An Automated High Aspect Ratio Mesher for Computational Fluid Dynamics

Phase II Final Report

Submitted by:

Ciespace Corporation
P.O. Box 4291
Lisle, IL 60532-9291
Email: cray@ciespace.com
http://www.ciespace.com

Contract Number: T8.01-9986
Report Number: NNX11CC71C

August 9, 2013

Project Summary

Whereas our Phase 1 work demonstrated successfully the feasibility of the proposed computational framework for generating a high aspect ratio mesh for computational fluid dynamics (CFD) in 2D domains, the Phase 2 work has extended it to 3D domains. The Phase 2 work has also commercialized the new solution by turning the developed mesher into an application programming interface (API) product for high aspect ratio meshing for CFD. At the heart of the work in Phase I and Phase II is a novel solution framework, Ciespace-CMU High Aspect Ratio Mesher (CHARM), for adaptation of meshes and CFD solutions using an automated high aspect ratio mesher with three core technologies: metric tensor conditioning, metric-tracing mesher, and cell-packing mesher.

Our Phase II project has pursued and successfully achieved five objectives: (1) Development of a computational method for the automatic generation of highly anisotropic three-dimensional (3D) meshes that align an input tensor metric; (2) Development of a 3D meshing API product that supports anisotropic mesh regeneration functionality; (3) Realization of an adaptive meshing framework that combines the developed anisotropic mesh generation method and CFD solvers, at least one NASA solver and one commercial solver; (4) Completing a feasibility study on the GUI software that supports highly anisotropic 3D meshing for CFD; (5) Further refinement of software development best practices to ensure continual improvement in product quality – functionality, usability, reliability, predictability and supportability.

The key innovation from the Phase 1 and Phase 2 work can also be extended and applied to non-CFD applications such as structural, thermal, noise, vibration, and electro-magnetic simulation.

Table of Contents

1 Identification and Significance of the Innovation

Computational fluid dynamics (CFD) simulations are routinely used while designing, analyzing, and optimizing air- and spacecraft. An important component of CFD simulations is mesh generation, or discretization into polygonal or polyhedral cells, of the domain being analyzed. The overall computational cost and accuracy of simulations depend heavily on mesh quality – the size, shape, and structure of the cells. Another important aspect of CFD simulation is that solutions can be achieved iteratively, with each subsequent pass decreasing error and increasing solution accuracy. This is facilitated by grid adaptation, in which output from the last simulation is used to improve the mesh for the next.

The primary innovation achieved in the Phase II effort is an adaptive remeshing framework called CHARM – the Ciespace-CMU High Aspect-Ratio Mesher. CHARM is capable of generating high aspect-ratio meshes given solver-based input in the form of a metric tensor field, and combines a specialized meshing algorithm for critical regions with a packing-based algorithm (BubbleMesh®) for the other regions. This combination allows the generation of quality meshes that can handle the directionality and high aspect-ratio requirements that are encountered in adaptive remeshing problems. The meshing framework, CHARM, has been developed as part of an API product that can coordinate flow-solver / mesh generation iterations with NASA's FUN3D CFD solver and an open source CFD solver, OpenFOAM.

The developed CHARM framework has enabled the generation of high aspect ratio, 2D and 3D meshes for computational fluid dynamics (CFD). While the creation of such a high aspect ratio mesh itself is a major technical challenge, integrating the mesher into an adaptive mesh and solution refinement iteration loop is another major technical challenge, worked on in the Phase II project. Figure 1 illustrates the developed solution framework, Ciespace-CMU High Aspect Ratio Mesher (CHARM), for adaptive refinement of meshes and CFD solutions using an automated high aspect ratio mesher.

The developed framework is advantageous over previous methods, which are typically based on the advancing front or Delaunay triangulation/tetrahedrization, in four aspects:

- The proposed mesher can create a high aspect ratio mesh without user's intervention. In a complex flow simulation, where an ideal mesh anisotropy and directionality change over the domain, it is not easy to create a high-performance mesh manually.
- The proposed metric tensor conditioner refines a noisy and erroneous metric tensor so metric-tracing mesher and cell-packing mesher can perform well together. Such metric tensor conditioning will accelerate the convergence of a CFD solution and improve the solution accuracy.
- The proposed metric-tracing mesher can create meshes with an extremely high aspect ratio. This enables the precise capturing of flow features such as bow shocks and wakes.

- The proposed cell-packing mesher can create quadrilateral meshes for two-dimensional domains and hex-dominant meshes for three-dimensional domains. These orthogonal mesh elements increase the accuracy of a CFD solution while keeping the element count lower than conventional triangular or tetrahedral meshes.

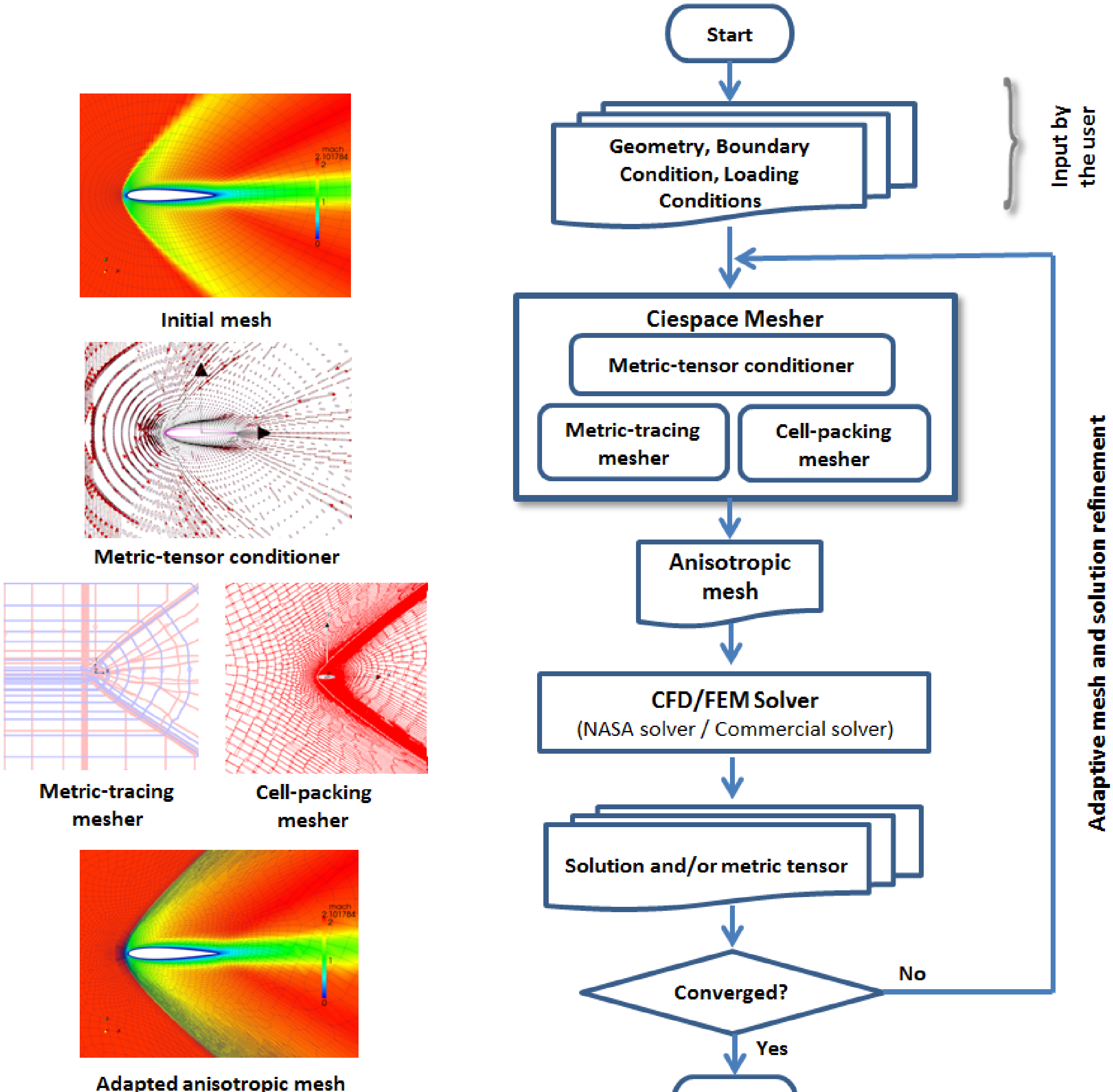


Figure 1: Ciespace-CMU High Aspect Ratio Mesher (CHARM): the proposed framework of adaptive refinement of mesh and CFD solution using Ciespace-CMU's automated high aspect ratio mesher. At the heart of the mesher are three innovative computational methods: metric tensor conditioning, metric-tracing mesher, and cell-packing mesher.

2 Technical Objectives

The technical objectives for this project are to:

- Objective 1: Develop a computational method for the automatic generation of highly anisotropic three-dimensional (3D) meshes that align an input tensor metric.
- Objective 2: Develop a 3D meshing API product that supports anisotropic mesh regeneration functionality.
- Objective 3: Realize an adaptive meshing framework that combines the developed anisotropic mesh generation method and CFD solvers, at least one NASA solver and one commercial solver.
- Objective 4: Perform a feasibility study on the GUI software that supports highly anisotropic 3D meshing for CFD.
- Objective 5: Refine further software development best practices to ensure continual improvement in product quality – functionality, usability, reliability, predictability and supportability.

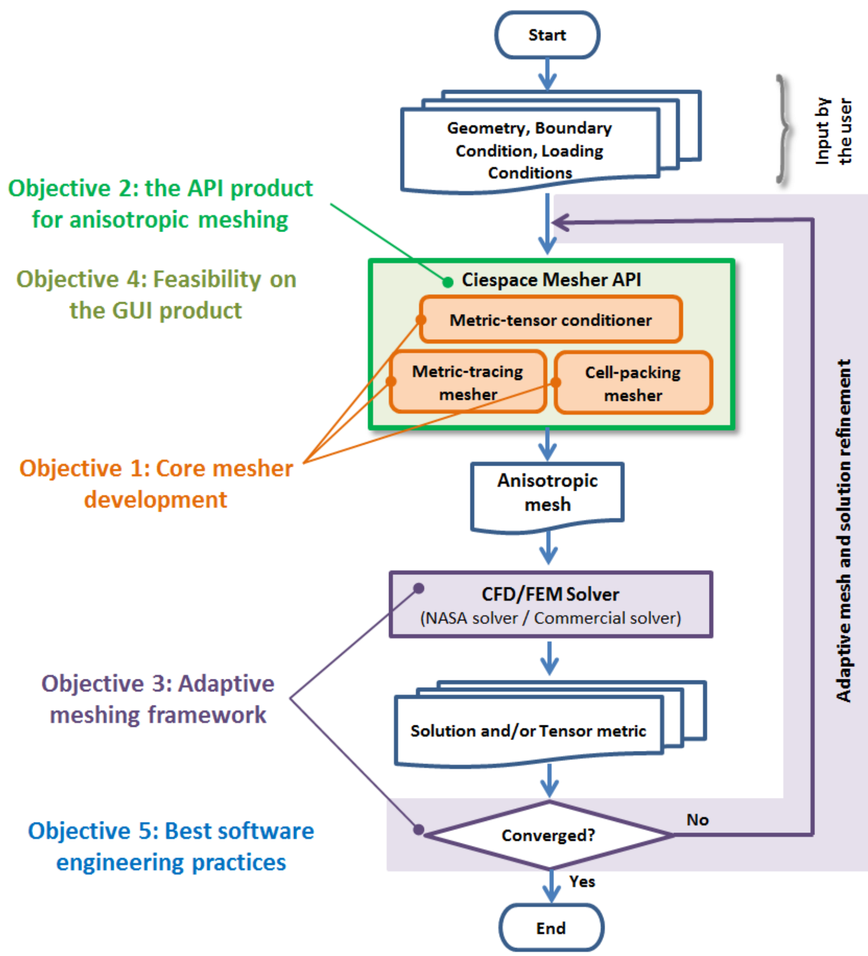


Figure 2: Five technical objectives of the Phase 2 work.

3 Work Plan

The Phase II work was completed by a team of:

- four individuals from the SBC: John Cray (JC), Marc Zinck (MZ) / Dr. Forest Rouse (FR), Tomotake Furuhata (TF), and Dr. Perry Miller (PM), and
- three individuals from the RI: Dr. Kenji Shimada (KS), Ved Vyas (VV), and Dr. Soji Yamakawa (SY)

3.1 Tasks

To meet the technical objectives of the project, the Phase II work was broken up into seven tasks:

1) **Tensor Metric**: Define the mathematical representation of a 3D tensor metric, and develop computational methods processing the tensor metric for highly anisotropic meshing for CFD.

    a. Define the mathematical representation of a 3D tensor metric and visualize the tensor metric (RI: VV, SY, KS; SBC: FR, TF)

    b. Develop computational methods for conditioning a 3D tensor metric such as fairing, blending and resampling (RI: VV, SY)

    c. Verify that the pre-conditioners perform well with noisy 3D tensor metric generated by CFD solvers (RI: VV, SY)

2) **Core Meshers**: Develop two core meshing methods: metric-tracing mesher and cell-packing mesher

    a. Design an integration method for metric-tracing mesher and cell-packing mesher (RI: VV, SY, KS; SBC: FR)

    b. Develop the metric-tracing mesher with highly anisotropic meshing (RI: VV, SY)

    c. Implement modification to packing based mesher to control mesh anisotropy and directionality (RI: SY, VV)

    d. Prototype, implement and test fine grain parallelism in core mesher to efficiently utilize multi-core compute nodes. (SBC: FR, TF; RI: SY, VV)

    e. Perform feasibility study investigating meshing using large scale parallelism in computing environments with thousands of nodes. (SBC: FR, PM, TF; RI: VV, SY)

3) **API Product**: Develop a meshing API product by combining metric –tracing mesher and cell-packing mesher.

   a. Define the high-level specifications of the API product based on the general requirements for integrating the API with CFD solvers (SBC: FR, TF, RI: KS, SY, VV)

   b. Define the detailed specifications of the API product (SBC: FR, TF, RI: SY, VV)

   c. Develop the API product that combines the metric-tracing mesher and the cell-packing mesher (SBC: FR; RI: VV, SY)

   d. Ensure that all the data format and numerical methods support sufficiently high precision for anisotropic meshing with an extreme aspect ratio of 1:10,000 and higher (SBC: FR, TF; RI: SY, VV)

   e. Verify the performance of the API product with test CFD problems (SBC: FR; RI: VV, SY, KS)

   f. Prepare user manuals of the API product (SBC: JC, FR)

4) **Solver Integration**: Realize an adaptive meshing framework that combines the developed anisotropic mesh generation method and NASA's CFD solvers as well as commercial FEM solvers

   a. Identification of at least one target NASA CFD solver (RI: KS; SBC: JC)

   b. Identification of at least one target commercial FEM solver (SBC: JC; RI: KS)

   c. Exchange binary data between the grid generation software and the solvers. (SBC: FR, PM, TF; RI: VV)

   d. Explore the use of both adjoint and feature-based mesh adaption. (RI: VV, SY)

5) **GUI Product Feasibility**: Conduct a feasibility study for the future GUI product in preparation for Phase 3.

   a. Start a conceptual design of the future GUI product in preparation for Phase 3 (SBC: JC, PM, FR; RI: KS)

   b. Identify unique requirements in the GUI product for highly anisotropic meshing for CFD (SBC: JC, PM, FR; RI: KS)

   c. Develop a partial prototype of the GUI product (SBC: PM)

6) **Software Engineering**: Use software development best practices to ensure continual improvement in product quality – functionality, usability, reliability, predictability and supportability

a. Use Ciespace software engineering processes and tools including source code repository system, build systems, issue tracking, static code analysis, coverage analysis and profiling. (SBC: FR, PM; RI: VV; SY)

b. Add software to the Ciespace continuous integration process which includes automated building, testing and delivery package generation for each repository revision. (SBC: FR, PM)

c. Identify a set of representative test cases to track algorithm improvements and add test cases to the Ciespace regression test suite. (SBC: JC, FR, PM)

d. Develop a plan to support the software in Ciespace's meshing API, the Meshing Toolkit. (SBC: JC, FR, PM)

e. Conduct periodic reviews of the software engineering process with the goal of finding efficiencies and improvements. (SBC: FR, PM, TF; RI:VV, SY)

7) Project Management

a. Conduct progress meetings every 3 months between the NASA CFD researchers and users, Ciespace R&D and CMU researchers. (SBC: JC, KK, FR, TF, PM; RI: KS, VV, SY)

b. Hold progress meetings every 2 weeks between Ciespace R&D and CMU researchers. (SBC: JC; FR, TF, PM; RI: VV, SY)

c. Oversee technical direction of project through regular meetings with CMU research staff. (SBC: JC, FR; RI: KS, VV)

d. Write and deliver interim report. (SBC: JC, FR; KS, VV, SY)

e. Write and deliver final and technical report. (SBC: JC, FR; RI: KS, VV, SY)

f. Build, test and deliver final Phase 2 version of meshing software. Conduct quarterly builds and test according to the milestones. (SBC: FR, PM)

g. Update marketing and business plans (SBC: JC; RI: KS)

3.2 Meeting the Technical Objectives

As demonstrated by the methods and results in Section 4, the proposed technical objectives have been satisfied in the following ways:

Develop a computational method for the automatic generation of highly anisotropic three-dimensional (3D) meshes that align to an input metric tensor field:

- Data structures are available to store metric tensor fields on unstructured meshes. Furthermore, methods have been implemented to provide fundamental operations such as interpolation on these metric tensor fields. Additional methods provide more extensive capabilities such as conditioning to remove noise and improve the directional “fairness” of such fields. Finally, metric tensor fields can be visualized using glyphs (symbols) that depict local anisotropy and directionality as well as integral manifolds such as streamlines that indicate how directionality varies over a domain.

- Work has been performed on the core meshers, based on cell-packing (BubbleMesh®) and metric-tracing, to support generation of anisotropic meshes based on metric tensor fields that are derived from solver outputs.

Develop a 3D meshing API product that supports anisotropic mesh regeneration functionality:

- A C++ API has been developed based on a component model. The API is designed for extensible and cross-platform use, among other benefits. The API covers geometry, meshing, solvers, and adaptation. Components in each category can be extended and even swapped out with alternatives developed by end-users of the API product.
- In particular, the cell-packing mesher is fully exposed through this API and can generate anisotropic meshes for CFD.

Realize an adaptive meshing framework that combines the developed anisotropic mesh generation method and CFD solvers, at least one NASA solver and one commercial solver:

- The target commercial solver and target NASA solver (FUN3D) have been integrated as solvers in the API.
- Adaptation logic has been implemented through components in the API product that can coordinate geometry, mesher, solver, and tensor field actions.

Perform a feasibility study on the GUI software that supports highly anisotropic 3D meshing for CFD:

- The feasibility study began with an effort toward conceptual design of the GUI. The produced designs covered CAE project management, workflow management, 3D model interaction, and collaboration.
- Requirements for CFD applications with anisotropy were determined in the context of geometry, meshing, and solutions.
- A web-based GUI was produced as part of Ciespace’s commercial strategy. The GUI leverages portions of the Ciespace API product on the backend to enable users to prepare geometry, generate CFD meshes, and generate CFD solutions.

Refine further software development best practices to ensure continual improvement in product quality – functionality, usability, reliability, predictability, and supportability:

- All of the practices detailed in Task 6 have been adopted.

3.3 Task Labor Categories and Schedules

The following table outlines the total time spent (in hours) for each of the major tasks defined for Phase II by person category:

Table 1: Task assignments and cost

REMOVED

Table 2 shows the task schedule assuming a 24-month project. Note that several of the Tasks, 6a, 7a, 7b, 7c, and 7d, are regular reoccurring tasks, like weekly meetings and source code commits into the software repository. These regular reoccurring tasks are not shown in Table 2.

Table 2: Task Schedule

Task Schedule							
Month	SBC				RI		
	JC	MZ	SD	TF	KS	VV	SY
1		6a	6a	6a	1a	1a, 6a	1a, 6a
2		1a		1a	1a	1a	1a
3	7a				7a	1b	1b
4	4a, 6c	6c	6c		4a	1b	1b
5	4a, 6c	6c	6c		4a	1c	1c
6	7a, 7d	7d			7a, 7d	1c, 7d	1c, 7d
Milestone 1	Completed Tasks: 1a, 1b, 1c, 2a, 3a, 4a						
7		3a, 2a		3a	2a, 3a	2a, 3a	2a, 3a
8	4b	3a, 6c	6c		4b	2b, 3a	2c, 3a
9	7a, 4b	3b, 6c	6c	3b	7a, 4b	2b, 2c, 3b	2c, 2b, 3b
10	4b	3b		3b	4b	2b, 2c	2c, 2b
11		3b				2b	2c
12	7a, 7d	3b, 7d			7a, 7d	2b, 7d	2c, 7d
Milestone 2	Completed Tasks 2b, 2c, 3b, 4b, 6c						
13	5a	3c, 5a	5a		5a	3c	3c
14	5a	3c, 4c, 5a	5a, 4c	4c		3c, 4c	3c
15	7a	3c, 4c, 5a	5a, 4c	4c	7a	3c, 4c	3c
16	5b	3c	5b		5b	3c	3c
17	3e, 5b	3d, 3e	5b		3d	3d, 3e	3d, 3e
18	7a, 7d, 3e	3d, 3e, 7d	5b		7a, 7d, 3e	3d, 3e	3d, 3d
Milestone 3	Completed Tasks: 3c, 3d, 4c, 5a, 5b						
19	3d, 7g	3e	5c		7g	4d	4d
20	3d, 7g	3e	5c		7g	4d	4d
21	7a, 3f	3f	5c		7a	4d	4d
22	6d, 7g	3f, 6d	5c, 6d	6d	7g	4d	4d
23	6d, 7g	6d	5c, 6d	6d	7g	4d	4d
24	7a, 7d, 7e	7d, , 7e, 7f	5c, 7e, 7f		7a, 7d, 7e	7e	7e
Milestone 4	Completed Tasks: 3e, 3f, 4d, 5c, 6d, 7f, 7g						

The milestones of the proposed project are:

Milestone 1 (at the end of 6 months) – The foundation for achieving the project objectives is established.

- The preparation steps are completed including the representation scheme of a 3D tensor metric (Task 1a), pre-conditioning of the metric. (Tasks 1b and 1c), and the integration method for the two core meshers (Task 2a).
- The concept design and high-level specifications of the API product are completed. (Task 3a)
- At least one target NASA CFD solver is identified. (Task 4a)

- Identify a set of representative test cases to track algorithm improvements and add test cases to the Ciespace regression test suite. (Task 6c)

Milestone 2 (at the end of 12 months) – Two core mesher components are completed, and the detailed API design is finalized.

- Two core meshers are developed: metric-tracing mesher (Task 2b) and cell-packing mesher (Task 2c).
- The detailed design of the API product is completed. (Task 3b)
- At least one target commercial FEM solver is identified. (Task 4b)

Milestone 3 (at the end of 18 months) – The API product is developed, and the data can be exchanged between the API product and selected CFD solvers. The conceptual design of the GUI prototype is completed.

- The API product that combines the metric-tracing mesher and the cell-packing mesher is developed. (Task 3c)
- The performance of the developed API product is verified with test CFD problems. (Task 3e)
- It is verified that the developed API product can handle anisotropic meshing with an extreme aspect ratio of 1:10,000 and higher. (Task 3d)
- The binary data exchange between the API product and the target CFD solvers is facilitated. (Task 4c)
- The conceptual design of the future GUI product is started, and the unique requirements in the GUI product for highly anisotropic meshing for CFD are identified. (Tasks 5a and 5b)
- Prototype, implement and test fine grain parallelism in core mesher to efficiently utilize multi-core compute nodes. Perform feasibility study using large scale parallelism in computing environments with thousands of nodes. (Tasks 2d and 2e)

Milestone 4 (at the end of 24 months) – Solver Integration is achieved, and the business plan is updated based on the API and GUI products. 3e, 3f, 4d, 5c, 6d, 7e, 7f, 7g

- The user manuals for the API product are written. (Tasks 3f)
- The two core meshers and the API product are tested with two types of input metric tensor: feature based and adjoint (Task 4d), and a plan to include the API into Ciespace meshing API is established (Task 6d).
- A partial prototype of the future GUI product is developed. (Task 5c)
- The final software and reports are submitted to NASA. (Tasks 7e, 7f and 7g)

4 Results of Innovation

Details of progress in various areas of this Phase II project are given in the following sections on metric tensors, CFD mesh generation, GUI development, and a study on large-scale parallelism.

4.1 Metric Tensor Representation, Conditioning, and Visualization

Various mathematical abstractions allow for the representation of mesh anisotropy and directionality requirements. The fundamental pieces of information considered here are:

1. An orthogonal set of directions that define desired mesh orientation
2. A corresponding set of desired mesh sizes along each direction

This information can be represented in multiple ways and can vary spatially. Furthermore, there are several operations that depend on this local information or its variation over the domain. Finally: certain representations may be more suitable for certain operations. This suggests conversion between representations is necessary.

Metric tensors are commonly used to generate anisotropic meshes. In three dimensions, these tensors can be stored as full 3x3 matrices (nine real values) or using just six values when exploiting their symmetry. The eigenvectors and eigenvalues of the metrics encode the desired directionality and anisotropy.

Solvers and adaptation codes, such as those in the FUN3D suite of tools, often compute and exchange metric tensor fields (metric tensors stored on a grid with some kind of support). They are amenable to interpolation and many other operations that are useful when generating anisotropic, adapted meshes. For these reasons, metric tensors and their decompositions have been adopted as primary representations for mesh requirements in our system.

As demonstrated in the previous report and during Phase I, there are several types of conditioning that need to be applied to the metric tensor fields that describe mesh directionality and anisotropy.

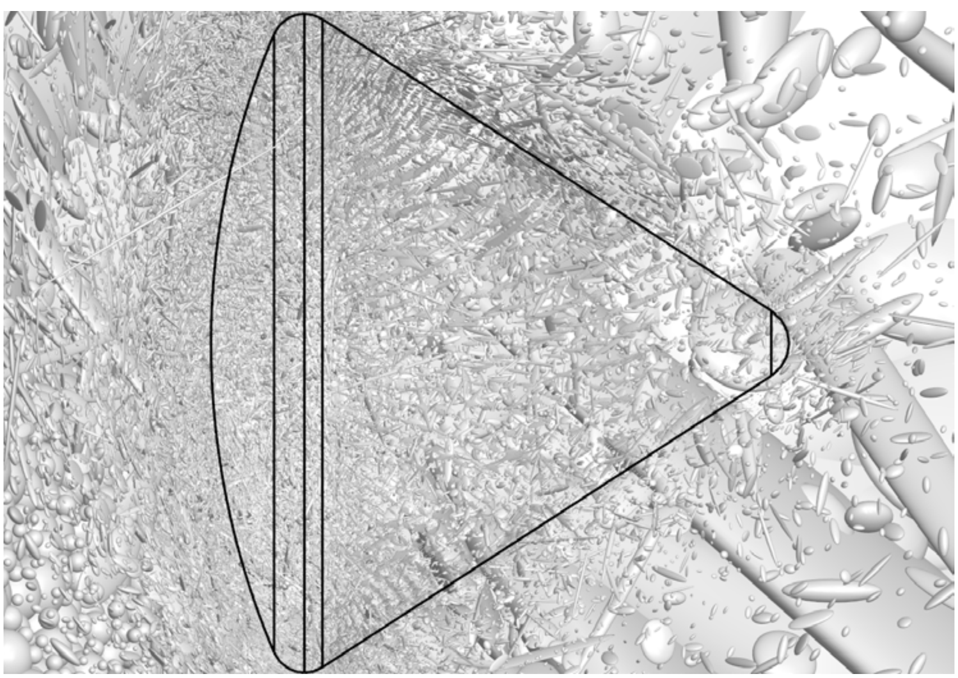
Some examples of necessary conditioning and the stages in which they are applied are:

- Inter-iteration: blending or limiting adaptation iteration to iteration changes in the field
- Meshing preparation: reducing noise and fairing the directionality of the field
- Integration of metric-traced mesh and cell-packed mesh: the metric field will be adjusted in the region between meshes generated by different algorithms to ensure a smooth transition

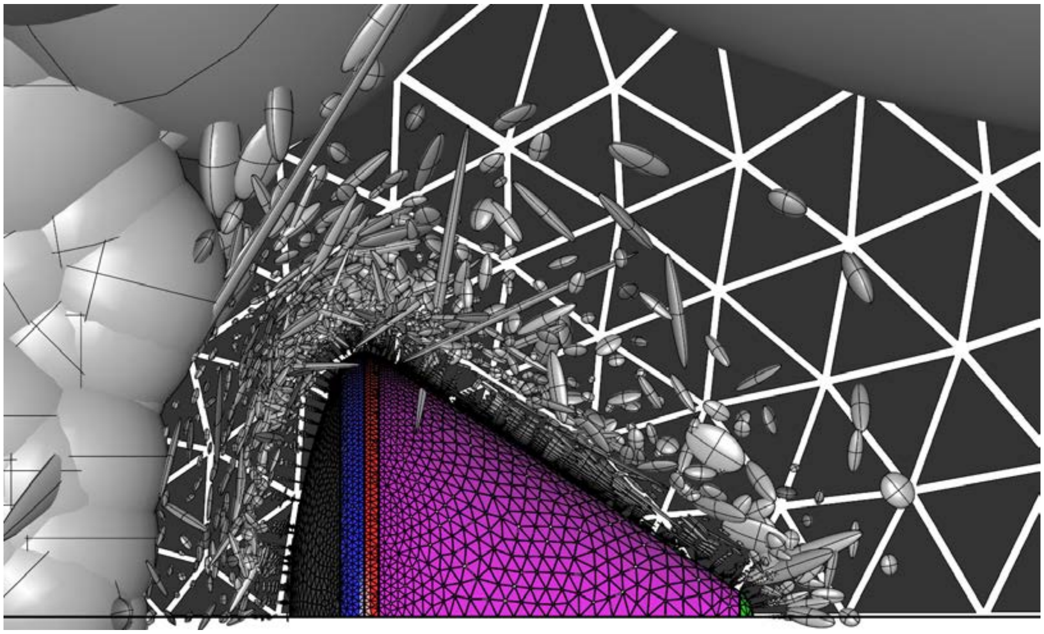
A 3D framework has been developed to support these forms of conditioning. The framework expands on our previous tensor field work in 3D and utilizes concepts from the Phase I work in 2D. Rudimentary operations such as interpolation and other queries are supported. Furthermore, fields can be synthesized based on user input and geometry information.

Most tensor field conditioning can be expressed as the application of local rules to a tensor stored on a node in a mesh and its neighboring tensors. A combination of number of levels (in the mesh), a local sphere radius, and a required minimum number of neighbors can be specified to generate the neighborhoods. This combination is designed to support PDE-based techniques that may utilize just the tensors on a node and its edge-connected neighbors, techniques that perform fitting (thus may require *n* samples) or filtering (arbitrary filter size), and techniques adapted from image-processing. The weighting applied to nodes/tensors in the neighbor can also be controlled. Some choices are uniform (unit), inverse distance, and Gaussian-based. Finally, some control is provided for conditioning steps that are iterative. The minimum and maximum number of global iterations can be controlled. If specified, these are hard constraints. Conditioning can also be terminated when, for instance, the maximum relative change in the tensor components (or invariants or other derived quantities) is below some threshold.

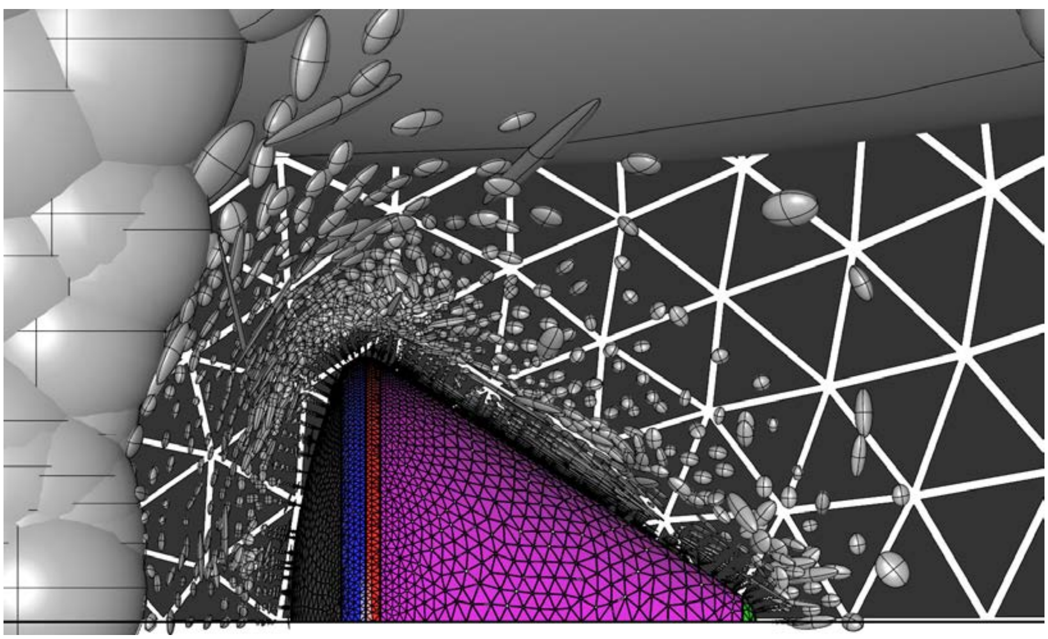
Figure 3 illustrates the results of noise removal using a Median filter in this conditioning framework. Shown is a cut-away view of the capsule model in which a layer of metric tensor ellipsoids is revealed. The tensor glyphs are drawn at the nodes of the underlying mesh at 0.35 scale. Discrete median filtering (described in previous reports) was applied using immediate neighbors (1-ring) over 1-5 global passes to successfully reduce noise in directionality and anisotropy.



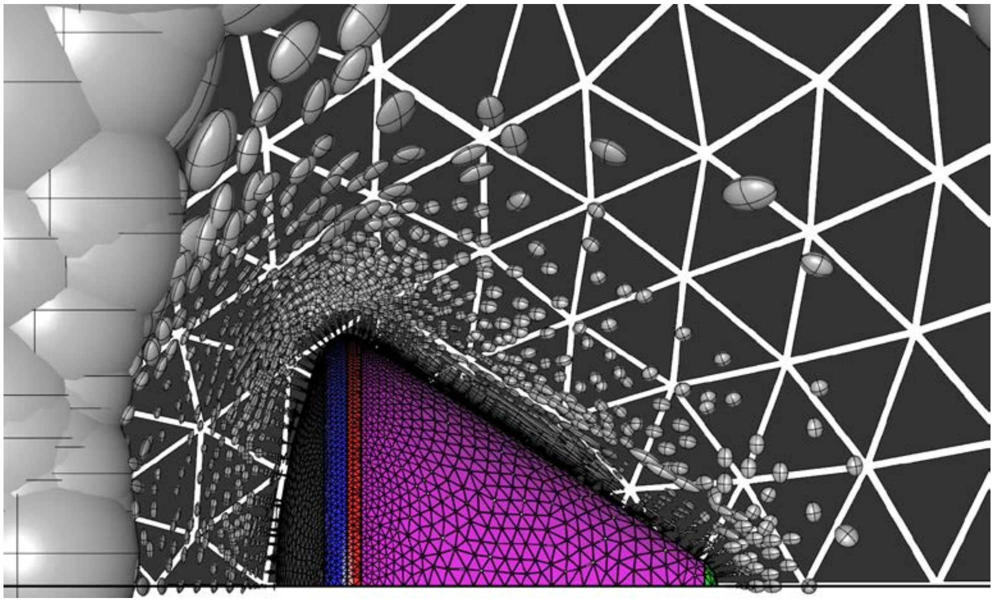
(a) Raw tensor data over whole domain



(b) Unfiltered input tensor data



(c) Median filter results (one pass)

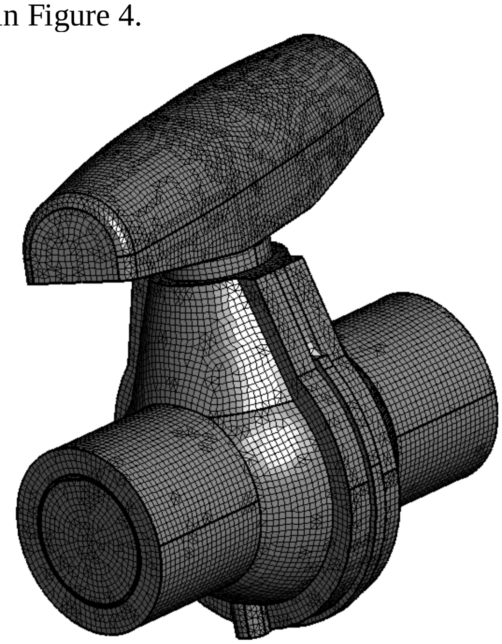


(d) Median filter results (five passes)

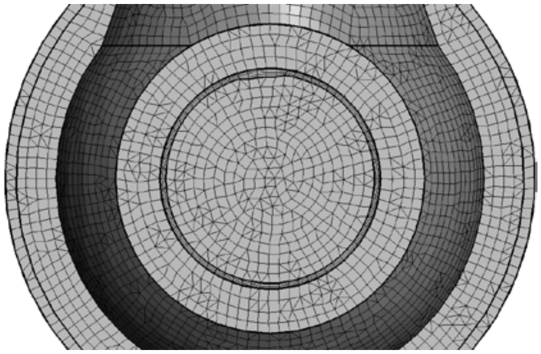
Figure 3. Noise removal results

4.2 CFD Mesh Generation using API Product

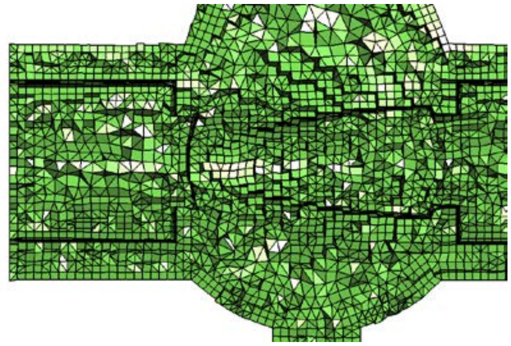
The current form of the API product is continually tested on various problems from our CFD test suite. The geometry, mesher, and solver components are exercised during the tests for quality assurance and performance monitoring purposes. This has also ensured that the API exposes sufficient options to address the specific needs of various analyses. Examples of various meshes and analysis results of a valve model, produced using the API product, are shown in Figure 4.



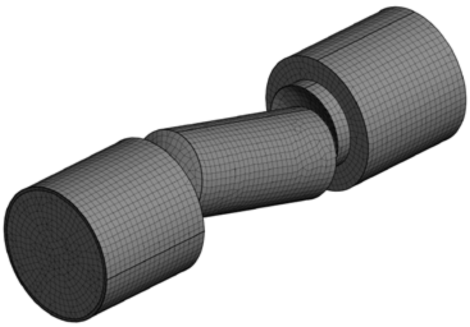
(a) Mesh of valve model fluid region and casing



(b) Side view of mesh

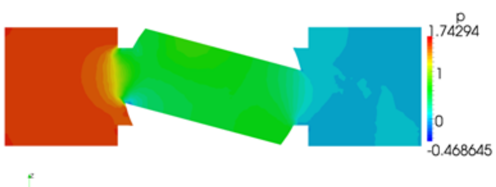


(c) Interior of both meshes

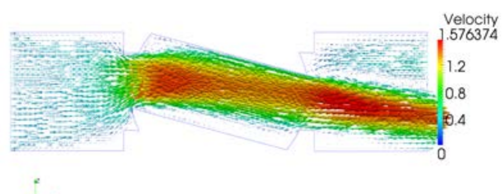


(d) Another hex-dominant mesh of the fluid region

(e) Interior of alternative hex-dominant mesh



(f) Pressure result from analysis on (d)



(g) Velocity field from analysis on (d)

Figure 4. Test CFD problem for API product verification

The results shown in Figure 4f and Figure 4g correspond to a steady, incompressible, turbulent (k-ε) flow with an inlet velocity of 0.3 and an outlet pressure of 0 (and zero-gradient at the inlet). The mesher was instructed to create boundary layers at the viscous walls, and additional types of mesh control can be specified as well.

4.3 GUI Product Feasibility

Over the course of this Phase II STTR, Ciespace has been engaged in an effort to develop an effective GUI product that leverages core functionalities in the areas of geometry, meshers, and solvers, and also collaboration and facilitates project management and collaboration.

Ciespace officially released version 1.0 of a web-based engineering services product near the end of this project. The user interface (UI) has three primary panes in an open project: the hierarchy and properties on the left, 3D views, and a workflow manager on the bottom.

The example in Figure 5 demonstrates an end-to-end workflow starting from a geometry imported into the system until the final post-processing visualization. The properties panel on the left reflects settings of the currently selected node (process) in the workflow, and activating a node in the workflow produces an appropriate visualization. The workflow describes sequences of operations on imported data and can be, edited, cloned, and branched among other things.

In the open project in Figure 5, a mesh generation node has been activated, revealing the properties panel and 3D mesh viewer. The Mesher1 node operates on the result of a series of steps including fluid volume extraction, model repair, and simplification. The properties expose control over mesh element types, boundary layer type and sizing, mesh sizing, regional controls, and more. It is easy to adjust properties and re-execute the nodes while exploring the space of parameters. Downstream nodes can be updated to use the latest output from their parents.

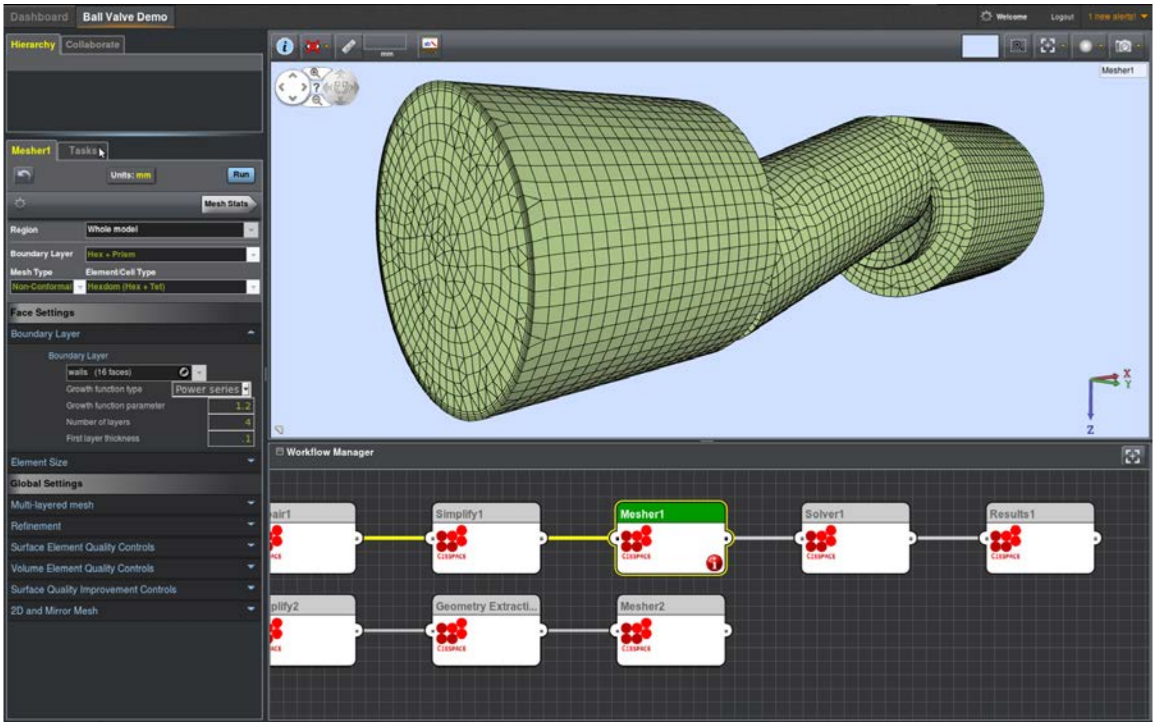


Figure 5. Ciespace v1.0 user interface with mesher node active

A solver node has been created from the results of the mesher node. Some of the available properties for solver nodes are displayed in Figure 6. Users can track residuals during the solve, and can perform visualization at solver checkpoints and completion.

Figure 6. Ciespace UI properties panel for solver node

The adaptation and high aspect ratio controls build on the UI design summarized here. This includes the addition of an adaptation node that regulates the adaptation process (i.e., convergence criteria, adaptation metric formulation, and adaptive meshing method) and provides progress information.

4.4 Large Scale Parallelism Feasibility Study

Followed by improvements to single-node, multithreaded performance of the core meshers was a feasibility study to evaluate the application of the meshers in a multi-node environment. After some consideration, variations of the following strategy were decided on:

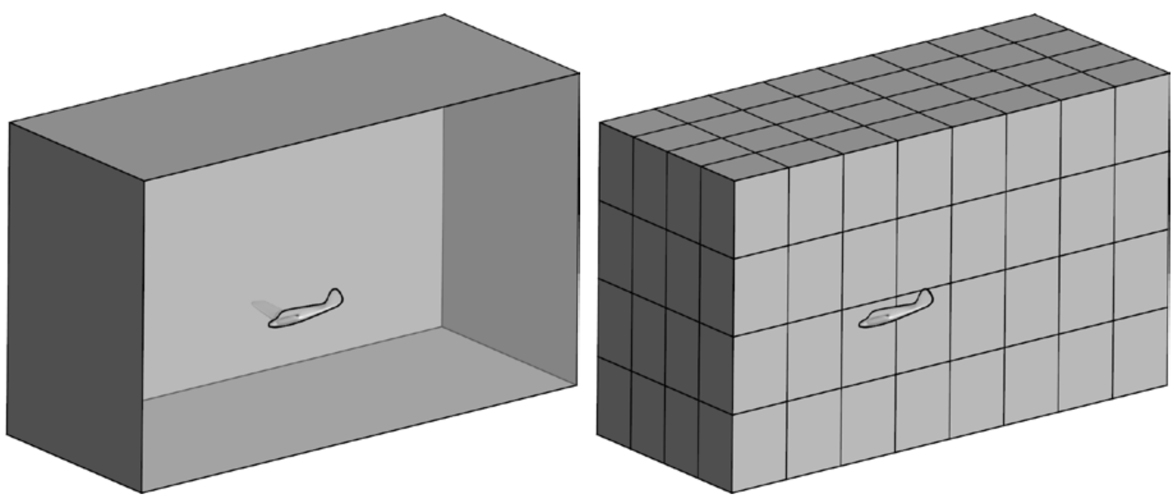
1. Decomposition of the input domain
2. Parallel meshing of the interface surfaces between adjacent partitions
3. Parallel volume meshing that maintains conformity to the interfaces

This strategy takes advantage of the current multi-core mesher without requiring significant architectural changes. Furthermore, its implementation may benefit from and benefit the current conformal assembly meshing capability since both are related. It also results in reasonably minimal inter-node communication, at least until the last step. The strategy appears to be feasible if a good partitioning and interface conformity may be obtained.

Because the input domain consists of a boundary representation only, the first step must produce a 3D decomposition by cutting the domain with surfaces. This must be sensitive to meshing requirements. For instance, the cutting surfaces may limit mesh quality based on the dihedral angles they form at intersections with the boundary.

Next, the set of cuts and the boundary produce a set of surfaces (including partition interfaces) that can be meshed in parallel. Finally, the volumes enclosed by each partition are volume meshed across all the distributed compute nodes. Each compute node will invoke multi-threaded operation of the mesher to perform the surface and volume meshing jobs it is given.

An external flow problem with symmetry was used for the preliminary study, as shown in Figure 7a. The input domain was successively decomposed into 2, 4, 8, 16, 32, 64, and 128 partitions (Figure 7b). The individual partitions were meshed sequentially on a single compute node using four threads and equivalent settings for each to ultimately produce a mesh on the order of 5-10 million cells. Thus, the distributed scenario was only simulated to a certain degree and the results do not include communication of partition data. The study assumes that the average of the CPU/wall-clock time covers represents the overall time for parallel execution of the meshers. Furthermore, conformal meshing of partition interfaces and reconstruction were not covered.

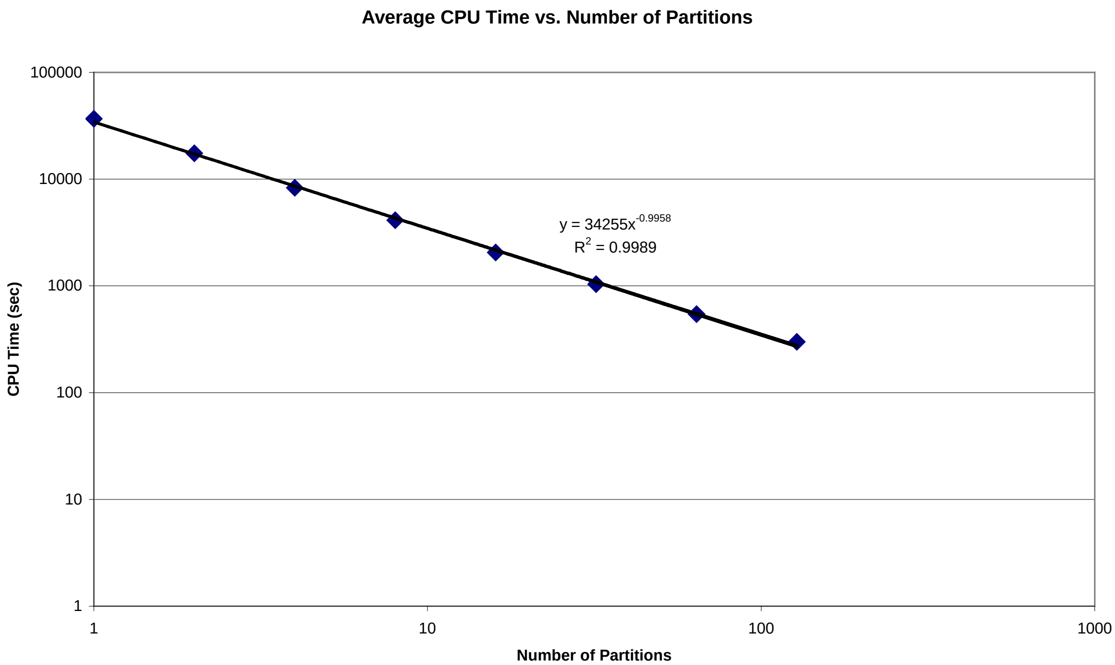


(a) Entire domain (b) Final decomposition into 128 partitions

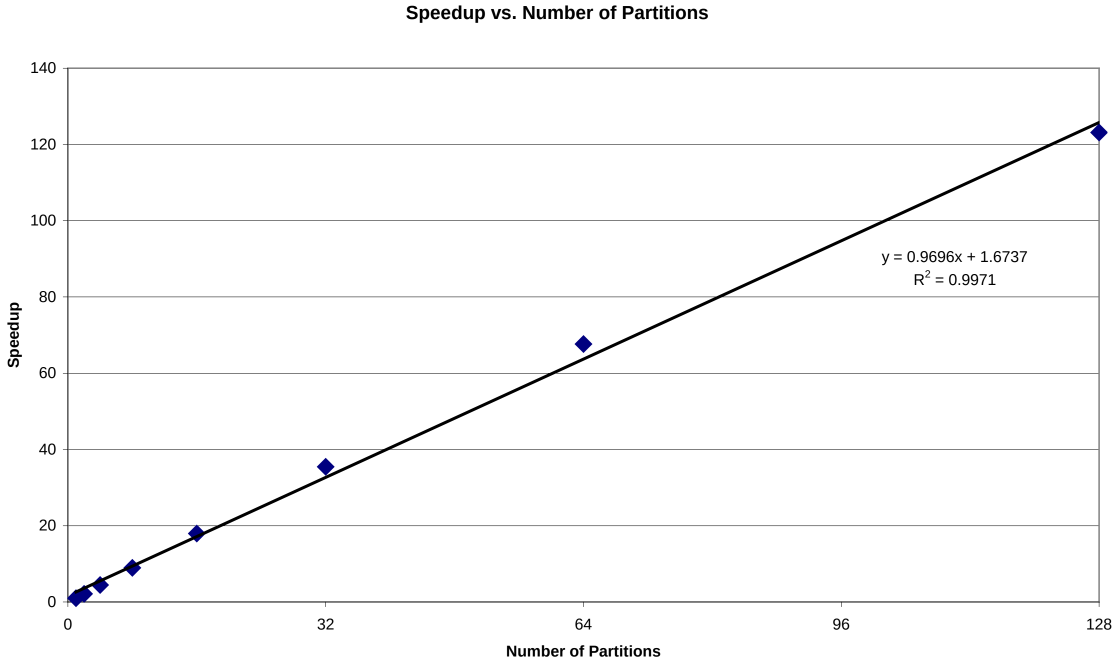
Figure 7. External flow domain for preliminary study

With these conditions in mind, a selection of the results from the study are plotted in Figure 8. These are based on the CPU and wall-clock times collected during each run. The first plot simply demonstrates that time to mesh decreases with decreasing problem size, as expected. Next, Figure 8b shows the speedup (calculated as time for a single-partition run divided by the time for the current number of partitions), which is essentially linear. Finally, an interesting note is on the CPU utilization measure, given by CPU time / number of CPUs / wall-clock time. Figure 8c indicates that CPU utilization is better with smaller problem sizes, which suggests both that single-node scability can be improved but also that an effective distributed strategy will be helpful.

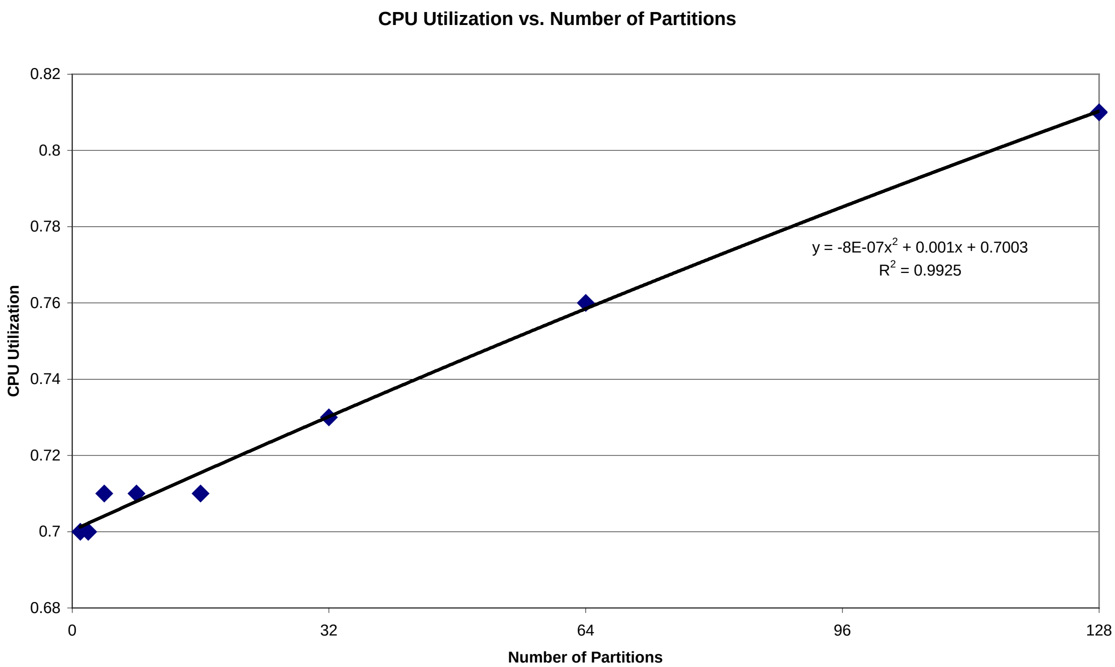
Finally, it is acknowledged that actually distributing and reconstructing data in the study will add sequential operations that limit the speedups presented so far. This is one of the topics that will be studied in more depth during Phase II-E, which in part is concerned with scalability to large problems.



(a) CPU time per “compute node” as number of partitions is increased



(b) Gained speedup as number of partitions is increased



(c) CPU utilization as number of partitions is increased

Figure 8. Performance results from preliminary study

5 Contacts

5.1 Key Contractor Participants

Participants from the SBC and RI (contractor) side have included the following:

- Dr. Kenji Shimada, Principal Investigator, Theodore Ahrens Professor of Engineering, Carnegie Mellon University
- John Cray, SBC PM / Point of Contact, Ciespace Corporation
- Marc Zinck, Developer, Ciespace Corporation
- Ved Vyas, Consultant to Ciespace Corporation, PhD in progress, Carnegie Mellon University
- Dr. Soji Yamakawa, Researcher, Carnegie Mellon University
- Tomotake Furuhata, Part-time Consultant to Ciespace Corporation
- Dr. Forest Rouse, Software Developer, Ciespace Corporation
- Dr. Perry Miller, Software Developer, Ciespace Corporation

5.2 Key NASA Participants

Participants from NASA have included the following:

- William Kleb, Technical Monitor / COTR, NASA Langley Research Center
- Kimberly Graupner, Administrative Liaison, NASA Langley Research Center
- Susan Tillman, Industrial Property Officer, NASA Langley Research Center

6 Technical Activities

6.1 Cumulative Technical Activities

The following activities were performed in the Phase II project:

- The representation scheme of a 3D metric tensor was defined.
- The pre-conditioning method for the metrics was developed.
- The cell-packing mesher has been fully integrated into the CHARM framework, and the metric-tracing mesher has been partially integrated.
- The concept design and high-level specifications of the API product were completed.
- Target NASA CFD solver was identified: FUN3D.
- A set of representative test cases to track algorithm improvements and add test cases to the Ciespace regression test suite were selected.
- The cell-packing mesher has been fully developed, and the metric-tracing mesher has been partially developed.

- The detailed design of the API product was completed.
- The target commercial FEM solver was identified: OpenFOAM.
- The API product was developed that fully exposes cell-packing mesher.
- The performance of the developed API product was verified with test CFD problems.
- The binary data exchange between the API product and the target CFD solvers was facilitated.
- The conceptual design of the future GUI product was started, and the unique requirements in the GUI product for highly anisotropic meshing for CFD were identified.
- Fine-grain parallelism in core mesher to efficiently utilize multi-core compute nodes was prototyped.
- The user manuals for the API product were written.
- The feature based metric tensor and the adjoint metric tensor were studied.
- A partial prototype of the future GUI product was developed.

6.2 Future Technical Activities

As illustrated in Figure 9, the Phase II technology and product have been developed through five Phase II Objectives. In the approved Phase II Enhancement project, we propose to enhance our Phase II technology and product in two respects:

- **Ability to handle larger-scale meshes with given memory storage**
- **Ability to yield more accurate solutions with improved solution convergence**

The former will be achieved by enhancing the two meshing components shown in Figure 9, the metric-tracing mesher and cell-packing mesher. The current components remesh the entire domain during each adaptation iteration of the illustrated loop. The drawback of this approach is rapid increase of computational and memory cost as the mesh is further refined. The proposed enhancement will minimize the computation cost and memory usage by remeshing only critical local regions.

The latter enhancement will be achieved by a scheme to transfer a previous solution onto the newly adapted mesh. This circumvents solver "cold-starts" in which the solution is regenerated from scratch in each adaptation iteration. The choice of transfer scheme and the immediate benefit of a better initial condition for the new solution have the potential to make the adaptation loop more computationally efficient and robust.

The proposed Phase 2 Enhancement project will modify and enhance the two core meshing components to support mesh adaptation through localized changes and modifications to an existing mesh. This also provides the opportunity to produce a viable metric-tracing mesher based on the research performed thus far in Phase II.

This new scheme has the potential to reduce resource utilization and improve mesh adaptation robustness. Furthermore, the reuse of solution values in unchanged portions of the mesh and remapping elsewhere may have a beneficial impact on per-iteration solution convergence and computation time.

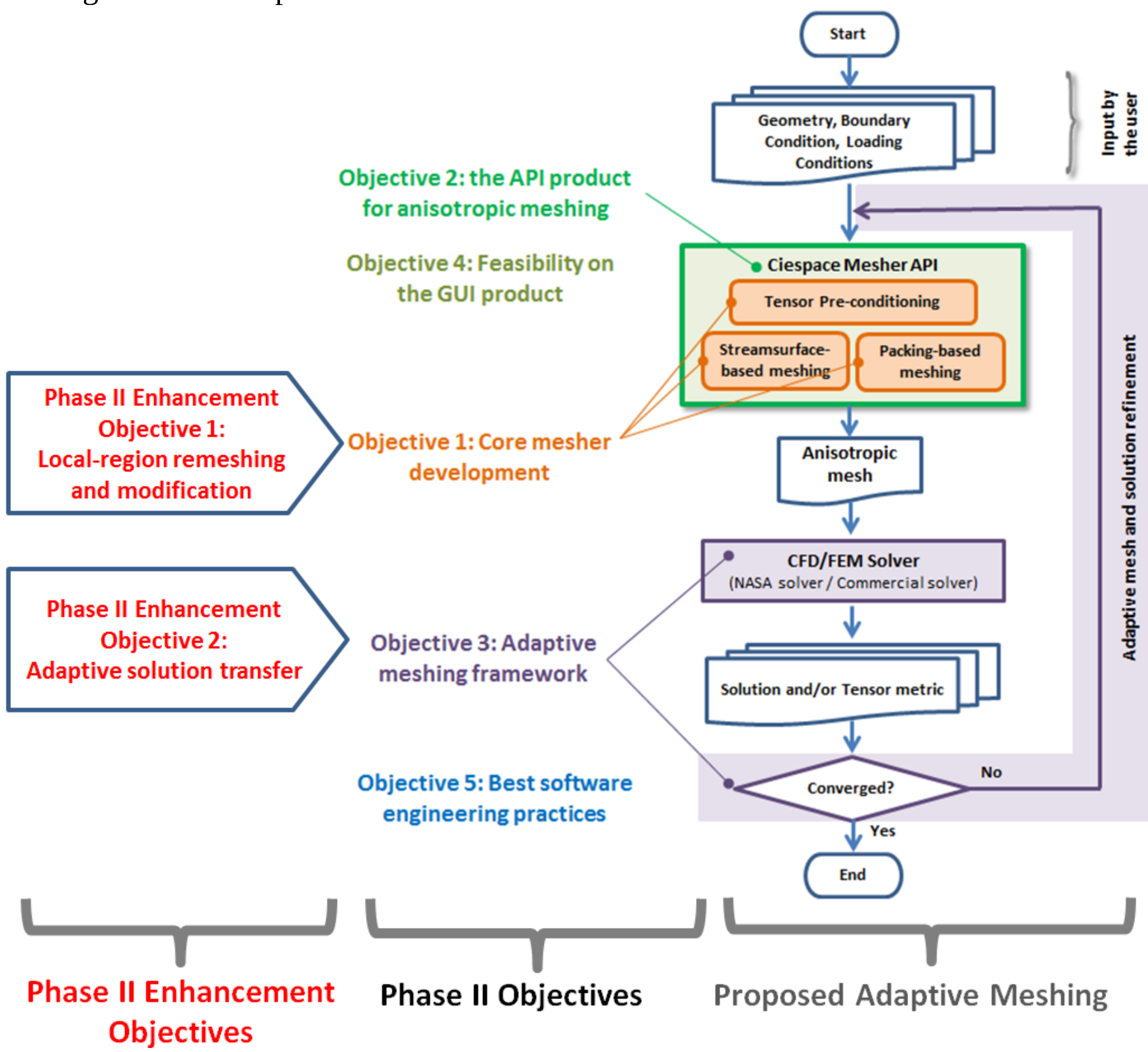


Figure 9. Objectives of the Phase II Enhancement project.

7 Potential Customer and Commercialization Activities

7.1 Cumulative NASA Potential Customer Activities

There are a number of NASA initiatives which could benefit from the Ciespace adaptive remeshing framework, including evaluation, design/redesign, and assessment of air- and spacecraft for current and future NASA missions. Since the current requirement is the use of a suitable CFD solver which can take mixed-element meshes as input for solution and provide metric output for adaptive refinement, activities for applicability would include:

- Application of the technology to programs already utilizing the FUN3D solver.
- Review and analysis of other CFD solvers used within the environment, for potential integration and usage of the Ciespace adaptive framework.
- Analysis of usage and value of high aspect ratio controllability, and/or increased quality derived from iterative solver refinement.

7.2 Cumulative Non-NASA Potential Customer Activities

Ciespace has already demonstrated successful partnership and cooperative R&D with Toray (3DTIMON), IBM (EMSURF), and NASA (FUN3D). The proposed innovation's solver-adaptive meshing capability supports even stronger differentiation, increasing further partnership and commercialization potential in cooperation with these firms.

Potential target customers for the innovation include:

- Other government programs, at agencies including DoD, Army Research, Naval Research, DARPA, and others, where high-variability physics (impact, explosion, collision, etc.) benefit from the ability to leverage high levels of directional, anisotropic control over finite elements, coupled with adaptive refinement to improve mesh and analysis quality.
- Targeting of other commercial users of the FUN3D solver, including CFD analysis projects at Honda Jet, SpaceX, Lockheed Martin, Boeing, BMI, Gulfstream, and Sikorski Aircraft.
- Increased usage and penetration into enterprise automotive, aerospace, and equipment manufacturing companies (companies exceeding $2B in annual revenue) in which Ciespace already has a footprint due to past relationships and BubbleMesh product usage or evaluation. These include Honda (U.S. and Japan); General Motors; BE Aerospace; John Deere; and Caterpillar.
- Other large enterprises in automotive, aerospace, and equipment manufacturing which Ciespace is approaching through a direct selling model. These include Chrysler; Ford; Boeing; McDonnell Douglas; Crown; NACCO; and Fiat.
- Mid-market companies (annual revenue between $200M and $2B) approached through direct referrals from large enterprises (as parts or equipment suppliers) and/or through other direct selling models.
- Indirect customers targeted through partnership with existing leading analysis/solver companies, enabling Ciespace to be the Preprocessing system

of choice for their customer base and leveraging the innovation of unique bi-directional solver integration. These partnerships include Toray (existing partner for 3DTimon solver with over 75 existing Ciespace customers); IBM (existing partner for EMSURF electromagnetic solver); Autodesk (for Moldflow solver, leveraging Ciespace injection molding experience); Solidworks (for Cosmos); ESI (across a suite of solvers for crash, vibration, metal forming, etc.); MSC (across a suite of solvers); etc.

- Indirect customers targeted through resellers in the U.S., Asia-Pacific, and Europe.

8 Resources Status

The following table details the accumulated work hours and costs for all resources assigned to the Phase II project:

REMOVED

The total cumulative costs incurred, as of 8/9/2013, are $599,996. As of this final report, 100% of the work (assigned tasks) has been completed.

9 References

1. V. Vyas and K. Shimada, “Metric-Field Conditioning for Mesh Adaptation Problems,” Proceedings of the 21st International Meshing Roundtable – Research Note, 2012.
2. “Ciespace Announces the Release of the CAE Industry’s First SaaS Application for 3D Engineering Design and Simulation,” http://www.ciespace.com/community/press-release/ciespace-cfd-product-launch/
3. “Ciespace Meshing – Meshing Tools for Engineering Analysis,” Ciespace White Paper, 2013 (http://www.ciespace.com/wp-content/uploads/Ciespace_Meshing_Whitepaper_040113.pdf)
4. “Ciespace CFD – An Open Cloud Platform for Engineering Analysis Powered by OpenFOAM,” Video Presentation of the Ciespace CFD product, 2013. (http://vimeo.com/59772964#)
5. “FUN3D – Fully Unstructured Navier Stokes,” Project Website (http://fun3d.larc.nasa.gov)