

A next-
generation
simulation tool
for electron,
thermal and
spin transport.

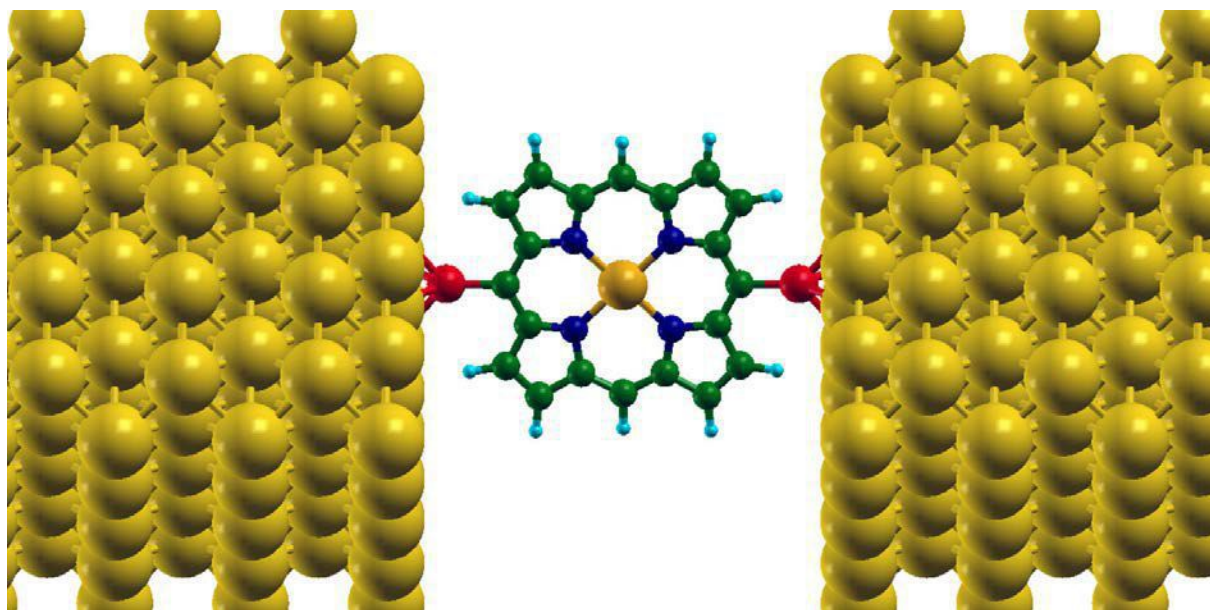
Gollum2

User Manual

July 2018



Universidad de Oviedo
La Universidad de Asturias



Gollum2 Copyright Holders

Jaime Ferrer	Universidad de Oviedo
Colin J. Lambert	Lancaster University
Hatef Sadeghi	University of Warwick
Victor M. García-Suárez	Universidad de Oviedo
Steven Bailey	Lancaster University

Gollum2 Authors

Currently-active authors:

Jaime Ferrer

Hatef Sadeghi

David Visontai

Victor M. Garcia-Suarez

Sara Sangtarash

Steven W. D. Bailey

Past authors include: Laszlo Oroszlany, David Zs. Manrique, Iain Grace, Ruben Rodriguez-Ferradas, Katalin Gillemot, Laith A. Algharagholy, Zain Al-Milli.

Gollum2 released in July 2018 is available for academic research under the terms of the **Gollum2 Academic License** found via the link: <https://github.com/gollumcode>

1 CONTENTS

1	Contents.....	0
2	Introduction	4
2.1	Updates and new functionalities: IMPORTANT CHANGES.....	4
2.2	The Advisory Board:	4
2.3	Citing Gollum.....	4
2.4	Gollum Units	4
2.5	Distributed Folder Structure	5
3	The scope of the Gollum project.....	6
4	Installation and Running.....	7
4.1	MATLAB version.....	7
4.1.1	Installing the MCR and running in Windows	7
4.1.2	Installing the MCR and running LINUX.....	7
4.1.3	Installing the MCR and running MAC.....	8
5	Test Examples and TUTORIALS.	10
5.1	The Gollum INPUT FILE GENERATOR.....	10
	Table 5.1.....	11
5.2	Overview	12
	Summary of the examples: TABLE 5.2.....	14
5.3	Step by step Tight-binding-based EXAMPLE: 1. single-atom_chain.spin	16
5.4	Step by step siesta-based example: 1.au_linear_chain	17
5.5	AiiDA plugin for GOLLUM.....	18
5.5.1	installing the plugin.....	18
5.5.2	Standard GOLLUM plugin	19
5.5.3	GOLLUM-Siesta workflow	19
6	Hamiltonian generation.....	20
7	Theoretical Approach.....	21
7.1	Nomenclature.	21

7.2	The surface Green's function for the current carrying leads.	23
7.3	The extended scattering region Hamiltonian.....	24
7.4	Hamiltonian Assembly	25
7.5	Scattering matrix and transmission in multi-terminal devices.....	25
7.6	Functionalities delivered by gollum2	28
7.7	Input files format and notation.....	28
8	the file 'input'	30
8.1	Mandatory Variable: Mode.....	30
8.2	A summary of the Mode function	31
8.3	Mandatory Variable: Leadp.....	31
8.4	Mandatory Variable: atom.....	32
8.5	Mandatory Variable: HamiltonianProvider	32
8.6	Partial mandatory variable variable: Path_EM	33
8.7	Mandatory Variable: PATH_LEADs.....	33
8.8	Partial Mandatory variable for modes 0, 1 and 3: ERange.....	33
8.9	Partial mandatory variable for modes 3 and 5: ThermoTRange.	33
8.10	Partial mandatory variable for mode 3: Thermosweep.	33
8.11	Partial mandatory variable for modes: Thermoefrange.	34
8.12	Partial mandatory variable for mode 3: Thermotemp.....	34
8.13	Partial mandatory variable for mode-3: ThermoSpinmethod.	34
8.14	Partial mandatory variable for Mode=4: Bias	34
8.15	Partial mandatory variable for mode 5: WRange	35
8.16	Optional variable for all Modes: variable: EFshift	35
8.17	Optional variable for Mode=4: variable: Bias_accuracy	35
8.18	Optional variable for Modes 1,2,3,4: variable: scissors	35
8.19	Optional variable for all Modes: User defined Gate PoTENTIAL	35
8.20	Optional variable for all Modes: VgateFACTOR	36
8.21	Optional variable for mode-4: User defined bias potential	36
8.22	Optional variable for mode-3: thermousesavetrm	36

8.23	Example input file.	36
9	The 'Extended Molecule' file.	38
9.1	An Example 'Extended_Molecule' file.....	38
9.2	Variable: nspin	38
9.3	Variable: Fermie	38
9.4	Variable: iorb.....	39
9.5	Variable: kpoints	39
9.6	Variable: HSM.....	39
10	THE 'Lead_i' input file.	40
10.1	Example 'Lead_i' file.	40
10.2	Variable: kpoints_lead	40
10.3	Variable: HSL	40
11	Output file format and notation.	41
11.1	The Universal files.	41
11.1.1	The Universal output at a glance: TABLE 10.1.1.....	41
11.1.2	Transmission (reflection) coefficients.	41
11.1.3	The number of open channels.....	42
11.1.4	Shot noise.	42
11.2	Additional Mode generated files.....	42
11.2.1	Mode generated files at a glance: Table 10.2.1.	42
11.2.2	Mode-2 generated files.	42
11.2.3	Mode-3 generated files.	43
11.2.4	Mode-4 generated files.	44
11.2.5	Mode-5 generated files.	44
12	References	45
13	Index	46

2 INTRODUCTION

2.1 UPDATES AND NEW FUNCTIONALITIES: IMPORTANT CHANGES.

Gollum2 can read output Wannier Hamiltonians from the Wannier90 code. So Gollum interfaces now with plane wave codes such as Castep, VASP, ABINIT and Quantum-Espresso.

The input file needed to run Gollum2 is now generated interactively avoiding tedious formatting for the user.

Exciting new functionalities allow the user to calculate:

- the phonon contribution to the thermal conductance
- a new range of thermal properties including the figure of merit ZT
- User defined potential profiles to apply bias and gate voltages
- The transmission and number of open channels is calculated for each value of k
- The band structure, density of states and number of open channels of a given lead
- Modularisation and use of sparsity has resulted in efficiency improvements where Gollum2 is up to 1.5 times faster and consumes up to three times less memory than previous versions of Gollum.

Updates to bug fixes together with code development will be posted on the web-site: <https://github.com/gollumcode>

2.2 THE ADVISORY BOARD:

All changes will be vetted by the advisory board:

Jaime Ferrer	ferrer@uniovi.es
Colin J. Lambert	c.lambert@lancaster.ac.uk
Victor García-Suárez	vm.garcia@cinn.es
Hatef Sadeghi	Hatef.Sadeghi@warwick.ac.uk
Steven Bailey	s.bailey@lancaster.ac.uk

2.3 CITING GOLLUM

A full description of the functionality and capabilities are described in:

"Gollum: a next-generation simulation tool for electron, thermal and spin transport", J. Ferrer, C. J. Lambert, V. M. García-Suárez, D. Zs. Manrique, D. Visontai, L. Oroszlany, R. Rodríguez-Ferradás, I. Grace, S. W. D. Bailey, K. Guillemot, Hatef Sadeghi, L. A. Algharagholi, New J. Phys. **16** 093029(2014), [doi:10.1088/1367-2630/16/9/093029](https://doi.org/10.1088/1367-2630/16/9/093029)

If the GOLLUM2 Package used wholly or partially to perform phonon related calculations, in addition to the above paper, the following paper must be cited:

"Oligoyne molecular junctions for efficient room temperature thermoelectric power generation", Hatef Sadeghi, Sara Sangtarash, and Colin J. Lambert, Nano Letters, 15(11), 7467 (2015). [doi: 10.1021/acs.nanolett.5b03033](https://doi.org/10.1021/acs.nanolett.5b03033)

2.4 GOLLUM UNITS

1. Energy (E) in eV.
2. Current intensities (I) in Amps.

3. Voltages (V) in volts.
4. Temperatures in Kelvins.
5. All energies refer to the Fermi energy (E_F) of the extended scattering region referred to as the extended molecule (EM) as shown in Figure 7.1.2.

2.5 DISTRIBUTED FOLDER STRUCTURE

The distribution contains the following folders and files

Gollum2		
quick start.pdf		
examples	Tight-binding	Folders 0-6 (see Table 5.3)
	DFT-based	Folders 1-8 (see Table 5.3)
	Wannier	Folders 1-6 (see Table 5.3)
	phonon	Folders 1-2 (see table 5.3)
gollum.linux		
gollum.mac		
gollum.windows		
manual		

You will find in folders 'gollum.linux', 'gollum.mac' and 'gollum.windows' the corresponding executable file plus a generated Readme.

The 'examples' folder is more complex but contains all the information to run the tutorial examples. Once the '.zip' files are extracted you will find folders 'Tight-binding', 'DFT-based', 'phonon' and 'Wannier' each of which will expand into the MACOSX versions and the Windows/Linux versions.

All the 'Tight-binding' examples labelled $n = 0-6$ contain:

- 'mode-0, mode-1, mode-2, mode-3, mode-4' folders, each of which contain the 'input' file needed to run Gollum with the resultant output files.
- ' n .TB.Guide.pdf' that gives details of the system and plots of the calculation, where n is the example label.
- 'Extended_Molecule' input to Gollum
- 'Lead_1' input to Gollum
- 'Lead_2' input to Gollum

All the 'Siesta-based' examples labelled $n = 1-8$ contain folders:

- 'mode-0, mode-1, mode-2, mode-3, mode-4' each of which contain the 'input' file needed to run Gollum with the resultant output files.
- 'Gollum_files' which contains the Extended_Molecule, Lead_1, Lead_2 and input files.
- ' n .DFT.Guide.pdf' which gives details of the system and plots of the calculation, where n is the example label.
- 'siesta_files' which contains the files needed to run the associated SIESTA ² calculations 'lead.fdf', 'emol.fdf' and the pseudopotential files ('.psf'). NOTE the '.fdf' files have the flags needed to extract the Hamiltonians using the DIRECT SIESTA INTERFACE METHOD shown in section 6.1.

The 'phonon' examples contain folders:

- 'mode-5' which contains the 'input' file needed to run Gollum with the resultant output files.
- 'siesta_files' which contains the files needed to run the SIESTA ² calculations.

The 'Wannier' examples labelled $n=1-6$ contain folders:

- 'mode-0, mode-1, mode-2, mode-3, mode-4' each of which contain the 'input' file needed to run Gollum with the resultant output files.

- *n*.W.90.Guide.pdf which gives details of the system and plots of the calculation, where *n* is the example label.

The folder ‘manual’ contains this documentation.

3 THE SCOPE OF THE GOLLUM PROJECT.

Gollum2 is a code that computes the electrical and thermal transport properties of multi-terminal nanoscale systems. The program can compute transport properties of either user-defined systems described by a tight-binding (or Hückel) Hamiltonian, or more material-specific properties of systems composed of real atoms described by DFT Hamiltonians. Gollum2 will interface easily with any DFT flavor:

- Localized basis set codes and can currently read information from all the latest flavors of the code SIESTA ².
- Plane wave codes such as Castep, VASP, ABINIT, Quantum-Espresso via the Wannier90 code.

Gollum2 is based on equilibrium transport theory, which means that it consumes much less memory than non-equilibrium Green’s function codes. The program has been designed for user-friendliness and takes a considerable leap towards the realization of ab initio multi-scale simulations of conventional and more sophisticated transport functionalities. These include:

- The ability to use either model tight-binding or DFT Hamiltonians.
- The ability to compute non-equilibrium current-voltage curves.
- Access to the full scattering matrix, including scattering amplitudes, phases and Wigner delay times.
- The ability to construct conductance statistics of single-molecule conductances, relevant to break-junction and STM measurements.
- Integration of a wide range of phenomena, including thermoelectrics, spintronics, superconductivity, Coulomb blockade, quantum pumps, Kondo physics and topological phases.
- The ability to describe multi-probe structures.

The Gollum work-flow shown in FIG. 3.1 below illustrates the versatility of the package to allow the user to generate the required Hamiltonians either ‘by hand’ from prescribed tight-binding parameters or by adapting the output of DFT codes. A suitable methodology follows the following route. Once the atomic arrangements are generated these are fed into the second stage, where the Hamiltonian matrix is generated. This stage is in practice independent of the previous geometry construction and can be run separately, taking only the output geometries of the first stage. The junction Hamiltonian can be generated using a variety of tools, some of which are listed in box II in FIG 3.1. A popular approach is the use of DFT codes that are able to write the Hamiltonian in a tight-binding language. In this way, model tight-binding Hamiltonians can also be easily generated. Other approaches involve the use of Slater-Koster or semi-empirical methods. In addition, Gollum has the ability to modify these Hamiltonian matrices. For example, the Hamiltonian matrix can be modified to include scissor corrections, Coulomb-blockade physics, a gate or bias voltage, a magnetic phase factor or a superconducting order parameter. Finally, stage III is the actual quantum transport calculation. This takes the Hamiltonian matrix as an input and calculates the S-matrix and associated physical quantities, such the electrical or spin current, the conductance or the thermopower.

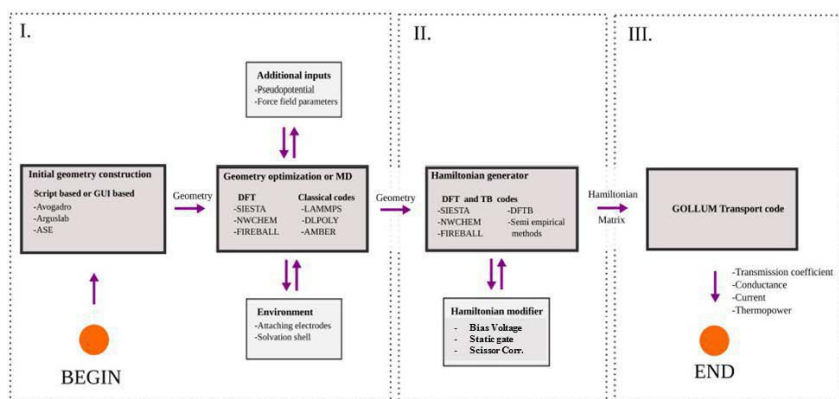


FIG.3.1: Typical Gollum workflow with various optional software tools.

4 INSTALLATION AND RUNNING.

Gollum2 is distributed as a standalone MATLAB executable. A copy can be obtained by logging onto the web page at <https://github.com/gollumcode> and completing the application form.

Upon request a user specific compatible version of Gollum will be provided.

Requests for the developer-version will be considered by the advisory board.

4.1 MATLAB VERSION.

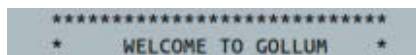
As Gollum is a standalone executable, all you need to run the programme is a Gollum-compatible MATLAB Compiler Runtime (MCR).

4.1.1 INSTALLING THE MCR AND RUNNING IN WINDOWS

From your Gollum2 distribution navigate to your version dependent executable in the 'Gollum.windows' directory to locate 'Gollum.exe'.

For a windows machine, to verify if you have the MCR installed simply: double click on Gollum.exe or run Gollum.exe from the command prompt.

If you have a compatible MCR, then you will see the following window:



There will be an 'Error message' at this stage which indicates that as yet you have not included the correct input files to run a calculation.

If you do not have the correct MCR then navigate to <http://www.mathworks.co.uk/products/compiler/mcr/> in MathWorks and locate the correct version to install.

Windows requires release Windows R2017a 64-bit.

Once the MCR is correctly installed you can make your first test.

You can of course place Gollum.exe in your path.

Copy from the examples folder supplied with your distribution an example of 'your choice' into a new directory of your choice. Place 'Gollum.exe' in the folder and run the calculation by double clicking 'Gollum.exe'. Your results of the calculation should match those to be found in the example matching 'your choice'.

4.1.2 INSTALLING THE MCR AND RUNNING LINUX

From your Gollum2 distribution navigate to your version dependent executable in the 'Gollum.linux' directory to locate 'Gollum' and 'run_Gollum.sh'.

For a LINUX machine verify that you have the MCR e.g. using the terminal command

locate MATLAB_Compiler_Runtime

If you do not have the correct MCR then navigate to <http://www.mathworks.co.uk/products/compiler/mcr/> in MathWorks and locate the correct version to install.

LINUX requires Linux R2017a 64-bit.

Install as root to place the application in your 'usr/local directory'.

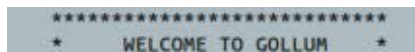
Enter your 'Gollum_linux' directory to find 'Gollum' and 'run_Gollum.sh' NOTE: ensure that you have full permissions in place for both files.

To test your MCR locate your MCR libraries which should be in
/usr/local/MATLAB/MATLAB_Compiler_Runtime/v9.2/

Then type

./run_Gollum.sh /usr/local/MATLAB/MATLAB_Compiler_Runtime/v9.2/

If you have a compatible MCR, then you will see the following window:



There will be an 'Error message' at this stage which indicates that as yet you have not included the correct input files to run a calculation.

The location of the MCR libraries and 'Gollum' can be placed in your PATH by editing your .bashrc or .cshrc files or by typing the following to set the path prior to each calculation.

In -s path-source/run_Gollum.sh run_Gollum.sh

In -s path-source/Gollum Gollum

For simplicity we outline the following procedure.

Copy from the examples folder supplied with your distribution an example of 'your choice' into a new directory of your choice. Place 'Gollum' and 'run_Gollum.sh' into the folder.

Type

./run_Gollum.sh /usr/local/MATLAB/MATLAB_Compiler_Runtime/9.2/

Your results of the calculation should match those to be found in the example matching 'your choice'.

For advanced users 'run_Gollum.sh' can be edited to include the location of the compiler in the path.

4.1.3 INSTALLING THE MCR AND RUNNING MAC

Enter the 'Gollum.mac' directory to locate the folder 'Gollum.app' which contains the executable and the files 'readme.txt' and 'run_Gollum.sh'

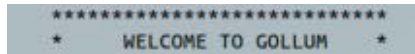
locate MATLAB_Compiler_Runtime

If you do not have the correct MCR then navigate to <http://www.mathworks.co.uk/products/compiler/mcr/> in MathWorks and locate the correct version to install.

Mac requires release Mac R2017a intel 64bit.

Once the MCR is correctly installed you can make your first test. Follow the instructions in the MATLAB readme file making sure that you place not only the 'run_Gollum.sh' but also the 'Gollum.app' in your test directory and type: `./run_Gollum.sh <mcr_directory>` where mcr_directory locates the MCR version you have installed.

If you have a compatible MCR, then you will see the following window:



There will be an 'Error message' at this stage which indicates that as yet you have not included the correct input files to run a calculation.

You can of course place 'Gollum' in your path using the Mac terminal in a similar way to the Linux procedure to run the code.

To run an example for simplicity we outline the following procedure:

1. Copy from the examples folder supplied with your distribution an example of your choice into a new directory. Place 'Gollum.app' and 'run_Gollum.sh' into the folder.

2. Type `./run_Gollum.sh <mcr_directory>`

Your results of the calculation should match those given in the examples.

5 TEST EXAMPLES AND TUTORIALS.

As a separate part of the tutorial distribution Dr Sadeghi has prepared a folder called 'Quick Start'. This will lead the user through an example, the details of which are shown in the file quick start.pdf, which will give an immediate grasp of how to use Gollum2. The tutorials and information in the following describe in greater detail the functionality of Gollum2.

Once the input data files relating to your system have been generated then the next task is to instruct Gollum exactly what you want to calculate by creating a suitable input file. This requires detailed knowledge of the system to be studied but to aid the user overcome the tedious task of writing their own input file the developers have created an input file generator. NOTE: This process is not automatic and you must understand the energy ranges, temperature ranges and variables that you use.

Gollum2 requires three files as a minimum to carry out a calculation: Leads_i where i indexes the number of leads in the simulation with the Extended_Molecule, which together provide the Hamiltonian data of the lead and scattering region (EM) respectively, in a Gollum2 compatible format (see Fig. 7.1.2 and the Examples) and the input file. The developers recommend that you use the input files given in the examples as templates as an aid to understanding how the input file instructs Gollum2 to carry out the required calculation.

The tutorials are sets of calculations located in 'examples' in your distribution. These are summarised in TABLE 5.1.

It is important to run each of the examples to ensure that you are happy with your compilation testing by comparing the results found in the output data 'mode' folders and the plots shown in the 'Guide.pdf'. Following all the examples and reading this document together with the Gollum ¹ paper will complete the tutorial.

5.1 THE GOLLUM INPUT FILE GENERATOR

One of the most important files needed to run Gollum is a suitable input file. You can either generate it using the variables description explained in the variable section by hand or use Gollum to generate an input file. If you run Gollum in a folder without an input file, Gollum will stop, generate a template start-up input file and return the following error message:

Error using read_input (line 43)

An "input" file has not been found! A blank "input" file has been created. Please run Gollum again in the presence of this blank "input" file to fill it up.

Error in Gollum (line 61)

Run Gollum again in the presence of this template start-up input file and Gollum enters into the **input file generator mode** by asking a list of questions which will enable you to generate a correct input file. You need to answer these questions based on your specific system and to assist the user a question is asked accompanied by a short description. The following example in Table 5.1 can be used as a guide.

In the left column the black and blue text are questions and the example answers respectively.

Note: In the presence of a Gollum input file, Gollum will not enter to the input file generator mode but start transport calculations based upon the input file provided.

TABLE 5.1

<i>Gollum Input Generator 2</i>	Comments
<i>Mode</i> = 1	The number of mode 0,1, 2, 3, 4 or 5. Described in sections 5.2 and 7.6
<i>E range</i> [<i>Emin Emax nE</i>] = [-3 3 1001]	Energy range for transport calculation. <i>Emin</i> : minimum energy, <i>Emax</i> : maximum energy, <i>nE</i> : number of points
<i>Number of atoms in the system</i> = 120	Total number of atoms in the extended molecule
<i>Number of Leads</i> = 2	Number of leads
<i>Hamiltonian Provider</i> (TBM or DFT) = dft	Hamiltonian Provider: tight-binding or DFT
<i>path/filename of Extended_Molecule file</i> = ../em.out	If Hamiltonian provider is DFT: Path to output file of the extended molecule calculation using Siesta or Wannier calculation If Hamiltonian provider is TBM: Path to tight-binding Extended_Molecule file
<i>Atom range in lead 1: either [atom_labels] or [initial:final]</i> = [1:10]	Atom range in lead 1: atom index if ordered [initial:final] meaning atom index range in lead 1 e.g. [1:10], if not ordered [atom_labels] meaning atom index in lead 1 e.g. [1:5,11,14,17,25,26]
"leadp" variable for lead 1: [No_PL terminating_PL separate_lead_Hamiltonian] = [2 2 0]	"leadp" variable in section 7.8
Closest atom in this atom range to the EM? "1" if the first atom or "2" if the last one = 2	In the above given atom range for lead 1, which atom is closer to the scattering region? E.g. 1 or 10 in the above example. If atom 1 (first atom) is closer type "1" if atom 10 (last atom) is closer type "2"
<i>Atom range in lead 2: either [atom_labels] or [initial:final]</i> = 111:120	Instruction is the same as lead 1 (above)
"leadp" variable for lead 2: [No_PL terminating_PL separate_lead_Hamiltonian] = [2 2 1]	
Closest atom in this atom range to the EM? "1" if the first atom or "2" if the last one = 1	
<i>path/filename of Lead 2 file</i> = ../lead/em.out ** warning! separate lead hamiltonian may be used in incorrect way.	If lead Hamiltonian read from separate file (see leadp variable description for more detail), here either the path to the output file need to be given in the case of DFT or the path to the Lead_2 file need to be given in the case of TBM
<i>Optional variables</i> 0 = none 1 = Gate potential profile	These are optional variables in section 7.6.9

2 = scissors 3 = EF_shift 4 = Outputing 5 = Lead GF Tolerance 6 = DOS calculation 7 = all Optional variables = 7	
Gate potential profile vector [vector with the length of number of atom] = <code>ones(1,120)</code>	Gate potential profile vector, this could be large for large numbers of atoms. For a non-uniform profile, we recommend you create it first in a separate text file and then copy and paste. You can use Matlab commands to create this vector e.g. <code>ones(1,120)</code> creates a column vector of ones with the length of 120 (number of atoms). Section 7.6.14
gate voltage factor = 0.2	gate voltage factor. Section 7.6.14
scissors correction [0_or_1 No_electron HOMO_shift LUMO_shift Screening_length] = <code>[1 10 -0.5 0.4 10000]</code>	scissors correction. Section 7.6.12
atom numbers that belong to the "Molecule" in the EM region [atom_labels] or [initial:final] = 61:70	This is to set right range of atoms for scissors correction. Here the atom index belongs to the molecule (that the number of electrons given above, e.g. "No_electron") should be given.
Fermi energy shift = -0.1	By default the DFT Fermi energy shifted to 0. Here you could change it.
Full output 1, otherwise 0 = 1	To see a full Gollum output insert 1, default is 0.
Lead GF Tolerance [$> 1e-5$] = $1e-7$	Tolerance to check if ingoing and outgoing channels are equal. In rare cases some systems are sensitive to this. You may change it with care. Default value is $10^{-7} = 1e-7$. See ... for detail
DOS calculation no = 0, yes = 1: 0	Flag to calculate DOS in mode 0 only. Section 5.2
An "input" file has been generated. Please run Gollum again to start your calculation.	

5.2 OVERVIEW

As a preamble, here is some basic information relating to the Mode command. More details are to be found in sections 7 and 8.

Mode-0: New Feature

In this mode, Gollum2 computes the band structure, the number of open channels $NOC(E)$ and optionally the density of states $DOS(E)$ as a function of energy E of a given lead or set of leads.

The advantages of this new feature are that the calculation is faster and that only the Hamiltonian of the

desired lead needs to be supplied (e.g.: there is no need to supply the Hamiltonian of the Extended Molecule, or of other non-interesting leads). As a hint in case that the DOS is computed, we suggest to increase the number of energy points to eliminate spurious spikes in the DOS.

Mode-1

In this mode, Gollum2 computes the equilibrium transport properties of a junction: the transmission $T(E)$, $NOC(E)$ and Shot Noise $SN(E)$ as a function of energy E .

Mode-2

In this mode, Gollum2 computes the zero-voltage, zero-energy (referred to E_F) spin-dependent Scattering matrix and Transmission coefficients.

Mode-3: New Feature

In this mode, Gollum2 computes the zero-voltage thermo-electric properties of the junction as a function of temperature. Such properties include the Peltier Coefficient, the Thermal conductance, the Thermopower, the Seebeck Coefficient and the Figure of Merit.

Important improvements now enable the user to calculate those spin-dependent thermo-electric properties over a temperature sweep or an energy sweep based upon the Fermi Energy, and under the assumption of charge-spin separation or non-separation.

Mode-4: New Feature

In this mode, Gollum2 determines the finite-voltage transmission coefficients and current-voltage $I-V$ curves of multi-lead junctions. Now the user can implement its own potential profile to describe the bias and/or gate voltages.






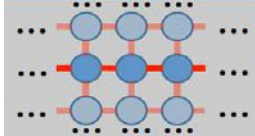
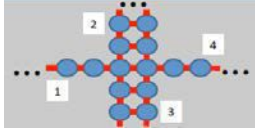

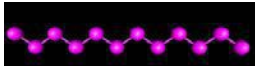


Mode-5: New Feature

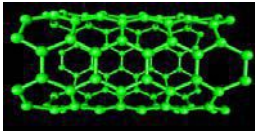

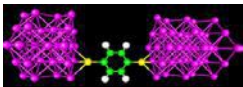
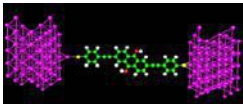






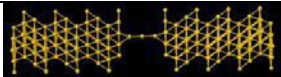
In this mode Gollum2 determines the phonon transmission and thermal conductance.

The force constant matrix must be calculated along with the corresponding xyz coordinates for each relaxation step. Using Siesta to do this you must edit the input file of Siesta (i.e. the fdf file) as follows. Type the line "MD.TypeOfRun fc" and set the displacement. The default value in Siesta is at 0.04 Bohr but we suggest changing this to a smaller value by including another line such as "MD.FCDispl 0.0189 Bohr".

The calculation produces a file labelled a file "systemlabel.FC" which contains the force constant matrix. To produce the coordinates for each relaxation step include the line "WriteCoorXmol T". This prints the xyz coordinates as a separate file "systemlable.xyz" needed for mode 5. See the example folders for various Siesta input "systemlabel.fdf" files. Note that you need to specify a frequency and a temperature range for the phonon transmission and thermal conductance calculations, respectively.

SUMMARY OF THE EXAMPLES: TABLE 5.2.

Guide-based information.		
Tight-binding	System	Comment
0.Single-level		In this simplest example of a single level some detail of the theory accompanies the calculations.
1.Single-atom_chain.Spin		Simulate a spin-polarized one-dimensional (1D) system with perfect transmission equal to the number of open channels or bands at a given energy.
2.Single-atom_chain.Non-collinear magnetism		Simulate a one-dimensional (1D) system with perfect transmission equal to the number of open channels or bands at a certain energy and non-collinear magnetic configuration.
3.Two-level_system.SAINT+ V_{gate}		Simulate a two level system (H2 molecule) coupled to metallic atomic chains. Check that the transmission has two peaks corresponding to the positions of the levels of the molecule. Apply a gate voltage to change the on site energies and the SAINT correction to increase the separation between both molecular levels.
4.Two-level_chain		Simulate a one-dimensional (1D) system with different orbitals and perfect transmission equal to the number of open channels or bands at a given energy.
5.Square_lattice		Simulate a two-dimensional (2D) system with perfect transmission equal to the number of open channels or bands at a given energy.
6.Four-probe_system		Simulate a four-probe system with two types of electrodes. Calculate the transmissions from one to other electrodes. See the effect of including a bias voltage on one of the electrodes. Calculate the currents.
Siesta-based	System	Comment
1.Au_linear_chain		Simulate a one dimensional (1D) system with perfect transmission equal to the number of open channels or bands at a certain energy.
2.Ir_zigzag_chain		Simulate a one dimensional (1D) system with perfect transmission equal to the number of open channels or bands at a certain energy, various channels and a non collinear magnetic configuration.
3.C_chain+gap.NDR		Simulate a one-dimensional (1D) system with a vacuum gap to see the effect of including a bias voltage that falls in the middle.
4.Ni_001_bulk		Simulate a magnetic three-dimensional (3D) system (transverse periodic boundary conditions and k -points) with perfect transmission equal to the number of open

		channels or bands at a certain energy and various channels.
5.CNT_5,5		Simulate a quasi-one-dimensional (q1D) system with perfect transmission equal to the number of open channels or bands at a certain energy and various channels.
6.Graphene		Simulate a perfect two-dimensional (2D) system transverse periodic boundary conditions and k -points along y with perfect transmission equal to the number of open channels or bands at a certain energy and various channels.
7.Au_001+BDT.SAINT		BDT (1,4-benzenedithiolate) molecule between gold electrodes. Correct the transport properties by applying a spectral adjustment (SAINT) to the occupied and unoccupied levels and shifting the Fermi level towards the HOMO.
8.Au_111+OPE		An OPE (oligophenyleneethylene) based molecule between gold electrodes.
Wannier	System	Comment
1.Na chain		A Sodium linear chain.
2.Na chain + gap		A Sodium linear chain with a defect (gap)
3. Au chain		Slightly zig-zag Au chain to lift degeneracy of Wannier functions.
4. Au chain + gap		Slightly zig-zag Au chain to lift degeneracy of Wannier functions with defect (gap).
5.(5,0) CNT		Quasi 1D system, which is metallic due to curvature effects.
6.(5,0) CNT + Si		Quasi 1D system, which is metallic due to curvature effects, with a substituted Si impurity.
phonon	System	Comment
1.Au chain + Au_111 leads		Phonon transmission and thermal conductance in a gold chain.

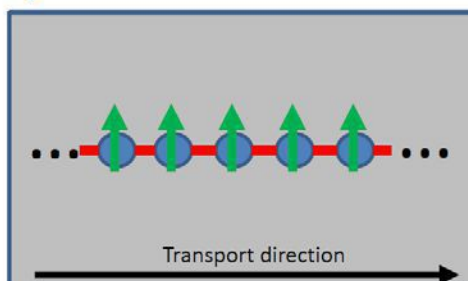
5.3 STEP BY STEP TIGHT-BINDING-BASED EXAMPLE: 1. SINGLE-ATOM_CHAIN.SPIN

Enter the directory: examples\Tight-binding\1.Single-atom_chain.Spin from your distribution. You will find the file 1.TB.Guide.pdf that contains the description below on this example system.

Single-atom chain. Spin polarized

Objectives

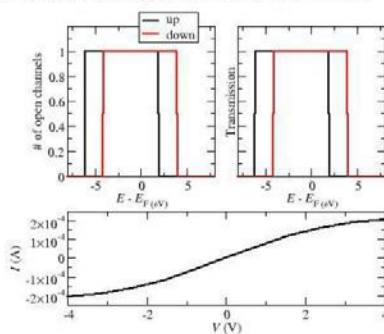
- Simulate a spin-polarized one-dimensional (1D) system with perfect transmission (equal to the number of open channels or bands at a certain energy).
- Check that the transmission exactly coincides with the number of open channels.



Results

- The transmission has a step-like shape which corresponds to the number of open channels.
- The spin-up and spin-down channels and transmissions have an exchange splitting of 2 eV.
- The current linearly increases (as expected, since it is equal to the integral of a constant function) until it saturates at large voltages (when the bias window starts covering the edges of the band, which are "rounded" by the effect of the voltage).

Transmission and number of open channels as a function of energy. Calculated with **Mode 1** of Gollum.



Current as a function of voltage. Calculated with **Mode 4** of Gollum.

JF. 2014

System description and parameters

Leads

- Unit cell with just one atom. One orbital per atom.
- Single-atom chain with spin polarization. The exchange splitting between the spin-up and spin-down states is 2 eV.

Extended molecule (EM)

- 5 atoms in the calculation.
- Same parameters as in the leads.

Gollum parameters

- Transmission coefficients calculated between -8.0 eV and 8.0 eV in 200 energy points.
- 2 principal layers on each electrode. The terminating principal layer is the second on each. Electronic structure obtained from the leads calculation.
- No SAINT correction or Fermi level shift.
- Bias voltage between -4.0 V and 4.0 V calculated in 11 voltage points. The bias shift is applied on the first terminating layer of the left and right electrodes only (4th column of the `atom` variable set to different from 0). This is an approximation since the voltage should not fall (or at least fall very slowly) on a perfect infinite system.

Tight-binding parameters

- Orthogonal basis set (overlap matrix = identity matrix).
- On-site energies (both leads and EM): -1.0 eV (spin-up) and 1.0 eV (spin-down).
- Nearest-neighbour coupling matrix elements (both leads and EM): -2.0 eV.

Transfer the files: 'input', 'Lead_1', 'Lead_2' and 'Extended_Molecule' to a new test folder of your choice. Edit the file 'input' and check that the paths to files 'Lead_1', 'Lead_2' and 'Extended_Molecule' are correct. Execute Gollum as described in SECTION 4 INSTALLATION and RUNNING and check your results.

The 'input' file is defined in SECTION 7.6. The 'Extended_Molecule' file is defined in SECTION 7.7, and contains the tight-binding Hamiltonian of the extended molecule. Similarly, the files 'Lead_1' and 'Lead_2', that are described in SECTION 7.8, contain the tight-binding Hamiltonian of the two leads.

The output data files are given for the various 'mode' directories and are described in SECTION 8.

5.4 STEP BY STEP SIESTA-BASED EXAMPLE: 1.AU_LINEAR_CHAIN

Enter the directory: examples\Siesta-based\1.Au_linear_chain.

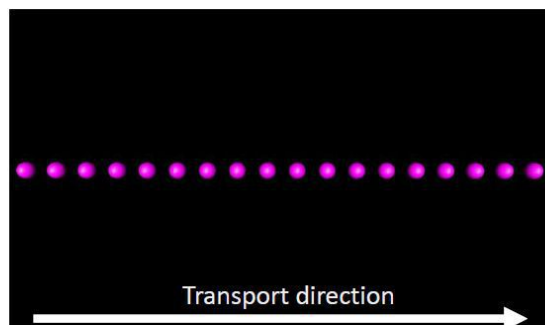
Gold linear chain

Objectives

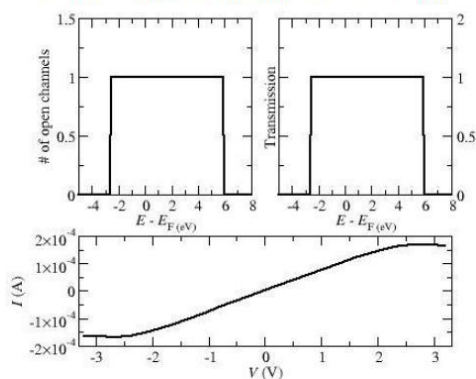
- Simulate a one-dimensional (1D) system with perfect transmission (equal to the number of open channels or bands at a certain energy).
- Check that the transmission exactly coincides with the number of open channels.

Results

- The transmission has a step-like shape which corresponds to the number of open channels. In this case there is only one s orbital in the valence of gold (the rest of states are included in the pseudopotential), so the transmission is equal to 1 in the range of energies that comprises the gold band.
- The current linearly increases (as expected, since it is equal to the integral of a constant function) until it saturates and slightly decreases (when the bias window starts covering the edges of the band, which are "rounded" by the effect of the voltage).



Transmission and number of open channels as a function of energy. Calculated with **Mode 1** of Gollum.



Current as a function of voltage. Calculated with **Mode 4** of Gollum.

System description and parameters

Leads

- Au leads made of only 2 gold atoms.
- Lattice vectors long enough along the transverse directions to make sure the system is 1D.

Extended molecule (EM)

- Perfect gold chain made of 18 atoms (9 unit cells of the electrodes) and with the same transverse lattice vectors of the leads.
- The system is also periodic along the transport direction (z) to avoid finite size effects.

Gollum parameters

- Transmission coefficients calculated between -5.0 eV and 8.0 eV in 200 energy points.
- 2 principal layers on each electrode. The terminating principal layer is the second on each. Electronic structure obtained from the leads calculation.
- No SAINT correction or Fermi level shift.
- Bias voltage between -3.2 V and 3.2 V calculated in 25 voltage points. The bias shift is applied on the first terminating layer of the left and right electrodes only (4th column of the `atom` variable set to different from 0). This is an approximation since the voltage should not fall (or at least fall very slowly) on a perfect infinite system.

Ab-initio (Siesta) parameters

- Basis set: Single- ζ (SZ); only the s orbital is included in the Au valence (the rest are hidden in the pseudopotential).
- LDA exchange and correlation functional. CA parametrization.
- Mesh cut-off: 150 Ry.
- 90 k-points along z in the lead calculation. 10 k-points in the EM calculation (to make sure the electronic structure exactly matches that of the leads). 1 k-point (Γ point) along the perpendicular directions.

In this case the definitions of the Hamiltonians are generated from ab-initio DFT parameters using SIESTA and are shown in 'Gollum_files\Leads_1 and 2' with the 'Extended_Molecule'. These files are generated using the method explained in SECTION 6 HAMILTONIAN GENERATION and in SECTION 6.2 DIRECT SIESTA INTERFACE METHOD. This makes use of the files 'lead.fdf' and 'emol.fdf' with 'Au.psf' provided within the distribution. You will need to obtain a licenced SIESTA version (<http://departments.icmab.es/leem/siesta/>) and understand how to run the code from the documentation.

Again transfer the files: 'input, Lead_1, Lead_2 and Extended_Molecule' to a new test folder. Execute Gollum as described in SECTION 4 INSTALLATION and RUNNING and check your results.

The 'input' file is defined in SECTION 8. The 'Extended_Molecule' file is defined in SECTION 9. The 'Lead_1 and Lead_2' files are defined in SECTION 10. The output data files are given for the various 'mode' directories and are described in SECTION 11.

5.5 AIIDA PLUGIN FOR GOLLUM

AiiDA (<http://www.aiida.net/>) is a flexible and scalable informatics infrastructure to manage, preserve, and disseminate the simulations, data, and workflows of modern-day computational science. It is interfaced to a series of ab-initio codes or related programs that allow to automate and store large amounts of simulations of materials and other types of data. The current version of Gollum can run under AiiDA, as explained in the webpage <http://aiida-gollum.readthedocs.io/en/latest/>. We give in the following a brief description of the main capabilities of the plugin.

5.5.1 INSTALLING THE PLUGIN

We assume the user has already installed the 0.12.0 version of AiiDA, which is the one under which the plugin fully works.

Download the plugin from <https://github.com/garsua/aiida-gollum/> and install it by doing from the top level directory 'pip install -e .' . Run also 'reentry scan -r aiida' to make sure all other plugins are discovered and registered.

5.5.2 STANDARD GOLLUM PLUGIN

The plugin has two main parts, the input and the output, which facilitate the design, analysis and postprocessing of transport calculations within the AiiDA framework.

The inputs should include some settings (the name of the localfolder that contains the Extended_Molecule and Lead_* input files, in case they are necessary, and the path of the -remote- directory where the Matlab MCR is located) and parameters (a dictionary with scalar and string variables and blocks, which are the basic elements of the Gollum input file). Here, the definition of each parameter is simpler than in the Gollum original input file (it is not necessary to specify the type of variable and the number of rows and columns). Complex data structures such as blocks with various rows and columns are defined by using an appropriate key and Python's multiline string constructor. There are three types of blocks: numerical (NBlock), string (SBlock) and the atom block (atom).

The output parser gets information from the Gollum output file (where the output of the run is redirected) and from data files generated by the run (transmission and open-channels files). The data include the number of open channels at the Fermi level (oc_ef), the maximum (oc_M) and minimum (oc_m) open channels, the transmission at the Fermi level (tt_ef) and the maximum (tt_M) and minimum (tt_m) transmission. All these values are converted to 'float'. The parser also distinguishes between spin-unpolarized and -polarized calculations. In the former case it gives the values ou_ef, ou_M, ou_m, tu_ef, tu_M and tu_m, for the up open channels and transmission and od_ef, od_M, od_m, td_ef, td_M and td_m for the down open channels and transmission, respectively.

The warnings list contains program warnings which do not stop the execution of Gollum. The errors list contains the last line of the output file when the execution stops for errors or for external reasons.

The output also includes array-data, which contains the open channels and the transmission in an array form (the energy in the x axis and the open channels or transmission in the y axis).

5.5.3 GOLLUM-SIESTA WORKFLOW

The Gollum-Siesta workflow produces files with the transmission and the number of open channels from the electronic structure calculated with Siesta.

Note that, in order for this workflow to work, it is also necessary to install the aiida_siesta plugin (<http://aiida-siesta-plugin.readthedocs.io/en/latest/>).

The inputs to the Gollum workflow include the Siesta code, the Gollum code, the structures of the leads and the extended molecule, the protocol (fast or standard Siesta calculation), the number of k-points in the leads and the extended molecule and some parameters for Gollum which are not defined in the workflow file.

The outputs include the number of open channels of the first electrode (we assume at the moment that both electrodes are equal) and the transmission between electrodes. In case of a spin-polarized calculation the output distinguishes between spin-up and down channels and transmission.

6 HAMILTONIAN GENERATION.

The Tight Binding Hamiltonians can be constructed by hand to generate the 'Lead_i' and 'Extended_Molecule' input files used by Gollum2, nevertheless we strongly recommend that the Hamiltonians are generated through a suitable DFT code. Gollum2 reads directly the Hamiltonians generated as output of SIESTA and those generated by Wannier90 from plane-wave codes.

7 THEORETICAL APPROACH

For a full description please refer to the Gollum¹ paper.

7.1 NOMENCLATURE.

Gollum describes open systems comprising an extended scattering region (coloured dark blue in FIGS. 7.1.1 and 7.1.2) connected to external crystalline leads (coloured light blue in FIGS. 7.1.1 and 7.1.2). Depending on the problem of interest and the language used to describe the system, the material (M) of interest forming the central part of the scattering region could comprise a single molecule, a quantum dot, a mesoscopic cavity, a carbon nanotube, a two-dimensional mono or multi-layered material, a magneto-resistive element or a region containing one or more superconductors. FIG. 7.1.1 shows an example of a 4-lead system whose central scattering region (labeled M) is a molecule. It is important to note that in an accurate ab initio description of such a structure, the properties of the leads closest to the molecule (or more generally the central scattering material) will be modified by the presence of the central scattering region (M) and by the fact that the leads terminate. In what follows, we refer to those affected portions of the leads closest to the central scatterer as 'branches' and include them as part of the 'extended scatterer' (denoted EM). Consequently within Gollum, a typical structure consists of an extended scatterer (EM), formed from both the central scatterer (M) and the branches. The extended scattering region is connected to crystalline current-carrying leads of constant cross-section, shown in light blue in the FIGS. 7.1.1 and 7.1.2. For an accurate description of a given system, the branches are chosen to be long enough such that they join smoothly with the (light blue) crystalline leads. Crucially, the properties of this interface region between the central scatterer M and the leads are determined by their mutual interaction and are not properties of either M or the electrodes alone.

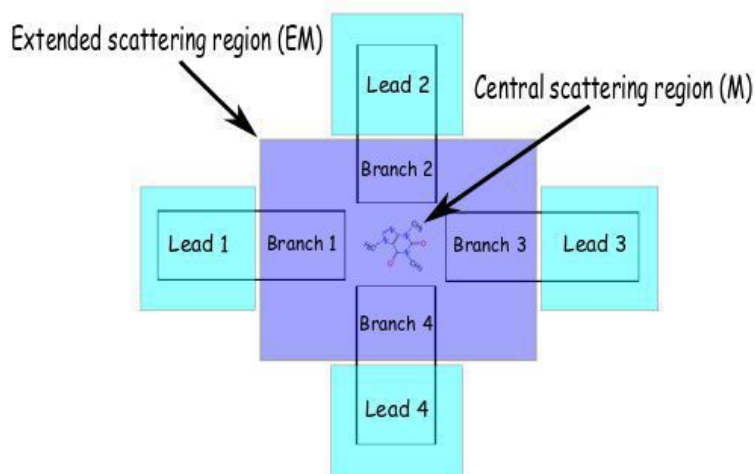


FIG. 7.1.1: Schematic plot of a four-terminal device, which includes an extended scattering region and four leads.

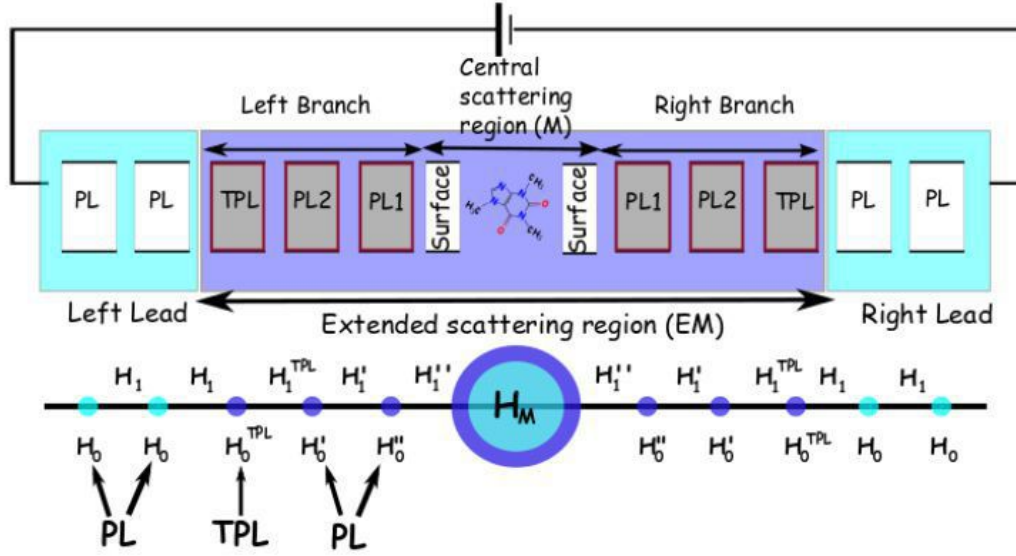


FIG. 7.1.2: (Top) Schematic two-terminal device, where electrons are driven from the left to the right lead through the extended scattering region. The leads are possibly kept at chemical potentials $\mu_{1,2} = -eV_{1,2}$, where $V = V_1 - V_2$ is the applied bias. (Bottom) Each lead is composed of an infinite chain of PLs with Hamiltonians H_0 , which are coupled with each other via the coupling Hamiltonians H_1 . The Extended Molecule comprises the actual Molecule, the electrodes surfaces and the Left and Right Branches. These branches contain several PLs up to the TPL. The TPL link the EM to the leads. Both leads are assumed to be identical in this figure for simplicity.

FIG. 7.1.2 shows a two-terminal device in more detail and introduces further terminology. The regions in light blue are called electrodes or leads and are described by perfect periodic Hamiltonians subject to chosen chemical potentials. Each lead i is formed by a semi-infinite series of identical layers of constant cross section, which we refer to as principal layers (PLs). FIG. 7.1.1 shows only two PLs per lead (coloured white), although an infinite number is implied. Furthermore in the figure, the leads are identical and therefore the lead index i has been dropped. These PLs are described mathematically by intra-layer Hamiltonians H_0^i . PLs must be chosen so that they are coupled only to their nearest neighbors by the Hamiltonians H_1^i . If each PL contains N^i orbitals then H_0^i and H_1^i are square $N^i \times N^i$ matrices.

The extended scatterer (EM) in dark blue is composed of a central scattering region (M) and branches. Each branch contains several PLs. These PLs have an identical atomic arrangement as the PLs in the leads. However, their Hamiltonians differ from H_0^i and H_1^i due to the presence of the central scattering region. PL numbering at each branch starts at the PL beside the central scattering region. The outermost PL at each branch of the EM region is called the terminating principal layer (TPL) and must be described by Hamiltonians $H_0^{i,TPL}$ and $H_1^{i,TPL}$ which are close enough to H_0^i and H_1^i , to match smoothly with the corresponding lead Hamiltonian. For this reason, Gollum requires that the EM contain at the very least one PL. The central scatterer (M) itself is described by an intra-scatterer Hamiltonian H_M^0 and coupling matrices to the closest PLs of the branches. In the example in FIG. 7.1.1, the central scattering region M comprises a molecule and the atoms forming the electrode surfaces. The surfaces in Gollum include all atoms belonging to the electrodes whose atomic arrangements cannot be cast exactly as a PL, due to surface reconstructions, etc. For simplicity, FIG. 7.1.1 shows the case of a symmetric system, although no such symmetries are imposed by Gollum. All Hamiltonians are spin-dependent, but again for notational simplicity, the spin index σ is not written explicitly.

7.2 THE SURFACE GREEN'S FUNCTION FOR THE CURRENT CARRYING LEADS.

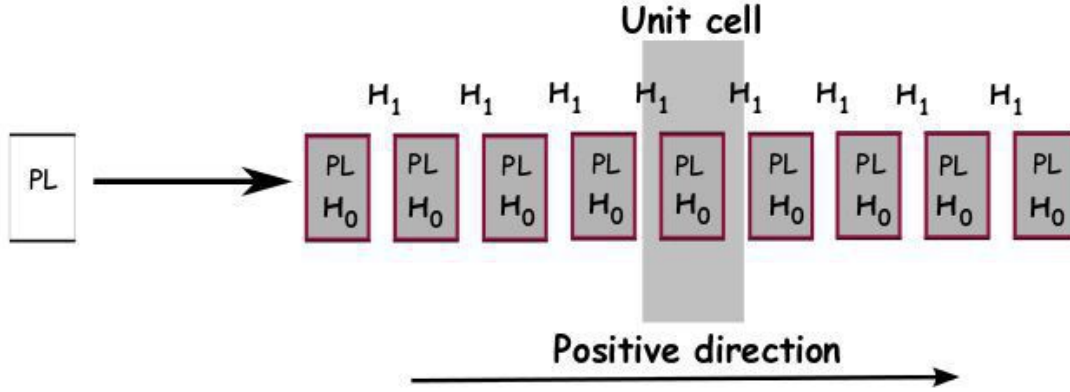


FIG.7.2.1 Illustrates the infinite system used to generate the leads Hamiltonians K_0^i and K_1^i where the positive direction is towards the scatterer. For non-orthogonal basis sets the overlap matrices ($S_{o,\pm 1}^i$) and (S^i) must have the same structure as the Hamiltonian matrices and

$$K_{o,\pm 1}^i = H_{o,\pm 1}^i - ES_{o,\pm 1}^i \quad (1)$$

$$K^i = H^i - ES^i \quad (2)$$

By expanding the Bloch eigenstates $|\Psi(k)\rangle = \sum_{n,\mu} e^{ikn} c_\mu(k) |\varphi(n,\mu)\rangle$ of the infinite system in a localised basis set a $N \times N$ secular equation can be found $(K_0 + K_1 e^{ik} + K_{-1} e^{-ik})C(k) = 0$ which if by choosing the energy (E) and solving for the allowed wave vectors produces $2N$ solutions with either real or complex wave vectors (k_p $p = 1 \dots 2N$) which Gollum solves to give the eigenvalues and eigenvectors $C(k_p)$.

$$\begin{pmatrix} -K_0 & -K_{-1} \\ I_N & 0_N \end{pmatrix} C(k_p) = e^{ik_p} \begin{pmatrix} K_1 & 0_N \\ 0_N & I_N \end{pmatrix} C(k_p) \quad (3)$$

The group velocities of the states corresponding to the real wave vectors is then given by

$$v(k_p) = i \frac{C(k_p)^\dagger (K_1 e^{ik_p} - K_{-1} e^{-ik_p}) C(k_p)}{C(k_p)^\dagger (S_0 + S_1 e^{ik_p} + S_{-1} e^{-ik_p}) C(k_p)} \quad (3a)$$

$v(k_p)$ has units of energy and is real and the fully-dimensioned group velocity is given by $v(k_p)(a/h)$ where a is the spacing between PL's in a given lead. The $2N$ wave vectors are split into two sets the first denoted k_p are real (complex) wave vectors that have $v_p > 0$ ($\text{Im } k_p > 0$) and therefore propagate (decay) towards the right of FIG. 7.2. These are called positive open (closed) channels. The second set are denoted \tilde{k}_p are real (complex) wave vectors that have $v_p < 0$ ($\text{Im } k_p < 0$) and therefore propagate (decay) towards the left of FIG. 5. These are called negative open (closed) channels. By using the dual vectors $D(k_p)$ and $D(\tilde{k}_p)$ which satisfy $D(k_p)^\dagger \cdot C(k_q) = \delta_{p,q}$ and $D(\tilde{k}_p)^\dagger \cdot C(\tilde{k}_q) = \delta_{p,q}$ which are found by inverting the $N \times N$ matrices:

$$Q = (C(k_1), \dots, C(k_N)) = (C_1, \dots, C_N)$$

$$\tilde{Q} = (C(\tilde{k}_1), \dots, C(\tilde{k}_N)) = (\tilde{C}_1, \dots, \tilde{C}_N)$$

$$(D_1, \dots, D_N) = (Q^{-1})^\dagger$$

$$(\tilde{D}_1, \dots, \tilde{D}_N) = (\tilde{Q}^{-1})^\dagger$$

the required transfer matrices are constructed:

$$T = \sum_1^N C_n e^{i k_n} D_n^\dagger \quad (3b)$$

$$\tilde{T} = \sum_1^N \tilde{C}_n e^{i \tilde{k}_n} \tilde{D}_n^\dagger \quad (3c)$$

The transfer matrices allow us to build the coupling matrix V , the self-energies Σ and the surface Green's functions $G_{S,0}^i$:

$$V = K_{-1}(T^{-1} - \tilde{T}) \quad (4)$$

$$\Sigma = K_1 T \quad (5)$$

$$G_{S,0}^i = -(K_0 + \Sigma)^{-1} \quad (6)$$

NOTE the procedure might fail if K_1 is singular. We adopt a decimation method to remove the offending degrees of freedom (see the Gollum paper for details [1]).

7.3 THE EXTENDED SCATTERING REGION HAMILTONIAN

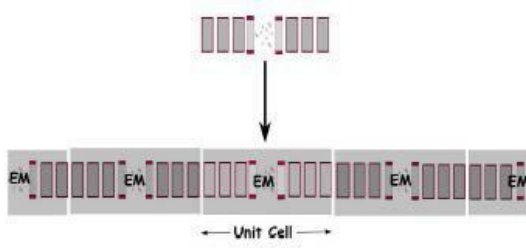


FIG.7.3.1. The infinite system where the unit cell is linked by periodic boundary condition.

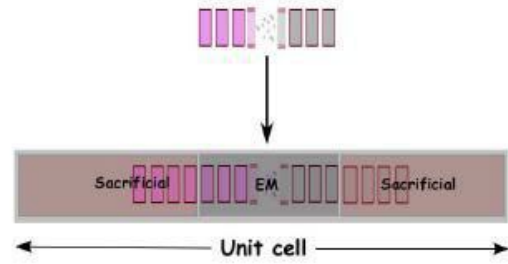


FIG.7.3.2. The super-cell containing the EM with vacuum buffer regions to left and right.

The extended scattering region Hamiltonian K^{EM} can be provided as a model Hamiltonian, or generated by a DFT or other material-specific program. One of the strengths of Gollum is an ability to treat interfaces with high accuracy. In a tight-binding description, tight-binding parameters of a particular material are often chosen by fitting to a band structure. However this does not solve the problem of choosing parameters to describe the interface between two materials. Often this problem is finessed by choosing interface parameters to be a combination of pure material parameters such as an arithmetic or geometric mean, but there is no fundamental justification for such approximations. Therefore we describe here methods to generate K^{EM} using a DFT program, where the inclusion of branches as part of the extended scatterer occurs naturally. FIG.7.3.1 shows an example of a junction where the electrodes are identical. The system is composed of super-cells formed from a central scatterer and PLs. There are periodic boundary conditions in the longitudinal direction, such that the TPL of one branch of a super-cell is linked smoothly to the TPL of a neighboring super-cell. Running a DFT program for such a super-cell then automatically generates K^{EM} . Provided the super-cells contain sufficient PLs, the Hamiltonians K_0^{TPL} and K_1^{TPL} associated with the TPLs will be almost identical to those generated from a calculation involving an infinite periodic lead, i.e. if the Hamiltonians K_0 and K_1 associated with the PLs are generated from a calculation involving an infinite periodic lead, then provided the super-cells contain sufficient PLs, these will be almost identical to K_0^{TPL} and K_1^{TPL} respectively. In this case then there will be minimal scattering caused by the junction between the TPL and the lead. Clearly there is a trade-off between accuracy

and CPU time, because inserting more PLs increases the size and cost of the calculation. In practice, the number of PLs retained in such a super-cell is increased in stages until the results do not change significantly as the number of PLs is increased further. There exist situations where the electrodes are dissimilar, either chemically, or because of their different crystalline structure, or because their magnetic moments are not aligned. In these cases, there cannot be a smooth matching between TPLs of neighboring super-cells in FIG.7.3.1. To address this situation, we use a setup similar to that displayed in FIG.7.3.2, where additional PLs are appended to the branches in the EM region. These additional PLs are terminated by artificial surfaces and surrounded by vacuum. The TPLs are then chosen to be one of the PLs near the middle of each branch and should be surrounded by enough PLs both towards its artificial surface and towards the central scattering region. Then the PLs placed between the TPL and the artificial surface can be discarded. These sacrificial PLs ensure that the chosen TPL is unaffected by the presence of the artificial vacuum boundary. Clearly, calculations of this sort are more expensive in numerical terms than those performed with super-cells generated as in FIG.7.3.1, because they contain many more atoms.

7.4 HAMILTONIAN ASSEMBLY

In an ab initio calculation of the transport properties of a junction, the DFT program produces the Hamiltonians and Fermi energy of the EM and leads in separate runs as the Hartree potential is defined up to a constant, which is usually different for the EM and for each lead. This usually means that the energy of the EM and of the corresponding lead PLs, as well as their Fermi energies do not agree with each other, so Eq. (1) must be rewritten as follows:

$$K_{o,\pm 1}^i = H_{o,\pm 1}^i - (E_F^i + E)S_{o,\pm 1}^i \quad (7)$$

where we have referred the energy of each lead to its own Fermi energy. To fix the Hamiltonian mismatch we define a realignment variable for each lead as follows:

$$\Delta^i = H_0^i(\mu, \mu) - H_0^{EM}(\mu, \mu) \quad (8)$$

where μ indicates a relevant orbital or group of orbitals. Then, the Hamiltonian of each lead is realigned with that of the EM

$$\begin{aligned} K_{o,\pm 1}^i &= H_{o,\pm 1}^i - (E_F^{EM} + E + \Delta^i)S_{o,\pm 1}^i \\ &= H_{o,\pm 1}^i - (E_F^i + E)S_{o,\pm 1}^i \end{aligned} \quad (9)$$

It turns out that the renormalized Fermi energies of each lead do not match perfectly with each other if the number of PLs in the EM region is not sufficiently large. This is the case when for efficiency reasons, it is desirable to artificially minimize the size of the EM. Sometimes it is advisable to choose the Fermi energy of one of the leads as the reference energy. In this case an overall shift is made by the quantity $\delta = E_F^1 - E_F^{EM}$.

7.5 SCATTERING MATRIX AND TRANSMISSION IN MULTI-TERMINAL DEVICES.

We note that the most general scattering state in a given lead i at a given energy E can be written as a linear combination of open and closed channels as follows

$$|\Phi^i(E)\rangle = \sum_{k_i} o_{k_i} \frac{|\Psi^i(k_i)\rangle}{\sqrt{v_{k_i}}} + \sum_{\bar{q}_i} o_{\bar{q}_i} \frac{|\Psi^i(\bar{q}_i)\rangle}{\sqrt{v_{\bar{q}_i}}} + |\chi^i\rangle \quad (10)$$

where k_i and \bar{q}_i denote here open positive and negative channels and $|\Psi^i(k_i)\rangle$ and $|\Psi^i(\bar{q}_i)\rangle$ are their normalized kets. Here, the contribution of all the closed channels in lead i is described by the ket

$|\chi^i\rangle$. Consequently, the number of electrons per unit time flowing between two adjacent PLs within the lead is $j^i(E) = \sum_{k_i \in i} |o_{k_i}|^2 - \sum_{\bar{q}_i \in i} |o_{\bar{q}_i}|^2$ (11)

We pick in this section the convention that positive direction in the lead means flow towards the EM region and vice versa. So positive (negative) open channels are also called incoming (outgoing) channels of lead i . With this notation the wave-function coefficients of the incoming open channels of a given lead are determined by the properties of the reservoir connected to the lead. The wave function coefficients of the open outgoing channels of lead i are obtained from the amplitudes of all incoming channels by

$$o_{\bar{q}_i}^i = \sum_{j,k} S_{\bar{q}_i k_j}^{ij} o_{k_j}^j \quad (12)$$

Where $\bar{q}_i(k_j)$ is an outgoing (incoming) dimensionless wave-vector of lead i or lead j . It is therefore convenient to assemble the wave functions of the M^i incoming (outgoing) open channels of a given lead i in the column vectors \bar{O}^i, O^i and all the scattering matrix elements connecting leads i and j into the matrix block S^{ij} of dimension $M^i \times M^j$. Therefore the above equation can be written

$$\begin{pmatrix} \bar{O}^1 \\ \bar{O}^2 \\ \dots \\ \bar{O}^P \end{pmatrix} = \begin{pmatrix} S^{11} & S^{12} & \dots & S^{1P} \\ S^{21} & S^{22} & \dots & S^{2P} \\ \dots & \dots & \dots & \dots \\ S^{P1} & S^{P2} & \dots & S^{PP} \end{pmatrix} \begin{pmatrix} O^1 \\ O^2 \\ \dots \\ O^P \end{pmatrix} \quad (13)$$

By normalizing the Bloch eigenvectors and their duals to unit flux by

$$C^i(k_i) = \frac{c^i(k_i)}{\sqrt{v_{k_i}}}, D^i(k_i) = \sqrt{v_{k_i}}(k_i) \\ C^i(\bar{q}_i) = \frac{c^i(\bar{q}_i)}{\sqrt{v_{\bar{q}_i}}}, D^i(\bar{q}_i) = \sqrt{v_{\bar{q}_i}}(\bar{q}_i) \quad (14)$$

The matrix elements of the scattering matrix block connecting leads i and j can then be written as

$$S_{\bar{q}_i k_j}^{ij} = D^i(\bar{q}_i) (G_S^{ij} V^i - I \delta_{ij}) C^i(k_i) \quad (15)$$

Here G_S^{ij} is the off diagonal block of the surface Greens function defined in Eq. 6 that connects leads i and j and V^i is also defined in Eq.6.

With the above if the incoming channel k_i of lead i is occupied with probability $f_{k_i}(E)$ (i.e. in Eq.10 $o_{k_i} = 1$ with probability $f_{k_i}(E)$), then the number of electrons per unit time entering the scattering region from reservoir i along channel k_i with energy between E and $E + dE$ is

$$dI_{\bar{q}_i}^{out}(E) = (dE/h) f_{k_i}(E) \quad (16)$$

And the number per unit time per unit energy leaving the scatterer and entering reservoir i along channel \bar{q}_i with energy between E and $E + dE$ is

$$dI_{\bar{q}_i}^{out}(E) = (dE/h) \sum_{j,k} |S_{\bar{q}_i k_j}^{ij}|^2 f_{k_i}(E) \quad (17)$$

In many cases, the incoming and outgoing channels of each lead i can be grouped into channels possessing particular attributes (i.e. quantum numbers) labelled α_i, β_i, \dots etc. This occurs when all incoming channels of a particular type α_i in lead i possess the same occupation probability $f_{\alpha_i}^i(E)$. For example, all quasi-particles of type α_i in reservoir i may possess a common chemical potential μ_{α_i} and $f_{\alpha_i}^i(E)$ may take the form $f_{\alpha_i}^i(E) = f(E - \mu_{\alpha_i})$, where $f(E)$ is the Fermi function. In this case, if the incoming and outgoing channels of type α_i

belonging to lead i possess wave-vectors $k_{\alpha_i}, \bar{q}_{\alpha_i}$, then the number of quasi-particles per unit time of type α_i leaving reservoir i with energy between E and $E + dE$ is

$$dI_{\alpha_i}^i(E) = (dE/h) \sum_{j, \beta_j} P_{\alpha_i \beta_j}^{i,j} f_{\beta_j}^j(E) \quad (18)$$

where

$$P_{\alpha_i \beta_j}^{i,j} = M_{\alpha_i}^i(E) \delta_{i,j} \delta_{\alpha_i \beta_j} - \sum_{\bar{q}_{\alpha_i} k_{\beta_j}} \left| s_{\bar{q}_{\alpha_i} k_{\beta_j}}^{ij} \right|^2 \quad (19)$$

and $M_{\alpha_i}^i(E)$ is the number of open incoming channels of type α energy E in lead i . Note that in the above summation \bar{q}_{α_i} runs over all outgoing wave vectors of energy E and type α_i of lead i and k_{β_j} runs over all incoming wave vectors of energy E and type β_j in lead j . If i and j are different leads then $s_{\bar{q}_i k_j}$ is often called the transmission amplitude and denoted $t_{\bar{q}_i k_j}$ whilst if they are of the same lead then $s_{\bar{q}_i k_i}$ is called the reflection amplitude $r_{\bar{q}_i k_i}$. Similarly if $i \neq j$ it is common to define the transmission coefficient as

$$T_{\alpha_i \beta_j}^{i,j} = \sum_{\bar{q}_{\alpha_i} k_{\beta_j}} \left| s_{\bar{q}_{\alpha_i} k_{\beta_j}}^{ij} \right|^2 \quad (20)$$

For $i = j$ we define the reflection coefficient as

$$R_{\alpha_i \beta_i}^{i,i} = \sum_{\bar{q}_{\alpha_i} k_{\beta_i}} \left| s_{\bar{q}_{\alpha_i} k_{\beta_i}}^{ij} \right|^2 \quad (21)$$

So that

$$dI_{\alpha_i}^i(E) = (dE/h) \left\{ \left[\sum_{\beta_i} M_{\alpha_i}^i(E) \delta_{\alpha_i \beta_i} - R_{\alpha_i \beta_i}^{i,i} \right] f_{\beta_i}^i(E) - \sum_{j \neq i, \beta_j} T_{\alpha_i \beta_j}^{i,j} f_{\beta_j}^j(E) \right\} \quad (22)$$

Note the unitarity of the scattering matrix requires that

$$\sum_{i, \bar{q}_{\alpha_i}, \alpha_i} \left| s_{\bar{q}_{\alpha_i} k_{\beta_j}}^{ij} \right|^2 = \sum_{j, k_{\beta_j}, \beta_j} \left| s_{\bar{q}_{\alpha_i} k_{\beta_j}}^{ij} \right|^2 = 1 \quad (23)$$

Hence the sum of the elements of each row and column of matrix P is zero:

$$\sum_{j, \beta_j} P_{\alpha_i \beta_j}^{i,j} = \sum_{i, \alpha_i} P_{\alpha_i \beta_j}^{i,j} = 0 \quad (24)$$

Or equivalently,

$$\sum_{\beta_i} R_{\alpha_i \beta_i}^{i,i} + \sum_{j \neq i, \beta_j} T_{\alpha_i \beta_j}^{i,j} = M_{\alpha_i}^i \quad (25)$$

and

$$\sum_{\alpha_i} R_{\alpha_i \beta_j}^{j,j} + \sum_{j \neq i, \alpha_i} T_{\alpha_i \beta_j}^{i,j} = M_{\beta_j}^j \quad (26)$$

From Eqs. 23 and 24 if $f_{\beta_j}^j(E)$ is independent of j and β_j then $dI_{\alpha_i}^i(E) = 0$ for all i and α_i . For this reason in the above equations $f_{\beta_j}^j(E)$ can be replaced by $\bar{f}_{\beta_j}^j(E) = f_{\beta_j}^j(E) - f(E)$ where $f(E)$ is an arbitrary function of energy which in practice is usually chosen to be the Fermi function evaluated at a convenient reference temperature and chemical potential. When comparing theory with experiment we are usually interested in computing the flux of some quantity Q from a particular reservoir. From Eq. 18 if the amount of Q carried by quasi particles of type α_i is $Q_{\alpha_i}(E)$ then the flux of Q from reservoir i is

$$I_Q^i = \int (dE/h) \sum_{\alpha_i, \beta_j} Q_{\alpha_i}(E) P_{\alpha_i \beta_j}^{i,j} \bar{f}_{\beta_j}^j(E) \quad (27)$$

In the simplest case of a normal conductor choosing $Q_{\alpha_i} = -e$ independent of α_i then the above equation yields the electrical current from lead i . Within Gollum α_i may represent spin and in the presence of superconductivity may represent hole (h) or particle (p) degrees of freedom. In the latter case the charge $Q_p = -e$ and $Q_h = +e$.

7.6 FUNCTIONALITIES DELIVERED BY GOLLUM2

1. Simulates multi-terminal devices
2. Reads either Tight-Binding or DFT Hamiltonians
3. Reads electronic structures computed with plane-wave codes via Wannier functions
4. Computes the full scattering matrix
5. Computes charge transport
 1. Number of open scattering channels
 2. Transmission and reflection coefficients
 3. Shot-noise
6. Computes heat transport
 1. Thermal conductance
 2. Thermopower (Seebeck coefficient)
 3. Peltier coefficient
7. Computes spin transport
 1. Collinear spins systems
 2. Non-collinear spins systems
 3. Systems with spin-orbit interaction
8. Computes zero-voltage as well as I-V curves
9. Handles vdW and LDA+U functionals
10. Scissors corrections scheme for strongly correlated systems
11. Computes band-structure of the leads and density of states (DOS) in the scattering region
12. Covers large samples enabling the analysis of the ballistic to diffusive regimes.
13. Phonons: Phonon transport and thermal conductance.

7.7 INPUT FILES FORMAT AND NOTATION.

Gollum can read the Hamiltonian of the extended scattering region (which includes a terminal principle layer as shown in FIG.4) and extract the PL to build the leads Green's function. It can also read stand-alone lead Hamiltonians to obtain the same information. Therefore Gollum requires as a minimum an 'input' file and an 'Extended_Molecule' file with an optional instruction to read 'Lead_i' files. The variables in the input file are designed to be clear and easy to interpret and apart from the Mode variable, which must appear first, the variables can be placed in arbitrary order and some variables are optional. The 'Extended_Molecule' file instructs

Gollum how to read the Hamiltonian, overlap matrix, orbitals, spin degrees of freedom and the number of transverse k points for the extended scattering region. All variables are mandatory.

8 THE FILE 'INPUT'.

Note the 'INPUT' file could now be created interactively as Gollum2 is initiated; nevertheless the format is shown as follows. The new functionalities are highlighted in bold font.

8.1 MANDATORY VARIABLE: MODE

As you have completed the examples to test your compilation you will note that the results appear in the folder 'mode-j' where $j=0\dots5$ (NOTE: mode-5 initiates the phonon calculations only). The concept of a mode in Gollum is employed to define the range of calculations to be carried out.

name: Mode

type: scalar j

NOTE: In all modes there are now restart flags, which enable the user to calculate the transmission coefficients once, store them and then use them to calculate further properties.

Mode 0 [$j=0$]

In this mode, Gollum computes the band structure, density of states (DOS) and number of open channels for a periodic system. A very important functionality which allows the user to determine the fundamental properties of the source of electrons when applied to the leads.

Hints: increase the number of energy points to eliminate spurious spikes in the DOS.

Mode 1 [$j=1$]

In this mode, Gollum computes the equilibrium transport properties of a junction.

Mode 2 [$j=2$]

In this mode, Gollum computes the zero-voltage, zero-energy spin-dependent Scattering matrix and Transmission coefficients.

Mode 3 [$j=3$]

In this mode, Gollum computes the zero-voltage thermo-electric properties of the junction. Those include the Peltier Coefficient, the Thermal conductance, the Thermopower, the Seebeck Coefficient and the Figure of Merit. Improvements enable the user to access the spin dependent thermo-electric properties over a temperature sweep or an energy sweep based upon the Fermi Energy.

Mode 4 [$j=4$]

In this mode, Gollum determines the finite-voltage transmission coefficients and current-voltage $I-V$ curves of multi-lead junctions. The user can implement now its own potential profile to describe the bias or gate voltage with or without scissors corrections.

Mode 5 [$j=5$]

In this mode Gollum determines the phononic thermal conductance.

8.2 A SUMMARY OF THE MODE FUNCTION

Mode	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
0	✓	✓	✓	✓	✓	✓	✓				✓			✓									
1	✓	✓	✓	✓	✓	✓			✓		✓	✓	✓										
2	✓	✓	✓	✓	✓				✓		✓	✓	✓										
3	✓	✓	✓	✓	✓	✓			✓		✓	✓	✓		✓	✓	✓		✓		✓	✓	
4	✓	✓	✓	✓	✓			✓	✓	✓	✓	✓	✓										
5	✓	✓	✓	✓					✓											✓			✓
MV(✓) = Mandatory variable: PMV(✓) = Partial mandatory variable: OV(✓)= Optional variable: scalar (s): matrix (m): string (st)																							

KEY	MV	PMV	OV	Definition
A: Mode	✓			Mode (s)
B: Leadp	✓			Lead PL and TPL properties (m)
C: atom	✓			EM atom properties (m)
D:HamiltonianProvider	✓			dft (Density Functional) or (Tight Binding) tbm (st)
E:Path_Leads	✓			Path to the leads file (st)
F: ERange		✓		Energy range and number of data points (m)
G: kRange		✓		k range and points mode-0 (m)
H: Bias		✓		Current-voltage range and number of data points (m)
I:Path_EM		✓		Path to extended molecule (st)
J G: Biasaccuracy			✓	Energy grid integration accuracy (s)
K: EF_shift			✓	Fermi energy shift overrides default shift to E _F (s)
L: scissor			✓	Scissor corrections (m)
M:VgateFactor or Vgate			✓	Multiplier for gate potential applied (s)
N: DOS		✓		DOS for mode-0 (m)
O:ThermoUseSaveTrm			✓	Use saved transmission data(0 or 1) (s)
P:ThermoSweep		✓		EF sweep = 0 Temperature sweep = 1 (s)
Q:ThermoEFRange		✓		Used if ThermoSweep = 0 (m)
R:ThermoEF		✓		Used if ThermoSweep = 1 (s)
S:ThermoTRange		✓		Used if ThermoSweep = 1 (m)
T:ThermoSpinMethod		✓		Spin method to calculate the thermo-electric properties
U:ThermoAccuracy			✓	Fineness of temperature grid.
V:ThermoTemp		✓		Temperature for ThermoSweep = 0 (s)
W:WRange		✓		Phonon frequency range (m)

8.3 MANDATORY VARIABLE: LEADP

This variable contains one row per branch (or lead) and sets the branch properties. The format is:

```
# name: leadp
# type: matrix
# rows: 2
# columns: 3
leadp(1) leadp(2) leadp(3)
```

For each row (branch), the columns are as follows:

leadp(1) Sets the number of PL in the branch.

leadp(2) Sets the terminating PL in the branch. Notice that leadp(2) must be equal to or smaller than leadp(1).

leadp(3) Sets the Hamiltonian used to compute the Lead Green's function (GF). Possible values are:

0. The lead GF is computed using the Hamiltonian and overlap matrices of the TPL, using the Hamiltonian of the EM, which is read from the file 'Extended_Molecule'.

± 1 . The Hamiltonian and overlap matrices of ideal leads are appended at the end of the TPL and become the new TPL. These are read from the file 'Lead_i', where i is the Lead number. The Lead GF is computed using this new TPL. The sign indicates whether the lead incoming channels are directed along the positive (-1) or negative (+1) z-axis.

As an example, for a junction containing two leads, lead 1 (left lead) should use leadp(3)=-1 and lead 2 (right lead) should use leadp(3)=1.

For DFT-based calculations, it is advisable to generate and use the files Lead1 and Lead2, setting leadp(3)= ± 1 . This is because even PLs far away from the electrode's surface feel the presence of the surface, which produces spurious gaps in the Open Channels.

8.4 MANDATORY VARIABLE: ATOM

This variable defines the properties of all the atoms in the Extended Molecule. There is therefore a row per atom. For example, for an EM region containing 26 atoms, the variable would look like

```
# name: atom
# type: matrix
# rows: 26
# columns: 6
atom(1) atom(2) atom(3) atom(4) atom(5) atom(6)
```

For each row, the columns are defined as follows:

atom(1): Atom number in the Extended Molecule.

atom(2): Sets the EM region where the atom is placed. Possible regions are "Molecule + Surface" (atom(2)=0) or one of the branches. E.g.: For Lead i, atom(2)=i.

atom(3): If atom(2) = i \neq 0, then this variable sets the PL in Lead i to which the atom belongs to. However, if the atom belongs to the "Surface + Molecule" region (atom(2)=0), then this variable is used to discern atoms belonging to the Molecule itself (atom(3)=0) or the surface (atom(3)=1), which is used to apply scissors corrections or include a Kondo U only in the molecule.

atom(4): If atom(4)=i, the variable instructs the program to shift the on-site matrix elements of the orbitals in the atom by the voltage of Lead i. If atom(4)=0, there is no shift.

atom(5) describes the user defined Bias potential (Default 0.0)

atom(6) describes the user defined Gate potential (Default 0.0)

Variable ATOM can have very many rows, so a simple MATLAB utility is provided to automate the generation of the file 'input'.

8.5 MANDATORY VARIABLE: HAMILTONIANPROVIDER

This variable defines the Hamiltonian kind: dft (Density Functional Theory) or tbm (Tight Binding model)

```
# name: HamiltonianProvider
# type: string
# rows: 1
tbm
```

8.6 PARTIAL MANDATORY VARIABLE VARIABLE: PATH_EM

Defines the path to your EM Hamiltonian file.

```
# name: Path_EM
# type: string
# rows: 1
../Extended_Molecule
```

8.7 MANDATORY VARIABLE: PATH_LEADS

Gives the path to your leads Hamiltonian files.

```
# name: Path_Leads
# type: string
# rows: 2
1 ../Lead_1
2 ../Lead_2
```

8.8 PARTIAL MANDATORY VARIABLE FOR MODES 0, 1 AND 3: ERANGE

This variable sets the energy range (E_{min} and E_{max}) and number of energy points nE used. The variable format is as follows:

```
# name: ERange
# type: matrix
# rows: 1
# columns: 3
ERange(1) ERange(2) ERange(3)
    ERange(1) Defines  $E_{min}$  measured in eV units.
    ERange(2) Defines  $E_{max}$  measured in eV units.
    ERange(3) Defines  $nE$ .
```

The variable is only read if Mode=0 , 1 or 3, because the program uses default values for modes 2 and 4. If Mode=2, E_{min} and E_{max} are set to 0 and nE to 1. If Mode=4, E_{min} and E_{max} are set to $\pm V_{max}/2$, and nE is set by the variable Bias_Accuracy.

8.9 PARTIAL MANDATORY VARIABLE FOR MODES 3 AND 5: THERMOTRANGE.

This variable controls the temperature range in the calculation of thermal properties. Its format is as follows:

```
# name: ThermoTRange
# type: matrix
# rows: 1
# columns: 3
TRange(1) TRange(2)TRange(3)
    TRange(1) defines the minimum temperature for the calculation  $T_{min}$ , measured in Kelvin degrees.
    TRange(2) defines the maximum temperature for the calculation  $T_{max}$ , measured in Kelvin degrees.
    TRange(3) defines the number of temperature points to be performed in the range ( $T_{min}$ ,  $T_{max}$ ).
```

8.10 PARTIAL MANDATORY VARIABLE FOR MODE 3: THERMOSWEEP.

Defines if the calculation of the thermal properties is carried out over an energy range or a temperature range.

```
# name: ThermoSweep
# type: scalar
0 (Energy) or 1(Temperature)
```

8.11 PARTIAL MANDATORY VARIABLE FOR MODES: THERMOEFRANGE.

Used if ThermoSweep = 0. Should be the same or smaller than the accompanying energy range (ERange)

```
# name: ThermoEFRange
# type: matrix
# rows: 1
# columns: 3
-8.00  8.00  200.00
```

8.12 PARTIAL MANDATORY VARIABLE FOR MODE 3: THERMOTEMP.

Defines the temperature of the calculation if ThermoSweep = 0 i.e.

```
# name: ThermoTemp
# type: scalar
300
```

8.13 PARTIAL MANDATORY VARIABLE FOR MODE-3: THERMOSPINMETHOD.

As the calculations are spin dependent there are two methods of calculating the Seebeck coefficient (S)

The default when ThermoSpinMethod = 0 is given by $S = \frac{L_i}{L_{\uparrow}L_{\downarrow}}$

The alternative where ThermoSpinMethod = 1 is given by $S^{\sigma} = \frac{L_1^{\sigma}}{L_0^{\sigma}}$

The first corresponds to the case where charge and spin are tied up while the second to the case of spin-charge separation.

```
# name: ThermoSpinMeth
# type: scalar
1
```

8.14 PARTIAL MANDATORY VARIABLE FOR MODE=4: BIAS

This variable controls the calculation of current-voltage curves and is only used in Mode=4. The variable contains one row per lead. In this example for a four lead junction the first row describes how the voltage in lead 1 is to be ramped up. The voltage for the remaining leads is fixed to a given value given by V2, V3, V4. The format of the variable is as follows:

```
# name: Bias
# type: matrix
# rows: 4
# columns: 3
Bias(1) Bias(2) Bias(3)
V2 0 0
V3 0 0
V4 0 0
```

Bias(1) defines the minimum voltage to be applied to the lead V_{min} , measured in volts.

Bias(2) defines the maximum voltage to be applied to the lead V_{max} , measured in volts.

Bias(3) defines the number of voltage points $nvolt$ to be used in the range (V_{min} , V_{max}).

NOTE: the user defined bias potential profile can be defined in column 5 of the atom variable.

8.15 PARTIAL MANDATORY VARIABLE FOR MODE 5: WRANGE

Phonon calculation where the frequency range is defined.

```
# name: WRange
# type: matrix
# rows: 1
# columns: 3
0.00    0.15    20001.00
```

8.16 OPTIONAL VARIABLE FOR ALL MODES: VARIABLE: EFSHIFT

Gollum uses the Fermi energy of the extended molecule to set the zero energy. This reference energy can be shifted up and down by setting EF_shift to positive or negative energy values (measured in eV).

```
# name: EF_shift
# type: scalar
+/- eV
```

8.17 OPTIONAL VARIABLE FOR MODE=4: VARIABLE: BIAS_ACCURACY

Bias_accuracy is used to set the energy grid in mode 4. Available values are 0, 1, 2 and 3, whereby the energy grid is set to 100, 250, 1000 and 3000 energy points, respectively. Consequently, the calculation is slowed down roughly by a factor of 3 or 10 respectively. **NOTE** that the 100 point calculation will not be accurate and should be used for pre-production runs.

```
# name: Bias_accuracy
# type: scalar
1
```

8.18 OPTIONAL VARIABLE FOR MODES 1,2,3,4: VARIABLE: SCISSORS

The atoms to which this correction is applied are set in variable: atom. If scissor corrections are applied then Gollum also prints the Molecule's Density of States (DOS) with and without scissor corrections, where the energy origin is chosen to be the Fermi energy of the EM region. This functionality is implemented only if the calculation is spin-unpolarized and the number of transverse k-points is one. The variable format is as follows:

```
# name: scissors
# type: matrix
# rows: 1
# columns: 5
scissors(1) scissors(2) scissors(3) scissors(4) scissors(5)
    scissors(1) Compute Scissor corrections (if 1) or not (otherwise).
    scissors(2) Number of electrons in the molecule.
    scissors(3) Energy value for the shift downwards of the HOMO.
    scissors(4) Energy value for the shift upwards of the LUMO.
    scissors(5) Defines the distance for the screening image charge, measured in Bohr.
```

NOTE: It is usually accompanied by a user defined gate potential in column 6 of the atom variable.

8.19 OPTIONAL VARIABLE FOR ALL MODES: USER DEFINED GATE POTENTIAL

The user can define the gate potential profile across the atoms in the system by using the sixth column in the atom variable:

```
# name: atom
```

```
# type: matrix
# rows: 5
# columns: 6
1      1      2      0      0.00  0.00
2      1      1      0      0.00  0.50
3      0      1      0      0.00  1.00
4      2      1      0      0.00  0.50
5      2      2      0      0.00  0.00
```

The above example is for a 6-atom chain. NOTE care must be taken to have a reasonable potential and ensure via the gate voltage factor, which effectively scales the potential, are set correctly for your system. Many of the Mode-4 example input files give good starting points.

8.20 OPTIONAL VARIABLE FOR ALL MODES: VGATEFACTOR

This variable sets the parameter for multiplying the gate potential profile applied to the EM in 7.6.13 above. This is useful to compare results by varying your potential magnitude. The variable format is as follows:

```
# name: VgateFactor
# type: scalar
Multiplier (in eV).
# name: Vgate
# type: scalar
Multiplier (in eV).
```

8.21 OPTIONAL VARIABLE FOR MODE-4: USER DEFINED BIAS POTENTIAL

The user can define the potential profile across the atoms in the Extended Molecule by using the fourth and fifth columns in the atom variable as follows:

```
# name: atom
# type: matrix
# rows: 3
# columns: 6
1      1      1      1      0.50  0
2      0      0      0      0.00  0
3      2      1      2      0.50  0
```

The above example is for a 3-atom chain. Column 4 is used to set the lead to which the voltage of the atom is pegged. Column 5 then is used to define the potential profile. NOTE care must be taken to have a reasonable potential and ensure via the gate voltage factor, which effectively scales the potential, are set correctly for your system. Many of the Mode-4 example input files give good starting points.

8.22 OPTIONAL VARIABLE FOR MODE-3: THERMOUSESAVETRM

ThermoUseSaveTrm selects the option to save and reuse the transmission data. This saves time when choosing the correct parameters to calculate the thermal properties.

```
# name: ThermoUseSaveTrm
# type: scalar
1
```

8.23 EXAMPLE INPUT FILE.

The following is an input file for a two-lead junction in Mode 3.

```
# name: Mode
```

```

# type: scalar
3
# name: ThermoSweep
# type: scalar
1
# name: ERange
# type: matrix
# rows: 1
# columns: 3
-8 8 6000.00
# name: ThermoTRange
# type: matrix
# rows: 1
# columns: 3
1 400 200.00
# name: ThermoUseSaveTrm
# type: scalar
1
# name: ThermoSpinMeth
# type: scalar
1
# name: HamiltonianProvider
# type: string
# rows: 1
tbm
# name: Path_EM
# type: string
# rows: 1
../Extended_Molecule
# name: Path_Leads
# type: string
# rows: 2
1 ../Lead_1
2 ../Lead_2
# name: leadp
# type: matrix
# rows: 2
# columns: 3
2 2 -1
2 2 1
# name: atom
# type: matrix
# rows: 5
# columns: 6
1 1 2 0.00 0.00 0
2 1 1 0.00 0.00 0
3 0 1 0.00 0.00 0
4 2 1 0.00 0.00 0
5 2 2 0.00 0.00 0

```


9 THE 'EXTENDED MOLECULE' FILE.

The 'Extended_Molecule' file enables Gollum to read the Hamiltonian and overlap matrices of the extended scattering region. It uses mandatory variables to define how Gollum reads the Fermi energy, the orbital information, the spin degrees of freedom and the number of transverse k points used in the simulation.

9.1 AN EXAMPLE 'EXTENDED_MOLECULE' FILE.

```
# name: nspin
# type: scalar
1
# name: FermiE
# type: scalar
0
# name: iorb
# type: matrix
# rows: 10
# columns: 5
1 6 0 0 1
1 6 1 0 1
2 6 0 0 1
.....
5 6 1 0 1
# name: kpoints EM
# type: matrix
# rows: 1
# columns: 3
0.0000000000E+00 0.0000000000E+00 1.0000000000E+00
# name: HSM
# type: matrix
# rows: 36
# columns: 7
1 1 1 1.0 0.0 -2.0 0.0
1 3 3 1.0 0.0 -2.0 0.0
1 5 5 1.0 0.0 -2.0 0.0
...
1 10 9 0.0 0.0 -0.3 0.0
```

9.2 VARIABLE: NSPIN

```
# name: nspin
# type: scalar
nspin
```

This variable defines whether the calculation is spin-unpolarized (nspin=1) or polarized (nspin=2).

9.3 VARIABLE: FERMIE

```
# name: FermiE
# type: scalar
0
```

This variable defines the value of the Fermi energy at the extended molecule (in eV), which is taken as the reference energy.

9.4 VARIABLE: IORB

```
# name: iorb
# type: matrix
# rows: 10
# columns: 4
iorb(1) iorb(2) iorb(3) iorb(4)
```

This variable contains information about the orbitals used in the simulation. The variable has as many rows as orbitals exist in the EM region.

iorb(1) Indicates the atom to which the orbital belongs.
iorb(2) Indicates the principal quantum number n .
iorb(3) Indicates the orbital quantum number l .
iorb(4) Indicates l_z , using real spherical harmonics.

Gollum only uses iorb(1). The rest are placed for information purposes.

9.5 VARIABLE: KPOINTS

```
# name: kpoints EM
# type: matrix
# rows: 1
# columns: 3
kpoints(1) kpoints(2) kpoints(3)
```

This variable sets the transverse kpoints used in the Extended Molecule. There are as many rows as kpoints. kpoints(1) and kpoints(2) give the values of k_x and k_y ; kpoints(3) gives the weight in the k summation.

9.6 VARIABLE: HSM

```
# name: HSM
# type: matrix
# rows: 36
# columns: 7
HSM(1) HSM(2) HSM(3) HSM(4) HSM(5) HSM(6) HSM(7) HSM(8) HSM(9)
```

This variables set the overlap $S(i, j, k)$ and $H(i, j, k, \uparrow\downarrow)$ for a given transverse k -point, given orbital pairs (i, j) and spin $(\uparrow\downarrow)$. The number of columns is 7 if the calculation is paramagnetic ($nspin=1$) or 9 if the calculation is spin polarized ($nspin=2$). There are as many rows as non-zero Hamiltonian or overlap matrix elements. If for given $(i, j, k, \uparrow\downarrow)$ all matrix elements are zero, then this row is skipped.

HSM(1) Defines the k -point number as defined in variable kpoints.

HSM(2) Defines the orbital number i as defined in variable iorb.

HSM(3) Defines the orbital number j as defined in variable iorb.

HSM(4) Provides $\text{Real}[S(i, j, k)]$

HSM(5) Provides $\text{Imag}[S(i, j, k)]$

HSM(6) Provides $\text{Real}[H(i, j, k)]$ or $\text{Real}[H(i, j, k, \uparrow)]$ ($nspin = 1$ or 2 , respectively).

HSM(7) Provides $\text{Imag}[H(i, j, k)]$ or $\text{Imag}[H(i, j, k, \uparrow)]$ ($nspin = 1$ or 2 , respectively).

HSM(8) Exists if $nspin=2$. It provides $\text{Real}[H(i, j, k, \downarrow)]$.

HSM(9) Exists if $nspin=2$. It provides $\text{Imag}[H(i, j, k, \downarrow)]$.

10 THE 'LEAD_I' INPUT FILE.

10.1 EXAMPLE 'LEAD_I' FILE.

There can be one of these files for each lead. The file will be read and used if the corresponding variable leadp(3) is set to ± 1 . The following is an example of a Lead file:

```
# name: nspin
# type: scalar
2
# name: FermiE
# type: scalar
0
# name: kpoints_Lead
# type: matrix
# rows: 1
# columns: 3
0.0000000000E+00 0.0000000000E+00 1.0000000000E+00
# name: HSL
# type: matrix
# rows: 1
# columns: 15
1 1 1 1.0 0.0 -1.0 0.0 1.0 0.0 0.0 0.0 -2.0 0.0 -2.0 0.0
```

10.2 VARIABLE: KPOINTS_LEAD

```
# name: kpoints Lead
# type: matrix
# rows: 1
# columns: 3
kpoints(1) kpoints(2) kpoints(3)
```

This variable sets the transverse kpoints used in the DFT simulation of the Lead. There are as many rows as kpoints. It is mandatory because the ordering of k-points in the DFT simulations of the Leads and of the Extended Molecule need not coincide.

kpoints(1) and kpoints(2) give the values of k_x and k_y ; kpoints(3) give the weight in the k summation.

10.3 VARIABLE: HSL

```
# name: HSL
# type: matrix
# rows: 4
# columns: 11
HSL(1) HSL(2) ... HSL(11) ... HSL(15)
```

This variables set the intra- and inter-PL overlap $S_{0,1}(i, j, k)$ and Hamiltonian matrix elements $H_{0,1}(i, j, k, \uparrow\downarrow)$ for a given transverse k -point, given orbital pairs (i, j) and spin ($\uparrow\downarrow$). The number of columns is 11 if the calculation is paramagnetic (nspin=1) or 15 if the calculation is spin-polarized (nspin=2). There are as many rows as non-zero Hamiltonian or overlap matrix elements. If for given $(i, j, k, \uparrow\downarrow)$ all matrix elements are zero, then this row is skipped. The ordering of the variable is:

For nspin = 1 (i, j, k) Real[S_0] Imag[S_0] Real[H_0] Imag[H_0] Real[S_1] Imag[S_1] Real[H_1] Imag[H_1]

For nspin = 2 (i, j, k) Real[S_0] Imag[S_0] Real[$H_{0,\uparrow}$] Imag[$H_{0,\uparrow}$] Real[$H_{0,\downarrow}$] Imag[$H_{0,\downarrow}$] Real[S_1] Imag[S_1] Real[$H_{1,\uparrow}$] Imag[$H_{1,\uparrow}$] Real[$H_{1,\downarrow}$] Imag[$H_{1,\downarrow}$]

11 OUTPUT FILE FORMAT AND NOTATION.

The output files are written for each lead i of a junction containing N leads. Some files are universal to all Modes but some are specific to each Mode defined in the input file. The output file for calculations with N leads will have $N+1$ columns, where the first column is the energy (E).

11.1 THE UNIVERSAL FILES.

11.1.1 THE UNIVERSAL OUTPUT AT A GLANCE: TABLE 10.1.1.

This table shows the universal output files as seen in the Results folder in the Examples.

	Key: spin refers to paramagnetic spin up(down) refers to spin-polarized $\sigma = 1$ or 2	Comment.
Mode-1-2-3-4	Open_channels_per_spin σ .gdat	The number of open channels per lead
Mode-1-2-3-4	Open_channels_up σ .gdat	The number of open channels per lead
Mode-1-2-3-4	Open_channels_down σ .gdat	The number of open channels per lead
Mode-1-2-3-4	Shot_noise_per_spin σ .gdat	Shot noise coefficients
Mode-1-2-3-4	Shot_noise_per_up σ .gdat	Shot noise coefficients
Mode-1-2-3-4	Shot_noise_per_down σ .gdat	Shot noise coefficients
Mode-1-2-3-4	T_per_spin σ .gdat	Transmission coefficients
Mode-1-2-3-4	Tup σ .gdat	Transmission coefficients
Mode-1-2-3-4	Tdown σ .gdat	Transmission coefficients
Mode-1-2-3-4	Bands σ .gdat	From the GF of lead i .
Mode-0	DOS_spin_up(down)_per_k1 E-k_spin_up(down)_per_k1 Open_channels_spin_up(down)_per_k1	DOS leads Bands leads G/G0 leads
Phonon Mode-5	Open_ph_channels σ .gdat Phonon_thermal_conductance σ .gdat Tph σ .gdat	Phonon open channels Phonon Thermal conductance Phonon transmission

NOTE-1. Scissor corrections shift the HOMO and LUMO levels upwards and downwards in energy. This has the effect of reducing the transmission and therefore the current across the junction around the Fermi energy.

NOTE-2. The results for paramagnetic or spin polarized calculations are determined by the input Hamiltonians which for example depend upon your flags in the SIESTA input file and the value of $nspin = 1$ for spin-unpolarized or $nspin = 2$ polarized in the Extended_Molecule input file.

11.1.2 TRANSMISSION (REFLECTION) COEFFICIENTS.

Mode-1-2-3-4	T_per_spin σ .gdat	Transmission coefficients
Mode-1-2-3-4	Tup σ .gdat	Transmission coefficients
Mode-1-2-3-4	Tdown σ .gdat	Transmission coefficients

Mode-1, Mode-2 and Mode-3 give zero bias calculations. For example the transmission coefficients are given by $T_{ij}(E, 0)$ $i(j) = 1 \dots N$ in the columns where $i \neq j$ and the reflection coefficients for $i = j$. Files are written to

give parametric spin calculations and up/down for spin-polarized calculations. The block of gdata is headed by a line stating the voltage and the first column is the energy (E) giving $N + 1$ columns in total.

Mode-4 gives finite bias results $T_{ij}(E, V)$ in the same format as the other Modes.

11.1.3 THE NUMBER OF OPEN CHANNELS.

Mode-1-2-3-4	Open_channels_per_spin σ .gdat	The number of open channels per lead
Mode-1-2-3-4	Open_channels_up σ .gdat	The number of open channels per lead
Mode-1-2-3-4	Open_channels_down σ .gdat	The number of open channels per lead

These files give the number of open channels as a function of energy in the lead. The number of open channels is a dimensionless quantity.

11.1.4 SHOT NOISE.

Mode-1-2-3-4	Shot_noise_per_spin σ .gdat	Shot noise coefficients
Mode-1-2-3-4	Shot_noise_per_up σ .gdat	Shot noise coefficients
Mode-1-2-3-4	Shot_noise_per_down σ .gdat	Shot noise coefficients

The file provides the Shot Noise coefficients $SN_{ij}(E, V = 0)$ using the same format as the transmission files. Shot noise coefficients are dimensionless.

11.2 ADDITIONAL MODE GENERATED FILES.

11.2.1 MODE GENERATED FILES AT A GLANCE: TABLE 10.2.1.

	Key: spin refers to paramagnetic spin up(down) refers to spin-polarized $i = 1..N$, where N = the number of leads.	Comment
mode-2	G_spinsum <i>i</i> .gdat	The spin-summed conductance G/G_0
mode-2	G_up(down) <i>i</i> .gdat	The spin conductance G/G_0
mode-2	G_tot <i>i</i> .gdat	The spin-summed conductance G/G_0
mode-2	S_matrix.gdat	The S matrix
mode-3	G_tot <i>i</i> .gdat	The spin-summed thermal conductance $G(T)$
mode-3	G_up(down) <i>i</i> .gdat	The spin thermal conductance $G(T)$
mode-3	Thermopower_tot <i>i</i> .gdat Thermopower_up(down) <i>i</i> .gdat	Spin-summed thermopower $S^e(T)$
mode-3	Peltier_tot <i>i</i> .gdat Peltier_up(down) <i>i</i> .gdat	Peltier coefficients $\Pi(T)$
mode-3	Thermal_conductance_tot <i>i</i> .gdat	Spin-summed thermal conductance $\kappa(T)$
mode-3	ZT <i>i</i> .gdat	ZT
mode-4	Charge_Current <i>i</i> .gdat	Current-voltage

11.2.2 MODE-2 GENERATED FILES.

11.2.2.1 CONDUCTANCE.

mode-2	G_spinsum <i>i</i> .gdat	The spin-summed conductance G/G_0
mode-2	G_down <i>i</i> .gdat	The spin-summed conductance G/G_0
mode-2	G_up <i>i</i> .gdat	The spin-summed conductance G/G_0

For each lead *i* the file G_spinsum*i*.gdat containing *N* columns where *N* is the number of leads is written. Each column *j* provides the spin-summed zero-voltage conductance G/G_0 . For spin-polarized calculations additional files G_up*i*.gdat and G_down*i*.gdat containing the spin-resolved conductances G/G_0 are also given.

11.2.2.2 S MATRIX.

Mode-2	S_matrix.gdat	The S matrix
--------	---------------	--------------

The S_matrix.gdat file contains the scattering matrix coefficients.

11.2.3 MODE-3 GENERATED FILES.

11.2.3.1 CONDUCTANCE.

Mode-3	G_spinsum <i>i</i> .gdat	The spin-summed thermal conductance $G(T)$
Mode-3	G_down <i>i</i> .gdat	The spin-summed thermal conductance $G(T)$
Mode-3	G_up <i>i</i> .gdat	The spin-summed thermal conductance $G(T)$

For each lead *i*, Gollum writes a file called G_spinsum*i*.gdat which contains the spin-summed, temperature dependent conductance ($G(T) = G/G_0$) between lead *i* and lead *j* ($j \neq i$). There is a row for each temperature (K) where the first column in each row provides the temperature and the remaining columns give $G(T)$. The spin resolved temperature-dependent conductances are given in G_down*i*.gdat and G_up*i*.gdat.

11.2.3.2 THERMOPOWER.

Mode-3	Thermopower_tot <i>i</i> .gdat Thermopower_up(down) <i>i</i> .gdat	Spin-summed thermopower
--------	---	-------------------------

In a similar layout to the conductance files for each lead *i* Gollum writes a file called Thermopower*i*.gdat which contains the spin-summed, temperature dependent thermopower $Q(T)$ between lead *i* and lead *j* ($j \neq i$).

11.2.3.3 PELTIER COEFFICIENTS.

Mode-3	Peltier_tot <i>i</i> .gdat Peltier_up(down) <i>i</i> .gdat	Spin-summed Peltier coefficients
--------	---	----------------------------------

In a similar layout to the conductance files for each lead *i* Gollum writes a file called Peltier*i*.gdat which contains the spin-summed, temperature dependent Peltier coefficient between lead *i* and lead *j* ($j \neq i$).

11.2.3.4 THERMAL CONDUCTANCE.

Mode-3	Thermal_conductance_tot <i>i</i> .gdat	Spin-summed thermal conductance
--------	--	---------------------------------

In a similar layout to the conductance files for each lead i Gollum writes a file called Thermal_conductance*i*.gdat which contains the spin-summed, temperature dependent thermal conductance between lead i and lead j ($j \neq i$).

11.2.4 MODE-4 GENERATED FILES.

11.2.4.1 CURRENT-VOLTAGE.

Mode-4	Charge_Current <i>i</i> .gdat	Current-voltage
--------	-------------------------------	-----------------

For each lead i , Gollum writes a file called Charge_Current*i*.gdat which contains the charge current as a function of voltage $I(V)$. The file gives the total charge currents flowing between lead i and lead j ($j \neq i$) as a function of voltage, measured in Amps.

11.2.4.2 SPIN CURRENT.

Mode-4	Spin_Current <i>i</i> .gdat	Spin-current
--------	-----------------------------	--------------

If the calculation is spin-polarized, a file called Spin Current*i*.gdat is also written for each lead i , containing the spin current, defined as $I_{\uparrow} - I_{\downarrow}$, as a function of voltage.

11.2.5 MODE-5 GENERATED FILES.

Phonon Mode-5	Open_ph_channels <i>σ</i> .gdat	Phonon open channels
	Phonon_thermal_conductance <i>σ</i> .gdat	Phonon Thermal conductance
	Tph <i>σ</i> .gdat	Phonon transmission

12 REFERENCES

1. "Gollum: a next-generation simulation tool for electron, thermal and spin transport"
J. Ferrer, C. J. Lambert, V. M. García-Suárez, D. Zs. Manrique, D. Visontai, L. Oroszlany, R. Rodríguez-Ferradás, I. Grace, S. W. D. Bailey, K. Guillemot, Hatef Sadeghi, L. A. Algharagholy, *New J. Phys.* **16** 093029(2014)
2. J. M. Soler, E. Artacho, J.D. Gale, A. García, J. Junquera, P. Ordejón, and D. Sánchez-Portal, *J. Phys.: Condens. Matter* **14**, 2745 (2002)
3. J. P. Lewis, P. Jelnek, J. Ortega, A. A. Demkov, D. G. Trabada, B. Haycock, H. Wang, G. Adams, J. K. Tomfohr, E. Abad, H. Wang and D. A. Drabold, *Phys. Stat. Solidi B* **248**, 1989 (2011).

C

Cite

Gollum paper 0, 3

conductance 42, 43

Current

(amps) Gollum units 3, 29, 42, 45

D

DFT-based

Example and tutorial 0, 4, 10, 11, 15, 17, 30

Direct SIESTA method 0, 17

E

EM

extended scattering region 3, 19, 20, 23, 24, 29, 30, 31,
33, 35, 36, 39, 41

Energy

(eV) Gollum units 3, 29, 33

Examples

Example and tutorial 0, 4, 8, 9, 10, 11, 12, 15, 17

Extended molecule

input, method 1, 18

F

Fermi energy

E_F (eV) 3, 18, 23, 24, 29, 33, 35, 36, 39

fortran 5, 18

G

Generation of Gollum files 0, 18

Gold lead 0, 11, 15, 16, 18

Gollum

paper 1, 0, 3, 4, 5, 6, 7, 8, 10, 14, 16, 17, 18, 19, 20, 21,
22, 23, 26, 27, 28, 33, 35, 36, 43, 44, 45

H

Hamiltonian

generation, theory 0, 1, 5, 17, 18, 20, 21, 22, 23, 27, 30,
35, 37, 38

I

Indirect SIESTA method 0, 17

Input file

variable

mandatory

partial

optional 1, 27

installation 0, 7

Interface 0, 17, 18

L

Lead-1 14, 16

M

mac 9

MCR

for Matlab 0, 7, 8, 9

Mode-1 28, 39, 40, 41

Mode-2 2, 28, 40, 41, 42, 43

Mode-3 2, 28, 29, 30, 40, 42, 43, 44

Mode-4 2, 28, 29, 40, 42, 45

Modes

input

output 1, 2, 27, 28, 29, 30, 31, 32, 33, 34, 38, 39, 40,
41, 42, 43, 44, 45

O

Output

universal 2, 38

P

PL

principle layer 20, 21, 27, 29, 30, 31, 38

R

running..... 0, 7, 17, 23

S

Shot noise 2, 39, 40, 41

SIESTA

DFT0, 4, 5, 16, 17, 39

Spin current 45

Step by step

tight binding and DFT 0, 12, 15

Summary

examples..... 0, 10

T

Temperature

(K) Gollum units 29, 34

Theoretical 1, 19

Tight-binding-based

Example and tutorial 0, 4, 10, 12

TPL

terminating principle layer 20, 23, 29, 30