

Grid-Supported Simulation of Vapour-Liquid Equilibria with GridSFEA

I.L. Muntean, E. Elts, M. Buchholz, and H.-J. Bungartz

Technische Universität München, Dept. of Informatics,
Scientific Computing in Computer Science,
Boltzmannstr. 3, 85748 Garching, Germany
{muntean,elts,buchholm,bungartz}@in.tum.de
<http://www5.in.tum.de>

Abstract. In order to benefit from grid computing, software applications in CSE often need to be substantially modified or rewritten to a large extent. To reduce the required grid know-how and effort the computational scientist (end user and software developer) needs for this task, we developed a framework for engineering simulations in grid environments (GridSFEA). This paper presents two novel features of GridSFEA: the integrated support for parameter investigations and the controlled execution of long-running simulations in grids. They allow the grid enabling of CSE applications with minimal or even without changes of their source code. Furthermore, the overhead for working in grid environments introduced by our approach, compared to working on classical HPC platforms, is very low. We provide two examples of using GridSFEA for performing vapour-liquid equilibria (VLE) simulations using Molecular Dynamics and Monte Carlo methods. To develop VLE models, parameter investigations are carried out. Large VLE scenarios are computed over a long time, to create test cases for the development of HPC software.

Keywords: grid application, grid services, HPC molecular simulation, vapour-liquid equilibria, two-centre Lennard-Jones, polar fluid.

1 Introduction

Recently, grid computing environments [1] evolved from research and experimental status towards production systems, providing scientists with access to large aggregated computing and storage resources. Although the scope of such environments is limited – scenarios of capability computing typically requiring high performance resources at one place – esp. capacity computing applications (e.g., Monte Carlo (MC) simulations, parameter studies) offer a huge potential for the grid. Despite this attractiveness (due to mechanisms for security or access to distributed resources e.g.) of computing grids, they still remain underutilised and underexploited by the computational science and engineering (CSE) community. One reason for this is esp. the tedious development of grid applications and the grid middleware know-how necessary for the scientist (developer) to master.

There is ongoing research to bridge the gap between CSE applications and the grid. On the one hand, development toolkits and libraries for grid applications have recently been embraced by the grid community. Such examples are Java CoG Kit [2], DRMAA [3], or JavaGAT [4]. They provide high-level API and tools for interacting with the grid, simplifying the grid software development a lot. Nevertheless, with this approach the code of existing CSE simulation software often still needs to be extended to get advantages from grids. On the other hand, frameworks for grid applications have emerged, mainly based on the above toolkits and libraries. Two representatives of such programs can be found in [5] and [6]. Since they both are tailored to their specific field, earth sciences and image processing, resp. it is hard to reuse them for other applications. Another approach is represented by complex grid tools, covering the entire range from applications to grid resource managers and brokers. Such examples are Nimrod/G [7], Condor-G[8], or GrADS[9]. Although they are nearly complete solutions for grid-enabling applications, existing simulation codes (still) need to be rewritten, to use provided features of these tools. Such an example is GrADS, where the migration of a computational job is possible only after the re-implementation of application-level checkpointing in the simulation code using the SRS library.

We present an approach suitable for enabling various CSE software applications without the need to adapt their code. The framework we introduce here – GridSFEA (Grid-based Simulation Framework for Engineering Applications) [10] – reuses the know-how about requirements of CSE applications gathered in previous experiences with frameworks for engineering processes [11,12]. It handles completely the interaction with the grid middleware Globus Toolkit 4 (GT4) [13] by using Java CoG Kit, DRMAA, an application portal based on OGCE2 [14], grid services, etc. The integration of an application is done by simple wrappers. So far, we have used GridSFEA for the development of classification algorithms based on sparse grids [15] and for computational fluid dynamics simulations. Here, we highlight its application to molecular simulation scenarios.

In this paper, we focus on two *easy-to-use* mechanisms available within GridSFEA. The first one refers to application-independent parameter studies, while the second one applies to checkpoint-based migration of long-running simulations in the grid. We evaluate these mechanisms in the context of Molecular Dynamics and Monte Carlo simulations of vapour-liquid equilibria (VLE) of pure substances and mixtures: First, we use GridSFEA for carrying out parameter space investigations needed to develop VLE models. Second, we show how our framework can be easily employed to manage the long-running execution of large VLE scenarios with millions of molecules in the grid. Those scenarios are needed to create test cases to be used in the development of HPC software.

The framework is briefly introduced in Sect. 2, with focus on the two mechanisms mentioned above. Section 3 presents computational aspects of the development of models and HPC simulation software for VLE. We discuss the numerical experiments carried out with GridSFEA in Sect. 4 and conclude with Sect. 5.

2 GridSFEA - A Simulation Framework for Engineering Applications in Grid

The GridSFEA framework aims at providing a viable complement to the classical way of computing HPC simulations. It enables various CSE software applications to easily run in grid environments with minimal or even without changes of their original source code. Within GridSFEA, a set of common functionalities, such as authentication, file transfer, or job coordination, are shared among all simulation applications. Additionally, the framework comprises enhancements specific to user scenarios.

2.1 Organisation of GridSFEA

In the realisation of GridSFEA, we reuse available (grid) community toolkits and libraries. GridSFEA currently works with GT4, which is the de-facto standard middleware for grid environments. The main components of our framework are shown in Fig. 1.

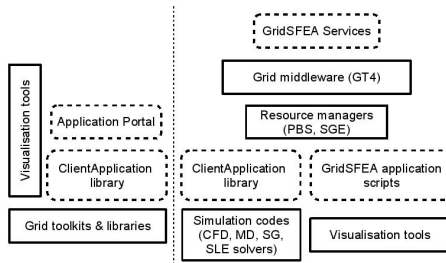


Fig. 1. The architecture of GridSFEA: tools running in the user (left) and the grid (right) environment, resp

GridSFEA Services are a collection of grid services for the management of simulations, of their data, and for the visualisation of the simulation results. They are developed based on WSRF and run on grid servers with GT4 installed. The **application portal** hosts web applications based on portlets, for interfacing the scientist with the simulation programs to compute with. Additionally, it reuses portlets from OGCE2 for basic interactions with the grid, such as portal authentication and file transfers. **Application scripts** are wrappers, adaptors, and utility programs (such as generators) written for integrating the various simulation programs into the framework. They typically run on the resources where the simulations are computed. The portal and the application scripts interact with the grid via the GridSFEA **ClientApplication library**, which is based on CoG Kit and DRMAA. The library acts as a glue for the framework. GridSFEA interacts with various categories of tools (see Fig. 1).

2.2 Enhanced Support for Parameter Space Investigations

The portal of our framework comprises two portlets designed for parameter studies. One is the SparseGrids portlet, a web application tailored to the requirements of the development process of classification algorithms [15]. The second one is the ParameterInvestigation portlet, that allows more general parameter studies. It uses a flexible and extensible data model for the parameter definition, based on XML Schema. Thus, it allows an intuitive specification of simulation parameters, such as enumerations, sequences, and combinations of the latter ones. Moreover, it has its own engine for parameter generation, passes parameters and other input data to a user-specified simulation program, automatically submits jobs to the grid and manages them. JSR128 portlets provided by a third-party can coexist with our portlets and mutually complement their functionality.

The parameter specification API and the generation engine are part of the ClientApplication library. Thus, they are used from both portal and command line-based tools of the framework. Furthermore, one can easily write adaptors or plugins for other parameter study tools and connect them to GridSFEA.

2.3 Long Running Simulations in Grid Environments

GridSFEA provides a job migration mechanism based on checkpoints that aims to automate the execution of long-running simulations in grids. Thus, it manages the simulation data for jobs computed on different grid resources at various locations. It aims to reduce the idle time of the continuation jobs in the queues. This is achieved by the GridSFEA services and the ClientApplication library. They record metadata about the input, output, result, and checkpoint files of each job. An application wrapper defines the interface between the framework and the program to be run in grid. This way, the simulation code remains unaffected.

To a regular user job we add new operations before and after the execution of the simulation. The preprocessing operations are the retrieval of the checkpoint information from the GridSFEA services and the transfer of the checkpoint and of other input data. The registration of the new checkpoint is done at the end of the job. Furthermore, we provide the computational scientist, both user and application developer, the possibility to specify or to plug-in post processing scripts to be performed on the simulation results. So far, we plugged in the new job the generation of a preview of the results using VMD [16] for MD and Paraview for CFD simulations. They are used as batch renderers. Similar postprocessing tasks can be integrated in GridSFEA with low programming effort.

The prerequisites to enable a simulation program to use the migration mechanism available in GridSFEA are: non-interactive use (requested by the batch execution mode), checkpoint-based resuming of computations, and a simple wrapper for specifying checkpoint and result files or for integrating postprocessing tasks. Furthermore, the application has to be installed in grid in advance.

3 VLE Simulations

For many technologically relevant tasks in chemical and thermal process engineering knowledge of vapour-liquid equilibria (VLE) has turned out to be necessary. Among the different ways to study VLE, such as experimental and theoretical approaches, molecular simulation plays an important role [17,18]. It is a reliable tool for predicting phase equilibria and the best way for gaining insights into the connection between phase behaviour and molecular interactions.

Here we focus on two distinct challenges from the field of VLE, both well suited to be tackled with the help of a grid. In Sect. 3.1, we describe a method for a fast elaboration of molecular models which involves extensive parameter studies. By using GridSFEA, we can profit to a great extent from the ideal possibilities a grid provides for such studies. The second example (Sect. 3.2) deals with the evaluation of parallelisation strategies for a wide range of large VLE simulations. Therefore, we need to run a moderate number of long-running HPC jobs. GridSFEA's job-migration facilities helps us to avoid to explicitly organise the resources for those jobs.

3.1 Elaboration of the 2CLJQ Fluid Model for VLE

The search for an appropriate interaction model for a given fluid is usually a time consuming process. In this section, we follow a new route to develop interaction models, proposed in [19], that allows fast adjustments of model parameters to experimental data *for a given class of pure fluids* and considerably reduces the time required for the development of new molecular models. We consider this model elaboration technique for the example of the two-centre Lennard-Jones plus pointquadrupole model (2CLJQ).

The idea is to study the thermodynamic properties of the 2CLJQ model fluid systematically and in detail over a wide range of model parameters with the help of GridSFEA. Using reduced coordinates, for the 2CLJQ fluid with fixed angle θ , only 2 parameters have to be varied: the LJ centre-centre distance¹ L^* and the pointquadrupole strength Q^{*2} .

Based on the results from the parameter study, it is straightforward to adjust the molecular interaction parameters of the 2CLJQ fluid to experimental data of real quadrupolar fluids. Properties like the critical value of the temperature e.g. are available as functions of the molecular interaction parameters [19]. Hence, the development of the molecular interaction model for a given substance is not more difficult than the adjustment of parameters of thermodynamic models. Thus, the present simulations are a reliable basis for adjustments of the model parameters Q^{*2} and L^* to experimental VLE data of real fluids.

3.2 Development and Evaluation of Parallelisation Strategies

For VLE simulations with large numbers of particles, the development and evaluation of parallelisation techniques is an important issue.

¹ All values with * are reduced values (transformed to a dimensionless form).

One typical property of VLE simulations is a very heterogeneous particle distribution. This necessitates the use of efficient parallelisation and load-balancing algorithms. We have developed a MD software for the simulation of NVT-ensembles [20] with a large number of particles. The design of our software allows us to easily switch between different parallelisation schemes [21], but we need to find a way of comparing those schemes. Therefore, we first have to investigate the possible simulation scenarios.

For the initial configuration of a NVT simulation, the **N**umber of particles, the **V**olume, and the **T**emperature have to be specified. In our case, we use a face-centred grid to set the particles initial positions. The particle distribution that evolves depends on the temperature and on the density (particles per volume). Basically each combination of temperature and density yields a different distribution of the particles in the domain. To cover most of those distributions for the evaluation of the parallelisation techniques, we should examine at least five different densities and five different temperatures, which leads to 25 scenarios. As we use $N = 5 \cdot 10^6$ particles for each scenario and simulate it for 10^5 time steps, the simulation of one of those scenarios on 64 processors takes more than 24 hours. Getting processing time for a long-running job is quite hard and obviously it is harder to get processing time for 25 of those jobs. But it is much easier to get several shorter blocks of processing time. Therefore, the simulation of all scenarios demands the possibility of job migration and support for parameter studies to reduce the administrative work for the user. By using GridSFEA, we get those benefits without having to interact during the simulation.

4 Case Studies

For each of the two challenges presented in Sect. 3, we describe – after introducing the experiments carried out – how to run the simulations with GridSFEA and evaluate the results with a special focus on the time overhead needed for the framework.

4.1 Systematic Investigation of the VLE of the 2CLJQ Model Fluid

Setup of the experiments. The systematic investigation of the VLE of the 2CLJQ model fluid was performed for a range of quadrupolar momentum $0 \leq Q^{*2} \leq 4$ and of elongation $0 \leq L^* \leq 0.8$, with steps 1 and 0.2, resp. Temperatures investigated ranged from 0.55% to 0.95% of the critical temperature. Combining these values, 125 input files were generated with GridSFEA.

To obtain the VLE data the Grand Equilibrium method [18] was used. Widom’s insertion method [22] was used to calculate the chemical potential in the liquid. The data obtained from liquid simulation was further used as phase equilibrium conditions for the vapour phase.

Results. The simulations have been computed with the parallel MD code ms2 [23]. They were carried out within the frame of the InGrid project [24]. GridSFEA services have been deployed on a GT4 container (version 4.0.3) running on

the grid node gt4.dgrid.hlr.de. The user provided in the ParameterInvestigation portlet the XML description of the parameters Q^{*2} , L^* , and the temperature factor, together with the name of the generator for ms2 input files. The portlet generated all parameter combinations and for each of them submitted a batch job to the grid. Thus, the user does not have to care about all the tedious administration overhead necessary to create jobs for the different parameter combinations, all that work is done by GridSFEA.

Table 1. Time for the submission with GridSFEA of a trivial job (hostname) and of the VLE simulation tasks: liquid phase (MD) and vapour phase (MC)

	Execution time (min)	Submission time (min)
trivial job	10^{-4}	0.27
liquid phase	42 – 254	0.27
vapour phase	41 – 103	0.27

In Table 1, we show the submission time for the VLE simulation tasks (liquid and vapour phases) together with the range of their execution time. Each of the generated jobs ran on sets of four processors. For both parts of the simulation, the submission time with GridSFEA was approximately the same. This time is measured from the moment the submission command was issued until the user program begins to run. It includes internal operations in the framework, such as file transfer, job submission, results retrieval, logging etc. Thus, the submission time is a measure of the time overhead introduced by GridSFEA at the execution of a user simulation on HPC resources. The overhead is independent from the number of employed processors and from the duration of the individual jobs. Furthermore, it has the same value as for the submission of a trivial job that returns the name of the system it runs on (hostname).

Figure 2 illustrates the strong influence of both the elongation and the quadrupolar momentum on the 2CLJQ VLE data for $Q^{*2} = 1$ and $Q^{*2} = 4$. Increasing the elongation or the quadrupolar momentum strongly influences the shape of the density coexistence curve. With the data obtained from this study, it

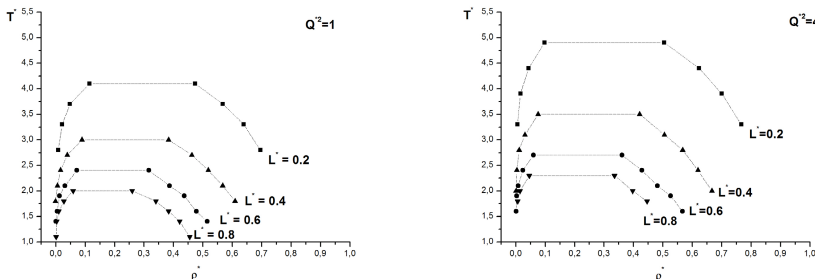


Fig. 2. Vapour-liquid coexistence of the 2CLJQ fluid

is straightforward to adjust the molecular interaction parameters of the 2CLJQ model fluid to experimental data of real quadrupole fluid and to develop molecular interaction model for various substances [25].

4.2 Examination of the Time Requirements for Large Scenarios

Scenarios and HPC environment. In Sect. 3.2, we motivated the examination of different scenarios. Now we look at some results for three selected scenarios with the densities $\rho^* = 0.15$, $\rho^* = 0.3$ and $\rho^* = 0.6$. All scenarios contain $N = 5 \cdot 10^6$ particles and have a temperature of $T^* = 0.85$. The simulations were done on a Linux Cluster with an InfiniBand 4x network. We used 16 nodes, each having 8 GB RAM and four Opteron 850 processors with 2.4 GHz. The parallelisation scheme used here is a domain decomposition without any load balancing.

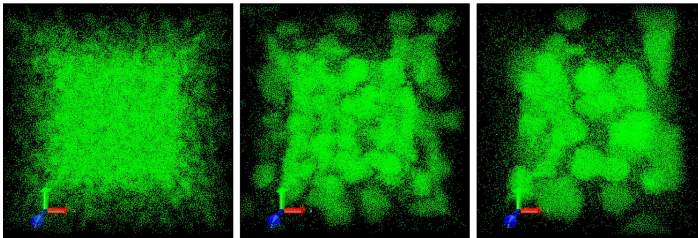


Fig. 3. Simulation of 5 million particles at $T^* = 0.85$ and $\rho^* = 0.15$ after 10,000, 50,000 and 100,000 time steps (visualisation with VMD [16])

Results. Fig. 3 shows the visualisation for the first scenario ($T^* = 0.85$, $\rho^* = 0.15$) after 10,000, 50,000 and 100,000 time steps. To get a clearer view, only a cubic section with one third of the original side length is shown. As the distribution changes significantly during the whole simulation, 100,000 time steps are needed. In the beginning, the particles in each scenario are uniformly distributed. As the imbalance increases during the simulation, the processing time increases too and remains instationary during the first 100,000 time steps. This indicates that the distribution is continuously changing throughout the simulation.

Table 2. Time results for the migration with GridSFEA of the VLE simulation with $\rho^* = 0.30$, $T^* = 0.85$

Scenario	Setup time (GridSFEA)	File transfer time	Computation time
first job	15 s	16 s	10.6 h
cont. job	38 s	20 s	11.4 h

We computed sets of jobs, with 10,000 time steps per job. Each job checkpointed its final state to a file and registered it with the GridSFEA services. At startup, each job retrieved the checkpoint information from the grid services and transferred it to the local machine. Table 2 shows the overhead introduced by the migration mechanism of our framework for the scenario with $N = 5 \cdot 10^6$ molecules at $T^* = 0.85$ and $\rho^* = 0.3$, compared to the effective simulation time.

Discussion. In both experiments introduced in this section, the grid enabling of the simulation programs was carried out without any changes of their source code. To use the migration mechanism in the second experiment, a simple wrapper for the MD application was added to the user space installation of GridSFEA. Similarly, further engineering simulation tasks can use this mechanism.

The overhead for automatically setting up a simulation job in GridSFEA is very small, below one minute. This makes our checkpoint-based migration suitable for typical computational engineering jobs, with execution times ranging from few hours to hundreds or thousands of hours. Nevertheless, for jobs with relatively short duration (several minutes), the migration mechanism from GridSFEA is not useful any more.

5 Conclusions

Using *state-of-the-art* grid technologies and tools, we developed the GridSFEA framework for performing engineering simulations in grid environments. With this work, we enabled the computational scientist to benefit from GT4-based grid environments for concrete tasks such as modeling VLE processes and preparing realistic simulation data for test cases to be employed in the development of HPC software for VLE. Our approach achieves the integration of application scenarios in the framework by means of wrappers, therefore without modifying the source code of the respective CSE program. The two features of GridSFEA that we introduced in this paper – support for parameter space investigations and for long running simulations – are not only useful for the case studies discussed here, but also for other similar CSE scenarios specific to capacity computing. As future work, we plan the improvement of the automated migration mechanism in GridSFEA and an opening of the framework to other grid middleware (e.g. Unicore). Furthermore, we intend to increase the number and type of CSE scenarios and applications handled by GridSFEA.

Acknowledgements. We thank HLRS and the German D-Grid initiative for providing within the InGrid project the computing resources employed for accomplishing this research. Furthermore, we acknowledge the collaboration on the VLE topic with B. Eckl and Dr. J. Vrabec at the Institute of Thermodynamics and Thermal Process Engineering, Universität Stuttgart.

References

1. Foster, I., Kesselman, C.: *The Grid: Blueprint for a New Computing Infrastructure*. Morgan Kaufmann, San Francisco (2005)
2. Thomas, M., et al.: Grid Portal Architectures for Scientific Applications. *Journal of Physics* 16, 596–600 (2005)
3. Tröger, P., Rajic, H., Haas, A., Domagalski, P.: Standardization of an API for Distributed Resource Management Systems. In: *Proc. 7th IEEE Int. Symp. on Cluster Computing and the Grid (CCGrid 2007)*, pp. 619–626 (2007)
4. Nieuwpoort, R.V., Kielmann, T., Bal, H.E.: User-Friendly and Reliable Grid Computing Based on Imperfect Middleware. In: *ACM, Supercomputing (SC 2007)* (2007)
5. Price, A.R., et al.: Optimization of integrated Earth System Model components using Grid-enabled data management and computation. *Concurrency Computat.: Pract. Exper.* 19, 153–165 (2007)
6. Jin, H., Zheng, R., Zhang, Q., Li, Y.: Components and workflow based Grid programming environment for integrated image-processing applications. *Concurrency Computat.: Pract. Exper.* 18, 1857–1869 (2006)
7. Abramson, D., Giddy, J., Kotler, L.: High performance parametric modeling with nimrod/g: Killer application for the global grid? In: *IPDPS 2000: Proc. of the 14th Int. Symp. on Parallel and Distributed Processing*. IEEE Computer Society Press, Washington, DC (2000)
8. Frey, J., et al.: Condor-G: A computation management agent for multi-institutional grids. *Cluster Computing* 5, 237–246 (2002)
9. Vadhiyar, S.S., Dongarra, J.J.: Self adaptivity in grid computing: Research articles. *Concurr. Comput.: Pract. Exper.* 17(2-4), 235–257 (2005)
10. Muntean, I.L.: GridSFEA - Grid-based Simulation Framework for Engineering Applications, <http://www5.in.tum.de/forschung/grid/gridsfea/>
11. Mundani, R.P., Bungartz, H.J., Niggl, A., Rank, E.: Embedding, Organisation, and Control of Simulation Processes in an Octree-Based CSCW Framework. In: *Proc. 11th Int. Conf. on Comp. in Civil and Building Eng.*, pp. 3208–3215 (2006)
12. Mundani, R.P., et al.: Applying Grid Techniques to an Octree-Based CSCW Framework. In: Di Martino, B., Kranzlmüller, D., Dongarra, J. (eds.) *EuroPVM/MPI 2005*. LNCS, vol. 3666, pp. 504–511. Springer, Heidelberg (2005)
13. Foster, I.: Globus Toolkit Version 4: Software for Service-Oriented Systems. In: Jin, H., Reed, D., Jiang, W. (eds.) *NPC 2005*. LNCS, vol. 3779, pp. 2–13. Springer, Heidelberg (2005)
14. OGCE: Open Grid Computing Environments: www.collab-ogce.org/ogce2/
15. Pflüger, D., Muntean, I.L., Bungartz, H.J.: Adaptive Sparse Grid Classification Using Grid Environments. In: Shi, Y., van Albada, G.D., Dongarra, J., Sloot, P.M.A. (eds.) *ICCS 2007*. LNCS, vol. 4487, pp. 708–715. Springer, Heidelberg (2007)
16. Humphrey, W., Dalke, A., Schulten, K.: VMD – Visual Molecular Dynamics. *J. Molecular Graphics* (1996)
17. Serbanovic, S., et al.: Vapour-liquid equilibria of the OPLS model for the binary systems of alkanes and alkanes + alcohols. *J. Serb. Chem. Soc* (2005)
18. Vrabec, J., Hasse, H.: Grand Equilibrium: vapour-liquid equilibria by a new molecular simulation method. *Molecular Physics* (2002)
19. Stoll, J., Vrabec, J., Hasse, H., Fischer, J.: Comprehensive study of the vapour-liquid equilibria of the two-centre Lennard-Jones plus point quadrupole fluid. *Fluid Phase Equilibria* (2001)

20. Allen, M.P., Tildesley, D.J.: Computer Simulation of Liquids. Oxford University Press, USA (1989)
21. Bernreuther, M., Buchholz, M., Bungartz, H.J.: Aspects of a Parallel Molecular Dynamics Software for Nano-Fluidics. In: Parallel Computing: Architectures, Algorithms and Applications, Int. Conf. ParCo 2007 (2007)
22. Heyes, D.: Chemical Potential, Partial Enthalpy and Partial Volume of Mixtures by NPT Molecular Dynamics. Molecular Simulation (1992)
23. Eckl, B., Vrabec, J.: ms2 – MD simulation program. Inst. of Thermodynamics and Thermal Process Engineering, University of Stuttgart
24. INGRID: Inovative grid technology in engineering, www.ingrid-info.de/
25. Vrabec, J., Stoll, J., Hasse, H.: A set of molecular models for symmetric quadrupole fluids. Journal of Physical Chemistry B (2001)