QuanEstimation.jl - Numerical Quantum Parameter Estimation with Julia

Huai-Ming Yu

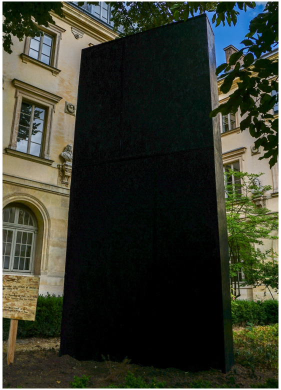
National Precise Gravity Measurement Facility,
School of Physics,
Huazhong University of Science and Technology

HuaimingYuuu@gmail.com

December 8, 2022

Classical Measurement and Standard Quantum Limit

- Standard quantum limit (shot-noise limit) $\sim 1/\sqrt{N}$



The Monolith in *2001: A Space Odyssey*

Pic. by Amandine Brige

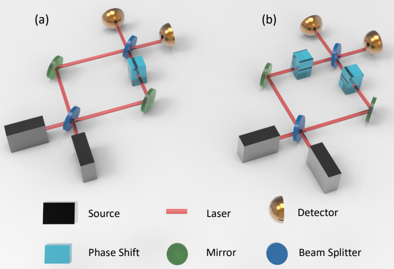
Central limit theorem

When N is large, the average of N independent and identically distributed (i.i.d.) random variables with finite variance σ^2 converges to a normal distribution with variance σ^2/N.

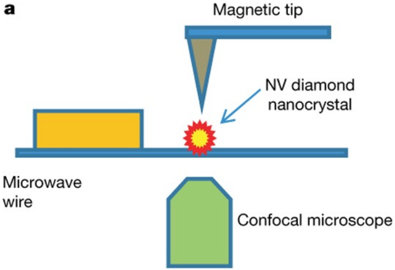
The classical measurement precision is bounded by the central limit theorem.

Towards Enhanced Precision Beyond The Standard Quantum Limit

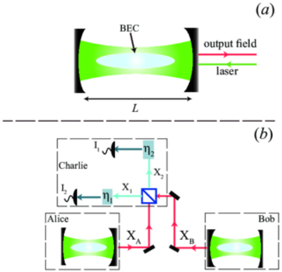
- Quantum metrology: Heisenberg limit



Phase estimation in the Mach-Zehnder interferometer[1]



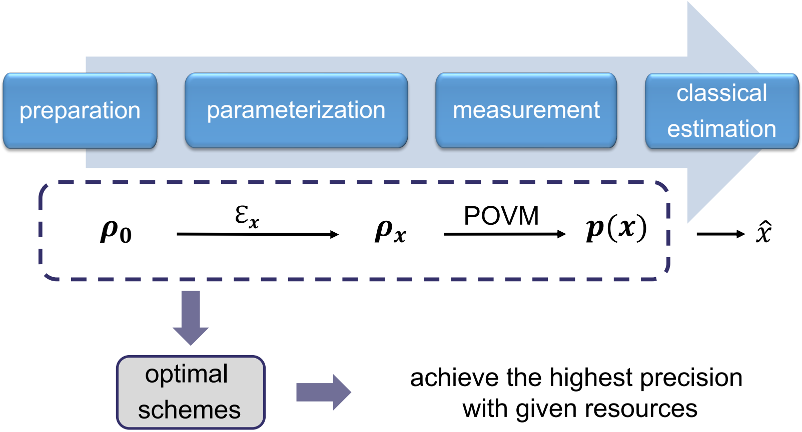
Magnetometry with NV center[2]



Trapped ultracold atoms inside the cavity[3]

[1] C. M. Caves, Phys. Rev. D **23**, 1693-1708 (1981); J. Liu et al., J. Phys. A: Math. Theor. **53**, 023001 (2020).
[2] G. Balasubramanian et al., Nature **455**, 648–651 (2008).
[3] M. Eghbali-Arani et al., J. Opt. Soc. Am. B 32, 798-804 (2015).

Quantum Parameter Estimation



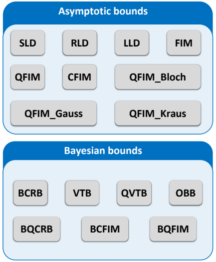
A typical scheme in quantum parameter estimation [4]

[4] J. Liu et al., J. Phys. A: Math. Theor. **53**, 023001 (2020).

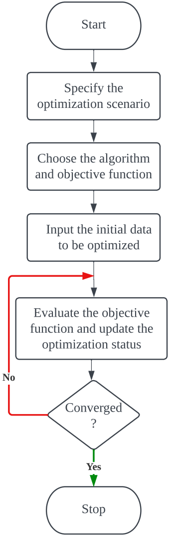
Motivation and Why Julia

What we need:

- Numerical toolbox of the mathematical tools in quantum parameter estimation.
- Optimal quantum metrological scheme design

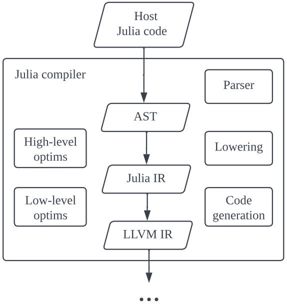


...



Why Julia?

- JIT compilation
- Type stability and multiple dispatch
- Composability

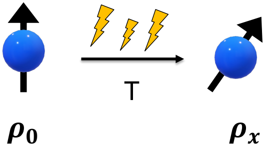


Parameterization

- Evolving the probe state under dynamics governed by master equation for a period of time

$$\partial_t \rho = \mathcal{L}\rho$$

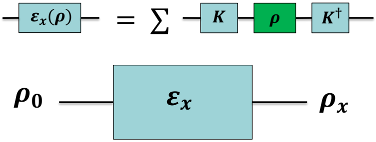
$\mathcal{L}$: superoperator.



- Sending the probe state through quantum channels

$$\rho(\mathbf{x}) = \sum_i K_i(\mathbf{x})\rho_0 K_i^\dagger(\mathbf{x})$$

K_i : Kraus operator.



Master Equation

For an open system, the dynamics is governed by

$$\partial_t \rho = \mathcal{L}\rho = -i\left[H_{\mathbf{x}}, \rho\right] + \sum_i \gamma_i \left(\Gamma_i \rho \Gamma_i^\dagger - \frac{1}{2}\left\{\rho, \Gamma_i^\dagger \Gamma_i\right\}\right)$$

- $H_{\mathbf{x}}$: Hamiltonian encoded with parameter(s) $\mathbf{x}$
  $H_{\mathbf{x}} = H_0(\mathbf{x}) + \sum_k u_k(t) H_k$
- Γ_i and γ_i : the ith decay operator and decay rate
- *expm* : matrix exponential method solver
- *ode* : ODE solvers from DifferentialEquation.jl

```julia
# `tspan` : time interval of the evolution
# `rho0` : initial state
# `H0` : free Hamiltonian
# `dH` : derivatives of the Hamiltonian
#        with respect to the paramters
# `decay` : decay operators & decay rates
# `Hc` : control Hamiltonian
# `ctrl` : control coefficients

# solve the dynamics wit h matrix exponatial
rho, drho = QuanEstimation.expm(
            tspan, rho0, H0, dH, decay, Hc, ctrl
            )

# solve the dynamics with ODE solver
rho, drho = QuanEstimation.ode(
            tspan, rho0, H0, dH, decay, Hc, ctrl
            )
```

Example for Dynamics

```
using QuanEstimation

# initial state
rho0 = 0.5*ones(2, 2)
# free Hamiltonian
omega = 1.0
sz = [1. 0.0im; 0. -1.]
H0 = 0.5*omega*sz
# derivative of the free Hamiltonian on omega
dH = [0.5*sz]
# time length for the evolution
tspan = range(0., 10., length=2500)
# dynamics
rho, drho = QuanEstimation.expm(tspan, rho0, H0, dH)
```

- Hamiltonian:

$$H_0 = \frac{1}{2}\omega\sigma_z$$

ω : paramter to be estimated

- Probe state:

$$\rho_0 = |+\rangle\langle+|$$

- Dynamics: $\gamma = 0$,

$$\partial_t \rho = -i[H_0, \rho]$$

Kraus Operator

Kraus operators $\rho(\mathbf{x}) = \sum_i K_i(\mathbf{x})\rho_0 K_i^\dagger(\mathbf{x})$

```
# define the channel with Kraus operators
channel = QuanEstimation.Kraus(rho0, K, dK)
# solve the parameterization results
rho, drho = QuanEstimation.evolve(channel)
```

- Amplititude damping channel

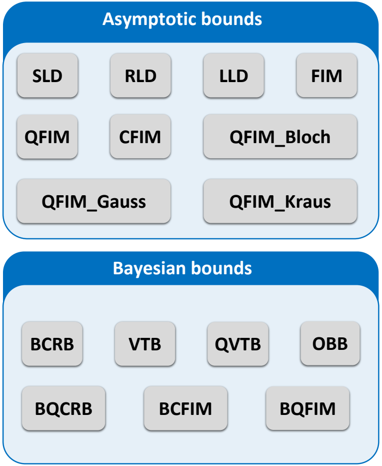
$$K_1 = \begin{pmatrix} 1 & 0 \\ 0 & \sqrt{1-\gamma} \end{pmatrix}, \quad K_2 = \begin{pmatrix} 0 & \sqrt{\gamma} \\ 0 & 0 \end{pmatrix}$$

- Probe state $\rho = |+\rangle\langle+|$

```
using QuanEstimation

# initial state
rho0 = 0.5*ones(2, 2)
# Kraus operators for the amplitude damping channel
gamma = 0.1
K1 = [1. 0.; 0. sqrt(1-gamma)]
K2 = [0. sqrt(gamma); 0. 0.]
K = [K1, K2]
# derivatives of Kraus operators on gamma
dK1 = [1. 0.; 0. -0.5/sqrt(1-gamma)]
dK2 = [0. 0.5/sqrt(gamma); 0. 0.]
dK = [[dK1], [dK2]]
# parameterization process
Kraus = QuanEstimation.Kraus(rho0, K, dK)
rho, drho = QuanEstimation.evolve(Kraus)
```

Quantum Metrological Tools

- Asymptotic Bounds
  - Quantum Cramér-Rao bounds
  - Holevo Cramér-Rao bound
  - Nagaoka-Hayashi bound
- Bayesian Bounds
  - Bayesian type of Cramér-Rao bounds
  - Ziv-Zakai bound
- ...



Quantum Cramér-Rao Bounds

- The most renown metrological tools in quantum parameter estimation

Quantum Cramér-Rao Bound

Let $\rho = \rho(\mathbf{x})$ be a parameterized density matrix and $\{\Pi_y\}$ a set of positive operator-valued measure (POVM), then the covariance matrix $\mathrm{cov}(\hat{\mathbf{x}}, \{\Pi_y\}) := \sum_y \mathrm{Tr}(\rho\Pi_y)(\hat{\mathbf{x}} - \mathbf{x})(\hat{\mathbf{x}} - \mathbf{x})^{\mathrm{T}}$ for the unknown parameters $\mathbf{x} = (x_0, x_1, \ldots)^{\mathrm{T}}$ and the corresponding unbiased estimators $\hat{\mathbf{x}} = (\hat{x}_0, \hat{x}_1, \ldots)^{\mathrm{T}}$ satisfies the following inequalities [5]

$$\mathrm{cov}\left(\hat{\mathbf{x}}, \{\Pi_y\}\right) \geq \frac{1}{n}\mathcal{I}^{-1}\left(\{\Pi_y\}\right) \geq \frac{1}{n}\mathcal{F}^{-1}$$

where n is the repetition of the experiment, $\mathcal{I}$ is the classical Fisher information matrix (CFIM) and $\mathcal{F}$ is the quantum Fisher information matrix (QFIM).

[5] C. W. Helstrom, *Quantum Detection and Estimation Theory* (1976); A. S. Holevo, *Probabilistic and Statistical Aspects of Quantum Theory* (1982)

Quantum Cramér-Rao Bounds

- Classical Fisher information matrix (CFIM)

$$\mathcal{I}_{ab} = \sum_y \frac{1}{p(y|\mathbf{x})}[\partial_a p(y|\mathbf{x})][\partial_b p(y|\mathbf{x})], \qquad p(y|\mathbf{x}) = \mathrm{Tr}(\rho \Pi_y)$$

- Symmetric logarithmic derivative (SLD) operator

$$\partial_a \rho = \frac{1}{2}(\rho L_a + L_a \rho)$$

- Quantum Fisher information matrix (QFIM)

$$\mathcal{F}_{ab} = \frac{1}{2}\mathrm{Tr}(\rho\{L_a, L_b\})$$

Quantum Cramér-Rao Bounds

```
# `rho` & `drho`: density matrix and
#                 its derivative.
# `M`: measurements, a set of POVM.
#      By default a symmetric
#      informationally complete POVM
#      (SIC-POVM).
CFIM(rho, drho, M; eps=1e-8)

# `rep` can be "original" or "eigen"
SLD(rho, drho; rep="original", eps=1e-8)
RLD(rho, drho; rep="original", eps=1e-8)
LLD(rho, drho; rep="original", eps=1e-8)

# `LDtype` cam be :SLD or :RLD or :LLD
QFIM(rho, drho; LDtype=:SLD,
exportLD=false,  eps=1e-8)

QFIM_Kraus(rho, K, dK;
LDtype=:SLD,exportLD=false, eps=1e-8)
```

Notes:

- SIC-POVM:
  A set of POVM $\{\frac{1}{d}|\phi_j\rangle\langle\phi_j|\}_{j=1}^{d^2}$ satisfying

$$|\langle\phi_j|\phi_k\rangle|^2 = \frac{d\delta_{jk}+1}{d+1}$$

- RLD and LLD:

$$\partial_a\rho = \rho\mathcal{R}_a\,, \quad \partial_a\rho = \mathcal{R}_a^\dagger\rho$$

Holevo Cramér-Rao Bound and Nagaoka-Hayashi Bound

- Holevo Cramér-Rao Bound (HCRB)

$$\min_{\mathbf{X},V} \mathrm{Tr}(WV),$$

$$\text{subject to} \begin{cases} \begin{pmatrix} V & \Lambda^{\mathrm{T}}R^{\dagger} \\ R\Lambda & \mathbb{1} \end{pmatrix} \geq 0 \\ \sum_i [\Lambda]_{ai}\mathrm{Tr}(\lambda_i\partial_b\rho) = \delta_{ab} \end{cases}$$

$$X_i = \sum_j [\Lambda]_{ij}\lambda_j, Z = \Lambda^{\mathrm{T}}R^{\dagger}R\Lambda.$$

```
HCRB(rho, drho, W; eps=1e-8)
```

- Nagaoka-Hayashi Bound (NHB)

$$\min_{\mathbf{X},\mathcal{Q}} \mathrm{Tr}\left((W\otimes\rho)\mathcal{Q}\right),$$

$$\text{subject to} \begin{cases} \begin{pmatrix} \mathcal{Q} & \mathbf{X}^{\mathrm{T}} \\ \mathbf{X} & \mathbb{1} \end{pmatrix} \geq 0 \\ \mathrm{Tr}(\rho X_a) = 0,\ \forall a \\ \mathrm{Tr}(X_a\partial_b\rho) = \delta_{ab},\ \forall a, b \end{cases}$$

```
NHB(rho, drho, W; eps=1e-8)
```

QFIM in Specific Scenario

- QFIM in the Bloch representation

$$\mathcal{F}_{ab} = (\partial_b \mathbf{r})^{\mathrm{T}} \left(\frac{d}{2(d-1)} G - \mathbf{r}\,\mathbf{r}^{\mathrm{T}}\right)^{-1} \partial_a \mathbf{r}$$

$$\rho = \frac{1}{d}\left(\mathbb{1} + \sqrt{\frac{d(d-1)}{2}}\mathbf{r}\cdot\lambda\right), G_{ij} = \frac{1}{2}\mathrm{Tr}(\rho\{\lambda_i, \lambda_j\})$$

```
# `r` and `dr`:
#   the Bloch vector and its derivatives
QFIM_Bloch(r, dr; eps=1e-8)
```

- QFIM for the Gaussian states

$$\mathcal{F}_{ab} = \mathrm{Tr}\left(G_a \partial_b D\right) + \left(\partial_a \langle \mathbf{R} \rangle^{\mathrm{T}}\right) D^{-1} \partial_b \langle \mathbf{R} \rangle$$

$$\mathbf{R} = (q_1, p_1, q_2, p_2, \dots), D_{ij} = \frac{1}{2}\langle\{[\mathbf{R}]_i, [\mathbf{R}]_j\}\rangle$$

$$q_i = \frac{1}{\sqrt{2}}(a_i + a_i^{\dagger}), p_i = \frac{1}{i\sqrt{2}}(a_i - a_i^{\dagger}), \langle \mathbf{R} \rangle = \mathrm{Tr}(\rho \mathbf{R})$$

```
# `R`, `dR`:
#  frist-order moment and its derivatives
# `D`, `dD`:
#  second-order moment and its derivatives
QFIM_Gauss(R, dR, D, dD)
```

Examples for Quantum Cramér-Rao Bounds

```julia
using QuanEstimation

# initial state
rho0 = 0.5*ones(2, 2)
# free Hamiltonian
omega = 1.0
sx = [0. 1.; 1. 0.0im]
sy = [0. -im; im 0.]
sz = [1. 0.0im; 0. -1.]
H0 = 0.5*omega*sz
# derivative of the free Hamiltonian on omega
dH = [0.5*sz]
# dissipation
sp = [0. 1.; 0. 0.0im]
sm = [0. 0.; 1. 0.0im]
decay = [[sp, 0.0], [sm, 0.1]]
# time length for the evolution
tspan = range(0., 50., length=2000)
# dynamics
rho, drho = QuanEstimation.expm(tspan, rho0, H0, dH,
decay)
# calculation of the CFI and QFI
I, F = Float64[], Float64[]
for ti in 2:length(tspan)
    # CFI
    append!(I, QuanEstimation.CFIM(rho[ti], drho[ti]))
    # QFI
    append!(F, QuanEstimation.QFIM(rho[ti], drho[ti]))
end
```

- Example for QFIM and CFIM

Dynamics: $H_0 = \frac{1}{2}\omega\sigma_3$,

$$\partial_t \rho = -i\left[H_0, \rho\right] + \gamma_+ \left(\sigma_+ \rho \sigma_- - \frac{1}{2}\left\{\rho, \sigma_-\sigma_+\right\}\right) + \gamma_- \left(\sigma_- \rho \sigma_+ - \frac{1}{2}\left\{\rho, \sigma_+\sigma_-\right\}\right)$$

Probe state: $\rho_0 = |+\rangle\langle+|$

Bayesian Cramér-Rao Bounds

- Three types of BCRB and BQCRB

$$\mathrm{cov}(\hat{\mathbf{x}}, \{\Pi_y\}) \geq \int p(\mathbf{x}) \left(B\mathcal{I}^{-1}B + \mathbf{b}\mathbf{b}^{\mathrm{T}}\right) \mathrm{d}\mathbf{x}, \quad \mathrm{cov}(\hat{\mathbf{x}}, \{\Pi_y\}) \geq \int p(\mathbf{x}) \left(B\mathcal{F}^{-1}B + \mathbf{b}\mathbf{b}^{\mathrm{T}}\right) \mathrm{d}\mathbf{x}$$

$$\mathrm{cov}(\hat{\mathbf{x}}, \{\Pi_y\}) \geq \mathcal{B}\mathcal{I}_{\mathrm{Bayes}}^{-1}\mathcal{B} + \int p(\mathbf{x})\mathbf{b}\mathbf{b}^{\mathrm{T}}\mathrm{d}\mathbf{x}, \quad \mathrm{cov}(\hat{\mathbf{x}}, \{\Pi_y\}) \geq \mathcal{B}\mathcal{F}_{\mathrm{Bayes}}^{-1}\mathcal{B} + \int p(\mathbf{x})\mathbf{b}\mathbf{b}^{\mathrm{T}}\mathrm{d}\mathbf{x}$$

$$\mathrm{cov}(\hat{\mathbf{x}}, \{\Pi_y\}) \geq \int p(\mathbf{x})\mathcal{G}\left(\mathcal{I}_p + \mathcal{I}\right)^{-1}\mathcal{G}^{\mathrm{T}}\mathrm{d}\mathbf{x}, \quad \mathrm{cov}(\hat{\mathbf{x}}, \{\Pi_y\}) \geq \int p(\mathbf{x})\mathcal{G}\left(\mathcal{I}_p + \mathcal{F}\right)^{-1}\mathcal{G}^{\mathrm{T}}\mathrm{d}\mathbf{x}$$

$$\mathcal{I}_{\mathrm{Bayes}} = \int p(\mathbf{x})\mathcal{I}\mathrm{d}\mathbf{x}, \ \mathcal{F}_{\mathrm{Bayes}} = \int p(\mathbf{x})\mathcal{F}\mathrm{d}\mathbf{x}$$

$$[\mathcal{I}_p]_{ab} = [\partial_a \ln p(\mathbf{x})][\partial_b \ln p(\mathbf{x})]$$

$$[\mathcal{G}]_{ab} = [\partial_b \ln p(\mathbf{x})][\mathbf{b}_a] + B_{aa}\delta_{ab}$$

```julia
# `btype` can be: 1, 2, 3.
# Bayesian Cramér-Rao Bound
BCRB(x, p, dp, rho, drho;
     M=M, b=b, db=db, btype=1, eps=1e-8)
# Bayesian Quantum Cramér-Rao Bound
BQCRB(x, p, dp, rho, drho;
      b=b, db=db, btype=1, LDtype=:SLD, eps=1e-8)
```

Bayesian Bounds

- Quantum Ziv-Zakai bound (QZZB)

$$\mathrm{var}(\hat{x},\{\Pi_y\}) \geq \frac{1}{2}\int_0^{\infty} \mathrm{d}\tau\tau\mathcal{V}\int_{-\infty}^{\infty} \mathrm{d}x \min\{p(x),p(x+\tau)\}\left(1-\frac{1}{2}||\rho(x)-\rho(x+\tau)||\right)$$

- Van Trees bound (VTB) and Tsang-Wiseman-Caves bound (QVTB)

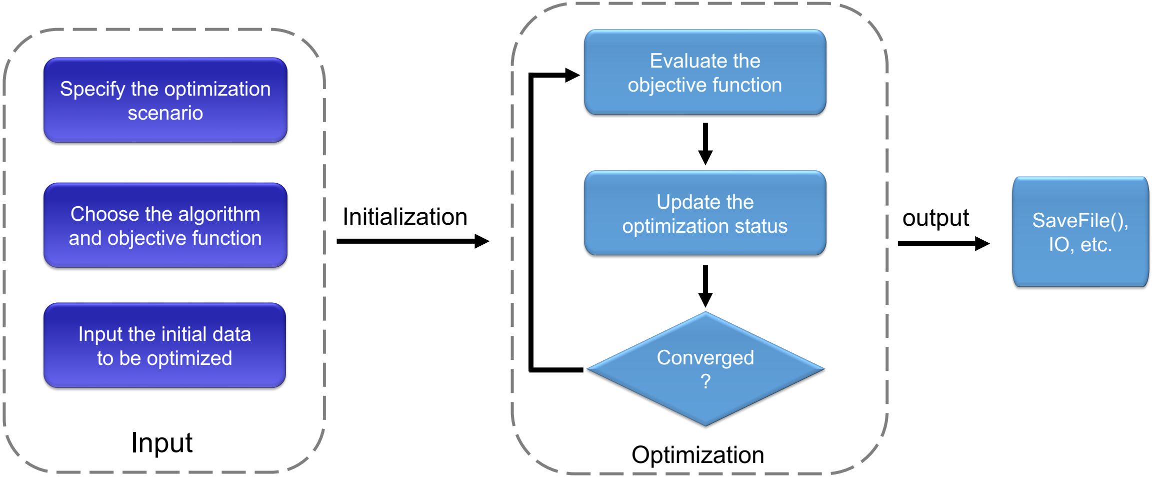
$$\mathrm{cov}(\hat{\mathbf{x}},\{\Pi_y\}) \geq (\mathcal{I}_{\mathrm{prior}}+\mathcal{I}_{\mathrm{Bayes}})^{-1}$$
$$\mathrm{cov}(\hat{\mathbf{x}},\{\Pi_y\}) \geq (\mathcal{I}_{\mathrm{prior}}+\mathcal{F}_{\mathrm{Bayes}})^{-1}$$

- Bayesian cost bound (BCB)

$$\bar{C} \geq \int p(\mathbf{x})(\mathbf{x}^{\mathrm{T}}W\mathbf{x})\mathrm{d}\mathbf{x} - \sum_{ab} W_{ab}\mathrm{Tr}(\bar{\rho}\bar{L}_a\bar{L}_b)$$

```
BCB(x, p, rho; W=W, eps=1e-8)

OBB(x, p, dp, rho, drho, d2rho, LDtype=:SLD, eps=1e-8)

VTB(x, p, dp, rho, drho; M=M,eps=1e-8)

QVTB(x, p, dp, rho, drho; LDtype=:SLD, eps=1e-8)

QZZB(x, p, rho; eps=1e-8)
```

Optimal Scheme Design



Optimization scenario and Algorithms

OptScenario	Supported Algortithms:
▶ Control Optimization $$H = H_0 + \sum_{k=1}^{K} u_k(t) H_k$$	Gradient ascent pulse engineering (GRAPE), auto-GRAPE, Particle swarm optimization (PSO), Differential evolution (DE), Deep Deterministic Policy Gradients (DDPG)
▶ State Optimization $$\|\psi\rangle = \sum_i c_i \|i\rangle$$	Gradient descent with automatic differentiation (AD), Nelder-Mead (NM), Reverse iterative (RI), PSO, DE, DDPG
▶ Measurement Optimization mtype = :Projection, :LC, :Rotation	PSO, DE, AD
▶ Comprehensive Optimization SM, SC, CM, SCM	PSO, DE, AD

Examples for Control Optimization

```
# initial guessed control coefficients
cnum = length(tspan)-1
ctrl = [zeros(cnum) for _ in 1:length(Hc)]
ctrl_bound = [-2., 2.]
# set the optimization type
opt = QuanEstimation.ControlOpt(ctrl=ctrl,
ctrl_bound=ctrl_bound, seed=1234)
# control algorithm: auto-GRAPE
alg = QuanEstimation.autoGRAPE(Adam=true,
max_episode=300, epsilon=0.01, beta1=0.90,
beta2=0.99)
# objective function: QFI
obj = QuanEstimation.QFIM_obj()
# input the dynamics data
dynamics = QuanEstimation.Lindblad(opt, tspan,
rho0, H0, dH, Hc, decay, dyn_method=:Expm)
# run the control optimization problem
QuanEstimation.run(opt, alg, obj, dynamics;
savefile=false)
```

Dynamics: $H_0 = \frac{1}{2}\omega\sigma_3$,

$$\partial_t \rho = -i\left[H_0, \rho\right] + \gamma_+\left(\sigma_+\rho\sigma_- - \frac{1}{2}\left\{\rho, \sigma_-\sigma_+\right\}\right) + \gamma_-\left(\sigma_-\rho\sigma_+ - \frac{1}{2}\left\{\rho, \sigma_+\sigma_-\right\}\right)$$

Probe state:

$$\rho_0 = |+\rangle\langle+|$$

Initial control coefficients:

$$u_{k,0}(t) \equiv 0$$

Docs:

`https://quanestimation.github.io/QuanEstimation/`

GitHub repositories:

- QuanEstimation/QuanEstimation:
  `https://github.com/QuanEstimation/QuanEstimation`
- QuanEstimation/QuanEstimation.jl :
  `https://github.com/QuanEstimation/QuanEstimation.jl`

Related works:

- M. Zhang et al., Phys. Rev. Research **4**, 043057 (2022).
- H.-M. Yu et al., in preparation.

Thank You!