Proceedings of GT2009
ASME Turbo Expo 2009: Power for Land, Sea and Air
June 8-12, 2009, Orlando, Florida, USA

GT2009-59817

APPLICATION OF TOPOLOGY-FREE OPTIMIZATION TO MANAGE COOLED TURBINE TIP HEAT LOAD

W N Dawes[1], WP Kellar & GA Richardson

CFD Laboratory, Department of Engineering,
University of Cambridge, Cambridge CB2 1PZ, UK
&
Cambridge Flow Solutions Ltd,
Compass House, Vision Park, Cambridge, CB4 9AD, UK

Abstract

The application of automated design optimization to real-world, complex geometry problems is a significant challenge—especially if the topology is not known *a-priori* like in turbine internal cooling. The long term goal of our work is to focus on an end-to-end integration of the whole CFD Process, from solid model through meshing, solving and post-processing to enable this type of design optimization to become viable & practical. In recent papers we have reported the integration of a Level Set based geometry kernel with an octree-based cut-Cartesian mesh generator, RANS flow solver, post-processing & geometry editing all within a single piece of software—and all implemented in parallel with commodity PC clusters as the target. The cut-cells which characterize the approach are eliminated by exporting a body-conformal mesh, with a viscous layer, guided by the underpinning Level Set. This paper develops this work further with a simple, generic study showing how the basic functionality can be scripted and then used as the infrastructure for automated optimization applied to an example in turbine cooling: managing blade tip heat lead.

Introduction

As confidence in flow simulations increases, so does application to ever more challenging cases. Current challenges in the world of turbomachinery include many examples of complex flows in complex geometries—like secondary air systems and turbine internal & external cooling. Conventional simulation tools, CAD systems, mesh generators, flow solvers & post-processors perform well individually but less so when attempts are made to script them together in an automatable system—especially when problem sizes become large and the geometry is very (or even arbitrarily) complex. The research reported in this paper was originally motivated by asking the questions: what would it take for the *entire* CFD Process from CAD-to-mesh-to-solver-to-post-processing to be inherently parallel, scalable and without any serial bottlenecks? What would it take for the geometry to be editable—without topological restriction? What would it take for this to be scriptable & automatable?

In a series of papers we have tried to chart a way forward. Dawes [1] proposed a methodology that was deliberately different from the standard, orthodox process chain which scripts together best-in-class tools (CAD import, mesh generation, solver, etc…).

[1] wnd@eng.cam.ac.uk

The essence of this new approach was the integration of a geometry kernel based on a Level Set approach with an octree-based cut-Cartesian mesh generator, RANS flow solver and post-processing all within a single piece of software. The basic building-block work was reported by Dawes [1] and the parallelization of the entire system was reported in Dawes [2]. Replacing the cut-cells with layers of body-conformal meshes was described in Dawes *et al* [3]. The ultimate goal of this research is to allow rapid prototyping design optimization to take place automatically on real geometries of arbitrary size & complexity.

This paper reports recent progress towards these goals. By exposing the core functionality of our software to external agents via appropriate interfaces and scripting we show that it is possible to build a fully automated end-to-end process capable of managing, editing, meshing and flow solving very complex geometries. We illustrate this with a scoping study: the optimization—with topology change—of the cooled tip of a gas turbine blade.

Managing cooled turbine tip heat load is one of the several key steps in managing turbine blade life. The geometry of high pressure ratio gas turbine blades has become very complex: trenched tips; dust holes and cooling holes within the trench fed from the internal, perhaps double-triple, passage; part-chord squealers; pressure-side cooling; cut-back trailing edges; etc. The associated flowfield is complex and relies for success on management of interactions between coolant flows, leakage flows and vortical structures. Traditionally design has relied on trial and error, insight and experience. Modern developments in automated design optimization have been difficult to apply due to the complex geometries, the need to consider topological geometry changes and the unreliability of mesh generators. This paper illustrates the potential of the BOXER paradigm.

The BOXER simulation system

Algorithmic architecture

The BOXER software represents an attempt to integrate the whole CFD Process "end-to-end" from CAD import through mesh generation flow solution & post-processing—including the ability to edit the geometry. BOXER has been implemented, at least to prototype standard, end-to-end in parallel—including all data flows between functional units. BOXER was inspired by exploration of the possibilities offered by the integration of the solid modeling directly with the mesh generation & with the flow solution—this inspiration combined ideas from solid modeling (see for example Samareh [4,5] & Haimes *et al* [6]) with virtual sculpting (see for example, Galyean *et al* [7], Perng *et al* [8] and Baeretzen [9]) combined in the context of a simple, cut-Cartesian mesh flow solver (see Bussoletti *et al* [10] or Aftosmis *et al* [11]). In a series of publications, Dawes [1] first set out these building blocks and showed their potential as a rapid prototyping design tool; then Dawes [2] showed how these building blocks could be efficiently implemented in parallel; then Dawes *et al* [3] showed how layer meshes or body-conformal meshes could be exported to overcome the cut-cell issue.

The backbone of BOXER is a very efficient octree data-structure acting *simultaneously* as a search engine, as a spatial occupancy solid model and as an adaptive, unstructured mesh for the flow solver. This provides unlimited geometric flexibility and very robust mesh generation. The solid model is initialised by the import of a tessellated surface from a variety of potential sources (most CAD engines have an STL export) or by direct interrogation of the CAD solid model kernel itself (Haimes *et al* [6]). The solid model is captured on the adaptive, unstructured Cartesian hexahedral mesh very efficiently by cutting the tesselated boundaries using basic computer graphics constructs developed for interactive 3D gaming (aimed at real-time collision detection). This geometry capture is very fast; for example, a body represented by about 1M surface triangles can be imported into a mesh of around 11M cells (with 6-7 levels of refinement) in approximately 2 minutes on a single, top-end PC—very much faster in parallel. The spatial occupancy solid model is sampled as a distance field and managed as a Level Set; this forms a solid modeling kernel to support the activities of the code. Adaptive mesh refine/de-refine for the flow and for the geometry, via the distance field, enables both moving bodies and topology editing.

Conformal mesh export to drive third-party flow solvers

The cut cells are the big disadvantage of the Cartesian approach—Dawes *et al* [3] described how viscous layers and/or body-conformal meshes may be generated using the underpinning Level Set distance field geometry kernel as a guide. The conformal mesh export (together with simple removal of the hanging nodes by inserting appropriate pyramids, tetrahedra etc.) allows third party flow solvers, like FLUENT®, to be used—and this was the case in this paper. The intention at this stage is to provide full topological freedom to more standard CFD design (i.e. flow solvers), thus to test the process against known and current engineering, as a building-block prior to our ultimate goal of completely self-contained rapid-prototyping.

Key to the success of this approach (as described in Dawes *et al* [3]) is the use of optimization algorithms to control the quality of the exported volume mesh. This optimization manages volume mesh quality using metrics like skew, warpage, etc.—in fact whatever metrics the third-party solver is sensitive to—and the volume mesh is then "managed" into an acceptable range so the exported mesh is *guaranteed* of solvable quality. This is absolutely crucial for the success—indeed for the very viability—of an automated process aimed at real, complex geometries likely to undergo topological change during optimization. The trade-off for the guaranteed solvable volume mesh is that sometimes surface mesh smoothness is compromised; typically most of the surface is mirror-like but difficult areas distort, but by no more than a fraction of the fundamental smallest mesh scale (so whether a solver can see the distortion is a moot point).

Viscous layers can be added also using the same methodology. The viscous layers are essentially the captured surface mesh (which is quad-dominant) extruded as a set of prisms back into the domain. The initial wall-normal extrusion distance, expansion ratio and target number of layers are user-specified. The actual number of layers achieved in practice is determined by mesh quality constraints managed by an optimizer. Hence, aerofoil-like surface are easy to add layer meshes to; within corners, for example, the ability to grow layers is suitably constrained.

Automatable, scriptable functionality

Design consists in performing analysis *repeatedly*, guided by appropriate metrics—and *changing the geometry*. The absolutely critical bottleneck here is changing the geometry and rebuilding a good quality mesh; this challenge is increased hugely for cases like turbine internal cooling since the topology is *a priori* unknown. An additional challenge is that this geometry editing & re-meshing activity must be automatable—and hence guaranteed.

Pitch-chord	0.96
Inlet angle/Exit angle	37°/56°
Inlet Mach number/Exit isentropic Mach number	0.35/0.72
Reynolds number	0.6E+06
Upstream reference stagnation pressure/coolant pressure	110kPa/117kPa
Upstream stagnation temperature/coolant temperature	400K/300K

Table 1: Basic parameters of the TG1 cooled turbine blade

We have addressed these challenges by adding a scripting capability to BOXER, implemented via the Lua language, to allow the core functionality to be driven by an external agent. Various geometry editing tools & techniques can be used—and these tools can be themselves arbitrarily defined and imported—their action on the geometry is also scriptable. The high-level scripting language is object-oriented both computationally and literally; example commands are:

- `boxer.loadCylinder(length, radius, point, axis)` to define the location of a cylindrical virtual tool;
- `boxer.useTool(-1)` to commit the tool geometry in a Boolean sense to the current overall geometry (i.e. to cut a hole; `(+1)` would add a cylinder-shaped protrusion instead);
- `boxer.loadArbitraryTool(STL-file)` to load an arbitrarily-defined sculpting object from STL;
- `boxer.outputViscousMesh(file, layer-height, number of layer cells)` to *automatically* output a body-conformal viscous mesh of the current state of the geometry.

The general approach is simply similar to that of other third-party codes; like FLUENT® (having a high-level journalling language) and FIELDVIEW® (with a high-level language very similar to Lua).

Topology-free optimization: a simple scoping study—application to manage cooled turbine tip heat load

Geometry, mesh & basic flowfield

The candidate geometry is a simplified cooled turbine blade with a trenched tip and designated TG1; this has been designed and tested and analysed as part of an EU FP6 project on turbine cooling—AITEB2 (see Janke *et al* [12]). The basic physical parameters are summarized in Table 1. Although clearly a rotor blade it was intended to be tested in a stationary cascade and was therefore simulated here without rotation. The blade internal geometry delivering the coolant to the trenched tip was a simple plenum not a fully-featured set of cooling channels.

A mesh containing about 7.8M cells was generated with BOXER direct from the STL representation of the geometry exported from Unigraphics CAD. The basic CAD in fact contained no cooling holes or dust holes; all such detail was generated entirely by geometry editing within the optimization process.

This mesh was then exported, after the addition of a viscous layer two cells deep, in body-conformal, hybrid form and in FLUENT format. Figure 1 shows general and detailed views of the mesh; in the actual optimization simulation the mesh was also adapted to follow the blade wakes. This mesh generation (including the editing of cooling holes etc into the geometry) was fully scripted and took about 18 minutes to accomplish. The exported mesh is generated to conform to the published FLUENT mesh quality metrics.

For flow simulations FLUENT 6.3.26 was used—run on a cluster of four PCs. The flow is subsonic so the pressure based solver was selected, with second order flux terms and the Spalart-Almaras turbulence model. Fully-converged simulations took around 36 wall-clock hours. Figures 2 & 3 show some general views of the simulation. In particular the blade surface Y+ values were in the range 10-15 over the whole blade (and obviously lower in the trench itself). The mid-span flowfield is perfectly classical—mid-loaded and with thin, well behaved-boundary layers. Some sense of the complex nature of the flow in the tip region is visible in Figure 2b.

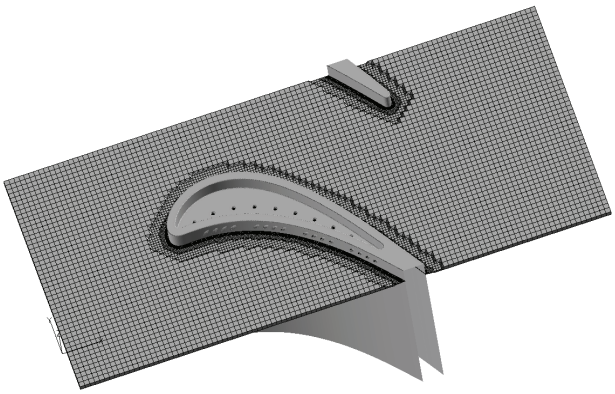
Objective function definition

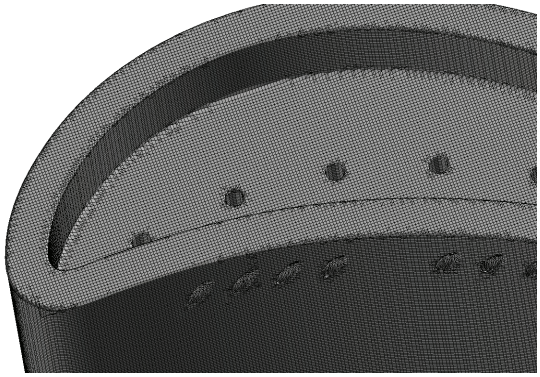
Two objective functions were considered. The first was the "cooled row efficiency", defined in Table 2 below, which is like the classical mass-weighted total pressure loss ratio with corrections to make allowance for the (presumed dissipated) kinetic energy of the cooling jets.

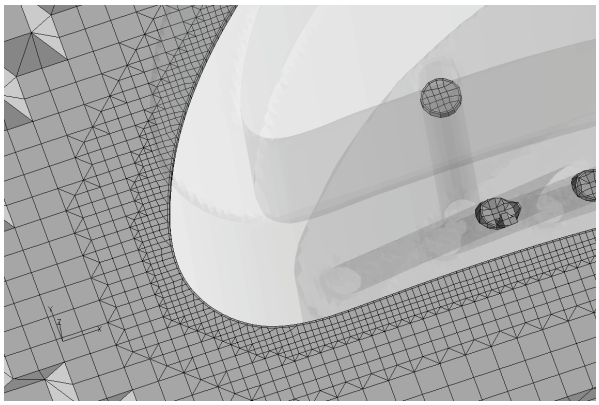
$$\xi_{ex} = 1 - \frac{(1+\sum_{j=1}^{n} c_{m,j})\, w_2^2}{w_{2,is}^2 + \sum_{j=1}^{n} c_{m,j}\, w_{c,is,j}^2}$$

$$= 1 - \frac{\left[1-\left(\frac{p_2}{p_{02}}\right)^{\frac{\kappa-1}{\kappa}}\right]\left(1+\sum_{j=1}^{n} c_{m,j}\cdot\frac{T_{0c,j}}{T_{01}}\right)}{1-\left(\frac{p_2}{p_{01}}\right)^{\frac{\kappa-1}{\kappa}} + \sum_{j=1}^{n} c_{m,j}\frac{T_{0c,j}}{T_{01}}\left[1-\left(\frac{p_2}{p_{0c,j}}\right)^{\frac{\kappa-1}{\kappa}}\right]}$$

Table 2: The "cooled row efficiency"







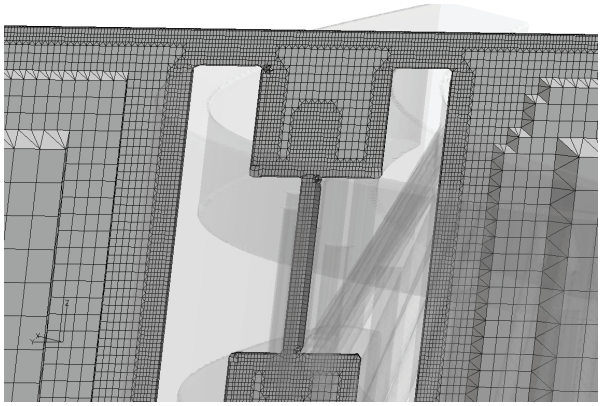


Fig. 1a: Basic blade-blade-mesh

Fig.1b: Detail view of the trenched tip

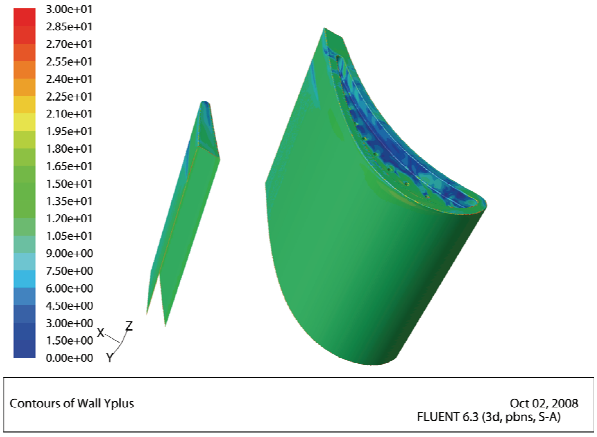


Fig.2a: Predicted Y+ values

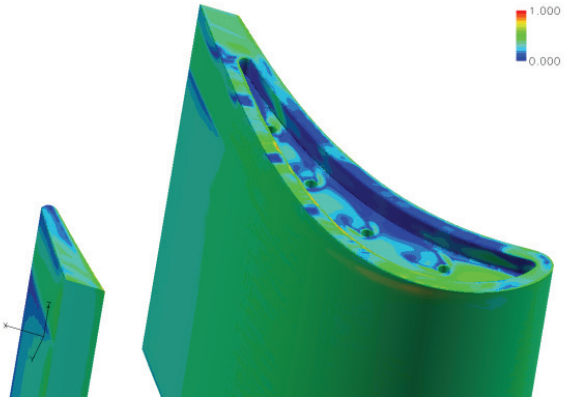


Fig.2b: Predicted surface isentropic Mach number

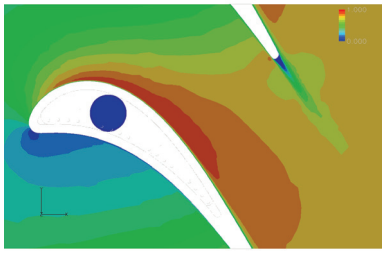


Fig.3a: Mid-span Mach numbers (range 0.0-1.0)

Fig.3b: Mid-span total pressure (range 80-110kPa)

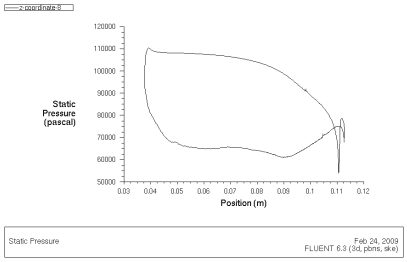


Fig.3c: Mid-span static pressure distribution

The second objective function was based on the classical "film-cooling effectiveness" defined as

$$\varepsilon = (T_\infty - T_{wall})/(T_\infty - T_{coolant})$$

which is really a measure of coolant delivery. The second objective function was evaluated as a single number by a simple area-average of the local effectiveness over the tip region of the blade as defined in Figure 4a as our main interest is tip heat load.

A composite objective function was also defined as the simple arithmetic average of the two individual functions.

Parameterization of the geometry

The geometry parameterization is a key step. This might be approached in various ways. The CAD solid model itself might be parametric – in which case those parameters would be scripted and then BOXER would simply regenerate meshes from each new solid model export. Otherwise BOXER provides for a set of virtual tools which can be used to edit the geometry–again via a suitable script. Conventional blading parameters like lean and twist could be used either via parameterized CAD or by mapping onto BOXER's toolset. Which approach is used depends on the circumstances and available data.

For the present proof of concept paper two aspects of the design of the turbine tip were parameterized for automated use within the optimization: the number of dust holes within the trenched tip and the depth of the trench itself.

To manage the dust holes a polynomial function was fitted based to the original location of the dust holes and leading/trailing edge location of the bottom squealer surface. Then variable numbers of new dust holes were equally spaced along this polynomial under the control of the optimizer (see Figure 4b).

The total area of the dust holes was kept constant and all dust holes in each set had equal diameters—this

keeps their total mass flow approximately constant. The dust holes were cut into the solid model of the blade by BOXER using cylindrical virtual tools as illustrated in Figure 4c—the optimizer controls the location of the tools via a script and then BOXER uses the tools to edit the underpinning Level Set solid model for the blade and then rebuilds the mesh—re-exporting it in body-conformal FLUENT format.

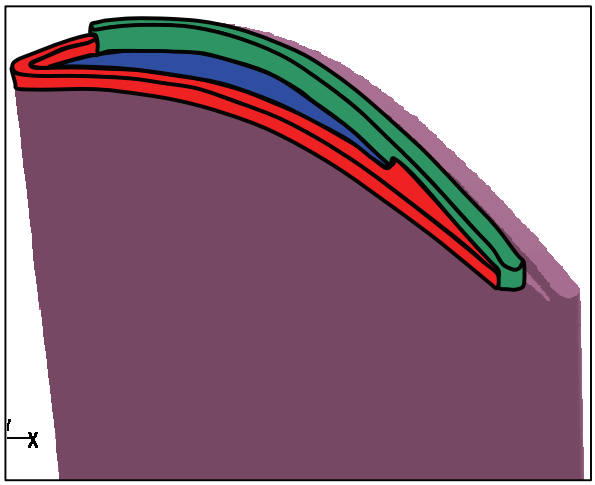


Fig.4a (top): the definition of the surface areas (red, blue and green) used to evaluate tip heat load

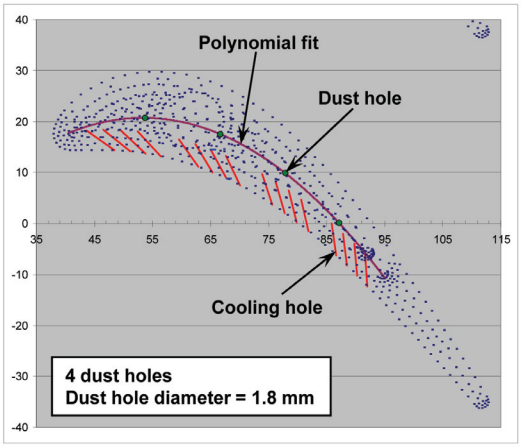


Fig.4b: Parameterization of the dust hole location

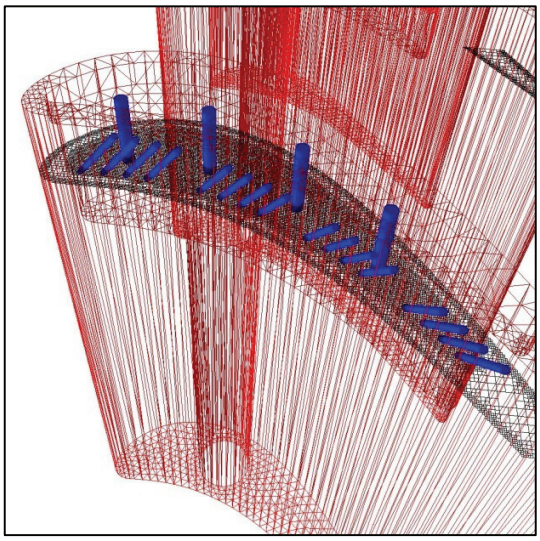


Fig.4c: The virtual tools used to edit the cooling and dust holes into the basic blade geometry

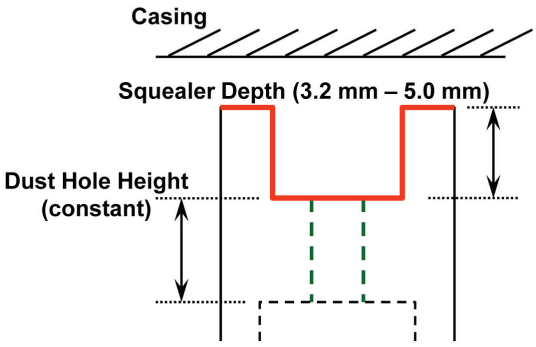


Fig.4d: Parameterization of the squealer trench

The depth of the trench is varied over a fixed range, as shown in Figure 4d, via BOXER's Free Form Deformation tool (see Sederberg *et al* [13]). This is a technique which encloses an object (in this case the trench) within a 4×4×4 3D spline box and then maps an adjustment to any of the spline control knots into a deformation of the enclosed object. (In this application the deformation is obviously a very simple one.) Again, after editing the Level Set solid model for the blade, BOXER rebuilds the mesh. This geometry editing/re-meshing/re-exporting phase takes about 20 minutes overall and is guaranteed to produce a mesh with solvable mesh quality metrics.

Optimization results

The optimizer selected for this study was a simple Design of Experiment—DoE—which produced a range of solutions for a range of parameters from which a simple Response Surface could then be constructed. Table 3 below shows the 3×3 parameter

range selected. All other aspects of the blade geometry—including the tip clearance—and all primary and secondary boundary conditions were kept the same. This is obviously a very simplified cartoon of a practical optimization but is intended as proof of concept.

4/3.2mm	4/4.1mm	4/5.0mm
7/3.2mm	7/4.1mm	7/5.0mm
10/3.2mm	10/4.1mm	10/5.0mm

Table 3: The parameter ranges selected for a simple Design of Experiment study: format is number of dust holes/squealer trench depth

The entire DoE ran completely automatically with no human intervention—topological geometry editing, mesh regeneration, flow solution & extraction of the objective functions—and on a cluster of 4 PCs took wall-clock 14 days.

Each simulation deserves attention in its own right from the point of view of better appreciating the revealed flow physics but there is not enough space here; instead some indicative results will summarize the results, then the optimization DoE itself will be discussed. A number of recent papers, for example Krishnababu *et al* [14], give extensive discussions of cooled turbine tip physics. Figure 5 & 6 show predicted blade surface stagnation temperature distributions for the suction and pressure sides respectively. Clear differences can be seen between the cases in terms of the three-dimensional flow structures in the tip region associated with different number of dust holes and differing over-tip leakage vortical structures.

There are two main physical effects at play in the tip region: the trench flow and the discharge from the dust holes. The trench depth affects the gross form of the overtip leakage flow: a blade with a shallow trench tends to have a single overtip leakage vortex structure, anchored from blade peak suction; with a deep trench this vortical structure is augmented by a powerful secondary flow driven chordwise within the trench by the blade leading edge stagnation. Meanwhile the dust holes deliver cooling air into the trench with approximately constant velocity ratio (the hole discharge velocity is set by the local static pressure field and the plenum pressure) but in cases with fewer, larger holes the discharge/free-stream

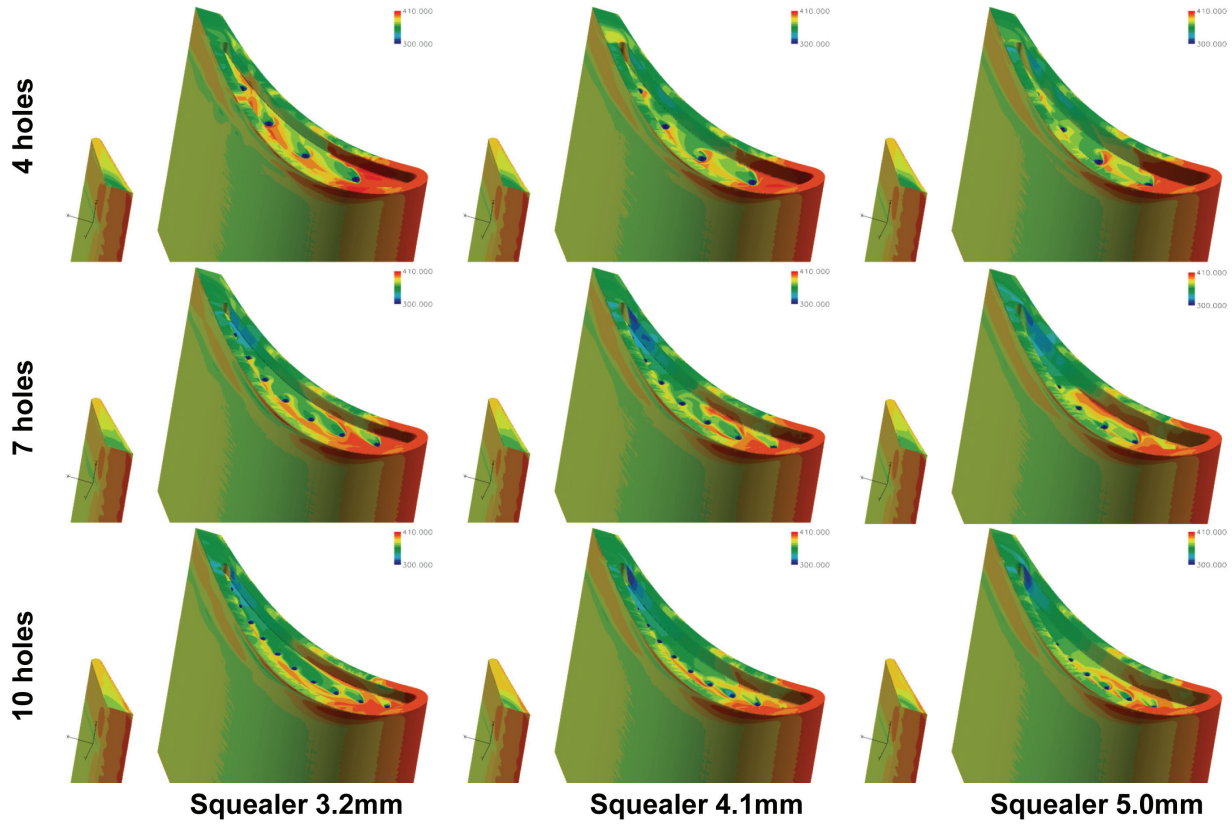


Fig.5: DoE results—surface stagnation temperature—suction side view (range 300-410K)

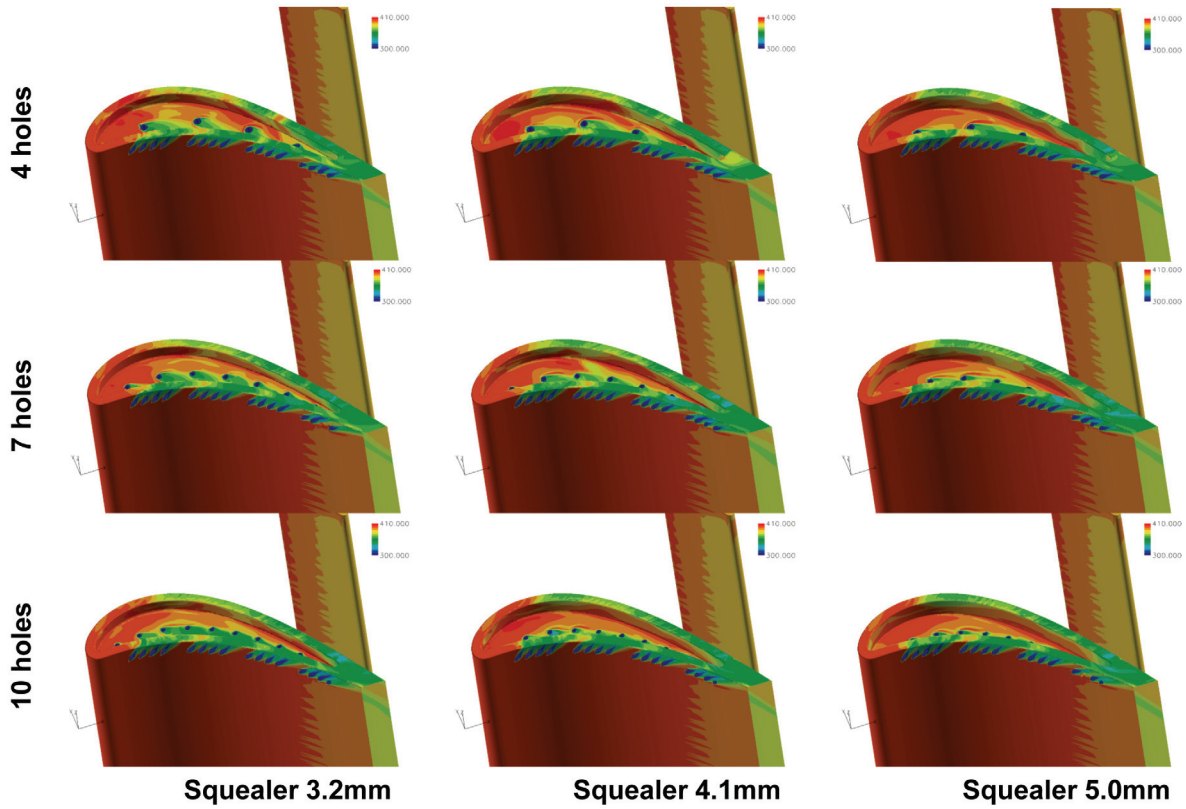


Fig.6: DoE results—surface stagnation temperature—pressure side view (range 300-410K)

momentum ratio is higher and so this leads to greater coolant penetration into the surrounding flow. This in turn leads to poorer coolant delivery to the blade surfaces within the trench and hence to poorer film cooling effectiveness. This is clearly visible in Figures 5 and 6.

Figure 7 attempts to illustrate further the way the highly three-dimensional overtip flow structures depend not only on the squealer trench depth but also on the interaction of this with the coolant discharge from the variable number of dust holes. As described above, with deeper trenches the overtip structure are more complex and with fewer dust holes more coolant is entrained into these structures and kept away from the blade surface.

In quantitative terms, Figure 8 show contours of a series of least squares quadratic Response Surfaces fitted to the various objective functions used in the DoE: film cooling effectiveness, cooled row efficiency, the row efficiency not corrected for the cooling kinetic energy loss and the combined objective function (defined above). The Response Surface plots are oriented in the same way as the flowfield plots in Figures 5-7 and the parameterization data in Table 3.

A number of interesting observations arise. In terms of blade row loss coefficient, the loss not corrected for the cooling kinetic energy (Fig.8b) is around 4.7% and varies little over the parametric geometry range (the weak trend being for reduced losses with reduced trench depth). When corrected for the cooling (Fig.8c) the blade row loss level is around 11.5% - much higher but as expected as coolant mixing is known to be a significant loss mechanism—again varying little over the geometric parameter range—mainly because the parameterization tried to keep the coolant mass flow approximately constant by varying dust hole diameter with number of holes.

By contrast, the film cooling effectiveness (Fig.8a) displays a much larger variation—33% to 37%—with geometric parameterization. For small squealer trench depths (left side of the plots) there is clearly a benefit to having a given coolant flow discharge in smaller individual amounts through a much larger number of dust holes; indeed in Figure 6 the delivery

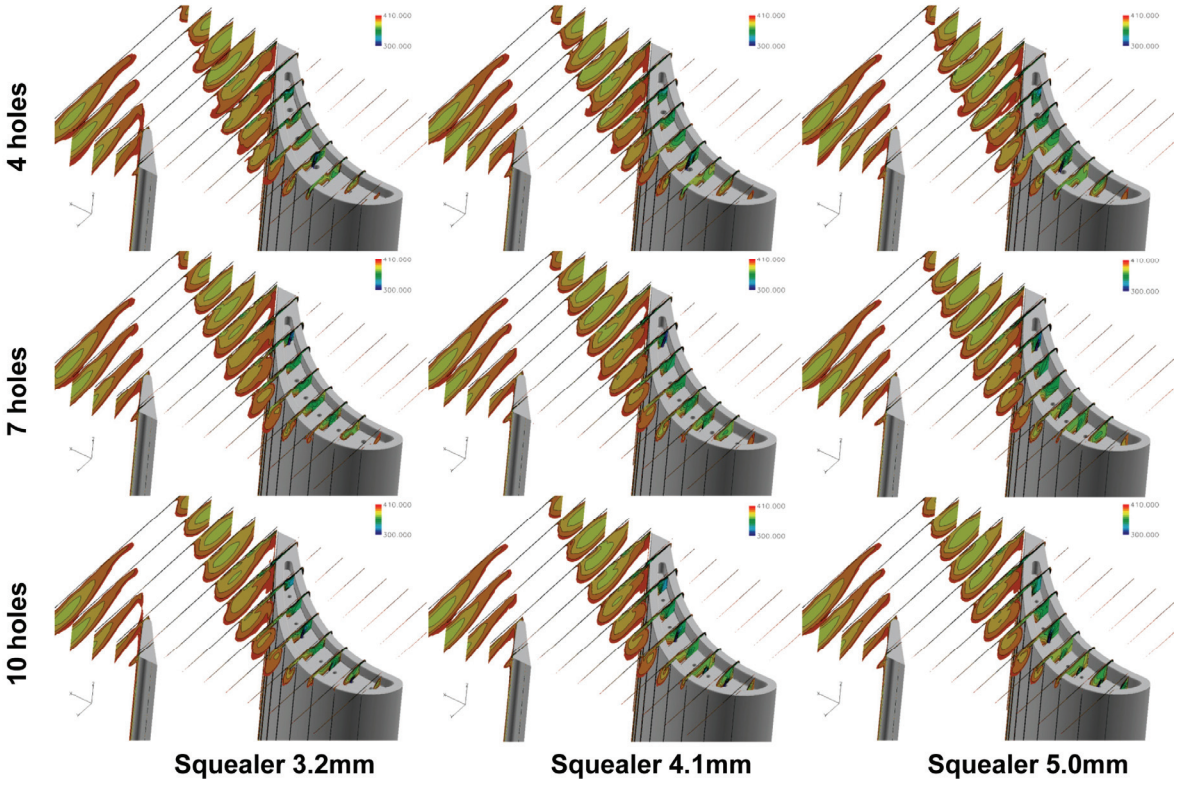


Fig.7: DoE results—thresholded stagnation temperature cut planes

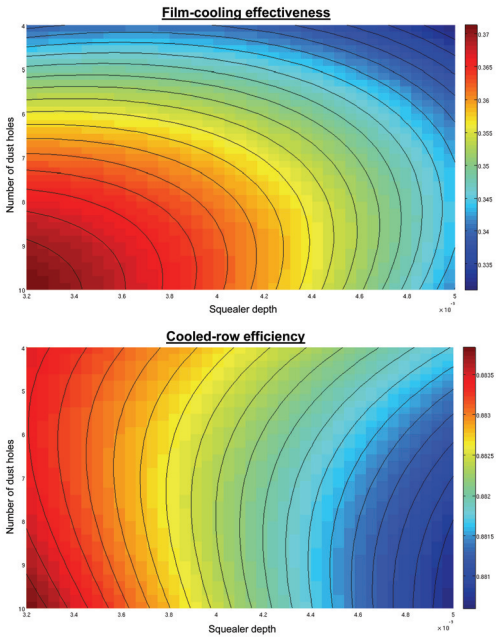


Fig.8a: film cooling effectiveness—range 0.335 (blue)-0.370 (red)

Fig.8b: cooled row efficiency—range 0.881 (blue)-0.884 (red)

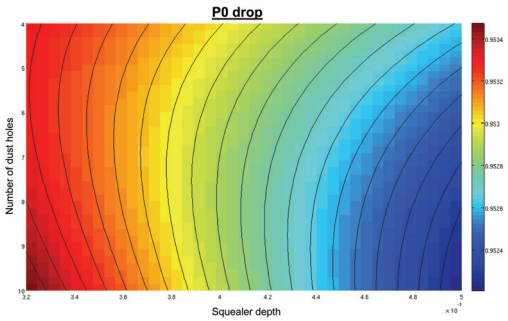


Fig.8c: row efficiency not corrected for the cooling kinetic energy loss—range 0.952-0.953

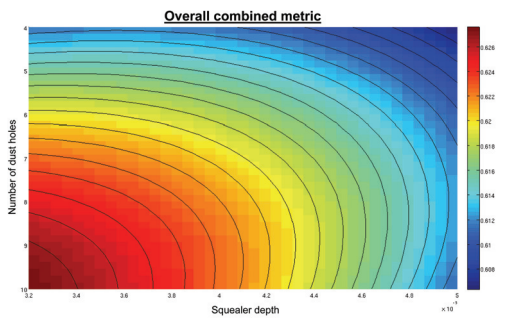


Fig.8d: combined objective function—range 0.608 (blue)-0.626 (red)

of coolant to the trench is visibly better in the case of 10 holes/3.2mm than 4 holes/3.2mm. For deeper trenches, the optimum number of dust holes becomes rather fewer approaching around 8 for a 5mm trench.

Finally, the Response Surface Model was used to predict the objective functions at a new, arbitrarily chosen parameterization and this "low-order model" result compared with the results from a newly generated and meshed geometry solved via the "high order model" FLUENT; this is typical of how a real optimization would proceed in practice. The results in Table 4 in fact show only reasonable agreement between the high order model (CFD) predictions and that derived from the low order model (RSM). This is not really surprising as the RSM is fitted to a very small sample in the first place. However, in the context of managing the exploration of an a priori unknown design space the RSM has suggested a better place to be and this is confirmed by the expensive high order model. Automated design optimization with the opportunity to effect topological geometric change is a large area of research and will be pursued in future publications.

Objective function	Low order model (RSM)	High-order model (CFD)
Film cooling effectiveness	0.372	0.387
Cooled row efficiency	0.884	0.883
Combined objective function	0.628	0.635

Table 4: Predicted performance for a new case with 11 dust holes and a 3mm deep squealer trench.

Concluding Remarks

This paper has described recent extensions to the BOXER paradigm. In particular, it is shown that by adding a scripting capability to the core functionality it is possible to automate the modeling & meshing of complex geometries without fixed topology as the basis for design optimization—driven here by a simple DoE algorithm.

As a scoping study the new approach was applied to the management of the tip heat load on a cooled gas turbine blade. We stress that this study should not be understood as developing an optimum design but as a demonstration that the speed, robustness and flexibility of the BOXER approach is indeed capable of enabling automated design optimization for real, complex geometries.

What was striking was that given the apparent success of the BOXER-based optimization the real limit to automated design improvement has become the parameterization itself. The choice made in this paper was very restrictive—and a more flexible parameterization allowing a much more extensive design space to be accessed could easily have been devised. The next pacing item, therefore, is in the development of optimization algorithms capable of handling many more than a small handful of parameters.

Acknowledgements

We would like to thank our fellow Partners in European Union FP6 programme "AITEB2"—Aerothermal Investigations on Turbine Endwalls and Blading" (FP6 proposal number 516113) led by Dr. Erik Janke for permission to use some of the results presented in this paper.

References

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

[2] Dawes WN "Towards a fully integrated parallel geometry kernel, mesh generator, flow solver & post-processor", 44th AIAA Aerospace Sciences Meeting & Exhibit, 9-12 January 2006, Reno, NV, AIAA-2006-45023

[3] Dawes WN, Harvey SA, Fellows S, Favaretto CF & Vellivelli A "Viscous Layer Meshes from Level Sets on Cartesian Meshes", 45th AIAA Aerospace Sciences Meeting & Exhibit, 8-11 January 2007, Reno, NV, AIAA-2007-0555

[4] Samareh, JA "Status & Future of Geometry Modelling and Grid Generation of Design and Optimisation" J. of Aircraft, Vol 36, no1, Feb 1999

[5] Samareh JA "Survey of shape parameterisation techniques for high-fidelity multi-disciplinary shape optimisation" AIAA Journal Vol.39, No.5, 2001

[6] Haimes R & Follen GL "Computational Analysis Programming Interface" Proc. 6th Int. Conf. On Numerical Grid Generation in Computational Field Simulations, Eds. Cross, Eiseman, Hauser, Soni & Thompson, July 1998.

[7] Galyean TA & Hughes JF "Sculpting: an interactive volumetric modelling technique" ACM Trans., Computer Graphics, vol.25, no.4, pp 267-274, 1991

[8] Perng K-L, Wang W-T, Flanagan M & Ouhyoung M "A real-time 3D virtual sculpting tool based on modified marching cubes", SIGGRAPH, 2001

[9] Baerentzen A "Volume sculpting: intuitive, interactive 3D shape modelling" IMM, May 2001
[10] Bussoletti JE, Johnson FT, Bieterman MB, Hilmes CL, Melvin RG, Young DP & Drela M "TRANAIR: solution-adaptive CFD modelling for complex 3D configurations" AIAA Paper 1995-0451, 1985
[11] Aftosmis MJ, Berger MJ & Melton JE, "Robust and efficient Cartesian mesh generation for component-based geometry" AIAA J., 36, 6, 952-, 1998
[12] Janke E, Hodson HP, Faccini, Popovic I, Lehmann K, Georgakis C, Pons L, Lutum E, Wallin F, Dawes WN & Favaretto F "Selected aerothermal CFD analyses of high pressure turbine topics within the AITEB-2 project", ECCOMAS-2008, 5th European Congress on Computational Methods in Applied Science and Engineering, Venice-Italy June 30-July 5, 2008
[13] Sederberg TW & Parry SR "Free-form deformation of solid geometric models" Computer Graphics Vol.20, No.4, 1986
[14] Krishnababu SK, Newton PJ, Dawes WN, Lock GD, Hodson HP, Hannis J & Whitney C "Aero-thermal investigation of tip leakage flow in axial flow turbines Part III—tip cooling" ASME Turbo Expo Montreal GT2007-27368 (accepted for J of Turbo TURBO-07-1081) 2007