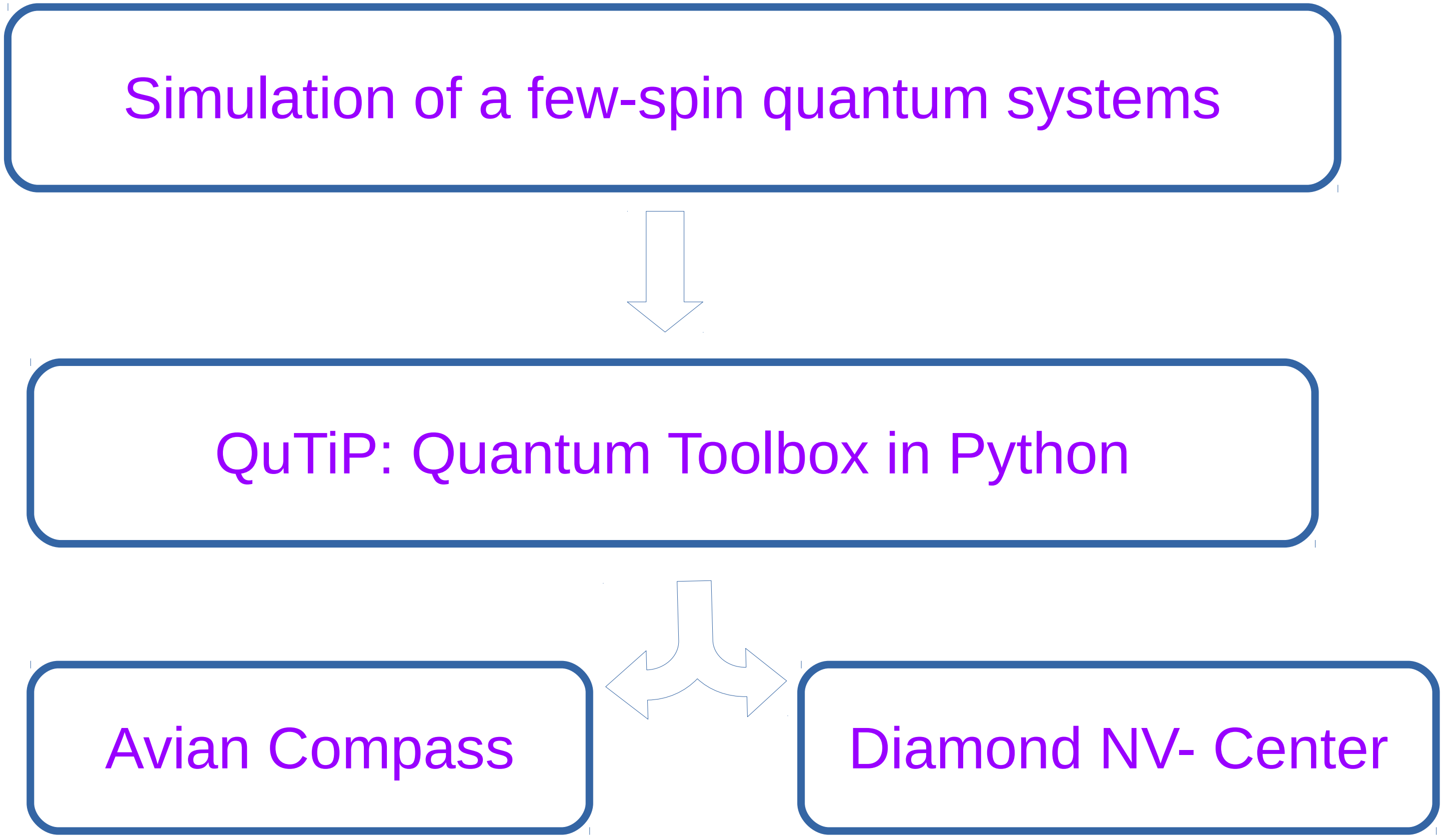
Simulation of Spin Systems: QuTiP Demonstration

Swaroop Ganguly **Vishvendra Poonia**

HRI

29th Feb 2016


Simulation of a few-spin quantum systems
QuTiP: Quantum Toolbox in Python
Avian Compass
Diamond NV- Center

QuTiP: Quantum Toolbox in Python

- Based on Python + Scipy framework
- Designed to be a general framework for solving quantum mechanics problems viz. quantum systems with few levels and harmonic oscillators
- Contains large number of functions for performing various quantum operations
- Well tested and open source
- Easy to generate complex Hamiltonians, Liouvillians, wave functions, trace/pratial trace etc
- ODE Solver for Schrodinger equation and master equations


QuTiP
bloch_redfield
brmesolve
bloch_redfield_solve
bloch_redfield_tensor
correlation
correlation_2op_1t
correlation_2op_2t
correlation_3op_1t
correlation_3op_2t
coherence_function_g1
coherence_function_g2
spectrum_correlation_fft
correlation_ss
correlation_4op_1t
correlation_4op_2t
spectrum_ss
spectrum_pi
eseries
esseries
esval
esspec
estidy
essolve
essolve
ode2es
floquet
floquet_modes
floquet_modes_t
floquet_modes_table
floquet_modes_t_lookup
floquet_states
floquet_states_t
floquet_wavefunction
floquet_wavefunction_t
floquet_state_decomposition
fsesolve
floquet_master_equation_rates
floquet_collapse_operators
floquet_master_equation_tensor
floquet_master_equation_steadystate
floquet_basis_transform
floquet_markov_mesolve
fmmesolve
mcsolve
mcsolve
mesolve
mesolve
odesolve
propagator
propagator
propagator_steadystate
rhs_generate
rhs_clear
rhs_generate
sesolve
sesolve
solver
Options
Result
SolverConfiguration
steadystate
steadystate
steady
build_preconditioner
stochastic
StochasticSolverOptions:
ssesolve
smesolve
ssepdpsolve
smepdpsolve
d1_psi_homodyne
d2_psi_homodyne
d1_psi_heterodyne
d2_psi_heterodyne
d1_psi_photocurrent
d2_psi_photocurrent
sop_H
sop_G
d1_rho_homodyne
d2_rho_homodyne
d1_rho_heterodyne
d2_rho_heterodyne
d1_rho_photocurrent
d2_rho_photocurrent
settings
load_rc_file
bloch
Bloch
bloch3d
Bloch3d
distributions
Distribution:
WignerDistribution
QDistribution
TwoModeQuadratureCorrelation
HarmonicOscillatorWaveFunction
HarmonicOscillatorProbabilityFunction
orbital
orbital
tomography
qpt_plot
qpt_plot_combined
qpt
visualization
hinton
sphereplot
matrix_histogram
complex_phase_cmap
matrix_histogram_complex
plot_energy_levels
wigner_cmap
energy_level_diagram
plot_wigner
plot_fock_distribution
plot_wigner_fock_distribution
wigner_fock_distribution
plot_expectation_values
plot_spin_distribution_2d
plot_spin_distribution_3d
complex_array_to_rgb
plot_qubism
plot_schmidt
wigner
wigner
qfunc
spin_q_function
spin_wigner
operators
jmat
spin_Jx
spin_Jy
spin_Jz
spin_Jm
spin_Jp
spin_J_set
sigmap
sigmam
sigmax
sigmay
sigmaz
destroy
create
qeye
identity
position
momentum
num
squeeze
squeezing
displace
commutator
qutrit_ops
qdiags
phase
zero_oper
subsystem_apply
subsystem_apply
superop_reps
super_to_choi
choi_to_super
choi_to_kraus
kraus_to_super
choi_to_chi
chi_to_choi
to_choi
to_chi
to_super
to_kraus
superoperator
liouvillian
liouvillian_ref
lindblad_dissipator
operator_to_vector
vector_to_operator
mat2vec
vec2mat
vec2mat_index
mat2vec_index
spost
spre
sprepost
continuous_variables
correlation_matrix
covariance_matrix
correlation_matrix_field
correlation_matrix_quadrature
wigner_covariance_matrix
logarithmic_negativity
random_objects
rand_herm
rand_unitary
rand_ket
rand_dm
rand_kraus_map
rand_super
states
basis
qutrit_basis
coherent
coherent_dm
fock_dm
maximally_mixed_dm
thermal_dm
fock
ket2dm
projection
qstate
ket
bra
state_number_enumerate
state_number_index
state_index_number
state_number_qobj
phase_basis
zero_ket
spin_state
spin_coherent
bell_state
singlet_state
triplet_states
w_state
ghz_state
three_level_atom
three_level_basis
three_level_ops
entropy
entropy_vn
entropy_linear
concurrence
entropy_mutual
entropy_conditional
participation_ratio
entangling_power
metrics
fidelity
process_fidelity
average_gate_fidelity
tracedist
hilbert_dist
bures_dist
bures_angle
expect
expect
variance
partial_transpose
partial_transpose
ptrace
qobj
Qobj
qobj_list_evaluate
dag
ptrace
dims
shape
isket
isbra
isoperket
isoperbra
isoper
issuper
isequal
isherm
tensor
tensor
super_tensor
composite
about
about
fileio
file_data_store
file_data_read
qsave
qload
graph
graph_degree
breadth_first_search
maximum_bipartite_matching
weighted_bipartite_matching
hardware_info
hardware_info
ipynbtools
version_table
HTMLProgressBar
parfor
plot_animation
parfor
parfor
reshuffle
permute
picloud
picloud
simdiag
simdiag
degen
sparse
sp_fro_norm
sp_inf_norm
sp_L2_norm
sp_max_norm
sp_one_norm
sp_reshape
sp_eigs
sp_expm
sp_permute
sp_reverse_permute
sp_bandwidth
sp_profile
testing
run
utilities
n_thermal
linspace_with
clebsch
convert_unit
convert_GHz_to_meV
convert_meV_to_GHz
convert_J_to_meV
convert_meV_to_J
convert_meV_to_mK
convert_mK_to_meV
convert_GHz_to_mK
convert_mK_to_GHz
version
view_methods

QuTiP: Basics

- Quantum states, operators etc. are represented by a data structure called Qobj

- Defining a state $|\psi\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

```
>> psi = Qobj([[1], [0]])
>> print psi
Quantum object: dims = [[2], [1]], shape = [2, 1]
Qobj data =
[[1]
 [0]]
```

- Operations on quantum states $|\psi_{up}\rangle = \begin{bmatrix} 1 \\ 0 \end{bmatrix}; |\psi_{down}\rangle = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$ $\quad |\psi\rangle = \frac{|\psi_{up}\rangle + |\psi_{down}\rangle}{\sqrt{2}}$

```
>> psi_up   = Qobj([[0], [1]])
>> psi_down = Qobj([[1], [0]])
>> psi      = (psi_up + psi_down) / sqrt(2)
>> print psi
Quantum object: dims = [[2], [1]], shape = [2, 1]
Qobj data =
[[ 0.70710678]
 [ 0.70710678]]
```

- Fock state

$$|\psi_{fock}\rangle = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix}$$

$$\rho_{fock} = |\psi_{fock}\rangle \langle \psi_{fock}|$$

- Tensor product

- Partial trace

```
>> print fock(4, 1)
Quantum object: dims = [[4], [1]], shape = [4, 1]
Qobj data =
[[ 0.]
 [ 1.]
 [ 0.]
 [ 0.]]
>> print fock_dm(4, 1)
Quantum object: dims = [[4], [4]], shape = [4, 4]
Qobj data =
[[ 0.  0.  0.  0.]
 [ 0.  1.  0.  0.]
 [ 0.  0.  0.  0.]
 [ 0.  0.  0.  0.]]

>> rho=tensor([fock_dm(2,0),fock_dm(2,1)])
>> print ptrace(rho,0)
Quantum object: dims = [[2], [2]], shape = [2, 2]
Qobj data =
[[ 1.  0.]
 [ 0.  0.]]
>> print ptrace(rho,1)
Quantum object: dims = [[2], [2]], shape = [2, 2]
Qobj data =
[[ 0.  0.]
 [ 0.  1.]]
```

Hamiltonian: Examples

- Two interacting qubit: $H = \sigma_z^{(1)} + \sigma_z^{(2)} + 0.01\sigma_x^{(1)}\sigma_x^{(2)}$

```
>> sz1  = tensor([sigmaz(), qeye()])
>> sz2  = tensor([qeye(), sigmaz()])
>> sxsx = tensor([sigmax(), sigmax()])
>> H = sz1 + sz2 + 0.1 * sxsx
>> print H
Quantum object: dims = [[2, 2], [2, 2]], shape = [4, 4]
Qobj data =
[[ 2.   0.   0.   0.1]
 [ 0.   0.   0.1  0. ]
 [ 0.   0.1  0.   0. ]
 [ 0.1  0.   0.  -2. ]]
```

- Qubit coupled to a harmonic oscillator bath: $H = \omega a^\dagger a + \epsilon\sigma_z + g(a^\dagger\sigma^- + a\sigma^+)$

```
>> N = 10; w = 1; eps = 1; g = 0.01
>> a = tensor([destroy(N), qeye(2)])
>> sz = tensor([qeye(N), sigmaz()])
>> sm = tensor([qeye(N), sigmam()])
>> H = w * a.dag() * a + eps * sz + g * (a.dag() * sm + a * sm.dag())
>> print H
Quantum object: dims = [[10, 2], [10, 2]], shape = [20, 20]
Qobj data =
...
```

Time Evolution: Unitary time evolution

$$H\,|\psi\rangle = i\hbar\frac{\partial}{\partial t}\,|\psi\rangle;\;\; |\psi(0)\rangle = psi_0$$

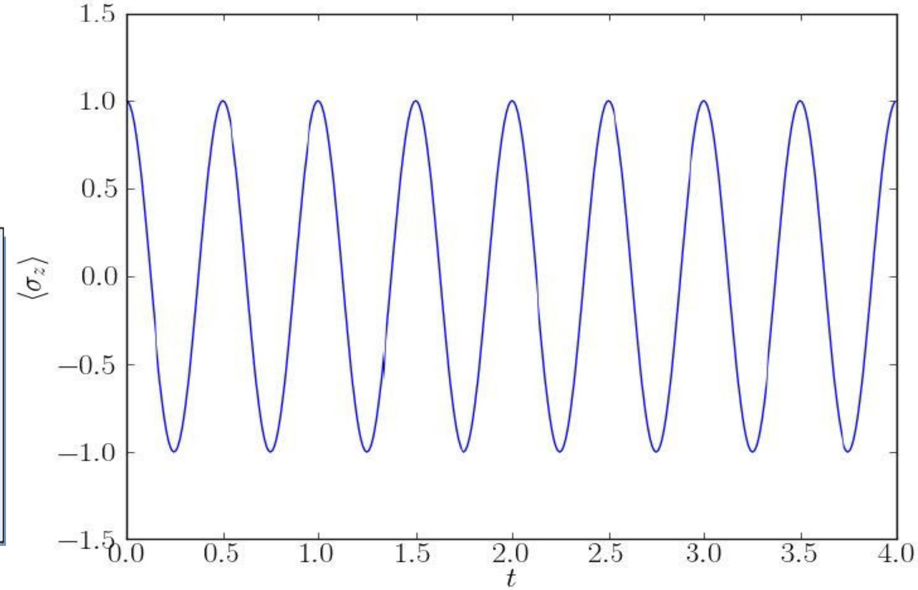
QuTiP: result = ode_solve(H, psi_0, tlist, [ ], expt_op_list)

- H = Hamlitonian
- psi_0 = initial wave function
- tlist = time instants at which result is to be stored
- expt_op_list = expectation value of these operator is stored (at specified time)
- result = state values stored at time instants specified in tlist

$$< A(t) >= \langle\psi(t)|\, A\, |\psi(t)\rangle$$

Example:

```
>> H = 2 * pi * sigmax()
>> psi0 = fock(2, 0)              # |0> state
>> tlist = arange(0.0, 4.0, 0.01)
>> expt_sz = ode_solve(H, psi0, tlist,[], [sigmaz()])
>> plot(tlist, expt_sz[0])
>> show()
```



Time Evolution: Master equation

$$\dot{\rho} = -\frac{i}{\hbar}[H,\rho] + \Gamma \sum_{i=1}^{8} (L_i \rho L_i^\dagger - \frac{1}{2}(L_i^\dagger L_i \rho + \rho L_i^\dagger L_i))$$

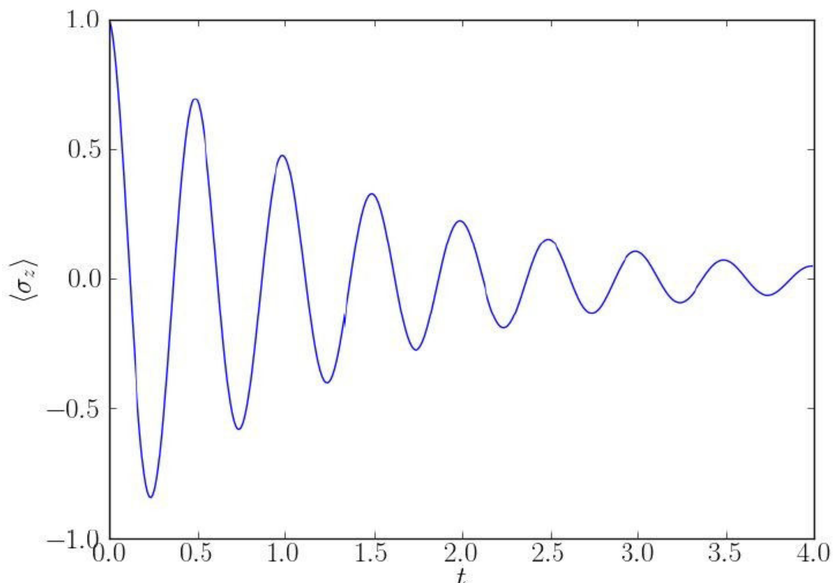
QuTiP: result = ode_solve(H, psi0, tlist, c_op_list, expt_op_list)

- H = Hamlitonian
- psi0 = an initial wave function
- tlist = time instants at which result is to be stored
- c_op_list = list of collapse operators
- expt_op_list = expectation value of these operator is stored (at specified time)

Alternatively: result = mesolve(H, rho0, tlist, c_ops, expt_ops, args={}, options=None)

Example:

```
>> H = 2 * pi * sigmax()
>> psi0 = fock(2, 0)              # |0> state
>> gamma1 = 1.0
>> c_ops = [sqrt(gamma1) * sigmam()] # relaxation
>> expt_ops = [sigmaz()]
>> tlist = arange(0.0, 4.0, 0.01)
>> expt_sz = ode_solve(H, psi0,tlist,c_ops,expt_ops)
>> plot(tlist, expt_sz[0])
```



QuTiP Demos: Inbuilt Examples

Qubit: demos()

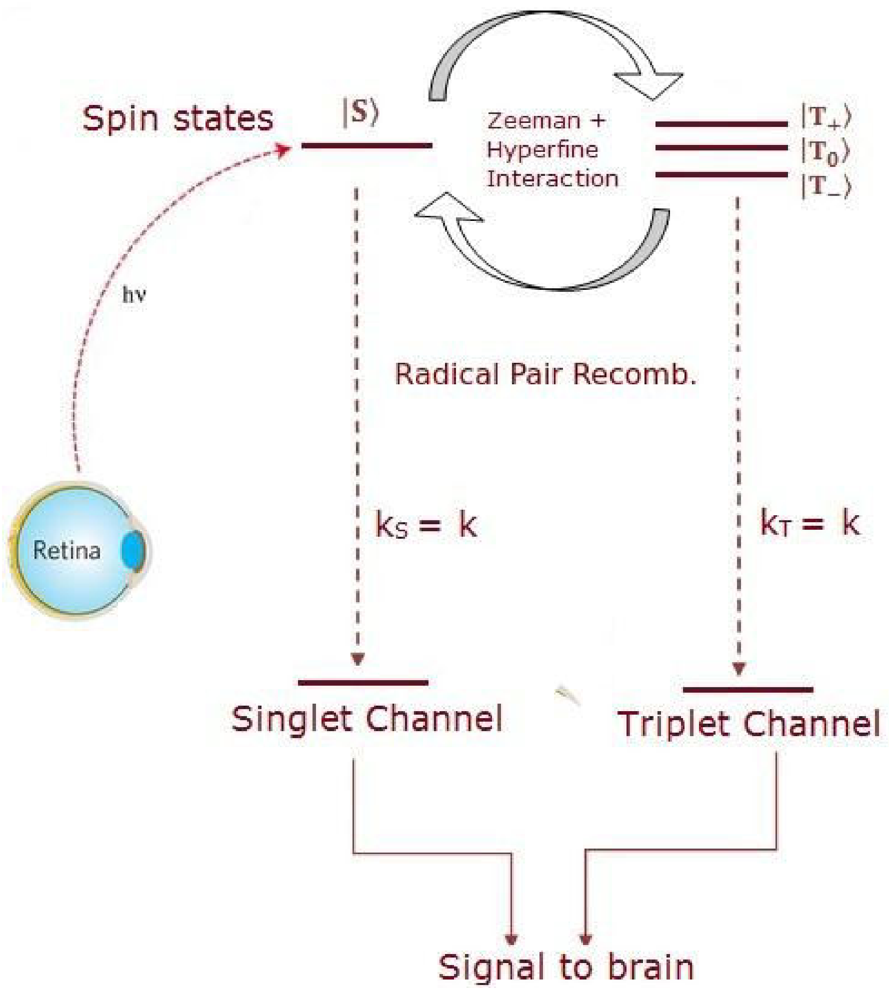
Rabi Oscillations

Bloch Sphere Spin Dynamics

Time Evolution and Dynamics

...... and many more

Avian Compass: RP Spin Dynamics



- Fraction of radical pairs recombining to singlet channel ⟶ Singlet Yield
- Fraction of radical pairs recombining to triplet channel ⟶ Triplet Yield
- Singlet and Triplet yields contain the information about the ambient magnetic field

Gauger Erik M., et al. *PRL,* 106.4, 040503, 2011

The Formalism

Lindblad Master Equation:

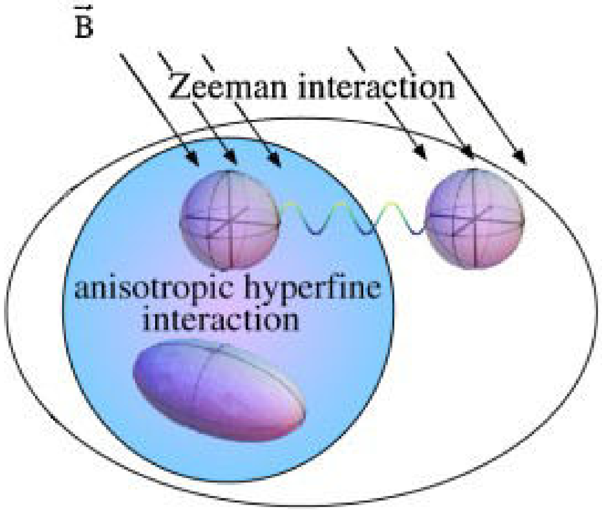
$$\dot{\rho} = \underbrace{-\frac{i}{\hbar}[H,\rho]}_{\text{Unitary evolution}} + \underbrace{k\sum_{i=1}^{8} P_i\rho P_i^\dagger - \frac{1}{2}(P_i^\dagger P_i\rho + \rho P_i^\dagger P_i)}_{\text{Recombination}}.$$

P_i: Recombination to singlet and triplet channels

$$P_1 = |SC\rangle\langle S\uparrow| \qquad P_5 = |SC\rangle\langle S\downarrow|$$
$$P_2 = |TC\rangle\langle T_0\uparrow| \qquad P_6 = |TC\rangle\langle T_0\downarrow|$$
$$P_3 = |TC\rangle\langle T_+\uparrow| \qquad P_7 = |TC\rangle\langle T_+\downarrow|$$
$$P_4 = |TC\rangle\langle T_-\uparrow| \qquad P_8 = |TC\rangle\langle T_-\downarrow|$$

Hamiltonian

$$H = \underbrace{\gamma \mathbf{B} \cdot (\hat{S}_1 + \hat{S}_2)}_{\text{Zeeman}} + \underbrace{\hat{I} \cdot \mathbf{A} \cdot \hat{S}_2}_{\text{Hyperfine}}$$



$$H = H_{nuc} \otimes H_{e1} \otimes H_{e2}$$

$$H_{hyp} = a_x I_x S_x + a_y I_y S_y + a_z I_z S_z$$

$$\mathbf{A} = diag(a_x, a_y, a_z)$$

$$H_{hyp} = a_x(\sigma_x \otimes I \otimes \sigma_x) + a_y(\sigma_y \otimes I \otimes \sigma_y) + a_z(\sigma_z \otimes I \otimes \sigma_z)$$

$$H_{Zeeman} = \gamma(B_x(S_{1x} + S_{2x}) + B_y(S_{1y} + S_{2y}) + B_z(S_{1z} + S_{2z}))$$

$$H_{Zeeman} = \gamma B_x(I \otimes \sigma_x \otimes I + I \otimes I \otimes \sigma_x) + \gamma B_y(I \otimes \sigma_y \otimes I + I \otimes I \otimes \sigma_y)$$
$$+ \gamma B_z(I \otimes \sigma_z \otimes I + I \otimes I \otimes \sigma_z)$$

Density Matrix

- Three spin system (Each spin 1/2): 8 dim Hilbert Space
- This Hilbert space is extended to 10 dims to account for two recombination channels

$$\rho = [\rho_{nuc} \otimes \rho_{elec1} \otimes \rho_{elec2}] + 2\, recomb.\, channel\ states$$

- The matrix elements corresponding to 9th and 10th dimension give the singlet and triplet yields

Initial State

- Initially, the RP is in singlet state: $|s\rangle = \frac{1}{\sqrt{2}}(|\uparrow\downarrow\rangle - |\downarrow\uparrow\rangle)$
- Initial state: $\rho(0) = \frac{1}{2} I \otimes (|s\rangle \otimes \langle s|)$

Projection Operator

Define: P_i Projection from singlet and triplet states to recombination channels

$$P_1 = |SC\rangle\langle S\uparrow|$$
$$P_2 = |TC\rangle\langle T_0\uparrow|$$
$$P_3 = |TC\rangle\langle T_+\uparrow|$$
$$P_4 = |TC\rangle\langle T_-\uparrow|$$

$$P_5 = |SC\rangle\langle S\downarrow|$$
$$P_6 = |TC\rangle\langle T_0\downarrow|$$
$$P_7 = |TC\rangle\langle T_+\downarrow|$$
$$P_8 = |TC\rangle\langle T_-\downarrow|$$

Projection Operators: Explicit Matrix form

$$P_1 = \begin{pmatrix} 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&1&-1&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0 \end{pmatrix} \quad P_2 = \begin{pmatrix} 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&-1&0&0&0\\ 0&0&0&0&0&0&0&0&0&0 \end{pmatrix}$$

$$P_3 = \begin{pmatrix} 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&1&1&0&0&0&0&0&0&0 \end{pmatrix} \quad P_4 = \begin{pmatrix} 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&1&1&0&0&0 \end{pmatrix}$$

$$P_5 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad P_6 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

$$P_7 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \end{pmatrix} \quad P_8 = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \end{pmatrix}$$

State Evolution

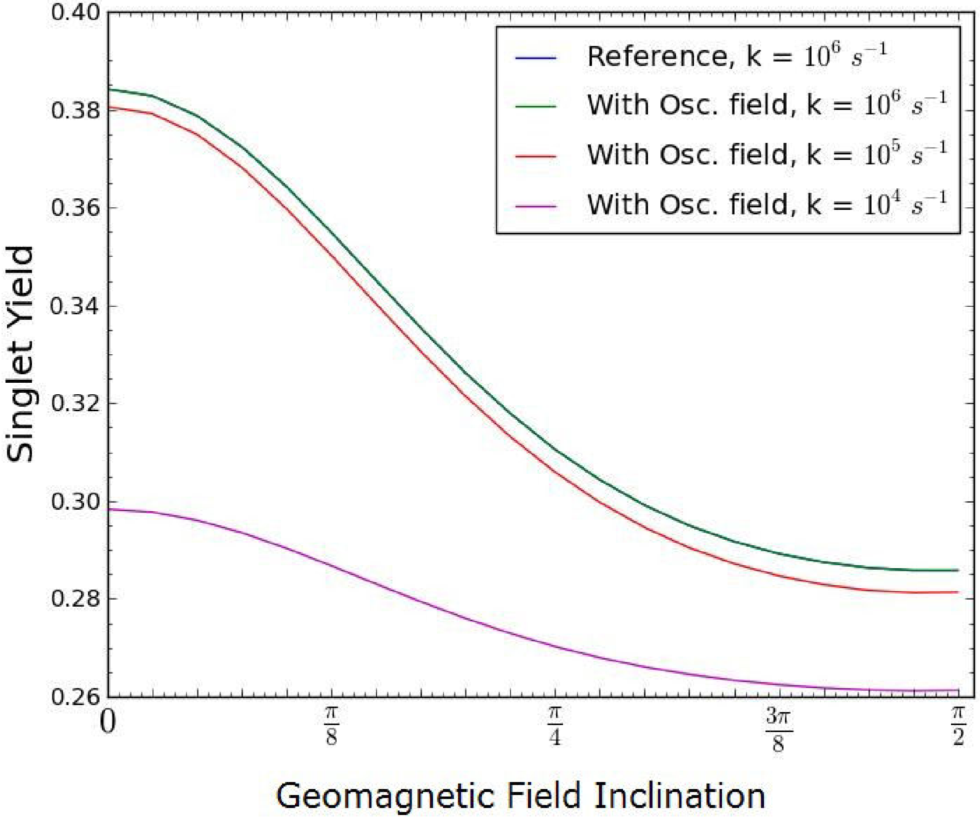
$$\dot{\rho} = -\frac{i}{\hbar}[H, \rho] + k\sum_{i=1}^{8} P_i \rho P_i^\dagger - \frac{1}{2}(P_i^\dagger P_i \rho + \rho P_i^\dagger P_i).$$

- After defining Hamiltonian, initial state and, projection operators this master equation is solved using mesolve function.

```
result = mesolve(H, rho0, tlist, c_ops, expt_ops, args={}, options=None)
```

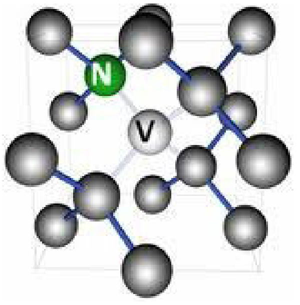
- result stores the density matrix at each time instant
- The 9th diagonal element is the singlet yield and 10th diagonal element is the triplet yield of the reaction
- The singlet and triplet yields are plotted against the magnetic field

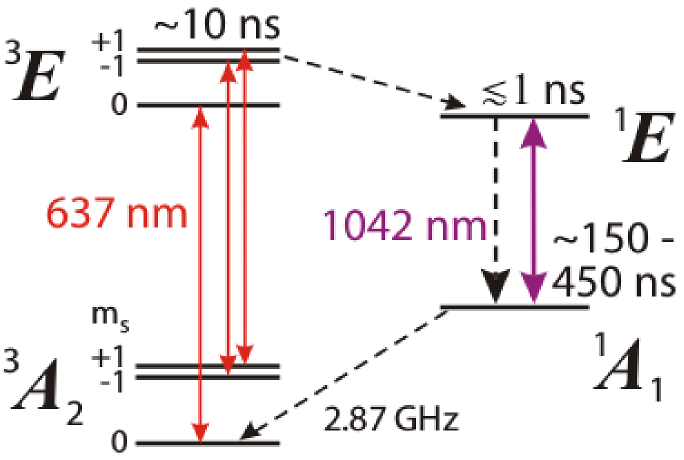
Results: Singlet yield vs geomagnetic field



Diamond NV Center – Electronic Structure

- NV^- center has triplet ground and excited states with ZFS
- **Optical transitions:** Optical pumping at 637nm Radiative and non-radiative transitions
- **Spin Dynamics:** Transition between the spin sublevels of a given spin state
- The ground and excited state can evolve under the magnetic field
- Fluctuating magnetic fields dephase the spin state





Ground State and Excited State Hamiltonian

- Ground State Hamiltonian

$$\hat{H}_{gs} = D_{gs}\left[\hat{S}_z^2 - S(S+1)/3\right] + A_{gs}^{\parallel}\hat{S}_z\hat{I}_z + A_{gs}^{\perp}\left[\hat{S}_x\hat{I}_x + \hat{S}_y\hat{I}_y\right] + P_{gs}\left[\hat{I}_z^2 - I(I+1)/3\right]$$

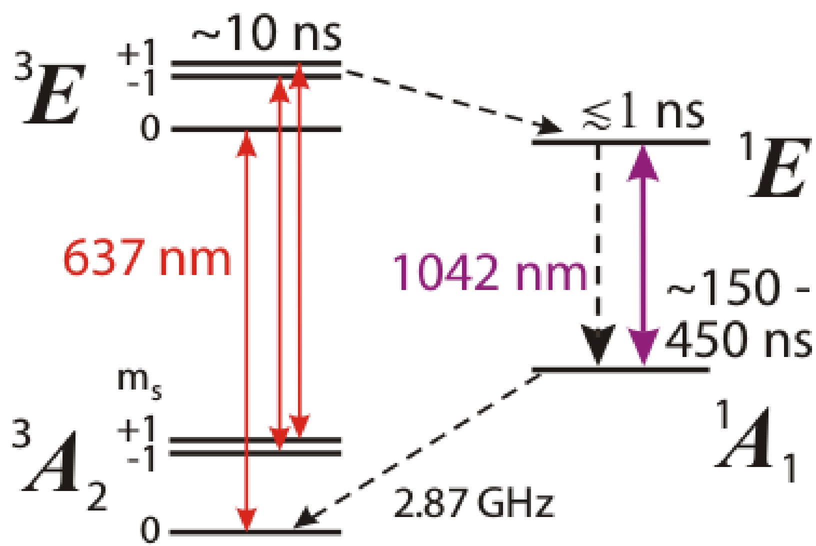
- Excited State Hamiltonian

$$\hat{H}_{es}^{RT} = D_{es}^{\parallel}\left[\hat{S}_z^2 - S(S+1)/3\right] + A_{es}^{\parallel}\hat{S}_z\hat{I}_z + A_{es}^{\perp}\left[\hat{S}_x\hat{I}_x + \hat{S}_y\hat{I}_y\right] + P_{es}\left[\hat{I}_z^2 - I(I+1)/3\right]$$

M. W. Doherty, et al. Physics Reports 528.1 (2013).

Simulation Scheme

- NV center is spin 1 system with spin 1 nucleus [Two 3 level systems]



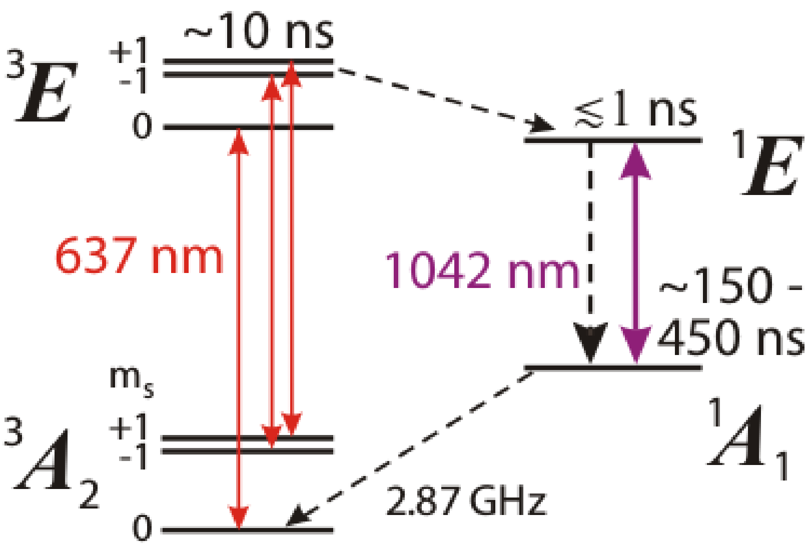
- Ground state and excited states evolve under 9 dimensional Hilbert space

- Total Hamiltonian

$$H = \begin{bmatrix} [H_{gs}] & 0 & 0 \\ 0 & [H_{es}] & 0 \\ 0 & 0 & [SS] \end{bmatrix}$$

H_{gs} = 9x9
H_{es} = 9x9
SS = 2x2

Projection Operators



$$H = \begin{bmatrix} [H_{gs}] & 0 & 0 \\ 0 & [H_{es}] & 0 \\ 0 & 0 & [SS] \end{bmatrix}$$

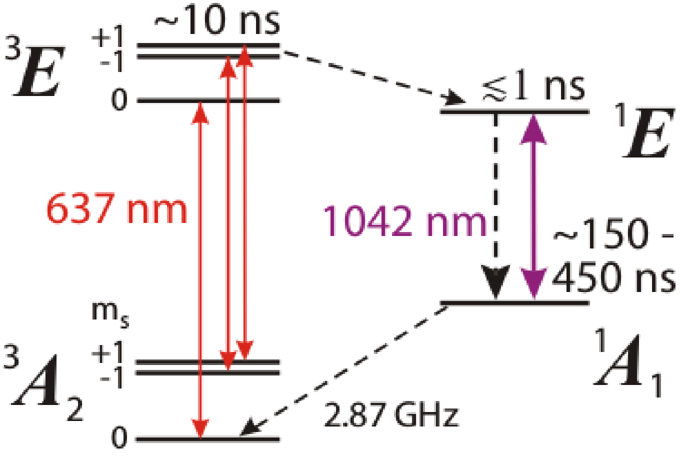
- P1 = 3E(m_s=+1, m_l=-1) --> 1E
- P2 = 3E(m_s=+1, m_l=0) --> 1E
- P3 = 3E(m_s=+1, m_l=+1) --> 1E
- P4 = 3E(m_s=-1, m_l=-1) --> 1E
- P5 = 3E(m_s=-1, m_l=0) --> 1E
- P6 = 3E(m_s=-1, m_l=+1) --> 1E
- P7 = 1E --> 1A
- P8 = 3E(m_s=0, m_l=-1) --> 3A(m_s=0, m_l=-1)
- P9 = 3E(m_s=0, m_l=0) --> 3A(m_s=0, m_l=0)
- P10 = 3E(m_s=0, m_l=+1) --> 3A(m_s=0, m_l=+1)
- P11 = 3E(m_s=+1, m_l=-1) --> 3A(m_s=+1, m_l=-1)
- P12 = 3E(m_s=+1, m_l=0) --> 3A(m_s=+1, m_l=0)
- P13 = 3E(m_s=+1, m_l=+1) --> 3A(m_s=+1, m_l=+1)
- P14 = 3E(m_s=-1, m_l=-1) --> 3A(m_s=-1, m_l=-1)
- P15 = 3E(m_s=-1, m_l=0) --> 3A(m_s=-1, m_l=0)
- P16 = 3E(m_s=-1, m_l=+1) --> 3A(m_s=-1, m_l=+1)

Phenomenological Master Equation

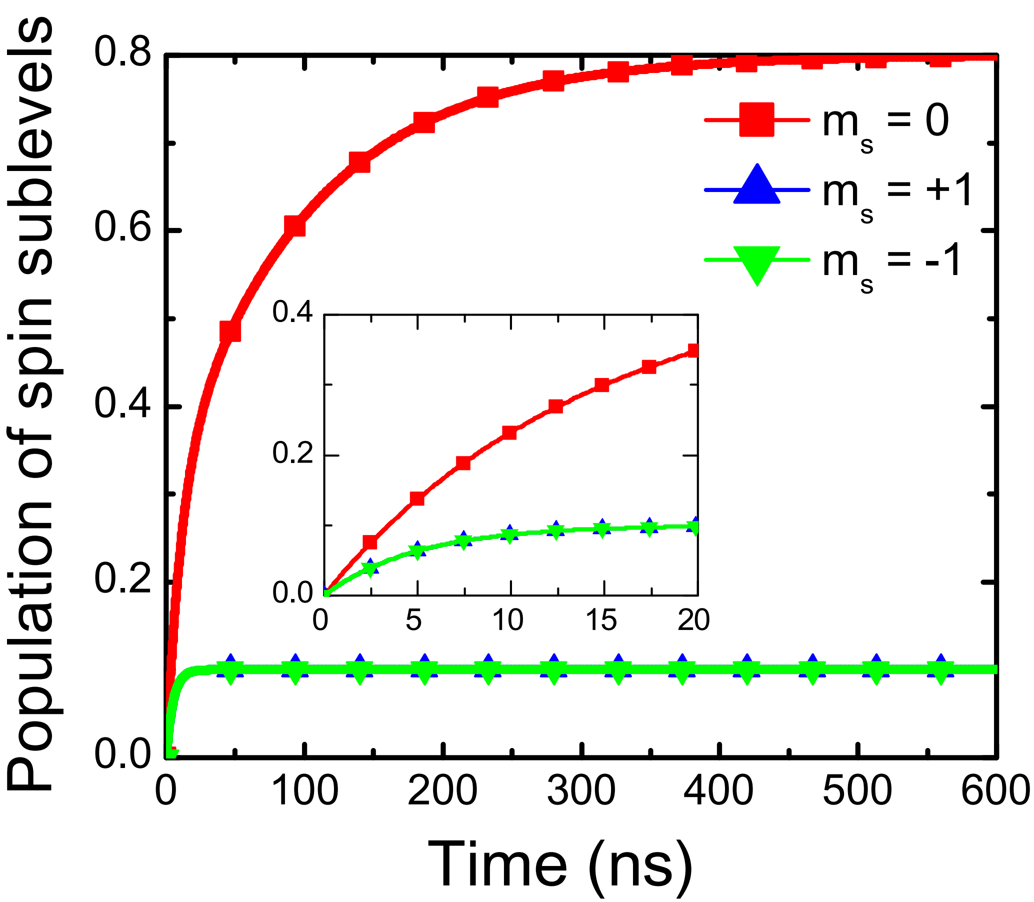
Phenomenological Master Equation:

$$\dot{\rho} = -\frac{i}{\hbar}[H,\rho] + \sum_{i=1}^{16} k_i \left[P_i \rho P_i^\dagger - \frac{1}{2}\left(P_i^\dagger P_i \rho + \rho P_i^\dagger P_i \right) \right]$$

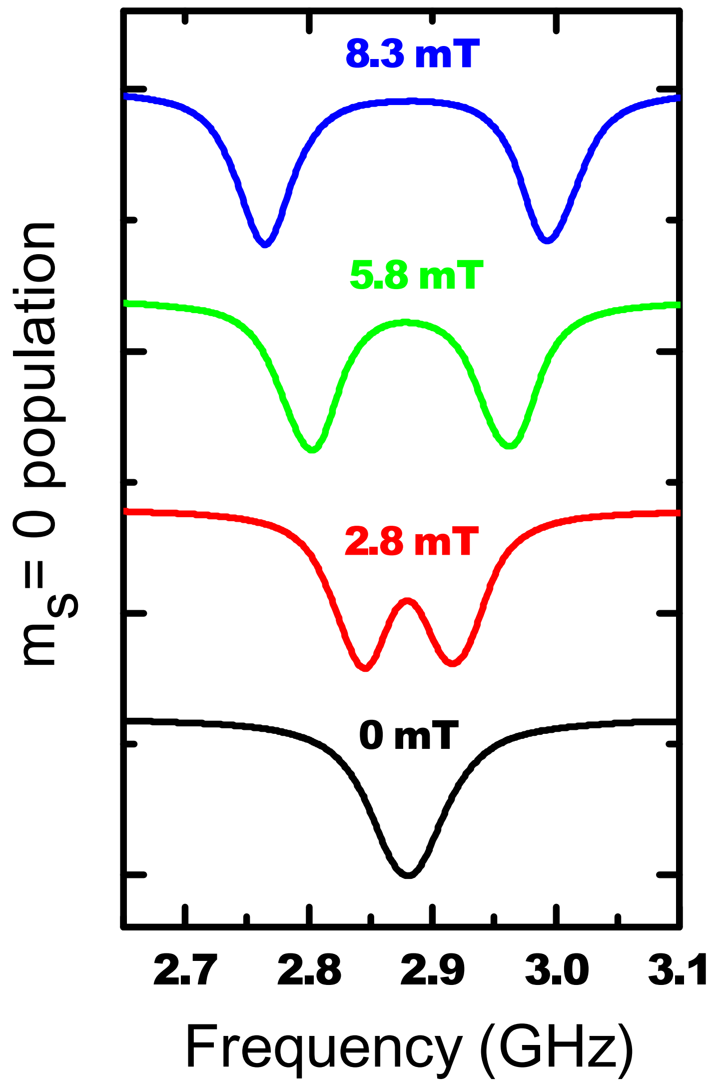
k_i is governed by various relaxation rates



Results



Optical Polarization



ODMR

Acknowledgements

Erik M. Gauger, Royal Society of Edinburgh Fellow

Martin B. Plenio, Ulm University

Thank you