INTEGRATED MODELING ENVIRONMENT “VIRTUAL DISCHARGE” FOR GAS DISCHARGE STUDY

A. G. SHISHKIN[1] AND S. V. STEPANOV[2]*

[1,2] Dept. of Computational Mathematics & Cybernetics, Moscow State University, Vorobjovy Gory, 119992, Moscow, Russia
* sergey.v.stepanov@gmail.com

ABSTRACT

Nowadays the gas discharges are well-studied and there are a number of application codes for plasma simulation. The key problem is that such codes are often proprietary and poorly documented. Another problem is that such codes cannot be used together due to different input/output data formats. If some plasma simulation code is not an open source one, the only way to reuse its numerical results is to write an application or script which will convert the output data to the proper format.

“Virtual Discharge” modeling environment was developed to allow users to build complicated modeling cases only with several mouse clicks by the help of simply-to-use graphical user interface. Having a chance to configure different code calculation chains and to define data convertors, a user can easily reuse third-party plasma simulation codes saving his time.

“Virtual Discharge” is developed with Java so it’s a cross-platform application. Being integrated with ScopeShell data analysis and visualization integrated shell [1] and Tadisys task distribution system [2] it is a powerful tool for complicated plasma simulation cases, which also supports distributed computations.

1. INTRODUCTION

Working on numerical experiments and code implementation, most users encounter the same difficulties - they need to implement custom input-output data convertors, setup and configure visualization software, increase calculation performance and throughput (fig. 1). In case of multi-model and multi-parameter experiments the problem becomes much more complicated as a user has to define input parameter arrays per setup and run all the setups sequentially. The more setups and configurations defined, the more difficult monitoring and analysis tasks are. It should be outlined that most numerical codes, that may be coupled with user’s self-developed codes, don’t provide any graphical user interface (GUI) and monitoring functionality.

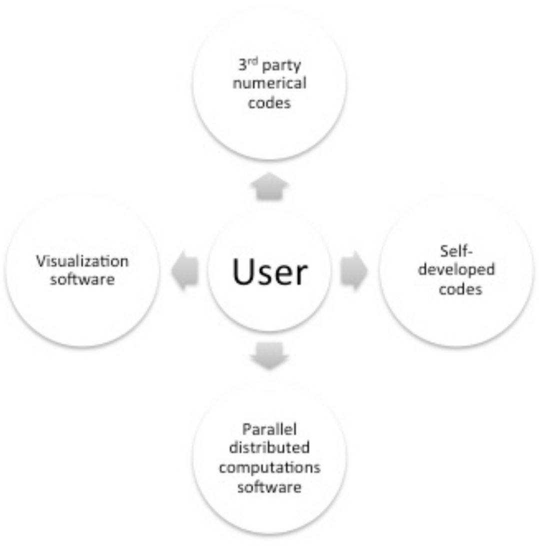

3rd party numerical codes
Visualization software
User
Self-developed codes
Parallel distributed computations software

Fig. 1 User interaction with modelling software

On the other hand, some modeling software solutions and codes are proprietary ones built with an “all-in-one” paradigm.

The problem is that such software solutions are not customizable at all, a user cannot couple it with some other codes to perform extra calculations and extract some data. At the same time, some of them are presented only to a few platforms and operating systems (OS), so it’s rather difficult to use it.

2. “VIRTUAL DISCHAGE” MODELING ENVIRONMENT

To overcome all the problems mentioned above the “Virtual discharge” solution was designed with the following key requirements and principles:

- Allow users to couple third-party numerical codes and solutions without custom data convertors development;
- Implement an easy-to-use graphical user interface (GUI);
- Support data monitoring, analysis and visualization;
- Support distributed parallel computations.

The aim is to develop a solution allowing to couple all the components together and make its usage transparent for a user.

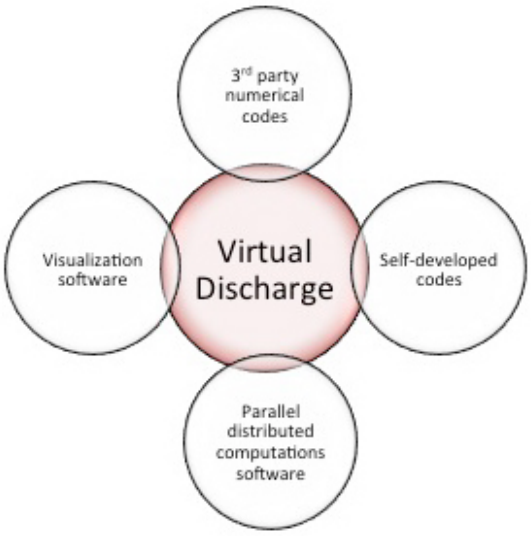
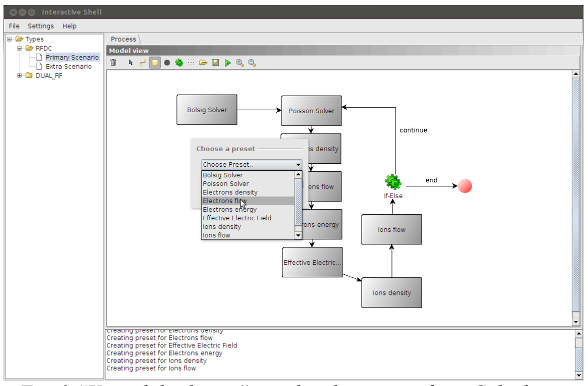

3rd party numerical codes
Visualization software
Virtual Discharge
Self-developed codes
Parallel distributed computations software

Fig. 2 “Virtual discharge” coupling as a main principle

At the same time, the described approach should have easy-to-use intuitive UI.

“Virtual discharge” developed in Java so it’s a platform-independent software that may be used in any environment. Third-party software and codes are integrated via special modules – computational blocks. Once integrated and configured, such blocks can be easily reused and shared with other users with the help of ImpEx module (Import-Export).

The main idea is to allow user to setup computational experiment just in "two mouse clicks". Having predefined computational blocks and built-in modules, one can easily setup computational chains with a few mouse clicks in “drag-n-drop” mode via intuitive graphical user interface. A number of presets – a set of parameters and input data – can be defined for each computational block. A user has an ability to simply drag and drop needed blocks and choose a proper preset.



```
Creating preset for Electrons flow
Creating preset for Effective Electric Field
Creating preset for Electrons energy
Creating preset for Ions density
Creating preset for Ions flow
```

Fig. 3 “Virtual discharge” graphical user interface. Calculation chains.

It should be mentioned, that a number of computation chains can be defined and grouped in a project. That approach helps user to easily work with multi-modal and multi-parameter tasks. Each computational chain can be easily cloned or exported with the ImpEx module, so all the defined chains can be easily shared with other users or stored in any version control system (VCS) since it is a simple XML file.

Calculation blocks sequence is defined in “drag-n-drop” mode just with a couple of mouse clicks connecting blocks with a “one-way” line. If a user needs to define an iterative calculation process, then it’s necessary to configure a “stop condition” with the help of a predefined block.

Each computational block is coupled with ScopeShell – an environment for data analysis, processing and visualization [1]. Defining file masks, convertors and formats, it’s easy to process a large set of files, that may be produced as an output data by any computational code (fig. 3). At the same time, ScopeShell allows user to preprocess input data with a set of commands if needed.

ScopeShell can be used in two modes – embedded or standalone ones. Embedded mode allows to use all the functions of the software just in the same environment. At the same time, it requires to store all the configuration files locally and it uses local resources, so, such approach may lead to a significant performance degradation in case of a large computation chain with a number of blocks with local ScopeShell instances. Standalone mode allows user to configure and run a ScopeShell instance remotely, not only at a local machine, and communicate with an instance via special protocol. Using such approach, it's possible to significantly decrease hardware resource usage. On the other hand, if a large set of input and output data is processed, it may take much time to transmit all the data from one machine to another over a network, so, a network throughput becomes important. One of the mentioned approaches should be chosen for a better performance and usability depending on the data being processed and infrastructure configuration.

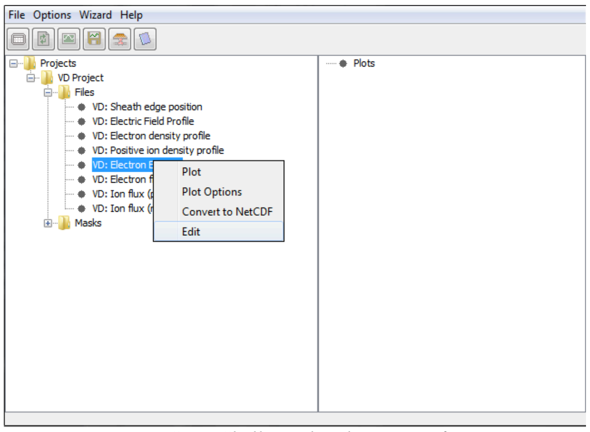


Fig. 4 ScopeShell graphical user interface.

Data visualization is performed with a number of third-party opensource packets. By default, "Virtual Discharge" is integrated with the GnuPlot packet [3], but other visualization tools can be easily integrated with the "Virtual Discharge" environment with the help of the integration module.

Each computational block can be calculated both locally and remotely. When a calculation chain is large, most of its computational blocks are supposed to be calculated remotely to significantly increase overall computational experiment performance. It should be underlined, that remote calculation should be used wisely. For example, if a computational task is simple enough, its possible that data transmission and remote setup overhead will be significantly more noticeable than the task calculation time. The best practice to use remote approach is to split computational blocks, that can be calculated in a parallel mode (or even concurrently), and run its calculations on different servers. In such case, performance will improve close to a linear mode.

Distributed calculations are supported via Tadisys software [2]. Tadisys is a client-server application that implements two key approaches to support distributed calculations. The first one is a custom solution to run computational tasks remotely. All the data transformed via sftp protocol to remote servers, after that, startup scripts are called. User has an ability to monitor remotely running processes and get results as soon as calculations are done. The second solution is built atop of the Hadoop project [4] and its subprojects. It implements BSP (Bulk Synchronous Parallel) computing paradigm, which allows run remote tasks easier and more effectively.

When calculation chains are defined and configured, one can start its calculation in GUI mode. While up and running, a user can monitor the ongoing process or processes, extract current results and visualize it.

3. GAS DISCHARGE STUDY

For simplicity, one case of studying dual-frequency gas discharge based on a two-dimensional fluid model is described.

Modelling task can be presented as a sequence of computational blocks – first, it's necessary to obtain electron transport parameters (ETP), electric field, electron density and its flux etc (fig. 5). All the calculation process can be easily defined and configured in the "Virtual Discharge" environment.

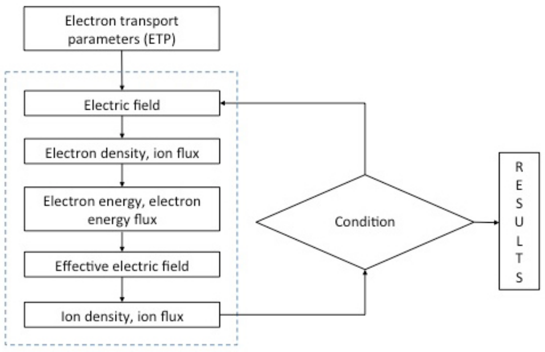
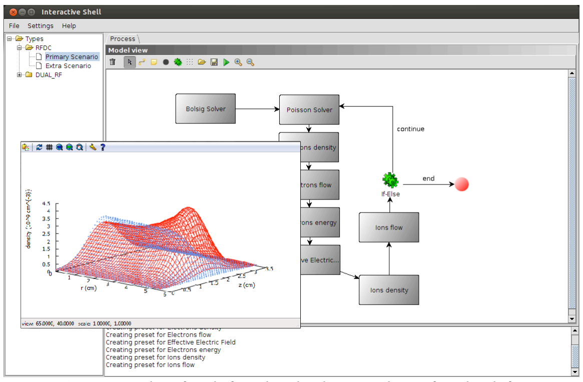

Electron transport parameters (ETP)
Electric field
Electron density, ion flux
Electron energy, electron energy flux
Effective electric field
Ion density, ion flux
Condition
RESULTS

Fig. 5 Dual-frequency discharge numerical experiment sequence

Electron Transport Parameters (ETP) are obtained with Bolsig+ solver [5], which is widely used for plasma discharge study. “Virtual Discharge” is integrated with Bolsig+ solver out-of-the-box, so ETP parameters can be easily obtained. When ETP parameters are obtained, the main iterative calculation process can be started. All the key parameters, such as electron and positive and negative ions densities, fluxes and electron energy are obtained with a self-developed code. While a calculation process is running, a user can obtain all the results, visualize it and monitor the ongoing calculation task progress (fig. 6).



```
Creating preset for Electrons flow
Creating preset for Effective Electric Field
Creating preset for Electrons energy
Creating preset for Ions density
Creating preset for Ions flow
```

Fig. 6 An example of a defined calculation chain for dual-frequency discharge study

“Virtual Discharge” environment is extremely useful when working with multi-model and multi-parameter tasks. Once the calculation chain is defined, it can be cloned, after that, it is possible to change some input data and parameters and run all the chains at once. When all the tasks are done, one can visualize, analyse and process all the results and compare it. Such approach gives a number of important parameters – for example, the dependency of key plasma parameters on a reactor configuration can be studied.

4. CONCLUSION

As shown above, “Virtual Discharge” modelling software may significantly increase overall numerical experiments performance. With the help of an easy-to-use graphical user interface, integration with data analysis, processing and visualization software and implementation of parallel distribution calculations it is possible to perform numerical studies in a more effective way.

5. REFERENCES

[1] D. P. Kostomarov, F. S. Zaitsev, A. G. Shishkin, and S. V. Stepanov. “The ScopeShell Graphic Interface: Support for Computational Experiments and Data Visualization”. Moscow University Computational Mathematics and Cybernetics. Vol. 34. Pp. 191-197. 2010.
[2] D. P. Kostomarov, F. S. Zaitsev, A. G. Shishkin, S. V. Stepanov, and E. P. Suchkov. “Automating Computations in the Virtual Tokamak Software System”. Moscow University Computational Mathematics and Cybernetics. Vol. 36. Pp. 165-168. 2012.
[3] GnuPlot website http://www.gnuplot.info
[4] Hadoop website http://hadoop.apache.org
[5] Bolsig+ Solver website http://www.bolsig.laplace.univ-tlse.fr