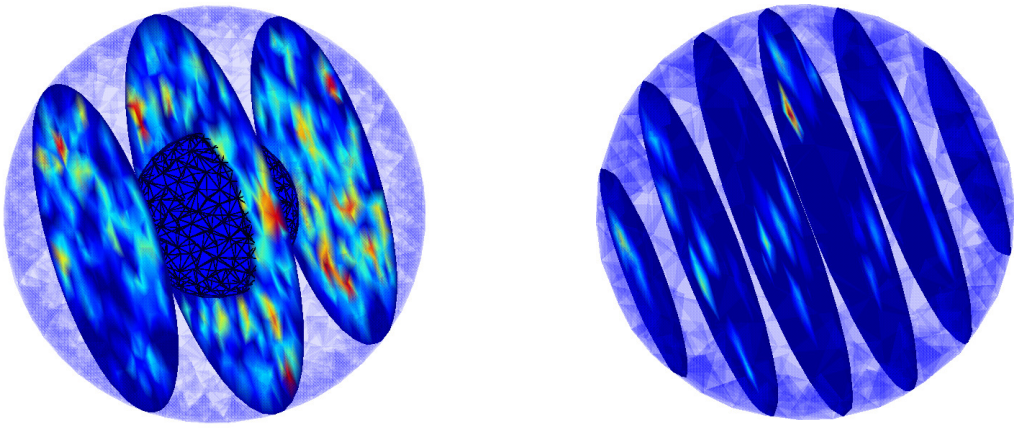
FLEXIBLE AND EFFICIENT SIMULATION OF STOCHASTIC REACTION-DIFFUSION NETWORKS

STEFAN ENGBLOM



Effective summary

This line of research is concerned with *novel and improved algorithms* as well as *new applications* for models of processes taking place inside living cells. The research is centered around the numerical software "URDME"[1], freely available under the GPL-license and with a growing community of users. In essence, URDME is a flexible simulator of chemical reaction networks in arbitrary geometries.

There are several openings for interesting and challenging projects within this software framework. Suggestions stated below include *high-performance software development*, *new models in molecular systems biology*, and improving the *simulation efficiency and flexibility*. Research of a more theoretical character include questions in *numerical stochastic analysis* as well as gaining a *better understanding of certain mathematical properties* of the modeling involved. Precise suggestions for projects suitable to courses at, say, the level of 5hp and all the way up to MSc/BSc-theses can be formulated upon request.

Interested candidates with a background in one or more of computer science, software engineering, computational physics, scientific computing, or molecular systems biology are more than welcome to contact me for further information.

[1] http://www.urdme.org

Background

According to classical chemistry, the concentrations of reacting substances obey rate-diffusion laws. Inside living cells, however, the number of molecules is often small and reactions between the molecules are more realistically described by a *random process*. One then pictures the molecules as diffusing more or less freely inside the cell by random motion and reactions happen whenever two molecules happen to meet.

A fruitful approach for simulating such *reaction-diffusion* models is to partition the biological cell into smaller sub-volumes. The molecules in a sub-volume can react with each other or move to an adjacent sub-volume according to certain probabilistic laws. A realization of the process is advanced in time by updating the number of molecules whenever such reaction- or diffusion events happen. A sample result is reproduced in the figure: the distribution of a species in a cell is shown to the left, and the nucleus is displayed to the right. Blue corresponds to a low number of molecules and red to a high number.

Project suggestions

The software URDME simulates such *event-based* stochastic models and relies on *Comsol Multiphysics* to generate input data. As a result of this setup, URDME is the only software capable of simulating these models in non-Cartesian geometries.

Your help is wanted to continue and improve this piece of software!

High-performance computing: The event-based algorithm is of general interest in computational physics. How can it best harness the power of modern multicore architectures?

Software flexibility: We like to add interfaces to *Python*, be able to make use of standard mesh-formats, and to display the results using free post-processing tools.

Modeling flexibility: There is ongoing work on the simulation of *microscopic models* and on *active transport*. How should these features best be incorporated?

Advanced modeling: In practical cases of interest, molecular movement is confined to a membrane or occurs in very crowded areas. How should this be modeled?

New applications: We are interested in including and experimenting with various reaction pathways inside living cells. Another type of models that would be interesting is pattern-formation mechanisms.

Algorithm efficiency: There exist several *approximate* algorithms. Ideally, one would like to rely on error-estimates which depend on the output wanted.

Mathematical properties: Dynamical systems in nature tend to remain close to critical points where the efficiency is the highest. This often makes it difficult to predict how model uncertainties propagate — a challenge which is highly relevant to take up when considering *inverse problems*.

URL: http://user.it.uu.se/~stefane
E-mail address: stefane@it.uu.se