

ERDC MP-21-3

Engineer Research and
Development Center



**US Army Corps
of Engineers®**
Engineer Research and
Development Center



In Situ Analysis and Visualization to Enable Better Workflows with CREATE-AV™ Helios

Andrew C. Bauer, James R. Forsythe, Jayanarayanan Sitaraman,
Andrew Wissink, Buvana Jayaraman, and Robert B. Haehnel

June 2021

The US Army Engineer Research and Development Center (ERDC) solves the nation's toughest engineering and environmental challenges. ERDC develops innovative solutions in civil and military engineering, geospatial sciences, water resources, and environmental sciences for the Army, the Department of Defense, civilian agencies, and our nation's public good. Find out more at www.erdclibrary.on.worldcat.org/discovery.

To search for other technical reports published by ERDC, visit the ERDC online library at <http://www.erdclibrary.on.worldcat.org/discovery>.

In Situ Analysis and Visualization to Enable Better Workflows with CREATE-AV™ Helios

Andrew C. Bauer

*US Army Engineer Research and Development Center
Information Technology Laboratory
3909 Halls Ferry Road
Vicksburg, MS 39180*

Robert B. Haehnel

*Cold Regions Research Engineering Laboratory
7 Lyme Road
Hanover, NH 03755*

James R. Forsythe

*US Navy NAVAIR 4.3.2.1
48110 Shaw Road
Patuxent River, MD 20670*

Janarayanan Sitaraman

*Parallel Geometric Algorithms LLC
Sunnyvale, CA 94087*

Andrew Wissink and Buvana Jayaraman

*US Army CCDC AvMC
Technology Development Directorate
Moffett Field, CA 94035*

Final report

Approved for public release; distribution is unlimited.

Prepared for U.S. Army Corps of Engineers
Washington, DC 20413

Under Program Element Number 0212040, AMSCO 644115AX400

Preface

This study was conducted for the Department of Defense (DoD) High Performance Computing Modernization Program (HPCMP). Funding for Computational Research and Engineering (CPE) was provided under Program Element 0212040, AMSCO 644115AX400 by the Deputy Assistant Secretary of the Army for Research and Technology and supported by the Future Vertical Lift Cross-Functional Team.

The work was performed by the Data Analysis and Assessment Branch (Dr. Michael M. Stephens, Chief) of the Supercomputing Research Center Division (Mr. Robert M. Hunter, Chief), U.S. Army Engineer Research and Development Center, Information Technology Laboratory (ERDC-ITL); and by the Terrestrial and Cryospheric Sciences Branch (Dr. John Weatherly, Chief), of the Research and Engineering Division (Mr. David Ringleberg, Acting Chief), U.S. Army Engineer Research and Development Center, Cold Regions Research Engineering Laboratory (ERDC-CRREL). At the time of publication, the Deputy Director of ERDC-ITL was Ms. Patti S. Duett, and the Director was Dr. David A. Horner. The Deputy Director of ERDC-CRREL was Mr. David B. Ringelberg, and the Director was Dr. Joseph L. Corriveau.

This paper was originally presented at the *American Institute of Aeronautics and Astronautics Scitech 2021 Forum* (Virtual Event, 19-21 January 2021) and published online 4 January 2021. This work was made possible by the HPCMP's allocation of computer time on the ERDC DSRC's Cray XC40, Onyx.

The Commander of ERDC was COL Teresa A. Schlosser and the Director was Dr. David W. Pittman.

DISCLAIMER: The contents of this report are not to be used for advertising, publication, or promotional purposes. Citation of trade names does not constitute an official endorsement or approval of the use of such commercial products. All product names and trademarks cited are the property of their respective owners. The findings of this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

DESTROY THIS REPORT WHEN NO LONGER NEEDED. DO NOT RETURN IT TO THE ORIGINATOR.

***In Situ* Analysis and Visualization to Enable Better Workflows with CREATE-AVTM Helios**

Abstract

The CREATE-AVTM Helios CFD simulation code has been used to accurately predict rotorcraft performance under a variety of flight conditions. The Helios package contains a suite of tools that contain almost the entire set of functionality needed for a variety of workflows. These workflows include tools customized to properly specify many *in situ* analysis and visualization capabilities appropriate for rotorcraft analysis. *In situ* is the process of computing analysis and visualization information during a simulation run before data is saved to disk. *In situ* has been referred to with a variety of terms including co-processing, covisualization, coviz, etc. In this paper we describe the customization of the pre-processing GUI and corresponding development of the Helios solver code-base to effectively implement *in situ* analysis and visualization to reduce file IO and speed up workflows for CFD analysts. We showcase how the workflow enables the wide variety of Helios users to effectively work in post-processing tools they are already familiar with as opposed to forcing them to learn new tools in order post-process *in situ* data extracts being produced by Helios. These data extracts include a various sources of information customized to Helios, such as knowledge about the near- and off-body grids, internal surface extracts with patch information, and volumetric extracts meant for fast post-processing of data. Additionally, we demonstrate how *in situ* can be used by workflow automation tools to help convey information to the user that would be much more difficult when using full data dumps.

Acronyms

API Application Programmer's Interface.

CFD Computational Fluid Dynamics.

CPE Computational Prototyping Environment.

CSD Computational Structural Dynamics.

GUI Graphical User Interface.

HIGen Helios Input Generator.

HPC High Performance Computing or High Performance Computer.

HUI Helios User Interface.

MPI Message Passing Interface.

VTK Visualization Toolkit.

XML Extensible Markup Language.

I. Introduction

As compute power grows, workflows for rotorcraft simulations are getting more complex and demanding. This is especially true for workflows that include high performance computing since both the fidelity of a simulation can increase while simultaneously multi-experiment runs are becoming more common to providing more encompassing information to the problem at hand.

For high fidelity single simulation runs, the data output can be extremely expensive to fully store and examine in a *post hoc* fashion. For high fidelity rotorcraft simulations the time-dependent nature of the flow requires many time steps, often many thousands to potentially tens of thousands for complex maneuvers. The typical method to reduce file IO, and thus reduce simulation run time, is to output full data dumps at a reduced temporal frequency. The risk with this method though is if the physics of interest occurs at a higher frequency than the data dump frequency and thus not all of the useful information will be available for *post hoc* analysis.

For multi-experiment runs such as parameter sweeps, uncertainty quantification, etc., often times the quantity of interest is not a single flow visualization but a summary report. In fact if there are enough runs then it becomes impossible to view results for each run in a timely fashion. Still though, the user may want to view outlier results whether they are the best results for optimization runs or worst results for failure analysis. Tools like Slycat [1] exist that are designed to view multi-experiment results by storing representative images for each experiment run.

While there are several general workflow tools available [2–4] that are focused on mesh-based, PDE solver workflows, it seems that there is enough variety in workflows and users' needs that each platform has some deficiency to necessitate more work in this area. For rotorcraft analysis this extension or customization comes in the form of multi-experiment runs to analyze performance characteristics over different flight envelopes such as maneuvers, hover, different design variations, various environmental conditions, etc. Beyond this, there are additional simplifications to reduce compute time that can also be added to workflows. Examples include using simplified geometry, blade element theory instead of full fluid structure interaction, actuator disk, etc. Adding to the complexity of these various rotorcraft analysis cases, there is also the question of user access to compute resources, choice of pre- and post-processing tools, and monitoring of computational jobs. All of these parts of workflow can be designed to work well together or can be developed separately and this is often reflected in the user experience and in their success or failure in using the simulation ecosystem. In this paper we discuss how *in situ* analysis and visualization fits in Helios's workflow ecosystem through a design that has evolved through much effort, several software releases and user feedback.

II. Background

In situ analysis and visualization has been around for over a half a century having first been done by Zajac in 1964 [5]. In the 1990s there were enough *in situ* systems developed that two survey papers were written [6, 7]. The interest for *in situ* waned for a bit until general production needs increased and new systems needed to be developed that scaled to higher levels of parallelism while also providing more robust functionality. This is when ParaView [8] and VisIt [9] began to add *in situ* capabilities to their software via their Catalyst [10] and Libsim [11] packages, respectively. While these new tools provided the needed *in situ* capabilities for simulation codes, the workflow for the domain expert was a step backwards from many of the bespoke systems previously available. This is because configuring the *in situ* output required extra effort on the part of the domain expert and this was not set up to work conveniently within their workflow.

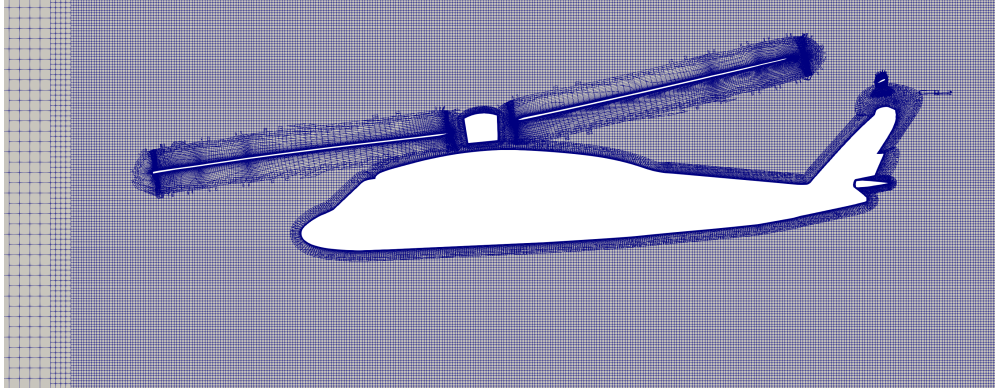


Fig. 1 Zoomed in view of Helios dual grid mesh with AMR off-body and body-fitted near-body grids.

Additionally, loading data extract output from these infrastructures was initially limited to certain post-processing tools that supported those file formats.

A. CREATE-AVTM Helios

CREATE-AVTM Helios, or Helios for short, is a high-fidelity rotorcraft analysis package developed collaboratively over the past decade by the US Army and the DoD High Performance Computing Modernization Program (HPCMP) CREATE program. Helios models the coupled aerodynamic and structural dynamics response of the vehicle using Computational Fluid Dynamics (CFD) codes for aerodynamics, Computational Structural Dynamics (CSD) to model structural bending, and trim software with multi-body dynamics to set the controls to achieve vehicle trim. In order to accurately resolve both the viscous turbulent flow near the wall and the wake that impacts interactional rotor-rotor and rotor-fuselage aerodynamics, Helios CFD uses a dual-mesh paradigm with a body-fitted near-body grid and a Cartesian AMR off-body grid for the wake. An example of the dual-mesh paradigm is shown in Figure 1.

Unlike self contained single-language simulation codes, Helios utilizes a Python-based object-oriented infrastructure to link together different multi-physics components: near- and off-body CFD solvers, CSD solver, overset connectivity package, multi-body dynamics, etc. Each component can be written in its own desired language and wrapped by a thin layer of Python code that maintains pointers to shared data and function APIs. The infrastructure orchestrates operations between the underlying components in order to carry out the simulation desired by the user. This infrastructure-based approach makes integration of *in situ* visualization straightforward. The visualization libraries can be readily integrated in the same fashion as the underlying physics solvers, using the same data. Currently at version 11.0, Helios includes a legacy and a cutting-edge pre-processing GUI, automated mesh generation, parallel near- and off-body solvers, and several post-processing scripts to help with common analysis operations. The production-quality code has more than 100 licensed users across the DoD, U.S. universities, and various U.S. aerospace companies. For this paper we will only discuss the parts of Helios that directly relate to the *in situ* processing and refer users to other references [12–14] for more detailed background information.

B. ParaView Catalyst

ParaView Catalyst [10] is the production-level *in situ* library that is integrated with Helios. It is used in-line, meaning that it is called directly from the simulation and uses the same compute resources as the simulation. The *in situ* processing can be run in either batch mode via C++ or Python code or with a human-in-the-loop by connecting to the ParaView GUI to modify the Python code operations. The most general use case is for users to set up their desired analysis and visualization output in the ParaView GUI and then export those operations in a Python script for use during the simulation run. Alternatively, developers familiar with the ParaView code base can program parameterized output based on a user's input to provide specific *in situ* analysis and visualization routines. To instrument a simulation code to work with Catalyst, the following three calls must be made from the simulation code:

initialize This initializes Catalyst and sets up all *in situ* processing. This is usually called soon after MPI is initialized for parallel simulation codes.

coprocess This is where the *in situ* analysis and visualization processing occurs and typically is called after each

time step. This does not imply that *in situ* processing occurs at each time step, only that the simulation checks with the *in situ* processing operations whether something needs to be done. If something does need to be computed, only then does this call take any significant compute resources for a given time step.

finalize This cleans up all Catalyst operations. This is usually called right before MPI is finalized for parallel simulation codes.

Besides doing the actual *in situ* computations in the **coprocess** call, this is also where the simulation data structures are copied into data structures that Catalyst understands. Since Helios uses Python as the main control language to efficiently manage the different CFD solvers and other libraries operating underneath the covers, we also use Python as the main control language for copying the Helios grid and field data structures into Catalyst grid and field data structures. For the fields in Helios, we are able to use NumPy to conveniently and efficiently copy to Catalyst data structures but for the unstructured and curvilinear grids we use Python-wrapped C++ code for computational efficiency.

III. User Workflows

Due to the wide variety of analysis requirements for rotorcraft, there are a variety of workflows that Helios is a key component of. The simplest is of a single design along with a single flight path. The complexity can grow to include variations on rotorcraft design, flight path, altitude, ambient conditions, mesh resolution, CFD solver choice, turbulence modeling, etc.

A. Basic Workflow

The basic workflow is initiated through Helios's pre-processing GUIs. The legacy GUI is called HUI and the new, more functional GUI is called HIGen. The basic workflow involves creating a mesh and then using one of the Helios pre-processors to create input files for a Helios simulation. After this, running the simulation, usually on an HPC machine, and then examining the output files to gain insight into the problem at hand. For Helios there are a variety of output files available including full data dumps, *in situ* extracts, log files and time-series information. In Helios version 11.0 the following *in situ* output is available, along with the required user information that needs to be specified during the pre-processing step in their workflow:

- Internal boundary surface extract which outputs the geometry of the rotorcraft along with the fields defined over that geometry. The user must specify the output frequency of this data extract.
- Cartesian interpolated extracts with chosen fields. The user must specify the geometry of the Cartesian grid (minimum and maximum coordinate and number of points in each direction), aggregation for file IO, output frequency, time averaging option with the frequency and what iteration to start time averaging the output.
- Cutting plane extracts. The user must specify an output filename, normal, origin, output frequency, set of offsets and optional sphere limiter to reduce the cut plane output to only be within that sphere volume.
- Iso-surface, or contour, extracts. The user specifies the field, output frequency, iso-surface value(s) and output filename. The iso-surface fields can be primitive or derived fields.
- Streamline extracts. The user specifies a set of seeds (point, line or point cloud), number of seeds, output frequency, and output filename.
- Taps extract, or sensor location extracts. The user specifies a set of locations and tap output frequency and automatically generates a .csv file for each tap location.
- Particle tracking extracts. The user specifies the same seed types as the streamline extracts, an output frequency, an output filename, and a particle reinjection frequency (0 for no reinjection). However, unlike other extracts which are only relevant to a particular time step, particle extracts must store both the current and previous times for integration and write out this information for restart.
- Volumetric grid extracts to examine the fields and near-body, off-body, or combined near-body and off-body grids the way the *in situ* processing computes over them. This is more useful as a debugging feature.

In addition to those specific data extracts, there are the global *in situ* options for Helios which are:

- Which iteration to begin outputting the *in situ* results at. Due to the complexity of the flow field and the difficulty in prescribing accurate initial conditions for rotorcraft, generally the first two or three full rotorblade rotations of the simulation are used to get rid of start up transients. Any *in situ* output through these iterations are usually wasted computations and the option to start outputting after this initial start up phase is useful to save compute time.

- Output format can be chosen to be legacy VTK XML format, aggregated VTK XML format or FieldView format. The difference between the legacy VTK XML format and the aggregated VTK XML format is that the aggregated VTK XML format produces less files by aggregating the data onto a smaller number of MPI processes. For situations with high MPI process counts and time iterations and/or large numbers of runs this can be a very important factor on HPC machines. As Helios developers we want to reach as many users as possible and allowing users to choose which post-processing tool best fits their workflow allows them to focus on their work instead of on learning new software. Figure 2 shows the same Helios boundary extract in a variety of post-processing packages.
- In Helios version 11.0 we added the ability to choose which fields to output. This provides the user with the convenience of computing expensive quantities *in situ*. Additionally, fields that are not needed *post hoc* do not have to be saved to disk, saving file IO.

The HIGen output section which includes *in situ*, or coviz in Helios terminology, is shown in Figure 3. This allows the user to set up the *in situ* processing just as they would do for any other simulation operations during a Helios run. Additionally, the *in situ* processing parameters are treated just as any other Helios input options would be so they can be archived, shared and easily reused in separate simulation runs.

IV. *In Situ* Integration

The *in situ* computations that are done within Helios are a combination of using the ParaView Catalyst library and bespoke integration. As previously discussed, there are two parts to the *in situ* integration with ParaView Catalyst. The first part is adapting the simulation data structures into data structures that Catalyst understands. The second part is the driver code that takes in the user input and runs the Catalyst *in situ* algorithms to output the requested data extracts. While most of the data extract integration was straightforward, the particle paths extracts, Cartesian grid extracts and field output selection bear some extra description due to their integration complexity.

A. Particle Paths Extracts

All of the other data extracts, described in Section III.A, that are computed inside of the Catalyst library are single time step algorithms. They are essentially stateless and do not require information from any previous time step to be computed. Particle paths are different in that they require both the previous and current time step (the Catalyst particle path time stepping algorithm only requires two time steps). Particle tracking in unsteady problems is particularly difficult with full-volume visualization because multiple timesteps of data must be stored on disk, only to be loaded into memory later by the *post hoc* tool that computes the particle paths. This problem is largely alleviated in *in situ* algorithms because the volume data is already in memory and never needs to be written to disk, but the implementation still requires some special considerations. The first is that the *in situ* algorithm must cache the previous time step grid and fields since it cannot rely on the simulation to store that information. Second is that for restarted simulations, the state of the particle path algorithm must also be restarted. Thus, whenever Helios outputs restart information the particle paths must participate in this action as well so that the proper particle path restart data is also available if a simulation restart is to be performed. On a simulation restart, the *in situ* particle path algorithm essentially seeds the paths with the particles from the restart file as well as making sure that subsequent seeds get reinjected at the appropriate time step. This is much finer grain control than is normally available with particle paths in post-processing applications. The advantage with the *in situ* particle path IO though is that only point-wise information is being stored instead of volumetric information, which is a huge reduction in size.

B. Cartesian Grid Extracts

In Helios version 10.1 we developed Cartesian grid extracts to provide efficient volumetric extracts over a portion of the domain. We implemented the *in situ* algorithm to interpolate the fields onto the Cartesian grid using internal Helios algorithms but utilized the VTK writers that are packaged with ParaView Catalyst to output the data. The advantage of this bespoke implementation is that due to the dual grid nature employed in Helios, grid search structures and interpolation are already implemented and constructed for use and this can be taken advantage of for interpolating onto the Cartesian grid. The advantage of using the VTK writers that are packaged with Catalyst is that they are well tested, efficient, portable, and can be easily swapped out when more efficient writers are developed inside of VTK. Additionally, this reduces the development burden on the Helios development team. The advantage of using Cartesian grid output from Helios is that the grid information is compact and essentially all of the storage cost is related to storing

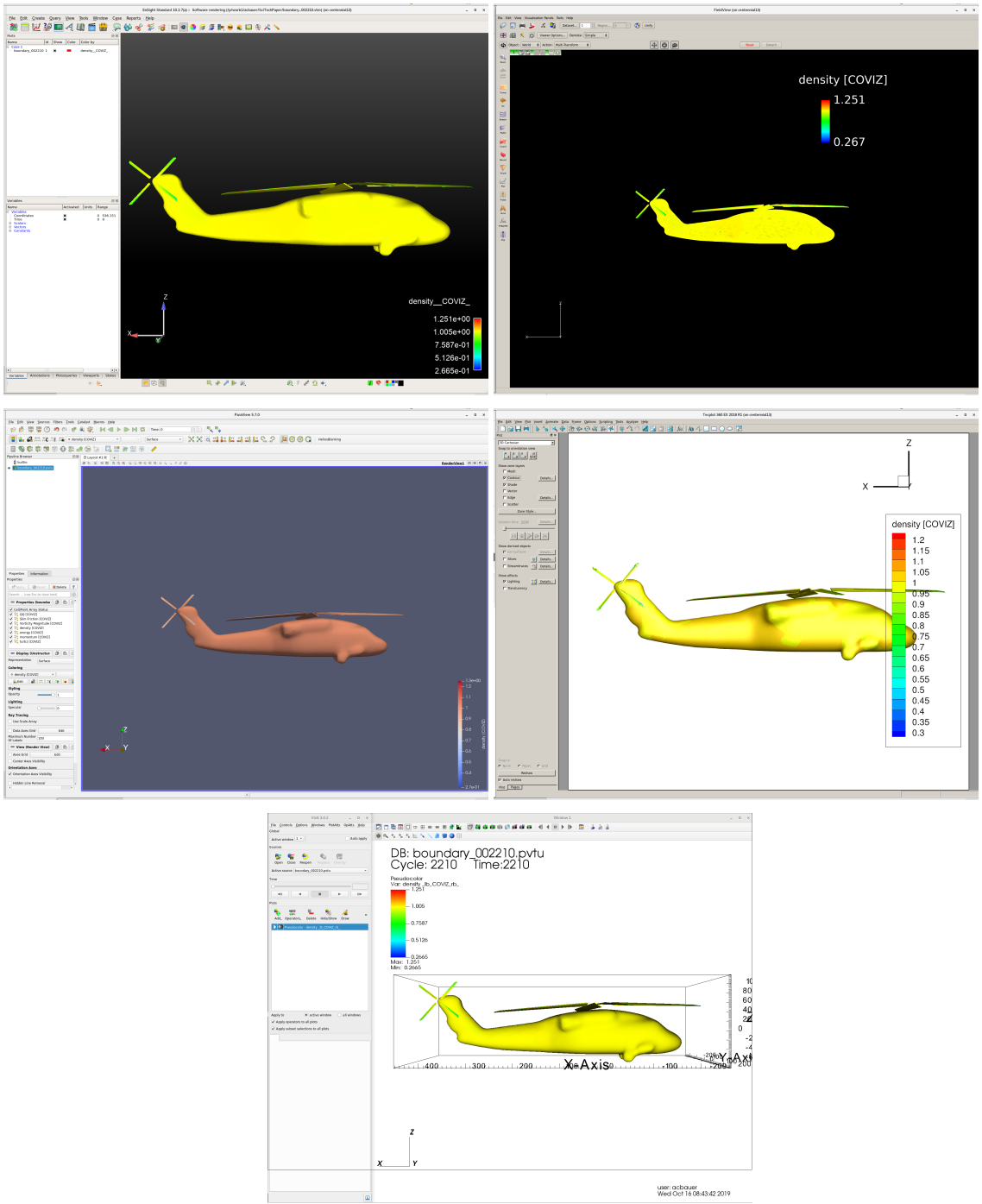


Fig. 2 Images for the boundary extract for EnSight Standard 10.2.7, FieldView 17, ParaView 5.7, Tecplot360 2018R1, and VisIt 3.0.2.

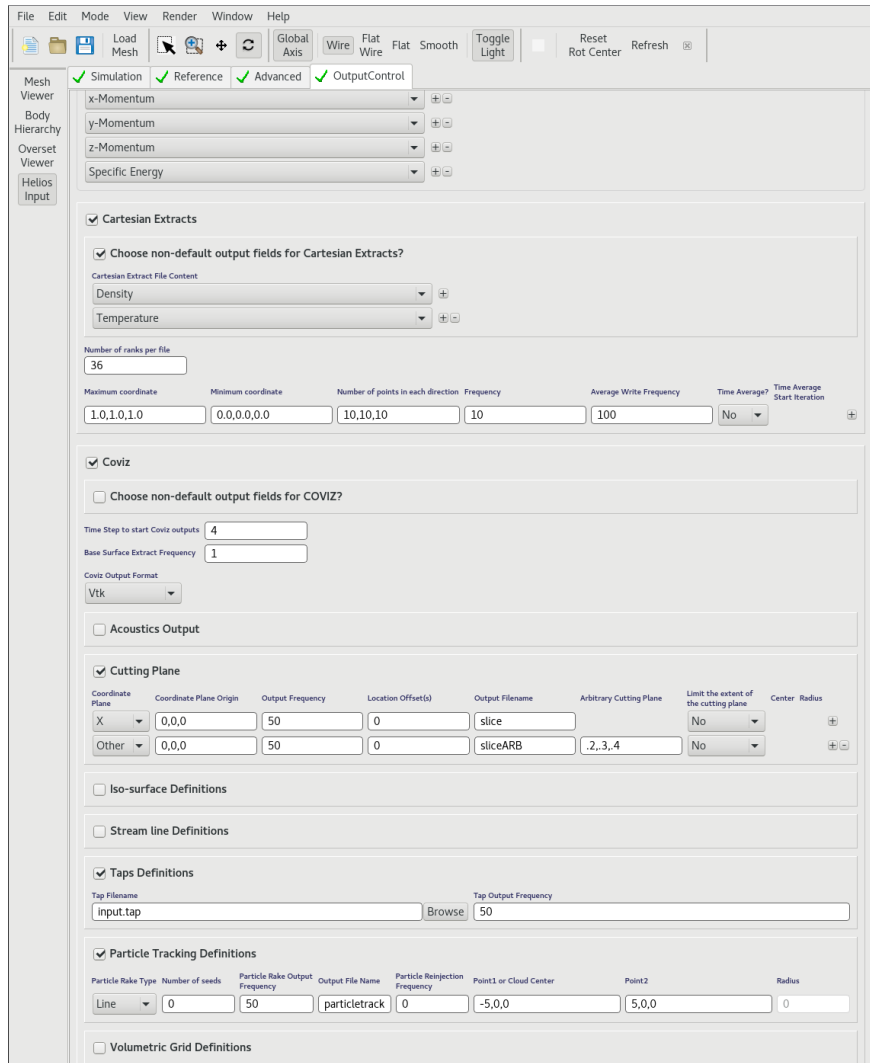


Fig. 3 Helios HIGen preprocessor showing all output options including coviz.

selected fields, which the user specifies. Additionally, many analysis and visualization operations are significantly faster on Cartesian grids due to the simple grid structure.

C. Field Output Selection

To reduce *in situ* processing as well as file IO, Helios allows the selection of fields to be written out in the data extracts. Before Helios version 11.0 this was done in an ad hoc manner. As users' needs grew, the output fields expanded beyond the original fields of density, momentum, energy, Q-criterion, and vorticity magnitude to also include turbulence quantities, temperature and skin friction over the boundary surface extract. Outputting temperature in the extracts was an optional choice and this could be done in dimensional or non-dimensional form. In Helios version 11.0 we extended this to being able to explicitly choose whether each available field variable is written out in all extracts or not. This includes both primitive and derived solution quantities in both non-dimensional and dimensional forms. Quantities like Q-criterion that require gradient information, and thus grid information, must be computed over the volumetric grid before the extract geometry is computed. Derived fields that are simple algebraic computations like velocity, enthalpy, etc., are computed over the extract geometry, e.g. the cutting plane, to reduce computational cost and memory overhead. After all of the requested derived fields are computed over the extracts, all non-requested fields are then excluded at the next stage of the processing pipeline before the writing stage of the *in situ* processing in order to ensure that only the requested fields are saved to disk.

V. In Situ For Complex Workflows

As mentioned earlier, for more complex workflows like collective sweeps, tradespace analysis, etc. we expect that the user will often not view the results of many of the individual simulation runs. At best, the user may look at highlight images of the run. The advantage that *in situ* brings here is that the user has already decided what is important to view during the problem set up. They will choose iso-surface fields and values, cut plane setups, and/or particle path extracts to go along with the surface extracts for later use. When trying to provide summary information for a run for quick perusal we only need to provide viewing mechanisms for these extracts instead of requiring the user to perform a full analysis on a full volume dataset.

A. Computational Prototyping Environment

The Computational Prototyping Environment [15], or CPE for short, is a web-based workflow platform. It is designed to work with Helios to assist with managing and monitoring jobs on HPC machines. The monitoring of jobs includes both accessing HPC queue system information as well as Helios output information. CPE currently includes three multi-experiment job launch capabilities with more planned in the future. The current three are collective sweep for hover performance analysis, speed sweep for forward flight performance analysis and ensemble analysis for turbulence model and grid sensitivity analysis. While not intended to replace all of the pre-processing parts of the workflow, CPE currently provides collective and speed sweeps multi-experiment workflows with more planned for in future releases. CPE's web-based interface provides authenticated access to the DoD's Distributed Supercomputing Resource Centers (DSRCs) and a rich environment that allows launching new HPC jobs, monitoring running jobs and examining output from completed jobs. The Helios output that can be viewed from CPE include:

- Log files which show informational and error messages that Helios generates during a simulation run.
- Time series tracking files of such information as model forces and coefficients.
- *In situ* output of cut plane, iso-surface and internal boundary data extracts.

For the *in situ* extracts that can be viewed in CPE, the user first chooses which experiment point and time step to view extracts for. Given these, the user then selects from a list of extracts available at that time step. Note that a subset of the cut plane, iso-surface and internal boundary data extracts may be available at that time step based on the user's configuration for that simulation run. Once the extracts are chosen for that time step then the user selects which fields to use for pseudo-coloring. If no fields are chosen then a default grey color is used. Several preset viewing directions are available as well to help the user inspect the results. An example of this is shown in Figure 4. The goal of the CPE visualization is not for publication worthy images but to give the user quick and easy access to the current status of the simulation to determine if any obvious errors have occurred.

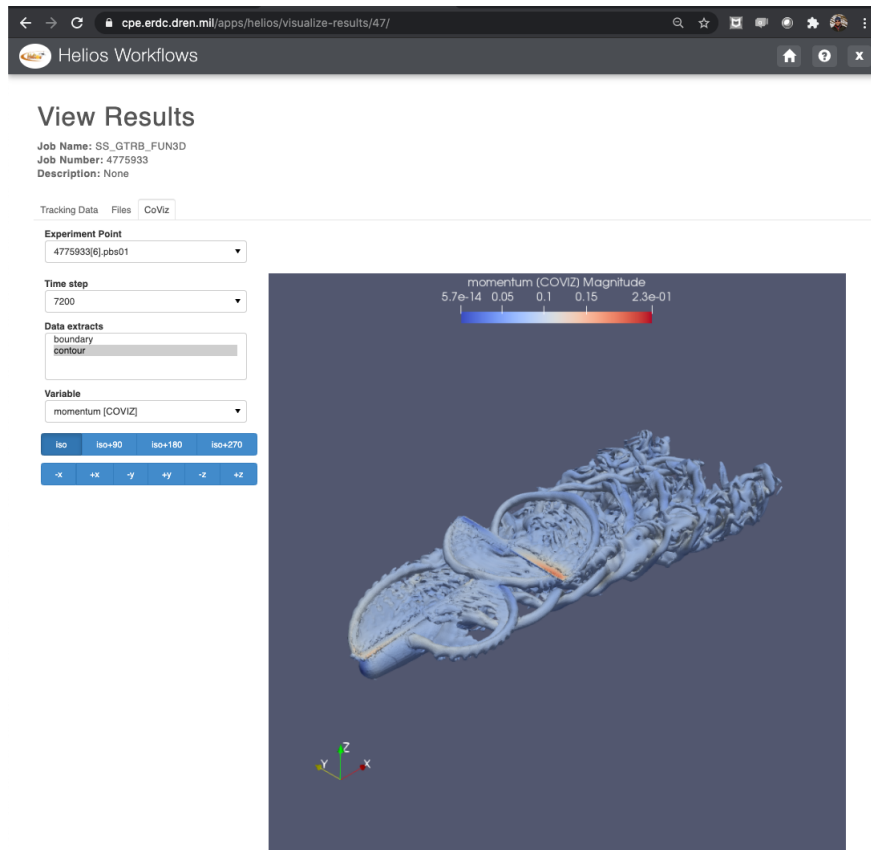


Fig. 4 Computational Prototyping Environment (CPE) platform presenting *in situ* output from a Helios run.

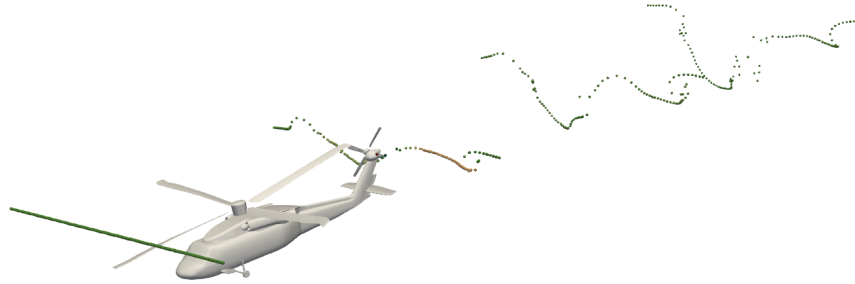


Fig. 5 Sample *in situ* particle path and internal surface output of an idealized UH-60 rotorcraft simulation.

VI. Results

Quantifying workflow simplification and automation tools is a difficult to impossible task, especially considering the variety of users that Helios supports as well as the tasks that Helios is used for. For *in situ* research the following are often used as stand-in metrics for comparisons:

Time to Insight: Here a comparison is made between the total time to compute a data product such as an image of a pseudo-colored iso-surface or a cut plane extract *in situ* and performing a full data dump and then computing that same data product *post hoc*. It ignores HPC queue times and other parts of the workflow such as the time to set up the *in situ* or *post hoc* operations as that is often user dependent.

Data Output Size: Here a comparison is made between full data dumps, which is often at lower frequencies, and data extract dumps at higher frequencies. This can give an idea of the data that the user may need to manage during post-processing but doesn't give any indication of time saved using *in situ* methods compared to working with full data dumps.

Even though data output size is not a perfect measure of the usefulness of *in situ* to our workflows, it can be inferred and is an easy measure. We examine results from an idealized geometry of a UH-60 rotorcraft in forward flight. We ran Helios on 880 MPI processes on the U.S. Department of Defense's Onyx HPC machine. We computed internal surface, cut plane and particle path extracts. The cut plane was chosen such that it was aligned with the flight path and bisected the UH-60 in half. The internal surface and cut plane extracts were chosen since their output file size are proportional to the full grid size while iso-surface output file size can be easily manipulated by choosing convenient contour values. For the particle path extracts, 100 particle seeds were reinjected every 1,440 time steps and data was output every 72 time steps. The particle path locations were updated after every simulation time step. The simulation was run for 5,647 iterations and shutdown automatically via queue time restrictions. Sample output of the internal surface and particle paths at time step 4,320 is shown in Figure 5. For file size comparison, the full data dump size after the first iteration was 19.7 GB. A single cut plane output was 61 MB and the internal boundary extract was 141 MB. The particle path extracts for 80 different outputs came to a total of 9.1 MB.

VII. Summary

We have provided an overview of the *in situ* processing in Helios and how it is used for a variety of workflows. We have demonstrated that the workflows do not require knowledge of any specific post-processing tool which reduces the barrier to entry for users. The benefits to users have been key in both improved analysis of rotorcraft and reduced analysis time. Future work includes a better user experience through more efficient *in situ* computation, reduced file IO time for extracts, improved integration with HIGen pre-processing tool to help visualize where particle path seeds, streamline seeds and cut planes are being placed, and a run-time update capability to modify certain *in situ* parameterized outputs during a simulation run. For CPE, planned improvements include adding in the capability to modify Helios's coviz output during a simulation run.

VIII. Acknowledgements

Material presented is a product of the CREATE (Computational Research and Engineering for Acquisition Tools and Environments) element of the U.S. Department of Defense HPC Modernization Program.

Funding for CPE was provided by the Deputy Assistant Secretary of the Army for Research and Technology and

supported by the Future Vertical Lift Cross-Functional Team. This work was performed by the U.S. Army Engineer Research and Development Center under the Computational Prototyping Environment program.

This work was made possible by the Department of Defense High Performance Computing Modernization Program's allocation of compute time on the Engineer Research and Development Center's Department of Defense Supercomputing Resource Center's Cray XC40, Onyx.

References

- [1] Crossno, P., "Slycat Website," , 2020. URL <https://github.com/sandialabs/slycat>.
- [2] Brown, N., Nash, R., Gibb, G., Prodan, B., Kontak, M., Olshevsky, V., and Der Chien, W., "The Role of Interactive Super-Computing in Using HPC for Urgent Decision Making," *High Performance Computing*, edited by M. Weiland, G. Juckeland, S. Alam, and H. Jagode, Springer International Publishing, Cham, 2019, pp. 528–540.
- [3] Directed Energy Directorate of the Air Force Research Lab and Stellar Science Ltd Co, "Galaxy: A framework for multi-site, cross-platform simulation and optimization," , 2018. URL https://www.stellarscience.com/Galaxy_White_Paper.pdf.
- [4] Kitware Inc., "Computational Model Builder Website," , 2020. URL <https://www.computationalmodelbuilder.org/>.
- [5] Zajac, E. E., "Computer-Made Perspective Movies as a Scientific and Communication Tool," *Communications of the ACM*, Vol. 7, No. 3, 1964, pp. 169–170. <https://doi.org/10.1145/363958.363993>.
- [6] Heiland, R., and Baker, M. P., "A Survey of Co-Processing Systems," , August 1998. URL <https://rheiland.github.io/coproc/CoprocSurvey.pdf>.
- [7] Mulder, J., Wijk, van, J., and Liere, van, R., "A survey of computational steering environments," *Future Generation Computer Systems*, Vol. 15, No. 1, 1999, pp. 119–129. [https://doi.org/10.1016/S0167-739X\(98\)00047-8](https://doi.org/10.1016/S0167-739X(98)00047-8).
- [8] Ahrens, J., Geveci, B., and Law, C., "ParaView: An End-User Tool for Large-Data Visualization," *The Visualization Handbook*, 2005.
- [9] Childs, H., Brugger, E., Whitlock, B., Meredith, J., Ahern, S., Pugmire, D., Biagas, K., Miller, M., Harrison, C., Weber, G. H., Krishnan, H., Fogal, T., Sanderson, A., Garth, C., Bethel, E. W., Camp, D., Rübel, O., Durant, M., Favre, J. M., and Navrátil, P., "VisIt: An End-User Tool For Visualizing and Analyzing Very Large Data," *High Performance Visualization—Enabling Extreme-Scale Scientific Insight*, 2012, pp. 357–372.
- [10] Fabian, N., Moreland, K., Thompson, D., Bauer, A. C., Marion, P., Geveci, B., Rasquin, M., and Jansen, K. E., "The ParaView Coprocessing Library: A Scalable, General Purpose In Situ Visualization Library," *Proceedings of the IEEE Symposium on Large-Scale Data Analysis and Visualization*, 2011, pp. 89–96. <https://doi.org/10.1109/LDAV.2011.6092322>.
- [11] Whitlock, B., Favre, J. M., and Meredith, J. S., "Parallel In Situ Coupling of Simulation with a Fully Featured Visualization System," *Eurographics Symposium on Parallel Graphics and Visualization*, 2011. <https://doi.org/10.2312/EGPGV/EGPGV11/101-109>.
- [12] Sankaran, V., Sitaraman, J., Wissink, A., Datta, A., Jayaraman, B., Potsdam, M., Mavriplis, D., Yang, Z., O'Brien, D., Saberi, H., Cheng, R., Hariharan, N., and Strawn, R., *Application of the Helios Computational Platform to Rotorcraft Flowfields*, chapter and pages. <https://doi.org/10.2514/6.2010-1230>, URL <https://arc.aiaa.org/doi/abs/10.2514/6.2010-1230>.
- [13] Sitaraman, J., Potsdam, M., Wissink, A., Jayaraman, B., Datta, A., Mavriplis, D., and Saberi, H., "Rotor Loads Prediction Using Helios: A Multisolver Framework for Rotorcraft Aeromechanics Analysis," *Journal of Aircraft*, Vol. 50, No. 2, 2013, pp. 478–492. <https://doi.org/10.2514/1.C031897>, URL <https://doi.org/10.2514/1.C031897>.
- [14] "A Dual-Mesh Unstructured Adaptive Cartesian Computational Fluid Dynamics Approach for Hover Prediction," *Journal of the American Helicopter Society*, Vol. 61, No. 1, 2016.
- [15] Haehnel, R. B., Christensen, S. D., Whitlow, J. L., Bauer, A. C., Meyer, A., Rangarajan, G., Wenren, Y., Harden, D., Hoch, B., Clark, S., and Eiseman, A., "A Computational Prototyping Environment interface for DoD CREATETM Helios Simulations," *AIAA Scitech 2021 Forum*, 2021 in press.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE (DD-MM-YYYY) June 2021		2. REPORT TYPE Final		3. DATES COVERED (From - To)	
4. TITLE AND SUBTITLE In Situ Analysis and Visualization to Enable Better Workflows with CREATE-AV™ Helios				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER 0212040A	
6. AUTHOR(S) Andrew C. Bauer, James R. Forsythe, Jayanarayanan Sitaraman, Andrew Wissink, Buvana Jayaraman, and Robert B. Haehnel				5d. PROJECT NUMBER 485610	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) See next page.				8. PERFORMING ORGANIZATION REPORT NUMBER ERDC MP-21-3	
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES) US Army Corps of Engineers Washington, DC 20314				10. SPONSOR/MONITOR'S ACRONYM(S) USACE	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION / AVAILABILITY STATEMENT Approved for public release; distribution is unlimited.					
13. SUPPLEMENTARY NOTES This paper was originally presented at the American Institute of Aeronautics and Astronautics Scitech 2021 Forum (Virtual Event, 19-21 January 2021) and published online 4 January 2021. Additional funding information AMSCO: 644115AX400					
14. ABSTRACT The CREATE-AV™ Helios CFD simulation code has been used to accurately predict rotorcraft performance under a variety of flight conditions. The Helios package contains a suite of tools that contain almost the entire set of functionality needed for a variety of workflows. These workflows include tools customized to properly specify many in situ analysis and visualization capabilities appropriate for rotorcraft analysis. In situ is the process of computing analysis and visualization information during a simulation run before data is saved to disk. In situ has been referred to with a variety of terms including co-processing, covisualization, coviz, etc. In this paper we describe the customization of the pre-processing GUI and corresponding development of the Helios solver code-base to effectively implement in situ analysis and visualization to reduce file IO and speed up workflows for CFD analysts. We showcase how the workflow enables the wide variety of Helios users to effectively work in post-processing tools they are already familiar with as opposed to forcing them to learn new tools in order post-process in situ data extracts being produced by Helios. These data extracts include various sources of information customized to Helios, such as knowledge about the near- and off-body grids, internal surface extracts with patch information, and volumetric extracts meant for fast post-processing of data. Additionally, we demonstrate how in situ can be used by workflow automation tools to help convey information to the user that would be much more difficult when using full data dumps.					
15. SUBJECT TERMS High performance computing; Computer simulation; Prototypes; Engineering; Web-based user interfaces					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT Unclassified	b. ABSTRACT Unclassified	c. THIS PAGE Unclassified			19b. TELEPHONE NUMBER (include area code)

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Andrew C. Bauer

*US Army Engineer Research and Development Center
Information Technology Laboratory
3909 Halls Ferry Road
Vicksburg, MS 39180*

Robert B. Haehnel

*Cold Regions Research Engineering Laboratory
7 Lyme Road
Hanover, NH 03755*

James R Forsythe

*US Navy NAVAIR 4.3.2.1
48110 Shaw Road
Patuxent River, MD 20670*

Janarayanan Sitaraman

*Parallel Geometric Algorithms LLC
Sunnyvale, CA 94087*

Andrew Wissink and Buvana Jayaraman

*US Army CCDC AvMC
Technology Development Directorate
Moffett Field, CA 94035*