Modelling 5/7 Ring Dislocations in InGaN Using CONQUEST Density Functional Theory Code

Harry Youel
University College London

September 28, 2022

Contents

Abstract

Gallium nitride (GaN) is commonly used in the production of light emitting diodes (LEDs), disk reading technology as well as many other semiconducting power devices. It is possible to alloy GaN with indium nitride (InN) to drive the emission of a prototype LED towards the UV spectrum. Threading dislocations, such as 5/7 rings, form when a GaN sample is grown from a substrate which contains defects. Dislocations can also occur when there is a difference in lattice parameters between a substrate and a grown GaN sample [1]. The arising dislocation cores can effect a device's efficiency.

This investigation aims to determine the energetically favorable positions of In atoms near commonly found GaN 5/7 ring dislocation cores, using CONQUEST denisty functional theory (DFT) code [2, 3, 4, 5]. The purpose of this report is to understand the properties and behaviour of such dislocation cores.

ASE (Atomic Simulation Environment) [6] was utilised to create an appropriate wurtzite basis cell which was then structurally relaxed using CONQUEST. Dislocations of varying x sizes were then formed using ASE's 'supercell' function before being edited to approximate the 5/7 rings [7], which were again structurally relaxed. An In atom was substituted for a Ga atom in a chosen plane and the energy was calculated for all possible In atom locations on that plane. The optimum value for said In atom was discovered to be located near the dislocation core (Fig 12).

Therefore, this investigation provides some evidence that an In atom's most energetically favourable position in an InGaN 5/7 ring dislocation cell is at the dislocation core (Fig 21).

Acknowledgements

This project would not have been possible without Prof David Bowler, as his continued support and guidance throughout this project was invaluable. I would also like to express my gratitude to Dr Shereif Mujahed, who provided essential advice and assistance during the completion of this project.

1 Introduction and Background Theory

This investigation utilises CONQUEST DFT code [2, 3, 4, 5] to investigate the energy of an InGaN dislocation sample. Electronic structure methods such as DFT have several uses including: computing physical and chemical properties of a system (energy, pressure, stress, etc.), identifying the band structure of a crystal as well as understanding reaction mechanics [8]. This investigation is primarily concerned with how the position of a doped In atom in a GaN dislocation core affects the system's energy and consequently the forces on the constituent atoms.

1.1 Why investigate Indium Gallium Nitride?

The material system in question is an In doped GaN dislocation core (InGaN). GaN is a semiconducting material which is widely used in areas such as: Blu-ray disk reading, radio frequency devices, lasers and photonics. In order to tune the bandgap of GaN for the emission of UV radiation ($\sim$ 400nm), indium nitride (InN) can be added to generate an InGaN alloy. This alloy can be used in photovoltaics, which is the central process in the production of solar technology [9, 10]. Using (In)GaN semiconductors over other semiconducting materials like silicon have the following advantages: increased efficiency, higher power density, higher switching frequency and lower system cost [9].

Dislocations in InGaN can affect device efficiency. It is important to investigate the most energetically favourable positions of In atoms within the wurtzite cell structure to provide an insight into a possible optimal InGaN atomic structure. This investigation focuses on modelling a 5/7 ring threading dislocation, which occurs naturally in InGaN. These dislocations can occur if there is a defect on the base substrate that the crystal sample is grown from, or if there is a difference in lattice parameters and thermal expansion coefficients between the substrate and sample [1].

1.2 Electronic Structure Methods

There are two main branches of electronic structure methods: the quantum chemistry approach (wavefunction based), and density-based methods (charge-density based). The former stems from the Hartree-Fock approximation, which involves constructing many-body wavefunctions using a combination of single-particle wavefunctions. The latter builds a spatially dependent charge density from the single-particle wavefunctions that make up the electronic structure [11].

1.3 DFT Overview

DFT (density functional theory) is an ab inito method which tackles the issue of solving many-body electronic Hamiltonians by applying a mean field approximation and re-expressing the electron-electron interaction term with an effective potential $V_{eff}(\mathbf{r})$. This effective potential is combined with the nuclei potential to construct a single external potential for each electron which is expressed in terms of the total charge density $n(\mathbf{r})$. The overall aim of DFT calculations is to determine the ground state charge density [11].

All many-body interactions are encompassed by a functional of charge density. This is referred to as the exchange-correlation functional $E_{XC}[n(\mathbf{r})]$ and contains: exchange effects, correlation effects and the kinetic energy differences between independent and many-body electrons [11].

1.4 Jacob's Ladder

As the exact form of Kohn and Sham's exchange-correlation functional is unknown, it must be approximated. John Perdew, who was an early pioneer of DFT, first coined the idea of *Jacob's Ladder* (Fig 1), which indicates the progression of functionals from simple LDSAs to the theoretical and yet unknown exact functional. Improving the exchange-correlation functional approximation is a very active field of research in DFT.

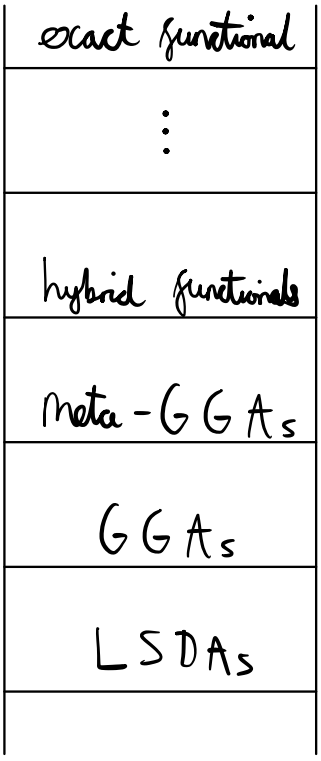

Figure 1: Jacob's Ladder illustration showing the progression of functionals.

1.4.1 LSDAs

Local spin density approximations (LSDAs) represent the most primitive approximation form. It assumes that the energy density is local and reflects the energy density that a uniform electron gas possesses ($\nabla n(\mathbf{r}) << 1$) [11, 12]. This can be represented by [13]:

$$E_{XC}^{LSDA}[n(\mathbf{r})] = \int n(\mathbf{r})\epsilon_{XC}(n(\mathbf{r}))d\mathbf{r}$$

Where the electron spin densities are $n(\mathbf{r}) = n_{up}(\mathbf{r}) + n_{down}(\mathbf{r})$ and $\epsilon_{XC}(n(\mathbf{r}))$ is the exchange-correlation energy per particle of a homogenous electron gas (jellium). The success of LSDAs arises due to the error cancellation between the exchange and correlation terms [11]. Limitations of this approximation include the tendency to over-bind as well as the lack of accuracy when modelling the formation or breaking of bonds [11, 12].

1.4.2 The GEA and GGAs

The gradient expansion approximation (GEA) attempts to improve upon LSDAs. This is achieved by considering the inhomogeneity of electron densities which gives rise to the inclusion of first order terms in $n(\mathbf{r})$ [12, 14]:

$$\epsilon_{XC}[n(\mathbf{r}), |\Delta n(\mathbf{r})|] = \epsilon_{LDSA}[n(\mathbf{r})] + f_1(n(\mathbf{r}))|\Delta n(\mathbf{r})| + f_2(n(\mathbf{r}))|\Delta n(\mathbf{r})|^2$$

The first order term in $\Delta n(\mathbf{r})$ is zero [12], which suggests that there is an issue surrounding the depletion of density around each electron (exchange-correlation hole). To circumvent this problem: firstly, holes with $n(\mathbf{r}) > 0$ are set to zero and secondly, a sum rule must be invoked ($n_{XC}(\mathbf{r}) = n_X(\mathbf{r}) + n_C(\mathbf{r})$) to truncate holes which violate this rule hence ensuring each exchange-correlation hole is negative definite [12, 13, 14]. Generalized gradient approximations (GGAs) arise from the resulting treatment of exchange-correlation holes and can be represented by [14]:

$$E_{XC}^{GGA}[n(\mathbf{r})] = \int n(\mathbf{r})\epsilon_{XC}[n(\mathbf{r}), \nabla n(\mathbf{r})]d\mathbf{r}$$

The extra degrees of freedom imparted from the addition of energy variation results in the existence of a family of GGAs including PBE96 which is utilised in the subsequent InGaN calculations. [11, 14].

1.4.3 PBE96

The Perdew, Burke and Ernzerhof 1996 functional (PBE96) was used for all DFT calculations involving InGaN and has the following form [13]:

$$F_X(s) = 1 + \kappa - \frac{\kappa}{1 + \frac{\mu s^2}{\kappa}}$$

Where s is a dimensionless density gradient $s = \frac{|\Delta n(\mathbf{r})|}{2k_F n(\mathbf{r})}$, $\mu \approx 0.21951$ and $\kappa \approx 0.804$. $F_X(s)$ is an enhancement factor which is related to exchange energy by [13]:

$$E_X^{GGA}[n(\mathbf{r})] = \int n(\mathbf{r})\epsilon_X[n(\mathbf{r})]F_X(s)d\mathbf{r}$$

With $\epsilon_X[n(\mathbf{r})]$ representing the exchange energy per particle. The correlation energy is defined as [13]:

$$E_C^{GGA}[n(\mathbf{r})] = \int n[\epsilon_C(r_s, \zeta) + H(r_s, \zeta, t)]d\mathbf{r}$$

Where $\epsilon_C(r_s, \zeta)$ is the correlation energy per particle, r_s is the local Seitz radius ($n = \frac{3}{4}\pi r_s^3 = \frac{k_F^3}{3\pi^2}$), ζ is the relative spin polarization and t is another dimensionless density gradient $t = \frac{|\nabla n|}{2\phi k_s n}$ [13].

1.4.4 Improvements on GGAs

Meta-GGA functionals add an extra layer of complexity due to the inclusion of the energy density's Laplacian [14]:

$$E_{XC}^{meta-GGA}[n(\mathbf{r})] = \int n(\mathbf{r})\epsilon_{XC}(n(\mathbf{r}), \nabla n(\mathbf{r}), \nabla^2 n(\mathbf{r}))d\mathbf{r}$$

Hybrid functionals attempt to combine approximations from the Hartree-Fock method (exact exchange) and a definite interacting homogenous electron gas (LSDAs and GGAs) by using a linear combination of exchange correlation energies [14]:

$$E_{XC} \approx \alpha E_{HF} + \beta E_{XC}^{GGA}$$

2 CONQUEST Methodology

Concurrent Order N Quantum Electronic Structure (CONQUEST) [2, 3, 4, 5] is a local orbital DFT code which was utilised in this investigation to model InGaN. The support functions are made up of basis functions which can be either: pseudo-atomic orbitals (PAOs) or b-splines (PAOs are the chosen basis functions for all InGaN calculations due to the relative lack of complexity). The most primative PAO construction arises when each support function is represented by a single PAO (which was used in this investigation), while larger systems may require multi-site support functions or on-site support functions [15]. Pseudo-atomic orbitals are defined as solutions to the Schrodinger equation for an isolated atom which utilises pseudo-potentials. PAOs are made up of radial functions multiplied by spherical harmonics [15]. The formal definition of a single particle density matrix is:

$$\rho(\mathbf{r},\mathbf{r'}) = \sum_n f_n \psi_n(\mathbf{r})\psi(\mathbf{r'})$$

This density matrix can be represented in terms of PAO support functions by [15]:

$$\rho(\mathbf{r},\mathbf{r'}) = \sum_{i\alpha j\beta} \phi_{i\alpha}(\mathbf{r}) K_{i\alpha j\beta} \phi_{j\beta}(\mathbf{r'})$$

Where $\phi_{ij}(\mathbf{r})$ represents the support functions and $K_{i\alpha j\beta}$ is the density matrix expressed in the same basis as the support functions [15]. The Kohn-Sham eigenstates are represented by:

$$\psi_{n\mathbf{k}}(\mathbf{r}) = \sum_{i\alpha} c_{i\alpha}^{n\mathbf{k}} \phi_{i\alpha}(\mathbf{r})$$

The density matrix $K_{i\alpha j\beta}$ is found using diagonalisation as the system in question typically contains ~ 100 atoms, while larger systems would be better suited to linear scaling.

Due to the self-consistency problem between the effective potential's components ($V_{eff}(\mathbf{r}) = V_{Har}(\mathbf{r}) + V_{XC}(\mathbf{r}) + V_{eI}(\mathbf{r})$) and the charge density, an iterative approach must be taken. Pulay mixing takes the form [11, 15]:

$$\rho_{n+1}^{in}(\mathbf{r}) = \sum_i \alpha_i [\rho_i^{in}(\mathbf{r}) + AR_i(\mathbf{r})]$$

Where the residual can be expressed by: $R(\mathbf{r}) = \rho^{out}(\mathbf{r}) - \rho^{in}(\mathbf{r})$ and the linear mixing factor (A) is set to 0.5 by default. The Pulay parameters are represented by α_i.

Kerker preconditioning was also used in order to prevent 'charge sloshing'. This phenomenon is encountered when the system has a particularly large cell parameter in one dimension. The significance of charge sloshing is prevalent, because a small variation in the input density can drastically change the output density due to the dependence on long-wavelength charge deviations [15, 16]. Kerker preconditioning prevents this from affecting the self-consistency procedure by applying the following function to the Fourier transform of the residual [15]:

$$(\frac{q^2}{q^2+q_0^2})\tilde{R}$$

The Kerker factor q_0 has a default value of 0.1 which marks the threshold for long-range real-space wavelengths (short-range reciprocal-space wavevectors) in the residual [16].

Periodic boundary conditions are also enforced on the cell. Bloch's theorem describes how the wavefunctions of atoms which are translated by a cell length in any specified direction only differ by a phase factor [11]:

$$\psi(\mathbf{r}+\mathbf{A}) = e^{i\mathbf{k}\cdot\mathbf{A}}\psi(\mathbf{r})$$

Where $\mathbf{A}$ is an integer multiple of the cell width in a particular direction and $\mathbf{k}$ is an appropriate wavevector [11].

2.1 Input file

The input file is denoted as:

```
Conquest_input
```

The command:

```
AtomMove.TypeOfRun static
```

signals the system's freedom to structurally relax. In this case, the option to restrict atomic movement is indicated by the 'static' setting while other options such as 'cg' allow the lattice constants to relax using the conjugate gradient approach as well as 'sqnm', which optimises the atomic positions. In order to optimise the lattice constants using the conjugate gradient approach the following lines must be added to the file:

```
AtomMove.OptCell T
AtomMove.OptCellMethod 1
```

Activating the Monkhorst-Pack (MP) mesh and specifying the number of k-points that are to be assigned to each dimension can be accomplished by adding:

```
Diag.MPMesh T
Diag.MPMeshX N
Diag.MPMeshY N
Diag.MPMeshZ N
```

Where generally each component is inversely proportional to the lattice constant in that dimension (e.g. a lattice with $a = 10a_0, b = 10a_0, c = 5a_0$ should be assigned a $1 - 1 - 2$ MP mesh).

In order to centre the MP mesh on the gamma point, the following command must be specified:

```
Diag.GammaCentred T
```

The coordinate file is specified by:

```
IO.Coordinates xxxxxx
```

and will be discussed in greater detail in the next subsection.

The integration grid's energy spacing is specified by the following command:

```
Grid.GridCutoff N
```

and must be a factor of 3, 4 or 5 in all directions to satisfy the fast Fourier transform algorithm requirements [15].

The number of chemical species and their ion files references are indicated here:

```
General.NumberOfSpecies 2

%block ChemicalSpeciesLabel
1   69.720  Ga_DZP
2   14.010  N_DZP
%endblock ChemicalSpeciesLabel
```

The chemical species label block has the following form: reference number (from coordinate file), atomic mass and ion file.

Changing the residual's tolerance in the self-consistency iteration can be achieved by inputting:

```
minE.SCTolerance N
```

Adjusting this value can affect the accuracy of the procedure but at the expense of incurring more iterations.

Kerker preconditioning can be activated and the Kerker factor can be set by adding:

```
SC.KerkerPreCondition T
SC.KerkerFactor xxx
```

Visualisation for Electronic Structural Analysis (VESTA) [17] was used to visualise structures that were created. In order to make the '.cell' file for VESTA to read, the following line can be added:

```
Process.CoordFormat.cell
```

2.2 Coordinate file

The atomic coordinate file specified in the input file details the lattice structure. The upper matrix is forbidden to have off-diagonal components as unit cells are required to be orthorhombic:

```
a   0.0 0.0
0.0 b   0.0
0.0 0.0 c
```

With a, b and c representing the lattice constants in x, y and z (a_0). The number of atoms and their fractional positions are specified below:

```
No.Atoms
    0.000   0.000   0.000   1   T/F T/F T/F
    0.500   0.500   0.500   2   T/F T/F T/F
       .       .       .    .    .   .   .
       .       .       .    .    .   .   .
       .       .       .    .    .   .   .
```

Where the integers (1,2,...) represent the type of atom as specified by the input file, while the T/F option indicates the atoms' freedom to move in each dimension.

2.3 Ion files

Ion files contain information concerning each atomic species such as the atomic number, atomic mass, valence charge and details surrounding PAOs. The frozen core approximation is utilised here such that the core electrons are fixed and do not impact the charge density. For example, Ga has three valence shells: 3d 4s 4p. These valence orbitals are referred to as zetas and unoccupied orbitals are referenced as polarisations [15]. The ion files used in this investigation are denoted 'DZP' (double zeta, single polarisation) and contain two radial functions for each orbital angular momentum value and a single radial function for unoccupied states.

2.4 Output file

The output file is denoted as:

```
Conquest_out
```

To begin the Pulay iterations, the GGA functional is stated as well as the chosen basis set:

```
The functional used will be GGA PBE96
Using PAOs as basis set for support functions
```

The self-consistency tolerance, linear mixing factor (up and down channels) [15] and the Kerker preconditioning factor is then displayed.

```
Starting self-consistency. Tolerance: 0.10000E-06

Starting Pulay mixing, A_up = 0.500 A_dn = 0.500
Spin non-polarised calculation.
     with Kerker preconditioning, q0 = 0.100
```

The Pulay mixing will then begin, outputting the Harris-Foulkes Energy, DFT Total Energy, Pulay iteration index and residual value until the self-consistency tolerance is reached.

```
Harris-Foulkes Energy:    -11016.500047744037147 Ha
DFT Total Energy:         -10992.006163539466797 Ha
Pulay iteration 1 residual: 0.94373E-03
```

For a static calculation, this will stop when the self-consistency tolerance is met and will output: forces on each atom, total stress and total pressure as well as some additional data such as the maximum total memory use and total run time.

If the structure's lattice constants have been instructed to relax using the conjugate gradient method, the process will repeat and the new cell dimensions will be outputted as well as the following indicator of a successful optimisation step:

```
GeomOpt - Iter: 1 MaxStr: 0.17872750 H: -0.10942971E+05 dH: 0.00202645
```

This will continue until the pressure tolerance level is satisfied or the change in enthalpy is sufficiently small that the optimiser terminates. A similar method will be utilised for an atomic positions optimisation, but in this case it is the maximum force that must reach a predetermined tolerance. As well as this, setting the cell optimisation method to '3' will simultaneously optimise the cell parameters and atomic positions.

3 Cell Creation

3.1 Wurtzite Basis Cell

GaN has a wurtzite basis cell structure. The approximate atomic position layout has the form shown in Fig 2.

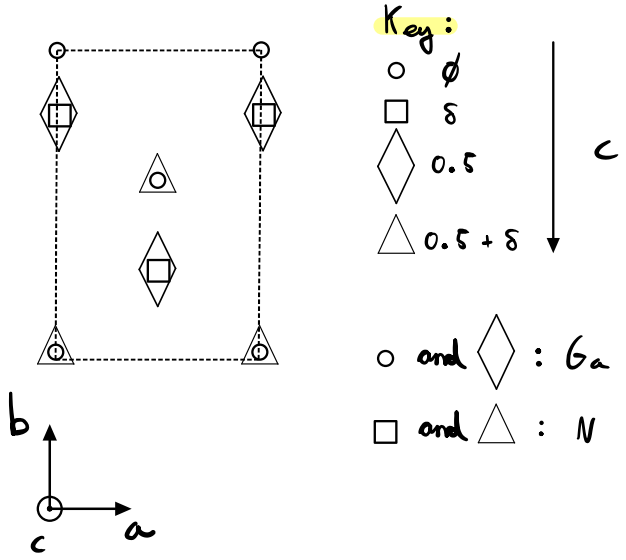

Figure 2: Illustration of wurtzite basis cell schematic (the z-dimension is out of page). δ and ϕ are both small intrinsic fractional distances within the cell.

The angles between layers in the z-direction were required to be 109.28° while the 'tripod' shaped tranches as viewed in the xy-plane had 60.00° angles.

3.1.1 Create approximate basis cell in ASE

The basis cell was prepared using ASE (Atomic Simulation Environment) [6], which is a Python package that can create an approximate bulk GaN wurtzite cell with the following command:

```
basis_cell = bulk('GaN',crystalstructure='wurtzite',a=3.186,orthorhombic=True)
write('GaN_basis_cell',format='conquest',fractional=True,atomic_order=['Ga','N'])
```

where a = 3.186$\mathring{A}$ is an approximate value for the x-direction cell length.

3.1.2 Optimise basis cell and export

Once the basis cell had been estimated, the cell's lattice parameters were optimised using the conjugate gradient method. Following this, the atomic positions were optimised to minimise the forces on the constituent atoms.

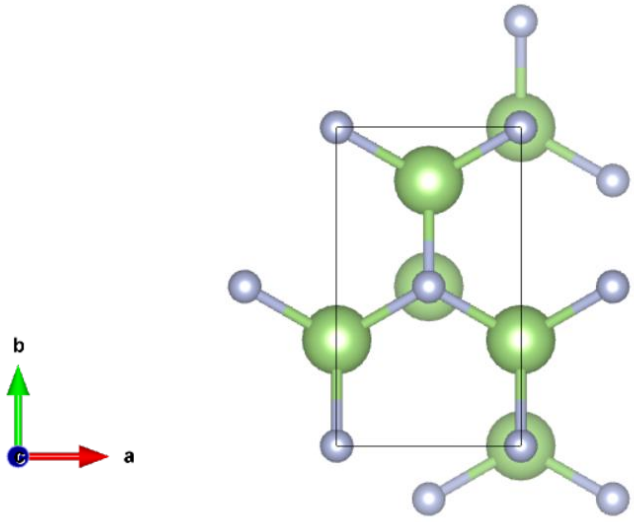

Figure 3: VESTA image of GaN Wurtzite basis cell (Ga - Green and N - Grey). The dimensions are as follows: a = 6.11 a_0, b = 10.60 a_0, c = 9.99 a_0.

Energy convergence tests on the basis cell found an appropriate grid cutoff and MP mesh grid of:

```
Grid.GridCutoff 350
Diag.MPMeshX      6
Diag.MPMeshY      4
Diag.MPMeshZ      4
```

These relatively computationally demanding parameters were effective due to the small number of atoms in the basis cell (8). Larger cells including the dislocations that were created would only be viable if computationally efficient parameters were used.

The grid cutoff is related to the defining energy spacing of the integration grid. The larger the energy spacing, the more accurate the numerical integral evaluation will be and hence, convergence is easier to reach when using a larger grid cutoff. The MP mesh grid is defined as the number of k-points in each direction (reciprocal space) to approximate integrals over the Brillouin zone [15]. Using a greater number of k-point samples will produce a more accurate result, hence convergence will be easily achieved.

3.2 5/7 Ring Dislocation Cells

The following method was adopted for the creation of the 5/7 ring dislocation [7]. The basic format of a dislocation was for the number of planes in the x-direction to decrease by one and then resolve itself hence creating two dislocation cores (5-4-5, 6-5-6 and 7-6-7). The periodic boundary conditions require an equal number of planes in the x-direction at both $y = 0$ and $y = b$ (the cell edge). The z-direction was chosen to be 1 cell thick to decrease computation time while the y-direction was 8 cells long and were distributed by 2-4-2 to allow equal spacing between the dislocation cores.

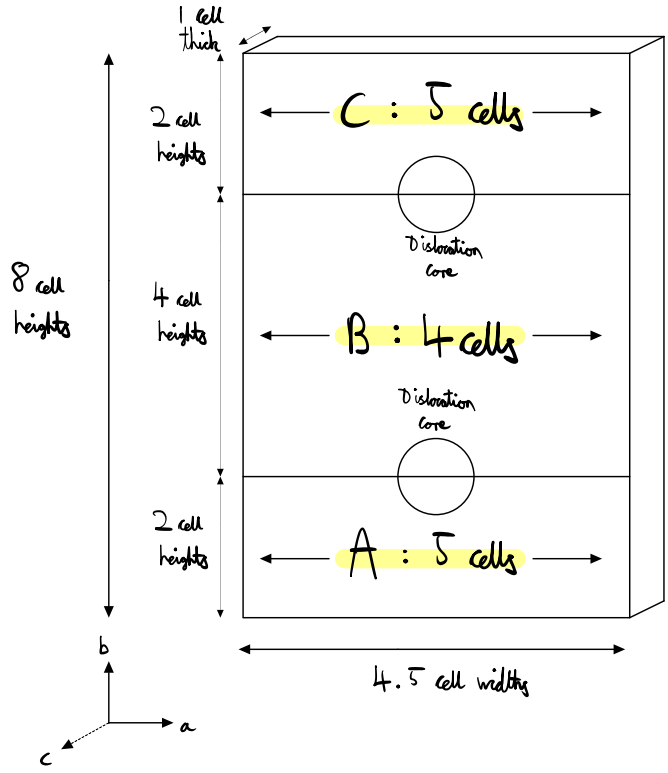

Figure 4: Illustration of 5-4-5 ABC dislocation format. Note part A is identical to part C.

The method for the 5-4-5 dislocation cell is as follows.

3.2.1 Create the three dislocation components

```
mulitplierA = [np.identity(3)[0]*5, np.identity(3)[1], np.identity(3)[2]]
partA = make_supercell(basis_cell, multiplierA)

mulitplierB = [np.identity(3)[0]*4, np.identity(3)[1], np.identity(3)[2]]
partB = make_supercell(basis_cell, multiplierB)

mulitplierC = [np.identity(3)[0]*5, np.identity(3)[1], np.identity(3)[2]]
partC = make_supercell(basis_cell, multiplierC)
```

3.2.2 Scale each component along the x-direction to the average width of both parts

```
scaleA = [[9/10,0,0],[0,1,0],[0,0,1]]
partA.set_cell(partA.get_cell()*scaleA*Bohr, scale_atoms=True)

scaleB = [[9/8,0,0],[0,1,0],[0,0,1]]
partB.set_cell(partB.get_cell()*scaleB*Bohr, scale_atoms=True)

scaleC = [[9/10,0,0],[0,1,0],[0,0,1]]
partC.set_cell(partC.get_cell()*scaleC*Bohr, scale_atoms=True)
```

The $\frac{9}{10}$ and $\frac{9}{8}$ factors scale all parts to have an x-width of 4.5 unit cells.

3.2.3 Combine the components

```
partAB = ase.build.stack(partA, partB, axis=1)
partABC = ase.build.stack(partAB, partC, axis=1)
```

Where 'axis=1' indicates that the components are to be stacked along the y-direction.

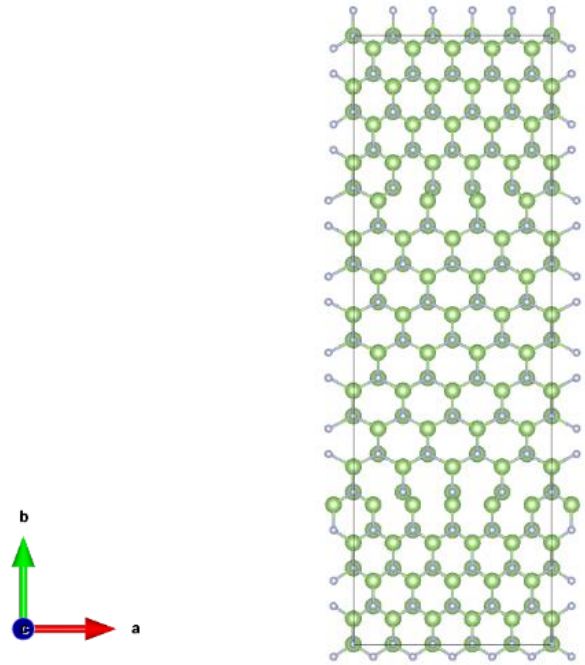

Figure 5: VESTA image of an unrelaxed 5/7 ring dislocation stage 3.

3.2.4 Move individual atoms to form first 5/7 ring

```
d_n_bond = 4.7*(1e-2)
d_ga_n_bond = 2.1*(1e-2)

d_n_7 = 5.4*(1e-2)
d_ga_7 = 4*(1e-2)
```

Where these small changes were estimated to match the appropriate form of the dislocation [7]. It is important to note that these particular values are fractional changes for a 5-4-5 dislocation and consequently they must be scaled for other dislocations e.g. for 6-5-6 the scale factor is $\frac{4.5}{5.5} = \frac{9}{11}$.

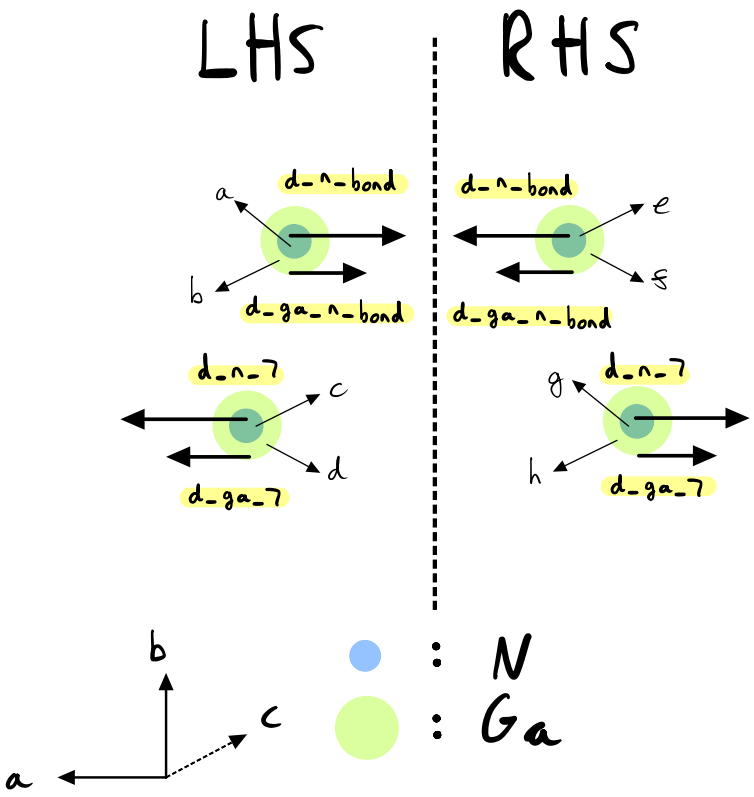

Figure 6: Illustration of required atom positions in the x-direction for a 5/7 ring dislocation.

The atoms that were required to have updated positions were identified and their x-positions were changed:

```
# LHS
# N-N Bond
atom_a_newx = partABC.get_scaled_positions()[a] - d_n_bond # N
atom_b_newx = partABC.get_scaled_positions()[b] - d_ga_n_bond # Ga
# 7 Ring
atom_c_newx = partABC.get_scaled_positions()[c] + d_n_7 # N
atom_d_newx = partABC.get_scaled_positions()[d] + d_ga_7 # Ga

#################################################################

# RHS
# N-N Bond
atom_e_newx = partABC.get_scaled_positions()[e] + d_n_bond # N
atom_f_newx = partABC.get_scaled_positions()[f] + d_ga_n_bond # Ga
# 7 Ring
atom_g_newx = partABC.get_scaled_positions()[g] - d_n_7 # N
atom_h_newx = partABC.get_scaled_positions()[h] - d_ga_7 # Ga
```

It is important to pay close attention to the direction of change for the LHS and RHS with respect to the x-coordinate direction.

```
new_pos_scaled = partABC.get_scaled_positions()

new_pos_scaled[a][0] = atom_a_newx
new_pos_scaled[b][0] = atom_b_newx
new_pos_scaled[c][0] = atom_c_newx
new_pos_scaled[d][0] = atom_d_newx
new_pos_scaled[e][0] = atom_e_newx
new_pos_scaled[f][0] = atom_f_newx
new_pos_scaled[g][0] = atom_g_newx
new_pos_scaled[h][0] = atom_h_newx

partABC.set_scaled_positions(new_pos_scaled)
```

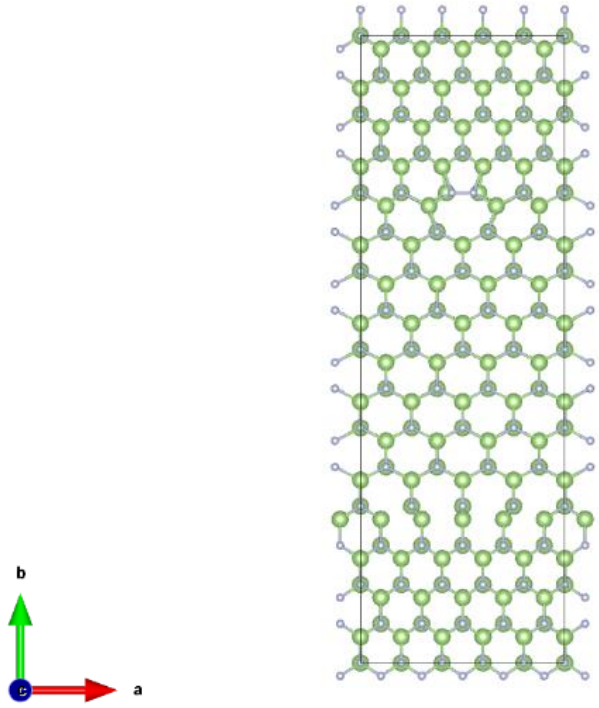

Figure 7: VESTA image of an unrelaxed 5/7 ring dislocation stage 4.

3.2.5 Delete atom planes

In order for the second dislocation (bottom) to possess the appropriate form, intermediate atom planes were deleted to match the 5/7 ring formation shown in the first dislocation core. The resulting cell after the appropriate atoms had been deleted can be visualised in Fig 7.

```
del_atoms = [...]
```

This is a list of atoms to be deleted in descending order. The order must be descending as the list updates after each redaction i.e. if 'Atom 5' is deleted, 'Atom 9' will now be 'Atom 8' while 'Atom 3' still has the same reference index.

```
iter = 0
while iter < len(del_atoms):
    del partABC[partABC.index for partABC in partABC
    if partABC.index == del_atoms[iter]]
    iter = iter + 1
```

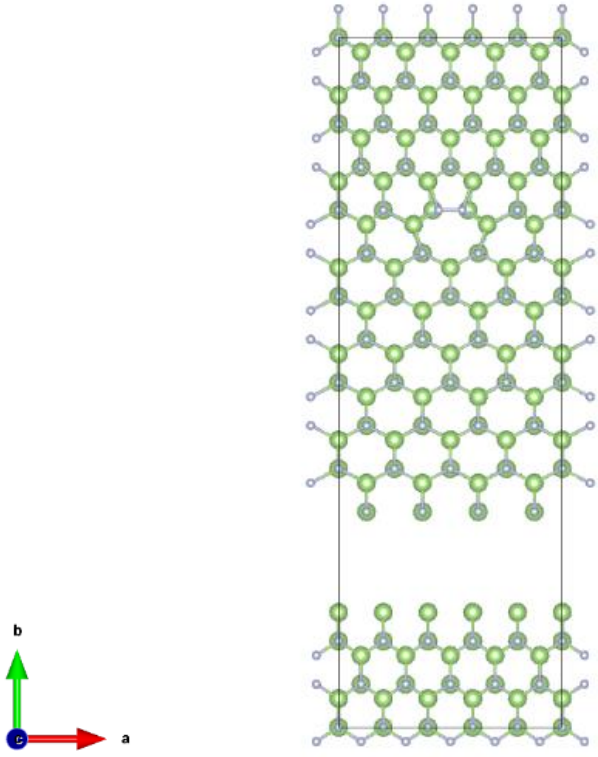

Figure 8: VESTA image of an unrelaxed 5/7 ring dislocation stage 5.

3.2.6 Shift atoms and repeat moving individual atoms

```
atoms_move = [...]
```

These are the atoms that are to be moved along the y-direction to fill in the missing volume from the deleted atoms (the bottom section in Fig 8).

```
pos = partABC.get_scaled_positions()

db = 0.12500027

iter = 0
while iter < len(atoms_move)
    newy = partABC.get_scaled_positions()[atoms_move[iter]][1] + db
    pos[atoms_move[iter]][1] = newy
    iter = iter + 1
```

Where the value *db* was estimated so the bond lengths on the parts' boundaries matched the appropriate bond lengths between other constituent atoms. The sign of *db* must also be carefully considered depending on the desired direction of the movement. The atomic movements around the new dislocation detailed in stage 4 must be again repeated to form the second 5/7 ring.

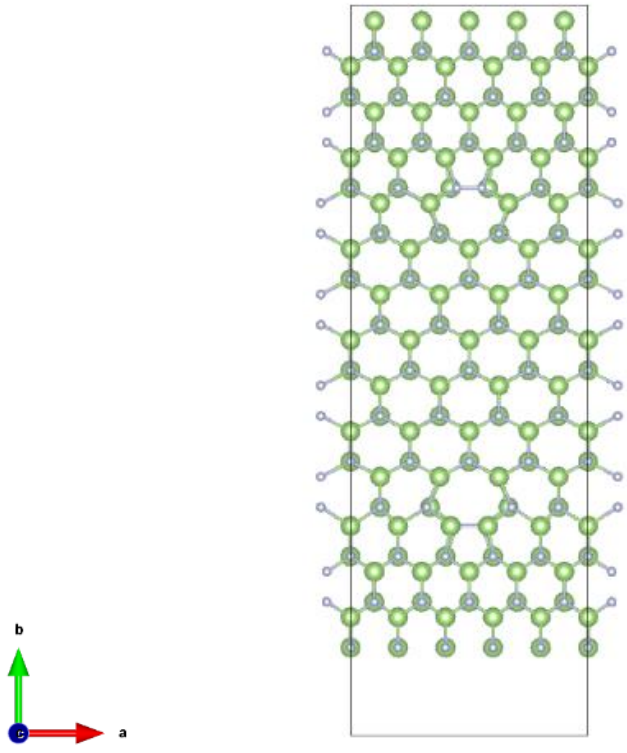

Figure 9: VESTA image of an unrelaxed 5/7 ring dislocation stage 6.

3.2.7 Re-scale cell length in y-direction to remove empty space

```
a = partABC.get_cell()[0][0]
b = 39.477445
c = partABC.get_cell()[2][2]
partABC.set_cell([[a,0,0],[0,b,0],[0,0,c]],scale_atoms=False)
```

Where the atoms' fractional positions must not be scaled in order to remove the empty space in the cell. The value of b is again estimated to match the bond lengths of boundary atoms with other constituent atoms which are not near dislocation cores.

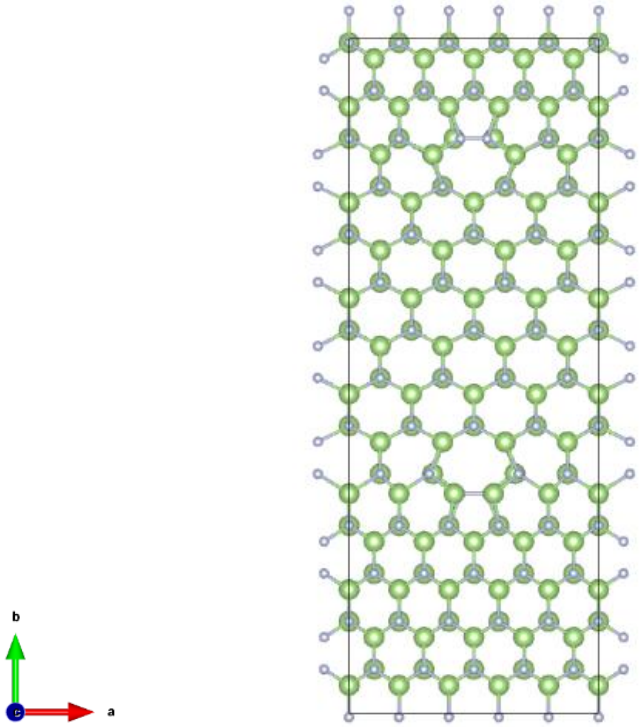

Figure 10: VESTA image of an unrelaxed 5/7 ring dislocation stage 7.

3.2.8 Optimise lattice constants and atomic positions

The cell parameters are optimised using the conjugate gradient routine and then the atomic positions are optimised. This process is repeated 3 or 4 times before finally a dual optimisation of cell parameters and atomic positions is performed leaving a fully structurally relaxed GaN dislocation cell.

The cell dimensions were optimised until the pressure and maximum force in each direction were <0.1 GPa and <0.001 $\frac{Ha}{a_0}$ respectively. An example of the notation that has been used for the different dislocations is Dis5-4-5 (8), where 5-4-5 denotes the number of cells across the x-direction in part A, B and C, while (8) represents the number of cells in the y-direction (before the atom planes were removed).

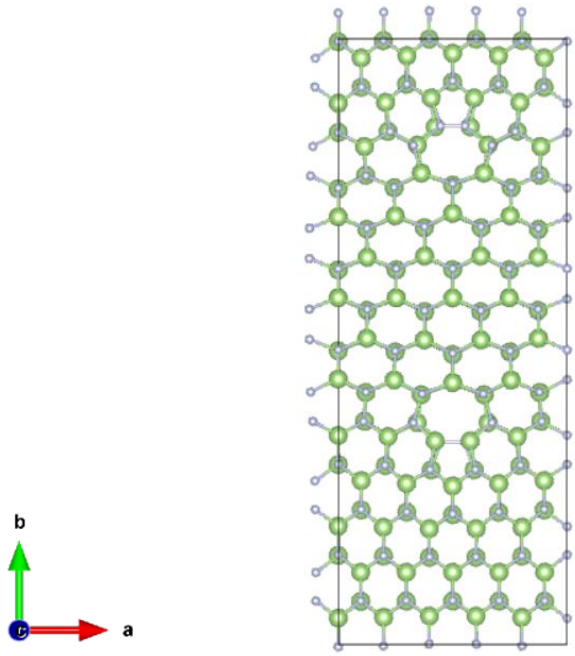

Figure 11: VESTA image of fully optimised 5/7 ring dislocation (Dis5-4-5 (8)).

4 Results

The following subsections outline the results for the three dislocation cores that were modelled. In each instance, the following findings are presented: the total energies of each possible In atom location on the $z = 0.5$ (fractional) plane, the forces of all atoms at said plane for the lowest, 2nd lowest and 3rd lowest energy sites, and the forces on all atoms for the highest energy site.

It is important to note that the DFT energy plots (Fig 12, 15 and 18) were calculated by replacing a single Ga atom with an In atom at each location on the plane and recording the DFT energy from a static test at all the possible sites. The force plots were subsequently generated using the lowest three energy sites and the highest energy site (which was established from the energy plot). The force plots' titles indicate the reference for the In atom replacement site. As well as this, the N atoms do not lie in the same plane as the Ga atoms, and infact have a z-position of ~ 0.625. Therefore, the force plots (Fig 13, 14, 16, 17, 19 and 20) show a projection of both z-planes (Ga and N) on the xy-plane.

4.1 Dis5-4-5 (8)

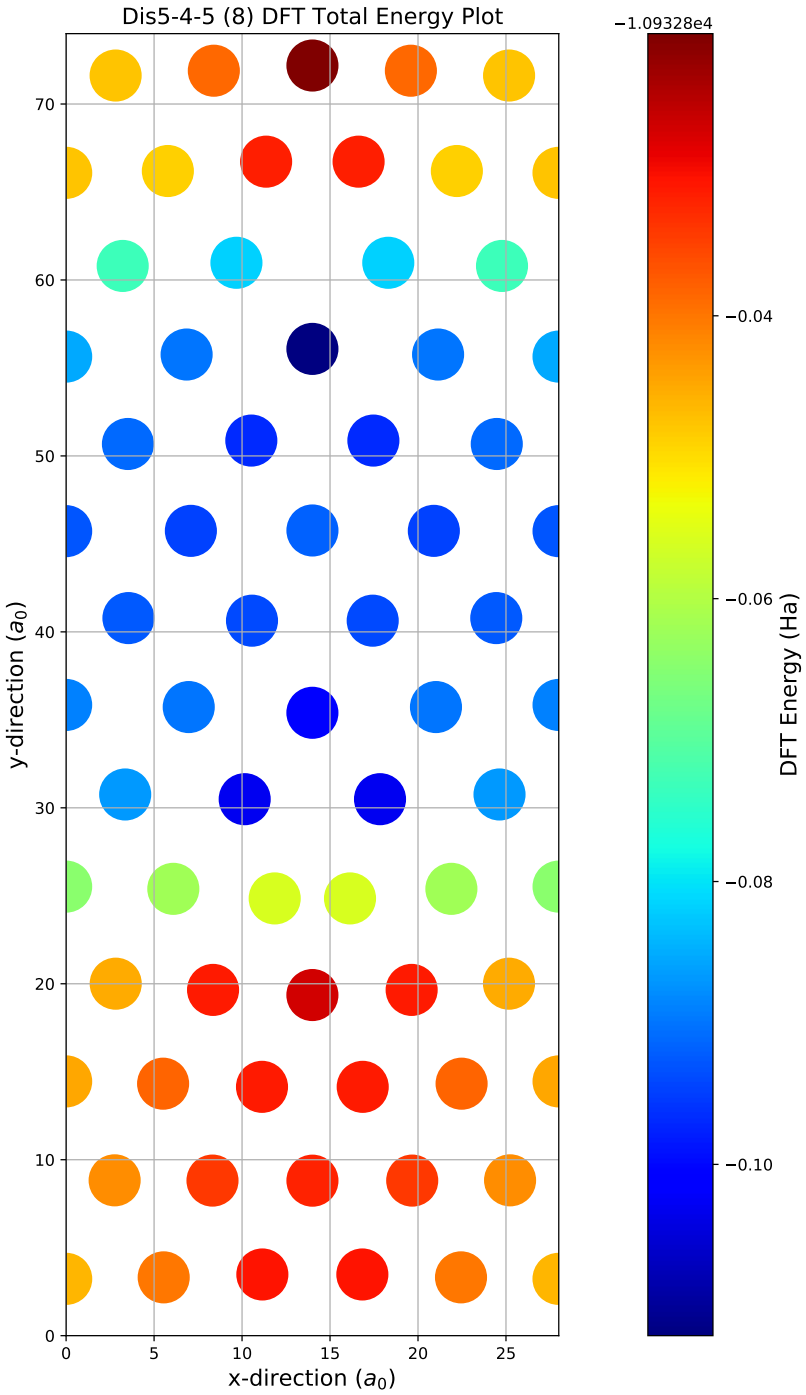

Figure 12: Dis5-4-5 (8) DFT total energy plot for In atoms at different Ga atom positions in a 5/7 ring dislocation.

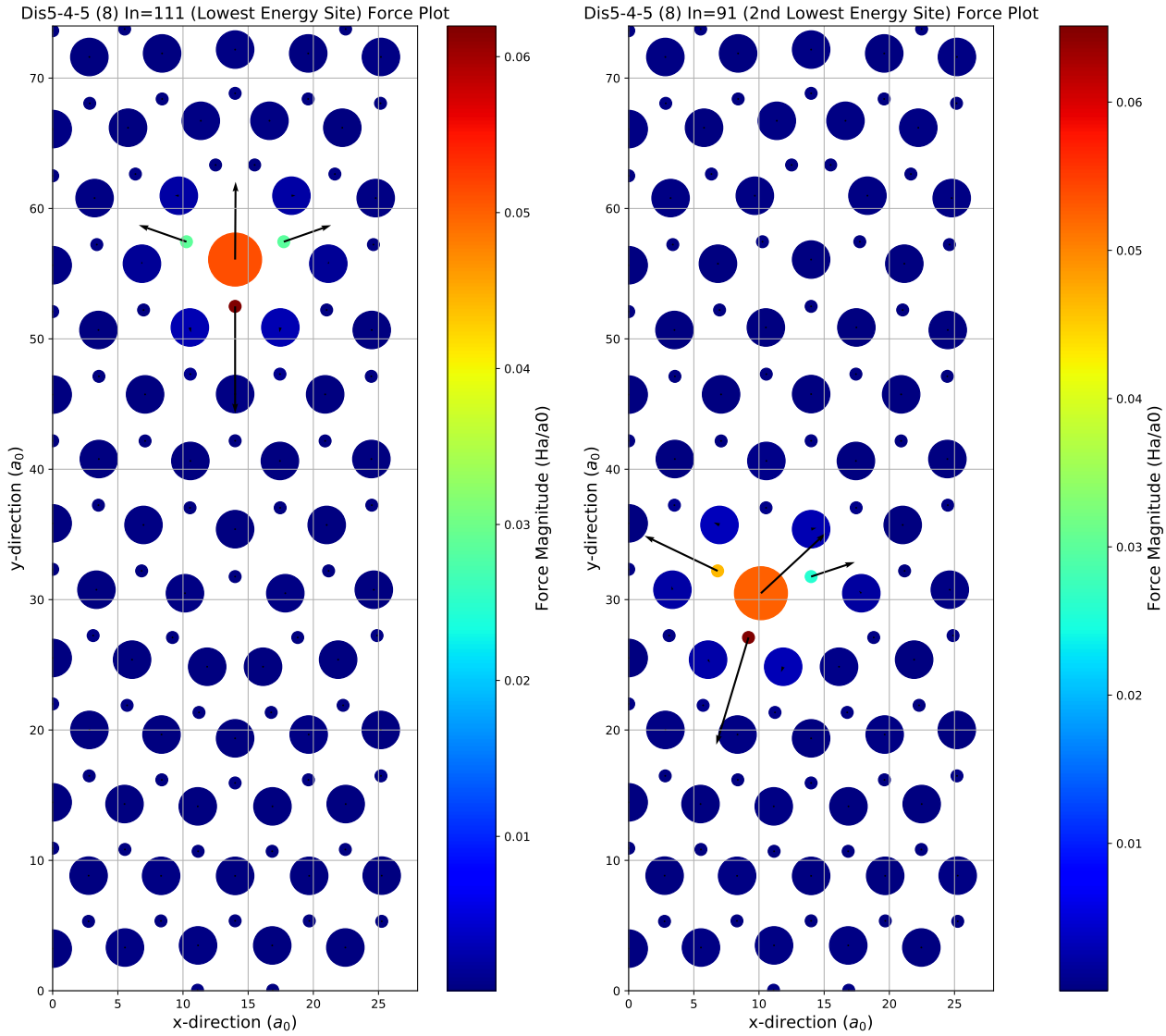

Figure 13: Forces of the lowest two energy sites in Dis5-4-5 (8) projected onto the xy-plane. The three data point sizes correspond to: N - smallest, Ga - medium and In - largest.

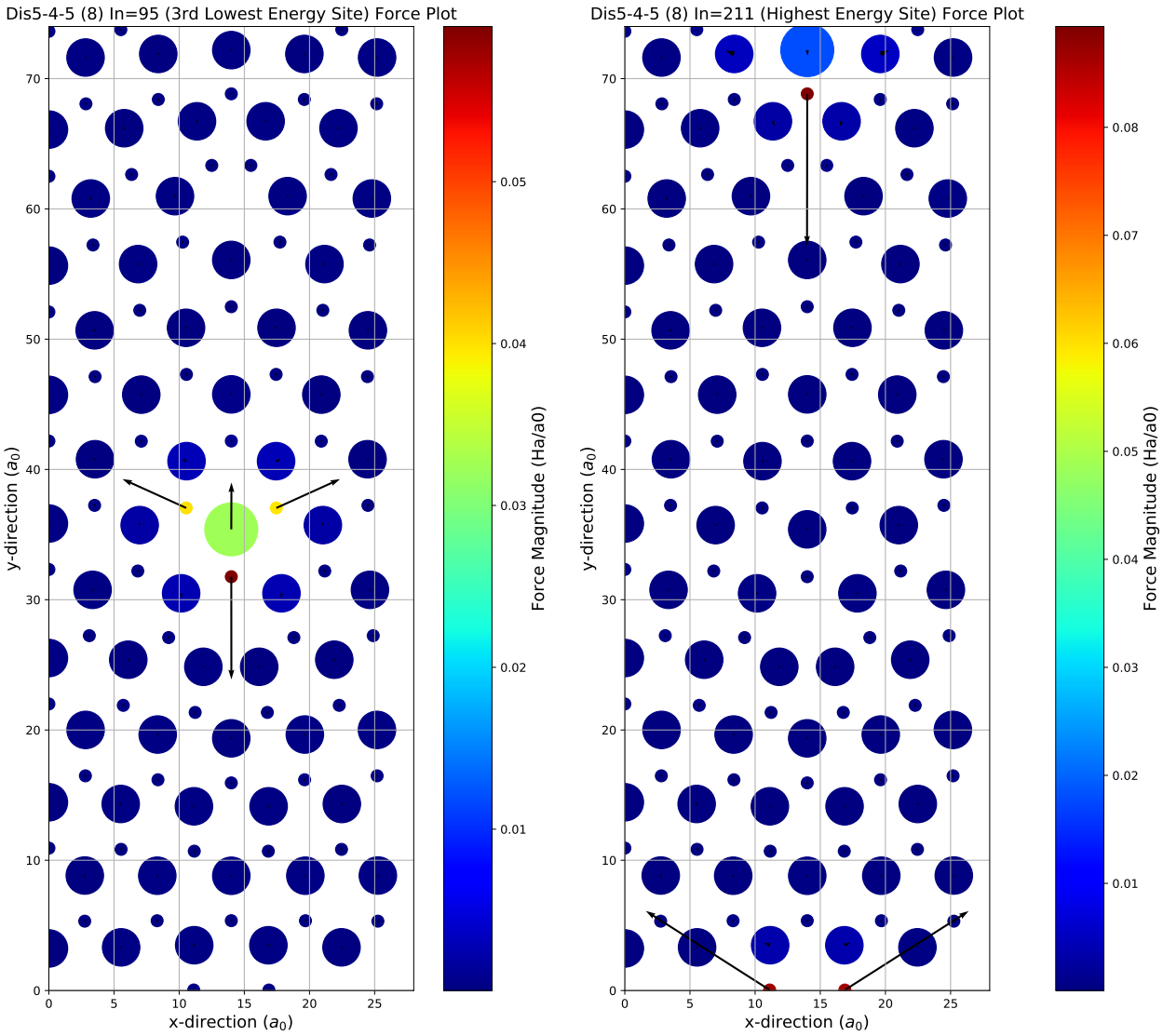

Figure 14: Forces of the third lowest and highest energy sites in Dis5-4-5 (8) projected onto the xy-plane. The three data point sizes correspond to: N - smallest, Ga - medium and In - largest.

4.2 Dis6-5-6 (8)

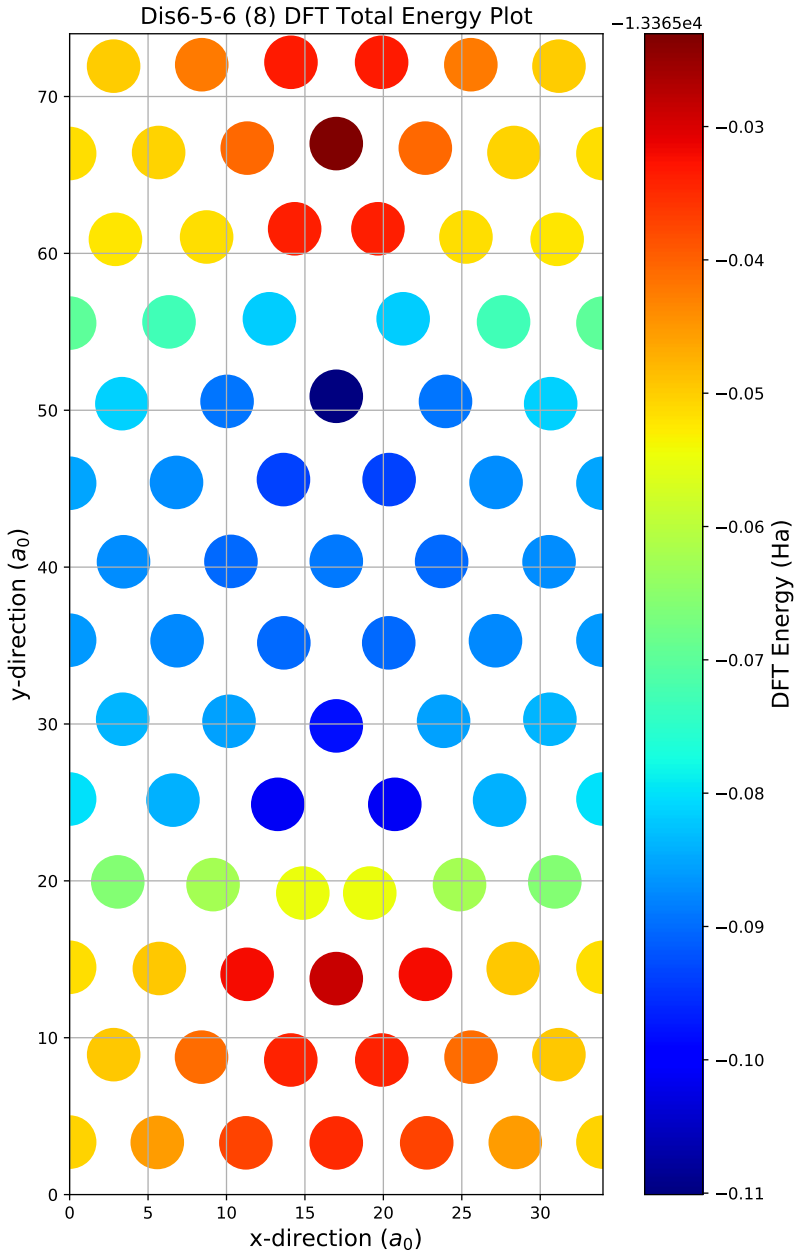

Figure 15: Dis6-5-6 (8) DFT total energy plot for In atoms at different Ga atom positions in a 5/7 ring dislocation.

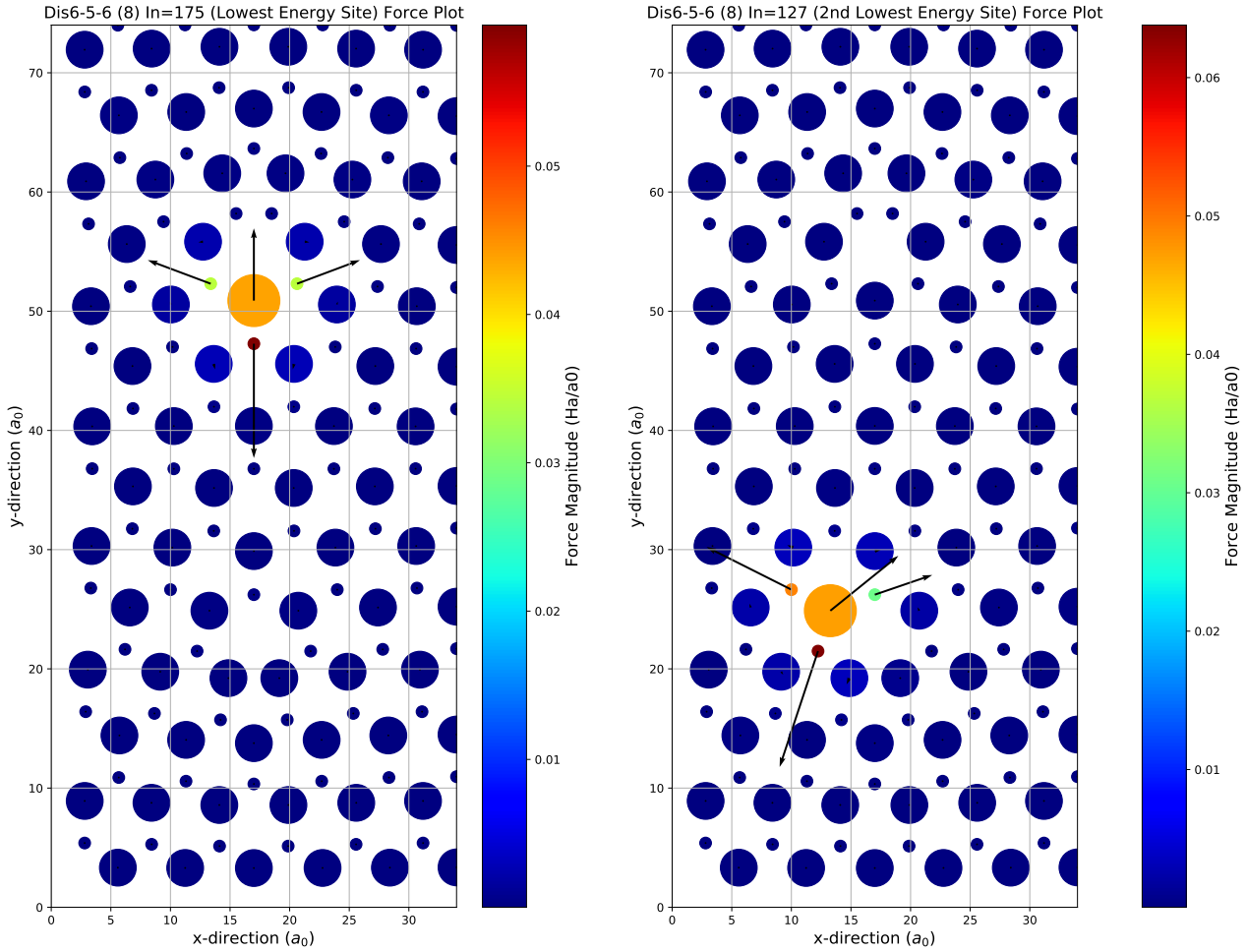

Figure 16: Forces of the lowest two energy sites in Dis6-5-6 (8) projected onto the xy-plane. The three data point sizes correspond to: N - smallest, Ga - medium and In - largest.

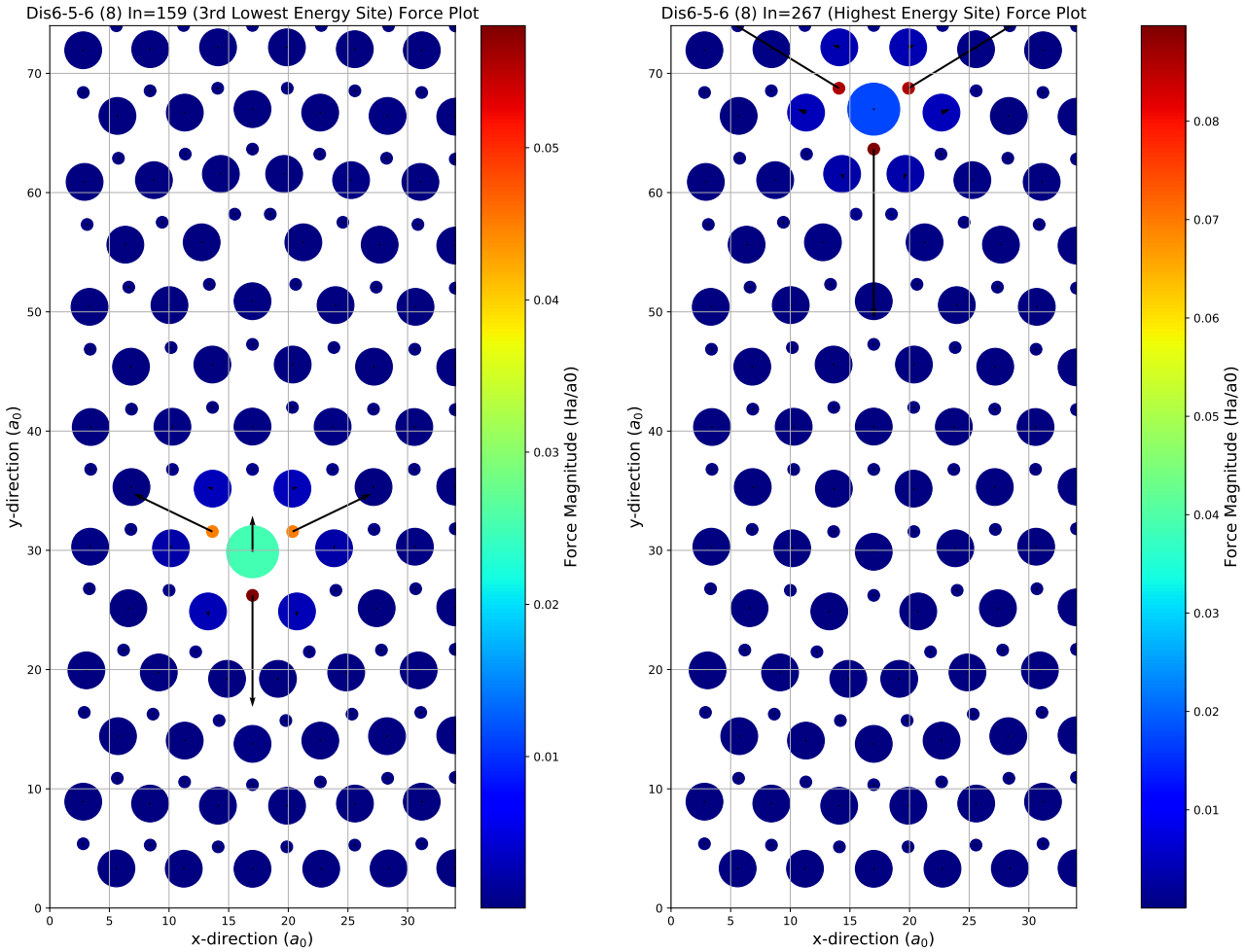

Figure 17: Forces of the third lowest and highest energy sites in Dis6-5-6 (8) projected onto the xy-plane. The three data point sizes correspond to: N - smallest, Ga - medium and In - largest.

4.3 Dis7-6-7 (8)

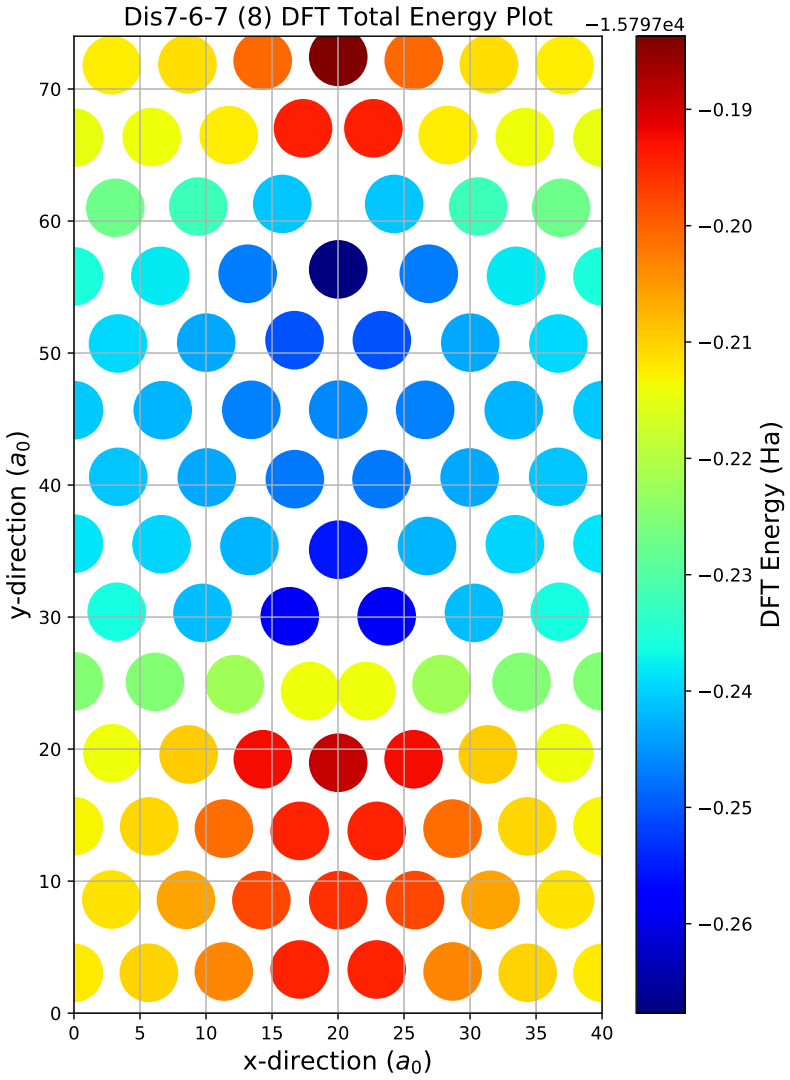

Figure 18: Dis7-6-7 (8) DFT total energy plot for In atoms at different Ga atom positions in a 5/7 ring dislocation.

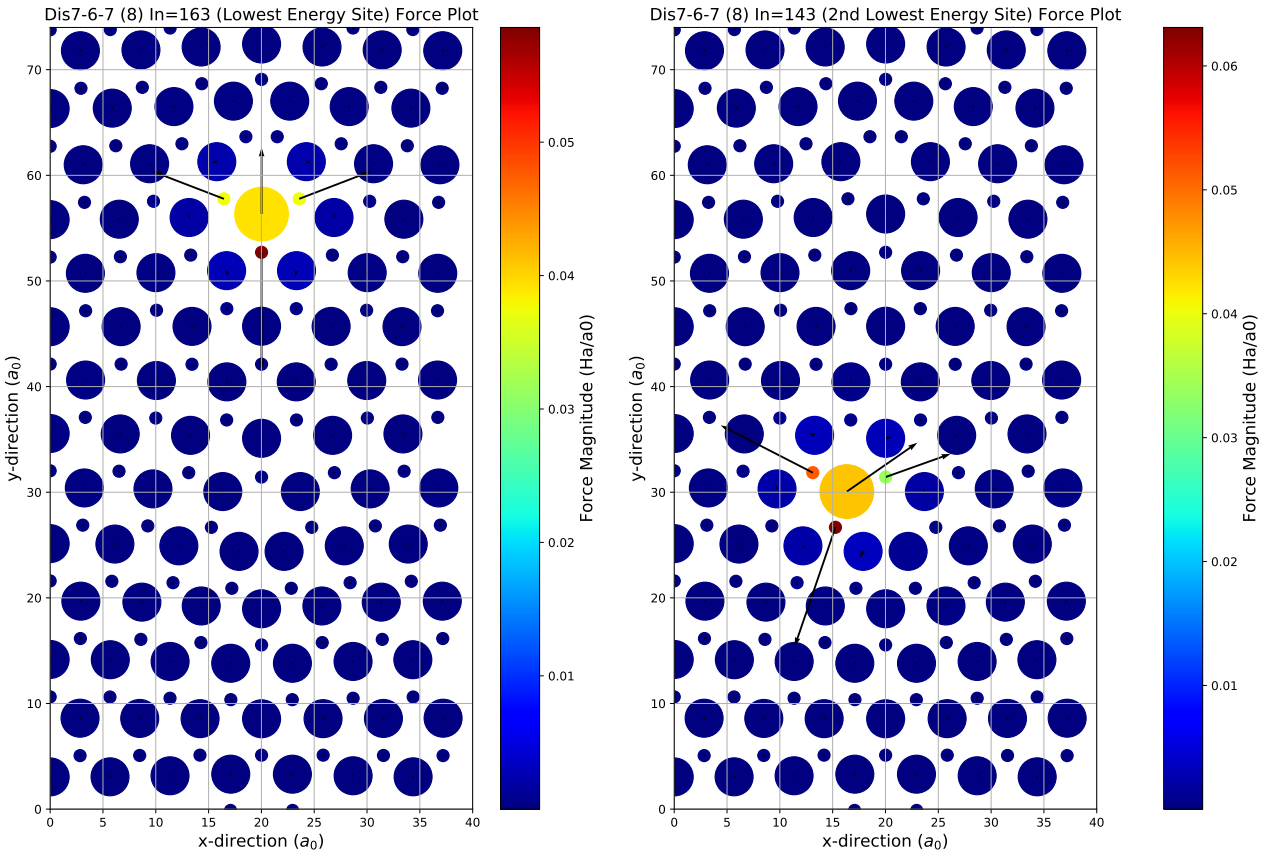

Figure 19: Forces of the lowest two energy sites in Dis7-6-7 (8) projected onto the xy-plane. The three data point sizes correspond to: N - smallest, Ga - medium and In - largest.

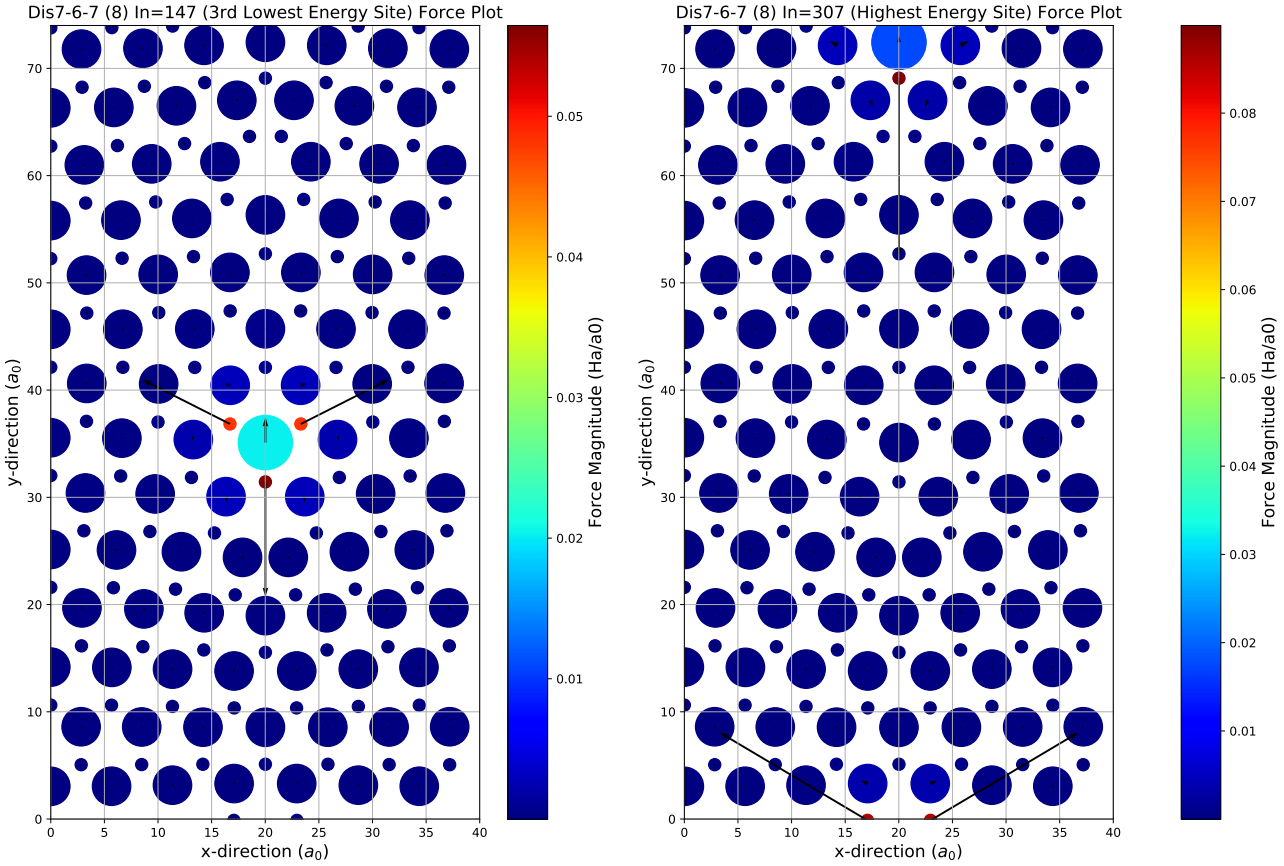

Figure 20: Forces of the third lowest and highest energy sites in Dis7-6-7 (8) projected onto the xy-plane. The three data point sizes correspond to: N - smallest, Ga - medium and In - largest.

5 Discussion

As Fig 12, 15 and 18 collectively suggest, the system's minimum DFT energy is achieved when the In atom is located near the dislocation cores. Specifically, the most energetically favourable In atom position is on the 7-ring at the point which is furthest from the N-N bond as shown in Fig 21.

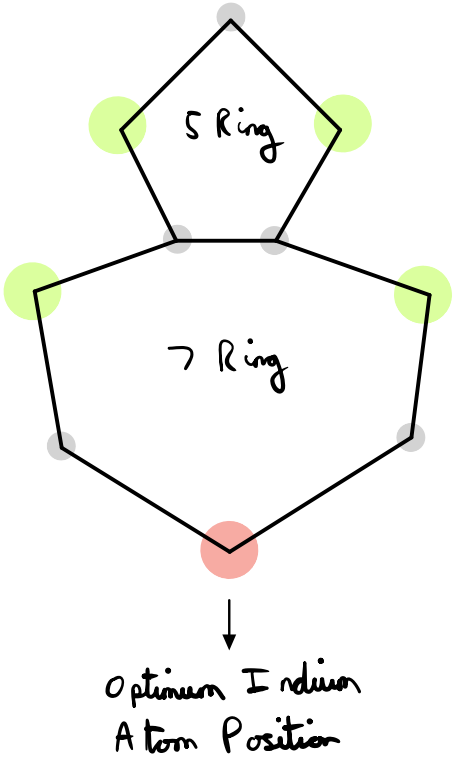

Figure 21: Illustration of optimum In atom location at a 5/7 ring (Ga - Green, N - Grey and In - Red).

The relative positions of the three lowest energy sites and the highest energy site remain the same across the three models which indicates some level of repeatability in the results of the 5/7 ring dislocation cores (Fig 13, 14, 16, 17, 19 and 20). The largest forces of the sites of interest tend to act on the substituted In atom and the three surrounding N atoms. This was expected as all the cells had been fully relaxed before the introduction of the In atom. As well as this, the DFT energy plots (Fig 12, 15 and 18) possess a resemblance in their energy variation.

Due to the periodic boundary conditions imposed on the system, dislocation cores will interact because of their proximity. In order to reduce the effects of this, the system could be extended in the y-direction as well as the x-direction. Alternative dislocation models which could be explored include: 8-7-8 (8), 5-4-5 (12) and 5-4-5 (16). Due to the increased number of atoms, these models would incur greater computation times. Despite this, a more clear energy landscape could be modelled with less horizontal and vertical interference from neighbouring dislocation cores.

References

[1] Antoine Bere and Anna Serra. Atomic structure of dislocation cores in gan. *Physical Review B*, 65:205323–, 05 2002.

[2] M J Gillan D R Bowler, T Miyazaki. Recent progress in linear scaling ab initio electronic structure techniques. J. Phys.: Condens. Matter 14 2781 2002.

[3] R Choudhury T Miyazaki, D R Bowler and M J Gillan. Atomic force algorithms in density functional theory electronic-structure techniques based on local orbitals. J. Chem. Phys. 121, 6186 2006.

[4] M J Gillan T Miyazaki D R Bowler, R Choudhury. Recent progress with large-scale ab initio calculations: the conquest code. J. Chem. Phys. 243, 989 2006.

[5] A S Torralba et al. Pseudo-atomic orbitals as basis sets for the o(n) dft code conquest. J. Phys.: Condens. Matter 20 294206 2008.

[6] Ask Hjorth Larsen et al 2017 J. Phys.: Condens. Matter 29 273002. The atomic simulation environment - a python library for working with atoms. June 2017.

[7] J.Chen G.Nouet I.Belabbas, P.Ruterana. The atomic and electronic structure of dislocations in ga-based nitride semiconductors. *Philosophical Magazine*, 86(15):2241–2269, 2006.

[8] Prof. Ursula Röthlisberger. Introduction to electronic structure methods, 2019.

[9] GaN Systems. Gallium nitride: The fundamental building block for power electronics. https://gansystems.com/gallium-nitride-semiconductor/#:~:text=Since%20the%201990s%2C%20it%20has,components%2C%20lasers%2C%20and%20photonics.

[10] AZO Optics. Indium gallium nitride (ingan) lasers – properties and applications. https://www.azooptics.com/Article.aspx?ArticleID=517#:~:text=InGaN-,Applications,LEDs, 2013.

[11] V.Brázdová D.R.Bowler. *Atomistic Computer Simulations - A Practical Guide*. 2013.

[12] P.Briddon. Ggas - a history. 2014.

[13] Matthias Ernzerehof John P.Perdew, Kieron Burke. *Generalized Gradient Approximation Made Simple*. Phys. Rev. Lett. 78, 1396 (1996).

[14] N.M.Harrison. *An Introduction to Density Functional Theory*. 2001.

[15] D.R.Bowler. Conquest: Read the docs. https://readthedocs.org/projects/conquest/.

[16] Lianheng Tong. Kerker preconditioning and wave-dependent metric. http://www.hector.ac.uk/cse/distributedcse/reports/conquest/conquest/node3.html, 2011.

[17] F.Izumi K.Momma. Vesta 3 for three-dimensional visualization of crystal, volumetric and morphology data. J.Appl.Cryst 44, 1272-1276, 2011.

Appendices

A File Location Details

The directory I have made containing my dislocation files is titled 'ForDavid' and it can be found on Midgard at '/homes/hyouel/'. Within this directory you will find the following sub-directories: dis5-4-5 (8), dis5-4-5 (12), dis5-4-5 (16), dis6-5-6 (8), dis7-6-7 (8) and dis8-7-8 (8). All these sub-directories contain their relevant coordinate file and input file.

The only file with any sort of optimisation are dis5-4-5 (8), dis6-5-6 (8) and dis7-6-7 (8) where you will find ~ 4 lattice parameter optimisations, ~ 4 atomic position optimisations and in one case a combined optimisation. The order these were conducted in is: latopt1, sqnm1, latopt2, sqnm2, latopt3, sqnm3 and combined. The directory '/dis5-4-5 (8) coords' documents the coordinate file after each one of these steps, starting with the un-optimised file: 'dis5-4-5 (8).coords'.

The file:

`in_dope_static`

contains files for each Ga atom which was replaced by an In atom e.g.

`103_in`

These results were used to create Fig 12, 15 and 18. I will also attach a supplementary file containing .cell files, coordinate files and the files used for both the energy and force plots to the email.

Please let me know if this is not clear or if anything is missing.

B Optimised GaN Basis Cell Coordinates

The full coordinate file reads (dimensions are in a_0 and atomic positions are fractional):

```
6.11879702829883065  0.00000000000000000 0.00000000000000000
0.00000000000000000 10.59680695883881718 0.00000000000000000
0.00000000000000000  0.00000000000000000 9.98510003070653696

8
0.99996825053545624  0.00000945659730831 0.00037632690819069 1 T  T  T
0.00000102706870009  0.33334558479177417 0.12447249408935390 2 T  T  T
0.99999903039348614  0.33323691324333116 0.50042624152000470 1 T  T  T
0.99998847932436785  0.00002818019279056 0.62447625477253177 2 T  T  T
0.49997054112306677  0.50000945667249919 0.00037632097296355 1 T  T  T
0.50000098033962115  0.83334558466431818 0.12447249430820959 2 T  T  T
0.49999902725008938  0.83323691206125350 0.50042624127874136 1 T  T  T
0.49998722442679949  0.50002818019126016 0.62447625662273221 2 T  T  T
```

C Static Test Input File Example

The input file for the static tests reads:

```
AtomMove.TypeOfRun               static

Diag.GammaCentred                T

SC.KerkerPreCondition            T

minE.SCTolerance                 1E-7
General.NumberOfSpecies          3

IO.Coordinates                   dis5-4-5_8_opt.coords

Grid.GridCutoff                  100

Diag.MPMesh                      T
Diag.MPMeshX                     2
Diag.MPMeshY                     1
Diag.MPMeshZ                     4

%block ChemicalSpeciesLabel
 1   69.720  Ga_DZP
 2   14.010  N_DZP
 3   114.820  In_DZP
%endblock ChemicalSpeciesLabel
```

The grid cutoff, mesh values and self-consistency tolerance were chosen to reduce computation time while preserving a small difference between the Harris-Foulkes energy and DFT total energy of $\sim 5 \times 10^{-6}$ Ha.

D Surface Fitting Python Code

The following function is an attempt to fit a polynomial plane to the Dis7-6-7 (8) energy data points using the pseudo-inverse.

```
def best_fit_poly(x,y,E,poly):

    """
    Fits an polynomial surface to 3D data points with powers up to and
    including the degree 'poly' using the pseudo-inverse matrix.
    This function is specifically written to fit Dis7-6-7_8
    but can be adapted to fit other dislocation cores.

    Inputs:
    x - horizontal data positions
    y - vertical data positions
    E - energy values at each data point
    poly - desired degree of fit

    Outputs:
    X & Y - meshgrid of new interpolated data points
    vals - interpolation 'E' values

    """

    b = E

    tri_nums = [1, 3, 6, 10, 15, 21, 28, 36, 45, 55, 78, 91, 105, 120, 136, 153, 171,
    190, 210, 231, 253, 276, 300, 325, 351]

    order_x = [(1,0),(2,1,0),(3,2,1,0),(4,3,2,1,0),(5,4,3,2,1,0),(6,5,4,3,2,1,0)
    ,(7,6,5,4,3,2,1,0),(8,7,6,5,4,3,2,1,0)]
    order_y = [(0,1),(0,1,2),(0,1,2,3),(0,1,2,3,4),(0,1,2,3,4,5),(0,1,2,3,4,5,6)
    ,(0,1,2,3,4,5,6,7),(0,1,2,3,4,5,6,7,8)]

    order_x = order_x[:poly]
    order_y = order_y[:poly]

    tri_val = tri_nums[poly]

    A = np.zeros((len(x),tri_val))

    counter = 1
    i = 0 # Data point
    j = 0 # Which order
    k = 0 # Value within order_x and order_y
```

```
while i < len(x):
    A[i][0] = 1
    while j < len(order_x):
        k = 0
        while k < len(order_x[j]):
            A[i][counter] = x[i]**order_x[j][k] * y[i]**order_y[j][k]
            counter += 1
            k += 1
        j += 1

    counter = 1
    j = 0
    i += 1

beta = np.linalg.pinv(A) @ b

x_coords = np.linspace(0,40,401)
y_coords = np.linspace(0,74,741)

X, Y = np.meshgrid(x_coords,y_coords)

num_of_points = 401*741
shape = num_of_points
num_of_rows = np.shape(X)[0]
num_of_columns = np.shape(X)[1]

vals = np.zeros(num_of_points)

new_vals = np.zeros(len(beta))

X = np.reshape(X,shape)
Y = np.reshape(Y,shape)

point = 0
degree = 0
j = 0
k = 0
counter = 1
while point < num_of_points:
    con = beta[0]
    j = 0
    counter = 1
    new_vals = np.zeros(len(beta))
    while j < len(order_x):
        k = 0
        while k < len(order_x[j]):
            new_vals[counter] = beta[counter] * X[point]**order_x[j][k]
```

```
                * Y[point]**order_y[j][k]
                counter += 1
                k += 1
            j += 1
        vals[point] = np.sum(new_vals) + con
        point += 1

vals = np.reshape(vals,(741,401))

return X,Y,vals
```

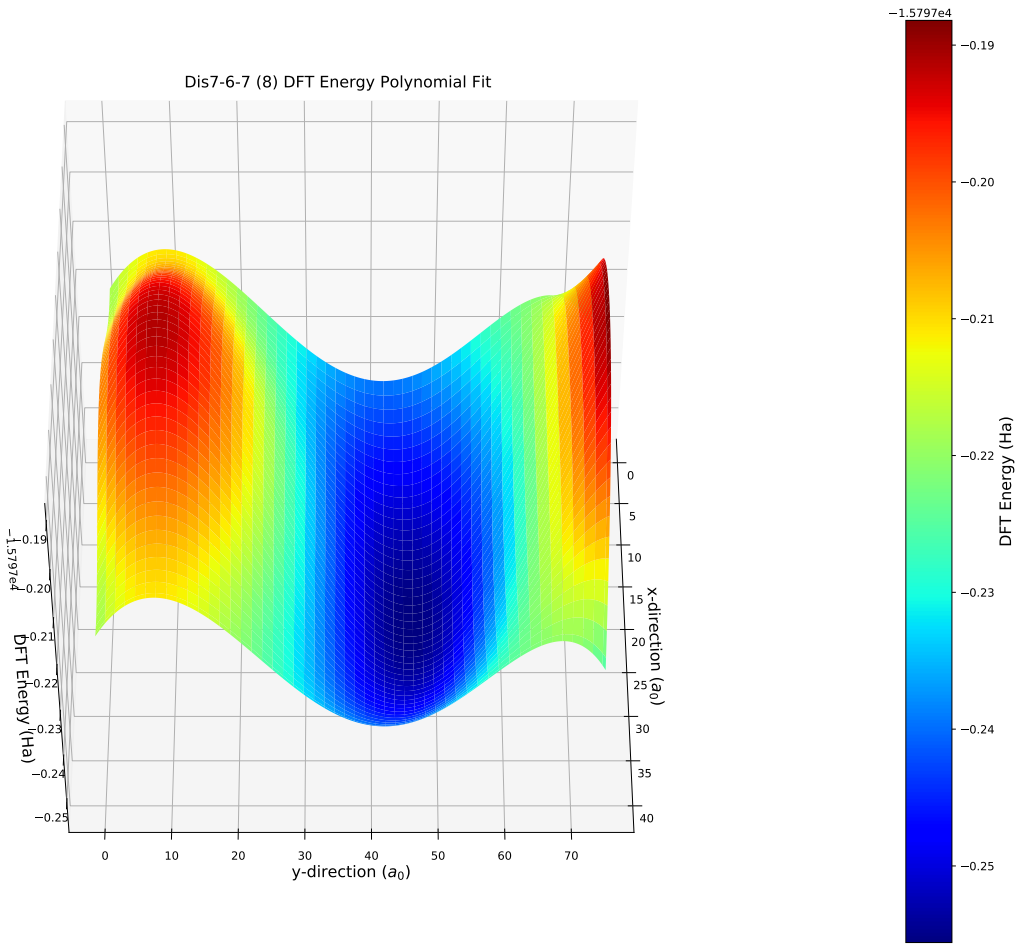

Figure 22: Dis7-6-7 (8) DFT total energy polynomial fit (5th degree).