Population annealing study of the frustrated Ising antiferromagnet on the stacked triangular lattice

Michal Borovský

Department of Theoretical Physics and Astrophysics, University of P. J. Šafárik in Košice, Slovakia

18th November 2015

Collaboration

- Dr. Martin Weigel (Applied Mathematics Research Centre, Coventry University, UK)
- Dr. Lev Yu. Barash (Landau Institute for Theoretical Physics, Chernogolovka, Russia)
- Dr. Milan Žukovič (UPJŠ, Košice, Slovakia)

Outline

Outline

Population annealing (PA)

Introduction

K. Hukushima and Y. Iba, *Population Annealing and Its Application to a Spin Glass*, AIP Conf. Proc. 690, 200 (2003).

- suitable for systems with rough free energy surfaces (spin glasses, frustrated spin systems, complex biomolecular systems, etc.)
- used as an alternative to parallel tempering
- combination of simulated annealing, population algorithms and sequential Monte Carlo method
- provides a good estimate of free energy

Population annealing

Algorithm

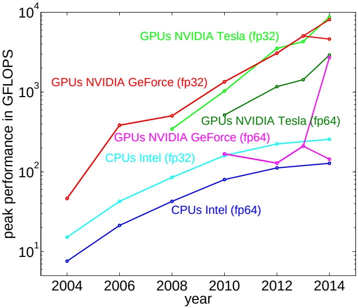
- initialize population of R_K replicas at $\beta_{K+1} = 0$
- for β_k from β_K to β_0 with step $\Delta\beta = \beta_k - \beta_{k+1}$
  - partition function ratio: $Q_k = \frac{1}{\tilde{R}_{\beta_{k+1}}} \sum_{j-1}^{\tilde{R}_{\beta_{k+1}}} \exp\left[-\Delta\beta E_j\right]$
  - for all replicas do:
    - normalize weights: $\tau_j = \frac{1}{Q_k} \exp\left[-\Delta\beta E_j\right]$
    - resampling: create $\mathcal{N}\left[\left(R_{\beta_k}/\tilde{R}_{\beta_{k+1}}\right)\tau_j\right]$ copies of replica ($\mathcal{N}\left[a\right]$ - Poisson random variate with mean value a)
  - calculate new size of a population $\tilde{R}_{\beta_k}$
  - equilibrate replicas for θ_k Monte Carlo sweeps
  - calculate observables and the free energy:

$$-\beta_k \tilde{F}(\beta_k) = \ln\Omega + \sum_{l=K}^{k} \ln Q_l$$

Outline

CPU vs. GPU

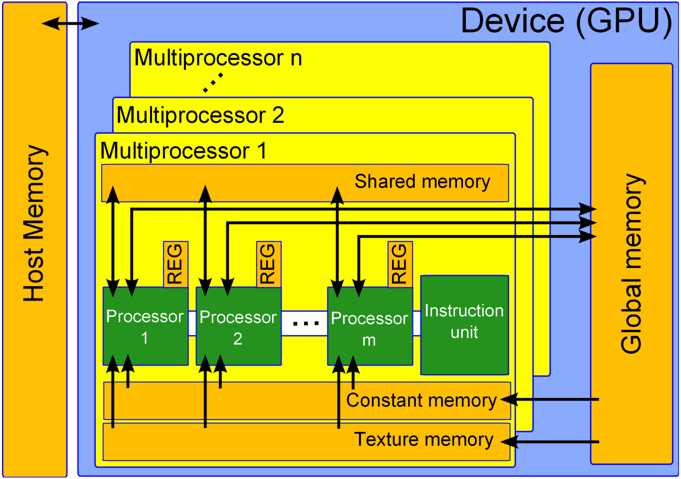
Performance comparison



rok	CPU - Intel	GPU - NVIDIA GeForce	GPU - NVIDIA Tesla
2004	Pentium 4 570J (3.8GHz)	6800 GT	-
2006	Core 2 Duo E6700 (2.66GHz)	7950 GT	-
2008	Core 2 Quad Q9400 (2.66GHz)	9800 GT (112 CUDA cores @ 600MHz)	C870 (128@600MHz)
2010	Core i7-980 (3.33GHz)	GTX 480 (448@607MHz)	C2070 (448@575MHz)
2012	Core i7-3770K (3.5GHz)	GTX 680 (1536@1006MHz)	K20 (2496@706MHz)
2013	-	GTX 780 Ti (2880@875MHz)	K40 (2880@705MHz)
2014	Core i7-4790K (4GHz)	GTX Titan Z(5760@705MHz)	K80 (4992@562MHz)
		GTX 980 (2048@1126MHz)	

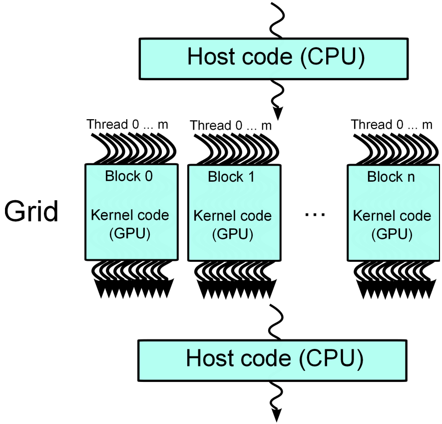
GPU CUDA architecture

Schematic depiction



M. Weigel, Journal of Computational Physics 231 (2012) 30643082

CUDA program



- GPU program:
  - Host code - ANSI C
  - Device code - ANSI C extended by keywords for kernels (parallel functions) and data structures
- NVIDIA C compiler (nvcc)
- Program execution:
  - THREAD
  - (WARP)
  - BLOCK
  - GRID
- SIMT - "single instruction multiple threads"

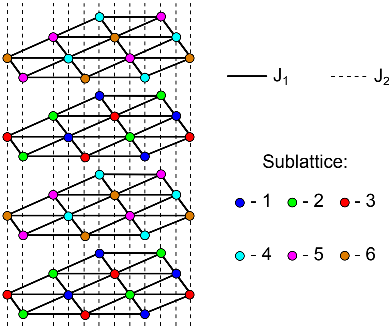
Parallelizing the PA algorithm

- 2 levels of parallelism:
  - over replicas (τ_i, Q) $\rightarrow$ 1 thread = 1 replica
  - over spins of each replica (MC update,E,M) $\rightarrow$ 1 block of threads - 8x8x8 block-wise coalesced array of spin values; 1 block = 1 replica
- use of parallel reduction algorithm for summing over replicas/spin values/local energy contributions
- parallel generation of long sequences of pseudo-random numbers - "cuRAND" Philox_4x32_10 ($p = 2^{128} \approx 10^{38}$)
- Boltzmann factor tabulation - texture memory

Outline

Stacked triangular Ising antiferromagnet

Sublattice partition and hamiltonian



Hamiltonian:

$$H = -J_1 \sum_{\langle i,j \rangle} S_i S_j - J_2 \sum_{\langle i,j \rangle} S_i S_k$$

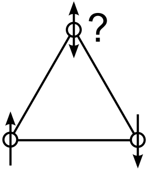
$S_i = \pm 1$ ... Ising spin variable

$J_1 < 0$ ... antiferromagnetic intralayer (interchain) interaction

$J_2 < 0$ ... antiferromagnetic interlayer (intrachain) interaction

---

Geometrical frustration:

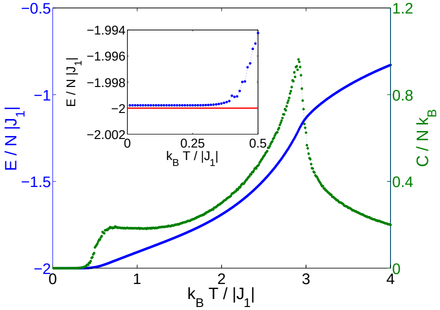


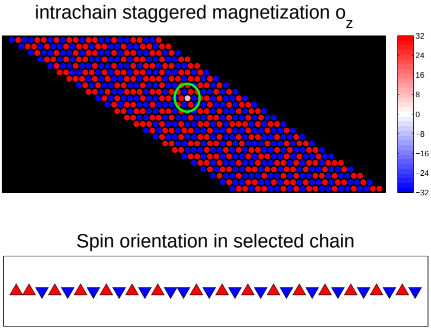
Stacked triangular Ising antiferromagnet

Kinetic freezing in a standard MCMC simulation

R.R. Netz and A.N. Berker, Phys. Rev. Lett. 66, 377 (1991).

$J_1 = J_2$, 24x24x32 spins ($L_z = 32$ layers), 10^5 MCMC sweeps (+20% for equilibration), $o_z = \sum_{k=1}^{L_z} (-1)^k S_k$, snapshot at $k_B T/|J_1| = 0.01$



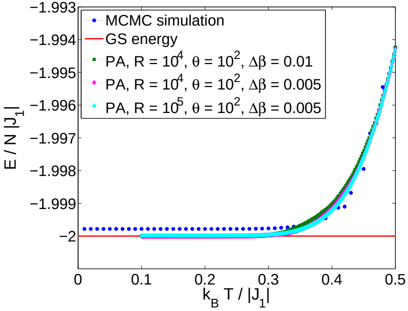


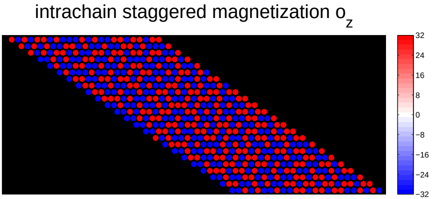
Outline

MCMC and PA comparison

GS energy and configuration

$J_1 = J_2$, 24x24x32 spins ($L_z = 32$ layers), snapshot at $k_B T/|J_1| = 0.1$





MCMC and PA comparison

Family entropy

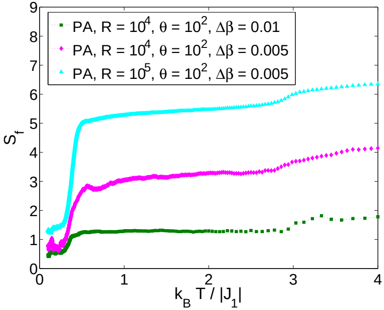
W. Wang, J. Machta, and H. G. Katzgraber, Phys. Rev. E 92, 013303 (2015)

Family entropy: $S_f = -\sum_i \nu_i \ln \nu_i$

$\nu_i \ldots$ fraction of the population with origin in the i-th replica

$e^{S_f} \ldots$ effective number of surviving families

equilibration requirement: $e^{S_f} \geq 100$ (or $S_f \gtrsim 4.6$)



Number of unique GS configurations:

- 171 (0.171% of the population size, $e^{S_f} = 3.7375$)
- 23 (0.23%, $e^{S_f} = 2.1845$)
- 32 (0.32%, $e^{S_f} = 1.5857$)

PA algorithm performance

Nvidia GTX Titan

24x24x32	$R = 10^3$	$R = 10^4$	$R = 10^5$
θ	$t_{SF}[ns]$	$t_{SF}[ns]$	$t_{SF}[ns]$
10^0	8.235	7.714	7.933
10^1	1.024	0.953	0.961
10^2	0.308	0.276	0.269
10^3	0.240	0.209	0.259
10^4	0.233	0.208	0.245
GPU memory used	17.62 MB	176.24 MB	1762.39 MB

Outline

Conclusions and perspective

Conclusions:

- we created optimized parallel GPU program of the PA algorithm for the frustrated stacked triangular Ising antiferromagnet
- system reached GS (Wannier-like phase with antiferromagnetically ordered spin chains) even for relatively small R and θ
- equilibration criterion was not met in all simulations for a low-T region

Perspective:

- choice of more effective high quality PRNG
- parallel resampling of replicas in the GPU global memory
- adaptive inverse temperature step $\Delta\beta_k$ - histogram overlap
- asynchronous multispin coding - bitwise operations
- multi-histogram reweighting

Thank you for your attention.

Population annealing

Weighted averaging

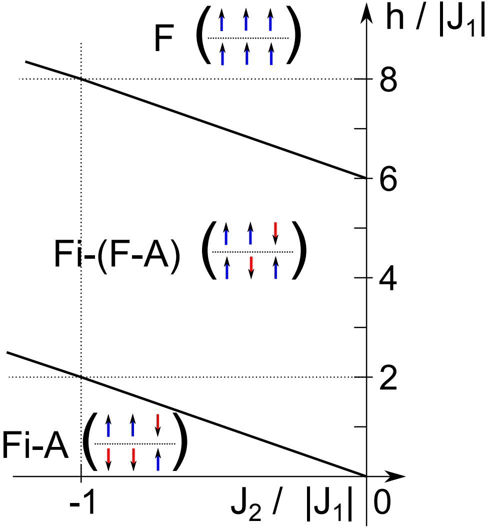
J. Machta, *Population annealing with weighted averages: A Monte Carlo method for rough free energy landscapes*, Phys.Rev.E 82, 026704 (2010)

- for not sufficient values of parameters $\tilde{R}_k$, $\Delta\beta$, $\theta_k \Rightarrow$ bias
- lets consider a set of the M independent runs of the algorithm with observables $\tilde{A}_r(\beta)$ and free energies $\tilde{F}_r(\beta)$
- weighted averaging: $\bar{A}(\beta) = \sum_{r=1}^{M} \tilde{A}_r(\beta)\omega_r(\beta)$,

  where $\omega_r(\beta) = \frac{\exp\left[-\beta\tilde{F}_r(\beta)\right]}{\sum_{r=1}^{M}\exp\left[-\beta\tilde{F}_r(\beta)\right]}$.
- unbiased free energy: $-\beta\bar{F}(\beta) = \ln\left[\frac{1}{M}\sum_{r=1}^{M}\exp\left[-\beta\tilde{F}_r(\beta)\right]\right]$
- weighted averaging errors - bootstrapping
- optimization - minimize $\mathrm{Var}(-\beta\tilde{F})$ using the same computational resources

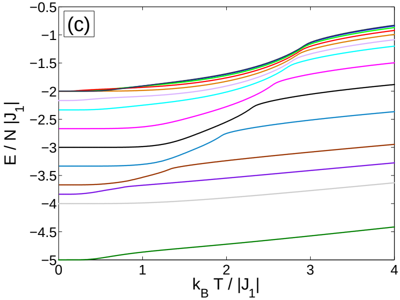
$h > 0$

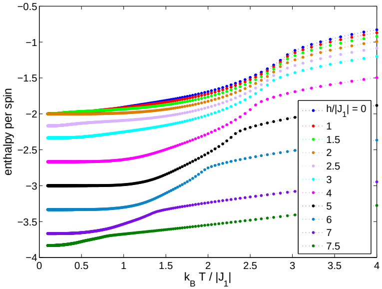
GS configurations



$h > 0$

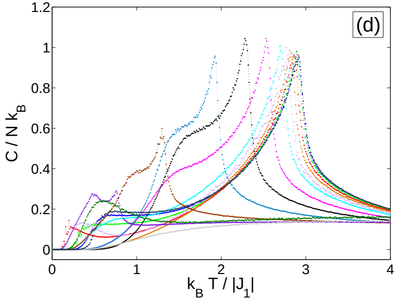
Enthalpy per spin

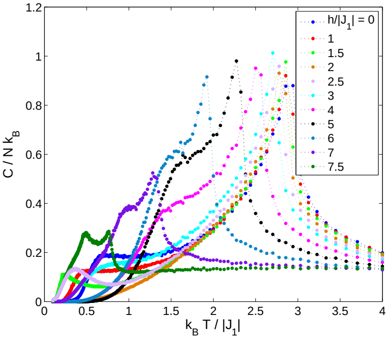




$h > 0$

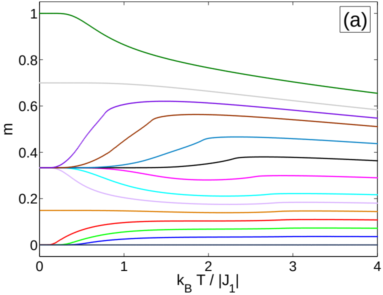
Heat capacity

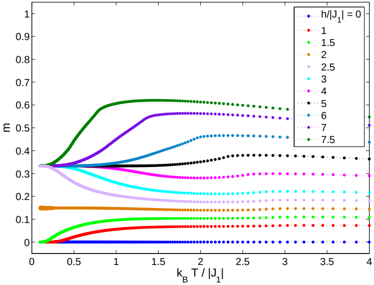




$h > 0$

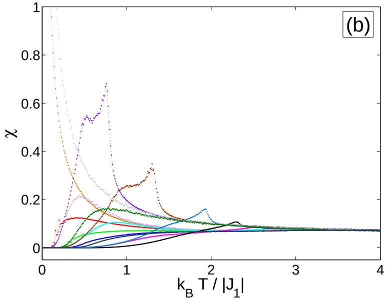
Total magnetization per spin

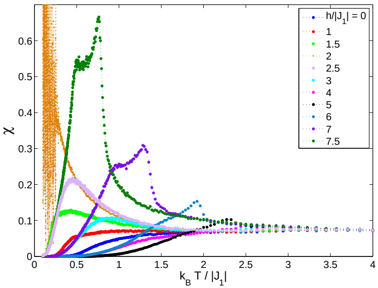




$h > 0$

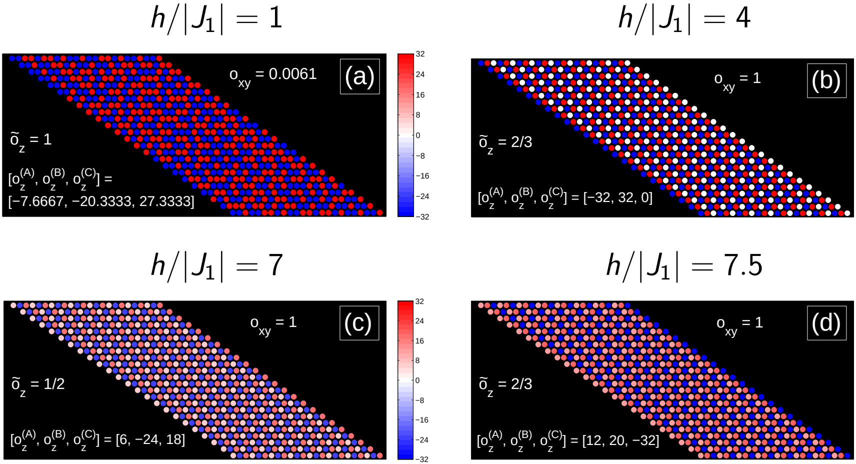
Magnetic susceptibility





$h > 0$

Ground state configurations



$h > 0$

Ground state configurations - degeneracy