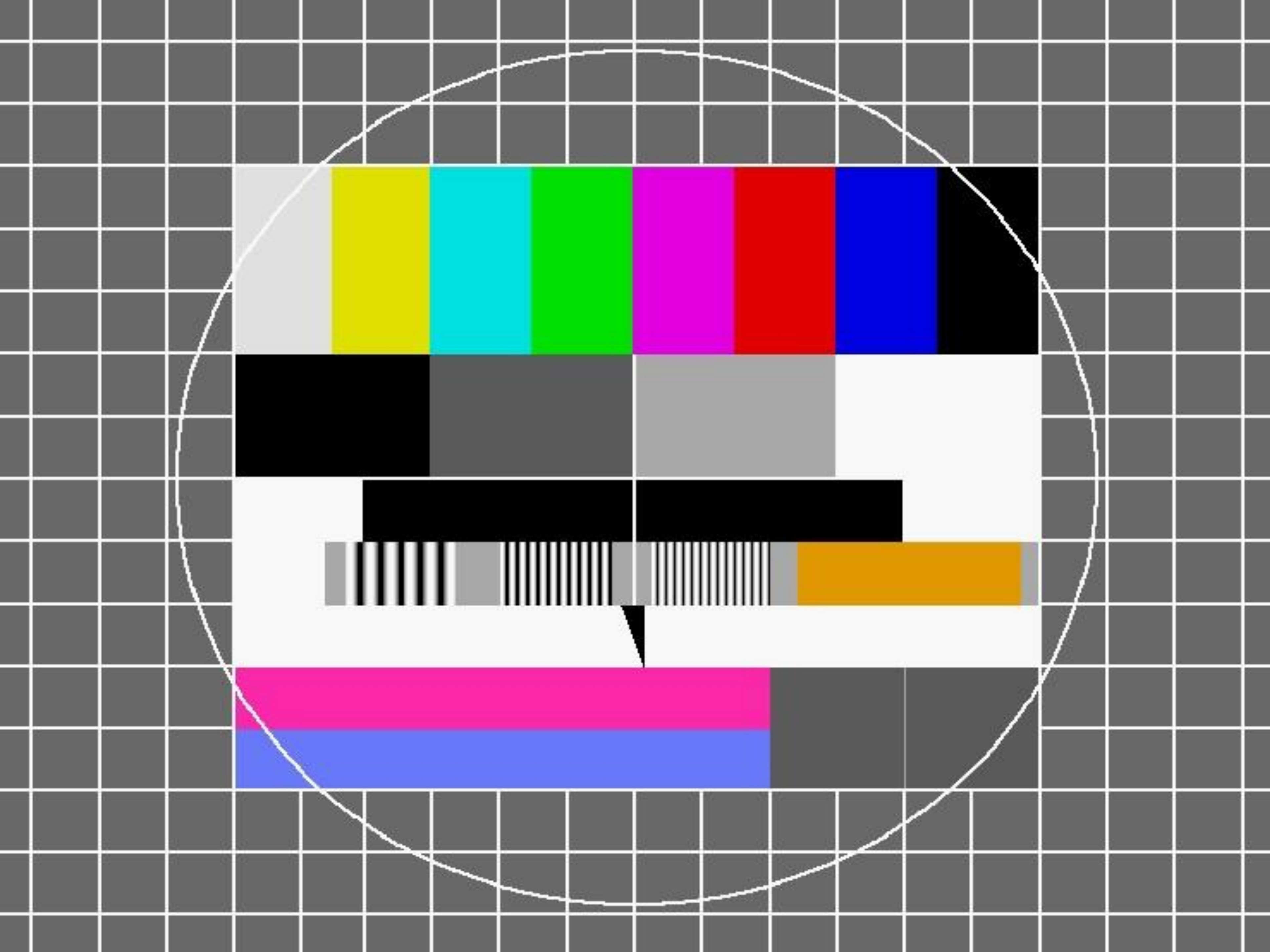


CENTRE FOR QUANTUM TECHNOLOGY, SINGAPORE

CLARENDON LABORATORY AND KEBLE COLLEGE, UNIVERSITY OF OXFORD

INSTITUTE FOR SCIENTIFIC INTERCHANGE, TURIN

Tensor networks vs. Monte Carlo for stochastic processes

Tomi Johnson

19th November 2014, UCL

Overview

1. The system – classical stochastic and pure quantum dynamics

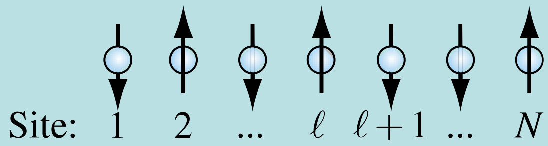
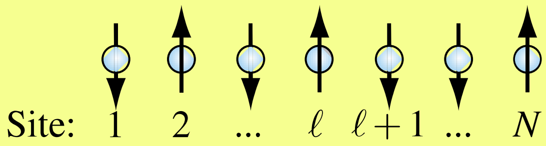
2. Our method – Tensor networks

3. Main rival – Dynamical Monte Carlo

4. Comparison for high variance observables

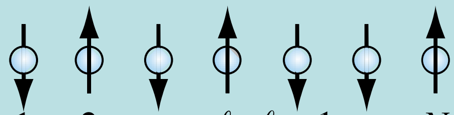
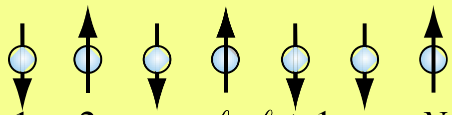
Representing the state of a system

Introducing the basics

	Quantum (pure states)	Stochastic (probabilistic mixtures)
		
Local configs (e.g. spin ½)	z_ℓ, d values $z_\ell = \pm 1$, $d = 2$	z_ℓ, d values $z_\ell = \pm 1$, $d = 2$
Global configs	$\mathbf{z} = (z_1, \ldots, z_N)$ $D = d^N$	$\mathbf{z} = (z_1, \ldots, z_N)$ $D = d^N$
State vector	$\lvert\psi\rangle = \sum_{\mathbf{z}} \psi(\mathbf{z}) \lvert\mathbf{z}\rangle$	$\lvert P\rangle = \sum_{\mathbf{z}} P(\mathbf{z}) \lvert\mathbf{z}\rangle$
Probabilities	$\lvert\psi(\mathbf{z})\rvert^2$	$P(\mathbf{z})$

Evolving the state of a system

Introducing the basics

	Unitary evolution	Markovian stochastic process
		
Master equation		$\frac{\partial P(\mathbf{z},t)}{\partial t} = \sum_{\mathbf{z}'\neq\mathbf{z}} \left(P(\mathbf{z}',t)H(\mathbf{z}',\mathbf{z}) - P(\mathbf{z},t)H(\mathbf{z},\mathbf{z}')\right)$
Rearrangement		$\langle\mathbf{z}\vert H\vert\mathbf{z}'\rangle = H(\mathbf{z}',\mathbf{z})$ for $\mathbf{z}'\neq\mathbf{z}$, $\langle\mathbf{z}\vert H\vert\mathbf{z}\rangle = -\sum_{\mathbf{z}'\neq\mathbf{z}} H(\mathbf{z},\mathbf{z}')$.
Evolution equation	$i\hbar\frac{\partial}{\partial t}\vert\psi(t)\rangle = H\vert\psi(t)\rangle$	$\frac{\partial}{\partial t}\vert P(t)\rangle = H\vert P(t)\rangle$
Evolution operator	$\vert\psi(t)\rangle = e^{-iHt/\hbar}\vert\psi(0)\rangle$	$\vert P(t)\rangle = e^{Ht}\vert P(0)\rangle$

Solving the representation problem

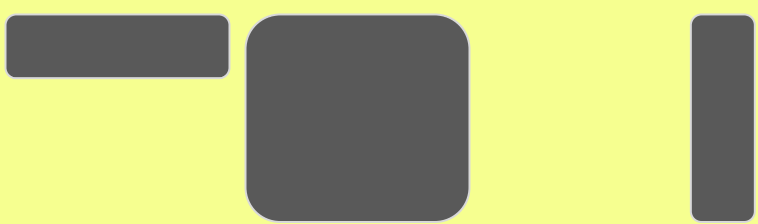
Recall: *Dimension explosion*

$$D = d^N$$

Mean Field $P(\mathbf{z}) = P^{[1]z_1} P^{[2]z_2} \dots P^{[N]z_N}$

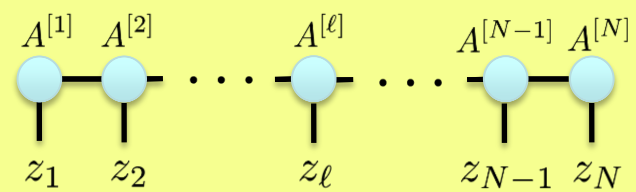
Matrix Product State

$$P(\mathbf{z}) = A^{[1]z_1} \quad A^{[2]z_2} \quad \dots \quad A^{[N]z_N}$$



$(1 \times \chi)$ $(\chi \times \chi)$ $(\chi \times 1)$

MPS as a tensor network



$$Nd\chi^2$$

Gibbs distribution compressibility

Gibbs distribution for e.g. the 1D Ising model

$$P(\mathbf{z}) = \exp[-\beta E(\mathbf{z})], \qquad E(\mathbf{z}) = -J \sum_{\ell=1}^{N} z_\ell z_{\ell+1} - \lambda \sum_{\ell=1}^{N} z_\ell$$

It factorises

$$P(\mathbf{z}) = A^{[1]z_1} \ldots A^{[\ell]z_\ell} \ldots A^{[N]z_N}$$

$1 \qquad 1 \times 2 \qquad 2 \times 2 \qquad 2 \times 1$

$$A^{[1]z_1}_{\mu_1} = \exp\left[\beta z_1 (J\mu_1/2 + \lambda)\right]$$

$$A^{[\ell]z_\ell}_{\mu_{\ell-1}\mu_\ell} = \exp\left[\beta z_\ell (J(\mu_{\ell-1} + \mu_\ell)/2 + \lambda)\right]$$

$$A^{[N]z_N}_{\mu_{N-1}} = \exp\left[\beta z_N (J\mu_{N-1}/2 + \lambda)\right]$$

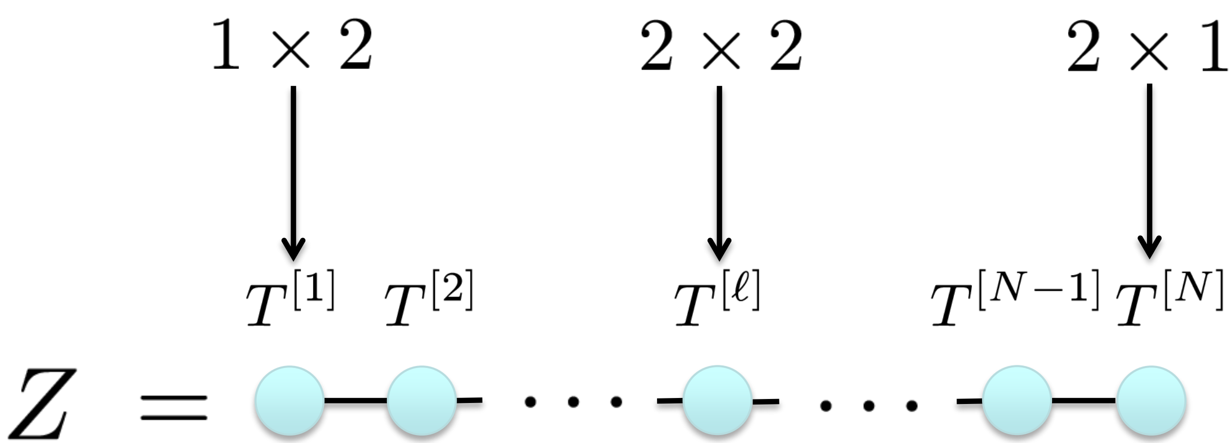
$$\chi = d = 2$$

$$\mu_\ell \in \{-1, 1\}$$

Transfer matrices

Transfer matrix method

$$Z = \sum_{\mathbf{z}} P(\mathbf{z}) = \left(\sum_{z_1} A^{[1]z_1} \right) \cdots \left(\sum_{z_N} A^{[N]z_N} \right) = T^{[1]} \dots T^{[N]}$$

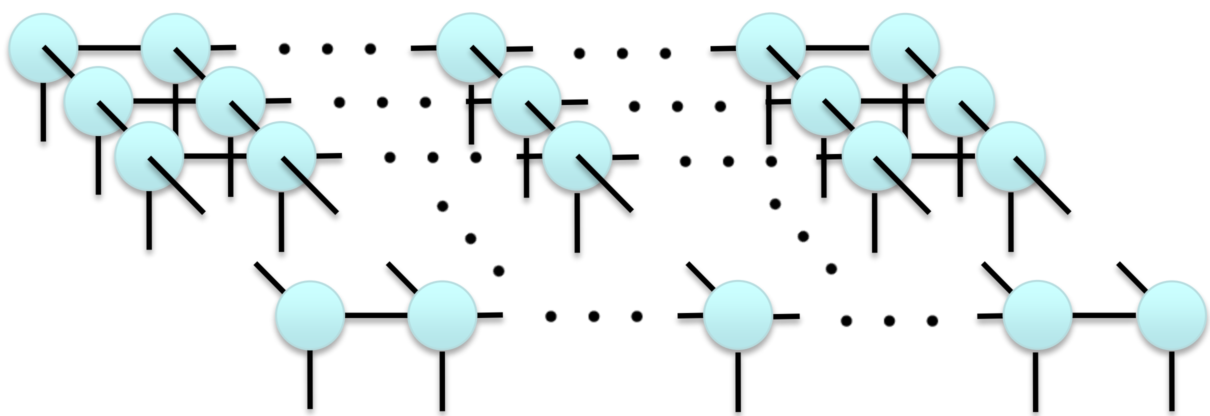


2D and beyond

Works with other geometries, e.g. 2D arrangement:

$$E(\mathbf{z}) = -J \sum_{\langle \ell \ell' \rangle} z_\ell z_{\ell'} - \lambda \sum_\ell z_\ell$$

$P =$



$$Nd^5$$

M. Levin and C. P. Nave, Phys. Rev. Lett. **99**, 120601 (2007).

Solving the evolution problem

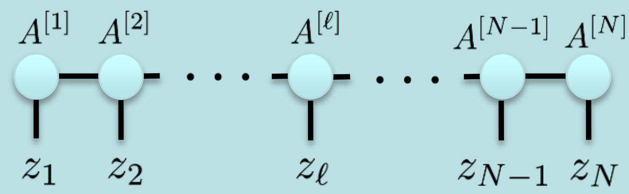
Recall: *Evolution operator*

$$|P(t)\rangle = \mathrm{e}^{Ht}|P(0)\rangle$$

Assume: *Nearest-neighbour*

$$H = \sum_{\ell=1}^{N-1} h_{\ell,\ell+1},$$

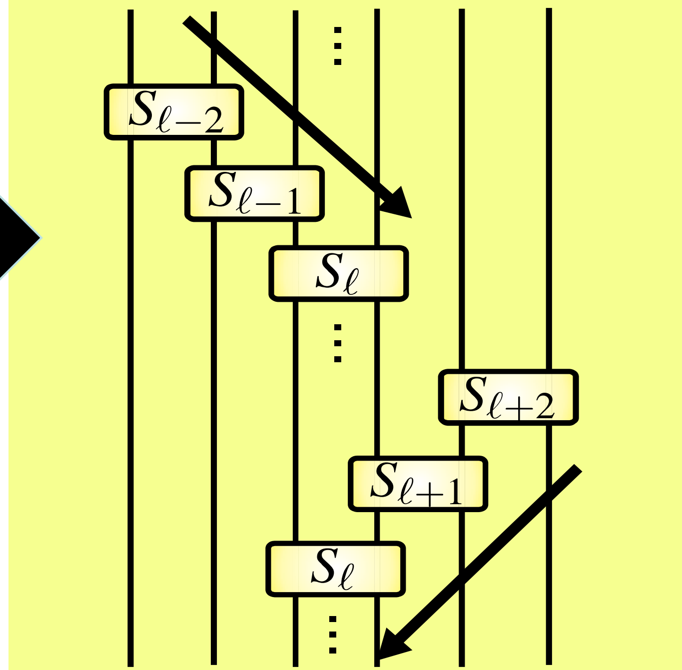
Recall: *Tensor Network*



$$\mathrm{e}^{Ht} = \mathrm{e}^{H\delta t}\mathrm{e}^{H\delta t}\cdots\mathrm{e}^{H\delta t}$$

$$\mathrm{e}^{H\delta t} = \underbrace{\left(\prod_{\ell=1}^{N-1}\mathrm{e}^{h_{\ell,\ell+1}\delta t/2}\right)}_{S_\ell}\left(\prod_{\ell=N-1}^{1}\mathrm{e}^{h_{\ell,\ell+1}\delta t/2}\right) + O\left(\delta t^3\right)$$

S_ℓ = two-site gate



No runaway of correlations in quench

Steady distributions, e.g. Gibbs, have limited correlations and are thus compressible.

A quench begins AND ENDS in a steady distribution.

Correlations can’t keep building up as for pure quantum systems.

The competitor – Dynamical Monte Carlo

Expected value

$$\langle O \rangle = \int [\mathrm{D}\mathbf{s}] O[\mathbf{s}] \mathcal{P}[\mathbf{s}]$$

Sample paths $\mathbf{s}^1, \cdots, \mathbf{s}^M$

Average $\bar{O}_M = \frac{1}{M} \sum_{m=1}^{M} O[\mathbf{s}^m]$

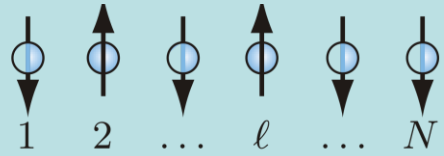
Typical error $\langle \Delta \bar{O}_M \rangle = \sqrt{\frac{\mathrm{Var}[O]}{M}}$

Number of paths needed $M = \frac{\mathrm{Var}[O]}{\epsilon \langle O \rangle^2} = \frac{v}{\epsilon}$

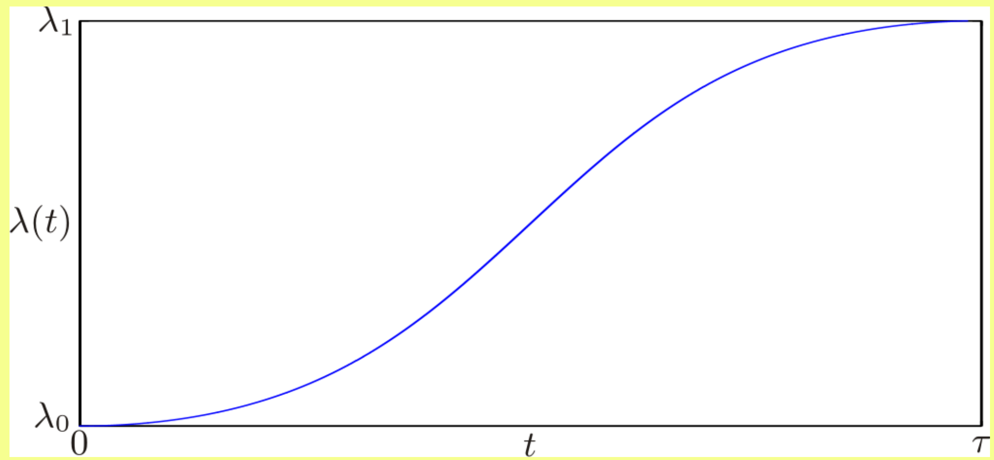
Large variance
=
Difficult

Quench

Recall: *Ising chain*



Drive away from equilibrium

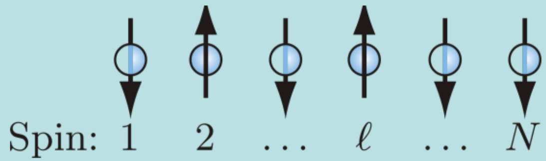


While it also thermalises

$$H(\mathbf{z},\mathbf{z}') = \frac{h}{1+\exp[-\beta(E(\mathbf{z})-E(\mathbf{z}'))]}$$

High variance observables

Recall: Ising chain



Configuration-dependent (magnetisation)

$$M(t) = \sum_{\ell} z_{\ell}(t)$$

Path-dependent (work done)

$$W(t) = -\int_0^t \mathrm{d}s M(s)\dot{\lambda}(s)$$

High-variance examples (exponentially scaling with N)

$$v = \mathrm{e}^{\mathcal{O}(N)}$$

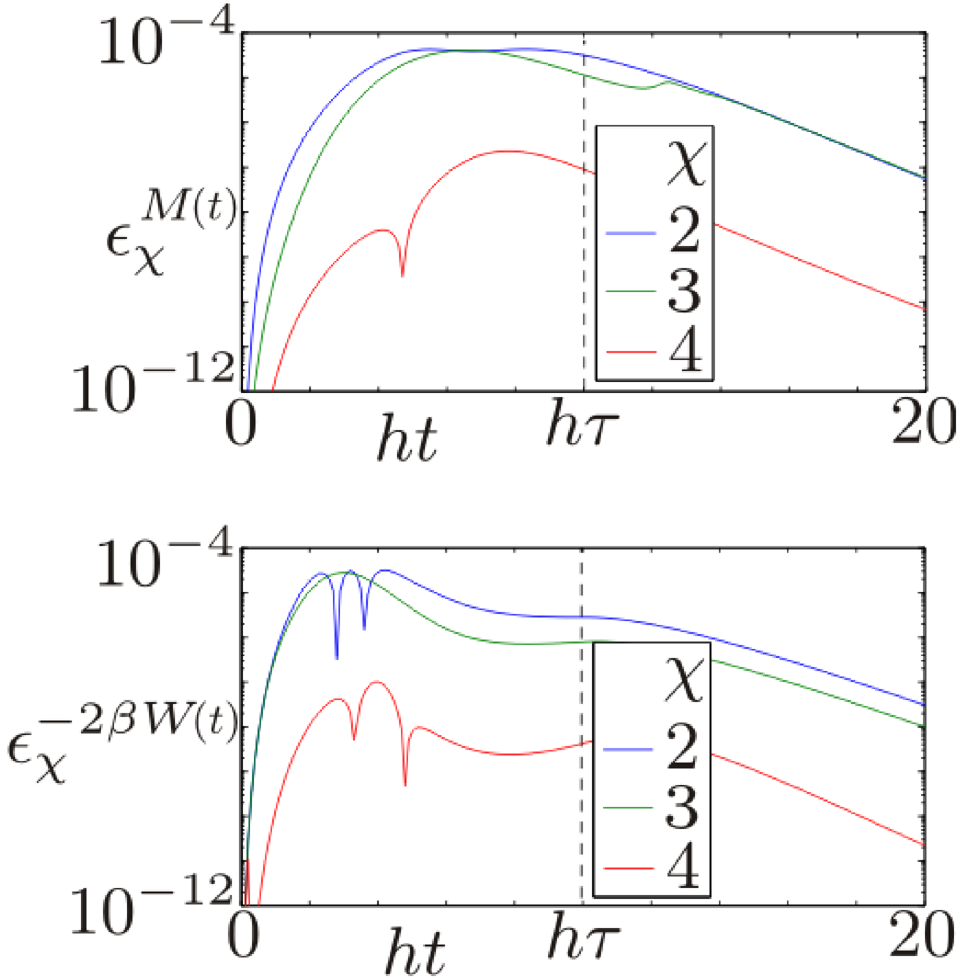
$$\mathrm{e}^{M(t)}$$

$$\mathrm{e}^{-2\beta W(t)}$$

$$\mathrm{e}^{-\beta W(t)}$$

Small system results

Fractional error due to compressibility at a single time, small system



Parameters

$$N = 8$$
$$h\tau = 10$$
$$\beta J = 1$$
$$\beta\lambda_0 = 0$$
$$\beta\lambda_1 = 1$$

Accurate compression

$$\epsilon_\chi^{-\beta W(t)} = 0$$

Exact compression

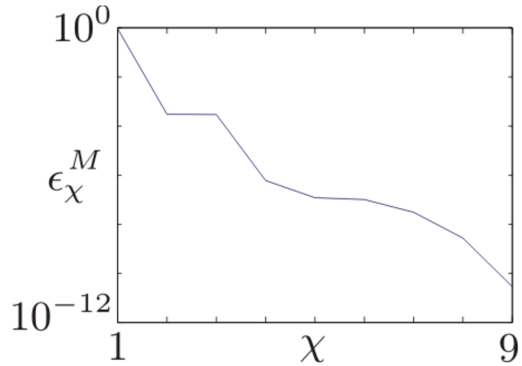
Large system results

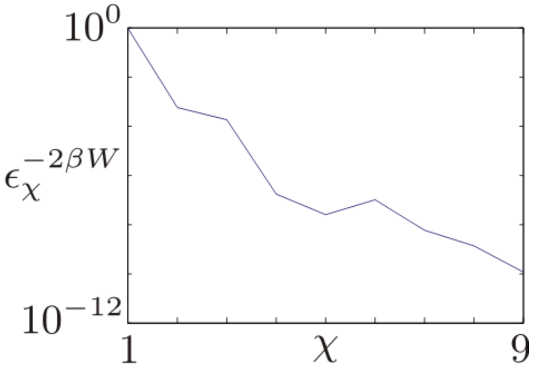
Recall: DMC and variance

$$M = \frac{\mathrm{Var}[O]}{\epsilon \langle O \rangle^2} = \frac{v}{\epsilon}$$

Parameters

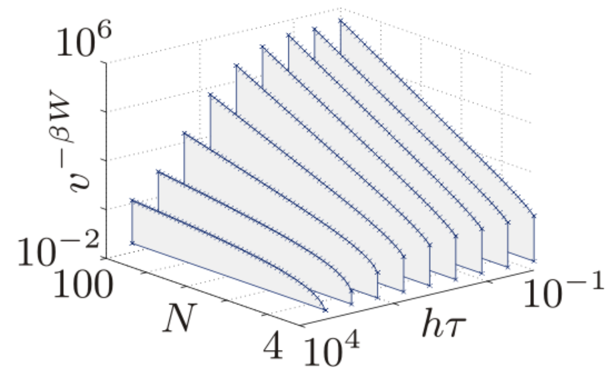
$$N = 200$$
$$h\tau = 10$$
$$ht = 10$$
$$h\delta t = 10^{-3}$$
$$\beta J = 1$$
$$\beta\lambda_0 = 0$$
$$\beta\lambda_1 = 1$$





$$\epsilon^{-\beta W} \approx \alpha N (h\delta t)^2$$

$$\alpha = 4.65 \times 10^{-2}$$



Exponentially increasing variance

Exponential samples needed to match tensor networks

Example: 200-spin calculation, taking an hour, but needing $\approx 10^{17}$ samples

Summary and future directions

1. Tensor networks simulate stochastic processes

2. Is this the way to estimate high variance observables?

3. Good option for tensor networks in 2D?

What else can we do with tensor networks – PDEs?

Thanks for listening

Thomas Elliott

Stephen Clark

Dieter Jaksch

T. H. Johnson, S. R. Clark, and D. Jaksch, Phys. Rev. E **82**, 036702 (2010)

T. H. Johnson, T. J. Elliott, S. R. Clark, and D. Jaksch, arXiv:1410.3319

Tensor Network Library, tensornetworktheory.org

Questions?