Appendix A

Extensions to Smeagol: scaling to large systems and high performance computing

In section (2.4.1) we presented the main features of the parallel version of *Smeagol*. In *Smeagol*, one needs to perform an integral over the energy (see equations (1.57) and (1.58). Numerically, this integral is discretised and becomes a sum over energy points weighted using Gauss-Legendre polynomials [101]. The Green function for a subset of energies (obtained by inverting equation (1.33)) is calculated in serial in each processor (POE method). Subsequently the partial sums for are gathered to obtain the full density matrix. This method is very efficient since the number of communication between CPUs is rather small - the Hamiltonian at the beginning of each self-consistent cycle is distributed over all the CPUs. Each CPU then works independently to calculate parts of the integrals and finally all the segments are added to obtain the total density matrix.

Although the scalability is good for the POE method, the equilibrium part of the density matrix (equation (1.57)) can be performed with a reasonably small number of points (between 50-150 depending on the system)[1] and with good precision. Therefore increasing the number of CPUs would not increase the overall speed up of the code (we could only increase the total number of energy points in the integral despite the need for more points) and we still need to perform the calculation of the Green function - a matrix inversion - in series for each energy point.

Hence the POE method only allows for parallelism up to approximately 128 processors. Albeit respectable for small clusters, the present form of parallelism implemented in Smeagol is not scalable to a higher number of processors. More

[1]he number of points for equation (1.57) is usually smaller than the one of (1.58) and therefore it defines the bottleneck of our calcualtion.

importantly, until recently great effort was placed in increasing the speed of individual processing units, but there has been a shift towards distributed architectures where the overall time for a calculation decreases by increasing the actual number of CPUs. In other words, instead of improving the performance of a single processor (a task hindered by technological barriers), computer companies are increasing the number of CPUs performing the same task. This change in paradigm is not only the realm of high performance computing, but has also made its way into desktops and laptops with the advent Duo Core technology.

Furthermore *Smeagol* needs to store in memory the Green function (and other auxiliary matrices). In DNA molecules attached to Au electrodes (see chapter 5) the number of degrees of freedom is in excess of 5,000 (5,000 × 5,000 matrices). At its present state we are reaching the limit of available memory per single compute node in state-of-the-art facilities for high performance computing.[2]

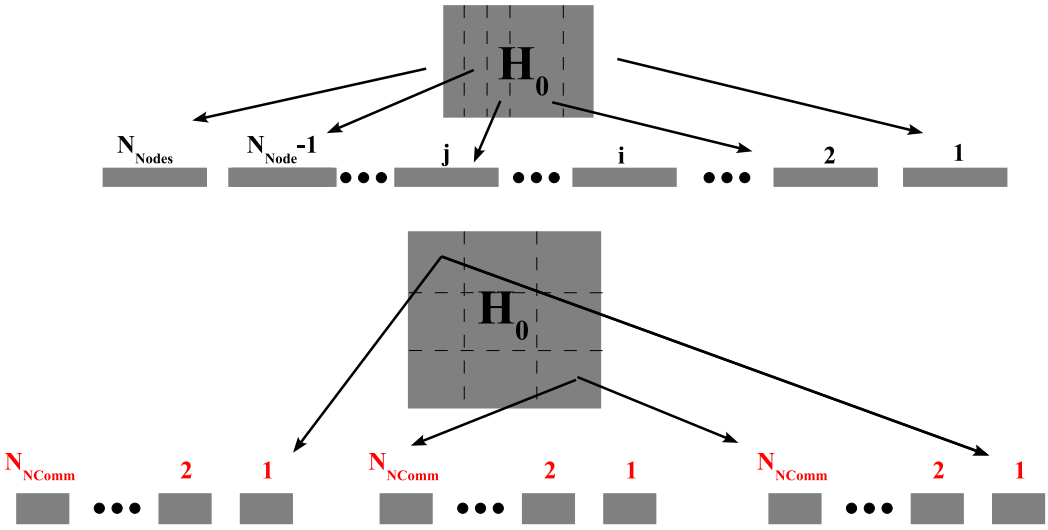

Figure A.1: Schematic represtation of redistribution of the Hamiltonian into groups of communicators. The Hamiltonian is innitially distributed over N_{Nodes} processors. It is then regrouped and redistributed N_{Comm} times over groups of CPUs comprising N_{NComm} processors each.

Facilities where a few thousand processors can be accessed at a time are becoming increasingly available to the scientific community. Numerical tools that can be used in such systems, however, remain relatively few, specially within the materials science community. Importantly, no electronic transport code as far as we know, has been

[2]Blue Gene architectures deserve special attention. In favour of low power consumption and scalability the memory available per CPU is approximately 500 Mb, *i.e.*, rather small. Therefore one must ensure that memory usage is small as possible while ensuring scalability.

designed to run in such facilities and no way of accurately treating the transport properties of large systems (in excess of 10,000 atoms) is yet available.

Charge transport in macromolecules for example present exciting pontential for future applications in molecular electronic devices. Furthermore, with the present computing hardware we can start to calculate electronic transport properties of quantum dots and the new generation of transistors from first principles. The numerical tools for such tasks, however, as mentioned eaerlier are still lacking.

A new approach is needed to deal with these two issues: 1) scalability up to thousands of CPUs and 2) memory storage of large matrices. One possible way to deal with this problem is to have a mixed scheme. In other words use the POE scheme already implemented in *Smeagol* together with the POO approach (discussed in section (2.4.1). In the latter case the orbitals are distributed over the processors without the need to store the entire matrices in one single processor. This can ensure better memory usage.

The way to do this is by using sets of communicators [146]. A communicator is nothing more than a subset of CPUs within a parallel task. Let N_{Nodes} be the number of CPUs available. Initially SIESTA distributes the Hamiltonian (the overlap matrix and the density matrix) over all processors. We then set N_{Comm} groups of CPUs, each group comprising

$$N_{\mathrm{NComm}} = \frac{N_{\mathrm{Nodes}}}{N_{\mathrm{Comm}}} \tag{A.1}$$

compute nodes.[3] The total Hamiltonian is then redistributed using the POO scheme over N_{NComm}.

Figure (A.1) shows a schematic representation of this procedure. To increase performance even further we split each group into a two-dimensional grid [147],

$$N_{\mathrm{NComm}} = N_x \times N_y \tag{A.2}$$

instead of the one-dimensional strips used by SIESTA. This ensures better scallability for parallel inversion algorithms specially if

$$N_x \sim N_y.^4 \tag{A.3}$$

This approach ensures better memory management overall. On average, we can increase memory availability (and system size) by approximately a factor N_{NComm}.

[3] We must ensure that N_{Nodes} is a multiple of N_{Comm}.

[4] The one dimensional grid is a special case with $N_y = 1$. As the number of processors increases $N_x \gg N_y$ and scalability becomes poor.

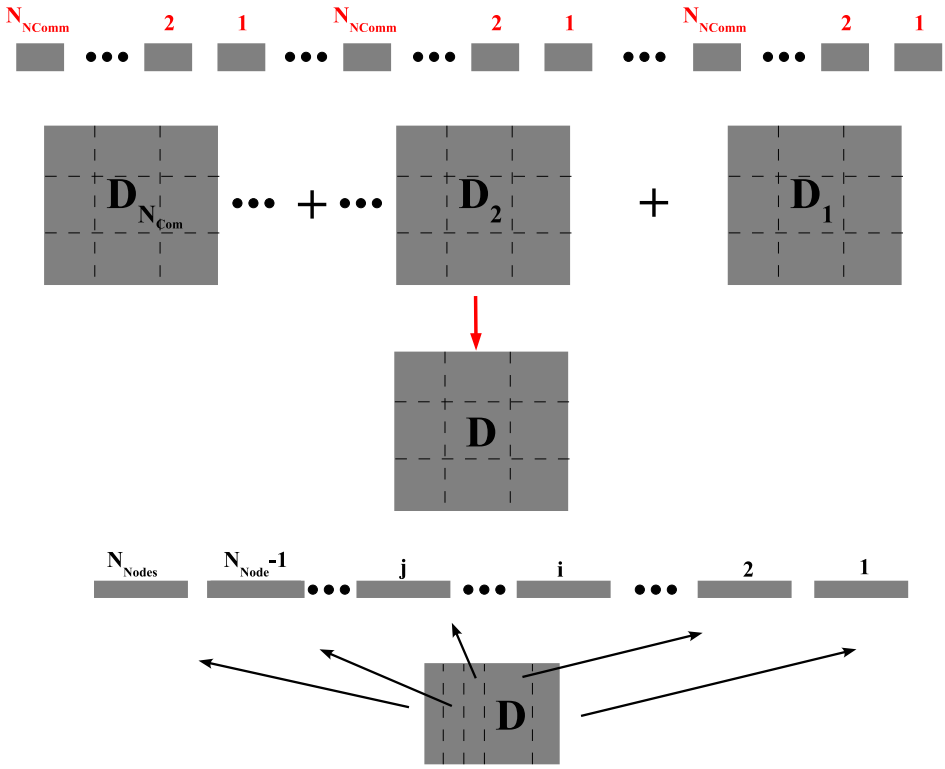

Figure A.2: Schematic represtation of the calculation of the density matrix. Each group of communicators calculates the inversion of the Green function in parallel for a different set of energies. The partial density matrices are then summed and collected to form the final density matrix which is subsequently redistributed over the initial N_{Nodes} processors.

Once this is done we can consider each set of communicators as a separate entity. Each group performs the inversion of equation (1.33) for a set of energies. Note that within each group the inversion is performed in parallel instead of in series as the standard POE method. But if we take the subset collectively the operations performed resemble the POE method. For that reason we call our new approach the mixed parallel over energy method (MPOE).

Because we work with groups of communicators each inversion is performed within N_{NComm} processors. Usually $N_{\text{NComm}} \leq 32$ processors which ensures good scalability of the inversion subroutine if we have good interconnects. Moreover, we can use the power of the POE method to scale up to 128 processors/communicators.

Therefore using the MPOE approach we can not perform calculations over

$$N_{\text{Nodes}} = N_{\text{Comm}} \times N_{\text{NComm}} \sim 128 \times 32 = 4,096 \text{ CPUs}, \tag{A.4}$$

while increasing memory availability by a factor of

$$N_{\mathrm{NComm}} \sim 32. \tag{A.5}$$

This means that while previously we could do calculations with 1,000 atoms, the MPOE method allows us to break the 10,000 atom barrier. Furthermore, with more efficient inversion algorithms one can increase N_{NComm} even more and consequently so can the system size.

The MPOE method has been implemented in *Smeagol* and tests have been performed on a 1024-processor BlueGene solution located at HPCC Edingburgh. While it is hard to compare these results with other benchmarks for *Smeagol* (see section (2.4.1)) given the peculiarities of the BlueGene architecture, the results look promising.

Appendix B

Publications stemming from this work

- Molecular-Spintronics: the art of driving spin through molecules, Stefano Sanvito and Alexandre Reily Rocha, submitted to J. Comp. Theo. Nano. Also cond-mat/0605239.

- Spin and Molecular Electronics in Atomically-Generated Orbital Landscapes, Alexandre Reily Rocha, Victor Garcia-Suarez, Steve W. Bailey, Colin J. Lambert, Jaime Ferrer and Stefano Sanvito, Phys. Rev. B. **73**, 085414 (2006), Also cond-mat/0510083.

- Towards Molecular Spintronics, Alexandre Reily Rocha, Victor Garcia-Suarez, Steve W. Bailey, Colin J. Lambert, Jaime Ferrer and Stefano Sanvito, Nature Materials **4**, 335 (2005).

- Conductance oscillations in zigzag platinum chains - suppression of parity effects, Victor Garcia-Suarez, Alexandre Reily Rocha, Steve W. Bailey, Colin J. Lambert, Stefano Sanvito and Jaime Ferrer, Phys. Rev. Lett. **95**, 256804 (2005), Also cond-mat/0505487.

- Single channel conductance of H2 molecules attached to platinum or palladium electrodes Victor Garcia-Suarez, Alexandre Reily Rocha, Steve W. Bailey, Colin J. Lambert, Stefano Sanvito and Jaime Ferrer, Phys. Rev. B. **72**, 045437 (2005). Also cond-mat/0412726.

- Asymmetric I-V characteristics and magnetoresistance in magnetic point contacts, Alexandre Reily Rocha and Stefano Sanvito, Phys. Rev. B **70**, 094406 (2004). Also cond-mat/0403351.

- Computational Spintronics in highly confined systems, Stefano Sanvito and Alexandre R. Rocha, CMSM 2004.

Appendix C

The Smeagol Manual

User's Guide

SMEAGOL (version 1.0)

June 29, 2005

Alexandre Reily Rocha and Stefano Sanvito

Department of Physics, Trinity College Dublin, IRELAND

Víctor Manuel García Suárez and Jaime Ferrer Rodríguez

Departamento de Física, Universidad de Oviedo, SPAIN

Steve Bailey and Colin J. Lambert

Department of Physics, Lancaster University, Lancaster, LA14YB, UK

smeagol@tcd.ie

http://www.smeagol.tcd.ie

Contents

1 INTRODUCTION **2**

2 THE SYSTEM SET UP **2**

3 THE LEADS CALCULATION **3**

3.1 Input files 3

3.2 Input flags 3

3.3 Output files 4

4 *I*-*V* CALCULATION **5**

4.1 Input files 5

4.2 Input flags 6

4.2.1 Matching of the Hartree Potential 10

4.2.2 Flags specific of k-point calculations 11

4.2.3 Flags specific of Γ point calculations 12

4.3 Output files 12

5 HOW TO RUN SMEAGOL **13**

6 PROBLEM HANDLING **14**

7 ACKNOWLEDGEMENTS **14**

1 INTRODUCTION

This is a comprehensive user's guide to the quantum electronic transport code SMEAGOL (Spin and Molecular Electronics Algorithm on a Generalized atomic Orbital Landscape) [1, 2]. SMEAGOL is based on the non-equilibrium Green's function (NEGF) formalism for one-particle Hamiltonian. In its present form it uses density functional theory (DFT) with the numerical implementation contained in the code SIESTA [3, 4]. However, SMEAGOL's computational scheme is very general and can be implemented together with any electronic structure methods based on localized basis sets. Alternative implementations are currently under investigation.

In the next sections we will explain how to set up a typical calculation and we will describe the various options. The reader of this user guide is supposed to have familiarity with the non-equilibrium Green's function formalism [5, 6, 7, 8, 9, 10, 11, 12] and with density functional theory [13]. In addition a good knowledge of SIESTA and the SIESTA's input files is necessary since only the SMEAGOL's commands are described here, although a complete input file needs the setting of flags proper of SIESTA. For these we refer to the SIESTA user's guide [4].

2 THE SYSTEM SET UP

SMEAGOL is designed for calculating two probe I-V characteristics. The typical system investigated is described in figure 1. It comprises two semi-infinite leads (left and right) and an extended molecule, which includes the region of interest and a few atomic layers of the leads. The electronic structure of the leads is not affected by the potential drop and it is computed only once at the beginning of the calculation.

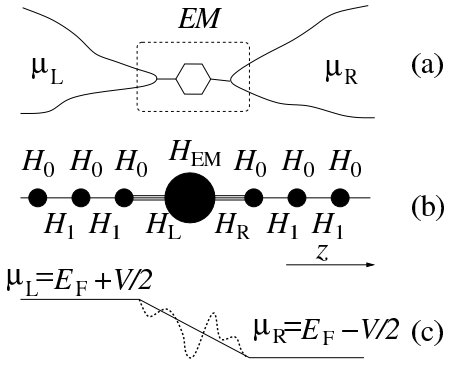

Figure 1: (a) Schematic two terminal device. Two leads are kept at the chemical potentials $\mu_{\rm L}$ and $\mu_{\rm R}$ and the transport is through the extended molecule *EM*. (b) The Hamiltonian is an infinite matrix comprising two block diagonal parts describing the leads and a part (finite) describing the extended molecule $H_{\rm EM}$. (c) Typical potential profile.

There are two fundamental steps in setting up a SMEAGOL's calculation: 1) calculating the electronic structure for the leads, and 2) calculating the I-V curve for a given system. The evalu-

ation of the electronic structure of the leads is essential and must be performed before attempting the calculation of the I-V. The files: `Systemlabel.HSL`, `Systemlabel.DM`, `bulklft.DAT` and `bulkrgt.DAT` are generated during the evaluation of the electronic structure of the leads. These are necessary for calculating the I-V characteristic.

3 THE LEADS CALCULATION

Here SMEAGOL evaluates the Hamiltonian $\mathcal{H}$, the overlap $\mathcal{S}$ and the density ρ matrices of the current/voltage probes (the leads). The self-energies for the semi-infinite electrodes can be obtained from the knowledge of the Hamiltonian and the overlap matrices of an infinite system, and therefore only a DFT calculation for a bulk system is needed. $\mathcal{H}$ and $\mathcal{S}$ should be written in the tridiagonal form described by Sanvito *et al.* [1, 14]. Figure (2) shows the typical setup for the electrodes.

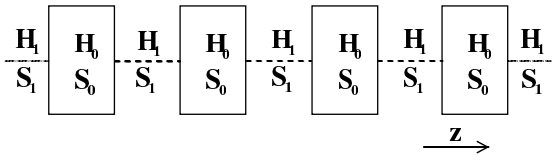

Figure 2: Periodic structure of the leads. H_0 and S_0 are the Hamiltonian and overlap matrix describing the interaction within one unit cell and H_1 and S_1 are the coupling and overlap matrix describing the interaction between adjacent unit cells. The arrow shows the direction of transport (z direction)

The same procedure is used for calculating the matrix elements of the Hamiltonian and overlap matrix in the case of k-points calculation (along the direction orthogonal to transport).

3.1 Input files

There are no additional input files than those used by SIESTA for an ordinary DFT calculation (input file, pseudopotentials ...). For these we refer to the SIESTA user's guide.

3.2 Input flags

The following input flags must be supplied to the `input.fdf` file in order to perform the bulk calculation.

BulkTransport (*Boolean*):

When this flag is set to true the leads Hamiltonian and overlap matrix will be written to the file `SystemLabel.HSL`. Moreover the files `bulklft.DAT` or `bulkrgt.DAT` (or both) will be created. These contain information about the system (Fermi energy, System label, unit cell, non-zero elements of the Hamiltonian and information about the super cell).

Default value: F

BulkLeads (*String*):

Define whether the calculated electronic structure should be used for the left-hand side lead, for the right-hand side lead of for both. In practise it defines the names and the extensions of the output files. There are three possible options

- `LR` or `RL` (the same output files will be used for both left- and right-hand side lead)
- `R` (the output files will be used for the right-hand side lead only)
- `L` (the output files will be used for the left-hand side lead only)

Default value: LR

3.3 Output files

bulklft.DAT :

File containing information about the left-hand side lead (number of states, Fermi energy, number of k-points, etc)

bulkrgt.DAT :

File containing information about the right-hand side lead (number of states, Fermi energy, number of $k$points, etc)

Systemlabel.HSL :

File containing the Hamiltonian and overlap matrices of the leads. The string `Systemlabel` is read from either `bulklft.DAT` or `bulkrgt.DAT`. If the two leads are the same one can use the same Systemlabel.HSL file to read the information for both the left and right leads (set the flag `BulkLeads` to `LR`). If the two systems are different, then two files with different names must be provided from two different leads calculations (set the flag `BulkLeads` to either `L` or `R`).

Systemlabel.DM :

File containing the density matrix of the leads. It is needed for calculating the I-V. The string `Systemlabel` is read from either `bulklft.DAT` or `bulkrgt.DAT`. If the two leads are made from the same materials and have the same geometrical arrangement one can use the same `Systemlabel.DM` file to read the information about both the left- and right-hand side lead (set the flag `BulkLeads` to `LR`). If the two leads are different, then two files with different names must be provided from two independent leads calculations (set the flag `BulkLeads` to either `L` or `R`). The `Systemlabel.DM` file is used to set the boundary condition at the interface between the scattering region and the electrodes.

4 *I-V* CALCULATION

Once the initial calculation for the leads has been performed we can proceed with computing the non-equilibrium transport properties of our system.

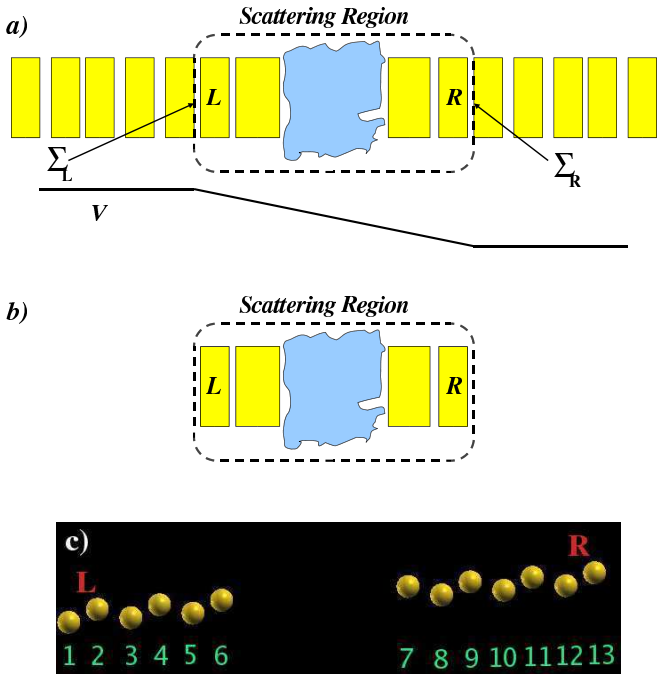

Figure 3: a) Sketch of a typical system simulated using SMEAGOL: two semi-infinite electrodes connected to a central scattering region. b) The typical unit cell for a SMEAGOL calculation. The reader must note that we include at least one layer of the leads both to the left- and to the right-hand side of the scattering region. The order of the atoms in the input file is only important in these two layers. The first atoms of the list (see `AtomicCoordinatesAndAtomicSpecies` in the SIESTA user's guide) must be those contained in the left-hand side lead unit cell, and they should be input with the same order than in the bulk leads calculation. In an analogous way, the last atoms of the list must conform with the input file for the right-hand side lead. c) An example of a system setup for two semi-infinite zig-zag wires separated by a vacuum region. The numbers for each atom have been indicated as well as the two atoms on either side corresponding to the the unit cells for the left- and right-hand side lead.

4.1 Input files

Several files generated during the construction of the leads must be used. These are: `bulklft.DAT`, `bulkrgt.DAT`, `Systemlabel.HSL` and `Systemlabel.DM`.

4.2 Input flags

The following input flags must be supplied to the file `input.fdf` in order to perform the transport (I-V) calculation. If any of these values is not supplied, the default value will be assumed. Special care should be taken for the variables that control the real and contour integrals, when calculating the non-equilibrium density matrix.

EMTransport (*Boolean*):

When `EMTransport` is set to `T`, SMEAGOL will perform an NEGF calculation (transport). Otherwise, the code will perform a standard SIESTA equilibrium ground state calculation for either a periodic system or a molecule.

Default value: F

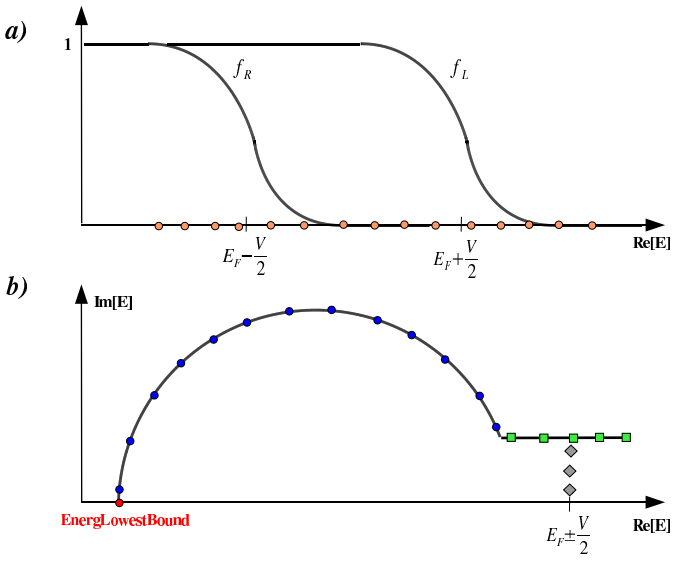

Figure 4: Sketches for the Green Function integration leading to the non-equilibrium charge density [1, 2]. a) The non-equilibrium part must be performed along the real energy axis, but is bound by the left- and right-hand side Fermi distributions functions, f_{L} and f_{R}. b) The equilibrium component is performed over a semicircular path in the complex energy plane. Here we show the starting position `EnergLowestBound`, the energy mesh along the two segments of the curve and the poles of the Fermi distribution.

NEnergReal (*Integer*):

Number or energy points along the real axis for the integration of the non-equilibrium part of the Green's function. The number of points is extremely sensitive to the type of calculation performed. The points are distributed between $-(\frac{V}{2} + 30.0\ k_B T)$ and $(\frac{V}{2} +$

30.0 k_BT). The number of points determines how fine the mesh is. Since the Green's function on the real axis can be ill-behaving (large number of singularities), a very fine mesh might be necessary. For equilibrium calculations, NEnergReal can be set to zero. In Fig. (4a) the real axis energy points are sketched as orange circles and `NEnergReal`= 15.

Default value: 0

NEnergImCircle (*Integer*):

Number of energy points along the semi-circle in the complex plane (equilibrium part of the Green's function integral). The circle starts at the energy `EnergLowesBound` and finishes at the complex point energy point ($E_F - 30.0\ k_BT,\ 2\ \mathrm{N_{Poles}}\pi\ k_BT$). The values for the energy points are calculated using a Gauss-Legendre algorithm along a semicircle. Fig. (4b) shows a sketch of the points along the contour in the complex energy plane. In this example, the points along the circle are represented by blue circles and `NEnergImCircle`= 13.

Default value: 15

NEnergImLine (*Integer*):

Number of energy points along the line in the complex plane where the integral of the equilibrium contribution to the density matrix is performed. This straight line starts at the point ($E - 30.0\ k_BT,\ 2\mathrm{N_{Poles}}\pi k_BT$) and finishes at point ($E_F + 30.0\ k_BT,\ 2\mathrm{N_{Poles}}\pi k_BT$) (green squares in Fig. (4b), `NEnergImLine`= 5). The position of the energy points is determined by a Gauss-Legendre algorithm.

Default value: 20

NPoles (*Integer*):

Number of poles in the Fermi distribution used to compute the contribution to the equilibrium charge density. The Fermi distribution is given by:

$$f(\epsilon) = \frac{1}{e^{\frac{\epsilon-\mu}{k_BT}} + 1} \tag{1}$$

The poles of $f(\epsilon)$ all lay on the complex plane with $i_n = \mu + (2 * n + 1)\, i\pi$ being the n-th pole. The number of poles specifies how far from the real axis the contour integral will be performed. The furthest away from the real axis the more well-behaving the Green's function is.

In Fig. (4b) the position of the poles are represented by grey diamonds (`NPoles`= 3).

Default value: 5

VInitial (*Physical*):

Value of the initial bias for the I-V calculation.

Default value: 0.0 eV

VFinal (*Physical*):

Value of the final bias for the I-V calculation.

Default value: 0.0 eV

NIVPoints (*Integer*):

Number of bias steps considered between the two limits V_{initial} and V_{final}. The current will be calculated only for these biases.

Default value: 0

Delta (*Double precision*):

Small imaginary part δ that accounts for the broadening of the localized energy levels of Green Function, $G = [\epsilon + i\delta - H_{\text{S}} - \Sigma_{\mathbf{L}} - \Sigma_{\text{R}}]^{-1}$.

Default value: 10^{-4}

EnergLowestBound (*Physical*):

Energy that specifies the beginning of the contour integral in the complex plane for the equilibrium contribution to the charge density. `EnergLowestBound` must be below both the lowest band of the leads and the lowest energy level of the scattering region. Red dot in Fig. (4b).

Default value: -5.0 Ry

NSlices (*Integer*):

Number of slices of the bulk Hamiltonian substituted into the left- and right-hand side part of scattering region. The Hamiltonian of these slices is not recalculated self-consistently and these "buffer" layers help the convergence, in particular when the two leads are different.

Default value: 1

AtomLeftVCte (*Integer*):

Number of the atom used to determine the onset of the bias ramp on the left-hand side. The number of the atom is taken from the order in which the atoms are specified in the SIESTA block `AtomicCoordinatesAndAtomicSpecies` (refer to the SIESTA manual for a description of this block).

Default value: 1

AtomRightVCte (*Integer*):

Number of the atom that is used to determine the onset of the bias ramp on the right-hand side. The number of the atom is taken from the order in which the atoms are specified in the SIESTA block `AtomicCoordinatesAndAtomicSpecies` (refer to the SIESTA manual for a description of this block).

For $z < z_{\text{AtomLeftVCte}}$ and $z > z_{\text{AtomRightVCte}}$, the potential added to the Hartree potential is a constant equal to $+\frac{V}{2}$ and $-\frac{V}{2}$ respectively. For $z_{\text{AtomLeftVCte}} < z < z_{\text{AtomRightVCte}}$

a bias ramp $V\,(z-z_0)$ is introduced, where z_0 is the midpoint between $z_{\mathrm{AtomLeftVCte}}$ and $z_{\mathrm{AtomRightVCte}}$.

Default value: Last Atom

TrCoefficients (*Boolean*):

If set to true the transmission coefficients as a function of energy $T\,(\epsilon, V)$ will be calculated and printed to the file `SystemLabel.TRC` (see section 4.3 below) for any bias point. Although the current is defined as

$$\frac{e}{h}\int_{-\infty}^{\infty} T\,(\epsilon, V)\,(f_{\mathrm{L}} - f_{\mathrm{R}})\,d\epsilon \tag{2}$$

we can calculate the Transmission coefficients for a wider range of energy values and a different number of energy points by using the variables `InitTransmRange`, `FinalTransmRange` and `NTransmPoints` which are defined below.

Default value: F

InitTransmRange (*Physical*):

Initial energy for the calculation of the transmission coefficients

Default value: -5.0 Ry

FinalTransmRange (*Physical*):

Final energy for the calculation of the transmission coefficients

Default value: 5.0 Ry

NTransmPoints (*Integer*):

Number of energy points uniformly distributed between `InitTransmRange` and `FinalTransmRange` for which the transmission coefficients are calculated. If `TrCoefficients` is set to false, this flag is ignored.

Default value: 100

PeriodicTransp (*Boolean*):

If the two electrodes are equal one may use periodic boundary conditions over the supercell. This procedure improves convergence and decreases the size of the unit cell. If the two leads are different then `PeriodicTransp` must be set to `F`.

Default value: T

SaveBiasSteps (*data block*):

Block containing information for printing the Hartree potential (`Systemlabel.VH`), the charge density (`Systemlabel.RHO`), the difference between the self-consistent and the atomic charge densities (`Systemlabel.DRHO`) and the total DFT potential

(`Systemlabel.VT`) at a specified bias step. The arguments in the block are integers indicating the bias steps at which to print these files. They range from from 0 up to `NIVpoints`. The appropriate SIESTA flag for printing any of these files must also be set to true.

Example :

```
% block SaveBiasSteps
0 2 3
% endblock SaveBiasSteps
```

SMEAGOL will print the requested file (for instance `Systemlabel.VT` if the SIESTA flag `SaveElectrostaticPotential` is set to true) at the bias steps 0, 2 and 3.

Default value: No Default

4.2.1 Matching of the Hartree Potential

One important information needed from the leads calculation is the average value of the Hartree Potential. SMEAGOL uses a Fast Fourier Transform (FFT) algorithm to solve the Poison equation in reciprocal space (refer to the SIESTA manual for more details). The $\vec{k} = 0$ term is ignored resulting in a Hartree Potential that is defined up to a constant. This might lead to a mismatch between the Hartree potential of the leads and that at the edges of the scattering region. This mismatch is better illustrated in Fig. 5 where we have plotted the average Hartree potential on the plane orthogonal to transport for the electrodes and two semi-infinite wires separated by 12 Å (our scattering region; see Fig. (3c)) using the post-processing tool `Pot.exe` (included in the Utils directory).

In order to compensate for the arbitrary zero of the potential we shift the Hartree Potential of the scattering region by a constant (see variables `HartreeLeadsLeft`, `HartreeLeadsRight` and `HartreeLeadsBottom` described below). In this way we force the match of the potential at the scattering region/leads interface.

HartreeLeadsLeft (*Physical*):

Position in space in the left-hand side lead where the Hartree potential of the scattering region should match that of the bulk.

Default value: 0.0 Ang

HartreeLeadsRight (*Physical*):

Position in space of the right-hand side lead where the Hartree potential in the scattering region should match that of the bulk.

Default value: 0.0 Ang

HartreeLeadsBottom (*Physical*):

Average value of the Hartree potential in the leads over a plane perpendicular to the transport direction. This parameter should be evaluated from the bulk leads calculation.

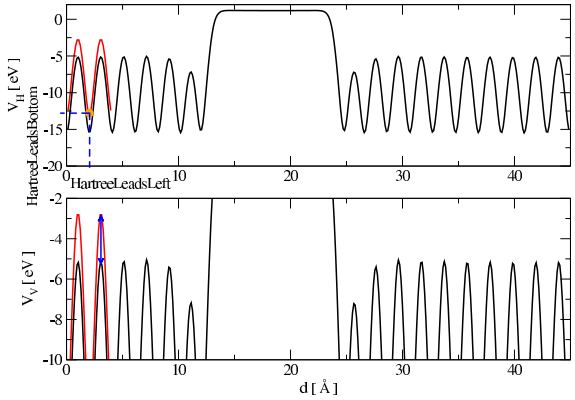

Figure 5: Shift of the Hartree potential between two SIESTA calculations: bulk gold (the leads) with two atoms in the unit cell (red line) and the parallel plate capacitor (scattering region) depicted in Fig. (3c). Note that the Hartree potential of the scattering region has converged to the bulk value after the first atomic layer except for a constant.

The fast Fourier Transform algorithm used in SMEAGOL to solve the Poisson's equation disposes the $k = 0$ term. This defines the Hartree potential up to an arbitrary constant. By defining `HartreeLeadsBottom` we shift the Hartree potential in the scattering region in order to match that of the leads.

Default value: 0.0 eV (no shift)

4.2.2 Flags specific of k-point calculations

These flags are used only when periodic boundary conditions in the direction orthogonal to the transport are considered.

SaveMemTranspK (*Boolean*):

By default, SMEAGOL calculates the self-energies during the first iteration and stores them in memory. There are two self-energies (one for the left- and one for right-hand side contact) for each energy point and each k-point. If the number of k-points and the size of the unit cell of the leads are large, large memory might be needed. In that case, `SaveMemTranspK` should be set to `T`, and the self-energies will be re-calculated after each iteration without the need for storing them in memory. Use with caution; this specification might increase considerably the time of the calculation.

Default value: F

TransmissionOverk (*Boolean*):

Allow to print the transmission coefficient as a function of both the k-vector in the transverse Brillouin zone and the energy. Note that the flag `TrCoefficients` provide only the

integral of the transmission coefficients over the k-points. `TrCoefficients` must be set to true. The file `SystemLabel.TRC.k.up (down)` contains the results (see section 4.3 below).

Default value: F

4.2.3 Flags specific of Γ point calculations

UseLeadsGF (*Boolean*):

If true the file `SystemLabel.LGF` is read from disk (see description of output files below). It is particularly useful when one wants to perform numerous calculations with the same set of leads and the same number of real energy points.

Default value: T

4.3 Output files

SystemLabel.CUR :

Four column vector containing the I-V characteristics. The first column corresponds to the bias in Rydberg, the second column corresponds to the total current and the third and fourth columns (in the case of a spin-dependent calculation) correspond to the spin-decomposed current.

SystemLabel.TRC :

Transmission coefficients as a function of energy for different biases. The first column corresponds to the energy, the second to the total transmission for both the spin direction and the following `nspin` columns are the spin-decomposed transmission coefficients.

SystemLabel.TRC.k.up (down) :

If `TransmissionOverk` is set to `T`, `SystemLabel.TRC.k.up` contains the transmission coefficients for each k point and energy for either up or down spins. The first two columns contain the perpendicular components of each k point, the third column contains the spin dependent transmission coefficient and the fourth column contains the energy.

SystemLabel.LGF :

For a Γ point serial calculation, this file contains the `NEnergReal` self-energies for both the right and left hand-side leads for all the energies lying on the real axis. This file might be useful when restarting a calculation. The user should note that, in this case, both `VInitial`, `VFinal` and `NEnergReal` must be the same as in a previous calculation.

SystemLabel.CHR :

File containing the total charge inside the scattering region after each iteration. The first column contains the iteration number, the second column contains the charge in units of e (the electron charge), the third column contains the bias in Rydberg and the fourth column contains the temperature also in Rydberg.

IV.SystemLabel.VH :

If `SaveElectrostaticPotential` is set to true (see SIESTA user's guide) and the block `SaveBiasSteps` is present, the file `IV.SystemLabel.VH` containing the Hartree potential for the biases specified in `SaveBiasSteps` will be generated.

IV.SystemLabel.VT :

If `WriteVT` is set to true (see SIESTA user's guide) and the block `SaveBiasSteps` is present, the file `IV.SystemLabel.VT` containing the total DFT potential for the biases specified in `SaveBiasSteps` will be generated.

IV.SystemLabel.RHO :

If `WriteRHO` is set to true (see SIESTA user's guide) and the block `SaveBiasSteps` is present, the file `IV.SystemLabel.RHO` containing the self-consistent charge density for the biases specified in `SaveBiasSteps` will be generated.

IV.SystemLabel.DRHO :

If `WriteDRHO` is set to true (see SIESTA user's guide) and the block `SaveBiasSteps` is present, the file `IV.SystemLabel.DRHO` containing the difference between the self-consistent and the atomic charge densities for the biases specified in `SaveBiasSteps` will be generated.

IV.SystemLabel.DMT :

If `SaveDMT` is set to true and the block `SaveBiasSteps` is present, the file `IV.SystemLabel.DM` containing the converged density matrix at each bias specified in `SaveBiasSteps` will be written.

5 HOW TO RUN SMEAGOL

Assuming that the user has already compiled SMEAGOL successfully and saved it as an executable (say smeagol.1.0), then we are ready to start a calculation. Initially the user must setup the input file for the left- and right-hand side leads. The input file must include the appropriate SIESTA and SMEAGOL flags as discussed in section (3.2) and in the SIESTA Manual. Let us assume, as a simple example, that the two leads are equivalent and the leads file is called `leads.fdf`. In the same directory where this file is, there must also be the pseudopotential files - either in `.psf` or `.vps` format. Then the user should type:

$ smeagol.1.0 < `leads.fdf` > `leads.out`

The calculation will yield the files `bulklft.DAT`, `bulkrgt.DAT`, `Systemlabel.HSL` and `Systemlabel.DM`. These should be moved to the directory containing the input file for the scattering region - say `scattering.fdf`. This directory must also contain pseudopotential files for all the atomic species involved in the calculation. It is important to note that the user must be consistent with regards to pseudopotentials, they should be identical to the ones used for the leads calculation.

Before starting the I-V calculation we must obtain `HartreeLeadsBottom`. For that we use the post-processing tool `Pot.exe` contained in the `Utils` directory of the SMEAGOL distribution. The file `Potential.dat` should be edited in order to process the file `SystemLabel.VH` (for the leads). The resulting output `SystemLabel.VH_DAT` can be viewed using a plotting tool such as gnuplot or xmgrace.

We are interested in the first two columns of this data file. They contain the planar average of the Hartree potential as a function of the z direction. A value for the Hartree Potential (`HartreeLeadsBottom`) should be chosen (usually the edges of the unit cell) and the position in the unit cell should be recorded and matched with the equivalent position in the scattering region (`HartreeLeadsLeft` and `HartreeLeadsRight`). Once this is done, we are ready to calculate the I-V characteristics:

$ smeagol.1.0 < `scattering.fdf` > `scattering.out`

The resulting files can be processed using a variety of plotting programs such as gnuplot and xmgrace for 2D-plots and OpenDX for 3D-plots and isosurfaces.

6 PROBLEM HANDLING

The users are encouraged to report problems and bugs to the SMEAGOL's developers at smeagol@tcd.ie. Patches and fixes will be uploaded to the web-site http://www.smeagol.tcd.ie/

7 ACKNOWLEDGEMENTS

This work is sponsored by Science Foundation of Ireland under the grant SFI02/IN1/I175. JFR and VMGS thank the spanish Ministerio de Educación y Ciencia for financial support (grants BFM2003-03156 and AP2000-4454). SB and CJL thank the UK EPSRC and the EU network MRTN-CT-2003-504574 RTNNANO. ARR thanks Enterprise Ireland (grant EI-SC/2002/10) for financial support. Traveling has been sponsored by the Royal Irish Academy under the International Exchanges Grant scheme.

References

[1] A. R. Rocha, V. M. García-Suárez, S. W. Bailey, C. J. Lambert, J. Ferrer and S. Sanvito, Nature Materials **4**, 335 (2005).

[2] A. R. Rocha, V. M. García-Suárez, S. W. Bailey, C. J. Lambert, J. Ferrer and S. Sanvito, in preparation.

[3] J. M. Soler, E. Artacho, J. D. Gale, A. Garcia, J. Junquera, P. Ordejón and D. Sanchez-Portal, J. Phys. Cond. Matter **14**, 2745-2779 (2002).

[4] http://www.uam.es/departamentos/ciencias/fismateriac/siesta/

[5] S. Datta, *Electronic Transport in Mesoscopic Systems*, Cambridge University Press, Cambridge, UK, 1995.

[6] H. Haug and A. P. Jauho, *Quantum Kinetics in Transport and Optics of Semiconductors*, Springer, Berlin, 1996.

[7] C. Caroli, R. Combescot, P. Nozieres, and D. Saint-James, J. Phys. C: Solid State Phys. **5**, 21 (1972).

[8] J. Ferrer, A. Martín-Rodero, and F. Flores, Phys. Rev. B **38**, 10113 (1988).

[9] J. Taylor, H. Guo, and J. Wang, Phys. Rev. B **63**, 245407 (2001).

[10] M. Bradbyge, J. Taylor, K. Stokbro, J.-L. Mozos, and P. Ordejón, Phys. Rev. B **65**, 165401 (2002).

[11] J. J. Palacios, A. J. Pérez-Jiménez, E. Louis, E. SanFabián, and J. A. Vergés, Phys. Rev. B **66**, 035322 (2002).

[12] A. Pecchia and A. Di Carlo, Rep. Prog. Phys. **67**, 1497 (2004).

[13] H. Hohenberg and W. Kohn, Phys. Rev. **136**, B864 (1964). W. Kohn and L.J. Sham, Phys. Rev. **140**, A1133 (1965).

[14] S. Sanvito, C. J. Lambert, J. H. Jefferson, and A. M. Bratkovsky, Phys. Rev. B, **59**, 11936 (1999).