Final Exam: DFT and Machine Learning
System: C-FCC

Noguera, V .; Maisincho, B .; Morales, D .; Palma, A . & Salazar, A .
School of Physical Sciences and Nanotechnology, Yachay Tech University, Urcuquí - Ecuador
School of Chemical Sciences and Engineering, Yachay Tech University, Urcuquí - Ecuador

---

DFT

ECUT energy

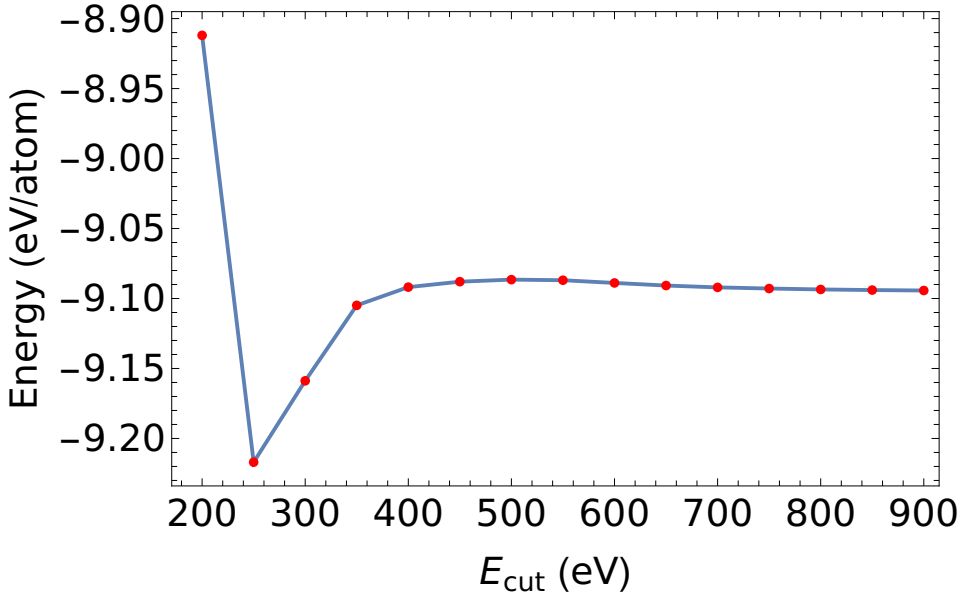
In[•]:=
```
wrk =
  "/home/alanp/wrk/dft_course_hands-on/final_exam/scripts_inputs";
SetDirectory[wrk];
```

In[•]:=
```
data1 = Import["ecut.dat"] // Map[#/{1, 2} &, #] &
```

Out[•]=
```
{{200, -8.91198}, {250, -9.21697}, {300, -9.15883}, {350, -9.1049}, {400, -9.09188},
 {450, -9.08798}, {500, -9.08651}, {550, -9.08694}, {600, -9.08889}, {650, -9.09074},
 {700, -9.09206}, {750, -9.09288}, {800, -9.09353}, {850, -9.09398}, {900, -9.09428}}
```

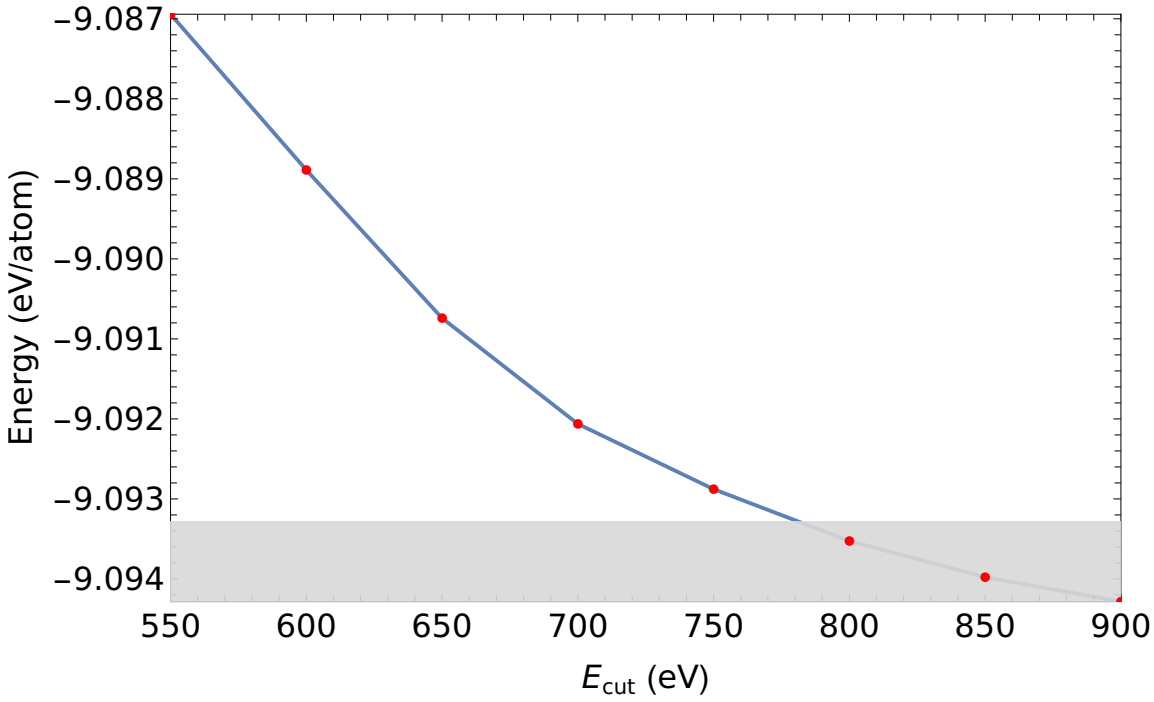
In[•]:=
```
data1 // ListPlot[#, Frame → True,
  FrameLabel → {"\!\(\*SubscriptBox[\(E\), \(cut\)]\) (eV)", "Energy (eV/atom)"},
  BaseStyle → {FontFamily → "Helvetica", FontSize → 19}, Joined → True, PlotRange → All,
  InterpolationOrder → 1, ImageSize → 500, Epilog → {Red, AbsolutePointSize[5], Point[#]}] &
```

Out[•]=



In[•]:=
```
data1 // ListPlot[#, Frame → True,
  FrameLabel → {"\!\(\*SubscriptBox[\(E\), \(cut\)]\) (eV)", "Energy (eV/atom)"},
  BaseStyle → {FontFamily → "Helvetica", FontSize → 16},
  PlotRange → {{#[[8, 1]], #[[-1, 1]]}, {#[[-1, 2]], #[[8, 2]]}}, Joined → True,
  PlotRange → All, InterpolationOrder → 1, Epilog →
   {{LightGray, Opacity[0.9], Rectangle[{#[[4, 1]], #[[-1, 2]]}, {#[[-1, 1]], #[[-1, 2]] + 10^-3}]},
    {Red, AbsolutePointSize[5], Point[#]}},
  ImageSize → 600, Epilog → {Red, AbsolutePointSize[5], Point[#]}] &
```

Out[•]=



THe ECUT energy to be used is 800.

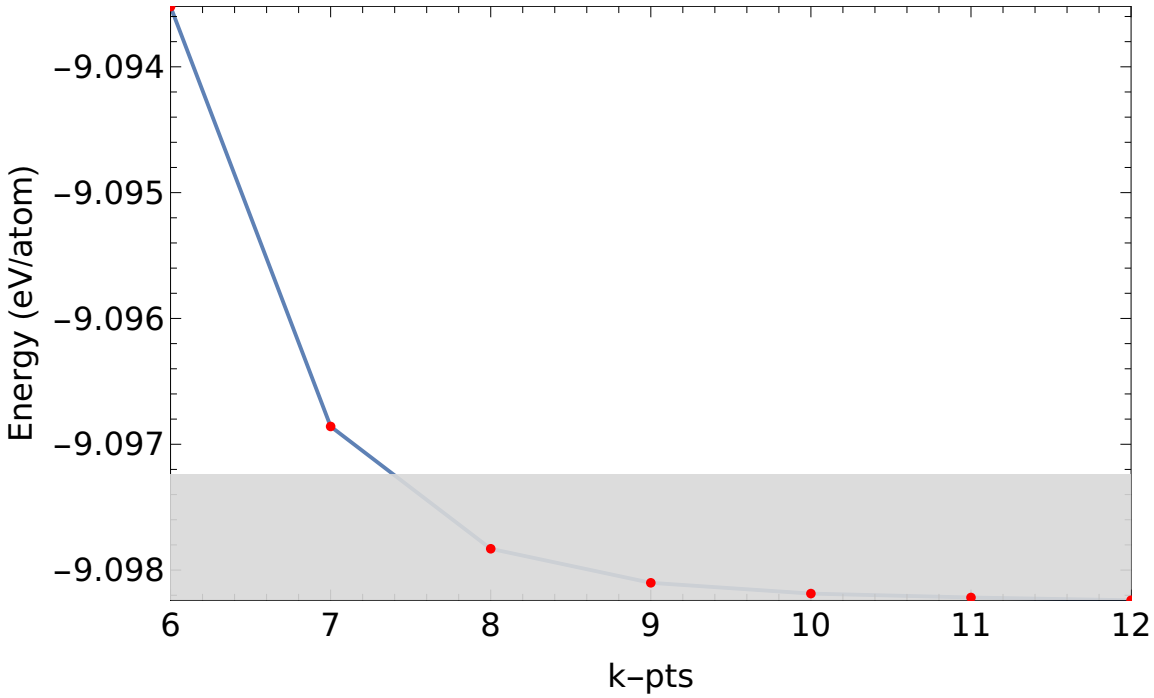
Analysis of k-points grids

In[•]:=
```
datakp = Import["kps.dat"] // Map[# / {1, 2} &, #] &
```

Out[•]=
```
{{4, -9.0314}, {5, -9.08104}, {6, -9.09352}, {7, -9.09686},
 {8, -9.09783}, {9, -9.0981}, {10, -9.09819}, {11, -9.09822}, {12, -9.09824}}
```

In[•]:=
```
datakp // ListPlot[#, Frame → True, FrameLabel → {"k-pts", "Energy (eV/atom)"},
  BaseStyle → {FontFamily → "Helvetica", FontSize → 16},
  PlotRange → {{#[[3, 1]], #[[-1, 1]]}, {Min[#[[All, 2]]], #[[3, 2]]}},
  Joined → True, PlotRange → All, InterpolationOrder → 1, Epilog →
   {{LightGray, Opacity[0.9], Rectangle[{#[[1, 1]], #[[-1, 2]]}, {#[[-1, 1]], #[[-1, 2]] + 10^-3}]},
    {Red, AbsolutePointSize[5], Point[#]}},
  ImageSize → 600, Epilog → {Red, AbsolutePointSize[5], Point[#]}] &
```

Out[•]=



Then for this system a kpts grid of 8×8×8will be used.

The equation of state (EOS)

The cross product is × is esc cross esc and dot prodct is .

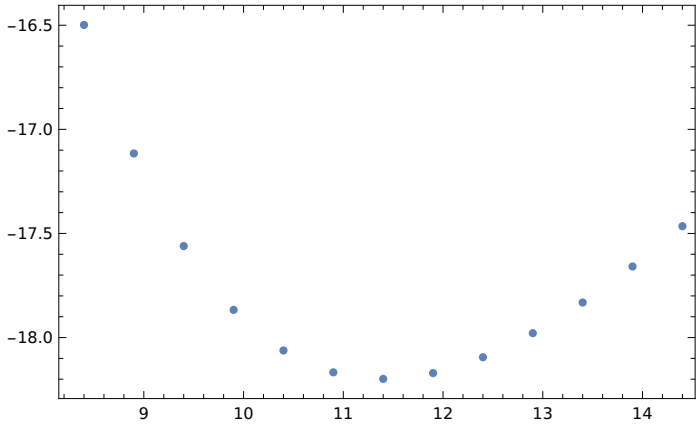
In[•]:=
```
cell = Import["cell.dat"];
cell // TableForm
```

Out[•]//TableForm=

8.4	-16.4987
8.9	-17.1158
9.4	-17.5607
9.9	-17.8672
10.4	-18.0622
10.9	-18.167
11.4	-18.1986
11.9	-18.1707
12.4	-18.0945
12.9	-17.979
13.4	-17.8316
13.9	-17.6586
14.4	-17.4651

In[•]:=
```
cell // ListPlot[#, Frame → True] &
```

Out[•]=



In[•]:=
```
eos = e0 + (9 v0 B0)/16 * ((((v0/v)^(2/3) - 1)^3 * B0' + ((v0/v)^(2/3) - 1)^2 * (6 - 4 (v0/v)^(2/3))))
```

Out[•]=
$$e0 + \frac{9}{16}\, B0\, v0 \left(\left(6 - 4 \left(\frac{v0}{v}\right)^{2/3}\right)\left(-1 + \left(\frac{v0}{v}\right)^{2/3}\right)^2 + \left(-1 + \left(\frac{v0}{v}\right)^{2/3}\right)^3 B0'\right)$$

In[•]:=
```
eos // PowerExpand // ExpandAll // Collect[#, v] &
```

Out[•]=
$$e0 + \frac{27\, B0\, v0}{8} - \frac{9}{16}\, B0\, v0\, B0' + \frac{-9\, B0\, v0^{5/3} + \frac{27}{16}\, B0\, v0^{5/3}\, B0'}{v^{2/3}} + \frac{\frac{63}{8}\, B0\, v0^{7/3} - \frac{27}{16}\, B0\, v0^{7/3}\, B0'}{v^{4/3}} + \frac{-\frac{9\, B0\, v0^3}{4} + \frac{9}{16}\, B0\, v0^3\, B0'}{v^2}$$

In[•]:=
```
model = k[1] + k[2] v^(-2/3) + k[3] v^(-4/3) + k[4] v^(-2);
```

In[•]:=
```
parameters = Table[k[i], {i, 4}]
```

Out[•]=
```
{k[1], k[2], k[3], k[4]}
```

In[]:=
```
nlm = NonlinearModelFit[cell, model, parameters, v]
```

Out[]=
$$\text{FittedModel}\left[21.6 - \frac{687.}{v^2} + \frac{1.29\times10^3}{v^{4/3}} - \frac{430.}{v^{2/3}}\right]$$

In[]:=
```
nlm["BestFitParameters"]
```

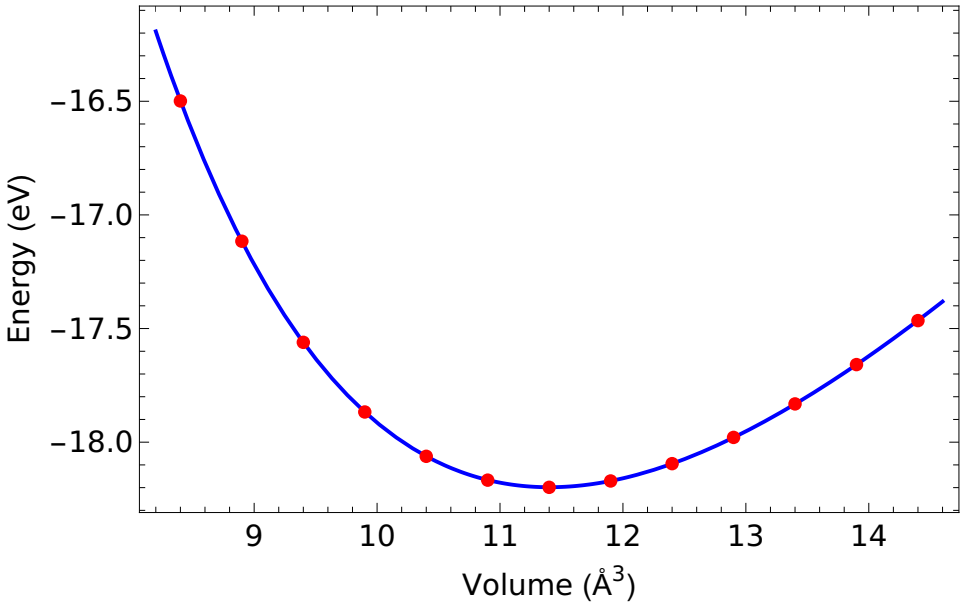
Out[]=
```
{k[1] → 21.6213, k[2] → -430.195, k[3] → 1293.12, k[4] → -687.44}
```

In[]:=
```
nlm[v]
```

Out[]=
$$21.6213 - \frac{687.44}{v^2} + \frac{1293.12}{v^{4/3}} - \frac{430.195}{v^{2/3}}$$

In[]:=
```
Plot[nlm[v], {v, (cell[[All, 1]] // Min) - 0.2, (cell[[All, 1]] // Max) + 0.2},
 Frame → True,
 FrameLabel → {"Volume (Å^3)", "Energy (eV)"},
 BaseStyle → {FontFamily → "Helvetica", FontSize → 15},
 PlotRange → All ,
 ImageSize → 500,
 Epilog → {Red, AbsolutePointSize[7], Point[cell]},
 Axes → None,
 PlotStyle → {AbsoluteThickness[2], Blue}]
```

Out[]=



In[]:=
```
nlm["RSquared"]
```

Out[]=
```
1.
```

The optimal value

In[]:=
```
opvol = FindMinimum[nlm[v], {v, 11, 12}]
```

Out[]=
```
{-18.1981, {v → 11.4004}}
```

To get the lattice parameter in this system we have::

In[•]:=
```
Solve[vol == s^3/4, s]
```

Out[•]=
$$\{\{s \to (-2)^{2/3}\ vol^{1/3}\}, \{s \to 2^{2/3}\ vol^{1/3}\}, \{s \to -(-1)^{1/3}\ 2^{2/3}\ vol^{1/3}\}\}$$

In[•]:=
```
2^(2/3) v^(1/3) /. opvol[[2]]
```

Out[•]=
```
3.57267
```

The bulk modulus B_0 is :

In[•]:=
```
bulkm = v D[nlm[v], {v, 2}] /. opvol[[2]]
```

Out[•]=
```
2.69232
```

In[•]:=
```
UnitConvert[Quantity[bulkm, "Electronvolts"/"Angstroms"^3], "GigaPascals"]
```

Out[•]=
```
431.357 GPa
```

B_0 ' is :

In[•]:=
```
-v/bulkm D[v D[nlm[v], {v, 2}], v] /. opvol[[2]]
```

Out[•]=
```
3.69364
```

In[•]:=
```
sol = Solve[{
    e0 + 27 B0 v0/8 - 9/16 B0 v0 B' == k[1],
    -9 B0 v0^(5/3) + 27/16 B0 v0^(5/3) B' == k[2],
    63/8 B0 v0^(7/3) - 27/16 B0 v0^(7/3) B' == k[3],
    -9 B0 v0^3/4 + 9/16 B0 v0^3 B' == k[4]} /. nlm["BestFitParameters"], {e0, B0, B', v0}]
```

Out[•]=
```
{{e0 → 199.826, B0 → -954.996, B′ → 5.63969, v0 → 0.920696},
 {e0 → -18.1981, B0 → 2.69232, B′ → 3.69364, v0 → 11.4004}}
```

In[•]:=
```
acal = 2^(2/3) v0^(1/3) /. sol[[2]]
```

Out[•]=
```
3.57267
```

The error predicted lattice parameters is :

In[•]:=
```
(acal - aexpt)/aexpt 100 /. {aexpt → 3.567}
```

Out[•]=
```
0.159044
```

Density of States and Partial Density of States (DOS)

```
In[•]:= wrk =
          "/home/alanp/wrk/dft_course_hands-on/final_exam/scripts_inputs/dos";
        SetDirectory[wrk];
```

Here write a comment about the cohesion energy: what is the definition?

```
In[•]:= input = Import["DOSCAR", "Table"];
```

```
In[•]:= input;
```

```
In[•]:= dos = input[[7 ;; 7 + 300]][[All, {1, 2}]];
```

```
In[•]:= dos[[1]]
Out[•]= {-13.754, 0.}
```

```
In[•]:= dos[[-1]]
Out[•]= {28.382, 0.}
```

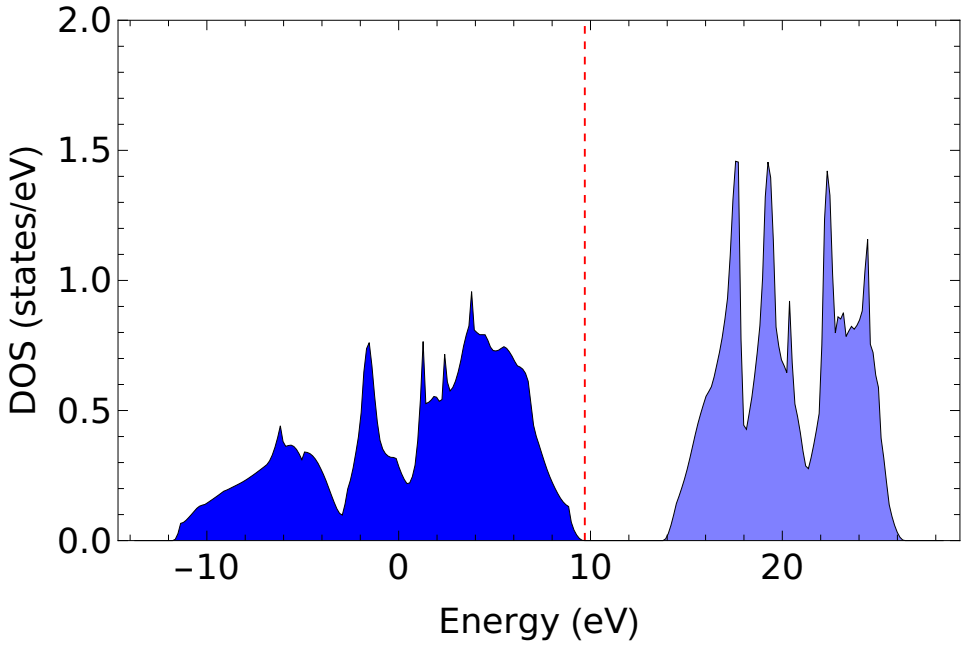
Fermi level

```
In[•]:= ef = input[[6, 4]]
Out[•]= 9.70367
```

Density of States (DOS)

In[•]:=
```
ListPlot[dos,
 Joined → True,
 Axes → False,
 Frame → True,
 FrameLabel → {"Energy (eV)", "DOS (states/eV)"},
 GridLines → {{ef}, None},
 GridLinesStyle → Directive[Red, Dashed, AbsoluteThickness[1]],
 BaseStyle → {FontFamily → "Helvetica", FontSize → 18},
 Filling → Axis,
 FillingStyle → Directive[Opacity[0.5], Blue],
 PlotStyle → {AbsoluteThickness[0.5], Black},
 ImageSize → 500,
 PlotRange → {All, {0, 2.0}}
]
```

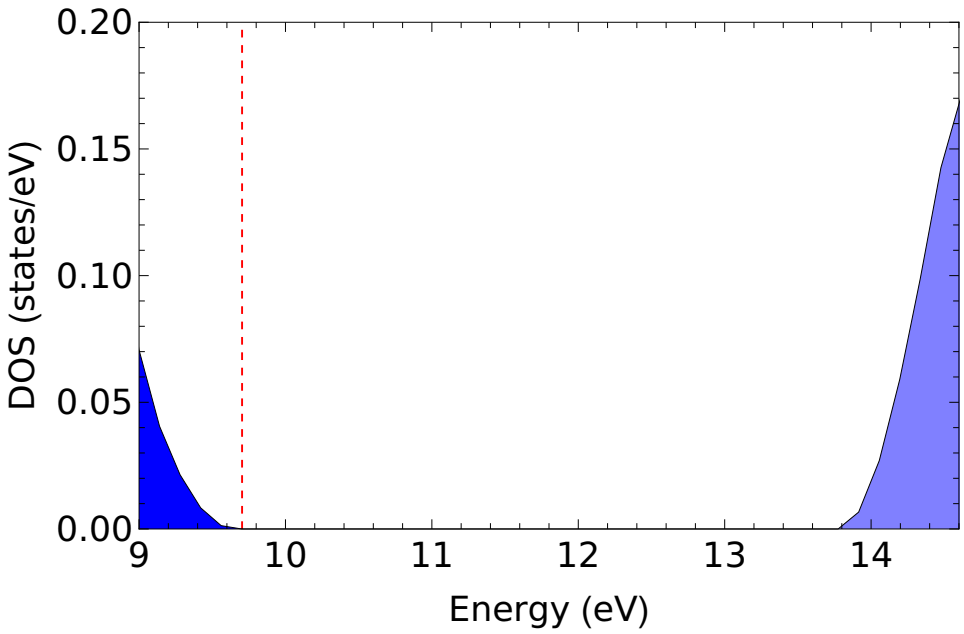
Out[•]=



Valence band on the left, conduction band on the right

In[•]:=
```
ListPlot[dos,
 Joined → True,
 Axes → False,
 Frame → True,
 FrameLabel → {"Energy (eV)", "DOS (states/eV)"},
 GridLines → {{ef}, None},
 GridLinesStyle → Directive[Red, Dashed, AbsoluteThickness[1]],
 BaseStyle → {FontFamily → "Helvetica", FontSize → 18},
 Filling → Axis,
 FillingStyle → Directive[Opacity[0.5], Blue],
 PlotStyle → {AbsoluteThickness[0.5], Black},
 ImageSize → 500,
 PlotRange → {{9, 14.6}, {0, 0.2}}

]
```

Out[•]=



The band gap is the following value

In[•]:=
```
13.77 - ef
```

Out[•]=
```
4.06633
```

Below, we are going to analyze the partial density of states (PDOS), the arrange of the data PDOS in according to the AO :
energy, s, p_y,p_z,p_x,d_{xy}, d_{yz}, d_{z2-r2}, d_{xz}, d_{x2-y2}

In[•]:=
```
dos1 = input[[7 + 300 + 2 ;; 7 + 300 + 2 + 300]];
dos2 = input[[7 + 300 + 2 + 300 + 2 ;; 7 + 300 + 2 + 300 + 2 + 300]];
```

In[•]:=
```
dos1[[1]]
```

Out[•]=
```
{-13.754, 0., 0., 0., 0., 0., 0., 0., 0., 0.}
```

In[•]:=
```
dos2[[1]]
```

Out[•]=
```
{-13.754, 0., 0., 0., 0., 0., 0., 0., 0., 0.}
```

In[•]:=
```
dosall = (dos1 + dos2) // Map[# / {2, 1, 1, 1, 1, 1, 1, 1, 1, 1} &, #] &;
```

In[•]:=
```
dosall[[1]]
```

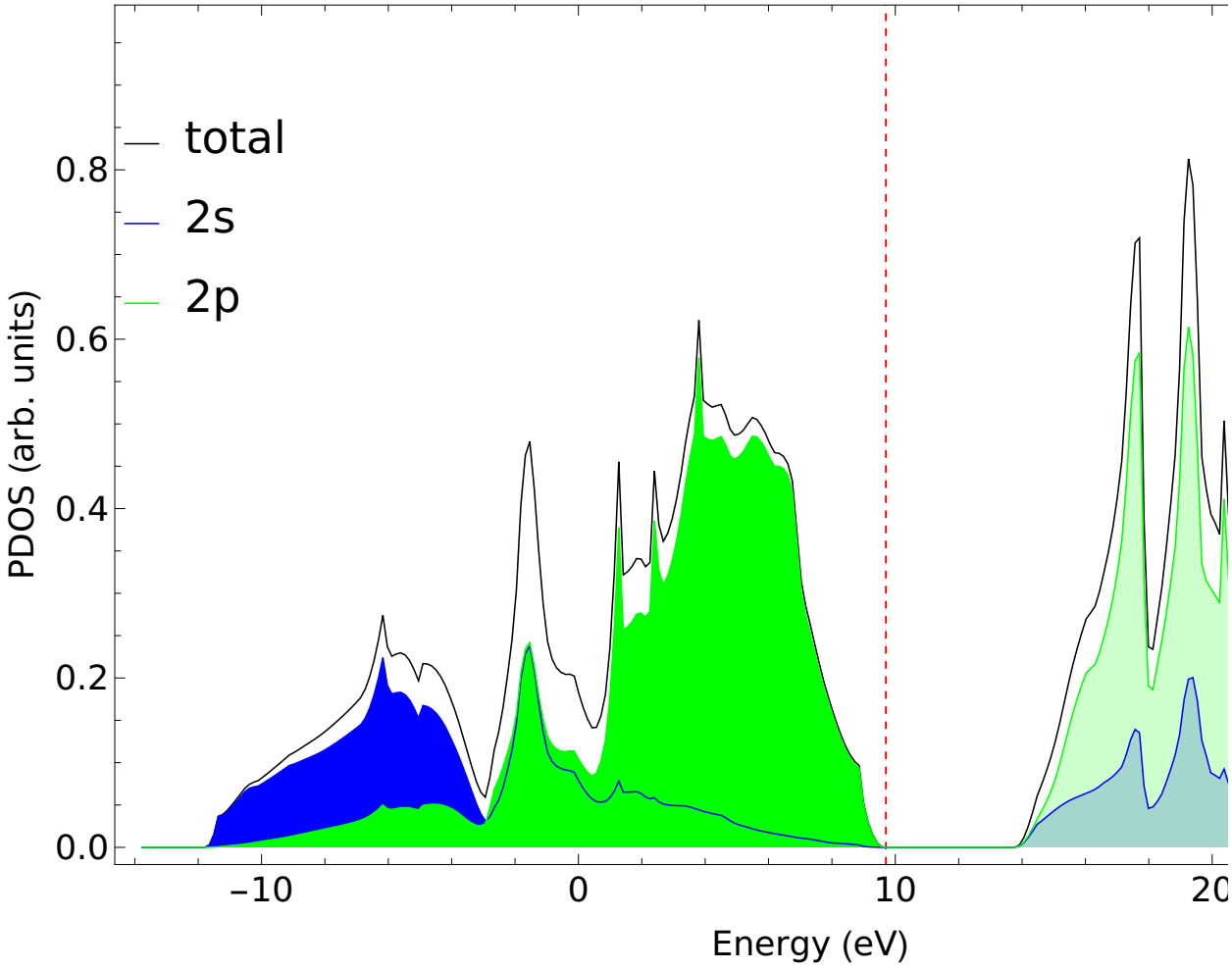
Out[•]=
```
{-13.754, 0., 0., 0., 0., 0., 0., 0., 0., 0.}
```

In[•]:=
```
f[total] = Map[{#[[1]], Plus @@ (#[[2 ;; 10]])} &, #] &;
```

In[•]:=
```
f[s] = Map[{#[[1]], #[[2]]} &, #] &;
f[p] = Map[{#[[1]], Plus @@ (#[[3 ;; 5]])} &, #] &;
f[d] = Map[{#[[1]], Plus @@ (#[[6 ;; 10]])} &, #] &;
f[py] = Map[{#[[1]], #[[3]]} &, #] &;
f[pz] = Map[{#[[1]], #[[4]]} &, #] &;
f[px] = Map[{#[[1]], #[[5]]} &, #] &;
```

In[•]:=
```
{dosall // f[total], dosall // f[s], dosall // f[p]} // ListPlot[#,
  Joined → True,
  GridLines → {{ef}, None},
  Filling → {2 → Axis, 3 → Axis},
  Frame → True,
  PlotRange → All,
  PlotLegends → Placed[{"total", "2s", "2p"}, {Scaled[{0, 0.75}], {0, 0.5}}],
  ImageSize → 800,
  Axes → False,
  GridLinesStyle → Directive[Red, Dashed, AbsoluteThickness[1.0]],
  PlotStyle → {{AbsoluteThickness[0.8], Black},
    {AbsoluteThickness[0.8], Blue}, {AbsoluteThickness[0.8], Green}},
  FrameLabel → {"Energy (eV)", "PDOS (arb. units)"},
  BaseStyle → {FontFamily → "Helvetica", FontSize → 18}
 ] &
```

Out[•]=



Now we have to compute the band structure. Bellow is the band structure of C-fcc along the path : L→Γ→X→K→Γ

Band Structure from PROCAR_OPT Along the path: L→Γ→X→K→Γ

The following notebook will allow you to create the band structure from any VASP calculation that produces a PROCAR_OPT file

To run this notebook, be sure you have in same directory PROCAR_OPT and KPOINTS_OPT.

%%%% you can run the lines below %%%%%%%%%%

In[•]:=
```
SetDirectory["/home/alanp/wrk/dft_course_hands-on/final_exam/scripts_inputs/dos"];
```

In[•]:=
```
Run["grep k-point PROCAR_OPT > kpt"];
Run["sed 's/-/ -/g' kpt > kpt2"];
Run["sed 's/k -point/k-point/g' kpt2 > kpts"];
Run["rm kpt kpt2"];
Run["grep band PROCAR_OPT > bandData"];
Run["grep \"# of bands\" PROCAR_OPT | tail -1 > numbands"];
```

In[•]:=
```
(*define the Fermi elevel take from previous calculations OUTCAR.st*)
ef = 9.70367048`;
```

In[•]:=
```
(*ef=input[[6,4]];*)
```

In[•]:=
```
(*This gets number of kp and bands*)
bandsinfo = Import["numbands", "Table"][[1]];
```

In[•]:=
```
bandsinfo[[4]] // TableForm
```

Out[•]//TableForm=
```
140
```

In[•]:=
```
{Nkpt, Nbands} = bandsinfo // {#[[4]], #[[8]]} &
```

Out[•]=
```
{140, 8}
```

In[•]:=
```
(*This gets the high symmetry points and divisions*)
path = Import["KPOINTS-bs", "Table"];
div = path[[2, 1]];
```

In[•]:=
```
(*This defines the high symmetry points labels and the length*)
hsympts = (If[(path[[5]] // Length) == 4, path[[5 ;; -1]] // Flatten // Partition[#, 4] & // #[[All, 4]] &,
      path[[5 ;; -1]] // Flatten // Partition[#, 5] & // #[[All, 5]] &] //
    Drop[#, {3, Length[#], 2}] &) /. "GAMMA" → "Γ"
NumGridL = (hsympts // Length) - 2;
```

Out[•]=
```
{L, Γ, X, K, Γ}
```

In[•]:=
```
(*This defines the band energies shifted to Ef*)
bands = Import["bandData", "Table"][[2 ;; Nbands Nkpt + 1]][[All, 5]] - ef // Partition[#, Nbands] &;
bandener = bands // Drop[#, {div, Nkpt - div, div}] &;
```

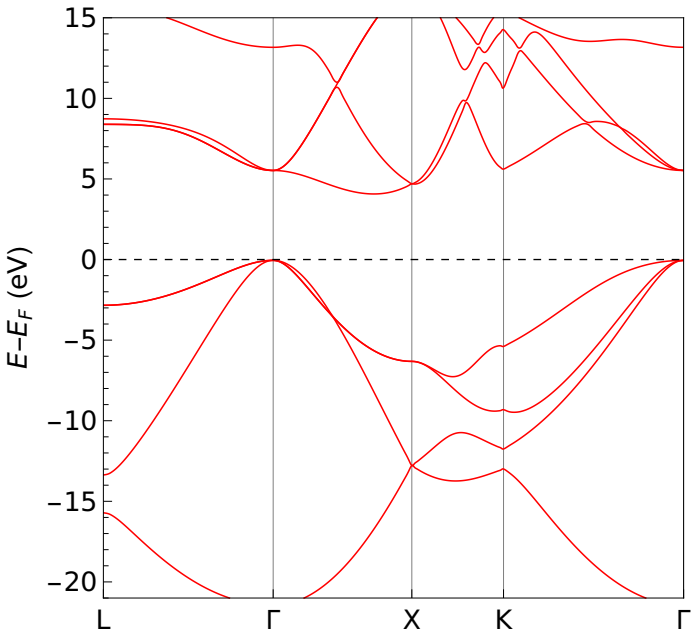
In[•]:=
```
(*This defines k-point coordinates*)
kp = Import["kpts", "Table"][[2 ;; Nkpt + 1]][[All, {4, 5, 6}]];
kpt = kp // Drop[#, {div, Nkpt - div, div}] &;
```

In[•]:=
```
(*This liniarizes k-point coordinates in 1D*)
kpt2 = Join[{0}, Table[Norm[kpt[[i + 1]] - kpt[[i]]], {i, (kpt // Length) - 1}]] // Accumulate;
```

In[•]:=
```
(*This creates the bands in kpt2 domain*)
bands2 = {kpt2, bandener} // Transpose;
Do[bd[i] = bands2 // Map[{#[[1]], #[[2, i]]} &, #] &, {i, Nbands}];
```

```
(*please provide the bands you want to plot*)
ListPlot[Table[bd[i], {i, 1, 8(*Nbands*)}],
 Frame → True,
 FrameLabel → {"", "E-E_F (eV)"},
 BaseStyle → {FontFamily → "Helvetica", FontSize → 14},
 Joined → True,
 InterpolationOrder → 2,
 AspectRatio → 1,
 AxesStyle → Directive[Black, Dashed, AbsoluteThickness[0.7]],
 FrameTicks → {{Automatic, None},
   {{kpt2[[Join[Table[i div - (i - 1), {i, 0, NumGridL}], {-1}]]], hsympts} // Transpose, None}},
 GridLines → {kpt2[[Join[Table[i div - (i - 1), {i, 0, NumGridL}], {-1}]]], None},
 PlotStyle → Directive[Red, AbsoluteThickness[0.8]],
 PlotRange → {{0, kpt2[[-1]]}, {-21, 15}}]
```

Out[•]=



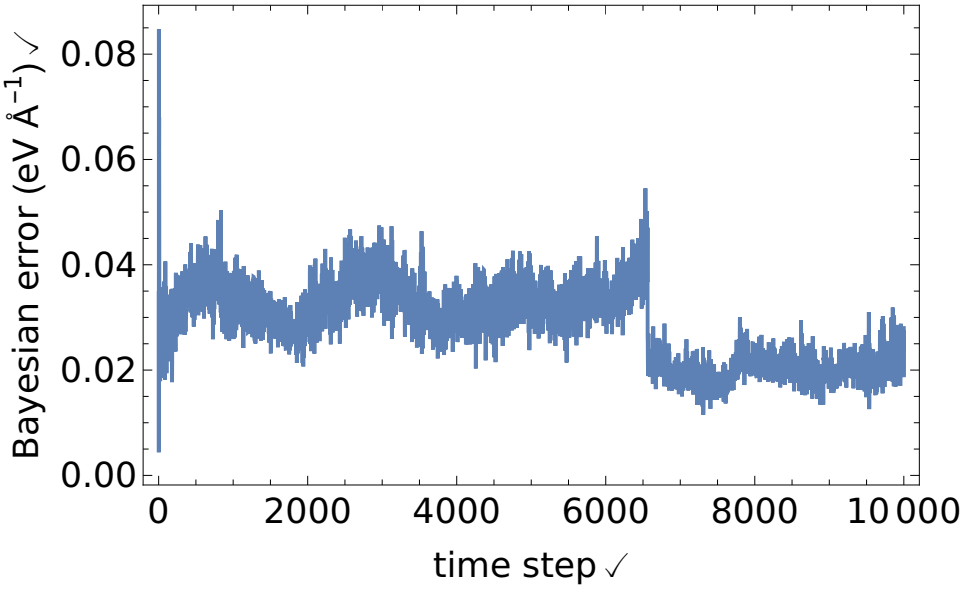
Machine Learning C(100) Surface

```
In[•]:= wrkdir = "/home/alanp/wrk/dft_course_hands-on/final_exam/machlearn";
        SetDirectory[wrkdir];

In[•]:= paramplot =
          {BaseStyle → {FontFamily → "Helvetica", FontSize → 18}, ImageSize → 500, Frame → True};
```

Bayesian error

```
In[•]:= beef = Import["ml/beef.dat"];
        beef // ListPlot[#, FrameLabel → ✓{"time step", "Bayesian error (eV Å^-1)"},
           Joined → True, PlotRange → All, paramplot] &
```

Out[•]=



This error comes from external surface data and isn't very large—it' s only concerning when it reaches around 0.1 or higher . It mainly reflects the quality of our prediction .

Let' s check the KPOINT grid

```
In[•]:= sc = Import["POSCAR-222", "Table"];

In[•]:= lattsc = sc[[2, 1]] × sc[[3 ;; 5]]

Out[•]= {{5.05252, 0., 0.}, {2.52626, 4.37561, 0.}, {2.52626, 1.45854, 4.12537}}
```

In[•]:=
```
{a, b, c} = lattsc;
Δk = Table[0.08 - 0.005 i , {i, 0, 10}];
Δk // Map[{#, Round[(b × c)/((a×b).c) // Norm / #], Round[(a × c)/((a×b).c) // Norm / #], Round[(b × c)/((a×b).c) // Norm / #]} &, #] & // TableForm
```

Out[•]//TableForm=

0.08	3	3	3
0.075	3	3	3
0.07	3	3	3
0.065	4	4	4
0.06	4	4	4
0.055	4	4	4
0.05	5	5	5
0.045	5	5	5
0.04	6	6	6
0.035	7	7	7
0.03	8	8	8

In[•]:=
```
pc = Import["POSCAR-pc", "Table"]; (*same as POSCAR-bulk*)
```

In[•]:=
```
lattpc = pc[[2, 1]] × pc[[3 ;; 5]]
```

Out[•]=
```
{{2.5145, 0., 0.}, {1.25725, 2.17762, 0.}, {1.25725, 0.725874, 2.05308}}
```

In[•]:=
```
{a, b, c} = lattpc;
Δk = Table[0.08 - 0.005 i , {i, 0, 10}];
Δk // Map[{#, Round[(b × c)/((a×b).c) // Norm / #], Round[(a × c)/((a×b).c) // Norm / #], Round[(b × c)/((a×b).c) // Norm / #]} &, #] & // TableForm
```

Out[•]//TableForm=

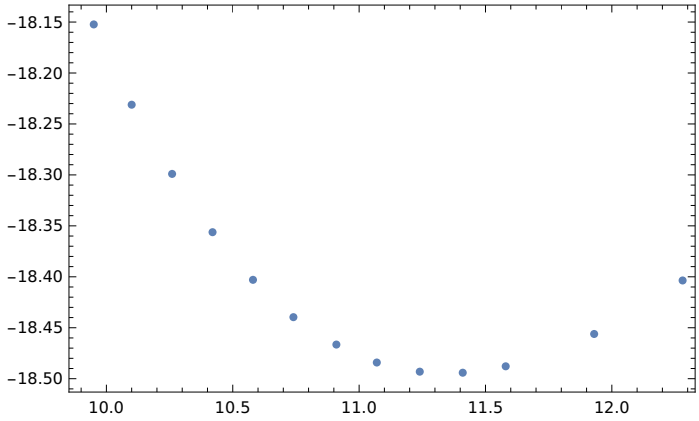
0.08	6	6	6
0.075	6	6	6
0.07	7	7	7
0.065	7	7	7
0.06	8	8	8
0.055	9	9	9
0.05	10	10	10
0.045	11	11	11
0.04	12	12	12
0.035	14	14	14
0.03	16	16	16

EOS MLFFF VS DFT

This is the computed EOS using ML:

In[•]:=
```
datML = Import["ml/eos-mlPC/cell-ml.dat"];
ListPlot[datML , Frame → True]
```

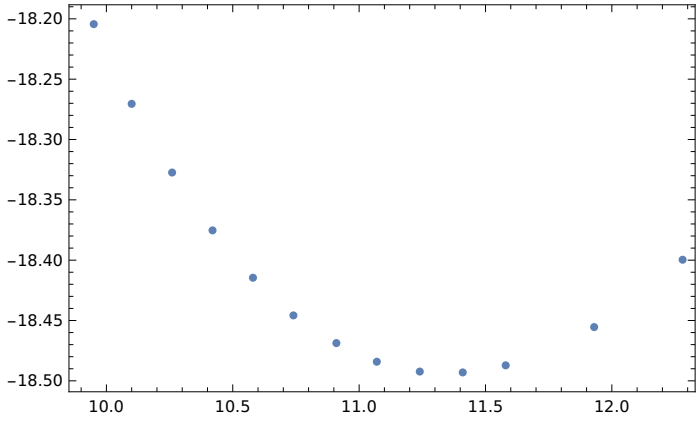
Out[•]=



This is the computed EOS using DFT

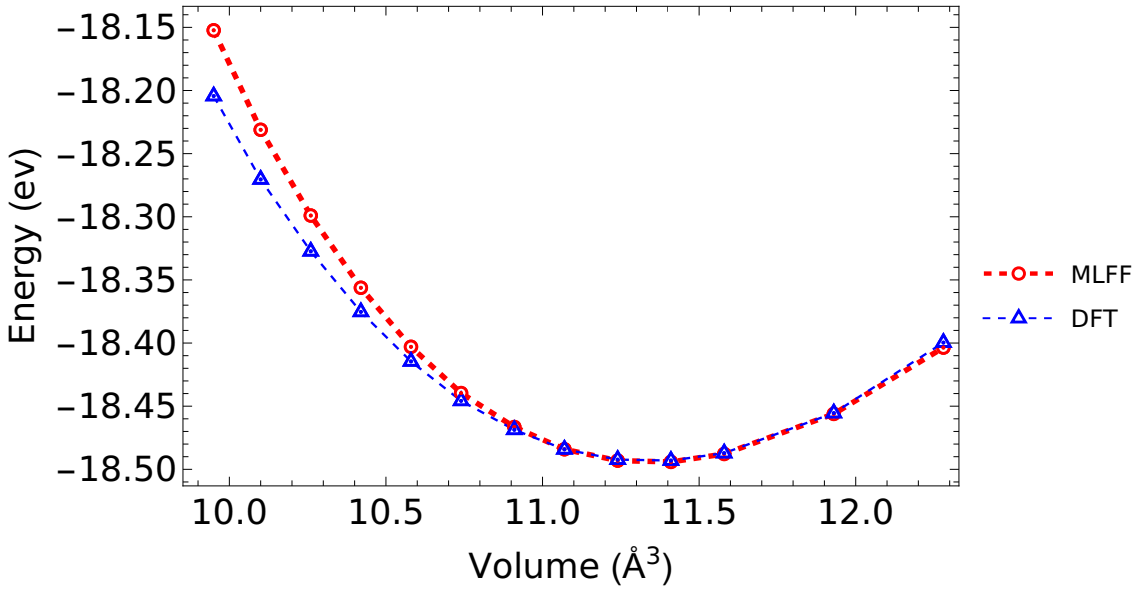
In[•]:=
```
dataAB = Import["dft-eosPC/cell-dft.dat"];
ListPlot[dataAB, Frame → True]
```

Out[•]=



```
In[•]:= ListLinePlot[{datML, dataAB},
 Frame → True,
 Joined → True,
 PlotStyle → {{Red, Thickness[0.007], Dashed}, {Blue, Thickness[0.003], Dashed}},
 FrameLabel → {"Volume (Å^3)", "Energy (ev)"},
 PlotLegends → {"MLFF", "DFT"},
 ImageSize → 500,
 paramplot,
 Mesh → Full,
 MeshStyle → PointSize[Small],
 PlotMarkers → "OpenMarkers"
]
```

Out[•]=



Both results nearly match perfectly, even though we only used 10,000 training steps.

KPOINTS Grid (slab)

```
In[•]:= slab = Import["POSCAR-100", "Table"];

In[•]:= latt = slab[[2, 1]] × slab[[3 ;; 5]]

Out[•]= {{2.5263, 0., 0.}, {0., 2.5263, 0.}, {0., 0., 13.5727}}

In[•]:= {a, b, c} = latt;

In[•]:= Δk = Table[0.14 - 0.005 i, {i, 0, 20}];
```

In[•]:=
```
Δk // Map[{#, Round[(b×c)/(a.(b×c)) // Norm / #], Round[(a×c)/(a.(b×c)) // Norm / #], Round[(a×b)/(a.(b×c)) // Norm / #]} &, #] & // TableForm
```

Out[•]//TableForm=

0.14	3	3	1
0.135	3	3	1
0.13	3	3	1
0.125	3	3	1
0.12	3	3	1
0.115	3	3	1
0.11	4	4	1
0.105	4	4	1
0.1	4	4	1
0.095	4	4	1
0.09	4	4	1
0.085	5	5	1
0.08	5	5	1
0.075	5	5	1
0.07	6	6	1
0.065	6	6	1
0.06	7	7	1
0.055	7	7	1
0.05	8	8	1
0.045	9	9	2
0.04	10	10	2

In[•]:=
```
slab = Import["surf/POSCAR-2x2l5v10-initNEW.POSCAR", "Table"];
```

In[•]:=
```
latt = slab[[2, 1]] × slab[[3 ;; 5]]
```

Out[•]=
```
{{5.0526, 0., 0.}, {0., 5.0526, 0.}, {0., 0., 14.5727}}
```

In[•]:=
```
{a, b, c} = latt;
```

In[•]:=
```
Δk = Table[0.14 - 0.005 i, {i, 0, 20}];
```

In[•]:=
```
Δk // Map[{#, Round[b×c / (a.(b×c)) // Norm / #], Round[a×c / (a.(b×c)) // Norm / #], Round[a×b / (a.(b×c)) // Norm / #]} &, #] & // TableForm
```

Out[•]//TableForm=

0.14	1	1	0
0.135	1	1	1
0.13	2	2	1
0.125	2	2	1
0.12	2	2	1
0.115	2	2	1
0.11	2	2	1
0.105	2	2	1
0.1	2	2	1
0.095	2	2	1
0.09	2	2	1
0.085	2	2	1
0.08	2	2	1
0.075	3	3	1
0.07	3	3	1
0.065	3	3	1
0.06	3	3	1
0.055	4	4	1
0.05	4	4	1
0.045	4	4	2
0.04	5	5	2

In[•]:=
```
Δk = 0.12;
```

In[•]:=
```
Δk // Map[{#, Round[b×c / (a.(b×c)) // Norm / #], Round[a×c / (a.(b×c)) // Norm / #], Round[a×b / (a.(b×c)) // Norm / #]} &, #] & // TableForm
```

Out[•]//TableForm=

0.12

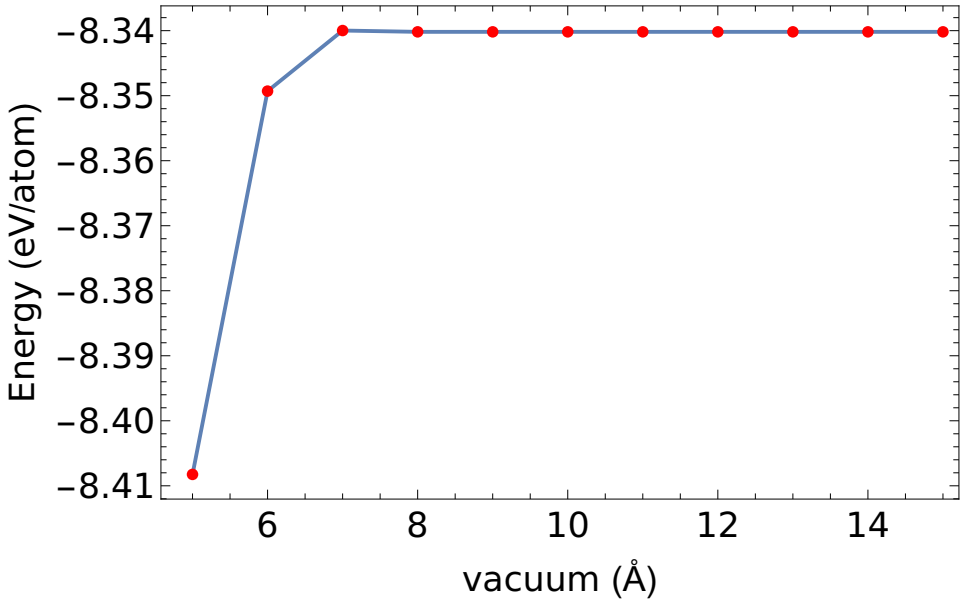
The change of energy per atom VS vacuum size

In[•]:=
```
vac = Import["surf/vacuum.dat"] // Map[# / {1, 5} &, #] &
```

Out[•]=
```
{{5., -8.40825}, {6., -8.3493}, {7., -8.33998}, {8., -8.34018}, {9., -8.34018}, {10., -8.34018},
 {11., -8.34018}, {12., -8.34018}, {13., -8.34018}, {14., -8.34018}, {15., -8.34018}}
```

In[•]:=
```
ListPlot[vac,
 Frame → True,
 FrameLabel → {"vacuum (Å)", "Energy (eV/atom)"},
 Axes → False,
 paramplot,
 PlotRange → All,
 Joined → True,
 Epilog → {Red, AbsolutePointSize[6], Point[vac]}
]
```

Out[•]=



We need to use at least 8 Å, but since we'll be computing STM images, around 10 Å is more appropriate—so we'll use 11 Å.

Surface Energies

In[•]:=
```
surfen = Import["surf/surfaces.dat"][[All, {1, 2}]];
```

In[•]:=
```
surfen // TableForm
```

Out[•]//TableForm=

```
2x2l5v10    -195.044
```

In[•]:=
```
A2x2 = 5.0526^2;
e[C] = -18.19862827 / 2;
e[H] = -3.3795;
area = {A2X2};
```

In[•]:=
```
nsi = 20;
nh = 8;
```

In[•]:=
$$\frac{1}{2\,A2x2}\left(Eslab - nsi\,e[C] - nh\,e[H]\right) /.\ Eslab \to surfen[[1,\ 2]]$$

Out[•]= 0.273771

This is the energy of our system $(eV/Å^2)$

DOS of C (100) Surface

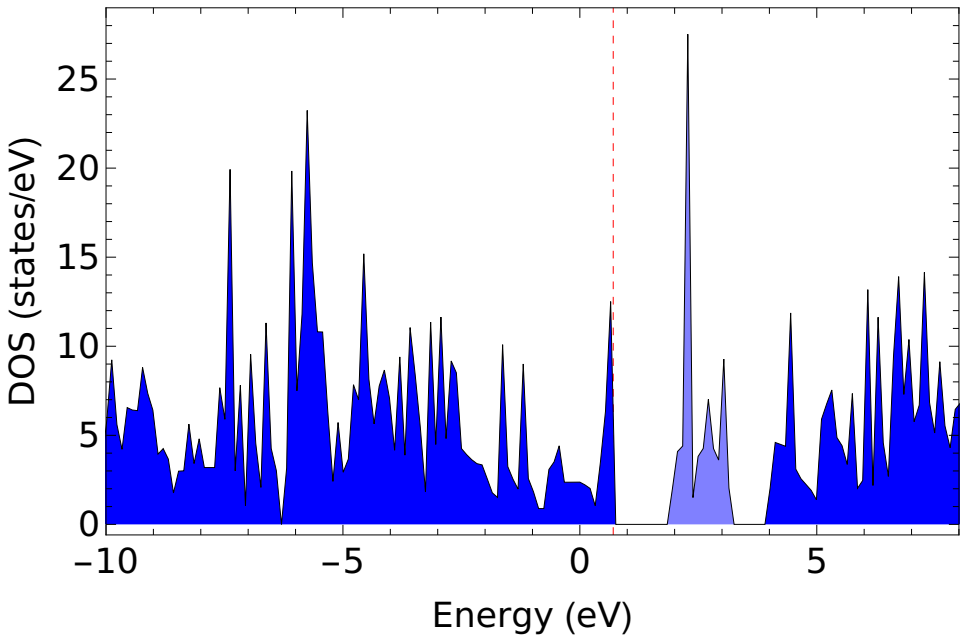
In[•]:=
```
input = Import["surf/DOSCAR-2x2l5v10.st", "Table"];
```

In[•]:=
```
dos = input[[7 ;; 7 + 300]][[All, {1, 2}]];
ef = input[[6, 4]]
```

Out[•]= 0.704167

In[•]:=
```
ListPlot[dos,
 Joined → True,
 Axes → False,
 FrameLabel → {"Energy (eV)", "DOS (states/eV)"},
 GridLines → {{ef}, None},
 GridLinesStyle → Directive[Red, Dashed],
 paramplot,
 PlotRange → {{-10., 8.0}, All},
 Filling → Axis,
 FillingStyle → Directive[Opacity[0.5], Blue],
 PlotStyle → {AbsoluteThickness[0.5], Black}
]
```

Out[•]=



This is the density of states of the surface, and we can observe a band gap—just like in the bulk

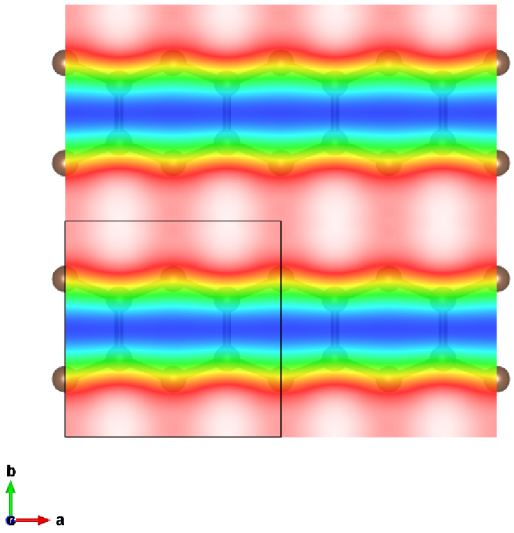
system, which also exhibited a band gap .

The band gap is:

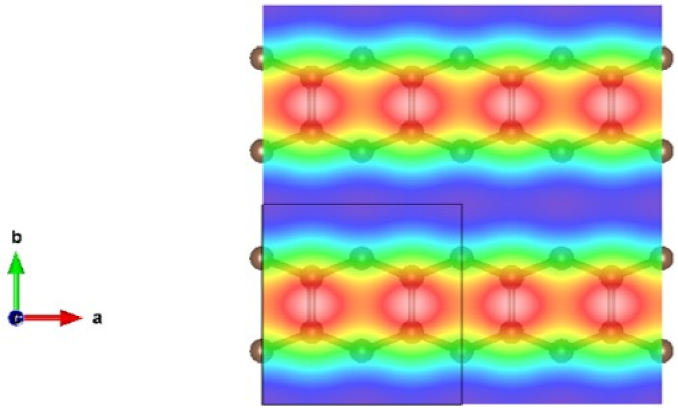
In[•]:=
```
1.952 - ef
```

Out[•]=
```
1.24783
```

Computed constant height mode STM image for a voltage 2.5 V for the slab 2*2 using VESTA



Computed constant height mode STM image for a voltage 2.5 V for the slab 2*2 using VESTA



Generation of STM Images for C(100) V 2x2 $\left(V_{VIAS} = -2.0\,V\right)$

```
In[•]:=
in = Import["surf/PARCHG-2x2l5v10+2.5.vasp", "Table"];

In[•]:= numat = Plus @@ in[[7]]
Out[•]= 28

In[•]:=
thelattice = in[[3 ;; 5]]
Out[•]= {{5.0526, 0., 0.}, {0., 5.0526, 0.}, {0., 0., 14.5727}}

In[•]:= in[[7 + numat + 3]]
Out[•]= {72, 72, 216}

In[•]:= stm = (in[[7 + numat + 4 ;; -1]] // Flatten);

In[•]:= stm // Dimensions
Out[•]= {1 119 744}

In[•]:= stm[[1]]
Out[•]= 2.0565

In[•]:= stm[[-1]]
Out[•]= 2.4507

In[•]:= divX = in[[7 + numat + 3, 1]]
Out[•]= 72

In[•]:= divY = in[[7 + numat + 3, 2]]
Out[•]= 72

In[•]:= zrange = in[[5, {2, 3}]]
Out[•]= {0., 14.5727}

In[•]:=
transformDATA =
  (
    #
          // Partition[#, divX] &
         // Partition[#, divY] &
        // Map[Append[#, First[#]] &, #, {2}] &
       // Map[Append[#, First[#]] &, #] &
      // Transpose[#, {3, 2, 1}] &
  ) &;

In[•]:= stm1 = stm // transformDATA;
```

In[•]:=
```
stm1[[1, 1]][[1]]
```

Out[•]=
```
2.0565
```

In[•]:=
```
stm1 // Dimensions
```

Out[•]=
```
{73, 73, 216}
```

Here the interpolation for stm1

In[•]:=
```
theInterpolation = ListInterpolation[#,
    {{0, in[[3, 1]]}, {0, in[[4, 2]]}, {0, in[[5, 3]]}}, PeriodicInterpolation → {True, True, False}] &;
```

In[•]:=
```
stminterpolation = stm1 // theInterpolation;
```

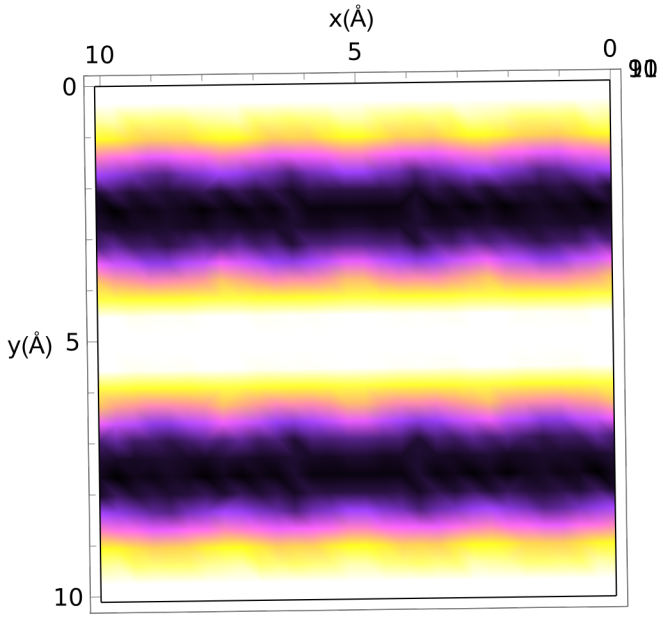
In[•]:=
```
stminterpolation[0, 0, 3]
```

Out[•]=
```
1.08688
```

In[•]:=
```
nx = 2;
ny = 2;
```

In[•]:=
```
nx = 2;
ny = 2;
ContourPlot3D[stminterpolation[x, y, z] == 0.1,
 {x, 0, nx thelattice[[1, 1]]},
 {y, 0, ny thelattice[[2, 2]]},
 {z, 8.6, 11.6},
 ColorFunction → Function[{x, y, z, f}, ColorData["SunsetColors"][z]],
 PlotPoints → 30,
 MaxRecursion → 0,
 BoxRatios → {1, ny thelattice[[2, 2]] / nx thelattice[[1, 1]], (11.6 - 8.6) / nx thelattice[[1, 1]]},
 AxesLabel → {"x(Å)", "y(Å)", None},
 ViewPoint → {0, 0, ∞},
 BaseStyle → {FontFamily → "Helvetica", FontSize → 13},
 ContourStyle → Directive[Opacity[1.]],
 Lighting → AmbientLight[GrayLevel[1.]],
 Mesh → None]
```

Out[•]=



This plot represent the computed constant current mode STM image for a voltage 2.5 V for the slab 2×2 using Mathematica. This image is a top view of the a contour plot of LDOS for a given value of LDOS. The dark colors represent the carbon atoms.

Generation of STM Images for C(100) 2x2 $\left(V_{VIAS} = -2.0\,V\right)$

In[•]:=
```
in = Import["surf/PARCHG-2x2l5v10-2.0.vasp", "Table"];
```

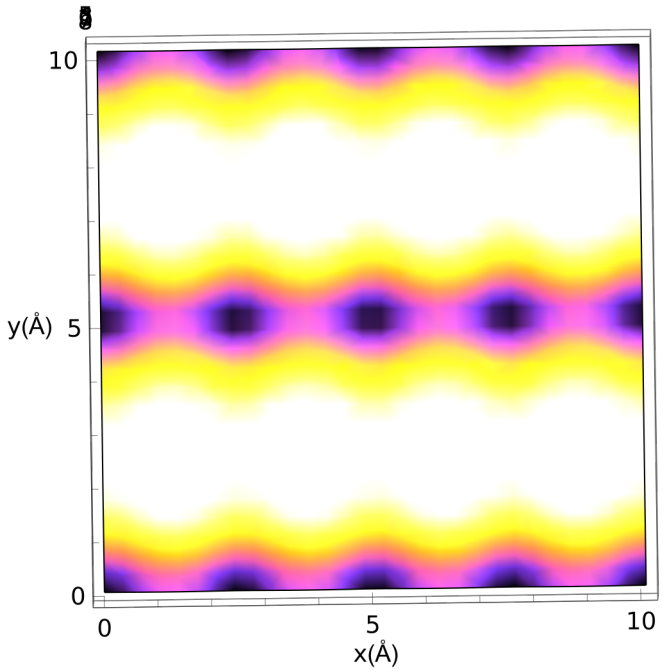
```
In[•]:= numat = Plus @@ in[[7]]
Out[•]= 28

In[•]:=
thelattice = in[[3 ;; 5]]
Out[•]= {{5.0526, 0., 0.}, {0., 5.0526, 0.}, {0., 0., 14.5727}}

In[•]:= in[[7 + numat + 3]]
Out[•]= {72, 72, 216}

In[•]:= stm = (in[[7 + numat + 4 ;; -1]] // Flatten);

In[•]:= stm // Dimensions
Out[•]= {1 119 744}

In[•]:= stm[[1]]
Out[•]= 0.60533

In[•]:= stm[[-1]]
Out[•]= 0.58824

In[•]:= divX = in[[7 + numat + 3, 1]]
Out[•]= 72

In[•]:= divY = in[[7 + numat + 3, 2]]
Out[•]= 72

In[•]:= zrange = in[[5, {2, 3}]]
Out[•]= {0., 14.5727}

In[•]:=
transformDATA =
  (
    #
          // Partition[#, divX] &
         // Partition[#, divY] &
        // Map[Append[#, First[#]] &, #, {2}] &
       // Map[Append[#, First[#]] &, #] &
      // Transpose[#, {3, 2, 1}] &
  ) &;

In[•]:= stm1 = stm // transformDATA;

In[•]:= stm1[[1, 1]][[1]]
Out[•]= 0.60533

In[•]:= stm1 // Dimensions
Out[•]= {73, 73, 216}
```

Here the interpolation for stm1.

```
In[•]:=
theInterpolation = ListInterpolation[#,
    {{0, in[[3, 1]]}, {0, in[[4, 2]]}, {0, in[[5, 3]]}}, PeriodicInterpolation → {True, True, False}] &;

In[•]:= stminterpolation = stm1 // theInterpolation;

In[•]:= stminterpolation[0, 0, 3]

Out[•]= 25.1768

In[•]:= nx = 2;
        ny = 2;
```

In[•]:=
```
nx = 2;
ny = 2;
ContourPlot3D[stminterpolation[x, y, z] == 0.1,
 {x, 0, nx thelattice[[1, 1]]},
 {y, 0, ny thelattice[[2, 2]]},
 {z, 5, 8.5},
 ColorFunction → Function[{x, y, z, f}, ColorData["SunsetColors"][z]],
 PlotPoints → 30,
 MaxRecursion → 0,
 BoxRatios → {1, ny thelattice[[2, 2]] / nx thelattice[[1, 1]], 11.6 - 8.6 / nx thelattice[[1, 1]]},
 AxesLabel → {"x(Å)", "y(Å)", None},
 ViewPoint → {0, 0, ∞},
 BaseStyle → {FontFamily → "Helvetica", FontSize → 13},
 ContourStyle → Directive[Opacity[1.]],
 Lighting → AmbientLight[GrayLevel[1.]],
 Mesh → None]
```

Out[•]=



This plot represent the computed constant current mode STM image for a voltage -2.0 V for the slab 2×2 using Mathematica. This image is a top view of the a contour plot of LDOS for a given value of LDOS. The dark colors represent the carbon atoms.