ARGESIM Benchmark on Parallel and Distributed Simulation

Felix Breitenecker, Gerhard Höfinger,Vienna University of Technology
Thorsten Pawletta, Sven Paletta, René Hinz, University Wismar, Germany
Felix.Breitenecker@tuwien.ac.at

This ASIM / ARGESIM Benchmark on Parallel and Distributed Simulation addresses benefits of parallel and distributed computing in the area of continuous, discrete, and hybrid simulation and in related areas. This new benchmark may be of interest for users of all types of parallel and distributed facilities. The spectrum may range from parallelisation strategies and strategies for distributing tasks, via general purpose programming languages to simulation languages, and from networks of workstations, via special parallel computers, to very high performance computers. The problems considered are a Monte-Carlo study for parameters in a dynamic mass-spring system, a case study for Lattice-Boltzmann simulation for fluid flow (famous cavity flow problem published by Hou et al.), and solution of the PDE for the swinging string with different approaches.

Introduction

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 ([1]). 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.

1 Contribution to Benchmark CP2

The *ARGESIM Benchmark on Parallel and Distributed Simulation* test benefits of parallel and distributed simulation with three case studies:

- Monte-Carlo study for parameters in a dynamic mass-spring system
- Case study for Lattice-Boltzmann simulation for fluid flow (famous cavity flow problem published by Hou et al.)
- Solution of the PDE for the swinging string with different approaches

Participation at this benchmark requires:

- Documentation of the algorithms for solving the case studies (one or more algorithms)
- Documentation of the strategy for parallelising or distributing the case studies (one or more strategies)
- Serial solution of the case studies
- Parallel / distributed solutions of the case studies
- Determination and documentation of efficiency of parallelisation

In detail, a contribution to this benchmark should for each case study describe first the approach or the algorithms for calculating solutions, followed by information about the method of parallelisation or distribution of tasks and subtasks. It is highly appreciated, if more than one solution for a particular case study is given, either using different parallelisation strategies or strategies for distribution, or by using different hardware environments, or by using different algorithms for calculating solutions.

In the following results of the case studies should be presented, based on a comparison of a serial solution and the parallel / distributed simulation of each case study.

For quantitative comparison of serial solution and parallel or distributed solutions, performance should be assessed in terms of the relative speed-up factor, a numerical value found by dividing the time for serial solution by the time for the parallel solutions (speed-up factor $\boldsymbol{f}$).

Measurements of time, whenever necessary, should be in terms of the total elapsed time for running the task. Furthermore, a rough indication should be provided for the (time) effort for implementing a parallel / distributed simulation (at best compared with implementing the serial solution).

Contributions to this benchmark will be published in the journal SNE – Simulation News Europe. Solutions sent in should not exceed four SNE pages and will be reviewed by the editorial board and by authors of the benchmark. Whenever possible, also model files and code should be provided, or linked.

C1 Case Study 1 Monte Carlo Study

The first case study is a Monte Carlo study. In a damped dynamic mass – spring system the damping factor is randomly disturbed, and the mean of a sample of dynamic outputs is to be calculated. The second order mass-spring system is described by the following ODE, where the damping factor d should be chosen as a random quantity uniformly distributed in [800, 1200]:

$$m\,\ddot{x}(t) + d\,\dot{x}(t) + k\,x(t) = 0$$

$$x(0) = 0,\ \dot{x}(0) = 0.1,\ k = 9000,\ m = 450$$

The task is to calculate a sample of r = 1000 results $x(t, d_i)$ of the motion (Figure 1 shows $x(t, 1000)$) and to calculate the mean motion $x_{mean}(t)$ over the time interval [0, 2] with a resolution (stepsize) of 0.01 (n = 200 steps):

$$x_i(t) = x(t, d_i), \quad i = 1, \ldots, r$$

$$x_{mean}(t) = \frac{1}{r}\sum_{i=1}^{r} x_i(t) = \frac{1}{r}\sum_{i=1}^{r} x(t, d_i)$$

As the model is a linear one, the solution can be provided also analytically, not only by using an ODE solver:

$$x(t, d) = Ke^{-\alpha t}\sin(\omega t)$$

$$\alpha = \frac{d}{2m},\ \omega^2 = \frac{k}{m} - \alpha^2,\ K = \frac{\dot{x}(0)}{\omega}$$

$$x_{mean}(t_j) = \frac{1}{r}K\sin(\omega t_j)\cdot\sum_{i=1}^{r} e^{-\frac{d_i}{m}t_j},\ j = 1, \ldots, n$$

While the ODE may be basis for a parallelisation of the varying damping factor, the analytical formula may be a basis for parallelisation of the 201 time instants, where a solution is to be calculated.

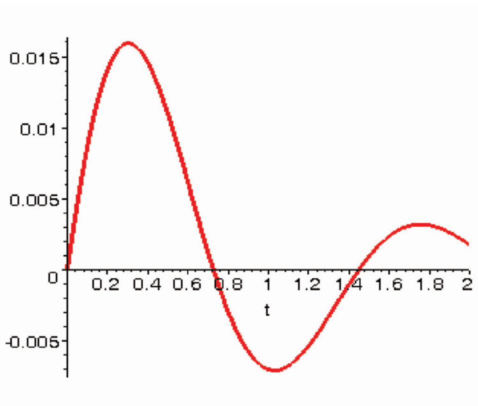

Figure 1: Plot of the analytical solution of the second order mass – spring system with d = 1000.

For documentation, we ask for a precise description of the parallelisation strategy used, and for comparison of the solutions we ask for a plot of the mean motion $x_{mean}(t)$ and of values for the speed-up factors $\boldsymbol{f}$.

C2 Case Study 2 Lattice Boltzmann Simulation

The second case study 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.

The Lattice Boltzmann Method ([2]) is derived from the *Lattice Gas* (*Cellular*) *Automata* (LGA), a cellular automata approach which considers single particles on lattice nodes. In contrast to LGA, LBM deals with distribution function values instead of single particles. The exact denomination for the Lattice Boltzmann Method is *Lattice Boltzmann BGK Method* (LBGK), caused by the special collision operator being introduced by Bhatnager, Gross and Krook in 1954.

In lattice gas cellular automata, space, time, particle velocity, and particle occupation state are all discrete. In LBM and LBGK, particle occupation state on nodes is replaced by single-particle distribution functions (real valued).

In 2D square LBM, a square lattice with unit spacing is used. Each node has eight nearest neighbours being connected by eight links (see Figure 2). Particles on nodes move along the axes and along the diagonals with discrete speed, furthermore, non moving particles with speed zero are allowed. The occupation of particles is represented by a single-particle distribution function. The distribution function represents the probability to find a particle at a certain node.

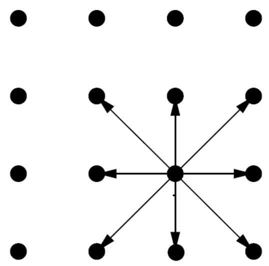

Figure 2: Nearest neighbour links of a lattice node.

The Lattice-Boltzmann BGK equation equals particle propagation terms and particle collision terms. In simulation, in each time step, two operations have to be performed: collision and propagation due to the equations ([3]).

The case study is based on on a special problem in fluid dynamics, the famous cavity flow problem published by Hou et al. in J. Comput. Physics ([4]), where the behaviour of an incompressible fluid in a square enclosure, driven by a constant stream on the top boundary is examined (see Figure 3).

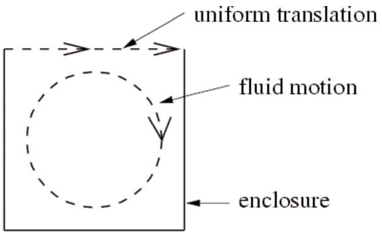

Figure 3: Lid-driven cavity flow.

For a description of the geometry matrix g, cell types are divided into wall cells (W), driving cells (D) and fluid cells (F). For a lattice size of 2×2, the matrix g is given in the following:

$$g_{2,2} = \begin{pmatrix} D & D & D & D \\ W & F & F & W \\ W & F & F & W \\ W & W & W & W \end{pmatrix}$$

The uniform translation on top of the cavity is given as $u_{0x} = 0.1$, $u_{0y} = 0$, where the Reynolds number is $Re = 1000$. At any grid point, the initial macroscopic velocity is $u_x = 0$, $u_y = 0$ and the initial density is $\rho\ = 1$.

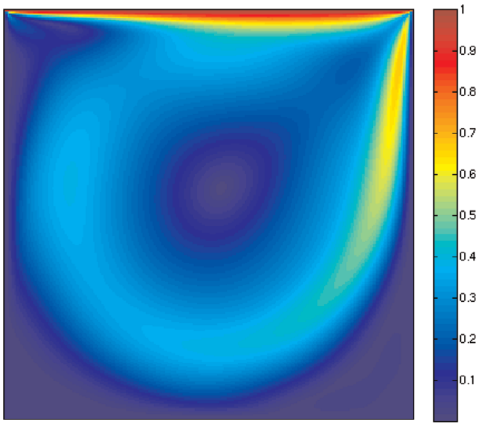

Figure 4: Relative macroscopic velocity magnitude (u/u_0) in cavity flow after 350000 iterations on a 257x257 grid.

The task is, to simulate the cavity flow with lattice size 257×257 for a number of 350.000 iterations. After this number of iterations, steady state is reached. Simulation results are shown in Figure 3.

For documentation, we ask for a precise description of the parallelisation strategy used, and for comparison of the solutions we ask for a plot of relative macroscopic velocity magnitudes (u / u_0) at steady state and for values of the speed-up factors $\boldsymbol{f}$ (please note, that also a serial solution is necessary for this purpose).

A problem discussion in detail and links to sequential reference implementations as well as to introductory materials for the lattice Boltzmann method are provided at WWW.MB.HS-WISMAR.DE/CEA/LBM.

C3 Case Study 3 – Solution of a Partial Differential Equation

The third case study is based on a second order partial differential equation describing a swinging string with length L fixed at both ends, excited at the beginning

$$u_{xx}(t,x) = \frac{1}{v^2} u_{tt}(t,x)$$

One approach for solving this PDE is the method of lines, using discretisation of space. Discretising the space into N equidistant intervals and replacing the differential quotient $u_{xx}(t,x)$ by a central difference quotient, a set of weakly coupled ODEs replaces the PDE:

$$u(0,t) = u(L,t) = 0, \quad u_t(x,0) = 0$$
$$u(0 \le x \le \frac{L}{2}, 0) = 2\frac{h}{L}x, \quad v = 0.6, \quad L = 0.5, \quad h = 0.05$$
$$u(\frac{L}{2} \le x \le L, 0) = 2h(1 - \frac{1}{L}x)$$

$$\frac{k^2}{v^2}\ddot{u}_i(t) = u_{i-1}(t) - 2u_i(t) + u_{i+1}(t), \quad i = 1, N-1$$
$$u_i(0) = 2\frac{h}{N}i, \quad i = 0, \dots, \frac{N}{2},$$
$$u_i(0) = 2h(1 - \frac{i}{N}), \ i = \frac{N}{2}, \dots, N, \quad \dot{u}_i(0) = 0$$

Also an analytical solution (approximation) can be calculated because of linearity. A classical separation approach $u(t,x) = X(x) \cdot T(t)$ can be used for calculating the solution.

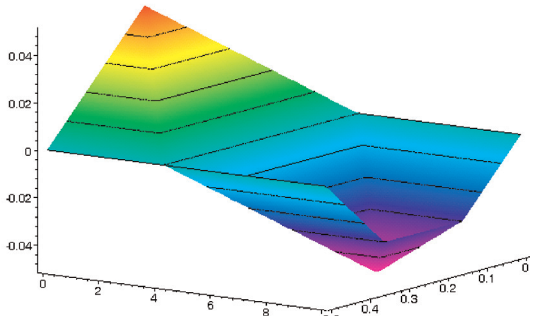

Figure 5: Surface plot for the swinging string – excitation in dependency of space and time.

This yields with given initial and boundary conditions a solution with a Fourier series ([5]):

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

Figure 5 shows a surface plot of the solution, Figure 6 and Figure 7 show solution lines in x and t, calculated with Fourier series (series cut at 100 summands).

In principle, also discretisation of space and time may be suitable. For instance, using for space discretisation a central difference quotient as in method of lines, and using for time backwards difference quotients (as well for PDE and for initial condition) yields a linear system for $u(t_k, x_i)$, which may be parallelised for solution. Of course, other algorithms for solving the PDE may be used, with varying grids etc, which can be parallelised or distributed appropriately.

In general, the system is to be solved with a spatial discretisation of $N = 500$ lines at the interval [0, 10] with time discretisation of 0.01s (m = 1000 steps).

For documentation, we ask for a precise description of the parallelisation strategy used, and for comparison of solutions we ask for plots of the lines

$$u\,(x=3L/4,\ t),\quad u\,(x=L/2,\ t)$$
$$u\,(x,\ t=15),\quad u\,(x,t=30),$$

and of a surface plot (excitation versus space and time). Furthermore, give values for the speed-up factors $\boldsymbol{f}$.

References

[1] F. Breitenecker, I. Husinsky, G. Schuster: *Comparison of Parallel Simulation Techniques – Definition.* Simulation News Europe SNE 10, March 1994.

[2] S. Chen, G. D. Doolen: Lattice Boltzmann method for fluid flows. Annual Review of Fluid Mechanics 30, p. 329-364, 1998.

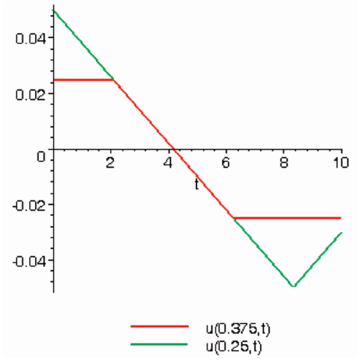

Figure 5: Solution of the PDE, excitation over time at x = 0.375 and x = 0.25.

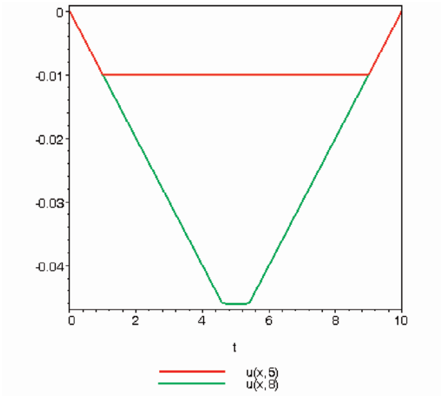

Figure 7: Solution of the PDE, excitation over space at t = 5 and t = 8.

[3] M. Krafczyk: *Lattice-BGK TUTORIAL. Version 1.0, 03/2001.* WWW.LSTM.UNI-ERLANGEN.DE/lbm2001/download/LBGK_tutorial.tgz

[4] S. Hou, Q. Hou, S. Chen, G. D. Doolen, A.C. Cogley: *Simulation of Cavity Flow by the Lattice Boltzmann Method.* J. Comput. Phys. Vol. 118, no. 2, May 1995.

[5] Ch. B. Lang, N. Pucker: *Mathematische Methoden in der Physik*; Akad. Verlag, Spektrum HTB, Heidelberg, 1998 ISBN 3-8274-0225-5

Corresponding authors:
F. Breitenecker, Vienna University of Technology,
Inst. f. Analysis and Scientific Computing, Wiedner Hauptstrasse 8-10, 1040 Vienna, Austria;
Felix.Breitenecker@tuwien.ac.at
Thorsten Pawletta, Wismar University,
Process and Environmental Engineering,
Phillip Müller Straße, 23952 Wismar,
Germany; *pawel@mb.hs-wismar.de*

Received: June 15, 2006
Revised: August 30, 2006
Revised extended version: January 15, 2007
Accepted: February 6, 2007