User Manual for AS_LBFG_v2 , AS_Calc_v2

Tiberius O. Cheche
Faculty of Physics, University of Bucharest, Bucharest 077125, Romania
e-mail: tiberius.cheche@unibuc.ro
tiberiuscheche@yahoo.com

1.1 Introduction

The code AS_LBFG_v2 is created to minimize the elastic energy of semiconductor heterostructures formed by inclusion of QD in a matrix of finite size. The zinc-blende symmetry is considered for the heterostructure compounds. Zinc-blende structure is the generic name given to crystals in which the unit cell has two different atoms; each type of atom is placed in a face-centered cubic (FCC) lattice and each atom is located in the center of a regular tetrahedron which has in the four tips the other type of atom. In this version of the code the shape of QD is of island (*pyramidal*, *semi-torus*, *cone*, *truncated cone*) or spherical (*core-shell*) type. Other QD shapes can easily be introduced in the code by implementing the subroutine which defines the desired shape. The island QD type stands on a *wetting layer* (WL) and is embedded in a matrix (rectangular or cylindrical box). The beneath WL matrix is called *substrate* and the above WL *matrix* out of QD is called *cap*.
The elastic parameters of InAs, GaAs, GaSb are input parameter in the input file type, VFF_v2.inp. The code AS_Calc_v2 is created to handle easier the data generated by AS_LBFG_v2 and calculate the strain for each atomic location from the initial configuration.

1. 2 Installation

The codes AS_LBFG_v2 , AS_Calc_v2 can be compiled by using a Fortran compiler. For example, to install the Intel Fortran Compiler into Microsoft Visual Studio one can follow instructions from
https://www.intel.com/content/www/us/en/developer/articles/training/intel-fortran-compiler-in-ms-visual-studio.html
Once the Fortran compiler is installed, new Fortran projects can be created in Microsoft Visual Studio. The source files can be downloaded from ref. [1].
The codes are also implemented on the Code Ocean platform. On the Code Ocean platform:
(i) The codes can be run by the user without calling any Fortran commands.
(ii) To chose the quantum dot (QD) shape for simulation the input parameters of the codes can be easily manipulated in the input type file VFF_v2.inp of AS_LBFG_v2 (see instructions in section 2.5 AS_LBFG_v2 . Info about input parameters). To obtain the strain field the code AS_Calc_v2 should be run; instructions of how to use this code can be found in section 3. AS_Calc_v2. Source files.
(iii) The codes can be modified by the user by using Fortran commands. A debugger is not implemented.

2. AS_LBFG_v2

The heterostructure considered is of type A_1A_2/A_1A_3 and A_1/A_2 (A_1, A_2, A_3 denote atom element type). Next, we consider as an example GaSb/GaAs heterostructure with A_1, A_2, A_3 as, Ga, Sb, As, respectively.

2.1 AS_LBFG_v2 . Zinc-blende structure

By considering *xy* planes for an FCC crystal grown in the (001) direction, the position of the atoms in the bulk zinc-blende structure is such that one of the two species of atoms (Ga or Sb, for example, in GaSb) is translated relative to the other one by a quarter of the lattice constant, *a*, in each of the three orthogonal directions of the *xyz* Cartesian system of coordinates. Thus there are 4 families of planes parallel to the plane $z=0$, which form the crystal. One starts with a Ga atom placed in the origin of *xyz* Cartesian system of coordinates and for the plane containing it the coordinates x, y, z of the Ga atoms is

$$x = ma\,,\; y = na\,,\; x = a/2 + ma\,,\; y = -a/2 + na\,,\; z = pa \qquad (1a)$$

with *m*, *p*, *n* integers. The second representative family of *z* planes for Ga atoms is translated according to the rule $x \to x$, $y \to y + a/2$, and $z \to z + a/2$; the position of the Ga atoms in this family of *z* planes is given by

$$x = ma\,,\; y = a/2 + na\,,\; x = a/2 + ma\,,\; y = na\,,\; z = a/2 + pa\,. \qquad (1b)$$

By the translations $x \to x - a/4$, $y \to y + a/4$, $z \to z + 3a/4$, from eq. (1a), we obtain the positions of the As atoms in the third representative family of *z* planes as

$$x = a/4(4m-1)\,,\; y = a/4(4n+1)\,,\; x = a/4(4m+1)\,,\; y = a/4(4n-1)\,,$$
$$z = a/4(4p+3)\,. \qquad (1c)$$

By the translations $x \to x + a/4$, $y \to y + a/4$, $z \to z + a/4$, from eq. (1a), we obtain the positions of As atoms in the fourth representative family of *z* planes as

$$x = a/4(4m+1)\,,\; y = a/4(4n+1)\,,\; x = a/4(4m+3)\,,\; y = a/4(4n-1)\,,$$
$$z = a/4(4p+1)\,. \qquad (1d)$$

2.2 AS_LBFG_v2 . Source files

VFF_v2.inp, VFF_Main_v2.f90, VFF_Modul_v2.f90, EnConfig_v2.f90, Sortx123_v2.f90, and VFF_Routines_v2.f90 are files of the code AS_LBFG_v2 . The files blas.f, lbfgsb_v2.f, linpack.f, and timer.f, also included in AS_LBFG_v2 , are implemented from code L-BFGS-B [2]; lbfgsb_v2.f is a slightly modified version (only a few format commands are changed) of file lbfgsb.f from code L-BFGS-B. The code L-BFGS-B minimizes a function of multiple variables subject to bounds by using the function gradient. As a sorting tool the subroutine hpsort_eps_epw [3] is contained by the file VFF_Routines_v2.f90.

- VFF_v2.inp encloses in its script comprehensive explanatory notes about the input parameters of the code.
- VFF_Main_v2.f90 leads the run. In this file, the setulb routine that minimizes the elastic energy is called. It contains a section which introduce bounds (imposed intervals of atom motion) on some atom locations (at the WL-matrix interface and beneath the WL) to simulate the bulk substrate and WL effects.

- VFF_Modul_v2.f90 is a module type file which introduces types of variables.
- EnConfig_v2.f90 encloses subroutines used to calculate the elastic energy and its gradient.
- Sortx123_v2.f90 encloses subroutines used to sort the neighbor atoms.
- VFF_Routines_v2.f90 encloses necessary subroutines.

2.3 AS_LBFG_v2 . Subroutines

In this section the file VFF_v2.inp, and subroutines from VFF_Main_v2.f90, VFF_Modul_v2.f90, EnConfig_v2.f90, Sortx123_v2.f90, and VFF_Routines_v2.f90 are briefly described. More explicit details are provided in the script of the code files.

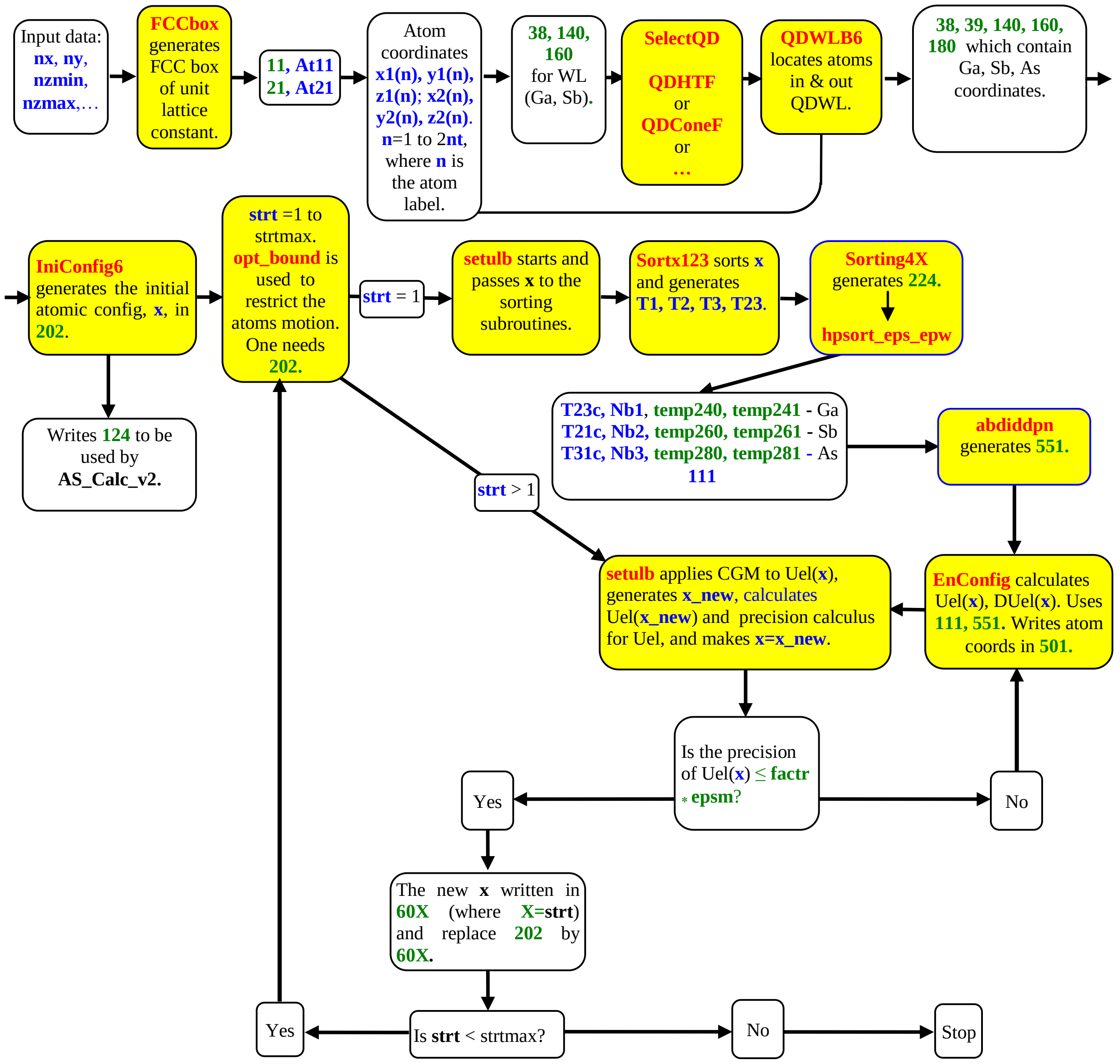
- FCCbox(nx,ny,nzmin,nzmax) generates the box of FCC crystal with latt_ct equal to unity (the coordinates are collected in file fort.11 for At1 and fort.21 for At2) which contains 8*na=8*(2*nx+1)*(2*ny+1)*(nzmin+nzmax+1) as total number of atoms (of type At1 and At2); nx, ny, nzmin, nzmax are integer. Each *xy* plane of the box contains 2(2*nx+1)*(2*ny+1) atoms, each atom being located in the center of a square with the edge of length unity. The *z*=0 plane contains atoms At1 (Ga) and one of this At1 type of atoms is located in the origin (0,0,0) of the Cartesian system of coordinates.
- QDPyramidF(x,y,z,log) generates the rectangular pyramid function in Cartesian coordinates with the base b=2Rc and height h.
- QDConeTruncF(x,y,z,log) generates truncated cone function in Cartesian coordinates with Rct-cone radius, ht-height of cone, hct-height truncated cone (distance between the centers of the two bases).
- QDConeF(x,y,z,log) generates cone function in Cartesian with Rc-cone radius; h-cone height.
- CSQDF(x,y,z,log) generates core-shell QD in Cartesian coordinates with RCO-core radius.
- QDHTF(x,y,z,log) generates semi-torus (ST) function in Cartesian coordinates with Rq-torus radius, Rt-radius of ST; Rt1-internal radius of ST; Rt2-outer radius of ST.
- SelectQD(x1,y1,z1,x2,y2,z2,log1,log2,QD) selects the QD type: QD=1 for semi-torus; QD=2 for cone; QD=21 for truncated cone; QD=3 for pyramid; QD=4 for core-shell.
- QDWLB6(QD,ss,x1,y1,z1,x2,y2,z2,log1,log2,i1IN,i1OUT,i1,i2,i3,WL) generates the FCC coordinates for IN (inside QD) and OUT (in the matrix) QD plus WL (QDWL). The written files: fort.140, fort.160, fort.180 contain coordinates of Ga, Sb, and As atoms, respectively; fort.38, fort.39 contain coordinates of Ga IN and (in the matrix) QDWL, respectively.
- In IniConfig6(x) files fort.140, fort.160, fort.180 are reordered as T1, T2, T3, and T23 (which combines files fort.140 and fort.160). The files fort.201, fort.202 which are generated contain the initial atomic configuration array, x, in the form: x={x(1), x(2), x(3), .., x(3*N13), x(3*N13+1),.., x(3*(N13+N2IN)), x(3*(N13+N2IN)+1), x(3*(N13+N2IN+N4OUT))}, which in Cartesian would mean (x1,y1,z1), (x2,y2,z2),...(xN,yN,zN); N12 is the number of At1 (Ga), N2IN is the number of At2 (Sb), N4OUT is the number of At3 (As) of the heterostructure operated by the code.
- EnConfig(QD,x,Uel,DUel,lbfg,Rcut,Rstart) calculates the elastic energy Uel and the gradient DUel for an atomic configuration array, x; lbfg is introduced to count the loops of EnConfig(...) in setulb; Rcut is an array of dimension 5, which sets the cut-off radius for

searching neighbor atoms (NAs). Rstart is set in VFF_v2.inp as the number of cut-off radii for which the minimization is done; in VFF_v2.inp, by default Rstart=strtmax=1, which means that only one minimization cycle is performed.

- abdiddpn(Rcut,Rstart) collects alfa(q), beta(q), did(q), dpn(q) (used later in EnConfig(…)) in file fort.551. alfa(q), beta(q), did(q), dpn(q) are those obtained in the initial configuration.
- elastic_ct calculates the elastic parameters α and β of binary compound for each material (GaSb, GaAs).
- Sorting4X(i1,T1,T2,T12c,Natt1,Natt2,Nb) finds the first 4 NAs of At_i1 belonging to the sequence T1. T2 is formed by the NA indexes corresponding to At_i1 as a principal atom (PA). T12c stores ascendingly the first 4 ordered distances (d12) between At_i1 (as a PA) and its NAs from T2; Natt1, Natt2 are dimensions of T1 and T2; Nb is a sequence containing the first 4 neighbor indexes ordered as a function of ascendingly associated distance d12 and the same 4 NAs with their index from the T2 sequence.
- Sortx123(x,lbfg) collects distances and indexes of a PA and its first 4 NAs. The initial input configuration, x={x(1),… x(3*(Nat1+Nat2+Nat3))}, is contained in file fort.202 and fort.201. Sortx123(…) generates T1, T2, T3, T23 for a given x configuration; T1, T2, T3, T23 are used by Sorting4X(…) to obtain files necessary to find the neighbors of: At_1(Ga) in temp240.dat; At_2(Sb) in temp260.dat; At_3(As) in temp280.dat. Files temp240.dat, temp260.dat, temp280.dat are used in abdiddpn(…) to obtain the data necessary to compute Uel(n,x) and DUel(LL,:). File fort.111 provides the indexes of each atom (as a PA) and its first 4 neighbors for configuration x.
- opt_bound(bopt,x,nbd,l,u,i) selects between constraints imposed to the atoms coordinates during the minimization process. Two slightly different sets of constraints imposed to the WL-matrix interface atoms may be chosen by setting the bopt parameter to optimize the evaluations in the *line search* of routine setulb.

We found that a sorting done for each minimization cycle generally changes the NA numerical labels (that is an initial NA of some PA can be replaced by another one) but the resulting strain field is more similar with that reported in the literature by sorting (and labeling) NAs only in the initial configuration and keeping unchanged then the NA labels in each loop involved by the minimization procedure. For this reason, after the sorting in the initial configuration we kept unchanged the NA labels of each PA in the minimization loops of the routine setulb.

2.4 AS_LBFG_v2 . Flowchart


Input data: nx, ny, nzmin, nzmax,…
FCCbox generates FCC box of unit lattice constant.
11, At11 21, At21
Atom coordinates x1(n), y1(n), z1(n); x2(n), y2(n), z2(n). n=1 to 2nt, where n is the atom label.
38, 140, 160 for WL (Ga, Sb).
SelectQD QDHTF or QDConeF or …
QDWLB6 locates atoms in & out QDWL.
38, 39, 140, 160, 180 which contain Ga, Sb, As coordinates.
IniConfig6 generates the initial atomic config, x, in 202.
strt =1 to strtmax. opt_bound is used to restrict the atoms motion. One needs 202.
strt = 1
setulb starts and passes x to the sorting subroutines.
Sortx123 sorts x and generates T1, T2, T3, T23.
Sorting4X generates 224. hpsort_eps_epw
Writes 124 to be used by AS_Calc_v2.
T23c, Nb1, temp240, temp241 - Ga T21c, Nb2, temp260, temp261 - Sb T31c, Nb3, temp280, temp281 - As 111
abdiddpn generates 551.
strt > 1
setulb applies CGM to Uel(x), generates x_new, calculates Uel(x_new) and precision calculus for Uel, and makes x=x_new.
EnConfig calculates Uel(x), DUel(x). Uses 111, 551. Writes atom coords in 501.
Is the precision of Uel(x) ≤ factr * epsm?
Yes
No
The new x written in 60X (where X=strt) and replace 202 by 60X.
Yes
Is strt < strtmax?
No
Stop

Color legend: yellow background box frame is for routines or subroutines; white background box frame is for intermediate data or commands; red is used for the name of routines or subroutines; blue is used for scalars, vectors, or arrays; green is used for the name of files.

2.5 AS_LBFG_v2 . Info about input parameters

In VFF_v2.inp the input parameters are introduced are as follows.

- QD holds for the shape of QD: QD=1 for hemi-torus; QD=2 for cone; QD=21 for truncated cone; QD=3 for pyramid; QD=4 for core-shell.
- bound sets the code to minimize the elastic energy by using bound and/or unbound variables. If bound=0 atomic positions are freely changed during the relaxation process (unbound variable). If bound=1 atomic positions have a limited range of displacement during the relaxation process (bound variable). In VFF_Main_v2.f90, for bound=1, a limited range of motion is set for (bound) atoms at the WL-QD, WL-cap, WL-substrate interfaces. If variable nbd(atom_coordinate)=0, the atom coordinate is unbound, and if nbd(atom_coordinate)=2, the atom coordinate is bound. For core-shell setting is bound=0.
- bopt works for QD=1, 2, 21, 3. It is a parameter which allows choosing between two sets of constraints imposed to the WL-matrix interface atoms. bopt=1 or 2 selects between the two settings.
- clt1 and clt2 are coefficients used for the substrate and WL, respectively, simulation.
- In VFF_Main_v2.f90: x(i), x(i+1), x(i+2) are the x, y, z, respectively, coordinates of the atom labeled as Mod(i-1,3)+1.
- In the code the lengths are expressed in Ångström unit.
- Rt is the torus radius, Rq is the tube radius of the torus.
- Rc is the base con radius or half the edge of the square base, h is the cone height.
- Rct is the cone radius, ht is the height of cone, hct is the height of truncated cone (distance between the centers of the two bases).
- RCO is the core radius, RCS is the core+shell radius.
- a1 is e.g., the GaAs lattice constant; a2 is e.g., the GaSb lattice constant.
- Nw is the number of mono-layers.
- nx, ny are the number of atoms at z=0 for x>0, y>0. nzmin is the number of atoms on negatíve z axis. nzmin is the number of atoms on positive z axis.
- C11out is elastic constant for GaAs. C12out is elastic constant for GaAs. C11in is elastic constant for GaSb, InAs. C12in is elastic constant for GaSb, InAs. For C11out, C12out, C11in, C12in see refs. [4, 5, 6].
- radius is the cut-off radius for searching and sorting NAs in the initial atomic configuration; its value is automatically changed in the program to ensure at least 4 NAs for each PA.
- radiuscut1 is cut-off radius for searching NAs in the 1st minimization process (default). radiuscut2 is cut-off radius for searching NAs in the 2nd minimization process (if set up). radiuscut3 is cut-off radius for searching NAs in the 3rd minimization process (if set up). radiuscut4 is cut-off radius for searching NAs in the 4th minimization process (if set up). radiuscut5 is cut-off radius for searching NAs in the 5th minimization process (if set up). radiuscut1,2,3,4,5 is close(larger) to dc12, dc13 (see subroutine elastic_ct).
- 4, 2.7, 3.7, 2.7, 1.8, 2.7 ! radius,radiuscut1,radiuscut2,radiuscut3,radiuscut4,radiuscut5-is by default setting.

- strtmax is the number of minimization processes (the default value is 1 and it corresponds to radiuscut1). strtmax=5 corresponding to radiuscut1,2,3,4,5.
- factr sets the accuracy parameter for the calculus of f = Uel:
  (f^k - f^{k+1})/max{|f^k|,|f^{k+1}|,1} <= factr*epsmch. Typical values for factr:
  1.d+12 for low accuracy;
  1.d+7 for moderate accuracy;
  1.d+1 for extremely high accuracy.
  In the simulations I used factr= $1.d+10 \div 1.d+11$ to obtain the convergence.
- latshift1, latshift2 set the lateral size (in the *xy* plane) of matrix and substrate boxes (see in VFF_Main_v2.f90).

2.6 AS_LBFG_v2 . Notes

- The index associated to the atoms is an ordering label (OL) which remains unchanged after it is assigned; each atom is seen both as a PA and a NA but has an only one OL:
  OL of Ga: 1,..., N13; OL of Sb: N13+1,..., N13+N2IN; OL of As: N13+N2IN +1,..., N13+N2IN+N4OUT;
  Example: A PA has OL=2 and its first 4 NAs in the unrelaxed (initial) config could be, e.g., OL=11, 12, 13, 15.
  The 4 neighbor OLs of any PA (which has its own OL) remains the same in each minimization step.

- To check the convergence during the minimization process the files fort.6ab0, where the number ab0=100*a+10*b, collect the atoms coordinates and the precision with which they are calculated. The files fort.111, fort.124, fort.201, fort.6ab0/601, temp240.dat, temp260.dat, temp280.dat are input files for the code AS_Calc_v2. At the end of minimization process the file fort.601 generated by AS_LBFG_v2 collects the atoms coordinates and the precision set in the input file for their calculus. Code AS_Calc _v2 calculates the strain tensor elements.

- The total estimated computation time is displayed on the screen at the first iteration as:

```
==================================================
 INFO:
  The estimated total time calculus is    …     s
 ==================================================
```

2.7 AS_LBFG_v2. Adding New QD shape

A new QD shape function can be added by:

1. Coding a subroutine following the examples QDPyramidF(..), QDConeTruncF(x,y,z,log), etc., from the file VFF_Routines_v2.f90;
2. Entering an integer to the parameter QD, for example QD=11, corresponding to the new QD shape function in VFF_v2.inp;
3. Calling the new created subroutine in SelectQDT(..);
4. Entering QD==11 in Subroutine QDWLB6(..) in the command line following the commented lines !@ and paying attention to the QD type, island or non-island;

5. Introducing QD==11 in VFF_Main_v2.f90 following the examples for the other QD values inside this Fortran file.

3. AS_Calc_v2. Source files

The strain field is calculated with data provided by AS_LBFG_v2 by using the files fort.111, fort.124, fort.201, temp280, temp240, temp260, and fort.601 or fort.6ab0 (e.g., fort.6010, fort.6120, ...). The files for the atomic positions generated by AS_LBFG_v2 are as follows: fort.38, fort.39, fort.140, fort.160, fort.180, fort.201.

- Strain_v2.f90 leads the run. The method of calculus the strain is that used in refs. [3] or [7]. Files which collect the strain tensor elements in various directions and plans are generated.
- Misfit_v2.f90 collects the misfit dislocations in files fort.502, fort.503, fort.504 for QD=1, 2, 21, 3 and in files fort.571, fort.572, fort.573, fort.574 for QD=4.
- Modul_Strain_v2.f90 is a module type file which introduces types of variables.
- StrainIni_v2.inp encloses comprehensive explanatory notes about the input parameters of the code.
- If a new QD shape function is be added then the subroutine created in code AS_LBFG_v2 should be added to AS_Calc_v2 also. QD==11 (see subsection 2.7) should be introduced in both Strain_v2.f90 and Misfit_v2.f90 files following the examples for the other QD values inside these Fortran files.

References

[1] https://github.com/tocheche/AS_LBFG_v2; https://github.com/tocheche/AS_Calc_v2
[2] C. Zhu, R. Byrd, J. Nocedal, J.L. Morales, L-BFGS-B (version 3.0), April 25, 2011. http://users.iems.northwestern.edu/~nocedal/lbfgsb.html
[3] S. Poncé, R. Margine, C. Verdi, F. Giustino, Copyright (C) 2010-2016; J. Noffsinger, B. Malone, F. Giustino, Copyright (C) 2007-2009, https://github.com/dceresoli/q-e-d-c/blob/master/EPW/src/sort.f90
[4] C. Pryor, J. Kim, L. W. Wang, A. J. Williamson, A. Zunger, J. Appl. Phys. 83 2548 (1998).
[5] D. Barettin, S. Madsen, B. Lassen, M. Willatzen, Commun. Comput. Phys., **11**, No. 3, pp. 797-830 (2012).
[6] http://www.ioffe.ru/SVA/NSM/Semicond/GaSb/mechanic.html
[7] P.M. Gullett, M.F. Horstemeyer1, M.I. Baskes, H. Fang, Modelling Simul. Mater. Sci. Eng. 16 (2008) 015001.