

Question

第四章作业-itebd (选做, 26分)

编写程序:

用ITEBD计算自旋为1的一维无限长Heisenberg模型的波函数并估算基态能量。奇异值分解可以调用cpplapack中的函数gesvd(...)或者你熟悉的函数。

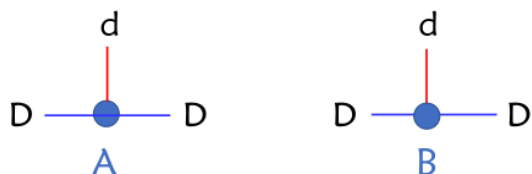
1.

```
BLASINT gesvd(const char &jobu, const char &jobvt, const BLASINT &m, const BLASINT &n, Complex* matrix, const BLASINT &lda, double* s, Complex* u, const BLASINT &ldu, Complex* vt, const BLASINT &ldvt, Complex* work, const BLASINT &lwork, double* rwork, BLASINT &info);
```

matrix = u s vt

- jobu, jobvt 取为 'A' ;
- m, n 为需要分解的矩阵 matrix 的行和列;
- lda, ldu, ldvt可以简单看作相应矩阵的列;
- work, rwork是额外需要的内存, 具体大小参见 lapack手册 zgesvd, dgesvd.

第四章作业 (选做, 26分)



2.

```
Complex* A = new Complex[D*d*D];
Complex* B = new Complex[D*d*D];
```

3. 基态能量的估算

$$e^{-\tau h}|\psi\rangle \approx \alpha|\psi\rangle \approx e^{-\tau E_0}|\psi\rangle.$$

分析

我们先介绍一下“Imaginary Time-evolving Block Decimation” (ITEBD)。

就我理解, 这是一种基于“power method”的一种求解矩阵最小本征值 (也就是基态) 的一种方法。通过“虚时演化+归一化”的方式将系统的状态还原到基态。

下面我们将具体来讲述该算法的基本原理:

1. 波函数的含时演化

在薛定谔绘景下, t 系统时刻的状态波函数可以表示为 (初始时刻记为时间零点):

$$|\Psi_t\rangle = \exp(-iHt) |\Psi_0\rangle$$

虚时情况下：

$$|\Psi_\tau\rangle = \frac{\exp(-H\tau) |\Psi_0\rangle}{\|\exp(-H\tau) |\Psi_0\rangle\|}$$

如果了解“power method”，那么应该可以看出虚时情况下的公式就是我们用来求解矩阵最小本征值的式子。

2. Power Method

对于厄米矩阵 $H \in \mathbb{C}^{n \times n}$ ，其最大本征值及相应的本征向量可以由下面的推导给出：

设 H 的本征值为 $\lambda_1, \lambda_2, \dots, \lambda_n$ ，对应的本征矢为 $\nu_1, \nu_2, \dots, \nu_n$ ，且 $|\lambda_1| > |\lambda_2| > \dots > |\lambda_n|$ 。

\mathbb{C}^n 中的向量 \mathbf{x}_0 可以写为 $\mathbf{x}_0 = c_1 \nu_1 + c_2 \nu_2 + \dots + c_n \nu_n$ 。（这是由厄米矩阵的本征空间是完备的决定的）

因此，我们有：

$$\begin{aligned} \mathbf{x}_1 &= H\mathbf{x}_0 = c_1 \lambda_1 \nu_1 + \dots + c_n \lambda_n \nu_n \\ &\dots \\ \mathbf{x}_k &= H^k \mathbf{x}_0 = c_1 \lambda_1^k \nu_1 + \dots + c_n \lambda_n^k \nu_n \\ &= \lambda_1^k \left[c_1 \nu_1 + c_2 \left(\frac{\lambda_2}{\lambda_1} \right)^k \nu_2 + \dots + c_n \left(\frac{\lambda_n}{\lambda_1} \right)^k \nu_n \right] \end{aligned}$$

当 $k \rightarrow \infty$ 时，只会剩下 ν_1 项，但是由于 λ_1^k 会趋向于无穷大或无穷小；因此，每次迭代都需要将 \mathbf{x} 归一化。

上面时一般的“power method”，只能求解绝对值最大的特征值和对应的特征向量。下面我们将介绍求解最小特征值及对应特征向量的方法。

还是上面的矩阵，只不过 $\lambda_1 < \lambda_2 < \dots < \lambda_n$ 。

则：

$$\mathbf{x} = e^{-\beta H} \mathbf{x}_0 = c_1 e^{-\beta \lambda_1} \nu_1 + \dots + c_n e^{-\beta \lambda_n} \nu_n$$

当 $\beta \rightarrow \infty, \mathbf{x} \rightarrow \nu_1$ 。

这就是我们上面给出的虚时演化波函数的式子。实际计算中，一般 $\tau \sim 0.01$ ，也可能和 λ_1 相关。

3. Matrix Product State

基于奇异值分解我们可以将系统的状态用多个矩阵乘积的形式表示^[1]：

SVD guarantees for an arbitrary (rectangular) matrix M of dimensions $(N_A \times N_B)$ the existence of a decomposition:

$$M = USV^\dagger,$$

where

- U is of dimension $(N_A \times \min(N_A, N_B))$ and has orthonormal columns (the **left singular vectors**), i.e. $U^\dagger U = I$; if $N_A \leq N_B$ this implies that it is unitary, and also $UU^\dagger = I$.
- S is of dimension $(\min(N_A, N_B) \times \min(N_A, N_B))$, diagonal with non-negative entries $S_{aa} \equiv s_a$. These are the so-called **singular values**. The number r of non-zero singular values is the **(Schmidt) rank** of M . In the following, we assume descending order: $s_1 \geq s_2 \geq \dots \geq s_r > 0$.
- V^\dagger is of dimension $(\min(N_A, N_B) \times N_B)$ and has orthonormal rows (the **right singular vectors**), i.e. $V^\dagger V = I$. If $N_A \geq N_B$ this implies that it is unitary, and also $VV^\dagger = I$.

Consider a lattice of L sites with d -dimensional local state spaces $\{\sigma_i\}$ on sites $i = 1, \dots, L$. In fact, while we will be naturally thinking of a one-dimensional lattice, the following also holds for a lattice of arbitrary dimension on which sites have been numbered; however, MPS generated from states on higher-dimensional lattices will not be manageable in numerical practice. The most general pure quantum state on the lattice reads

$$|\psi\rangle = \sum_{\sigma_1, \dots, \sigma_L} c_{\sigma_1 \dots \sigma_L} |\sigma_1, \dots, \sigma_L\rangle,$$

where we have exponentially many coefficients $c_{\sigma_1 \dots \sigma_L}$ with quite oblique content in typical quantum many-body problems. Let us assume that it is normalized. Can we find a notation that gives a more local notion of the state (while preserving the quantum non-locality of the state)? Indeed, SVD allows us to do just that. The result may look quite forbidding, but will be shown to relate profoundly to familiar concepts of quantum physics. There are three ways of doing this that are of relevance to us.

Left-canonical matrix product state. In a first step, we **reshape** the state vector with d^L components into a matrix Ψ of dimension $(d \times d^{L-1})$, where the coefficients are related as

$$\Psi_{\sigma_1, (\sigma_2 \dots \sigma_L)} = c_{\sigma_1 \dots \sigma_L}.$$

An SVD of Ψ gives

$$c_{\sigma_1 \dots \sigma_L} = \Psi_{\sigma_1, (\sigma_2 \dots \sigma_L)} = \sum_{a_1}^{r_1} U_{\sigma_1, a_1} S_{a_1, a_1} (V^\dagger)_{a_1, (\sigma_2 \dots \sigma_L)} \equiv \sum_{a_1}^{r_1} U_{\sigma_1, a_1} c_{a_1 \sigma_2 \dots \sigma_L},$$

where in the last equality S and V^\dagger have been multiplied and the resulting matrix has been reshaped back into a vector. The rank is $r_1 \leq d$. We now decompose the matrix U into a collection of d row vectors A^{σ_1} with entries $A_{a_1}^{\sigma_1} = U_{\sigma_1, a_1}$. At the same time, we reshape $c_{a_1 \sigma_2 \dots \sigma_L}$ into a matrix $\Psi_{(a_1 \sigma_2), (\sigma_3 \dots \sigma_L)}$ of dimension $(r_1 d \times d^{L-2})$, to give

$$c_{\sigma_1 \dots \sigma_L} = \sum_{a_1}^{r_1} A_{a_1}^{\sigma_1} \Psi_{(a_1 \sigma_2), (\sigma_3 \dots \sigma_L)}.$$

Ψ is subjected to an SVD, and we have

$$\begin{aligned} c_{\sigma_1 \dots \sigma_L} &= \sum_{a_1}^{r_1} \sum_{a_2}^{r_2} A_{a_1}^{\sigma_1} U_{(a_1 \sigma_2), a_2} S_{a_2, a_2} (V^\dagger)_{a_2, (\sigma_3 \dots \sigma_L)} \\ &= \sum_{a_1}^{r_1} \sum_{a_2}^{r_2} A_{a_1}^{\sigma_1} A_{a_1, a_2}^{\sigma_2} \Psi_{(a_2 \sigma_3), (\sigma_4 \dots \sigma_L)}, \end{aligned}$$

where we have replaced U by a set of d matrices A^{σ_2} of dimension $(r_1 \times r_2)$ with entries $A_{a_1, a_2}^{\sigma_2} = U_{(a_1 \sigma_2), a_2}$ and multiplied S and V^\dagger , to be reshaped into a matrix Ψ of dimension $(r_2 d \times d^{L-3})$, where $r_2 \leq r_1 d \leq d^2$. Upon further SVDs, we obtain

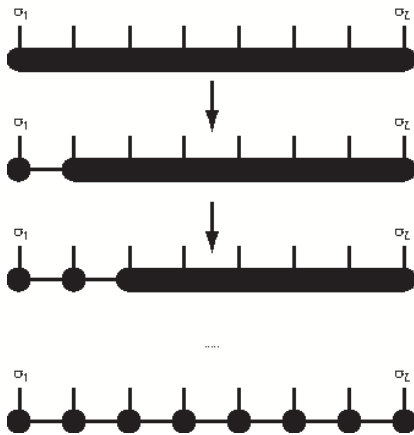
$$c_{\sigma_1 \dots \sigma_L} = \sum_{a_1, \dots, a_{L-1}} A_{a_1}^{\sigma_1} A_{a_1, a_2}^{\sigma_2} \dots A_{a_{L-2}, a_{L-1}}^{\sigma_{L-1}} A_{a_{L-1}}^{\sigma_L}$$

or more compactly

$$c_{\sigma_1 \dots \sigma_L} = A^{\sigma_1} A^{\sigma_2} \dots A^{\sigma_{L-1}} A^{\sigma_L},$$

where we have recognized the sums over a_1, a_2 and so forth as matrix multiplications. The last set of matrices A^{σ_L} in fact consists of column vectors. If we wish, dummy indices 1 may be introduced in the first and last A to turn them into matrices, too. In any case, the (arbitrary) quantum state is now represented exactly in the form of a **matrix product state**:

$$|\psi\rangle = \sum_{\sigma_1, \dots, \sigma_L} A^{\sigma_1} A^{\sigma_2} \dots A^{\sigma_{L-1}} A^{\sigma_L} |\sigma_1, \dots, \sigma_L\rangle.$$



Graphical representation of an iterative construction of an exact MPS representation of an arbitrary quantum state by a sequence of singular value decompositions.

上面介绍的是左正则的，还有右正则以及混合正则。但这不是我们本次计算的重点，我们只想要知道系统的状态可以写成矩阵乘积的形式就可以了。

还需要强调的一点是：

The maximal dimensions of the matrices are reached when for each SVD done the number of non-zero singular values is equal to the upper bound (the lesser of the dimensions of the matrix to be decomposed). Counting reveals that the dimensions may maximally be $(1 \times d)$, $(d \times d^2)$, \dots , $(d^{L/2-1} \times d^{L/2})$, $(d^{L/2} \times d^{L/2-1})$, \dots , $(d^2 \times d)$, $(d \times 1)$, going from the first to the last site (I have assumed L even for simplicity here). This shows that in practical calculations it will usually be impossible to carry out this exact decomposition explicitly, as the matrix dimensions blow up exponentially.

具有周期边界条件的MPS：

For periodic boundary conditions, the natural generalization of the MPS form is to make all matrices of equal dimensions ($D \times D$); as site L connects back to site 1, we make the MPS consistent with matrix multiplications on all bonds by taking the trace (see Figure below):

$$|\psi\rangle = \sum_{\sigma} \text{tr}(M^{\sigma_1} \dots M^{\sigma_L}) |\sigma\rangle \quad (\text{MPS for PBC}).$$

While *a priori* not more accurate than the other, it is much better suited and computationally far more efficient.



Representation of a periodic boundary condition MPS; the long line at the bottom corresponds to the trace operation.

4. Heisenberg Model

Imaginary Time-Evolution Block Decimation(ITEBD) (4.7)

1. $e^{A+B} = e^A e^B$?
2. $e^{A+B} = e^A e^B e^{-\frac{1}{2}[A,B]}$ (disentangling theorem)
3. $e^{A+B} = e^A e^B e^{-\frac{1}{2}[A,B]} e^{\frac{1}{6}(2[B,[A,B]] + [A,[A,B]])} \dots$ (Zassenhaus formula)

考虑一维 Heisenberg 模型 $\mathcal{H} = \sum_i \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_{i+1}$, 定义: $h_i = \hat{\mathbf{S}}_i \cdot \hat{\mathbf{S}}_{i+1}$

$$\begin{aligned} e^{-\tau \mathcal{H}} &= e^{-\tau(\mathcal{H}_{\text{odd}} + \mathcal{H}_{\text{even}})} = e^{-\tau \mathcal{H}_{\text{odd}}} e^{-\tau \mathcal{H}_{\text{even}}} + O(\tau^2) \\ &= e^{-\tau(\dots + h_1 + h_3 + \dots)} e^{-\tau(\dots + h_2 + h_4 + \dots)} + O(\tau^2) \\ &= \dots e^{-\tau h_1} e^{-\tau h_3} \dots e^{-\tau h_2} e^{-\tau h_4} \dots + O(\tau^2) \\ &\approx \dots e^{-\tau h_1} e^{-\tau h_2} e^{-\tau h_3} e^{-\tau h_4} \dots \end{aligned}$$

$$h_i = S_i^z S_{i+1}^z + \frac{1}{2}(S_i^+ S_{i+1}^- + S_i^- S_{i+1}^+)$$

5. 自旋梯子算符

$$S^\pm |j, m\rangle = \sqrt{(j \mp m)(j \pm m + 1)} |j, m \pm 1\rangle$$

6. 基本步骤

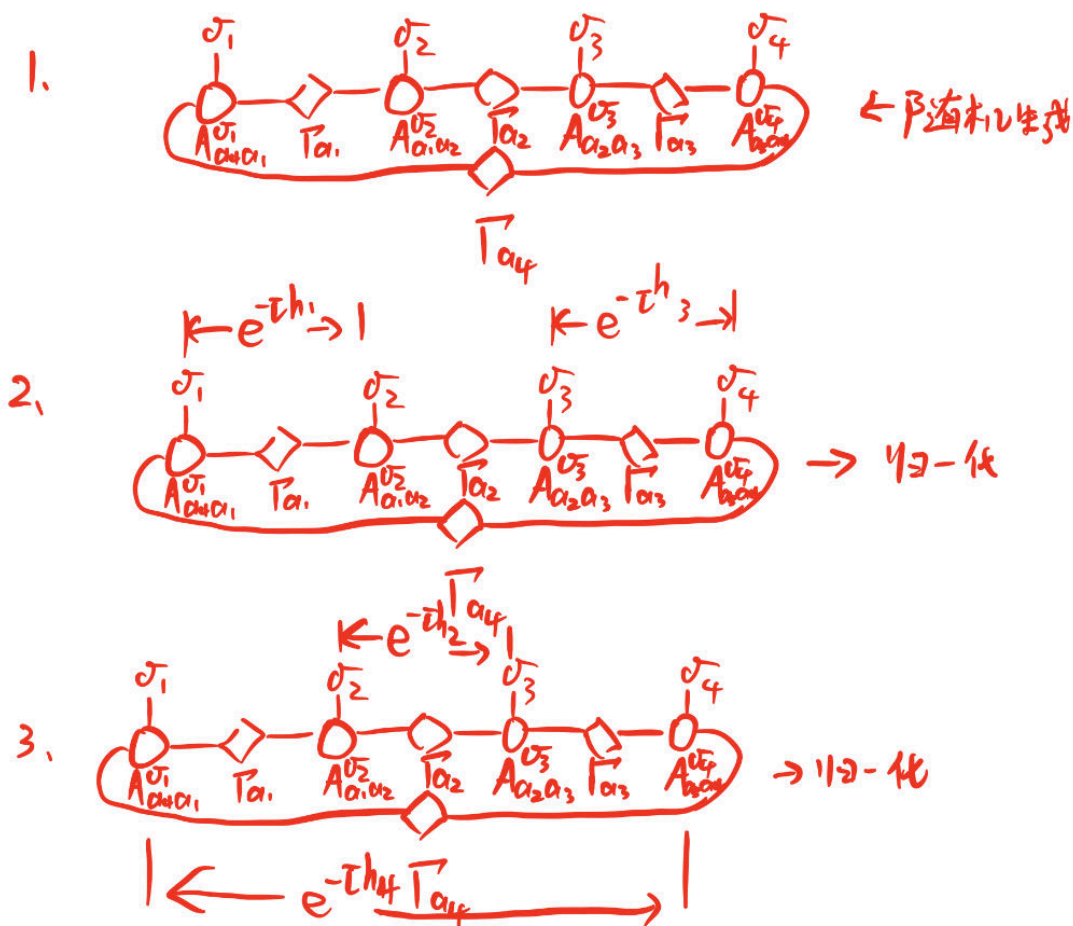
ITEBD (4.7)

基本步骤:

1. 随机产生一个具有周期性结构的MPS;
 2. 将Heisenberg链中编号为奇数的键作用到该MPS, 得到一个新的MPS;
 3. 将Heisenberg链中编号为偶数的键作用到该MPS, 得到一个新的MPS;
 4. 多次反复作用直到收敛。
- 注意每次作用后, 将奇异值归一化, 通过归一化系数估算基态能量并判断计算是否收敛,
 - 估算的方法: $e^{-\tau h}|\psi\rangle \approx \alpha|\psi\rangle \approx e^{-\tau E_0}|\psi\rangle$ 。

要点:

1. 图和指标的对应;
2. 利用奇异值分解做近似;
3. 记得要“归一化”;



代码编写

1.参数

我们首先需要确定参数的个数，为了是我们的程序更加灵活，我们期望其理论上可以计算任意自旋和周期的heisenberg model；因此，参数如下：

- ☒ 周期（或者粒子个数）L（本程序仅适用于偶数）
- ☒ 粒子的自旋量子数 s
- ☒ 矩阵非物理指标的维数 D
- ☒ 单位虚时 τ
- ☒ 数值零 jz
- ☐ 。。。

2. 自旋算符的矩阵表示

我们需要计算z方向的自旋算符以及两个梯子算符，通过循环的方式完成。虽然他们都是实数域内的，但是考虑到波函数是复的；因此，我们将它们设为复的。

3. h_i 的矩阵表示

$$h_i = S_i^z S_{i+1}^z + \frac{1}{2}(S_i^+ S_{i+1}^- + S_i^- S_{i+1}^+)$$

$$S^\pm |j, m\rangle = \sqrt{(j \mp m)(j \pm m + 1)} |j, m \pm 1\rangle$$

由于虚时演化算符是 $\exp(-\tau h_i)$ ，因此，虚时演化算符对应的矩阵可以使用`torch::matrix_exp`来计算，这是专门用来计算矩阵指数的函数；当然，当 τ 比较小的时候，也可以用 $I - \tau h_i$ 来近似。

采用直乘的算法即可。

4. 生成随机的周期MPS

随机生成上面说的周期MPS的张量即可，注意张量的维度以及物理指标的位置。至于有没有么正化问题不大，经过一次更新就好了。

当然，这样有极小的可能还没有迭代就截止了；可以打印一下迭代次数来检验。问题不大。

5. 虚时演化

将哈密顿量作用在MPS上并更新奇异值，这里需要注意，我们将用Hilbert子空间进行近似，因此只取前D向最大的；libtorch内置的svd奇异值的排序也正是descending的。

技术细节见代码；我们取偶数个粒子。

6. 改进

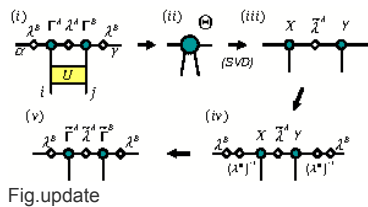
计算过程中发现数值稳定性不是很好，可能是 τ 对于相应的精度来说太大了。

由于改变 τ 就需要改变相应的哈密顿量对应的矩阵，因此，我们可以适当根据循环次数调整 τ 的值。

7. 总结

写这个程序还是很不容易的，特别是更新MPS那一块。

主要的步骤是来自文献^[2]的流程图：



In order to update the MPS after gate U has been applied, see Fig.update, we first contract the tensor network (i) into a single tensor $\Theta_{\alpha i j \gamma}$ (ii). We then compute the *singular value decomposition* of Θ according to the index bipartition $[\alpha i] : [j \gamma]$, namely $\Theta = \sum_{\beta} X_{[\alpha i] \beta} \tilde{\lambda}_{\beta}^A Y_{\beta [j \gamma]}$ as in(iii). We introduce λ^B back into the network (iv) and form tensors $\tilde{\Gamma}^A$ and $\tilde{\Gamma}^B$ in(iv) by attaching to X and Y the inverse of the Schmidt coefficients λ^B . All such matrix manipulations are achieved with $O(d^2 \chi^2)$ space and $O(d^3 \chi^3)$ and need to be performed only once in order to update the MPS for the whole infinite chain.

结果

通过改变一下参数，我们可以的到基态能量。（波函数就把MPS乘起来就好，这里就不给出了）

- ☒ 周期（或者粒子个数）L（本程序仅适用于偶数）
- ☒ 粒子的自旋量子数 s

- ☑ 矩阵非物理指标的维数 D
- ☑ 单位虚时 τ

算法的数值稳定性是非常不错的；至于与学长给的参考值有细微差别，原因在于对奇异值的归一化的方式上。学长应该是截断后在计算二范数，但我认为时间演化算符作用后的归一化系数应该是一个整体，归一化系数不应该只是截断后的。

把"torch::Tensor E = torch::norm(S,2);"中"S"改为"Gamma[i]"就可以得到和学长一样的结果。两个结果相差不大。

```
torch::Tensor E = torch::norm(S,2);
Gamma[i] = Gamma[i]/E.item<double>();
Gamma[i] = Gamma[i]/torch::norm(Gamma[i]).item<double>();
g_E += std::log(E.item<double>())/(-tau);
```

118

`A[i] = torch::einsum("a,abc->abc",{torch::reciprocal(Gamn`

PROBLEMS

OUTPUT

DEBUG CONSOLE

TERMINAL

PORTS

GITLENS

●

●

M

M

M

M

M

○

●

heaven@heaven:~/Desktop/doc/Linear_Algebra/ITEBD/build\$./mps

times is :9384

the ground energy is:-1.37409

●

heaven@heaven:~/Desktop/doc/Linear_Algebra/ITEBD/build\$./mps

times is :9369

the ground energy is:-1.37409

●

heaven@heaven:~/Desktop/doc/Linear_Algebra/ITEBD/build\$./mps

times is :7332

the ground energy is:-1.37409

●

heaven@heaven:~/Desktop/doc/Linear_Algebra/ITEBD/build\$./mps

times is :7579

the ground energy is:-1.37409

●

heaven@heaven:~/Desktop/doc/Linear_Algebra/ITEBD/build\$./mps

times is :12190

the ground energy is:-1.37409

○

heaven@heaven:~/Desktop/doc/Linear_Algebra/ITEBD/build\$

周期（或者粒子个数）L (本程序仅适用于偶数)	4	6	8	4	4	4	4	4	4	4
粒子的自旋量子数 s	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0	1.0/2
矩阵非物理指标的维数 D	4	4	4	4	4	6	8	9	10	4
单位虚时 τ	0.01	0.01	0.01	0.05	0.005	0.01	0.01	0.01	0.01	0.01
基态能量 g_E	-1.37409	-1.37409	-1.37409	-1.37676	-1.37379	-1.39972	-1.40112	-1.40118	-1.40125	-0.441079

1. Schollwöck, U. (2011). The density-matrix renormalization group in the age of matrix product states. Annals of Physics, 326(1), 96–192.
<https://doi.org/10.1016/j.aop.2010.09.012> pdf ↗

2. Vidal, G. (2007). Classical Simulation of Infinite-Size Quantum Lattice Systems in One Spatial Dimension. Physical Review Letters, 98(7), 070201.
doi:10.1103/physrevlett.98.070201 pdf ↗