Interactive Visualisation of Multi-Fidelity Design Optimisation

Graham Pullan*, Aljaz Kotnik†
Whittle Laboratory, Department of Engineering, University of Cambridge, Cambridge, UK

Marcus Meyer‡
Rolls-Royce Deutschland Ltd, Eschenweg 11, 15827 Blankenfelde-Mahlow,Germany

The modern industrial aerodynamic design process involves a large number of simulations, spanning a multi-dimensional design space. If an automated optimisation process is employed, hundreds of simulations can easily be obtained as the set of input parameters is refined. In this paper, we address the challenge of visualising and interpreting these computations. By emphasising interactivity as a core principle, the design process can be navigated and filtered at a high level, while subsets of simulations are interrogated in detail. Using a client-server web-based approach, statistics (for example, response surfaces and rank correlations) react dynamically to changes in the filter settings applied to the database of cases, allowing the engineer to develop an understanding of the connections between input parameters and the flow field. As optimal design candidates are determined, it is common to employ more expensive, higher fidelity, computations. We show how unsteady computations can also be interrogated in the same visualisation framework. In particular, the ability to interactively change time level, across multiple views of both three-dimensional objects and one-dimensional line data, is demonstrated. The goal is to reduce the friction and latency associated with asking questions of the simulation data so that engineers can more rapidly evaluate designs.

I. Introduction

The repeated use of computational simulations is a key component of the aerodynamic design process. An important output of computational fluid dynamics has always been the identification of flow field and performance metric trends as input parameters (geometry or boundary conditions) are varied. Modern compute hardware allows multi-dimensional design spaces to be sampled and evaluated quickly, for example using a Design of Experiments strategy. The result is the routine generation of 100s of simulations. Software tools are needed to support engineers in their decision making based on this wealth of data[1–3].

During the evaluation of a design, the engineer may want to both refine the optimal choice of input parameters, and also employ higher fidelity simulations to evaluate viable candidates with increased accuracy. These additional computations could be higher resolution steady Reynolds-Averaged Navier-Stokes (RANS), or unsteady RANS, or Large Eddy or even Direct Numerical Simulation. These higher fidelity simulations are best evaluated within the context of the larger number of exploratory (lower fidelity) simulations that comprise their ancestry in the design process.

The goal of this paper is to contribute to the tools needed to allow engineers to visualise and interpret the large number of simulations, of differing fidelities, that arise from the modern aerodynamic design optimisation process. We stress the importance of an interactive approach that enables the engineer to obtain answers to questions as they arise. By reducing the 'friction' in the navigation and visualisation of point, line and surface data, the engineers can better identify connections between input parameters and output metrics, and ultimately define improved design concepts.

II. Interactive web-based visualisation

A. Overall concept

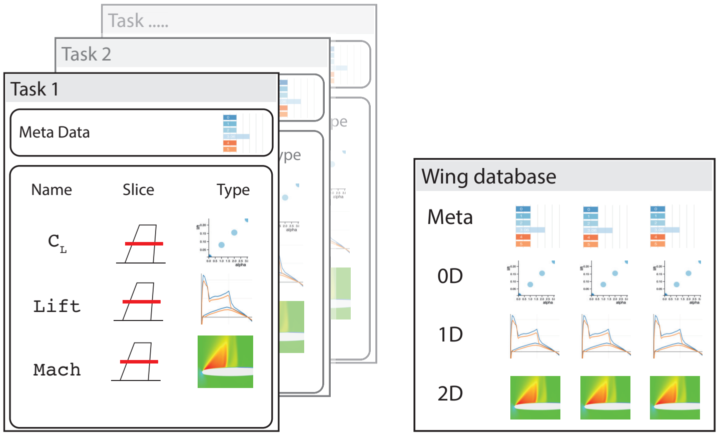
A natural abstraction is the client-server model, Fig. 1, where the server holds the repository of simulation results and the client displays the visualisation. By implementing the client-side software as a web browser application, we benefit in two ways. First, the web-based visualisation platform is inherently portable across all devices (desktop, laptop,

*Professor of Computational Aerothermal Design, AIAA Senior Member

†Research student

‡Specialist, CFD Methods

tablet and phone). Second, we are able to leverage the ecosystem of libraries that support graphics and interactive visualisation in the browser[4, 5].



(a) Server database

(b) Client interface

Fig. 1 Data is arranged by simulation case, "Task", on the server. On the client, Tasks are selected, using their Meta Data, and then examined in detail via comparative plots. From [6]

A core assumption is that the data is hierarchical with at least two levels, Fig. 1(a). Each simulation has a row of data in a high-level 'metadata' table that typically includes input parameters, and key output parameters, for each computation. At the detailed level, the server stores a collection of line and surface data for each simulation. This data is expected to be pre-computed, but could also be extracted on-the-fly by the server from the full simulation output. Interactivity is enhanced by transferring all of the metadata at the start of the visualisation session so that subsequent filtering is unimpeded by data transfer; detailed data is then only transferred for the subset of simulation of interest at any one time, Fig. 1(b).

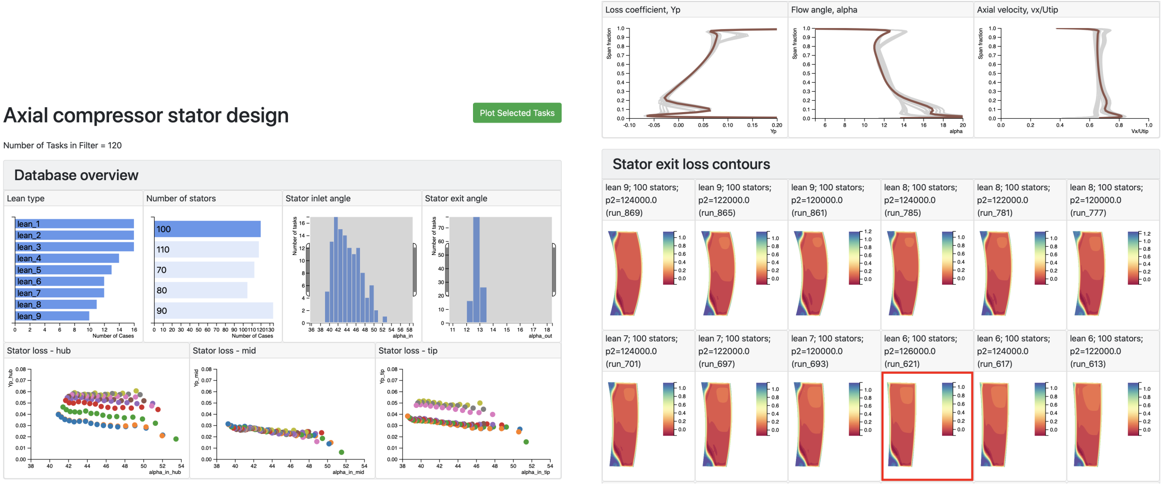
B. Implementation in dbslice

dbslice[6] is an open source implementation of the concept outlined in Section II.A. Figure 2 shows two screenshots from a compressor design study demonstration available at the dbslice website[7]. In Fig. 2(a), the user has already filtered the database of 590 simulations down to a subset of 120 by selecting only those cases with 100 stator blades. During the selection process, the scatter plots of hub, mid-span and tip loss respond to the changing filter settings on-the-fly. It becomes apparent that there is a strong linkage between lean type and the loss in the hub and tip regions. In Fig. 2(b), the contour plots of loss at stator exit show that the high losses are caused by corner separations. All the plot are linked – when the user moves the cursor over a point in a scatter plot, the same case is highlighted in all the plots, enabling the user to quickly evaluate the performance of a particular design in multiple contexts.

III. Extension of dbslice for Design Optimisation

A. Goals for user experience

dbslice is designed to be extendable such that additional plot types can be added easily. In design optimisation, two outcomes are sought. First, an understanding of the relative influence of the input parameters (e.g. geometry definitions) on the chosen output metric. Second, a set of input parameters that achieve optimum performance under given constraints. The goal is to extend *dbslice* so that the discovery of linkages between inputs and outputs, and the likely set of optimal inputs, is enhanced through interactive manipulation of the data.


Axial compressor stator design
Plot Selected Tasks
Number of Tasks in Filter = 120
Database overview
Lean type
Number of stators
Stator inlet angle
Stator exit angle
lean_1
lean_2
lean_3
lean_4
lean_5
lean_6
lean_7
lean_8
lean_9
Number of Cases
100
110
70
80
90
Number of Cases
alpha_in
alpha_out
Stator loss - hub
Stator loss - mid
Stator loss - tip
alpha_in_hub
alpha_in_mid
alpha_in_tip
Loss coefficient, Yp
Flow angle, alpha
Axial velocity, vx/Utip
Span fraction
Yp
alpha
Vx/Utip
Stator exit loss contours
lean 9; 100 stators; p2=124000.0 (run_869)
lean 9; 100 stators; p2=122000.0 (run_865)
lean 9; 100 stators; p2=120000.0 (run_861)
lean 8; 100 stators; p2=124000.0 (run_785)
lean 8; 100 stators; p2=122000.0 (run_781)
lean 8; 100 stators; p2=120000.0 (run_777)
lean 7; 100 stators; p2=124000.0 (run_701)
lean 7; 100 stators; p2=122000.0 (run_697)
lean 7; 100 stators; p2=120000.0 (run_693)
lean 6; 100 stators; p2=126000.0 (run_621)
lean 6; 100 stators; p2=124000.0 (run_617)
lean 6; 100 stators; p2=122000.0 (run_613)

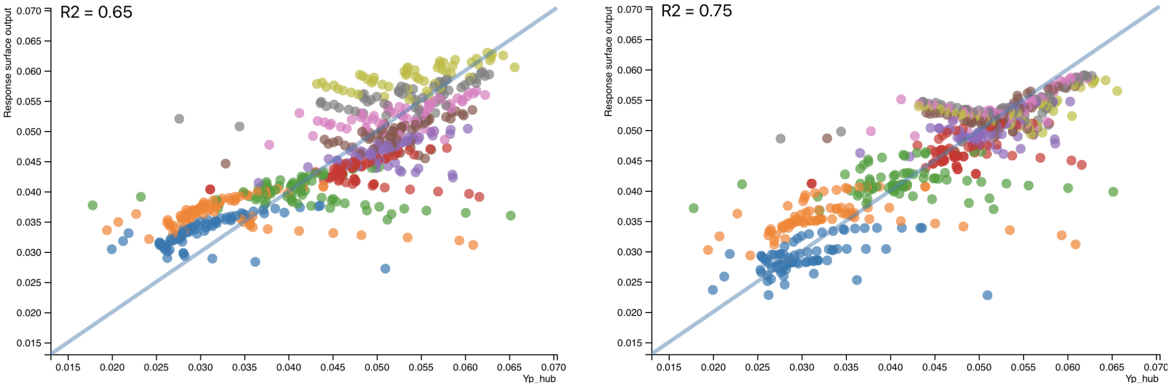
(a) Filter on 100 blades

(b) Selected case has hub separation only

Fig. 2 Visualisation of a dataset of 590 compressor stator simulations (scatter points and lines are coloured by lean type).

B. New visualisation functionality

A focus of the extensions implemented for design optimisation is to *process*, not just visualise, the metadata on the client, through fitting response surfaces and by using statistics, to provide insights into the data. In each of the following cases, the plots are computed based on the set of cases in the current filter. This allows outliers to be removed, or a particular area of the design space to be focused on.



(a) Linear response surface

(b) Quadratic response surface

Fig. 3 Visualisation of the response surface fit to the compressor stator hub loss data.

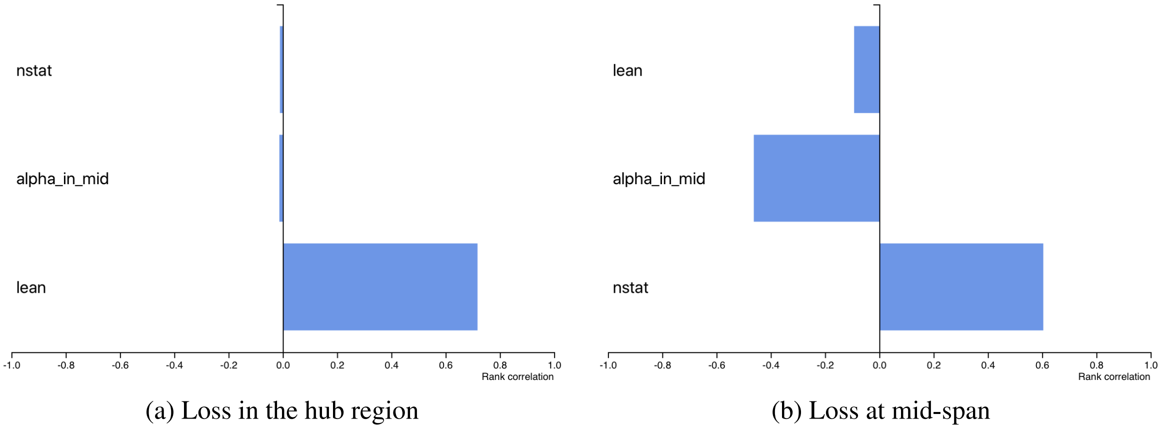
Response surface correlation Two response surfaces, linking inputs x_i to an output y have been implemented. These are: linear,

$$y = \beta_0 + \sum \beta_i \, x_i \tag{1}$$

and quadratic (keeping only the diagonal quadratic terms),

$$y = \beta_0 + \sum (\beta_{1i} \, x_i + \beta_{2i} \, x_i^2) \tag{2}$$

where β are the coefficients determined by a least-squares fit to the metadata. Figure 3 shows an example of this type of plot using the ‘hub loss’ output parameter from the compressor design study data of Fig. 2. As expected, the quadratic fit shows improved correlation with the data. The points are coloured by the 'lean' input parameter and this illustrates the strong dependency between the stator hub loss and lean.



(a) Loss in the hub region

(b) Loss at mid-span

Fig. 4 Spearman rank correlation coefficient showing the links between input parameters and the loss in the hub and mid-span regions of the compressor stator.

Rank correlations The Spearman Rank Correlation Coefficient is implemented to highlight correlations in the ordered rank index of a given output y to each input x_i. A rank correlation coefficient of 1 means that y increases continually as x_i increases, but says nothing about the function that connects x_i and y. Figure 4(a) shows, again, the dominant correlation of the compressor stator hub loss with the lean input parameter. Mid-span loss, however, Fig. 4(b) is only weakly dependent on lean, and has a stronger correlation with the number of stators ('nstat') and the inlet flow angle ('alpha_in_mid').

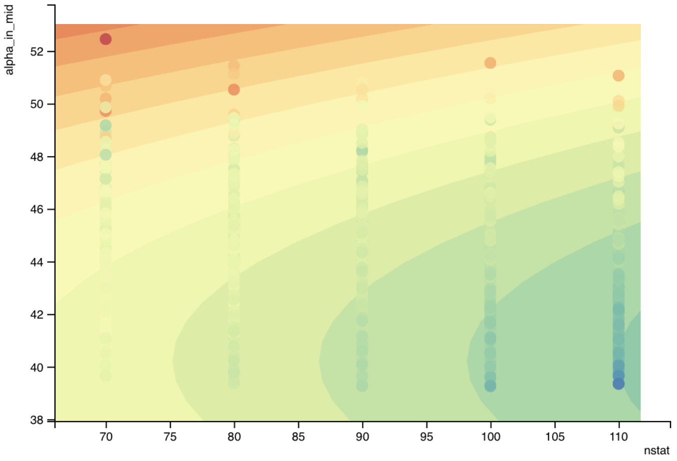


Fig. 5 Two input variable (number of stators and inlet flow angle) response surface for mid-span loss. Points and surface are both coloured by mid-span loss.

Response surface for two inputs In the case of two input parameters, the response surface can be visualised in a contour plot, Fig. 5. The individual data points are also shown and, in the case of Fig. 5, they are coloured with the same parameter as the response surface itself; this allows a visual assessment of the ability of the response surface to represent the underlying data.

Grouping of related simulation data Each simulation result (point, line or surface) displayed by *dbslice* is normally considered to be distinct from all other results. However, it is often desirable to alert the user to connections between

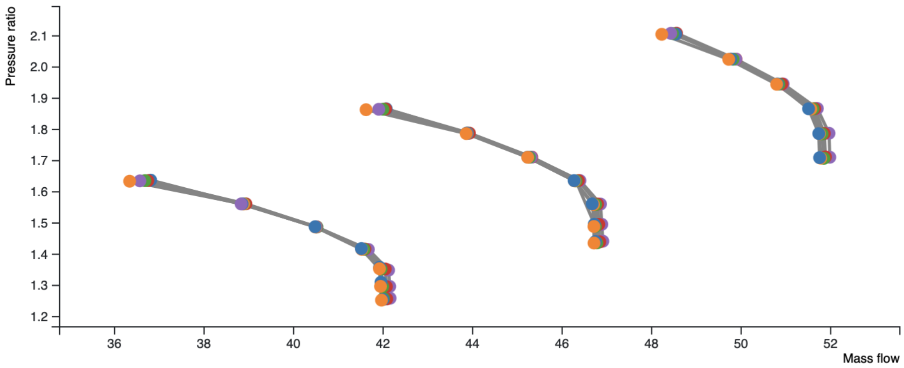


Fig. 6 Compressor characteristic lines formed by grouping points which share the same geometry and rotational speed

points that can be determined from the metadata. In the optimisation visualisation of Section IV, for example, steady and unsteady RANS simulations of the same geometry are connected. In Fig. 6, compressor pressure-ratio versus mass flowrate characteristics are produced by connecting simulations with the same geometry and rotational speed.

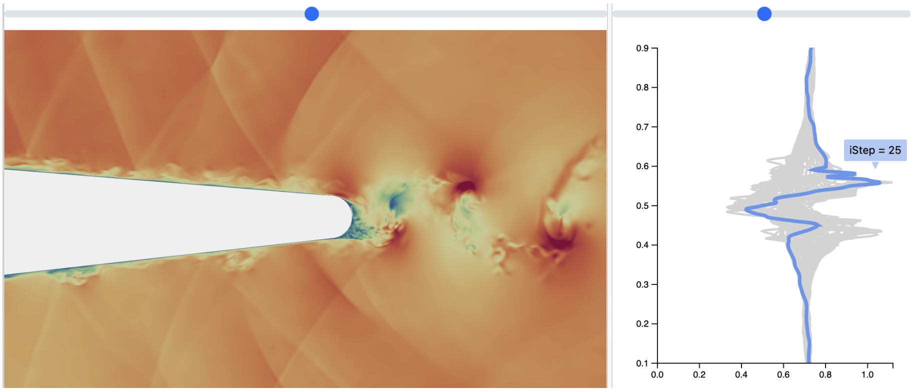


Fig. 7 Unsteady transonic trailing edge simulation: (left) Mach number snapshot; (right) Mach number profile within the wake. The two plots are kept in sync as the time-sliders are moved (at the top of each plot) or the user hovers over one of the lines in the exit distributions

Synchronising plots of unsteady data by time level When unsteady simulation results are available at multiple time instances, *dbslice* allows the user to select the time level currently displayed by moving a 'time slider' positioned above the corresponding plot. *dbslice* can also synchronise the time levels of multiple plots so that all surfaces and lines are displayed at the same time level. In addition, hovering over an individual line from a plot showing lines at different time levels will also change the time slider and synchronise all corresponding plots; this is useful for determining the mechanism responsible for an outlier result that occurs at a particular time instant. Figure 7 shows results from an LES simulation of a turbine trailing edge at high subsonic Mach numbers. During the vortex shedding cycle, shock waves are formed that propagate upstream over the blade surfaces, Fig. 7(a). Downstream of the blade, the shed vortices cause characteristic over- and under-shooting of the Mach number profile in the wake, Fig. 7(b). Moving the time-slider causes the plots to move to the corresponding time level. The plots are configured so that the sliders move together and hence the Mach number contours and line plots are kept in sync. The user can also hover over any of the lines in Fig. 7(b) and the time levels in each plot will 'snap' to the time associated with that wake profile line.

IV. Application to turbine optimisation

A. Case and optimisation approach

The application selected for the present paper deals with the three-dimensional aerodynamic shape optimization of a high pressure turbine rotor, as shown in Fig. 8, which is routinely performed during the design process after the initial two-dimensional section design has been completed. Current industrial practice is to employ steady RANS simulations for this optimization exercise.

The optimization parameters selected for this task are movements of the designed sections on several radial heights, as shown in Fig. 9. In our case, we select five design sections equally spaced between hub and tip with two free parameters each, a rotation (called "ROTATE") around the centroid and a shift in circumferential direction (called "THETASHIFT") on sections 1,5,11,16,21. This leads to ten optimization parameters $\mathbf{x} \in \mathbb{R}^{10}$ with prescribed lower ($\mathbf{x}_l$) and upper bounds ($\mathbf{x}_u$). The deformed aerofoil is obtained by radial spline interpolation between these sections.

The goal of the aerofoil shape optimization is to improve the efficiency η of the turbine stage as shown in Fig. 8 without changing the operating point of the turbine, i.e. the normalised massflow (also known as capacity) c and the rotor reaction r have to be kept at their prescribed values (denoted by $\cdot^*$). This leads to a single-objective formulation with two equality constraints,

$$\max_{\mathbf{x}} \eta \quad \text{subject to} \tag{3}$$

$$c = c^* \tag{4}$$

$$r = r^* \tag{5}$$

$$\mathbf{x}_l <= \mathbf{x} <= \mathbf{x}_u \tag{6}$$

The optimization task is solved in a sequence of steps:

1) A space-filling set of points (also called Design of Experiments - DoE) is computed, in our case using the optimal Latin Hypercube (oLHS) approach. At those points (which all define a different and randomly selected aerofoil shape), the 3D RANS CFD simulation is performed and the values for η, c and r are recorded.
2) After the simulations of all DoE points have been completed, an optimization procedure, see e.g. [8] and [9], selects promising optimal candidates for objective and constraints and iterates until convergence is reached. These optimal candidates are again simulated using steady RANS and added to the database.
3) In order to gain further understanding of the improved flow features, promising optimal candidates are then assessed using simulation types with higher accuracy, e.g. nonlinear harmonic balance methods [10] or, as in our case, unsteady RANS simulations [11]. It is also possible to incorporate those different fidelity levels into the optimization algorithm itself, as shown in [12] and [13]. In both cases we obtain simulation data for the same geometry with different fidelity and associated accuracy levels.

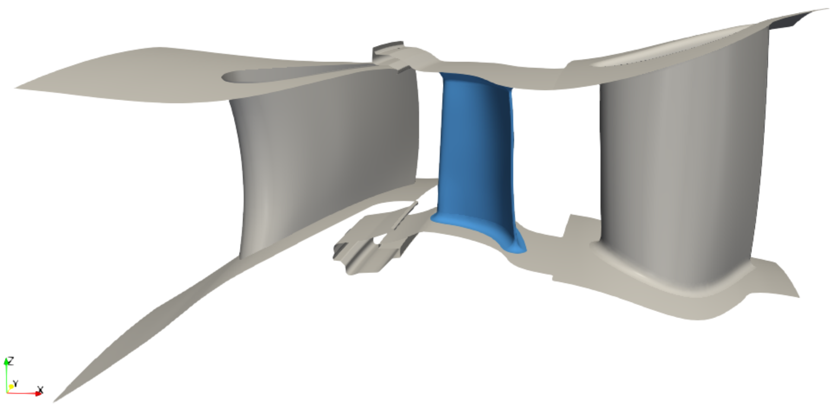


Fig. 8 Geometry of the CFD setup used for the aerodynamic shape optimization of the rotor (depicted in blue)

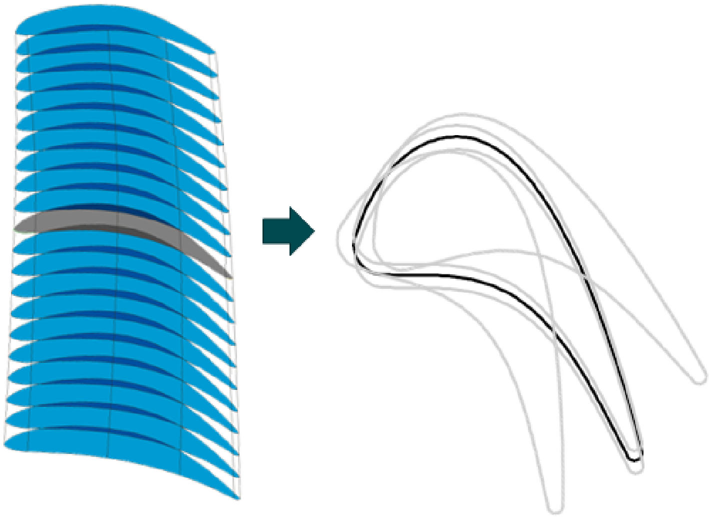


Fig. 9 Aerofoil section parameterization

In the following sections, visualisations of the computations are presented in three sections of increasing fidelity: Design of Experiment simulations, candidate optimal designs and, finally, the unsteady computations. The goal of *dbslice* is an interactive simulation exploration environment and so the figures presented here can only give snapshots of the functionality; the full demonstration is available at the *dbslice* web site[7].

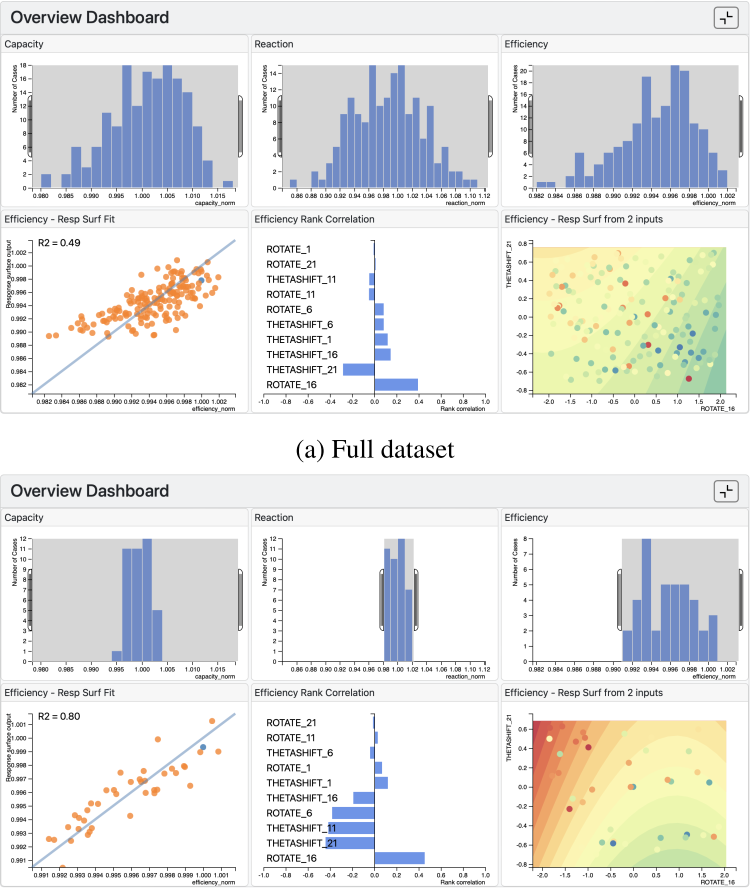
B. Visualisation of Design of Experiment simulations

The metadata (a table of input parameters and output performance metrics with one row for each simulation) is read in from a csv or json file. Figure 10 shows two views of the metadata 'dashboard' for the turbine design optimisation visualisation. The top row of the dashboard has histograms of normalised reaction, capacity and efficiency. The second row comprises the three new plot types: a quadratic response surface for efficiency using all 10 input parameters; a rank correlation between efficiency and each input parameter; a response surface for efficiency using the two most influential input parameters (based on the rank correlation) - the points are also coloured by efficiency. The engineer can interactively filter the data set, Fig. 10(b), so that only those simulations that meet the capacity constraint are considered, and low efficiency outliers are removed; the quality of the response surface fit improves as a result. The key outcome is that, while several inputs influence the efficiency, the dominant parameters control the blade tip: ROTATE_16 (rotation of the profile at section 16 of 21); and, THETASHIFT_21 (circumferential shift of the profile at section 21).

C. Visualisation of candidate optimal designs

Based on the DoE cases shown in the preceding section, promising optimal candidates, as defined by the EGO approach[9] are then simulated. These are shown in the screenshots of metadata, Fig. 11, and detailed data, Fig. 12. Figure 11 now has an additional bar chart that allows selection of cases based on their type: 'baseline', 'DOE set', or 'optimal set'. The data has again been filtered on the reaction constraint and to remove low efficiency outliers. The points in the efficiency versus capacity scatter pot (upper-right plot) are now coloured by design type showing that the optimal candidates (green cases) have the highest efficiency. In Fig. 11, the user has again filtered the cases according to their reaction (close to the target) and efficiency (removing low efficiency outliers). The user has moved their mouse cursor to the highest efficiency point in the scatter plot of capacity versus efficiency (to right plot, 'run 204') and the same simulation is highlighted in the response surface plots.

In addition to the metadata, the user is able to transfer additional data, such as line and surface plots, from the server to look at details of the flow field. As an example, Fig. 12 shows radial distributions (of axial velocity, stagnation pressure and stagnation temperature) at blade exit, and views of the 3D blade coloured by static pressure, for a selection of cases from the optimal set. The user still has access to the metadata dashboard (not shown in Fig. 12), typically above the more detailed results on the screen, and can easily modify the selection of cases displayed. The line plots (csv or json files) are small enough to be read automatically as the metadata filters are adjusted without causing a distracting



(a) Full dataset

(b) Filtered to adhere to capacity constraint and eliminate low efficiency runs

Fig. 10 Visualising Design of Experiment simulations. Top row: histograms of reaction; capacity; and efficiency. Bottom row: efficiency response surface fit; rank correlation of input parameters with output efficiency; efficiency response surface based on two inputs (green-blue is high efficiency)

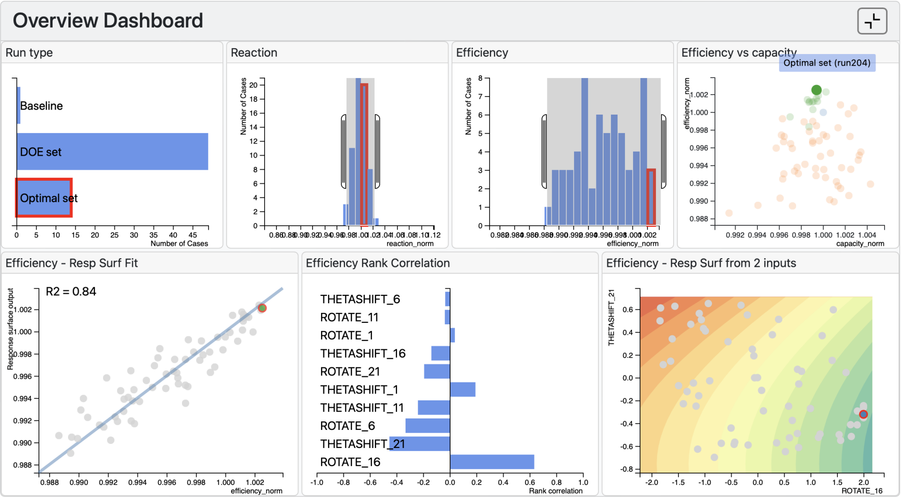


Fig. 11 Dashboard including baseline, DoE, and two sets of optimal candidates. Filtered on capacity constraint and to remove low efficiency cases. Optimal candidates are green and red points, DoE cases are orange.

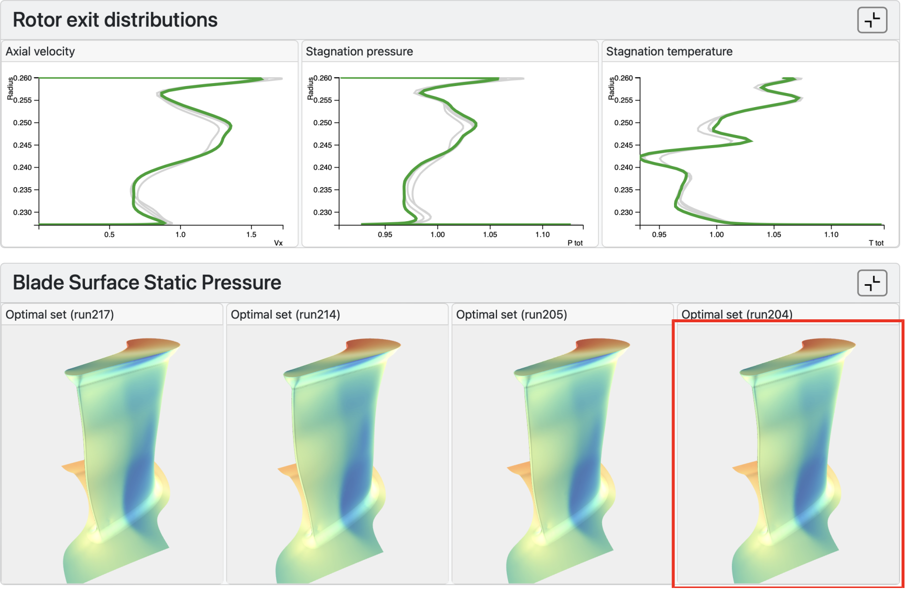


Fig. 12 Exit radial distributions and views of the blade surface static pressure for a selection of simulations - the case with highest efficiency is highlighted. All 3D views change in sync when any one case is rotated or zoomed.

latency for the user. The data for the 3D surfaces are read only when the user presses a ‘plot selected cases’ button at the top of the screen. To reduce load time, a the surface data is stored as a binary files created by pre-processing the VTK output from Paraview[14]; these files store the triangle vertices, indices and property arrays in exactly the format needed by webGL for rendering on the GPU, so minimal computational work is required by the web browser. In Fig. 12, the highest efficiency case (‘run204’) is highlighted. The renders of the blades are configured so that they all rotate, zoom and pan together when any one of the views is adjusted. This allows the engineer to inspect and compare particular features of the design. In this case, it is clear that high efficiency designs all have a characteristic S-shaped trailing edge due to the profile changes in the tip region.

D. Visualisation of unsteady simulations

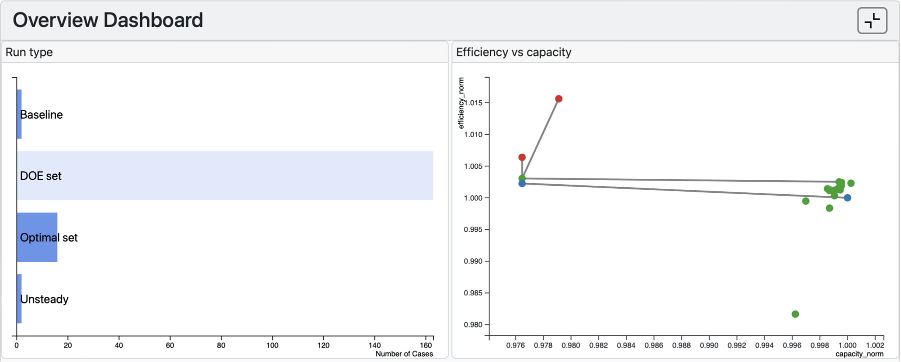


Fig. 13 Baseline and optimal unsteady simulations are added to the available cases (left). The lines joining the points in the scatter plot (right) connect cases which share the same geometry.

Having arrived at a small number of candidate designs, higher fidelity computations are then used for more accurate assessments. In our turbine blade optimisation case, 3-row unsteady RANS simulations are used. As is common in turbomachinery, the cost of the unsteady computation has been reduced by a slight change to the blade count of each row. To allow a fair comparison, a steady RANS computation is also run with the new blade numbers. Figure 13 shows that the user has selected the baseline, optimal and unsteady computations. There are two unsteady simulations (shown as red points) and each is connected to two steady computations (with the original and modified blade counts). The engineer can quickly see that the change to the blade numbers has caused a 2.5% drop in capacity, and that the unsteady runs have a higher efficiency (by up to 1%) than the steady computations.

Figure 14 shows data from snapshots from the two unsteady computations. Line plots of blade exit axial velocity distributions, and 3D renders of the blades coloured by surface static temperature (and an additional iso-surface at constant temperature) are shown. The line plots have been loaded from csv files. The surfaces have been exported as VTK data during the solver run using the Paraview-Catalyst plugin[15] and subsequently compressed into the *dbslice* binary file format. By storing the parameter of interest (static temperature in Fig. 14) at single precision and only storing the vertices once for the blade (since the geometry is fixed), the total file size is reduced from 1.11GB to 266MB for 30 time levels (a compression ratio of more than 4). As in Fig. 12, both 3D views are kept at the same camera rotation and zoom when either one is interactively changed by the user. In addition, each plot now has a time slider to determine which snapshot is displayed. The time level is also kept in sync so that changing the slider of one plot will change all four plots. The time level can also be altered by hovering over the axial velocity lines. This interactivity allows the user to interrogate the unsteady flow field in the same framework as the larger number of steady computations and the statistics from the Design of Experiment cases (all of the previous cases remain displayed and can be accessed by the user by scrolling in the web-browser). By moving between different fidelities in an interactive and frictionless way, the engineer can uncover links between design parameters, steady flow performance and unsteady flow mechanisms.

V. Conclusions

The use of computational fluid dynamics as part of design optimisation routinely generates hundreds of simulations. These computations are typically of different types: design-space-spanning cases (following a Design of Experiments

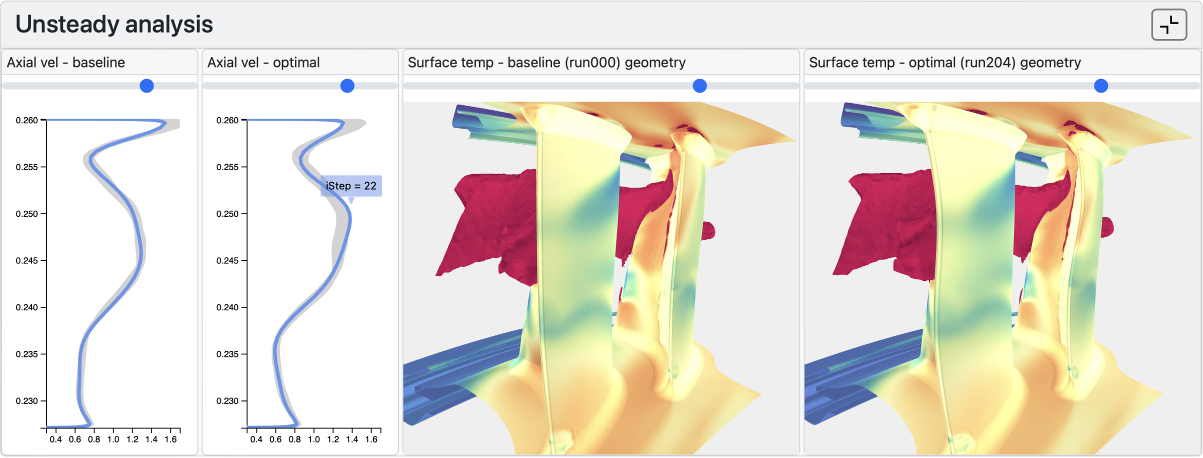


Fig. 14 Unsteady snapshot data for the baseline and optimal simulations. Line plots of blade exit axial velocity distributions, and 3D renders of surface static temperature with an iso-surface at constant temperature. The time-level can be changed by moving the slider above each plot, or by hovering over a line in the velocity distributions. The time level for each plot, and the 3D view of the blades and iso-surfaces, are kept in sync as they are modified by the user.

approach, for example); a set of potential optimal candidates; and a subset of design candidates assessed by more computationally expensive, higher fidelity, simulations. An interactive visualisation framework is presented that allows all the computations performed as part of an optimisation to be navigated and visualised together. The goal is to provide design review meetings with results displayed in context, and to present data, in real time, that responds to the questions and ideas raised by the participating engineers.

The following three contributions are highlighted:

1) By always displaying the best performing design candidate in the context of its ‘peer group’ set of optimised cases, and also with its ‘ancestry’ of design-space-spanning computations, the user is better able to make connections (qualitative and quantitative) between input parameters and the output metrics of interest.
2) As the set of cases under consideration is filtered (according to design constraints, for example) additional processing is performed to construct visualisations that assist the user in identifying the influence of key input parameters. These plots, which respond interactively to the the current set of cases, include response surfaces and Spearman rank correlation coefficients.
3) The increased amount of data associated with unsteady computations creates a visualisation challenge. In the present work, time-averaged results are presented alongside the steady cases selected by the user. If the steady and unsteady cases are related (they share the same blade geometry, for example) the associated points on a scatter plot are joined by lines. In displaying time-accurate results, the user can change the time-level by moving a slider. All points showing time-accurate line or surface data remain in sync (in time and, for three-dimensional surfaces, in the rotation and zoom of the view). This allows the user to compare performance and flow mechanisms at identical times across multiple design candidates.

Acknowledgments

The turbine application example presented in this paper was created within the framework of the DARWIN research project (20D1911A), funded by Rolls-Royce Deutschland Ltd & Co KG and the Bundesministerium für Wirtschaft und Klimaschutz. Rolls-Royce Deutschland’s permission to publish this work is gratefully acknowledged.

References

[1] Slotnick, J., Khodadoust, A., Alonso, J., Darmofal, D., Gropp, W., Lurie, E., and Mavriplis, D., “CFD Vision 2030 Study: A Path to Revolutionary Computational Aerosciences,” *NASA/CR-2014-218178*, 2014.

[2] Victor, B., "The Humane Representation of Thought," 2014. URL http://worrydream.com/TheHumaneRepresentationOfThought/note.html.

[3] Pullan, "Making Use of Our Data," *Journal of the Global Power and Propulsion Society*, 2021.

[4] Bostock, M., "d3.js Data-Driven Documents," *https://d3js.org*, 2022.

[5] threejs, "threejs - JavaScript 3D library," *https://threejs.org*, 2022.

[6] Pullan, G., and Li, X., "dbslice - Flexible, Web-Based, Database-Driven Visualization," *AIAA 2018-1173*, 2018.

[7] dbslice, "dbslice - Interactive, hierarchical, database-driven plotting," *http://www.dbslice.org*, 2022.

[8] Sacks, J., Welch, W. J., Mitchell, T. J., and Wynn, H. P., "Design and Analysis of Computer Experiments," *Statistical Science*, Vol. 4, No. 4, 1989, pp. 409–423.

[9] Jones, D. R., Schonlau, M., and Welch, W. J., "Efficient Global Optimization of Expensive Black-Box Functions," *Journal of Global Optimization*, Vol. 13, 1998, pp. 455–492.

[10] Wang, F., and di Mare, L., "Computation of multistage flows using a fourier approach," *AIAA Journal*, Vol. 60(1), 2022, p. 345–359.

[11] Giulio Zamboni, P. A., "Unsteady Interaction Between the Leakage and the Main Passage Flow in a High Pressure Turbine Rig, GT2016-56041," *Proceedings of ASME Turbo Expo*, 2016.

[12] Kennedy, M., and O'Hagan, A., "Bayesian calibration of computer models," *Journal of the Royal Statistical Society*, 2001.

[13] Le Gratiet, L., "Multi-fidelity Gaussian process regression for computer experiments," Ph.D. thesis, Oxford University Oxford, UK, 2013.

[14] Ahrens, J., Geveci, B., and Law, C., *An End-User Tool for Large Data Visualization, Visualization Handbook*, Elsevier, 2005.

[15] "ParaView Catalyst Overview," https://www.paraview.org/Wiki/ParaView/Catalyst/Overview, 2022. Accessed: 2022-11-28.