Montomoli F, Ciani A & D'Ercole M "Design for additive manufacturing: internal channel optimisation" ASME paper GT2016-57318, Seoul, 2016.

[6] Iseler J & Martin TJ "Flow topology optimisation of a cooling passage for a high-pressure gas turbine blade" ASME paper GT2017-63618, Charlotte NC, 2017.

[7] Ferster KK, Kirsch KL & Thole KA "Effects of geometry and spacing in additively manufactured micro channel pin-fin arrays" ASME paper GT2017-63442, Charlotte NC, 2017.

[8] Stimpson CK, Snyder JC, Thole KA & Mongillo D "Effectiveness Measurements of Additively Manufactured Film Cooling Holes" ASME paper GT2017-64903, Charlotte NC, 2017.

[9] Dawes WN "Building Blocks Towards VR-Based Flow Sculpting" 43rd AIAA Aerospace Sciences Meeting & Exhibit, 10-13 January 2005, Reno, NV, AIAA-2005-1156.

[10] Dawes WN, Kellar WP & Harvey SA "Viscous Layer Meshes from Level Sets on Cartesian Meshes" 45th AIAA Aerospace Sciences Meeting & Exhibit, 8-11 January 2007, Reno, NV, AIAA-2007-0555.

[11] Dawes WN, Kellar WP, Harvey SA "A practical demonstration of scalable parallel mesh generation" 47th AIAA Aerospace Sciences Meeting & Exhibit, 9-12 January 2009, Orlando, FL, AIAA-2009-0981.

[12] www.nondot.org/sabre/Mirrored/.../gpbb35.pdf
Chapter 35: *Bresenham is Fast and Fast is Good*

[16] Breen DE & Whitaker RT, "A Level-Set Approach for the Metamorphosis of Solid Models", IEEE Transactions on Visualization and Computer Graphics, Vol 7, No. 2, pp. 173-192, April-June 2001.

[17] Mourad HM, Dolbow J & Garikipati K "An assumed-gradient finite element method for level set equation. International Journal for Numerical Methods in Engineering", 2005 00:1-6

[18] Fries T-P, Byfut A, Alizada A, Wah Cheng K & Schroder A "Hanging nodes and XFEM", International

Journal for Numerical Methods in Engineering, 2000;00:1–6

[19] Lorensen WE & H. E. Cline HE, “Marching cubes: a high-resolution 3D surface construction algorithm. In Proc. of ACM SIGGRAPH 87, pages 163-170, 1987.

[20] Carnevale M, Salvadori S, Manna M & Martelli F “A comparative study of RANS, URANS, and NLES approaches”, 10th European Conference on Turbomachinery, Fluid dynamics & Thermodynamics, April 2013.

[21] Harrison KL & Bogard DG “Comparison of RANS turbulence models for prediction of film cooling performance”, Proceedings of ASME Turbo Expo 2008. GT2008-51423