Overview on the Cardiac ElectroPhysiology Simulator (CEPS)
Nejib Zemzemi, Yves Couderère, Florian Caro, Marc Fuentes

To cite this version:
Nejib Zemzemi, Yves Couderère, Florian Caro, Marc Fuentes. Overview on the Cardiac ElectroPhysiology Simulator (CEPS). Meeting at the Lyric, Jun 2015, Bordeaux, France. <hal-01204600>

HAL Id: hal-01204600
https://hal.archives-ouvertes.fr/hal-01204600
Submitted on 24 Sep 2015

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L’archive ouverte pluridisciplinaire HAL, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d’enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.
Overview on the Cardiac ElectroPhysiology Simulator (CEPS)

Nejib Zemzemi, Yves Coudière, Florian Caro, Marc Fuentes,
INRIA Bordeaux Sud-Ouest, Institut Mathématiques Bordeaux, Institut de Rythmologie et
Modélisation Cardiaque
nejib.zemzemi, yves.coudiere@inria.fr, florian.caro@inria.fr,
marc.fuentes@inria.fr

Goals

- Develop a modular code called CEPS (Cardiac ElectroPhysiology Simulator) useful for doctors and for applied mathematician researchers.
- Develop a parallel code in order to take account multiple scales (from the macroscopic scale to the microscopic scale).
- Develop a kernel framework useful for researchers in medicine and in applied mathematics.
- Develop a parallel code in order to be efficient with clusters like Plafirim, Curie, or personal computer...
- Develop useful tools for installation of the code and validation test cases.

Actors

A lot of persons work in the CEPS code, see the not exhaustive list below

- Juhoor Mehdi (old ADT), the foundation of CEPS with a lot of contribution of Nejib Zemzemi who contributes on the framework of CEPS.
- Marc Fuentes (SED), to help us on everything on the code (compilation, development,…).
- Pierre Elliott Bécue works with CEPS for simulation at the microscopic scale
- Gerard Antoine, works with CEPS on the implementation of the bilayer atria model in CEPS.
- Florian Caro works on numerical methods and on microscopic scale with PE Bécue.
- Charlie Douaunli-tontsi works on high order time numerical schemes with Charles Pierre.
- Yves Coudière will works on high ordrer Finte Volume methods, thanks to the framework of CEPS.
- Those schemes are devoted to be implemented in CEPS.
- Students and PhD thesis for the future.

What it is done currently in CEPS

- Mono domain model is developed in CEPS

\[
\begin{align*}
(C_m \partial_t (\varepsilon g), \nabla \varphi) &\quad &\text{dans } \Omega, \\
\partial_{\nu} (\varepsilon g) - \varphi &\quad &\text{dans } \partial \Omega, \\
\varphi - \varepsilon g &\quad &\text{sur } \partial \Omega, \\
\n &\quad &\text{dans } \partial \Omega,
\end{align*}
\]

where \( \varepsilon \) and \( g \) denote the unknown vector for the ionic variables and the electric potential. Parameters \( \varepsilon \) and \( C_m \) are physical data and \( g \) denotes the conductivity tensor of the medium.

- About 11 300 lines of C++ (whith header files but without test files)

<table>
<thead>
<tr>
<th>Language</th>
<th>files</th>
<th>blank</th>
<th>comment</th>
<th>code</th>
</tr>
</thead>
<tbody>
<tr>
<td>C/++ + Header</td>
<td>98</td>
<td>2915</td>
<td>7357</td>
<td>11343</td>
</tr>
<tr>
<td>C++</td>
<td>55</td>
<td>1602</td>
<td>3000</td>
<td>7610</td>
</tr>
<tr>
<td>CMake</td>
<td>16</td>
<td>74</td>
<td>96</td>
<td>311</td>
</tr>
<tr>
<td>SUM</td>
<td>169</td>
<td>4591</td>
<td>10453</td>
<td>19264</td>
</tr>
</tbody>
</table>

- Validation

  - Unit test case.

  - Validation test case (comparison between exact solution and numerical solution).

<table>
<thead>
<tr>
<th>Test</th>
<th>Description</th>
<th>Time (sec)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1/16</td>
<td>utility.numeric</td>
<td>0.19</td>
</tr>
<tr>
<td>2/16</td>
<td>linearAlgebra.distributedVector</td>
<td>3.39</td>
</tr>
<tr>
<td>3/16</td>
<td>linearAlgebra.distributedHaloVector</td>
<td>1.12</td>
</tr>
<tr>
<td>4/16</td>
<td>linearAlgebra.distributedMatrix</td>
<td>1.24</td>
</tr>
<tr>
<td>5/16</td>
<td>linearAlgebra.linearSystem</td>
<td>1.17</td>
</tr>
<tr>
<td>6/16</td>
<td>geometry.geometry</td>
<td>1.36</td>
</tr>
<tr>
<td>7/16</td>
<td>ionicmodels</td>
<td>6.58</td>
</tr>
<tr>
<td>8/16</td>
<td>pde.common</td>
<td>0.12</td>
</tr>
<tr>
<td>9/16</td>
<td>pde.boundaryConditions</td>
<td>1.34</td>
</tr>
<tr>
<td>10/16</td>
<td>pde.assemblers</td>
<td>10.00</td>
</tr>
</tbody>
</table>