cellular_raza - Novel Flexibility in Design of Agent-Based Models in Cellular Systems

Jonas Pleyer[1], Christian Fleck[1]
[1]Freiburg Center for Data-Analysis and Modeling

Agent-Based Models (ABMs) allow researchers to describe complex cellular systems in a mechanistic manner but can also abstract over less-known processes. It is often desirable to exchange only parts of the model eg. changing the spatial representation of cells from a spherical interaction potential to an elliptical. Existing tools lack in flexibility and cannot change their internal representation of cells. To solve these problems we created `cellular_raza`, a novel library that offers previously unknown flexibility in model design while retaining excellent performance.

Features

- Generic Progamming allows for unparalleled flexibility
- Parallelized (via OS-threads and Domain-decomposition)
- Produces deterministic results
- Modular
- No inherent assumptions
- User has complete control over every parameter and functionality
- Free software (GPLv2.0)

Scaling Behavior

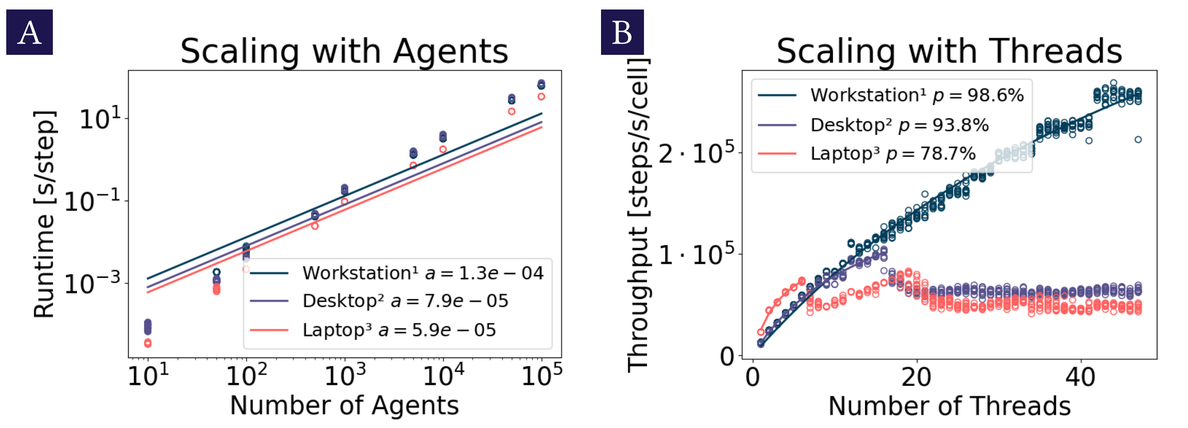

Figure 1: (A) Linear fit $f(x) = ax$ of scaling with increasing amounts of agents. (B) Amdahl's Law with up to $p = 98.6\%$ parallelized parts of the executed code resulting in a 21.5 times speedup.

[1]Workstation, AMD 3960X (24C/48T) @3.8GHz-4.5GHz, 64Gb DDR4 3200MT/s
[2]Desktop, AMD 3700x (8C/16T) @3.6GHz-4.4GHz, 32Gb DDR4 3200MT/s
[3]Laptop, Intel 12700H (6+8C/12+8T) 45W @3.5GHz-4.7GHz 32Gb DDR5 4800MT/s

Cellular Properties as Rust Traits

Abstract traits are used to define cellular interactions via force mechanics. Users implement traits and obtain full control over cellular behavior.

```rust
pub trait Interaction<Pos, Vel, Force, Inf = ()> {
  /// Get additional information of cellular properties (ie. for
  /// cell-specific interactions). For now, this can also be used
  /// to get the mass of the other cell-agent. In the future, we
  /// will probably provide a custom function for this.
  fn get_interaction_information(&self) -> Inf;

  /// Calculates the force (velocity-derivative) on the
  /// corresponding external position given external velocity.
  /// By providing velocities, we can calculate terms that are
  /// related to friction.
  fn calculate_force_between(
    &self,
    own_pos: &Pos,
    own_vel: &Vel,
    ext_pos: &Pos,
    ext_vel: &Vel,
    ext_info: &Inf,
  ) -> Option<Result<Force, CalcError>>;
}
```

Roadmap

- Stabilize user API
- Additional backends (GPUs, MPI)
- Multi-Scale
- Stochastic processes
- Restarting simulations
- Advanced error handling
- Support common export formats (such as `*.vtk` files)

Branching patterns of *Bacillus subtilis* in 2D & 3D

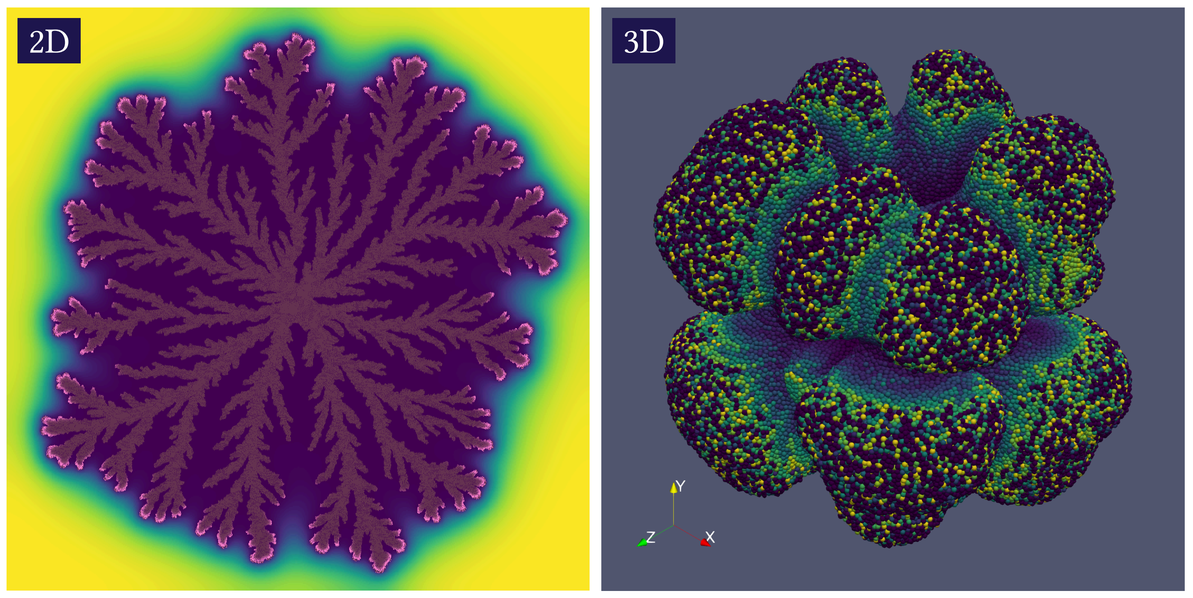

Figure 2: Spatio-Temporal patterns inspired by [1], [2]. Cells (~500,000) consume extracellular nutrients, grow, divide and self-organize into a branched pattern. Brighter colors indicate higher nutrient concentrations.

Cell Sorting in 3D

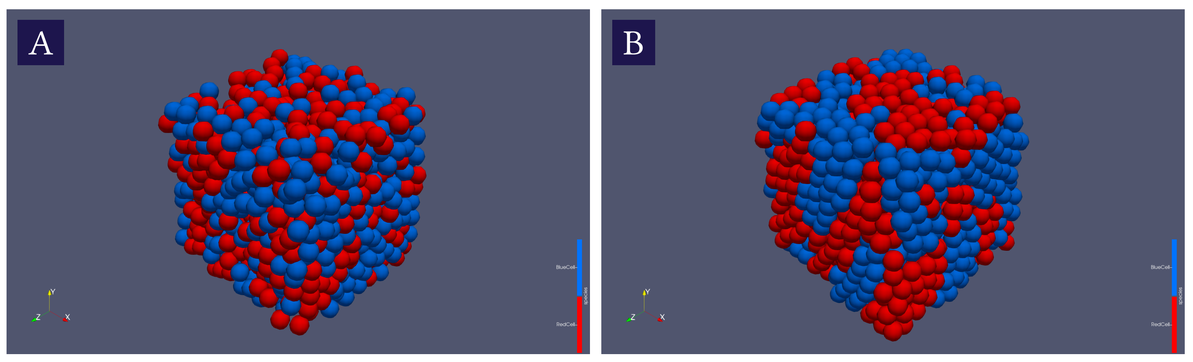

Figure 3: Cells with species-specific interactions. The initially randomized state (A) organizes itself and the two species get separated (B).

Semi-Vertex Models

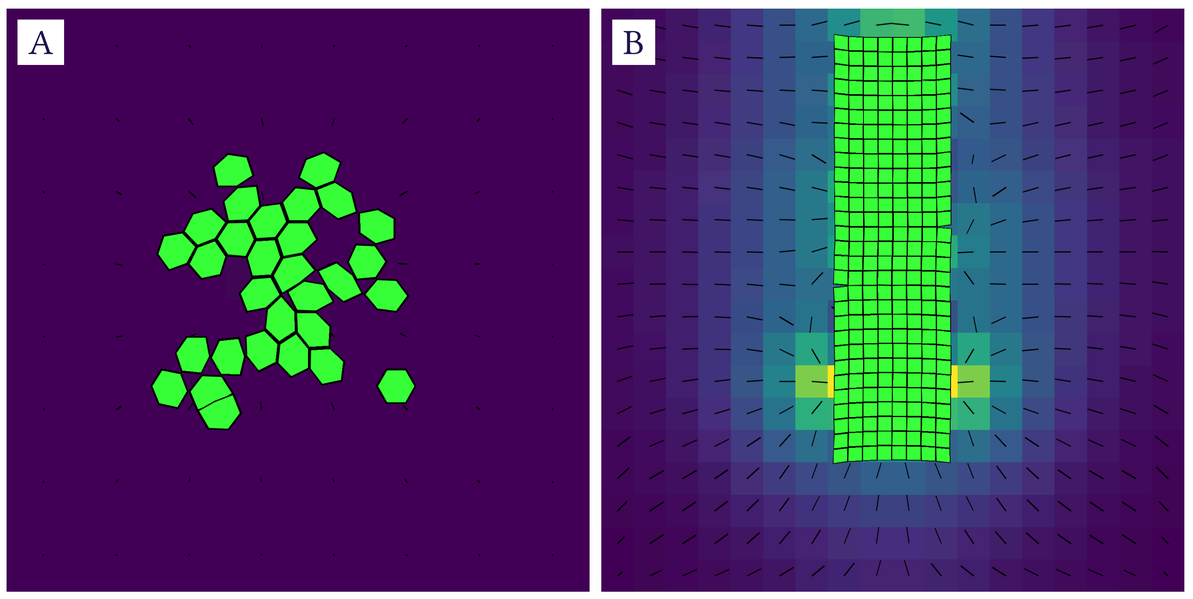

Figure 4: Freely motile semi-vertex models with (A) 6 and (b) 4 vertices. Vertices attract each other but will be repelled once inside another cell.

Sources

[1] K. Kawasaki, A. Mochizuki, M. Matsushita, T. Umeda, and N. Shigesada, "Modeling Spatio-Temporal Patterns Generated byBacillus Subtilis", Sep. 1997, doi: 10.1006/jtbi.1997.0462.

[2] M. Matsushita *et al.*, "Interface Growth and Pattern Formation in Bacterial Colonies", Jan. 1998, doi: 10.1016/S0378-4371(97)00511-6.