Collective Cell Behaviour – a Cell-Based Parallelisation Approach for a Phase Field Active Polar Gel Model

Simon Praetorius and Axel Voigt

Institute of Scientific Computing, Technische Universität Dresden, 01062 Dresden, Germany
E-mail: {simon.praetorius, axel.voigt}@tu-dresden.de

We consider a continuum model for collective cell movement. Each cell is modelled by a phase field active polar gel model and the cells interact via steric interactions. We provide a finite element implementation with a parallel efficiency of at least 0.5 in the number of cells. This is achieved by considering each cell on a different processor and various improvements to reduce the communication overhead to deal with the cell-cell interactions. We describe implementation details and demonstrate results for up to 768 cells.

1 Introduction

Computational modelling of cell movement (and related topics such as tissue dynamics) is a large and very active topic at the frontiers between physics, biology, mathematics and biomedical engineering. Various computational approaches have been applied to this topic. We here consider a phase field approach, where the most recent progress in modelling of single and collective cell behaviour was achieved[1–6]. With the provided highly parallel implementation of the phase field active polar gel model[7,8,6] previous results can be extended to larger numbers of cells and simulations now allow the study of emerging phenomena, such as cluster formation and phase-separation, which so far could only be modelled using coarse grained particle models, e.g. Refs. 9–12. Our approach now allows to consider for example the influence of shape change and elastic interaction on contact on the overall dynamics. The shape evolution of cells is thereby described by a geometric evolution law and the active motion by a splay instability in the cell bulk that induces a net-polarisation of the cell. Each cell is described by its own phase field variable and polarisation field. The interaction is based on a steric interaction term dependent on the distance of the cells.

While the phase field active polar gel model for one cell can be efficiently solved using adaptive finite elements in space and standard time discretisation methods[4] and an extension to a moderate number of cells can be handled using an OpenMP parallelisation[6], these approaches are no longer sufficient for hundreds of cells. Also classical domain decomposition approaches would be inefficient to handle hundreds of coupled partial differential equations, due to a huge memory and communication overhead. We therefore consider a different parallelisation approach, based on a physical decoupling. Each processor will handle the evolution of one cell and communication is only needed to consider the interaction between cells. The approach is related to adaptive full domain covering meshes[13,14] but achieves a better performance. Each processor handles a mesh of the whole domain, which is adaptively refined according to its phase field variable. Communication thus requires the interaction of variables which are defined on differently refined meshes. Such interactions have been considered in sequential discretisations using the multi-mesh approach[15,16], which we adapt and parallelise. Our approach will in principle allow scaling

with the number of cells as their local evolution is perfectly parallelisable. Only the interaction requires special care as its effort increases with the number of cells. To deal with this interaction we adapt parallel concepts used for particle methods.

After briefly introducing the model we describe in the following sections the implementation of both, the local evolution equation and a communication procedure that allows an efficient parallel setup and discuss some physical results.

2 Mathematical Model

We consider $i = 1, \dots, N$ phase field variables ϕ_i and polarisation fields $\mathbf{P}_i$, for which the coupled evolution equations read

$$\begin{aligned}
\partial_t \phi_i + v_0 \nabla \cdot (\phi_i \mathbf{P}_i) &= \gamma \Delta \mu_i \,, \\
\mu_i := \frac{\delta \mathcal{F}}{\delta \phi_i} &= \frac{1}{Ca}\Big(-\epsilon \Delta \phi_i + \frac{1}{\epsilon} W'(\phi_i)\Big) \\
&\quad + \frac{1}{In}\Big(B'(\phi_i) \sum_{j \neq i} w(\phi_j) + w'(\phi_i) \sum_{j \neq i} B(\phi_j)\Big) \\
&\quad + \frac{1}{Pa}\Big(-\frac{c_1}{2} \|\mathbf{P}_i\|^2 - \beta \nabla \cdot \mathbf{P}_i\Big) \,, \\
\partial_t \mathbf{P}_i + (v_0 \mathbf{P}_i \cdot \nabla)\mathbf{P}_i &= -\frac{1}{\kappa} \mathbf{H}_i \,, \\
\mathbf{H}_i := \frac{\delta \mathcal{F}}{\delta \mathbf{P}_i} &= \frac{1}{Pa}\Big(-c_1 \phi_i \mathbf{P}_i + c_1 \|\mathbf{P}_i\|^2 \mathbf{P}_i - \Delta \mathbf{P}_i + \beta \nabla \phi_i\Big) \,,
\end{aligned} \tag{1}$$

in $\Omega \times (0, T]$ for some simulation end time $T > 0$ and a two-dimensional domain Ω. We assume in the following periodic boundary conditions for Ω. The model results as an H^{-1} gradient flow of the energy $\mathcal{F}$ w.r.t. ϕ_i and an L^2 gradient flow w.r.t. $\mathbf{P}_i$ with

$$\begin{aligned}
\mathcal{F}[\{\mathbf{P}_i\}, \{\phi_i\}] = \sum_i \frac{1}{Ca} \int_\Omega \frac{\epsilon}{2} \|\nabla \phi_i\|^2 + \frac{1}{\epsilon} W(\phi_i) \, \mathrm{d}\mathbf{x} \\
+ \frac{1}{In} \int_\Omega B(\phi_i) \sum_{j \neq i} w(\phi_j) \, \mathrm{d}\mathbf{x} \\
+ \frac{1}{Pa} \int_\Omega \frac{1}{2} \|\nabla \mathbf{P}_i\|^2 + \frac{c_1}{4} \|\mathbf{P}_i\|^2 (-2\phi + \|\mathbf{P}_i\|^2) + \beta \mathbf{P}_i \cdot \nabla \phi_i \, \mathrm{d}\mathbf{x}
\end{aligned}$$

considering the surface energy, interaction terms and a polar liquid crystal energy to model the actin filaments in the cells. The parameters Ca, In and Pa act as weighting between these contributions. The surface energy is a classical Ginzburg-Landau function with double-well potential $W(\phi) = \frac{1}{4}(\phi^2 - 1)^2$ and interface thickness ϵ. The interaction term considers $B(\phi) = \frac{3}{\epsilon\sqrt{2}} W(\phi) \approx \delta_\Gamma$ an approximation of the surface delta function and

$$w(\phi_j) = \exp(-d_{\phi_j}^2 / \epsilon^2), \quad \text{with} \quad d_{\phi_j}(\mathbf{x}) = -\frac{\epsilon}{\sqrt{2}} \ln \frac{1 + \phi_j(\mathbf{x})}{1 - \phi_j(\mathbf{x})} \tag{2}$$

an approximation of a short range interaction potential, with signed distance function computed from the tanh-profile of the phase field functions. The polar liquid crystal energy

is of Frank-Oseen type and c_1, β are parameters controlling the deformation of the polarisation field $\mathbf{P}$ within the cell bulk and the anchoring on the cell interface. Activity is introduced in the Eqs. 1 by a self-propulsion term, with velocity v_0. For more details we refer to Refs. 4, 6, 17, 16. Fig. 1 shows the phase field variable of one cell together with its polarisation field and the net polarisation (a), as well as a snapshot of the interaction of 48 cells, for which only the cell boundaries Γ_i and the net polarisation in each cell are shown (b).

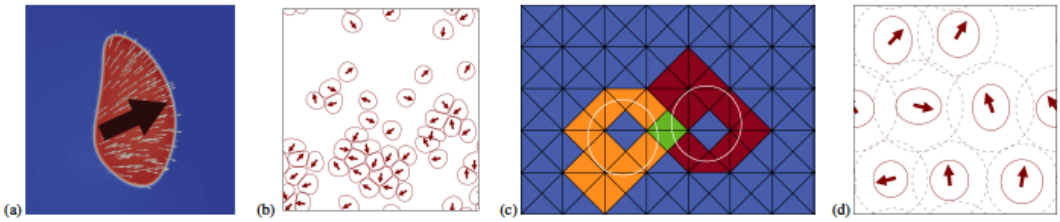

Figure 1. (a) Phase field variable $\phi_i \approx 1$ (red) cell bulk and $\phi_i \approx -1$ (blue) and polarisation field (white arrows) together with net polarisation (black arrow). (b) Evolution of 48 cells by cell activity. The arrows indicates the net polarisation of each cell. The cell boundaries Γ_i are the 0 level lines of the phase field variables. (c) The grid-elements of two overlapping interaction-regions are highlighted. (d) Cells with enclosing balls (dashed lines) defining neighbourhoods by overlap.

3 Implementation

The evolution equations are solved by using the finite-element method and are implemented in the framework AMDiS[18,19]. Therefore, the domain Ω is partitioned in conforming triangulation $\mathcal{T}_i$, all resulting as an adaptive refinement of the same coarse triangulation $\mathcal{T}$. A simple discretisation in time by semi-implicit backward Euler discretisation is applied, utilising a sequential operator splitting approach to separately solve for ϕ_i and $\mathbf{P}_i$ and allow to evaluate the interaction terms in the old-timestep. As each cell is described by its own phase field ϕ_i and its own polarisation field $\mathbf{P}_i$ the integration of the dynamic equations can be separated. We assign each cell to its own processor. This physical decomposition guarantees a perfect scaling in the number of cells N for the local problem, even if made more complex by incorporating more physical aspects of cell motility and shape evolution, and the interaction only requires communication between the processing units. With an increase in the number of cells the communication procedure gets more and more complicated as in principle each cell can interact with all other cells. However, this N^2 complexity can be improved.

Due to the short interaction range and the structure of the interaction potential in Eq. 2 which is of exponential decay, a cut-off distance r_c from the cell-interface can be defined and only cells within an overlapping thin shell $\Gamma_i^{r_c} := \Gamma_i \times [-r_c, r_c]$ need to be considered for interaction. Within the finite element discretisation we thus need to consider the set of neighbouring cells $\mathbb{J}_i$ interacting with cell i, i.e., $\mathbb{J}_i := \{j \,:\, \Gamma_i^{r_c} \cap \Gamma_j^{r_c} \neq \emptyset\}$ and $\mathcal{T}_{ij}$ the set of grid-elements overlapping the interaction region $\Gamma_i^{r_c} \cap \Gamma_j^{r_c}$, such that

$$\sum_{j \in \mathbb{J}_i} \sum_{T \in \mathcal{T}_{ij}} \int_T (B'(\phi_i) w(\phi_j) + B(\phi_j) w'(\phi_i)) \vartheta \, \mathrm{d}\mathbf{x}\,, \qquad \forall \vartheta \in V, \tag{3}$$

with testfunction ϑ and finite-element space V, is sufficient for the integration of the interaction term in Eqs. 1. In Fig. 1 (c) two cells are sketched, each with an interaction region $\Gamma_i^{r_c}$ overlapping some grid-elements (in red and yellow). The intersection of these grid-elements (green) builds the support of the interaction term between the two cells. For the construction of the neighbour lists $\mathbb{J}_i$ and lists of overlapping interface grid elements $\mathcal{T}_{ij}$, we follow a two-step procedure: (1) Determine which cell needs to communicate with which other cell. (2) Exchange the indices of the grid elements in the interface between neighbouring cells. For the first step, we enclose each cell in a ball and communicate just the ball centre and radius, leading to a simple construction of a Verlet list for overlapping balls, see Fig. 1 (d). For cell i the algorithms reads

```
center = argmin(d_Γi(x) for x in Ω)
radius = max([distance(center, x) for x in Γi]) + rc

# communicate with all cells
centers, radii = all_gather( (center, radius) )

Ji = []
for j in cells:
  if j != i and distance(centers[j], center) < radii[j] + radius:
    Ji.append(j)
```

Listing 1. Build neighbour list.

where `distance(a,b)` respects the periodic boundary conditions of the domain Ω. The radius of a cell is any radius large enough to enclose the cell in a ball around the centre incorporating an interaction radius r_c. For the second step, we need to collect interface elements and to communicate these elements between neighbouring cells.

```
interface = []
for T in T(Ω):
  if any([abs(d_φi(v)) < rc for v in vertices(T)]):
    interface.append(T)

for j in Ji:
  interfacej = send_recv(interface, _to=j)
  Tij = [T for T in interface if T in interfacej]
```

Listing 2. Collect and exchange overlapping elements.

where a (coarse-grid) element $T \in \mathcal{T}(\Omega)$ is assumed to be a tuple of vertices (v_0, v_1, v_2). Since the phase field variable is nearly constant away from its interface and has a steep transition from -1 to 1 in the interface region, an adaptively refined grid is used to resolve this interface smoothly. This leads to a large number of grid elements that would need to be interchanged in the `interface` list and thus increases the communication cost rapidly. Instead of this strategy only the coarse-grid elements are collected and exchanged. However, for the interaction we need the phase fields from neighbouring cells and thus need to communicate the (fine-grid) phase field data on the overlapping coarse-grid elements.

```
for j in 𝕁_i:
  for T in 𝒯_ij:
    φ_j|_T = send_recv(φ_i|_T, _to=j)
    assemble(B'(φ_i)w(φ_j) + B(φ_j)w'(φ_i), _on=T) # see Eq. 3
```

Listing 3. Communicate phase field data.

Since each phase field lives on a different refined grid with a common coarse-grid, the phase field data communicated must incorporate position of the fine-grid values (DOFs) in the overlapping elements. A compact data-structure to represent all positions and values of a refined coarse-grid element is described in Refs. 15, 19, 16 and is based on the representation of a binary refinement tree as a 64-Bits structure-code.

```
visited = [False]*num_dofs
values = []
for t in leaf_elements(T): # pre-order traverse
  for dof in DOFs(t):
    if visited[dof]:
      continue

    values.append(φ_i[dof])
    visited[dof] = True
```

Listing 4. Collect DOFs to communicate.

On the receiver side, the structure-code generates a refinement of a coarse-grid element and afterwards, the data can be distributed to the leaf DOFs. This allows to reconstruct the phase field solution on another processor and afterwards the evaluation of Eq. 3 with two differently refined phase field representations.

4 Numerical Experiments

We study the motion and interaction of N cells distributed initially in a rectangular periodic domain. The shape and position of each cell is a perturbation of a circular object with a fixed radius $R = 3$ in a triangular lattice. Two different setups are considered. `setup-1` considers a domain of fixed size. The cells are packed in a rectangular cluster of local high packing fraction. In `setup-2` the cells are distributed in the whole domain such that a given volume fraction φ of cells is achieved. This requires the scaling of the domain size and thus changes the local problem. The initial orientation $\mathbf{P}_i(t = 0)$ is set to a normalised random vector constant inside the cells and zero outside in both settings.

In all simulations the parameters $\epsilon = 0.15$, $\gamma = \kappa = 1$, $Pa = 1, In = 0.05$, $Ca = 0.025$, $v_0 = 2.5$ and $c_1 = 10$ are fixed. We here consider only different values of β. It's influence on the emergence of collective motion can be seen in Fig. 2, where the global orientational order $\theta(\mathbf{d}) = \big(\operatorname{atan}(d_y/d_x) + \pi\big)/(2\pi)$ is analysed. It is 1 if all cells are oriented in the same direction and 0 if all polarisations $\mathbf{d} = \mathbf{P}_i$ or cell velocities $\mathbf{d} = \mathbf{v}_i$ are independent. The results are in excellent agreement with the phase field crystal simulations in Ref. 12 and thus validate this coarse-grained description. Other effects, e.g. the influence of the deformability of the cells, characterised by Ca, will be studied elsewhere. We are here more concerned with the scaling properties. In Fig. 3 the two configurations `setup-1` and `setup-2` are considered for an increasing number of cells. While for `setup-1` an efficiency above 0.5 can be achieved for 768 cells, `setup-2` be-

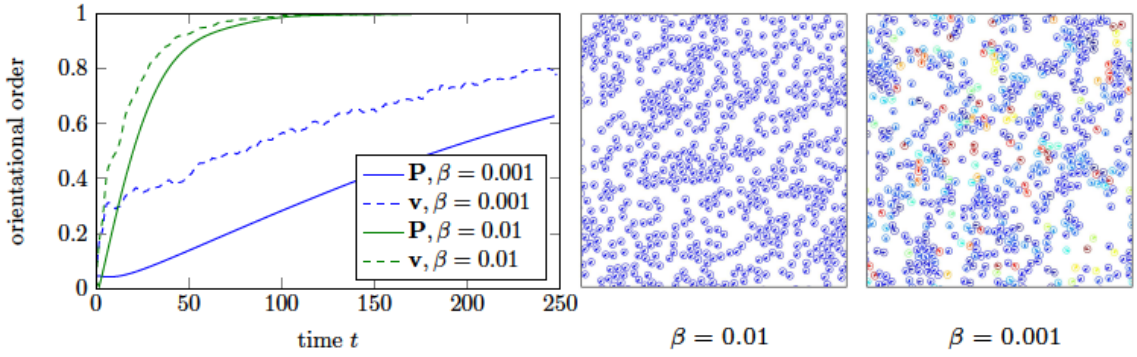

Figure 2. (left) Orientational order θ for two different values of β simulated for $N = 576$ cells in `setup-2`. (centre) Configuration for $\beta = 0.01$ at time $t = 200$. (right) Configuration for $\beta = 0.001$ at time $t = 200$, different cell colours indicate different orientations.

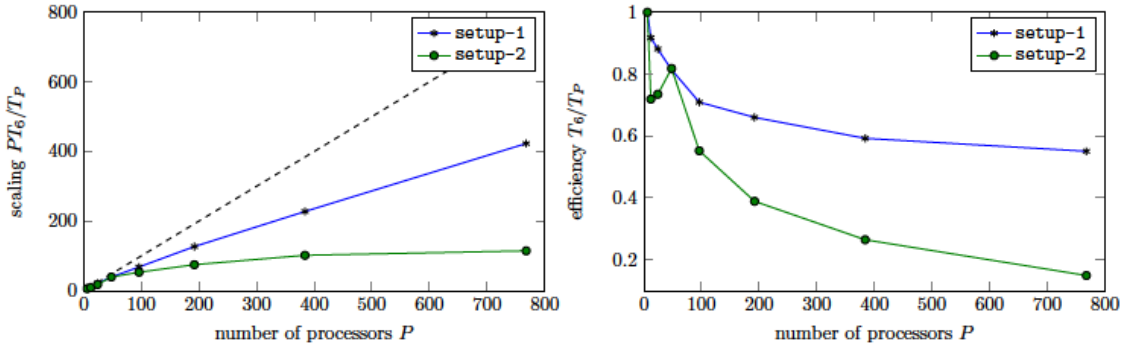

Figure 3. (left) Weak scaling for two different configurations. The dashed line corresponds to ideal scaling. (right) Corresponding parallel efficiency. In both cases the simulation time is compared to a configuration with 6 cells.

comes inefficient for large cell numbers. This results from the adjustment of the coarse grid, which leads to changes in the local problems and an increased complexity.

One parameter to adjust is the resolution of the coarse-grid, as it controls the communication cost. To determine the optimal background mesh we vary the resolution of the coarse-grid, while keeping the fine-grid resolution on the cell interface fixed. The domains away from the cell interface are resolved with the coarse-grid resolution. Thus, a finer coarse-grid results in a larger total number of DOFs. Measuring the total simulation time and comparing for various coarse-grid element sizes, defined as the maximal edge length in the coarse-grid, we find the optimal background mesh to have an element-size of $6 - 8$. With the mean cell radius $R = 3$, this gives an element-size similar to the cell diameter, see Fig. 4 (left). A more detailed view on the timings, see Fig. 4 (right), shows that the update of $\mathbb{J}_i$ and $\mathcal{T}_{ij}$ gets cheaper for larger coarse-grid elements, whereas the communication costs increase. Since the overall number of DOFs also increases, the time to assemble and invert the linear system makes the biggest contribution in the time differences.

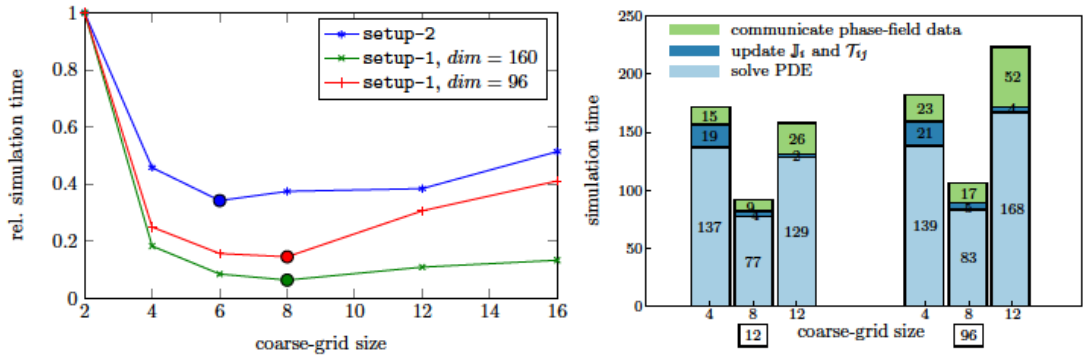

Figure 4. (left) Comparison of simulation time for different coarse-grid element sizes. Three setups are considered. The highlighted marker determines the minimal simulation time. (right) The partitioning of the total simulation time corresponding to the red curve for two different numbers of processors (12 and 96) and three different coarse-grid element sizes. Inside the bars, the partial times of the simulations steps are given.

5 Conclusion and Outlook

We here consider a phase field active polar gel model to simulate cell movement. While the model has been used before for single cells and the interaction of moderate numbers of cells, our implementation allows for collection of cells in the order of 1.000, which could be reached before only with coarse-grained models. Each cell is considered by a separate set of phase field and polarisation variables. This allows to handle complex physical models for shape change and activity mechanisms and to parametrise each cell individually. The parallelisation of the huge system of partial differential equations considers each cell on its own processing unit. This allows perfect scaling in the number of cells for the local problem, only the cell-cell interaction requires communication between the processing units. Our parallelisation approach reduces this communication costs by construction of a communication list and by sending only values on coarse-grid elements in an overlapping region. The results show an optimal balance between communication cost and cost due to computing time for coarse grid elements which are of the size of the cells. This leads to a moderate scaling behaviour tested on up to 768 processors.

Further increasing the efficiency could be achieved by combining MPI based parallelisation with a local shared-memory based parallelisation. If the local computing power per processor allows, each unit can also handle not just a single, but a set of cells, all on different grids and with different phase fields. This would reduce the costs for communication further, while increasing the workload per processor. Also an extension to three space dimensions is possible, as long as the complexity for a single cell can be handled by one processing unit. We thus expect the approach to be valuable for further investigations in collective cell dynamics.

Acknowledgements

The authors gratefully acknowledge the computing time granted by the John von Neumann Institute for Computing (NIC) and provided on the supercomputer JURECA at Jülich Supercomputing Centre (JSC).

References

1. F. Ziebert, S. Swaminathan, and I. S. Aranson, *Model for self-polarization and motility of keratocyte fragments*, J. R. Soc. Interface, **9**, 1084–1092, 2012.
2. E. Tjhung, D. Marenduzzo, and M. E. Cates, *Spontaneous symmetry breaking in active droplets provides a generic route to motility.*, P. Natl. Acad. Sci. USA, **109**, 12381–12386, 2012.
3. F. Ziebert and I. S. Aranson, *Effects of adhesion dynamics and substrate compliance on the shape and motility of crawling cells*, Plos One, **8**, e64511, 2013.
4. W. Marth, S. Praetorius, and A. Voigt, *A mechanism for cell motility by active polar gels*, J. R. Soc. Interface, **12**, 20150161, 2015.
5. J. Löber, F. Ziebert, and I. S. Aranson, *Collisions of deformable cells lead to collective migration*, Sci. Rep., **5**, 9172, 2015.
6. W. Marth and A. Voigt, *Collective migration under hydrodynamic interactions: a computational approach*, Interface Focus, **6**, 20160037, 2016.
7. K. Kruse and F. Jülicher, *Actively contracting bundles of polar filaments*, Phys. Rev. Lett., **85**, 1778–1781, 2000.
8. K. Kruse, J. F. Joanny, F. Jülicher, J. Prost, and K. Sekimoto, *Asters, vortices, and rotating spirals in active gels of polar filaments*, Phys. Rev. Lett., **92**, 078101, 2004.
9. D. Grossman, I. S. Aranson, and E. B. Jocob, *Emergence of agent swarm migration and vortex formation through inelastic collisions*, New J. Phys., **10**, 023036, 2008.
10. J. Palacci, S. Sacanna, A. P. Steinberg, D. J. Pine, and P. Chaikin, *Living crystals of light-activated colloidal surfers*, Science, **339**, 936–940, 2013.
11. J. Stenhammar, R. Wittkowski, D. Marenduzzo, and M. E. Cates, *Activity-Induced phase separation and self-assembly in mixtures of active and passive particles*, Phys. Rev. Lett., **114**, 018301, 2015.
12. F. Alaimo, S. Praetorius, and A. Voigt, *A microscopic field theoretical approach for active systems*, New J. Phys., **18**, 083008, 2016.
13. R. E. Bank and M. Holst, *A new paradigm for parallel adaptive meshing algorithms*, SIAM Review, **45**, 291–323, 2003.
14. S. Vey and A. Voigt, *Adaptive full domain covering meshes for parallel finite element computations*, Computing, **81**, 53–75, 2007.
15. A. Voigt and T. Witkowski, *A multi-mesh finite element method for Lagrange elements of arbitrary degree*, J. Comput. Sci., **3**, 420–428, 2012.
16. S. Ling, W. Marth, S. Praetorius, and A. Voigt, *An adaptive finite element multi-mesh approach for interacting deformable objects in flow*, Comput. Methods Appl. Math., **16**, 475–484, 2016.
17. W. Marth, S. Aland, and A. Voigt, *Margination of white blood cells - a computational approach by a hydrodynamic phase field model*, J. Fluid Mech., **709**, 389–406, 2016.
18. S. Vey and A. Voigt, *AMDiS: adaptive multidimensional simulations*, Comput. Vis. Sci., **10**, 57–67, 2007.
19. T. Witkowski, S. Ling, S. Praetorius, and A. Voigt, *Software concepts and numerical algorithms for a scalable adaptive parallel finite element method*, Adv. Comput. Math., **41**, 1145–1177, 2015.