Manual for TimeESR v0.4 & v0.5

Jose Reina-Gálvez[1,*] and Nicolás Lorente[2,3,†]

[1] *Center for Quantum nanoscience, Ewha Womans University, Seoul, Republic of Korea*
[2] *Centro de Física de Materiales CFM/MPC (CSIC-UPV/EHU), 20018 Donostia-San Sebastián, Spain*
[3] *Donostia International Physics Center (DIPC), 20018 Donostia-San Sebastián, Spain*

(Dated: July 27, 2022)

I. DIFFERENCE BETWEEN V0.4 AND V0.5

Version v0.5 is a rewrite of v0.4 changing the flow of the code with the aim of reducing memory requirements. The flow of the code is easily seen in the main file `TimeESR.f90`. For version v0.4, the flow is to read the data from the input files, diagonalize the quantum-dot Hamiltonian (or nanostructure Hamiltonian) and compute the rates that will be used in the Runge Kutta solver. By storing all rates in arrays, the code is very fast. But as soon as 5E5 points in time are needed, the code needs Gb of storage. This gets quickly bad as more spins or larger spins are added to the nanostructure (or quantum dot). Version v0.5 solves this by computing rates only when the Runge-Kutta solver needs it. The memory requirements dramatically go down but the time performance is about $5\times$ the time used in version v0.4 (latest test seem to show that actually the performance of v0.5 is comparable to v0.4, so I discontinue v0.4, Nicolás).

None of the versions is parallel because parallelizing Runge Kutta algorithms is difficult (?). We do not see a way to do this right now (parallel Runge Kutta exists but it is not easy to implement as we have done here).

II. THEORY AND APPROXIMATIONS

We proceed like in Refs. [1] to derive the quantum master equation (QME) by treating the coupling between the impurity and the reservoirs to the lowest (second) order in perturbation theory in H_T. This approximation is the Born-Markov approximation. The reduced density matrix in the adsorbate's eigenstate basis set is

$$\rho_{lj}(t) = \mathrm{Tr}\left[\hat{\rho}_T(t)|l\rangle\langle j|\right], \tag{1}$$

with the trace taken over all the degrees of freedom of the total system and $\hat{\rho}_T(t)$ the time-dependent density matrix of the full system [1, 2]. The equation of motion for $\rho_{lj}(t)$ is

$$\begin{aligned}\hbar\dot{\rho}_{lj}(t) - i\Delta_{lj}\rho_{lj}(t) = \sum_{vu}\Big\{\left[\Gamma_{vl,ju}(t) + \Gamma^*_{uj,lv}(t)\right]\rho_{vu}(t) - \\ \Gamma_{jv,vu}(t)\rho_{lu}(t) - \Gamma^*_{lv,vu}(t)\rho_{uj}(t)\Big\},\end{aligned} \tag{2}$$

where we have denoted $\Delta_{l,j} = E_l - E_j$. All indices (l, j, v, u) refer to many-body eigenstates of the Hamiltonian for the nanostructure or quantum dot, H_{QD}.

The above QME, Eq. (2), is physically meaningful in the limit of weak coupling between impurity and electrodes. Here, weak means that the induced broadening of the impurity levels by the couplings to the electrodes is smaller than the typical separation between levels, $\Delta_{l,j}$. In this way, we make sure that the dynamics induced by the electrode is a small perturbation of the intrinsic impurity dynamics. Only in this limit can a Lindblad form be written for the equation of motion of the reduced density matrix. As we shall see, this restricts the systems to those that have $\Delta_{l,j}$ within the applied DC bias.

The tunnel coupling between the impurity and the two reservoirs is described by the Hamiltonian

$$H_T(t) = \sum_{\alpha k\sigma}\left(w_\alpha(t)c^\dagger_{\alpha k\sigma}d_\sigma + w^*_\alpha(t)d^\dagger_\sigma c_{\alpha k\sigma}\right). \tag{3}$$

* galvez.jose@qns.science
† nicolas.lorente@ehu.eus

Here we introduce a time-dependent hopping induced by the driving which explicitly reads:

$$w_\alpha(t) = w_\alpha^0 \left[1 + A_\alpha \cos(\omega t)\right], \tag{4}$$

The driving is then assumed to produce a modulation of the hopping integrals, and all other effects are neglected. The advantage of this is that we only need to diagonalize once the impurity's Hamiltonian (or adsorbates) to define the above basis set. (NB: this is tricky... we need to do more research on this, in particular we are oversimplifying the bias-dependence, Nicolás).

For the case of one single orbital, we consider all possible configurations for zero, one and two electrons in the atom of the tunnel junction or adsorbate. These configurations are $|p\rangle$ with $p = \downarrow, \uparrow, \emptyset, 2$ (this is the order in the code). The first two account for one electron states while the third and forth are labelling the spin singlets with zero and two electrons, respectively. The nanostructure's or quantum dot Hamiltonian in this basis reads:

$$\begin{aligned} H_{QD} = &\sum_p \varepsilon_p |p\rangle\langle p| + \sum_{i=1}^{N_m}\sum_{j=1}^{3} g_{i,j}\mu_B S_{i,j} B_{i,j} \\ &+ \sum_{(i1,i2)}\sum_{j=1}^{3} J_j(i1,i2) S_{i1,j} S_{i2,j} + \sum_{i=1}^{N_m} (B_{20}O_{20} + B_{40}O_{40} + B_{44}O_{44} + B_{22}O_{22})_i, \end{aligned} \tag{5}$$

where $\varepsilon_\sigma = \varepsilon + g\mu_B B_z \sigma$ for $\sigma = \uparrow$ or $\downarrow$, $\varepsilon_2 = 2\varepsilon + U$ and $\varepsilon_\emptyset = 0$ for versions v04 and v05 of the code. N_m is the number of sites, spins or molecules in the nanostructure, where the orbital is a spin-1/2 molecule. O_{nm} refer to the Stevens operators with the corresponding coefficient for each site i. $J_j(i1,i2)$ is the component j of the exchange interaction in the pair $(i1,i2)$ and the Zeeman interaction is also assumed to be anisotropic (depends on j) and site dependent (we can assume that magnetic heavy atoms that are nearby behave classically and they just produce a local magnetic field of component j on site i, of magnitude $B_{i,j}$ or the local structure of the electrodes or ...). For more info you can read Ref. [3, 4]. The tunneling Hamiltonian in this $|p\rangle$ basis set is:

$$H_T(t) = \sum_{\alpha k\sigma} \left(w_\alpha(t) c_{\alpha k\sigma}^\dagger |\emptyset\rangle\langle\sigma| + w_\alpha(t) c_{\alpha k\sigma}^\dagger |\bar{\sigma}\rangle\langle 2| + h.c. \right) \tag{6}$$

where $\bar{\sigma}$ indicates the opposite to the σ spin projection.

In the presence of a complex nanostructure with one orbital and then a spin system, such as the one taken into account to write Eq. (5), the basis set is taken as the tensorial product

$$|n\rangle = |p\rangle \bigotimes_{i=1}^{N_m} |S_i, S_{z,i}\rangle. \tag{7}$$

In `TimeESR` the basis set appears through the expression of the vectorial spin operators that is used to set up the Hamiltonain, Eq. (5). This is done in module `KrondProd.f90` using Pauli matrices that depend on the value of each local spin, and just keeping track of the matrix element of the spin operator where the local spin enters. This is also done for the orbital part, defining an extended Pauli matrix for it. By using just the relevant indices, the tensorial product is simple and efficient. The maximum local spin that `TimeESR` can deal with is $S = 5/2$ because we hardwired all Pauli matrices up to $S = 5/2$ in `spin_parameters.f90` (it is possible to write an algorithm for any spin S, but...).

It is convenient to express both the adsorbate and tunneling Hamiltonians in the many-body basis of eigenstates of the adsorbate Hamiltonian because in our problem, we assume that the quantum impurity is not subjected to time-dependent fields. The eigenstate basis is given by,

$$H_{QD}|l\rangle = E_l|l\rangle. \tag{8}$$

Accordingly, we write $H_T(t)$ in terms of the Hubbard operators $|l\rangle\langle j|$ obtained from the adsorbate many-body eigenstates [5],

$$H_T(t) = \sum_{\alpha k\sigma lj} \left(w_\alpha(t) c_{\alpha k\sigma}^\dagger \lambda_{lj\sigma} |l\rangle\langle j| + h.c. \right),$$

that explicitly contains the matrix element that reflects the change of basis set

$$\lambda_{lj\sigma} = \langle l| \left(|\emptyset\rangle\langle\sigma| + |\bar{\sigma}\rangle\langle 2| \right) |j\rangle. \tag{9}$$

We expand the adsorbate's Hamiltonian by including a spin Hamiltonian that takes into account a complex magnetic nanostructure described by quantum spins.

Equation (14) of Ref. [1] is the rate for long pulses. Indeed, it is obtained at a time t when the initial time $t_n \to \infty$. Let us assume now, that we are looking into a pulse for time $t \in [t_n, t_{n+1})$. The general expression for the rate is [1]:

$$\begin{aligned}\Gamma_{vl,ju}(t) &= \frac{1}{\hbar}\sum_{\alpha,\sigma}[\lambda_{vl\sigma}\lambda^*_{uj\sigma}w_\alpha(t)D_{\alpha\sigma}\int_{-\infty}^{\infty}d\epsilon f_\alpha(\epsilon)\\ &\times\int_0^\infty w^*_\alpha(t-\tau)e^{-\frac{i}{\hbar}\Delta_{ju}\tau}e^{\frac{i}{\hbar}\epsilon\tau}e^{-\tau/\tau_c}d\tau\\ &+\lambda^*_{lv\sigma}\lambda_{uj\sigma}w^*_\alpha(t)D_{\alpha\sigma}\int_{-\infty}^{\infty}d\epsilon(1-f_\alpha(\epsilon))\\ &\times\int_0^\infty w_\alpha(t-\tau)e^{-\frac{i}{\hbar}\Delta_{ju}\tau}e^{-\frac{i}{\hbar}\epsilon\tau}e^{-\tau/\tau_c}d\tau\\ &=\frac{1}{\hbar}\sum_{\alpha,\sigma}[\lambda_{vl\sigma}\lambda^*_{uj\sigma}w_\alpha(t)D_{\alpha\sigma}I_1\\ &+\lambda^*_{lv\sigma}\lambda_{ju\sigma}w^*_\alpha(t)D_{\alpha\sigma}I_2,]\end{aligned} \tag{10}$$

where I_1 is the term:

$$I_1=\int_{-\infty}^{\infty}d\epsilon f_\alpha(\epsilon)\int_0^\infty w^*_\alpha(t-\tau)e^{-\frac{i}{\hbar}\Delta_{ju}\tau}e^{\frac{i}{\hbar}\epsilon\tau}e^{-\tau/\tau_c}d\tau, \tag{11}$$

and I_2 is defined by:

$$I_2=\int d\epsilon(1-f_\alpha(\epsilon))\int w_\alpha(t-\tau)e^{-\frac{i}{\hbar}\Delta_{ju}\tau}e^{-\frac{i}{\hbar}\epsilon\tau}e^{-\tau/\tau_c}d\tau, \tag{12}$$

we use that the hopping contains the driving in the form of a finite pulse for time $t \in [t_n, t_{n+1})$, where we exclude $t = t_{n+1}$, then we write the hopping matrix elements as:

$$w_\alpha(t) = w^0_\alpha(1 + A_{n,\alpha}\cos(\omega t)), \tag{13}$$

defining $\delta = 1/\tau_c$ and if we assume $\delta \to 0^+$, then I is just

$$\begin{aligned}I = w^{0*}_\alpha&\int_{-\infty}^{\infty}d\epsilon f_\alpha(\epsilon)\left(\frac{i}{\epsilon-\Delta_{ju}+i\delta}\right.\\ &\left.+\ A_n\int_0^\infty d\tau\cos(\omega(t-\tau))e^{i(\epsilon-\Delta_{ju}+i\delta)\tau}\right).\end{aligned} \tag{14}$$

Let us rename the time integral of the expression for I as I_a, such that

$$I_a = A_n\int_{-\infty}^{\infty}d\epsilon f_\alpha(\epsilon)\int_0^\infty\cos(\omega(t-\tau))e^{i(\epsilon-\Delta_{ju}+i\delta)\tau}. \tag{15}$$

After straightforward algebra, with a change of variables $x = t - \tau$, we reach the expression

$$\begin{aligned}I_a &= \frac{A_n}{2}\int_{-\infty}^{\infty}d\epsilon f_\alpha(\epsilon)e^{i(\epsilon-\Delta_{ju}+i\delta)t}\theta(t_{n+1}-t)\\ &\times[\frac{e^{-i(\epsilon-\Delta_{ju}-\omega)t}-e^{-i(\epsilon-\Delta_{ju}-\omega)t_n}e^{\delta t_n}}{-i(\epsilon-\Delta_{ju}-\omega)+\delta}\\ &+\frac{e^{-i(\epsilon-\Delta_{ju}+\omega)t}-e^{-i(\epsilon-\Delta_{ju}+\omega)t_n}e^{\delta t_n}}{-i(\epsilon-\Delta_{ju}+\omega)+\delta}].\end{aligned} \tag{16}$$

here, we have neglected the terms on δt because they are always infinitesimal.

Expression Eq. (16) becomes

$$\begin{aligned}I_a &= \frac{A_n}{2}\int_{-\infty}^{\infty}d\epsilon f_\alpha(\epsilon)e^{i(\epsilon-\Delta_{ju}+i\delta)t}\theta(t_{n+1}-t)\\ &\times[\frac{e^{-i(\epsilon-\Delta_{ju}-\omega)t}}{-i(\epsilon-\Delta_{ju}-\omega)+\delta}\\ &+\frac{e^{-i(\epsilon-\Delta_{ju}+\omega)t}}{-i(\epsilon-\Delta_{ju}+\omega)+\delta}].\end{aligned} \tag{17}$$

If we assume that the Lamb shift is zero (WARNING: this is a really bad approximation that can yield wrong results, in v0.4 on we keep the Lamb shift, see section IV), this yields

$$I_a = \frac{A_n}{2}\int_{-\infty}^{\infty} d\epsilon f_\alpha(\epsilon) \times [\pi\delta(\epsilon - \Delta_{ju} - \omega)e^{i\omega t} + \pi\delta(\epsilon - \Delta_{ju} + \omega)e^{-i\omega t}] \tag{18}$$

And the final expression for the full rate, Eq. (10), becomes:

$$\Gamma_{vl,ju}(t) = \frac{1}{2}\sum_{\alpha,\sigma}[\lambda_{vl\sigma}\lambda^*_{uj\sigma}\gamma_{\alpha\sigma}\ (\ f_\alpha(\Delta_{ju}) + f_\alpha(\Delta_{ju} - \hbar\omega)\frac{A_\alpha}{2}e^{-i\omega t} + f_\alpha(\Delta_{ju} + \hbar\omega)\frac{A_\alpha}{2}e^{i\omega t}\ ) \times (1 + A_\alpha\cos(\omega t)) + \cdots \tag{19}$$

This is a nice and transparent way of writing Eq. (14) of Ref.[6]. Indeed, if $\hbar\omega$ is negligible in front of the excitation energies Δ_{ju} then we see that expression, Eq. (19), reduces to

$$\Gamma_{vl,ju}(t) \approx \frac{1}{2}\sum_{\alpha,\sigma}[\lambda_{vl\sigma}\lambda^*_{uj\sigma}\gamma_{\alpha\sigma}f_\alpha(\Delta_{ju})[1 + A_\alpha\cos(\omega t)]^2 + \cdots \tag{20}$$

where $\gamma_{\alpha\sigma}$ is used to include the driving, by modifying it to:

$$\gamma^0_{\alpha\sigma} = 2\pi|w^0_\alpha|^2 D_\alpha \times (1/2 + \sigma P_\alpha) = \gamma^0_\alpha \times (1/2 + \sigma P_\alpha), \tag{21}$$

where the spin-dependent density of states of the electrodes is defined using the spin polarization P_α of electrode α and its total density of states D_α. We can also define a non-spin-polarized γ^0_α. We also define the driving by means of another effective width:

$$\gamma^1_\alpha = A_\alpha\gamma^0_\alpha. \tag{22}$$

To obtain the actual amplitude of the pulse at a time t, we just define a profile for the pulse, between 0 and 1, and that becomes $A_{n,\alpha}$. (This is not 100% correct, need to rewrite it, maybe it is more clear in the explanation on pulses later on, Nicolás).

A. Short pulses

For short pulses, the term $e^{\delta t_n} = 1$. The consequence is a perfect cancellation of the different terms in Eq. (15), and $I_2 = 0$. Then for short pulses, the rate becomes

$$\Gamma_{vl,ju}(t) = \frac{1}{2}\sum_{\alpha,\sigma}[\lambda_{vl\sigma}\lambda^*_{uj\sigma}\gamma_{\alpha\sigma}f_\alpha(\Delta_{ju})[1 + A_\alpha\cos(\omega t)] + \cdots \tag{23}$$

This is a startling result because comparing Eq. (20) with Eq. (23) we see that for small driving, $A_\alpha < 1$, the long-pulses present a **double** driving, because $[1 + A_\alpha\cos(\omega t)]^2 \approx [1 + 2A\cos(\omega t)]$ as compared with $[1 + A_\alpha\cos(\omega t)]$ for short pulses, in Eq. (23).

If the driving is large, $A_\alpha > 1$, then the difference is more dramatic because the driving difference is between linear (short pulses) to quadratic (long pulses).

`TimeESR` has been coded using the short-pulse expression, Eq. (23). We need to take this into account when comparing the continuous wave (CW) ESR spectra of the Floquet code with the `TimeESR`. In particular, the driving will be roughly half for `TimeESR` under the same set of parameters.

B. Computation of $\lambda_{ij\sigma}$

A key element of the code is $\lambda_{ij\sigma}$, given by Eq. (9). We explain here how this is implemented in versions v04 and v05. These version only consider one electronic orbital of the nanostructure (impurity) that couples to the electrodes and permits us to have an electron current in the system. We already have a multi-orbital version, but needs debugging.

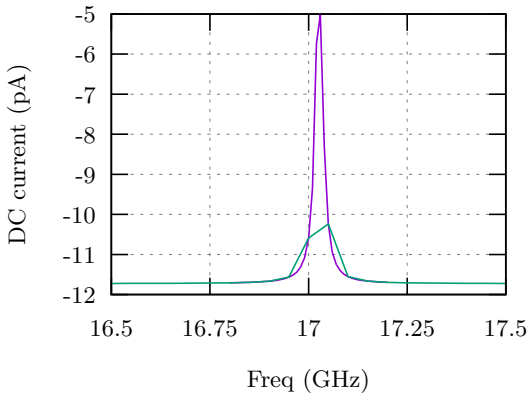

FIG. 1. CW spectra for the case of Ref. [6] computed with real rates. The agreement is excellent. The difference between the two gaphs is the sampling in Frequencies: the purple curve contains 5× more points. In these calculations the spin polarization of the left electrode is 100%. The bias for the left electrode is -15 mV, and for the right one is 15 mV. $U = 100$ meV and $\epsilon = -10$ meV. $\gamma_L^0 = 0.1$ μeV, $\gamma_L^1 = 0.15$ μeV (the driving is 1.5, but keep in mind the factor of two with Floquet). For the right electrode $\gamma_R^0 = 5.$ μeV, $\gamma_R^1 = 0$.

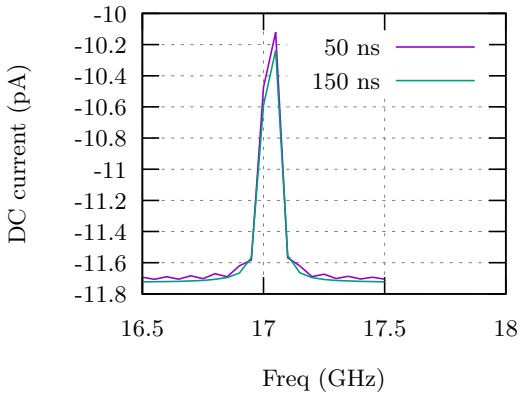

FIG. 2. Convergence with duration of the time propagation. The sampling of points correspond to the green curve of Fig. 1.

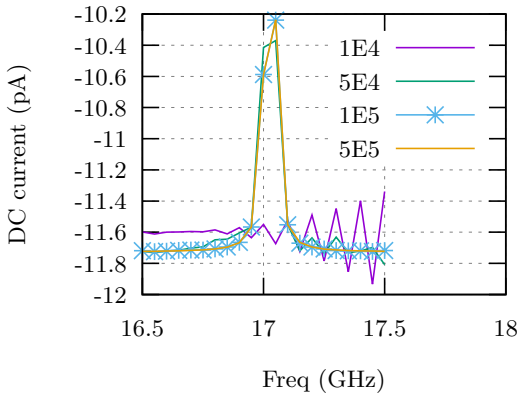

FIG. 3. Convergence with number of points for a fixed final time (different step in time). The main difference between the cyano and the yellow curve is the smoother behavior of the latter. As a rule of thumb, results are converged with about 10^5 points when the time limit is long enough.

After diagonalization of the H matrix, the one written in Eq. (5), the coefficients of the eigenstates are stored in the same matrix in the code. Then eigenstate $|i\rangle$ can be expressed as

$$|i\rangle = \sum_n H(n,i)|n\rangle, \tag{24}$$

where $\{|n\rangle\}$ is the basis set used to write Eq. (5). The basis set is given by the tensorial product first of the orbital degrees of freedom and next of the rest of the spin structure (see discussion after Eq. (5).

Then we can compute

$$\begin{aligned}\lambda_{ij\sigma} &= \langle i|\left(|\emptyset\rangle\langle\sigma| + |\bar{\sigma}\rangle\langle 2|\right)|j\rangle \\ &= \sum_{n,m} H^*(n,i)H(m,j)\langle n|\left(|\emptyset\rangle\langle\sigma| + |\bar{\sigma}\rangle\langle 2|\right)|n\rangle,\end{aligned} \tag{25}$$

using that n is given by two indices in our tensorial product (see module `KrondProd.f90`) that we call $i2, i4$ for n and $i3, i5$ for m. Then we easily find

$$\lambda_{ij\sigma} = \sum_{i2,i3,i4} H^*(i4 + (i2-1)\times N/N_{block}(1), i)\; H(i4 + (i3-1)\times N/N_{block}(1), i) \times (\delta_{i2,3}\delta_{\sigma,i3} + \delta_{i2,\bar{\sigma}}\delta_{4,i3}). \tag{26}$$

Here, we have used that $|\emptyset\rangle$ corresponds to state 3 such that we get a Kronecker delta, $\delta_{i2,3}$, etc... $N_{block}(1)$ is the dimension of the first block entering the tensorial product of the basis set, which is always 4 for v04 and v05, but we will generalize in future versions. All this is found at the end of module `H_QD.f90` (of course we do not implement the Kronecker δ, we just replace it by all possible values in the final expression).

III. RESULTS AND CONVERGENCE

A. Convergence on CW spectra

Figure 1 shows the results of a CW calculation using the average of the last two periods of the current at long times. The results is in excellent agreement with Ref. [6]. The main difference with the calculations of Ref. [6] is the above problem with the driving. It is roughly half for the present calculation.

Using 5×10^5 time points for all graphs, but cutting the propagation at 50 ns, vs extending it to 150 ns, leads to a big difference as can be seen in Fig. 2. Even though the populations seem to have reached their long-time limit at 50 ns, the DC current is not converged yet. The actual values of the time limits depend on the value of γ_α. The smaller γ_α, the larger the propagation time. However, the sampling of the time interval depends on other magnitudes.

Obviously changing the number of time points is crucial for the correct behavior. The Runge-Kutta algorithm altogether fails for very few points (large time step).

IV. FINITE BROADENING CALCULATIONS

To include the Lamb shift that is needed to get correct results, we have replaced the infinitesimal δ in the above rates, Eqs. (16) and (17), by finite $1/\tau_c$ that represents a broadening of the impurity Green's functions. Strictly speaking this broadening should not be there because we are keeping everything to the lowest order in the hopping (causing the broadening). However, we think it is physical and it improves the results greatly as you can see in the following. As a rule of thumb, the value of $1/\tau_c$ should be of the order of $\gamma_R + \gamma_L$ because it corresponds to the broadening caused by the electrodes. In reality, it is probably way more complex than this and we just leave it as an adjustable parameter.

Other approximations: we tried first to implement Eqs. (16) and (17) with the finite broadening. It is impractical because you need to perform an integration in energy for each value of the propagation in time. We then approximate the energy integrals using the fact that for long pulses the t_n value disappears. We need to think this better.

Then it is straightforward to get the new rates. It all boils down to replacing $\pi f_\alpha(\Delta_{ju})$ by I_{11} and $\pi(1 - f_\alpha(-\Delta_{ju}))$ by I_{21}, where

$$I_{11} = i\int_{-\infty}^{\infty} d\epsilon \frac{f_\alpha(\epsilon)}{\epsilon - \Delta_{ju} + i/\tau_c}. \tag{27}$$

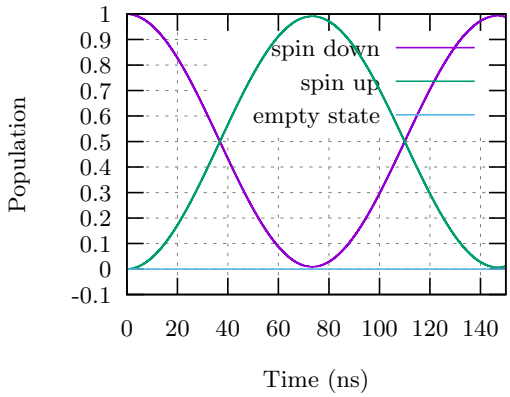

FIG. 4. Population of the spin down (initial population) spin down and empty states. The frequency is on resonance with the peak of the CW spectra. Here, the magnetic field is perpendicular and .60827625 T, such that the resonance frequency is 17.02717787 GHz . A slightly tilted magnetic field leads to a difference of spin up and down populations (typically we use $(0.6, 0, 0.1)T$ this is why we have this weird $.608\cdots$ when we make it perpendicular to the electrode's spin polarization).

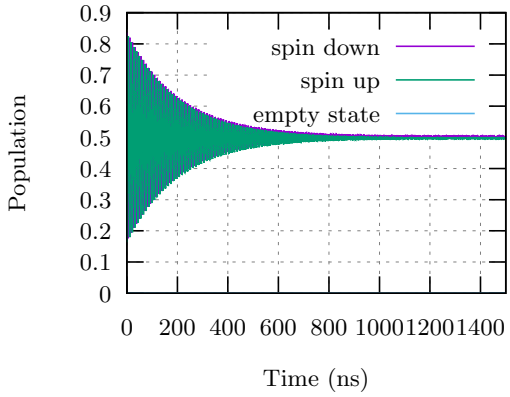

FIG. 5. Same as before but for much longer times, such that decoherence is visible. Here, we kept the same number of points in the Runge-Kutta algorithm, but further increasing the times limits led to having populations without oscillations, showing that a "seemingly reasonable" solution can be obtained when the time step is just a bit too large for being converged, and giving totally misleading results.

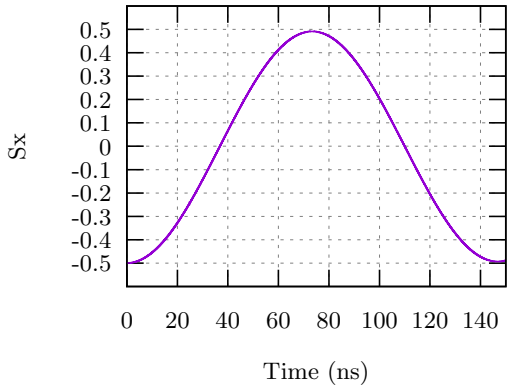

FIG. 6. Low-bias results: On-resonance $\langle \hat{S}_x \rangle$ as a function of time.

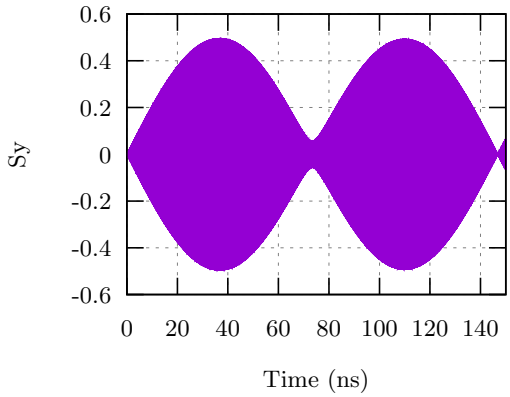

FIG. 7. Low-bias results: On-resonance $\langle \hat{S}_y \rangle$ as a function of time showing the expected Larmor and Rabi oscillations.

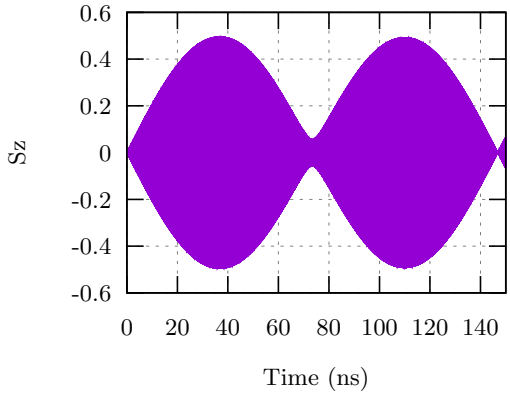

FIG. 8. Low-bias results: On-resonance $\langle \hat{S}_z \rangle$ as a function of time showing the expected Larmor and Rabi oscillations.

and

$$I_{21} = -i \int_{-\infty}^{\infty} d\epsilon \frac{1 - f_\alpha(\epsilon)}{\epsilon + \Delta_{ju} - i/\tau_c}. \tag{28}$$

where we have again neglected $\hbar\omega$ in front of Δ_{ju}.

A. Low-bias calculations

Here all channels are closed, and the current is different from zero because the Green's function has an intrinsic broadening given by the coupling to the electrodes.

Despite of using the electron current coming frpm just the broadening, the results seem to make sense, and are probably correct thanks to the use of a QME that seems to have the right ingredients.

Indeed, we recover perfect Rabi oscillations (only at half the period the populations do not reach one ore zero, but they do at the full period!), see Figs. 4 and 5. This last figure was performed extending the propagation time up to 1.5 μs, to reveal that decoherence is operative. Indeed, there is decoherence because there is some real part to the rates coming from the finite broadening of the impurity's Green's function.

The spin behavior is exactly what we would expect from a two level system: the spin in x (along the magnetic field) goes from $-1/2$ to $1/2$ in half the Rabi period (π-pulse), while the other two components of the spin precess at the driving frequency.

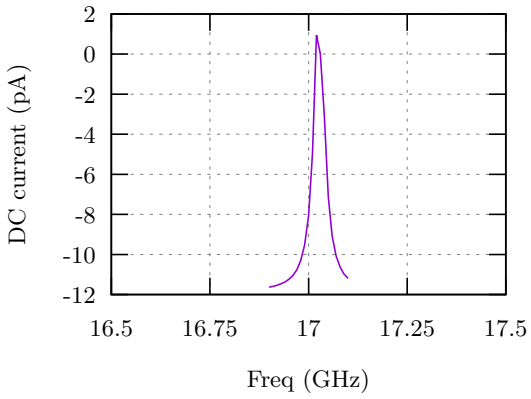

FIG. 9. CW spectra using the full rate with all channels open. In these calculations the spin polarization of the left electrode is 100%. Same as Fig. 1 except that the driving is 2.0 instead of 1.5. Notice the whopping efficiency of the ESR peak of more than 100%!!

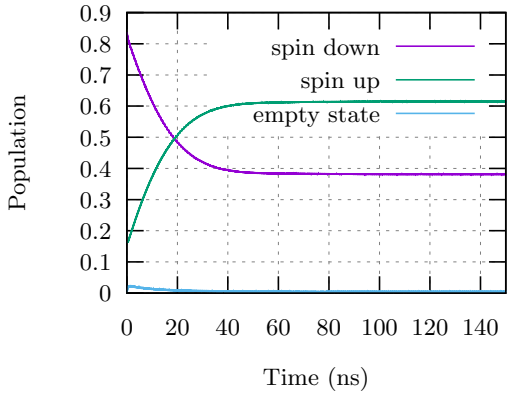

FIG. 10. Large-bias populations for $f = 17.02$ GHz. All channels are open, decoherence wins over. But there is an interesting crossing of the populations with frequency.

V. LARGE-BIAS CALCULATIONS

When the bias is larger than the thresholds to empty or occupy the orbital (in the present case we only have $0 \Longleftrightarrow \uparrow, \downarrow$ because we have done the calculation with a large charging energy, $U = 100$ meV), decoherence hits hard.

Figure 11 shows the populations of these three states near the Larmor frequency, $f = 17.027$ GHz. As can be seen in the CW calculation, Fig.9, 17.03 GHz is not on resonance. This is further seen because the corresponding population should be equal to 0.5 just on resonance. Indeed, Figs. 10 and 11 and 12 show how the final populations cross around 0.5 as the frequency crosses the resonance value.

VI. HOW TO RUN THE CODE

In `./Time_ESR/Example/` there is a bash script that repeatedly launches the code for a set of driving frequencies. It fills p the `TimeESR.input` (contains all the variables related to the QME time solution) and `H_QD.input` (contains all the data on the impurity, including magnetic field -which is local, so there are actually as many magnetic fields as spins or molecules or impurities or whatever you want to call them). The script produces a continuous wave ESR spectrum, using a single long pulse with well defined driving frequency.

You can change the script to do loops on other quantities and produce whatever you deem necessary. And you do not need the script, you can eventually just execute TimeESR.out as produced by the Makefile (check out the name you have for the executable in the Makefile) in so far as `TimeESR.input` and `H_QD.input` are in the running directory and TimeESR.out is in one of the directories contained in your `$PATH` variable.

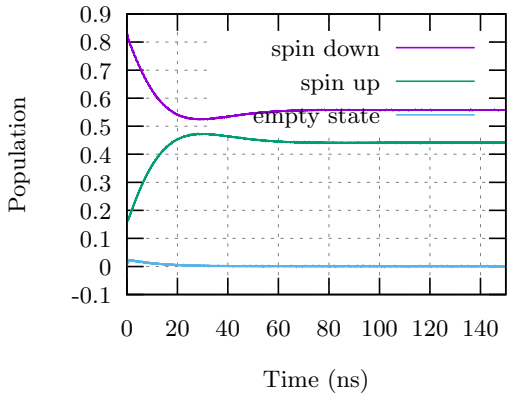

FIG. 11. Large-bias populations for $f = 17.03$ GHz. All channels are open, decoherence wins over. But there is an interesting crossing of the populations with frequency.

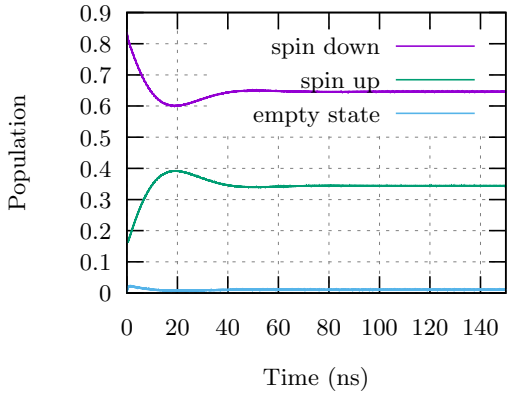

FIG. 12. Large-bias populations for $f = 17.04$ GHz. All channels are open, decoherence wins over. But there is an interesting crossing of the populations with frequency.

A. Pulse sequence

Pulses are coded to be really versatile. Let me start by the above example of a single long pulse. The idea is that the duration of all pulses has to be the same as the time propagation time. In the header of `TimeESR.input`,

```
****************************************************************
************ Please keep this FORMAT ***************************
****************************************************************
500000 !Ntime, number of points used in the time propagation
0.  150. !Initial and final time in nanoseconds
-----------------Pulse definition block-------------------------
1 ! Number of pulses (must contain zero pulses too)
1 ! Maximum number of frequencies
0.  150.  !times for first pulse of zero amplitude
1. ! Amplitude (loop on Freq is faster)
17.02717787 ! first  pulse frequency in GHz
0. ! phase shift in radians for the first interval (no loop on Freq)
-----------------electrode set-up-------------------------------
```

we see that the second line contains the propagation duration (0. 150.) that goes from time 0 to time 150 nanoseconds. Then, the sum of all pulses has to be 150 nanoseconds (as is here).

The "Pulse definition block´´ starts with the number of times the driving is going to change. It should contain then pulses that are zero. The second line is the number of driving frequencies. Some ESR experiments (ENDOR

-Electron nuclear double resonance- but this has been extended to ESR too, some people call it DEER) contain more than one driving frequency. And this is really the key to be able to do complex operations with spins.

Next the Amplitude for the pulse block comes. It has to be less or equal to one (because the actual amplitude of the driving is set later in the code in the variables `gamma_L_1` and/or `gamma_R_1`, the ratio of `gamma_R_1` to `gamma_R_0` is the maximum amplitude of the driving for that electrode -our theory assumes that the electrode-level hopping is the one that is modulated by the applied microwave electric field). And finally you can assume that you have a phase shift for each pulse. The pulses are of the form given by Eq. (29), but we have generalized them here for several frequencies, the hopping with electrode α becomes:

$$w_\alpha(t) = w_\alpha^0(1 + \sum_{n,m} A_{n,m,\alpha}\cos(\omega_m t)), \tag{29}$$

where m refers to the driving frequency m ($f_m = \omega_m/2\pi$) and n to the pulse interval.

Say you have a sequence of pulses where you produce a pulse for 10 ns with frequency $f_1 = 10$ GHz, then you rest for 30 ns (no pulse) and then you apply a pulse that is half the intensity for 20 ns at a frequency $f_2 = 15$ GHz. The input code is:

```
****************************************************************
************ Please keep this FORMAT ************************
****************************************************************
100000 !Ntime, number of points used in the time propagation
0.  60. !Initial and final time in nanoseconds
-----------------Pulse definition block-----------------------------
3 ! Number of pulses (must contain zero pulses too)
2 ! Maximum number of frequencies
0.  10.  !times for first pulse of amplitude 1
1. ! Amplitude (loop on Freq is faster)
10 ! first  pulse frequency in GHz
0. ! phase shift in radians for the first interval (no loop on Freq)
10.  40.  !times for second pulse of zero amplitude
0. ! Amplitude (loop on Freq is faster)
0. ! first  pulse frequency in GHz
0. ! phase shift in radians for the first interval (no loop on Freq)
40.  60.  !times for third pulse of amplitude 0.5
.5 ! Amplitude (loop on Freq is faster)
15 ! first  pulse frequency in GHz
0. ! phase shift in radians for the first interval (no loop on Freq)
-----------------electrode set-up-----------------------------------
```

...(not finished, see below for complete examples).

Next example is the complete file for a case of a single long pulse but with **two frequencies** (like in DEER):

```
****************************************************************
************ Please keep this FORMAT ************************
****************************************************************
500000 !Ntime, number of points used in the time propagation
0.  10. !Initial and final time in nanoseconds
-----------------Pulse definition block-----------------------------
1 ! Number of pulses (must contain zero pulses too)
2 ! Maximum number of frequencies
0.  10.  !times for first pulse of zero amplitude
1. ! Amplitude (loop on Freq is faster)
16.289 ! first  pulse frequency in GHz
1. ! Amplitude (loop on Freq is faster)
15.071 ! second  pulse frequency in GHz
0. ! phase shift in radians for the first interval (no loop on Freq)
-----------------electrode set-up-----------------------------------
0.2 ! in meV : gamma_R_0= 2*pi*W_R_0*W_R_0*rho
```

```
0.01 ! in meV : gamma_L_0 2*pi*W_L_0*W_L_0*rho
0.0 ! in meV: gamma_R_1= 2*pi*W_R_0*W_R_1*rho
0.01 ! in meV: gamma_L_1= 2*pi*W_L_0*W_L_1*rho
100.0 ! Cutoff (meV) for integral in Lambshift
.002 ! Broadening of Green's function (meV)
500000 ! Number of points in I11 and I21 (see Manual)
-----------------bias, temperature, spin polarization-----------------
2.0  ! in mV right electrode bias
-2.0 ! in mV left electrode bias
.1   ! Temperature in Kelvin
0.0   ! Spin polarization of electrode R, between -1 and 1
1.0   ! Spin polarization of electrode L, between -1 and 1
0    ! 0 is left and 1 is right electrode
-----------------output file names-----------------------------------
.true. !.true. write POPULATIONS.dat
C.dat ! file name for the time-dependent current
S.dat ! file name for the current Fourier transform
ESR.dat ! file DC current
-----------------READ previous RUNS: DANGEROUS-----------------------------------------
.false. ! Keep to false to avoid reading previous runs
.true. ! Plot time evolution of spins
------------------END of INPUT-----------------------------------
```

B. Output files

`Spin_distribution.dat` contains the average of the spin on x, y and z for each of the impurities (called "Site´´ in the file, the idea is that we have a single orbital conencted to many spins, and each spin is on one site, or impurity or molecules or whatever of your full quantum dot or nanostructure, so this is BEFORE coupling to the electrodes...) AND for each eigenstate of the diagonalization of the total impurity or quantum dot or nanostructure. The file writes all eigenstates with their corresponding energy in GHz, together with the square value of the spin of each eigenstate and the spin, S, obtained from the square assuming that the square is a good quantum number ($S(S+1)$, if it is not, it will look weird).

As you can see in the input example `TimeESR.input` you can choose the name of the output files for the time-dependent current, the DC current (the average current over the last two oscillations at the end of the propagation interval). The possibility to have a file with the Fourier transform of the current is disabled now, but we can easily put it back (we decided not to include Fast-Fourier routines for plotting something of not much interest).

If the last entry is set to `.true.` the file SpinDynamics.dat contains the real part of $\langle S_x\rangle$, $\langle S_x\rangle$, $\langle S_x\rangle$, $\sqrt{\langle S_x\rangle^2+\langle S_x\rangle^2+\langle S_x\rangle^2}$ for each spin of the nanostructure:q . The first column is the time in nanoseconds and in blocks of 4 columns the information for each spin si plotted. If you want to plot the evolution of the component y for the second spin in your structure, you need to plot the seventh column against the first one. The square is plotted to see if the modules of the vector is roughly constant (should not be but if you start plotting a spin vector it is good to see how bad it gets).

C. Transport properties, bias, temperature, driving

It is important to be aware of the transport regime of the calculation. The code is conceived in the low-hopping approximation (otherwise there is no way to end up with a linear QME) and this forces us to be in the sequential regime. This regime is also mean on-resonance regime, because the transport channel for transmitting one electron through the orbital is open. This means that an electron hops from one of the electrodes (the one with the "good´´ bias) onto the orbital and the state of the full impurity changes because there is an extra electron in the system. This channel is open when the initial and final energy of the full system including electrodes are equal. This boils down to comparing $\Delta_{i,j}$ where i and j are the initial and final states of the impurity, with the applied bias (paying attention at where the bias drop, we always assume a symmetric drop in our calculations, but this can be done more realistically).

The present code uses a finite broadening of the bare Green's function of the impurity, which is a questionable approximation, but seems to yield realistic results. The broadening is given by the variable Γ_C in the code. And we

expect it to be of the order of the broadenings used for computing the hoppings.

The temperature enters in two places: (i) the Fermi function that appears everywhere in the calculation of the rates and (ii) in the initial population, where we assume thermal populations to initialize the propagation (seems to be the realistic experimental case).

Finally the driving is set by the ratio of `gamma_R(L)_1` to `gamma_R(L)_0` for electrode R (or L). Ratios that go above 1 lead to unphysical populations sometime. This needs to be checked. In principle a ratio larger than one seems to violate the conditions of small coupling of the theory. This may not be the case, depending on the total couplings to the electrodes.

A simple example of input for the code, `Time_ESR.input`, is:

```
****************************************************************
************ Please keep this FORMAT ***************************
****************************************************************
500000 !Ntime, number of points used in the time propagation
0.  12. !Initial and final time in nanoseconds
-----------------Pulse definition block-----------------------------
1 ! Number of pulses (must contain zero pulses too)
1 ! Maximum number of frequencies
0.  12.  !times for first pulse of zero amplitude
1. ! Amplitude (loop on Freq is faster)
17.02717787 ! first  pulse frequency in GHz
0. ! phase shift in radians for the first interval (no loop on Freq)
-----------------electrode set-up-----------------------------------
0.2 ! in meV : gamma_R_0= 2*pi*W_R_0*W_R_0*rho
0.01 ! in meV : gamma_L_0 2*pi*W_L_0*W_L_0*rho
0.0 ! in meV: gamma_R_1= 2*pi*W_R_0*W_R_1*rho
0.01 ! in meV: gamma_L_1= 2*pi*W_L_0*W_L_1*rho
100.0 ! Cutoff (meV) for integral in Lambshift
.002 ! Broadening of Green's function (meV)
500000 ! Number of points in I11 and I21 (see Manual)
-----------------bias, temperature, spin polarization---------------
20.0  ! in mV right electrode bias
-20.0 ! in mV left electrode bias
.1    ! Temperature in Kelvin
0.0   ! Spin polarization of electrode R, between -1 and 1
1.0   ! Spin polarization of electrode L, between -1 and 1
0     ! 0 is left and 1 is right electrode
-----------------output file names----------------------------------
.true. !.true. write POPULATIONS.dat
C.dat ! file name for the time-dependent current
S.dat ! file name for the current Fourier transform
ESR.dat ! file DC current
-----------------READ previous RUNS: DANGEROUS----------------------
.false. ! Keep to false to avoid reading previous runs
.true. ! Plot time evolution of spins
------------------END of INPUT-------------------------------------
```

where a total bias of 40 mV is applied, the left electrode is supposed to be 100% spin polarized, and the current is computed on the left electrode.

Equations (27) and (28) are one marked as I11 and I21 in the above file. These integrals are solved with 500000 points using the trapèze-rule in the module `QME.f90` between -100.0 and +100.0 meV of integration limits. These limits need to be checked against bias, as a rule of thumb, the cutoff should be several times the applied bias. But the energy step in the integral needs to be about five times smaller than the broadening Γ_C. In the present example we have a broadening, Γ_C of 0.002 meV, and the energy step is 200 meV/500000 = 0.0004 meV or 5 times smaller. This is not costly in the calculation, but it is critical to avoid numerical problems.

Finally, it is important to keep the previous-to-last entry as `.false.` unless a script is being used to perform some loops on variables of `Time_ESR.input` and the result of the Hamiltonian diagonalization can be reused. The danger is that we may think that we are changing the parameters of the Hamiltonian input, but our calculation is taken the previous values of the running directory if this key is kept to `.true.`

D. The Hamiltonian input

The file `H_QD.input` contains the input for the Hamiltonian. As above, the format is not free and you really need to write on each line what is expected or the code will crash.

Let us write here a typical example of an orbital that interacts with an extra spin 1/2. This can be a simple model for an atom that will only fill in and empty one orbital at low bias, but will see its other electron (in a low-lying level) to undergo an electron spin resonance.

```
*********************************************************
************ Please keep this FORMAT ********************
*********************************************************
2  !Number of spins (molecules or sites)
------ Spin properties: as many blocs as Number of Spins ----
0.5 ! First spin has to be 0.5 for Transport!
0 0 0 0 ! Stephen Coefficients: B20 B22 B40 B44
0 0 1 ! Axis for Stephen Operators
.60827625 0 0.0 ! Local magnetic field in Teslas
2.0 2.0 2.0 ! Local gyromagnetic factors
---------------------------------------------------------------
0.5 ! Second Spin!
0 0 0 0 ! Stephen Coefficients: B20 B22 B40 B44
0 0 1 ! Axis for Stephen Operators
.60827625 0 0.0 ! Local magnetic field in Teslas
2. 2. 2. ! Local gyromagnetic factors
------------ End of spin properties ---------------------------
1  !Number of exchange connected pairs
1  2 !Firs pair of connected pairs
-12500. -12500. -12500. !Jx, Jy, Jz in GHz
------------ End of exchange interactions ---------------------
-1005.0 ! eps_QD  electronic level in meV
1000.0000 ! Hubbard U (meV)
------------ End of electronic interactions -------------------
Hamiltonian.output ! Name of the output file with H
8 ! Number of states to print in Spin_distribution.dat
*********************************************************
************ END of INPUT FILE H_QD.input ***************
*********************************************************
```

Since there are two spins (the orbital one and the other spin) the input requires two sites. There is a block for each spin, that specifies the value of the spin (0.5 for this spin-1/2 case), the Stephen coefficients (give the anisotropy of the system, for spin 1/2 there is none). The axis for the Stephen operators (where the z is for the anisotropy of the atom with respect to the spin-polarization of the electronic current, that is always in z for this code).

Both the local field and the gyromagnetic factors are taken as vectors.

After the blocks for the spins, the exchange interaction block starts. You start by saying how many pairs of spins are exchange-connected and then you write a block for each pair. The exchange interaction is also a vector, to allow for anisotropic exchange interactions, and they are expressed in GHz.

Finally, the on-site energy of the electronic level is written with respect to the Fermi energy without bias, and the Hubbard U energy too, in meV.

As a final remark, do not hesitate to edit the code to find why the reading is crashing: probably you are missing a line, or there is a funny character where there should be a digit. The reading of `H_QD.input` is done in the module `H_QD.f90`, but the reading of `Time_ESR.input` is done in the module `io.f90`

VII. TROUBLESHOOTING

The complexity of the calculation grows exponentially with the spin size. Version v05 is the only one that can tackle more than two spins. However convergence is tricky and at the time of writing these lines, we need a lot more work to do running the code, understanding it and improving the input parameters.

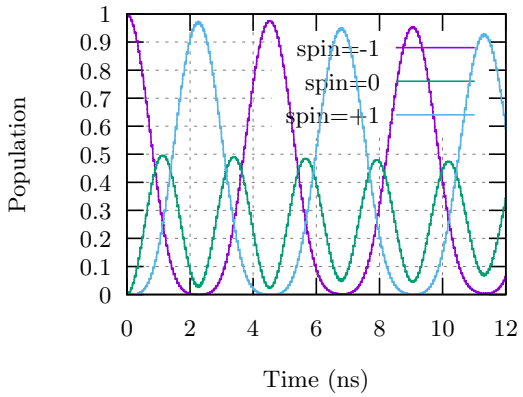

FIG. 13. Populations for the case of one level and one spin-1/2 interacting via a large exchange coupling for $f = 17.027$ GHz (this corresponds to the example of `H_QD.input`). All channels are closed, because we changed the bias to be +2 and -2 mV in the above input file, `Time_ESR.input`. The wiggles ontop of the Rabi oscillation seem to be real and are not caused by a lack of precision (so far as we could check). The finite broadening of the Green's function allows for a direct coupling between $S_z = -1$, $S_z = 0$ and $S_z = +1$.

It is easy to get `NAN`'s running the code. The usual cause is the Runge Kutta going nuts because the time step is way too long. By massively increasing the `Ntime` variable (see examples of `Time_ESR.input` above) the problem is solved, because Runge Kutta is extremely stable and robust. We have the impression that as you increase the Hilbert space (more/larger spins?) the requirements of precision are higher. We might need to change the integration algorithm. But before doing that, we can use bigger computers, and run over longer periods of times, etc...

It seems (not thoroughly tested yet) that Γ_C is critical for avoiding `NAN`'s. Too large values of the broadening leads to unbalance real-imaginary rates and the results stop being physical and eventually the populations blow up. The way to go is to reduce it, making sure that the step in the integration is still some 5 times smaller. Using the trapèze rule is probably the best we can do, fancier algorithms will be slower and less reliable.

All channels closed seem to work, actually thanks to having a finite Γ_C. This is how we decrease the decoherence to levels that allow us to have Rabi oscillations. But it is a bit spooky to use a theory for the sequential regime and then use it in the cotunneling one. In principle, it is wrong. Having said this, it may be physically correct because you are using the tails of the level resonance to do the electron transport and this is probably more likely than doing some higher-order cotunneling transfer. The problem is that we rely on a parameter that is not really justified, Γ_c.

---

[1] J. Reina-Gálvez, N. Lorente, F. Delgado, and L. Arrachea, All-electric electron spin resonance studied by means of floquet quantum master equations, Phys. Rev. B **104**, 245435 (2021).

[2] B. Bhandari, R. Fazio, F. Taddei, and L. Arrachea, From nonequilibrium green's functions to quantum master equations for the density matrix and out-of-time-order correlators: Steady-state and adiabatic dynamics, Phys. Rev. B **104**, 035425 (2021).

[3] D.-J. Choi, N. Lorente, J. Wiebe, K. von Bergmann, A. F. Otte, and A. J. Heinrich, Colloquium: Atomic spin chains on surfaces, Reviews of Modern Physics **91**, 041001 (2019), publisher: American Physical Society.

[4] C. Wolf, F. Delgado, J. Reina, and N. Lorente, enEfficient ab initio multiplet calculations for magnetic adatoms on mgo, The Journal of Physical Chemistry A **124**, 2318 (2020).

[5] A. C. Hewson, *The Kondo Problem to Heavy Fermions*, Cambridge Studies in Magnetism (Cambridge University Press, 1993).

[6] J. Reina-Gálvez, C. Wolf, and N. Lorente, Many-body effects in all-electrical electron spin resonance, .