TranSIESTA Update
&
TBTrans Update

Nick Papior Andersen

November 19, 2012

This document is intended for notification about the changes in the code TranSIESTA. The author of this document has worked on optimising and restructuring the TranSIESTA code to be more usable and easier to maintain. To get an idea of how much code that has been recoded a `diff` with `grep -e "[^-+]"` yields a total of > 15.000 lines of code. Thus a lot has changed and we wish to test this version for future full implementation.

The author would like to thank Mads Brandbyge and Alberto Garcia for fruitful discussions and guidance.

The document will provide an overview of the things that has been implemented as well as emphasising what should be noticed and/or changed in regards of simulations, in general, no parameters should be changed and all current FDF-inputs are valid (with a few default changes on file names, and file overwrites). Only the internal structure has changed as well as the addition of new features.

Furthermore, there will be a description of a new TBTrans utility, which provides a significant speed-up.

There will be three types of messages in this document:

NOTICE Something to be noticed.

BEWARE Something radically different than the previous implementations, check the output files of the information specified here.

REPORT BACK Please report back your findings.

1 Bug fix release – rev. 23

Several bugs in the compilation of TranSIESTA/TBTrans have been corrected:

- Pointer mapping of sized pointer is a FORTRAN 2003 standard (m_ts_contour.F90)
- Removed intent of pointer passings (m_hs_matrix.f90)
- Added dummy routine for non-MPI compilation (m_glob_sparse.F90)
- Renamed `TS.ReUseGF` to `TS.TBT.ReUseGF` (TBtrans_rep)

- Bugfix in units of AtomPDOS and COOP (TBtrans_rep)
- Bugfix for parallel runs, regarding FDF files (TBtrans_rep)
- Several bugfixes for option handling (TBtrans_rep)
- Added spin factor for AtomPDOS/COOP calculations (TBtrans_rep)

  NOTICE Please test the DOS calculation to see if has been corrected in all cases

- Added more documentation in the manual

2 TranSIESTA

2.1 General changes

TranSIESTA now checks for incorrect unit cells. That is if the unit cell vectors 1 or 2 extends into the 3. unit vector, i.e. the transport direction.

NOTICE If you have unit vectors which are extending into a *forbidden* region and TranSIESTA does not stop, please notify the developers

TranSIESTA will now stop if the electrode TSHS files does not exist upon option reading. This ensures that the execution will stop immediately, as opposed to the former versions which first stopped in the TranSIESTA routine.

BEWARE If TranSIESTA does not stop and the electrode files does not exist, please notify the developers

TranSIESTA will calculate the GF files at the start of the simulation. This helps to check for settings in the electrode before spending too much time in the calculation. Furthermore, the GF files naming scheme has changed to conform with the <SystemLabel>. That is the GF file names have changed to:

`Left.GF` $\to$ <SystemLabel>.TSGFL,

`Right.GF` $\to$ <SystemLabel>.TSGFR.

However, the names can be changed at will using the regular FDF-keys (see the manual).

The GF files are now defaulted to **not** be overwritten in case they exist. This also means that in case they do not match, TranSIESTA will stop instead of overwriting the GF file. This enables the user to maintain several GF files and interchange them at will. Furthermore you will not loose any data which you could possibly reuse. Several checks are made to ensure the correctness of the GF file:

- Fermi level shift in electrode
- Number of energy points
- Number of atoms/orbitals

- Atomic coordinates of the electrode
- Repetition used
- Number of used **k**-points
- Number of spin
- Energy points on the contour line
- Weights of energy points

It should be near impossible for TranSIESTA to use a non-conforming GF file.

NOTICE The former versions of TranSIESTA did not check this elaborately

BEWARE If a GF file is used where it should not, please notify the developers

2.1.1 General speed-improvements

Quite a few algorithms have been optimised to increase the throughput. However also additional code has been added, so this could decrease the performance.

It is expected to show a great performance increase in the electrode creation due to the parellelisation.

REPORT BACK Be kind to return any speed changes recorded to the developers, notice that the number of SCF may differ between TranSIESTA versions and return speed per SCF, for best clarity. Refer to the TS_calc which is the "pure" SCF calculation time.

2.2 Repetition of the electrodes

NOTICE This is a new feature, please explore the different mechanisms and speed-improvements this facilitates

We use Bloch's theorem in acquiring the surface Green's function for the electrodes. An example of what it does can be seen in Fig. 1. If one does not have an electrode which has periodicities, the repetition scheme can not be used.

Using the repetition scheme can be done by using this transformation of existing electrodes:

1. Reduce the unit cell to the minimal unit cell in the x and y directions
2. Increase the **k**-point sampling the same factor as the reduction (in each direction)
3. Recalculate electrodes
4. Insert flags for TranSIESTA to recognise the repetition
5. Ensure that the electrodes are sorted according to the repetition (see Fig. 1)

2.2.1 Repetition algorithm

The expansion can be viewed as the following:

```
ia = 1 ! Full system atom count
do iaE = 1 , NUsedAtoms ! Electrode atom count
  do j = 0 , NA2
    do i = 0 , NA1 ! Here 'ucellE' is the electrode unit cell
      xa(1:2,ia) = xaE(1:2,iaE)+ucellE(1:2,1)*i+ucellE(1:2,2)*j
      xa(3   ,ia) = xaE(   3,iaE)
       a = a + 1
    end do
  end do
end do
```

which means that the repetition is along the first unit cell vector and then the second unit cell vector, see Fig. 1. TranSIESTA will automatically quit if the coordinates or the number of orbitals representing the atoms are not consistent.

BEWARE If the code does not stop and the outputted comparison of the electrode and scattering region electrode positions does not match, please notify the developers.

BEWARE If the code does not stop and the **k**-point sampling is wrong, please notify the developers.

2.2.2 Flags for the repetition

NOTICE These are *new* flags for using the repetition

The repetition can be enabled by using these flags:

```
TS.ReplicateA1Left <integer>
TS.ReplicateA2Left <integer>
TS.ReplicateA1Right <integer>
TS.ReplicateA2Right <integer>
```

They each represent the repetition of the electrode in the $\hat{\mathbf{u}}_i$ direction, $i \in \{1,2\}$.

2.2.3 Effect of repetition

The repetitions main advantage is the speed-up of generating the GF files for calculating the scattering region density matrix. I.e. the speed-up is entirely connected to the calculation of the surface-Green's function using the recursive methods. The Green's function files will also be a lot smaller. Roughly a factor of $R_{A1} \times R_{A2}$ will be saved in the file size. This will enable calculations on larger systems where the number of energy points is far greater.

However, the reduction of the electrode size also affects the self-consistent run. The smaller surface-Green's function must be repeated out which takes some computational time. For all tests performed by the developer, the computational gain in the GF generation outweighs the cost of expanding at each SCF cycle.

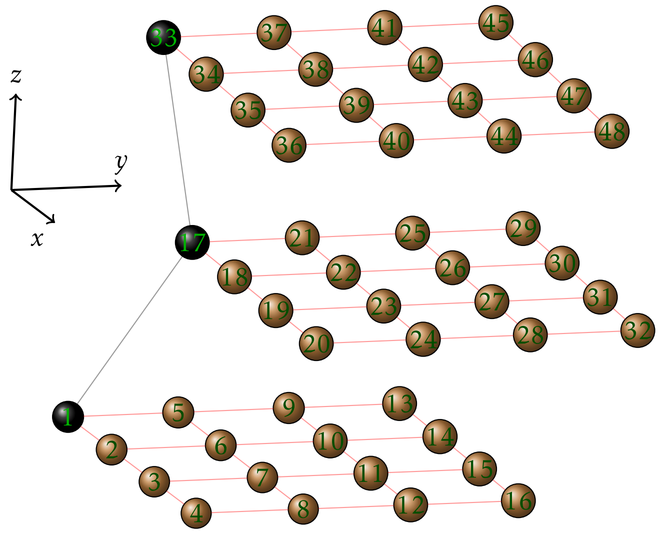

Figure 1 | Copper electrode of $4 \times 4 \times 3$ atoms. Employing the method of unit repetition the electrode can be described by only using the 3 atoms marked black. Through Bloch's theorem the 3 atoms can be expanded to the full system. They are numbered as to how they should be sorted in the `TranSIESTA` structure.

REPORT BACK If you ever encounter a case where the repeated cell is slower than the full GF, please notify the developers.

2.3 Parallelisation of the GF file generation

The generation of the GF files has been fully parallelised across the energy points. This means a speed up roughly equal to the number of processors. You should notice this speed up immediately. Furthermore, this will better allow for `TranSIESTA` to run on super-computers which require a certain load percentage. As in the previous `TranSIESTA` versions, try to use a number of energy points which is divisible by the number of processors used. This will not cost any computational time, only the GF-file size will increase.

In the option reading `TranSIESTA` will output a notice section if the number of energy points is not divisible by the number of processors. Also in cases of voltage applied the load will not be symmetric about electrode equilibrium and non-equilibrium contour integration points. Thus it is always advisable to require the following[1]:

$$(\texttt{NPoles} + \texttt{NCircle} + \texttt{NLine})\%\texttt{Nodes} = 0 \tag{1}$$

$$\texttt{NumPoints}\%\texttt{Nodes} = 0 \quad \text{, in case of applied voltage.} \tag{2}$$

REPORT BACK If you experience bad scalability, please inform the developers.

[1] % is the modulo operator.

2.4 Informational output

The code has been updated to supply a lot more information during the SCF cycle as well as in the initialisation of TranSIESTA.

The first thing you will notice is that after reading the options TranSIESTA will output how it recognises the different regions (buffer-layers, electrodes and device). It will show a new block in the following format:

```
transiesta: Atomic coordinates and regions (Ang):
//////////////////////////////////////////////
/      0.0000000      0.0000000      0.0000000  /     Left buffer
//////////////////////////////////////////////
##############################################
#      1.4615550      0.8438292      2.3867093  #     Left electrode
#      1.4615550     -0.8438292      4.7734186  #
##############################################
       0.0000000      0.0000000      7.1601280
       1.4615550      0.8438292      9.5468373        Device
       1.4615550      0.8438292     19.0936746
##############################################
#      1.4615550     -0.8438292     21.4803839  #
#      0.0000000      0.0000000     23.8670932  #     Right electrode
#      1.4615550      0.8438292     26.2538025  #
##############################################
//////////////////////////////////////////////
/      1.4615550     -0.8438292     28.6405119  /     Right buffer
//////////////////////////////////////////////
```

This makes debugging electrode positions easier.

BEWARE If you encounter atomic coordinates being handled incorrectly, please inform the developers.

After the options TranSIESTA will generate the GF files. Please notice here that the following has changed with respect to the contour points:

- energy values are now shown in eV and **not** in Ry,
- the equilibrium weights where shown with a minus sign as opposed to the internal use. Thus the equilibrium weights have changed their sign with respect to the previous versions (only for informational purposes, the internal implementation has not changed),
- the energy points have been assigned a “type” as opposed to a header, this is merely for design purposes,
- the parallel distribution is not shown as the parallelisation has been simplified. However, the distribution will be a consecutive distribution (for those interested).

When creating the GF files the output will look something like this:

```
Creating Green's function file for: left
 Left unit cell (Ang):
  2.9231  0.0000  0.0000
  1.4616  2.5315  0.0000
  0.0000  0.0000  7.1601
 Structure of the Left electrode  | System electrode:
     X (Ang)   Y (Ang)   Z (Ang)  |   X (Ang)   Y (Ang)   Z (Ang)
     0.00000   0.00000   0.00000  |   0.00000   0.00000   0.00000
     1.46156   0.84383   2.38671  |   1.46156   0.84383   2.38671
     1.46156  -0.84383   4.77342  |   1.46156  -0.84383   4.77342
 WARNING: Connections across 2 unit cells or more in the transport direction.
 WARNING: This is inadvisable.
 WARNING: Please increase the electrode size in the transport direction.
 WARNING: Will proceed without further notice.
 Electrodes with transport k-points  (Bohr**-1) and weights:
   1    -0.42655E+00   0.24627E+00   0.31250E-01
   2     0.56873E+00   0.32836E+00   0.15625E-01
 Atoms available    / used atoms   :      3 /      3
 Orbitals available / used orbitals:     27 /     27
 q-points for expanding electrode (Bohr**-1):
   1     0.00000E+00   0.00000E+00   0.10000E+01
 Fermi level shift in electrode :        0.50000  eV
Done creating 'Left.GF'.
```

It will list the following information:

Unit cell The unit cell found in the electrode TSHS file

Electrode comparison List the structure of the electrode compared to the system configured electrode (including the expansion of any repeated electrode). If the coordinates are not the same TranSIESTA will stop

List electrode k-points In case of repetition the **k**-points of the electrode are not the same as those of the system, if they do not conform to the system TranSIESTA will stop

Size information Number of atoms/orbitals used for generating the GF file (these are *not* repeated out)

Repetition q-points In case of repetition, $R_{A1} \times R_{A2}$ q-points will be used

Fermi level shift Is determined from the applied voltage

As seen above a warning message will occur in cases of atoms having connections across 3+ unit cells. This has implications on the generated surface-Green's function. The calculation of the surface-Green's function using the recursive method requires that there is only connections in the principal layer. However, if the connections are very weak they will not be as important and convergence can still be reached.

This warning is directly linked to the used supercell for the electrode calculation.

BEWARE If the warning message occurs, you should try and increase your electrode size and see if it changes the SCF cycles and final result

BEWARE If the electrode supercell is more than 2 in the $\mathbf{a}_3$ direction and the warning does not show up, please notify the developers.

After creating the GF-files (or if they already exist) an information regarding the used atoms and orbitals is again printed out:

```
Left : GF atoms     / Expanded atoms    :      3 /      3
Left : GF orbitals / Expanded orbitals :     27 /     27
Right: GF atoms     / Expanded atoms    :      3 /      3
Right: GF orbitals / Expanded orbitals :     27 /     27
```

the expanded key-word refer to the repeated out cell, and is thus equal to $R_{A1} \times R_{A2}$ times the atoms/orbitals in the GF.

2.5 Charge overview

The new code also provides more information about the charge distributions. Upon entry to the TranSIESTA routine an initial SIESTA charge distribution block will be shown (unless it is a TranSIESTA continuation run, in which case it is the charge distribution of the previous iteration).

Initial charge distribution is listed as such:

```
Efermi from SIESTA                     :    -2.79988
Total charge                    [Qt0]  :   110.00000
Charge in update region         [Qc]   :    44.31382
Charge outside update region    [Qcn]  :    65.68618
Left electrode                  [L]    :    32.69903
Left electrode/device           [L-C]  :     0.30826
Device                          [C]    :    43.69730
Device/right electrode          [C-R]  :     0.30826
Right electrode                 [R]    :    32.69902
Other                           [O]    :     0.28814
```

which shows, total charge, and the charge contained in the region of update, etc. If buffer regions exist, they will also be shown. In this case other will refer to the charges between the left and right electrode (in the SIESTA solution method).

BEWARE If the charges seem way off, please notify the developers

During the SCF cycle of TranSIESTA a new block will show like this:

```
ts-charge:         O        L      L-C        C      C-R        R       Qt
ts-charge:     0.288   32.699    0.374   43.718    0.177   32.699  109.955
```

which is directly linked to the notation used in the initial charge distribution overview. From this one can see that charges are moved from the [C-R] region to the [L-C] region. Qt is the total charge currently in the full system (this should be as close to Qt0 as possible. Again if buffer regions exist, they will also be shown.

2.6 Charge Correction

NOTICE This is a new feature, see the manual at `TS.ChargeCorrection`

NOTICE The charge fluctuations can be quite large, for very simple systems and/or irregularities in the region charges. If so, please try the charge correction scheme.

If the charge fluctuations are excessive, a charge correction code has been implemented. It enables the rescaling of the charges in certain regions to maintain a somewhat constant charge in the unit cell.

REPORT BACK If using the charge correction algorithm please notify the developers upon the results and/or the effect it has had

3 TBTrans

NOTICE The new TBTrans utility is found in `Util/TBTrans_rep`

In order to utilise the repetition and the new parallel algorithm across the energy points, a new TBTrans has been created. It is much more integrated into the TranSIESTA code and utilises almost all routines from TranSIESTA instead of having the same.

It is written entirely in Fortran 90.

The transmission calculation is done with a better parallelisation scheme than any of the previous TBTrans utilities as it calculates the **k**-point Hamiltonian in advance in parallel runs (which pays of if the communication of the Hamiltonian is faster than the creation of the Hamiltonian from the sparse matrix).

It is parallel across the **k**-points (somewhat) and the energy points (fully).

As the transmission can with benefit be calculated with a finer **k**-grid a new block for TBTrans is introduced. It is called: `%block tbt_kgrid_Monkhorst_Pack` which is equivalent to the regular **k**-grid input. However, it is only read by TBTrans. If it does not exist it will read in `%block kgrid_Monkhorst_Pack`.

NOTICE New feature: `%block tbt_kgrid_Monkhorst_Pack`

3.1 Informational output

In general there is not much left over from the older TBTrans versions. Much of the output will be similar to that of TranSIESTA.

In TBTrans a regional output of the system is also created. Within this it looks something like:

```
Atomic coordinates and regions (Ang):
###############################################
#     0.0000000     0.0000000     0.0000000  #
#     1.4615550     0.8438292     2.3867093  #     Left electrode
#     1.4615550    -0.8438292     4.7734186  #
###############################################
```

```
      0.0000000     0.0000000     7.1601280
      1.4615550     0.8438292     9.5468373   *    Device
      1.4615550     0.8438292    19.0936746   *
      1.4615550    -0.8438292    21.4803839   *
##############################################
#     0.0000000     0.0000000    23.8670932  #
#     1.4615550     0.8438292    26.2538025  #    Right electrode
#     1.4615550    -0.8438292    28.6405119  #
##############################################
```

It is the same as shown in TranSIESTA, however, the * in the device region denotes the PDOS atoms, see TS.TBT.PDOSFrom and TS.TBT.PDOSTo in the manual.

BEWARE If this shows an incorrect regional partition, please notify the developers

3.2 Atomic PDOS

NOTICE Use key TS.TBT.AtomPDOS

The new TBTrans has reenabled the calculation of the projected density of states on the selected atoms (PDOSFrom–PDOSTo). This is much like the Mulliken populations.

BEWARE Has not been tested, needs vigorous testing to ensure the correctness, please notify the developers on any kind of knowledge about this.

NOTICE This does not exist in the previous version of TBTrans

REPORT BACK Please inform the developers on any successful usage

3.3 COOP

NOTICE Use key TS.TBT.COOP

The new TBTrans has reenabled the calculation of the Crystal-Orbital OverlaP. It does not need any post-processing.

BEWARE Has not been tested, needs vigorous testing to ensure the correctness, please notify the developers on any kind of knowledge about this.

NOTICE This does not exist in the previous version of TBTrans

REPORT BACK Please inform the developers on any successful usage

3.4 Report speed improvements

As this version of TBTrans is not limited to parellelisation across **k**-points it should perform much better, as well as a lot of clearing of unnecessary code.

REPORT BACK Please report to the developers about the speed improvements of using this executable.