

Contributions Towards the NASA 2030 Vision for Simulation

Bill Dawes¹

Cambridge University Engineering Department, Cambridge UK

Will Kellar, John Verdicchio, Tamas Racz, Nabil Meah, Andrey Kudryavtsev, Rich Evans, Matt Hunt,
Phil Tiller, Carlos Vasco, Lu Yi

Cambridge Flow Solutions Ltd, The Bradfield Centre, Cambridge Science Park, Cambridge CB4 0GA, UK

Kai Liu

BoXeR Solutions KK, Chuo-ku, Kobe, Japan, 651-0087

In the future the aim of simulation will be to develop & improve a whole product, not just separate components individually – a systems-based approach. This will require a complete re-think of current, conventional, simulation process chains encompassing geometry, meshing, CFD/FEA & post-processing. The NASA 2030 Vision¹ study has charted a Roadmap to this new world; the aim of this paper is to set out work we have completed, are performing at the moment & are planning in support of this Vision. Our work makes contributions to geometry, geometry & meshing integration, geometry editing & management, meshing itself, coupling with simulation & post-processing. Examples are given to illustrate our thinking.

I. Introduction

The basics of CFD & FEA were established in the 1970-1990's and have matured to significantly contribute to reduced wind tunnel & physical testing time. However, looking forward, simulations will get bigger and bigger as system & sub-system level models replace component-based design; geometries will become more complex, with ever greater fidelity; the associated meshing processes ever more challenging; simulations will become more coupled as conjugate aero-thermal-mechanical analysis & FSI become the norm; visualisation & extraction of meaningful engineering data will become a “big data” problem. The aim will be to *develop & improve a whole product*, not just separate components individually – a *systems-based* approach. To illustrate this, Figure 1 shows a complete turbocharger simulation, and complete underhood context – and integration with overall product aerodynamics.

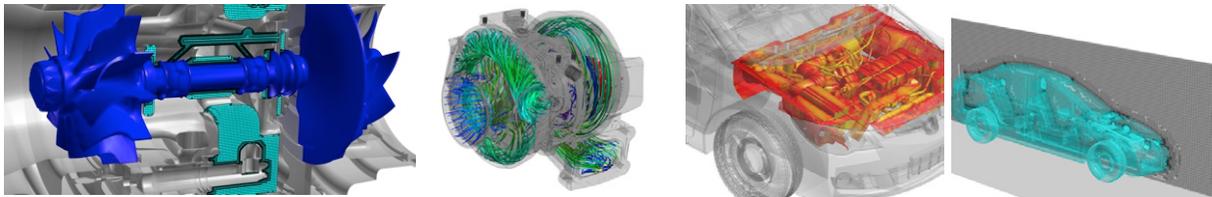


Fig.1: From component to system level simulation: a turbocharger in an underhood environment

Overall simulation requirements for this example can be summarised as:

- Very large mesh : complete engine bay + component CHT mesh + external aero domain will be ~1Bn cells
- “Geometry editing” = very quick and substantial geometry change, without affecting simulation process speed (e.g. “cut-paste” a complete new manifold design, maybe in real time)

¹ AIAA Senior Member

- Integrated tools for simulation, not simply assembly of existing packages via “loose-coupling”
- Data handling (full parallelism) and visualisation to match size & ambition of simulation

The NASA Vision 2030 study¹, in an outstanding act of leadership, has attempted to chart the way ahead. As summarised by Warwick² the study makes clear that “having relied on mature algorithms and ridden the wave of ever-decreasing commodity hardware costs, the CFD community now finds itself poorly positioned to capitalize on the rapidly changing HPC architectures” – HPC hardware “is on the cusp of a paradigm shift in technology that may require rethinking of current CFD algorithms and software”.

Figure 2 shows the Technology Roadmap from the NASA 2030 Vision. The core of any simulation system is the geometry and the meshing systems which deliver that geometry to the multi-disciplinary simulation systems. We have been working for a number of years³⁻⁶ with the aim of developing a scalable, tightly coupled geometry & meshing system capable of dealing efficiently with geometries of arbitrary size and complexity and capable of close coupling with conjugate simulation systems.

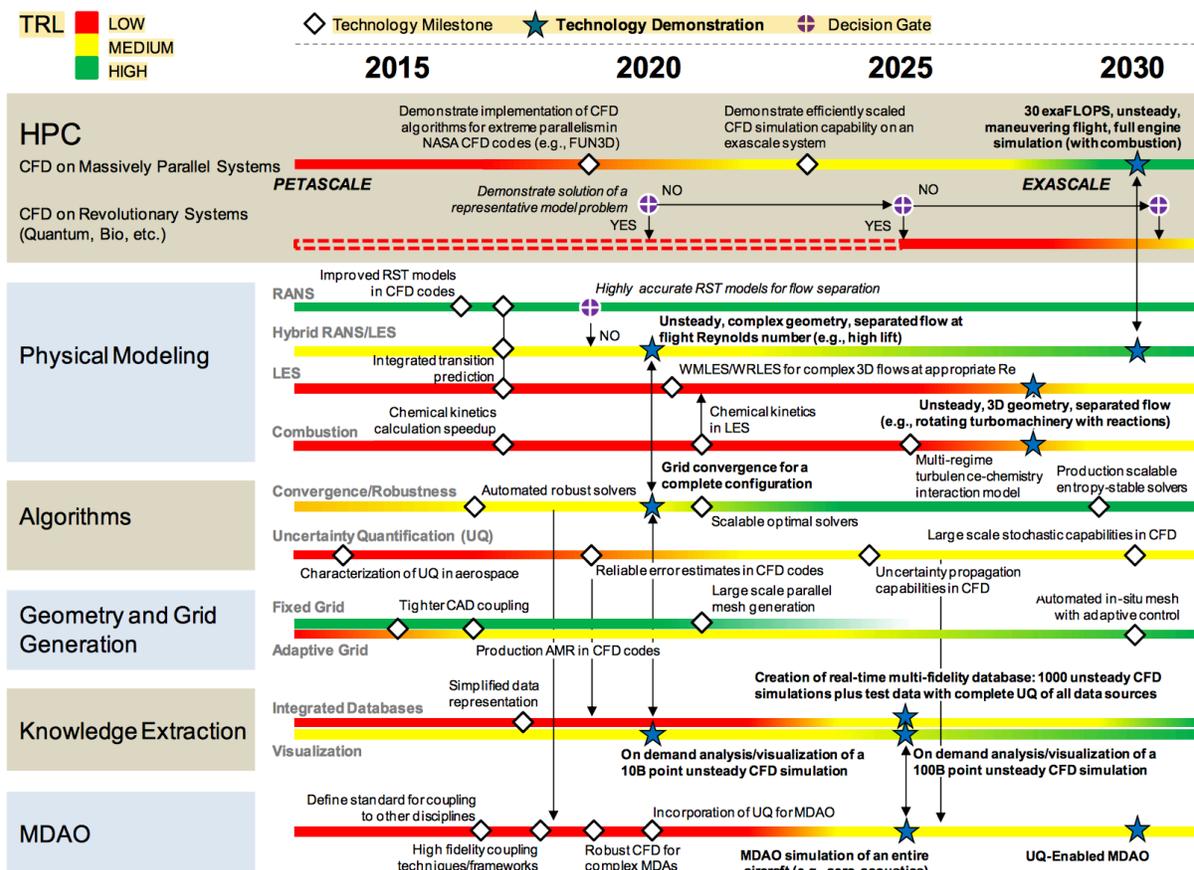
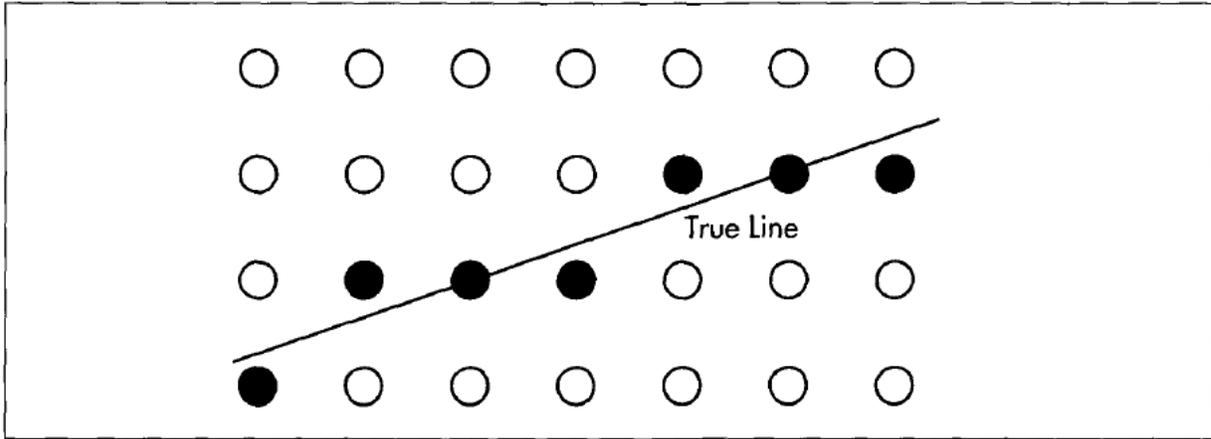


Fig.2: The NASA 2030 Vision Technology Roadmap¹

The aim of this paper is to set out work we have completed, are performing at the moment & are planning in support of the NASA 2030 Vision. Our work makes contributions to geometry, geometry & meshing integration, geometry editing & management, meshing itself, coupling with simulation & post-processing. Examples are given to illustrate our thinking; we call our software system *Boxer*.

II. Digital Geometry Model

The famous Bresenham line algorithm (1962) was developed as a way of representing a line via discrete pixels – “rasterisation” on the newly emerging Cathode Ray Tube terminals. As Figure 2 illustrates⁷, the closest pixels to the line are illuminated. This is essentially the core idea in digital photography – a picture – in 3D this becomes *geometry*.



Approximating a true line from a pixel array.

Fig.3: The Bresenham Line Algorithm (1962)⁷

Our *Boxer*⁸ software is built on Digital Geometry using generalised³ 3D versions of the fundamental Bresenham algorithm; Figure 4 illustrates this. 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.

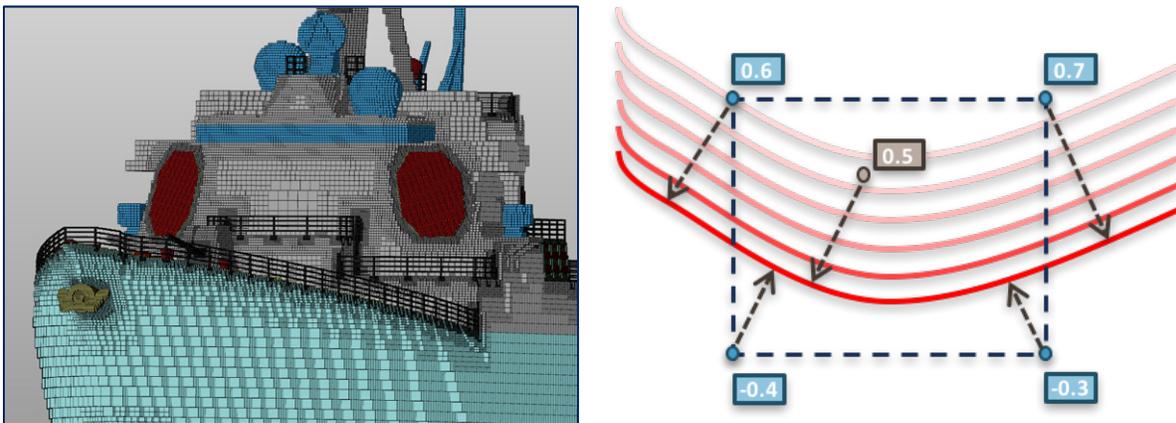


Fig.4: The Digital Geometry Kernel in *Boxer*; on the left the 3D voxel image; on the right the Distance Field storing sub-voxel scale geometry information

There are two key advantages of this approach:

- Digital Geometry can be distributed onto any cluster - enables true parallel scalability
- Geometry editing & management is supported in a very general, topology-independent way

Looking ahead at the Simulation System of the Future, geometry will need to be available throughout the 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 modeling itself must be capable of being implemented & scaling in parallel – this is trivial for our Digital Geometry kernel but very difficult to imagine with a kernel based on traditional NURBS/BREP constructs.

An engineer presented the idea for a "filmless camera" to Kodak executives in 1975, but was laughed out of the room⁹. In 2012 Kodak declared bankruptcy, having failed to adapt to the digital world. Leaving behind analogue geometry & meshing and moving on to the digital world was referred to by Chawner *et al*¹⁰ as a potential "Kodak moment".

III. Scalable Meshing Integrated with the Geometry

We have developed a meshing process, tightly coupled to this Digital Geometry model. The stages in the meshing process are sketched in Figure 5. Stage 1 represents the digital capture of the geometry; Stages 2-4 develop a body-fitted mesh; Stages 5-6 add layer mesh cells. All six stages are implemented in parallel and each is coupled in parallel to the Digital Geometry model to support activities like mesh adaption in a fully scalable way.

The meshing process can be viewed as a "volume-to-surface" approach that avoids the prior generation of a surface mesh by "capturing" and "re-constructing" the geometry as the volume mesh is generated. The surface mesh (of triangles & quads) is an *output* from the process – in effect the volume mesh *predicts* where the surface mesh must lay to permit good volume cell quality. This is especially useful in small gaps (like airfoil to flap/slat gaps) which can be difficult to volume mesh with more conventional meshing approaches if the previously generated surface mesh cell size is not similar either side of the gap.

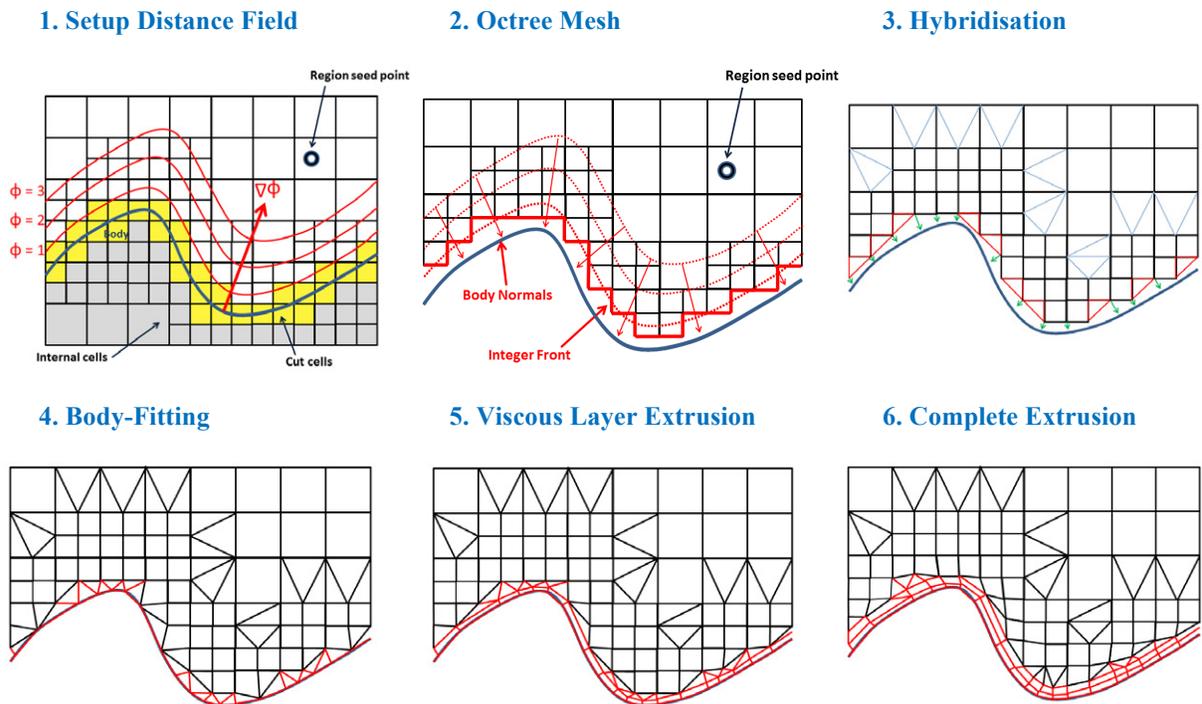


Fig.5: The stages in the meshing process: Stage 1 represents the digital capture of the geometry; Stages 2-4 a body-fitted mesh; Stages 5-6 add layer mesh cells

As an *example*, Figure 6 shows detail views of the *Boxer* surface mesh for the NASA HL-CRM geometry.

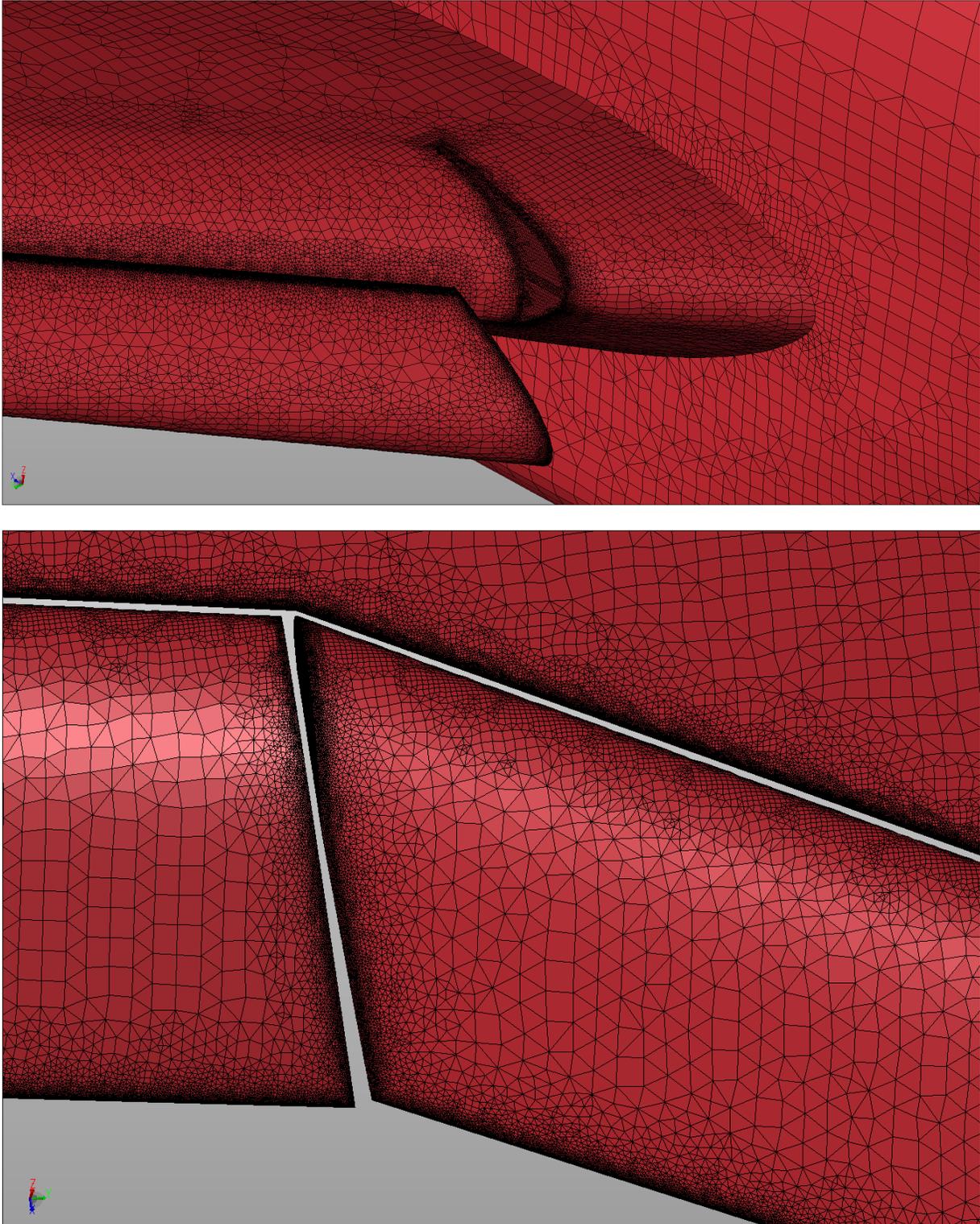


Fig.6: Detail views of the *Boxer* surface mesh from the NASA HL-CRM geometry

A second *example* shown in Figure 7 is based on a HP turbine rotor including cooling air system, shroud and under-hub. The mesh is for CHT/multidisciplinary simulation; multiple fluid & solid meshes are produced automatically from a single meshing template – a fully conformal fluid-solid mesh interface enables straightforward coupling of CHT models.

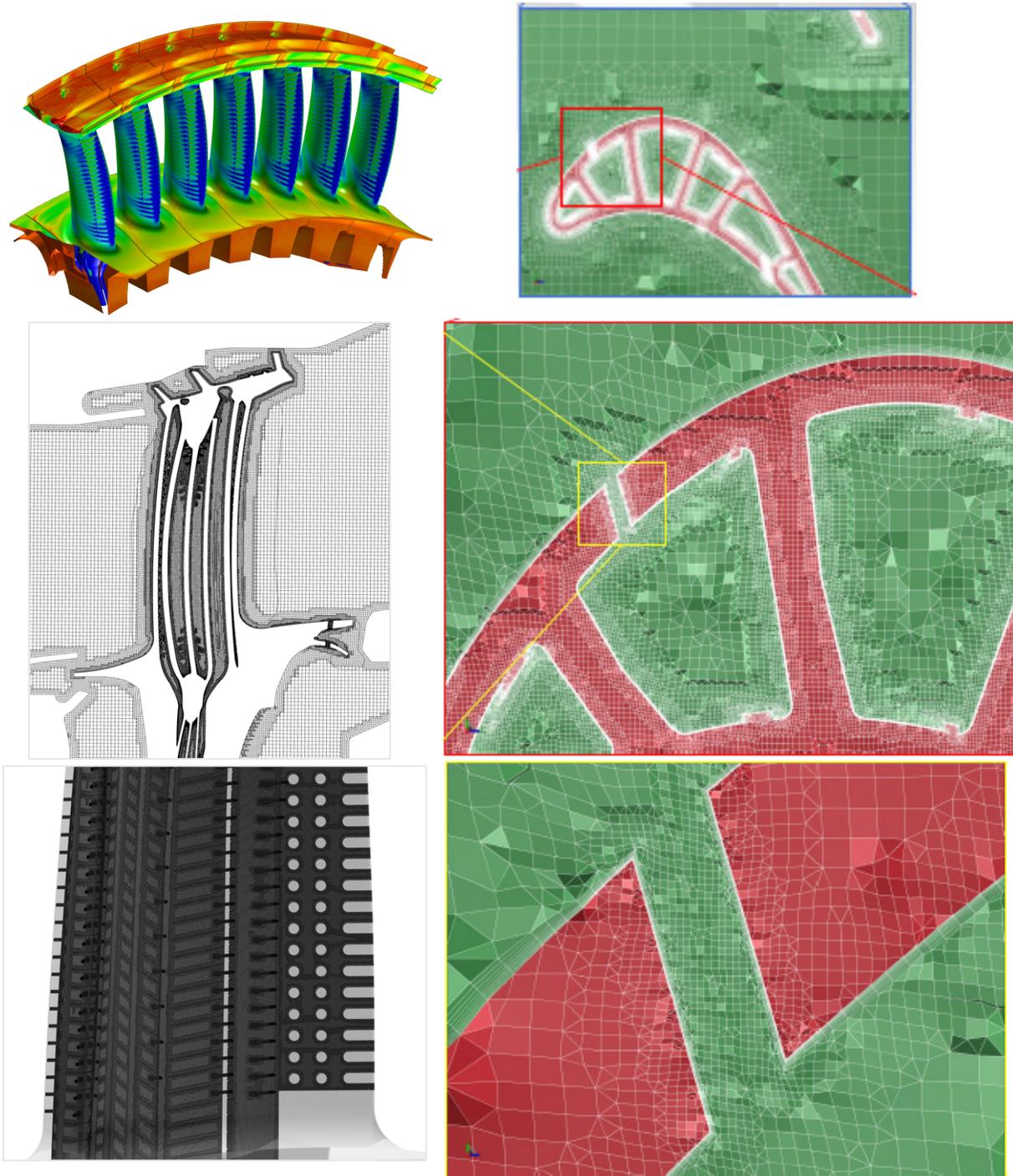


Fig. 7: A conjugate mesh for a CHT simulation of a cooled gas turbine blade

The underlying philosophy of *Boxer* is to generate a mesh that is always runnable, through a process that is entirely dependable and will take a predictable amount of time, rather than being open-ended. A number of significant benefits arise from the present approach. Some of these benefits stem from the overall meshing procedure, some from detailed choices at an algorithmic level and others simply from the way in which the software has been written and implemented. Together, the architecture and parallelization allow the software to run on the widest range of hardware platforms, literally from a laptop to an HPC cluster, whilst delivering excellent performance scalability. Users have demonstrated performance retention in meshes exceeding 1Bn cells.

We have also developed libraries which can elevate basic P1 (second order) *Boxer* meshes to Higher Order P2 (third) & P3 (fourth order); this needs knowledge of the underlying geometry. Figure 8 shows for the LARC trap-wing case a relatively coarse P1 mesh resolution (too coarse probably for decent CFD), but nevertheless the underlying digital geometry model is a good match to the shape. High-order element conversion elevates the mesh to P2 (third order) as shown in the detail view. We support higher order tetrahedrons, pyramids, prisms & hexahedra all up to P3.

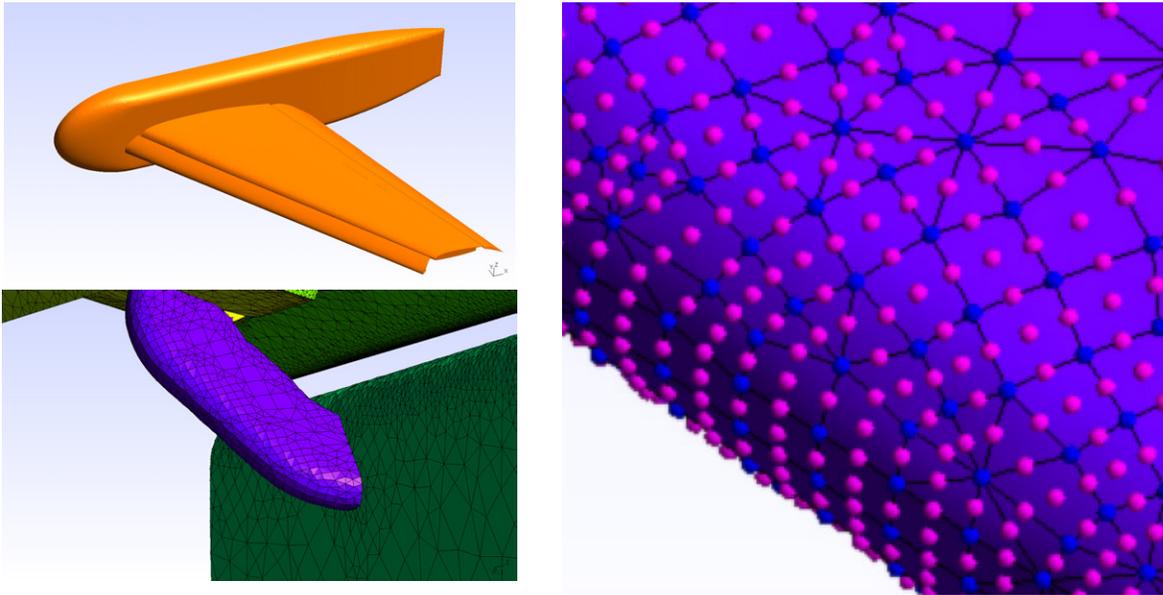


Fig.8: Elevation of a P1 (second order) mesh on the LARC trap-wing (left) to a Higher Order P2 (third order) mesh (right, detail)

IV. Geometry Editing & Management

Continuing in the theme of exploring approaches & technology which can be used in an end-to-end integrated parallel simulation environment we have been experimenting¹¹ with Boolean summation of geometry parts & Free Form Deformation of geometry; both are very easy to implement using our Digital Geometry kernel – and both classes of operation can be implemented in parallel and scale naturally with the geometry sizes and meshes.

By way of illustration, Figure 9 shows first using Boolean subtraction to create cylindrical cooling holes in a gas turbine blade followed a three-parameter *Inverse* FFD to create fan shaped hole exits. In *Inverse* mode a few (three here) parameters are imposed (red, green & blue in the Figure); these few parameters can exert a lot of influence over the design space allowing extensive exploration for new designs. The FFD tri-cubic spline is used to blend the effects of these few parameters smoothly across all the hole. The alternative (which is the more usual in other implementations) is *forward* mode when every single tri-cubic spline control point needs attention leading to far too many parameters for effective design space exploration.

As an extra aid to automated optimization we construct (and give scripted control of to the user) one *super-parameter* to control *all* the cooling holes. All of the other FFD boxes for the other holes are slightly different –

aligned to their own local reference frames – this is just a mapping - this enables the local shape deformations e.g. “make hole wider” to be applied relative to each of the individual FFD boxes’ reference frames

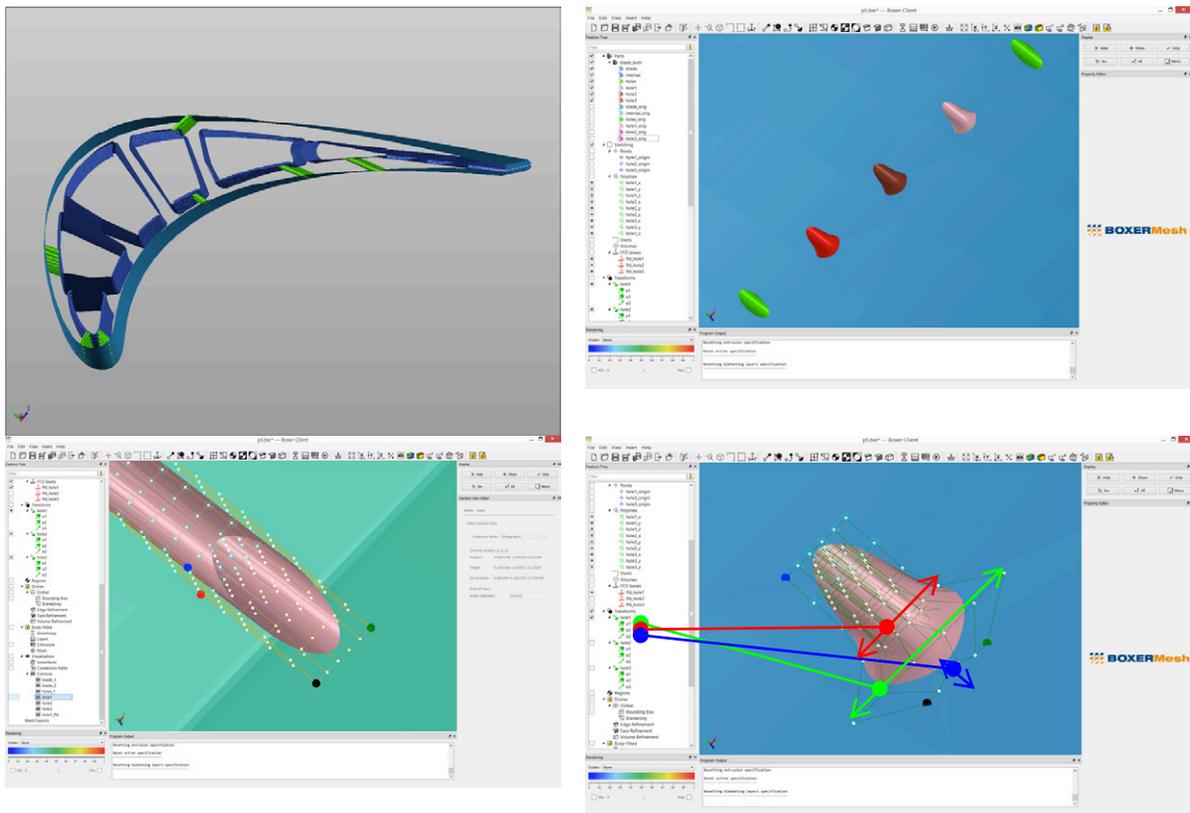


Fig.9: Using Boolean subtraction to create cylindrical cooling holes in a gas turbine blade; a three-parameter inverse FFD creates fan shapes; one *super-parameter* controls all

We are also exploring Morphing geometries by modifying the Distance Field which supports our Digital Geometry kernel. Very much inspired by the exciting work of Breen & Whitacker¹² we have been implementing within *Boxer* all the infrastructure for mesh inputs & mesh deformation in accordance with a derived Surface Deformation Field. The Surface Deformation Field can be driven by a whole range of physics-based or heuristic algorithms (see for example Adalsteinsson & Sethian¹³) – including adjoint-style information from an associated FEA or CFD simulation.

Our current work is directed at establishing adjoint-style geometry perturbations from some shape A to some other shape B, via the digital level-set geometry model expressed on background octree mesh(-es). Naturally, in the spirit of NASA 2030, we have been implementing this methodology into the *Boxer* environment in a scalable, parallel manner; the morphing algorithm itself looks very much like a flow evolution equation but acting on the Distance Field^{12,13} and is trivial to implement in parallel. Figure 10 sketches the basic infrastructure for mesh inputs & mesh deformation within *Boxer*.

In terms of design space exploration the intention and guiding idea is to be able to breed useful new but unexpected candidate designs from combinations of existing ones (here an Italian and American car...) very much inspired by the work of Nishino *et al*¹⁴ (see also Baerentzen¹⁵).

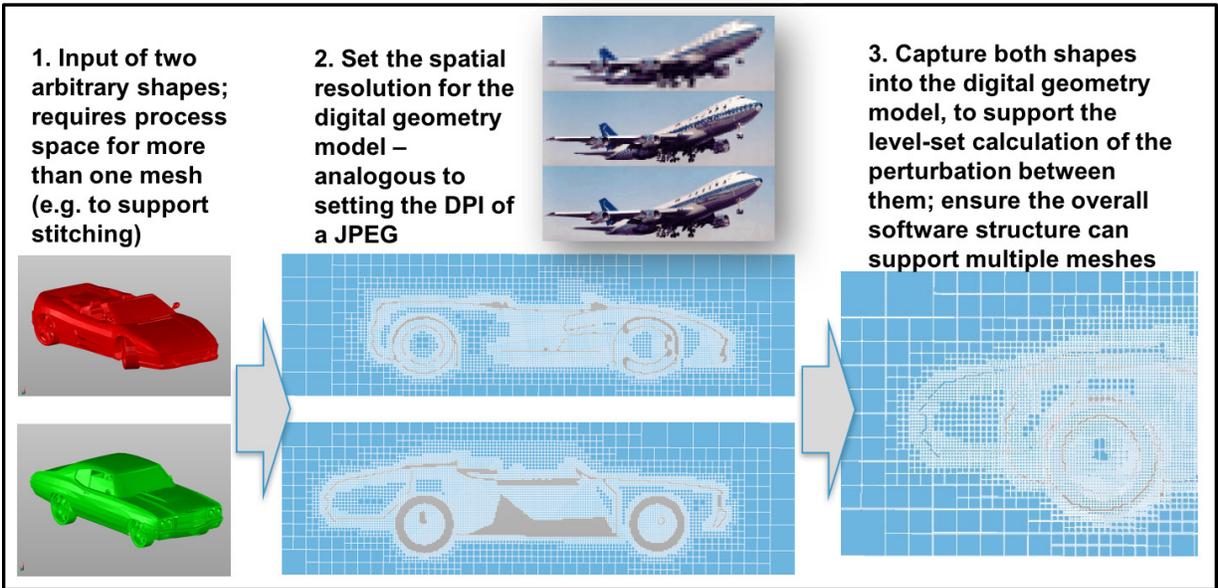


Fig.10: The basic infrastructure for mesh inputs & mesh deformation within *Boxer*

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). Accordingly we have implemented a simple, automated *morph-mesh-solve* workflow. A simple result of this process is illustrated in Figure 11: given a sphere (source) and a cube (target) the automated workflow computes first the level set morphing operation, and saves the intermediate shapes, next, the meshes are generated, and finally a flow solver is used to compute the drag coefficient. We used here the flow solver Fluent™ (version 18.1) for a Reynolds number of ~ 30000 with as turbulence model $k - \omega$. 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.

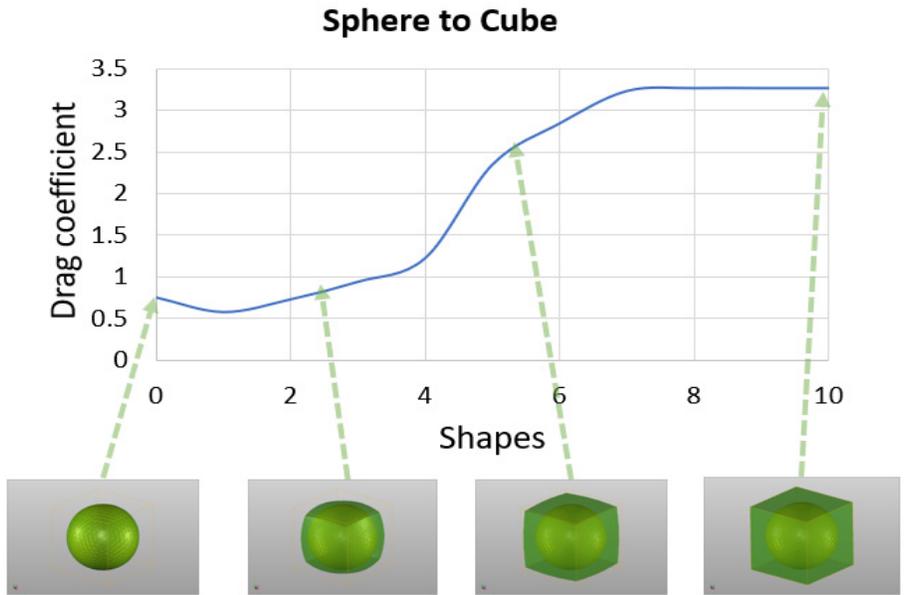


Fig.11: Drag coefficient for a sphere morphing to a cube

It is worth noting that the shape minimising the drag coefficient is not quite a sphere (according to RANS anyway) but one that strongly resembles a sphere 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.

It follows that this two-shape morphing is a building block for more complex operations, such as a multi-objective optimisation problem. We restrict the following analysis to three shapes in total; using the concept of *micro-Genetic Algorithm*, a powerful, yet simple, optimisation technique can be developed which enables the automated workflow to populate the design space progressively. Following the specification of *three* 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 choice of the source/target shapes, or automatically.

Figure 12 illustrates this process. 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. As any genetic algorithm the termination criterion is not easy to define. Our method very rapidly yields a solution, and the user can pursue the exploration of the space by selecting shapes far in the design space from the best one to bring diversity in the population, hence increasing the chance to reveal better candidates. Other advantages of such optimisation technique include its robustness since the results do not vary significantly between each run.

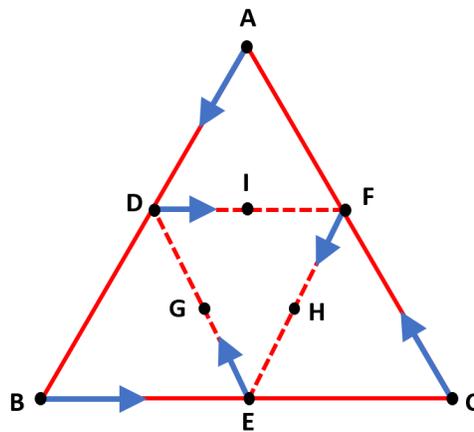


Fig.12: Design space for three shapes morphing under the management of a Genetic Algorithm

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; the objective here is not really to produce a new design but to show the potential of the present morph-solve workflow to allow rich & interesting new design spaces to be created and explored.

Figure 13 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.

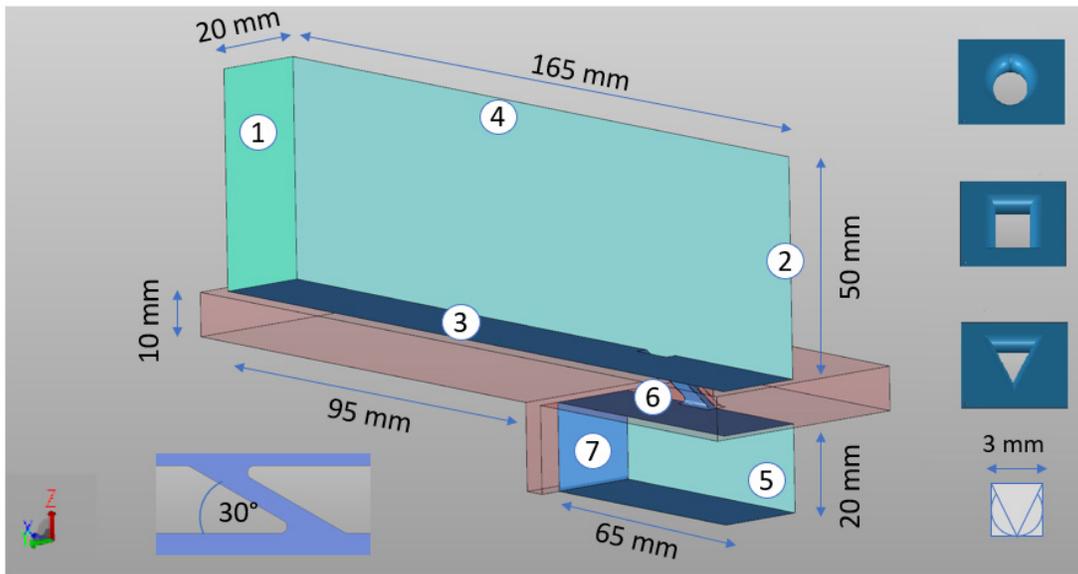


Fig.13: Dimensions of the domain, with the part of the geometry which is morphed (red), and 1 – Pressure outlet, 2 - Cross flow inlet (stagnation inlet), 3 – Test section (adiabatic), 5 – Plenum inlet, 6 – Film cooling hole (adiabatic walls, half model); the three basic seed shapes are circular, square & triangular and are shown on the right hand side with their relative position.

A typical mesh, for the square cooling hole, is shown in Figure 14, 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.

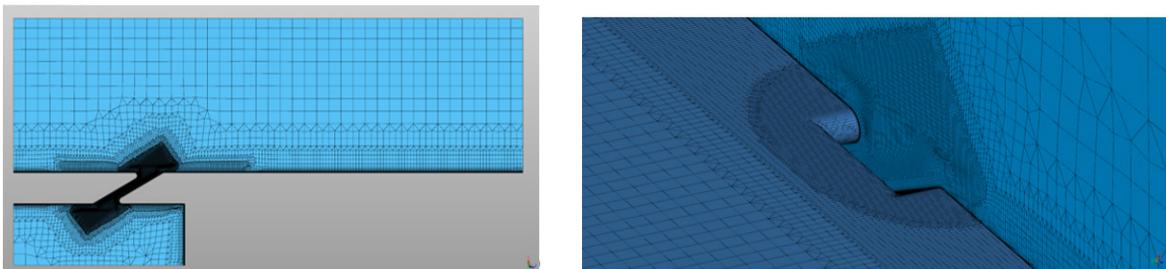


Fig.14: Mesh used for the rectangular cooling hole, (top: isosurface of the full system, and bottom: zoom on the hole, with an isosurface at the hole centre plane)

Simulations were run with the Fluent™ flow solver used in RANS mode with the turbulent model $k - \omega$ SST employed and with hot primary flow with boundary conditions summarised below in Table 1.

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:

$$\eta = \frac{T_{cross\ flow,inlet} - T_w}{T_{cross\ flow,inlet} - T_{plenum,inlet}}$$

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

Table 1. Boundary conditions

The cooling efficiency averaged over the test surface area is plotted vs the average mass flow rate in Figure 15 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.

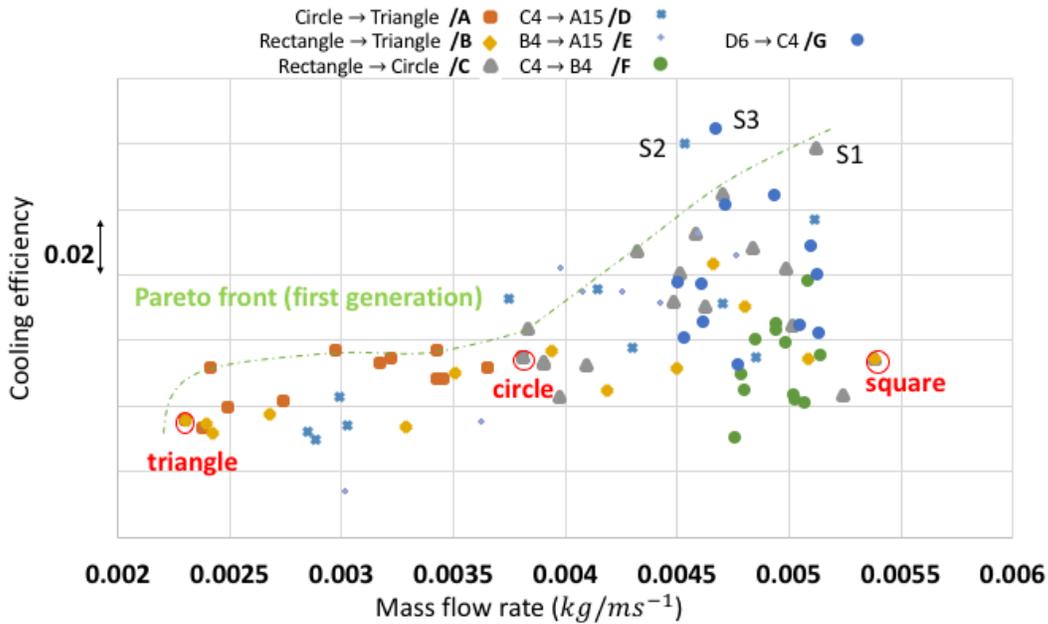


Fig.15: Average cooling efficiency vs average mass flow rate for all the shapes generated

V. Coupling with Simulation

In *loose* coupling, the current norm, all data transfer between geometry, mesher, FEA/CFD etc. is in serial, scripted. As an example of this we show a loose coupled simulation of a bicycle, in scripted 6DOF motion, within a velodrome in the vicinity of Mount Fuji. This is not at all a fanciful simulation; at the highest levels of competition the conditions within the velodrome, and the interactions with the cyclists, are known to be affected by the overall environment of the velodrome and the winds/weather of the surroundings.

First of all the geometry must be assembled; the terrain comes from satellite data, the velodrome and bicycle from different CAD models: see Figure 16. Next meshes are generated as illustrated in Figure 17; our approach is very

efficient at handling these very large ranges of scale. Flow simulation then follows (we used Fluent™ in RANS mode) and Figure 18 shows both the general flow in the velodrome itself and the detail flow near to the bicycle.

Finally, Figure 19 illustrates a basic overtaking manoeuvre...

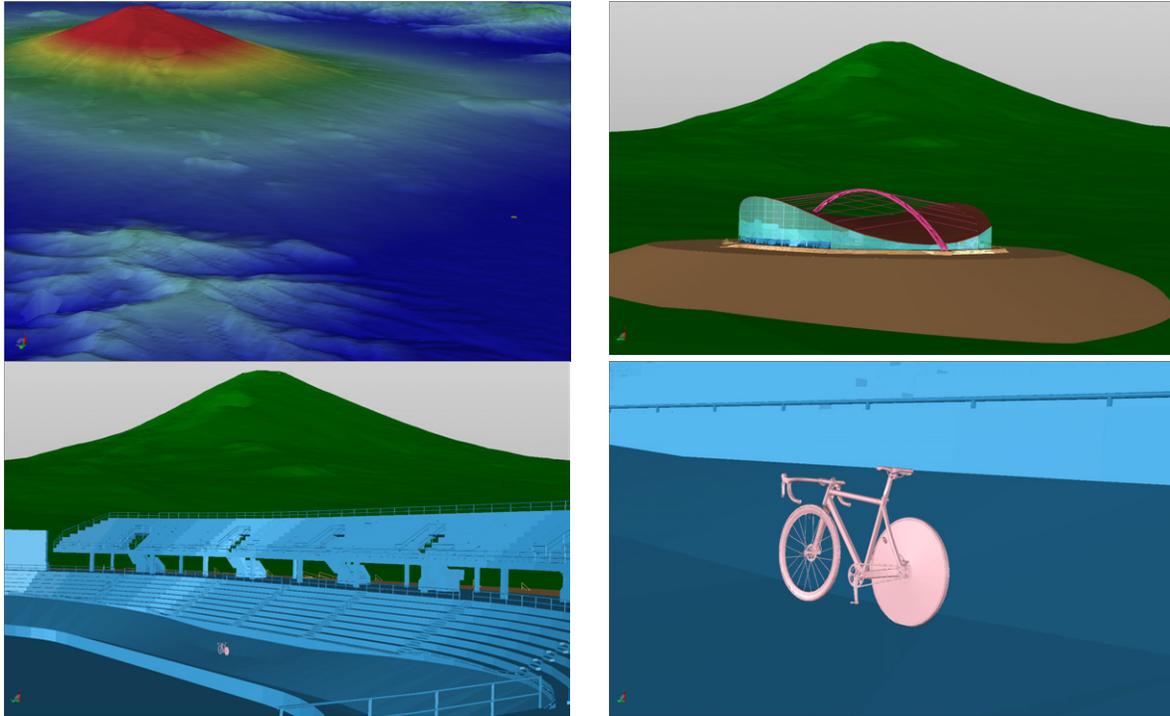


Fig.16: Assembling the geometry

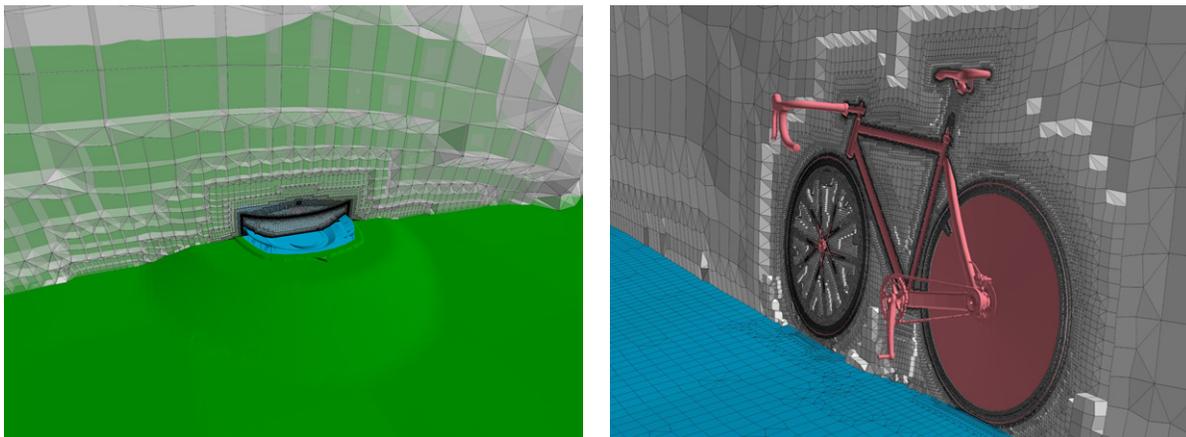


Fig.17: Meshes

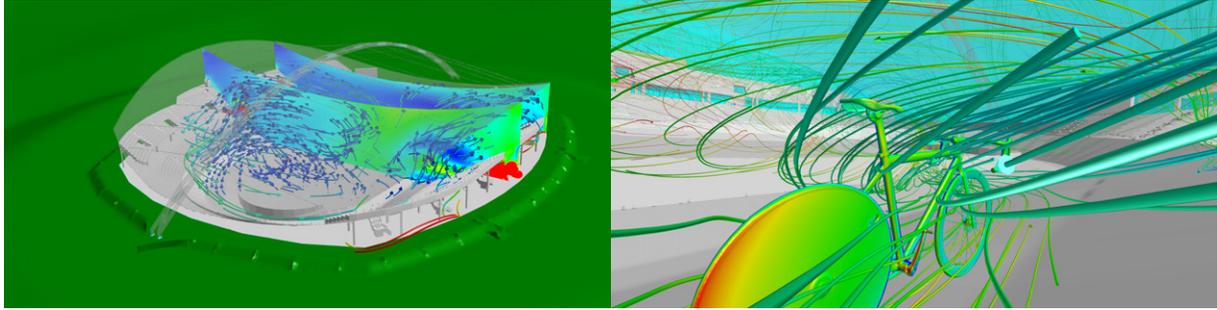


Fig.18: RANS flow simulation on the scale of the velodrome and the bicycle.

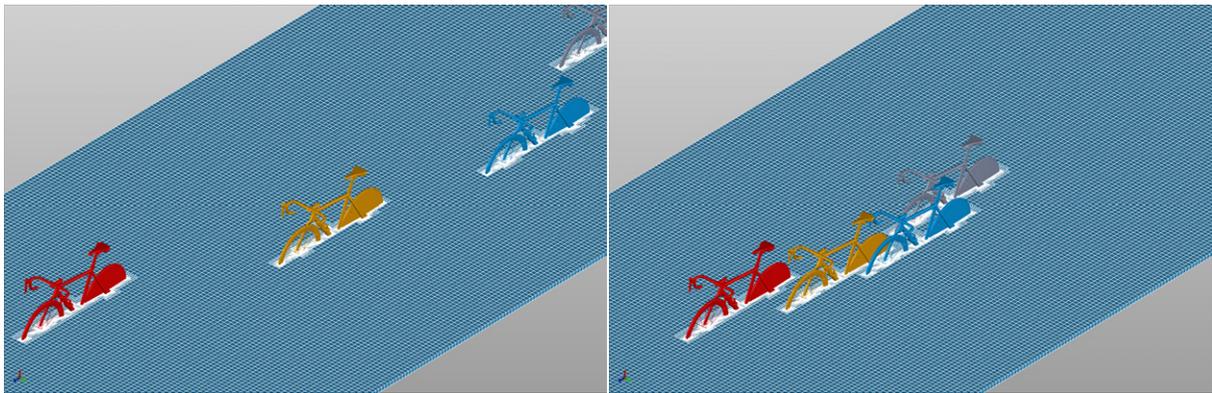


Fig.19: A scripted 6DOF overtaking manoeuvre.

In terms of resource, this is a very efficient and economic simulation consuming as follows:

- Time to assemble the geometry & set up the mesh 0.5 days; time to mesh 300 minutes ~100 million cells
- Time to flow solve ~500 minutes on 60 parallel cores

For very large datasets and coupled simulations like CHT or FSI – this data transfer becomes a serious, serial bottleneck; going forward in the spirit of NASA 2030, we must plan for data to be exchanged in parallel and the *Boxer* Environment is designed to support this.

For *close* coupling (eg. for quasi-steady conjugate simulations) *Boxer* reaches out to other applications via an API giving access to shared, distributed memory – we call this “BMF” – and it is based on HDF5, the emerging standard for parallel coupling. HDF5 implements cross-platform parallel I/O and presents a single file as a directory-like structure & permits concurrent views by multiple processes. The first release of HDF5 was ~17 years ago and since then has seen increasing usage. For us this is work-in-progress and the intention is that BMF will become a CFS-provided source code library allowing eg., N _meshes to couple with M _solves in core in parallel.

For *strong* coupling (eg. for FSI with reduced frequencies of $O(1)$ or for flutter) even an HDF5-type connectivity is likely to be too restrictive on conjugate meshes numbering in the Billions. It seems likely to us that the simulation tools (FEA, CFD,...) must be directly integrated with the mesher and geometry kernel end-to-end in parallel sharing the same in-core data structures. We are exploring this with our own rather simple but very fast RANS solver *NEWT*. The first version of *NEWT* was developed nearly 20 years ago, Dawes¹⁶; the basic solver was a very simple finite volume Runge-Kutta time marcher with k - ϵ turbulence modelling – but very fast, robust and accurate. The code was highly validated, mostly for turbomachinery applications but also in the process industry and motor sport. In recent years *NEWT* has been brought up to date and extended – in particular to run on *Boxer* hybrid meshes and to low Mach numbers and incompressible fluids.

VI. Post-processing & Visualisation

Hand-in-hand with advances in simulation technology goes the post-processing of the massive “big data” unsteady results. This brings the huge challenge of storage and computing cost for practical I/O operation. There are two types of basic post-processing requirement for unsteady data: one is the transient data extraction and recording such as flow quantities on probe points, surface and iso-surface data-extraction; another is the analysis of unsteady volume data such as unsteady flow structures and their coherent connections.

For the former it seems self-evident that the post-processing & visualisation algorithms must be implemented in parallel and closely/strongly coupled into the solver in-core data structures; the VisIt software for In Situ visualisation¹⁷ seems a promising way to go here. This in turn needs to be tightly coupled back to the geometry and mesher for solution adaptive mesh refinement. This would complete an *end-to-end parallel coupled simulation system*.

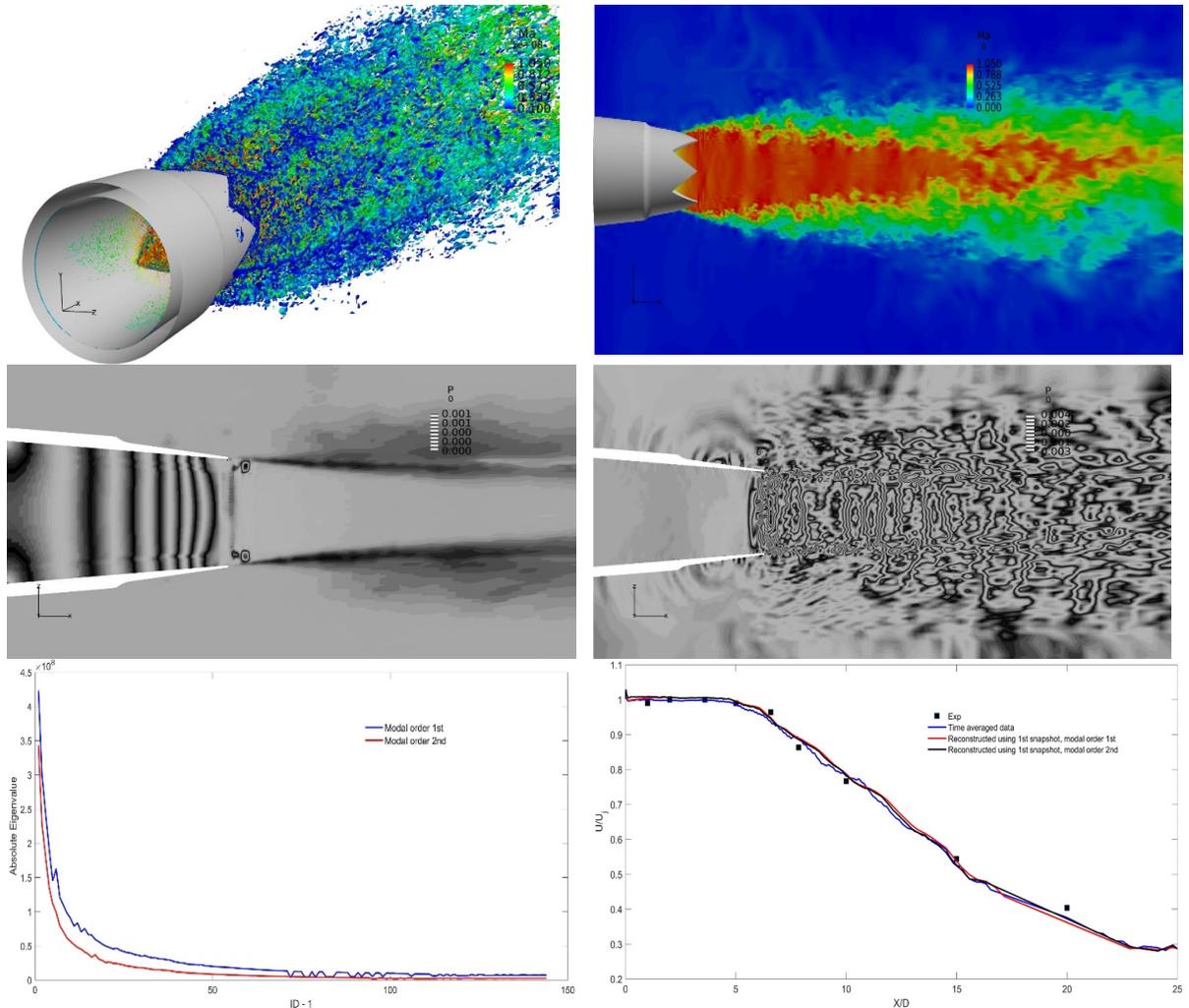


Fig.20: Instantaneous Q-criterion, instantaneous Mach number and 1st & 9th POD modes from an LES simulation of the NASA Acoustic Reference Nozzle; bottom left shows a plot of Eigenvalues(not including 1st mode) from the HPOD analysis; bottom right shows a comparison of reconstructed, time-averaged centreline u-velocity using the 1st POD mode.

For the post-processing analysis of unsteady volume data, we have proposed a new Hierarchical Proper Orthogonal Decomposition (HPOD) method¹⁸ with two main objectives: one is to perform reduced order analysis for large scale high order simulations on limited computing resource, for the analysis of flow mechanisms and extraction of industrially interesting information; another is to explore the relations between different frequency parts of flow field,

aiming to construct a multi-level local filter, which could help to reduce computing cost dramatically for high fidelity simulations. In HPOD, multi-level solutions in orthogonal modal space are extracted *on-the-fly* during the high order large eddy simulations, the memory requirement and computing cost of POD analysis on low order modal solutions are much less than the analysis of original nodal high order solutions, which makes the POD analysis affordable on modest computing resource even for very large scale simulations. These lower order modal solutions could be regarded as the filtered low frequency part of the flow field, which contains the main flow information of most interest to industry, whose connection with higher order parts of the complete modal solutions can be investigated as part of the POD analysis. Figure 20 illustrates this with some results from a LES simulation we performed¹⁸ on a standard nozzle. It can be seen that maybe only 50 eigenmodes are needed to reconstruct the flow – a potentially enormous saving in resource and wall-clock time.

VII. Conclusions

The objective of this paper was to describe contributions we are trying to make towards the NASA 2030 Vision for the Simulation Systems of the Future. The paper has described contributions in five main areas:

Digital Geometry: there are two key advantages of our approach: Digital Geometry can be distributed onto any cluster - enables true parallel scalability; geometry editing & management is supported in a very general, topology-independent way; geometry will need to be available throughout the simulation process chain to support solution adaptive meshes, FSI and automated design optimisation – it follows therefore that it is essential to adopt a geometry kernel which can be implemented in parallel.

Meshing: The underlying philosophy of *Boxer* is to generate a mesh that is always runnable, through a process that is entirely dependable and will take a predictable amount of time, rather than being open-ended; for this to be implemented in parallel and hence scalable to geometries of arbitrary size & complexity – and to be tightly integrated with the Digital Geometry kernel to support mesh adaption & FSI.

Geometry editing & Management: continuing in the theme of exploring approaches & technology which can be used in an end-to-end integrated parallel simulation environment, we have been experimenting with Boolean summation of geometry parts, Free Form Deformation and Morphing of geometry; all are very easy to implement using our Digital Geometry kernel – and all classes of operation can be implemented in parallel and scale naturally with the geometry sizes and meshes.

Simulation itself: our ambitious overall objective is RANS closely coupled with geometry and meshing so that candidate designs can be evaluated and geometries changed in real-time...

Post-processing: post-processing & visualisation algorithms must be implemented in parallel and close coupled into the solver in-core data structures; this in turn needs to be close-coupled back to the geometry and mesher for solution adaptive mesh refinement; in this paper, for the post-processing analysis of unsteady volume data, we have outlined an additional approach based on a new Hierarchical Proper Orthogonal Decomposition (HPOD).

Acknowledgements

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

References

- ¹Slotnick J *et al* “CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences” NASA/CR-2014-218178
- ²Warwick G “Computing Crunch” Aviation Week & Space Technology, p18, May 19 2014
- ³Dawes WN “Building Blocks Towards VR-Based Flow Sculpting” 43rd AIAA Aerospace Sciences Meeting & Exhibit, 10-13 January 2005, Reno, NV, AIAA-2005-1156
- ⁴Dawes 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

- ⁵Dawes 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
- ⁶Demargne AAJ, Evans ROE, Tiller PJ & Dawes WN " Practical and Reliable Mesh Generation for Complex, Real-world Geometries" AIAA-2014_0199
- ⁷www.nondot.org/sabre/Mirrored/.../gpbb35.pdf Chapter 35: Bresenham is Fast and Fast is Good
- ⁸www.cambridgeflowsolutions.com
- ⁹<http://www.telegraph.co.uk/technology/2016/05/13/biggest-mistakes-in-tech-history/>
- ¹⁰Chawner JR et al "The Path to and State of Geometry and Meshing in 2030": Summary" AIAA 2015-3409
- ¹¹Dawes WN, Kellar WP, Harvey SA "Towards topology-free optimisation: an application to turbine internal cooling geometries" 46th AIAA Aerospace Sciences Meeting & Exhibit, 7-10 January 2008, Reno, NV, AIAA-2008-0925
- ¹²Breen DE & Whitacker RT "A Level-Set approach for the metamorphosis of solid models" IEEE Transactions in Visualisation & Computer Graphics, vol.7, no.2, April-June 2001
- ¹³Adalsteinsson D & Sethian JA, "A level set approach to a unified model for etching, deposition & lithography II: three dimensional simulations" J.Comput.Phys, 122, pp348-366, 1995
- ¹⁴Nishino H, Takagi H, Cho S-B & Utsumiya K "A 3D modelling system for creative design" 15th International Conference on Information Networking, ICOIN-15, Beppu, Japan, pp 479-, 2001
- ¹⁵Baerentzen A, "Volume sculpting: intuitive, interactive 3D shape modelling" IMM, May 2001
- ¹⁶Dawes WN "Simulating unsteady turbomachinery flows on unstructured meshes which adapt both in time and space", in *ASME International Congress on Gas Turbine and Aeroengine* (pp. 12 pages). New York, USA: American Society of Mechanical Engineers, 1993
- ¹⁷Whitlock B *et al* "In Situ visualisation using VisIt" Scientific Computing SC11, 2011
- ¹⁸LuYu, Kai Liu & Dawes WN "Efficient and affordable high order, high fidelity Large Eddy Simulations for industrial level problems" SciTech AIAA-2017-xxxx

-oOo-