Computer Physics Communications 192 (2015) 130–137

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc

A parallel implementation of an off-lattice individual-based model of multicellular populations

Daniel G. Harvey [a], Alexander G. Fletcher [b], James M. Osborne [a,c], Joe Pitt-Francis [a,*]

[a] Computational Biology Group, Department of Computer Science, University of Oxford, Parks Road, Oxford, OX1 3QD, UK

[b] Wolfson Centre for Mathematical Biology, Mathematical Institute, University of Oxford, Andrew Wiles Building, Radcliffe Observatory Quarter, Woodstock Road, Oxford, OX2 6GG, UK

[c] Computational Science, Microsoft Research, 21 Station Road, Cambridge, CB1 2FB, UK

ARTICLE INFO

Article history:
Received 10 September 2014
Received in revised form
28 January 2015
Accepted 10 March 2015
Available online 7 April 2015

Keywords:
Computational biology
Individual-based models
Off-lattice
Parallelisation
Scaling analysis

ABSTRACT

As computational models of multicellular populations include ever more detailed descriptions of biophysical and biochemical processes, the computational cost of simulating such models limits their ability to generate novel scientific hypotheses and testable predictions. While developments in microchip technology continue to increase the power of individual processors, parallel computing offers an immediate increase in available processing power. To make full use of parallel computing technology, it is necessary to develop specialised algorithms. To this end, we present a parallel algorithm for a class of off-lattice individual-based models of multicellular populations. The algorithm divides the spatial domain between computing processes and comprises communication routines that ensure the model is correctly simulated on multiple processors. The parallel algorithm is shown to accurately reproduce the results of a deterministic simulation performed using a pre-existing serial implementation. We test the scaling of computation time, memory use and load balancing as more processes are used to simulate a cell population of fixed size. We find approximate linear scaling of both speed-up and memory consumption on up to 32 processor cores. Dynamic load balancing is shown to provide speed-up for non-regular spatial distributions of cells in the case of a growing population.

1. Introduction

Individual-based simulation enables the modelling of complex systems, such as multicellular populations in biology, at increasingly detailed resolution [1]. An advantage of this approach is that the collective behaviour emerging from the interactions of heterogeneous individuals can be studied in a way that is not possible using classical methods. Furthermore, it is often possible to parameterise individual-based models using quantities that may be more readily measured experimentally, leading to an increasingly symbiotic relationship between modelling and the experimental sciences.

There are, however, a number of difficulties with the individual-based approach. One practical limitation is the computational cost of simulating such models. Simulations typically consist of a large number of interacting individuals, each of which has an internal state that evolves in response to interactions with other individuals according to pre-determined rules. The need to store the state of each of the n individuals in a simulation leads to a memory requirement that grows as $\mathcal{O}(n)$. Worse, for pairwise interactions between individuals, the processing load grows as $\mathcal{O}(n^2)$. This is a problem that is keenly felt in individual-based modelling of multicellular populations, where each cell is represented as a discrete entity. For large systems, computational cost can make simulations intractable, either as a result of excessive memory consumption or excessive processing time.

A common approach in the face of such computational limitations is to employ a continuum model. While it is possible to derive continuum models in certain circumstances from their individual-based counterparts under certain limits using coarse-graining techniques [2–4], or to combine continuum and discrete models in a hybrid system [5,6], such approaches are largely restricted to simplifications of the original model and many of the benefits of the individual-based approach are consequently lost. By increasing the computational resources available for simulation, it is possible to significantly increase the scale of model that may be

* Corresponding author.
E-mail addresses: danielharvey458@gmail.com (D.G. Harvey), alexander.fletcher@maths.ox.ac.uk (A.G. Fletcher), james.osborne@cs.ox.ac.uk (J.M. Osborne), joe.pitt-francis@cs.ox.ac.uk (J. Pitt-Francis).

http://dx.doi.org/10.1016/j.cpc.2015.03.005
0010-4655/

practically simulated using an individual-based approach without placing restrictions on the modelling methodology.

Parallel computing technology enables multiple single computational units to execute a program, such as a simulation of an individual-based model. This affords a significant increase in the available memory and computing power for a single simulation, potentially increasing the scale of simulation that is possible. The fact that such an approach has seen limited use in the field of individual-based modelling of cell populations in biology may be due to the difficulty in constructing computational algorithms that may be executed on parallel computing technology. The typical development cycle of software for such simulations is to begin with a simple serial algorithm, which is iteratively extended to more detailed modelling until the limits of computational tractability are hit. At this point it is often too costly to alter the software so that it may be executed in parallel, as there tend to be many explicit and implicit assumptions of serial execution within the program.

1.1. Individual cell-based modelling approaches

A variety of different discrete modelling approaches have been developed to describe how individual cells interact within a population. These range in complexity from simple cellular automata, where each cell is represented by a single site on a lattice, with cell state updated by pre-defined rules, to physical models where individual cells are partitioned into mesoscopic elements to better capture cell mechanics and deformation.

By imposing that the physical representation of cells is restricted to a fixed geometric lattice and updating their location in terms of rules or the minimisation of a heuristic potential function, lattice-based models such as cellular automata [7,8] and the cellular Potts model (CPM) [9–11] offer a simple description of the dynamics of cells within the population. This typically makes computation less expensive. However, parameters used to define such physical models are often difficult to relate to experimentally measurable quantities since they typically arise from heuristic rules. Furthermore, it is more difficult to define a time scale for such models so that simulation progression may be compared with temporal experimental data. In many cases these drawbacks are not critical to the biological hypotheses being investigated and lattice-based models provide an ideal framework. However, when the physical interaction of cells forms a key part of the model (for example, in studying contact inhibition of cell growth [12] an off-lattice approach is often favoured.

Common off-lattice approaches include cell-centre models, where each cell is represented by a single point in space with an associated volume (defined, for example, through a Voronoi tessellation), and vertex models, where each cell is represented by a polygon (or polyhedron), whose vertices are shared with neighbouring cells [13,14]. Here we focus on developing an algorithm for cell-centre models, which have been used to study biological processes such as solid tumour development [12,15], intestinal homoeostasis and carcinogenesis [16–18] and ductal carcinoma *in situ* [19].

1.2. Parallel approaches

Efforts to develop parallel algorithms for individual-based models of cell populations have largely been focused on the CPM. A parallel algorithm has been reported for the CPM that allows $\mathcal{O}(10^6)$ cells to be simulated on approximately 25 processors [20]. The authors decompose the lattice into spatial regions divided between the processors and employ a lock-based algorithm where neighbouring sites are locked (not updated) while a lattice site is being updated to ensure the model is correctly simulated in parallel. Although one might expect that imposing a wait on each processor using the locking algorithm during each update step would lead to large inefficiencies of the algorithm, efficiencies of between 80% and 90% are reported for simulations of $\mathcal{O}(10^6)$ cells. Similar approaches, based on decomposing space between processors, have been used to parallelise the algorithms of related lattice-based models [21,22].

Another notable example, from the wider field of agent-based modelling of multicellular populations, is the parallel functionality provided by the Flexible Large-scale Agent-based Modelling Environment (FLAME) [23]. Using the Message Passing Interface (MPI), FLAME has demonstrated 80% parallel efficiency on over 400 processes for a benchmark simulation, although the authors note that decomposing agents based on their state variables (for example, a space decomposition) gives rise to load balancing problems, where different processes may be responsible for a disproportionate amount of computation [24].

Some recent developments in parallel cell population simulations have focused on the use of General Purpose Graphics Processing Units (GPGPUs), which require modest capital outlay and achieve significant performance improvements over serial CPU algorithms. For example, a GPGPU approach has been shown to substantially improve the performance of the subcellular element model [25], a class of off-lattice model that includes a fine-grained description of cell mechanics [26]. By moving computationally costly tasks, such as neighbour identification, to the GPGPU the authors were able to achieve an 18-fold increase in performance. Nevertheless, they were only able to report on simulations where the total number of cells grew to approximately 5000, and such simulations took almost 10 h to compute. In other work a simulation of a cell population has been implemented in FLAME using a GPGPU combined with a spatial decomposition approach, obtaining speed-ups of 10,000 [27]. While such large speed-ups are available from GPGPU technology, they are limited in their memory capacity, and for detailed models of individual agents, may still pose a high communication overhead when transferring agents from the CPU to the GPGPU.

The greatest progress in the use of high-performance computing technology in cell-based modelling has come in the lattice-based paradigm. It has been suggested that simulating individual-based models with a high degree of connectivity between the individuals on distributed HPCs is ineffective due to communication latency and bandwidth restrictions, and that shared-memory approaches such as GPGPUs are of greater benefit [22]. However, GPGPU technology, while providing valuable speed-up for relatively small simulations, lacks the memory and processing power to tackle new scales of problem. On the other hand it has been shown that distributed HPC methods can be a viable option for large-scale simulation of cell population models [20].

To address the lack of suitable parallel methodologies for off-lattice, individual-based models of cell populations, in this paper we describe a novel parallel algorithm for a class of such models. Our algorithm combines techniques for domain decomposition from computational physics with a novel load balancing algorithm that improves efficiency of simulations where the spatial domain is not known ahead of runtime. In contrast to GPGPU or shared memory approaches, which are tied to a specific computational system design, we choose to use the architecture-independent Message Passing Interface (MPI) to allow for an implementation that is portable between multi-threaded and multi-core shared-memory and grid-based distributed memory parallel computing systems. By implementing inter-process communication in a model-agnostic manner, our implementation is able to support parallel simulations of unbounded variety by modification of the individual-cell model, and of the cell–cell interaction rules. Our parallel algorithm only places the restriction of spatial locality on cell–cell interaction laws.