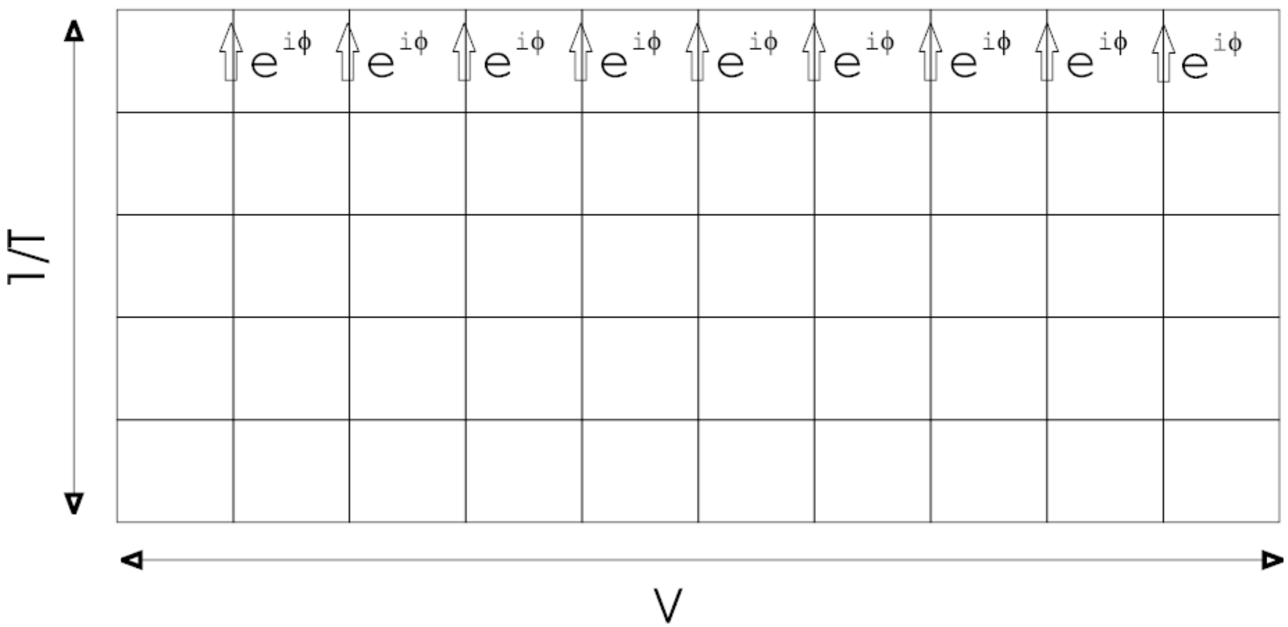
Finite density simulations using a determinant estimator

Andrei Alexandru
Anyi Li & Keh-Fei Liu

Overview

- Motivation and introduction
- The algorithm: Hybrid Noisy Monte Carlo
- Determinant estimator
- Algorithm check and volume dependence
- Conclusions and outlook

Canonical partition function

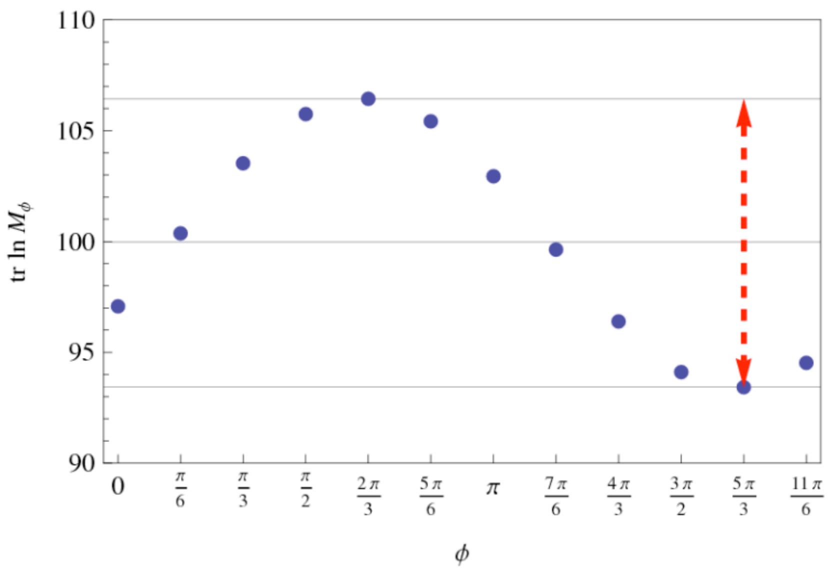


$$Z_C(V, n, T) = \int DU\, e^{-S_G[U]} \det{}_n M^2(U)$$

$$\det{}_k M \equiv \frac{1}{2\pi} \int d\phi e^{-ik\phi} \det M_\phi^2 \qquad \text{Constraint: } n_u + n_d = k.$$

Projected determinant

- To compute the projected determinant we need to evaluate the determinant for all phases -- not feasible
- We use an approximation where we employ a discrete Fourier transform



$$\det{}'_k M^2(U) = \frac{1}{N}\sum_{\varphi_i} e^{-ik\varphi_i} \det M^2_{\varphi_i}(U)$$

Exact determinant simulations: scaling

- We carried out a study on 4^4 lattices that used LU decomposition to compute the fermionic determinant
- To produce physical results we need at least 6^3x4 lattices.
- The scaling of the exact algorithm goes like V^4
- The estimator method was expected to scale like V^2
- As we will show the estimator scales with V^3, which still makes it the method of choice when moving to larger volume.

Hybrid Noisy Monte Carlo

$$Z_C(V,k,T) = \int DU\, e^{-S_G(U)} \,\mathrm{det}'_k\, M^2(U)$$
$$= \int DU D\xi\, e^{-S_G(U)} \det M^2(U)\, f_k(U,\xi)$$

$$\int D\xi\, g(U,\varphi,\xi) = \frac{\det {M_\varphi}^2(U)}{\det M^2(U)}$$

$$f_k(U,\xi) = \frac{1}{N}\sum_{\varphi_i} e^{-ik\varphi_i} g(U,\varphi_i,\xi)$$

The updating process

$$Z_C(V,n,T) = \int DUD\xi\; e^{-S_G(U)} \det M^2(U) f_n(U,\xi)$$

$$= \int DUD\xi\; \underbrace{e^{-S_G(U)} \det M^2(U)\, | f_n(U,\xi) |}_{\text{Simulation measure}} \underbrace{\frac{f_n(U,\xi)}{| f_n(U,\xi) |}}_{\text{phase}}$$

$$(U,\xi) \xrightarrow{HMC+Acc/rej} (U',\xi) \xrightarrow{Acc/rej} (U',\xi')$$

The accept/reject steps are based on the ratios $\frac{| f_n(U',\xi) |}{| f_n(U,\xi) |}$ and $\frac{| f_n(U',\xi') |}{| f_n(U',\xi) |}$.

The estimator

To set up the estimator we write

$$\frac{\det M_\phi^2}{\det M^2} = e^{2\mathrm{Tr}(\ln M_\phi - \ln M)}$$

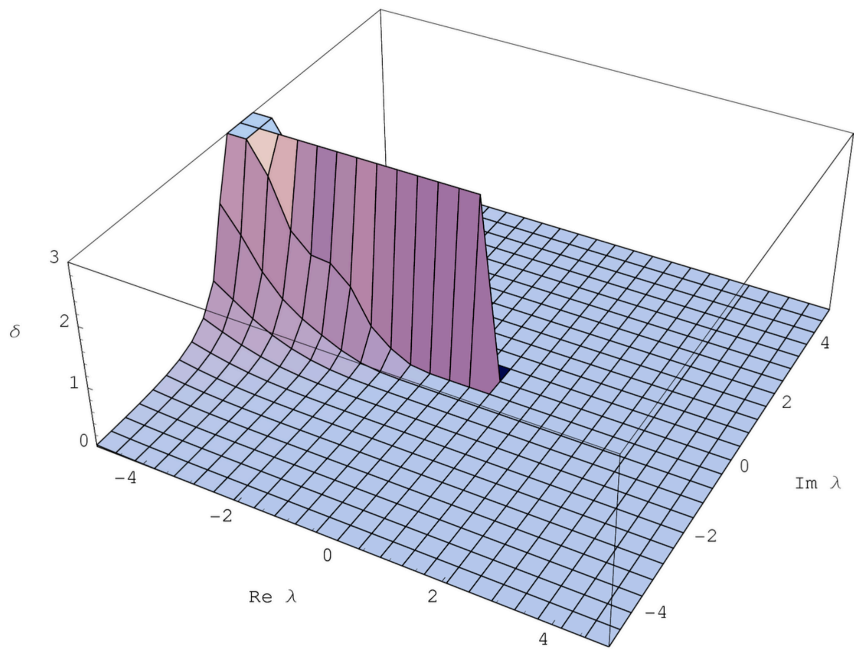
We first develop an estimator for the exponent:

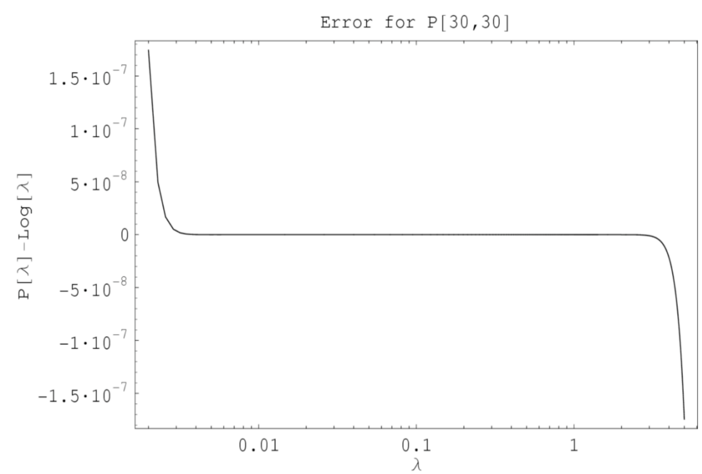
- Use Z(4) noise for trace
- Use Pade approximation for log M
- Improve the estimator using unbiased subtraction

Use the trace estimator with Bhanot - Kennedy method to turn it into and unbiased estimator for the determinant

Exponent estimator: Pade approximation

$$\ln\det M = \mathrm{Tr}\ln M \cong b_0\,\mathrm{Tr}\,I + \sum_{k=1}^{K}\mathrm{Tr}\frac{b_k}{c_k + M}$$





Trace improvement

$$\mathrm{Tr}\ln M = \int d\eta\,\eta^{+} M \eta$$

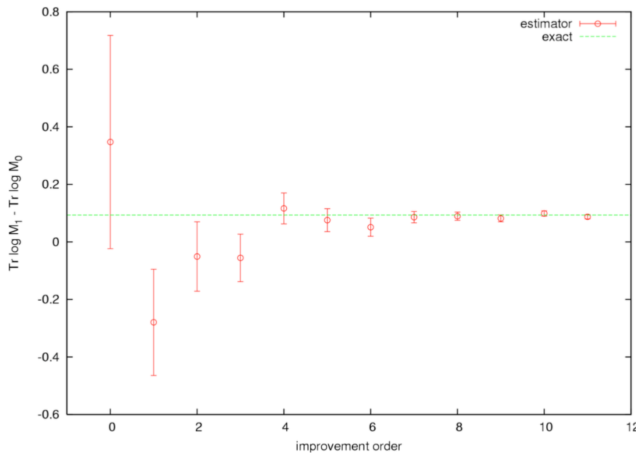
$$\mathrm{Tr}\ln M \cong b_0\,\mathrm{Tr}\,I + \sum_{k=1}^{K} \mathrm{Tr}\frac{b_k}{c_k + M}$$

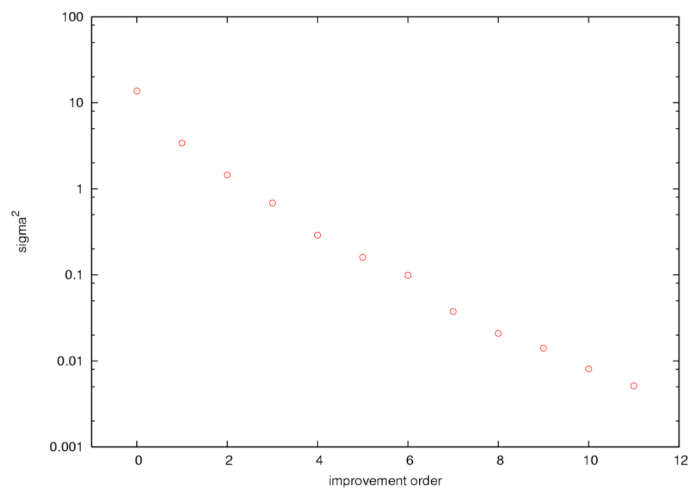
$$\mathrm{Tr}\frac{1}{c+M} = \mathrm{Tr}\Big(\frac{1}{c+M} - \sum_i O_i\Big)$$

$$\mathrm{Tr}\frac{1}{c+M} = \frac{1}{1+c} + \frac{\kappa}{(1+c)^2}D + \frac{\kappa^2}{(1+c)^3}D^2 + \ldots$$

$$O_i = \frac{\kappa^i}{(1+c)^{i+1}}\Big(D^i - \frac{1}{\mathrm{Tr}\,1}\mathrm{Tr}\,D^i\Big)$$

Number of loops for a 4^4 lattice: 4 - 10, 6 - 112, 8 - 2884, 10 - 84360.





Bhanot-Kennedy estimator

$$\det M = e^{Tr \ln M}$$

$$Tr \ln M = \int d\eta \; g(\eta)$$

$$\langle g_1(\eta) \rangle = Tr \ln M, \quad P(g_1 = 0) = 0$$

$$\langle g_2(\eta) \rangle = \frac{1}{2} Tr \ln M, \quad P(g_2 = 0) = \frac{1}{2}$$

$$\vdots$$

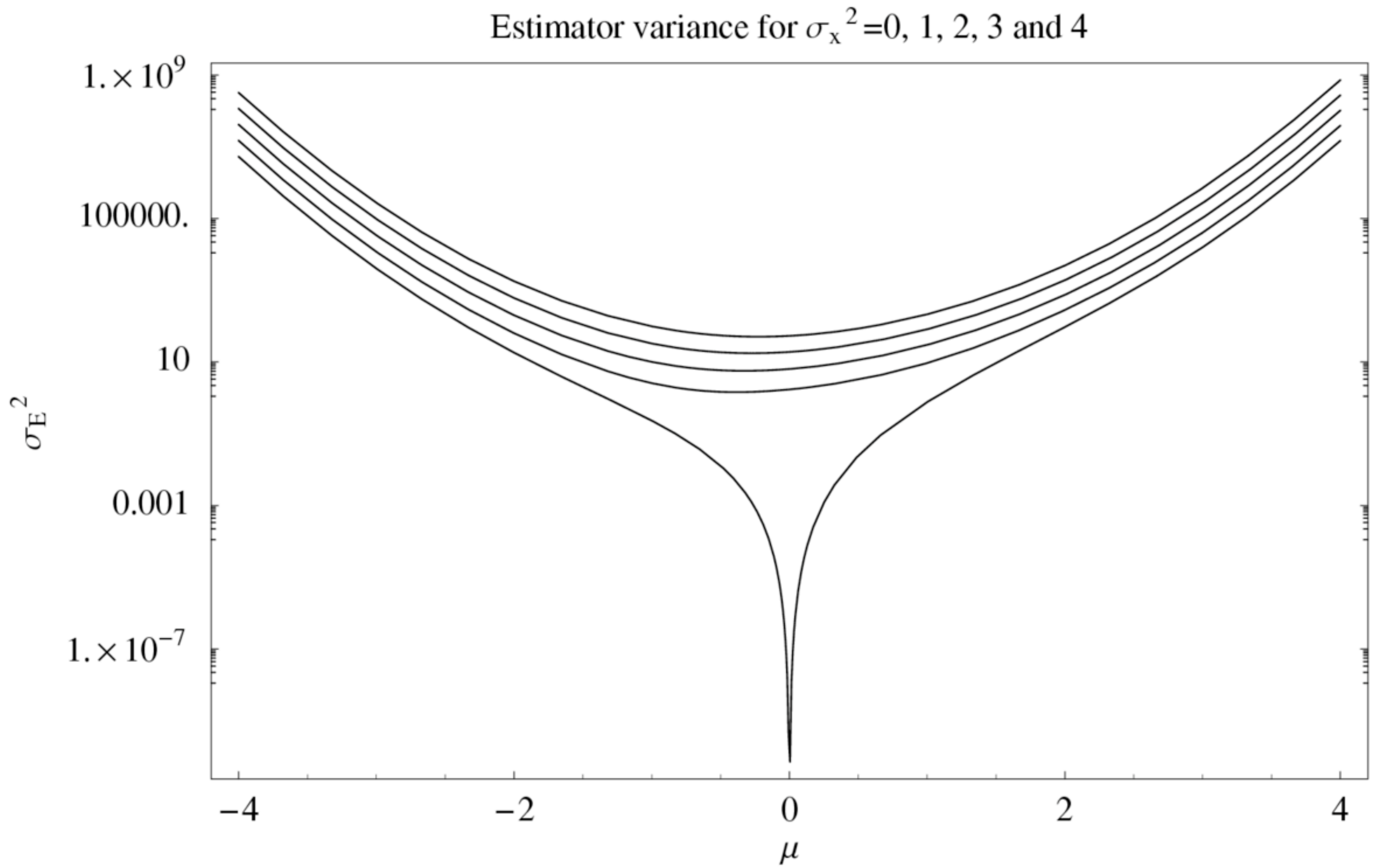
$$\langle g_k(\eta) \rangle = \frac{1}{k} Tr \ln M, \quad P(g_k = 0) = \frac{k-1}{k}$$

$$\vdots$$

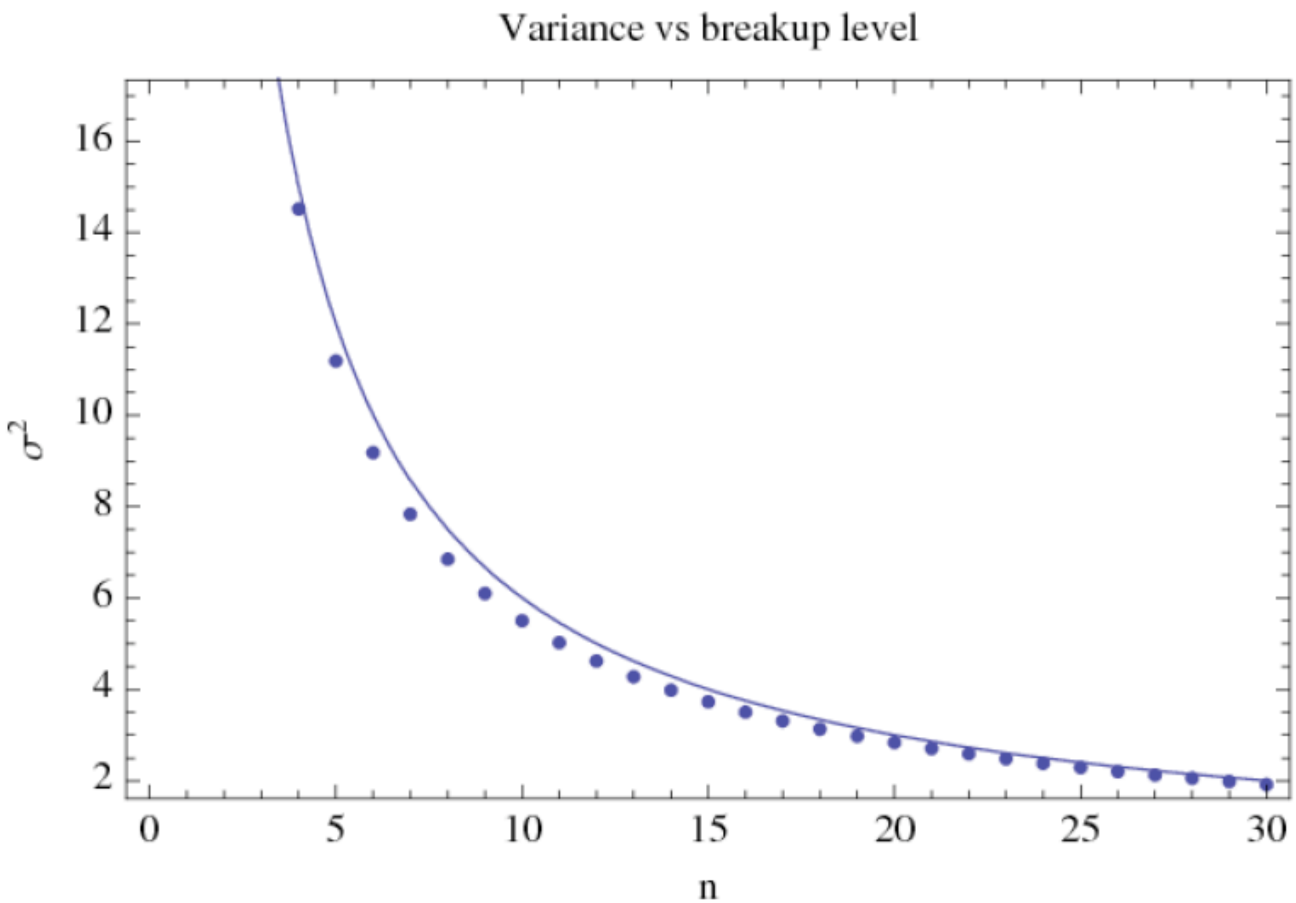
$$f(\eta_1, \eta_2, \ldots) = 1 + g_1(\eta_1) + g_1(\eta_1) g_2(\eta_2) + g_1(\eta_1) g_2(\eta_2) g_3(\eta_3) + \ldots$$

$$\langle f(\eta_1, \eta_2, \ldots) \rangle = \det M$$

Variance

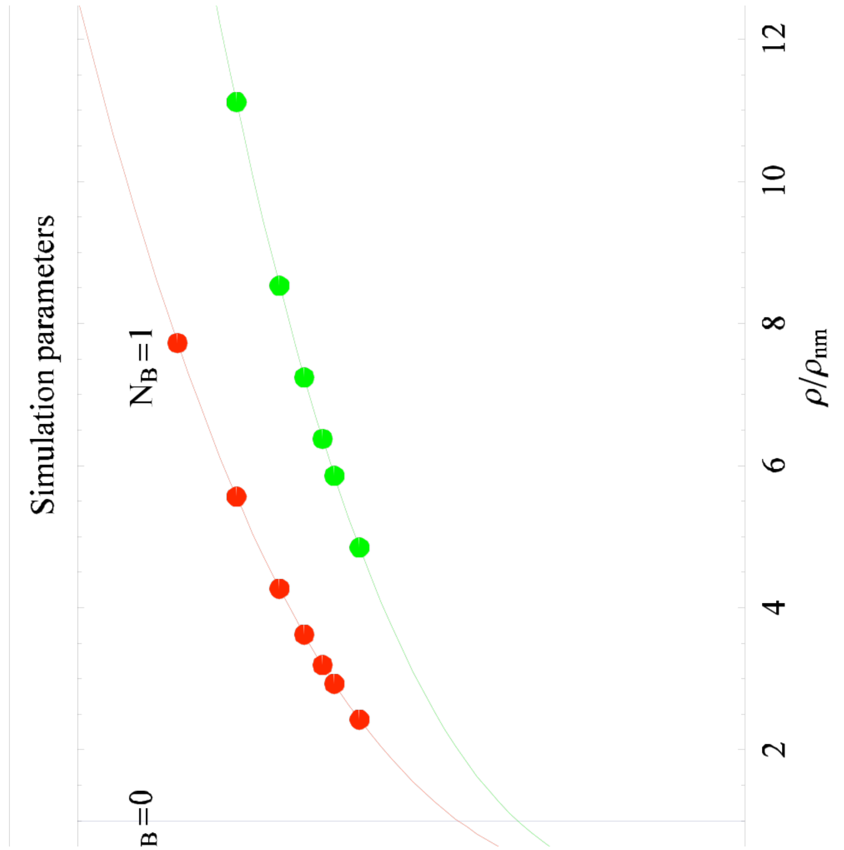

Estimator variance for $\sigma_x{}^2$=0, 1, 2, 3 and 4
$1.\times 10^9$
100000.
10
0.001
$1.\times 10^{-7}$
$\sigma_E{}^2$
−4
−2
0
2
4
μ

Estimator breakup



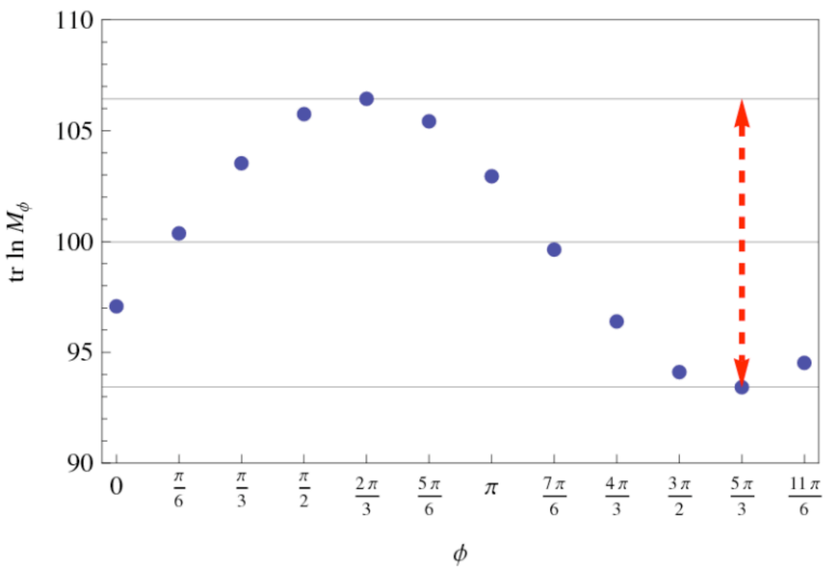
$$E(\mu, \sigma^2) \sim e^{\mu} \qquad\qquad E(\mu, \sigma^2) \to E\left(\frac{\mu}{n}, \frac{\sigma^2}{n^2}\right)^n$$

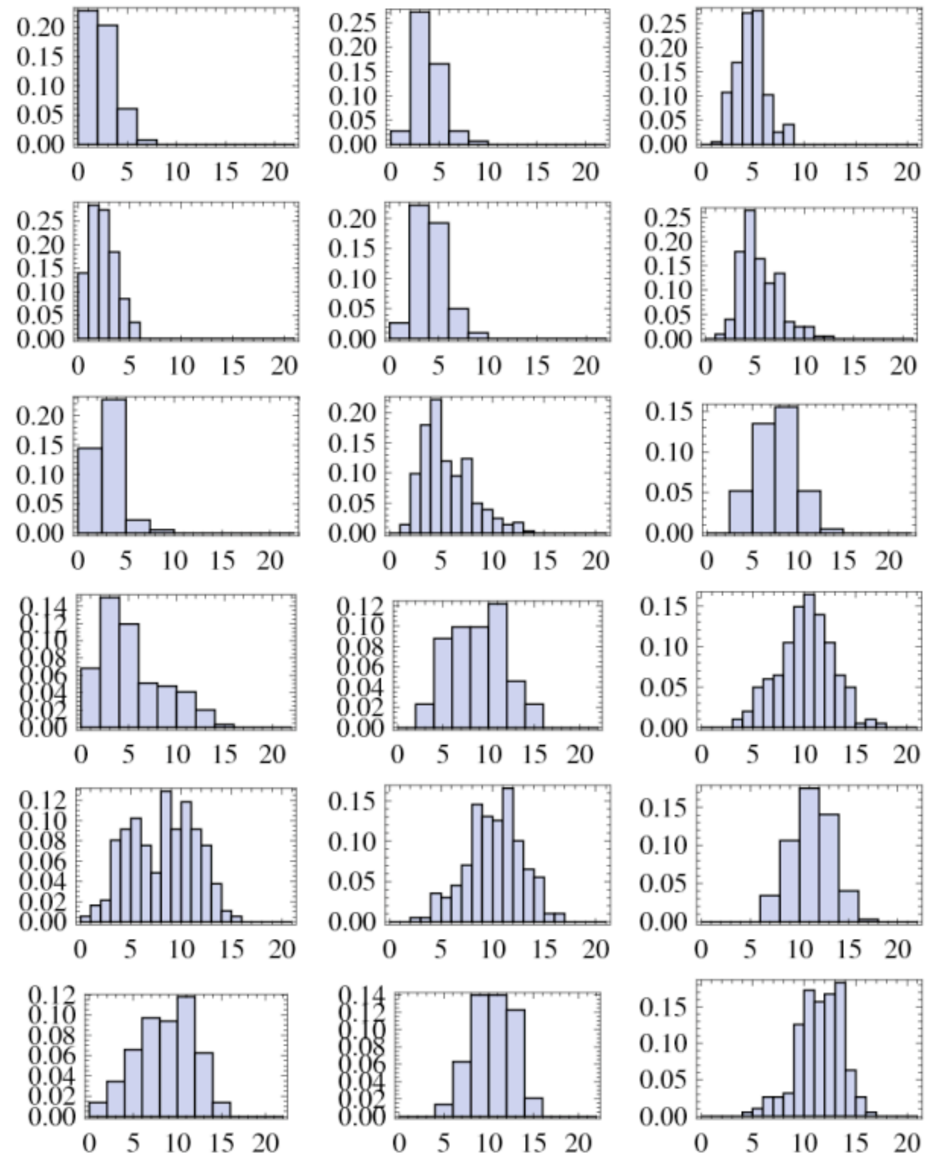
Testing the algorithm



We run the program at the same parameters as in our previous study: kappa=0.158, N=12 and the same values of beta

Determining the breakup level

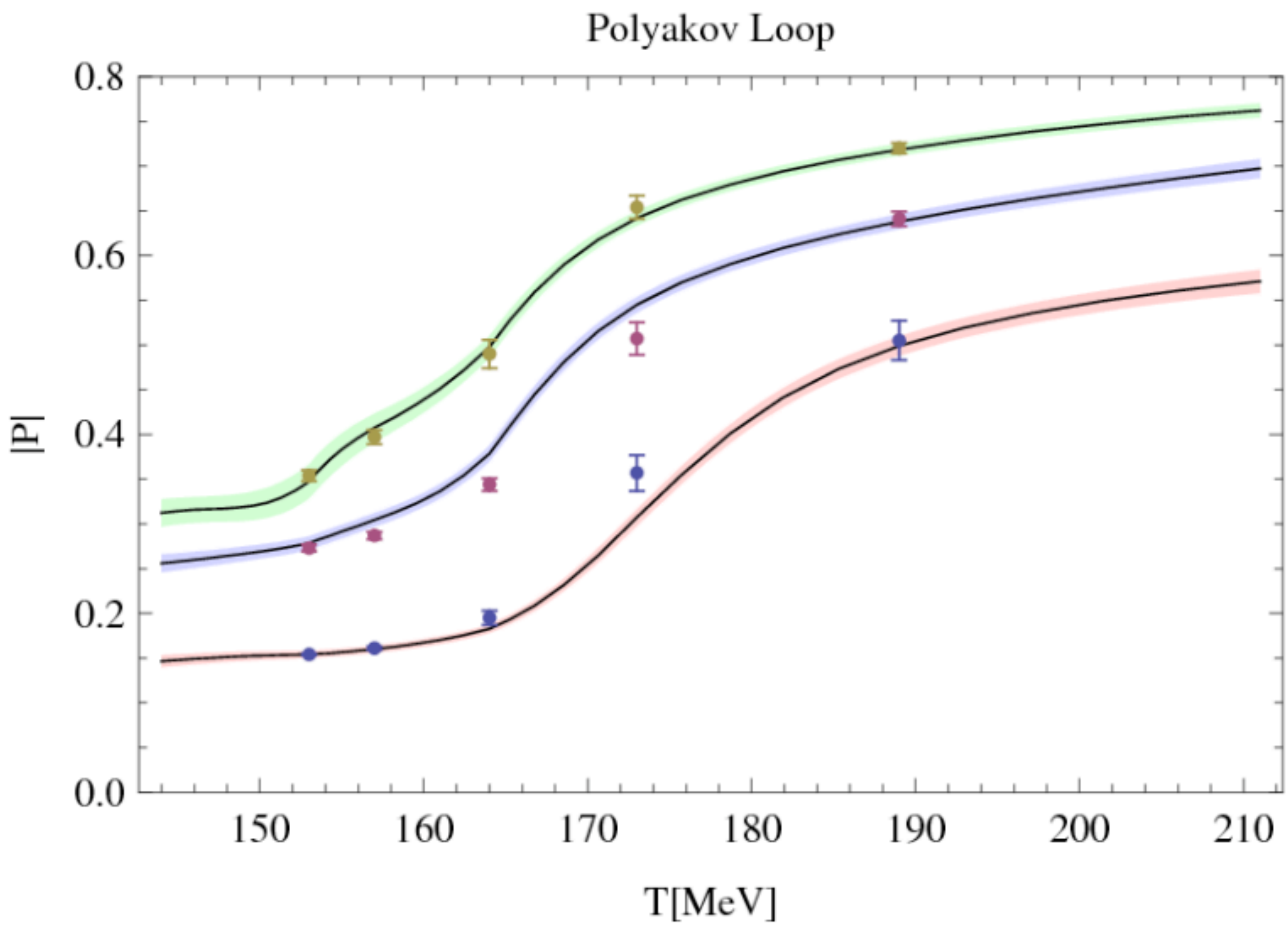




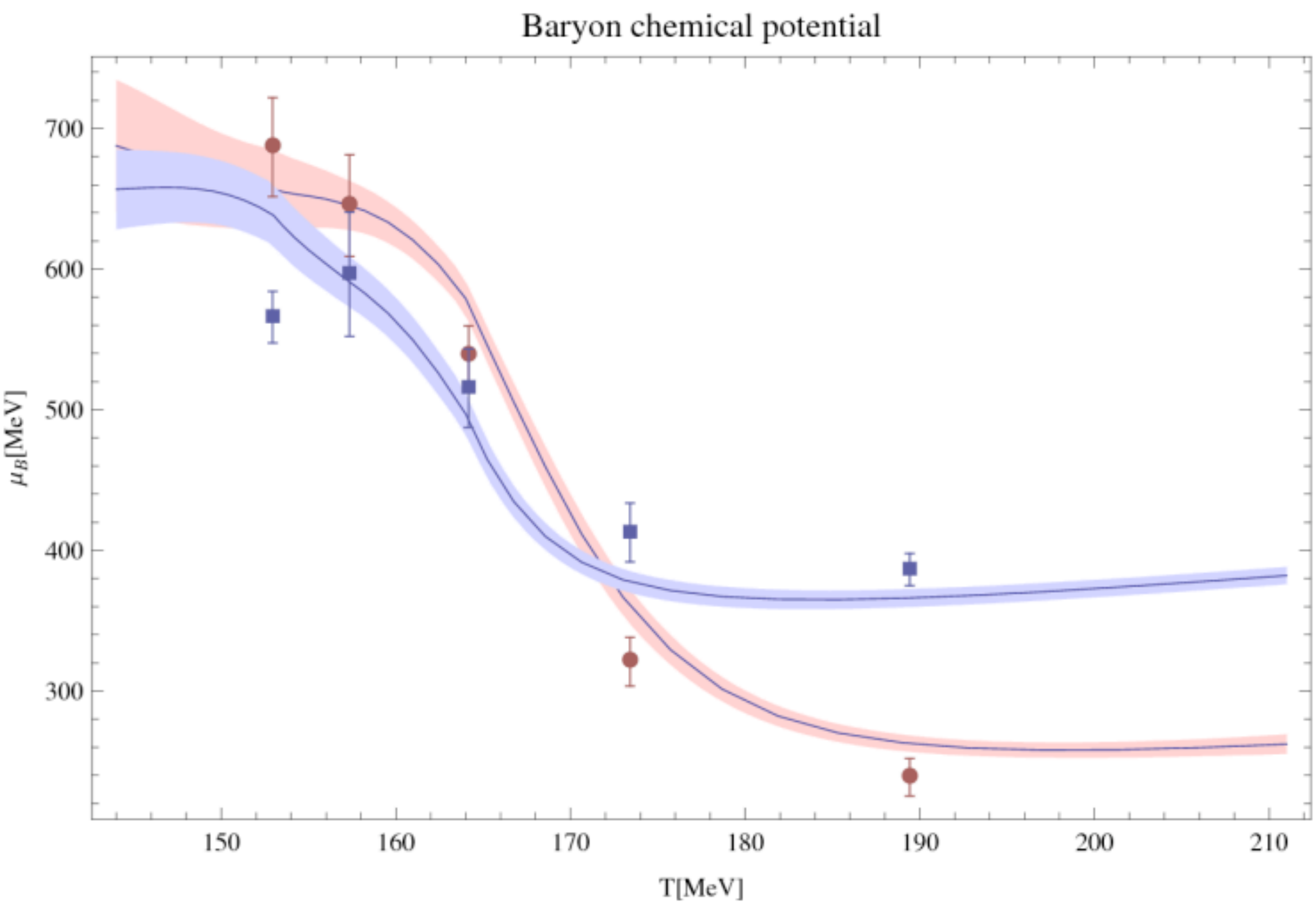
Simulation details

- We run the program at the same parameters as in our previous study: kappa=0.158, N=12 and the same values of beta
- On 4^4 lattices the estimator takes less time than the exact calculation (100s vs 140s)
- We used small HMC trajectories so that the gauge acceptance rate stays above 50%.
- The acceptance rate for gauge updates matches or is slightly less that in the exact runs using the same trajectory length
- The acceptance rates for the stochastic field is very high 65%-95% (this indicates that the estimator has small variance)

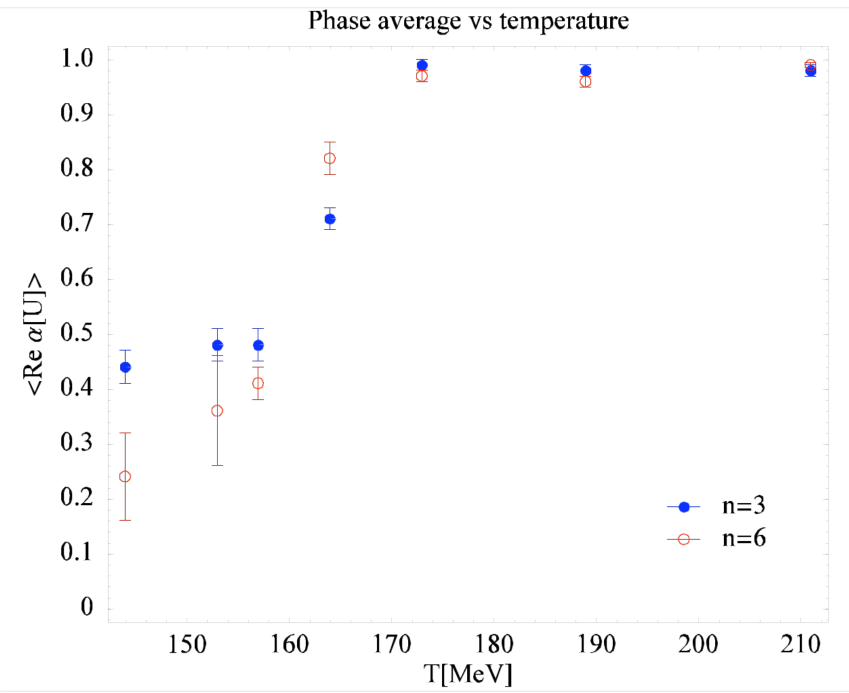
Estimator vs exact simulations

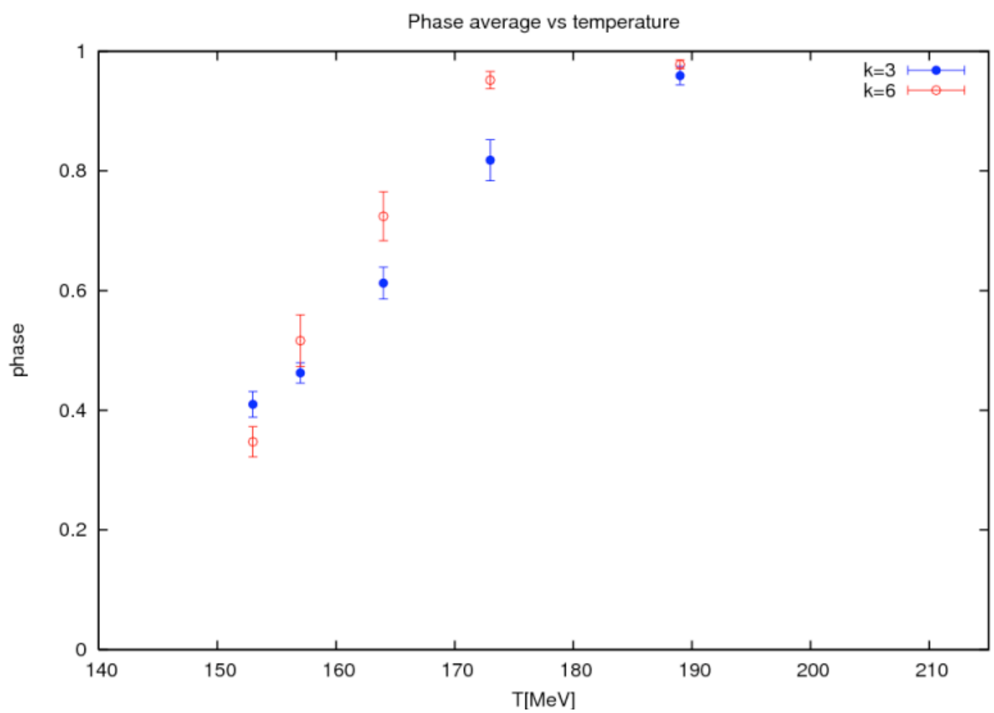


Estimator vs exact simulations

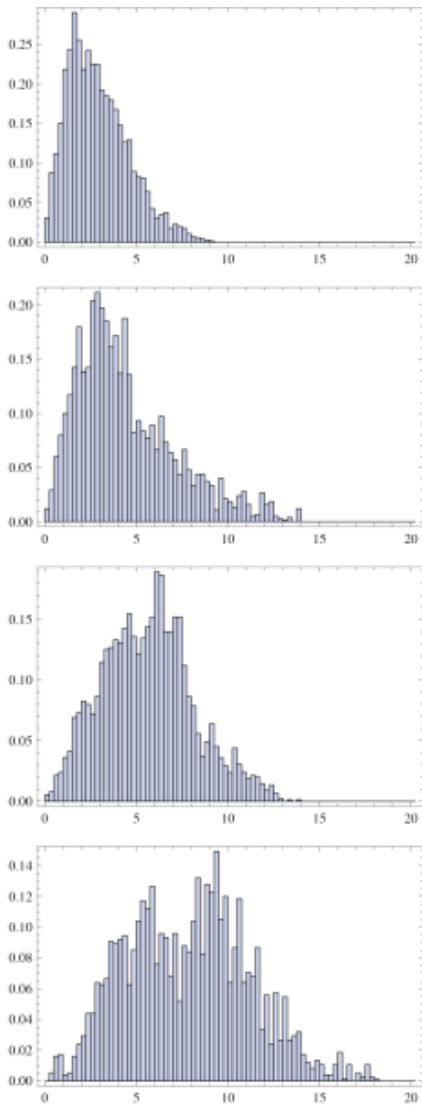


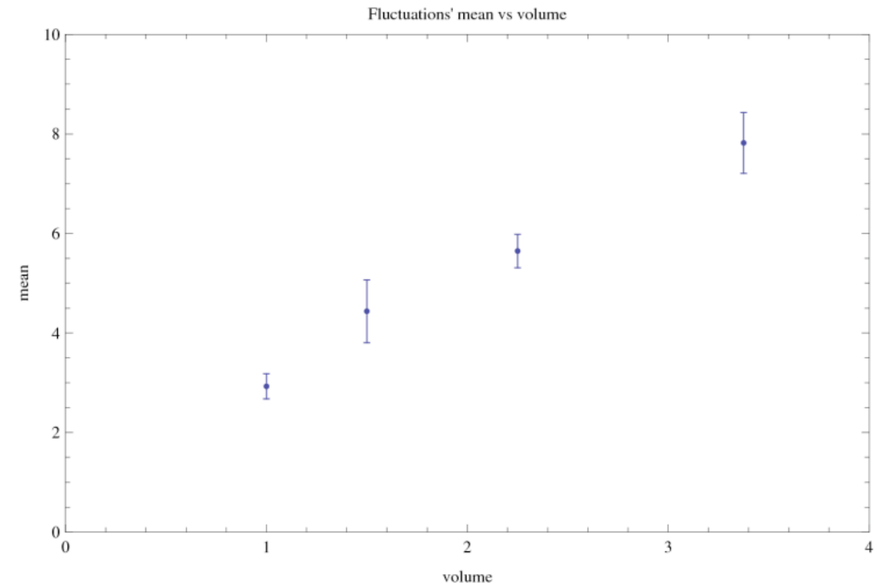
Sign problem

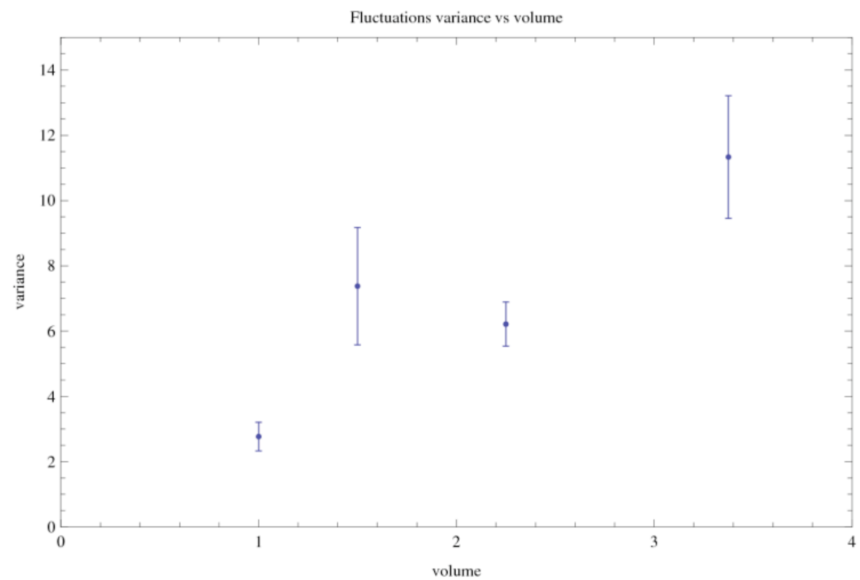




Volume dependence







Algorithm scaling

- We find that our algorithm should scale with V^3:
  - one factor of V because the dirac matrix inversion scales with V
  - one factor of V to keep the same density.
  - one factor of V because the breakup level has to be increased with V
- In conclusion the estimator method scales with V^3 -- still better than the exact method's V^4
- Since already on 4^4 lattices the estimator method is faster for 6^3x4 this method seems to be the clear winner.

Conclusions and outlook

- The stochastic algorithm is correct and it is faster than the exact one.
- The sign oscillations are comparable to the ones in the previous study.
- The algorithm scales with $V^{3.}$
- The new algorithm makes 6^3x4 simulations feasible.
- We plan to run this new algorithm 6^3x4 lattices and scan the parameter space looking for the phase transition line.