Infinite DMRG and the Matrix Product Toolkit

Ian McCulloch

University of Queensland
Centre for Engineered Quantum Systems (EQuS)

21/9/2015

Outline

Infinite MPS - a brief review

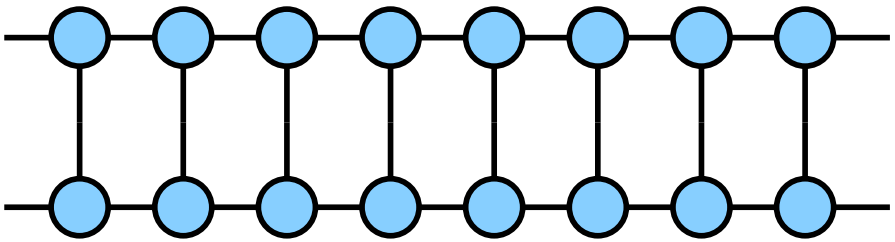
- Start with an MPS

$$|\psi\rangle = \mathrm{Tr} \sum_{s_i} A_1^{s_1} A_2^{s_2} A_3^{s_3} A_4^{s_4} \cdots |s^1\rangle\ |s^2\rangle\ |s^3\rangle\ |s^4\rangle \cdots$$

- Choose a unit cell size L, and make the MPS *infinite*
- no trace – the boundaries are at infinity
- eg, with a 2-site unit cell,

$$|\psi\rangle = \sum_{s_i} \cdots A_1^{s_1} A_2^{s_2} \quad A_1^{s_3} A_2^{s_4} \quad A_1^{s_5} A_2^{s_6} \quad A_1^{s_7} A_2^{s_8} \cdots$$

- All quantities are a limit in some sense, eg wavefunction norm



Matrix Product Operators

- At each iteration we have a set of *block operators*, acting on the m-dimensional auxiliary space
- It is natural to use a Matrix Product approach to constructing the block operators used in DMRG

Ising model $H = \sum_{<i,j>} S_i^z S_j^z + \lambda \sum_i S_i^x$, adding a site to the block:

(identity operator)	I	$\rightarrow$	$I \otimes I_{\text{local}}$
(z-spin acting on right-most site)	S^z	$\rightarrow$	$I \otimes S^z_{\text{local}}$
(block Hamiltonian)	H	$\rightarrow$	$\lambda I \otimes S^x_{\text{local}} + S^z \otimes S^z_{\text{local}} + H \otimes I_{\text{local}}$

In matrix form:

$$\underbrace{\begin{pmatrix} I & S^z & H \end{pmatrix}'}_{\text{new block operators}} = \underbrace{\begin{pmatrix} I & S^z & H \end{pmatrix}}_{\text{old block operators}} \times \underbrace{\begin{pmatrix} I & S^z & \lambda S^x \\ & & S^z \\ & & I \end{pmatrix}}_{\text{local}}$$

Matrix Product Operators

This form can represent many operators

- fermionic $c^\dagger_{k=0}$: $W_{c^\dagger_{k=0}} = \begin{pmatrix} P & c^\dagger \\ & I \end{pmatrix}$, $P = (-1)^N$, J-W string
- finite momentum $b^\dagger_k$: $W_{b^\dagger_k} = \begin{pmatrix} e^{ik} & b^\dagger \\ & I \end{pmatrix}$

Advantages of the MPO representation: *arithmetic operations!*

$H_1 + H_2$	direct sum of the MPO representations
$H_1 \times H_2$	direct product of the MPO representations

also derivatives, etc

This preserves the lower triangular form.

In the thermodynamic limit:

$$\langle A \rangle_L = \text{polynomial function of } L$$

Examples:

- Energy: $\langle H \rangle_L = L\,\epsilon$
- Hamiltonian block operator matrix elements to restart a calculation
- Single-mode approximation: $\langle S^-_k H S^+_k \rangle_L / \langle S^-_k S^+_k \rangle_L$

Matrix Product Operators

This form can represent many operators

- fermionic $c^\dagger_{k=0}$: $W_{c^\dagger_{k=0}} = \begin{pmatrix} P & c^\dagger \\ & I \end{pmatrix}$, $P = (-1)^N$, J-W string
- finite momentum $b^\dagger_k$: $W_{b^\dagger_k} = \begin{pmatrix} e^{ik} & b^\dagger \\ & I \end{pmatrix}$

Advantages of the MPO representation: *arithmetic operations!*

$H_1 + H_2$	direct sum of the MPO representations
$H_1 \times H_2$	direct product of the MPO representations

also derivatives, etc

This preserves the lower triangular form.

In the thermodynamic limit:

$$\langle A \rangle_L = \text{polynomial function of } L$$

Examples:

- Energy: $\langle H \rangle_L = L\,\epsilon$
- Hamiltonian block operator matrix elements to restart a calculation
- Single-mode approximation: $\langle S^-_k H S^+_k \rangle_L / \langle S^-_k S^+_k \rangle_L$

Types of operator

The Matrix Product Toolkit uses 3 different kinds of MPO

- *FiniteMPO* – an MPO that has finite support
  evaluation is straightforward from a canonical form

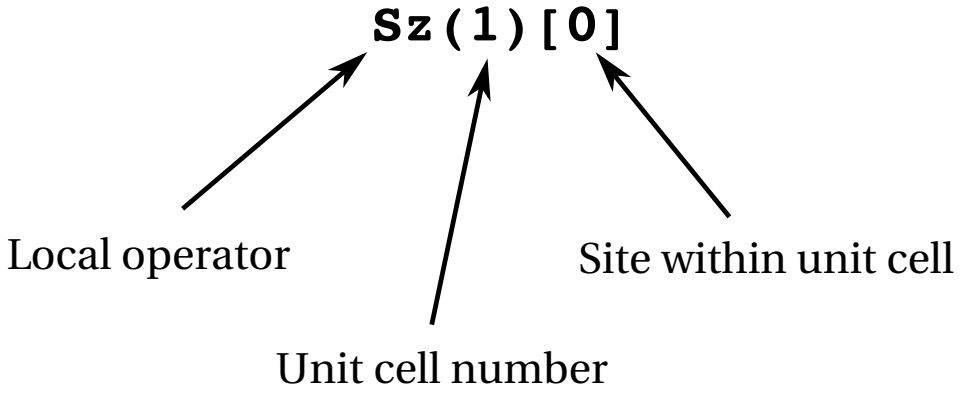
- *TriangularMPO* – an MPO in explicitly upper-triangular form
  Reasonably efficient algorithm for obtaining the expectation value as the coefficients of a polynomial function in L.

- *ProductMPO* – a representation for an infinite product of operators
  examples: Suzuki-trotter decomposition $\exp[\text{commuting terms}]$, Jordan-Wigner fermion strings, ...

Finite MPO's in the Matrix Product Toolkit

Finite MPO's are constructed using C++ or on the command line

- *Local operators* – defined on a single site (C++ class `LatticeSite`)



- *expressions* `+ * ^ exp`, commutator brakets, $SU(2)$ vector operations...
- some built-in operators, such as swap gates: `swap(a,b)[i,j]` – swap site i of unit cell a with site j of unit cell b.
- All user-defined operators are defined relative to a unit cell

Example: `BondCurrent(0) = i*(BH(0)*B(1) - B(0)*BH(1))`

- Then `BondCurrent(n)` will represent `i*(BH(n)*B(n+1) - B(n)*BH(n+1))`
- If the unit cell is a single site, then local operators are automatically 'promoted' to unit cell operators. `BH(n)` instead of `BH(n)[0]`

Triangular MPO's

Triangular MPO's represent infinite sums and polynomials of finite operators

- the basic way to construct a TriangularMPO is a sum over unit cells

$$S^z_{\text{total}} = \sum_n S^z(n)$$

```
TriangularMPO SzTotal = sum_unit(Sz(0));
```

- Finite momentum

$$S^z_k = \sum_n e^{ikn} S^z(n)$$

```
TriangularMPO Sz_k05 = sum_k(0.5*pi,Sz(0));
```

- 'kink' operators

$$S^z_{\text{kink}} = \sum_n \left(\prod_{j<n} (-1)^{S^z(j)} \right) S^z(n)$$

```
sum_kink(exp(i*pi*Sz(0)), Sz(0))
```

- and various expressions, sum, product, ...

Product MPO's

A ProductMPO is a repeated product of operators, often unitary.

- construct as a product over unit cell operators

$$P = \prod_j (-1)^N$$

```
ProductMPO P = prod_unit(exp(i*pi*N(0)))
```

- or as a string of local operators

```
string(exp(i*pi*N))
```

ProductMPO's are not used so often – but useful for string order parameters, probing symmetries of symmetry-protected topological states, etc

Limited set of operations

- *no* addition
- *no* scalar multiplication
- can multiply two ProductMPO's

Lattice files

The *lattice file* is a C++ program that defines:

- one more more `LatticeSite`'s – defines local Hilbert space for a single site and associated operators
- The `UnitCell`, as an array of `LatticeSite`'s
- additional operators

```
LatticeSite Site = SpinSite(0.5);
UnitCell Cell(Site);
InfiniteLattice Lattice("Spin chain", Cell);
UnitCellOperator Sx(Cell, "Sx"),
                 Sy(Cell, "Sy"),
                 Sz(Cell, "Sz");

Lattice["H_xx"] = sum_unit(Sx(0)*Sx(1));
Lattice["H_yy"] = sum_unit(Sy(0)*Sy(1));
Lattice["H_zz"] = sum_unit(Sz(0)*Sz(1));
```

Function declarations

- Can also define *functions*

```
Lattice.function("H")
    ("K", arg("alpha")=0.0, arg("U")=0.0, arg("J")=1.0)
 = "J*(H_J1 + cos(pi*alpha)*(H_J2 + H_J0)
  + sin(pi*alpha)*(H_Jc2 - H_Jc0))
  + K*H_K + U*H_U";
```

- use `mp-lattice-info` to see what a lattice contains

```
Description: Bosonic 3-leg ladder with flux
Command line: bosehubbard-flux-3leg-u1 -N 1 -o lat
Date: Thu, 17 Sep 2015 15:41:50 +0200
SymmetryList: N:U(1)
Unit cell size: 3
   site [0] is: Boson, maximum number of particles per site = 1
   site [1] is: Boson, maximum number of particles per site = 1
   site [2] is: Boson, maximum number of particles per site = 1

Unit cell operators:
   I           - transforms: 0, commutes: Bosonic
   j0          - transforms: 0, commutes: Bosonic
...
Lattice operators:
   H_J         - nearest neighbor leg hopping = H_J0 + H_J1 + H_J2
   H_J0        - nearest neighbor leg 0 hopping
   H_J1        - nearest neighbor leg 1 hopping
.....
Lattice functions:
   H           - Hamiltonian, parametized by K, alpha (flux), U, J
   H{K, alpha=0, U=0, J=1} = J*(H_J1 + cos(pi*alpha)*(H_J2 + H_J0) + sin(pi*alpha)*(H_Jc2 - H_Jc0)) + K*H_K + U*H_U
```

iDMRG

Currently two iDMRG programs – `mp-idmrg` (two-site updates) and `mp-idmrg-s3e` (one-site updates)
Input configuration:

- Hamiltonian operator
  example: `lattice:"H_xx + H_yy + H_zz"`
- Either:
  - an existing iMPS wavefunction
  - or specify the wavefunction unit cell size and quantum number for a random initial state `-b -a -u <unitcellsize> -q <quantumnumber>`

  Note: the wavefunction unit cell can be an arbitrary multiple of the Hamiltonian unit cell

Still in development:

- stopping criteria
- checkpoint/restart
- MPI
- multi-threading (but works reasonably well with multi-threaded BLAS/LAPACK)

Expectation values

Finite operators:

- `mp-iexpectation` works in much the same way as for finite DMRG
  ```
  mp-iexpectation psi lattice:"Sz(0)*Sz(100)"
  mp-iexpectation psi lattice:
  "(Sz(0)*exp(i*pi*(Sz(1)+Sz(2)+Sz(3)))*Sz(4))^2"
  ```
- might eventually be merged and `mp-expectation` will do both finite and infinite MPS (but allowed MPO operations are different...)
- for correlation functions, better to look at the transfer matrix spectrum

Product operators:

- use in conjunction with calculating overlaps – see next slide

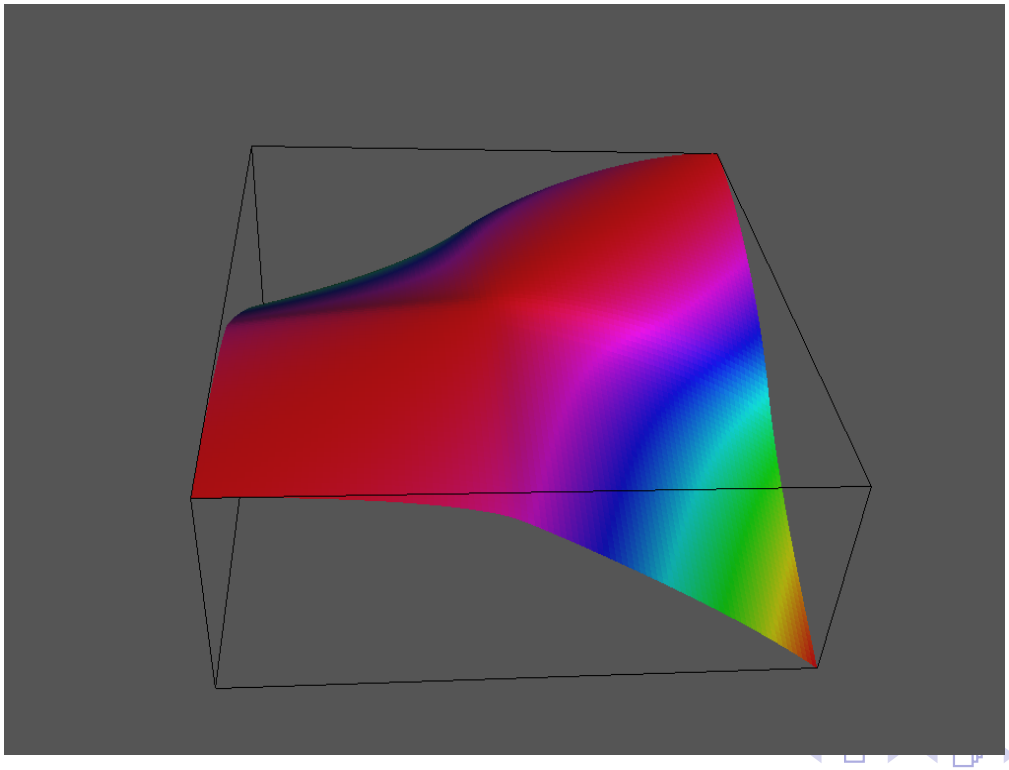
Triangular operators:

- separate tool – `mp-imoments`

Overlap of iMPS

In the thermodynamic limit, two iMPS are either the identical state, or they are orthogonal.
The asymptotic overlap is obtained as d^L, where $|d| \leq 1$, and L is the number of unit cells
The *fidelity per site* is sometimes suggested as a measure for probing phase transitions,

$$\langle \psi(a) | \psi(b) \rangle$$



Overlap of iMPS

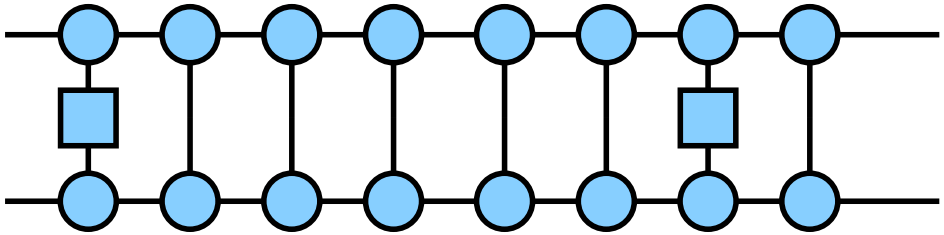
The tool `mp-ioverlap` calculates the overlap of two wavefunctions

- *symmetries*: need to calculate overlap in each symmetry sector
- symmetry sector depends on the nature of the two wavefunctions; generally not possible to predict which sector is meaningful
- overlap of a wavefunction with itself always has eigenvalue 1 in the scalar symmetry sector
- overlap in other symmetry sectors is a fast way of calculating the correlation length
- can include a ProductMPO as a string term

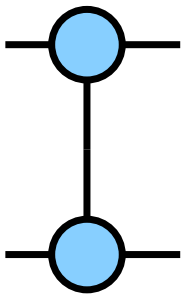
In fact, the overlap gives an entire spectrum, not just a single number.

- not sure if the remaining spectrum is meaningful for overlaps between different wavefunctions – `mp-ioverlap` only calculates one eigenvalue in each symmetry sector (but easy to extend this if it is useful)

Transfer matrix spectrum



The form of correlation functions are determined by the eigenvalues of the *transfer operator*



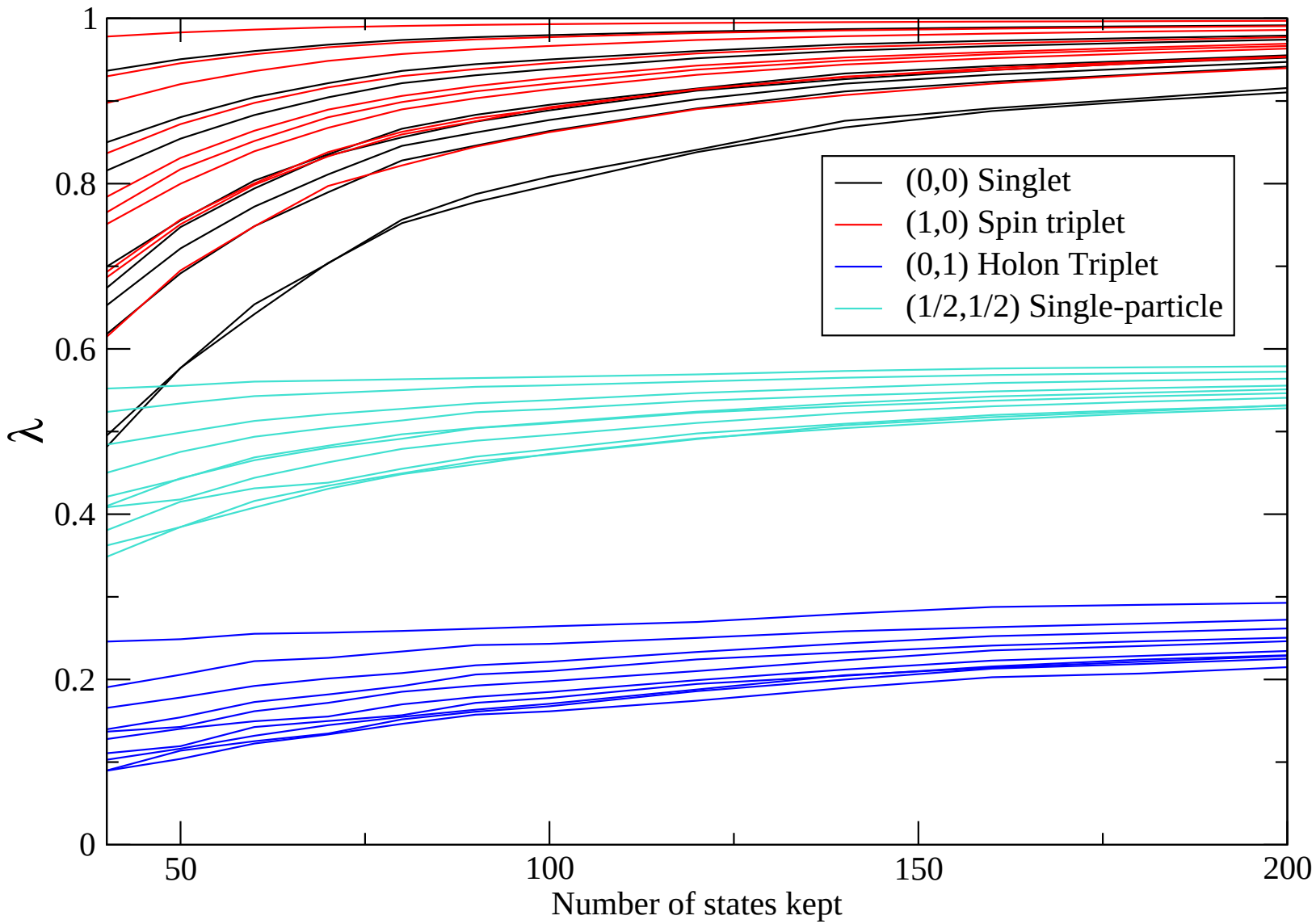
- All eigenvalues ≤ 1
- One eigenvalue equal to 1, corresponding to the identity operator

Expansion in terms of eigenspectrum λ_i:

$$\langle O(x)O(y)\rangle = \sum_i a_i \lambda_i^{|y-x|}$$

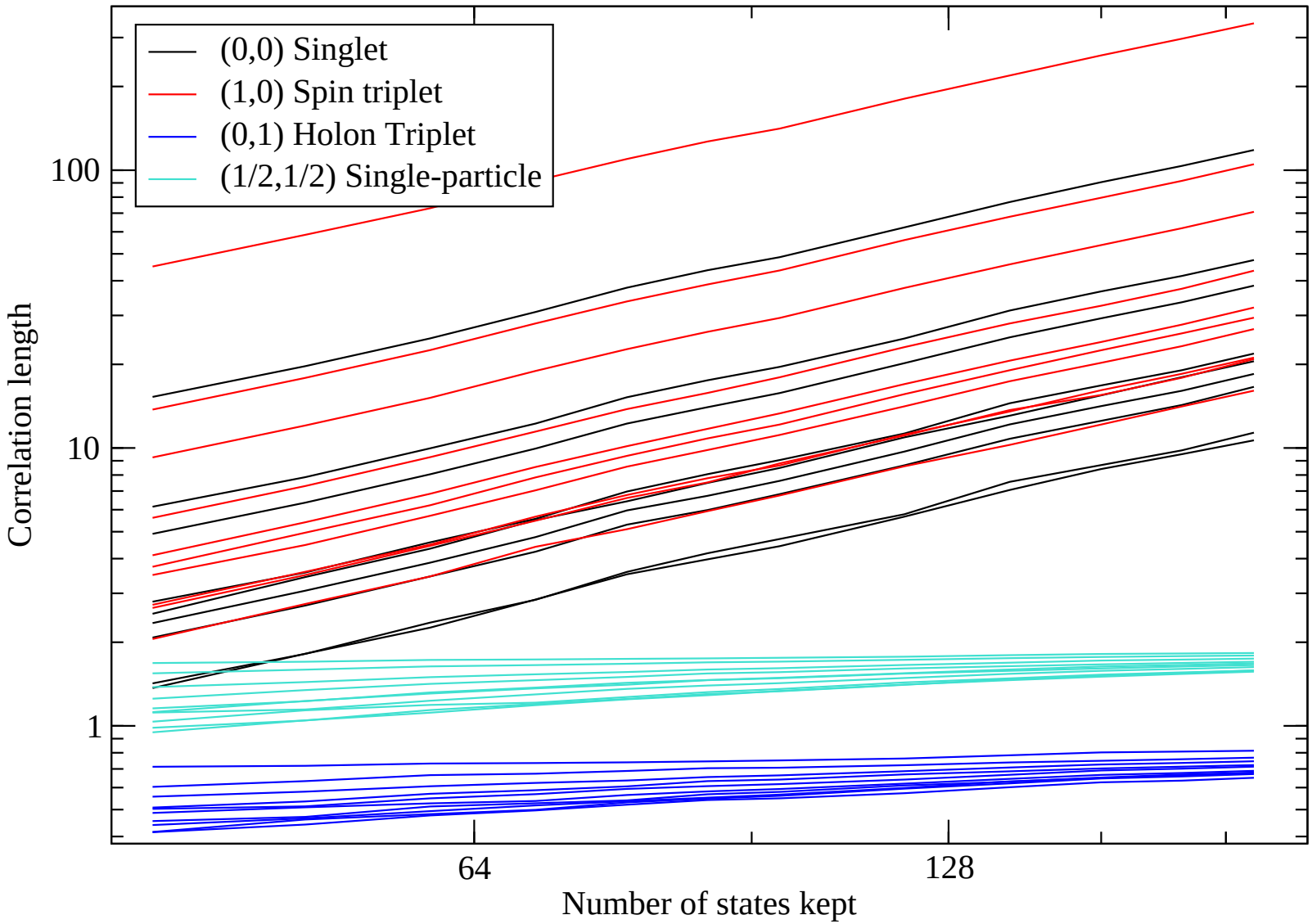
Hubbard Model transfer matrix spectrum

Half-filling, U/t = 4



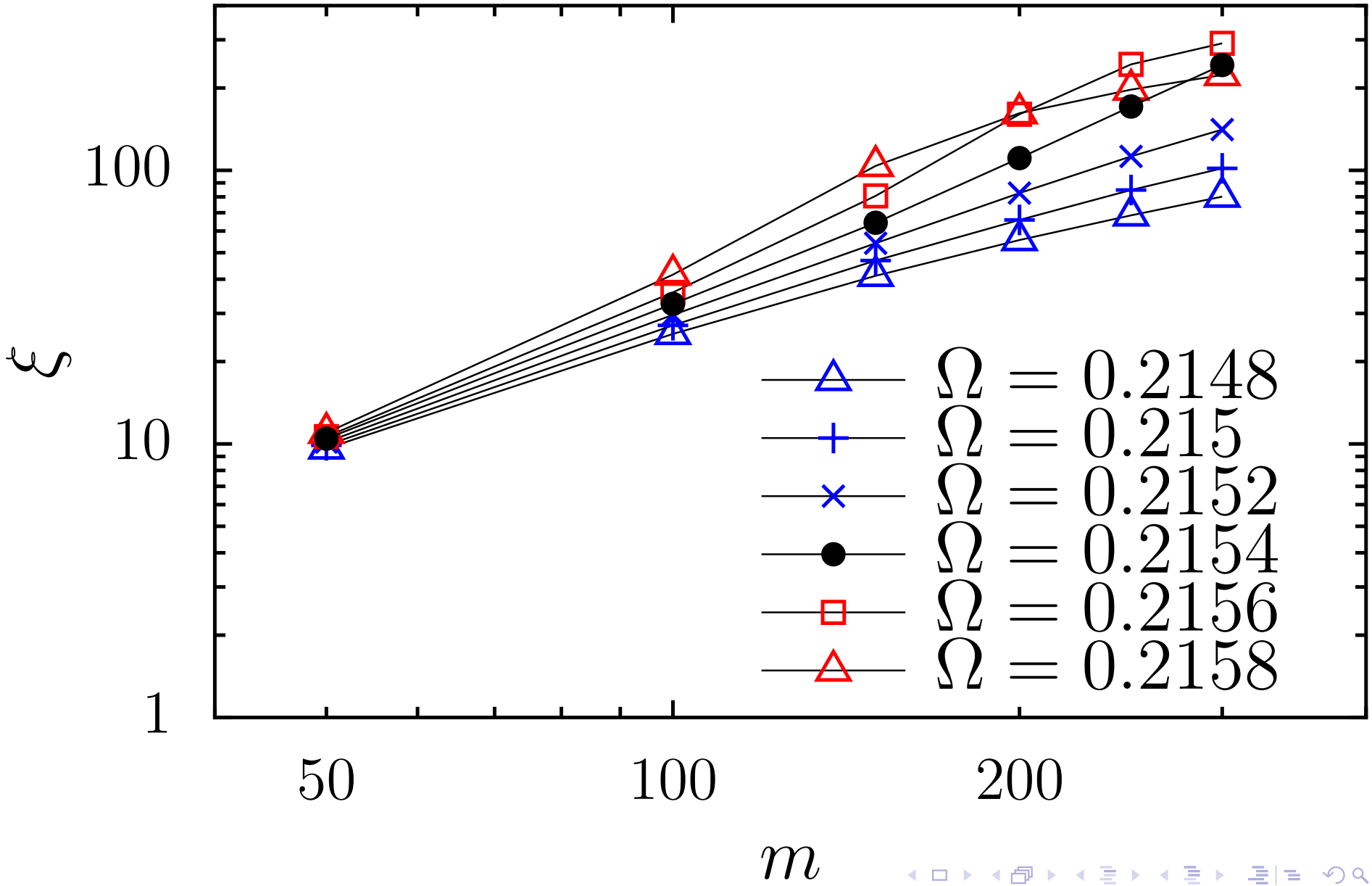
Hubbard model transfer matrix spectrum

Half-filling, U/t=4



Critical scaling example

Two-species bose gas with linear tunneling Ω, from F. Zhan et al, Phys. Rev. A 90, 023630 2014



Transfer matrix spectrum

The transfer matrix spectrum is obtained using `mp-ispectum`

- default operation calculates 10 eigenvalues in each symmetry sector
- optional ProductMPO to use as a string operator
- can also calculate expansion coefficients c_λ of finite operators

$$\langle A(0)B(n)\rangle = \sum_\lambda c_\lambda^{AB}\ \lambda^n$$

- Example: spin-1 chain string order parameter:

```
mp-ispectrum psi -q 0 --real --corr --string lat:"prod_unit(exp(i*pi*(Sz(0))))"
--left lat:"Sz(0)" --right lat:"Sz(0)"

#sector     #n   #real                 #corr_length          #overlap_0_0_real
0           0    1                     -2.8147497671066e+14  -0.37432512029978
0           1    -0.83444514484204     5.5252198820973       -1.9549209642218e-23
0           2    -0.79566947093406     4.3750002235372       -1.0129973942408e-23
0           3    -0.72968550984391     3.1731762919434       -8.9289307735515e-23
0           4    0.66983859476809      2.4955174333598       2.3215621446014e-15
0           5    0.66712025003964      2.4704474917566       -0.0051910662146261
0           6    0.65888947798557      2.3969349706944       -0.0011925525910177
0           7    -0.63571357447144     2.2074705711093       -7.3219774753827e-22
0           8    0.60464705041238      1.9876354014288       -1.3146871827368e-17
0           9    0.60383398633243      1.9823335390762       7.7647474263361e-18
```

CFT Parameters

For a critial mode, the correlation length increases with number of states m as a power law,

$$\xi \sim m^{\kappa}$$

[T. Nishino, K. Okunishi, M. Kikuchi, Phys. Lett. A **213**, 69 (1996)
M. Andersson, M. Boman, S. Östlund, Phys. Rev. B **59**, 10493 (1999)
L. Tagliacozzo, Thiago. R. de Oliveira, S. Iblisdir, J. I. Latorre, Phys. Rev. B **78**, 024410 (2008)]

This exponent is a function *only* of the central charge,

$$\kappa = \frac{6}{\sqrt{12c} + c}$$

[Pollmann et al, PRL 2009]

Note: in practice this usually isn't a good way to determine c – better to use entropy scaling

Scaling dimensions

Suppose we have a two-point correlator that has a power-law at large distances

$$\langle O(x)O(y)\rangle = |y-x|^{-2\Delta}$$

As we increase the number of states kept m the correlation length increases, so the region of validity of the power law increases.

- Take two different calculations with m_1 and m_2
- Correlation lengths ξ_1 and ξ_2
- We expect: $\dfrac{O(\xi_2)}{O(\xi_1)} = \left(\dfrac{\xi_2}{\xi_1}\right)^{\Delta}$
- for x large, we have: $O(x) \simeq a\,\lambda^x$ (with $\xi = -1/\ln\lambda$)
- Prefactor a is overlap of operator O with next-leading eigenvector of transfer operator

$$a \propto \xi^{-\Delta}$$

This gives directly the operator scaling dimensions by direct fit

Scaling dimensions

Suppose we have a two-point correlator that has a power-law at large distances

$$\langle O(x)O(y)\rangle = |y-x|^{-2\Delta}$$

As we increase the number of states kept m the correlation length increases, so the region of validity of the power law increases.

- Take two different calculations with m_1 and m_2
- Correlation lengths ξ_1 and ξ_2
- We expect: $\dfrac{O(\xi_2)}{O(\xi_1)} = \left(\dfrac{\xi_2}{\xi_1}\right)^{\Delta}$
- for x large, we have: $O(x) \simeq a\ \lambda^x$ (with $\xi = -1/\ln\lambda$)
- Prefactor a is overlap of operator O with next-leading eigenvector of transfer operator

$$a \propto \xi^{-\Delta}$$

This gives directly the operator scaling dimensions by direct fit

Scaling dimensions

Suppose we have a two-point correlator that has a power-law at large distances

$$\langle O(x)O(y)\rangle = |y-x|^{-2\Delta}$$

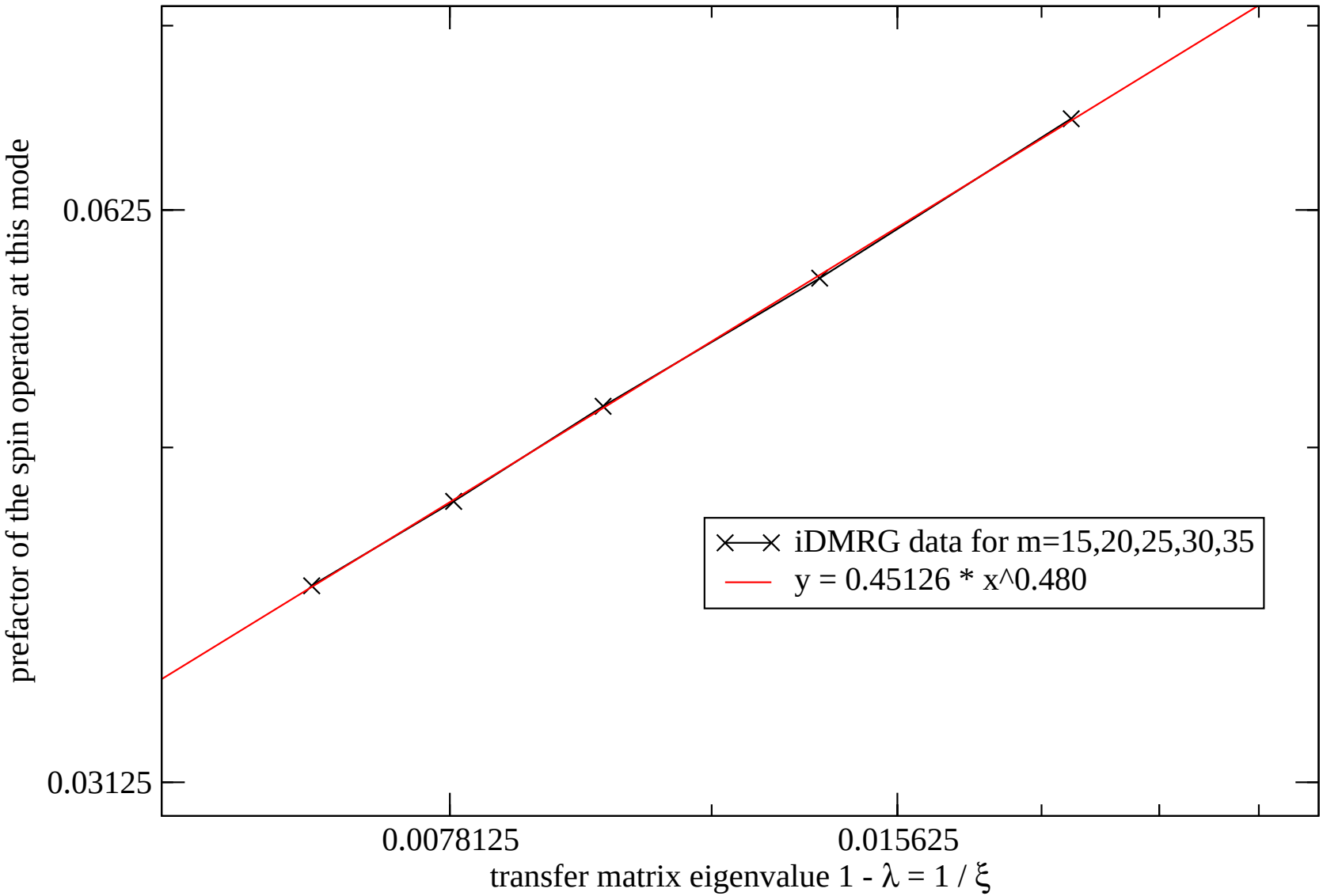
As we increase the number of states kept m the correlation length increases, so the region of validity of the power law increases.

- Take two different calculations with m_1 and m_2
- Correlation lengths ξ_1 and ξ_2
- We expect: $\dfrac{O(\xi_2)}{O(\xi_1)} = \left(\dfrac{\xi_2}{\xi_1}\right)^{\Delta}$
- for x large, we have: $O(x) \simeq a\,\lambda^x$ (with $\xi = -1/\ln\lambda$)
- Prefactor a is overlap of operator O with next-leading eigenvector of transfer operator

$$a \propto \xi^{-\Delta}$$

This gives directly the operator scaling dimensions by direct fit

Heisenberg model fit for the scaling dimension



Generalized Scaling

Alternative viewpoint (Vid Stojevic et al, Phys. Rev. B **91**, 035120 (2015))

Scaling relation for large s:

$$O(s\xi) \simeq a\lambda^{s\xi} = a\lambda^{-s/\ln\lambda} = ae^{-s}$$

So we obtain Δ by scaling $O(s\xi)$ versus $s\xi$

$$O(s\xi) \propto (s\xi)^{-\Delta}$$

However, this also works for s small, eg $s \ll 1$, because for $s \ll 1$, the correlation function is already (approximately) power-law.

the scaling relation works for any $0 < s < \infty$!

But: $O(s\xi) \simeq e^{-s}\xi^{-\Delta}$ only for $s \gg 1$

Generalized Scaling

Alternative viewpoint (Vid Stojevic et al, Phys. Rev. B **91**, 035120 (2015))

Scaling relation for large s:

$$O(s\xi) \simeq a\lambda^{s\xi} = a\lambda^{-s/\ln\lambda} = ae^{-s}$$

So we obtain Δ by scaling $O(s\xi)$ versus $s\xi$

$$O(s\xi) \propto (s\xi)^{-\Delta}$$

However, this also works for s small, eg $s \ll 1$, because for $s \ll 1$, the correlation function is already (approximately) power-law.

the scaling relation works for any $0 < s < \infty$!

But: $O(s\xi) \simeq e^{-s}\xi^{-\Delta}$ only for $s \gg 1$

Generalized Scaling

Alternative viewpoint (Vid Stojevic et al, Phys. Rev. B **91**, 035120 (2015))

Scaling relation for large s:

$$O(s\xi) \simeq a\lambda^{s\xi} = a\lambda^{-s/\ln\lambda} = ae^{-s}$$

So we obtain Δ by scaling $O(s\xi)$ versus $s\xi$

$$O(s\xi) \propto (s\xi)^{-\Delta}$$

However, this also works for s small, eg $s \ll 1$, because for $s \ll 1$, the correlation function is already (approximately) power-law.

the scaling relation works for any $0 < s < \infty$!

But: $O(s\xi) \simeq e^{-s}\xi^{-\Delta}$ only for $s \gg 1$

Moments and cumulants

How close is a variational state to an eigenstate of the Hamiltonian?

- Sometimes there is an algorithmic measure, often not.

The square of the Hamiltonian operator determines the energy variance

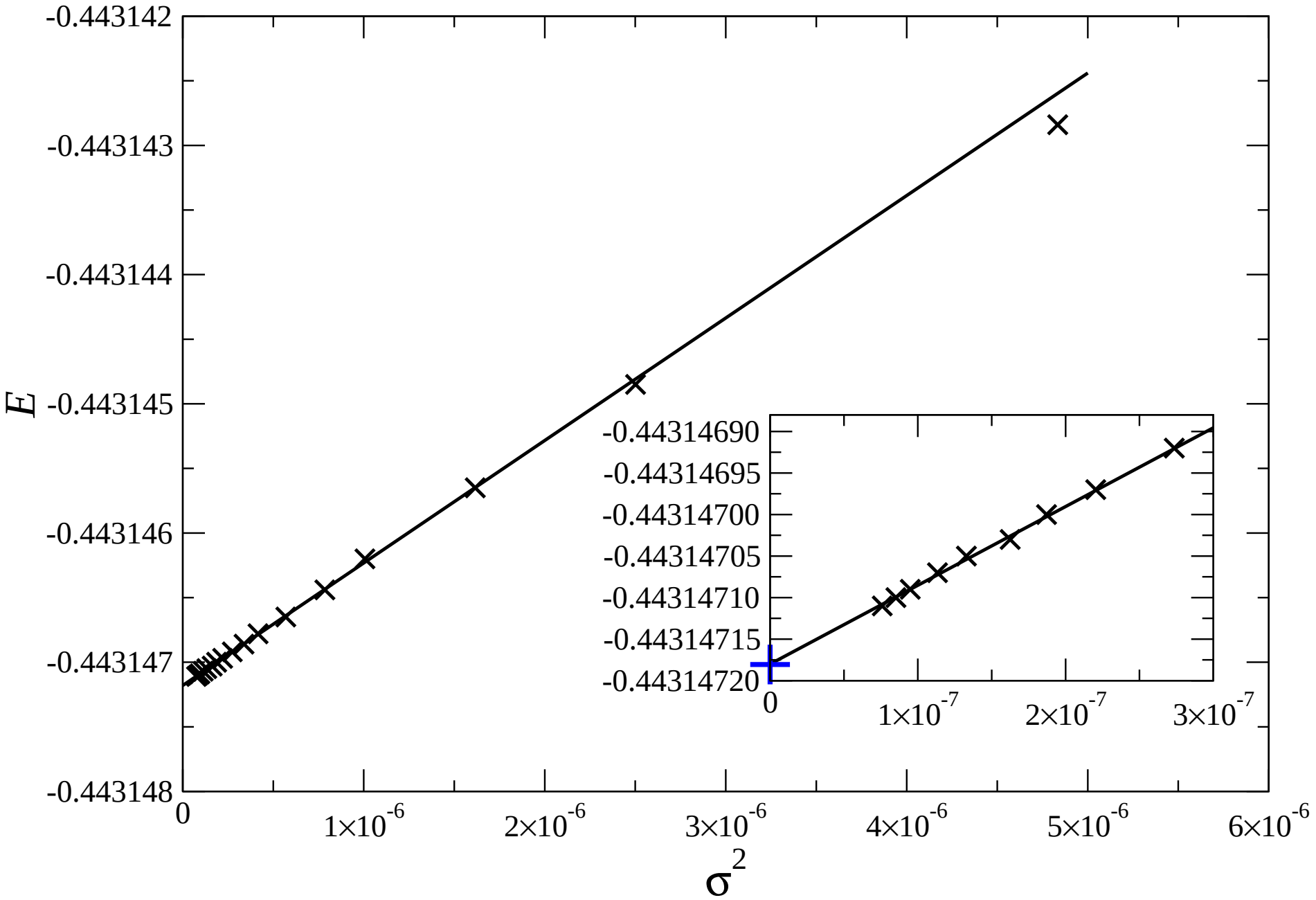
$$\langle H^2 \rangle_L - \langle H \rangle_L^2 = \langle (H - E)^2 \rangle_L = L\sigma^2$$

- A universal measure for the quality of a variational wavefunction
- lower bound for the energy: there is always an eigenstate within σ of E

We can easily construct an MPO representation of H^2

Spin 1/2 Heisenberg Model

Energy per site scaling with variance (exact energy = -ln 2 + 0.25 = -0.44314718056)



Momentum distribution

Momentum-dependent operators have a simple form,

$$b_k^\dagger = \sum_x e^{ikx} b_x^\dagger$$

$$W_{b_k^\dagger} = \begin{pmatrix} e^{ik}I & b^\dagger \\ & I \end{pmatrix}$$

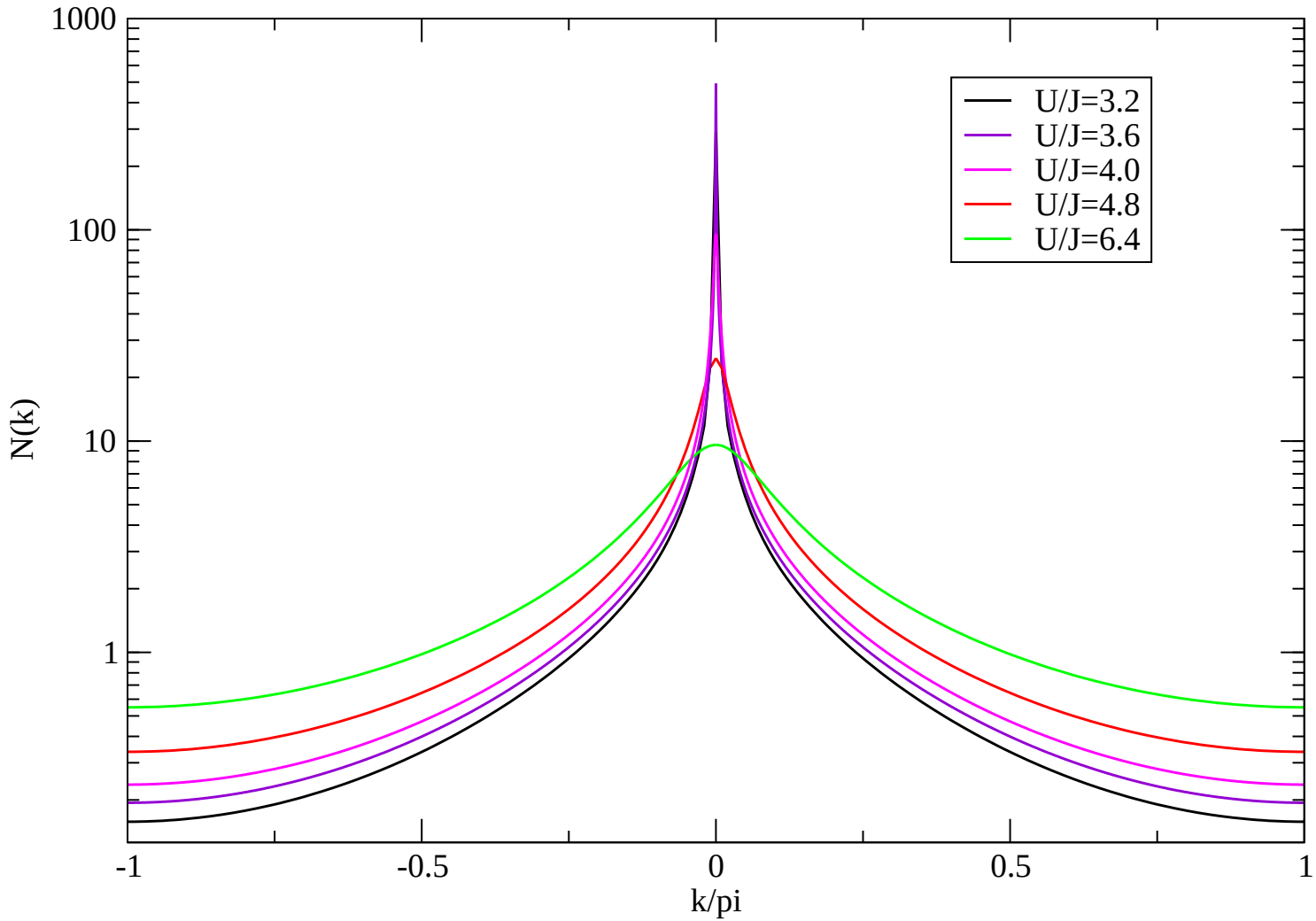
Momentum occupation:

$$n(k) = \frac{1}{L} b_k^\dagger b_k$$

- Broken $U(1)$ symmetry: $\langle b^\dagger \rangle \neq 0$ hence $n(k=0) \propto L$ (extensive)
- With $U(1)$ symmetry: $n(k=0)$ is finite, but diverges with m in superfluid phase

Bose-Hubbard Model N(k)

Infinite 1D, one particle per site



Higher moments

It is straight forward to evaluate a local order parameter, eg

$$M = \sum_i M_i$$

The first moment of this operator gives the order parameter,

$$\langle M \rangle = m_1(L)$$

It is also useful to calculate higher moments, eg

$$\langle M^2 \rangle = m_2(L)$$

or generally

$$\langle M^k \rangle = m_k(L)$$

These are polynomial functions in the system size L.

Evaluating higher moments

The MPToolkit tool is `mp-imoments`

- Can calculate expectation value of almost any triangular MPO
- Finite momentum, kink operators, arbitrary powers of operators,...
- Efficient implementation: $O(Nm^3[M + k\xi])$, MPO dimension M, unit cell size N
- limited by computation time, dimension of the MPO
- Numerically quite stable – condition number of linear solver is ξ

```
#mp-imoments psi lattice:H_J1 --power 4
#Date: Mon, 21 Sep 2015 01:25:48 +0200
#quantities are calculated per unit cell size of 1 site
#moment #degree #real                  #imag                  #magnitude             #argument(deg)
1       1       -1.4014840387142       5.5081024986921e-18    1.4014840387142        180
2       1       1.4998562322677e-09    1.5053693933704e-17    1.4998562322677e-09    5.7506386940758e-07
2       2       1.9641575107706        -6.7296812726876e-18   1.9641575107706        -1.9630927371092e-16
3       1       1.0581567136403e-08    9.3552789252297e-18    1.0581567136403e-08    5.0655823629312e-08
3       2       -6.3060720603692e-09   -1.730862224427e-17    6.3060720603692e-09    -179.99999984274
3       3       -2.7527354008656       -2.6200958995394e-18   2.7527354008656        -180
4       1       8.3035341269522e-08    -7.9223396910221e-16   8.3035341269522e-08    -5.4665473908416e-07
4       2       -5.9319642531364e-08   4.7062972194418e-16    5.9319642531364e-08    179.99999954543
4       3       1.7675737022197e-08    -1.9692692448952e-17   1.7675737022197e-08    -6.3833726602583e-08
4       4       3.8579147271166        -1.4175824902037e-17   3.8579147271166        -2.1053211267068e-16
```

Cumulant expansions

Express the *moments* m_i in terms of the *cumulants per site* κ_j,

$$\begin{aligned} m_1(L) &= \kappa_1 L \\ m_2(L) &= \kappa_1^2 L^2 + \kappa_2 L \\ m_3(L) &= \kappa_1^3 L^3 + 3\kappa_1\kappa_2 L^2 + \kappa_3 L \\ m_4(L) &= \kappa_1^4 L^4 + 6\kappa_1^2\kappa_2 L^3 + (3\kappa_2^2 + 4\kappa_1\kappa_3)L^2 + \kappa_4 L \end{aligned}$$

κ_1 is the order parameter itself
κ_2 is the variance (related to the susceptibility)
κ_3 is the skewness
κ_4 is the kurtosis

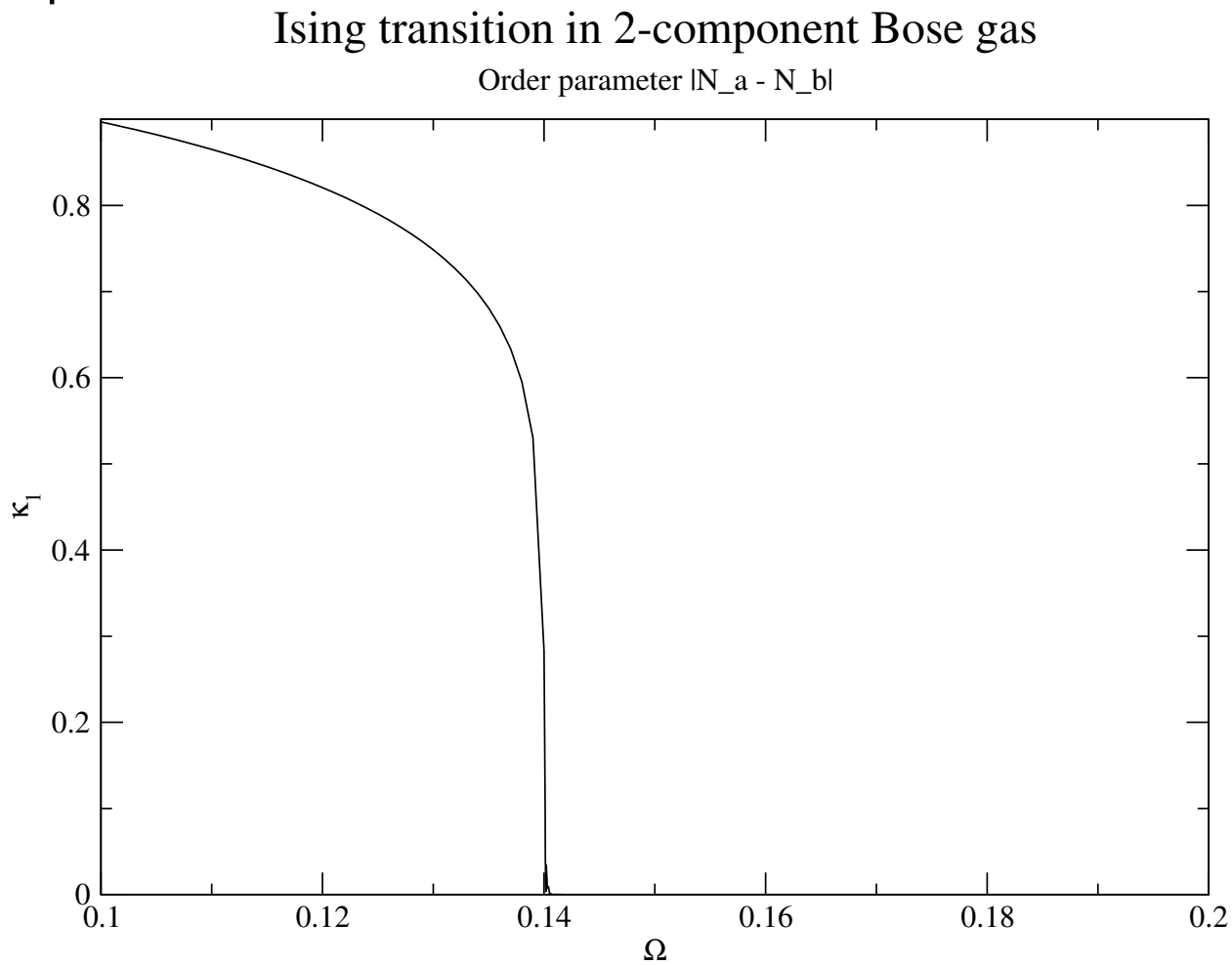
The cumulants per site κ_k are well-defined for an iMPS

```
#mp-imoments psi lat:H_J1 --power 4 --cumulants
#Date: Mon, 21 Sep 2015 01:30:50 +0200
#quantities are calculated per unit cell size of 1 site
#cumulant #real                  #imag                  #magnitude             #argument(deg)
1         -1.4014840387142       5.5081024986921e-18    1.4014840387142        180
2         1.4998562322677e-09    1.5053693933704e-17    1.4998562322677e-09    5.7506386940758e-07
3         1.0581567136403e-08    9.3552789252297e-18    1.0581567136403e-08    5.0655823629312e-08
4         8.3035341269522e-08    -7.9223396910221e-16   8.3035341269522e-08    -5.4665473908416e-07
```

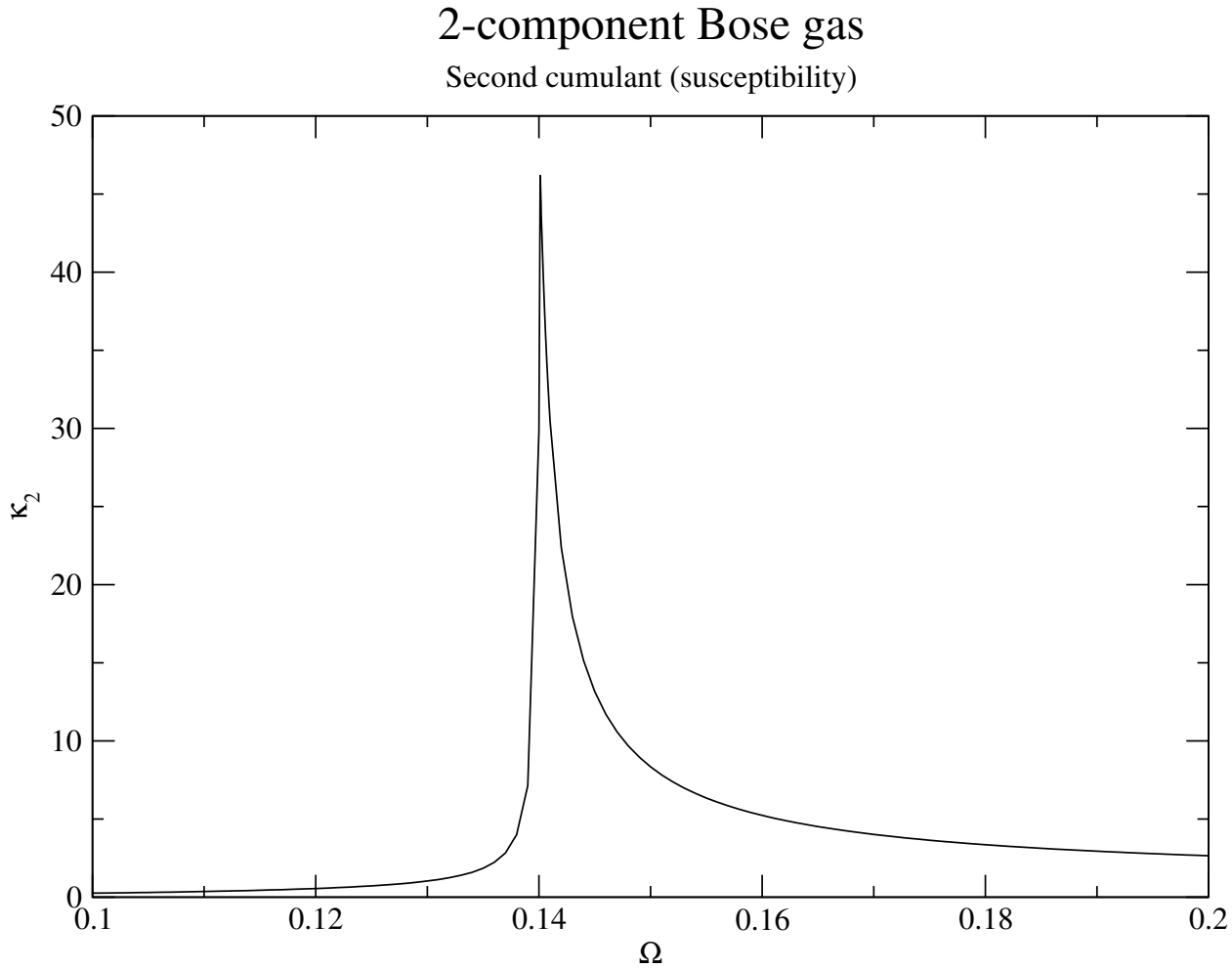
iMPS for two-component bose gas

The cumulant expansion already gives a lot of information

κ_1 is the order parameter itself

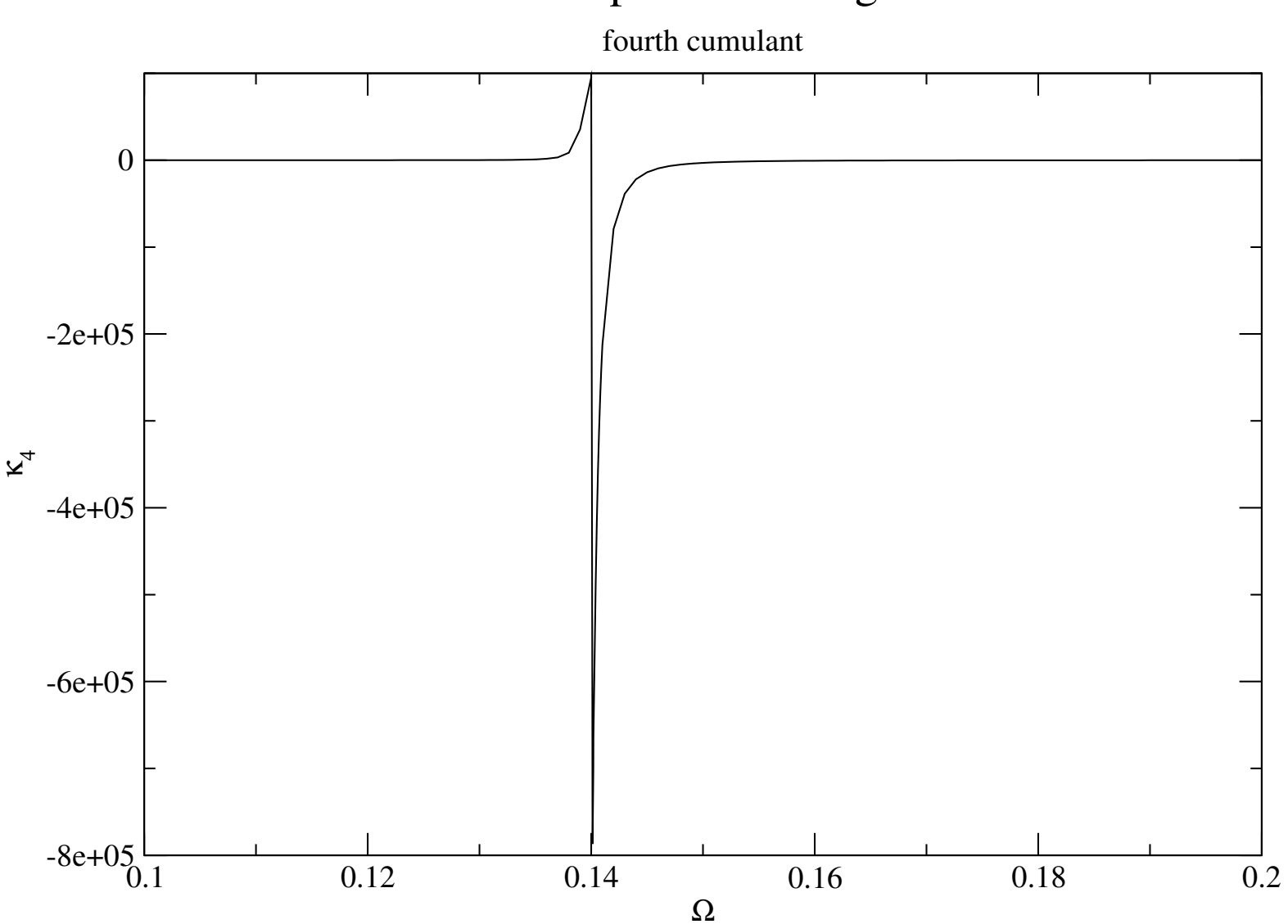


The second cumulant gives the susceptibility



Different to a finite-size scaling, the susceptibility exactly diverges at the critical point.
Sufficiently close to the critical point, it looks mean-field-like (so will generally give the *wrong exponent!*)

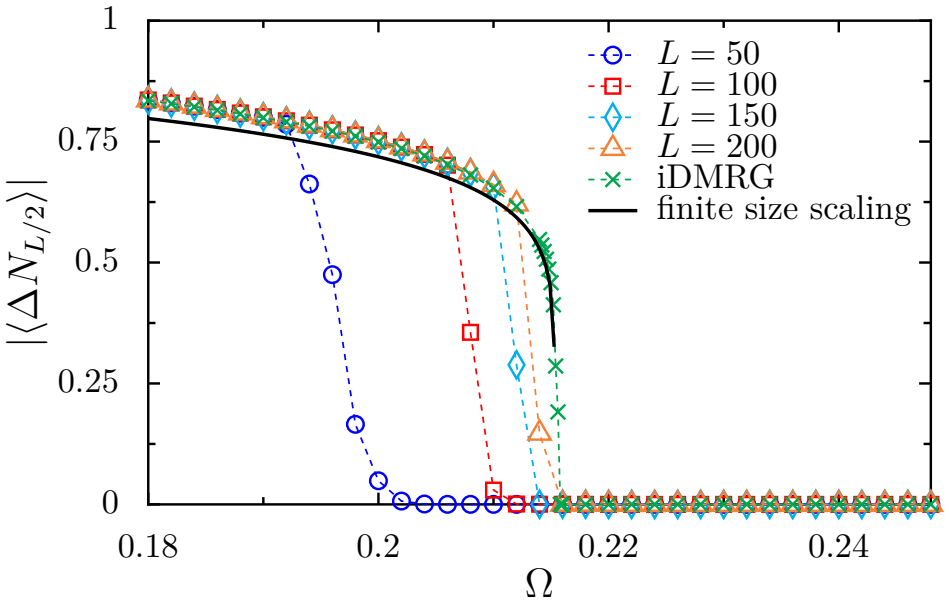
The fourth cumulant changes sign at the transition.

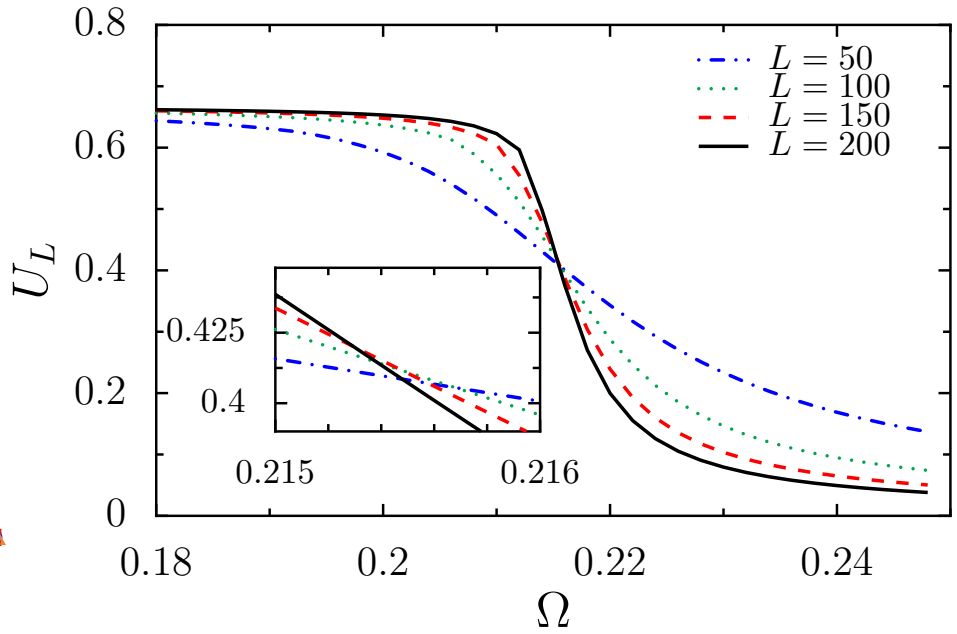


Binder cumulant

For *finite* systems, the Binder cumulant of the order parameter cancels the leading-order finite size effects

$$U_L = 1 - \frac{\langle m^4 \rangle}{3\langle m^2 \rangle^2}$$





The 2-component Bose-Hubbard model, with a linear coupling between components, has an Ising-like transition from immiscible (small Ω) to miscible (large Ω).

Binder Cumulant for iMPS

Naively taking the limit $L \to \infty$ for the Binder cumulant doesn't produce anything useful:

- if the order parameter $\kappa_1 \neq 0$,

$$U_L = 1 - \frac{\langle m^4 \rangle_L}{3\langle m^2 \rangle_L^2} \to \frac{2}{3}$$

- if $\kappa_1 = 0$, then $m_4(L) = 3k_2^2L^2 + k_4L$
  Hence

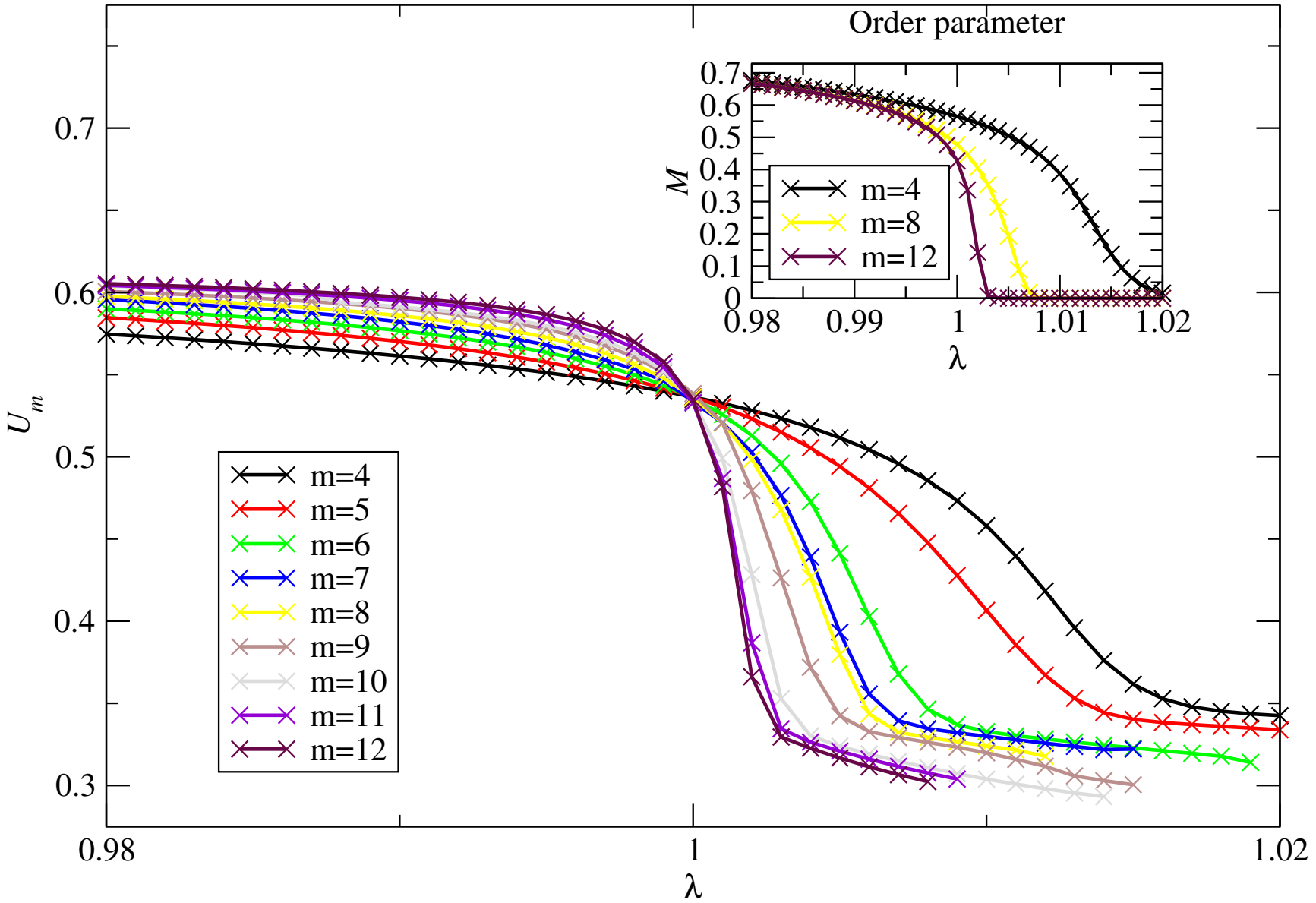
$$U_L = 1 - \frac{3k_2^2L^2 + k_4L}{3k_2^2L^2} \to 0$$

- Finally, a step function that detects whether the order parameter is non-zero

Better approach, in the spirit of finite-entanglement scaling: Evaluate the moment polynomial using $L \propto$ correlation length

Transverse field Ising model

Binder cumulant, scale factor s=5



String parameters

Order parameters do not have to be local

Mott insulator string order parameter

$$O_P^2 = \lim_{|j-i|\to\infty} \langle \Pi_{k=i}^{j} (-1)^{n_k} \rangle$$

We can write this as a correlation function of 'kink operators',

$$p_i = \Pi_{k<i} \, (-1)^{n_k}$$

This turns the string order into a 2-point correlation function:

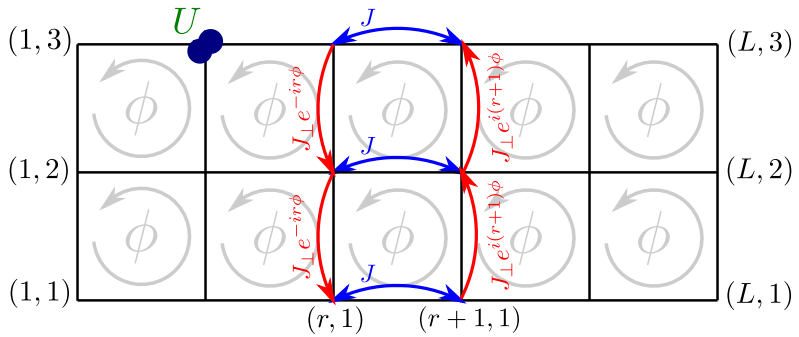
$$O_P^2 = \lim_{|j-i|\to\infty} \langle \, p_i \, p_j \, \rangle$$

Or as an order parameter:

$$P = \sum_i p_i$$

Then $O_p^2 = \frac{1}{L^2} \langle P^2 \rangle$

Real example: 3-leg Bose-Hubbard model with flux phase (F. Kolley, M. Piraud, IPM, U. Schollwoeck, F. Heidrich-Meisner, NJP 17 (2015) 092001)



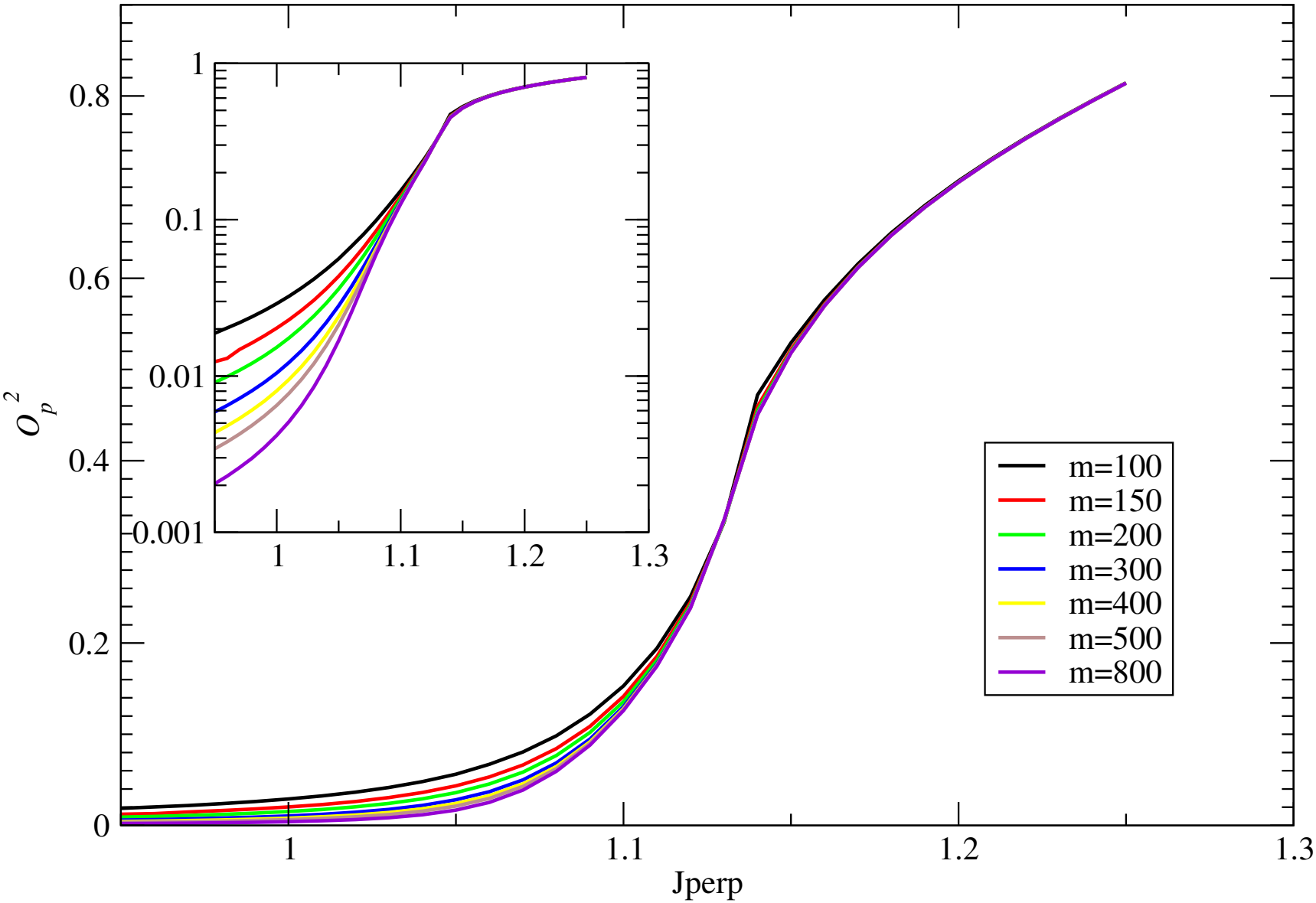
For density $n = 1/3$ (one particle per rung), near flux $\phi \sim \pi$, there is a transition from a Mott to critical as a function of $J_\perp$

- P has a simple MPO representation
- Hence we can calculate higher moments of P.

```
mp-imoments psi lattice:"sum_kink(exp(i*pi*N(0)), N(0))^2"
```

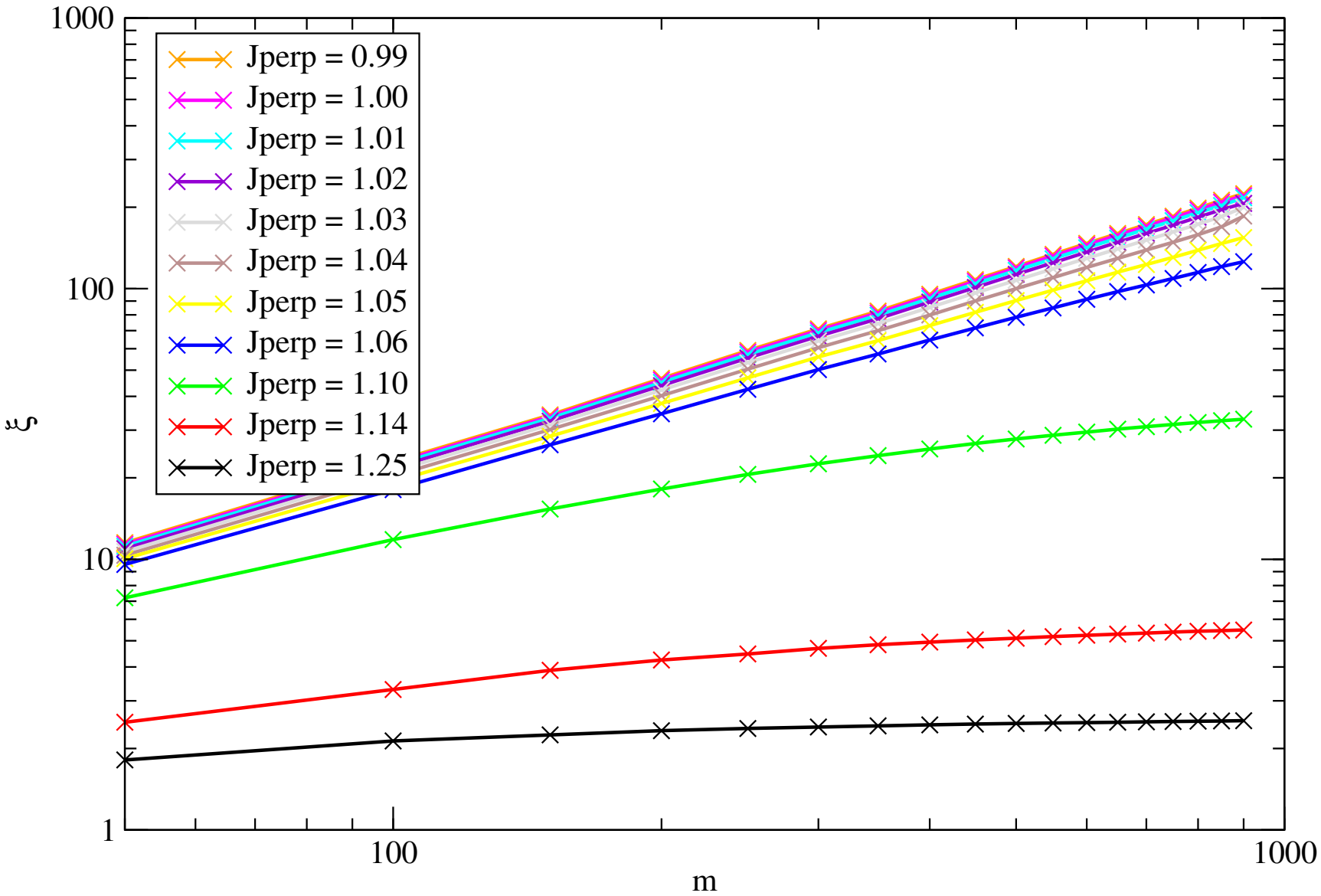
Bose-Hubbard Ladder

String order parameter



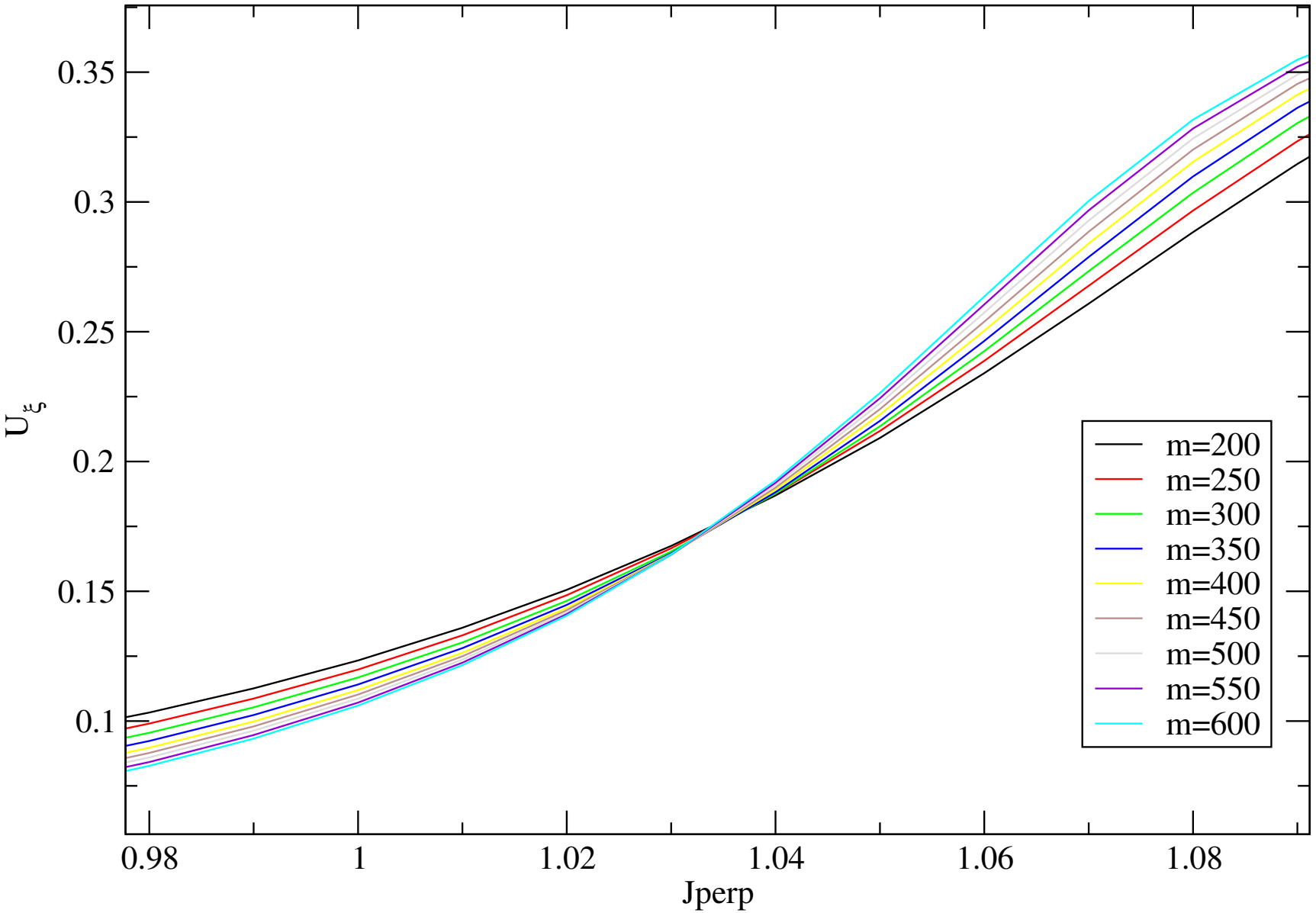
Bose=Hubbard Ladder

Scaling of correlation length



Bose-Hubbard Ladder

String parameter Binder cumulant



Other iMPS tools in the MPToolkit

- `mp-aux-algebra`: determine the algebra of symmetry operations (eg is it a projective representation?), used for *symmetry-protected topological states*
- OR; construct the matrix representation explicitly and examine with eg Matlab with `mp-aux-matrix`
- `mp-wigner-eckart`: project an $SU(2)$ wavefunction down to $U(1)$, or remove symmetries completely.
- Spatial reflection, conjugation, ...
- Improved `mp-info`

```
$ mp-info psi
Wavefunction is an InfiniteWavefunction in the left canonical basis.
Symmetry list = N:U(1)
Unit cell size = 6
Quantum number per unit cell = 2
Number of states = 2974

Attributes:
Hamiltonian=lat:H{alpha=0.9,K=0.5}
LastEnergy=-3.3848080922103367

Last history entry:
Date: Sun, 20 Sep 2015 21:38:18 +0200
./mp-idmrg-s3e -H "lat:H{alpha=0.9,K=0.5}" -w psi -m 2000..3000x40 -f 1e-4 --evolve 0.003
```

Future developments

- Merger of finite and infinite DMRG: *Infinite Boundary Conditions (IBC)*
  - Finite system attached to semi-infinite 'leads' at one or both ends
  - Ideal for local purturbations to a translationally invariant intial state
  - Should also be useful for excitation gaps
- Time evolution – no algorithms in the toolkit yet (except poor man's iTEBD: `mp-iapply` with bond evolution operators)
- More MPO operations: the derivative of an MPO, eg evaluate

$$\frac{\mathrm{d}}{\mathrm{d}k}\, n(k)$$

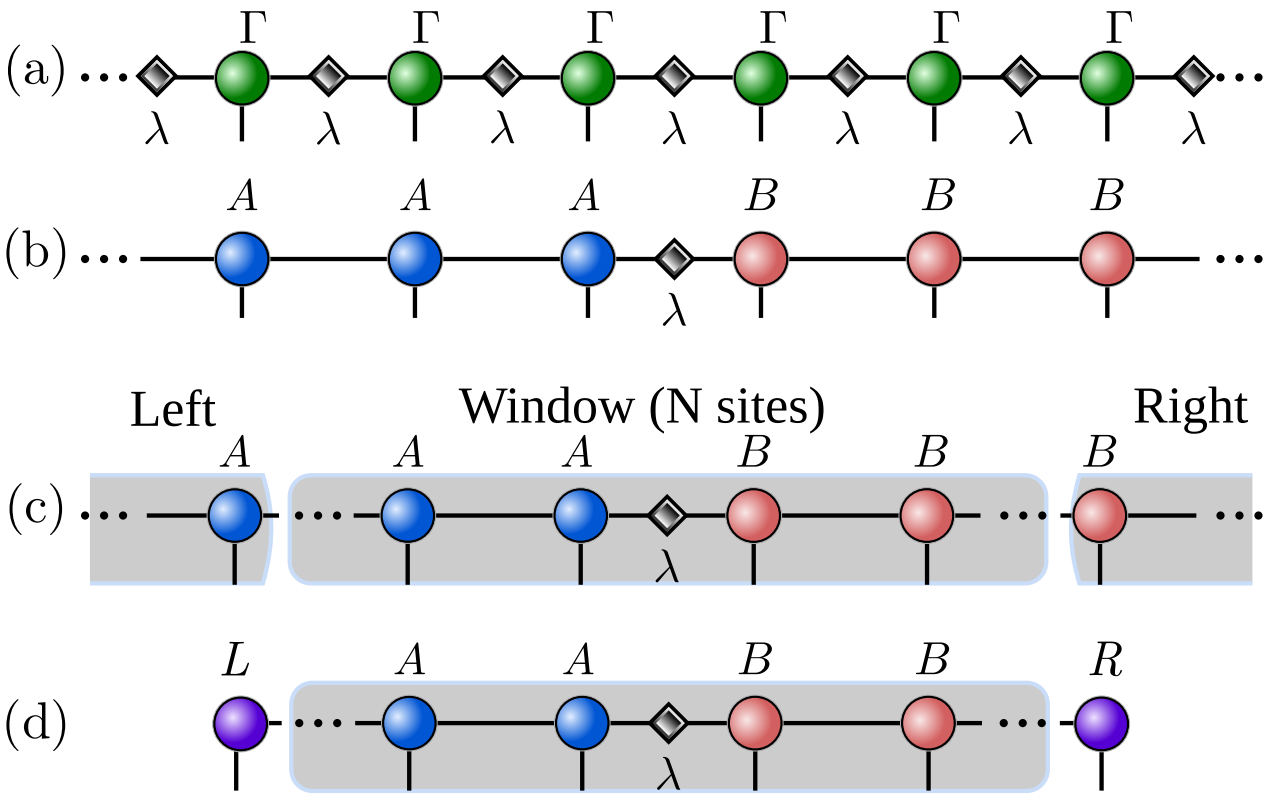
  at the Fermi surface
- iMPS perturbation theory: related to moment expansions and derivatives
- Translationally invariant excitations (Hagemann, Verstraete etc proof of principle)
- Finite temperature – reuse a lattice file for finite temperature calculations
- coarse-graining transformations, changing the local basis (eg maximum number of bosons) – functionality already exists in C++ but no easy way to access it
- Better handling of real vs complex wavefunctions
- Performance improvements and parallelization

Infinite boundary conditions

H.N. Phien, G. Vidal, IPM, Phys. Rev. B 86, 245107 (2012), Phys. Rev. B 88, 035103 (2013)

(see also Zauner et al 1207.0862, Milsted et al Phys. Rev. B 155116 (2013))

Local perturbation to a translationally invariant state



Map infinite system onto a finite MPS, with an effective boundary

- Key point: Even if the perturbation is correlated at long range, only the tensors at the perturbation are modified
- Decompose the Hamiltonian

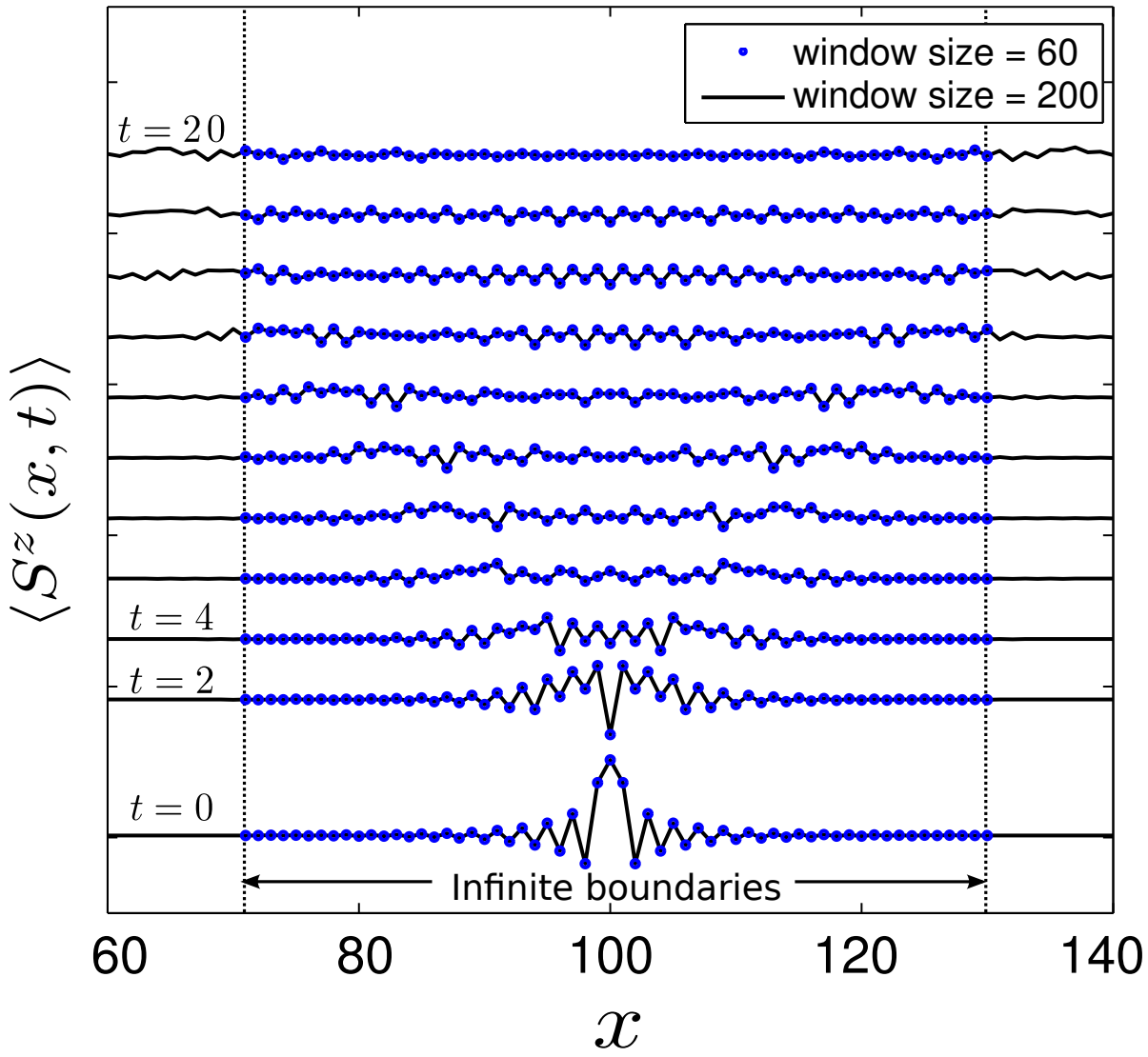
$$H = H_L + H_{LW} + H_W + H_{WR} + H_R$$

- We can calculate H_L and H_R by summing the infinite series of terms from the left and right
- Away from the perturbation the wavefunction is approximately an eigenstate, so

$$\exp itH_L \sim I$$

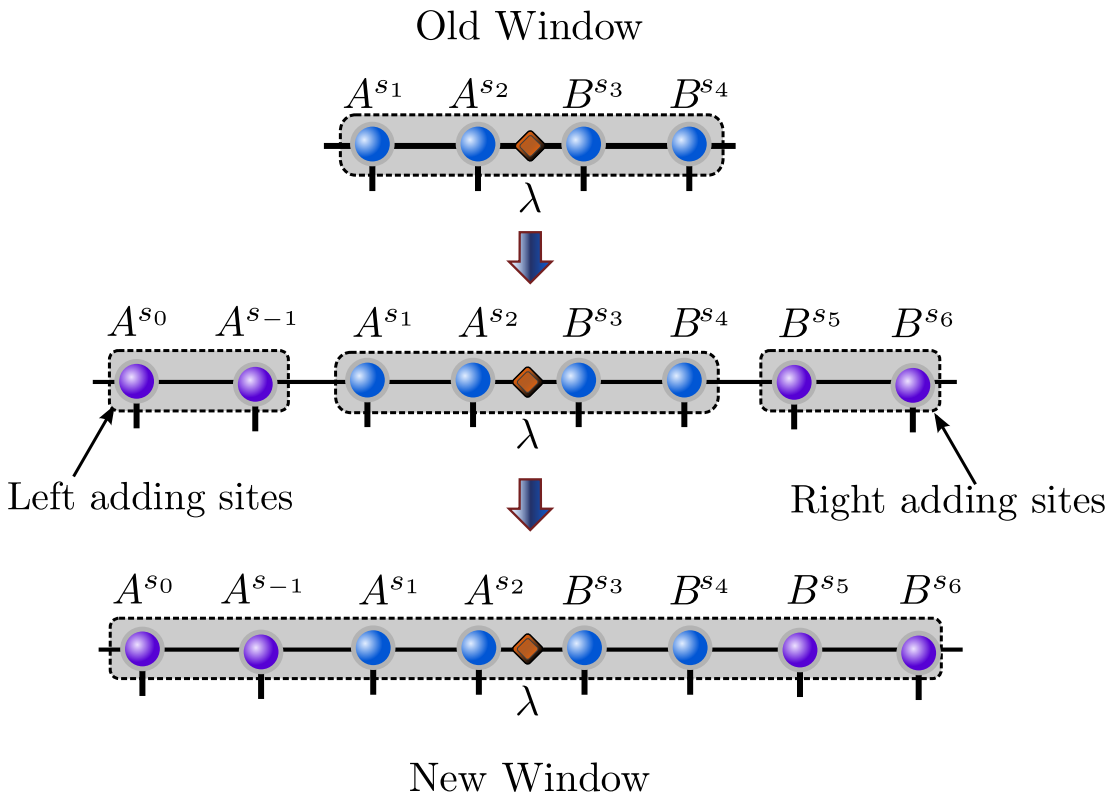
and we don't leave the Hilbert space of the semi-infinite strip

Spin-1 Heisenberg chain, S^+ initial perturbation

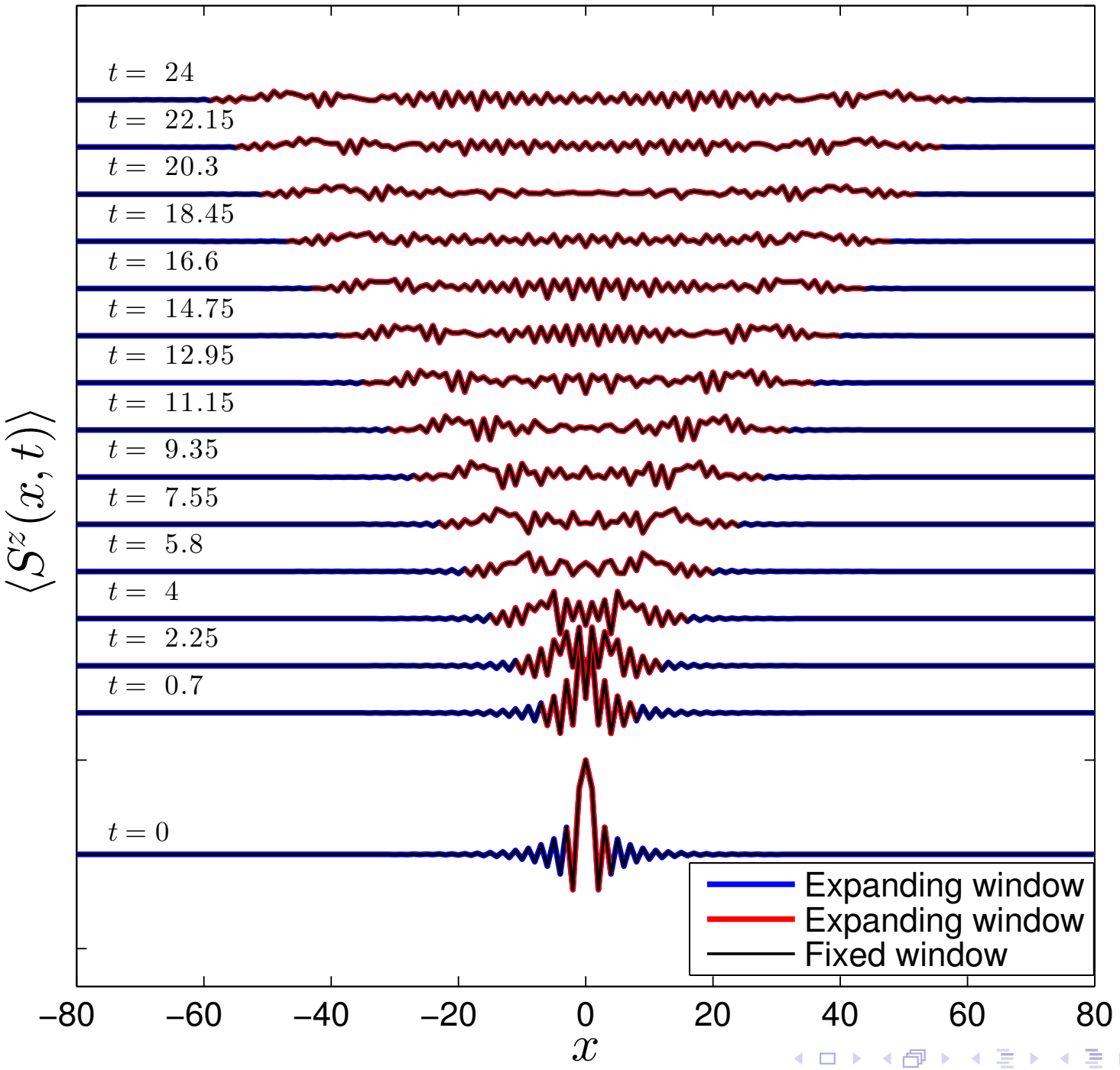


Resize the window

We can do better - why keep the size of the window fixed?
Window expansion - incorporate sites from the translationally-invariant section into the window

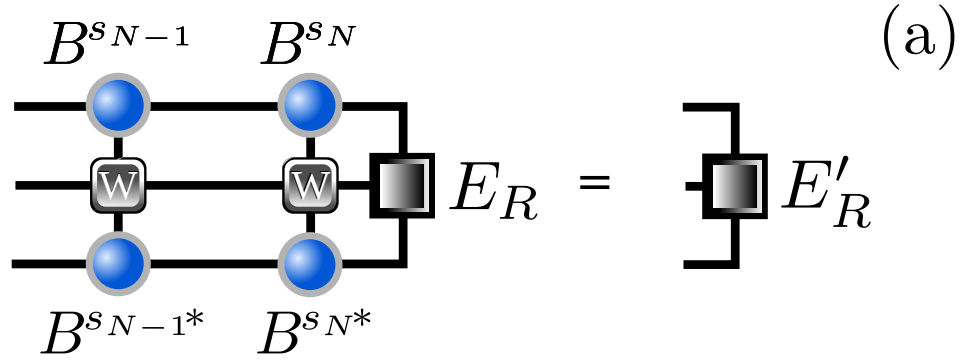


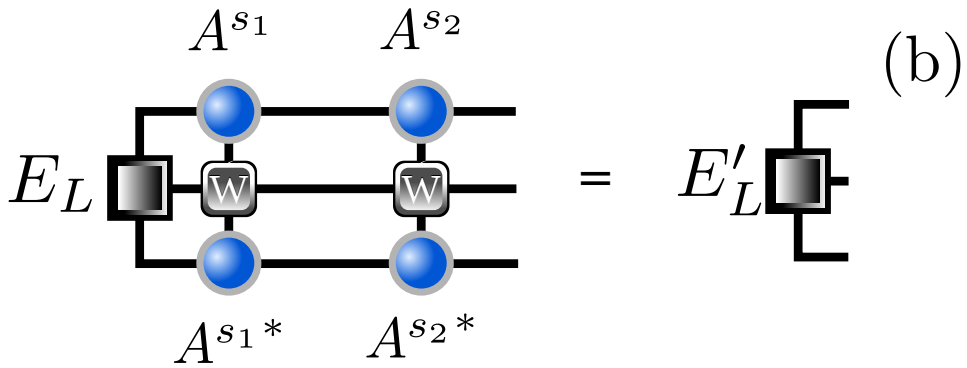
Criteria for expanding: is the wavefront near the boundary?
(Calculate from the fidelity of the wavefunction at the boundary)


$t = 24$
$t = 22.15$
$t = 20.3$
$t = 18.45$
$t = 16.6$
$t = 14.75$
$t = 12.95$
$t = 11.15$
$t = 9.35$
$t = 7.55$
$t = 5.8$
$t = 4$
$t = 2.25$
$t = 0.7$
$t = 0$
$\langle S^z(x,t)\rangle$
Expanding window
Expanding window
Fixed window
−80
−60
−40
−20
0
20
40
60
80
x

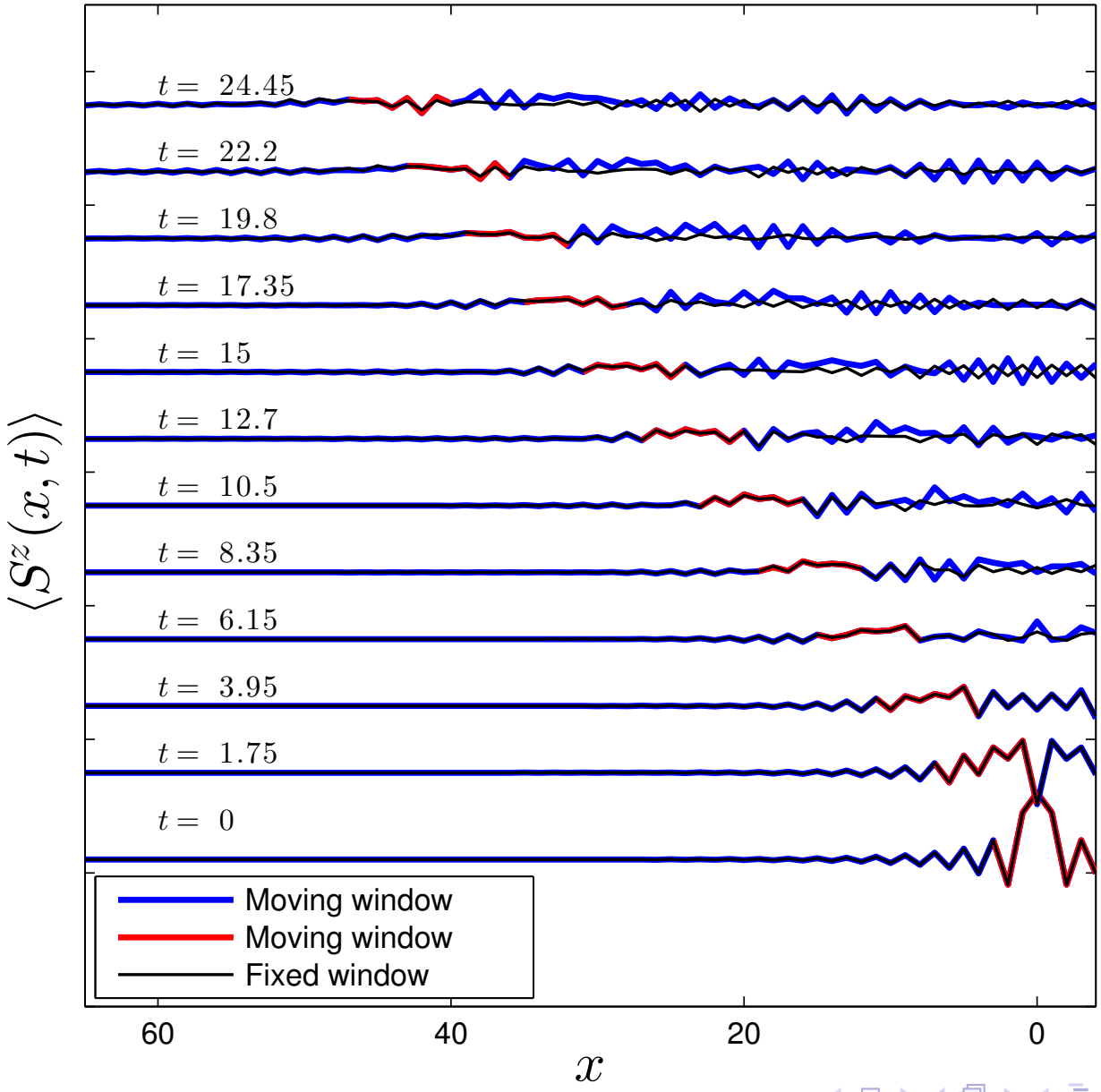
Window contraction

Window contraction - incorporate tensors from the window into the boundary
Contract the MPS and Hamiltonian MPO





Follow the wavefront



Broken symmetries

Finite size MPS: No broken symmetries (to $O(\text{truncation error})$)
Infinite size MPS: The Ansatz can break all symmetries

even continuous symmetries in one dimension

How to understand this?

- Matrix elements connecting symmetry sectors vanish as $\sim \exp(-L) \quad \to 0$
- Continuous symmetries cannot break in *exact* 1D because the associated goldstone modes would destroy the order parameter completely (percolation threshold!)
- But if the goldstone modes are gapped due to finite basis size, the symmetry can break
- Alternatively: in order to get a finite correlation length we must perturb the Hamiltonian with a *relevant* perturbation. No reason why that perturbation should not break any (or all) symmetry.

Broken symmetries

Finite size MPS: No broken symmetries (to $O(\text{truncation error})$)
Infinite size MPS: The Ansatz can break all symmetries

even continuous symmetries in one dimension

How to understand this?

- Matrix elements connecting symmetry sectors vanish as $\sim \exp(-L) \quad \to 0$
- Continuous symmetries cannot break in *exact* 1D because the associated goldstone modes would destroy the order parameter completely (percolation threshold!)
- But if the goldstone modes are gapped due to finite basis size, the symmetry can break
- Alternatively: in order to get a finite correlation length we must perturb the Hamiltonian with a *relevant* perturbation. No reason why that perturbation should not break any (or all) symmetry.

Prototypical example: Bose-Hubbard model

$$H = \frac{U}{2}\sum_i N_i(N_i-1) - J\sum_{<i,j>} b_i^\dagger b_j + b_j^\dagger b_i - \mu N$$

Mean-field approximation: $\beta = \langle b \rangle$

$$H_{\text{MF}} = \frac{U}{2}\sum_i N_i(N_i-1) - J\beta\sum_i (b_i^\dagger + b_i) - \mu N$$

Mean field Hamiltonian breaks $U(1)$ particle number conservation
Groundstate is an $m = 1$ infinite MPS (product state!)

$$|\psi\rangle = (|0\rangle + a_1|1\rangle + a_2|2\rangle \ldots)^{\otimes L}$$

- An iMPS with no symmetries reduces to mean-field
- Imposing quantum number symmetries reduces the quality of the variational state (for fixed m)
- But usually worth the cost in computational efficiency

Prototypical example: Bose-Hubbard model

$$H = \frac{U}{2}\sum_i N_i(N_i-1) - J\sum_{<i,j>} b_i^\dagger b_j + b_j^\dagger b_i - \mu N$$

Mean-field approximation: $\beta = \langle b \rangle$

$$H_{\text{MF}} = \frac{U}{2}\sum_i N_i(N_i-1) - J\beta\sum_i (b_i^\dagger + b_i) - \mu N$$

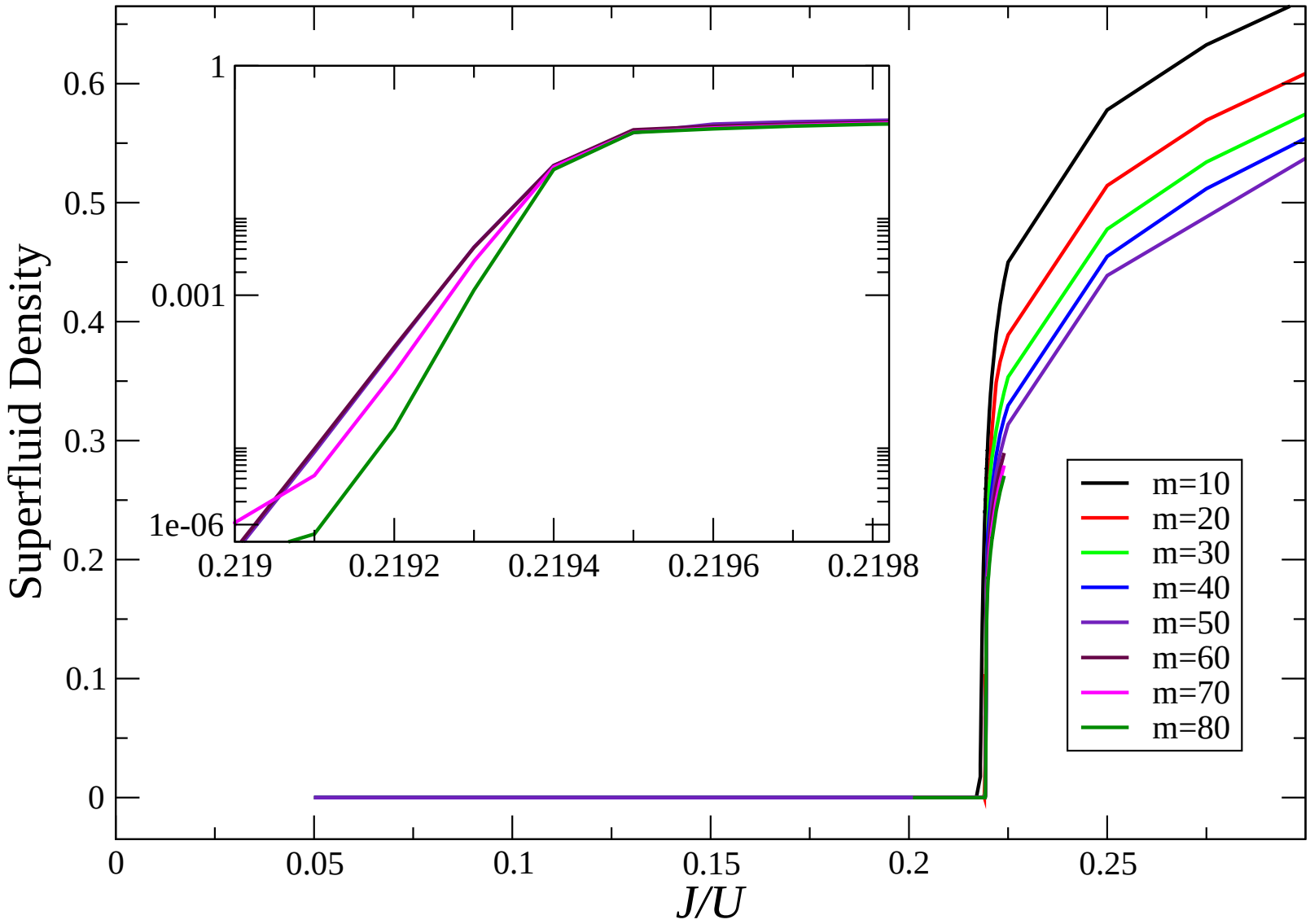
Mean field Hamiltonian breaks $U(1)$ particle number conservation
Groundstate is an $m = 1$ infinite MPS (product state!)

$$|\psi\rangle = (|0\rangle + a_1|1\rangle + a_2|2\rangle \ldots)^{\otimes L}$$

- An iMPS with no symmetries reduces to mean-field
- Imposing quantum number symmetries reduces the quality of the variational state (for fixed m)
- But usually worth the cost in computational efficiency

Bose-Hubbard Model Mott-Superfluid Transition

μ=0.25

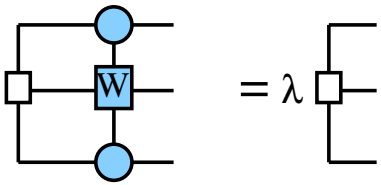


Expectation values of MPO's - arxiv:0804.2509

We have seen that we can write many interesting operators in the form of a matrix product operator

- Can we evaluate the expectation value of an arbitrary MPO?

If the MPO has no Jordan structure, this is a simple eigenvalue problem



For a lower triangular MPO, this doesn't work.

- But we can make use of the triangular structure

$$\begin{pmatrix} I & & \\ S^z & & \\ \lambda S^x & S^z & I \end{pmatrix}$$

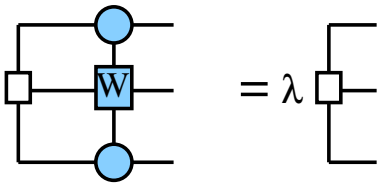
- index by index, each component is a function only of the previously calculated terms

Expectation values of MPO's - arxiv:0804.2509

We have seen that we can write many interesting operators in the form of a matrix product operator

- Can we evaluate the expectation value of an arbitrary MPO?

If the MPO has no Jordan structure, this is a simple eigenvalue problem

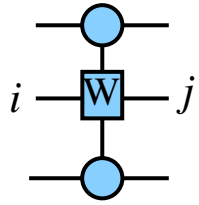


For a lower triangular MPO, this doesn't work.

- But we can make use of the triangular structure

$$\begin{pmatrix} I & & \\ S^z & & \\ \lambda S^x & S^z & I \end{pmatrix}$$

- index by index, each component is a function only of the previously calculated terms



Choose bond indices i, j of W_{ij}, and denote $T_{W_{ij}}$
Example:

$$\begin{pmatrix} I & & \\ S^z & & \\ \lambda S^x & S^z & I \end{pmatrix}$$

Eigentensor is $\begin{pmatrix} E_1 & E_2 & E_3 \end{pmatrix}$

- Starting from E_3:

  $$E_3 = T_I(E_3) = I$$

  is equivalent to the orthogonality condition - E_3 is just the identity
- E_2:

  $$E_2 = T_{S^z}(E_3) = T_{S^z}(I) = S^z$$

- E_1: doesn't reach a fixed point, $E_1 = E_1(L)$ depends on the number of iterations L

  $$\begin{aligned} E_1(L+1) &= T_I(E_1(L)) + T_{S^z}(E_2) + T_{\lambda S^x}(E_3) \\ &= T_I(E_1(L)) + C \end{aligned}$$

  where C is a constant matrix, $C = T_{S^z}(S^z) + \lambda S^x$

Fixed point equations for E_1:

$$E_1(L+1) = T_I(E_1(L)) + C$$

Eigenmatrix expansion of T_I:

$$T_I = \sum_{n=1}^{m^2} \lambda_n |\lambda\rangle\langle\lambda|$$

giving

$$E_1^{(n)}(L+1) = \lambda_n E_1^{(n)}(L) + C^{(n)}$$

- Since $\lambda_1 = 1$ by construction, this motivates decomposing into components parallel and perpendicular to the identity:

  $$E_1(L) = E_1'(L) + e_1(L)\, I$$

  where $\mathrm{Tr}\, E_1'(L)\rho = 0$

- Component in the direction of the identity:

$$e_1(L+1) = e_1(L) + \mathrm{Tr}\, C\rho$$

Has the solution

$$e_1(L) = L\ \mathrm{Tr}\, C\rho$$

is the energy

- Component perpendicular to the identity:

$$E_1'(L+1) = T_I(E_1'(L)) + C'$$

where $C' = C - (\mathrm{Tr}\, C\rho)\, I$

$$E_1'(L+1)_n = \lambda_n E_1'(L)_n + C_n'$$

Since all $|\lambda_n| < 1$ here, this is a geometric series that converges to a fixed point (independent of L),

$$(1 - T_I)(E_1') = C'$$

Linear solver for the unknown matrix E_1'

- Component in the direction of the identity:

$$e_1(L+1) = e_1(L) + \mathrm{Tr}\, C\rho$$

Has the solution

$$e_1(L) = L\ \mathrm{Tr}\, C\rho$$

is the energy

- Component perpendicular to the identity:

$$E_1'(L+1) = T_I(E_1'(L)) + C'$$

where $C' = C - (\mathrm{Tr}\, C\rho)\, I$

$$E_1'(L+1)_n = \lambda_n E_1'(L)_n + C_n'$$

Since all $|\lambda_n| < 1$ here, this is a geometric series that converges to a fixed point (independent of L),

$$(1 - T_I)(E_1') = C'$$

Linear solver for the unknown matrix E_1'

Summary:

- Decompose eigentensor into components parallel and perpendicular to the identity
- The component parallel to the identity is the energy per site
- The perpendicular components reach a fixed point and give the Hamiltonian matrix elements

$$
\begin{array}{lcll}
E_3 & = & I & \text{Identity operator} \\
E_2 & = & Sz & \text{Sz block operator} \\
E_1(L) & = & E' + Le_1 & \text{Hamiltonian operator + energy per site}
\end{array}
$$

Generalization to arbitrary triangular MPO's

arXiv:1008.4667

At the i^{th} iteration, we have

$$E_i(L+1) = T_{W_{ii}}(E_i(L)) + \underbrace{\sum_{j>i} T_{W_{ji}}(E_j(L))}_{= C(L)}$$

Basic idea:

- if $W_{ii} = 0$, then $E_i = C$
- if $W_{ii} \neq 0$, then solve $(1 - T_{W_{ii}})(E_i) = C$

The result will be a polynomial function of L

- solve separately for the coefficient of the k-th power of n
- If the diagonal element is unitary, then obtain the eigenvalues of magnitude 1
- If any eigenvalues are complex, then expand also in fourier modes

Generalization to arbitrary triangular MPO's

arXiv:1008.4667

At the i^{th} iteration, we have

$$E_i(L+1) = T_{W_{ii}}(E_i(L)) + \underbrace{\sum_{j>i} T_{W_{ji}}(E_j(L))}_{= C(L)}$$

Basic idea:

- if $W_{ii} = 0$, then $E_i = C$
- if $W_{ii} \neq 0$, then solve $(1 - T_{W_{ii}})(E_i) = C$

The result will be a polynomial function of L

- solve separately for the coefficient of the k-th power of n
- If the diagonal element is unitary, then obtain the eigenvalues of magnitude 1
- If any eigenvalues are complex, then expand also in fourier modes