Electro-Biological Simulation using a Web Front-End

P. Schwaha, R. Heinzl, G. Mach, C. Pogoreutz, S. Fister, and S. Selberherr

Institute for Microelectronics, TU Wien
Gußhausstraße 27-29/E360, A-1040 Vienna, Austria
E-mail: {schwaha | heinzl | mach | selberherr}@iue.tuwien.ac.at

KEYWORDS
Simulation, Biology, Web Application, Linux

ABSTRACT
We have developed a web application for the simulation in the field of electro-biology, particularly, for the analysis of a weakly electrically active fish. We investigated the charge relaxation due to the electric activity of the fish. The governing relaxation equation is discretized with the finite volume method. The discretized equations are then implemented using a generic scientific simulation environment. To ensure high responsiveness and scalability we also included means to distribute the calculations to computers on a network.

INTRODUCTION

The pursuit and the acquisition of new insights is common to all sciences. However, the methodologies and tools utilized to advance on the goal are quite different, due to the very different natures of the various scholarly disciplines. Simulation is a technique that has been applied very successfully in several fields of science, such as physics, and has even spawned its own branch of research: scientific computing. While simulations have proven to be very important to understand the various phenomena in nature, some areas of research have all but neglected this technique.

In order provide a means to quickly implement even complex algorithms directly in C++ we developed the scientific simulation environment (GSSE) [5, 6, 8, 13] which not only provides high performance but also takes into account the often neglected topic of software quality. Although its roots can be traced to satisfy the needs of technology computer aided design (TCAD) [4, 11, 12, 14], its application is not limited to this field of research. Investigations concerning this during several student projects have been very positive [10].

One of these projects, that dealt with electric phenomena in biological organisms, also revealed that high performance is not the only important factor when cooperating with scientists with very different backgrounds. In this context the benefit of appropriate visualization should also not be underestimated [2], as it often determines how well the simulation results can be interpreted by the target audience.

The goal therefore was to provide an user interface that is easily accessible by people from different fields of research with a minimum of effort. By following modern application design rules, that components should be implemented orthogonally [7] it is made possible that the underlying simulation remains unmodified, which reduces sources of errors.

Because access to the world wide web has become all but ubiquitous in the last years, it is a promising choice for a user interface. 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.

THE SUBJECT UNDER INVESTIGATION

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 species of fish, such as *Gnathonemus petersii* from the family of Mormyridae [3, 15, 16], use them for detection of their prey and for communication among their own kind.

The nose region of *G. 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.

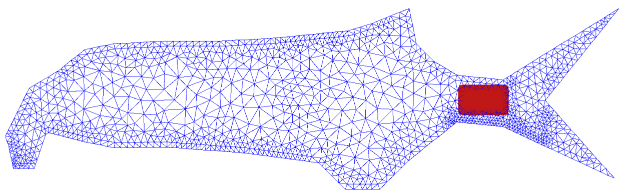

Figure 1: A triangulated model of an electrically active fish. The position of the electrically active organ is marked in red.

The goal of our simulations is to analyze the electric field generated by the fish, when different objects are introduced in the vicinity of the fish. This is difficult to obtain reliably with live specimen. The web application is described in the following section.

THE WEB APPLICATION

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

- Choose and modify the simulation parameters.
- Start the simulation with the selected parameters.
- Process 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. 2.

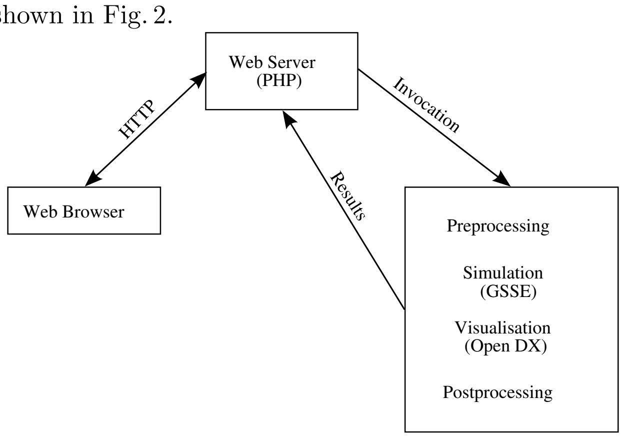

Figure 2: Overview of the web application's architecture.

The user contacts the web server using 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. After the simulation has finished the post processing 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 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. 4. 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 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. We have thereby

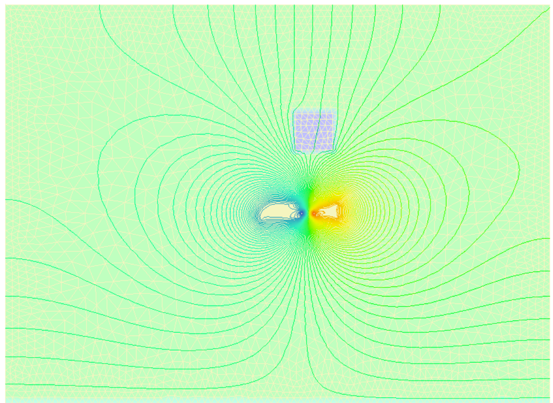

Figure 3: The results of a simulation performed with a very fine mesh.

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 to be used from the web application. This not only reduces maintenance, but also removes an unnecessary source of errors.
We use OpenDX [9] to generate images from the results of the calculation, because of its scripting capabilities. A resulting image is shown in Fig. 3.
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 are available at `http://tutorial.gsse.at/webapplication`. The modeling and implementation of the electric phenomena is discussed in the next section.

SIMULATION ENVIRONMENT

To realize a simulation we require a model of the physical phenomenon that can then be implemented. Therefore we first have to establish the theoretical foundations of our simulation.
We model the electrical phenomena due to the fish by a relaxation equation which can be directly derived from Maxwell's equations in a quasi-electrostatic case. It takes the following shape:

$$\partial_t \operatorname{div}(\varepsilon \operatorname{grad} \Psi) + \operatorname{div}(\gamma \operatorname{grad} \Psi) = P \qquad (1)$$

Here Ψ is the electrostatic potential, while ε and γ

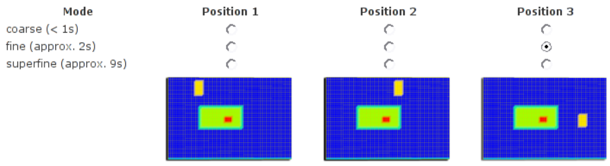

Figure 4: Different geometrical placement and meshes.

describe the permittivity and the conductivity respectively. The separation of charge, which is actively performed by the fish, is modeled by the inhomogeneity P. Equation 1 has to be solved numerically, because a closed form analytical solution does not exist for the general case.

GSSE

The GSSE has been developed in order to provide a reusable high performance framework for scientific computing at a high semantic level with C++.
Equation 1 is discretized using the finite volume discretization scheme [1] 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 arbitrary spatial dimension, illustrates the expressive power of the GSSE:

```
equation = sum<vertex_edge>
[ sum<edge_vertex>
  [ linearize(psi) ]
  * (area / dist) * (gamma * delta + eps)
] + vol * ((P * deltat) + rho)
```

All error prone details related to traversal or data access operations are taken care of by the GSSE, therefore, it can be used on structured as well as unstructured spatial discretizations called grids or meshes respectively.

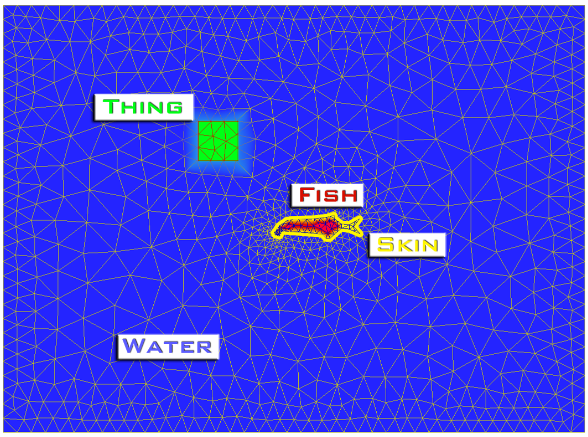

Figure 5: Simulation setting

The simulation domain is divided into several parts as shown in Fig. 5. 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.

CONCLUSION

We have shown how we used a web application as an interface to a simulation composed of orthogonal components. The core is implemented using the high performance GSSE, ensuring a minimum of runtime of the simulation itself. A caching mechanism helps to further conserve computational resources and enables fast responses. Our modular and orthogonal design furthermore enables an outsourcing of calculations from the web server. The ease of use of a web application hides the complexities and concentrates on presenting valid results and thereby makes the simulation more accessible to non-domain experts.

REFERENCES

[1] T. Barth and M. Ohlberger. Finite Volume Methods: Foundation and Analysis, 2004.

[2] W. Benger, G. Ritter, and R. Heinzl. The Concepts of VISH. In *Proc. of the 4th High-End Visualization Workshop*, pages 28–41, Tyrol, Austria, June 18–22 2007.

[3] Josef Dudel, Randolf Menzel, and Robert F. Schmidt. *Neurowissenschaft. Vom Molekül zur Kognition.* Springer, Berlin, 2001.

[4] T. Grasser (Editor). *Advanced Device Modeling and Simulation.* World Scientific Pub Co Inc, 2003.

[5] R. Heinzl. *Concepts for Scientific Computing.* Dissertation, Technische Universität Wien, 2007.

[6] 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 2006.

[7] 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 2006.

[8] R. Heinzl, M. Spevak, P. Schwaha, and S. Selberherr. A Generic Topology Library. In *Proc. of the Object-Oriented Programming Systems, Languages, and Applications Conf.*, pages 85–93, Portland, OR, USA, October 2006.

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

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

[11] C. Jungemann and B. Meinerzhagen. *Hierarchical Device Simulation. The Monte Carlo Perspective.* Springer, 2003.

[12] 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.

[13] 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.

[14] S. Selberherr. *Analysis and Simulation of Semiconductor Devices.* Springer, Wien–New York, 1984.

[15] G. von der Emde and S. Fetz. Distance, Shape and More: Recognition of Object Features during Active Electrolocation in a Weakly Electric Fish. *J. Exp. Biology*, 210(17):3082–3095, 2007.

[16] Wilfried Westheide and Reinhard Rieger. *Spezielle Zoologie. Teil 2: Wirbel- oder Schädeltiere.* Elsevier, 2003.