A HIGH PERFORMANCE WEBAPPLICATION FOR AN ELECTRO-BIOLOGICAL PROBLEM

Philipp Schwaha, René Heinzl, Georg Mach, Siegfried Selberherr
Institute for Microelectronics
TU Wien
1040 Wien, Austria
E-mail: {schwaha | heinzl | mach | selberherr}@iue.tuwien.ac.at

Claudia Pogoreutz, Susanne Fister
Faculty of Life Sciences
Universität Wien
1090 Wien, Austria

KEYWORDS

Simulation, Biology, Web Application

ABSTRACT

We present a web application for the simulation of a biological system. An electrically active fish and the corresponding charge relaxation derived from the Maxwell equations are presented. The transition from the equations to the numerical methods is performed with a finite volume discretisation scheme. The final approach to solve the system of resulting equations makes use of a high speed generic scientific simulation environment. The ability to delegate calculations to other hosts further ensures the high performance levels achieved by the highly optimised simulations.

INTRODUCTION

The Institute for Microelectronics has a long standing tradition of developing simulation software in the field of technology computer aided design (TCAD) [Selberherr et al.(1980)], [Sabelka and Selberherr(1998)], [IμE(2004)]. The fast pace in this field [ITRS(2005)] and the high demands placed on all simulation components has led to a large variety of available robust high performance techniques of dealing with the numerical solution of differential equations.

Recently the high demands on software quality have induced the development of a generic scientific simulation environment (GSSE) [Heinzl et al.(2006b)], [Schwaha et al.(2006)Schwaha, Heinzl, Spevak, and Grasser]. This environment offers capabilities to implement algorithms at a semantically high abstraction level and high performance directly in C++. As the name already suggests it is the aim of this development to provide an environment which is applicable to a broad range of scientific problems. Several projects [IμE(2006)] were initiated in order to test the extent to which this goal has been reached.

During one of these projects dealing with electric phenomena in biological organisms it became obvious, that it is not only necessary to provide a high performance simulation application, but that an appropriate interface is of utmost importance, especially when cooperating with scientists from different fields of research. It also became apparent, that simulation is able to illustrate complex theories and models, if the results can be visualised properly. This also demands that the user interface has to be accessible to a multitude of people with different background knowledge. Such an interface has to be implemented orthogonally to the underlying simulation, to be consistent with modern application designs [Heinzl et al.(2006a)]. A web interface fulfils these requirements, as it presents an easily accessible, widely accepted and generally available interface.

The following gives a short outline of the electro-biological problem treated.

TOPIC

Electric phenomena are common in biological organisms such as the discharges within the nervous system, but usually remain within a small scale. In some organisms, however, the electric phenomena take a more prominent role. Some fish species, such as *Gnathonemus petersii* from the family of Mormyridae [Westheide and Rieger(2003)], [Dudel et al.(2001)], use them for detection of their prey and for communication among their own kind.

The Mormyridae live in fresh water habitats of South Africa, especially in rivers. The animals are crepuscular, which means that they are inactive during the day, but start to feed as soon as the sun goes down. They feed by searching for small prey that lives in the muddy benthic substrate, but also catch plankton close to the substrate. The nose region of *Gnathonemus petersii* is modified to form a chin or snout like form. The shape representing the fish in the simulations is given in Fig. 1. The snouts possess very sensitive receptor cells which help to find small prey in the mud. These animals are very interesting for biologists firstly because of their electric sense, secondly because of their behaviour: The play behaviour they exhibit is unique within the bone fish family.

The up to 30 cm long fish actively generates electric pulses with an organ located near its tail fin (marked in Fig. 1). The distortions of the generated field are

Proceedings 21st European Conference on Modelling and Simulation
Ivan Zelinka, Zuzana Oplatková, Alessandra Orsoni ©ECMS 2007
ISBN 978-0-9553018-2-7 / ISBN 978-0-9553018-3-4 (CD)

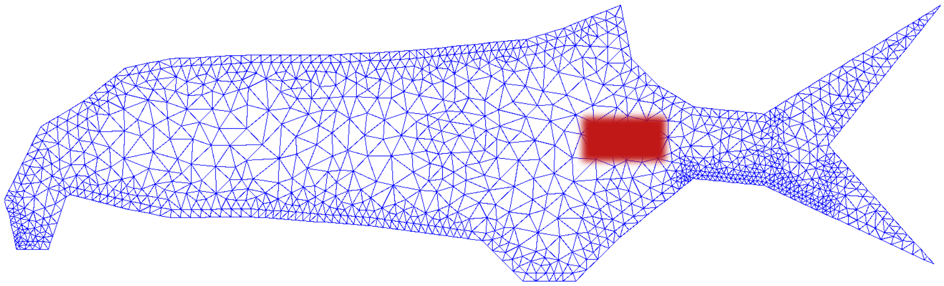

Fig. 1. A triangulated model of an electrically active fish. The electrically active organ is marked in red.

then picked up by the fish's receptors in its nose region. It can thereby distinguish between inanimate objects, prey, predators, or members of its own kind.

The goal of our simulation is to approximate the electric field generated by the fish and to determine how the electric field changes at the detecting cells, when different objects are introduced in the vicinity of the fish. This is difficult to obtain reliably with live specimen. The modelling and implementation of the electric phenomena is discussed in the next section.

Simulation Environment

To realise a simulation we require a model of the physical phenomenon, which can then be implemented. Therefore we first have to establish the theoretical foundations of our simulation.

For our case we derive the equation system based on a quasi-electrostatic system directly from the corresponding Maxwell equations [Maxwell(1873)]. We use the conservation law of charge, where ϱ represents the space charge and $\vec{J}$ the current density

$$\partial_t \, \varrho + \operatorname{div} \vec{J} = 0 \tag{1}$$

and the divergence theorem (Gauss's law), where $\vec{D}$ represents the dielectric flux

$$\operatorname{div} \vec{D} = \varrho \tag{2}$$

With the terms of a dominant electric field and the material properties we use

$$\vec{D} = \varepsilon \vec{E} \tag{3}$$

$$\vec{J} = \gamma \vec{E} \tag{4}$$

where $\vec{E}$ represents the electric field, ε the permittivity, and γ the conductivity.

Then we get

$$\partial_t \, \varrho + \operatorname{div} \gamma \vec{E} = 0 \tag{5}$$

$$\operatorname{div} \varepsilon \vec{E} = \varrho \tag{6}$$

and finally by incorporating Equation 6 into Equation 5 and the assumption of constant conductivity and permittivity we obtain:

$$\partial_t \operatorname{div} \vec{E} + \frac{\gamma}{\varepsilon} \operatorname{div} \vec{E} = 0 \tag{7}$$

The charge separation which is actively taking place within parts of the simulation domain is modelled as an inhomogeneity $-P$, giving:

$$\partial_t \operatorname{div} \vec{E} + \frac{\gamma}{\varepsilon} \operatorname{div} \vec{E} = -P \tag{8}$$

Finally we, assume the existence of an electrostatic potential defined as:

$$\vec{E} = - \operatorname{grad} \Psi, \tag{9}$$

Equation 8 can be transformed to the following form:

$$\partial_t \operatorname{div} (\varepsilon \operatorname{grad} \Psi) + \operatorname{div} (\gamma \operatorname{grad} \Psi) = P \tag{10}$$

Equation 10 has to be solved numerically, because a closed form analytical solution does not exist for the general case. We use the GSSE for our implementation.

GSSE

Equation 10 is discretised using the finite volume discretisation scheme [Selberherr(1984)] and is then implemented using the facilities offered by the GSSE. The following snippet of code, which already represents all relevant parts of the final application and can be used for an arbitrary spatial dimension, illustrates the expressive power of the GSSE:

```
equation = sum<vertex_edge>(_v)
[
 Orient(_v, _1) *
 sum<edge_vertex>(_e)
 [
   lineqn(psi(_1)) * Orient(_e, _1)
 ]
 * (area(_1) / dist(_1))
 * (gamma(_1) * deltat + eps(_1))
] +  vol(_1) * ((P(_1) * deltat) + rho(_1)))
```

The generic traversal mechanisms encapsulates the corresponding dimensional properties and grid types with the `sum<vertex_edge>` traversal operations [Heinzl et al.(2006c)]. While this satisfies the topological requirements of the finite volume discretisation method, additional functionality is required from the framework to correctly assemble the equation system.

The `Orient()` function implements an automatic mechanism to evaluate the sign corresponding to the vertices forming a given edge. The `lineqn()` represents a linearised equation [Spevak et al.(2006)], which

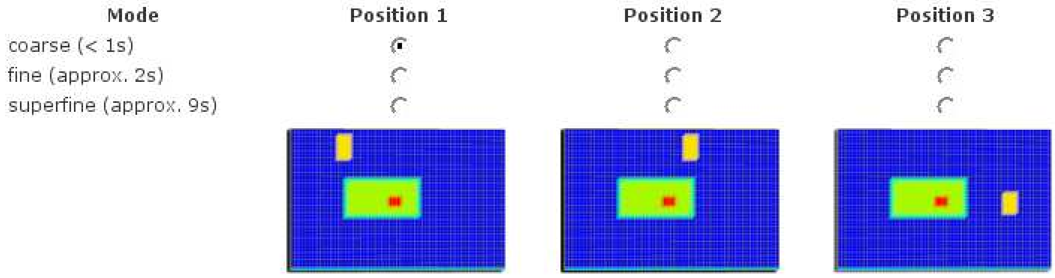

Fig. 2. Different geometrical placement and meshes.

is capable to describe the residual values of the linearised model of the discretised equation. Functional programming is used with the mechanism of the unnamed function object _1. Automatic data accessors [Heinzl et al.(2006c)] for `gamma()`, `eps()`, and `rho()` are used to ensure that the correct locality of the quantities can be determined. Functional programming in C++ uses local scopes which are indicated in the code snippets by `[...]`. Due to the local scopes, the outer function objects must explicitly make quantities available by stating _v and _e. `area()` and `dist()` are used to calculate the values demanded by the finite volume scheme.

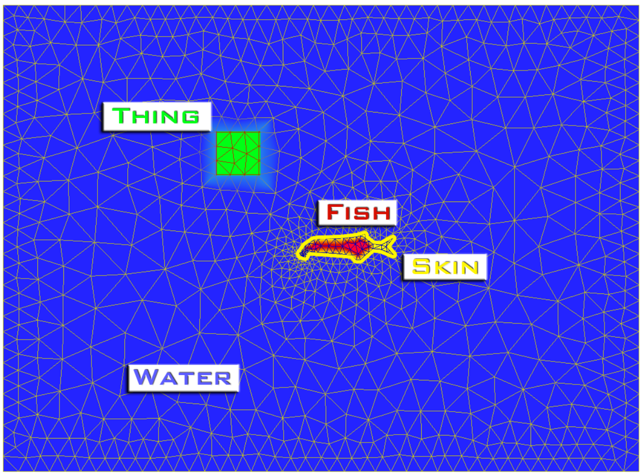

Fig. 3. Simulation setting

The simulation domain is divided into several parts as shown in Fig. 3. The parts include the fish itself, its skin, which serves as insulation, the water the fish lives in, and an object which represents either an inanimate object or prey. The parameters of each part can be adjusted separately.

While the shown specification of the simulation retains a lot of high level semantics and yields a highly performant executable, an appropriate interface must also be provided to make it as accessible as possible.

The Web Application

Access to the world wide web has become all but ubiquitous in the last years. Most people are familiar with the interfaces and semantics established in web pages and HTML forms. It can be therefore viewed as common ground for people of scientifically different backgrounds. From this point of view web pages are well suited as an interface to applications that bridge the gap between different branches of science.

A web application as a front end to the simulation environment has to fulfil several tasks:

- Choose and adjust the simulation parameters.
- Invoke the simulation with the simulation.
- Take the results of the simulation corresponding to the parameters and present them to the user.

A web application has to accomplish these tasks without interweaving the simulation itself with the presentation and control parts. In addition the strain on the web server itself should be kept to a minimum, in order to ensure low latencies. A model of such an architecture is shown in Fig. 4.

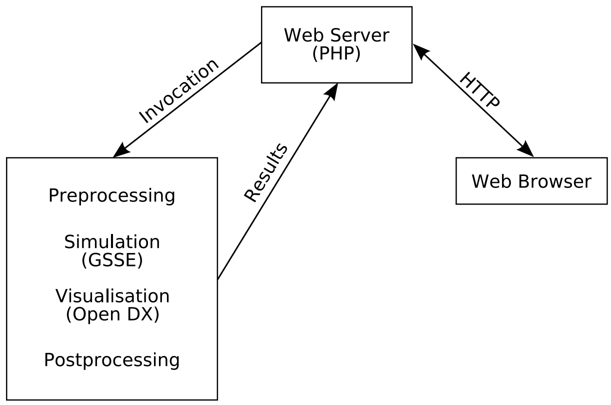

Fig. 4. Overview of the web application's architecture.

The user contacts the web server with a standard web browser. After some preprocessing the web server invokes the simulation. This is the point at which the simulation can be easily delegated to another computer with appropriate simulation applications and the required computing power. After the simulation has finished the postprocessing module generates appropriate output which the web server sends back to the user's web browser. The whole process has to be accomplished in a time frame of seconds in order to remain comparable to that of standard web browsing. Therefore, both, the simulation and the generation of a response, need to be implemented in a performance conscious way.

Our Implementation

Our implementation of the web application can be accessed under `http://webapplication.gsse.at/` to

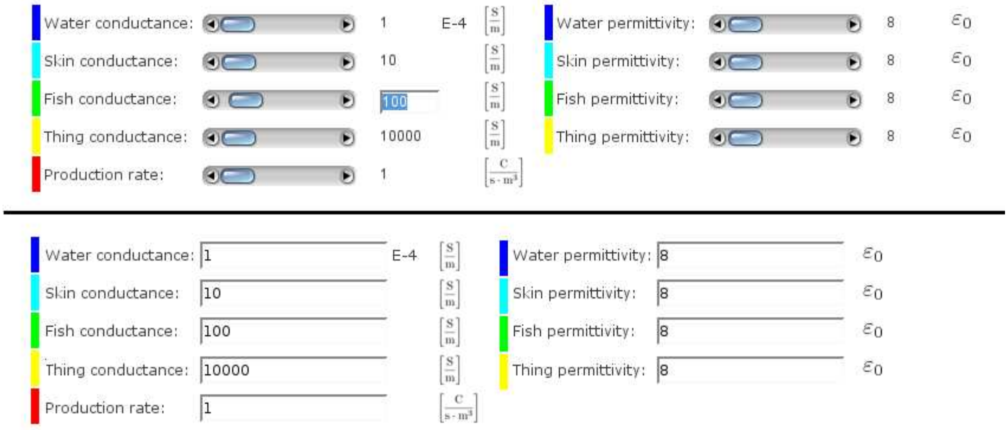

Fig. 5. Parameter input modes. top: default mode; bottom: expert mode.

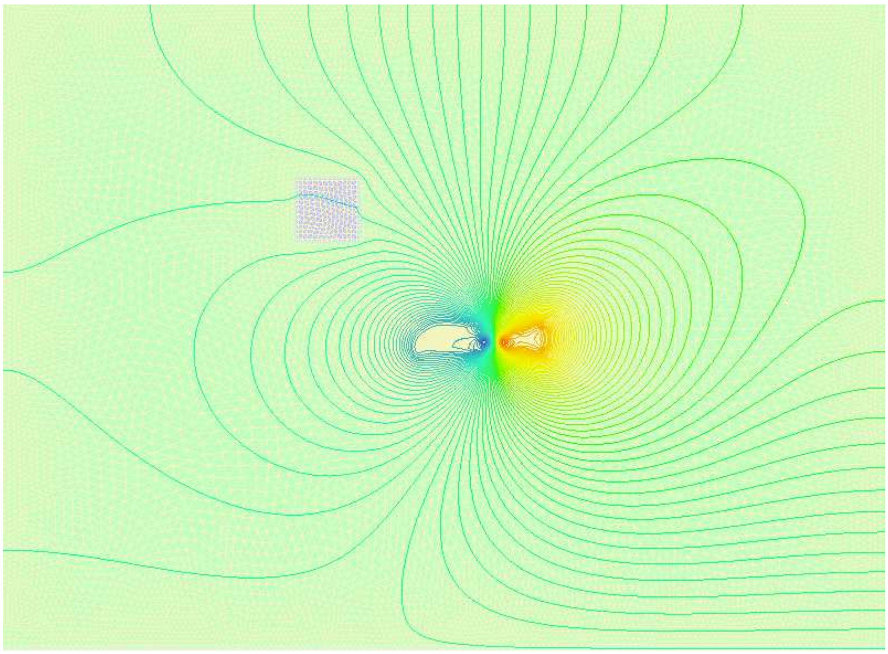

Fig. 6. The results of a simulation performed with a very fine mesh.

give the reader the opportunity for experimentation. The user can choose from three different geometric constellations of the fish and an object (thing) as shown in Fig. 2. Three different mesh resolutions are available with approximate simulation times given for each of them.

Apart from the geometric configuration it is of course also possible to adjust the parameters of the simulation using HTML forms. In our case this includes the conductivities and permittivities of the various simulated objects (thing, fish, skin, water) and the rate of charge separation (production rate) inside the fish. As shown in Fig. 5 two modes of entry are available, one using Java script sliders (default) and another one using HTML input boxes (expert). The reason for the two modes is, that by using the sliders it is possible to restrict the range of input to the simulation to safe values, thereby ensuring the proper operation of the underlying simulation. However, this type of restriction is cumbersome during experimentation. Therefore the HTML input boxes do not pose any restriction, allowing entry of arbitrary values.

We implemented our web application using PHP [PHP]. The choice of PHP was motivated by the facts, that it, on the one hand side, integrates into web servers very easily, is especially well suited to generate HTML pages, and on the other hand side provides a multitude of facilities to execute external programs, enabling both local and remote invocation of our GSSE based simulation tool.

As has already been stated, the communication time between the simulating node and the web server must be appropriately short. In our setup the communication time was kept below 0.2 seconds when using a separate computer for simulation. With GSSE the simulation times themselves are kept in the same order of magnitude, at least for the coarser meshes.

In this fashion we have ensured the orthogonality of the core simulation and the presentation of the results. Furthermore, we do not need to specially adapt the simulation software in any way for the web application. This not only reduces maintenance, but also removes an unnecessary source of errors.

After the simulation has finished the results must be transformed into an image which can be viewed by the user. We use the open data explorer [IBM(1993)] to generate the images. It has been chosen because of its scripting capabilities. A resulting image is shown in Fig. 6.

The parameters entered into the web application and passed to the simulation are recorded along with the results. It is thereby possible to efficiently cache the results which have already been requested at least once. When the same parameters are entered for another run of the simulation, the calculation is skipped and the cached result is returned. This simple mechanism helps to conserve computer resources.

Additional details concerning the development available at `http://tutorial.gsse.at/webapplication`.

Conclusion

We have presented a web application used as an interface to a simulation composed of orthogonal components. The core is realised using the high performance GSSE, enabling fast responses. Our orthogonal design furthermore makes it possible to outsource calculations from the web server, while the caching mechanism helps to conserve computational recourses. The ease of use of a web application also makes the simulation more accessible to non-domain experts as it hides the complexities and concentrates on presenting valid results.

Author Biographies

SUSANNE FISTER was born in 1985 in Kufstein, Austria. She studies biology (genetics and microbiology) at the University of Vienna.

RENÉ HEINZL was born in Vienna, Austria, in 1977. He studied electrical engineering at the Technische Universität Wien, where he received the degree of Diplomingenieur in 2003. He joined the Institute for Microelectronics in November 2003, where he is currently working on his doctoral degree. His research interests include programming paradigms, high performance programming techniques, performance analysis, process simulation, scientific visualisation, algebraic topology, mesh generation and adaptation for TCAD.

GEORG MACH was born in Vienna, Austria, in 1979. He studies electrical engineering at the Technische Universität Wien. He joined the Institute for Microelectronics in June 2003, where he is currently working on his diploma thesis.

CLAUDIA POGOREUTZ was born in Kufstein, Austria, in 1985. She studies zoology with emphasis on marine fauna at the University of Vienna.

PHILIPP SCHWAHA was born in Vienna, Austria, in 1977. He studied electrical engineering at the TU Wien. He joined the Institute for Microelectronics in June 2004, where he is currently working on his doctoral degree. His research activities include circuit and device simulation, algebraic structures, and software development.

SIEGFRIED SELBERHERR was born in Klosterneuburg, Austria, in 1955. He received the degree of Diplomingenieur in electrical engineering and the doctoral degree in technical sciences from the Technische Universität Wien in 1978 and 1981, respectively. Prof. Selberherr has been holding the venia docendi on Computer-Aided Design since 1984. From 1988 to 1999 he was the Head of the Institute for Microelectronics. From 1998 to 2005 he served as Dean of the Faculty of Electrical Engineering and Information Technology. His current research topics are modelling and simulation of problems for microelectronics engineering.

References

[PHP] *PHP: Hypertext Processor.* URL `http://www.php.net/`. `http://www.php.net/`.

[Dudel et al.(2001)] Josef Dudel, Randolf Menzel, and Robert F. Schmidt. *Neurowissenschaft. Vom Molekül zur Kognition.* Springer, Berlin, 2001. ISBN 3540413359.

[Heinzl et al.(2006a)] R. Heinzl, M. Spevak, P. Schwaha, and T. Grasser. Concepts for High Performance Generic Scientific Computing. In *Proc. of the 5th MATHMOD*, volume 1, Vienna, Austria, February 2006a.

[Heinzl et al.(2006b)] R. Heinzl, M. Spevak, P. Schwaha, and T. Grasser. A High Performance Generic Scientific Simulation Environment. In *Proc. of the PARA Conf.*, page 61, Umea, Sweden, June 2006b.

[Heinzl et al.(2006c)] R. Heinzl, M. Spevak, P. Schwaha, and S. Selberherr. A Generic Topology Library. In *Library Centric Sofware Design, OOPSLA*, pages 85–93, Portland, OR, USA, October 2006c.

[IBM(1993)] *IBM Visualization Data Explorer.* IBM Corporation, Yorktown Heights, NY, USA, third edition, February 1993.

[IμE(2006)] IμE. *IμE Summer of Code 2006.* Institut für Mikroelektronik, Technische Universität Wien, Austria, 2006. `http://www.summerofcode.at`.

[IμE(2004)] IμE. *MINIMOS-NT 2.1 User's Guide.* Institut für Mikroelektronik, Technische Universität Wien, Austria, 2004. `http://www.iue.tuwien.ac.at/software/minimos-nt`.

[ITRS(2005)] ITRS. International Technology Roadmap for Semiconductors, 2005.

[Maxwell(1873)] James Clerk Maxwell. *A treatise on electricity and magnetism.* 1873.

[Sabelka and Selberherr(1998)] R. Sabelka and S. Selberherr. SAP — A Program Package for Three-Dimensional Interconnect Simulation. In *Proc. Intl. Interconnect Techn. Conf.*, pages 250–252, Burlingame, California, June 1998. ISBN 0-7803-4285-2.

[Schwaha et al.(2006)Schwaha, Heinzl, Spevak, and Grasser] P. Schwaha, R. Heinzl, M. Spevak, and T. Grasser. Advanced Equation Processing for TCAD. In *Proc. of the PARA Conf.*, page 64, Umea, Sweden, June 2006.

[Selberherr(1984)] S. Selberherr. *Analysis and Simulation of Semiconductor Devices.* Springer, Wien–New York, 1984. ISBN 3-211-81800-6.

[Selberherr et al.(1980)] S. Selberherr, A. Schütz, and H.W. Pötzl. MINIMOS—A Two-Dimensional MOS Transistor Analyzer. *IEEE Trans. Electron Dev.*, ED-27(8):1540–1550, 1980.

[Spevak et al.(2006)] M. Spevak, P. Schwaha, R. Heinzl, and T. Grasser. Automatic Linearization using Functional Programming for Scientific Computing. In *Proc. of the ICCAM Conf.*, page 147, Leuven, Belgium, July 2006.

[Westheide and Rieger(2003)] Wilfried Westheide and Reinhard Rieger. *Spezielle Zoologie. Teil 2: Wirbel- oder Schädeltiere.* Elsevier, 2003. ISBN 978-3-8274-0900-3.