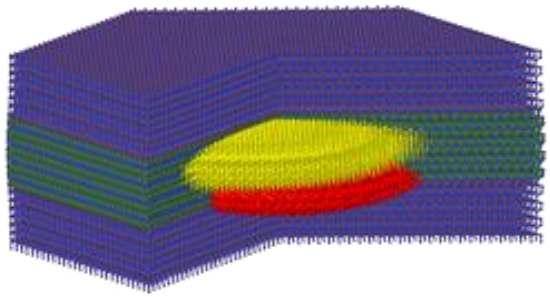
Atomistic

tiberCAD multiscale simulation



Atomistic is a set of tools for the simulation of nanostructures based on atomistic approaches. It contains a Module for structure relaxation of atomic structures based on **Valence Force Field** and a Module for **Empirical Tight Binding** calculations. An Atomistic Generator allows to generate an atomic basis associated with the finite element mesh which belongs to a given physical region, Empirical Tight Binding module allows for atomistic-based calculations of electronic and optical properties of Nanowires. Quantum Dots and Quantum Wells. Eigenstates, eigenfunctions and quantum density of a given system can be obtained by solving a tight-binding Hamiltonian.

- Built-in Atomistic Generator for several crystal structures
- Atomistic-based Empirical Tight Binding calculations of optoelectronic properties of Nanowires, Quantum Dots and Quantum Wells
- Accurate parameterization for several materials, including GaN/AlGaN/InGaN systems
- Random alloy approach to treat in a fundamental way nanometric features in active regions, such as alloy fluctuations

Atomistic

Atomistic block

```
Device
{
Region  ….

Region ….

Atomistic   <atomistic_structure_name>
{
   reference_region

    print = (xyz, gen, tgn)
    random_alloy = true
}

…
```

Generation of atomic basis associated with the FEM mesh which belongs to a given physical region, based on the material specifications and the growth directions defined for that region.

Atomistic block

Atomistic <*atomistic_structure*>
{
reference_region = <*ref_for_lattice_building*>
regions = <*region*|*cluster*>

passivation = *yes*|*no*

print = (*xyz*, *gen*, *tgn*)

random_alloy = *true*|*false*
}

…

region whose material and growth directions are taken as a reference for lattice construction

hydrogen passivation is applied to the structure

if true, a random alloy structure is generated

Atomistic block

Atomistic <*atomistic_structure*>
{
periodicity = true

for a 1D mesh, add periodicity in *x* direction; for a 2D mesh, add periodicity along *x* and/or *y*; for a 3D mesh, add periodicity along *x,y* and *z*

supercell_size_y = <*value*>
supercell_size_z = <*value*>

to build a supercell in the *yz* plane

load_structure = <*path*>
}

Load external atomic structure in *xyz, gen* file formats

…

Atomistic

Module ***VFF:***
Valence Force Field

Module VFF {
  name = *somename*
  regions = *set_of_regions*
  atomistic_structure = *tb*

boundary_conditions = *free_standing* | *substrate*| *all_around.*

Solver {
absolute_tolerance = ….
}

B.C.:
- All atoms relax,
- atoms belonging to an initial layer are fixed,
- all the outer atoms are fixed

tiberlab

Module ***empirical_tb***

Module empirical_tb{
  name = *somename*
  regions = *set_of_regions*
  atomistic_structure = *tb*

Solver {
  num_valence_eigenvalues = <*value*>
  num_conduction_eigenvalues = <*value*>
  }
……

On this atomic basis, **Empirical tight binding** is solved to find eigenvalues an wavefunctions

Module ***empirical_tb***

Module empirical_tb{

Harrison_scaling = true | false
strain_simulation = < *sim_name*>

potential_simulation = <*sim_name*>

Solver {
guess_conduction
guess_valence
}

Displacements from Elasticity or VFF projected to ETB Hamiltonian, through **scaling** of coupling matrix elements

Potential profile projected to ETB Hamiltonian: shift of on-site elements

Atomistic

Module ***optics_tb***

Module opticstb{
  name = <*somename*>
  regions = <*set_of_atomistic_regions*>

 initial_state_model = <*etb_sim*>
 final_state_model = <*etb_sim*>

}

optical matrix elements are calculated from the ETB simulations specified

Atomistic

Example: ***Qdot ETB***

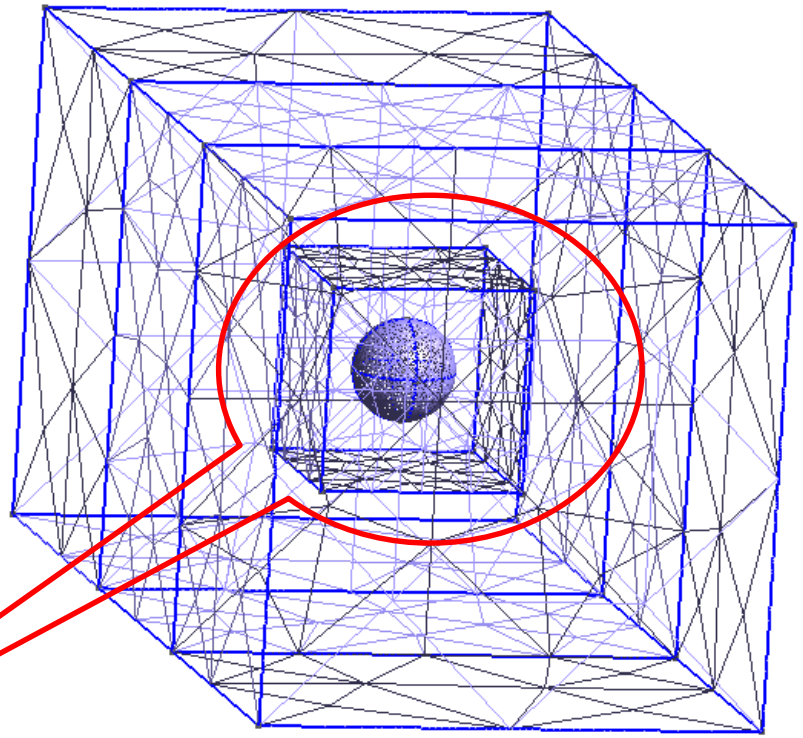
```
Region ball
  {
    material = GaN

    Doping {
      Nd = 1e15
      type = donor
      Ed = 0.025
    }
}
Region  qbox
  {
    material = AlN

    Doping {
      Nd = 1e15
      type = donor
      Ed = 0.025
    }}

    Cluster atomistic
  {
    regions=(ball, qbox)
  }


  }
```

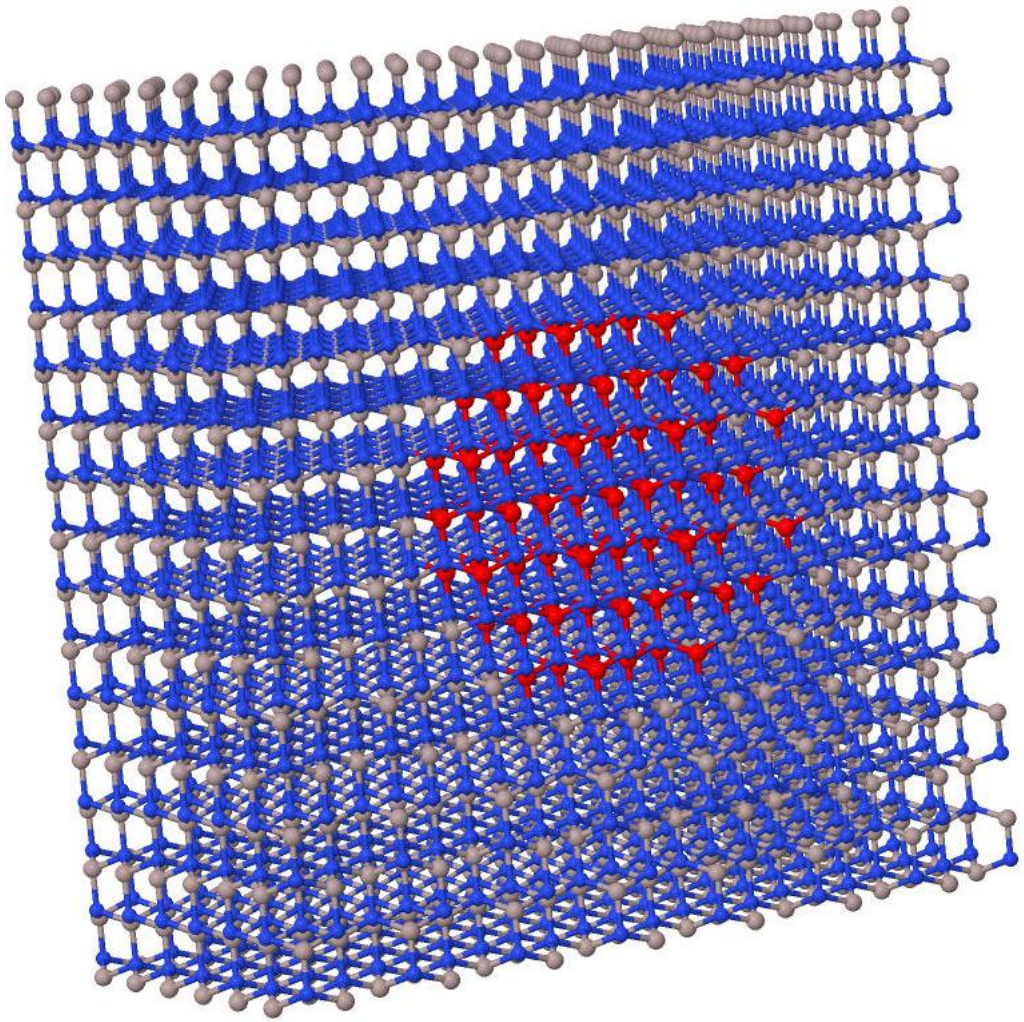


Atomistic

Example: ***Qdot ETB***

```
Atomistic tb1
{
  regions = atomistic
  reference_region = nside

  passivation = yes
  print = (gen, xyz, tgn)
}
```



Atomistic

Example: ***Qdot ETB***

```
Module empirical_tb
{
  regions = atomistic
  name = tb1

  atomistic_structure = tb1
  potential_simulation = dd


  plot =(tbstates, MeshStatesNodes )

  Solver
  { num_valence_eigenvalues = 1
    num_conduction_eigenvalues = 1
    long_tolerance = 1e-4
  }

}
```