Proceedings of the 2016 Winter Simulation Conference
T. M. K. Roeder, P. I. Frazier, R. Szechtman, E. Zhou, T. Huschka, and S. E. Chick, eds.

SIMULATING 3-D BONE TISSUE GROWTH USING REPAST HPC: INITIAL SIMULATION DESIGN AND PERFORMANCE RESULTS

John T. Murphy

Argonne National Laboratory
Global Security Sciences

9700 South Cass Avenue
Lemont, IL 60439, USA

Elif Seyma Bayrak
Mustafa Cagdas Ozturk
Ali Cinar

Department of Chemical and
Biological Engineering
Illinois Institute of Technology
10 W 33rd St,
Chicago, IL 60616, USA

ABSTRACT

Bone is one of the most implanted tissues worldwide. Bone tissue engineering deals with the replacement and regeneration of bone tissue; outcomes are determined by complex biological interactions, making it difficult to design an optimal tissue growth environment. Agent-Based Modeling (ABM) is a powerful tool to simulate such a system. We present a simulation of engineered bone tissue growth using the Repast HPC toolkit, an ABM tool for high-performance computing environments. We use this example to provide preliminary performance tests on new features (not yet publicly released) of Repast HPC that accommodate operations common to biological modeling: 3-Dimensional parallelized spatial simulation and diffusion in 3 dimensions. Repast HPC is a general ABM toolkit, and the performance documented here should be representative of performance on other simulations. Using the baseline Repast HPC tools provides flexibility for continued model development and improvement.

1 INTRODUCTION

Bone is one of the most implanted tissues worldwide. Bone tissue engineering deals with the replacement and regeneration of bone tissue; outcomes are determined by complex biological interactions, making it difficult to design an optimal tissue growth environment. Agent-Based Modeling (ABM) is a powerful tool to simulate such a system. We present in this paper a simulation of engineered bone tissue growth using the Repast HPC toolkit (http://repast.sourceforge.net), a free and open-source ABM tool for high-performance computing environments. We focus on new features in the Repast HPC package that are designed to accommodate operations that may be common in biological agent-based modeling. Our intent is to provide a basic overview of these features, their application in a specific simulation, and their rough performance estimates. Importantly, we do not claim that this implementation is highly optimized. Repast HPC is designed to be a flexible and easy-to-use toolkit that emphasizes simplicity over performance. The existence of these features and their potential for application in other biologically motivated simulations is promising, and our hope is to facilitate the use of these tools in future work.

We begin our discussion by presenting the motivation for the simulation and an overview of the biological processes we will examine. We then turn to the Repast HPC toolkit and the newly implemented features that enable the biological simulation to be performed. We next present the simulation in more specific detail, and elucidate the specific model components and their roles in the simulation. We follow

this with a description of the specific sets of test runs performed, the results of those tests, and a discussion of our intended next steps both for the bone model simulation and for Repast HPC.

2 MOTIVATION: ENGINEERED BONE TISSUE

The loss or failure of an organ or tissue is highly frequent, devastating problem with a substantial cost in human health care (Langer 1999). Bone is one of the most implanted tissue worldwide since bone defects above a critical size do not heal by itself. Bone tissue engineering deals with the replacement and regeneration of bone tissue with practical approaches (Amini et al. 2012). Use of synthetic or natural biomaterial scaffolds that supports the migration, proliferation, and differentiation into bone cells is one approach in bone tissue engineering (Damien and Parsons 1991; Langer and Vacanti 1993). However, the success of these biomaterials in repairing large size bone defects is limited due to their lack of the osteogenic and properties (Petite et al. 2000). A promising alternative is using the cells with osteogenic potential in these biomaterials to promote bone formation. Mesenchymal stem cells (MSC) have been widely used in bone regeneration applications (Manassero et al. 2013).

One of the challenges in the field of bone tissue engineering is exploring the complex interactions behind biological events and utilizing this knowledge to design an optimal tissue growth environment. Animal studies may represent the system adequately however they often fall short due to long time frame and difficult data collection resulting with sacrificing the animal. Computational models when they are properly developed can offer the ability to perform experiments *in silico* and accurately predict the interplay between the cells, biomaterial and signaling molecules. The number of variables that contribute to the formation of engineered tissues and the multicellular response to these signals present a challenging optimization problem that must be addressed for the ultimate success of tissue engineering.

Agent-Based Modeling (ABM) is a powerful tool to simulate main actors of a system based on their interactions with their local environment. It has great advantages including simulating of each individual's behavior, holding their history, allowing them to adapt and learn complex behavior. It is a natural choice and has been widely used to model biological systems (Qutub et al. 2009; Bayrak et al. 2015b). One main concern of the modelers is the computational heaviness of ABMs that limits the use of this technique in biological systems where one cubic centimeter of tissue can encompass approximately one billion cells.

Discovering the full potential of agent-based simulation models depends on the computational power available. Recent advances in the capacity and availability of high-performance computing resources and platforms have expanded the ability of agent-based models of biological systems to accomplish simulation in more realistic scales. Cockrell et al. (2015) implemented a model of gut which extends an previous model of gut epithelium using a custom parallel computational approach to run anatomic scales simulations. Perrin et al. (2010) developed a large-scale agent-based model, with explicit implementation of lymph nodes to investigate changes in the dynamics of HIV disease progression. In another work, two multi-scale existing model od in-stent restenosis and cerebrovascular blood flow was combined using HPC approach to run larger scales up to peta scale, incurring a coupling overhead of 1–10% of the total execution time (Groen et al. 2013). This paper focuses on use of Repast HPC for the development of a high performance agent-based modeling (ABM) framework that describes integrated cell behavior (vascularization and tissue growth) in a system based on interactions with its environment (degradable scaffold), other cells, and its own intracellular decisions/states.

3 REPAST HPC: OVERVIEW AND ENHANCEMENTS

Repast HPC is a toolkit for Agent-Based Modeling that is specifically designed for high-performance computing applications (Collier and North 2012). Internally, model construction can follow a paradigm that mirrors that of the Repast Simphony toolkit (North et al. 2013) on which Repast HPC is based. In both tools, a central collection called a ‘context’ contains the agents in the model. Contexts also contain ‘projections’, which structure agent interactions and relationships; the two kinds of projections are spaces and networks.

In Repast HPC, contexts and projections work together to parallelize the simulation. The assumption is that all agent interaction will be with agents that are adjacent to one another in at least one projection. Typically a spatial projection is used to divide the simulation to achieve load balancing; as part of the parallelization strategy, agents that were on adjacent processes and that are positioned within a 'buffer zone' a small distance from the boundary between processes were copied as non-local or 'ghost' agents, thereby making their information available to the adjacent process. Changes to 'ghost' agents cannot propagate back to the 'real' copy of the agent, but by appearing as copies on the adjacent process, the information in non-local agents was available for local agents to consider and act upon. Network projections are not typically used for load balancing- no automated way to do so is currently in Repast HPC- but the principle of adjacency applies, and agents connected by a network link, no matter how distant they may be according to a spatial projection, are copied onto each other's processes and can interact in this way.

For use in this simulation, two features not previously available in Repast HPC were required: the ability to construct 3-Dimensional spaces in which agents could move and interact, and the ability to easily calculate diffusion of chemicals through the simulation space.

3.1 N-Dimensional Spaces

Previous releases of Repast HPC have been limited to 2-Dimensional spaces. While adequate for many agent-based models, a large class of additional models based on real-world dynamics would require 3-Dimensional spaces (e.g. fish in schools, birds in flocks, etc.). Perhaps less intuitively, some agent-based models might require higher dimensionality (N-Dimensional) spaces; one common application would be 'opinion spaces,' where an individual's location represents not a physical position but rather a point along multiple axes representing some measure of sentiment toward N independent topics.

The MPI specification (MPI Forum 1994) already incorporates the capability to define communicators that arrange the processes in an abstract N-Dimensional Cartesian topology. Moreover, the underlying 'Point' object in Repast HPC has, since the original release, had the ability to represent points in an N-Dimensional space. What was lacking, however, was an ability to use a higher-dimensional space as a basis of synchronization.

To achieve this, the new code implemented a more generalized algorithm for calculating buffer zones and passing agent information to adjacent processes. This more generalized algorithm is capable of handling any arbitrary number of dimensions. A number of restrictions remain: space must be evenly divided in each dimension across the n processes in that dimension, and the overall space must be a N-Dimensional rectangle. However, N may be any number. Note that although special-purpose algorithms could theoretically achieve a performance benefit for common values of N (2 and 3), no such implementation is provided; instead, the generalized algorithm is now used for all spaces.

3.2 Diffusion

As described below, the simulation of bone model angiogenesis requires that cell movement be driven by the gradient in the concentration of a chemical secreted by certain specific kinds of cells. To achieve this diffusion, a special 'diffusion' module was added to Repast HPC. This new model defines a storage array that consists fundamentally of an array of values; it is not, therefore, a collection of agents or pointers to agents. The diffusion implementation works with N-Dimensional spaces; this matches the agent spaces described in the previous section.

The implementation of N-Dimensional spaces exchanges values across processes from regions of the space that abut each process. It relies on customized MPI data types that are defined recursively across the N dimensions. The implementation should therefore inherit whatever optimization the MPI implementation offers on the system on which the simulation is run.

The diffusion calculations are not optimized for performance; instead, they are optimized for ease of use and extension to specific applications. A *DiffusionLayer* class stores the values and provides

synchronization across processes using the recursive MPI data types. An abstract *Diffusor* class is provided that can be extended, and the specific diffusion algorithm required can be implemented in the child class. When the *DiffusionLayer* class is asked to perform diffusion, it loops through the collection of cells in the N-Dimensional space and at each point passes a block of cells with radius r on all dimensions to the *Diffusor* class; these cells are indexed in a standard way that allows the *Diffusor* class to access any that are needed (and omit any that are not), and to perform whatever operation has been specified to calculate the value that should replace the current value at the central cell. At the boundaries of the space, the block of cells used may be truncated; the diffusor class must be crafted to deal with this, but the indexing scheme for the block of cells provided to it will make any truncations clear.

It should be clear that for any specific diffusion algorithm a more efficient strategy for performing the diffusion calculations could be created. The implementation here trades efficiency in model execution for ease of use and simplicity in implementation: in keeping with traditional Repast HPC practice, the user is not required to write any MPI code, and instead need only implement the diffusion algorithm via a *Diffusor* class, and Repast HPC handles all diffusion calculations and cross-process synchronization.

4 MODEL STRUCTURE

The Repast HPC model described here is being conducted in parallel with laboratory research described in (Somo et al. 2015). A Java version of the simulation model has already been created (Bayrak et al. 2015a), and work to validate the Java model against the empirical data is in progress.

4.1 Vascular Bone Formation

Bone is a highly vascularized tissue. Angiogenesis, the process of growing new blood vessels from preexisting ones, is the main mechanism vascularization of biomaterial scaffold . Angiogenesis is driven by chemical cues including vascular endothelial growth factor (VEGF). The main steps involved in the process of angiogenesis can be summarized as (Adams and Alitalo 2007): (1) Expression of a pro-angiogenic factors in the tissue in response to local stimuli, (2) angiogenic factors diffuse through tissues and reach the blood vessels, (3) cells lining the walls of capillaries (Endothelial Cells), become activated and migrate towards the higher angiogenic factor concentrations, (4) ECs behind the leading cell start proliferation in order to expedite extending the newly formed sprout. When these sprouts grow, they fuse together to stabilize and form loops that can carry blood, a process known as anastomosis (Paweletz and Knierim 1989).

MSCs are pluripotent adult stem cells that can differentiate into different cell types and widely used in bone tissue engineering (Arvidson et al. 2011). One of the rationales for the use of MSCs in bone tissue engineering is their ability to release factors such as VEGF that influence cell survival, proliferation and promote angiogenesis (Giuliani et al. 2013). Several data indicate that MSCs can provide a local microenvironment that supports bone regeneration and angiogenesis.

4.2 Cell Types and Simulation Environment

Endothelial cells (ECs) are the main actors in angiogenesis process, and are responsible for capillary growth. The angiogenesis simulation model is constructed based on the EC actions, given the biomaterial microenvironment. Behavior of ECs has been abstracted in a rule-base from the real angiogenesis process observed in experimental studies. ECs become activated with the presence of a certain amount of angiogenic growth factor (i.e. VEGF) concentration in their local environment. Activated ECs (tip cell) migrate through the gradient of VEGF. While the EC is migrating, it elongates until it reaches the maximum observable size of a single EC. Elongation results in proliferation of ECs (stalk cells), which eventually form the capillary network. VEGF is needed for newly formed capillary survival as well. Endothelial cells may apoptose due to growth factor deficiency and results in capillary regression (Tang et al. 2004). A VEGF threshold is implemented in the ABM to account for the capillary regression during

vascularization process. The regions occupied by the scaffold are not allowed to EC invasion, when an EC encounters part of a scaffold, it searches new routes to keep moving through the gradient. Parameters and detailed rule base governing the EC behavior are derived from the literature and reported elsewhere (Bayrak et al. 2015a). The flow chart for the main methods is illustrated in Figure 1.

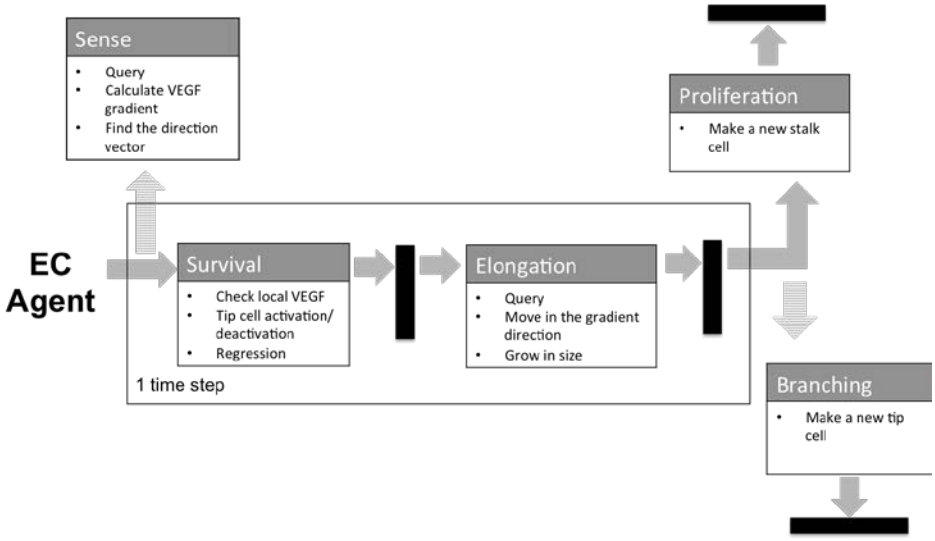

Figure 1: EC agent rule-base.

MSCs are the cells that form bone tissue after a differentiation process. A rule base is derived for tissue formation from stem cell differentiation based on a comprehensive literature survey (Bayrak et al. 2014). The MSC agent is capable of sensing the VEGF concentration in its neighborhood, migrating towards its gradient, going through cell cycle, growing in size, proliferating and releasing VEGF in the environment. MSC agent first searches the space and calculates the distance to the closest stable blood vessel(i.e anastomosed) since only anastomosed capillaries are considered as having blood flow and a source for oxygen and nutrition. If there is no blood vessel in the close vicinity (0-100μm), the cell agent will update its state to quiescent phase (G0). Hypoxic cell state (H) will be assigned to the MSC cell if there is no stable blood vessel in a further distance (100-200μm),and will start releasing VEGF. The time that cells can spend under hypoxia is limited (Deschepper et al. 2011). If no blood vessel reaches the close vicinity of a hypoxic cell, it will go through cell apoptosis. The oxygen and nutrition distribution is not explicitly modeled in space, however oxygen diffusion from the blood vessel is assumed to be a few hundred microns. If the cell is in the vicinity of a stable blood vessel, then it will go through the cell cycle and spend the required time at each state and complete the actions before it proceeds to the next state in the cycle.

The three-dimensional scaffold model with spherical pores were generated computationally and described elsewhere (Somo et al. 2015). For weak scaling studies larger size of scaffolds were created by reflecting the scaffolds at the original boundary in one dimension to keep the pore openings and porosity (pore volume divided by the total volume) consistent.

The **VEGF concentration values** were implemented using the new *DiffusionLayer* functionality described above; the diffusion algorithm used was a sum of the current grid cell's concentration minus the amount it would diffuse outward plus the amount it would receive from the immediately adjacent grid cells (radius 1, maximum of 26 adjacent cells, with fewer at the edges of the space). Simulation visualization for tissue growth is illustrated in Figure 2.

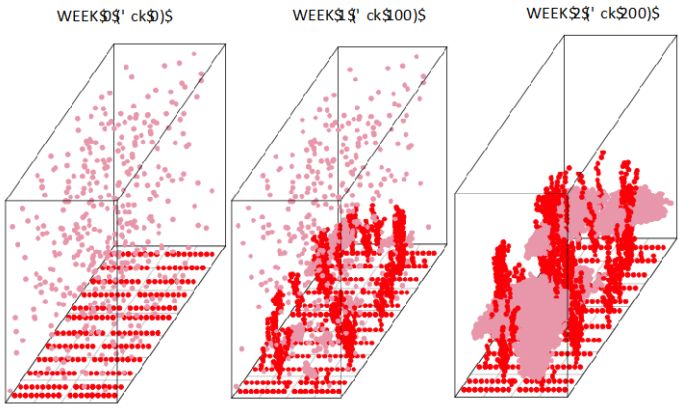

Figure 2: Simulation visualization for 2-week vascularization (red) and tissue growth (pink).

4.3 Repast HPC Projections

The Bone Tissue simulation model includes three projections: a fine-resolution discrete space, a coarse-resolution discrete space, and a network projection. The simulation is parallelized by dividing the physical space represented across multiple processes. Each process runs independently until there is a need to ensure that the information across processes is brought up to date and synchronized. Synchronization of the simulation is required when agents that might interact across processes have changed their internal states, or when some agents may have moved from one processor to another or been removed from the simulation environment. In Figure 1, black rectangles show the algorithm's synchronization points.

4.3.1 3-D Spaces

The requirement that MSC agents search for nearby cells can potentially lead to a high computational cost. If the agents' locations were only maintained in a continuous space, it would be necessary to calculate the distance from the active agent to all potential partner agents, and, unless some additional storage strategy were used, to do this for each active agent at each search. This simulation uses a discrete space, and in a discrete space the individual cells that are potentially within the search radius can be extracted. The discrete space provides a list of agents within these specific cells, and all agents outside this region are ignored. Despite this, for this simulation the search radius is large enough compared to the smallest spatial unit that the number of potential matching cells is very high: typically a region of 40x40x40 = 64,000 cells is needed, leading to a high computational demand.

To avoid this, Repast HPC uses multiple spaces: the positions of agents are maintained in both a fine-scaled space and much coarser space. Instead of 40x40x40, this coarser grid covers the same volume, but using only 3x3x3=27 cells. The resulting indexing permits much faster search. The only additional burden on the simulation is the need to ensure that all agent movement is done in both spaces, not just one.

4.3.2 Networks

The simulation also tracks blood vessels using a network projection; in this network projection, blood vessels are represented as linked cells forming a chain. This chain can cross process boundaries using Repast HPC's built-in functions, so no additional burden on the user is required.

5 PERFORMANCE TESTING

5.1 Test Design: Strong and Weak Scaling

Tests were performed in two ways. Strong scaling was tested by establishing a baseline simulation with 1,024,000 MSC agents in a space that measured 320 x 800 x 200, and then increasing the number of processes in the Z, X, and Y dimensions and further subdividing the space, as given in Table 1:

Table 1: Strong scaling design of runs.

Number of Processes	Procs: Z	Procs: X	Procs:Y	MSC Agent Count	Total MSC Agents
32	4	4	2	32000	1024000
64	8	4	2	16000	1024000
128	8	8	2	8000	1024000
256	8	8	4	4000	1024000
512	8	8	8	2000	1024000
1024	8	16	8	1000	1024000

For weak scaling tests, the size of the space was extended by expanding the Z dimension, and the number of agents were increased from a baseline, as given in Table 2:

Table 2: Weak scaling design of runs.

Number of Processes	Procs: Z	Procs: X	Procs:Y	MSC Agent Count	Total MSC Agents	zdim
16	4	2	2	12800	204800	100
32	8	2	2	12800	409600	200
64	16	2	2	12800	819200	400
128	32	2	2	12800	1638400	800
256	64	2	2	12800	3276800	1600
512	128	2	2	12800	6553600	3200
1024	256	2	2	12800	13107200	6400

5.2 Operations Tested

The performance tests given here report the performance for both the entire cycle of each simulation time step and for specific subcomponents of that time step. The results may be divided broadly into 'processing' and 'communication'; the former refers to the action of the bone model simulation, and the latter to the exchange of cross-process information handled by Repast HPC. Processing includes searching for stable vessels, MSC Agents movement and growth, and MSC Agents releasing GF; communication includes calls to three Repast HPC core functions:

- Synchronize balance: marking agents that need to be shared across processes
- Synchronize agent states: copying current agent state variables to non-local copies
- Synchronize agent status: updating all non-local agents to reflect movement of agents across processes or removal from the simulation

Additionally, total communication time, including Repast HPC's 'Synchronize projection information' operation, is assessed.

In addition to these, the diffusion operation described above was also assessed separately; this operation, 'updateVegFMap' includes both the calculation of the new values to represent diffusion and the communication of those values to adjacent processes via MPI.

Runs on the Vesta system (see below) are time-limited; consequently, we have considered all runs by limiting them to match the results obtained from the run with the fewest completed time ticks. For all results here we consider only the first 23 ticks of simulation time, for which we calculate averages for all operations listed. In practice, the time per tick for the simulation may change greatly as the simulation progresses due to the number and positions of the agents and the increased demand for processing and communication time. However, the initial period is comparable across all runs presented here.

5.3 Resources Used

All performance tests were done on the BG/Q Vesta system at the Argonne Leadership Class Facility, Argonne National Laboratory (USA) (https://www.alcf.anl.gov/cetus-and-vesta). Processors on the Vesta system are equipped with PowerPC A2 1600 MHz processors (1.6 GHz) and include 16 GB of memory. One core per processor was used for all text, and no multithreading was used.

6 RESULTS

6.1 Weak Scaling

Results from weak scaling tests are given in Figure 3 (left column). Overall times, shown at the top left, exhibit little increase as the size of the problem is increased. Of the simulated biological processes, the somewhat unusual curve for mscMoveAndGrow, and an anomalous value at 128 processes in mscRelease, are unexplained at this time. Of the Repast HPC communication processes, synchronization balancing is essentially unaffected by increasing the scale, but both synchronization of agent status and agent state climb. These, however, are small components of the overall communication costs, much of which occurs in the synchronize projection information task. The *updateVegfMap* call performing diffusion is essentially unchanged by increasing scale.

6.2 Strong Scaling

The right column of Figure 3 presents the results from the strong scaling runs as described in Table 1. Communication and Processing times are presented independently, and their total is also shown. The increase in processors by a factor of 32x improves performance by approximately 10x. As might be expected, the improvement is more marked in the processing time than in the communication time.

Performance for individual processes of *mscSearchForStableVessel*, *mscMoveAndGrow*, and *mscRelease* show improvement is apparent for all cases, though an anomalous uptick for *mscMoveAndGrow* at 256 processes is unexplained.

Performance for Repast HPC core functions are also given in Figure 3 (right column). SynchronizeProjectionInfo is the most expensive operation but scales well. Most of the performance costs from scaling are incurred in *synchronizeAgentStatus*, which trends upward as the scale increases, likely because this operation is the most highly affected by increasing the number of inter-process communications needed. Even though the size of each interprocess operation may be reduced, the number of these operations is increased, leading to a slightly higher performance cost as scaling is increased.

Performance for Repast HPC core functions are also given in Figure 3 (right column). SynchronizeProjectionInfo is the most expensive operation but scales well. Most of the performance costs from scaling are incurred in *synchronizeAgentStatus*, which trends upward as the scale increases, likely because this operation is the most highly affected by increasing the number of inter-process communications needed. Even though the size of each interprocess operation may be reduced, the number of these operations is increased, leading to a slightly higher performance cost as scaling is increased.

The performance of the diffusion routines (*updateVegfMap*) shows a speedup of almost exactly 32x, from a slowest speed of more than 10.85 seconds per operation to a fastest of under .35 seconds.

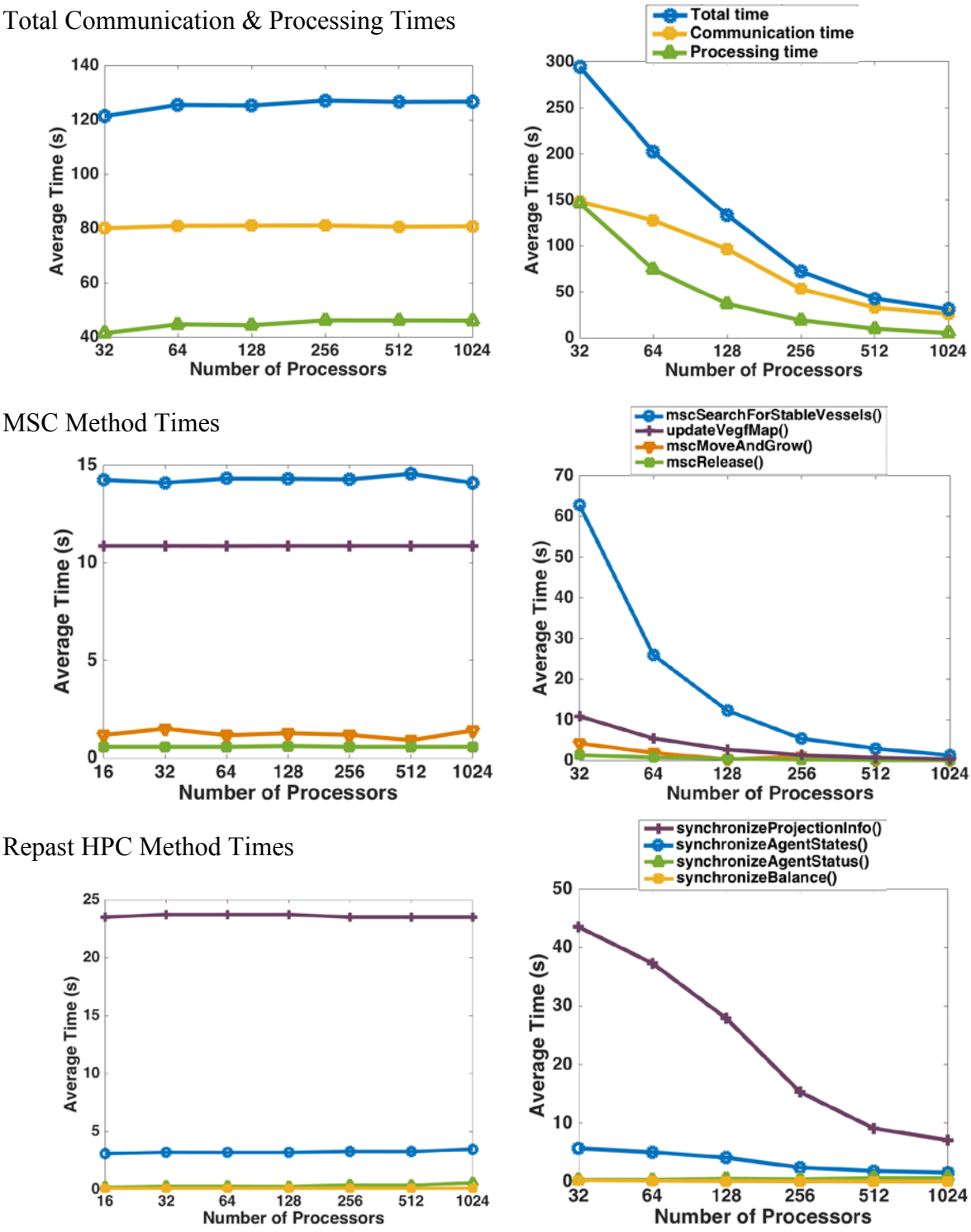

Figure 3: Weak Scaling Results (L) and Strong Scaling Results (R).

7 CONCLUSIONS AND NEXT STEPS

These initial tests are at considerably smaller scales than will be needed for biologically valid and medically relevant simulations. However, the capabilities of both Strong and Weak scaling will be

valuable when the simulations are scaled to real-world sizes: weak scaling will allow the problem size to be expanded to scales that include a much larger volume, approaching the sizes of bone grafts in use in medical contexts today; strong scaling can allow these larger simulations to make use of much higher numbers of processes- moving, for example, from 1024 processes on Vesta to ALCF's Mira, which can run simulations with as many as 786,432 processes (https://www.alcf.anl.gov/mira).

The choice to continue using Repast HPC's baseline and generic functionality, instead of customized and highly optimized code, is deliberate. The cell agents and biological processes used in these simulations represent only a small and preliminary part of a picture that is known to be richer. Repast HPC makes it easy to add new behaviors to existing cells and to expand the kinds of cells incorporated into the simulation, without the need to extensively revise the code or write custom MPI calls. This carries a recognized cost in terms of performance, but preserves flexibility appropriate to this stage of our project's development. The ability to rapidly develop new variants, and to test them first in small runs and then at larger scales quickly, will be key.

Repast HPC will enable further exploration and modeling of the biological processes in bone tissue growth. Expected next steps include the addition of different type of bone cell agents and growth factors released by these cells. The new 3-Dimensional spaces, and diffusion within those spaces, allow Repast HPC to put these additional biological avenues within easy reach.

ACKNOWLEDGMENTS

This research used resources of the Argonne Leadership Computing Facility, which is a DOE Office of Science User Facility supported under Contract DE-AC02-06CH11357. This study is based on research supported in part by the National Science Foundation (IIS- 1125412). We gratefully acknowledge the comments from three anonymous reviewers.

REFERENCES

Adams, R. H., and K. Alitalo. 2007. "Molecular Regulation of Angiogenesis and Lymphangiogenesis." *Nature Reviews Molecular Cell Biology* 8 (6):464-478.

Amini, A. R., C. T. Laurencin, and S. P. Nukavarapu. 2012. "Bone Tissue Engineering: Recent Advances and Challenges." *Critical Reviews in Biomedical Engineering* 40 (5):363-408.

Arvidson, K., B. M. Abdallah, L. A. Applegate, N. Baldini, E. Cenni, E. Gomez-Barrena, D. Granchi, M. Kassem, Y. T. Konttinen, K. Mustafa, D. P. Pioletti, T. Sillat, and A. Finne-Wistrand. 2011. "Bone Regeneration and Stem Cells." *Journal of Cellular and Molecular Medicine* 15 (4):718-46.

Bayrak, E. S., B. Akar, N. Xiao, H. Mehdizadeh, S. I. Somo, E. M. Brey, and A. Cinar. 2015a. "Agent-Based Modeling of Vascularization in Gradient Tissue Engineering Constructs." *IFAC-PapersOnLine* 48 (8):1240-1245.

Bayrak, E. S., H. Mehdizadeh, B. Akar, S. I. Somo, E. M. Brey, and A. Cinar. 2014. "Agent-Based Modeling of Osteogenic Differentiation of Mesenchymal Stem Cells in Porous Biomaterials." In *36th Annual International Conference of the IEEE Engineering in Medicine and Biology Society* 2924-2927.

Bayrak, E. S., T. Wang, A. Cinar, and C. Undey. 2015b. "Computational Modeling of Fed-Batch Cell Culture Bioreactor: Hybrid Agent-Based Approach." *IFAC-PapersOnLine* 48 (8):1252-1257.

Cockrell, R. C., S. Christley, E. Chang and G. An. 2015. "Towards Anatomic Scale Agent-Based Modeling with a Massively Parallel Spatially Explicit General-Purpose Model of Enteric Tissue (SEGMEnT_HPC)." *PLoS ONE* 10 (3):e0122192.

Collier, N., and M. North. 2012. "Parallel Agent-Based Simulation with Repast for High Performance Computing." *SIMULATION 89 (10):1215-1235.*

Damien, C. J., and J. R. Parsons. 1991. "Bone Graft and Bone Graft Substitutes: A Review of Current Technology and Applications." *Journal of Applied Biomaterials* 2 (3):187-208.

Deschepper, M., K. Oudina, B. David, V. Myrtil, C. Collet, M. Bensidhoum, D. Logeart-Avramoglou, and H. Petite. 2011. "Survival and Function of Mesenchymal Stem Cells (Mscs) Depend on Glucose to Overcome Exposure to Long-Term, Severe and Continuous Hypoxia." *Journal of Cellular and Molecular Medicine* 15 (7):1505-1514.

Giuliani, N., G. Lisignoli, M. Magnani, C. Racano, M. Bolzoni, B. Dalla Palma, A. Spolzino, C. Manferdini, C. Abati, D. Toscani, A. Facchini, and F. Aversa. 2013. "New Insights into Osteogenic and Chondrogenic Differentiation of Human Bone Marrow Mesenchymal Stem Cells and Their Potential Clinical Applications for Bone Regeneration in Pediatric Orthopaedics." *Stem Cells International* 2013:11.

Groen, D., N, J. Borgdorff, C. Bona-Casas, J. Hetherington, R. W. Nash, S. J. Zasada, I. Saverchenko, M. Mamonski, K. Kurowski, M. O. Bernabeu, A. G. Hoekstra, and P. V. Coveney. 2013. " Flexible Composition and Execution of High Performance, High Fidelity Multiscale Biomedical Simulations." *Interface Focus* 3 (2):20120087-20120094.

Langer, R. 1999. "Selected Advances in Drug Delivery and Tissue Engineering." *Journal of Controlled Release* 62 (1-2):7-11.

Langer, R., and J. P. Vacanti. 1993. "Tissue Engineering." *Science* 260 (5110):920-926.

Manassero, M., V. Viateau, M. Deschepper, K. Oudina, D. Logeart-Avramoglou, H. Petite, and M. Bensidhoum. 2013. "Bone Regeneration in Sheep Using Acropora Coral, a Natural Resorbable Scaffold, and Autologous Mesenchymal Stem Cells." *Tissue Engineering Part A* 19 (13-14):1554-1563.

MPI Forum 1994. "MPI: A Message-Passing Interface Standard." University of Tennessee.

North, M. J., N. T. Collier, J. Ozik, E. R. Tatara, C. M. Macal, M. Bragen, and P. Sydelko. 2013. "Complex Adaptive Systems Modeling with Repast Simphony." Complex Adaptive Systems Modeling 1 (1):1-26.

Paweletz, N., and M. Knierim. 1989. "Tumor-Related Angiogenesis." *Critical Reviews in Oncology/Hematology* 9 (3):197-242.

Perrin, D., H. J. Ruskin, and M. Crane. 2010. "Model Refinement through High-performance Computing: an Agent-Based HIV example." *Immunome Research* 6 (1):1-9.

Petite, H., V. Viateau, W. Bensaid, A. Meunier, C. de Pollak, M. Bourguignon, K. Oudina, L. Sedel, and G. Guillemin. 2000. "Tissue-Engineered Bone Regeneration." *Nature Biotechnology* 18 (9):959-963.

Qutub, A. A., F. Mac Gabhann, E. D. Karagiannis, P. Vempati, and A. S. Popel. 2009. "Multiscale Models of Angiogenesis: Integration of Molecular Mechanisms with Cell- and Organ-Level Models." *IEEE Engineering in Medicine and Biology Magazine* 28 (2):14-31.

Somo, S. I., B. Akar, E. S. Bayrak, J. C. Larson, A. A. Appel, H. Mehdizadeh, A. Cinar, and E. M. Brey. 2015. "Pore Interconnectivity Influences Growth Factor Mediated Vascularization in Sphere Templated Hydrogels." *Tissue Engineering Part C Methods* 20:20.

Tang, K., E. C. Breen, H. P. Gerber, N. M. Ferrara, and P. D. Wagner. 2004. "Capillary Regression in Vascular Endothelial Growth Factor-Deficient Skeletal Muscle." *Physiological Genomics* 18 (1):63-69.

AUTHOR BIOGRAPHIES

JOHN T. MURPHY, Ph.D., is a Fellow at the Computation Institute at the University of Chicago and a Computational Engineer at Argonne National Laboratory. At Argonne he is a member of the Global Security Sciences Division's System Science Center, Social and Behavioral Analysis Group. He specializes in computational social science and agent-based modeling of complex systems, especially coupled human and natural systems, and in the construction of ABM toolkits. His PhD is in Anthropology and Archaeology from the University of Arizona. His email address is johntmurphy@uchicago.edu.

ELIF SEYMA BAYRAK, Ph.D., is a Senior Engineer at Amgen in Digital Integration and Predictive Technologies group under Process Development organization. She received her Ph.D. degree from Illinois Institute of Technology in Chemical Engineering in Chicago. She focused on developing high performance agent-based models for bioengineering applications. She obtained her bachelor's degree from Hacettepe University in Ankara, Turkey. Her email address is ebayrak@amgen.com.

MUSTAFA CAGDAS OZTURK, Ph.D., is a Postdoctoral Fellow at the Chemical and Biological Engineering Department at Illinois Institute of Technology. His current research involves the development of high performance computing applications in the biomedical field. He holds a Ph.D. degree in Chemical Engineering from Illinois Institute of Technology, where he developed intelligent control systems for industrial applications. He obtained his B.S. and M.S. degrees from Hacettepe University in Ankara, Turkey. His email address is mozturk@iit.edu.

ALI CINAR, Ph.D., is a Professor of chemical engineering and biomedical engineering at the Illinois Institute of Technology, Chicago, IL, USA, and since 2004, he has been the Director of the Engineering Center for Diabetes Research and Education. He received the Ph.D. degree in chemical engineering from Texas A&M University. His current research interests include agent-based techniques for modeling, supervision, and control of complex and distributed systems, modeling of diabetes, angiogenesis and tissue formation, and adaptive control techniques for artificial pancreas systems for patients with diabetes. He has published two books, more than 190 technical papers in refereed journals and conference proceedings. Dr. Cinar is a senior member of the IEEE and a fellow of the AIChE. His email address is cinar@iit.edu.