THE ADAPTIVE MULTI-SCALE SIMULATION INFRASTRUCTURE - DESIGN AND IMPLEMENTATION OF A SOFT TISSUE SIMULATION

W.R. TOBIN, D. FOVARGUE, V.W.L. CHAN, M.S. SHEPHARD

Abstract.

1. Note. This article predominantly discusses multi-scale simulations - wherein two or more separate physical scales are involved in the simulation - the discussion is equally applicable to multi-model simulations - wherein two or more numerical or physical models operating on the same domain (possibly with differing discretizations) are involved in the simulation - as well as to simulations which are both.

2. Introduction. Most physical models and implementations using numerical methods operate at a single scale characteristic to the problem. The scale at which the preponderance of attributes governing the underlying physics being modelled are measured and the resolution of the numerical solution is considered acceptable for the desired accuracy of the simulation.

Introduction of influence from scales orders of magnitude removed from the primary scale can take place in several different ways. This influence can be incorporated during the mathematical development of the physical model itself. In this case the multi-scale features are incorporated into the numerical implementation automatically, as a result of their presence in the mathematical model.

Multi-scale influences can also be incorporated in a more strictly numerical sense, combining physical models operating at different scales referencing and effecting shared physical fields and values [12] [17]. Numerically multi-scale simulations associate many numerical implementations of single-scale physical models and cause them to interact through passing key physical terms between scales. This scale-coupling will involve up-/down-scaling operations to account for differences in dimension, as well as possibly dimensionalizing terms computed in a non-dimensional domain.

In order for a numerically multi-scale approach to be effective the domain at the scale of interest (or engineering scale) must be sufficiently large that conducting the entire simulation with terms and granularity orders of magnitude smaller would make the simulation computationally intractable. Thus some computationally reasonable subset of the engineering scale is assigned tertiary scales to simulate. The precise choice of location of the tertiary simulations is largely dependent on the specific numerical algorithms used to implement the engineering scale simulation.

Often the subset(s) of the domain associated with the tertiary scale(s) are locations in the engineering domain where the impact of the tertiary scales has the most impact on the engineering scale simulation, in order to leverage the most power from the multi-scale approach.

In the concurrent multi-scale paradigm, all scales are computed simultaneously and the required inter-scale coupling values are provided just-in-time [18].

In the sequential multi-scale paradigm, tertiary scales compute parameters which are converted to quantities of utility to a primary scale. Each scale (though not necessarily each scale-task, representing an individual operation or simulation occurring in a scale) must be processed sequentially [9] and may be blocked on the completion of the previous scale. This multi-scale model maps intuitively to phased parallel communication models found in the commonly used bulk synchronous parallel abstraction

[16].

2.1. Adaptive Multi-scale Simulation Infrastructure. The Adaptive Multi-scale Simulation Infrastructure (amsi) is a set of libraries and tools developed at the Scientific Computation Research Center (SCOREC) at Rensselaer Polytechnic Institute. Amsi is designed to support the implementation and execution of dynamic numerically multi-scale simulations on massively-parallel HPC machines. Currently amsi provides interfaces for describing and managing the execution of individual scales in a multi-scale simulation, for defining and managing the scale-coupling communication required by such simulations, and for planning and enacting scale-sensitive load balancing operations on individual scales in a multi-scale system.

The interfaces provided by the amsi libraries are intended to facilitate the coupling of functional (possibly legacy) simulation codes already in usage for single-scale problems, in order to avoid development overhead associated with casting a simulation into a set of framework-specific components. The currently implemented interfaces are physics-agnostic, operating only on computational quantities with no model of a term's relation to the mathematical derivation of the multi-scale interaction. Incorporation of meta-information and services targeted toward specific multi-scale physical couplings represents a desirable extension of the amsi libraries, constituting a rich area of future work.

Amsi operates by maintaining a set of minimal simulation metadata in order to model various quantities of interest during simulation execution. A decentralized approach toward control is taken; dynamic control decisions are made and implemented at runtime during collective operations. In order to avoid the introduction of unnecessary parallel barriers into the code, amsi control decisions are only made during operations which are already collective over the set of processes effected by a given control decision.

2.2. Novelty. Frameworks and runtime systems have been developed which are intended for the development of specific multi-scale problems [11] [] [] and these frameworks have in some cases been expanded to facilitate the implementation of additional multi-scale problems [5].

These systems suffer from the requirement that an application developer must cast the problem of interest into a set of predefined components (interfaces, tasks, etc.) in order to take full advantage of such frameworks. Leveraging a component-based design is valuable from a software engineering perspective, but also introduces new complexities involved in defining the set of components and how they interoperate in the framework shifting some complexity up a level of abstraction. The core features of amsi used for multi-scale simulation management and communication are provided with no requirements of adherence to a specific component definition.

simulation flow control is determined solely by the runtime system. A framework may also limit specific features of the numerical implementation to those provided by the framework. Domain discretization models and associated adaptive processes – structured and unstructured meshing and associated refinement algorithms, particle-in-cell methods, etc. – convergence algorithms for both the minimization of the relevant residuals (Newton-Raphson, etc.) as well as iterative solution of the resulting linear system (KSP, GMRES, ILU, etc.). Amsi supplies several numerical components consisting of an adaptive finite element system, linear algebraic solver, and nonlinear convergence operators available as both simple C++ class interface definitions, as well as implementations built on top of other software supplied by SCOREC [**?**] and the PETSc linear algebraic solver [3] [2] [4]. Usage of the supplied numerical compo-

nents is not required to make use of the multi-scale meta-modeling, communication, dynamic instantiation, and scale-sensitive load balancing facilities supplied by amsi.

While implementation of desired numerical operations into a framework is of course possible, it may prove an additional undesirable overhead for an application programmer interested only in utilizing the multi-scale framework to implement simulation, rather than implementing the various numerical operations often perfected in widely available packages and libraries [] [] [].

A configuration file is supplied during command invocation containing the initial configuration of the process sets and their assignment to individual scales, as well as declaring the coupling between scales in the simulation. The deceleration of a scale and the initialization of the process set associated with the scale are considered a single atomic operation in the current implementation.

Process sets are mathematical sets of MPI ranks implemented so as to take advantage of any mathematical conveniences to minimize explicit storage whenever possible. Process sets are required to be non-overlapping in order for amsi control mechanisms to operate correctly. Ongoing developments are targeted at removing this requirement to allow a richer set of dynamic control options and increase overall utilization in the case of coupled scales blocking on one another.

Deceleration of a coupling relation between two scales is required if any communication between the scales is to occur during the simulation (through the amsi libraries), as the coupling deceleration during simulation initialization is used to instantiate data structures specific to scale-coupling operations. This limits memory overhead by not allocating space for unneeded coupling operations.

Scales are declared to be related if some quantities of interest (such as physical tensor field values) are transmitted between them for scale-coupling. This coupling relationship is assumed to be asymmetric, thus if scale A is related to scale B, scale B is not implicitly related to scale A, though this may be separately declared. For brevity this may be written as A B for A is coupled to B.

While the primary scale (the scale associated with producing the solution for the overall simulation) is often thought of as controlling the tertiary scales in terms of dynamic instantiation, this is not assumed in the amsi implementation. Any scale is capable of informing the process set associated with a coupled scale of the addition or deletion of individual scale-tasks. At present it is dependent on the developer to implement the capability of a scale to accept additional scale-tasks, though algorithms and hooks for developer-designed distribution algorithms are supplied to ease the implementation.

Scale coupling communication is guided by data distributions and communication patterns. A data distribution is a representation of individual units of scale-coupling data distributed across a single scale-task. The format of the coupling data is arbitrary (so long as it can be serialized and provided for use by MPI communication primitives). A data distribution represents the smallest unit of data important to the scale-coupling communication, which is necessarily distinct for various multi-scale use cases.

All communication and planning operations in the amsi libraries use an implicit local numbering for data distribution terms, essentially denoted by the term's index in a serialized communication buffer.

If a local installation of the Zoltan [8] [6] load balancing library is available, the load balancing planning algorithms of Zoltan can be used underlying the scale-sensitive load balancing planning of amsi. A minimal set of load-balancing algorithms are provided with amsi. Since there is no general solution to the dynamic load bal-

FIG. 2.1. *biotissue multi-scale problem*

ancing problem any set of provided load-balancing algorithms will prove insufficient for some use case. Thus amsi provides users with the ability to create and register their own load-balancing algorithms specific to the requirements of a given numerical implementation.

Due to the necessity of retaining inter-scale linkages through the load-balancing processes, the actual data migration undertaken to implement a specific load balancing plan is typically conducted internally by the amsi system which is provided with buffers of data being load-balanced by each process on the appropriate scale. However, in order to maintain usability in more general situations, it is also possible to simply inform the amsi system that the current load-balancing plan has already been accomplished by user-implemented data migration. This functionality combined with the ability to use user-designed load-balance planning algorithms allows amsi to work with entirely external load-balancing libraries and algorithms, allowing scale-sensitive load-balancing to be used even when a specific multi-scale use case falls outside the bounds of capabilities built into the system.

2.3. Soft Tissue Simulation. The Biotissue code is a multi-scale simulation of soft organic tissues, consisting of an engineering-scale simulation using the finite element method, and a micro-scale quasistatics force-balance simulation. The Cauchy Momentum Balance Equation for a body in static equilibrium is used as the governing equation for the global simulation. Specific quantities needed to compute the elemental tangent stiffness matrices are instead supplied by the micro-scale force-balance simulations, which occur at each numerical integration point in the engineering-scale mesh, see figure 2.1.

The incremental displacements (the displacement deltas generated by the previous Newton iteration) on each engineering-scale finite element node are used to distort the a dimensionless representative volume element (RVE) resulting in each micro-scale node being displaced. Each RVE is a dimensionless unit cube containing a connected collagen fiber network, for the present results the fibers are implemented as truss elements and their distribution is generated using Delauney triangulation. The micro-scale boundary condition necessitates that the force exerted by each fiber network node located on the boundary of the RVE be zero. The micro-scale global Jacobian is formulated along with the force vector, wherein the boundary condition is enforced by setting specific rows of the force vector associated with boundary nodes

to zero.

Another Newton-Raphson iterative process occurs at micro-scale to converge to the set of displacements resulting in force equilibrium for all the fibers within the fiber network constituting the RVE. After a micro-scale simulation converges to a solution, the Jacobian and force vector are then evaluated again at the solution position, the force exerted along the boundary of the RVE is summed for each of the principal and shear components of a stress tensor, which is then dimensionalized and sent back to the engineering-scale for use in the elemental tangent stiffness matrix formulation. More detail into the derivation of this multi-scale system can be found in [13] [1] [14] . Discussion of the dimensionalization of the dimensionless force terms produced by each fiber network can be found in [15] [7].

Thus the simulation as a whole is essentially doubly nonlinear, as during each engineering-scale Newton iteration every micro-scale RVE must undergo a full Newton-Raphson convergence process.

2.4. Biotissue Scale-Sensitive Load Balancing. Scale-sensitive load balancing is conducted on the micro-scale RVE distribution at locations in the code where communication bottlenecks are already present. This prevents the introduction of new bottlenecks which could result in reduced performance. Macro-scale Newton-iterations are the most frequently occuring multi-scale communication bottleneck at which we chose to conduct the load balancing operation, and macro-scale incremental load steps the least-frequent. These represent the two extreme cases considered in our analysis on the efficacy of the scale-sensitive load-balancing scheme adopted for the biotissue simulation.

Each micro-scale RVE simulation involves the assembly and solve of a linear system of equations on the order of 10,000 unknowns for each micro-scale Newton iteration. While the total solve time is dependent on the number of nonlinear iterations and the hardware executing the code, this system is sufficiently small that parallelization of the RVE code itself is unwarranted, as the introduction of communication overhead into the RVE solution process would result in decreased performance, despite the modest gain to computation time. However, as each RVE represents an independent task to be accomplished by the simulation, they may be freely distributed across the processes assigned to the micro-scale computation. The only restriction to their (re)distribution for load balancing purposes is that the relation to a specific macro-scale numerical integration point must be maintained.

As each RVE is a nonlinear simulation their is no a-prior estimate of the computational load each RVE represents, so the optimal initial distribution is simply treating the weight of each RVE as 1. Thus each micro-scale process is assigned either $\left\lfloor \frac{\text{num\_rves}}{\text{num\_micro\_procs}} \right\rfloor$ or $\left\lceil \frac{\text{num\_rves}}{\text{num\_micro\_procs}} \right\rceil$. Once the first load-balancing event is reached, regardless of the granularity at which load-balancing is occuring for a given case, each RVE has recorded the total number of newton iterations it has conducted since the previous load-balancing event. This number is used as a heuristic weighting metric to determine the computational load of each individual RVE. The iteration count metric was chosen as each iteration takes a similar level of work to complete, up to the difference in the number of RVEs for each unique fiber network. Further, assuming linear incremental loading, regions in the engineering-scale mesh located near stress concentrators will result in 'more interesting' boundary conditions for the micro-scale problems related to numerical integration points in that region, which may result in the RVE requiring more Newton iterations to converge to an adequate solution, since the boundary conditions may lie farther from the initial state of the

RVE in the regime of convergence for the nonlinear problem.

3. Results.

4. Future Work. Work on scale-balancing is ongoing for the Biotissue test problem. Using this work as a basis, a set of algorithms to provide support for more general scale-balancing operations will be developed and packaged into amsi.

Embedded fiber-matrix RVEs separately parallelized and incorporated into multiscale model, used at most error-sensitive locations [10] [19] [20].

REFERENCES

[1] Balaji Agoram and Victor H Barocas, *Coupled macroscopic and microscopic scale modeling of fibrillar tissues and tissue equivalents*, Journal of biomechanical engineering, 123 (2001), pp. 362–369.

[2] Satish Balay, Shrirang Abhyankar, Mark F. Adams, Jed Brown, Peter Brune, Kris Buschelman, Lisandro Dalcin, Victor Eijkhout, William D. Gropp, Dinesh Kaushik, Matthew G. Knepley, Lois Curfman McInnes, Karl Rupp, Barry F. Smith, Stefano Zampini, and Hong Zhang, *PETSc users manual*, Tech. Report ANL-95/11 - Revision 3.6, Argonne National Laboratory, 2015.

[3] ———, *PETSc Web page*. `http://www.mcs.anl.gov/petsc`, 2015.

[4] Satish Balay, William D. Gropp, Lois Curfman McInnes, and Barry F. Smith, *Efficient management of parallelism in object oriented numerical software libraries*, in Modern Software Tools in Scientific Computing, E. Arge, A. M. Bruaset, and H. P. Langtangen, eds., Birkhäuser Press, 1997, pp. 163–202.

[5] Martin Berzins, Justin Luitjens, Qingyu Meng, Todd Harman, Charles A Wight, and Joseph R Peterson, *Uintah: a scalable framework for hazard analysis*, in Proceedings of the 2010 TeraGrid Conference, ACM, 2010, p. 3.

[6] E. G. Boman, U. V. Catalyurek, C. Chevalier, and K. D. Devine, *The Zoltan and Isorropia parallel toolkits for combinatorial scientific computing: Partitioning, ordering, and coloring*, Scientific Programming, 20 (2012), pp. 129–150.

[7] Preethi L Chandran and Victor H Barocas, *Deterministic material-based averaging theory model of collagen gel micromechanics*, Journal of biomechanical engineering, 129 (2007), pp. 137–147.

[8] Karen Devine, Erik Boman, Robert Heaphy, Bruce Hendrickson, and Courtenay Vaughan, *Zoltan data management services for parallel dynamic applications*, Computing in Science and Engineering, 4 (2002), pp. 90–97.

[9] Carlos J. Garcia-Cervera, Weiqing Ren, Jianfeng Lu, and E. Weinan, *Sequential multiscale modeling using sparse representation*, Communications in Computational Physics, 4 (2008), pp. 1025–1033.

[10] Spencer P Lake, Mohammad F Hadi, Victor K Lai, and Victor H Barocas, *Mechanics of a fiber network within a non-fibrillar matrix: model and comparison with collagen-agarose co-gels*, Annals of biomedical engineering, 40 (2012), pp. 2111–2121.

[11] Steven G Parker, James Guilkey, and Todd Harman, *A component-based parallel infrastructure for the simulation of fluid–structure interaction*, Engineering with Computers, 22 (2006), pp. 277–292.

[12] V. B. Shenoy, R. Miller, E. B. Tadmor, D. Rodney, R. Phillips, and M. Ortiz, *An Adaptive Finite Element Approach to Atomic-Scale Mechanics - The Quasicontinuum Method*, eprint arXiv:cond-mat/9710027, (1997).

[13] T. Stylianopoulos, *Multiscale mechanical modeling of soft biological tissues*, PhD thesis, University of Minnesota, 2008.

[14] Triantafyllos Stylianopoulos and Victor H Barocas, *Multiscale, structure-based modeling for the elastic mechanical behavior of arterial walls*, Journal of biomechanical engineering, 129 (2007), pp. 611–618.

[15] ———, *Volume-averaging theory for the study of the mechanics of collagen networks*, Computer methods in applied mechanics and engineering, 196 (2007), pp. 2981–2990.

[16] Leslie G Valiant, *A bridging model for parallel computation*, Communications of the ACM, 33 (1990), pp. 103–111.

[17] E Weinan, Bjorn Engquist, et al., *The heterognous multiscale methods*, Communications in Mathematical Sciences, 1 (2003), pp. 87–132.

[18] Xiaowei Zeng and Shaofan Li, *Recent developments on concurrent multiscale simulations*, in Advances in engineering mechanics, Qing-Hua Qin and Bohua Sun, eds., Nova Science, New York, 2010.

[19] L Zhang, SP Lake, VH Barocas, MS Shephard, and RC Picu, *Cross-linked fiber network embedded in an elastic matrix*, Soft matter, 9 (2013), pp. 6398–6405.

[20] Lijuan Zhang, Spencer P Lake, Victor K Lai, Catalin R Picu, Victor H Barocas, and Mark S Shephard, *A coupled fiber-matrix model demonstrates highly inhomogeneous microstructural interactions in soft tissues under tensile load*, Journal of biomechanical engineering, 135 (2013), p. 011008.