LA-UR-11-05546

Approved for public release;
distribution is unlimited.

Title: Hydra-TH L3 Milestone (THM.CFD.P3.02)

Author(s):
Mark A. Christon
Robert B. Lowrie
Nathan Barnett
Jozsef Bakosi
Marianne M. Francois

Intended for: CASL L3 Milestone POR3

Los Alamos National Laboratory, an affirmative action/equal opportunity employer, is operated by the Los Alamos National Security, LLC for the National Nuclear Security Administration of the U.S. Department of Energy under contract DE-AC52-06NA25396. By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes. Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy. Los Alamos National Laboratory strongly supports academic freedom and a researcher's right to publish; as an institution, however, the Laboratory does not endorse the viewpoint of a publication or guarantee its technical correctness.

Form 836 (7/06)

CASL-U-2011-0183-000

Hydra-TH L3 Milestone (THM.CFD.P3.02)

Mark A. Christon, Rob Lowrie, Nathan Barnett,
Jozsef Bakosi and Marianne Francois
Los Alamos National Laboratory
Los Alamos, NM 87544

With contributions from:
Emillian L. Popov, Oak Ridge National Laboratory

1 Executive Summary

This report describes the work carried out for completion of the Thermal Hydraulics Methods (THM) Level 3 Milestone THM.CFD.P3.02 for the Consortium for Advanced Simulation of Light Water Reactors (CASL). The Level 3 milestone was comprised of 8 sub-tasks that included: 1) initial integration of Hydra-TH and Hydra on LANL/ORNL HPC compute resources, 2) generation of initial results on test problems, 3) establishing development guidelines, 4) implementation of periodic BC's, 5) extension of testing infrastructure, 6) generation of GTRF results (contingent on obtaining SNL meshes), 7) assessment of documentation, 8) modification of the VERA-CFD plan as appropriate.

A brief overview of Hydra-TH is presented, and followed by a detailed summary of each of the 8 milestone tasks. At the time of this writing, the GTRF meshes from Sandia have just become available, and time has not permitted us to carry out any calculations on the GTRF. Instead, a series of problems from the Hydra-TH test suite are presented in §3.2 along with a scaling study in §3.6.2.

The problems presented from the Hydra-TH test suite indicate demonstrate second-order spatial convergence even on highly-skewed meshes. Additional data may be found in the Hydra-TH Theory Manual [7].

The scaling study presented in §3.6.2 indicates that there are some limitations with the Exodus-II file format and we are actively testing the upper-

CASL-U-2011-0183-000

bounds of the file-format to ensure that it can scale into the multi-billion element range. The parallel scaling demonstrated with Hydra-TH shows reasonable scaling into the 1024 processor range, and will be probed further on Jaguar in the near future. The scaling study did reveal some difficulties with the algorithmic scaling of the ML preconditioner – conjugate gradient combination that will be assessed further on Jaguar as well. These studies have also demonstrated that Hydra-TH can easily scale past the 10^8 element range. Preliminary testing in collaboration with the primary author of the Spider mesh generator indicates it can produce "analysis grade" meshes for the GTRF geometries, and scale into the 10^9 range. Future efforts will demonstrate the ability of Hydra-TH to use the high-resolution hybrid meshes produced by the Spider mesh generator.

2 Introduction & Background

Hydra-TH refers to the hybrid finite-element/finite-volume incompressible/low-Mach flow solver in the Hydra toolkit that is being used for CASL thermal-hydraulics applications. Hydra-TH is built as one of a number of virtual physics using the Hydra multiphysics toolkit. The Hydra toolkit is written in C++ and provides a rich suite of components that permits rapid application development, supports multiple discretization techniques, provides I/O interfaces to permit reading/writing multiple file formats for meshes, plot data, time-history, surface-based and restart output. Data registration is used to provide the ability to register variables at appropriate locations (e.g., node, element, dual-edge, etc), and provides integrated and automatic output and restart capabilities along with memory management. The toolkit also provides run-time parallel domain decomposition with data-migration for both static and dynamic load-balancing. Linear algebra is handled through an abstract virtual interface that makes it possible to use popular libraries such as PetSC and Trilinos. The use of output delegates provides the ability to develop lightweight physics-specific output kernels with minimal memory overhead that can be tailored to a specific physics, e.g., computation of vorticity, helicity, enstrophy for large-eddy simulations.

Hydra's toolkit model for development provides lightweight, high performance and reusable code components for agile development. Currently the toolkit supports finite-element based solvers for time-dependent heat conduction, time-dependent advection-diffusion, time-dependent incompressible

flow, multiple Lagrangian hydrodynamics solvers, rigid-body dynamics, etc. In addition, unstructured-grid finite-volume solvers are available for solving time-dependent advection-diffusion, Burgers' equation, the compressible Euler equations, and the hybrid incompressible/low-Mach Navier-Stokes equations.

2.1 Hydra-TH Overview

The Hydra-TH theory manual [7] presents the theoretical background for the hybrid finite-element/finite-volume incompressible/low-Mach flow solver based on the Hydra toolkit. By design, Hydra-TH was built with the idea of handling both single and multi-component flows. Although not currently used in Hydra-TH, the Hydra toolkit provides a number of interfaces for using interface reconstruction for volume-tracking, front-tracking (via FronTier [12]), and it is anticipated that these will be used in the future for CASL applications.

Hydra-TH uses a hybrid finite-element/finite-volume discretization for the incompressible/low-Mach Navier-Stokes equations. All transport variables are cell-centered and treated with a conservative discretization that includes a high-resolution monotonicity-preserving advection algorithm. The spatial discretization is formally derived using a discontinous-Galerkin framework that, in the limit, reduces to a locally-conservative finite-volume method. The high-resolution advection algorithm is designed to permit both implicit and explicit advection with the explicit advection targeted primarily at volume-tracking with interface reconstruction. The time-integration methods include backward-Euler and the neutrally-dissipative trapezoidal method. The coding for an optional BDF2 time-integrator has also been provided, but is not currently used in Hydra-TH. The implicit advective treatment delivers unconditional stability for the scalar transport equations, and conditional stability for the momentum transport equations. A sharp stability estimate for the momentum equations is not tractable, but operational experience shows that the algorithm is stable for $20 \leq CFL \leq 40$. For steady-state problems, backward-Euler provides additional damping that, in conjunction with $20 \leq CFL \leq 40$, provides a computationally efficient solution method. For URANS and LES computations, the trapezoid rule is neutrally dissipative, and delivers optimal performance for the more moderate CFL requirements for transient flow.

The solution algorithm used in Hydra-TH is based on a second-order

incremental projection algorithm. Projection methods are the most computationally efficient solution method available for solving the time-dependent Navier-Stokes equations. Over the past 20+ years, projection methods have enjoyed widespread adoption and have been applied to complex flow problems ranging from mold filling (volume-tracking) to atmospheric dispersion, chemically reacting flows, and exterior aerodynamics (see for example [2, 3, 10, 22, 15, 16, 17, 18, 19, 21, 13, 4, 8, 5, 9, 1, 20, 23, 24, 25]). The projection method also permits treating the momentum equations in a coupled manner (see for example [23, 24]. Although not currently used, Hydra-TH provides the underlying coding to couple arbitrary degrees-of-freedom in multiple transport equations. In addition, Hydra-TH has been implemented to permit rapid conversion to SIMPLE-based solution methods for steady-state problems. Extension to fully-coupled Newton-Krylov solution strategies can also easily be incorporated using inheritance with the virtual physics hierarchy.

In order to address fluid-structure problems, Hydra-TH uses an arbitrary Lagrangian-Eulerian (ALE) formulation and provides a mesh-deformation interface that can support multiple different mesh smoothing algorithms. Details on the ALE formulation may be found in the Hydra-TH theory manual [7]. The added-mass terms are computed for the structural coupling and can be exported for any structural solver. For explicit coupling, Hydra-TH provides a pressure-stabilized algorithm based on Nitche's varational method that circumvents the stability limitations associated with highly flexible structures and near unity fluid/solid density ratios. For conjugate heat transfer, there are multiple alternatives available in Hydra-TH that include explicit coupling with third-party heat conduction solver, internal coupling using the existing heat conduction solver supported by the Hydra toolkit and the multiphysics manager, or direct integration (with continuous meshing). The calculation of exported fields for both fluid-structure interaction and conjugate heat transfer are implemented for explicit coupling methods, and can be easily extended for use in tightly-coupled solution strategy. It is anticipated that driving CASL applications will determine the most suitable FSI/CHT solution strategy.

The linear algebra interface in Hydra-TH provides a number of linear algebra options that include both native solvers for testing/evaluation, and a rich set of solvers provided by PetSC. These solvers include the conjugate-gradient method (CG), bi-conjugate gradient squared (BCGS), generalized minimum residual (GMRES) and it's flexible counter part (FGMRES). For

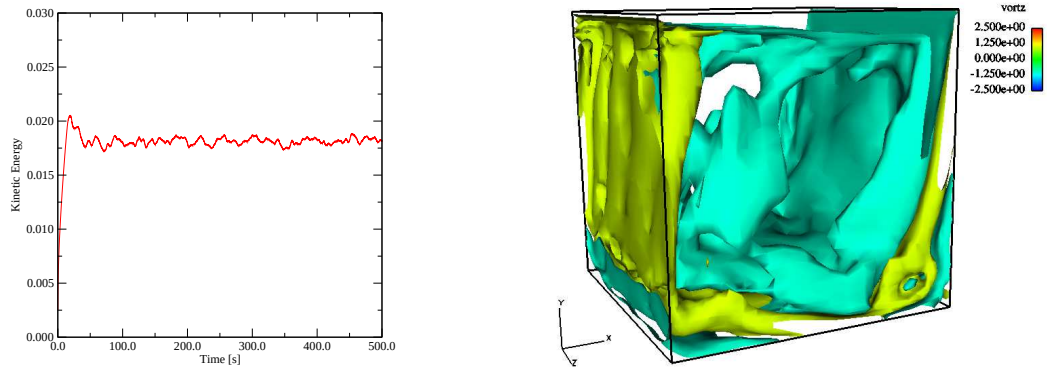
the pressure equation, the Trilinos ML preconditioner is used with CG while the transport equations typically use either Jacobi or ILU(0) preconditioning and FGMRES/GMRES.

The Hydra-TH flow solver was developed to make use of hybrid meshes and uses hex, tet, pyramid and wedge elements to permit meshing extremely complex geometries. All boundary and initial conditions are implemented using node, side (surface), and element sets permitting flexibility in the development of complex models. Material sets provide a simple and natural way to prescribe initial material interfaces for multi-fluid problems, and for prescribing material properties in conjugate heat transfer problems.

Hydra-TH builds on the Hydra toolkit to provide a number of non-Newtonian viscosities as well as the ability to handle temperature-dependent properites (see the Hydra-TH theory manual [7]). In addition, the use of output delegates permits Hydra-TH to provide a rich suite of output variables that are automatically tied to user input. Field output data may be requested as either element-centered or node-centered. In addition to the solution variables (e.g., velocity, temperature, turbulent kinetic energy, dissipation rate, etc.), Hydra-TH currently provides the following field output: displacements (ALE calculations only), vorticity, helicity, enstropy, the second-invariant of velocity gradient, i.e, the 'Q' criteria, processor ID (MPI rank), and turbulent eddy viscosity. Because output delegates are registered at run-time, it is trivial to add additional field output for visualization and debugging purposes. Surface field output variables include total traction, shear traction, normal traction, wall shear force, $y+$ and $y*$, heat flux, and the normal heat flux. Time history output variables include the primitive variables and turbulent eddy viscosity, enstrophy, vorticity, helicity, average pressure, surface area, average velocity, force, pressure forces, viscous forces, mass flow, volume Flow, heat flow, and average temperature.

The use of large-eddy simulation implies a required suite of post-processing tools for extracting useful averages and statistics. Hydra-TH provides an interface for performing windowed time-averaging suitable for post-simulation generation of ensemble averages. The calculation of derived statistics is outlined in §5.6 of the Hydra-TH theory manual[7]. A brief example is presented to illustrate the ideas.

In this example, a large-eddy simulation of a $Re = 10,000$ flow in a lid-driven cavity was performed using the first-author's older code GILA[6]. For the simulation, the dimensionless lid velocity was assumed to be $u_1 = 1$. Cavity dimensions were 1 unit in the x, y and z directions. Additional details

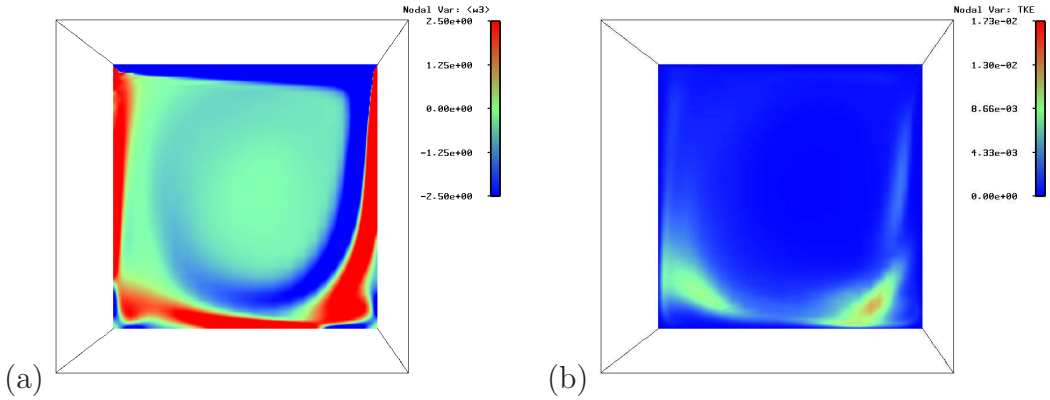


(a) (b)

Figure 1: $Re = 10,000$ lid-driven cavity (a) time history of global kinetic energy, (b) snapshot of instantaneous z-vorticity isosurfaces.

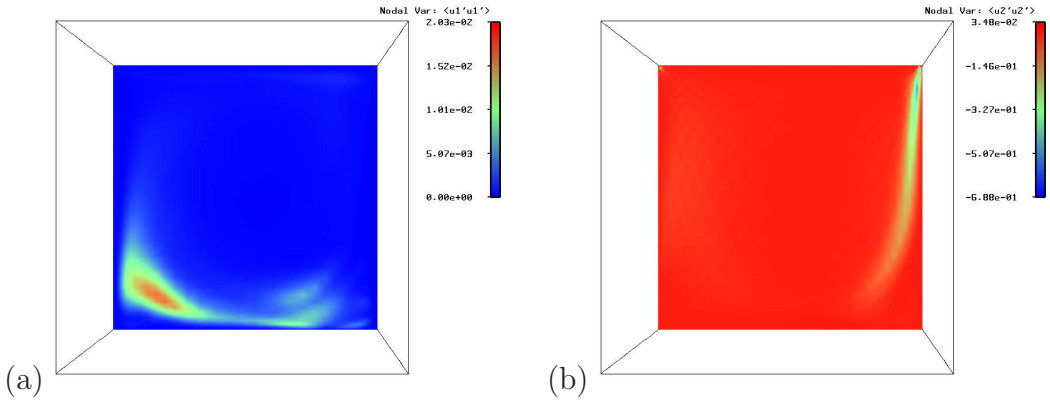
on this problem may be found in Zang, Street and Koseff [26]. A grid with $32 \times 32 \times 32$ elements was used and the simulation was run for approximately 400 dimensionless time units with 32 time planes containing average statistics being written.

Figure 1 (a) shows the time-history of kinetic energy, and Figure 1 (b) shows the instantaneous z-vorticity. Relative to the kinetic energy time-history plot, time planes 200 through 400 were used to derive the averages and statistics. The time-averaged z-vorticity and turbulent kinetic energy are shown in Figure 2. The $\langle u_1' u_1' \rangle$ and $\langle u_2' u_2' \rangle$ components of the Reynolds stress tensor are presented in Figure 3.



(a) (b)

Figure 2: Cutplanes showing the time-averaged (a) z-vorticity and (b) turbulent kinetic energy.



(a) (b)

Figure 3: Cutplanes showing the time-averaged Reynolds stress components (a) $\langle u_1' u_1' \rangle$ and (b) $\langle u_2' u_2' \rangle$.

3 Milestone Accomplishments

3.1 Hydra-TH Integration (Task-1)

Hydra uses CMake to manage compilation/installation processes which greatly simplifies porting to new platforms. In addition, several scripts have been developed to aid in porting/building the code and all of its dependent libraries on multiple systems, e.g., LINUX, Mac OS, etc. Hydra has been ported to the Conejo/Mapache LINUX clusters as well as multiple desktop systems at Los Alamos National Laboratory. Conejo and Mapache are both Linux based clusters with approximately 4900 Intel Xeon x5550 processors each. A preliminary port of some Hydra components and supporting libraries have been carried out on Jaguar at Oak Ridge National Laboratory. A series of test calculations have been carried out using Hydra's regression/verification suite in addition to a series of scaling studies as discussed in §3.2 and §3.6.

3.2 Initial Flow Calculations (Task-2)

This task was directly tied to the the integration/improvement of Hydra's testing infrastructure. Approximately 40% of the Hydra regression tests, i.e., those that were identified to be the most relevant to Hydra-TH have been incorporated in the new testing infrastructure at Los Alamos National Laboratory (see §3.5). In addition, a number of the Hydra test suite problems have been integrated and representative results for three of these problems are presented here.

3.2.1 Lid-Driven Skew Cavities

The first problem is a verification test that consists of five lid driven skewed cavity problems based on the work by Erturk and Dursun [11]. The geometry configuration is shown generically in Figure 4. On the bottom and side walls, no-slip/no-penetration boundary conditions were prescribed. Along the top "lid", a no-penetration boundary condition along with a unit lid velocity are prescribed. A single nodal pressure was prescribed in the bottom right-hand corner to set the hydrostatic pressure level.

The verification suite consists of five skewed cavities with $\alpha = 15, 30, 45, 60, 90^o$. Each skewed cavity uses three grids with 32×32, 128×128 and 256×256 elements. For all computations, $CFL = 10$ and backward-Euler time inte-

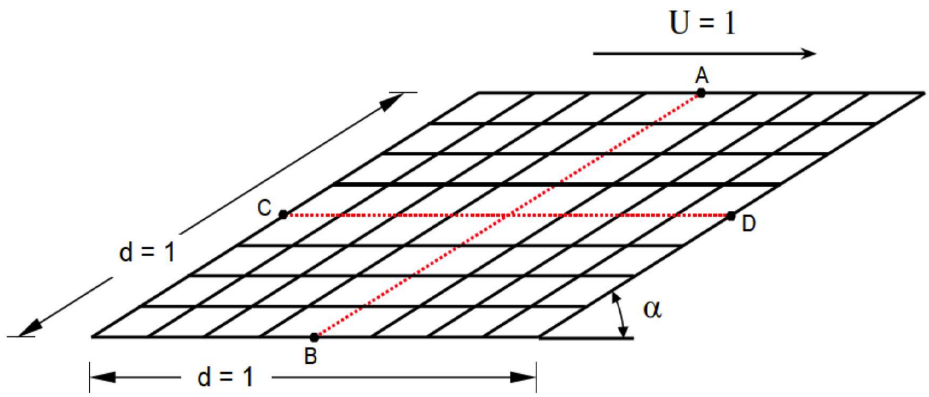


Figure 4: Skewed lid driven cavity geometry (reproduced from [11] without permission).

gration is used since the goal is a steady-state solution. Time history plots of the global kinetic energy indicate that a steady-state solution is reached by ≈ 10 time units. All problems for this verification test are run for 40 time units. The kinetic energy vs. time plots for the 128×128 grids are shown in Figure 5. Velocity data is extracted along the red center lines shown in Figure 4 for direct comparison with the reference data provided by Erturk and Dursun. The x-velocity profile is plotted against the vertical centerline, and the y-velocity profile is plotted against the horizontal centerline as shown in Figures 6 – 10.

All of the lid driven cavity problems achieve a steady-state (as verified by the global kinetic energy and velocity time-histories), and this provides a convenient way to assess the convergence behavior as the mesh is refined. All of the cavity meshes used uniform meshing, albeit with severely skewed elements for the 15° cavity. Table 1 shows the behavior of the kinetic energy as a function of the x-mesh size (h) which clearly indicates $O(h^2)$ convergence in all velocity components.

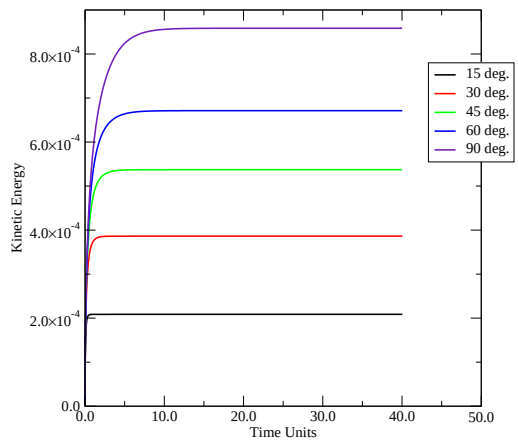
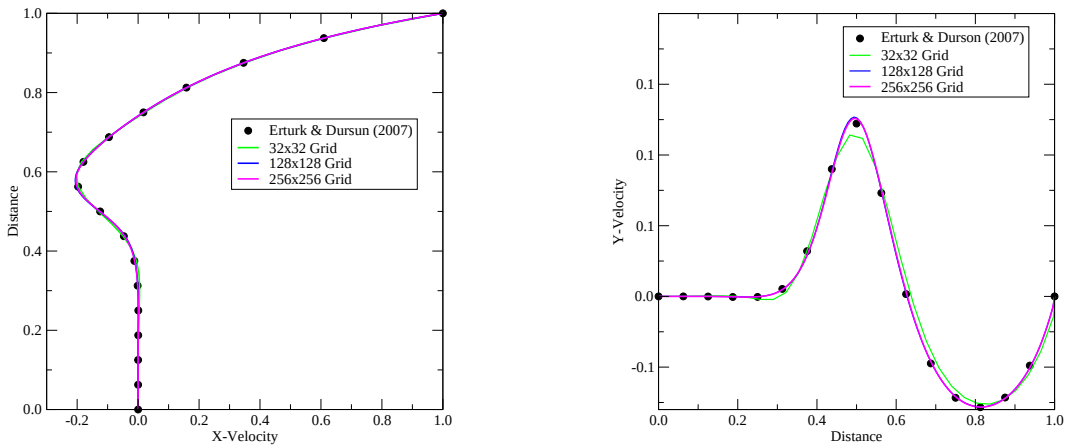


Figure 5: Kinetic energy vs. time for the 128×128 grids for $\alpha = 15, 30, 45, 60, 90^o$.

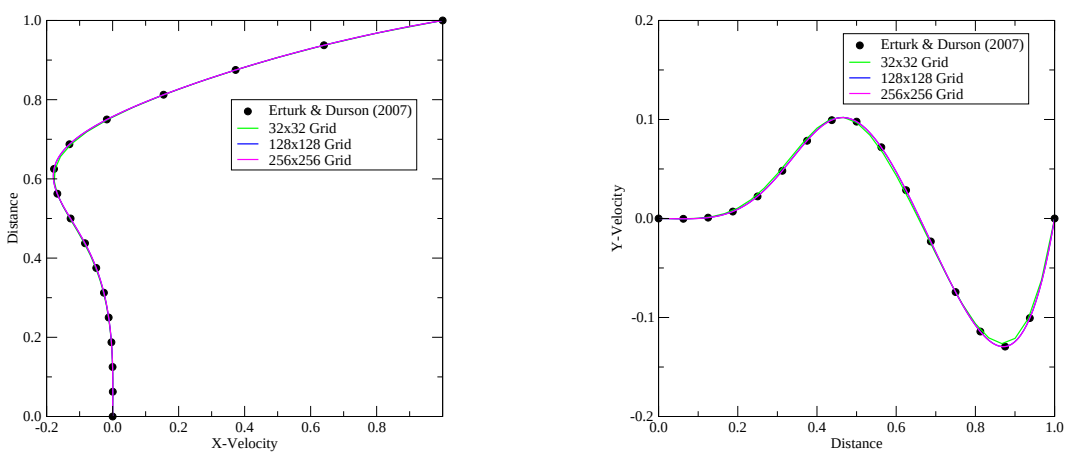
Cavity Angle	Global Kinetic Energy Correlation
15°	$0.00020907 - 0.006877\ h^2$
30°	$0.00038731 - 0.010046\ h^2$
45°	$0.00053854 - 0.014590\ h^2$
60°	$0.00067314 - 0.019690\ h^2$
90°	$0.00086136 - 0.029630\ h^2$

Table 1: Convergence behavior of the global kinetic energy vs. h for the lid-driven skewed cavities.



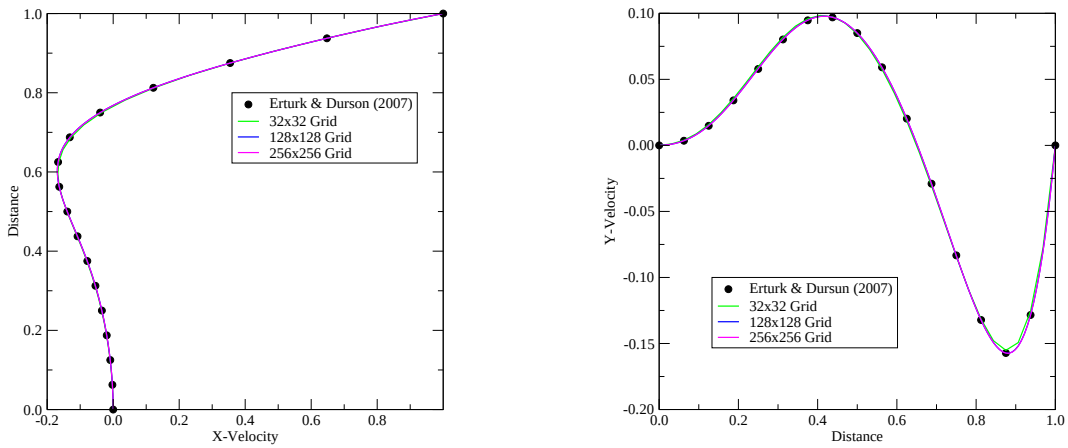
(a) (b)

Figure 6: 15° lid-driven cavity: (a) x-velocity, (b) y-velocity.



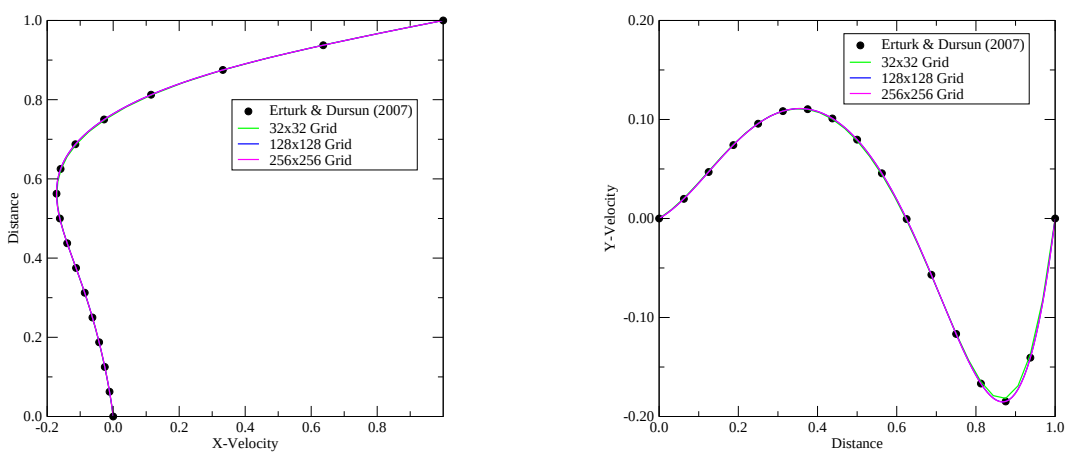
(a) (b)

Figure 7: 30° lid-driven cavity: (a) x-velocity, (b) y-velocity.



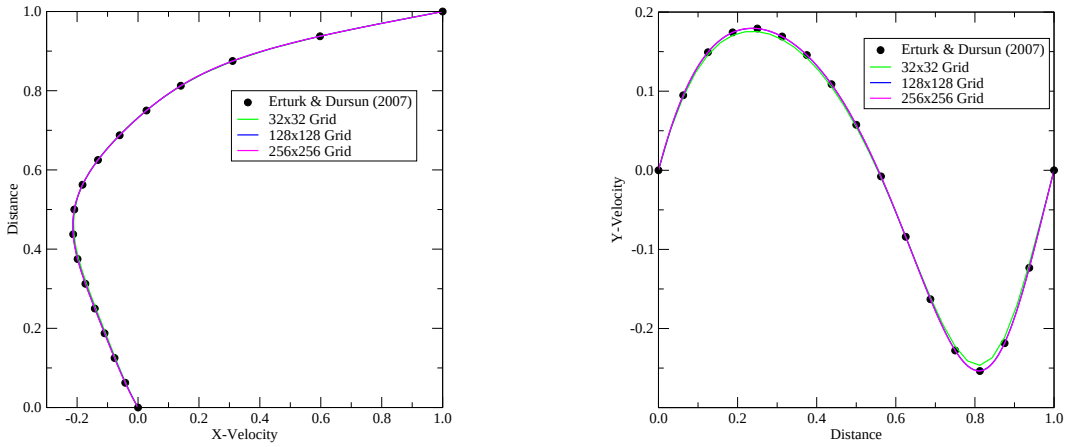
(a) (b)

Figure 8: 45° lid-driven cavity: (a) x-velocity, (b) y-velocity.



(a) (b)

Figure 9: 60° lid-driven cavity: (a) x-velocity, (b) y-velocity.



(a) (b)

Figure 10: 90° lid-driven cavity: (a) x-velocity, (b) y-velocity.

Mesh	No. of Elements	No. of Nodes
1	2520	5266
2	4068	8424
3	5902	12150
4	7630	15654
5	10080	20612

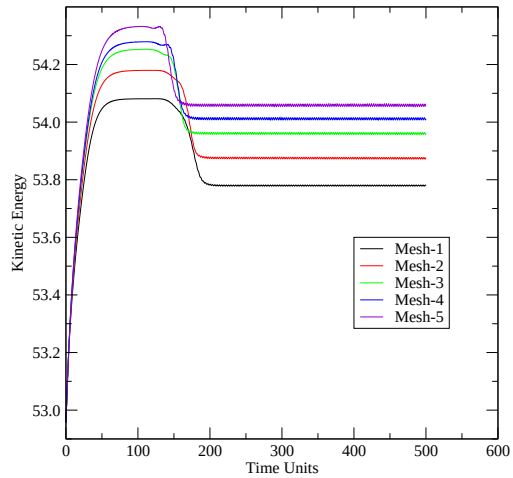
Table 2: Vortex shedding meshes.

3.2.2 Vortex Shedding

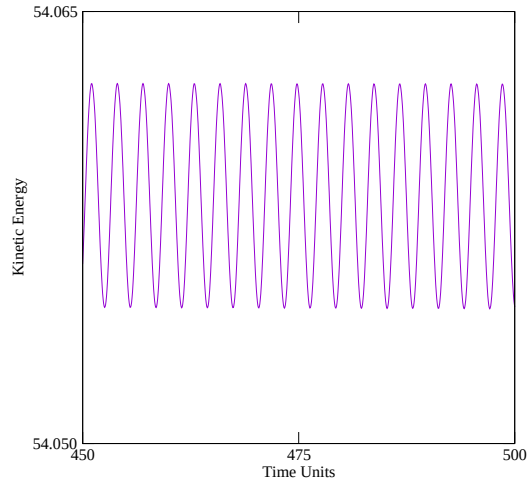
Similar to the lid-driven skew cavity test, flow past a circular cylinder is also one of the Hydra-TH verification problems. This flow problem is steady-state for $Re <\approx 50$, and becomes unsteady with a clearly defined vortex street for $Re > 50$. The transient behavior is a manifestation of a Hopf bifurcation, the first of which occurs at $Re = 47$.

The flow domain for this problem consists of a circular cylinder with a unit diameter, $-8 \leq x \leq 25$ and $-8 \leq y \leq -8$. The meshes are constructed using hex elements (1 element through the thickness). Five meshes are used with increasing resolution as shown in Table 2. The Reynolds number based on diameter is $Re = 100$. A constant inlet velocity is prescribed as $\mathbf{v} = (1, 0, 0)$. No-slip/no-penetration boundary conditions are used at the cylinder wall, and the x-velocity at the top and bottom of the domain is prescribed to match the inlet velocity, i.e., so-called "tow-tank" conditions. All calculations are run with a fixed $CFL = 2$ condition using the trapezoidal-rule time integrator. All calculation are run for 500 time units.

Figure 11 shows the time-history of the global kinetic energy for the 5 cases. All of the calculations go through a process of quasi-steady symmetric flow up until approximately 150 time units. At this point in time, the unstable mode has been saturated, and the flow transitions to a steady-periodic vortex shedding process with a well developed vortex street downstream from the cylinder. Here, the global kinetic energy indicates the spatial convergence. Figure 12(a) shows a snapshot of the x-velocity at 500 time units, and Figure 12(b) shows the z-vorticity. As expected, for each calculation, a single frequency is observed with a near-perfect sinusoidal variation in the kinetic energy as a function of time – see Figure 11(a). Due to space limitations, a detailed convergence study of this basic problem is not included.

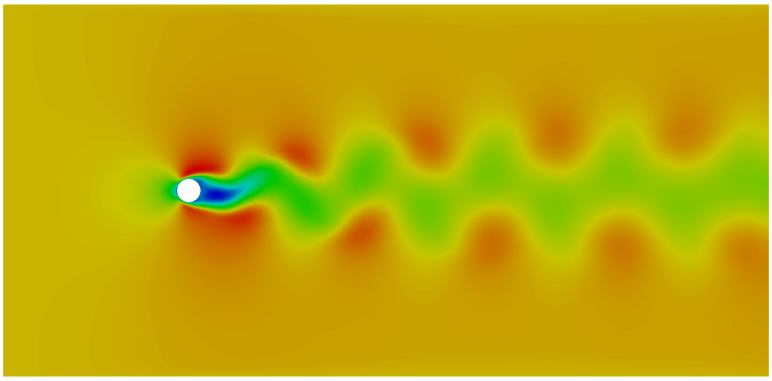


(a)

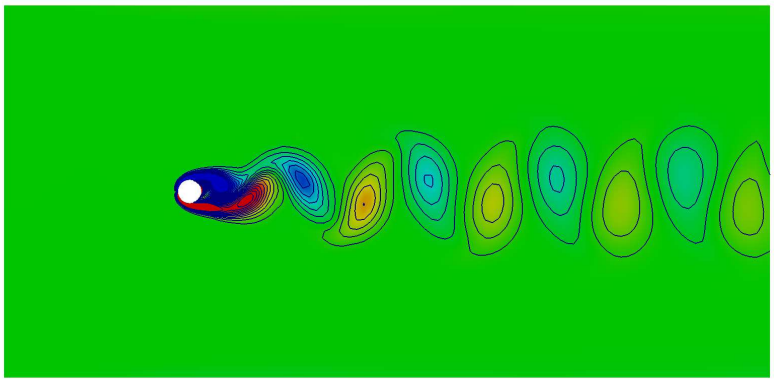


(b)

Figure 11: Time-history of the global kinetic energy. (a) Time-history of sequence of 5 meshes, (b) Time-history for mesh-5 for $450 \leq t \leq 500$.



(a)



(b)

Figure 12: Snapshot of the (a) instantaneous x-velocity and (b) the z-vorticity for mesh-5 at 500 time units.

3.2.3 Ahmed's Body

We present one final problem from the Hydra test suite that exercises the RNG k-ϵ, Spalart-Allmaras, Detached-Eddy Simulation (DES), and the implicit Large-Eddy Simulation (ILES) in Hydra-TH. The geometry for Ahmed's body is shown in Figure 13. Here, the RNG k-ϵ uses the so-called "scalable" or y^+-insensitive wall functions.

Remark 3.1 *The scalable wall functions used with the RNG k-ϵ model are important for thermal problems and working fluids with a large/small Prandtl number (Pr). In the case of a high-Pr, the thermal boundary layer dictates the near-wall mesh resolution and would not be acceptable for the momentum equations. The scalable wall function approach circumvents this difficulty.*

The Reynold number for this problem is based on the free-stream velocity, characteristic body length, density and molecular viscosity, which yields $Re = 4.29 \times 10^6$. No-slip/no-penetration boundary conditions are applied on the "ground" and Ahmed's body. Tow-tank conditions are prescribed at the front/back and upper flow domain boundaries. Inlet conditions prescribe a constant x-velocity, i.e., $\mathbf{v} = (1, 0, 0)$, and the homogeneous Neumann conditions for the transport equations are used at the outflow boundary.

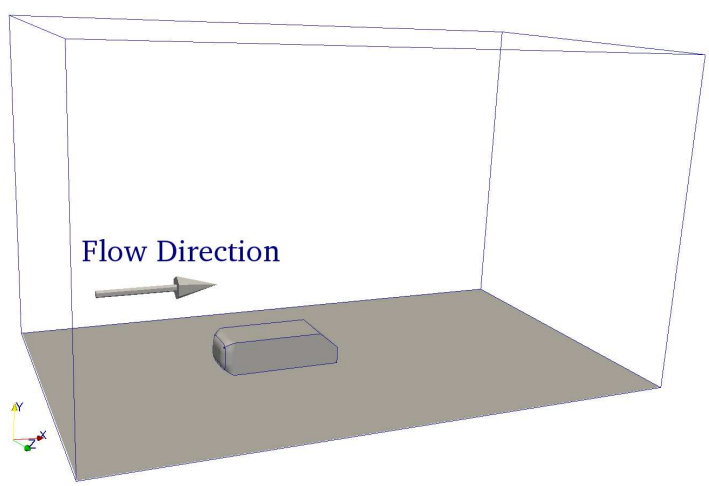


Figure 13: Flow past Ahmed's body with a 40° slant-back at $Re = 4.29 \times 10^6$.

The RNG k-ϵ and Spalart-Allmaras models are primarily designed to capture the long time-averaged mean quantities in an inhomogeneous turbulent flow, such as the Ahmed body problem. In contrast, DES and ILES capture, at different spatial fidelity, the time-dependent behavior, and thus provide a much richer statistical description than the RANS models (k-ϵ and Spalart-Allmaras).

The results presented here are primarily qualitative in nature, but serve to demonstrate the turbulence modeling capabilities in Hydra-TH. In addition, there are some interesting aspects of the side-by-side comparison of turbulence models that may be relevant to the modeling choices made for the GTRF problem. In the Ahmed body calculations, both the RNG k-ϵ and Spalart-Allmaras models used a $CFL = 10$ and achieved steady-state after about 2 time units. The ILES and DES computations used $CFL = 2$ and were also run for 2 time units which is sufficient for the ILES and DES models to establish a statistically stationary state as well.

Figure 14 shows cutplanes of velocity magnitude isosurfaces, and Figure 15 shows isosurfaces of the helicity. The RNG k-ϵ and Spalart-Allmaras results are very similar, and as expected only capture the primary helical structures downstream of the Ahmed body. In contrast, the instantaneous snapshots of the velocity and helicity for the DES and ILES calculations show a richer flow structure and indicates that both models are capturing secondary and tertiary helical structures in the wake. In the future, we hope to assess this range of turbulence models on the GTRF geometry when it becomes available.

3.3 Development Guidelines (Task-3)

A series of development processes have been put in place at LANL for the Hydra-TH development team. The processes are based on a combination of practices adapted from those used at Dassault-Systemés SIMULA for the Abaqus suite of tools, and those commonly employed in agile development practices. Because the team is relatively small, a formal document has not been put in place to outline the practices. Here, we summarize the general practices that will be followed.

All development of new features in Hydra-TH will be initiated with a process of collecting requirements and summarizing the software design. These documents are collected and stored in the Hydra git repository. After a preliminary implementation, a design review is conducted to assess the de-

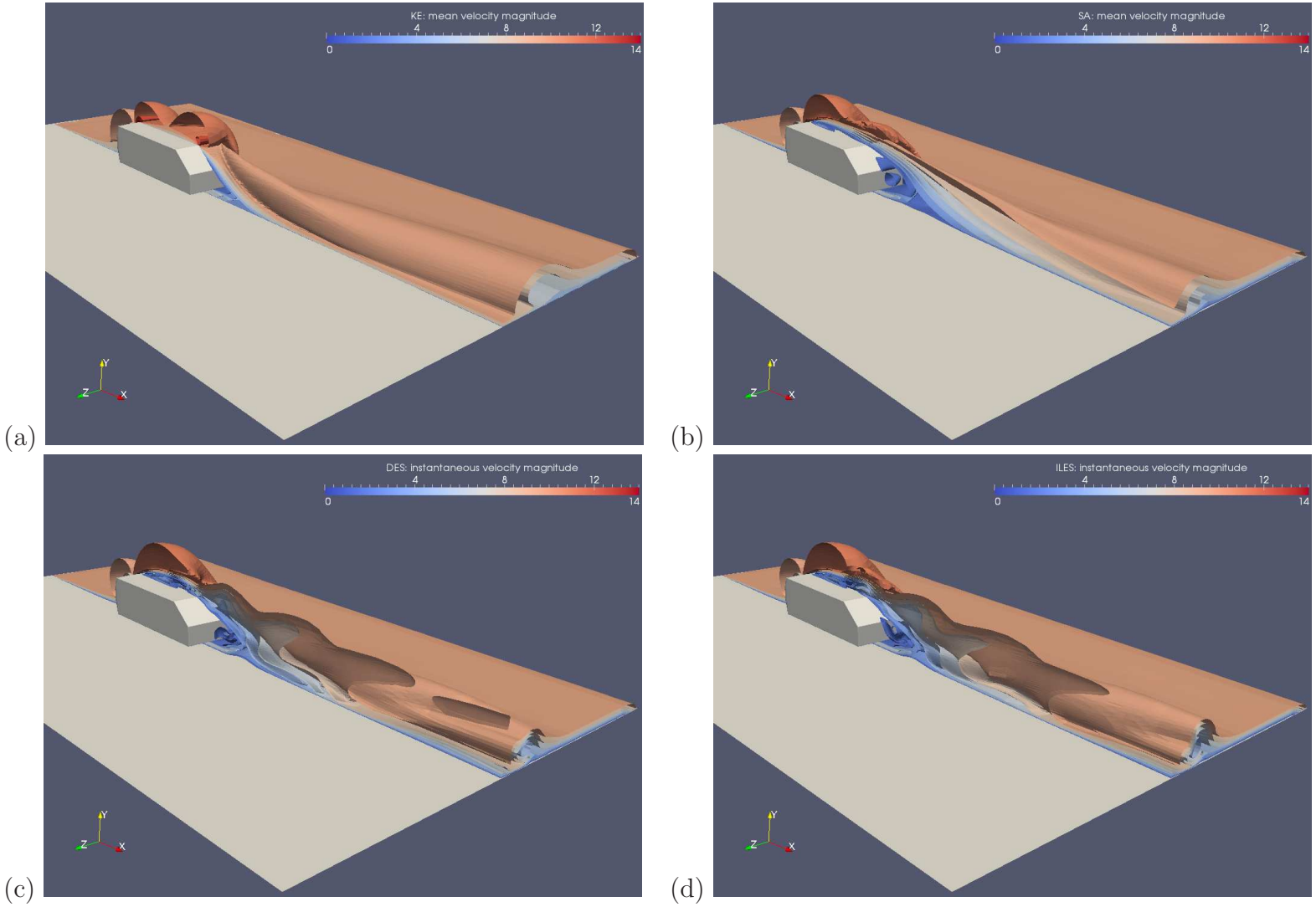


Figure 14: Cut-plane of velocity isosurfaces at 2 time units for (a) RNG k-ϵ, (b) Spalart-Allmaras, (c) DES, (d) ILES.

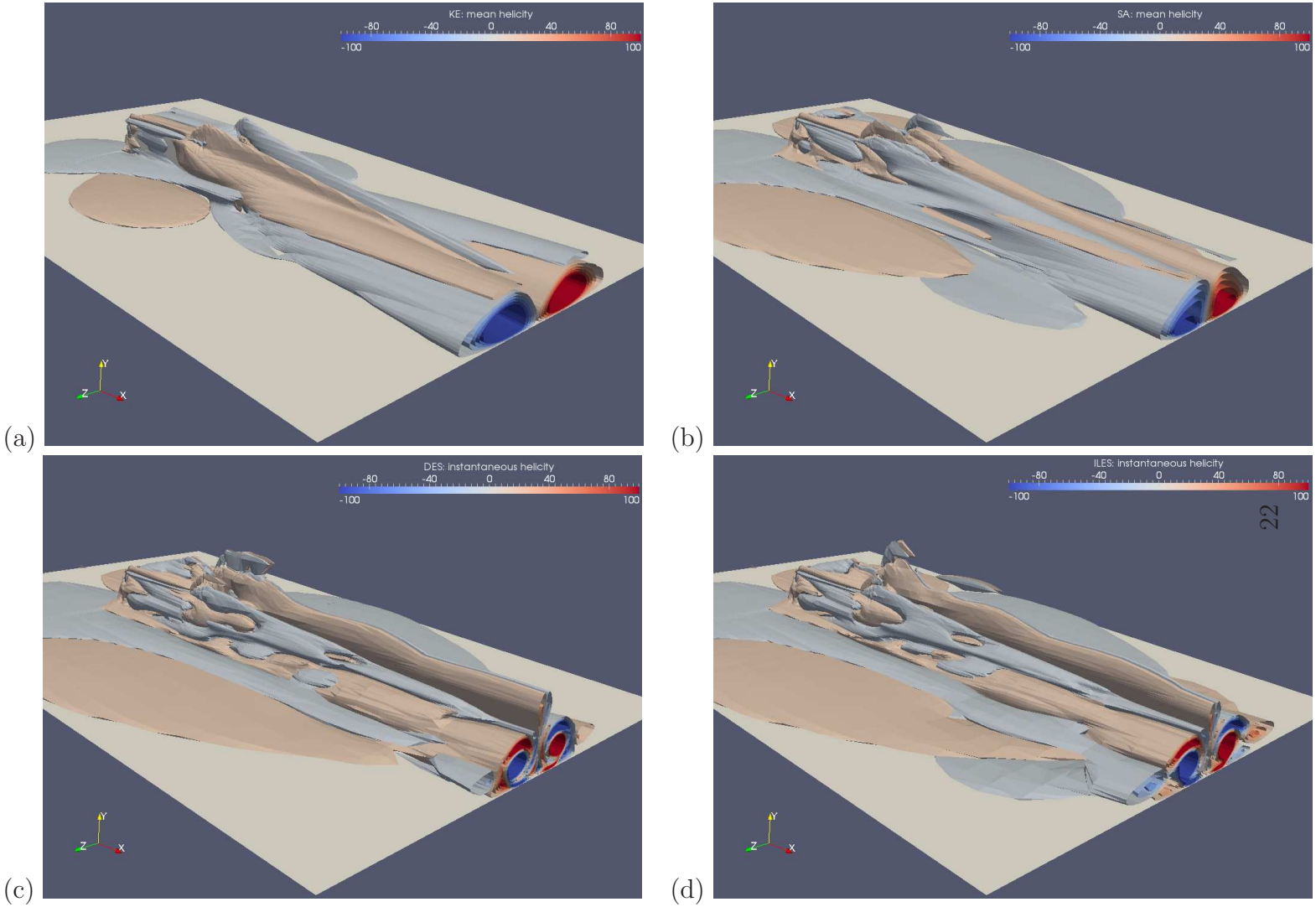


Figure 15: Cut-plane of helicity ($\mathbf{v} \cdot \omega$) isosurfaces at 2 time units for (a) RNG k-ϵ, (b) Spalart-Allmaras, (c) DES, (d) ILES.

sign, implementation, computational performance and usability. Following the design review, a revised implementation is put in place to address any concerns/questions that have been raised. Each developer is responsible for executing the regression test suite on 2 platforms in serial and parallel and ensuring that all tests pass successfully before a code update is made. New features require the addition of one or more regression tests and a companion verification problem when necessary.

As a general rule, regression tests will be run in a "nightly grind" with test results posted to *http://hydra.lanl.gov*. The verification and customer model suite of tests will be executed on a bi-weekly basis with test results posted to *http://hydra.lanl.gov*. (Note: *http://hydra.lanl.gov* is on the internal LANL intranet.)

3.4 Periodic BC's (Task-4)

Task-4 is focused on the design and implementation of periodic boundary conditions to support the 3×3 geometry. The requirements for periodic boundary conditions include 1) the ability to specify periodic conditions using the side-set conventions supported in Hydra-TH, 2) prescribe multi-direction periodicity, e.g., periodic in x, y and z-coordinate directions, 3) treatment of the hydrostatic pressure mode, and 4) error-handling for mis-aligned periodic conditions.

At the time of this writing, the initial design has been completed for the periodic conditions, however, the implementation has not been completed. For the transport equations, Hydra constructs ghost-cells on all boundaries for general communications and use in prescribed boundary conditions. Extension of the treatment of ghost cells to handle periodic conditions is relatively straightforward, but do introduce additional communications patterns that will be accounted for in the extension of Hydra's communicator classes for element and dual-edge centered data. The pressure-Poisson equation also requires additional communication for the construction of the elliptic operator. However, it is anticipated that the existing linear algebra interfaces are adequate for managing the assembly of the extended linear system. The hydrostatic mode will be eliminated by use of a modified constraint equation that will replace the current method of pegging a single hydrostatic pressure.

The implementation of the periodic boundary conditions will be continued based on the staffing levels and development requirements for the POR4 milestones still being developed.

3.5 Testing Infrastructure (Task-5)

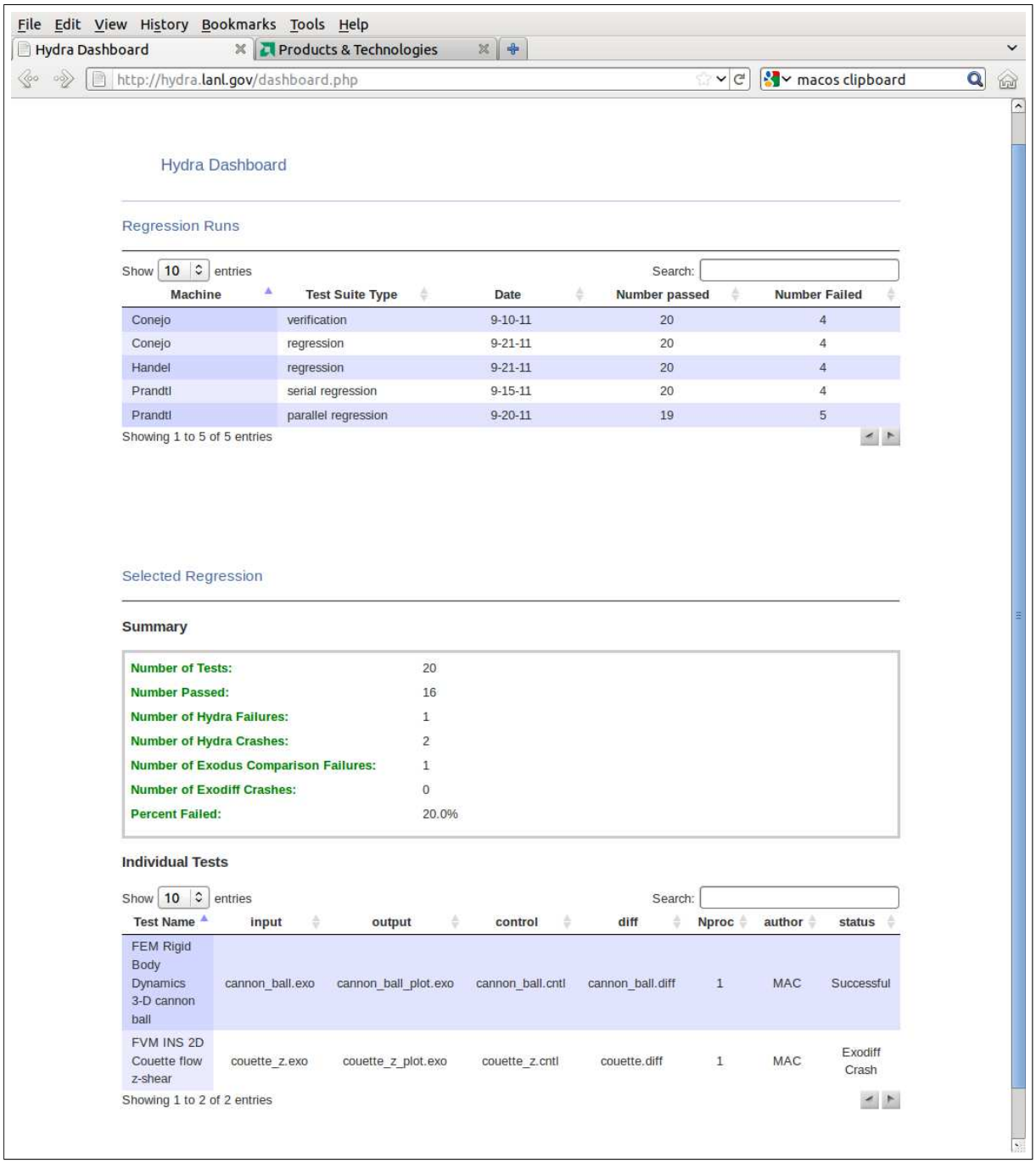
This task has focused on two primary objectives. First, the migration of existing Hydra regression, verification and customer model problems into the LANL development environment. Second, the improvement of the testing infrastructure for Hydra/Hydra-TH.

As a point of departure, we note that there are a series of test-suites for Hydra/Hydra-TH that include short-running serial regression tests, parallel regression tests, verification, and "customer" model problems. The customer model problems, such as the Ahmed body, are not necessarily categorized strictly as verification or validation problems, but represent typical application problems that are demanding and for which some comparison data, either numerical or experimental, are available.

Each test suite has a corresponding XML description file that, for each test, contains the test name, local path to the test, input files, baseline output date (golden files), the number of processors to be used for execution, the primary author, the comparison tool, etc. A python script (rtest.py) is used to execute the test suites on a given platform and generates summary data interactively for the tests. This permits the python script to be used by developers for pre-checking testing. For test failures, rtest.py generates a new XML description of all failed tests that can be re-executed once the test failures have been corrected.

In addition, rtest.py can optionally generate an XML file that may be stored or pushed to the php server on *http://hydra.lanl.gov* and stored in a mysql database. For nightly-grinds and bi-weekly verification/model testing, the results are similary stored in the mysql database, and the results are posted to *http://hydra.lanl.gov/dashboard.php*. Sample output from the Hydra testing dashboard is shown in Figure 16.

The Hydra Dashboard supports general search capabilities for regression type, execution date, machine name, compiler type, operating system, operating system release date, system processor, test name, test path, input and output files, number of processors, author, status, etc. Selecting a specific machine, test suite type or date exposes a summary of tests. Each test can be probed for in-depth information.



File Edit View History Bookmarks Tools Help

Hydra Dashboard | Products & Technologies

http://hydra.lanl.gov/dashboard.php

Hydra Dashboard

Regression Runs

Show 10 entries Search:

Machine	Test Suite Type	Date	Number passed	Number Failed
Conejo	verification	9-10-11	20	4
Conejo	regression	9-21-11	20	4
Handel	regression	9-21-11	20	4
Prandtl	serial regression	9-15-11	20	4
Prandtl	parallel regression	9-20-11	19	5

Showing 1 to 5 of 5 entries

Selected Regression

Summary

Number of Tests:	20
Number Passed:	16
Number of Hydra Failures:	1
Number of Hydra Crashes:	2
Number of Exodus Comparison Failures:	1
Number of Exodiff Crashes:	0
Percent Failed:	20.0%

Individual Tests

Show 10 entries Search:

Test Name	input	output	control	diff	Nproc	author	status
FEM Rigid Body Dynamics 3-D cannon ball	cannon_ball.exo	cannon_ball_plot.exo	cannon_ball.cntl	cannon_ball.diff	1	MAC	Successful
FVM INS 2D Couette flow z-shear	couette_z.exo	couette_z_plot.exo	couette_z.cntl	couette.diff	1	MAC	Exodiff Crash

Showing 1 to 2 of 2 entries

Figure 16: Snapshot of the Hydra testing dashboard.

3.6 GTRF Results (Task-6)

This task was directed at generating results on the GTRF meshes used by Drekar, and it was contingent on the meshes being supplied by Sandia National Laboratory in a timely manner. At the time of this report was written, the CUBIT input files for the GTRF have just been released to the Hydra-TH development team. Currently, the Hydra-TH team is attempting to regenerate the GTRF meshes using CUBIT version 13.0, and beginning to setup a series of calculations on the GTRF 3×3 geometry. Unfortunately, there is insufficient time to permit completion of the calculations and reported them here.

Due to the delays in obtaining the GTRF 3×3 meshes, a number of activities have been carried out that will be reported on here. First, as Hydra-TH was designed for use with hybrid meshes consisting of hex, tet, pyramid and wedge elements, a collaborative effort with the primary author of the Spider mesh generator (see http://www.meshing.at) has been initiated. The Spider mesh generator is marketed commercially as Hybrid/Hexpress by NUMECA. For simplicity, we refer to the product as Spider.

3.6.1 Spider Mesh Generator

Spider provides a unique capability for geometry import and cleanup as a part of its shrink-wrap process. Shrink-wrapping refers to the process of generating a closed CFD volume appropriate for mesh generation. Spider uses an octree mesh generation procedure that produces hex-dominant meshes and permits insertion of graded boundary layers. Closing "dirty" geometry is accomplished with the "hole-searcher" tool which provides a rapid mechanism for geometry cleanup/repair. Spider currently uses a shared memory (openMP) model for parallelism, and while limited in this respect promises to scale into the billion-cell range relatively easily, albeit with sufficient memory. Spider/Hybrid consists of two main components. The GUI, and the mesh generator. The mesh generation works fully in batch mode. Tools like the hole-searcher may be run in batch mode or using the GUI.

As a general guideline, the memory required to run Spider is about 500 MBytes per million nodes. A minimum of 4 GBytes is required. As a general guideline, about 1 GByte of disk is required to store a two-million nodes mesh in native (*.HPM) format. Currently, Spider supports the following file formats: Fluent (*.msh), StarCD (*.cel), Patran Neutral (*.neu), AVL Fire

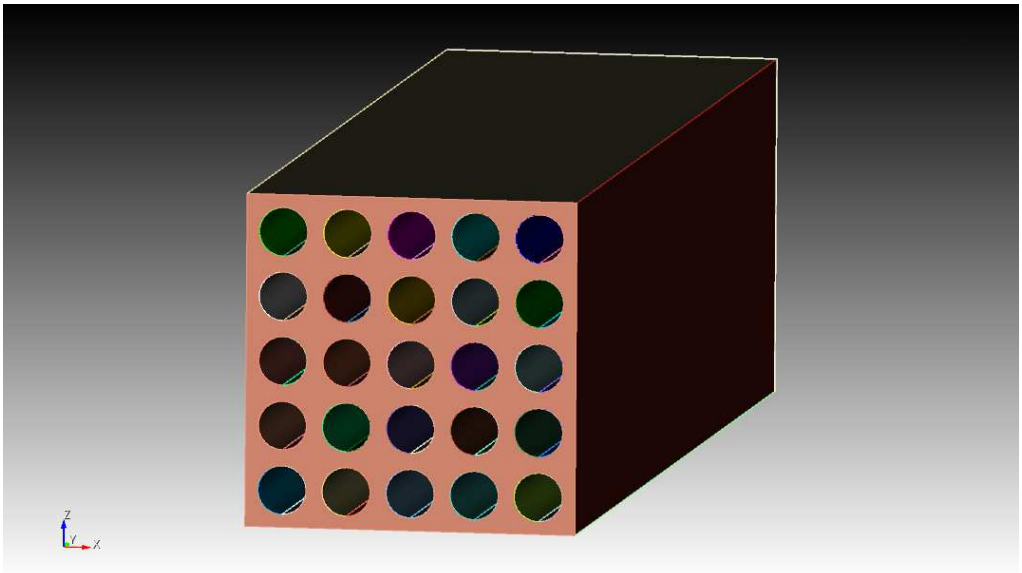


Figure 17: 5×5 GTRF test geometry.

(*.flma), Nastran (*.nas), and CEDRE (*.dat). At the time of this writing, NUMECA and the primary developer have agreed to provide support for writing Exodus-II files for hybrid meshes suitable for use with Hydra-TH. In addition, this capability may be useful for other lab codes that make use of the Exodus-II file format.

As a part of the collaboration with both ORNL, the Spider author, and NUMECA, several representative GTRF test meshes have been constructed using Spider. All meshes were generated by the author of Spider as a part of the collaboration. Emillian Popov (ORNL) provided the generic (non-NDA) 5×5 geometry shown in Figure 17. This geometry has representative internal supports and was thought to be a reasonable test case for the shrink-wrapping, hole-searcher, meshing process used by Spider.

The geometry was imported from a IGES file using CUBIT, and exported as an STL file for using in Spider. A 90 million element mesh was generated using the STL input with 6 layers of all-hex boundary layer elements. Figure 18 shows the 90 million element mesh from a perspective view. Figure 19 shows the surface mesh on the interior of the 5×5 geometry and reveals the ability to accurately capture the spacer structure and associated spacer-rod juncture. Figure 20 shows a cut-plane through the interior of the mesh

revealing the ability to lay boundary layer mesh around all the complex interior regions of the 5×5 geometry. Starting with the STL input file, the reported run time for mesh generation was approximately 45 minutes on a 12-core workstation for the 90 million element mesh and insertion of boundary layer elements.

Remark 3.2 *We note in passing that the ability to insert wall boundary layer elements is thought to be very important for turbulence models that make use of wall functions and require a specific wall-normal distance for the first element from the wall. In addition, this capability is important for to get accurate wall boundary meshes in order to compute accurate traction/heat flux for FSI/CHT.*

Based on input from Emillian Popov (ORNL), for full-core modeling, a coarser mesh was generated. For shrink-wrap meshing, the concern is that the resolution of the base geometry can degrade with coarse meshes, i.e., when the specified resolution h is too large to capture some geometry. Figure 21 shows the surface mesh associated with a mesh containing approximately 7 million elements with one layer of hex elements at the boundary. In comparison to Figure 19, there is some minor degradation of the geometry near the spacers. However, this appears to be an adequate mesh for analysis purposes. More testing is required with Hydra-TH to assess the behavior of these grids on the flow solution.

One final experiment has been conducted with Spider to assess the ability to scale to the billion-cell range. The Spider author scaled the $5x5$ geometry to 1 billion cells using 80 core PGV SuperServer and used approximately 250 GBytes of memory. Initial results indicate that the parallel scaling was non-ideal, and the author is investigating the scaling issues at this time.

3.6.2 Scaling Studies

In order to assess Hydra-TH scaling, the channel flow originally defined by Kim, et al. [14] was selected in lieu of the SNL GTRF meshes. In part, this geometry was chosen to permit the use of a custom, light-weight mesh generator that can produce extremely large meshes. This was done to exercise the Exodus-II file format and understand its current limitations for large-scale CFD problems, and to evaluate the various read/write options in Hydra-TH.

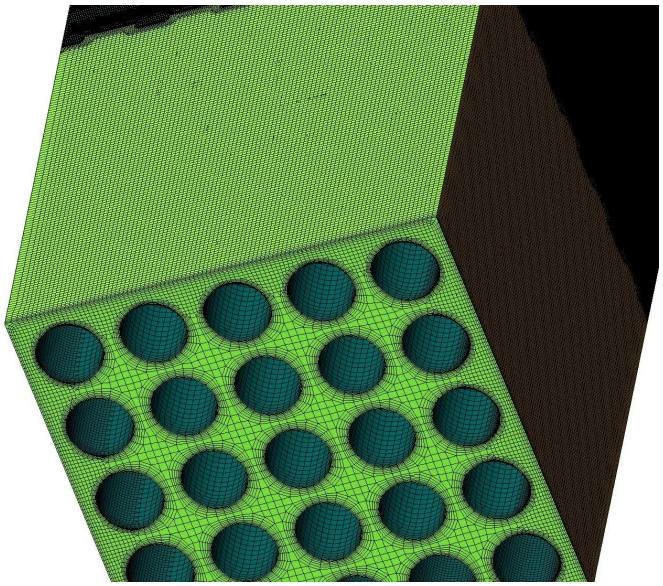


Figure 18: Mesh for the 90 Million element 5×5 GTRF test geometry.

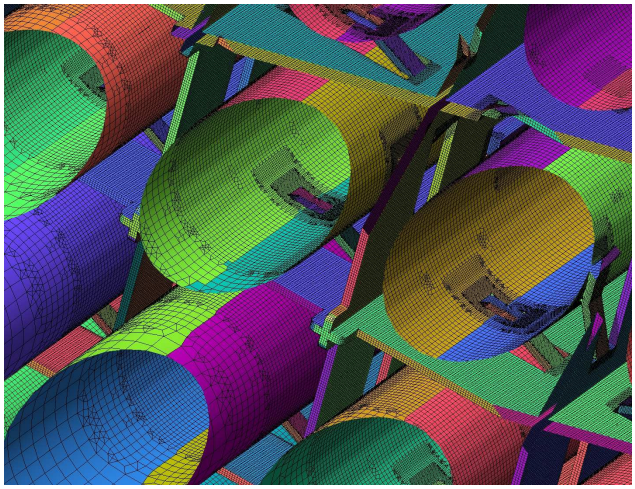


Figure 19: Internal surface mesh for the 90 Million element 5×5 GTRF test geometry.

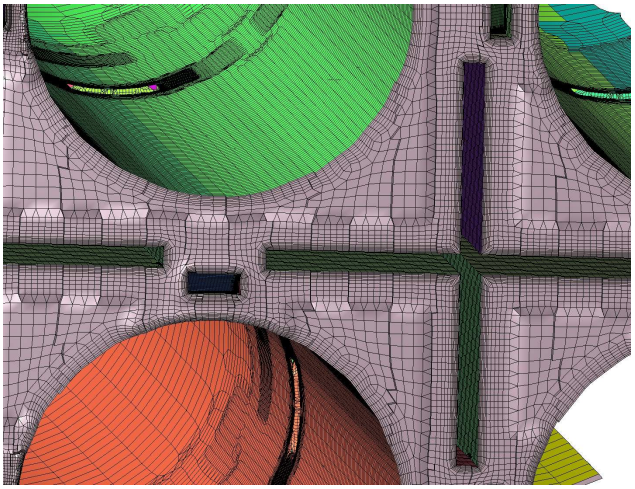


Figure 20: Internal mesh for the 90 Million element 5×5 GTRF test geometry.

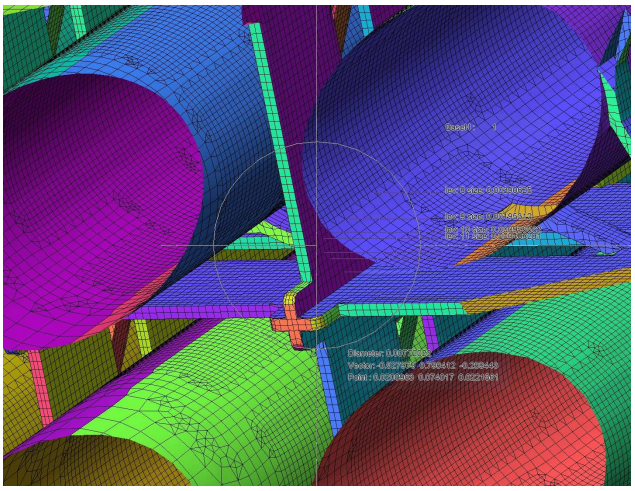


Figure 21: Internal surface mesh for the 2 Million element 5×5 GTRF test geometry.

Mesh	N_x	N_y	N_z	*No. Elements*	*DOF*
1	563	90	282	14,288,940	5.7392E+7
2	704	112	342	27,754,496	1.1139E+8
3	880	140	440	54,208,000	2.1741E+8
4	1100	176	550	106,480,000	4.2682E+8

Table 3: Parameters used for the channel meshes.

The channel geometry is rectangular with $L_x = 4\pi\delta$, $L_y = 2\delta$, and $L_z = 2\pi\delta$ where $\delta = 1/2$. Using this geometry, a series for 4 grids were generated using the parameters shown in Table 3. Here, the grid sizes were chosen to respect the original problem definition, and not to yield perfect element distributions across processors. This also provides a more realistic assessment of the parallel scaling with arbitrary unstructured meshes using complex geometry in contrast to a perfect power of 2 scaling in mesh size.

The shear Reynolds number was defined as $Re_\tau = u_\tau\delta/\nu = 395$. No-slip/no-penetration conditions were prescribed at the upper/lower walls, and a no-penetration condition was specified on the front/back planes. Periodic conditions would be preferred for the front/back planes of the domain, but for the purposes of a scaling study, this boundary condition is adequate. In addition, the $L_z = 2\pi\delta$ is large enough to support a reasonable long-wavelength variation in the cross-stream direction.

For all computations, the pressure-Poisson equation was solved using CG with the ML (multilevel) preconditioner. A minimum coarse-grid size of 1000 equations was specified with an incomplete-Cholesky smoother. The momentum transport equations used the ILU(0) preconditioner with FGMRES and a space of 15 restart vectors. All timings were measured over 500 time steps and include the solution time for the momentum transport equations, the pressure-Poisson equation (PPE), the total solution time, and the grind time (wall-clock time required to advance one element one time step).

For each grid, a series of 3 calculations were performed using an increasing number of processors. For example, for the mesh-1, 3 calculations were performed using 32, 64, and 128 processors. For mesh-2, 64, 128, and 256 processors were used. All calculations were carried out on the Conejo and Mapache clusters at Los Alamos National Laboratory.

Figure 22 shows the scaling results for the 4 channel meshes. Due to the user load on Conejo and queue contention, mesh-4 was run on the Mapache

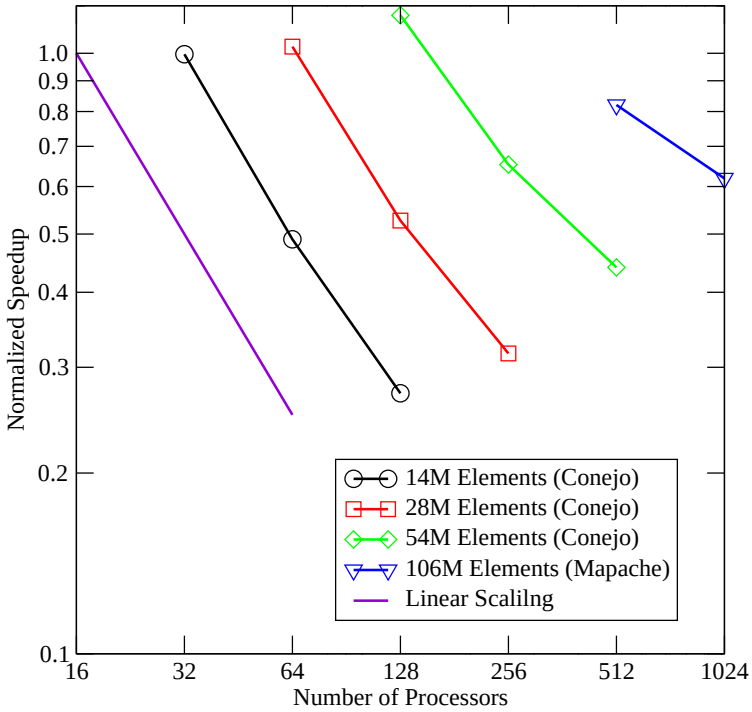


Figure 22: Normalized speedup for the 4 channel flow meshes. Here, the solution wall clock time for each calculation has been normalized by the average solution time for the 4 meshes on the minimum number of processors.

cluster. Unfortunately, the timing data for Mapache appears to be inconsistent with the timing data collected on Conejo. The base processor (Intel Xeon x5550), clock speed and interconnect is the same on both machines, so the cause of the differences is not clear at this time.

In Figure 22, the curve for each of the 4 meshes indicates the strong-scaling behavior. The variability in timings for this study make it difficult to present weak-scaling results. Ideally, a constant normalized speedup could be achieved with a factor of two increase in mesh resolution and a corresponding 2× increase in number of processors. We are aware of two factors that degrade both the strong and weak scaling in this study.

Mesh	No. of Processors	Minimum Iterations	Maximum Iterations	Average Iterations
1	32	12	22	15
	64	14	24	16
	128	14	24	16
2	64	14	22	15
	128	14	24	15
	256	14	24	16
3	128	18	32	21
	256	20	34	23
	512	18	56	24
4	512	22	36	24
	1024	22	36	25

Table 4: Minimum, maximum and average number of iterations required to solve the pressure-Poisson system using the ML preconditioner with CG.

The first and perhaps most significant factor is the effect of load-balancing. Hydra-TH currently uses ParMETIS, and it has been observed that, for very regular structured meshes, the graph-based partitioning can create very poor partitions that in turn affect the performance of the ML preconditioner for the PPE. Examination of the sub-domain geometry for several of the computations has indicated that this is the case. At this time, we are working to improve the partitioning algorithms for the special case of structured meshes.

The second factor is the algorithmic scaling associated with the ML preconditioner and CG solver. Table 4 shows the minimum, maximum and average number of iterations required to solve the PPE. For mesh-1 and mesh-2, the minimum, maximum and averages are essentially constant. However, mesh-3 shows a relatively large ($\approx 40\%$) increase in the average number of iterations to solve the PPE. This increase translates into increased wall-clock time and communications overhead. This is reflected in the increased normalized speedup for the mesh-3 (54M elements) in Figure 22. Similarly, the increase in average iterations with increasing number of processors for mesh-3 is associated with the deviation from ideal scaling. At this time, we are beginning to explore alternative preconditioner/solver combinations for the PPE.

Comments on the Scalability of Exodus-II Files

During the channel flow scaling studies, it was found that the `EXODUS_LARGE_MODEL` environment variable was required to generate meshes Exodus-II files with $O(1.0e+8)$ elements. Discussions with the Exodus-II developers revealed that this is a fundamental limitation in the older versions of the netcdf library which provides the database format for Exodus-II. The approach for circumventing this limitation is to use the netcdf-4 file format which is built on the hdf5 library.

3.7 Documentation (Task-7)

Task-7 focused on the generation of this report, and the assessment of documentation for Hydra-TH. The Hydra-TH theory manual has been revised and updated to reflect the current status of the code[7]. At this time, a user manual is in place, but omits many of the current features in Hydra-TH. This manual will be revised in the future and also released as an LA-UR.

3.8 VERA-CFD Implementation Plan (Task-8)

Finally, the VERA-CFD implementation plan has been updated to include work related to Hydra-TH. The Hydra-TH theory manual has been placed in the CASL implementation plan sharepoint.

4 Summary

A series of 8 tasks have been carried out as defined by the Hydra-TH L3 Milestone (THM.CFD.P3.02). Task 4 is still under development, and task 6 was conditional based on delivery of the GTRF meshes from Sandia National Laboratories. A series of verification problems from the Hydra-TH test suite were presented where second-order spatial accuracy was demonstrated on highly skewed meshes. In addition, the turbulence modeling capabilities in Hydra-TH both in terms of steady-state RANS models and time-accurate DES/LES models have been demonstrated on a problem that involves massive separation and complex helical structures in the flow – similar in character to those exhibited downstream from the GTRF spacers.

In lieu of the SNL GTRF meshes, a series of scaling studies were undertaken on the LANL clusters Conejo and Mapache. Although reasonable

strong scaling was demonstrated, the algorithmic scaling of the ML preconditioner requires additional investigation. In addition, efforts are underway to improve the run-time parallel load-balancing to achieve more consistent generation of sub-domains for regular meshes.

Based on the scaling studies, we are further examining/testing the Exodus-II file format to ensure that it will adequately scale to the 10^9 element range. As suggested in §3.6.2, there are limitations to the Exodus-II file format depending on the use of environment variables and versions of the netcdf, hdf5 libraries. This exercise will feed the collaborative interactions for the Spider mesh generator, and provide a tool that may be used to generate large-scale GTRF meshes in the future.

Future efforts will be directed, in part, towards a study of the GTRF using multiple turbulence models, the demonstration of calculations using Hydra-TH with hybrid meshes generated by the Spider mesh generator, and revised scaling studies.

ACKNOWLEDGMENTS

The authors would like to thank Linda Weltman for her efforts in getting the authors access to the Oak Ridge National Laboratory scientific computing resources.

References

[1] H. T. Ahn, M. Shashkov, and M. A. Christon, *The moment-of-fluid method in action*, Communications in Numerical Methods in Engineering, 25 (2008).

[2] J. B. Bell, P. Colella, and H. M. Glaz, *A second-order projection method for the incompressible navier-stokes equations*, Journal of Computational Physics, 85 (1989), pp. 257–283.

[3] J. Blasco, R. Codina, and A. Huerta, *A fractional-step method for the incompressible navier-stokes equations related to a predictor-multicorrector algorithm*, International Journal for Numerical Methods in Fluids, 28 (1998), pp. 1391–1419.

[4] M. A. Christon, *Domain-based parallelism and the projection algorithm for transient, viscous incompressible flow*, in preparation for Computer Methods in Applied Mechanics and Engineering, (1998).

[5] ——, *The new incompressible flow capabilities in LS-DYNA*, in 6th International LS-DYNA Users Conference 2000, Dearborn, Michigan, April 2000, ETA.

[6] M. A. Christon, *GILA – User Manual, version 1.0 – NS-SS*, SAND2003-1781, Sandia National Laboratories, Albuquerque, New Mexico, May 2003.

[7] M. A. Christon, *Hydra-TH Theory Manual*, Tech. Rep. LA-UR 11-05387, Los Alamos National Laboratory, September 2011.

[8] M. A. Christon and D. E. Carroll, *An unstructured-grid, parallel, projection solver for computing low-speed flows*, in International Conference on Computational Engineering Science, S. N. Atluri and P. E. O'Donoghue, eds., Atlanta, Georgia, October 1998, Tech Science Press, pp. 845–850.

[9] M. A. Christon and R. S. Patil, *A finite element projection method for low-Mach number reacting flows*, in Third MIT Conference on Computational Fluid and Solid Mechanics, K. J. Bathe, ed., New York, June 2005, Elsevier, pp. 617–622.

[10] S. J. Cummins and M. Rudman, *An sph projection method*, Journal of Computational Mechanics, 152 (1999), pp. 584–607.

[11] E. Erturk and B. Dursun, *Numerical solutions of 2-d steady incompressible flow in a driven skewed cavity*, ZAMM - Journal of Applied Mathematics and Mechanics, 87 (2007), pp. 377–392.

[12] J. Glimm, J. W. Grove, X. L. Li, and D. C. Tan, *Robust computational algorithms for dynamic interface tracking in three dimensions*, SIAM Journal for Scientific Computations, 21 (2000), pp. 2240–2256.

[13] J.-L. Guermond and L. Quartapelle, *A projection FEM for variable density incompressible flows*, Journal of Computational Physics, 165 (2000), pp. 167–188.

[14] J. Kim, P. Moin, and R. D. Moser, *Turbulence statistics in fully developed channel flow at low reynolds number*, Journal of Fluid Mechanics, 177 (1987), pp. 133–166.

[15] M. Lai, J. B. Bell, and P. Colella, *A projection method for combustion in the zero mach number limit*, in Eleventh AIAA Computational Fluid Dynamics Conference, AIAA, 1993, pp. 776–783.

[16] M. F. Lai, *A Projection Method for Reacting Flow in the Zero Mach Number Limit*, PhD thesis, University of California at Berkeley, 1993.

[17] H. Najm, P. S. Wyckoff, and O. M. Knio, *A semi-implicit numerical scheme for reacting flow*, Journal of Computational Physics, (1998), pp. 381–402.

[18] R. B. Pember, A. S. Almgren, W. Y. Crutchfield, L. H. Howell, J. B. Bell, P. Colella, and V. E. Beckner, *An embedded boundary method for the modeling of unsteady combustion in an industrial gas-fired furnace*, Tech. Rep. UCRL-JC-122177, Lawrence Livermore National Laboratory, October 1995.

[19] E. G. Puckett, A. S. Almgren, J. B. Bell, D. L. Marcus, and W. J. Rider, *A high-order projection method for tracking fluid interfaces in variable density incompressible flows*, Journal of Computational Physics, 130 (1997), pp. 269–282.

[20] S. P. Schofield, M. A. Christon, V. Dyadechko, R. V. Garimella, R. B. Lowrie, and B. K. Swartz, *Multi-material incompressible flow simulation using the moment-of-fluid method*, International Journal for Numerical Methods in Fluids, 63 (2010), pp. 931–952. (Los Alamos National Laboratory LA-UR-09-00733).

[21] M. Sussman, A. S. Almgren, J. B. Bell, P. Colella, L. H. Howell, and M. L. Welcome, *An adaptive level set approach for two-phase flows*, Journal of Computational Physics, 148 (1999), pp. 81–124.

[22] The Truchas Team, *Truchas physics and algorithms*, Tech. Rep. LA-UR-03-0166, Los Alamos National Laboratory, 2003.

[23] S. Tu and S. Aliabadi, *Development of a hybrid finite volume/element solver for incompressible flows*, International Journal for Numerical Methods in Fluids, 20 (2007), pp. 177–203.

[24] S. Tu, S. Aliabadi, R. Patel, and M. Watts, *An implementation of the spalart-allmaras des model in an implicit unstructured hybrid finite volume/element solver for incompressible turbulent flow*, International Journal for Numerical Methods in Fluids, 30 (2009), pp. 1051–1062.

[25] M. W. Williams, D. Kothe, D. Korzekwa, and P. Tubesing, *Numerical methods for tracking interfaces with surface tension in 3-D mold filling processes*, in Proceedings of FEDSM '02, Montreal, Canada, July 14-18 2002, 2002 ASME Fluid Engineering Division Summer Meeting, ASME.

[26] Y. Zang, R. L. Street, and J. R. Koseff, *A dynamic mixed subgrid-scale model and its application to turbulent recirculating flows*, Physics of Fluids A, 5 (1993), pp. 3185–3196.