Jupiter's Turkish Taco

by:

Santiago Copado Obregón
Sebastián Domínguez Calderón
Deniz Adigüzel
Lorenzo Matteo Salvatori

CONTENT

1. PROBLEM DEFINITION

Classical optimization problem

Hamiltonian

H

2. ANALOG HAMILTONIAN

$$\mathbf{H} = \sum_{i \in S} b_i v_i + [\sum_{i \in S} b_i d_i + \sum_{j=0}^{nqbin} p_j(2^j) - d_{max}]^2$$

$$\mathbf{H} = \sum_{i \in S} -(\frac{1 - Z(i)}{2}) v_i + [\sum_{i \in S} (\frac{1 - Z(i)}{2}) d_i + \sum_{j=0}^{nqbin} (\frac{1 - Z(j)}{2})(2^j) - d_{max}]^2$$

- hours needed : many
- errors made : lots of
- number of tears shed : yes
- slept ? lol

Vision

EXACT EIGENVALUES = BRUTEFORCE

Goal : find that the ground state is the correct solution for several easy cases

[((-46+0j), '1110100000'), ((-45+0j), '1111000000'), ((-45+0j), '11010111000')......

QOAO VS ADIABATIC

Step 1

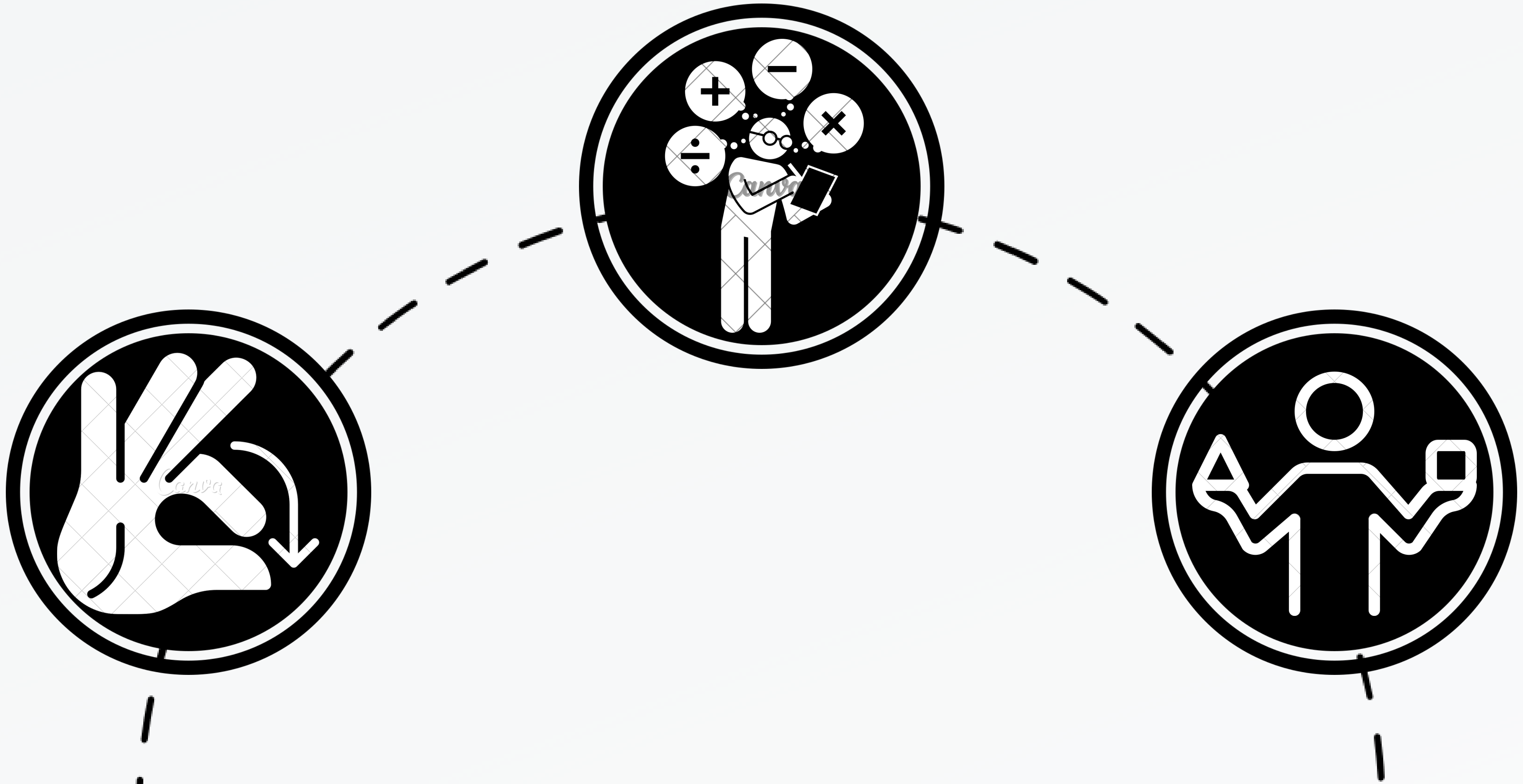
Reduce the hamiltonian
11 qubits -> 6 qubits

Step 2

Solve the new hamiltonian using the QAOA algorithm.
and using Quantum Adiabatic Evolution

Step 3

Compare the
different solutions



Reduced hamiltonian example

$$\mathrm{H} = -\sum_{i=0}^{L-1} \frac{1}{2} v[i] \cdot \left(\frac{1-Z(i)}{2}\right) + \left(\sum_{i=0}^{L-1} \frac{1}{2} d[i] \cdot \left(\frac{1-Z(i)}{2}\right) - d_{max}\right)^2$$

QAOA algorithm

Quantum Adiabatic Evolution

VALUES

QAOA

```
from qibo.symbols import Z, X
from qibo.hamiltonians import SymbolicHamiltonian

def build_cost_hamiltonian(values: list[int], duration: list[int], m

    cost_hamiltonian = 0

    L = len(values)
    d_max = max_duration

    term_1 = sum(values[i] * (1 - Z(i)) / 2 for i in range(L))

    a = sum(duration[i] * (1 - Z(i)) / 2 for i in range(L))
```

- [('9.95 %', '010111'), ('9.49 %', '011110'), ...

...The best solution is: 010111

- the solution is: 011101

ADIABATIC

```
from qibo.hamiltonians.hamiltonians import Hamiltonian


# build initial (H0) and target (H1) hamiltonians
H0 = build_initial_hamiltonian(nqubits=nqubits)
H1 = build_cost_hamiltonian(values=values, duration=duration, max_duration=max_duration)

# calculate the dense hamiltonian from the symbolic hamiltonian
H1_dense = Hamiltonian(nqubits, H1.calculate_dense().matrix)
H0_dense = Hamiltonian(nqubits, H0.calculate_dense().matrix)

dt = 0.1

T = 20


def s(t): return t

# construct the adiabatic model
adiabatic_model = AdiabaticEvolution(H0_dense, H1_dense, s, dt)
```

CSFQ - ISING SCHEDULES SETUP

$$\hat{H}(t) = \sum_i \left(h_i^x(t)\hat{X} + h_i^z(t)\hat{Z} \right) + \frac{1}{2}\sum_{i,j} J_{ij}(t)\hat{Z}_i\hat{Z}_j$$

$$\hat{H}(t_0) = \sum_i h_i^x(0)\hat{X}_i = \sum_i \hat{X}_i$$

$$\hat{O}_i := \left(\frac{1 - \hat{Z}_i}{2} \right)$$

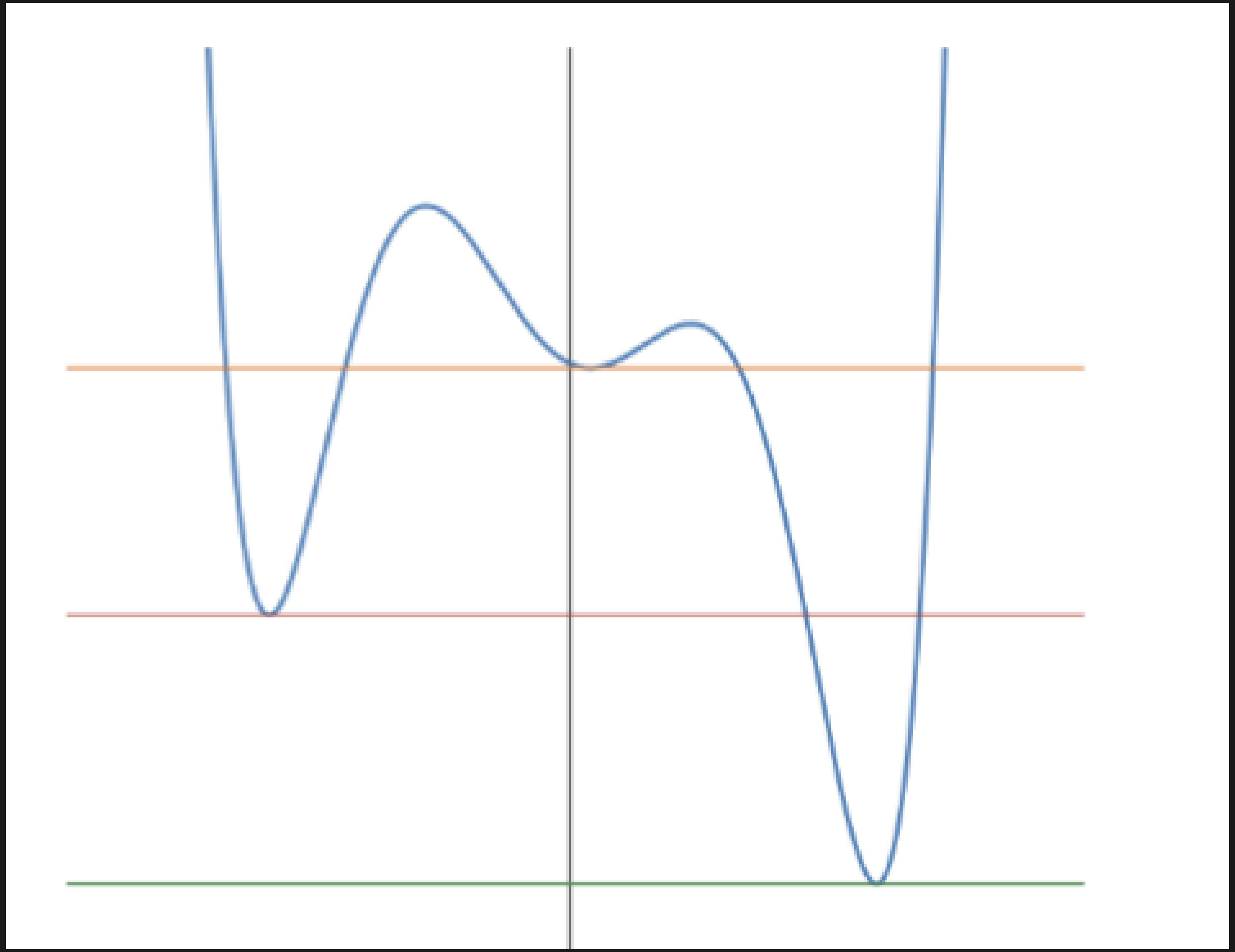
$$\hat{H}(t_f) = -\sum_i v_i \hat{O}_i + \left(\sum_i \hat{O}_i d_i + \sum_j 2^j \hat{O}_j - d_{max} \right)^2$$

$$\hat{H}(t_f) \approx -\sum_i (v_i + 2 d_{max} d_i) \hat{Z}_i + \sum_{i,j} d_i d_j \hat{Z}_i \hat{Z}_j$$

$$\hat{H}(t_f) \approx -\sum_i h_i^z(t_f) \hat{Z}_i + \sum_{i,j} J_{ij}(t_f) \hat{Z}_i \hat{Z}_j$$

$$t : t_0 \longrightarrow t_f$$
$$h_i^x(t) : 1 \longrightarrow 0$$
$$h_i^z(t) : 0 \longrightarrow v_i + 2d_i d_{max}$$
$$J_{ij}(t) : 0 \longrightarrow d_i d_j$$

Intuition:
Change potential to tilted washboard slowly, so previous ground state fits new potential ground state



HOW SLOW IS SLOW ENOUGH?

PERTURBATION THEORY (NONDEGENERATE)

$$H(t) = H^0 + e^{t/\tau} H^1, \qquad -\infty \le t \le 0$$

$$\tau \gg 1/\omega_{\min}$$

Slow enough = time associated to smallest transition frequency

Steps need to be longer than “slow enough”

Small frequency = very slow steps
High frequency = quick steps

Energy Plots

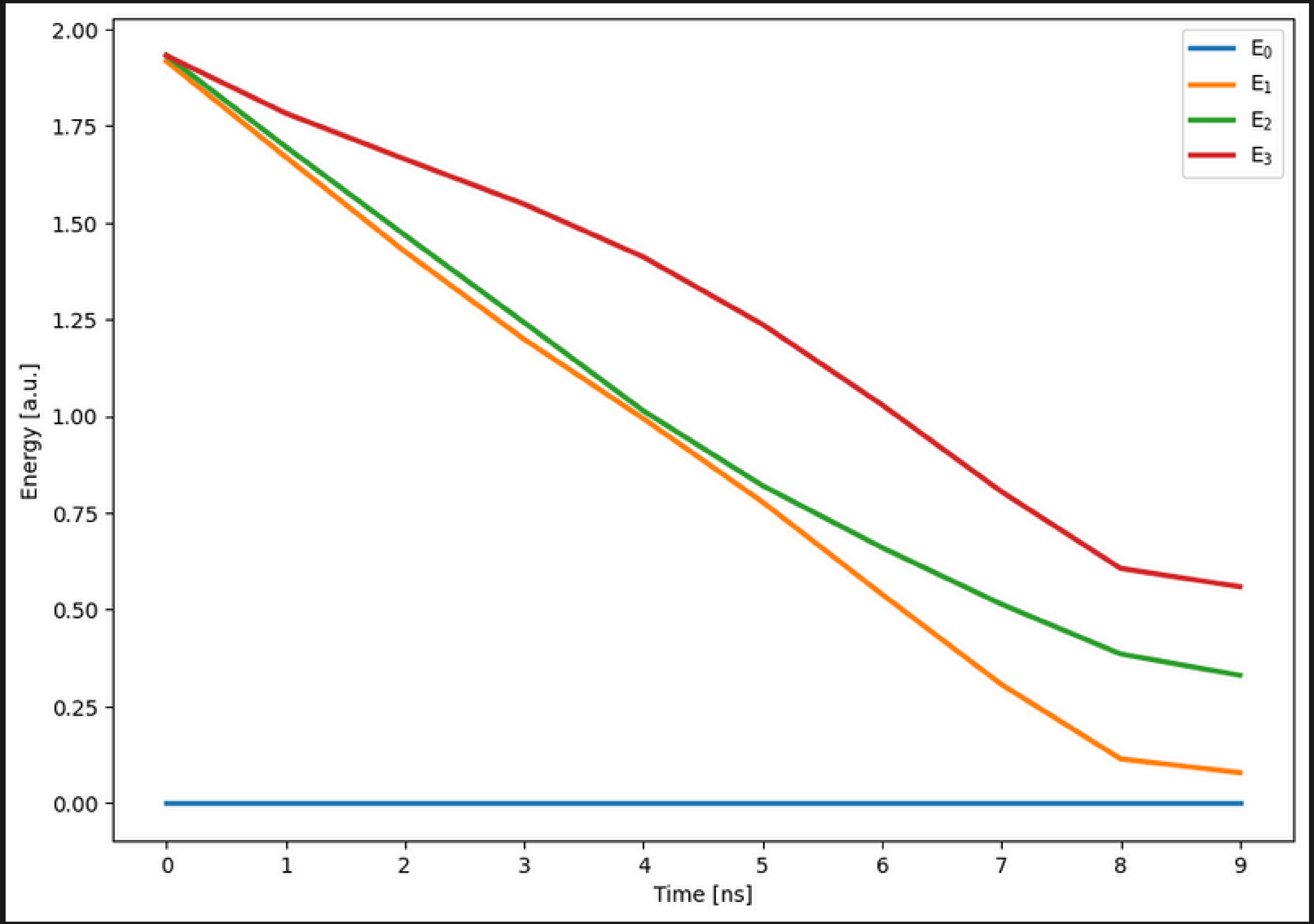


Fig 1: Energy levels of the states with a linear Schedule

Energy Plots

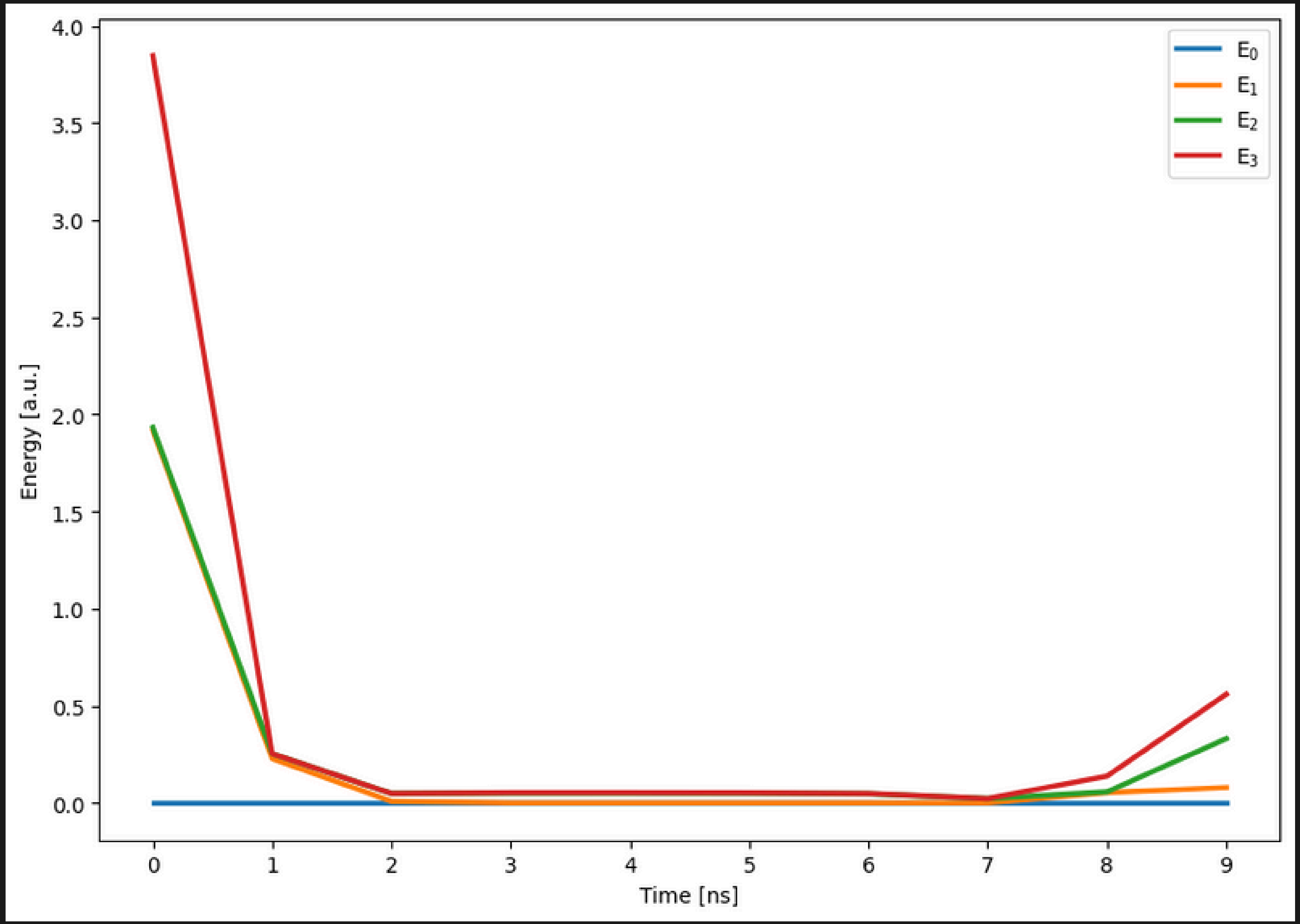


Fig 2: Energy levels of the states with an exponential Schedule

Fluxes

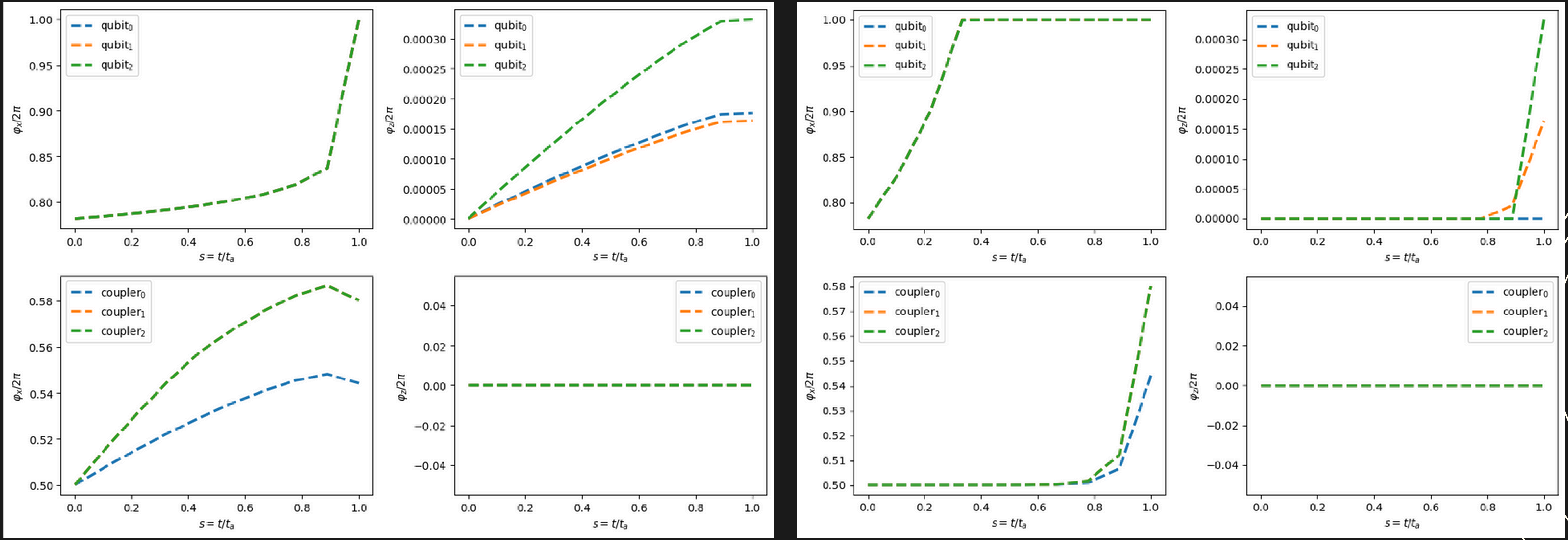


Fig 3-4: Time evolution of calculated fluxes. Left: linear, Right:exponential schedules

FINAL REMARKS

Let’s assume a homogeneous isotropically constant fun time throughout this hackathon

…

Everyone was having fun (we hope…)