Computer Simulation of Molecular Electrochemistry Experiments—A Rational Approach

Kai Ludwig and Bernd Speiser

Institut für Organische Chemie, Universität Tübingen
mechanism, transport, geometry
(physicochemical model)
mechanism, transport, geometry
(physicochemical model)

partial differential equations
(mathematical model)
mechanism, transport, geometry
(physicochemical model)

partial differential equations
(mathematical model)

coloration profiles, current/potential curves
(numerical model)
uses of simulation in molecular electrochemistry:
- analyze experimental data by comparison to calculations
- predict experimental responses for complex systems

numerical calculations: “digital simulation” (Feldberg, 1969)

programs for specific purposes vs. software packages
Simulation Packages

- purpose: help non-programmers to apply simulation in everyday work

- selection of more or less extended and popular simulation packages in the literature:
  - EASI (Speiser, since 1990) — fixed list of mechanisms
  - Elsim (Bieniasz, since 1992) — complex mathematical relationships
  - Digisim (Feldberg, Rudolph, et al., since 1994) — “any” mechanism, windows style, CV
  - DigiElch (Rudolph, 2004) — variation of Digisim
Common Problems with Simulation Packages

- complex code
  (difficult to understand and maintain)

- extension difficult or impossible
  (mechanisms, geometries, transport phenomena, experiments)

- proprietary code
  (distribution as binary executable)

- limited testing
  ("open science" based on access to code)
A Possible Rational Solution

based on two paradigms:

- use object-oriented programming techniques
- provide open-source code
A Possible Rational Solution

based on two paradigms:

- use object-oriented programming techniques
- provide open-source code
definition:
“writing program text decomposed in modules”
(wikipedia.org) which encapsulate data and actions
Object-Oriented Methods I

- definition:
  “writing program text decomposed in modules” (wikipedia.org) which encapsulate data and actions

- use: large scale software projects
definition: “writing program text decomposed in modules” (wikipedia.org) which encapsulate data and actions

use: large scale software projects

possible advantages:
- production of flexible, extensible code
- re-use of tested code
- localized debugging and maintenance
- better understanding of real system
Object-Oriented Methods II

three common steps of working with the object-oriented paradigm:

- analysis: identify concepts (in the real system)
- design: describe interaction, define classes (construction plans for objects)
- programming: implement in a programming language (write the code)
Object-Oriented Methods III

- object: instantiation of a class
- a program: collection of interacting objects
- relations between classes (and objects):
  - exchange of information
  - composition
  - derivation
Object-Oriented Methods III

- object: instantiation of a class
- a program: collection of interacting objects
- relations between classes (and objects):
  - exchange of information
  - composition
  - derivation
Object-Oriented Methods III

- object: instantiation of a class
- a program: collection of interacting objects
- relations between classes (and objects):
  - exchange of information
  - composition
  - derivation
Object-Oriented Methods III

- object: instantiation of a class
- a program: collection of interacting objects
- relations between classes (and objects):
  - exchange of information
  - composition
  - derivation
Object-Oriented Methods III

- object: instantiation of a class
- a program: collection of interacting objects
- relations between classes (and objects):
  - exchange of information
  - composition
  - derivation
- objects are “building blocks” of program
Technical Aspects of EChem++

- programming language: C++
- operating system: Linux
- extended use of libraries
  
<table>
<thead>
<tr>
<th>Library</th>
<th>Website</th>
</tr>
</thead>
<tbody>
<tr>
<td>Quantity</td>
<td>quantity.sourceforge.net</td>
</tr>
<tr>
<td>Spirit</td>
<td>spirit.sourceforge.net</td>
</tr>
<tr>
<td>GiNaC</td>
<td><a href="http://www.ginac.de">www.ginac.de</a></td>
</tr>
<tr>
<td>GMM++</td>
<td><a href="http://www.gmm.insa-tlse.fr/getfem">www.gmm.insa-tlse.fr/getfem</a></td>
</tr>
<tr>
<td>VTK</td>
<td>public.kitware.com/VTK</td>
</tr>
<tr>
<td>Qt</td>
<td><a href="http://www.trolltech.com">www.trolltech.com</a></td>
</tr>
<tr>
<td>tree.hh</td>
<td><a href="http://www.damtp.cam.ac.uk/user/kp229/tree">www.damtp.cam.ac.uk/user/kp229/tree</a></td>
</tr>
</tbody>
</table>
Example Objects in EChem++ I

excitation functions: induce changes during experiment

class Segment, class ExcitationFunction
Example Objects in EChem++ I

excitation functions: induce changes during experiment
class Segment, class ExcitationFunction
Example Objects in EChem++ I

excitation functions: induce changes during experiment

class Segment, class ExcitationFunction
Example Objects in EChem++ I

excitation functions: induce changes during experiment
class Segment, class ExcitationFunction
**Example Objects in EChem++ I**

excitation functions: induce changes during experiment

class Segment, class ExcitationFunction
Example Objects in EChem++ II

reaction network:
translate mechanisms into internal representation

class ReactionNetwork, class Law
Example Objects in EChem++ II

reaction network:
translate mechanisms into internal representation
class ReactionNetwork, class Law
Example Objects in EChem++ II

reaction network:
translate mechanisms into internal representation

class ReactionNetwork, class Law
Example Objects in EChem++ II

reaction network:
translate mechanisms into internal representation
class ReactionNetwork, class Law
Example Objects in EChem++ III

numerical integration:
solve differential equations by adaptive multilevel finite element method
class RotheRosenbrock, class MFEM
Example Objects in EChem++ III

numerical integration:
solve differential equations by adaptive multilevel finite element method

class RotheRosenbrock, class MFEM

integrate systems of

\[ \frac{\partial c}{\partial t} = D \frac{\partial^2 c}{\partial x^2} + \rho(c) \]

subject to appropriate initial and boundary conditions
numerical integration:
solve differential equations by adaptive multilevel finite element method

**class RotheRosenbrock, class MFEM**
Proposed Electrochemical Simulation High Level Modules
Example Simulation I: pseudo-first order reaction

\[
A \quad \Leftrightarrow \quad B + e^- \quad E^0, k_s, \alpha, D
\]

\[
B + C \quad \rightarrow \quad D \quad k
\]

assume: cyclic voltammetry, \( c_C = 100 \times c_A, k = 100 \text{ s}^{-1} \)

treat follow-up reaction as

- second order (e.g., in DigiSim)
- pseudo-first order (power rate law: \( r = kc_B \); this work)

decouple stoichiometry and kinetics
Example Simulation II: non-triangular wave forms

\[ A \rightleftharpoons B + e^- \quad E^0, k_s, \alpha, D \]

assume: accumulation of B in diffusion layer by prolonged potential-controlled oxidation

vary excitation functions:

use of more complicated, non-triangular waveforms
Example Simulation III: two working electrode system

\[
\begin{align*}
A & \Leftrightarrow B + e^- & E_1^0, k_{s,1}, \alpha_1, D_1 \\
A + e^- & \Leftrightarrow C & E_2^0, k_{s,2}, \alpha_2, D_2 \\
B + C & \Leftrightarrow 2A & K, k
\end{align*}
\]

assume: generate oxidation and reduced form at two working electrodes

specify electrode geometry: \( d = 50 \mu m \)
\( k = 10^7 \text{ l mol}^{-1} \text{ s}^{-1} \)
\( E_1^0 = +0.25 \text{ V}, E_2^0 = -0.25 \text{ V} \)

define complex geometries with multiple boundaries (currently only 1D)
EChem++ and Open Source Paradigm

open-source software is open for . . .

- use
- modification
- redistribution

in source form

apart from licence considerations, this is promoted within EChem++ by . . .

- modular, object-oriented design
- extendability
- free availability on the internet
Conclusion: The EChem++ Simulation Package

- simulation software for molecular electrochemistry
- code based on object-oriented methods
- provides framework for extensions to additional processes, conditions, etc.
- already in present state some improvements above existing software
- open-source character
Invitation

download, test, report, contribute

http://echempp.sourceforge.net
Acknowledgements

- financial support:
  Deutsche Forschungsgemeinschaft

- co-operation:
  excitation function: L. Rajendran
  graphical user interface: A. Millers