Adaptive Parallel Simulation of a Two-Timescale-Model for Apoptotic Receptor-Clustering on GPUs

Schöll, Alexander; Braun, Claus; Daub, Markus; Schneider, Guido; Wunderlich, Hans-Joachim

Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM'14) Belfast, United Kingdom, 2-5 November 2014

doi: http://dx.doi.org/10.1109/BIBM.2014.6999195

Abstract: Computational biology contributes important solutions for major biological challenges. Unfortunately, most applications in computational biology are highly computeintensive and associated with extensive computing times. Biological problems of interest are often not treatable with traditional simulation models on conventional multi-core CPU systems. This interdisciplinary work introduces a new multi-timescale simulation model for apoptotic receptor-clustering and a new parallel evaluation algorithm that exploits the computational performance of heterogeneous CPU-GPU computing systems. For this purpose, the different dynamics involved in receptor-clustering are separated and simulated on two timescales. Additionally, the time step sizes are adaptively refined on each timescale independently. This new approach improves the simulation performance significantly and reduces computing times from months to hours for observation times of several seconds.

Preprint

Adaptive Parallel Simulation of a Two-Timescale Model for Apoptotic Receptor-Clustering on GPUs

Alexander Schöll*, Claus Braun*, Markus Daub†, Guido Schneider† and Hans-Joachim Wunderlich*

*Institute of Computer Architecture and Computer Engineering, University of Stuttgart,
Pfaffenwaldring 47, D-70569, Stuttgart, Germany
Email: {alexander.schoell, claus.braun, wu}@informatik.uni-stuttgart.de

†Institute of Analysis, Dynamics, and Modeling, University of Stuttgart,
Pfaffenwaldring 57, D-70569, Stuttgart, Germany
Email: {Markus.Daub, Guido.Schneider}@mathematik.uni-stuttgart.de

***Abstract*—Computational biology contributes important solutions for major biological challenges. Unfortunately, most applications in computational biology are highly compute-intensive and associated with extensive computing times. Biological problems of interest are often not treatable with traditional simulation models on conventional multi-core CPU systems.**
This interdisciplinary work introduces a new multi-timescale simulation model for apoptotic receptor-clustering and a new parallel evaluation algorithm that exploits the computational performance of heterogeneous CPU-GPU computing systems. For this purpose, the different dynamics involved in receptor-clustering are separated and simulated on two timescales. Additionally, the time step sizes are adaptively refined on each timescale independently.
This new approach improves the simulation performance significantly and reduces computing times from months to hours for observation times of several seconds.

***Keywords*-Heterogeneous computing, GPU computing, parallel particle simulation, multi-timescale model, adaptive Euler-Maruyama approximation, ligand-receptor aggregation**

I. Introduction

Within the last decades, *computational biology* evolved to a dynamic and very important research area. It provides indispensable tools that enable solutions for major biological challenges. Unfortunately, most applications in computational biology, such as complex Monte Carlo methods [1]–[8], are highly compute-intensive and associated with extensive computing times.

The simulation of *ligand-receptor aggregation* is a very good example for the application of a spatial stochastic Monte Carlo method in computational biology, which is associated with an extraordinary high computational effort. Different particle models with directional interactions between the involved particles have been proposed, for instance for ligand-receptor systems in [9] or for patchy particles in [10]. In [11], a model of a ligand-receptor system motivated by apoptotic receptor-clustering has been introduced, which considers both the deterministic motion caused by directed interactions between particles and the stochastic Brownian motion of particles. Based on [12], this two-component particle model has been further extended by a third particle type (*receptor homodimer*) in [13]. However, since the model is based on a stochastic process, a significant number of simulation runs with various particle configurations is required to draw reliable conclusions, which causes extensive computing times. Conventional computing systems with multi-core CPUs are no longer sufficient to cope with the complexity and computational demand of such sophisticated models.

Heterogeneous computing systems comprise of latency-optimized multi-core CPU architectures and dedicated accelerator architectures like throughput-optimized graphics processing units (GPU). They provide the required flexibility and performance to enable the practical use of complex mathematical simulation models in computational biology. However, the fundamentally different characteristics of the involved processor architectures pose major challenges for the design of such models, as well as the partitioning and mapping of their evaluation algorithms.

In this work, a new particle model and a parallel evaluation algorithm are introduced which utilize heterogeneous computing systems and in particular the parallel nature of graphics processing units. For the first time, this new approach allows to tackle the simulative investigation of ligand-receptor systems at relevant timescales with high performance.

II. Extrinsic Pro-Apoptotic Signaling Pathway

The extrinsic pro-apoptotic signaling pathway is initiated by signaling-competent ligand-receptor aggregates on the cell membrane. According to [12], the receptor under consideration is a molecule designed for highly efficient apoptosis induction, which consists of the extracellular domains of the TNF receptor type 1 (TNFR1) and the cytoplasmic part of the Fas receptor [14], a so-called TNFR1-Fas chimera. The membrane distal cysteine rich domain of TNFR1-Fas receptors enables their homodimerization [15]. The ligand under consideration is the soluble TNF which exists as a homotrimer being able to bind up to three TNFR1-Fas receptors. TNFR1-Fas receptors move randomly on the cell membrane, and the soluble form of TNF perform a diffusive

motion through the extracellular space.
In the following, we introduce a particle model describing the stochastic and deterministic translation and rotation of the molecules. Together with appropriate binding conditions, the simulation of the particle model then allows for the investigation of ligand-receptor aggregates comprising several receptors cross-linked by ligands.

III. Two-Timescale Particle Model of Receptor-Clustering

In this section, we present four significant extensions to the particle model originally introduced in [11] which has been extended by a third particle type in [13]. Thus, the particle model comprises TNFR1-Fas monomers, TNFR1-Fas dimers and TNF ligands, shortly denoted with monomers, dimers and ligands, respectively, see Figure 1. Besides the

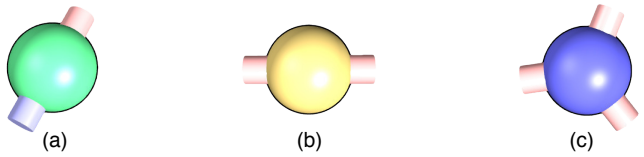

Figure 1: Different particle types - monomer (a), dimer (b) and ligand (c) - involved in the particle model. The trivalent ligand provides three indistinguishable binding sites for monomers or dimers, the bivalent dimer possesses two indistinguishable binding sites for ligands while the bivalent monomer has two binding sites - one for ligands and the other for the self-association with another monomer.

location of the particles, we also take the orientation of their binding sites into account and introduce equations of motion for both the translation and rotation of the particles. The numerical simulation of the three-component particle model especially the computation of the particles' interaction requires a very small step size of the temporal discretization of the model equations which causes an enormous computational cost. In order to achieve numerical simulations on biological relevant timescales comprising several minutes we establish an adaptive Euler-Maruyama time scheme for the two-timescale particle model. Starting point of our extensions is the system of stochastic differential equations introduced in [13].

- First, we assume that ligands primarily diffuse through the three-dimensional extracellular space and randomly impact on the cell membrane. For simplicity, we introduce an artificial two-dimensional simulation domain in close vicinity to the cell membrane, see Figure 2, instead of considering the complete three-dimensional extracellular space since we are solely interested in the ligand-receptor aggregation on the cell membrane. From this artificial simulation domain, a random number of ligands move towards the cell membrane where they interact with or bind to monomers or dimers.
- Secondly, we separate the particle dynamics into a pure diffusion of the particles and a deterministic motion caused by the interaction between the particles. Therefore, we split the set of particles into three

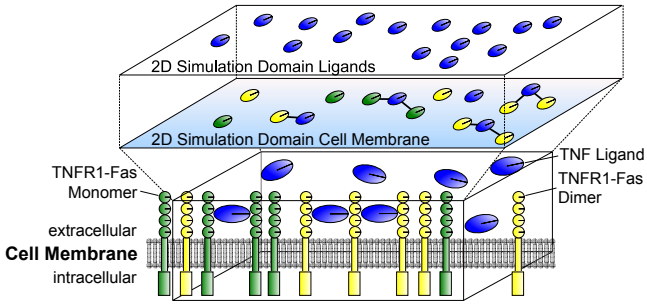

Figure 2: Sketch of the particle model with two simulation domains. The lower two-dimensional domain models the cell membrane while the upper domain displays the diffusive motion of the ligands in the three-dimensional extracellular space. Due to the diffusion, ligands can move from the upper simulation domain towards the lower one where they can associate with receptors and thereby form ligand-receptor aggregates.

classes: the free diffusing particles $\mathbb{M}_f$, $\mathbb{D}_f$, $\mathbb{L}_f$, the interacting particles $\mathbb{M}_i$, $\mathbb{D}_i$, $\mathbb{L}_{i,\text{temp}}$, and the bound particles $\mathbb{M}_b$, $\mathbb{D}_b$, $\mathbb{L}_b$, while the total number of ligands in the system remains constant. In this connection, particles are bound if the distance between them is smaller than a certain threshold value R_CUT and the orientation of their binding sites is appropriate. Particles are interacting if their distance is smaller than R_CUT but the orientation of the binding sites is unsuitable.
- Thirdly, we introduce different timescales for the diffusion and interaction of the particles. Since the interaction is strong and short-ranged, the simulation of the interaction process requires a very small timescale while the diffusion process can be simulated on a larger timescale.
- Fourthly, we assume that two bound receptor monomers form a dimer, i.e., the binding of two monomers is irreversible.

System of stochastic and ordinary differential equations

Following the first three extensions to simplify the particle model, we have a system of ordinary differential equations (ODEs) for the coordinates of interacting particles

$$\mathrm{d}\mathbf{x}_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l;\tau} = 6\mu^2_{\mathrm{M/D/L}}\mathbf{F}_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}(\boldsymbol{\xi}_{\mathbb{P}^t_i;\tau}, \boldsymbol{\varphi}_{\mathbb{P}^t_i;\tau})\mathrm{d}\tau, \quad (1)$$

$i \in \mathbb{M}^t_i$, $j \in \mathbb{D}^t_i$, $l \in \mathbb{L}^t_b \cup \mathbb{L}^t_{i,\text{temp}}$, where the matrix $\boldsymbol{\xi}_{\mathbb{P}^t_i;\tau}$ contains the coordinates of all interacting particles and $\boldsymbol{\varphi}_{\mathbb{P}^t_i;\tau}$ the angles describing the orientation of the binding sites, and a system of stochastic differential equations (SDEs)

$$\mathrm{d}\mathbf{x}_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l;t} = \sqrt{2}\mu_{\mathrm{M/D/L}}\mathrm{d}\widetilde{\mathbf{W}}_{\mathrm{trans},t,\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}, \quad (2)$$

$i \in \mathbb{M}^t_f$, $j \in \mathbb{D}^t_f$, $l \in \mathbb{L}^t_f$, for the coordinates of the free diffusing particles. For the particle rotation, we obtain for the bound particles a system of ODEs

$$\begin{aligned}\mathrm{d}\varphi_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l;\tau} =& \mu^2_{\mathrm{M/D/L}}\zeta^2_{\mathrm{M/D/L}}\times \\ & g_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}(\boldsymbol{\xi}_{\mathbb{P}^t_b;\tau}, \boldsymbol{\varphi}_{\mathbb{P}^t_b;\tau})\mathrm{d}\tau,\end{aligned} \quad (3)$$

$i \in \mathbb{M}^t_b$, $j \in \mathbb{D}^t_b$, $l \in \mathbb{L}^t_b$, where $\boldsymbol{\xi}_{\mathbb{P}^t_b;\tau}$ contains the coordinates of all bound particles and $\boldsymbol{\varphi}_{\mathbb{P}^t_b;\tau}$ the angles of the

corresponding particles. The particles which are not bound perform a random rotation described by a system of SDEs

$$\mathrm{d}\varphi_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l;t} = \sqrt{2}\mu_{\mathrm{M/D/L}}\zeta_{\mathrm{M/D/L}}\mathrm{d}\widetilde{W}_{\mathrm{rot},t,\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}, \quad (4)$$

$i \in \mathbb{M}_f^t \cup (\mathbb{M}_i^t \backslash \mathbb{M}_b^t)$, $j \in \mathbb{D}_f^t \cup (\mathbb{D}_i^t \backslash \mathbb{D}_b^t)$, $l \in \mathbb{L}_f^t$. In (2) and (4), the terms $\mathrm{d}\widetilde{\mathbf{W}}_{\mathrm{trans},t,\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}$ and $\mathrm{d}\widetilde{W}_{\mathrm{rot},t,\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}$ denote increments of Wiener processes modeling the Brownian motion of the particles. The interaction forces $\mathbf{F}_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}$ are given by a superposition of gradients of a truncated (12,6)-Lennard-Jones potential or the sole repulsive part of this potential dependent on the mutual orientation of the interacting particles. The interaction torsional moments $g_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}$ are defined by a cubic function with point symmetry to the origin and zeros at $\pm\pi$ in case of monomers, $\pm\pi/2$ in case of dimers, and $\pm\pi/3$ in case of ligands according to the number of indistinguishable binding sites, cf. [13] and Figure 1. Moreover, the parameters $\mu_{\mathrm{M/D/L}}$ and $\zeta_{\mathrm{M/D/L}}$ in (1)-(4) are given by

$$\mu_{\mathrm{M/D}}^2 = \frac{k_B T \bar{t}}{L^2 \beta_{\mathrm{M/D}}} \text{ with } \beta_{\mathrm{M/D}} = 6\pi\eta_{\mathrm{cm}} R_{\mathrm{M/D}}, \quad (5)$$

$$\mu_{\mathrm{L}}^2 = \frac{k_B T \bar{t}}{L^2 \beta_{\mathrm{L}}} \text{ with } \beta_{\mathrm{L}} = 6\pi\eta_{\mathrm{es}} R_{\mathrm{L}}, \quad (6)$$

$$\zeta_{\mathrm{M/D}}^2 = \frac{\beta_{\mathrm{M/D}} L^2}{\gamma_{\mathrm{rot,M/D}}} \text{ with } \gamma_{\mathrm{rot,M/D}} = 8\pi\eta_{\mathrm{cm}} R_{\mathrm{M/D}}^2 d_{\mathrm{cm}}, \quad (7)$$

$$\zeta_{\mathrm{L}}^2 = \frac{\beta_{\mathrm{L}} L^2}{\gamma_{\mathrm{rot,L}}} \text{ with } \gamma_{\mathrm{rot,L}} = 8\pi\eta_{\mathrm{es}} R_{\mathrm{L}}^3, \quad (8)$$

where L denotes the length scale and $\bar{t}$ the timescale for eliminating the dimensions of the magnitudes, η_{cm} the viscosity of the cell membrane, η_{es} the viscosity of the extracellular space, $R_{\mathrm{M/D/L}}$ the radius of the corresponding particles, d_{cm} the thickness of the cell membrane, and $k_B T$ the thermal energy of the system composed of the Boltzmann's constant k_B and the temperature T. Finally, the motion of the particles is completely described by the equations (1)-(4). In the following section, we shortly describe the Euler-Maruyama approximation for the introduced SDEs and the explicit Euler scheme for the ODEs.

Numerical Approximation of the ODEs and SDEs

In order to simulate the formation of ligand-receptor aggregates, the systems of stochastic and ordinary differential equations are solved numerically. For this purpose, we introduce the discretization $0 = t^0 < t^1 < t^2 < \cdots < t^n < t^{n+1} < \cdots < t^N = T$ of the large scaled time interval $[0, T]$ with an initially equidistant step size $\Delta t = t^{n+1} - t^n$. Since the interaction between the particles occurs on a smaller timescale, we introduce the discretization of each time interval $[t^n, t^{n+1}]$, namely $t^n = \tau^{n,0} < \tau^{n,1} < \cdots < \tau^{n,\eta} < \tau^{n,\eta+1} < \cdots < \tau^{n,H} = t^{n+1}$ with an initially equidistant step size $\Delta\tau^n = \tau^{n,\eta+1} - \tau^{n,\eta}$. Then, the approximation of the solution to the equations (1)-(4) at time t^n is summarized in the matrices $\boldsymbol{\xi}_{\mathbb{P}_f^n}^n$ and $\boldsymbol{\varphi}_{\mathbb{P}_f^n}^n$ on the large timescale, and $\boldsymbol{\xi}_{\mathbb{P}_i^n}^{n,\eta}$, $\boldsymbol{\xi}_{\mathbb{P}_b^n}^{n,\eta}$, $\boldsymbol{\varphi}_{\mathbb{P}_i^n}^{n,\eta}$, and $\boldsymbol{\varphi}_{\mathbb{P}_b^n}^{n,\eta}$ on the small timescale. In the latter terms, the sets $\mathbb{P}_{i/b}$ are though superscribed with an index n which indicates that the sets are updated on the large timescale. Finally, the explicit Euler method for the system of ODEs (1) reads

$$\Delta\mathbf{x}_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}^{n,\eta} = 6\mu_{\mathrm{M/D/L}}^2 \mathbf{F}_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}(\boldsymbol{\xi}_{\mathbb{P}_i^n}^{n,\eta}, \boldsymbol{\varphi}_{\mathbb{P}_i^n}^{n,\eta})\Delta\tau^n, \quad (9)$$

$i \in \mathbb{M}_i^n$, $j \in \mathbb{D}_i^n$, $l \in \mathbb{L}_b^n \cup \mathbb{L}_{i,\mathrm{temp}}^n$. The Euler-Maruyama approximation of the system of SDEs (2) is given by

$$\Delta\mathbf{x}_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}^n = \sqrt{2}\mu_{\mathrm{M/D/L}}\Delta\widetilde{\mathbf{W}}_{\mathrm{trans},n,\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}, \quad (10)$$

$i \in \mathbb{M}_f^n$, $j \in \mathbb{D}_f^n$, $l \in \mathbb{L}_f^n$, where the increments

$$\begin{aligned}&[\Delta\widetilde{\mathbf{W}}_{\mathrm{trans},n,\cdot}]_{dim=1,2} \\ &= [\widetilde{\mathbf{W}}_{\mathrm{trans},n+1,\cdot}]_{dim=1,2} - [\widetilde{\mathbf{W}}_{\mathrm{trans},n,\cdot}]_{dim=1,2}\end{aligned} \quad (11)$$

are $\mathcal{N}(0, \Delta t)$ distributed random variables and can be written as $[\Delta\widetilde{\mathbf{W}}_{\mathrm{trans},n,\cdot}]_{dim=1,2} = [\mathbf{Z}_{\mathrm{trans,n},\cdot}]_{dim=1,2} \cdot \sqrt{\Delta t}$ with $[\mathbf{Z}_{\mathrm{trans,n},\cdot}]_{dim=1,2} \sim \mathcal{N}(0,1)$. The index dim indicates the two components of the two-dimensional Wiener process $\widetilde{\mathbf{W}}_{\mathrm{trans}}$. For the approximation of (3), we obtain

$$\begin{aligned}\Delta\varphi_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}^{n,\eta} =& \mu_{\mathrm{M/D/L}}^2 \zeta_{\mathrm{M/D/L}}^2 \times \\ & g_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}(\boldsymbol{\xi}_{\mathbb{P}_b^n}^{n,\eta}, \boldsymbol{\varphi}_{\mathbb{P}_b^n}^{n,\eta})\Delta\tau^n,\end{aligned} \quad (12)$$

$i \in \mathbb{M}_b^n$, $j \in \mathbb{D}_b^n$, $l \in \mathbb{L}_b^n$, and the approximating equations of (4) are given by

$$\begin{aligned}\Delta\varphi_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}^n =& \sqrt{2}\mu_{\mathrm{M/D/L}}\zeta_{\mathrm{M/D/L}} \times \\ & \Delta\widetilde{W}_{\mathrm{rot},n,\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l},\end{aligned} \quad (13)$$

$i \in \mathbb{M}_f^n \cup (\mathbb{M}_i^n \backslash \mathbb{M}_b^n)$, $j \in \mathbb{D}_f^n \cup (\mathbb{D}_i^n \backslash \mathbb{D}_b^n)$, $l \in \mathbb{L}_f^n$. Again,

$$\Delta\widetilde{W}_{\mathrm{rot},n,\cdot} = \widetilde{W}_{\mathrm{rot},n+1,\cdot} - \widetilde{W}_{\mathrm{rot},n,\cdot} \sim \mathcal{N}(0, \Delta t) \quad (14)$$

can be written as $\Delta\widetilde{W}_{\mathrm{rot},n,\cdot} = Z_{\mathrm{rot},n,\cdot} \cdot \sqrt{\Delta t}$. Evaluating the right-hand side of equations (9), (10), (12), and (13) yields the variations in the coordinates $\Delta\mathbf{x}_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}$ and the angles $\Delta\varphi_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}$ on the different timescales. Starting with $\mathbf{x}_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}^0$, $\mathbf{x}_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}^{0,0}$, $\varphi_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}^0$, and $\varphi_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}^{0,0}$ for the initially free and interacting particles, respectively, we iteratively obtain the coordinates $\mathbf{x}_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}^{n+1}$ and the angles $\varphi_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}^{n+1}$, $n = 0, 1, 2, \ldots$, on the large timescale, and the coordinates $\mathbf{x}_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}^{n,\eta+1}$ and the angles $\varphi_{\mathrm{M}_i/\mathrm{D}_j/\mathrm{L}_l}^{n,\eta+1}$, $n = 0, 1, 2, \ldots, \eta = 0, 1, 2, \ldots$, on the small timescale.

A. Adaptive Euler-Maruyama Scheme

In unfavorable cases, particles may diffuse in a way such that the particles almost lie upon each other or the interacting particles run through each other if the time step of the discretization is too large. To avoid these cases we refine the time steps of the discretization adaptively. For this purpose, we choose different criteria for the refinement of the time steps of the diffusion and the interaction. In the following, we give the main ideas of the adaptive Euler-Maruyama scheme, originally introduced in [16] and now adapted to our two-timescale particle model comprising ordinary and

stochastic differential equations which are merged on the large timescale. For a mathematically rigorous description, we refer the interested reader to [16].

i) Adaptive time scheme for interacting particles.
In order to avoid the crossing of two particles due to their interaction for a given step size $\Delta\tau^n$, we refine the step size adaptively. For this purpose, we compute virtual coordinates of the particles for step sizes $\Delta_{k_i}\tau^n := 2^{-k_i}(1-\Sigma_i)\Delta\tau^n, k_i = 0, 1, 2, \ldots$, such that two conditions are fulfilled, where the term $(1-\Sigma_i)$ denotes the part of the time interval $[\tau^{n,\eta}, \tau^{n,\eta+1}]$ which is not yet simulated:
   I) First, the distance between the particles for the new virtual coordinates must be larger than a certain threshold R_CUT_MIN, hence, $|\mathbf{x}_{\mathrm{P}_i}^{n,\eta+\Delta_{k_i}\tau^n} - \mathbf{x}_{\mathrm{P}_j}^{n,\eta+\Delta_{k_i}\tau^n}| > \mathrm{R\_CUT\_MIN}$, $\mathrm{P}_i, \mathrm{P}_j \in \mathbb{P}_i^n$.
   II) Secondly, the distance between the new virtual coordinates of one particle and its original coordinates must be smaller than the distance between the new coordinates of the other interacting particle and the original coordinates of the former particle, shortly written as $|\mathbf{x}_{\mathrm{P}_i}^{n,\eta+\Delta_{k_i}\tau^n} - \mathbf{x}_{\mathrm{P}_i}^{n,\eta}| < |\mathbf{x}_{\mathrm{P}_j}^{n,\eta+\Delta_{k_i}\tau^n} - \mathbf{x}_{\mathrm{P}_i}^{n,\eta}|$, $\mathrm{P}_i, \mathrm{P}_j \in \mathbb{P}_i^n$.

   Finally, k_i is chosen as the smallest value such that both criteria are fulfilled for all interacting particles.

ii) Adaptive time scheme for diffusing particles.
Considering the diffusing particles, we similarly formulate two criteria for the adaptive refinement of the time step. Again, we compute virtual coordinates according to the discretized SDEs for the particle coordinates with the step sizes $\Delta_{k_f}t := 2^{-k_f}(1-\Sigma_f)\Delta t, k_f = 0, 1, 2, \ldots$, such that the following two conditions are fulfilled, where $(1-\Sigma_f)$ is the part of the interval $[t^n, t^{n+1}]$ which is not yet simulated:
   I) First, the distance between the diffusing particles after the diffusion step must be larger than a certain threshold R_CUT_MIN, hence, $|\mathbf{x}_{\mathrm{P}_i}^{n+\Delta_{k_f}t} - \mathbf{x}_{\mathrm{P}_j}^{n+\Delta_{k_f}t}| > \mathrm{R\_CUT\_MIN}$, $\mathrm{P}_i, \mathrm{P}_j \in \mathbb{P}_f^n$.
   II) Secondly, the distance between the new virtual coordinates of a diffusing particle and an interacting particle after the performed interaction time steps must be larger than the threshold R_CUT_MIN, i.e. $|\mathbf{x}_{\mathrm{P}_i}^{n+\Delta_{k_f}t} - \mathbf{x}_{\mathrm{P}_j}^{n,H}| > \mathrm{R\_CUT\_MIN}$, $\mathrm{P}_i \in \mathbb{P}_f^n$, $\mathrm{P}_j \in \mathbb{P}_i^n$.

In summary, we introduced in this section four extensions of the particle model, and additionally established an adaptive Euler-Maruyama time scheme for the two-timescale particle model.

IV. Parallel Model Evaluation Algorithm

A. Heterogeneous CPU-GPU Computing Systems

Heterogeneous computing systems comprising of multi-core CPU architectures and many-core GPU architectures deliver tremendous performance. However, they also require careful partitioning and mapping of algorithms. Latency-optimized CPUs are well-suited for performing small numbers of individual and independent tasks as fast as possible. Throughput-optimized GPUs excel on data-parallel workloads where a single stream of instructions is applied in parallel to a very large number of different data elements [17], [18]. Such GPU architectures exhibit rich memory systems with large register sets, shared memories and different levels of caches. All of these memories are associated with different capacities and access latencies which force a skillful management by the software developer. In addition, program execution on GPUs is often not as flexible as on CPUs since high performance can only be achieved if large numbers of threads follow the same control flow. This makes branches and thread divergence expensive and poses a challenge to the developer.

The parallel evaluation algorithm described in the following section exploits the special characteristics of heterogeneous computing systems. It executes setup, control and analysis steps on the CPU, while it utilizes the data-parallel GPU for the computation of particle interactions. The interplay of both architectures leads to substantial performance improvements as will be shown in Section V.

B. Parallel Evaluation Algorithm

The developed mathematical model has been mapped to a grid-based, stochastic particle simulation, tailored to modern heterogeneous computing systems. A simulation comprises three different kinds of particles (monomers, dimers, ligands) in different numbers, which reside in separate lists. Each particle is assigned to one of the three particle classes (free diffusing, interacting and bound), which is indicated with a flag. The spatial simulation domain is organized in grids. Each grid is composed of equally-sized grid cells.

The computation of forces, torsional moments and positions of particles is independent and can be computed in parallel. Each particle is therefore processed by its own thread. Since a biological relevant simulation comprises thousands of particles, these threads are executed on the GPU.

A large number of simulations has to be performed with different realizations in order to draw reliable statistical conclusions. Since modern GPUs comprise a tremendous number of simple arithmetic processing units which exceed the number of particles in a single simulation, multiple different simulation instances are simulated on a single GPU device in parallel. The execution of each simulation instance is controlled by parallel threads on the multi-core CPU. Therefore, it is ensured that diverging execution flows of the parallel executed simulation instances are not serialized on the GPU.

Simulation Algorithm Overview

The simulation algorithm, which is shown in Figure 3, comprises eight steps which are executed on the CPU and the GPU in parallel. In the first step, the setup of the simulation environment is executed on the CPU. Here, the available GPU devices are allocated and initialized, and

simulation parameters are transferred to the GPU. For each simulation instance, a CPU thread is created which controls the execution of the main simulation loop on the GPU.

In the second step, the random impact of ligands on the cell membrane is simulated on the GPU. To determine the subset $\mathbb{L}^n_{i,\text{temp}}$ of the available ligand particles in parallel, $|\mathbb{L}^n_{i,\text{temp}}|$ threads generate uniformly distributed random values in the range of the ligand indexes $[0, \mathbb{L}^n_f[$. Afterwards, each thread puts one chosen ligand particle on the cell membrane in parallel.

The evaluations for free diffusing particles are performed in the third step (*Adaptive diffusion*). The evaluations for interacting and bound particles are performed in the fourth step (*Adaptive interaction*). The diffusion and interaction steps are evaluated adaptively with respect to the timescales of the different particle classes. The progress on the different timescales is continuously synchronized at the time points t^n. During the evaluations in the third and fourth step, each thread iteratively traverses the particles of the current grid cell and in special cases, the particles from neighboring cells. In these steps, spatial coherence within the memory hierarchy of the GPU is exploited to improve the overall performance. For this purpose, the particle data of each grid cell is explicitly shared among the threads which perform evaluations in a common cell. Due to the utilization of fast shared memory for the particle data of each grid cell, the number of slow memory accesses to the off-chip GPU memory is significantly reduced.

In the fifth step, the randomly impacted ligands are evaluated. Ligands which are then neither interacting nor bound to surrounding particles leave the cell membrane and are therefore assigned to the free diffusing particle class.

In the sixth step, pairs of monomers which are appropriately located and oriented to each other to form aggregates are deleted from the particle lists. The resulting dimers are located at the center between monomers with respect to their positions and orientations. The emerging particle data for the resulting dimers is put in the respective particle list. Therefore, the capacity of the particle list which contains the dimers has to be increased during the execution on the GPU. Dynamic memory management generally has to be performed in sequential steps to avoid race conditions. For example, race conditions may occur if multiple parallel threads increase the particle list at the same time and overwrite the recently added particle data mutually. Therefore, a large number of simultaneously emerging dimers decreases performance significantly if memory management is performed in sequential steps. To allow a concurrently performed dynamic memory management on the GPU device, the particle lists are extended by *dummy* particles which are inactive during simulation. If an additional dimer is required, a dummy particle is activated to represent the emerging dimer particle. For each monomer, one specific dummy particle in the dimer particle list is available. In this way, it is ensured, that pairs of monomers can be merged to dimers concurrently while avoiding race conditions and performance overhead due to serialized memory management. The main simulation loop is repeated until the total simulation time has been reached. The dashed boxes *sampling* and *aggregate detection* in Figure 3 are additional steps, which are performed at user-defined intervals. During a sampling step, the location and orientation of the particles is transferred to the CPU. While the main simulation loop is continued on the GPU, additional threads are created on the CPU to perform the aggregate detection in parallel. In this step, the particle information is evaluated to find potential ligand-receptor aggregates and their size. The results of the sampling and aggregate detection steps are then stored.

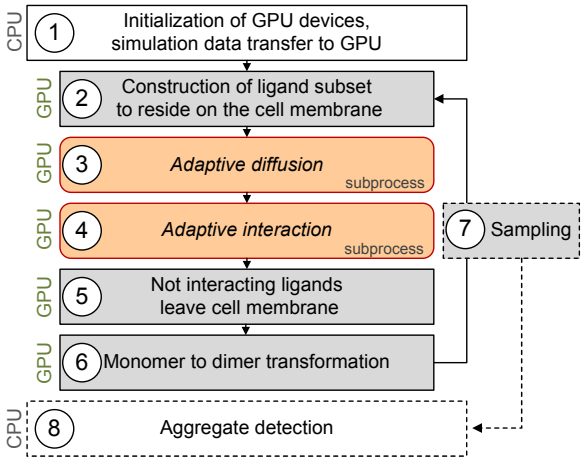

Figure 3: Overview of the model evaluation algorithm with the main simulation loop.

Grid-Based Mapping

In [13], we presented mapping strategies for grid-based particle simulations to achieve high performance on GPUs. Now, the adaptive refinement scheme induces additional requirements, which impels the development of further optimization strategies. As explained before, the spatial simulation domain is subdivided into equally-sized grid cells. According to the model, the interactions between particles decrease with increasing distance. Beyond the appropriately chosen distance threshold value R_CUT, interactions are numerically neglectable. In [13], we showed that the size of the grid cells has a direct influence on the achievable simulation performance. The number of sorting steps can be reduced by choosing large grid cells, since particles stay longer within the same cell. Too large grid cells induce a computational overhead, because the number of unnecessarily interacting particles per cell increases. Small grid cells reduce the number of unnecessary interactions but induce additional sorting overhead. Compared to equidistant time steps, the adaptive refinement scheme leverages particle translations on larger time steps. Large translations increase the tendency of particles to leave grid cells, which results in additional grid update steps. To exploit the achievable simulation performance, the size of the grid cells must be increased. Larger grid cells increase the number of particles

which do not interact with particles from neighboring cells. The reason is that their distance to the grid cell borders is greater than R_CUT. Additionally, these particles will only interact with particles in neighboring grid cells, which are also located in their respective R_CUT border region (see Figure 4).

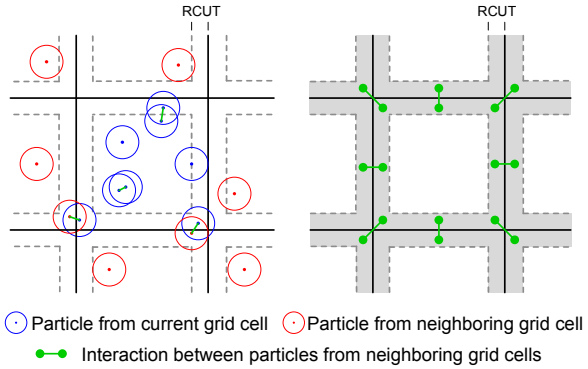

Figure 4: Scenarios for different particle interactions within and between cells.

Without further optimization, the algorithm would have to access the complete particle data from neighboring grid cells, even if there are no possible interactions. Since memory bandwidth is a scarce resource, the overall performance is decreased significantly. To ensure that only particle data of the neighboring R_CUT border region is accessed, the grid cells are subdivided. Therefore, each cell contains two areas, which comprise the R_CUT border region and the center of the grid cell. After the following sorting step, the particles which are located in the R_CUT border region are therefore aligned with each other (see Figure 5).

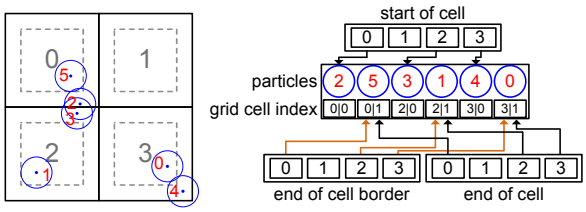

Figure 5: Extended grid cell and particle list organization using border regions.

The resulting alignment allows the algorithm to identify the portion of particles which are able to interact with particles from neighboring grid cells without accessing the particle data from neighboring cells in every single evaluation step. This reduces the overall number of memory accesses, since only particle data from neighboring cells are accessed which are actually required for evaluation.

Adaptive Evaluation Algorithm

Figure 6 shows the steps of the algorithm that generates refined time step sizes to find particle translations on the GPU that do not violate the rules formulated in Section III-A. This algorithm is executed in the third and fourth step of the main simulation loop (Figure 3). The algorithm additionally ensures that every adaptively refined timescale is synchronized at the time points t^n. Therefore, the algorithm continues to evaluate the particle classes using refined time steps until the time point t^n is reached. In the following, k represents the parameter k_i and k_f (see Section III-A) and describes the level of refinement. The simulation time step is calculated from k, which results in a fraction of Δt and $\Delta\tau^n$, respectively. In the first step, the refinement parameter k is initialized to zero, which induces no refinement. In the second step, a set of normally distributed random numbers is generated in parallel for each particle.

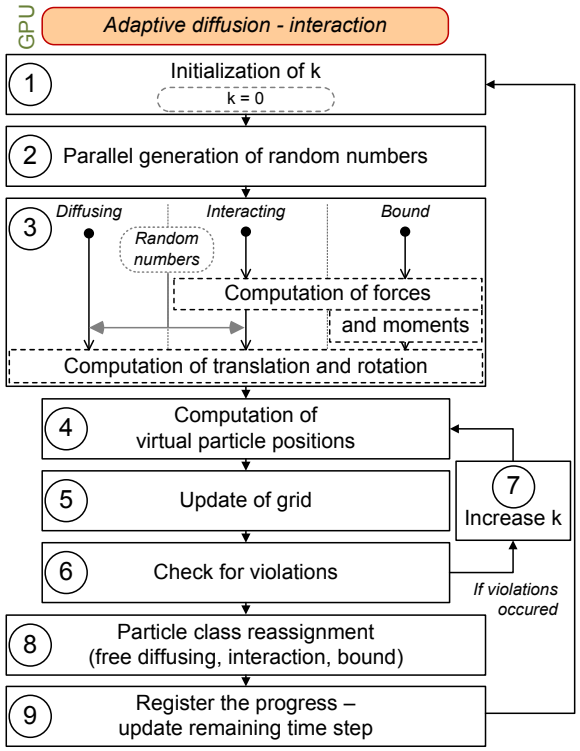

Figure 6: Overview of the algorithm, which chooses optimistic time steps and refines them adaptively.

In the third step, the evaluations for the different particle classes are performed. For free diffusing particles, translations of position and rotations are computed using the generated random numbers. For interacting and bound particles, the forces between all pairs of particles are determined to calculate the translations. Additionally, the rotation of the particles is evaluated: The rotation of interacting particles is calculated by using the generated random numbers whereas the rotation of bound particles is derived from the torsional moments which are calculated between all pairs of particles. The adaption loop is entered in the fourth step. The virtual coordinates and orientations are calculated using the respective translations and rotations. Here, the translations and rotations are refined with regard to the current level of refinement k and the remaining part of the time step Δt and $\Delta\tau^n$, respectively.

In the fifth step, all particles are reassigned to the grid cells, according to their virtual coordinates in the simulation domain. The resulting virtual coordinates are checked for violations in the sixth step, as described in Section III-A. If violations occurred, then the level of refinement k is increased in the seventh step.
With every iteration of the adaption loop, the translation and rotation of the particles are refined until all violations are resolved. In the eighth step, all particles are reassigned to their corresponding particle classes. The progress with respect to the common time step Δt and $\Delta \tau^n$, respectively, is registered in the ninth step. The algorithm is repeated, until the complete time step Δt and $\Delta \tau^n$, respectively, has been fully evaluated.

V. Experimental Results

The implementation of the adaptive multi-timescale evaluation algorithm has been evaluated with respect to computational performance and the biological indications of the apoptotic receptor-clustering. The hardware platform consists of an Intel X5680 with 3.33 GHz and 64 GB RAM. The system hosts four Nvidia Tesla C2070 GPUs with 448 processing cores at 1.15GHz and 6 GB GDDR5 RAM per device. The machine runs a linux operating system with CUDA version 5.0 and a GNU GCC 4.6.2 compiler tool chain.

A. Evaluation of Computational Performance

For the evaluation of computational performance, suitable simulation setups were chosen. The simulation runs were then validated against the previously published results [13] with respect to computation time.
In order to evaluate the influence of refinements on computation times, multiple simulation runs with different numbers of particles were executed. To ensure comparability, each of these simulation runs consisted of 1 ms simulation time. Table I shows the computation times compared to our previous results.

Total particles	4608	9216	18432	36864	64512
Prev. work (s)	1105	1824	3250	6847	13638
This work (s)	1	9	50	598	2437
Avg. ref.	1.93	10.56	64.0	445.72	1024.0

Table I: Computation time and average refinement over different numbers of particles.

The results show that increasing numbers of particles lead to increasing average refinement levels (*avg. ref.*) and to longer computation times. This can be explained by the fact that an increasing number of particles leads to a higher probability that particles almost lie upon each other or interacting particles run through each other. The utilization of particle classes reduces the number of evaluations, since free diffusing particles are neglected during expensive computations for interacting particles.

As discussed before, a significant number of simulation runs is required to draw reliable conclusions, since the model is based on stochastic differential equations. A typical simulation setup of biological relevance comprising 2496 monomers, 2496 dimers and 1344 ligands has been chosen. Up to 8 parallel simulation instances per device were performed. The execution of the simulation instances was equally distributed to 4 GPU devices. Table II shows the achieved reduction in computation time for simulation runs over 1 ms simulation time.

Par. Instances	4	8	16	24	32
Prev. work (s)	2082	3045	5203	7591	9778
This work (s)	12	17	30	43	57
Avg. ref.	8.11	7.41	7.67	7.52	8.22

Table II: Computation time and average refinement with different numbers of parallel instances on multiple GPUs.

The results show a significant reduction of computation time by an approximate factor of 175 compared to the previously published results [13]. The adaptive timescale approach reduces the computational effort in every simulation instance, since it chooses the largest time step sizes which do not result in violations. During the simulations, the average refinement ranges between 7.41 and 8.22, which corresponds to dividing almost every time step size into 8 steps.

B. Evaluation of Biological Indication

The results in Section V-A indicate that the extensions of the particle model and the tailoring of the algorithms to the GPU allow for the simulation of ligand-receptor aggregation on biological relevant timescales. In the following, we consider a particle configuration with 2496 monomers and 2496 dimers initially uniformly distributed on the cell membrane and a constant total number of 1344 ligands initially diffusing through the extracellular space. In each time step on the diffusion timescale, an $\mathcal{N}(192, 64)$ distributed random number of ligands impact on the cell membrane.
The simulation of $T = 10\,\mathrm{s}$ with a maximal step size of the diffusion timescale $\Delta t = 10^{-6}$ and a maximal step size of the interaction timescale $\Delta \tau^n = 10^{-7}$ took about 32 hours. Figure 7 shows the evolution of ligand-receptor aggregates on a section of the simulation time corresponding to $0.5\,\mathrm{s}$. We observe that the temporal evolution of ligand-receptor aggregates occurs on a very short timescale of less than $0.5\,\mathrm{s}$, and afterwards, the number of ligand-receptor aggregates remains constant. This simulation was purely for testing and shows the expected behavior.
The evolution of ligand-receptor aggregates significantly depends on the total amount of initially diffusing ligands and the mean values of ligands that impact on the cell membrane. Additionally, the influence of the amount of receptor monomers and dimers on the formation of ligand-receptor aggregates has to be studied. In particular, the occurrence of large ligand-receptor aggregates and their most stable structure are interesting points to investigate. However, a

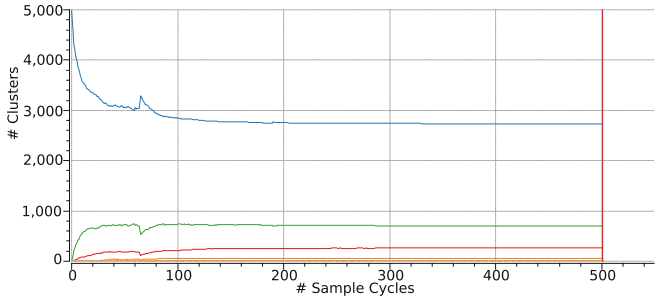

Figure 7: Evolution of ligand-receptor aggregates on the cell membrane: single particles (blue graph), aggregates of size two (green graph), aggregates of size three (red graph), and aggregates of size four (orange graph).

systematic study of the model for varying ligand numbers on biological relevant timescales, which is now possible due to the introduced model extensions and optimizations of the algorithms, is subject of current research and beyond the scope of this work.

VI. Conclusion

In this paper, the outcomes of an interdisciplinary collaboration for the extensive parallel simulation of apoptotic receptor-clustering have been presented. Timescales of biological relevance motivated the extension of a previously introduced model and the development of a new evaluation algorithm, tailored to heterogeneous CPU-GPU computing systems, in order to reduce the computational effort significantly. For this purpose, the particle dynamics were separated into a pure diffusion and the particles' interaction, allowing the simulation on different timescales. Moreover, an adaptive refinement of the time steps was independently implemented for both processes. The three-dimensional diffusion of the ligands in the extracellular space was represented by a second simulation domain while only a random subset of the ligands interact with or bind to the receptors on the cell membrane. Along with several algorithmic optimizations, the model extensions lead to significant performance enhancement, i.e., compared to our previous approach, the simulation of several seconds is now possible in a few hours instead of several months.

VII. Acknowledgment

The authors would like to thank the German Research Foundation (DFG) for financial support of their projects within the Cluster of Excellence in Simulation Technology (EXC 310/1) at the University of Stuttgart.

References

[1] C. Guo and H. Levine, "A Thermodynamic Model for Receptor Clustering", *Biophysical Journal*, vol. 77, no. 5, pp. 2358–2365, 1999.

[2] Y. Shi, "Clustering and Signalling of Cell Receptors", *Physica A: Statistical Mechanics and its Applications*, vol. 311, no. 1-2, pp. 199–212, 2002.

[3] J. R. Stiles, T. M. Bartol *et al.*, "Monte Carlo Methods for Simulating Realistic Synaptic Microphysiology Using MCell", in *Computational Neuroscience: Realistic Modeling for Experimentalists*. CRC Press, Boca Raton, FL, 2001, pp. 87–127.

[4] R. A. Kerr *et al.*, "Fast Monte Carlo Simulation Methods for Biological Reaction-Diffusion Systems in Solution and on Surfaces", *SIAM Journal on Scientific Computing*, vol. 30, no. 6, pp. 3126–3149, 2008.

[5] K. Mayawala, D. G. Vlachos, and J. S. Edwards, "Spatial Modeling of Dimerization Reaction Dynamics in the Plasma Membrane: Monte Carlo vs. Continuum Differential Equations", *Biophysical Chemistry*, vol. 121, no. 3, pp. 194–208, 2006.

[6] K. Radhakrishnan *et al.*, "Mathematical Simulation of Membrane Protein Clustering for Efficient Signal Transduction", *Annals of Biomedical Engineering*, vol. 40, no. 11, pp. 2307–2318, 2012.

[7] M. I. Monine *et al.*, "Modeling Multivalent Ligand-Receptor Interactions with Steric Constraints on Configurations of Cell-Surface Receptor Aggregates", *Biophysical Journal*, vol. 98, no. 1, pp. 48–56, 2010.

[8] C. Braun *et al.*, "Acceleration of Monte-Carlo Molecular Simulations on Hybrid Computing Architectures", in *30th IEEE International Conference on Computer Design (ICCD'12)*, Sept 2012, pp. 207–212.

[9] T. Ruiz-Herrero *et al.*, "A Tunable Coarse-Grained Model for Ligand-Receptor Interaction", *PLoS Computational Biology*, vol. 9, no. 11, p. e1003274, 2013.

[10] A. W. Wilber *et al.*, "Reversible Self-Assembly of Patchy Particles into Monodisperse Icosahedral Clusters", *The Journal of Chemical Physics*, vol. 127, no. 8, p. 085106, 2007.

[11] M. Falk *et al.*, "Modeling and Visualization of Receptor Clustering on the Cellular Membrane", in *2011 IEEE Symposium on Biological Data Visualization (BioVis)*, Providence, RI, October 2011, pp. 9–15.

[12] V. Boschert *et al.*, "Single Chain TNF Derivatives with Individually Mutated Receptor Binding Sites Reveal Differential Stoichiometry of Ligand Receptor Complex Formation for TNFR1 and TNFR2", *Cellular Signalling*, vol. 22, no. 7, pp. 1088–1096, 2010.

[13] C. Braun *et al.*, "Parallel Simulation of Apoptotic Receptor-Clustering on GPGPU Many-Core Architectures", in *Proceedings of the IEEE International Conference on Bioinformatics and Biomedicine (BIBM'12)*, Philadelphia, PA, October 2012, pp. 1–6.

[14] A. Krippner-Heidenreich *et al.*, "Control of Receptor-Induced Signaling Complex Formation by the Kinetics of Ligand/Receptor Interaction", *Journal of Biological Chemistry*, vol. 277, no. 46, pp. 44 155–44 163, 2002.

[15] M. Branschädel *et al.*, "Dual Function of Cysteine Rich Domain (CRD) 1 of TNF Receptor Type 1: Conformational Stabilization of CRD2 and Control of Receptor Responsiveness", *Cellular Signalling*, vol. 22, no. 3, pp. 404–414, 2010.

[16] H. Lamba, J. C. Mattingly, and A. M. Stuart, "An Adaptive Euler–Maruyama Scheme for SDEs: Convergence and Stability", *IMA Journal of Numerical Analysis*, vol. 27, no. 3, pp. 479–506, 2007.

[17] "General-Purpose Computation on Graphics Hardware, http://gpgpu.org."

[18] J. Nickolls and W. Dally, "The GPU Computing Era", *IEEE Micro*, vol. 30, no. 2, pp. 56–69, march-april 2010.