YEAR 3:

**Major activities**

Developments during year three focused on several aspects of this project including:

* Continued development of the general multiscale communication library, AMSI (Adaptive Multimodel Simulation Infrastructure), using the multiscale fibrous soft tissue model as a test bed.
* Completion of working (but unverified) version of two scale soft tissue model with fiber-only RVEs in which all parallelism is performed by AMSI.
* Execution of above mentioned model on AMOS supercomputer (IBM Blue Gene/Q) utilizing up to 8000 processes.
* Development of migration and other load balancing algorithms to improve scaling within both the general framework of AMSI and the soft tissue model.

The Adaptive Multimodel Simulation Infrastructure (AMSI) has been developed to provide a set of support libraries and functionalities for the development, implementation, and execution of general multimodel simulations. Key areas of functionality for AMSI include managing the parallel execution space of multimodel simulations, scale-linking communications, and dynamic scale-sensitive load-balancing operations.

AMSI provides runtime support for these key areas through maintaining a real-time minimal meta-data model of various qualities of the simulation, with respect to parallel execution requirements for simulation scale-tasks, scale-linking operations and the actual data representations of such scale-linking communications, and many additional quantities. This meta-data itself is intended to be generalized and extensible such that additional functionalities depending on an extended meta-model may be implemented by AMSI users, though at present the meta-data system is limited to managing those pieces of meta-data required for currently implemented AMSI functionality.

Over the past year the AMSI system has changed the multiscale parallel communication functionalities to be more generalized and less application-specific, as the initial implementation was tied closely to those functionalities needed by the primary test application. Additionally, the simulation meta-model was expanded in order to accommodate more dynamic runtime capabilities, such as the runtime scale-dependency sensitive load-balancing operations – including dynamic expansion/contraction of individual scale tasks as adaptive processes on one scale can result in the need to introduce and remove supporting scales – which have been developed and are currently being expanded and generalized. The expansion of the meta-model and the capabilities it allowed, as well as the performance of the test-application – on the AMOS (IBM BlueGene/Q) supercomputer at CCI/RPI with utilization up to 4000 cores – with respect to communication and parallel blocking was discussed in the “Parallel Infrastructure for Multiscale Simulations” minisymposia talk at the SIAM Parallel Processing Conference 2014.

This infrastructure has been integrated into the multiscale soft tissue model which now exclusively utilizes the general mechanisms supplied by AMSI for multiscale data communication instead of ad-hoc methods typically present in multiscale simulations. The general nature of AMSI allows it to be utilized in any parallel multimodel simulation, though at present it has – outside of test cases – been used exclusively with the multiscale soft tissue model which is an RVE-based hierarchical two scale model.

The above mentioned soft tissue model has been developed and is in a working state, but is still being verified. The current working version employs fiber-only RVEs at a dynamically-determinable subset of the integration points of the macroscale finite element analysis, constituting a two scale hierarchical model. The mechanical aspects of the current model are not novel in the current version, but the computer science aspects including AMSI’s provided support for multiscale communication and multiscale scaling and load-balancing concerns are. The current implementation supports several features not included in the previous iterations such as initial multiscale load balancing of RVEs and dynamic determination not only of whether an RVE should be used in lieu of a constitutive model at each engineering-scale numerical integration point, but also which type of RVE should be used. The verified coupled fiber-matrix RVE is currently being integrated into the simulation as an RVE choice. Software aspects have also been improved, including reduced memory footprints and a high-level interface specification for conducting finite element analysis (now included as part of AMSI) for use in the macroscale analysis, the fiber-matrix RVE, and a planned intermediate scale to be included in the near future.

The model has been run successfully on the AMOS supercomputer at CCI/RPI. Load balancing and migration of RVEs are being developed to improve runtimes and scalability. Weak scaling tests have been performed up to 8000 microscale processes and 32 macroscale processes and are presented below.

**Specific Objectives**

The specific objectives of this project are to:

* Define abstractions and methods that bridge physics and mathematics formalisms to the models and computational methods needed for component-based adaptive multimodel analysis. This infrastructure must maintain a clear understanding of all the relations and transformations executed and relate them to a multiscale design specification. Such capabilities are essential to tracking design sensitivities and uncertainties within a multiscale design process.
* Implement interoperable components that support the relations and transformations associated with the domain, model and field interactions of the abstracted simulation components.
  + Inclusion of high-level abstract interfaces supporting the implementation of finite element analysis codes in the AMSI system is one of the first concrete steps toward this goal. Generalizing this abstraction further to accommodate interactions between the specified areas is an area of future work. Fortunately this work has proven to be predominantly orthogonal to the work being conducted on other areas of AMSI, and usage of the multiscale parallel operations of the AMSI system in no way requires adherence to any specific interface or set of behaviors.
* Define and implement a methodology to supports the full range of adaptive model, scale linking and discretization control techniques needed for multiscale simulations.
* Develop multilevel dynamic load balancing techniques to support the scalable execution of adaptive multiscale simulations on massively parallel computers.
  + Current developments are focused primarily in the above two areas. Support for dynamic addition and deletion of supporting scales/models has been implemented, along with the capability to define how these modifications to the parallel structure effect load-balancing. Support for general dynamic migration of parallel scale-tasks is the current focus of implementation.
* Develop multiscale simulation applications that demonstrate the effectiveness of the tools and technologies developed.
  + The current primary test-application for the AMSI suite of tools is the fibrous soft tissure multiscale simulation, which has provided all AMSI related results (outside of fairly trivial hard-coded test cases which are designed to test the AMSI system, but do not provide a meanignful use-case) up to the present. Incorporation of AMSI facilities in additional multiscale simulation codes is a critical step in expanding development and functionality for AMSI.

RESULTS and achievements

Initial results of the AMSI parallel execution environment control and multiscale coupling communication systems were produced and presented at the SIAM Parallel Processing Coference in Portland, Oregon in February 2014. The lecture, entitled “Parallel Infrastructure for Multiscale Communication” discussed the implementation abstractions used by the AMSI system, how these abstractions applied to the implementation and execution of the soft tissue example case, and various results and metrics on the performance of the AMSI features being used to support the soft tissue application.

Additionally, since the SIAM conference in February, support for more dynamic management of the multiscale discretization of the parallel execution space has been implemented, and initial results generated for dynamic introduction/removal of supporting scales. An initial implementation for dynamic scale-sensitive load-balancing operations is currently nearing completion, and there should be results pertaining to this development in the near future.

The following weak scaling plots compare load balancing algorithms for newly added RVEs. The plots show timing results from the multiscale soft tissue model when running a uniaxial tension case on a standard tensile test geometry. The recorded times are from the first nonlinear iteration of two load steps, corresponding to elongation of the geometry by approximately 0.3% and 0.6%. For the first load step, each element has a 0.25 probability of utilizing a fiber-only RVE. These RVEs are evenly distributed across the microscale processes using the same algorithm, which results in the equivalent timing and scaling results between the algorithms as seen in Figure 1. During the second load step, elements again have 0.25 probability of having an RVE, so some RVEs will be maintained and are unable to move from their current microscale process. The newly added RVEs are arranged in one of three ways:

1. No load balancing (NLB): The macroscale ordering of elements is used to place RVEs in the communication pattern with no regard for work balance.
2. Load balancing 1 (LB1): Spreads the newly added RVEs evenly across the microscale processes without regard for their current workload.
3. Load balancing 2 (LB2): Assigns new RVEs to the lightest loaded microscale processes first.

As seen in Figure 2, both load balancing algorithms greatly reduce runtimes, as expected. The second load balancing is faster at these process counts, but the added complexity of the algorithm may begin to weigh in as the model is further scaled up.

Timings of all portions of the first nonlinear iteration for both the first and second load step with LB2 are shown in Figure 3 (A) and (B), respectively. The largest differences are the RVE times, and as expected, the macroscale matrix solve is unaffected. The assemble time (which is the total assemble time minus the RVEs and multiscale communication) and the multiscale communication are slightly changed but the same behavior is seen. The total solve time increases in accordance with the increase in RVE time. Timing results up to eight macroscale processes are averages of two runs. Meshes contain approximately 2000 elements per macroscale process. Future results will include migration of RVEs and should therefore show better scaling of the RVE times at non initial load steps.

Figure : Weak scaling comparison of RVE time and total nonlinear iteration time for the three load balancing algorithms for first load step.![A description...](data:None;base64,)

Figure : Weak scaling comparison of RVE time and total nonlinear iteration time for the three load balancing algorithms for second load step.![A description...](data:None;base64,)

Figure : Weak scaling plots of total solve, matrix solve, assemble, multiscale communication, and RVE times. (A) First load step. (B) Second load step.