Introduction into DFTB

Marcus Elstner

July 25, 2006

1 installing dftb

dftb is partially written in ratfor, that means it has to be precompiled with ratfor.

The ratfor source code is located in DFTB/ratfor compile this an make it available for the makefile of dftb. i.e. move the executable to your bin directory

source code is in: DFTB/src/ to compile the code, you have to specify the size: gmake ATOMS=300 ORBITALS=1200

1.1 input for dftb

dftb < input > output
structure file in gen format is specified in inputfile 'input'

The input file contains information about the calculation, i.e. wether a geometry optimization, MD, or single energy evaluation is desired. It further specifies the Slater-Koster parameters to be read, therefore, it determines which atom types are contained in the system. here is an example of an input file:

mode max-force scc-on scctol read-charges dispersion external-field-:charges
'structure.gen'
charge
constraints
'output.gen'
'L1 L2 L3 L4 ... LN
'potential:atom$_1$-atom$_1$'
'potential:atom$_1$-atom$_2$'

'potential:atom_1-atom_3'
...
'potential:atom_1-atom_N'
'potential:atom_2-atom_1'
...
'potential:atom_2-atom_N'
'potential:atom_N-atom_1'
...
'potential:atom_N-atom_N'
stepsize atomic-temperature electronic-temperature scalfactor number-of-steps

- integer mode: specifies the type of calculation
  1 ... MD with scaling of velocities according to atomic-temperature (last line)
  propability of scaling in each step is given by scalfactor (last line)
  2 ... MD without scaling of velocities according to atomic-temperature
  3 ... Steepest descent relaxation (velocities are set to zero after each step)
  4 ... Conjugate gradient relaxation
  5 ...
  6 ... Mulliken analysis and atomic energy calculation
  7 ... Option number 6 + prints out the eigenvectors
  8 ... BFGS relaxation in internal coordinates
  9 ... Option number 8 + Z-matrix input
  10... Excitation spectrum via linear response
  11... Option 1, read and read velocities from VELOC.DAT (restart option)
  12... Option 2, read and read velocities from VELOC.DAT (restart option)

- max-force: If the total force acting on each atom in the structure is smaller than fmax, the conjugate gradient or steepest descent routine is converged and the program terminates.

- logical scc-on:
  if .T. self-consistent charge mode: see Elstner PRB 1998
  if .F. non-self-consistent charge mode: see Elstner PRB 1995
  real scftol: the convergence criterion (Energy) for the charge SCF convergence.

- logical read-charges: dftb outputs a file CHR.DAT, which contains the atomic (Mulliken) charges: if read-charges=.T. this file is read to accelerate the initial SCF cycle
- logical dispersion: enables the use of empirical dispersion energy on top of the DFTB energy function (Elstner et al JCP 2000): needs additional file, see below
- string external-field-charges:
    - external-field-charges = 'CH' read set of external charges from file EXTCHARGES.INP:
      next
      x-1 y-1 z-1 charge-1
      ...
      x-next y-next z-next charge-next
    - external-field-charges = 'EF' read external electric field
      3 icycle
      x Ex
      y Ey
      z Ez
      icycle: switch field on at cycle icycle, see below
    - external-field-charges = 'NO' this option enables dftb to move system into center of mass
- string: structure.gen: name of structure file: contains initial structure in the 'gen' format as described below
- integer charge: net charge in the system
- integer constraints: there are two options:
    - e.g. 'constraints = N'
      only allow the first N atoms from the structre file to be moved
    - 'constraints = -N': constrain for N atoms x y z coordinates independently
      example:
      -3
      1 0 1 0
      3 1 0 0

5 0 0 0
constrains 3 atoms:
atom # 1: fix x and z coordinate
atom # 3: fix y and z coordinate
atom # 5: fix x, y and z coordinate
a 0 fixes the respective coordinate, a 1 leaves it free

- output.gen: output structure, same format as input.gen

- L1, L2, L3:L basis set size L for atomtypes 1, 2, 3 etc.:
  1 for H, 2 for O, N, C, 3 for S, P, Zn

- 'potential:atom$_1$-atom$_1$'
  'potential:atom$_1$-atom$_2$'
  ...
  paths to the slko-files: e.g.
  /home/sp2/elstner/oo
  ...,
  see example below

- integer stepsize: time step in a.u. for an MD simulation
  should be between 10 and 30 a.u. is there is a hydorgen in the system
  stepsize for relaxation

- real atomic-temperature: temperature for the atoms in K

- real electronic-temperature: Temperature of the electrons, important for metals, but also as a convergence accelerator

- real scalfactor: use this for equikinetical ensemble (mode 2), gives propability that velocities are rescaled according to Temperature at the current step. scalfactor=0.1

- integer number-of-steps: dftb writes in modes 3-10 the output structure to the file output.gen001 after number-of-steps. multiple inputlines are possible, see below.

Structure file 'input'

The structure is given in the 'gen' format. This is e.g:

```
N period
atoms
1 izp1 x y z
2 izp2 x y z
...
N izpN xyz
0.0 0.0 0.0
a1 a2 a3
b1 b2 b3
c1 c2 c3
```

description:

- integer N: number of atoms
- character period:
  - period = 'S': dftb performs calculation for a 'supercell' (S) using periodic boundary conditions, which are given after the coordinate section by:
    0.0 0.0 0.0
    a1 a2 a3: unitcell vector
    b1 b2 b3: unitcell vector
    c1 c2 c3: unitcell vector
  - period = 'C': dftb performs calculation for a molecular 'cluster' (C), no boundary conditions are used, therefore the section defining the unitcell vectors is not necessary
- the block in the middle defines the atomtypes and cordinates: first row is the atom number, the second row defines the atomtyp, which is also an integer in dftb. atomtypes are defined by number 1...

1.2 Makeinp gen2xyz xyz2gen

these are scripts which help converting creating and converting input and output of dftb:

Makeinp creates the input files:
Makeinp 0 o h
creates the input file doh.dftb, which allows for geometry optimization of a system containing the atomtypes o and h, where o is atom-1 and h is atom-2

specified by the potential as discussed above. The structure is in the file in.gen.

In the second row of Makeinp you have to specify the path to the parameter files for atom-1 and atom-2 in this example for oo.spl, oh.spl and hh.spl.

gen2xyz and **xyz2gen** convert the gen to the xyz format:
For the structure in example h2o:

```
gen2xyz in.gen o h
```

converts the gen-format in.gen into the xyz-format, works vice versa with:

```
xyz2gen h2o.xyz O H
```

All the scripts of course work with more atom types as well.

1.3 geometry relaxation

See: examples/h2o

```
dftb < doh.dftb
```

performs a geometry relaxation of the H2O molecule, starting from the structure in in.gen:

Output:

```
* dftb (version 26.11.1988)**
enter mode, fmax, scf, scftol, read charge, dispers, EXT
relaxation mode, conjugate gradient relaxation
enter filename for input structure
infile: in.gen
Input total charge
Total charge set to 0.0000000000000E+000
Number of moveable atoms
No constraints set
enter filename for output structure
enter file name for sk-data of pair 1 1
skfile for pair 1 1:
slko/oo.spl
enter file name for sk-data of pair 1 2
skfile for pair 1 2:
slko/oh.spl
enter file name for sk-data of pair 2 1
skfile for pair 2 1:
```

```
slko/oh.spl
enter file name for sk-data of pair 2 2
skfile for pair 2 2 :
slko/hh.spl
total number of electrons: 8.000000
enter deltat, tatom, telec, wvscale, maxrun
icycle iter niter e(total) e(bandstr) e(bind[kcal]) Edis deltax(max) fmax
===================================================
1 1 / 6 -4.069828 -4.124109 -242.403926 0.000000 0.081592 0.065973
1 2 / 5 -4.074483 -4.154285 -245.325329 0.000000 0.131364 0.023198
1 3 / 5 -4.065288 -4.201479 -239.555007 0.000000 0.120888 0.107737
1 4 / 5 -4.074546 -4.158137 -245.364591 0.0000000.000173 0.026333
1 5 / 4 -4.074546 -4.158073 -245.364601 0.000000 0.067156 0.026274
1 6 / 5 -4.077064 -4.153680 -246.944992 0.0000000.108122 0.011612
1 7 / 5 -4.077508 -4.141894 -247.223717 0.0000000.043256 0.013335
1 8 / 5 -4.077865 -4.147250 -247.447489 0.00000 0.078919 0.007764
1 9 / 5 -4.076432 -4.164989 -246.548130 0.000000 0.064378 0.035234
1 10 / 5 -4.077938 -4.150720 -247.493298 0.000000 0.064378 0.000253

* * * * * end of DFTB * * * * *
```

The first lines of the output document the input as it is read in:
10 optimization steps are performed as indicated by iter,
niter is the number of scf sreps,
e(total) is the total energy in H,
deltax(max)is the maximal displacement and
f(max) is the maximal force in the respective step.

The optimization stops, when either the force criterion maxfor is met or the number of steps is exceeded. The optimized structure is written to out.gen001.
SPE.DAT contains the Kohn-Sham energies and occupations numbers.
CHR.DAT contains the Mulliken charges of the atoms first row and for the orbitals (s, px, py, pz, dxx ...) in the following columns.
REST.DAT contains dipolemoment (D)calculated from the Mulliken charges.
ENERGY.TMP contains the same information as is also printed to standard out.

1.4 Dispersion

1) Reference paper:
M. Elstner, P. Hobza, T. Frauenheim, S. Suhai, and E. Kaxiras, Hydrogen bonding and stacking interactions of nucleic acid base pairs: a density-functional-theory based treatment, J. Chem. Phys. **114** (2001) 5149
2) Input:
a) in inputfile:
first line has to contain a T in the second last entry, i.e.:

```
4 0.0003 T 0.000000001 F T 'NO'
```

see manual
b) you need a file called
DISPERSION.INP
every line contains information for
the atomtypes, as defined in the input file.

i.e. for water dimer : see directory H2O

```
0.560 0.560 0.5600 0.000 3.8 3.8 3.8 3.8 3.15
0.386 0.386 0.000 0.000 3.5 3.5 3.5 3.5 0.8
```

first line: O
second line: H

(for more atomtypes, see DISPERSION.INP-ONCHSP)

first 4 entries: .560 0.560 0.5600 0.000
polarizabilities for different hybridization of atom (see Reference paper)
3.8 3.8 3.8 3.8: the corrsponding R_0, see Reference paper.
3.15: Slater Kirckwood effective electron number, Reference paper.

```
dftb< doh.dftb> doh.out
```

doh.out shows:

```
...
Slater-Kirkwood dispersion switched on:
read C6, Rvdw (eV,A) from DISPERSION.INP
careful, parameter determined from number of H atoms
```

```
icycle iter niter e(total) e(bandstr) e(bind[kcal]) Edis deltax(max) f(max)
==========================================================================
1 1 / 10 -8.161279 -8.302421 -498.377103 -0.000135 0.000000 0.000300
```

Edis= -0.000135 a.u. is the Dispersion energy:
careful: it is already included in e(total),e(bandstr) and e(bind[kcal]).

In force field programs, atom types are defined and the corresponding VdW parameters are acciciated to the atoms hybridization state: However, in the input of an quantum chemical program, we do not have this information:

Solutions:
a) use same polarizabilities for all hybridization states:
not done up to now
b) What is done while reading from DISPERSION.INP is the following,
and only works for DNA bases (but better to check every time):
- the program counts the number of H-atoms, which are attached
to the respective atom:
if the atom has 0 H atom neighbors,
the first entry is used
if the atom has 1 H atom neighbors,
the second entry is used
if the atom has 2 H atom neighbors,
the third entry is used
if the atom has 3 H atom neighbors,
the forth entry is used

for O, they are all the same (0.56), and O has maximum 2 H-neighbors

This is a crude way to assigning the values to hybrid states, and can be refined in the following way:

After one run, the program writes DISP.CHECK
which lists in the first N lines (N = number of atoms)
the C6 coefficient, R_0 and effective electron number
as the program determines it:

now, you can do:

```
cp DISP.CHECK DISP.INP
```

edit DISP.INP in order to assign the VdW parameters you think to appro-

priate
rm DISPERSION.INP,
and the program will read from DISP.INP and not from DISPERSION.INP.

another example: DNA atat strand is found directory DNA

1.5 vib

To install, check directory DFTB/vib/src/

make the executable phonir by typing make!

to run a frequency cacluation, there are several steps to perform:

for example, see vib/h2o :
doh.in is the input file for h2o, in.gen contains the strucuture, the necessary slko-files are in slko/

in.gen contains the optimized structure, you can check by optimizing: dftb< doh.dftb

FREQUENCIES: you need CalcVib.Gen and MoveAtom.Gen in your actual directory:

type "./CalcVib.Gen", as output you get:

```
===
'usage: CalcVib.Gen dftbname inputname Atom1 Atom2 Displacement
where dftbname is the name of the dftb version to use
inputname is the name of the control file for dftb
Atom1 is the atom to start with / 0 means: do header calc.
(Atom2 will then be ingnored...)
Atom2 is the last atom to be moved...
Displacement in Angstroems...
structure file has to be named in.gen '
===
```

dftbname has to contain the actual path to the executable!

```
./CalcVib.Gen ../../src/dftb doh.in 0 3 0.01
```

this job calculates the forces, dipolemoments for displaced atoms of the molecule:
the results are written in:
CHARGE DIPOLE FORCE ENERGY POSITION

the program ../src/phonir constructs the hessian from these data, and calculates the eigenvalues and eigenvectors:

```
../phonir
```

```
Usage: PhonIR project evecprint
```

project and evecprint are logical:
if project= t: project out the translational and rotational degrees of freedom:
if evecprint=t: print out the eigenvectors:

```
../phonir t f
```

will give you the frequencies, zero point energy and optimal geometry: