

# COMPARING THE EFFICIENCY OF PARALLELISATION TECHNIQUES FOR SIMULATION TASKS BY MEANS OF THE ARGESIM BENCHMARKS FOR PARALLEL SIMULATION

René Fink<sup>1</sup>, Felix Breiteneker<sup>2</sup>, Gerhard Höfinger<sup>2</sup>, Sven Pawletta<sup>3</sup>

<sup>1</sup>IAV GmbH

Philipp-Müller-Straße 12, 23966 Wismar, Germany

<sup>2</sup>Vienna University of Technology

Inst. f. Analysis and Scientific Computing

Wiedner Hauptstraße 8-10, 1040 Vienna

<sup>3</sup>Wismar University of Applied Sciences

Research Group Computational Engineering and Automation

Philipp-Müller-Straße, PF 1210, 23952 Wismar, Germany

*rene.fink@iav.de* (René Fink)

## Abstract

In 1994, ARGESIM has set up the ARGESIM Comparison on Parallel Simulation Techniques (CP1). There, three test examples have been chosen to investigate the types of parallelisation techniques best suited to particular types of simulation tasks. From 1994 until 2003, 12 solutions of the CP1 benchmark have been published in *EUROSIM – Simulation News Europe*. Primarily, these contributions differed with respect to applied hardware and software and, of course, runtime results. Regarding the implemented solving algorithms and parallelisation strategies, there were almost no differences between published solutions. The new ARGESIM Benchmark on Parallel and Distributed Simulation (CP2) extends the previous comparison, addressing not only simulation software and predefined given algorithms, but also allowing use of different algorithms for solving the tasks and comparing different strategies for parallelisation or distribution of the tasks. In this paper, the 1994's ARGESIM Comparison on Parallel Simulation Techniques (CP1) is discussed and regarding *EUROSIM – Simulation News Europe* contributions are summarized. Out of this summarization, drawbacks of the CP1 showing the need of a new benchmark's design are discussed. Finally, the new 2006's ARGESIM Benchmark on Parallel and Distributed Simulation (CP2) is presented.

**Keywords:** Parallel Simulation, Distributed Simulation, ARGESIM Benchmark.

## Presenting Author's Biography

René Fink. The author is originated from Wismar University's Research Group Computational Engineering and Automation. From 2000 until 2006 he worked on a research project regarding SCE based Parallel Processing, also being the topic of his dissertation. Since 2007 he is employed as development engineer at the IAV GmbH, department Powertrain Mechatronics Gasoline Engines Systems.



## 1 Introduction

*EUROSIM – Simulation News Europe* (SNE) features a series of comparisons of simulation software. Simulation languages are compared in terms of their features for modelling and experimentation using simple and easily comprehensible models drawn from a number of different application areas.

In 1994, ARGESIM has set up the ARGESIM Comparison on Parallel Simulation Techniques (CP1, s. [1]). There, three test examples have been chosen to investigate the types of parallelisation techniques best suited to particular types of simulation tasks.

From 1994 until 2003, 12 solutions of the CP1 benchmark have been published in SNE. Primarily, these contributions differed with respect to applied hardware and software and, of course, runtime results. Regarding the implemented solving algorithms and parallelisation strategies, there were almost no differences between published solutions.

The new ARGESIM Benchmark on Parallel and Distributed Simulation (CP2, s. [14]) extends the previous comparison, addressing not only simulation software and predefined given algorithms, but also allowing use of different algorithms for solving the tasks and comparing different strategies for parallelisation or distribution of the tasks.

## 2 ARGESIM Comparison on Parallel Simulation Techniques (CP1)

In SNE issue 10 (March 1994), three benchmark problems for comparison of parallel simulation techniques have been published. All examples belong to the field of continuous simulation, so problem descriptions were given via differential or partial differential equations. Beside the equations, there were no instructions about sequential or parallel implementation algorithms. SNE readers were encouraged to implement sequential and parallel solutions and to publish results with comments regarding speedup factor, parallelisation strategies and implementation effort.

In the following sections, the CP1 benchmark examples are briefly described.

### 2.1 Monte Carlo Study

In the first example, the mean motion course of a second order mass-spring system should be ascertained. Therefore, the system has to be simulated for 1000 simulation runs with a uniformly distributed random damping factor (see Fig. 1).

### 2.2 Coupled Predator-Prey Population

In the second example, five coupled predator-prey systems have to be simulated in one simulation run only. System description was given by 10 first order differential equations. It was expected that no improvements may be found through parallelisation, but readers were encouraged to publish negative results, as well. Figure 2 shows the example's simulation results.

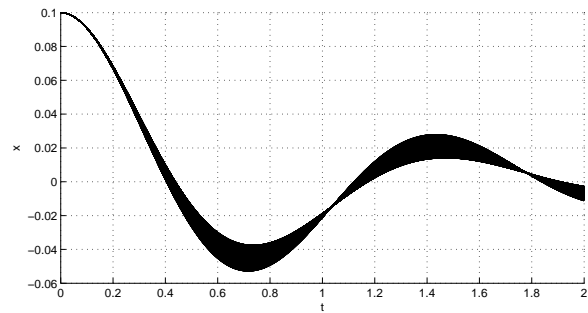


Fig. 1 Monte Carlo study – simulation results (1000 simulation runs)

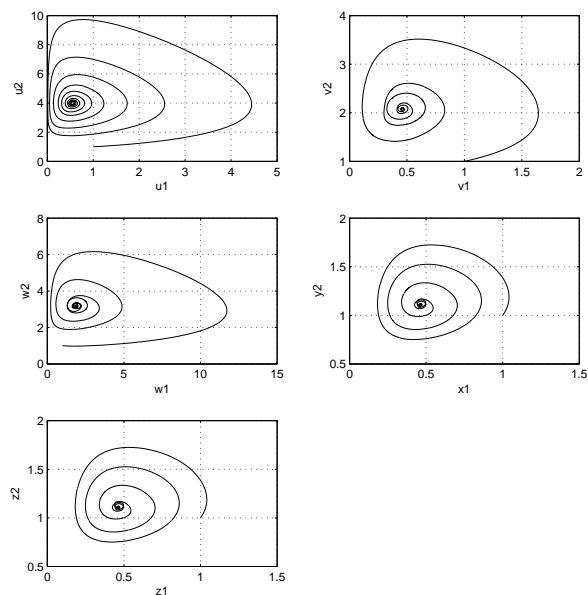


Fig. 2 Coupled predator-prey population – simulation results

### 2.3 Partial Differential Equation

In the third example, the motion of a swinging rope, being fixed on one end and forced on the other end, has to be simulated. System description was given by a partial differential equation. Furthermore, spatial discretisation following the method of lines was proposed for solving the problem numerically. Following this proposal, a system of  $2 * N$  ( $N$ : number of equidistant intervals) loosely coupled ordinary differential equations arose. Users were encouraged to use values of  $N = 800$  or higher. Figure 3 shows the simulation results of the third CP1 example at four points in time.

## 3 Contributions on CP1

From March 1994 until December 2003, 12 solutions for the CP1 benchmark were published in SNE, being summarised in Table 1. Published solutions show differences in applied hardware and software as well as in task distribution. Regarding applied solving algorithms and parallelisation strategies, solutions show no differ-

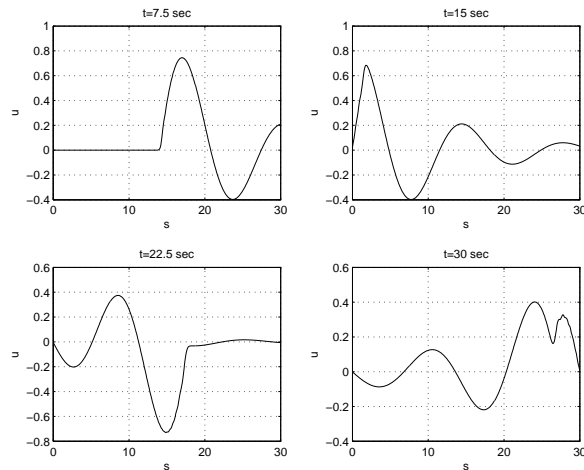


Fig. 3 Partial differential equation – simulation results

ences. In the following sections, discovered distinctions between CP1 solutions are presented. In this presentation, quantitative results are not of interest since hardware parameters and sequential implementation parameters vary between solutions. Thus, significant quantitative comparisons are difficult to realize. Instead of this, only qualitative attributes are considered in the following sections.

Notation	Date
FORTRAN, PVM [2]	March '94
SLIM [3]	July '94
Mosis/Cogent XTM [4]	July '94
FSIMUL-P [5]	Nov. '94
C, Linda [6]	Nov. '94
SIMUL_R PARALLEL [7]	Nov. '94
MATLAB, PSI [8]	March '95
ACSL [9]	July '96
ASCET-RS [10]	Nov. '96
Mosis/Parsytec [11]	Nov. '96
CPSS [12]	March '97
MATLAB, DP-Toolbox [13]	Dec. '03

Tab. 1 Published CP1 solutions in SNE

### 3.1 Distinctions in hardware

Regarding applied hardware, three parallel processing hardware types could be identified:

- Cluster of workstations
- Cluster of transputers
- Shared memory multiprocessors

*Cluster of workstations* consist out of multiple workstations with own physical memory and conventional microprocessors, being connected via Ethernet. *Cluster of transputers* consist out of multiple processors with own physical memory and special communication features. Interprocessor communication is established via

a switch or 2D grid. *Shared memory multiprocessors* consist out of multiple processors with shared physical memory. Interprocessor communication is established by the system bus. Since all processors are identically, they are also called *symmetric multiprocessors* (SMP).

Table 2 shows the hardware types of published CP1 solutions. It demonstrates that in most of CP1 solutions a cluster of transputers was used. Such clusters have been widespread small scale parallelisation platforms in the 90's but disappeared almost completely today. Furthermore, in only one CP1 contribution, shared memory multiprocessors are used.

Hardware type	Solution
Cluster of Workstations	FORTRAN, PVM FSIMUL-P MATLAB, PSI MATLAB, DP-Toolbox
Cluster of Transputers	SLIM Mosis/Cogent XTM C, Linda SIMUL_R PARALLEL ASCET-RS Mosis/Parsytec CPSS
SMP	ACSL

Tab. 2 Hardware types of published CP1 solutions

### 3.2 Distinctions in Software

Similarly to hardware structures, the applied CP1 solutions software can be divided into three types with different levels of specialisation:

- Programming language
- Simulation language
- Simulation system

In *programming language* based solutions, programs are implemented in a conventional programming language like Fortran or C. In *simulation language* based solutions, implementation happens in a specialised programming language, often dedicated for only continuous simulations. In *simulation system* based solutions, implementation and simulation runs take place within an interactive developing environment where models are described by linking graphical blocks.

In Table 3, CP1 solutions are assigned to the listed software types. It is shown that simulation languages and programming languages are dominating software types in CP1 contributions. An explanation of this fact can be the relatively low level of specialization, which eases access to parallel libraries.

### 3.3 Distinctions in task distribution

Beside hardware and software, CP1 solutions can also be differentiated by the way of task distribution in the first example, Monte Carlo study. In this example, the

Software type	Solution
Programming language	FORTRAN, PVM C, Linda MATLAB, PSI MATLAB, DP-Toolbox
Simulation language	SLIM Mosis/Cogent XTM SIMUL_R PARALLEL ACSL Mosis/Parsytec CPSS
Simulation system	FSIMUL-P ASCET-RS

Tab. 3 Software types of CP1 solutions

parallel program can be divided into independent tasks which only need to communicate at the beginning and end of task execution. In such program structures, the way of task distribution significantly influences the parallel program runtime.

In general, task distribution can be either statically or dynamically. In *static* task distribution, task to processor mapping is determined at the begin of a parallel program run, for example by assigning a fixed number of tasks to every processor. In *dynamic* task distribution, task to processor mapping is determined during parallel program execution, for example by assigning tasks on demand to idle processors.

In Table 4, the way of task distribution in CP1 Monte Carlo study is summarised. It is shown that in most of CP1 contributions, static task distribution is applied, where dynamic distribution and both ways of distribution are only implemented in one contribution, each.

Task distribution	Solution
statically	SLIM Mosis/Cogent XTM FSIMUL-P SIMUL_R PARALLEL MATLAB, PSI Mosis/Parsytec MATLAB, DP-Toolbox
dynamically	C, Linda CPSS
both	FORTRAN, PVM

Tab. 4 Task distribution in CP1 Monte Carlo study

#### 4 ARGESIM Benchmark on Parallel and Distributed Simulation (CP2)

The analysis of CP1 contributions has shown that individual solutions show little differences in solution algorithms and parallelisation strategies. Therefore, a new comparison has been designed addressing primarily these aspects. Again, the new comparison comprises three benchmark examples, partially derived from CP1.

In the following sections, CP2 benchmark examples are briefly described.

##### 4.1 Monte Carlo Study

This example extends the first CP1 example by enabling the use of an analytical solution beside the classical ODE solver solution. The expression for the analytical solution is in substance given as:

$$x(t, d) = Ae^{-dt} \sin(\omega t) \quad (1)$$

Where  $t$  is the time vector and  $d$  is the vector of damping factors. Since the analytical solution is given, parallelisation can take place in either time domain or damping factor domain or both. On the other hand, if an ODE solver is used, parallelisation can only take place in damping factor domain where independent simulation runs are distributed over parallel processes.

##### 4.2 Lattice-Boltzmann Simulation

This example addresses the lattice Boltzmann method (LBM) for fluid flows, which is widespread in parallel simulation domains today.

The purpose of the lattice Boltzmann method is to simulate fluid behaviours in complex geometries efficiently in parallel. Traditional fluid simulations, which are based on numerical solutions of the Navier Stokes equations have limited parallel potential and can hardly handle complex geometries.

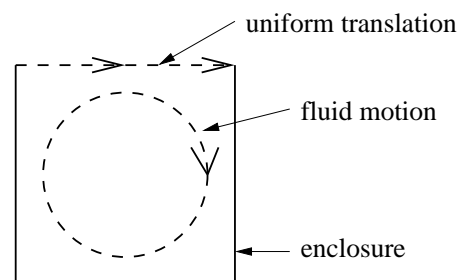


Fig. 4 Lattice-Boltzmann Simulation – problem description

In this example, the behaviour of an incompressible fluid in a square enclosure, driven by a constant stream on the top boundary has to be examined (see Figure 4). The task is, to simulate the cavity flow with lattice size 257x257 for a number of 350000 iterations. After this number of iterations, steady state is reached. Simulation results are shown in Figure 5.

In this example, parallelisation can only be achieved in form of a distributed model. This means, the model is split up into parts, being simulated by parallel processes which communicate at each iteration.

##### 4.3 Solution of a Partial Differential Equation

This example modifies and extends the third CP1 example “partial differential equation”.

Modification takes place in terms of boundary and initial conditions. In CP1 case study, space domain initial

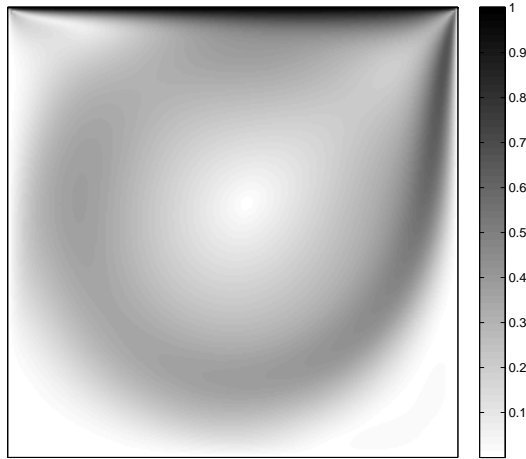


Fig. 5 Lattice-Boltzmann Simulation – simulation results (relative macroscopic velocity magnitude ( $u/u(0)$ ))

conditions are constant and time domain boundary conditions vary (see Fig. 3). In CP2 case study on the other hand, space domain initial conditions vary and time domain boundary conditions are constant. Simulation results of the CP2 case study are shown in figure 6.

Similarly to the first CP2 example, the usage of analytical solutions is permitted. An approximation for solving the problem analytically is given by the following Fourier series:

$$u(x, t) = \frac{8h}{\pi^2} \sum_{j=0}^{\infty} \frac{(-1)^j}{f^2} \sin\left(\frac{f\pi x}{L}\right) \cos\left(\frac{f\pi vt}{L}\right) \quad (2)$$

with  $f = 2j + 1$

In analogy to the first CP2 example, parallelisation of the analytical solution can happen in several domains. Since the sequential implementation of the analytical solution has to loop independently over  $t$ ,  $x$  and  $f$ , parallelisation can take place arbitrary in one of them. On the other hand, when solving the problem numerically by an explicit ODE solver, parallelisation can only happen in form of a distributed model as mentioned above.

## 5 Conclusion

In this paper, the 1994's ARGESIM Comparison on Parallel Simulation Techniques (CP1) was presented and regarding SNE solution contributions were summarised and compared.

It was shown, that CP1 solutions show differences in hardware, software and task distribution. Regarding hardware, three types of parallel processing hardware have been identified: clusters of workstations, clusters of transputers and shared memory multiprocessors. It was shown that clusters of transputers are the dominating parallel platform in CP1 contributions because of their relatively wide distribution in small scale parallel processing installations during the 90's.

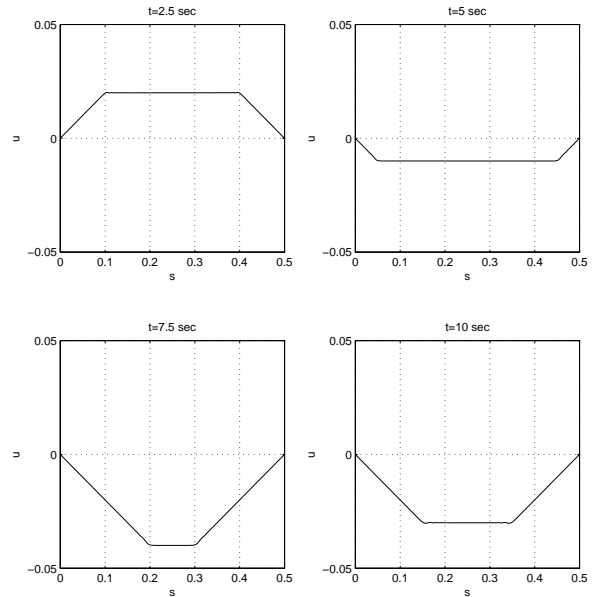


Fig. 6 CP2 partial differential equation – simulation results

Regarding software, again three types have been identified in CP1 solutions: programming language, simulation language and simulation system. These types differ in their specialisation level, where programming languages show the lowest level and simulation systems show the highest level of specialisation. It was shown that simulation languages are the dominating software type in CP1 contributions, followed by programming languages.

Regarding task distribution in CP1 example “Monte Carlo study”, two types of distribution have been identified: static distribution and dynamic distribution. Thereby, static distribution was the dominating type, where dynamic distribution or both types are implemented by only one contribution, each.

Finally, the new ARGESIM Benchmark on Parallel and Distributed Simulation (CP2) was presented and modifications towards CP1 have been pointed out. Thereby, two CP2 examples represent extensions to CP1 examples, where one example problem, “Lattice-Boltzmann Simulation” replaces a former CP2 example. It was shown that the main difference between CP1 and CP2 is the ability to implement alternative sequential solutions which enlarges the bandwidth of parallelisation and task distribution strategies. This property of the new benchmark enables the identification of advantages and disadvantages of different simulation methods with respect to parallel processing.

## 6 References

- [1] F. Breiteneker, I. Husinsky, and G. Schuster. Comparison of Parallel Simulation Techniques – Test Problems. *EUROSIM – Simulation News Europe*, 10:21–22, March 1994.

- [2] F. Breitenecker, I. Husinsky, and G. Schuster. Comparison of Parallel Simulation Techniques – Sample Solution. *EUROSIM – Simulation News Europe*, 10:24–25, March 1994.
- [3] M. Ostroversnik, and D. Murray-Smith. Comparison of Parallel Simulation Techniques – Parsytec Transputer Supercluster / SLIM. *EUROSIM – Simulation News Europe*, 11:34–35, July 1994.
- [4] G. Schuster, and F. Breitenecker. Comparison of Parallel Simulation Techniques – Cogent XTM / “mosis”. *EUROSIM – Simulation News Europe*, 11:36–37, July 1994.
- [5] P. Dellwig. Comparison of Parallel Simulation Techniques – Heterogeneous MP-system / FSIMUL-P. *EUROSIM – Simulation News Europe*, 12:32–33, November 1994.
- [6] G. Schuster, and F. Breitenecker. Comparison of Parallel Simulation Techniques – Cogent XTM / Linda, C. *EUROSIM – Simulation News Europe*, 12:34–35, November 1994.
- [7] R. Ruzicka. Comparison of Parallel Simulation Techniques – Cogent XTM / SIMUL\_R PARALLEL. *EUROSIM – Simulation News Europe*, 12:36–37, November 1994.
- [8] S. Pawletta, and W. Drewelow. Comparison of Parallel Simulation Techniques – Workstation Cluster / MATLAB/PSI. *EUROSIM – Simulation News Europe*, 13:38–39, March 1995.
- [9] K. Schwarz, and F. Breitenecker. Comparison of Parallel Simulation Techniques – Shared Memory Multiprocessors / UNIX. *EUROSIM – Simulation News Europe*, 17:37–38, July 1996.
- [10] J. Schäuuffele. Comparison of Parallel Simulation Techniques – ASCET-RS. *EUROSIM – Simulation News Europe*, 18:37–38, November 1996.
- [11] G. Schuster, and W. Zeller. Comparison of Parallel Simulation Techniques – mosis, Parsytec Supercluster / Parsytec Explorer. *EUROSIM – Simulation News Europe*, 18:39–40, November 1996.
- [12] G. Hanf, and R. Hohmann. Comparison of Parallel Simulation Techniques – Multiprocessor System with Physically Distributed Memory / CPSS. *EUROSIM – Simulation News Europe*, 19:46–47, March 1997.
- [13] R. Fink, S. Pawletta, and T. Pawletta. A Matlab based Solution to ARGESIM “Comparison of Parallel Simulation Techniques” using DP-Toolbox. *EUROSIM – Simulation News Europe* 38/39:46–47, December 2003.
- [14] F. Breitenecker, G. Höfinger, R. Fink, S. Pawletta, and T. Pawletta. ARGESIM Benchmark on Parallel and Distributed Simulation. In *EUROSIM – Simulations News Europe*, Special Issue: Parallel and Distributed Simulation Methods and Environments, 16/2:69–71, September 2006.