Recent development of the DFT+NEGF code `TranSIESTA`, performance improvements and N-terminal

Nick Papior Andersen, Mads Brandbyge

The Technical University of Denmark, Ørsteds Plads, 2800 Kgs. Lyngby,
Center for Nanostructured Graphene (CNG), DTU Nanotech
Denmark

nickpapior@gmail.com

DFT+NEGF methods are increasingly important tools for the interpretation and investigation of physical phenomena in experiments. In this work we present a re-implementation of the `TranSIESTA`-code [1] based on `SIESTA` [2].

The NEGF method relies on inverting the left hand side to obtain the Green's function (here shown with orbital indices μ, ν)

$$\left.\frac{1}{z\mathbf{S} - \mathbf{H} - \sum_i \mathbf{\Sigma}_i}\right|_{\mu\nu} = \mathbf{G}_{\mu\nu}, \tag{1}$$

where $\mathbf{H}$, $\mathbf{S}$ and $\mathbf{\Sigma}_i$ are respectively the Hamiltonian, overlap and self-energy matrices. The density matrix is directly related to the Green's function in equilibrium through $\mathbf{D}_{\mu\nu} \propto \mathbf{G}_{\mu\nu}$.

The Hamiltonian will inherently contain *many* zero elements due to the linear combination of atomic orbital (LCAO) approach in `SIESTA`. By taking advantage of this we can improve the performance and reduce memory requirements of `TranSIESTA`. An entire re-write of the code have surmounted to drastic performance improvements and memory reductions. We have implemented three variants of solution methods for the code, 1) a block tri-diagonal inversion algorithm which scales linearly with constant block sizes [3, 4] (LAPACK/BLAS), 2) a sparse inversion algorithm based on the MUltifrontal Massively Parallel sparse direct Solver (MUMPS, [5, 6]) which selectively calculates inverse elements of a matrix, and 3) the direct method using LAPACK/BLAS routines [7].

The block tri-diagonal and MUMPS routines can thus drastically decrease memory requirement of the code by removing unneeded elements from the Hamiltonian and overlap on the left hand side of Equation (1). The density matrix is equivalently only defined in the LCAO space which allows to only calculate a subset of matrix elements of the Green's function. This further reduces the memory requirement *and* the computational complexity.

We here present data for our block tri-diagonal routine with an example of pristine graphene using a 2 hexagon wide electrode and device. The basic unit cell is made up of 24 atoms and we increased the length of the system by multiplying this unit cell going up to ≈ 350 atoms. Our calculations are performed on a 12 core XeonE5-2620@2GHz with 32GB of ram (≈ 2.67 GB per core) and with the exact same settings for v3.2 and our work. The timings are seen in Fig. 1 together with the speedup of both the equilibrium calculation and the non-equilibrium calculations. At around 300 atoms v3.2 exhausts the memory in non-equilibrium which does not allow us to extend the analysis of comparison. However this work allows extending the system to 2160 atoms and still a single non-equilibrium SCF iteration step takes only 2351 s. Note that this is nearly a fourth of the time spent in v3.2 for 300 atoms!

Our implementation is generalised to handle N-terminal devices in `TranSIESTA` which will provide calculations of complex systems such as those schematised in Fig. 2. Furthermore the calculation of slabs ($N = 1$) within the method of NEGF will provide reduction in system sizes due to the removal of the mirror plane by imposing strict bulk, thus removing spurious non-bulk effects associated with the surface layers.

Center for Nanostructured Graphene

References

[1] M. Brandbyge *et al.* "Density-functional method for nonequilibrium electron transport". In: *Physical Review B* 65.16 (Mar. 2002), pp. 1–17.

[2] J. M. Soler *et al.* "The SIESTA method for ab initio order-N materials simulation". In: *Journal of Physics: Condensed Matter* 14.11 (Mar. 2002), pp. 2745–2779.

[3] E. M. Godfrin. "A method to compute the inverse of an n-block tridiagonal quasi-Hermitian matrix". In: *Journal of Physics: Condensed Matter* 3.40 (Oct. 1991), pp. 7843–7848.

[4] M. G. Reuter and J. C. Hill. "An efficient, block-by-block algorithm for inverting a block tridiagonal, nearly block Toeplitz matrix". In: *Computational Science & Discovery* 5.1 (July 2012), p. 014009.

[5] P. R. Amestoy *et al.* "A Fully Asynchronous Multifrontal Solver Using Distributed Dynamic Scheduling". In: *SIAM Journal on Matrix Analysis and Applications* 23.1 (Jan. 2001), pp. 15–41.

[6] P. R. Amestoy *et al.* "Hybrid scheduling for the parallel solution of linear systems". In: *Parallel Computing* 32.2 (Feb. 2006), pp. 136–156.

[7] E. Anderson *et al. LAPACK Users' Guide*. Third. Society for Industrial and Applied Mathematics, 1999.

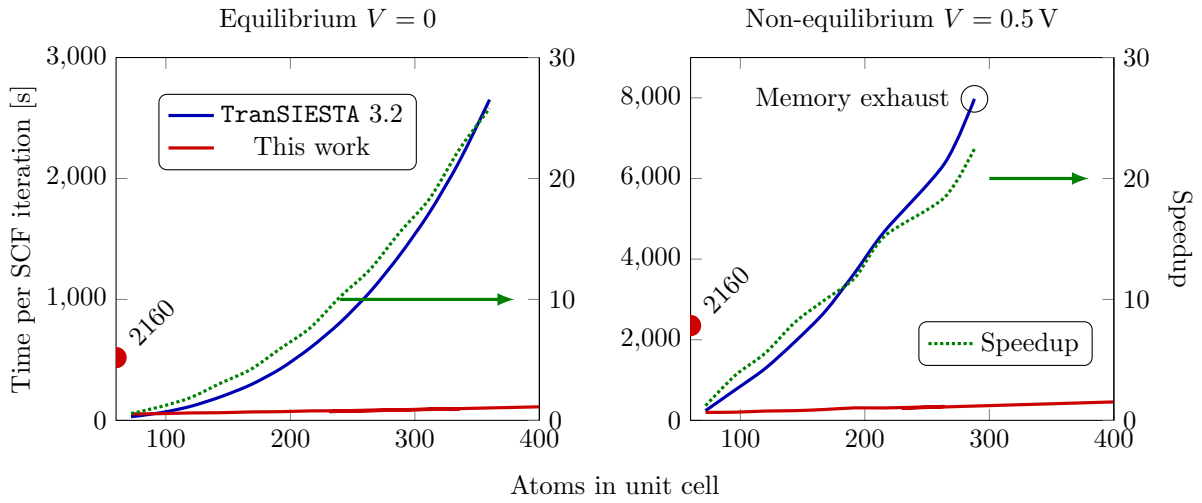

Figure 1: A direct comparison of a pristine graphene calculation showing speed-ups in excess of 20×. We could not go to higher number of atoms for v3.2 due to limited memory on the machine (2.67GB per core), yet this work could handle 2160 atoms with a timing of 518 s (Eq.), 2351 s (non-Eq.) per iteration.

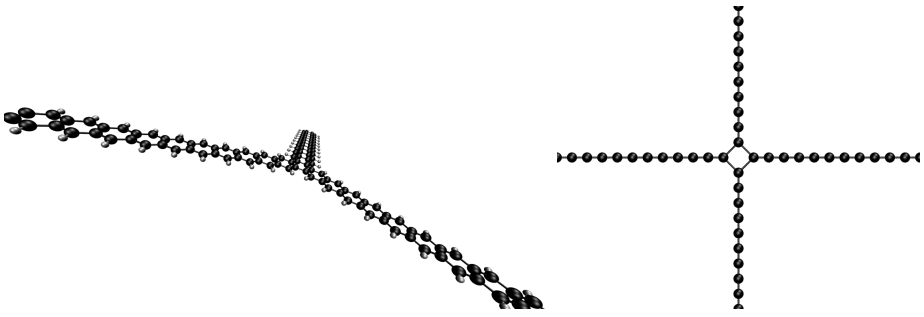

Figure 2: N-terminal setups are now possible in the `TranSIESTA`-code ($N \geq 1$).