



**UvA-DARE (Digital Academic Repository)**

**Distributed multiscale computing**

Borgdorff, J.

[Link to publication](#)

*Citation for published version (APA):*  
Borgdorff, J. (2014). Distributed multiscale computing

**General rights**

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

**Disclaimer/Complaints regulations**

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <http://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.

# C

## Application and model details

### C.1 Tokamak plasma (TTE, fusion)

The chosen Fusion application TTE [72] simulates the time evolution of profiles in the core plasma of a Tokamak. This application consists of three submodels: a transport equations solver to evolve the profiles, a fixed-boundary equilibrium code to provide geometry of the plasma and a turbulence code to calculate anomalous transport coefficients. Different versions of such submodels have been developed within the EDFA Task Force on Integrated Tokamak Modelling (ITM<sup>1</sup>). The structured data for these submodels are stored in a local database and accessed through a specific library on the Gateway cluster in Garching (DE). The coupled application is implemented using MUSCLE 2.

We have chosen to run this benchmark using only two submodels for sake of conciseness (the equilibrium is considered fixed during this simulation): the transport solver coming from the ETS [40] and turbulence coming from a gyrofluid approximation flux-tube code GEM [124].

The use case corresponds to a small Tokamak plasma (e.g ASDEX-Upgrade), where ETS evolves temperature and density profiles for electrons and one ion species, and GEM runs on 8 flux surfaces, each of which is calculated independently, using time averaging to provide transport coefficients on transport timescale. We have run this case

---

<sup>1</sup>ITM: <http://portal.efda-itm.eu/itm/portal/>

on the Gateway cluster in Germany (GEM up to 256 cores) using fully the database access, and on the Helios supercomputer in Japan (GEM up to 1024 cores) using a tedious ad-hoc setting to simulate the database. With a transport time step at 0.01s, such application can simulate 10s of physical time in less than 17 hours on 1024 cores (Helios). The model has improved our understanding of the different submodel parameters impact on simulation stability at longer time frames (time averaging, equilibrium model limitations, geometric representation), which is one of the main obstacle to overcome before applying such models to predictive simulations on ITER sized tokamaks. When distributed simulations will be allowed on Helios supercomputer, it will be possible to run routinely such simulations on small to medium Tokamak cases in order to validate the approach using experimental data stored in the local database. Finally, the main goal is to run large Tokamak cases (ITER will require a grid at least 64 times bigger), to replace the gyrofluid approximation by a gyrokinetic code and use additional submodels (e.g heating sources) to complete the description of the physics.

## C.2 Cerebrovascular blood flow (HemeLB, biomedicine)

In this application we use distributed multiscale computing to connect flow models of the major vessels arterial tree to models of local cerebral vasculature [60]. We do this to improve the accuracy of our simulations of cerebral aneurysms [18], and to understand the effect of patient-specific properties (e.g., heart rate or the structure of the circle of Willis) on the flow dynamics in local cerebral vasculature and aneurysms. Our main motivation to rely on MAPPER is to allow each code to be used in an optimal environment, and to minimise the time to completion for the overall system.

The HemeLB model couples two submodels together: PyNS and HemeLB. We use PyNS [100] to model the arterial tree in the human body, and HemeLB to model a local network of arteries in the brain. The two applications exchange pressure values at the boundaries, with the HemeLB simulation domain being a small subset of the overall network modelled in PyNS.

We configured PyNS, which is a 0/1 dimensional Discontinuous Galerkin solver, with a 70 bpm heart rate, 90 mmHg mean pressure, and 5.98 L/min cardiac output. The submodel runs for 4,000 time steps on a local desktop at UCL in London (4 cores), as it has a number of Python dependencies and does not scale to large core counts. PyNS exchanges pressure values with HemeLB at every time step.

We configured HemeLB, which is a 3 dimensional lattice-Boltzmann simulation environment, with a vascular network model with 4.2 million lattice sites, each site sized  $3.5 \times 10^{-5}$  m. It is run for 40,000 time steps on the HECToR supercomputer (512 or 2048 cores), which is located at EPCC in Edinburgh. HemeLB exchanges pressure values with PyNS at every 10 time steps (for a total of 4,000 exchanges).

The codes are coupled with the MPWide communication library.

### C.3 Irrigation network (Canals, hydrology)

The water service demands are keeping increasing rapidly. They are driven by several factors like the impact of climatic conditions, agriculture, overpopulation, etc. [102]. Today, the technical heritage to manage resources water and complex irrigation systems becomes somehow insufficient to cope with the new urgent needs and to predict in advance solutions for emergency cases (e.g. natural hazards). One way to deal with this is to use numerical models and computer simulations as decision making tool.

One of our interest in collaboration with the Geneva electricity company is to simulate a section of the *Rhone* river (13 km), from the *Geneva* city down to *Verbois* water dam [114]. This simulation investigates the possibility to study specific questions namely the way to flush sedimentation in some critical area by changing the water levels. This requires a 3D free surface (3DFS) model with sediment to study these critical areas and 1D shallow water to study areas where the flow behaviour is stable. The 3DFS model requires supercomputers capabilities contrary to 1D model which can be executed on clusters or desktop machines. Coupling 3DFS and 1D models results in a multiscale models and simulation [16]. In addition, using 3DFS with high resolution to get more accurate results can not be afforded by what a local centre can provide in term of computing power. The MAPPER framework allows to easily connect these different components and run them on a distributed grid of supercomputers [17].

### C.4 Reverse-engineering of gene-regulatory networks (MultiGrain, systems biology)

Gene regulation networks (GRNs) can be viewed as the “cognitive architecture” in living cells, they control how much product (protein, RNA) is synthesised and when

in response to input from the external or internal environment. Modelling and simulation of GRNs is an unsolved problem [101]. Furthermore, future models of GRNs are likely to be considerably more complex (number of genes, biochemical and physical levels involved) than their current counterparts. Our vision is to enable modelling and simulation of large gene-regulation networks that exhibit an inherent sub-division particularly in the time dimension. Such gene networks typically consist of dozens or hundreds of genes. They are difficult to handle with conventional modelling and simulation approaches due to the conceptual and computational complexities involved.

MultiGrain/MAPPER is a distributed multiscale computing application, designed for modelling and simulation of large GRN systems based on MAPPER multiscale computing solutions. MultiGrain is a Java library providing a number of GRN reverse-engineering features, including flexible specification of GRN rate laws, flexible algorithms separating GRN structure inference and time-course data fitting, and distributed optimisation based on a particle swarm optimisation [81] approach enabled by MAPPER components including MUSCLE 2 and QosCosGrid. Consequently, MultiGrain is able to run on a variety of hardware from personal computers to multi-processor architectures, computer clusters or even supercomputers.

## C.5 Clay-polymer nanocomposites (Nano, nanomaterials)

In this application we assess the materials properties of clay-polymer nanocomposites. The nanocomposites fall within the realm of the emergent area known as nanotechnology, where materials are designed and built at the atomic level, an area of science currently at the forefront of academic, industrial, health and public interest. Tailoring the clay structure on the nanometer scale produces composites with novel material properties, which can be significantly different from bulk properties and have already been applied in numerous commercial applications, such as in the automotive industry, biodegradable food packaging and in oil drilling fluids.

Our multiscale model involves 4 stages of execution:

1. Parametrising the potentials of clay sheet edges with CPMD [1], using up to 64 cores on a local cluster (Mavrino at University College London).
2. Modelling basic clay sheets and polymers using LAMMPS [2] in All-Atom mode

(the former with the previously calculated potentials), using up to  $\sim 1024$  cores on HECToR.

3. Iteratively coarse-graining and optimising the potentials using the Iterative Boltzmann Inversion (a few of the potentials were done using force matching), to construct accurate coarse-grained potentials, using  $\sim 32$  cores on Mavrino or HECToR.
4. Executing coarse-grained production simulations using these newfound potentials, using 1000s or 10000s of cores on HECToR.

In this paper we presented the performance of our simulations for one particular production system, which contains montmorillonite clay and uncharged polyvinyl alcohol.

## C.6 In-stent restenosis (ISR<sub>3D</sub>, biomedicine)

A common expression of coronary heart disease is arteriosclerosis: thickened and hardened blood vessels caused by a build-up of atheromatous plaque. This is associated with a partial occlusion of the blood vessel lumen, called a stenosis. A common intervention to open up the lumen is balloon angioplasty, assisted by a stent that keeps the lumen open after the intervention. An adverse response to the stent, a regrowth of tissue, may result in a re-occlusion of the lumen, called in-stent restenosis [80, 109]. The exact causes and key factors of in-stent restenosis are not yet known [46]. The three-dimensional model of in-stent restenosis ISR<sub>3D</sub> models the response to the stent-assisted angioplasty, taking into account tissue damage due to the stent placement, smooth muscle cell proliferation and migration, shear stress caused by blood flow, and the effect of local drug diffusion if a drug-eluting stent [32, 137, 139].

ISR<sub>3D</sub> has a cyclic coupling topology, alternating mainly between a smooth muscle cell submodel and a blood flow submodel. The smooth muscle cell code is parallelised with OpenMP, limiting it to a single compute node, while the blood flow code uses a lattice Boltzmann library, Palabos<sup>2</sup>, that has MPI parallelisation and scales well [20]. Depending on the blood vessel geometry and the resolution of the blood flow solver, a simulation consisting of 1500 time steps takes a few days to more than a week.

---

<sup>2</sup>Palabos: <http://palabos.org>