

Exercise sessions 12–13: Singular value decomposition.

Exercise 1: Over-defined system of linear equations [6 pts]

Consider an over-defined system of linear equations

$$\begin{pmatrix} 3 & 2 \\ 4 & 5 \\ 1 & 1 \end{pmatrix} \cdot \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 1 \\ 4 \\ 1 \end{pmatrix}$$

1. Find $x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$ that minimizes the norm of residual $r(x) = \|Ax - b\|$ by computing the Penrose inverse of the matrix A . You can use Matlab's command `pinv`.
2. Illustrate that your solution indeed minimizes the norm of residual $r(x)$ by plotting its magnitude in the vicinity of the solution x . We suggest to use Matlab's command `contour` for this purpose.

Submit: your report as a PDF file as well as your scripts. The points are distributed in the following way:

- [3 pts] for the solution of the system;
- [3 pts] for the plot of the residual.

Exercise 2: Image compression using SVD and the Ising model [10 pts]

Part A - Image compression

The singular value decomposition can be used for image compression. The idea is to perform the singular value decomposition of the matrix representing the image and store only a subset of selected significant singular values and corresponding vectors. The approximate image can then be easily reconstructed using this reduced amount of data.

1. Download photograph of Julia Yeomans available on Moodle (file `yeomans.jpg`) and load into Matlab using `imshow`. Using Matlab's `svd`, perform the singular value decomposition of the image. Plot the obtained singular values on a logarithmic scale. What is the rank of the image?
2. Approximate the original image by a lower-rank matrix of rank N . On the same figure, plot the original image as well as the approximated images for the number of singular values $N = 4, 20, 100$. Comment on the quality of the approximated images by making a connection with the previous point. How much memory is required to store the compressed versions of the original image?

Part B - Monte Carlo simulations of the Ising model

In the second part of this exercise, we want to make a direct application of image compression to gain some understanding of a physical system from its instantaneous configurations.

The Ising model is probably one of the simplest and yet richest models in physics. It was invented as a means to understand magnetism, and to this day remains a highly important topic in mathematics and theoretical physics. In its simplest form, it models a magnetic material with spins on a lattice represented by integer variables $\sigma_i = \pm 1$. The cooperative interaction between the spins is introduced as a nearest-neighbor interaction as given by the following Hamiltonian:

$$H = -J \sum_{\langle i,j \rangle} \sigma_i \sigma_j \quad J > 0, \quad (1)$$

where $\langle i, j \rangle$ stands for nearest-neighbor bonds on a lattice. On the square lattice, this model is exactly solvable (an extraordinary result by Onsager), and features a second order phase transition at $kT_c/J = \frac{2}{\ln(1+\sqrt{2})}$. Above this temperature, the spins are disordered (paramagnetic phase), whereas below this temperature, they tend to order, and the system exhibits a non-zero net magnetization (ferromagnetic phase). One nice feature of second-order phase transitions is that, at criticality, an additional symmetry called scale invariance appears. This symmetry corresponds to the fact that, at this temperature, there is no characteristic length scale appearing. Intuitively, this means that the ferromagnetic islands that appear at this temperature can be of any size, whereas at lower temperature, there is a characteristic size of the islands. For more information, c.f. for instance Julia Yeoman's book *Statistical mechanics of phase transitions*.

The Ising model can be studied, for instance, using Monte Carlo simulations, as we did in the Solid State Physics II course using a Jupyter notebook. Files `Fig1.png`, `Fig2.png` and `Fig3.png` correspond to snapshots of spin configurations of the square lattice Ising model taken at equilibrium. One of them is taken below the critical temperature, one above, and one exactly at the critical temperature.

1. Download the three snapshots. Using Matlab's `svd`, perform the singular value decomposition of the images (notice that they are not in RGB). Plot the obtained singular values on a logarithmic scale, for all three images on the same graph so that these singular values can be compared.
2. Approximate the images by lower-rank matrices of rank $N = 50$.
3. Can you predict which snapshot has been taken at which temperature from the image? And can you predict it from the decay of the singular values? Explain your reasoning.

Submit: your report as a PDF file as well as your scripts. The points are distributed in the following way:

- [4 pts] for the plot of the singular values (1 plot for Julia Yeomans' picture, 1 figure with 3 plots for the Ising snapshots)
- [2 pts] for the plot of the approximated images (1/2 per image)
- [1 pt] for the answers to the question about the compressed image of Julia Yeomans
- [3 pts] for the answers to the question about the relation between the image and the temperature, and the relation between the singular values and the temperature.

Exercise 3: Compressing a quantum ground state using SVD [14 pts]

Part A - Schmidt decomposition and iterative Schmidt decomposition

After 20 years of development, tensor networks have become a crucial tool for studying lattice spin systems. One of the key ideas of tensor networks is that ground states of many-body Hamiltonian have a special feature that allows them to be compressed with high efficiency. In particular, in this exercise, we are going to see how to write a wavefunction as a *matrix product state* (see below), and how that allows to compress the function and thus study large systems.

Consider a 1D spin chain model with N sites and a local Hilbert space of dimension d (for instance, $d = 2$ for a spin-1/2 chain, $d = 3$ for a spin-1 chain, etc.).

1. Given N and d , what is the dimension of the Hilbert space of the full chain?

In many-body physics, one is typically interested in ground-state or low-energy properties. The power of tensor networks lies in the fact that even though the size of the Hilbert space is exponential in the number of sites, the region of the Hilbert space where the ground state might lie is much smaller. We are going to illustrate this by compressing the ground state of the AKLT model, which is a spin-1 model, for two chain sizes.

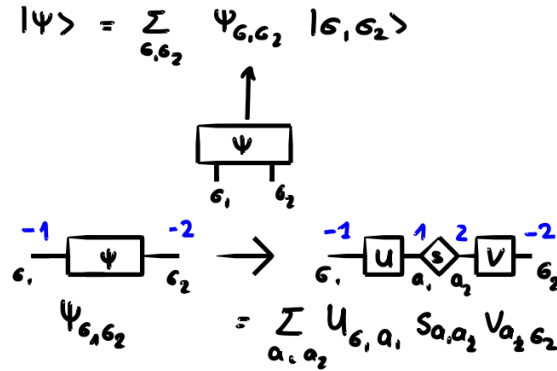


Figure 1: Decomposition of a two-sites wavefunction. The blue indices correspond to one way of giving the indices of the tensors as an input to the `ncon.m` function. The SVD applied to a wavefunction typically goes by the name of *Schmidt decomposition*.

Let us start with a Schmidt decomposition of a two-site wave function. The wavefunction can be considered as a rank-2 tensor (i.e. a matrix).

1. Load the wavefunction in `aklt2sites.mat`. Perform a SVD decomposition $\text{psi} \rightarrow U, s, V$.
2. What is the rank of the singular values matrix? We will call it the *bond dimension*.
3. Recover the wavefunction from `U, s` and `V`. These matrices can be seen as rank-2 tensors. Use the provided `ncon.m` function to recover `psi` (see Fig. 1).

To be able to treat cases with more sites, we need to implement what we will call the iterative Schmidt decomposition. This method is described below and illustrated in Fig. 2. Write the function `function [Alist, slist, Vlist]= iterativeschmidt(psi, threshold)` that implements the following:

- Given a wavefunction ψ and a threshold
- `d1 = size(psi,1); d2 = size(psi,2); n = length(size(psi))-2`
- `psi` $\xrightarrow{\text{reshape}}$ `[d1*d2, d2(n-1)*d1]`
- `psi` $\xrightarrow{\text{SVD}}$ `U, s, V`
- Truncate `U, s` and `V` based on the rank `chi` of the singular values matrix.
- `Alist` $\leftarrow (U*s \xrightarrow{\text{reshape}} [d1, d2, chi])$
- `Vlist` $\leftarrow V$
- `red` $\leftarrow (V \xrightarrow{\text{reshape}} [d2*chi, d2⁽ⁿ⁻²⁾*d1])$
- Keep reducing the resulting part by successive SVDs until only `d1` is left.

Because this procedure amounts to writing the wavefunction as a product of tensors, the resulting description of the wavefunction is called a *matrix product state*. Test your implementation with the `test_iterativeschmidt.m` code provided. Notice that it uses the `ncon.m` function provided as well. Pay attention to the reshaping steps.

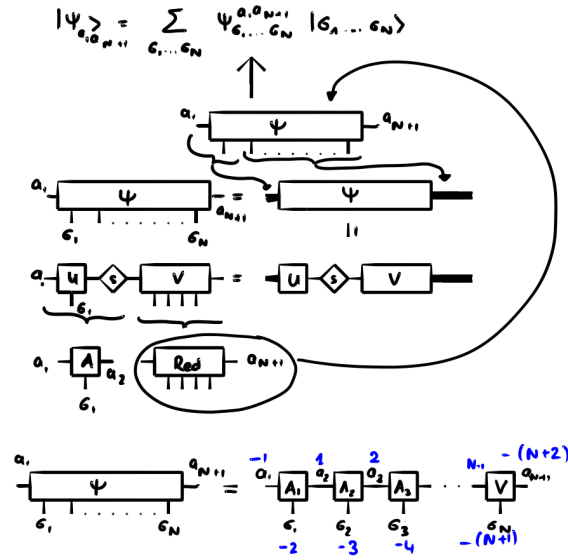


Figure 2: Illustration of the iterative Schmidt decomposition. The blue indices correspond to one way of giving the indices of the tensors as an input to the `ncon.m` function to reconstruct the wavefunction.

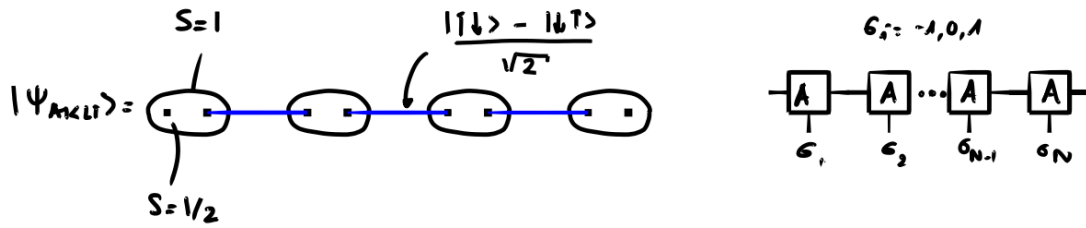


Figure 3: Illustration of the AKLT ground state wavefunction written as a translation invariant matrix product state. The key idea is that each spin 1 can be described using two virtual spins 1/2, projected onto their spin 1 sector.

Part B - The AKLT model

The AKLT Hamiltonian is a spin-1 chain Hamiltonian given by

$$H_{\text{AKLT}} = \sum_i \mathbf{S}_i \cdot \mathbf{S}_{i+1} + \frac{1}{3} (\mathbf{S}_i \cdot \mathbf{S}_{i+1})^2 + \frac{2}{3}. \quad (2)$$

1. Load the wavefunctions in `aklt_open.mat` and `aklt2sites_open.mat`, and apply the iterative Schmidt decomposition to these wavefunctions.
2. The `ncon` function is made to simplify the contraction of tensors. You can see examples of how to use it in the `test_iterativeschmidt.m` code. Explain how to use it to recover the wavefunctions from the iterative decompositions which you performed, and show that this works.
3. How many numbers were needed to fully described the wavefunctions `psiaklt` and `psiaklt2` in their original form? How many are needed in the compressed form of `Alist` and `V`?

We can understand this very impressive result from a physical standpoint. First, one needs to notice that

$$H_{\text{AKLT}} \propto 2 \sum_i \left((\mathbf{S}_i + \mathbf{S}_{i+1})^2 - 0 \right) \left((\mathbf{S}_i + \mathbf{S}_{i+1})^2 - 2 \right) \quad (3)$$

1. Consider the wavefunction where each spin $S = 1$ on each site is decomposed into two virtual $S = 1/2$ spins, and where each pair of neighbouring spin $1/2$ are antisymmetrized in a singlet state $S = 0$. This wavefunction can be described by

$$|\psi\rangle_{a_1, a_{N+1}} = \sum_{\{\sigma_i\}} A_{a_1}^{\sigma_1} \cdots A_{a_{N+1}}^{\sigma_N} |\sigma_1, \dots, \sigma_N\rangle \quad (4)$$

where

$$A^1 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad A^0 = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 1 \\ 0 & 0 \end{pmatrix} \quad A^{-1} = \frac{1}{\sqrt{2}} \begin{pmatrix} 0 & 0 \\ -1 & 0 \end{pmatrix} \quad (5)$$

(see Fig. 3). Using the `ncon.m` function, show that this is indeed the ground state wavefunction of the AKLT Hamiltonian.

2. Except from the fact that we work with open boundary conditions here, one can see that the wave function built that way is in principle translation invariant, because on each site it is described by the same tensor A . Is that suprising? Was it clear from your decomposition of `psiaklt2`? What does that tell you about the decomposition of a wavefunction as a tensor product?
3. (Bonus - 2pts) Can you prove by hand that this wavefunction has to be the ground state of the AKLT Hamiltonian?

Submit: your report as a PDF file, `iterativeschmidt.m`, as well as your scripts. The points are distributed in the following way:

- [1 pt] for the question about the Hilbert space dimension
- [3 pts] (1 per question) for the three questions about `aklt2sites.mat`
- [4 pts] for the implementation of the iterative Schmidt decomposition
- [2 pts] for a correct Schmidt decomposition and reconstruction of each wavefunction
- [2 pts] for the correct answer about the number of degrees of freedom in the description of the wavefunction in their usual forms and in their matrix product state form.
- [1 pt] for building the translational invariant ground state
- [1 pt] for the answer about the translation invariance and the differences between the A tensors.