From Fabry-Perot oscillations to Coulomb blockade in a quantum wire

P3TMA computer project by Varvara Petrova
Supervised by Adeline Crépieux

Centre de Physique Théorique, Aix-Marseille University

February 20, 2017

1 Introduction

Understanding the effect of the interactions on the transition between the Coulomb-blockade and the Fabry-Perot regime in the low-dimensional electronic devices is a long standing problem in condensed matter physics. A lot of progress in understanding of disorder effects has been achieved over the last 20 years, especially for one-dimensional systems (Luttinger theory, bosonization). The Dyson equation technique developed in [1], enables us to explore the interplay between interactions and impurities in a quantum wire. The aim of this project is to take the solutions of Dyson equation for the retarded Green function of the system with two impurities and to draw, using Mathematica, its density of states as a function of position and energy. We will study how the interaction and impurity potential strength changes its profile.

2 Theoretical framework

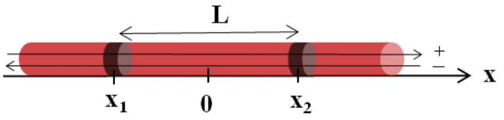


FIG. 1: One-dimensional wire with two impurities located at positions $x_{1,2} = \pm L/2$. The right (+) and left (-) chiralities are denoted by right and left arrows [1].

We consider an infinite one-dimensional interacting quantum wire (QW) with two impurities at positions $x_{1,2} = \pm L/2$ for a given length L as illustrated in figure 1. The impurities model metallic contacts. They are described by backward scattering potentials $\lambda^B_{1,2}$ and forward scattering potentials $\lambda^F_{1,2}$, giving rise to two independent conducting channels corresponding to (+) and (-) collective excitations of (free) bosons. The Hamiltonian can be written as $H = H_0 + H_{int} + H_{imp}$, where H_0 describes the non interacting QW without impurities:

$$H_0 = -i\hbar v_F \sum_{r=\pm} r \int_{-\infty}^{+\infty} \psi_r^\dagger(x), \partial_x \psi_r(x) \mathrm{d}x$$

with v_F is the Fermi velocity, $\psi_r^\dagger(x)$ and $\psi_r(x)$ are the creation and annihilation fermionic operators associated to the right moves (r=+) and left moves (r=-). The Hamiltonian H_{int} describes the Coulomb interaction in the wire:

$$H_{int} = \tfrac{1}{2}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \hat{\rho}(x)V(x,x')\hat{\rho}(x')\mathrm{d}x\mathrm{d}x',$$

where $\hat{\rho}(x) = \sum_{r,r'} \psi_r^\dagger(x)\psi_{r'}(x)$ is the density operator, and V is the Coulomb potential which is assumed to be short range due to screening effects by metallic gates or inter-wire coupling [1]. The impurity Hamiltonian $H_{imp} = H_B + H_F$ is composed of backward-scattering term:

$$H_B = \sum_{r=\pm}\sum_{i=1,2}\int_{-\infty}^{+\infty} \lambda_i^B \psi_r^\dagger(x)\psi_{-r}(x)\mathrm{d}x,$$

and forward-scattering term:

$$H_F = \sum_{r=\pm}\sum_{i=1,2}\int_{-\infty}^{+\infty} \lambda_i^F \psi_r^\dagger(x)\psi_{-r}(x)\mathrm{d}x.$$

In this model we assume that impurities are localized, i.e. $\lambda_i^{F,B} = \Gamma_i^{F,B}\delta(x-x_i)$. The local density of states (LDOS) can be obtained from the generalized retarded Green's function as follows:

$$\rho(x,\omega) = -\tfrac{1}{\pi}\sum_{r,r'} \mathrm{Im}\left\{G_{r,r'}^R(x,x;\omega)\right\} \tag{1}$$

Authors of [1] established the Dyson equation for the retarded Green's function $G_{r,r'}^R(x,x;\omega)$ defined as the Fourier transform of: $G_{r,r'}^R(x,x';t,t') = -i\Theta(t-t')\times$ $\times\langle\{\psi_r(x,t),\psi_{r'}^\dagger(x',t')\}\rangle$ where Θ is the Heaviside function, $\{\,,\,\}$ refers to the anti-commutator and $\langle\,\rangle$ to the thermo-average. Assuming that (i) Coulomb interactions are strongly attenuated with distance and weak in comparison to the energy and (ii) $|x-x_i|\omega \gg v_F$ (this is equivalent to neglect the contributions mixing the impurity potentials and the Coulomb potential), they derived the following expression for the retarded Green's function $G_{r,r'}^R(x,x;\omega)$:

$$\begin{aligned} G_{r,r'}^R(x,x';\omega) &= g_r^R(x,x';\omega)\delta_{r,r'} \\ &+ \sum_{i=1,2} g_r^R(x,x_i;\omega)\left[\Gamma_i^B G_{-r,r'}^R(x_i,x';\omega) + \Gamma_i^F G_{r,r'}^R(x_i,x';\omega)\right] \end{aligned} \tag{2}$$

where g_r^R are the Green's functions of a clean interacting homogeneous wire, associated to $H_0 + H_{int}$. Their form has been derived explicitly in [2] using the Tomonaga-Luttinger theory. In our case [1]:

$$\begin{aligned} g_r^R(x,x';\omega) &= \frac{-e^{irk_F(x-x')}K^2\omega_+}{2\hbar v_F\omega_c\sqrt{\pi}\mathbf{\Gamma}(1+\gamma)}\left(\frac{2i|x-x'|\omega_c}{aK\omega_+}\right)^{\frac{1}{2}-\gamma} \\ &\times\left[\mathbf{K}_{\gamma-\frac{1}{2}}\left(\frac{K|x-x'|\omega_+}{ia\omega_c}\right) - \mathrm{sgn}\left(r(x-x')\right)\mathbf{K}_{\gamma+\frac{1}{2}}\left(\frac{K|x-x'|\omega_+}{ia\omega_c}\right)\right] \end{aligned} \tag{3}$$

where $\omega_+ = \omega + i\epsilon$ ($\epsilon > 0$ is infinitesimal), k_F is the Fermi momentum, $\omega_c = v_F/a$ is a cut-off frequency, a being the small-distance cut-off of the Tomonaga-Luttinger theory. $\mathbf{\Gamma}$ and $\mathbf{K}$ are respectively the gamma and modified Bessel function, K is the interaction parameter ($K = 1$ in the absence of interaction, $0 < K < 1$ for the repulsive Coulomb interaction, $K > 1$ for attractive interaction) and $\gamma = (K + K^{-1} - 2)/4$. Eq. (3) is defined for $x \neq x'$. $g_r^R(x, x; \omega)$ can be approximated, using the expansion of the Bessel functions $\mathbf{K}_\alpha$ at small argument:

$$\mathbf{K}_\alpha \overset{z \to 0}{\sim} \tfrac{1}{2} \left[\mathbf{\Gamma}(\alpha) \left(\frac{2}{z} \right)^\alpha + \mathbf{\Gamma}(-\alpha) \left(\frac{z}{2} \right)^\alpha \right] \left(1 + \mathcal{O}(z^2) \right) \tag{4}$$

Introducing (4) into (3) and neglecting the divergent parts [2], we obtain:

$$g_r^R(x, x; \omega) \approx -\frac{\mathbf{\Gamma}(\frac{1}{2} - \gamma) K^2 \omega_+}{4\hbar v_F \omega_c \sqrt{\pi}\, \mathbf{\Gamma}(1 + \gamma)} \left(\frac{K\omega_+}{2i\omega_c} \right)^{2\gamma - 1} \tag{5}$$

The expressions of $G_{r,r}^R(x_i, x'; \omega)$ and $G_{-r,r}^R(x_i, x'; \omega)$ can be extracted from the Eq. (2) by solving a linear set of equations [1]:

$$G_{r,r}^R(x_i, x'; \omega) = \frac{\left(1 - \chi_r^{\bar{i}\bar{i}}\right) g_r^R(x_i, x'; \omega) + \chi_r^{i\bar{i}} g_r^R(x_{\bar{i}}, x'; \omega)}{\left(1 - \chi_r^{11}\right)\left(1 - \chi_r^{22}\right) - \chi_r^{12}\chi_r^{21}} \tag{6}$$

and

$$\begin{aligned} G_{-r,r}^R(x_i, x'; \omega) &= D^{-1} \sum_{j=1,2} g_{-r}^R(x_i, x_j; \omega) \Gamma_j^B G_{r,r}^R(x_j, x'; \omega) \\ +D^{-1} \Gamma_i^B \Gamma_{\bar{i}}^F &\left[g_{-r}^R(x_i, x_{\bar{i}}; \omega) g_{-r}^R(x_{\bar{i}}, x_i; \omega) - g_{-r}^R(x_i, x_i; \omega) g_{-r}^R(x_{\bar{i}}, x_{\bar{i}}; \omega) \right] G_{r,r}^R(x_i, x'; \omega) \end{aligned} \tag{7}$$

with $\bar{i} = 1$ when $i = 2$, and $\bar{i} = 2$ when $i = 1$. The quantities D and χ_r^{ij} are defined by:

$$\begin{aligned} D = \left[1 - \Gamma_1^F g_{-r}^R(x_1, x_1; \omega)\right] &\left[1 - \Gamma_2^F g_{-r}^R(x_2, x_2; \omega)\right] \\ &- \Gamma_1^F g_{-r}^R(x_1, x_2; \omega) \Gamma_2^F g_{-r}^R(x_2, x_1; \omega) \end{aligned} \tag{8}$$

$$\begin{aligned} \chi_r^{ij} = \Gamma_i^F g_r^R(x_i, x_j; \omega) + D^{-1} \Big[&\Gamma_1^B g_r^R(x_i, x_{\bar{j}}; \omega) \Gamma_2^B g_{-r}^R(x_{\bar{i}}, x_i; \omega) \\ &+ \Gamma_j^B g_r^R(x_i, x_j; \omega) \Gamma_{\bar{j}}^F g_{-r}^R(x_1, x_2; \omega) \Gamma_{\bar{j}}^B g_{-r}^R(x_2, x_1; \omega) \\ &+ \Gamma_i^B g_r^R(x_i, x_j; \omega) \Gamma_i^B g_{-r}^R(x_i, x_i; \omega) \left(1 - \Gamma_{\bar{j}}^F g_{-r}^R(x_{\bar{j}}, x_{\bar{j}}; \omega)\right) \Big] \end{aligned} \tag{9}$$

Remarks. – For the simplicity, we shall restrict ourself to the symmetrical configuration with $\Gamma_{1,2}^{F,B} \equiv \Gamma$. For the obvious numerical reason, the (infinite size) wire will be truncated.

Dimensional analysis and rescaling. – Since $[\lambda_{1,2}^{F,B}] = \mathrm{E}$ and $[\psi] = \frac{1}{\sqrt{\mathrm{L}}}$, we have $[\Gamma] = \mathrm{E.L}$, in the generic notation for energy and length. All the Green's functions depending on $(x, x'; \omega)$ are homogeneous to $\mathrm{E}^{-1}\mathrm{L}^{-1}$. In view of this, all the equations can be rescaled with respect to the characteristic length a and characteristic energy $\hbar\omega_c$. The dimensionless version of the equations (3) and (5) reads: if $x \neq x'$

$$g_r^R(x, x'; \omega) = \frac{-e^{irk_F(x-x')}K^2\omega_+}{2\sqrt{\pi}\mathbf{\Gamma}(1+\gamma)} \left(\frac{2i|x-x'|}{K\omega_+}\right)^{\frac{1}{2}-\gamma}$$
$$\times \left[\mathbf{K}_{\gamma-\frac{1}{2}}\left(-iK|x-x'|\omega_+\right) - \mathrm{sgn}\left(r(x-x')\right)\mathbf{K}_{\gamma+\frac{1}{2}}\left(-iK|x-x'|\omega_+\right)\right]$$

and

$$g_r^R(x, x; \omega) \approx -\frac{\mathbf{\Gamma}(\frac{1}{2}-\gamma)K^2\omega_+}{4\sqrt{\pi}\mathbf{\Gamma}(1+\gamma)} \left(\frac{K\omega_+}{2i}\right)^{2\gamma-1}$$

The quantities D and χ_r^{ij} have naturally no dimension. For this reason, the dimensionless version of the equations (2) and (6)-(9) is identical to the original one, provided that Γ is measured in unities of $\hbar v_F$ and the Green's functions in the inverse, i.e. $(\hbar v_F)^{-1}$.

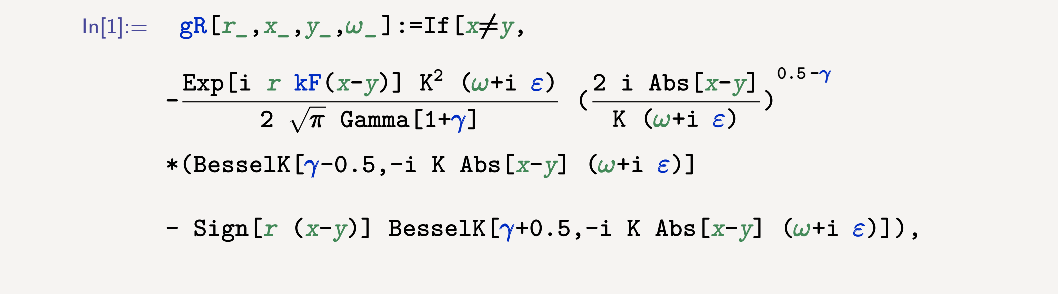
Parameters. – With the above mentioned simplification, the list of free physical parameters reduces to L, k_F, Γ and K. Additionally, our program will contain a purely technical parameter ϵ and such derived quantities as $\gamma = (K + K^{-1} - 2)/4$ and $x_{1,2} = \pm L$.

3 Numerical methods

3.1 Naive approach

Since Mathematica supports nested functions[1], the problem can be solved by direct inputting of the above formulas. With special functions `Gamma[ ]`, `BesselK[ ]`, `KroneckerDelta[ ]` and an auxiliary function `bar[i_]:=(If[i==1,Return[2]];1)`, the task is straightforward. Obviously, regardless it's transparency, the resulting program is extremely inefficient. The execution time t_{exec} per 2D-graph of the LDOS is $t_{exec} \sim 27$ min.

Nevertheless, at this stage we have prepared the basis for the further work on the code and defined functions that will be henceforth conserved, in particular the function $g_r^R(x, y; \omega)$:



```
In[1]:=  gR[r_,x_,y_,ω_]:=If[x≠y,

          Exp[i r kF(x-y)] K² (ω+i ε)    2 i Abs[x-y]  0.5-γ
        - ───────────────────────────  (─────────────)
             2 √π Gamma[1+γ]              K (ω+i ε)

        *(BesselK[γ-0.5,-i K Abs[x-y] (ω+i ε)]

        - Sign[r (x-y)] BesselK[γ+0.5,-i K Abs[x-y] (ω+i ε)]),
```

[1] Here, this term is to be understood fist of all as functions involving the other exteriorly defined ones.

```
  (Gamma[0.5-γ] K^2 (ω+i ε)) (-0.5 i K (ω+i ε))^(2 γ-1)
 -─────────────────────────────────────────────────────]
             4 √π Gamma[1+γ]
```

3.2 The main algorithm

Thanks to the self-consistency of the Eq. (2), the retarded Green's functions, $G^R_{r,r}(x_i, x'; \omega)$ and $G^R_{-r,r}(x_i, x'; \omega)$, may be computed by solving a system of linear equations. Indeed, applying Eq. (2) to $x = x_i$, $i = 1, 2$, one obtains, after some algebra, a matrix equation:

$$MX = B \tag{10}$$

where M is an 8x8 block matrix:

$$M = \begin{pmatrix} 1-\Gamma g^R_+(x_1,x_1;\omega) & -\Gamma g^R_+(x_1,x_1;\omega) & -\Gamma g^R_+(x_1,x_2;\omega) & -\Gamma g^R_+(x_1,x_2;\omega) \\ -\Gamma g^R_-(x_1,x_1;\omega) & 1-\Gamma g^R_-(x_1,x_1;\omega) & -\Gamma g^R_-(x_1,x_2;\omega) & -\Gamma g^R_-(x_1,x_2;\omega) \\ -\Gamma g^R_+(x_2,x_1;\omega) & -\Gamma g^R_+(x_2,x_1;\omega) & 1-\Gamma g^R_+(x_2,x_2;\omega) & -\Gamma g^R_+(x_2,x_2;\omega) \\ -\Gamma g^R_-(x_2,x_1;\omega) & -\Gamma g^R_-(x_2,x_1;\omega) & -\Gamma g^R_-(x_2,x_2;\omega) & 1-\Gamma g^R_-(x_2,x_2;\omega) \end{pmatrix} \otimes \mathbb{1}_2$$

$\mathbb{1}_2$ is the two-dimensional identity matrix, X an eight-component column-vector: $X(x';\omega) = {}^t\Big(G^R_{+,+}(x_1,x';\omega), G^R_{+,-}(x_1,x';\omega), G^R_{-,+}(x_1,x';\omega), G^R_{-,-}(x_1,x';\omega)$, $G^R_{+,+}(x_2,x';\omega), G^R_{+,-}(x_2,x';\omega), G^R_{-,+}(x_2,x';\omega), G^R_{-,-}(x_2,x';\omega)\Big)$ as well as the second member B given by: $B(x',\omega) = {}^t\Big(g^R_+(x_1,x';\omega), 0, 0, g^R_-(x_1,x';\omega), g^R_+(x_2,x';\omega), 0, 0,$ $g^R_-(x_2,x';\omega)\Big)$.

The Eq. (10) can be easily resolved by means of `LinearSolve[ ]` function, provided that $\det(M) \neq 0$. Another way of solving (10) is to invert the matrix M and then multiply B by the inverse ($X = M^{-1}B$). However, this is not as efficient as using `LinearSolve[ ]`, according to Wolfram documentation.
With these improvements, the execution time is divided by four, compared to t_{exec} of the first method, i.e. $t_{exec} \sim 7$ min.

Alternatively, Eq. (10) can be solved by diagonalizing the matrix M into $D = P^{-1}MP$, where P is the change of basis matrix. With $X' := P^{-1}X$ and $B' := P^{-1}B$, Eq. (10) becomes $DX' = B'$. The solution is then given by: $X = PX' = P(PD)^{-1}B$. The diagonalization procedure can be performed with the help of `Eigenvalues[ ]` and `Eigenvectors[ ]` functions:

```
In[2]:=  matriceD[ω_]:=DiagonalMatrix[Eigenvalues[M[ω]]];
         matriceP[ω_]:=Transpose[Eigenvectors[M[ω]]];
```

or (preferentially) with `Eigensystem[ ]`:

```
In[3]:=  DPm[ω_]:=Eigensystem[M[ω]];
         matriceD[ω_]:=DiagonalMatrix[DPm[M[ω]][[1]]];
         matriceP[ω_]:=Transpose[DPm[M[ω]][[2]]];
```

In both cases, the vector X (called `GSet` in the program), may be computed by the two equivalent ways, either as:

```
In[4]:=  GSet[x_,ω_]:=matriceP[ω].LinearSolve[matriceD[ω],
           Inverse[matriceP[ω]].B[x,ω]]
```

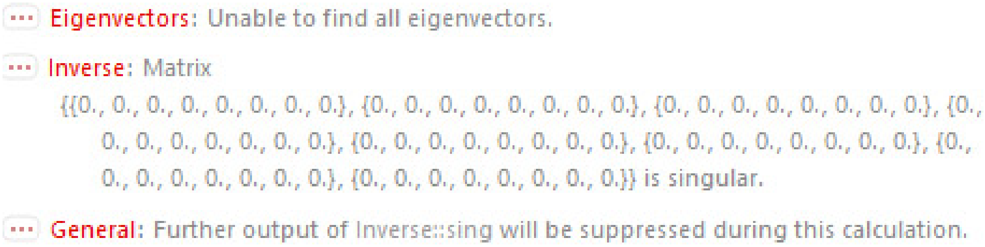
or:

```
In[5]:=  GSet[x_,ω_]:=matriceP[ω].Inverse[matriceP[ω].matriceD[ω]].B[x,ω]
```

Notice that in the numerical computation, because of the finite precision, M is expected a priori to be non-degenerate.

3.3 Problem encountered with `Eigen`-family functions

The program built with `Eigen`-family functions issues a sequence of error messages. Among them:



```
··· Eigenvectors: Unable to find all eigenvectors.

··· Inverse: Matrix
      {{0., 0., 0., 0., 0., 0., 0., 0.}, {0., 0., 0., 0., 0., 0., 0., 0.}, {0., 0., 0., 0., 0., 0., 0., 0.}, {0.,
         0., 0., 0., 0., 0., 0., 0.}, {0., 0., 0., 0., 0., 0., 0., 0.}, {0., 0., 0., 0., 0., 0., 0., 0.}, {0.,
         0., 0., 0., 0., 0., 0., 0.}, {0., 0., 0., 0., 0., 0., 0., 0.}} is singular.

··· General: Further output of Inverse::sing will be suppressed during this calculation.
```

Rewriting the concerned part of the code in a slightly different manner:

```
In[6]:=  matriceD[m_]:=DiagonalMatrix[Eigenvalues[m]];
         matriceP[m_]:=Transpose[Eigenvectors[m]];
         Sol[p_,d_,b_]:=p.LinearSolve[d,b];
         GSet[x_,ω_]:=Sol[matriceP[M[ω]],matriceD[M[ω]],Inverse[
         matriceP[M[ω]]].B[x,ω]]
```

turns out to be helpless because of it's equivalence to the previous one.
Numerical evaluation of both `GSet[x,ω]` and final result `ρ[x,ω]` in an arbitrary point[2] (x, ω) can still be performed (without getting any warning message):

```
In[7]:=  GSet[100.,0.005]
         Dimensions[GSet[100.,0.005]]
         ρ[100.,0.005]
```

```
Out[7]=  {0.1874591152531146+0.9872730529348801 i,
         -0.0840573000078053-0.05400824435728041 i,...
         ...,0.9322326482409091+0.37575964511326926 i}
         {8}
         0.29032764489593443
```

Moreover, `ρ[x,ω]` has been successfully plotted using the inappropriate for our goal `MatrixPlot[ ]` function (the suitable one is `DensityPlot[ ]`). The problem must

[2] `Random[ ]` function generating arbitrary $-L \leq x \leq L$ and $\omega_{min} \leq \omega \leq \omega_{max}$ has been used to check this 15 times.

arise from the fact that whilst `LinearSolve` works on both numerical and symbolic matrices, `Eigen`-family functions work on numerical ones (according to Wolfram documentation). It is less likely – but not excluded – that the problem is one of the Mathematica's bug.
Other error messages (too long to be reported here) suggest to avoid the syntax `function[arguments][[index]]`. All symbolic expressions involving `GSet[x,ω][[i]]` become actually problematic in the presence of `Eigenvectors` and `Eigensystem`, which is not the case with the pure `LinearSolve` method described at the beginning of § 3.2.

3.4 Optimisation

Taking into account the previous observations, the program needs to be restructured in order to:

- Get rid of the `function[arguments][[index]]` syntax.
- Minimise memory and time consuming redundancies of all nature, especially repeated invoking of functions for their evaluation at the same point.

With this we expect the symbolic computation to reduce in favour of numerical one and therefore the problem of `Eigen`-family functions to be resolved. The `Module[]` is an expedient routine to achieve this. We use it to set up local variables in the body of the functions.
First of all, we optimize the definition of the matrix M. We replace the heavy and inefficient expression:

```
In[8]:= p11[ω_]:=Γ gR[1,x1,x1,ω]; p12[ω_]:=Γ gR[1,x1,x2,ω];
        p21[ω_]:=Γ gR[1,x2,x1,ω]; p22[ω_]:=Γ gR[1,x2,x2,ω];
        m11[ω_]:=Γ gR[-1,x1,x1,ω]; m12[ω_]:=Γ gR[-1,x1,x2,ω];
        m21[ω_]:=Γ gR[-1,x2,x1,ω]; m22[ω_]:=Γ gR[-1,x2,x2,ω];
        M[ω_]:=
        {{1-p11[ω],0,-p11[ω],0,-p12[ω],0,-p12[ω],0},
        {0,1-p11[ω],0,-p11[ω],0,-p12[ω],0,-p12[ω]},
        {-m11[ω],0,1-m11[ω],0,-m12[ω],0,-m12[ω],0},
        {0,-m11[ω],0,1-m11[ω],0,-m12[ω],0,-m12[ω]},
        {-p21[ω],0,-p21[ω],0,1-p22[ω],0,-p22[ω],0},
        {0,-p21[ω],0,-p21[ω],0,1-p22[ω],0,-p22[ω]},
        {-m21[ω],0,-m21[ω],0,-m22[ω],0,1-m22[ω],0},
        {0,-m21[ω],0,-m21[ω],0,-m22[ω],0,1-m22[ω]}};
```

by a more compact and resource saving one:

```
In[9]:= M[ω_]:=Module[{C1,C2,A},
        C1={gR[1,x1,x1,ω],gR[-1,x1,x1,ω],gR[1,x2,x1,ω],gR[-1,x2,x1,ω]};
        C2={gR[1,x1,x2,ω],gR[-1,x1,x2,ω],gR[1,x2,x2,ω],gR[-1,x2,x2,ω]};
        A=ConstantArray[0,{4,4}];
        A[[All,1]]=A[[All,2]]=C1;A[[All,3]]=A[[All,4]]=C2;
        KroneckerProduct[IdentityMatrix[4]-Γ*A,IdentityMatrix[2]]]
```

Notice the use of `KroneckerProduct[ ]` instead of `TensorProduct[ ]`: only in this way the resulting matrix M is properly shaped. Similarly, we use `Module` to rebuild the function ρ`[x,ω]`:

```
In[10]:= ρ[x_,ω_]:=
         Module[{B,evals,evecs,matriceD,matriceP,GSet,Gpp,Gmp,Gpm,Gmm},
         B={gR[1,x1,x,ω],0,0,gR[-1,x1,x,ω],gR[1,x2,x,ω],0,0,gR[-1,x2,x,ω]};
         {evals,evecs}=Eigensystem[M[ω]];
         matriceD=DiagonalMatrix[evals];
         matriceP=Simplify[Transpose[evecs]];
         GSet=matriceP.LinearSolve[matriceD,Inverse[matriceP].B];

         Gpp=gR[1,x,x,ω]+Γ gR[1,x,x1,ω] (GSet[[3]]+GSet[[1]])
         +Γ gR[1,x,x2,ω] (GSet[[5]]+GSet[[7]]);

         Gmp=Γ gR[-1,x,x1,ω] (GSet[[3]]+GSet[[1]])
         +Γ gR[-1,x,x2,ω] (GSet[[5]]+GSet[[7]]);

         Gpm=Γ gR[1,x,x1,ω] (GSet[[4]]+GSet[[2]])
         +Γ gR[1,x,x2,ω] (GSet[[6]]+GSet[[8]]);

         Gmm=gR[-1,x,x,ω]+Γ gR[-1,x,x1,ω] (GSet[[4]]+GSet[[2]])
         +Γ gR[-1,x,x2,ω] (GSet[[6]]+GSet[[8]]);

          Im[Gpp]+Im[Gmp]+Im[Gpm]+Im[Gmm]
         -─────────────────────────────────]
                        π
```

The obtained program is advantageously fast. With the command:

```
In[11]:= Timing[fig=DensityPlot[ρ[x,ω],{x,-L,L},{ω,ωmin,ωmax},
         PlotPoints→100,ColorFunction→ColorData["BlueGreenYellow"],
         PlotLegends→Automatic,FrameLabel→{"x/a",ω/ωc}]]
         Export["fig_K_0.7_gamma_10_esp_1_em6.png",fig]
```

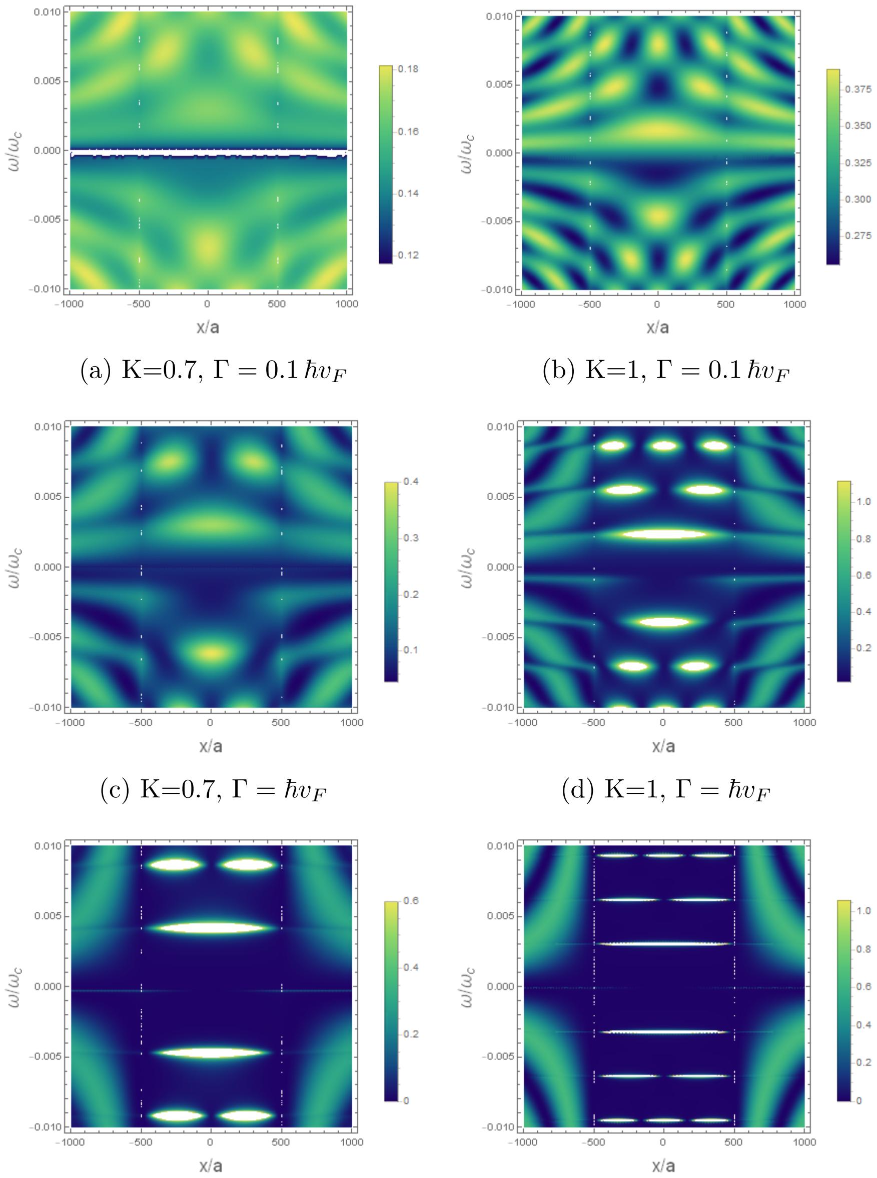
the LDOS graph is generated in $t_{exec} \sim 3$ min! Unfortunately, during the execution, we get (almost) the same error messages as previously, but this does not prevent us from getting the correct result.

4 Results and discussion

We consider the quantum wire with two impurities located at positions $x_1, 2 = \pm L/2$ with $L = 1000a$. At first we take $k_F = 0$. We plot the two-dimensional profile of the density of states as a function of position and energy in the region $[-L,\, L] \times [-0.01\omega_c,\, 0.01\omega_c]$. We study its response to the increasing impurity potentials in interacting and non interacting regime controlled by the parameter K.

The main algorithm (simple `LinearSolve` method) and its optimized version produce identical results. The convergence is obtained for $\epsilon < 10^{-5}$. One also need to adjust the number of initial samples to be used in each direction by specifying the option `PlotPoints` in `DensityPlot[ ]` function. Results obtained with $\epsilon = 10^{-6}$ and `PlotPoints` → `100` (see figure 2) are in accordance with the expected LDOS profiles.

The naïve method generates the expected result only in the regime of weak impurities ($\Gamma \sim 0.1\,\hbar v_F$). In the intermediate and strong impurity regime there appears regions with negative DOS that reveals the existence of an error in the code. Since the error has never been identified, this version of code was finally abandoned.



(a) K=0.7, $\Gamma = 0.1\,\hbar v_F$

(b) K=1, $\Gamma = 0.1\,\hbar v_F$

(c) K=0.7, $\Gamma = \hbar v_F$

(d) K=1, $\Gamma = \hbar v_F$

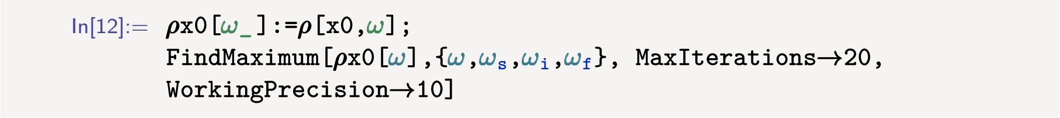
(e) K=0.7, $\Gamma = 10\,\hbar v_F$

(f) K=1, $\Gamma = 10\,\hbar v_F$

FIG. 2: LDOS in the presence of Coulomb interactions ($K = 0.7$) and for a non interacting wire ($K = 1$) as a function of position and energy.

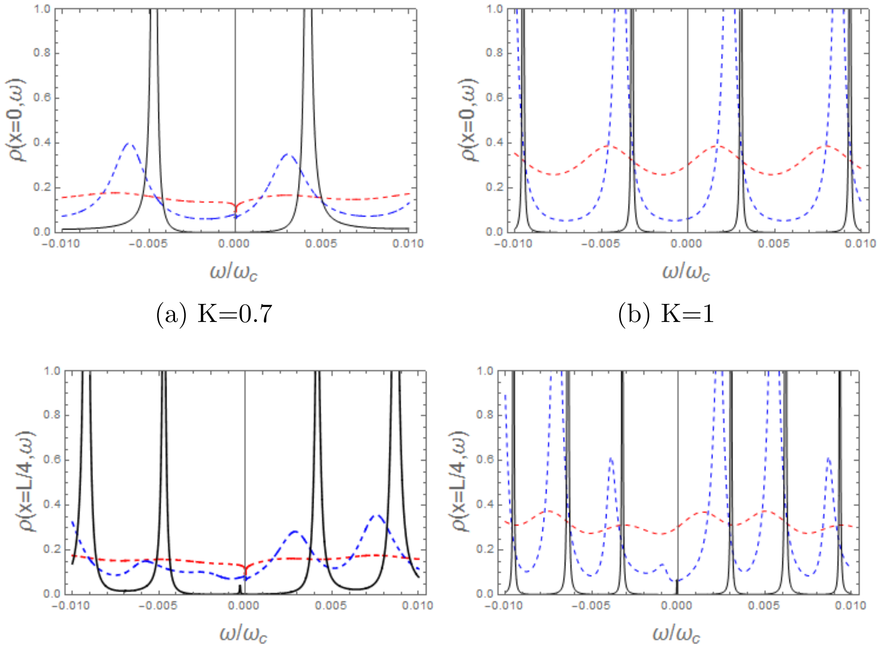
As illustrated in figures 2b and 3b, for small-impurity potentials ($\Gamma \lesssim 0.1\,\hbar v_F$), the LDOS of the non interacting QW is odd in energy. More precisely, $\rho(x,\omega) - \rho_0$ with $\rho_0 \simeq 0.32$) is odd on ω for all x. This fact can be anticipated analytically threw a perturbative expansion of Eq. (2) for $K = 1$ [1]. This is not rigorously the case for the interacting wire. Though it tends towards the same behaviour but with a smaller ρ_0 (see figure 2a).

In the intermediate impurity regime ($\Gamma \sim \hbar v_F$), in both cases the LDOS profile is neither odd, nor even whilst it becomes even for strong-impurity potentials. Moreover, in this regime ($\Gamma \sim 10\,\hbar v_F$), energy levels become discrete in the central region of the wire (between impurities). This is characteristic to the Coulomb-blockade regime. The weaker the interaction is ($K \to 1$), the thinner, wider and higher the corresponding density peaks are. Its periodicity in energy is determined by the ratio $\pi v/L$ with $v = v_F/K$, which is characteristic to a finite size wire of length L [1]. This has been checked numerically by plotting LDOS as a function of energy at fixed position x_0 and by finding the maxima of the function $\rho(x_0, \omega)$, using the command:



```
In[12]:=  ρx0[ω_]:=ρ[x0,ω];
          FindMaximum[ρx0[ω],{ω,ωs,ωi,ωf}, MaxIterations→20,
          WorkingPrecision→10]
```

(`FindMaximum[ ]` searches for a local maximum from the starting point ω_s, stopping the search if ω ever gets outside the range ω_i to ω_f). As shown in figure 3, for $x = L/4$ all peaks at energy levels $n\pi v/L$ with $n \neq 0$ an integer are visible. For $x = 0$, only peaks corresponding to an odd n are present.
LDOS has also been plotted as a function of position at two fixed frequencies, $\omega_0 = 0.01\omega_c$ and ω_0 corresponding to the first maximum of $\rho(0, \omega)$ when $\Gamma = 10\hbar v_F$ (value close to $\pi v/L$). The obtained graphs (see figure 4) mainly show that interactions reduce the amplitude of the density oscillations and that LDOS is zero on the impurities sites.



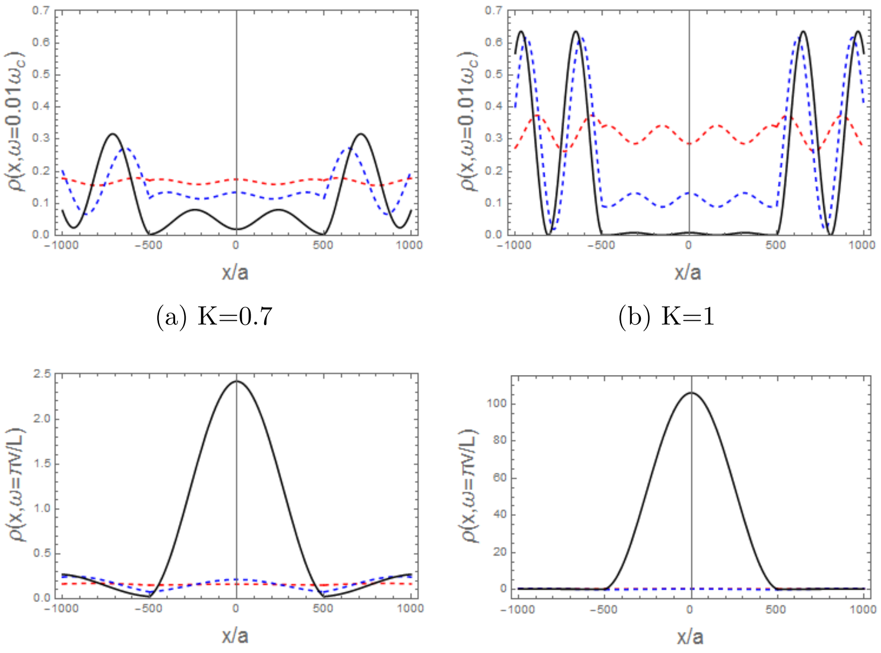
(a) K=0.7

(b) K=1

(c) K=0.7

(d) K=1

FIG. 3: LDOS in the presence of Coulomb interactions ($K = 0.7$) and for a non interacting wire ($K = 1$) as a function of energy at two different positions, $x = 0$ (upper graphs) and $x = L/4$ (bottom graphs), for $\Gamma = 0.1\,\hbar v_F$ (red dashed lines), $\Gamma = \hbar v_F$ (blue dashed lines) and $\Gamma = 10\,\hbar v_F$ (black solid lines).



(a) K=0.7

(b) K=1

(c) K=0.7

(d) K=1

FIG. 4: LDOS in the presence of Coulomb interactions ($K = 0.7$) and for a non interacting wire ($K = 1$) as a function of position for two different frequences, $\omega = 0,01\omega_c$ (upper graphs) and $\omega \approx \pi v/L$ (bottom graphs), for $\Gamma = 0.1\,\hbar v_F$ (red dashed lines), $\Gamma = \hbar v_F$ (blue dashed lines) and $\Gamma = 10\,\hbar v_F$ (black solid lines).

Lastly, let us mention that LDOS peaks disappear completely for $K \leq 0.4$ and that in the case $k_F \neq 0$, LDOS exhibits a strong oscillatory behaviour.

5 Conclusion

We built the two Mathematica programs that correctly calculate the density of states for an interacting quantum wire in presence of two impurities of equal potential strength. We worked at large distances and energies range. We explored different methods of solving a system of linear equations. Eigensystem[ ] function is the most efficient in terms of the execution time, but may be problematic when applied to heavy symbolic expressions. The code was optimized by means of `Module[ ]` function. The importance of `PlotPoints`, `MaxRecursion`, `Maxterations` has also been noticed. From the physical point of view, we observed that the main effect of interactions is to reduce the amplitude of the (Fabry-Perot) oscillations in the weak impurity regime and the Coulomb-blockade peaks in the strong-impurity regime. Further systematic calculations based on `FindMaximum[ ]` function need to be implemented in order to perform a more qualitative analysis of the evolution of LDOS peaks amplitude and spacing with the interaction strength K.

Acknowledgments

I would like to thank my supervisor, Adeline Crépieux, for her guidance, encouragement and helpful correspondence. This work has been carried out thanks to the support of the A∗MIDEX grant (№ANR-11-IDEX-0001-02) funded by the French Government “Investissements d’Avenir” program.

References

[1] R. Zamoum, M. Guigou, C. Bena, and A. Crépieux, “Density of states of interacting quantum wires with impurities: A Dyson equation approach,” vol. 90, p. 085408, Aug. 2014.

[2] B. Braunecker, C. Bena, and P. Simon, “Spectral properties of Luttinger liquids: A comparative analysis of regular, helical, and spiral Luttinger liquids,” vol. 85, p. 035136, Jan. 2012.