Simphony: a python package to simulate point-defect spin dynamics

Péter Boross[1], Győző Egri[1], and András Pályi[1,2]

[1]Qutility @ Faulhorn, Budapest, Hungary

[2]Institute of Physics, Budapest University of Technology and Economics, Budapest, Hungary

Qutility

qutility.io

Abstract

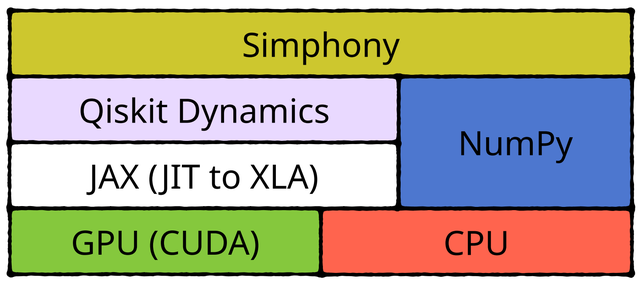
Electronic and nuclear spins around point defects are important building blocks of devices of quantum communication, quantum sensing, and quantum computing. We introduce Simphony, our python package we have developed to simulate quantum dynamics of such point-defect spin ensembles. Our goal with Simphony is to enable resource-efficient simulation with a short, simple, high-level python code. Using Simphony, the user's code and the input data is a short description of the setup (multi-spin configuration, noise model, static and dynamic control parameters). The computed output includes, e.g., the time-dependent wave function or density matrix, or the properties of the logical gate created by the pulse sequence (gate fidelity, process matrix, etc). Furthermore, Simphony supports GPU acceleration, enables the simulation of a collection of interacting point defects, and allows the definition of general point defects, beyond the NV-center paradigm. Simphony enables simple, efficient, and noise-aware design and optimisation of pulse sequences for various applications in quantum communication, sensing, and computing.

What Simphony does

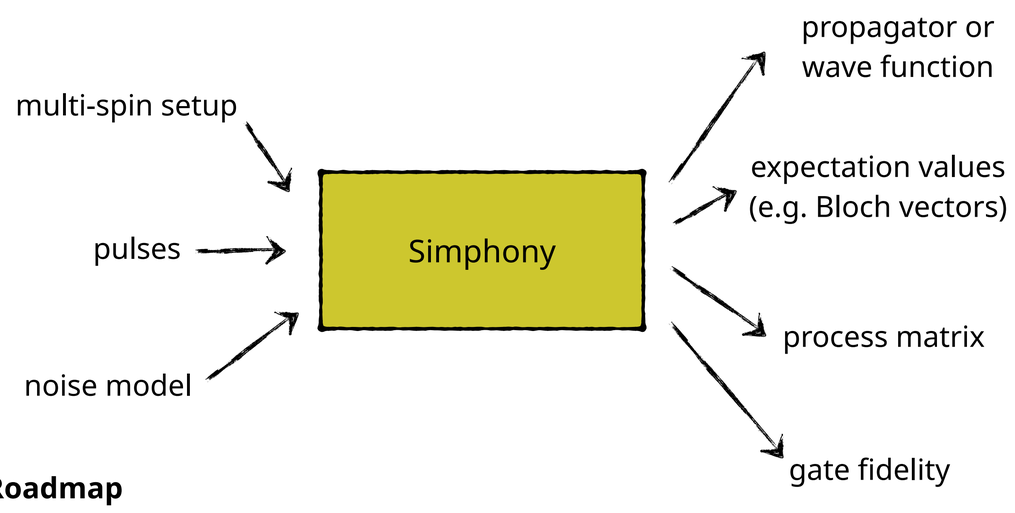
AI4QT project

simulating pulses → native gates → compiling circuits to central-spin register

Dependency diagram



Input-output



Roadmap

- Integration of ^{13}C hyperfine database
- Integration of point-defect Hamiltonians beyond NV centers
- Scalable registers of coupled NV centers
- Lindblad-type noise models
- Non-Markovian system-bath interaction
- Photophysics dynamics: initialisation and readout

Acknowledgement

We thank V. Ivády for useful discussions. This research was supported by the National Research Development and Innovation Office of Hungary within the German-Hungarian EUREKA programme (Project No. 2020-1.2.3-EUREKA-2022-00029)

National Research, Development and Innovation Office Hungary

PROJECT FINANCED FROM THE NRDI FUND

Example: DDRF pulse as a noise-resilient gate

Multi-spin setup

```
spin_e = simphony.Spin(
    dimension = 3,
    name = 'NV-e',
    gyromagnetic_ratio = 28020, # MHz/T
    qubit_subspace = (0, -1),
    zero_field_splitting = 2880 # MHz
)

spin_C = simphony.Spin(
    dimension = 2,
    name = '13C',
    gyromagnetic_ratio = 10.71, # MHz/T
    qubit_subspace = (-1/2, 1/2),
    zero_field_splitting = 0
)

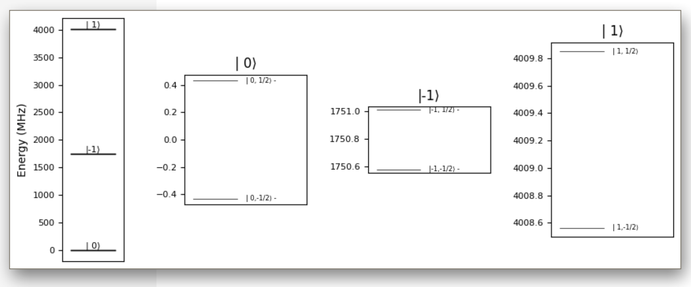
hyperfine = simphony.Interaction(spin_e, spin_C)
hyperfine.zz = 0.213154 # MHz
hyperfine.xz = 0.003 # MHz
hyperfine.zx = 0.003 # MHz

static_field = simphony.StaticField()
static_field.z = 0.0403 # T

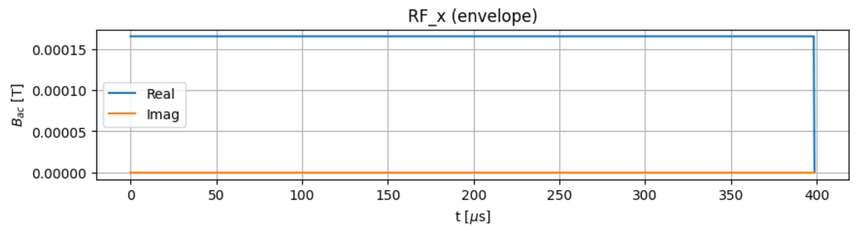
driving_field_MW = simphony.DrivingField(direction = [1, 0, 0], name = 'MW_x')
driving_field_RF = simphony.DrivingField(direction = [1, 0, 0], name = 'RF_x')

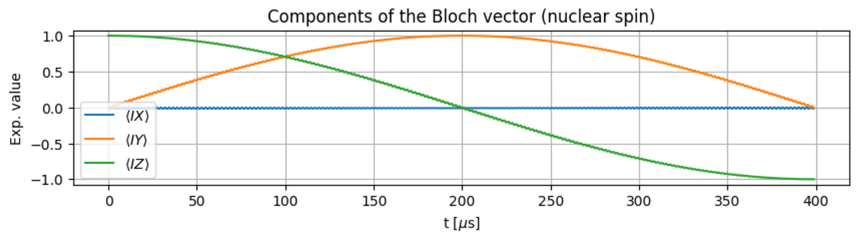
model = simphony.Model()

model.add_spin(spin_e)
```

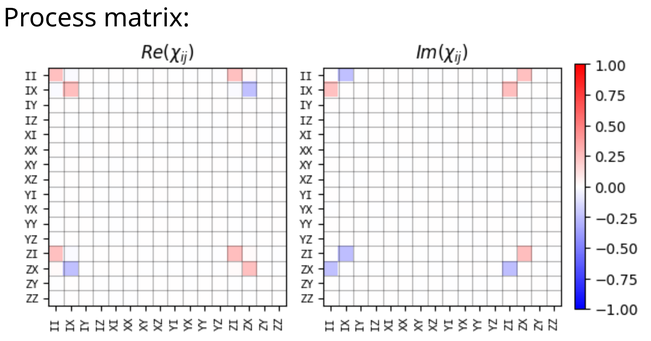


Conventional NMR (noiseless)





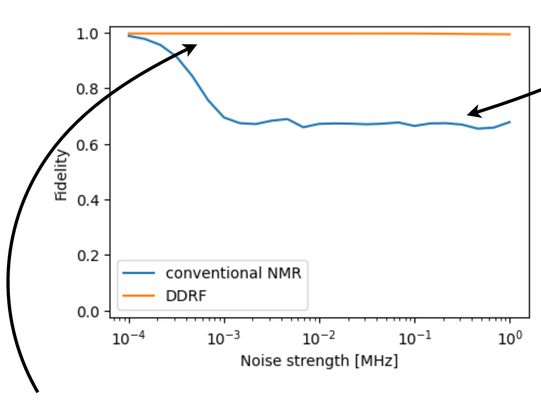
Process matrix:



Ideal gate:
$RX(\pi)$ controlled on $m_S = -1$

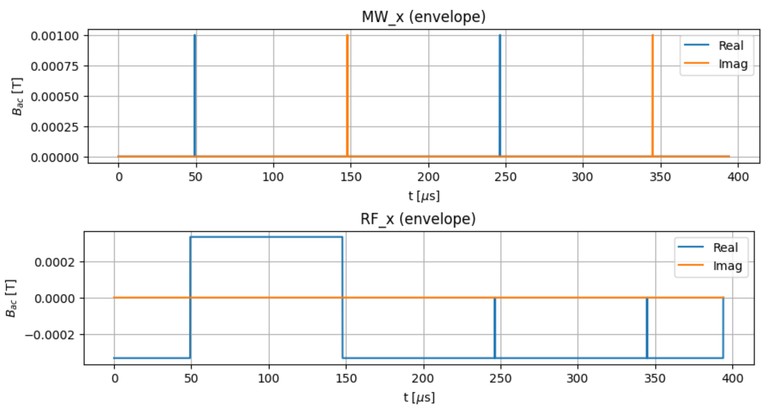
Average gate fidelity:
99.87%

Conventional NMR: noise-induced fidelity reduction



noise model:
quasistatic Larmor-frequency fluctuations

DDRF pulse is robust against noise



Fidelity:
99.34%
(1 MHz noise strength)

