Enabling HPC Simulation Workflows for Complex Industrial Flow Problems

Cameron W. Smith
Rensselaer Polytechnic Institute
110 8th Street
Troy, NY 12180
518-276-3401
smithc11@rpi.edu

Steven Tran
Rensselaer Polytechnic Institute
110 8th Street
Troy, NY 12180
518-276-6795
trans@rpi.edu

Onkar Sahni
Rensselaer Polytechnic Institute
110 8th Street
Troy, NY 12180
518-276-8560
sahni@rpi.edu

Farhad Behafarid
Rensselaer Polytechnic Institute
110 8th Street
Troy, NY 12180
518-276-6795
behaff2@rpi.edu

Mark S. Shephard
Rensselaer Polytechnic Institute
110 8th Street
Troy, NY 12180
518-276-8044
shephard@rpi.edu

Raminderjeet Singh
Indiana University
2709 E 10th Street.
Bloomington, IN 47408
812-856-0762
ramifnu@iu.edu

ABSTRACT
The use of simulation based engineering taking advantage of massively parallel computing methods by industry is limited due to the costs associated with developing and using high performance computing software and systems. To address industries ability to effectively include large-scale parallel simulations in daily production use, two key areas need to be addressed. The first is access to large-scale parallel computing systems that are cost effective to use. The second is support for complete simulation workflow execution on these systems by industrial users. This paper presents an approach, and set of associated software components, that can support industrial users on large-scale parallel computing systems available at various national laboratories, universities, or on clouds.

Categories and Subject Descriptors
I.6.7 [**Simulation and Modeling**]: Simulation Support Systems – *environments;* D.2.13 [**Software**] Reusable Software – r*eusable libraries*

Keywords
simulation based engineering, science gateway, parallel workflows

1. INTRODUCTION
High performance computing hardware and software has been advanced to the point that it is possible to address many of the large-scale simulations that are needed to advance science, address national needs and design superior products. The execution of massively parallel simulations at national laboratories and in academia is becoming the norm, while industry, which has a real need to employ these methods, is not applying them on a regular basis. The bottom line reason that industry is not using massively parallel simulations on a daily basis is exactly the reason one would imagine it to be – it's the time and cost. Although the current analysis software is capable of performing the simulations required, it is not integrated into simulation workflows that are of industrial quality and robustness. Thus industrial attempts to use such software tools become labor intensive and end up taking a substantial amount of time of an industrial person, even of those who are modeling and simulation experts and most experienced. This in-turn makes the process too time consuming and costly for the industry. In addition, industry access to simulation workflows that can be executed on massively parallel computers requires the use of complex remote systems.

This paper discusses two areas of development underway to make it possible for industry to apply massively parallel simulation in a timely, and cost, effective manner for execution of complete industrial relevant simulation workflows. The first is the integration of a set of parallel components to support the application of unstructured mesh-based simulation workflows on massively parallel computers. Emphasis is placed on being able to accept geometry-based problem specifications including engineering design parameters and to automatically perform all steps associated with an unstructured mesh simulation on massively parallel computers in order to provide reliable simulation results. Section 2 discusses a set of components that can effectively be coupled with unstructured mesh analysis codes to support parallel adaptive simulations. The second development is a web-based gateway that provides industrial users an easy to use interface for the execution of these simulations on massively parallel computer systems. Section 3 discusses the use of XSEDE science gateway technologies to implement the interface. Section 4 provides two example applications of these simulation capabilities.

2. Components for Parallel Adaptive Simulation of Using Unstructured Meshes
End-to-end automated parallel adaptive simulation workflows require the interactions of multiple software components. Figure 1 shows the set of components needed for reliable unstructured

XSEDE '15, July 26 - 30, 2015, St. Louis, MO, US A

ACM 978-1-4503-3720-5/15/07..$15.00
DOI: http://dx.doi.org/10.1145/2792745.2792786

mesh simulations with a focus on the role of the parallel mesh infrastructure. To effectively support the integration with multiple parallel analysis components, as well as alternative meshing and visualization technologies, the parallel mesh structures and services interact through functional interfaces [3][18]. These interfaces (or APIs) are designed specifically for an in-memory passing of information that is needed between different simulation components in going from the problem specification to the simulation results. At the highest level simulation information is specified in terms of attributes on a domain description of the problem, typically a CAD model, and parameters defining the physics model. Boundary-based geometric model and mesh representations, building on the abstraction of topological entities and their adjacencies, are ideally suited for the specification of, and maintaining of, the relationship between domain descriptions [4][11][14]. Problem description attributes are transformed into mathematical fields specified over geometric sub-domains, through the relationships between the geometric model and mesh while physics model parameters are transformed for selection of partial differential equations and discretization methods. Combined, this information is used to determine the desired output fields. In the case of a mesh-based simulation, the domain information is discretized (transformed) into a mesh that is adapted during the simulation. The mesh-based analysis discretizes the mathematical model and solves the resulting algebraic systems to determine vectors of unknowns that correspond to distribution coefficients for solution fields discretized over the mesh.

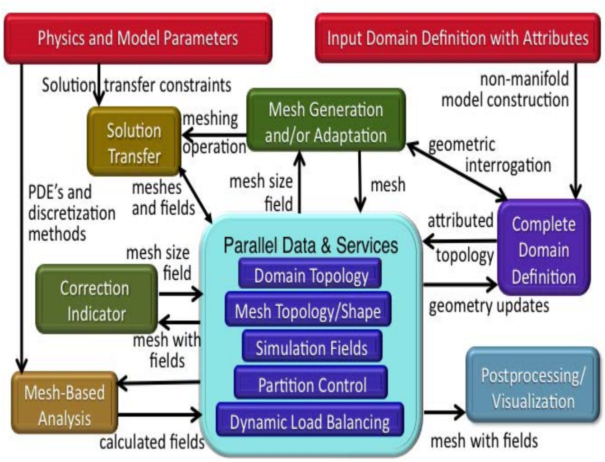

Figure 1. Components for parallel, adaptive mesh-based simulation workflows.

A summary of the components in the simulation workflow include:

Domain definition: Non-manifold boundary representations supported by CAD systems provide an effective representation of the geometric domain that can also be effectively coupled with generation/adaptation and analysis attributes for the simulation workflow [3][23][24]. An effective component for supporting the analysis model is Simmetrix' GeomSim [25][26] with functional interfaces to several CAD systems (NX, Pro/E, SpaceClaim, and SolidWorks) and modeling kernels (Parasolid [19], ACIS [1], and Granite [13]). It has capabilities to correct poorly defined geometry, combine multiple CAD models to define proper non-manifold models, automatically remove small features, when desired, and create complementary domains.

Parallel mesh infrastructure: This infrastructure houses the discretization of the domain, the mesh, in a distributed manner for use by the analysis procedures to solve the discretized mathematical models over the domain of interest. By maintaining the linkages between both the geometric model and the simulation fields, the parallel mesh infrastructure supports the operations of automatic mesh generation and adaptive mesh control procedures that can support the automated execution of reliable simulations. The parallel mesh infrastructure must provide a means to support a full set of mesh information in a distributed manner [22] as well as to support the operations required to maintain that representation as the parallel distribution is changed and/or the mesh is modified [30].

Parallel mesh-based analysis: The procedures used to solve the mathematical physics problems of interest must execute in a scalable manner and be able to effectively deal with complex anisotropic meshes appropriate for the physics problems of interest. The PHASTA code being used in the examples discussed in Section 0 meets the requirements to solve complex flow problems on adaptively constructed meshes [7][31] including meshes as large as 92 billion elements solved on 786,432 cores [21]. Other parallel analysis codes with which the current parallel mesh infrastructures has been integrated include fluid flow codes of FUN3D [10] and Proteus [16], Albany for solid mechanics [6], ACE3P for electromagnetics [17], and M3D-C1 for plasma MHD [15].

Parallel mesh generation and adaptation: To avoid the mesh generation bottleneck it must be fully automated and include methods for creating desired mesh types and gradations including anisotropic meshes being driven from a geometric model based specification of the mesh control information. In addition, procedures to automatically adapt the meshes for the purpose of controlling discretization errors, tracking specific solution features and accounting for evolving geometry are needed. The Simmetrix' MeshSim components are used for the meshing needs of the examples discussed in Section 0. Some meshes generated by MeshSim are shown in Figure 2 while Figure 3 shows an adaptively defined mesh.

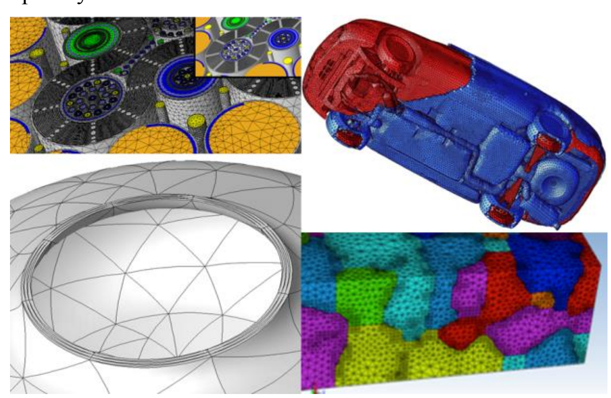

Figure 2. Clockwise from upper left: Reactor core mesh; boundary layer mesh on complex geometry; mesh on an evolving material microstructure; curved elements with multiple elements through the thickness.

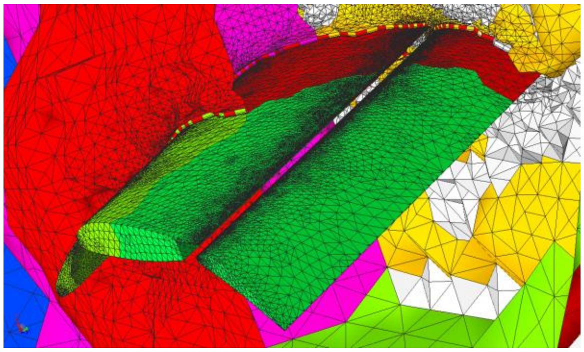

Figure 3. Adapted mesh of NASA's trap wing with boundary layer adaptation.

The mesh generation and adaptation procedures operate in parallel using distributed memory message passing [8][9]. A one billion element mesh was created on a CAD model in about six minutes on 224 processors. Meshes with up to 13 billion elements have been generated on 2048 processors. The parallel mesh adaptation procedures have created meshes as large as 92 billion elements on 786,432 cores [21].

Dynamic load balancing: One of the requirements of scalability parallel calculations is the ability to quickly, in parallel, repartition already distributed meshes to maintain load balance as the mesh evolves and/or different operations are carried out using the mesh. The Zoltan library provides a number of options for geometric and graph based partitioning to execute global load balancing [5][33]. The repeated execution of superior graph-based global load balancing becomes costly at large core counts. In addition, graph-based procedures can only account for information that can be directly mapped into a specific graph or hypergraph. In an effort to provide for more effective dynamic load balancing the library of Partitioning using Mesh Adjacencies, ParMA [27][29], is being developed to supplement the Zoltan partitioners. ParMA, combined with Zoltan geometric and graph based methods, has produced partitions of meshes with several billion elements with entity imbalances less than five percent on over one million processors [27].

3. Science Gateway

Execution of a component based workflow for PHASTA computational fluid dynamics simulations on HPC systems is through a web-browser based science gateway. A science gateway is a community-developed set of tools, applications, and data collections that are integrated through a portal or a suite of applications. These gateway technologies support development of an easy to use interface for running automated parallel PHASTA simulations on HPC systems that abstracts away complexities such as data management, job schedulers, and run-time environment setup. The PHASTA science gateway was created using the PHP Reference Gateway for Airavata (PGA) [2] and is hosted in the XSEDE gateway hosting environment [32]. PGA is a general-purpose gateway framework developed to enable scientific application in a browser environment. It provides user management, application cataloging and experiment management.

Simulations, called 'experiments' in the gateway, are defined by uploading a set of input files that specify the problem definition, simulation parameters, and required compute resources. depicts the experiment creation interface. The problem definition inputs include the complete definition of the analysis domain via the geometric model, typically from a CAD software such as SolidWorks, NX, etc., and the unstructured mesh. Also included are simulation specification information associated with the appropriate entities defining the domain, such as physical attribute information (e.g., loads, boundary conditions, material properties) and simulation parameters (e.g., initial mesh control, time steps, convergence requirements, solver options, etc.). Lastly, the compute resource inputs specify the HPC system, TACC's Stampede system for this example, the node and core count, and the max run time.

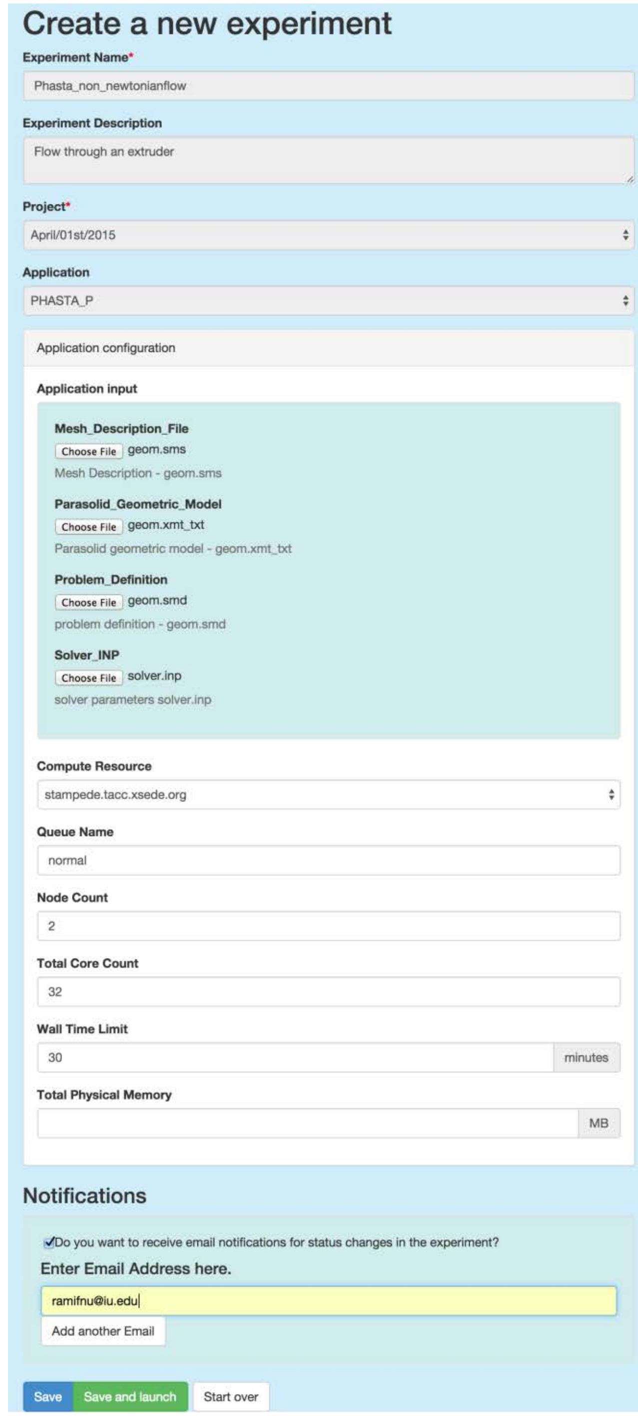

Figure 4. PHASTA gateway experiment creation.

Once the experiment is defined, clicking the 'Save and launch' button shown in will execute the PHASTA workflow. The PHASTA workflow is composed of three basic steps: 1) Pre-processing reads in the unstructured mesh, geometric model, and problem definition information and produces the data structures that PHASTA needs for analysis execution. 2) Partitioning and load-balancing increases the size of the partition as needed for execution on the specified number of processors (N), and balances the number of degree of freedom holding mesh entities that are assigned to each part for efficient analysis execution [27][29]. 3) Flow analysis execution on N processors. An additional fourth step that performs in-memory mesh adaptation may be included in the workflow.

The experiment execution request is supported through APIs provided by SciGaP [20]. SciGaP APIs process the user request, create a job scheduler script specific to a compute resource (PBS, SLURM, etc.), and monitor the status of a job, as shown in Figure 5. SciGaP also supports email notifications triggered by job status changes; an important mechanism for effective interactions with scheduled HPC systems. At the end of an experiment, the SciGaP service moves the outputs to the PGA storage location for users to download. In the future, we will integrate with data movement services like Globus online [12] to enable direct data movement to user systems.

Figure 5: PHASTA gateway experiment management interface.

4. Application to Complex Industrial Flow Problems

We present two complex industrial flow problems to demonstrate the component based simulation workflows discussed in Section 1. The first problem involves a twin-screw extruder, which contains two intermeshing rotating screws mounted on shafts in a closed barrel. The second problem case includes modeling of active flow control devices on rotor blades, in which the complex physics occurs where the geometry is the most complex. For both these problems the PHASTA gateway discussed in Section 3 is critically important for enabling industrial user interactions with these massively parallel simulation technologies.

The geometry of the first problem is composed of complex curved surfaces with corners and thin gaps in between them at which critical physics occurs as the raw material is pushed through the extruder. High aspect-ratio semi-structured boundary layer meshes are constructed over these complex surfaces, including in thin gaps, and appropriately graded into the general unstructured mesh in the remainder of the domain. The mesh in different regions of the domain is shown in Figure 6 and the corresponding numerical solution is shown in Figure 7. For this problem with multiple threads of the screw and tighter gaps, meshes can range from 20 to 50 millions elements in order to obtain accurate solution. In-turn to obtain the solution in a reasonable timeframe, less than a few hours, flow analysis component requires on the order of 500 cores, which is available on the XSEDE systems. The gateway based simulation workflow accounts for this.

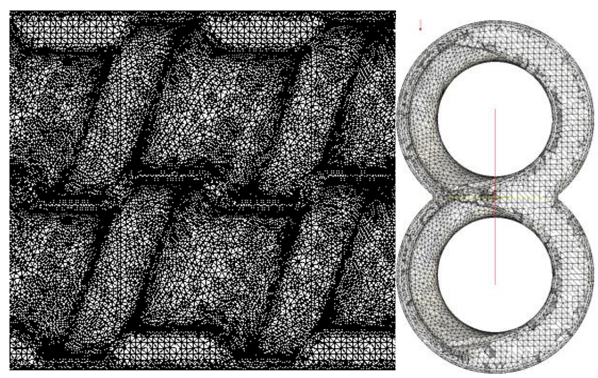

Figure 6. Twin-screw extruder mesh: (left) two threads of the screw and (right) cross-section across the extruder.

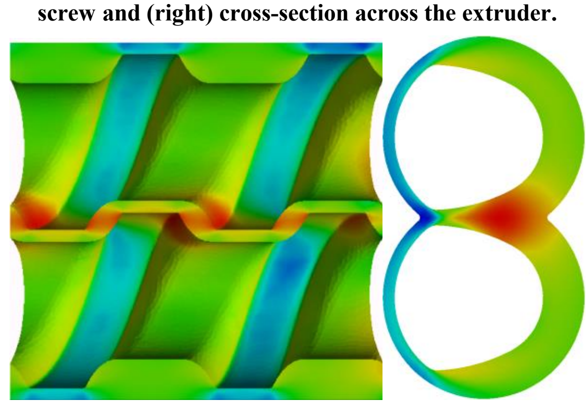

Figure 7. Twin-screw extruder axial velocity: (left) two threads of the screw and (right) cross-section across the extruder.

In the second problem the focus is on active mitigation of the dynamic stall phenomenon that is encountered on rotorcraft blades in a forward flight. This problem case is of critical importance because dynamic stall phenomenon is a significant factor in limiting the maximum speed capability and the flight envelope of the rotorcraft. This is especially true for rotorcrafts in high-speed and high-thrust conditions, where the air flow behavior over a blade is tremendously different between advancing and retreating sides of the blade. During dynamic stall large fluctuations and hysteresis are observed in aerodynamic forces and moments. Thus, active flow control mechanisms to mitigate dynamic stall [28] are very beneficial to expand the maximum speed and flight envelope for rotorcrafts. To accurately capture the flow physics associated with active flow control devices strongly graded elements, from small highly anisotropic elements near the device to relatively large elements in other portions of the domain, are created. These relatively small flow control devices are present within the boundary layer region, which is effectively resolved in the simulation only with the use of highly anisotropic semi-structured meshes. In turn, drastic transition occurs in lateral mesh sizes in the vicinity of the flow control devices as seen in Figure 8. Such a mesh leads to an effective resolution of the relevant flow structures resulting from the flow control device, see Figure 9. Since the Reynolds number in this problem case is in few millions, large meshes with up to 250 million elements are required especially for performing large-eddy simulations in order to resolve the relevant fine-scale turbulent structures. As with the previous problem, to obtain solution in a reasonable timeframe on these meshes the gateway executes the simulation workflow such that the flow analysis is carried out on large core counts, i.e., on the order of 2,500 cores – as available on the XSEDE systems.

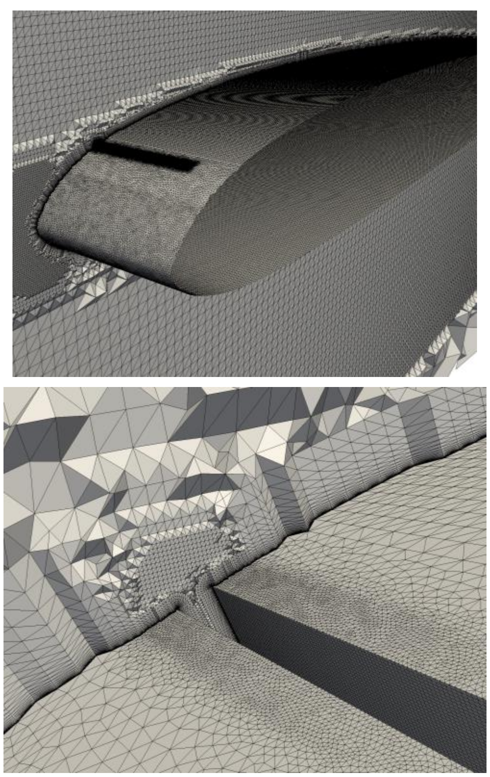

Figure 8. Synthetic jet on a blade (near the leading edge): (top) surface and cross-sectional mesh and (bottom) mesh around a synthetic jet orifice.

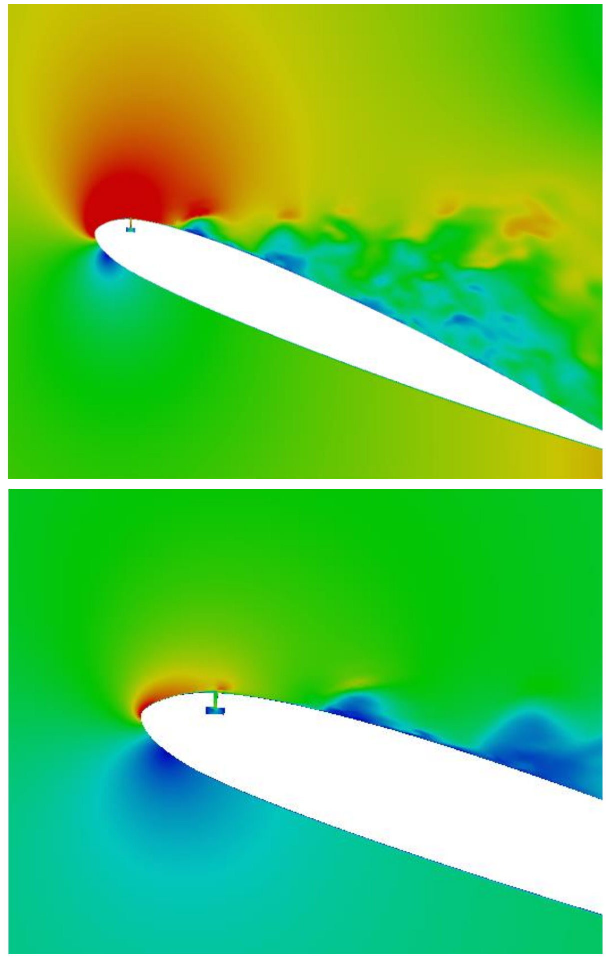

Figure 9. Flow field around a synthetic jet on a blade.

5. Closing Remarks

Applying leading HPC software and hardware systems to solve industrially relevant problems requires cost effective means to define and execute scalable simulation workflows. We define industrial workflows using a set of mature interoperable software components based on the unstructured mesh methods favored by industry. Industrial users easily execute the workflows on massively parallel computers using a single simplified web based interface. The ability to execute these capabilities on industrially relevant engineering design problems that can only be effectively addressed using massively parallel calculations has been demonstrated.

Near-term gateway developments will be driven by user feedback. One such development will satisfy industrial data retention requirements for sensitive intellectual property by supporting deletion of simulation data after execution. Future efforts towards increased simulation workflow efficiency will include integration of in-memory mesh adaptivity, and parallel mesh generation.

6. ACKNOWLEDGMENTS

Integration of the unstructured mesh simulation workflows with the XSEDE Science Gateway is supported by an XSEDE Industry Challenge Program project entitles “HPC Simulation Workflow to for Engineering Innovation”. Efforts on the development of parallel mesh-based simulation components is supported in part by the U.S. Department of Energy, Office of Science, Office of

Advanced Scientific Computing Research, under award DE-SC00066117 (FASTMath SciDAC Institute) while efforts to develop HPC workflows for industry is supported in part by the National Science Foundation under grants OCI-1126125 and IIP-1237555, and Empire State Development, Division of Science, Technology & Innovation through the NY State High Performance Computing Consortium (https://hpc2.org/). We gratefully acknowledge the use of the resources at the Texas Advanced Computing Center at the University of Texas at Austin, the Rensselaer Center for Computational Innovations and the Leadership Computing Facility at Argonne National Laboratory.

7. REFERENCES

[1] ACIS website. 2015. www.spatial.com/products/3d-acis-modeling.

[2] Airavata PGA: https://github.com/apache/airavata-php-gateway

[3] Beall, M.W., Walsh, J., Shephard, M.S. 2004. A comparison of techniques for geometry access related to mesh generation, *Engineering with Computers*, 20(3):210-221

[4] Beall, M.W., Shephard, M.S. 1997. A General Topology-Based Mesh Data Structure, *Int. J. Num. Meth. Engng*., 40:1573-1596.

[5] Catalyurek, U.V., Boman, E.C., Devine, K.D., Bozdag, D., Heaphy, R.T., 2009. A repartitioning hypergraph model for dynamic load balancing, *J. Parallel Distributed Computing*, 69:711-724.

[6] Chen,Q., Ostien, J.T., Hansen, G. 2014. Development of a Used Fuel Cladding Damage Model Incorporating Circumferential and Radial Hydride Responses, *Journal of Nuclear Materials*, 447(1-3):292-303.

[7] K.C. Chitale, O. Sahni, M.S. Shephard, S. Tendulkar and K.E. Jansen. 2015. "Anisotropic adaptation for transonic flows with turbulent boundary layers", *AIAA Journal*, 53(2):277-295, 2015.

[8] de Cougny, H.L., Shephard, M.S. 1999. Parallel Refinement and Coarsening of Tetrahedral Meshes, *Int. J. Numer. Meth. Engng.*, 46:1101-1125.

[9] de Cougny, H.L., Shephard, M.S. 1999. Parallel volume meshing using face removals and hierarchical repartitioning, *Comp. Meth. Appl. Mech. Engng*., 174(3-4):275-298.

[10] FUN3D website, http://fun3d.larc.nasa.gov/index.html.

[11] Garimella, R.V. 2002. Mesh data structure selection for mesh generation and fea applications, *Int. J. Numerical Methods Engineering*, 55(4):451:478.

[12] Globus Online: https://www.globus.org/

[13] PTC Granite website. 2015. http://www.ptc.com/product/creo/3d-cad/parametric/extension/granite-interoperability-kernel

[14] Haimes, R., Crawford, C. 2003. *Unified Geometry Access for Analysis and Design*, *Proc. 12 Int. Meshing Roundtable*, SAND-2003-3030P, pp. 21-32.

[15] S.C. Jardin, N. Ferraro, X. Luo, J. Chen, J. Breslau, K.E. Jansen and M.S. Shephard. 2008. The M3D-C1 approach to simulating 3D 2-fluid magnetohydrodynamics in magnetic fusion experiments, *Journal of Physics: Conference Series*, Volume 125, 012044, 7 pages, 2008.

[16] Kees, C. E., Farthing, M. W., & Bazilevs, Y. 2011. Parallel computational methods and simulation for coastal and hydraulic applications using the Proteus toolkit. In *Supercomputing 11: Proceedings of the PyHPC11 Workshop*.

[17] Luo, X.-J., Shephard, M.S., Lee, L.-Q., Ge, L., Ng, C., (2011). *Moving Curved Mesh Adaptation for Higher Order Finite Element Simulations*, Engineering with Computers, 27(1):41-50.

[18] Ollivier-Gooch ,C., Diachin, L.F., Shephard, M.S., Tautges, T., Kraftcheck, J., Leung, V., Luo, X.-J., Miller, M., 2010. An Interoperable, Data-Structure-Neutral Component for Mesh Query and Manipulation, *Transactions on Mathematical Software*, 37(3)29:1-29:28.

[19] Parasolid website. 2015. http://www.plm.automation.siemens.com/en_us/products/open/parasolid/

[20] Pierce, M., Marru. S., Demeler, B., Singh, R., and Gorbet, G. 2014. The apache airavata application programming interface. *In Proceedings of the 9th Gateway Computing Environments Workshop (GCE '14)*. IEEE Press, Piscataway, NJ, USA, 25-29. DOI=10.1109/GCE.2014.15

[21] M. Rasquin, C. Smith, K. Chitale, E.S. Seol, B. Matthews, J. Martin, O. Sahni, R. Loy, M.S. Shephard and K.E. Jansen. 2014. Scalable fully implicit finite element flow solver with application to high-fidelity flow control simulations on a realistic wing design, *Computing in Science and Engineering*, 16(6):13-21.

[22] Seol, E.S., Shephard, M.S., 2006. Efficient distributed mesh data structure for parallel automated adaptive analysis, *Engineering with Computers*, 22(3-4):197-213.

[23] M.S. Shephard and C.W. Smith. 2014. HPC Simulation Workflows for Engineering Innovation, *XSEDE '14*, Atlanta, GA, USA ACM 978-1-4503-2893-7/14/07. http://dx.doi.org/10.1145/2616498.2616556

[24] M.S. Shephard, J.E. Kolb and C. Smith. 2014. Bringing HPC to Engineering Innovation. *Computing in Science & Engineering*, 15(1): 16-25, Jan./Feb. 2013, DOI: 10.1109/MCSE.2012.128.

[25] M.S. Shephard, C.W. Smith, D.A. Ibanez, B. Granzow, M.W. Beall, and S. Tendulkar. 2015. Developing scalable components for massively parallel adaptive simulations, *Proc. NAFEMS World Congress,* San Diego, CA.

[26] Simmetrix website. 2015. http://www.simmetrix.com.

[27] Smith, C.W., Rasquin, M., Ibanez, D., Jansen, K. E., Shephard, M.S. 2015. Application Specific Mesh Partition Improvement, Submitted *SIAM Journal on Scientific Computing*.

[28] Tran, S.A, Fisher, A.E., Corson, D., and Sahni, O., 2015. Dynamic Stall Alleviation for an SC1095 Airfoil using Synthetic Jet Actuation, AIAA 2015 SciTech Meeting, AIAA Paper 2015-1038.

[29] Zhou, M., Sahni, O., Shephard, M.S., Devine, K.D., Jansen, K.E., 2010. Controlling unstructured mesh partitions for massively parallel simulations, *SIAM J. Sci. Comp*., 32(6):3201-3227.

[30] Zhou, M., Xie, T., Seol, E.S., Shephard, M.S., Sahni, O., Jansen, K.E. 2012. Tools to Support Mesh Adaptation on Massively Parallel Computers, *Engineering with Computers*, 28(3):287-301.

[31] M. Zhou, O. Sahni, H.J. Kim, C.A. Figueroa, C.A. Taylor, M.S. Shephard, and K.E. Jansen. 2010. Cardiovascular Flow Simulation at Extreme Scale. *Computational Mechanics,* 46:71-82, DOI: 10.1007/s00466-009-0450-z

[32] XSEDE gateway hosting: https://portal.xsede.org/knowledge-base/-/kb/document/ayfa

[33] Zoltan website (2015). http://www.cs.sandia.gov.