Free Electron Tutorial for LMPG

Derek Stewart
Sandia National Laboratories
Livermore, CA
dstewar@sandia.gov

July 1, 2003

Chapter 1

Free Electron Barrier

1.1 Transmission Basics

In the lmpg code, the transport is calculated based on a layered Green's function approach which describes the system in a LMTO framework. Transmission through the system is found by first determining the Green function matrix for the device region and then calculating the coupling to the external leads. The ballistic transmission in this case is given by:

$$T = Tr\left(\Gamma_{11} G^{R}_{1N} \Gamma_{NN} G^{A}_{N1}\right) \tag{1.1}$$

where $G^{R,A}$ are the retarded and advanced Green's functions connecting the ends of the device region. The Γ terms describe the coupling to the external leads and can be expressed in terms of the left and right surface Green's functions. The trace in the expression above is over the available states in the principal layer.

1.2 The Ctrl File

We start with a simple test case, the free electron potential step. The necessary information for this run is contained in the file *ctrl.step* and an atom file *xa.step*. The *ctrl.step* file is the main source of input for lmpg. This information is divided up by keywords in the ctrl file. Many of these keywords can be used in general for all of Mark's LMTO based codes, however some are specific for the 2D layered case we will be examining. Information for many of these keywords can be found in the doc section provided with the LMTO code.

In the case of transport, we are interested primarily in the principal Green function or PGF flag. This tells the code what to do specifically for 2D systems. This flag is listed first in the ctrl file, *ctrl.step.* For this flag, various options such as MODE,SPARSE,GFOPTS, etc.. are listed. For now, we will concern ourselves only with the option MODE. MODE tells the program lmpg whether to run a self-consistent calculation (MODE=1) to determine the proper electronic structure for the interface region or whether to run in (MODE=5) to calculate transport properties. Since we are dealing with a free electron system, we do not need to worry about achieving self-consistency. Currently the MODE is set to the variable pgf, which one line above is given by $pgf = 5$. Lines with a percent sign in front are used to denote external variables. These terms can be changed at run time so the number of k-points, PGF mode, and other things can be altered quickly.

The next flag of immediate importance is BZ, which stands for Brillioun zone. This flag determines how many divisions of k points will be done in the BZ for the three directions (NKABC = nkx nky nkz). In the PGF calculations, we deal with a 2D Brillioun zone, so nkz is set to 1. In this run, the external variable $nk1$ sets the number of divisions in k_x and k_y equal to 4. The energy range of the calculation is also described in this system. The nature of the energy calculation and integration technique is specified by the EMESH parameter. The EMESH parameter has the following format

$$\text{EMESH} = \text{n\_points mode e\_start e\_end imag\_e}$$

where n_points gives the number of energy points on the contour. The EMESH mode describes what type of energy contour we use. For self-consistent calculations where the determination of the DOS is crucial, it is much faster to use a energy contour that sweeps out into the complex plane (Emesh mode = 10). E_start and e_end describe the start and end points on the energy contour. For Emesh mode = 10, imag_e determines the distribution of energy points on the complex contour. In many cases, it is useful to cluster data points near the Fermi energy.

For transport calculations, we are calculating a quantity that is only analytic on the real axis, *i.e.* the transmission coefficient. In this case, we must use a energy contour that stays as close to the real axis as possible. Energy contours nearly on the real axis are given by EMESH mode equals 1. In this case, *imag_e* describes the imaginary part of energy used for the calculations. For the free electron case, we are only interested in the transmission, so we use the real axis energy contour.

Next the types of atoms and their positions are given by the flags CLASS and SITE. The CLASS flag provides information on the atom label and atomic number. Since we are dealing with free electrons, we have only one atom type, XA, with $Z = 0$. This is just an empty sphere used to represent empty space. The SITE flag lists the position of each atom in the layered structure in terms of x,y,z coordinates and the principal layer index. In this system we have a total of 9 principal layers.

Finally, the START flag can be used to determine the number of iterations for a self-consistent calculation ($NIT = 1$) and the level of RMS convergence necessary to stop the program ($CNVG = 10^{-6}$). It can also be used to provide information on individual atoms. In the free electron case, we specify individual potential values for each atom. Here, we have a potential step ($V = 0.1$) which is 4 principal layers thick. The potential specified can be adjusted to easily match several free electron profiles.

1.3 Running the program

Assuming the lm programs have compiled successfully, we can now start to calculate some transmission results. For almost every input file used in the lm family, the structural information of the system must first be prepped by the *lmstr* program. In lm runs, we only need to use the suffix of the ctrl filename to run programs.

```
lmstr step
```

This program will send out a flurry of data to the screen and then generate two structure files associated with the ctrl file. Once this is completed, we can run lmpg and calculate the transmission through the system.

To do this, we will use the *lmpg* program which should be in the main directory of your lm distribution. Type the following:

```
lmpg -vnit=1 -vnk1=4 -vpgf=5 step > step.out &
```

The dashed terms with the letter *v* stand for variables that can be set at run time. The variables listed in the command line are nit for the number of iterations, nk1 for the number of divisions in the kx or ky BZ directions and pgf for the PGF mode. All of these variables have a default definition in the file. I have listed them here to give you practice with the variable option. For this run, all program output will be sent to a file called *step.out*. The program could take a few minutes to run. The left and right surface Green functions are calculated first for necessary energies and then stored in files for easy access. For large systems, this portion can take some time and generate large files. Once lmpg is done, we can grab information about the total transmission and the transmission in different channels.

The transmission will be calculated at three different $k_{||}$ points for a range of energies. Lmpg produces two output files which contain the relevant information about transmission for the system under study. Jz.step contains a list of different energies, the spin channel, and the real and imaginary parts of the transmission in column format.

The file jzk.step breaks down the transmission at each energy into contributions from the different $k_{||}$ points. The first column gives the energy. The second column indicates the k point index and the third and fourth columns show the real and imaginary parts of the transmission.

Let't look just at the contribution to transmission for electrons with $k_{||} = 0$. These electrons have the k index 1. We can collect them into a separate file by using the grep commmand.

```
grep ' 1 ' jzk.step > jzk.1.step
```

Using gnuplot, you can plot out the transmission as follows:

```
gnuplot prompt> plot 'jzk.1.step' using 1:3
```

or if you want the energy in eV

```
gnuplot prompt> plot 'jzk.1.step' using (13.6*$1):3
```

You will see basically no transmission for electron energies lower than the potential barrier. Once the energy of the electron is greater than the potential barrier, there are some oscillations in the transmission around 1 that are related to the quantum nature of the interaction. These oscillations are well described in standard quantum physics books. The oscillations can be described analytically and are related to the step height and step width. When you plot the transmission for the second and third k points, the jump

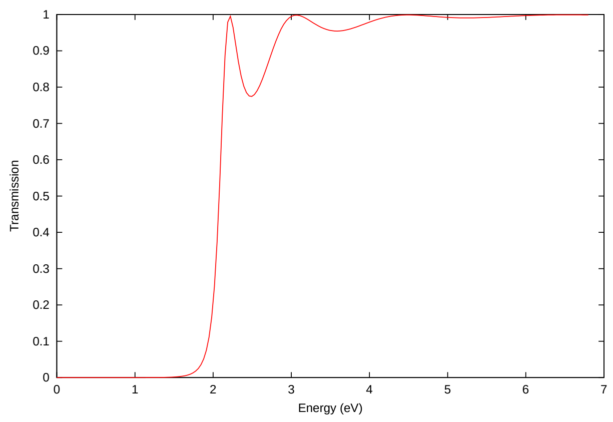

Figure 1.1: Transmission for electrons at the Γ point across the potential barrier.

up in transmission will be higher in energy. This is due to the fact that the electron is striking the potential barrier at a glancing angle. Therefore all of the electron energy is not in the z direction. For the k point 1, the electron strikes at normal incidence to the potential barrier, so all the energy is in the z direction. For the 2nd and 3rd k points, since only a fraction of the total energy is used to propagate in the z direction, a correspondingly higher total energy is required to transmit across the potential barrier.

We can also examine the total transmission across the potential barrier summed over the k points. To plot the total transmission, use the following gnuplot command:

```
gnuplot prompt> plot 'jz.step' using (13.6*$1):3
```

This will give a transmission curve which has a series of steps. Each step corresponds to the energy where an additional electron at a different k point is able to transmit over the barrier. You should be able to compare the individual transmission results for the different jzk and see a similar shift in transmission.

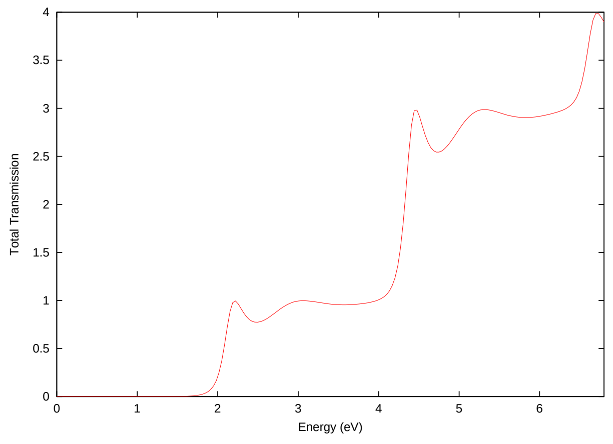

Figure 1.2: Total transmission for a potential barrier.

1.4 Adjusting the Free Electron Potential

To change the potential the free electron experiences, you will need to adjust the potentials given to the atoms below the START command in the ctrl.step file. Don't forget that the potentials are given in units of Rydbergs.

```
START NIT=1 FREE=F BEGMOM=F CNTROL=T CNVG=1D-6 RDVES=T
ATOM=XA V={0}
ATOM=XA2 V={0}
ATOM=XA3 V={0}
ATOM=XA4 V={0}
ATOM=XA5 V={0}
ATOM=XA6 V={0}
ATOM=XA7 V={0.1}
ATOM=XA8 V={0.1}
ATOM=XA9 V={0.1}
ATOM=XA10 V={0.1}
ATOM=XA11 V={0.1}
ATOM=XA12 V={0.1}
ATOM=XA13 V={0.1}
ATOM=XA14 V={0.1}
ATOM=XA15 V={0}
```

```
ATOM=XA16 V={0}
ATOM=XA17 V={0}
ATOM=XA18 V={0}
ATOM=XA19 V={0}
ATOM=XA20 V={0}
ATOM=XA21 V={0}
ATOM=XA22 V={0}
```

Beside each atom a potential is specified. To increase the potential step, just change V=0.1 to something like 0.3. The increase the width of the potential step, increase the number of atoms with a potential shift. You can also adjust the potential shape to consider a free electron model of a leaky quantum well.

I hope this helps to get you started on the general procedure for running the lmpg code. Good luck!

1.5 Contact Info

For additional information and to report bugs contact Derek Stewart at dstewar@sandia.gov