1 Introduction

The program calculates the transmission in the L-C or C-R interface. The system the program can calculate is written in the top part of Fig.1. The effects of semi-infinite electrodes, L or R, is calculated using the surface Green function, G_L or G_R. Then, the program takes into account of The effects electrode to the C region via self-energy Σ_L and Σ_R. In this way, the program solves the system shown in the bottom of Fig.1 which is equivalent to the top of the figure. The program can also apply the source-drain bias between L and R.

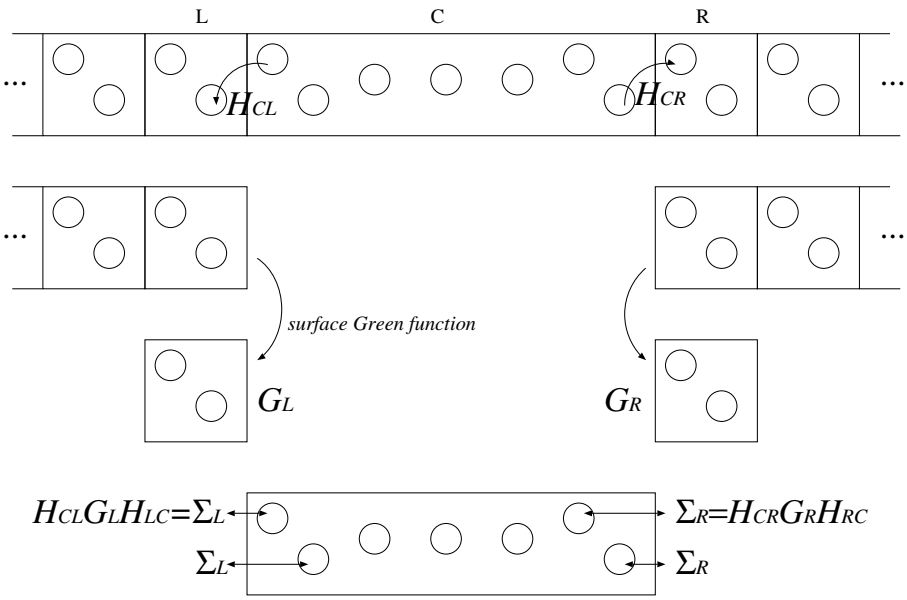

Figure 1:

2 Surface Green functions

What is necessary to calculate is Hamiltonian and overlap matrixes of electrodes, L and R. They are H00_e[side], S00_e[side] , H01_e[side] and S01_e[side] , side=0 for L and side=1 for R.

For L, H01_e[side] and S01_e[side] are transfer matrix to the left of the unit cell. For R, H01_e[side] and S01_e[side] are transfer matrix to the right of the unit cell. See also Fig.2.

The calculated Green functions are G_L and G_R.

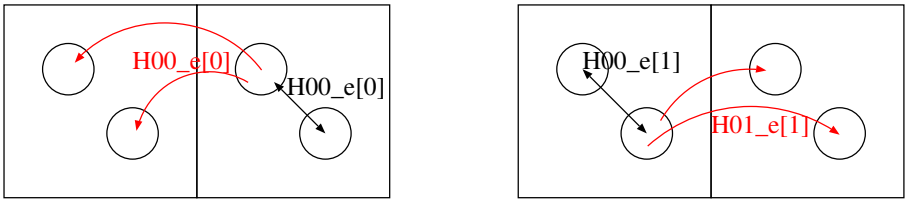

Figure 2:

3 Green function in the central region

What is necessary to calculate is Hamiltonian and overlap matrixes of the central region and to the electrodes, L and R. They are HCC, SCC, HCL, SCL and HCR and SCR. And the surface Green functions of the electrodes, G_L and G_R are necessary, of course.

The Green function of the central region, G, is

$$\begin{aligned} G(\omega) &= \{\omega S_{CC} - H_{CC} - \Sigma_R(\omega) - \Sigma_L(\omega)\}^{-1} && (1) \\ \Sigma_L(\omega) &= H_{CL} G_L(\omega) H_{LC} && (2) \\ \Sigma_R(\omega) &= H_{CR} G_R(\omega) H_{RC} && (3) \end{aligned}$$

The Lesser Green function of the central region is

$$G^< = G^R \Sigma^< G^A \simeq G^R(-(\Sigma_L^R - \Sigma_L^A) f(\omega - \mu_L) - (\Sigma_R^R - \Sigma_R^A) f(\omega - \mu_R)) G^A$$

where $G^R(\omega) = G(\omega + i\delta)$ and $G^A(\omega) = G(\omega - i\delta)$.

The Density matrix is evaluated as

$$\rho_{ij} = \mathrm{Im} \int_\infty^{E_1} d\omega (\frac{-1}{\pi}) G^R(\omega) + \int_{E_1}^\infty d\omega \frac{1}{2\pi i} (G_{ij}^<(\omega))$$

Analytic continuation is possible for the former integral.

$$\mathrm{Im} \int_\infty^{E_1} d\omega (\frac{-1}{\pi}) G^R(\omega) = \mathrm{Im} \int_C^{E_1} d\omega (\frac{-1}{\pi}) G(\omega)$$

The final expression is

$$\rho_{ij} = \mathrm{Im} \int_C^{E_1} d\omega (\frac{-1}{\pi}) G(\omega) + \int_{E_1}^\infty d\omega \frac{1}{2\pi i} (G_{ij}^<(\omega))$$

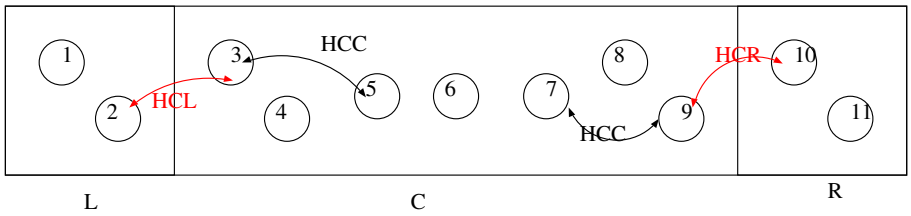

Figure 3:

4 Transmission

Once the Hamiltonian and overlap matirxes are solved. You can calculate transmission for any ω.

$$J = -\frac{e}{h} \int d\omega \mathrm{Tr}[(f(\omega - \mu_L) - f(\omega - \mu_R))(\Gamma_L G_C^R \Gamma_R G_C^A)]$$

where $\Gamma_L = i(\Sigma_L^R - \Sigma_L^A)$ and $\Gamma_R = i(\Sigma_R^R - \Sigma_R^A)$

5 Overlapping region

5.1 $\rho_{ij} \rightarrow \rho(r)$

$$\rho(r) = \sum_{ij} \phi_i(r)\rho_{ij}\phi_j(r)$$

The secular equation is solved only in the C region. It means the indexes i and j for ρ_{ij} that you can solve is in the C region. There exists ovelap between L and C region, however you can not include such effects. Thus you can not calculate $\rho(r)$ near the L region. Therefore the program employs the charge density $\rho(r)$ of the L region for such parts.

5.2 $\phi_i \rightarrow H$

The program define the potential, or the charge density, only in the C region. Thus, if the wavefunction ϕ_i or ϕ_j spans outside the C region, the program can not define the Hamiltonian matrix $\langle\phi_i|H|\phi_j\rangle$. Therefore the program use the Hamiltonian or overlap matrix of the L region for such parts.

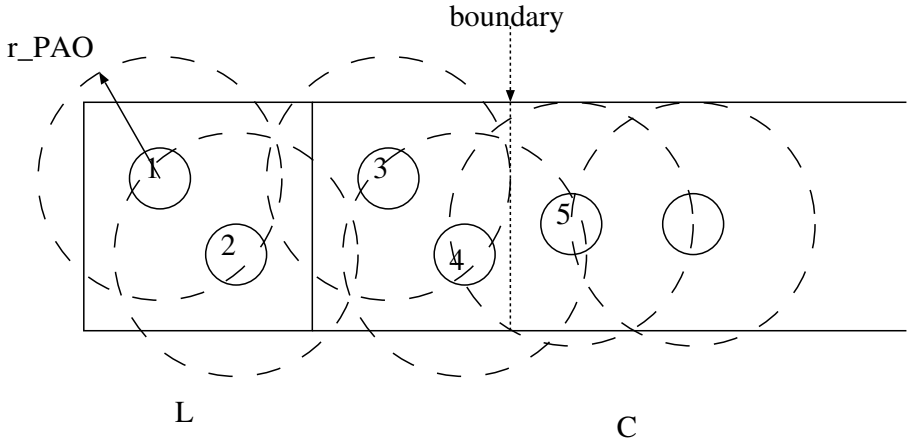

Figure 4: To evaluate $\rho(r)$ near the L region, the wavefunction ϕ_2 and the density matrix ρ_{2i} are necessary. Thus the program can not evaluate $\rho(r)$ whose r is left of the boundary. To evaluate $\langle\phi_3|H|\phi_3\rangle$, the potential in the L region is necessay.

6 Poisson equation

The program solves

$$\triangle V_H(r) = -4\pi\rho(r)$$

It is solved in this way.

$$\triangle(V_C(r) + V_B(r)) = -4\pi(\rho(r) + 0) \quad (4)$$

$$\triangle V_C(r) = -4\pi\rho(r) \quad (5)$$

$$\triangle V_B(r) = 0 \quad (6)$$

$\triangle V_C(r) = -4\pi\rho(r)$ is solved via FFT as a periodic system. $\triangle V_B(r) = 0$ is solved with the boundary condition.

$V(r)$ are transformed to $V(G_\parallel, z)$. Now you have $V_C(G_\parallel, z_i)$ in the C region, $i = 1, 2, ..., l$. The bounary condition is $V_B(G_\parallel, z_0)$ and $V_B(G_\parallel, z_{l+1})$ The correction to the $V_C(G_\parallel, z_i)$, $i = 0$ and $l + 1$, is $dV_H(G_\parallel, z_i) = V_B(G_\parallel, z_i) - V_C(G_\parallel, z_i)$.

6.1 $G_\parallel \neq 0$

$$\begin{aligned} V_B(r) &= \int dG_\parallel e^{iG_\parallel r_\parallel} \left\{ dV_H(G_\parallel, z_{l+1}) \left(e^{-G_\parallel (z_{l+1}-z)} - e^{-G(z_{l+1}-2z_0+z)} \right) \right. \\ &\left. + dV_H(G_\parallel, z_0) \left(e^{-G_\parallel (z-z_0)} - e^{-G(2z_{l+1}-z_0-z)} \right) \right\} / \left(1 - e^{-2G_\parallel (z_{l+1}-z_0)} \right) \end{aligned} \tag{7}$$

6.2 $G_\parallel = 0$

Here $G_0 = 0$.

$$\triangle V_B(G_0, z) = 0 \tag{8}$$

$$V_B(G_0, z) = a(z - z_0) + b \tag{9}$$

a and b is determined from

$$V_B(G_0, z_0) = dV_H(G_0, z_0) + b \tag{10}$$

$$V_B(G_0, z_{l+1}) = dV_H(G_0, z_{l+1}) + a(z_{l+1} - z_0) + b \tag{11}$$

A Interpolation

$\rho(r)$ and $V(r)$ are evaluated on the FFT grid. However, the FFT grid of the C region is different from that of the electrode, because the unit cell is different. The value must be interpolated.

An input data set

$$H((\gamma + n)/N_{in})$$

where $n = 0, N_{in} - 1$ and assuming the periodic boundary bondition. The interpolated data set

$$H((\alpha + \beta m)/N_{out})$$

where m is integer and can be beyond the range of [0:$N_{out} - 1$]

Forward transformation

$$h(k) = \sum_{n=0,N_{in}-1} H(n+\gamma) \exp(-2\pi i k(n+\gamma)/N_{in}) \tag{12}$$

$$= \exp(-2\pi i k\gamma/N_{in}) \sum_{n=0,N_{in}-1} H(n+\gamma) \exp(-2\pi i k n/N_{in}) \tag{13}$$

$$= \exp(-2\pi i k\gamma/N_{in}) h^0(k) \tag{14}$$

where

$$h^0(k) = \sum_{n=0,N_{in}-1} H(n+\gamma) \exp(-2\pi i k n/N_{in})$$

can be calculated via FFT.

Barkward transformation of $H((\alpha + \beta n)/N_{out})$.

$(\alpha + \beta n)/N_{out}$ can be beyound the range of [0:1].

$$H'(\alpha + \beta n) = \sum_{k=0,N_{in}-1} h(k) \exp(2\pi i k(\alpha + \beta n)/N_{out}) \tag{15}$$

$$= \sum_{k=0,N_{in}-1} \exp(-2\pi i k\gamma/N_{in}) h^0(k) \exp(2\pi i k(\alpha + \beta n)/N_{out}) \tag{16}$$

$$= \sum_{k=0,N_{in}-1} h^0(k) \exp(2\pi i k((\alpha + \beta n)/N_{out} - \gamma/N_{in})) \tag{17}$$

Note, the transformation bove can not give all of $(\alpha+\beta n)/N_{out}$ correctly. In order to interpolate data, use the formula below

$$H(\alpha+\beta n) = \sum_{k=0,N_{in}/2} h^0(k)\exp(2\pi ik((\alpha+\beta n)/N_{out}-\gamma/N_{in})) \quad (18)$$

$$+ \sum_{k=-N_{in}/2,-1} h^0(k)\exp(-2\pi ik((\alpha+\beta n)/N_{out}-\gamma/N_{in})) \quad (19)$$

where $h^0(k-N_{in})=h^0(k)$.

A Structure of the program

```
Input_std
    TRAN_Input_std
        TRAN_Set_SurfOverlap  (read data of electrode calculated previously)
    TRAN_Input_std_Atoms
truncation
    TRAN_adjust_Ngrid
    TRAN_adjust_Grid_Origin
DFT
    TRAN_Set_Electrode_Grid  (interpolate griddata of electrodes)
    TRAN_Allocate_Cregion
    scf_loop {
        Poisson | TRAN_Poisson  (solve Poisson equation with the boundary condition)
        Cluster_DFT | Band_DFT | TRAN_DFT  (calculate CDM from H and S)
        TRAN_Overwrite_Densitygrid  (overwrite the griddata of electrodes)
    }
    TRAN_Output_Trans_HS  (output H and S to calculate transmission later)
    TRAN_Deallocate_Cregion
```

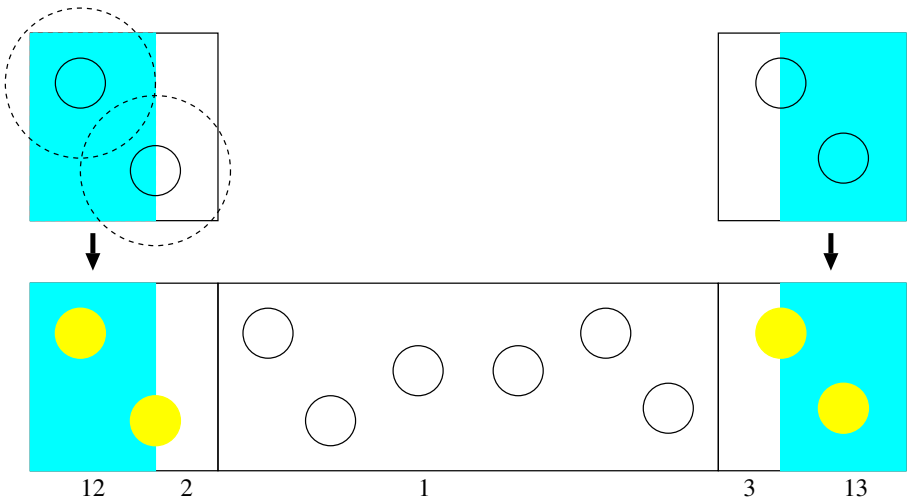

Figure 5: Electrode region = [0:TRAN_grid_bound[0]] for left, [TRAN_grid_bound[0],Ngrid1-1]. They are shown in blue. TRAN_region[1:atomnum] = 12 (left, yellow), 2 (left, open), 1 (center), 3 (right, open), 13 (right, yellow).

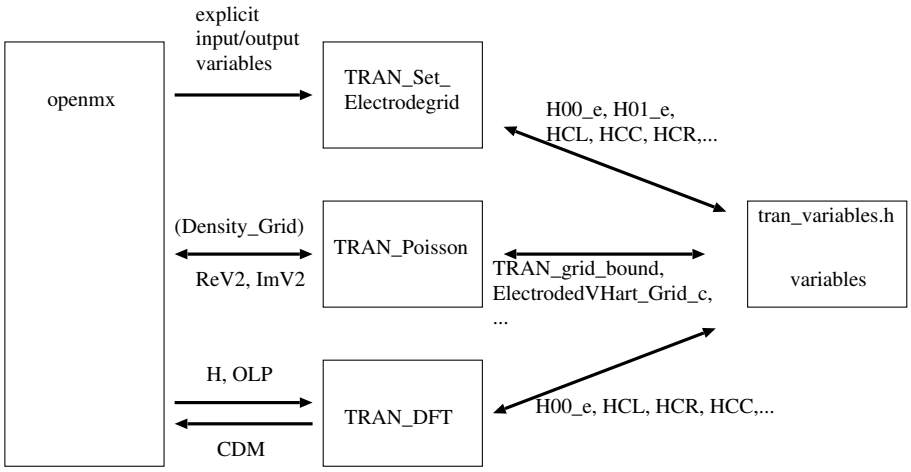
explicit
input/output
variables
openmx
TRAN_Set_
Electrodegrid
H00_e, H01_e,
HCL, HCC, HCR,...
(Density_Grid)
ReV2, ImV2
TRAN_Poisson
TRAN_grid_bound,
ElectrodedVHart_Grid_c,
...
tran_variables.h
variables
H, OLP
CDM
TRAN_DFT
H00_e, HCL, HCR, HCC,...

Figure 6: