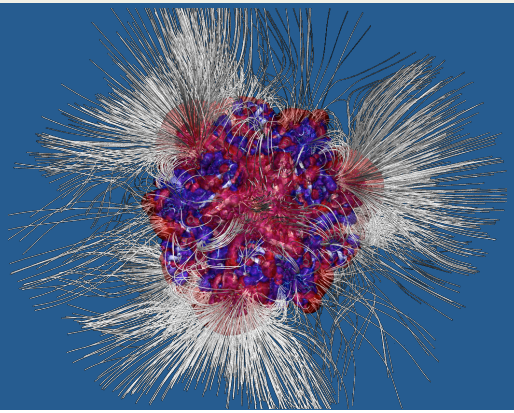
INCREASING THE REALISM OF MOLECULAR DYNAMICS SIMULATIONS: INCORPORATION OF FAST ELECTROSTATIC CALCULATIONS USING GPU DEVICES



NUNO MICAELO

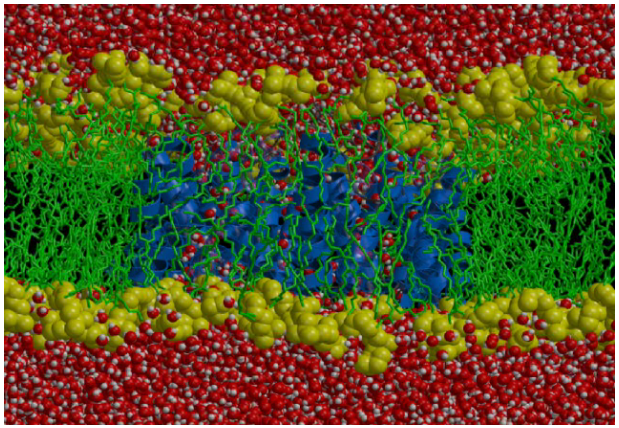
Molecular Modelling and Simulation Lab, Minho University
www.simulation.quimica.uminho.pt

Objective

- Increase de realism of molecular simulations (MS). How?
  - Include new physical models not yet present on the molecular modelling software.
  - We compromise the accuracy of our simulations due to the "poor" computational performance (algorithms/ hardware)
  - MS limitations:
    - Length of the simulations
    - System size

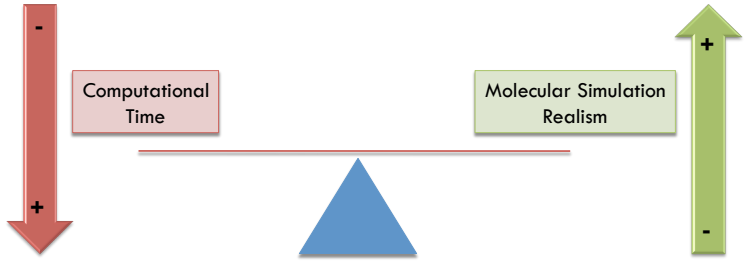
Molecular simulations. What is it?

- Use of a physical model to simulate reality



Increasing the realism of MS with new physical models

- Simulations are far from being perfect
- New physical models can be incorporated in the simulations with a cost in computational time
- Tradeoffs:

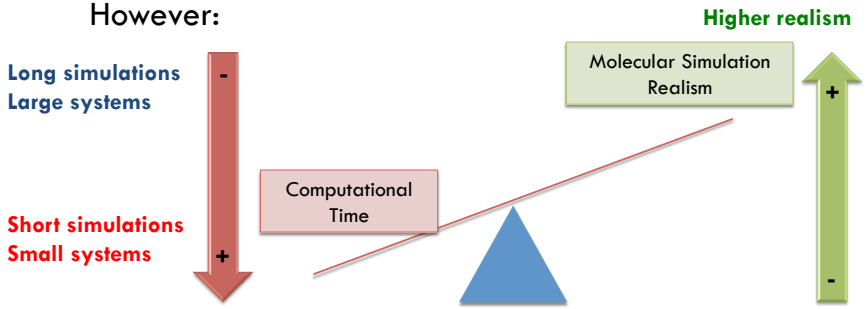


Incorporation of pH effects in MS

- pH is a measure of the acidity/basicity of a medium
- Blood pH is 7.4. A slight change of pH can be fatal (many diseases are related to a change 0.5 pH units)
- pH effects are not included in simulations! Why?
  - Require the calculation of the **electrostatic potential** using the Poisson-Boltzmann (PB) equation.
  - The PB equation must be calculated hundreds to million of times in a typical MS simulation.

Incorporation of pH effects in MS

- A prototype of MS simulations using pH effects have been developed and validated (Baptista, 2009). However:



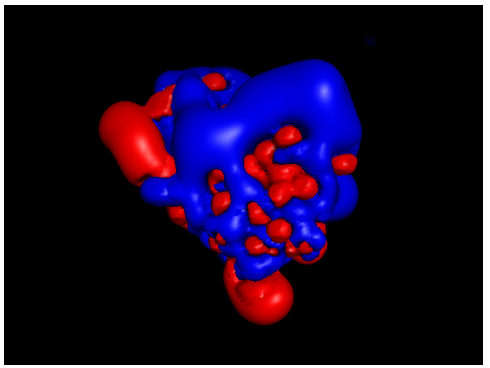
How to overcome this computational problem. Challenges.

- Develop and implement and efficient resolution of the 3D Poisson-Boltzmann (PB) equation using GPU.
- Why calculate PB in GPU?
  - Massive matrix operations
  - Opportunity to develop parallel algorithms with local and shared memory paradigms
  - Literature have been focused on 2D problems. Ours is 3D (more data, more complex, more challenging).
  - Cooperative use of CPU and GPU

GPU 3D Poisson-Boltzmann applications

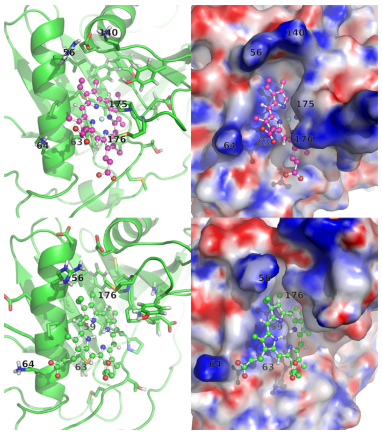
- Application of PB equation in molecular modelling studies:
  - pH
  - **Protein stability**
  - Protein function
  - Drug Design
  - Drug properties
  - Bioengineering

Electrostatic potential surface (Blue: positive Red: negative)



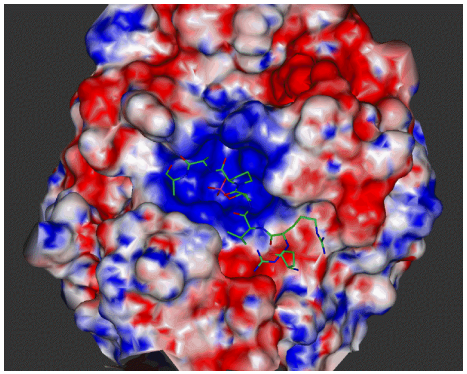
GPU 3D Poisson-Boltzmann applications

- Application of PB equation in molecular modelling studies:
  - pH
  - Protein stability
  - **Protein function**
  - Drug Design
  - Drug properties
  - Bioengineering



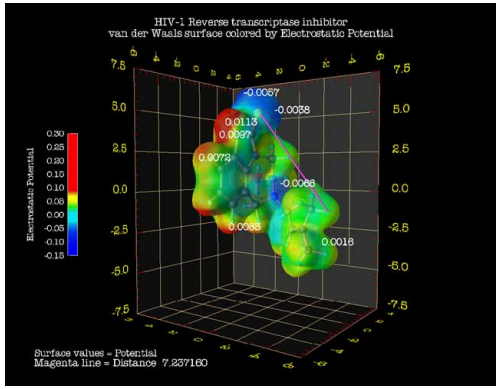
GPU 3D Poisson-Boltzmann applications

- Application of PB equation in molecular modelling studies:
  - pH
  - Protein stability
  - Protein function
  - **Drug Design**
  - Drug properties
  - Bioengineering



GPU 3D Poisson-Boltzmann applications

- Application of PB equation in molecular modelling studies:
  - pH
  - Protein stability
  - Protein function
  - Drug Design
  - **Drug properties**
  - Bioengineering



PB equation

- The electrostatic potential at position **r** is the result of the molecule atomic charges and solvent
- Poisson equation:

$$\nabla \cdot \varepsilon(r) \nabla \phi(r) = -4\pi\rho(r)$$

- $\phi(r)$: potential at position **r**
- $\varepsilon(r)$: dielectric constant at position **r**
- $\rho(r)$: charge density at positon **r**

PB equation

- The Poisson-Boltzmann equation includes the effect of ions in solution

$$\nabla\cdot\varepsilon(r)\nabla\phi(r) = -4\pi\rho(r) + \lambda(r)\kappa^{-2}\varphi(r)$$

$$\kappa^{-2} = \varepsilon\kappa^{2} = \frac{8\pi e^{2} N_a I}{k_B T}$$

- $\lambda(r)$: 1 if solvent, 0 if solute.

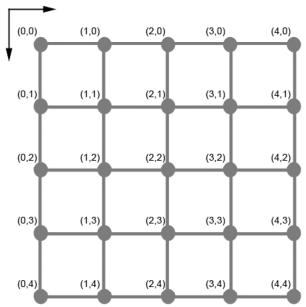
Electrostatic potential calculation using the Poisson equation

- The Poisson equation can be solved using the successive over-relaxation (SOR) algorithm
- SOR is an iterative method for solving partial differential equations
- The Poisson equation has the general form of:

$$\frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = -\frac{\rho}{\varepsilon}$$

Electrostatic potential calculation using the Poisson equation (2D)

- Poisson equation solution for a Ω 2D space
- Uniform grid is used to discretize the Ω space into Ωh



Electrostatic potential calculation using the Poisson equation (2D)

- The approximation of the second-derivatives by the second-order central differences on Ωh leads to a set of n×n×n algebraic equations of the type:

$$u_{i,j-1} + u_{i,j+1} + u_{i-1,j} + u_{i+1,j} - 4u_{i,j1} = h^2 f_{i,j}$$

for i,j,k = 2,3, n-1

  - u: potential
  - h: grid spacing
  - $f_{i,j}=\rho_{i,j}/\varepsilon_{i,j}$
  - n: grid size

Electrostatic potential calculation using the Poisson equation (2D)

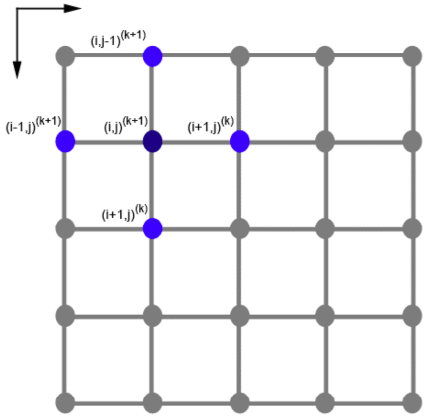
- The SOR method generates a sequence of approximations u(k), k = 1,2,3, … n to the solution until a convergence factor is satisfied or a maximum number of iterations is reached.
- The approximation u(k+1) at the (k+1) iteration is computed using the results u(k) obtained in the k^{th} iteration and the most recent values at nodes (i-1,j) and (i,j-1), thus, for each (i,j) node and $\omega \neq 0$, is given by:

Electrostatic potential calculation using the Poisson equation (2D)

$$u_{i,j}^{(k+1)} = u_{i,j}^{(k)} + \frac{\omega}{4}\left(u_{i,j-1}^{(k+1)} + u_{i,j+1}^{(k)} + u_{i-1,j}^{(k+1)} + u_{i+1,j}^{(k)} - 4u_{i,j}^{(k)} - h^2 f_{i,j}\right)$$

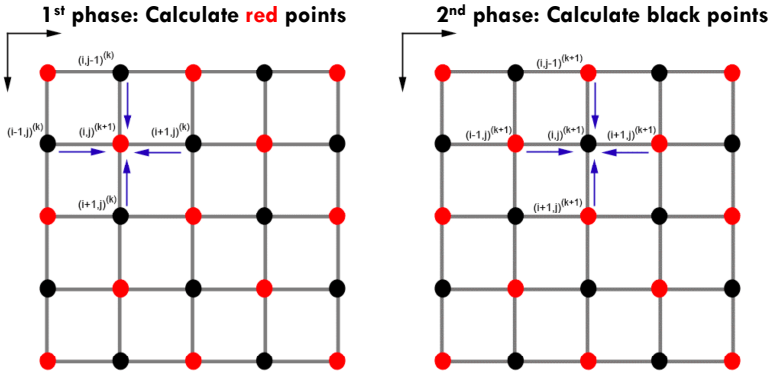
- Where u at node (i,j) of $(k+1)^{th}$ iteration relies on the values of the four neighboring nodes, (i,j+1) and (i-1,j) of the previous kth iteration and, (i-1,j) and (i,j-1) recently calculated nodes of the current iteration
- Optimal values of ω lies in (0,2) and determines the speed of convergence

Electrostatic potential calculation using the Poisson equation (2D)



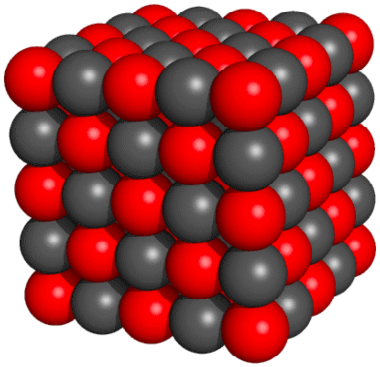
Electrostatic potential calculation using the Poisson equation (2D)

- Parallel Red-Black ordering technique



Electrostatic potential calculation using the Poisson equation

- Our problem is in 3D!
  - Massive data
  - Massive calculations



Electrostatic potential calculation using the Poisson equation

- The red nodes are calculated in phase 1 and are given by:

$$u_{i,j,z}^{(k+1)} = u_{i,j,z}^{(k)} + \frac{\omega}{6}\left(u_{i,j-1,z}^{(k)} + u_{i,j+1,z}^{(k)} + u_{i-1,j,z}^{(k)} + u_{i+1,j,z}^{(k)} + u_{i,j,z-1}^{(k)} + u_{i,j,z+1}^{(k)} - 6u_{i,j,z}^{(k)} - h^3 f_{i,j,z}\right)$$

- and in phase 2 the black nodes are given by:

Previously calculated red nodes

$$u_{i,j,z}^{(k+1)} = u_{i,j,z}^{(k)} + \frac{\omega}{6}\left(u_{i,j-1,z}^{(k+1)} + u_{i,j+1,z}^{(k+1)} + u_{i-1,j,z}^{(k+1)} + u_{i+1,j,z}^{(k+1)} + u_{i,j,z-1}^{(k+1)} + u_{i,j,z+1}^{(k+1)} - 6u_{i,j,z}^{(k)} - h^3 f_{i,j,z}\right)$$

2D/3D-PB GPU Project

- **Part 1.** Develop and implement and efficient resolution of the 2D Poisson-Boltzmann (PB) equation using GPU
- **Part 2.** 3D Poisson-Boltzmann (PB) equation using GPU
- Opportunity to explore different types of parallelism/ scalability and optimization strategies of numeric algorithms.
- Research with high impact in the Molecular Modelling scientific community
- Incorporation in on-going projects at MMSG(UM) and MS(ITQB-UNL) labs

2D/3D-PB GPU Project

- Contact: micaelo@quimica.uminho.pt
- or Google: **Nuno Micaelo**
- www: nunomicaelo.com

Questions?