Computational Science Technical Note **CSTN-144**

Halo Gathering Scalability for Large Scale Multi-dimensional Sznajd Opinion Models Using Data Parallelism with GPUs

K. A. Hawick and D. P. Playne

2012

The Sznajd model of opinion formation exhibits complex phase transitional and growth behaviour and can be studied with numerical simulations on a number of different network structures. Large system sizes and detailed statistical sampling of the model both require data-parallel computing to accelerate simulation performance. Data structures and computational performance issues are reported for simulations on single and multi-core processing devices. A discussion of optimal data structures for performance on Graphical Processing Units using NVIDIA's Compute Unified Device Architecture (CUDA) is also given. System size and memory layout tradeoffs for different processing devices are also presented.

Keywords: opinion formation model; interdisciplinary simulation; data-parallelism; GPU; CUDA

BiBTeX reference:

```
@INPROCEEDINGS{CSTN-144,
        author = {K. A. Hawick and D. P. Playne},
        title = {Halo Gathering Scalability for Large Scale Multi-dimensional Sznajd
                Opinion Models Using Data Parallelism with GPUs},
        booktitle = {Proc. Int. Conf. on Parallel and Distributed Processing Techniques
                and Applications (PDPTA'12)},
        year = {2012},
        pages = {95-101},
        address = {Las Vegas, USA},
        month = {16-19 July},
        publisher = {CSREA},
        institution = {Computer Science, Massey University},
        keywords = {opinion formation model; interdisciplinary simulation; data-parallelism;
                GPU; CUDA},
        owner = {kahawick},
        timestamp = {2012.05.03}
}
```

Technical Report CSTN-144

Halo Gathering Scalability for Large Scale Multi-dimensional Sznajd Opinion Models Using Data Parallelism with GPUs

K.A. Hawick and D.P. Playne
Computer Science, Institute for Information and Mathematical Sciences,
Massey University, North Shore 102-904, Auckland, New Zealand
{k.a.hawick, d.p.playne }@massey.ac.nz; Tel: +64 9 414 0800 Fax: +64 9 441 8181

March 2012

ABSTRACT
The Sznajd model of opinion formation exhibits complex phase transitional and growth behaviour and can be studied with numerical simulations on a number of different network structures. Large system sizes and detailed statistical sampling of the model both require data-parallel computing to accelerate simulation performance. Data structures and computational performance issues are reported for simulations on single and multi-core processing devices. A discussion of optimal data structures for performance on Graphical Processing Units using NVIDIA's Compute Unified Device Architecture (CUDA) is also given. System size and memory layout tradeoffs for different processing devices are also presented.

KEY WORDS
Sznajd opinion formation model; complex system; simulation; data-parallelism; GPU; CUDA; memory halo

1 Introduction

Opinion formation models [21] such as the Sznajd model [40] display some interesting phase transitional and complex behaviours that rely upon computer simulation for their study. Opinion formation models can capture individual behaviour at a microscopic level that manifests itself as a macroscopic or system-wide outcome when implemented with system of many participating agents. The Sznajd model of opinion formation exhibits complex phase transitional and growth behaviour and can be studied with numerical simulations on a number of different network structures [18, 35].

One of the main points of interest in simulating this

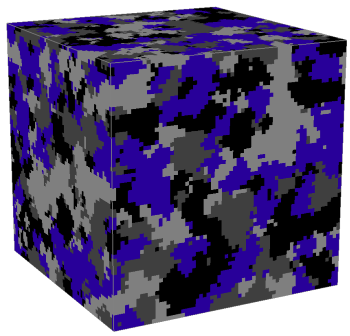

Figure 1: Multi-opinion Sznajd Model simulated on a 40^3 periodic lattice with 4 opinions in the population.

sort of model is to study the dynamics [29, 43] and kinetics [5, 11] of the system both from the perspective of a well-defined model of phase transitional behaviour but also for to study the spread of influence of opinions [13, 22, 44] for comparison with real sociological phenomena.

Sznajd models can be simulated on a range of network structures with varying number of neighbouring opinion sites [33]. Results have been reported for one dimensional systems [32]; triangular lattices [23] and generalised heterogeneous graphs and small-world [1] and Watts-Strogatz graph structures [42]. In this present paper we focus on square lattices but our simulation system can also manage arbitrary dimensional hyper-cubic structures in 3 dimensions and higher. Figure 1 is a system snapshot from our simulation, and shows the ongoing formation of spatial groups of like-minded opinion holding agents arranged on a cubic symmetry lattice.

This class of model can capture important sociological behaviours such as conformity [36]; crowd dynamics [25]; political extremism and relative agreements [9]; and voter herding and the role of independents [20]. Sznajd models can be usefully compared with real sociological systems such as political elections [34]; proportional and majority elections [41]; and other sociological and knowledge exchange scenarios [28].

The Sznajd model and its variants are also of interest as abstract models for the study of dynamics and criticality. Numerical experiments and data obtained from simulations allow comparisons with theoretical predictions. An area of theoretical comparison is with non-equilibrium [10] and growth theories including Fokker-Planck theory, fluctuation dissipation theory [12] and other relaxation modelling equations [26].

Comparisons are drawn in the literature between the Sznajd model and related systems like the Ising model [37]. The Sznajd system has been described as a "push" model as influence is pushed outwards from an updated cell, whereas the Ising model is a "pull" model since it changes a cell's value based upon the value of its immediate neighbours [4,39]. Other areas of comparison are based on the use of externally applied opinion or magnetic field biases [3,7]. Active areas of development include incorporation of bulk parameters such as an effective temperature into Sznajd-like models [24,38].

In summary, the Sznajd model and related models are an important class of complex system to study numerically. Large and fast simulations are necessary to support useful comparisons with realistic real systems and with theoretical predictions. We can achieve large fast simulations using data-parallel computing techniques.

Graphical Processing Units (GPUs) are finding many important uses as engines for data parallelism, over and above their original purpose for accelerating graphics systems. Opinion formation models such as the Sznajd system are relatively well suited to this form of parallel simulation and GPUs enable a rapid exploration of the parameter space of such models is possible interactively and with good statistical sampling on relatively large model systems.

Compute Unified Device Architecture (CUDA) [30] is NVIDIA's proprietary programming language that is widely used in programming computational model simulations including Ising systems. In this present paper we show how the Sznajd model can be implemented on GPU systems but that there are some unexpected complications due to the particular nature of the Sznajd model neighbourhoods. We also show that there is an interesting tradeoff between achieving realistically large system sizes the scalability requirements of simulating a system long enough to reach a well defined consensus opinion point.

Our paper is structured as follows: In Section 2 we summarise the Sznajd model and describe how we implemented it in Section 3. We include details on memory utilisation issues and a simple and more memory efficient implementation. We present some performance benchmark data on multiple GPU systems in Section 4. We offer some areas for further work in Section 5 and some conclusions in Section 6.

2 Sznajd Model Formulation

The Sznajd model formulation used in this research is formulated in the same manner as described in [18]. The simulation consists of some very simple microscopic rules that result in complex macroscopic behavior. Each update consists of selecting a cell and a random neighbor. If the two cells have the same opinion they convince their direct neighbors. This particular variation of the Sznajd model update can be applied for Sznajd systems of any dimensionality and is therefore useful for the parallel algorithmic and scalability analysis presented in this present work.

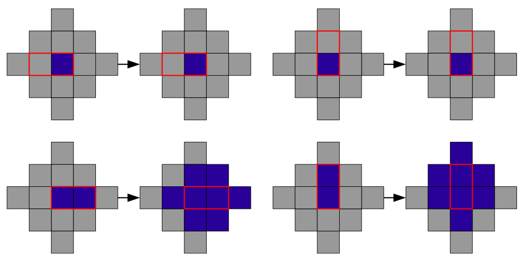

Figure 2: The update rules for the Sznajd Model, if both selected cells agree, they convince their neighbors. If they disagree the neighbors are unchanged.

The Sznajd model is constructed as follows:

1. each of the N lattice cells represents a voter agent holding a single opinion
2. starting conditions are chosen for a random mixture of Q different opinion states.
3. each voter is updated at each time step
4. upon choosing a voter, we randomly look at one of its neighbours.
5. if the voter and the chosen neighbour hold the same opinion they "persuade" all their immediate neighbours to this opinion

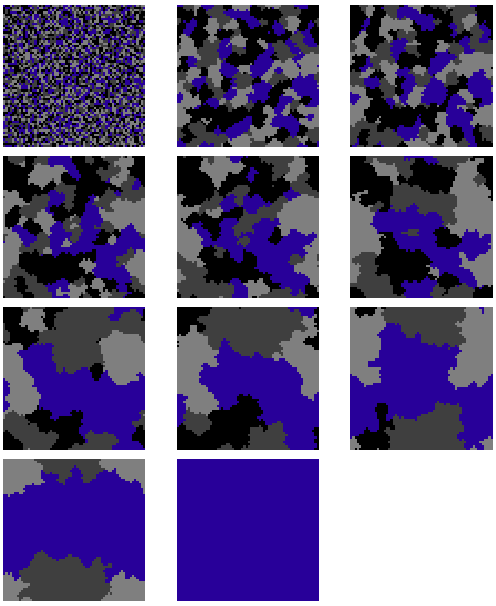

Figure 3: Sznajd Model on 64×64 periodic lattice at times: 0,1,2,4,8,16,32,64,128,256 and final configuration after random 25/25/25/25 start.

6. this process repeats until consensus is reached (whereby all voters hold the same final opinion state)

There are a number of properties that can be measured to study the changes in the population of opinion agents: the time to achieve consensus t_c; the time t_q to eliminate a species and yield only $q < Q - 1$ states present in the system; and the mean fluctuation sizes in these. These can be studied for different sizes of system N and also for different dimensionalities and indeed differing neighbour influence regions.

The Sznajd model and models like it can be run to full consensus under many dynamical conditions [29]. The meshes we study do lead (eventually) to a consensus outcome. The consensus state is a stable state since there are no thermal or spontaneous changes of opinion in the model we study. However although the consensus state will be arrived at eventually in finite time for a finite system, these completion times do grow with system size. We explore this effect and the implications for the maximum feasible system sizes that we can study.

Figure 2 shows the transition rules for the Sznajd model. The model behaviour is best understood from examination of a time series of model configurations. Figure 3 shows a two dimensional lattice of Sznajd opinion-holding agents evolving over logarithmic times from a random initial mix of opinions to a single consensus.

3 Data-Parallel Implementation

Sznajd model simulations present an challenge for Graphical Processing Units. In previous investigation of lattice-based simulations [15, 17, 19, 27, 31] GPUs generally exhibit the best performance for large system sizes as small simulations struggle to fully utilize the computational throughput of GPU architectures.

The nature of the Sznajd model means that the number of steps required for a system to reach consensus grows rapidly with system size. Because of this relationship, simulations of large system sizes that are well-suited to the GPU architectures require a very large number of steps to reach consensus. Even on powerful GPU architectures these large system sizes take an in feasibly long time to compute the thousands of runs required for statistical analysis.

This means an implementation must be developed that can perform effectively for small system sizes $N = 128^2$ to $N = 256^2$.

3.1 Sznajd Model Memory Access

The Sznajd model has a relatively large memory access pattern or memory halo. Each cell update has the potential to change the values of twelve different neighbouring cells. To perform parallel updates, it is important that these updates do not overlap or else the behavior of the model will be affected.

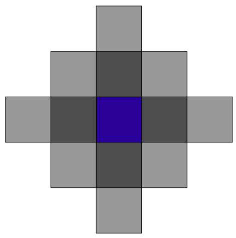

Figure 4: The potential memory halo of a Sznajd model update. The blue cell is the cell selected for update, the dark grey cells represent the immediate neighbors and the light grey cells represent cells whose values could potentially be affected.

3.2 Naive GPU Implementation

The initial idea of implementing the Sznajd model was to implement a checkerboard style update commonly used to simulate the Ising model [19]. The checkerboard update or red/black update performs a parallel update on every cell belonging to one half of the checkerboard. This update method is applicable to the Ising model because it reads from it's nearest neighbors and only changes the value of the cell being updated. Because the Sznajd model has quite a different memory halo, a different checkerboard pattern is required.

To ensure that no two parallel update ever write to the same cell, the memory halos of the updated cells must not overlap. One possible checkerboard design is to split the lattice into 5×5 (2D) or 5^3 (3D) sections. Each update selects the same cell out of this section and updates it. Each update will randomly select one neighbor but this is already accounted for in the memory halo. Importantly the memory halos of the updates never overlap, this can be seen in Figure 5.

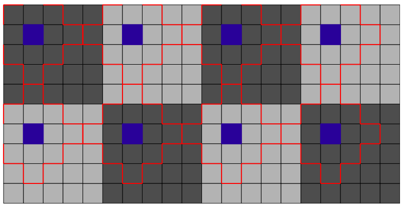

Figure 5: Update pattern of the naive GPU implementation, the update pattern ensures no cell value is changed by two different updates.

This implementation has regular memory access patterns and would be a perfect candidate for lattice crinkling which rearranges data in a lattice for better memory access [16]. However, during initial testing a problem was discovered with this implementation. When two neighboring cells are updated one after another, there is a tendency for an opinion to propagate in that direction. Because the cells are updated in a regular pattern, any neighboring updates will occur across the entire lattice. This means opinions propagate in one direction across the entire lattice and causes the lattice to 'jitter' during the course of the simulation.

Any correlation between updates such as this have the potential to affect the behavior of the model and skew any results. For this reason the Sznajd checkerboard update has not been used for any results gathering and instead an alternative implementation was developed to remove the correlation.

3.3 Improved GPU Implementation

The improved Sznajd update method is similar to the checkerboard update but with a few important differences. Instead of splitting the lattice into 5×5 sections, the improved method splits the lattice into 8×8 sections. These 8×8 sections are further split into 4×4 quadrants. Instead of the same cell in each section being updated at the same time, each quadrant of each section is updated at the same time. This update will randomly select a cell from the quadrant and update that cell. This breaks the correlation seen in the previous implementation while still insuring that the memory halos from two update never overlap. This update method is shown in Figure 6.

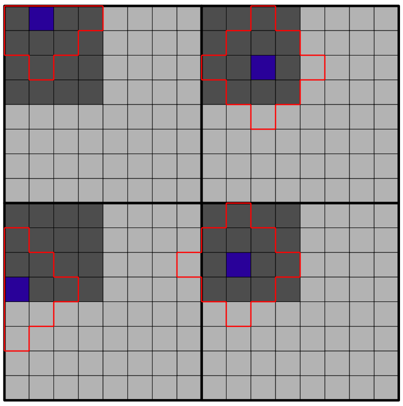

Figure 6: The update pattern of the improved GPU implementation. One random cell in each quadrant is updated

While this update method can be used to simulate the Sznajd model, it does severely restrict the computational load of each update. Each parallel update only processes one cell out of an entire 8x8 cell. This means for an N^2 system only $\frac{N^2}{8^2}$ cells can be updated in parallel. To perform an entire system update, 8^2 parallel updates are required. This severely restricts the performance of the simulation which is already limited to small system sizes due to the nature of the model but is necessary to break correlation between sequential updates.

3.4 Optimising the Implementation

Unfortunately due to the semi-random access pattern of this update method, previously successful optimization techniques such as data crinkling are not applicable [14, 19]. However, some optimization strategies such as bit-packing can be used. We are investigating Sznajd systems with $Q = \{2..16\}$, this means only 4 bits of storage are required for each cell. This allows the values of 8 cells to be packed into a single 32-bit integer. This conveniently allows each row of the 8x8 sections to be stored in a single integer.

This not only helps reduce the storage requirements for a Sznajd system but also helps memory access. Because each line of every 8x8 section will be stored in sequential memory addresses, they can be read in a single coalesced memory read. Once these values

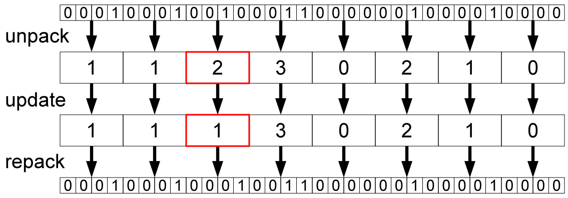

Figure 7: The process of unpacking integers, updating the simulation and re-packing them into the integers for storage.

are read, they will be unpacked into a shared memory array which will be used to compute the simulation. The threads computing the update will change the values in this shared memory array and then repack them into integers which will be written back to GPU memory. This process is illustrated in Figure 7.

4 Performance and Results

Due to the highly restricted and sparse update method coupled with very limited system sizes, the GPU implementation struggles to provide the kind of speedups seen in many simulations. For the system sizes appropriate for Sznajd simulations (64^2 - 256^2) the computational power of the GPU cannot be fully utilized. However, the GPU implementation can still performance faster than the CPU version for system sizes $N >= 160^2$.

As the system size increases, more of the computational power of the GPU can be used and the speedup it provides will increase. This performance improvement becomes increasingly important as the behavior of larger systems is investigated.

The performance of the GPU implementation has been compared to the un-optimised version and a CPU version for a range of system sizes and number of opinions. Varying the number of opinions had little impact on the performance of any simulation and only the results for $Q = 2$ are presented.

The CPU implementation has been written in C, compiled with gcc 4.5 and executed on a 2.67GHz Intel Core i7 920. Both the un-optimised (GPU1) and optimised (GPU2) implementations have been compiled using CUDA 4.0 and executed on a GTX580. The performance results of these three Sznajd model simulations are shown in Figure 8.

The improved performance of the GPU2 implementation allows larger system sizes to be investigated. The number of steps to consensus of the Sznajd model has been investigated for system sizes of $N = \{64^2...256^2\}$ and $Q = \{2...10\}$. The results of this experiment are presented in Figure 9 and appear to shown some unexpected results.

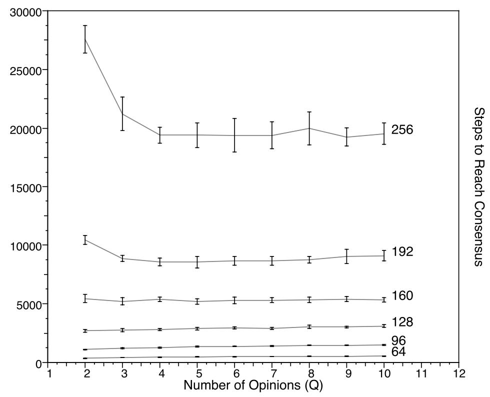

Figure 9: The number of simulation steps required for the Sznajd system to reach consensus.

Previous work with system sizes of $N <= 96^2$ has shown that the number of steps required to reach consensus increases as the number of opinions increases [18]. However, the findings from the previous experiment shows that for Sznajd systems of $N >= 192^2$ and $Q = 2$ require more steps to reach consensus than systems with higher values of Q.

5 Discussion

There are a number of variations to the simple Sznajd model that have been explored in the literature. The notion that two voters with the same opinion will have a strong influence on their immediate neighbours is not unreasonable, although it is possible to vary the number of agreeing voters required to cause a local sway of opinion. Triples and plaquettes of four have been tried. It appears that a pair is enough to capture the essential model behaviour however, and we have used the simplest "voter pairs" in this paper.

A simple regular mesh is not particularly realistic and a number of other meshes and graph networks including preferential and scale free network structures have also been studied and reported in the literature. Again, it appears that the square mesh captures the essence of the model behaviour and we use a simple square mesh in the work reported here. However we do consider the neighbourhood of influence of the agreeing pair of voters.

There are various ways the Sznajd model could be extended. One interesting possibility is modelling opin-

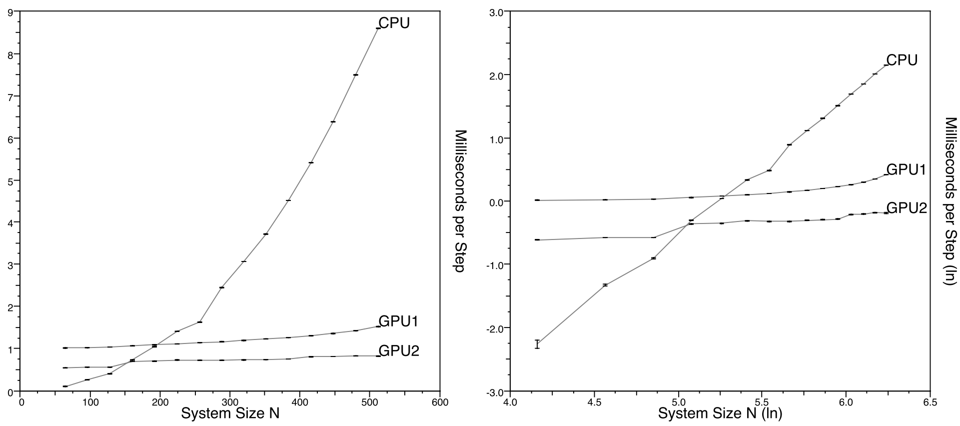

Figure 8: The milliseconds per time-step of three Sznajd model simulations - the CPU, GPU1 (un-optimised) and GPU2 (optimised) implementations. The results are shown in normal scale (left) and log-log scale (right).

ions that are continuous rather than discrete [2]. This sort of simulation would require use of floating point capabilities, which present a different set of performance tradeoffs on GPUs, which typically share floating point units across the their cores.

Another area for development is the study of agent reputations and how the reputation - or long term behaviour of an agent - affects its power of influence on its neighbouring agents [6, 8]. Temporal memory effects like reputation will require storage of agent histories and will considerably change the memory requirements. This will lower the feasibility of containing the simulation state within cache.

6 Conclusion

We have discussed the importance of Sznajd opinion models and their role in simulation studies of real sociological systems and as models of phase transitional and complex emergent behaviour. We have described the need for large simulations that must be run to the point of single opinion consensus.

We have experimented with various data and multi parallel implementations and have identified a tradeoff that makes good use for multiple GPU devices attached to a CPU. The computational performance tradeoff space is not trivial and is dominated by the halo gathering scalability effects.

The Sznajd model neighbour-hood and the push nature of the information propagation means that the simulated system has to be arranged with a relatively large crinkle length, to safely delegate compute responsibility across the data parallel cores of a GPU. While this supports good scalability for large simulated system sizes, we are constrained to having systems sizes that are not so large as to make it infeasible to achieve single opinion consensus. Numerical experiments require us to run the simulations to this point for sensible comparisons with theory. The multi-GPU approach addresses this by achieving good job throughput.

Some areas for further development of Sznajd-like models have also been identified. These include the use of continuous opinion dynamics and historical reputation effects that would further exercise the floating point and memory management properties of GPUs.

References

[1] Bagnoli, F., Barnabei, G., Rechtman, R.: Small-world bifurcations in an opinion model. Tech. Rep. arXiv:0909.0117v3, University of Florence (December 2009)

[2] Biswas, S.: Mean field solutions of kinetic exchange opinion models. Phys. Rev. E 84, 056106 (2011)

[3] Candia, J.: Advertising and irreversible opinion spreading in complex social networks. Int. J. Modern Physics C 20, 1–20 (2009)

[4] Castellano, C., Pastor-Satorras, R.: Irrelevance of information outflow in opinion dynamics models. Phys. Rev. E 83, 016113 (2011)

[5] Chowdhury, K.R., Ghosh, A., Biswas, S., Chakrabarti, B.K.: Kinetic exchange opinion model: solution in the single parameter map limit. arXiv 1112.5328v1, Saha Institute of Nuclear Physics, Kolkata, India (December 2011)

[6] Crokadakis, N., Forgerini, F.L.: Competition among reputations in the 2d sznajd model: Spontaneous emergence of democratic states. Tech. rep., Universidade Federal Fluminense, Rio de Janeiro, Brazil (2011)

[7] Crokidakis, N.: Effects of mass media on opinion spreading in the sznajd sociophysics model. Physica A; Stat. Mech and its Applications 391, 1729–1734 (2012)

[8] Crokidakis, N., de Oliveira, P.M.C.: The sznajd model with limited persuasion: competition between high-reputation and hesitant agents. J. Stat. Mech: Theory and Experiment Online, P11004 (2011)

[9] Deffuant, G., Amblard, F., Weisbuch, G., Faure, T.: How can extremism prevail? a study based on the relative agreement interaction model. Journal of Artificial Societies and Social Simulation 5(4), 1–27 (2002), `http://jasss.soc.surrey.ac.uk/5/4/1.html`

[10] Descalzi, O., Marti, A.C., Masoller, C., Rosso, O.A.: Topics on non-equilibrium statistical mechanics and nonlinear physics. Phil. Trans. Roy. Soc. A 367, 3151–3156 (2009)

[11] During, B.: Kinetic modelling of opinion leadership. SIAM News December, 1–12 (2011)

[12] During, B., Markowich, P., Pietschmann, J.F., marie Therese Wolfram: Boltzmann and fokker-planck equations modelling opinion formation in the presence of strong leaders. Proc. Roy. Soc A 465(2112), 3687–3708 (September 2009)

[13] Fortunato, S.: Damage spreading and opinion dynamics on scale free networks. Physica A: Statistical mechanics and Its Applications 348, 683–690 (2005)

[14] Hawick, K.A., Playne, D.P.: Automatically Generating Efficient Simulation Codes on GPUs from Partial Differential Equations. Tech. Rep. CSTN-087, Computer Science, Massey University (2010), submitted to Elsevier, J. Comp. Sci.

[15] Hawick, K.A., Playne, D.P.: Modelling, Simulating and Visualizing the Cahn-Hilliard-Cook Field Equation. International Journal of Computer Aided Engineering and Technology (IJCAET) 2(CSTN-075), 78–93 (2010)

[16] Hawick, K.A., Playne, D.P.: Hypercubic Storage Layout and Transforms in Arbitrary Dimensions using GPUs and CUDA. Concurrency and Computation: Practice and Experience 23(10), 1027–1050 (July 2011)

[17] Hawick, K.A., Playne, D.P.: Numerical Simulation of the Complex Ginzburg-Landau Equation on GPUs with CUDA. In: Proc. IASTED International Conference on Parallel and Distributed Computing and Networks (PDCN). pp. 39–45. No. CSTN-070, IASTED, Innsbruck, Austria (15-17 February 2011)

[18] Hawick, K.: Multi-party and spatial influence effects on opinion formation models. In: Proc. IASTED International Conference on Modelling and Simulation (MS 2010). No. CSTN-032, Calgary, Canada (June 2010), paper 696-035

[19] Hawick, K., Leist, A., Playne, D.: Regular Lattice and Small-World Spin Model Simulations using CUDA and GPUs. Int. J. Parallel Prog. 39(CSTN-093), 183–201 (2011)

[20] Hisakado, M., Mori, S.: Digital herders and phase transition in a voting model. J. Phys. A: Math. and Theor. 44(27), 275204 (July 2011)

[21] Kacperski, K., Holyst, J.A.: Opinion formation model with strong leader and external impact: A mean field approach (1999)

[22] Kempe, D., Kleinberg, J., Tardos, E.: Maximizing the spread of influence through a social network. In: Proc. 9th ACM SIGKDD Int. Conf. on Knowledge Discovery and Data MiningKDD'03 (2003)

[23] Kondrat, G., Sznajd-Weron, K.: Percolation framework in Ising-spin relaxation. Phys. Rev. E 79, 011119–1–5 (2009)

[24] Kondrat, G.: How to introduce temperature to the 1 dimensional sznajd model? Physica A: Stat. Mech and its Applications 390, 2087–2095 (2011)

[25] Koster, G., Seitz, M., Treml, F., Hartmann, D., Klein, W.: On modelling the influence of group formations in a crowd. Contemporary Social Science 6(3), 397–414 (November 2011)

[26] de la Lama, M., Szendro, I.G., Iglesias, J.R.: Van kampen's expansion approach in an opinion formation model. Eur. Phys. J. B 51, 435–442 (2006)

[27] Leist, A., Playne, D., Hawick, K.: Exploiting Graphical Processing Units for Data-Parallel Scientific Applications. Concurrency and Computation: Practice and Experience 21, 2400–2437 (December 2009), CSTN-065

[28] Lella, L., Licata, I.: A new model for the organizational knowledge life cycle. In: Minati, G., Abram, M., Pessa, E. (eds.) Processes of Emergence of Systems and Systemic Properties, pp. 215–228. Springer (2007)

[29] Martins, A.C.R.: Bayesian updating rules in continuous opinion dynamics models. J. Stat. Mech. P02017(arXiv:0807.4972v1), 1–14 (2009)

[30] NVIDIA® Corporation: CUDA™ 2.1 Programming Guide (2009), `http://www.nvidia.com/`, last accessed May 2009

[31] Playne, D., Hawick, K.: Data Parallel Three-Dimensional Cahn-Hilliard Field Equation Simulation on GPUs with CUDA. In: Proc. 2009 International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA'09) Las Vegas, USA. No. CSTN-073 (13-16 July 2009)

[32] Przybyala, P., Sznajd-Weron, K., Tabiszewski, M.: Exit probability in a one-dimensional nonlinear q-voter model. Phys. Rev. E 84, 031117 (2011)

[33] Schwammle, V., Gonzalez, M.C., Moreira, A.A., Jr. Andrade, J.S., J., H.H.: The spread of opinions in a model with different topologies. Phys. Rev. E 75, 066108 (2007)

[34] Sobkowicz, P.: Modelling opinion formation with physics tools: Call for closer link with reality. Journal of Artificial Societies and Social Simulation 12(1),

1–11 (January 2009)
[35] Staffer, D.: Monte carlo simulations of sznajd models. Journal of Artificial Societies and Social Simulation 5(1), 1–9 (January 2001)
[36] Sznajd-Weron, K., Tabiszewski, M., Timpanaro, A.M.: Phase transition in the sznajd model with independence. Europhysics Letters 96, 48002 (2011)
[37] Sznajd-Weron, K.: Dynamical model of ising spins. Phys. Rev. E 70, 037104–1–4 (2004)
[38] Sznajd-Weron, K.: Sznajd model and its applications. Acta Physica Polonica B 36(8), 2537–2547 (2005)
[39] Sznajd-Weron, K., Krupa, S.: Inflow versus outflow zero-temperature dynamics in one dimensional. Phys. Rev E 74(74), 031109–1–8 (2006)
[40] Sznajd-Weron, K., Sznajd-Weron, J.: Opinion evolution in closed community. Int. J. Modern Physics C 11(6), 1157–1165 (2000)
[41] Timpanaro, A.M., Prado, C.P.C.: Generalized sznajd model for opinion propagation. Phys. Rev. Lett. 80(2), 021119 (2009)
[42] Timpanaro, A.M., Prado, C.P.C.: Coexistence of interacting opinions in a generalized sznajd model. Phys. Rev. E 84, 027101 (2011)
[43] Woloszyn, M., Stauffer, D., Kulakowski, K.: Phase transitions in nowak-sznajd opinion dynamics. Physica A 378, 453–458 (2007)
[44] Xu, X.J., Wu, Z.X., Chen, G.: Epidemic spreading in lattice-embedded scale-free networks. Physica A 377, 125–130 (2007)