

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

Jonas Pleyer¹, Christian Fleck¹

¹Freiburg 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.

cellular_raza

- New Agent-Based Modeling Tool
- Extensive use of generics allows for unparalleled flexibility
- Written in Rust
- Parallelized (via OS-threads)
- Produces deterministic results
- Modularized
- No inherent assumptions about cells or domain
- User has complete control over every parameter and functionality
- Free software (GPLv2.0)

Rust Traits are Cellular Properties

This shows the abstract trait used to define cellular interactions via force mechanics. Users implement these traits to specify cellular behavior.

```
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>>;
}
```

Notice that even the types themselves such as position, force and velocity are generic, meaning this allows not only to switch between 2D and 3D but also to represent cells entirely differently as desired. The user is able and encouraged to modify and adjust these types as needed.

Cell Sorting in 3D

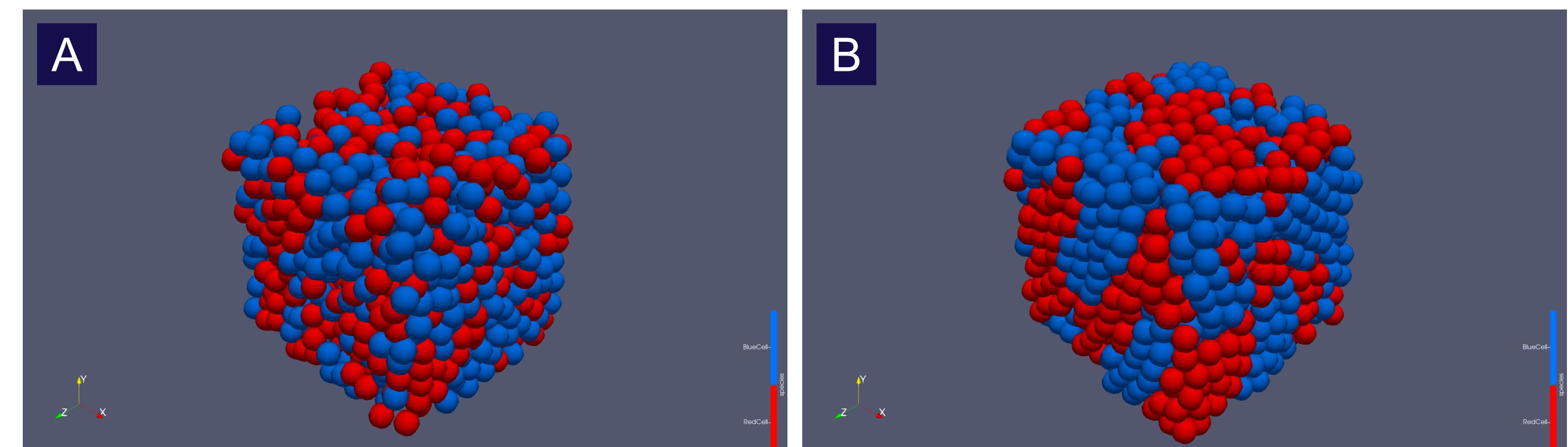


Figure 1: Cells with species-specific interactions. The initially randomized state (A) organizes itself and separates the two species (B).

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

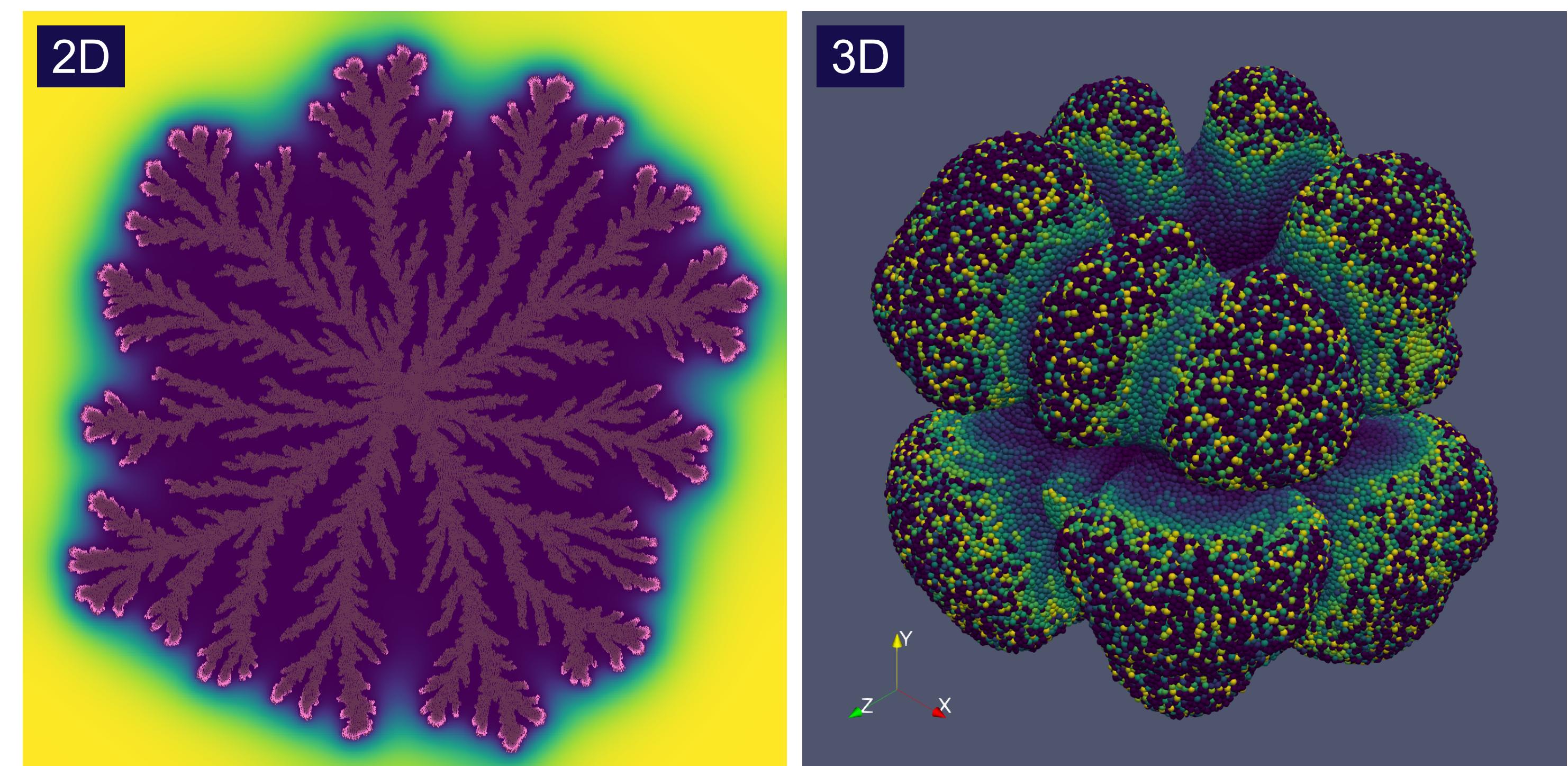


Figure 2: Spatio-Temporal patterns inspired by [1, 2]. Cells (~500,000) consume extracellular nutrients, grow, divide and self-organize into a branched pattern. Brighter colors show higher nutrient concentrations.

Roadmap

- Stabilize user API
- Develop additional backends
 - GPUs
 - MPI compatibility
- Multi-Scale in time for different cellular properties
- Optimize implementation of stochastic processes
- Restarting Simulations
- Advanced Error handling strategies and accuracy correction algorithms
- Support common export formats (such as *.vtk files)

Sources

- [1] K. Kawasaki, A. Mochizuki, M. Matsushita, T. Ueda, and N. Shigesada, "Modeling Spatio-Temporal Patterns Generated by *Bacillus subtilis*," *J. Theor. Biol.*, vol. 188, no. 2, pp. 177–185, Sep. 1997, doi: 10.1006/jtbi.1997.0462. Accessed: Apr. 26, 2023. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0022519397904628> (181 citations (Crossref) [2023-04-26])
- [2] M. Matsushita, J. Wakita, et al., "Interface growth and pattern formation in bacterial colonies," *Physica A: Statistical Mechanics Its Appl.*, vol. 249, no. 1-4, pp. 517–524, Jan. 1998, doi: 10.1016/S0378-4371(97)00511-6. Accessed: Apr. 7, 2023. [Online]. Available: <https://linkinghub.elsevier.com/retrieve/pii/S0378437197005116> (106 citations (Crossref) [2023-04-10])