

# SCALE-SENSITIVE LOAD BALANCING FOR NUMERICALLY MULTI-SCALE SIMULATIONS

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

## Abstract.

**1. Note.** While 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 different discretizations) are involved in the simulation - as well as to multi-scale-multi-model simulations.

**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 modeled are measured], and the scale at which the resolution of the solution is considered commensurate with the desired accuracy of the simulation.

Introduction of influence from scales orders of magnitude different 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 but operating on some of the same physical fields and values [7] [11]. Numerically multi-scale simulations associate many numerical implementations of single-scale physical models and cause them to interact in a meaningful way by passing key physical terms from some scales and supplying them to others, typically this will involve up-/down-scaling equations to take into account the difference in scale, referred to as compression and reconstruction operators in [11] for upscaling and downscaling, respectively. Clearly this is only useful in cases where terms from the various incorporated scales contribute meaningfully to the solution at the primary scale of interest. The domain at the scale of interest (for brevity the engineering scale) is sufficiently large that conducting the entire simulation with a granularity orders of magnitude smaller would make the simulation computationally intractable. Thus specific locations in the engineering scale domain - typically locations which are used during the simulation to determine the response of the system as a whole - have ..

Among numerically multi-scale simulations, there are two primary paradigms: concurrent and sequential multi-scale. In the sequential multi-scale paradigm, values required to continue the simulation at the current scale which are provided from associated scales are computed prior to the point at which they are

needed. In the concurrent multi-scale paradigm, both scales are computed simultaneously and the required inter-scale coupling values are provided just-in-time.

**2.1. Load Balancing.** As problem sizes scale - whether in terms of the underlying physical model used to simulate the problem or the granularity of the domain discretization or problem domain itself - single-scale simulations must be conducted on massively parallel machines in order to achieve meaningfully responsive results. This is accomplished by distributing problem data across the execution space of the machine, and keeping communication/blocking points to a minimum.

In order to maximize the computational gain granted by parallelization, load balancing must be used. The general load balancing problem depends on the implementation of the model used in a simulation; in terms of the data/tasks to be balanced as well as the necessary coordination between processes in order to advance the simulation. Further, the characteristics of the machine must be taken into account, both with respect to the hetero/homogeneity of the computational resources of the machine and the communications facilities employed. Finally combining the other dependencies, the load being balanced must have some metric denoting the computational demand associated with the discrete units capable of being distributed across the machine, as well as (possibly) the communication cost associated with that distribution. For a more in-depth discussion of load balancing theory, see [1, 2].

Dynamic load balancing is a difficult problem, though many widely-used numerical methods and/or specific implementations of methods have developed relatively mature load balancing algorithms or heuristics [5]. In particular the finite element method - used in the primary scale of the Biotissue problem - has a mature set of load-balancing methods based on graph methods [3, 4] used to distribute entities of the discretized domain and associated tensor field values across the execution space. The computational demand on each node can be estimated as a function of the number of individual elements that must be processed by each node, but due to dynamic mesh refinement processes used to minimize error estimators across the numerical problem domain, the global number of elements associated with the discretization may vary wildly over the course of a multi-load step simulation [6].

Using numerical methods for which load balancing schemes have been developed in a multi-scale simulation introduces a new set of complex constraints to the problem, due primarily to inter-scale dependencies. While it may be the case that such dependencies can be ignored, as in the case where all associated scales - those scales in the simulation which share a one- or two-way communication relationship with a given scale - must be fully reconstructed following a load-balancing operation. This reduces to the single-scale load balancing problem with an additional phase to reinitialize associated scales. When it is not possible to destroy or reinitialize associated scales, inter-scale dependencies

must be maintained in order to continue computation. This operation is referred to as scale-sensitive load balancing, as only a single scale in a simulation is being considered while maintaining the validity of the global simulation.

Beyond this introduction of new constraints to load balancing operations at each scale in a multi-scale system, the multi-scale system as a whole may greatly benefit from the use of scale-balancing. Scale-balancing is the balancing of scale-tasks against each other to minimize rendezvous time and maximize time spent in useful computation. Scale-balancing is a difficult prospect as it involves managing many dynamically-changing scales which themselves are being load-balanced using scale-sensitive load-balancing techniques, and represents a significant and promising area of future work.

**2.2. 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.

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.

During simulation initialization simulation scales are declared and then defined by associating them with process sets, and declaring their scale-linking relations. Process sets are simply mathematical sets of process ranks implemented so as to take advantage of any mathematical conveniences to minimize explicit storage whenever possible. At present process sets must be non-overlapping to take advantage of the simulation control portions of the AMSI libraries, but for the simulation metadata modeling, scale-coupling communication, and multi-scale load balancing operations, the process sets associated with related scales may overlap. Scales are declared to be related if some quantities of interest (such as tensor field values) are transmitted between them for scale-coupling.

If a local installation of the Zoltan [4] [2] 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

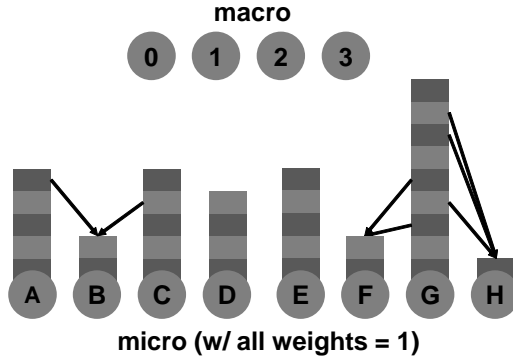


FIG. 3.1. *scale-sensitive load-balancing: planning*

solution to the dynamic load balancing 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.

### 3. Methods.

**3.1. Scale-Sensitive Load Balancing.** Scale-sensitive load balancing is a three-phase process: a planning phase, migration phase, and scale-syncing phase are required to complete a single load-balancing event.

The planning phase is a collective operation on the set of processes associated with the scale being load-balanced. During this phase weighting constraints and other information is provided to a planning algorithm which produces a plan for redistributing information such that the weighting metric is balanced. For this phase AMSI typically utilizes planning algorithms provided by the Zoltan library, though as noted above any planning algorithm may be registered and used with the system. A simple example is provided in figure 3.1, wherein each unit of load-balanced data has equal weight, a more concrete case will be discussed in section 3.3.

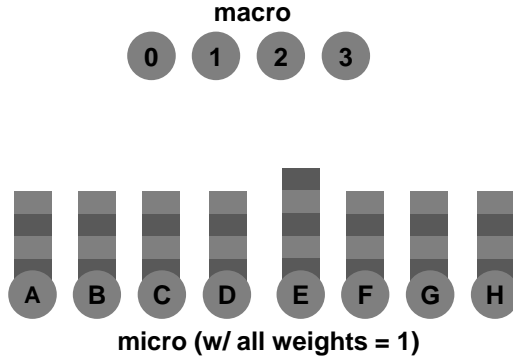


FIG. 3.2. *scale-sensitive load-balancing: migration*

The migration phase is also collective on the set of processes associated with the scale being load-balanced. During this phase the plan from the previous step is put into action. AMSI provides support for automatic migration, if a user is able to provide a memory buffer of serialized load-balancing data in the same order as the weights were provided to the planning stage. The simple example continues in figure 3.2. A user may also implement the migration function themselves and instead call the AMSI migration interface with an empty buffer, which indicates the user has already completed the migration. This is necessary as there are minimal pieces of metadata (minimally two integers for each piece of migration data, which are typically appended during the AMSI automatic migration and stripped off at the receiving end) which must be distributed by the AMSI system for the third phase to complete.

The scale-syncing phase is collective over the set of process associated with the scale being load-balanced, and at least one associate scale. In this phase the appended metadata redistributed with the migration data is used to provide information to associate scales about the movement of the data at the scale being load-balanced. The simple example concludes in figure 3.3. Currently associate scales must interpret this data themselves - which consists of the scale-ranks of the origin and destination of the piece of data and the implicit ID assigned to the data due to its ordering in the migration buffer - but the implementation of domain relation metadata in the AMSI system in the future should mitigate this requirement.

**3.2. Example Problem: Biotissue.** 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

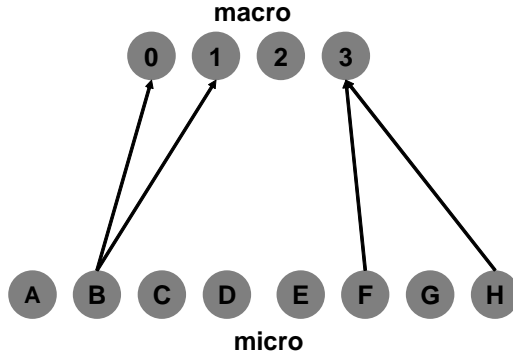


FIG. 3.3. *scale-sensitive load-balancing: scale-syncing*

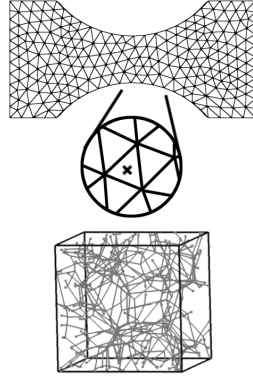


FIG. 3.4. *biotissue multi-scale problem*

each numerical integration point in the engineering-scale mesh, see figure 3.4.

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 [8] [1] [9] . Discussion of the dimensionalization of the dimensionless force terms produced by each fiber network can be found in [10] [3].

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.

**3.3. 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 occurring multi-scale communication bottleneck at which we chose to conduct the load balancing operation, and macro-scale load incrementing the least-frequent. These represent the two extreme cases considered in our analysis on the efficacy of our load-balancing scheme.

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 there 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 occurring 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



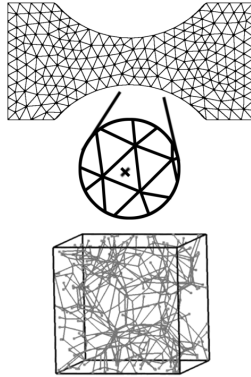


FIG. 4.1. *biotissue problem domain*

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.

**4. Results.** The domain chosen for the problem can be seen in 4.1.

Using previous iterations as weight metric in inter-load incrementing load balancing case. Incremental loading versus more dynamic loading cases.

**5. 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 multi-scale model, used at most error-sensitive locations [6] [12] [13].

## 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] 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.
- [3] 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.
- [4] 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.
- [5] KAREN D. DEVINE, ERIK G. BOMAN, ROBERT T. HEAPHY, BRUCE A. HENDRICKSON, JAMES D. TERESCO, JAMAL FAIK, JOSEPH E. FLAHERTY, AND LUIS G. GERVASIO,



- New challenges in dynamic load balancing*, Applied Numerical Mathematics, 52 (2005), pp. 133 – 152. {ADAPT} '03: Conference on Adaptive Methods for Partial Differential Equations and Large-Scale Computation.
- [6] 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.
  - [7] 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 Quasicon- tinuum Method*, eprint arXiv:cond-mat/9710027, (1997).
  - [8] T. STYLIANOPOULOS, *Multiscale mechanical modeling of soft biological tissues*, PhD thesis, University of Minnesota, 2008.
  - [9] 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.
  - [10] ———, *Volume-averaging theory for the study of the mechanics of collagen networks*, Computer methods in applied mechanics and engineering, 196 (2007), pp. 2981– 2990.
  - [11] E WEINAN, BJORN ENGQUIST, ET AL., *The heterognous multiscale methods*, Commu- nications in Mathematical Sciences, 1 (2003), pp. 87–132.
  - [12] 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.
  - [13] 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.