Eleventh International Conference on Computational Fluid Dynamics (ICCFD11), Maui, Hawaii, USA, July 11-15, 2022

ICCFD11-3603

Industrial and Biomedical CFD Workflows Enhanced via Co-processing for Knowledge Capture and Computational Steering

Steve M. Legensky, Earl P.N. Duque, Dave A. Amels and Brad Whitlock*
Marcus Meyer, Axel Gerstenberger and Paolo Adami**
Rene Thümmler***
Corresponding author: sml@ilight.com

* Intelligent Light, Rutherford, New Jersey, 07070, United States
** Rolls-Royce Deutschland, Dahlewitz, 15827 Blankenfelde-Mahlow, Germany
*** CFD Consultants GmbH, Gartenstraße 82, 72108 Rottenburg, Germany

Abstract: Over the past decade, the convergence of accessible HPC and high-fidelity simulation has amplified a bottleneck in analysis workflows: writing, managing and reading very large files. There has been much research performed on mechanisms to access information directly from the memory of the running solver code, known variously as 'in situ' or 'co-processing'. This paper describes a new co-processing approach for enhancing CFD workflows that provides services for visualization, data science, job monitoring, provenance capture and logging along with computational steering of the solver code. The software, called Kombyne™ accomplishes these functions with a very low code footprint, few to no external dependencies and direct support for 'in transit' operation where the workload and memory requirements are delegated to a separate process, working in tandem with the solver code.

This paper documents three recent examples of Kombyne™ applied to industrial-scale CFD analyses. In the first case, high-fidelity frequency analysis of a cylinder in supersonic crossflow was needed to calibrate data science workflows with experiment. The unique requirement here is obtaining a block of sample points at every solver timestep, to ensure that no temporal aliasing artifacts were present in the data used for the FFT analysis. The second application is a biomedical workflow intended to demonstrate the feasibility of using quick-turnaround automated CFD to advise surgeons on the condition of a patient's aneurysms[1]. The third example uses "Computational Steering" where the Rolls-Royce production flow solver Hydra was instrumented with the Kombyne™ software to enable on-the-fly changes of the computational setup for the production of compressor maps for gas turbine engines[2]. The common theme in these three applications is greatly increased functionality and turnaround speed in existing CFD codes by augmenting them with co-processing.

Keywords: Computational Fluid Dynamics, Co-processing, Computational Steering, Knowledge Capture, Frequency Analysis, Computer-aided Surgery.

1 Introduction

Advances in HPC and modelling techniques have led to a steady increase in the size and fidelity of solutions. Unsteady calculations have become more common as researchers seek to capture more of the dynamics of underlying phenomena. As size and fidelity increase, there is a requisite increase in the amount of information required for the calculation, with meshes composed of billions of cells requiring huge amounts of memory and the correspondingly large files for checkpointing and post-processing. The situation is further exacerbated in the unsteady domain, where tens of thousands of timesteps may result.

In the vast majority of these large scale computations, a large amount of memory is required for high fidelity solutions to the Navier-Stokes equations and associated physical models. However, computation of engineering quantities such as forces/moments and mass flow along with visualization of model geometries, cut planes, wakes and vortices requires only about 10% of the mesh and field data residing in memory. Given that file I/O is much slower than memory access and network transfer is slower still, the concept of going directly from solver memory to extracted surfaces, subvolumes or even rendered images has the potential to improve the wall clock time performance of modelling and simulation workflows by one to three orders of magnitude and is today essential to certain production workflows [3]. Figure 1 illustrates how co-processing functions are invoked during solver code iterations.

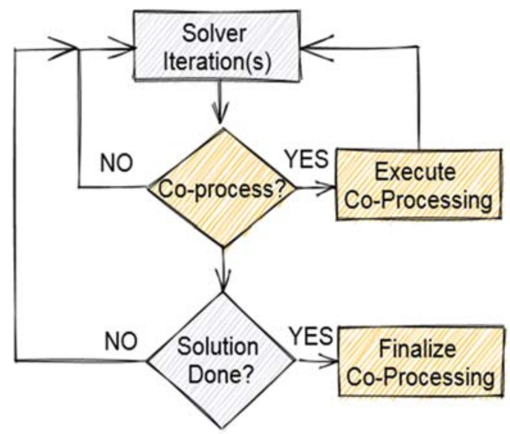

Figure 1: In situ iteration with Co-Processing

Several software solutions have been developed to implement these mechanisms, such as VisIt/Libsim[4], Paraview/Catalyst[5], and SENSEI. These tools are developed with visualization as the primary goal and are highly dependent upon the VTK graphics libraries [kitware, USA]. While co-processing has improved analysis and visualization for extremely large data, this achievement has been hard won. Two of the most prominent state-of-the-art in situ infrastructures, VisIt's Libsim and ParaView's Catalyst have been created from full fledge post-processing tools, which rely on very large and complex software libraries. This makes them hard to build, hard to integrate, hard to get working, inefficient, a risk to crash the simulation, or require mixed-language programming. SENSEI, currently under development by a team directed by the Lawrence Berkeley Laboratory with R&D performed by Intelligent Light and kitware, suffers from the same dependency burden.

1.1 Kombyne™

Based on the need for more effective and easy-to-integrate co-processing [6], Intelligent Light in 2018 proposed to create a next-generation software toolset that could provide both in situ and in transit operation. Development funding was obtained via Small Business Innovative Research (SBIR) grants from the Department of Energy, Office of Science. Brad Whitlock, a Senior Software Engineer at Intelligent Light lead this effort, after having spent over a decade working on scalable visualization and in situ workflows with the Visit/Libsim codes at Lawrence Livermore National Laboratories. Whitlock's experience, combined with the many years of CFD practice at Intelligent Light (via FieldView) drove a focused effort to resolve the major drawbacks of the existing software tools mentioned above:

- Difficult to build and port to new computer systems
- Difficult to instrument simulations
- May require unacceptable amount of runtime or memory from simulation

- May introduce error conditions (such as out of memory) that crash simulation
- Lack of readiness for heterogeneous computer architectures

Initially called “SCOREBOARD”, the project went through research and development phases both in-house and through externally funded efforts with industry [7]. The product was re-branded as Kombyne™ and was launched commercially in December of 2021. Kombyne™ is a high performance, parallel data analysis library that is designed to integrate with diverse simulations and get the most out of HPC hardware. Kombyne™ efficiently applies in situ and in transit data analysis techniques to create targeted data extracts that can be orders of magnitude smaller than typical simulation data. Kombyne™ shortens the time to insight, compared to post hoc methods and other in situ–only solutions, by enabling simulation and analysis to execute simultaneously. Kombyne™ also includes unique simulation monitoring and steering capabilities, bringing HPC data analysis for many jobs to a browser interface.

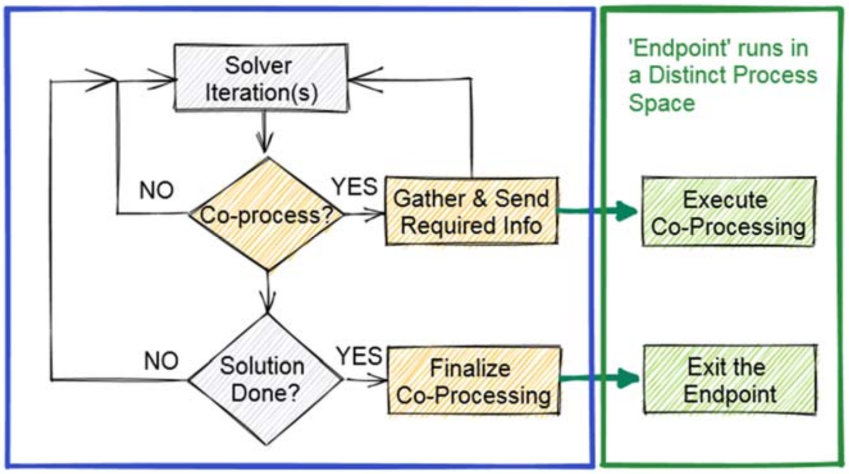

Figure 2 - In transit workflow with Endpoint operating in a separate process space

In transit is similar to in situ but it adds the step of **staging data to separate analysis resources** where data are processed (shown in Figure 2). Practitioners are often hesitant to include libraries that may compete for time and memory with the host simulation, or even worse, cause it to terminate. Kombyne™ mitigates this risk by sending simulation data to an analysis program called the “Endpoint”, which receives and processes data. Intelligent Light designed Kombyne™ for in transit to minimize code added into the simulation, reduce risk, and to reduce time and resource competition between simulation and analysis by making these operations occur in the Endpoint.

As shown in Figure 3, the fact that the in transit endpoint operates in a distinct process space on separate cores The solver code does not have to pause for the endpoint to complete its work thereby enabling more iterations per wall clock hour. Whether in the solver address space, or in an endpoint, Kombyne™ provides services, computes extracts and creates data products:

- Create cutting planes, iso-surfaces, boundary surfaces with populated with field variables
- Sample the meshes with arbitrary or structured coordinate patterns
- Render images of the extracted surfaces into PNG or Cinema [8]
- Execute expressions and triggers

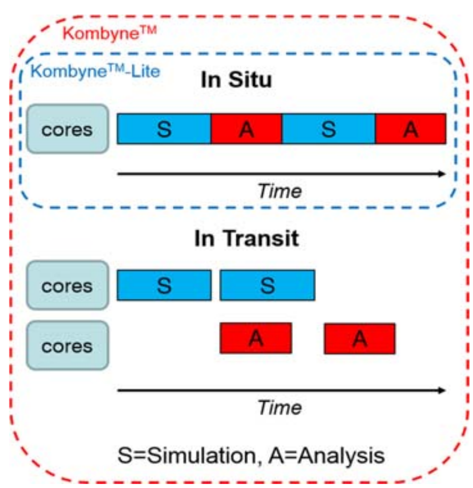

Figure 3 - In situ versus in transit operation

Data extracts enable the simulation to save meaningful results that would not be possible any other way. "Data Extract" is a broad term that encompasses reduced-size data products generated in situ. Data extracts can contain geometry-based subsets that focus on regions of interest, they can be images of the simulation data, or even computed metrics derived from the simulation data. Kombyne™ can produce various types of data extracts. For example, Cinema databases are an example of an image extract that can be explored interactively from multiple camera angles in a Cinema viewer. Better data extracts that retain more of the simulation's data at compact sizes are needed

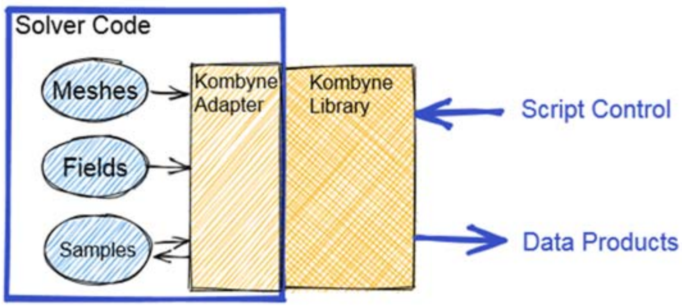

Figure 4 - Integrating Kombyne into a solver code

Integrating Kombyne™ into a solver code or data collection device is made easy due to the solver-oriented design of the data interface and the low external dependency count. This is one of the most important design goals for Kombyne™. The mesh and field interfaces are designed for zero copy and avoid the need for mixed language programming.

The key components are the Adapter and the Kombyne™ Library (Figure 4). The adapter header code and documentation are provided under an open source BSD license, simplifying potential intellectual property issues. Code is written to expose the meshes, fields and other quantities (samples) to the Kombyne™ API. Data is usually pointed to by reference, so that copies are not made at runtime. The interface is in the pattern of publisher-subscriber: Kombyne™ knows where things are, and accesses them when called for in a co-processing operation. The Kombyne™ library comes in two forms: Kombyne™ and Kombyne™ Lite. Kombyne™ Lite lacks some capabilities (such as in transit) and is available from the Intelligent Light website via a free download. Either form is available as static or dynamically loaded, for a wide variety of operating systems and hardware.

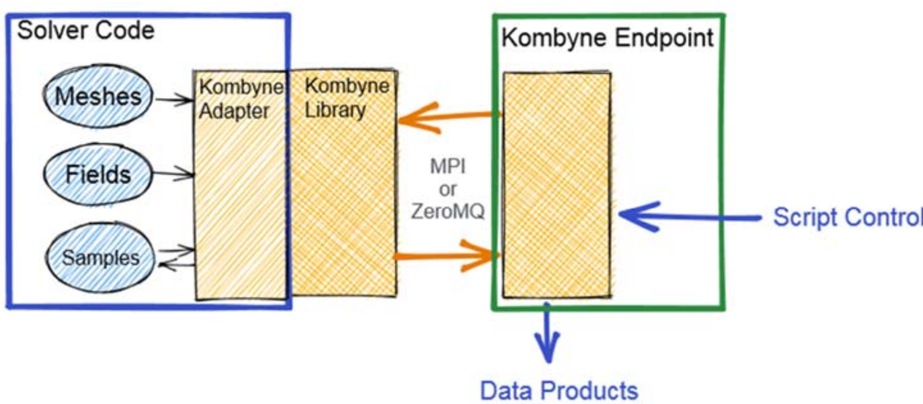

Figure 5 - Using a Kombyne Endpoint

Once a solver has been 'instrumented' with Kombyne™ for in situ operation, it is also ready for communication to an Endpoint process (Figure 5). The Endpoint can communicate with the solver process through one of two transports: MPI for tightly coupled operation and ZeroMQ for connection over an IP-protocol physical link. The configuration Kombyne™ will use at runtime is chosen by a command line option at solver startup time. **When used with an Endpoint, Kombyne™ automatically moves the computation of extracts, renderings and any user functionality added to a custom built Endpoint into the Endpoint and out of the solver process space.** When used with the ZeroMQ transport, the Endpoint can run on a separate compute resource, connected via IP.

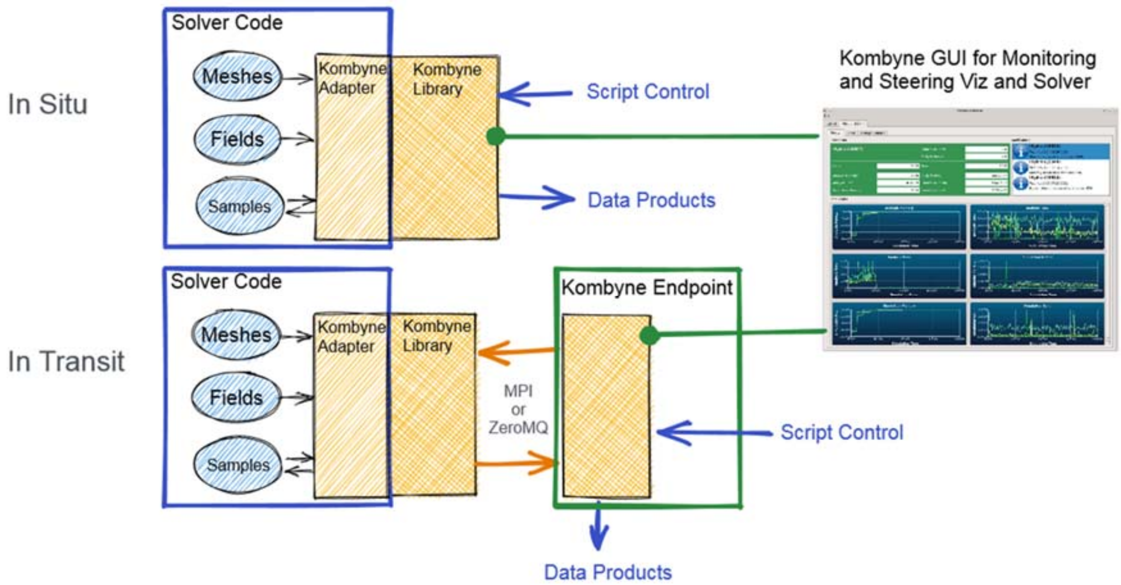

Figure 6 - Kombyne provides a monitoring a steering GUI for your jobs

It should be noted that Kombyne™ is 'all parallel, all the time'. While it can be operated in serial mode as well, Kombyne™ is fully MPI compliant and uses OpenMP as well. When tightly coupling an Endpoint process to a solver, a 'split MPI communicator' technique is used. This is very scalable and has been tested on the NERSC Cori system at up to 16K cores. When coupling via the ZeroMQ transport, the MPI domain for the solver and Endpoint are distinct. This the allocation of cores/nodes for the Endpoint is independent from that of the solver. One could have the solver running on 1024 compute cores and the Endpoint using 32 cores on a 'fat node'.

Finally, Kombyne™ provides a GUI for monitoring the operating status, resource utilization and data product production. Strip charts are available that show the amount of CPU, memory and networking resources utilized by the solver code and Kombyne™. This is very useful for debugging and tuning. Events may be reported through the Kombyne™ API and appear as notifications. The solver can also be paused, continued or stopped. All of this is accomplished in the single web or Qt GUI for all active jobs of a given user. As shown in Figure 6, this service is available whether you are using strict in situ or in transit co-processing.

Other tabs in the GUI support runtime computational steering of the data product creation and any solver variables the were selected for steering when the adapter was built. For the data products, extracts can be turned on or off and parameters such as iso-surface value or slice location can be changed. On the solver side, things like free stream alpha can be exposed to facilitate a drag polar, time step can be adjusted for unsteady calculations and as described in Section 4, boundary conditions can be manipulated to obtain engineering performance plots.

In the next three sections of this paper, examples of Kombyne™ applications will be surveyed. In each one, an existing solver code is instrumented with Kombyne™ to achieve research or performance results that are beyond the capability of the underlying CFD code. In that regard Kombyne™ provides a simple way to get more from the solver code(s) being used and allows a user to gain control of the simulation through a modern GUI.

2 Time Fidelity for Frequency Analysis: Cylinder in Crossflow

Unsteady CFD calculations are more frequently used today as the power and memory size of HPC systems grows and the need for predictive results are pressing for more accurate modelling. At the same time, the size of the file outputs from traditional CFD workflows provide a priori limitations on how much information can actually be stored and accessed. It is commonplace for users of remote HPC resources to wait days for network transfer of bulk output files, when then may only be interested in behavior at walls or very near to them.

Intelligent Light's Applied Research team had a requirement to calibrate a solver workflow for boundary layer separation point oscillation behavior of a cylinder in cross flow for which there was experimental data as reference (Figure 8). In more typical workflow, result files would be saved off at intervals such that the total file space need would fit within available resource. Or a 'rule of thumb' method would be used to determine the output rate. Frequency analysis is used in many types of engineering and experimental fluid mechanics but the rate limiting forced by file sizes creates roadblocks to use with high fidelity CFD.

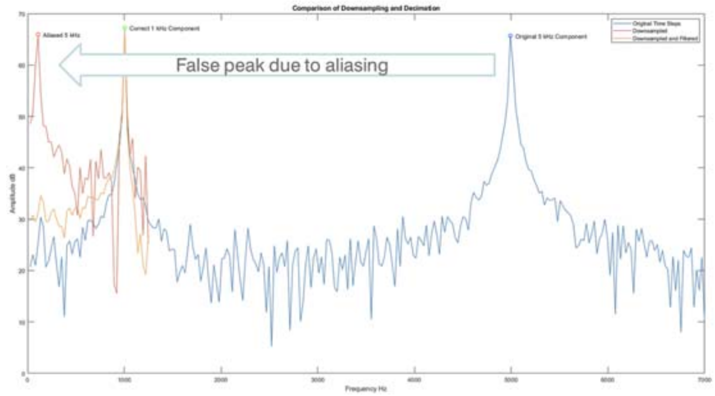

Figure 7 - Aliasing of results due to undersampling

Examining flow physics through results that are undersampled with respect to time can lead to spurious results. In the signal processing world, these principles are well understood. But a user performing FFT or DMD on undersampled data may observe false harmonic peaks unless a decimation filter is used. In Figure 7, the FFT of fully time resolved data is shown in blue. There are true 5kHz and 1kHz peaks. When downsampled timesteps are used directly, a 0.5kHz harmonic appears which is false: an artifact (shown by the red line). If the undersampled data is properly filtered (decimated) the orange line is much more faithful to the original data, shows the same peaks and does not exhibit aliasing. If the researcher is analyzing results without a reference as to the proper frequency behavior, it is very easy to be fooled by the false harmonics.

Kombyne™ permits data sampling in regions of interest at the solver iteration rate. In other words, one can obtain the most detailed data with respect to time due to the fact that the location and number of points that are saved as data products can be limited to just what is needed. For the cylinder in crossflow case, the target of study is the unsteady behavior of the separation point that moves up and down on the downstream side.

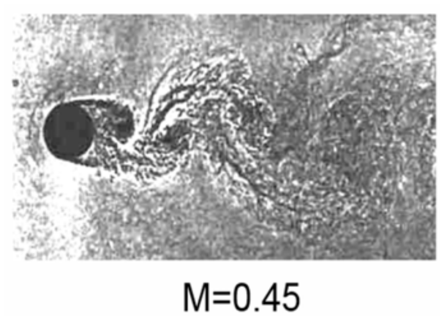

Figure 8 - From Ackerman [9]

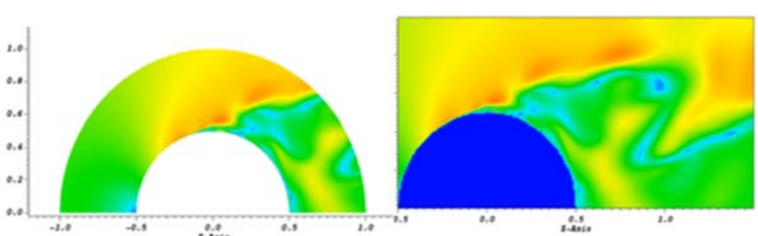

Figure 9 - Simulation results sampled in regions

Two structured sampling regions were specified in the Kombyne™ script that defines the data products: a annular cylinder and a box. Each sample contained about two million points, with the original gird containing almost 30 million points. The important fact is that these points were sampled at every converged solver timestep and written directly to Matlab files for subsequent DMD (Dynamic Mode Decomposition) computation. For visualization, VTK files were written every 200 timesteps. The flow solver used here is NASA's OVERFLOW2 v2.2n, which was instrumented with Kombyne™.

The Matlab files resulting from the simulation were read by the DMD analysis tool, which along with POD and FFT analysis tools comprise the Intelligent Light Data Analytics Suite (DAS). Figure 10 shows the GUI, a power spectral density (PSD) plot (above) and a plot of that mode within the CFD flow field. Comparing that plot with the unsteady CFD animation, we can see that the higher values of the 1kHz mode intensity (red, orange and yellow) correspond to the extent of the oscillatory motion of the separation point. Thus, it is possible to determine the movement extent of a dynamic flow field feature in a numerical, automated fashion, provided that the input data is of sufficient fidelity in time.

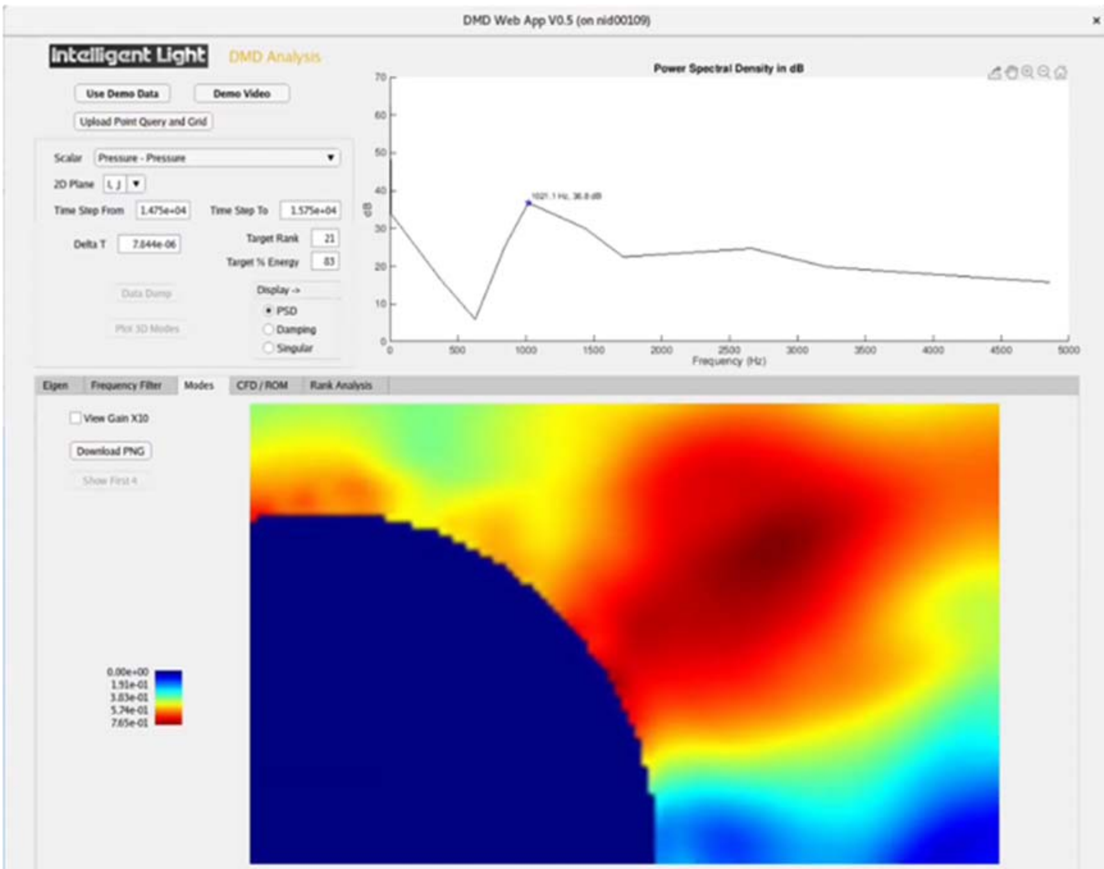

Figure 10 - DMD analysis shows a 1kHz energy peak in the PSD, the intensity of that mode is then plotted

Some conclusions and opportunities from this study:

- Frequency space analysis can be very useful in determining cause and effect of variable loads and other phenomena which are much more difficult to find via strip chart plots or animations.
- The unsteady nature of separation, re-attachment, vortices and shocks can not only be understood with this type of analysis but potentially used in flow path design.
- Artifacts coming into any kind of POD, DMD or FFT can cause confusing and or inaccurate results. Issues such as time fidelity and CFD solver numerics embedded in the code such as limiters or turbulence models can contaminate not only frequency or eigen analysis but machine learning applications as well. In this study, OVERFLOW2 was run with the turbulence model turned off, with a very fine mesh and timestep.
- Kombyne™'s structured sampling and direct output to Matlab (or CSV for python analytics)

greatly simplifies the development of automated knowledge capture workflows that are repeatable, of high fidelity and fast.

3 Rapid Knowledge Capture: CFD for Aneurysm Diagnostics

The use of robots, advanced scans and AI/ML in medicine offer promise to achieve better outcomes. In a research project funded by the German government, Dr. Thomas Wagner of the Universitätsklinikum Regensburg (University Hospital Regensburg) teamed up with CFD Consultants GmbH to evaluate a CFD-based approach to determining the need for surgical intervention. A cerebral aneurysm is a sack-like bulge of the vascular wall of a brain artery. The risk when an aneurysm is detected is that it may rupture, with dire consequences. As with many medical issues, assessing this risk is not simple, as there are many factors that influence the evaluation. One of the factors is the behavior of blood flow through the aneurysm.

It is not an easy task to observe the blood flow through an aneurysm, but with the help of CFD, it may be possible to understand the risk factors through simulation. There has been much research on the use of CFD to help predict the probability of rupture (see [10] for reference), the effective use of such a capability in a surgical environment faces two huge challenges: the neurosurgeon/medics are typically not CFD experts and the turnaround time for the entire process needs to be reliable and fast. The goal of the research effort therefore was to prototype a workflow that could rapidly convert well known medical scans, into meshes, automatically set the (complex) boundary conditions, run the CFD (OpenFOAM) efficiently and then extract/present the results (Figure 11).

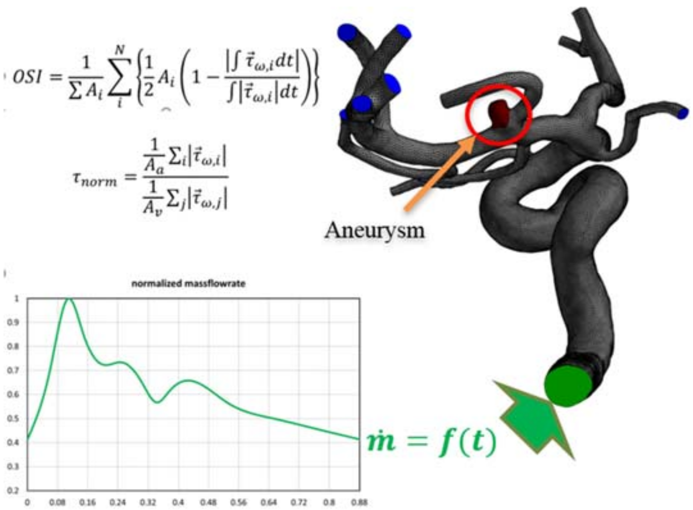

Figure 11 - Prediction functions, unsteady inflow & complex

The presentation by CFD Consultants (see [1]) details the pipeline used to go from raw medical imaging data to a 3D mesh (using cfMesh from the OpenFOAM toolbox), and setting the complex boundary conditions for blood flow (non-Newtonian). OpenFOAM's configurability was very important in meeting the constraints of this effort. However, the magnitude of the solution problem remained vast: an unsteady run with four heartbeat cycles needs between 15,000 and 30,000 timesteps. This can take 2-3 weeks to run and generate up to 100GB of disk space.

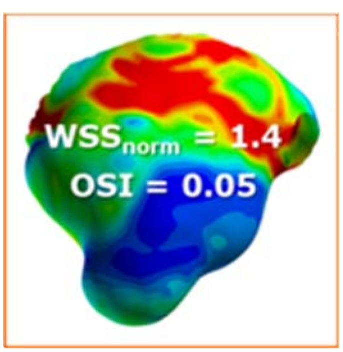

Figure 12 - Output image for medics including WSS & OSI values

The interesting outcome is the oscillating shear stress and the normalized wall shear stress as scalar numbers for the medics as in Figure 12. The medics then can decide if it is necessary to operate or not, of course also on the basis of a lot more numbers. The goal is to have a fully automated workflow from getting the real life data from a scan of the brain over to the automatic segmentation that creates a geometry as an STL that is used to generate the mesh run the simulations, and get back the numbers and images. The steady state simulation shall be run within 30 minutes to get first results and the unsteady simulations shall run within two days. The normalized wall shear stress and the oscillation shear index are highly mesh sensitive, so a boundary layer is necessary. Testing showed that polyhedral meshes show the best results at the minimum cell size. Polyhedral meshes were therefore used and the Kombyne™ pipelines were enhanced to handle them.

The post-processing step, getting useful information quickly to the medical team, is one area of potential gain, through the use of Kombyne™ workflows integrated into OpenFOAM. This can greatly reduce the amount of information saved by focusing in the surfaces of the blood vessels and the aneurysm: this is where the shear stress is computed. Surface data in an unstructured data set can be as little as 3% of the entire simulation storage.

CFD Consultants' software engineers implemented the Kombyne™ OpenFOAM adapter for this effort. Those familiar with the code know that OpenFOAM is NOT a single application; it is a set of multiple tools and solver applications and libraries (the v2012 installation consists of 310 executables and 156 shared libraries!). The engineers chose the functionObjects mechanism to enable a shared library integration, using fvMeshFunctionObject as base class. This method provides simple access to mesh and fields through reference pointers so that a 'zero copy' implementation is possible. An in transit workflow was implemented for co-processing of the simulation results as the solver ran. The ZeroMQ transport was used to communicate with the Endpoint.

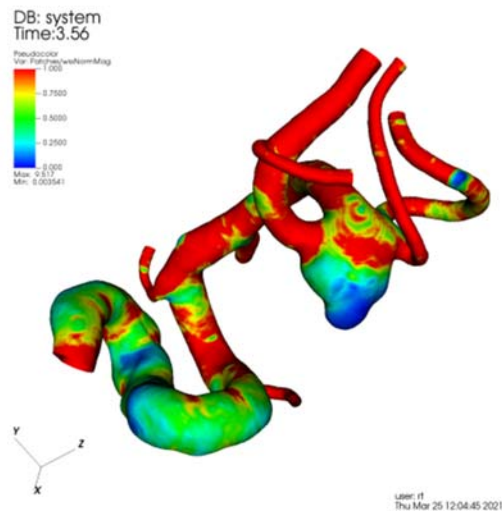

Figure 13 – Single timestep, blood vessels with aneurysm, colored by Wall Shear Stress

The published results of the research project were:

- A working medical diagnostic pipeline was demonstrated that used segmentation of scans created by medical imaging to produce STL files that formed the input to an automated polyhedral meshing step. OpenFOAM was used to calculate steady state and unsteady simulations and to derive the important diagnostic quantities oscillating shear stress and the normalized wall shear stress.
- OpenFOAM was instrumented with Kombyne™ via the functionObjects mechanism and the Kombyne™ software was enhanced to handle the polyhedral meshes and boundaries needed for this project.
- An in transit workflow was used to directly create surface extracts of the blood vessels and the aneurysm that were colored by the shear stress scalars. This produced actionable results much more rapidly than saving the volume files and reading them back into visualization software. In addition to single images, unsteady animations were also accessible to medics to increase their understanding (Figure 13).

4 Computational Steering: Turbomachinery Performance Maps

Rolls-Royce Deutschland (RRD) is a world leader in aeroengine products and has been funded under a German government initiative known as "PRESTIGE" whose purpose is to extend the state-of-the-art in computational and experimental techniques used in design. Intelligent Light was selected as a subcontractor to address co-processing workflows for massive simulations of full 360 degree compressors, some on the order of several billion cells, fully unsteady. Kombyne™ has been integrated into the RRD production CFD code "Hydra" for the PRESTIGE effort and this enabled the lead researcher, Marcus Meyer, to also use the computational steering capability in Kombyne™.

From Dr. Meyer's paper[2]: "Compressor maps for gas turbine engines show the relationship between efficiency, pressure ratio and corrected mass flow for a given corrected shaft speed. In conventional CFD workflows, several different operating conditions must be computed which means that several separate jobs must be submitted to converge the simulation for each prescribed exit mass flow while monitoring efficiency and pressure ratio until they settle. Since several points are required to accurately capture each single speed curve, this process becomes tedious and subject to error when performed manually."

Like so much in engineering analysis, a low dimensional performance plot, populated by highly complex 3D analysis forms the basis for comprehension and design. Obtaining the actual performance plot in the form shown in the adjacent Figure 14 is the objective of this computational study. Each point (such as that labelled "DP") requires a converged CFD solution. Python scripts have been used to create the input decks for the solver and to harvest the required statistics, but a new solver start and run is required for each point using standard techniques. The manual nature of starting each solver run and monitoring is error prone and time consuming. Copying, possibly converting units and then plotting the results can also be sources for error.

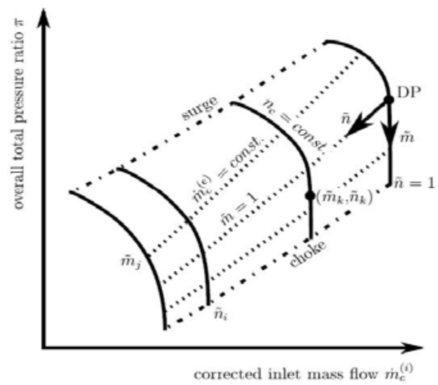

Figure 14 - Notional Compressor Map [11]

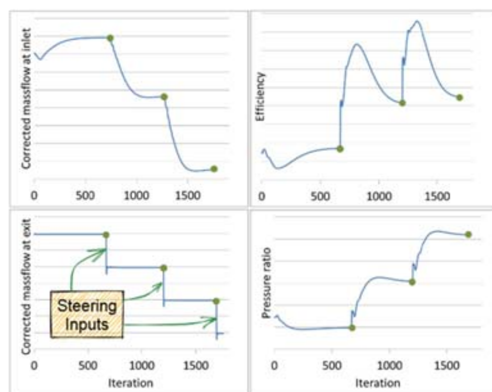

Figure 15 - Computational Steering

Instead of using the manual method to vary inputs to many solver runs, the Kombyne™ interface was used to accomplish "computational steering". When Hydra converged to the initial design point, the solver is paused and provided with a new value for the corrected massflow at the exit. In a sense, this is like running a virtual engine in a test cell and adjusting physical operation as it is running. This process and the convergence paths taken by the dependent variables is shown in Figure 15. The dots along the paths show the desired computational results to be used in preparing the compressor map.

The process of collecting values, steering to new points and also obtaining extract data products to be visualized can be controlled via the scripting capability in Kombyne™ called "triggers', essentially conditional behavior in the co-processing workflow. An example of these data products is shown in Figure 16. Here a series of boundary surfaces showing hub, shroud and blades along with a scalar-colored sampling surface at mid-height are output in VTK format and visualized in ParaView.

Dr. Meyer concluded his paper with the following observations on the benefits of computational steering to the preparation of the compressor performance plots when compared to submitting a number of jobs on the same computational resource:

- When running a single steering run on N cores in contrast with the usual approach of running M jobs, each on N/M cores, file I/O time is reduced, since grid and initial conditions are read only once instead of M times.
- Steering can reduce the number of iterations required, since the changed exit boundary

condition is deliberately chosen to be close to the previous one.

- In order to be more computationally efficient, the solver code should possess good strong scaling.

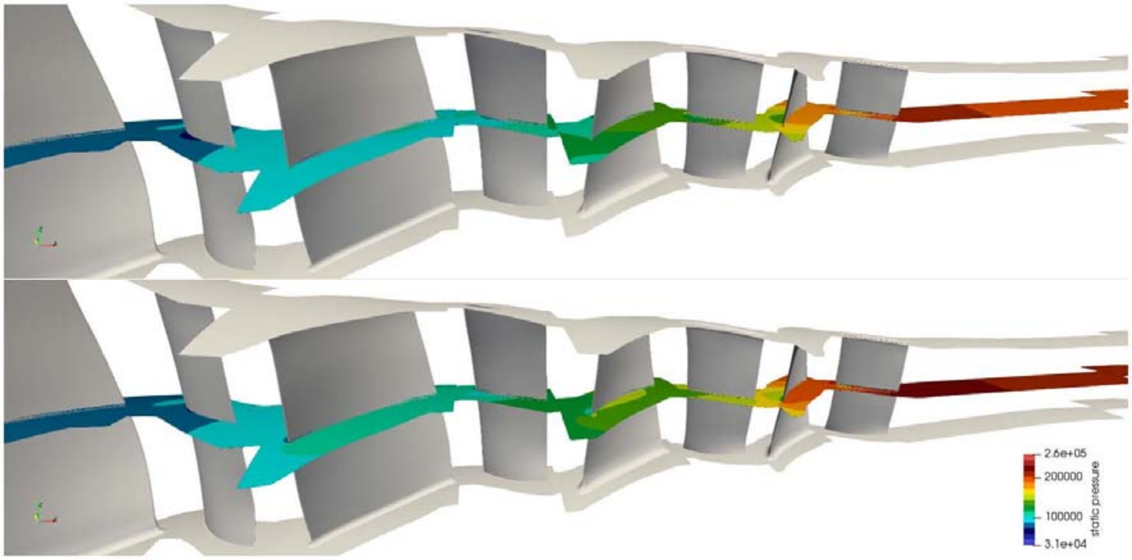

Figure 16 - Static Pressure Mid-height for first two converged operating points

5 Conclusions and Future Work

Co-processing is the strongest and most applicable mechanism for compressing analysis workflows within the existing technology framework of CFD solvers. The technique exploits the fact that knowledge capture for research and engineering can separated from the fine discretization in space and time for PDE solutions. The benefits of a co-processing work flow with targeted data products (matrices, surfaces, subregions, etc.) are many, here are a few:

- Simulation fidelity and breadth of analysis space does not have to be limited by disk resource or network copy time.
- Reduced order models, surrogate models, slice or subvolume extracts can be conveniently stored, retrieved and manipulated, giving more value to the digital assets resulting from CFD.
- Frequency-based analysis has many current applications and potential new uses are arising from eigen analysis and machine learning. However, users must take care to ensure that the fidelity and time and/or space is sufficient for their application. Co-processing can be very useful here.

Regarding future work in Kombyne™, efforts are underway to support high order element types [12], flexible topologies whereby multiple solvers can communicate with a single Endpoint, direct support for reduced order model production and consumption and GPU-resident operation.

6 Acknowledgements

The Kombyne™ software is based upon work supported by the Department of Energy under Award Number DE-SC0018633. Phase I & Phase II SBIR funding was used for this development with subsequent ‘Phase III’ support provided by RRD via the PRESTIGE program, NASA & JAXA.

The cylinder in crossflow work was supported partially through RRD for another German research program known as DARWIN, funding provided by RRD and the German government.

For the aneurysm work, CFD Consultants acknowledges support from:

For the turbomachinery work, RRD states: The work presented in this paper was conducted within the framework of the PRESTIGE research project (20T1716A), funded by Rolls-Royce Deutschland Ltd & Co KG and the Bundesministerium für Wirtschaft und Energie. Rolls-Royce Deutschland’s permission to publish this work is greatly acknowledged.

References

[1] R. Thümmler. *Fast automatic simulation pipeline for aneurysms*. 9th OpenFOAM Conference, Held virtually, October 2021.

[2] M. Meyer et al. *Computing 3D CFD Compressor Maps of a Multi-Stage Axial Compressor using Computational Steering,* AIAA SciTech 2022, https://doi.org/10.2514/6.2022-1723.

[3] S. Tsutsumi, N. Fujita, H. Ito and D. Obinata, *Best practice for in-situ visualization of large-scale data in supercomputer*, Parallel CFD2017 at University of Strathclyde, 2017.

[4] B. Whitlock, J. M. Favre and J. S. Meredith, *Parallel In Situ Coupling of a Simulation with a Fully Featured Visualization System*, in Eurographics Symposium on Parallel Graphics and Visualization, Llandudno, Wales, UK, 2011.

[5] U. Ayachit, A. Bauer, B. Geveci, P. O'Leary, K. Moreland, N. Fabian and J. Mauldin, *ParaView Catalyst: Enabling In Situ Data Analysis and Visualization*, in ISAV 2015, Austin, TX, 2015.

[6] L. Biven, *ASCR Workshop on In Situ Data Management*, Rockville, MD, 2019.

[7] B. Whitlock, E. P.N. Duque, C. Langanella, *Scoreboard: Management and Creation of In Situ and In Transit Data Extractions via Computational Steering*, AIAA SciTech 2021, https://doi.org/10.2514/6.2021-1599

[8] J. Ahrens, S. Jourdain, P. O'Leary, J. Patchett, D. Rogers and M. Petersen, "*An image-based approach to extreme scale in situ visualization and analysis*," in IEEE Supercomputing, New Orleans, 2014.

[9] Ackerman, J. R., Gostelow, J. P., Rona, A., & Carscallen, W. E. (2009). *“Measurements of fluctuating pressures on a circular cylinder in subsonic cross flow.”* AIAA journal, 47(9), 2121-2131.

[10] Xiang et al.; *Hemodynamic-Morphologic Discriminants for Intercranial Aneurysm Rupture*; 2015

[11] Janke, C., Goller, M., Martin, I., Gaun, L., and Bestle, D., “*3D CFD compressor map computation of a multi-stage axial compressor with off-design adjusted rotor geometries*,” ASME Turbo Expo 2016: Turbomachinery Technical Conference and Exposition, American Society of Mechanical Engineers Digital Collection, 2016

[12] "*FMS: CEED's high-order Field and Mesh Specification*," 14 12 2020. [Online]. Available: https://github.com/CEED/FM