Project title

Your Name

(Dated: July 8, 2020)

A paper usually includes an abstract, a concise summary of the work covered at length in the main body of the paper. Please also write a short abstract of your project.

I. GUIDELINES

Please write a short paper about your project. There are no strict length limits, but please write **at least two pages in the layout of this template** (including figures, excluding references and code listings). Please structure your paper in a scientific way, and include your references and your code. There are LaTeXpackages you can use to preserve the indentation of your code, e.g. the `listings` package which is demonstrated in the Appendix A.

This sample document makes use of of REVTeX 4.2, therefore you will need to install it to be able to compile this document yourself. Further information can be found in the REVTeX 4.2 documentation included in the distribution or available at `http://journals.aps.org/revtex/`.

A. Example citations

By default, citations are numerical[1], some more citations [2–6].

B. Exampe figure

Including and referring to figures is as usual, see for instance Fig. 1

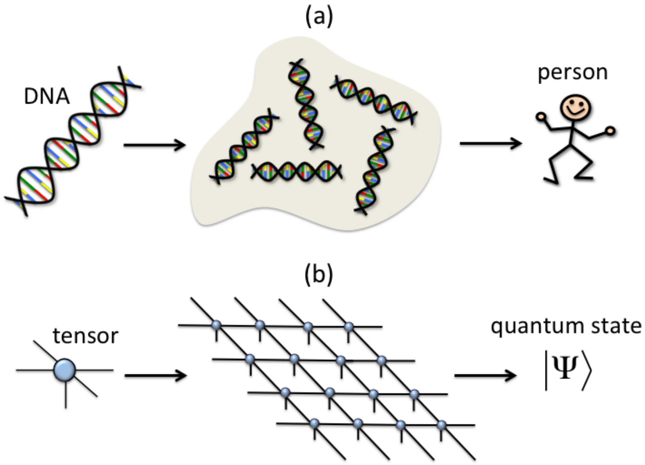

FIG. 1. Example of a figure [7].

[1] A. Einstein, Yu. Podolsky, and N. Rosen (EPR), Phys. Rev. **47**, 777 (1935).

[2] R. P. Feynman, Phys. Rev. **94**, 262 (1954).

[3] N. D. Birell and P. C. W. Davies, *Quantum Fields in Curved Space* (Cambridge University Press, 1982).

[4] J. G. P. Berman and J. F. M. Izrailev, Stability of nonlinear modes, Physica D **88**, 445 (1983).

[5] E. Witten, (2001), hep-th/0106109.

[6] E. B. Davies and L. Parns, Trapped modes in acoustic waveguides, Q. J. Mech. Appl. Math. **51**, 477 (1988).

[7] R. Orus, A practical introduction to tensor networks: Matrix product states and projected entangled pair states, Annals of Physics **349**, 117 (2013), 1306.2164.

Appendix A: Code listing

Please copy your code in the appendix.

```
"""

Module to generate the Hamiltonian of the transverse field Ising model.

H = -J sum_i sigma^x_i sigma^x_{i+1} - g sum_i sigma^z i.

Used in the solution of exercise 5.1

"""

```

```
import numpy as np
import scipy
from scipy import sparse
import scipy.sparse.linalg
import matplotlib.pyplot as plt

Id = sparse.csr_matrix(np.eye(2))
Sx = sparse.csr_matrix([[0., 1.], [1., 0.]])
Sz = sparse.csr_matrix([[1., 0.], [0., -1.]])
Splus = sparse.csr_matrix([[0., 1.], [0., 0.]])
Sminus = sparse.csr_matrix([[0., 0.], [1., 0.]])


def singesite_to_full(op, i, L):
    op_list = [Id]*L  # = [Id, Id, Id ...] with L entries
    op_list[i] = op
    full = op_list[0]
    for op_i in op_list[1:]:
        full = sparse.kron(full, op_i, format="csr")
    return full


def gen_sx_list(L):
    return [singesite_to_full(Sx, i, L) for i in range(L)]


def gen_sz_list(L):
    return [singesite_to_full(Sz, i, L) for i in range(L)]


def gen_hamiltonian_periodic(sx_list, sz_list, g, J=1.):
    """ assumes periodic boundery conditions """
    L = len(sx_list)
    H = sparse.csr_matrix((2**L, 2**L))
    for j in range(L):
        H = H - J *( sx_list[j] * sx_list[(j+1)%L])
        H = H - g * sz_list[j]
    return H
```