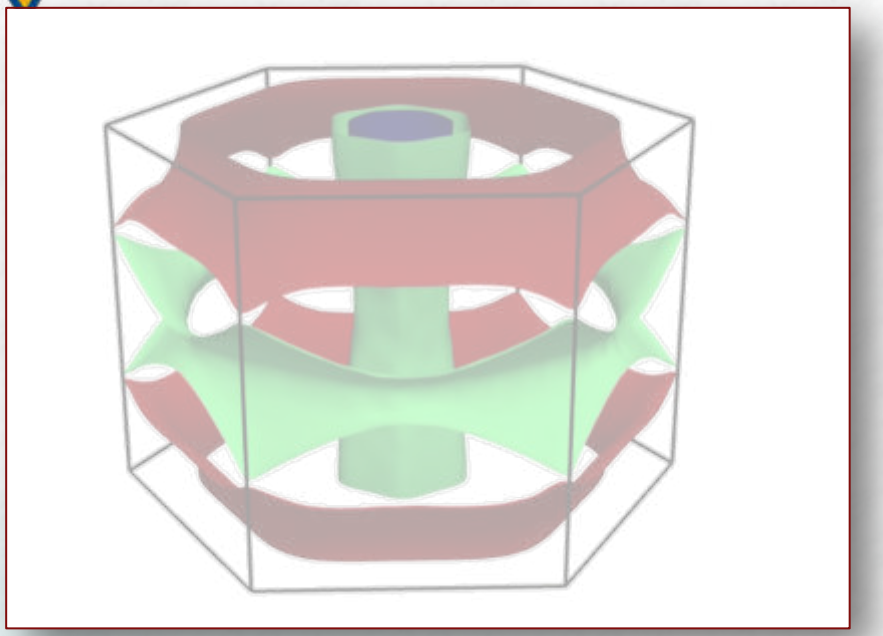
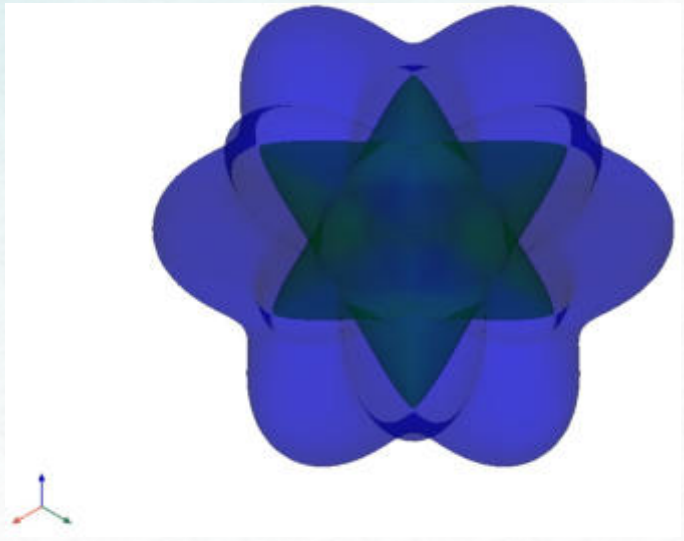
Department of Physics and Astronomy
West Virginia University



Recent Developments in Pyprocar and MechElastic



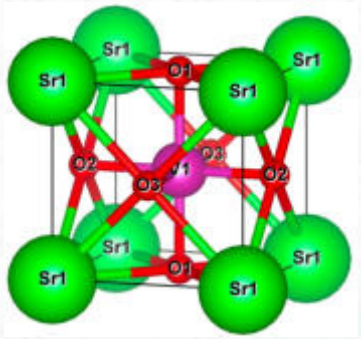
Pyprocar Contributors

Uthpala Herath, Pedram Tavadze, Xu He, Eric Bousquet, Sobhit Singh, Reese Boucher, Logan Lang, Freddy Farah, Francisco Muñoz and Aldo H. Romero

MechElastic Contributors

Logan Lang , Sobhit Singh, Viviana Dovale-Farelo, Uthpala Herath, Pedram Tavazohi, François-Xavier Coudert, Aldo H. Romero

Pyprocar



DFT

Numerical data

```
PROCAR lm decomposed
# of k-points:  200         # of bands:   48         # of ions:    5

 k-point     1 :    0.00000000 0.00000000 0.00000000     weight = 0.00500000

band     1 # energy  -29.05690371 # occ.  2.00000000

ion      s     py     pz     px     dxy    dyz    dz2    dxz  x2-y2    tot
    1  0.972  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.972
    2  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
    3  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
    4  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
    5  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000
tot    0.974  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.000  0.974
```

```
 0.000000000      1.203497247      0.000000   0.000000   0.000000   0.000000   0.000215   0.0
 0.7213197256E-01 1.206644141      0.029906   0.000944   0.000000   0.000944   0.007378   0.0
 0.1442639451     1.215535536      0.014263   0.001967   0.000000   0.001967   0.007869   0.0
 0.2163959177     1.226439061      0.004364   0.002027   0.000000   0.002027   0.008587   0.0
 0.2885278902     1.235537941      0.000056   0.001289   0.000000   0.001289   0.010137   0.0
 0.3606598628     1.233460228      0.002649   0.000051   0.000000   0.000051   0.003660   0.0
 0.4327918353     1.228146745      0.000000   0.000002   0.000000   0.000002   0.000000   0.0
 0.5049238079     1.206604221      0.000000   0.000011   0.000000   0.000011   0.000886   0.0
 0.5770557805     1.195644770      0.000000   0.000000   0.019435   0.000000   0.000000   0.0
 0.6491877530     1.198648720      0.000000   0.004414   0.000000   0.004414   0.035955   0.0
 0.7213197256     1.172068978      0.000000   0.001263   0.000000   0.001263   0.017216   0.0
 0.7934516981     1.143774531      0.000000   0.000369   0.000000   0.000369   0.016328   0.0
 0.8655836707     1.139814551      0.000000   0.000000   0.000000   0.000000   0.000000   0.0
```

```
 k =   0.0000000000  0.0000000000  0.0000000000
==== e(   1) =   -50.21339 eV ====
     psi = 0.478*[#   1]+0.478*[#  14]+0.010*[#   2]+0.010*[#  15]+0.004*[#  28]
           +0.004*[#  32]+0.004*[#  36]+0.004*[#  40]+0.002*[#   9]+0.002*[#  22]

    |psi|^2 = 1.000
==== e(   2) =   -50.21339 eV ====
     psi = 0.477*[#   1]+0.477*[#  14]+0.009*[#   2]+0.009*[#  15]+0.005*[#  28]
           +0.005*[#  32]+0.005*[#  36]+0.005*[#  40]+0.002*[#   9]+0.002*[#  22]

    |psi|^2 = 1.000
==== e(   3) =   -24.41820 eV ====
     psi = 0.485*[#   3]+0.485*[#  16]+0.006*[#   6]+0.006*[#  19]+0.002*[#  27]
```

Li3Mn1O5

- Simple, single line commands

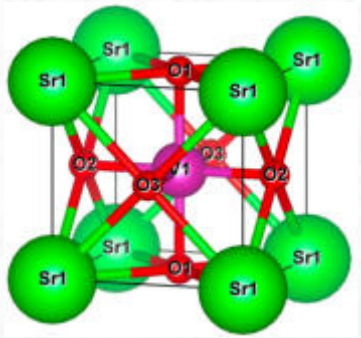
```
$pyprocar.bandsplot('PROCAR',mode='parametric',
orbitals=[4,5,6],code='vasp'…)
```

- Supports multiple DFT codes

VASP, Quantum Espresso, Abinit, Elk, Lobster
Siesta*, DFTB+*

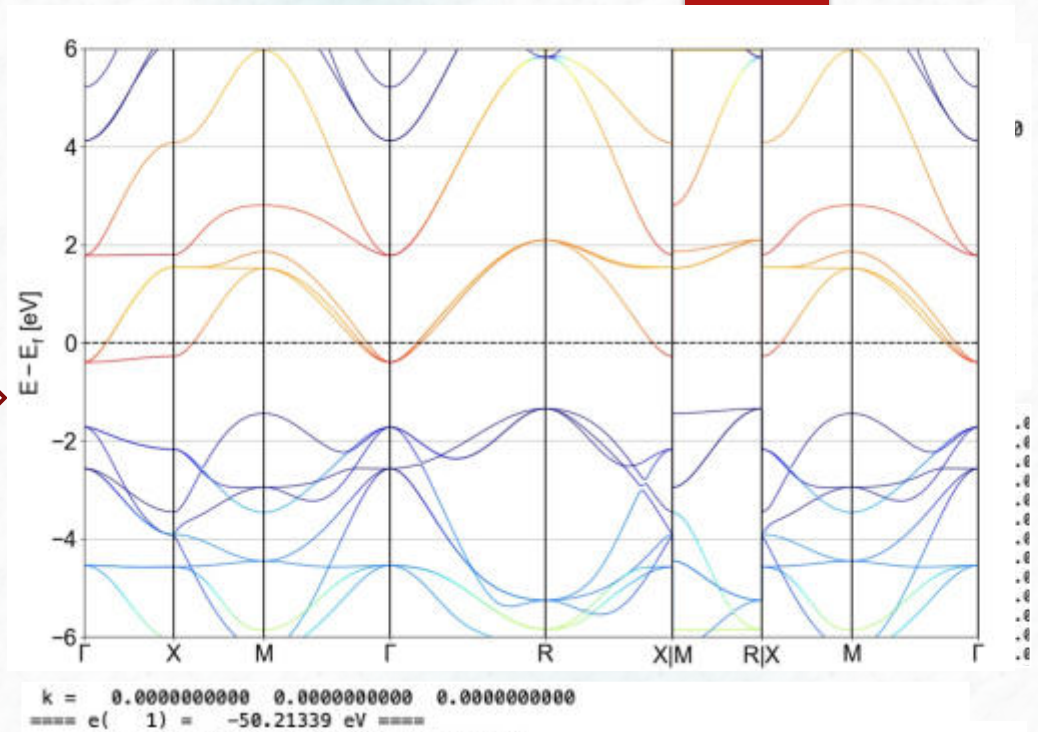
Herath, Uthpala, et al. "PyProcar: A Python library for electronic structure pre/post-processing." *Computer Physics Communications* 251 (2020): 107080.

Pyprocar



DFT

Numerical data



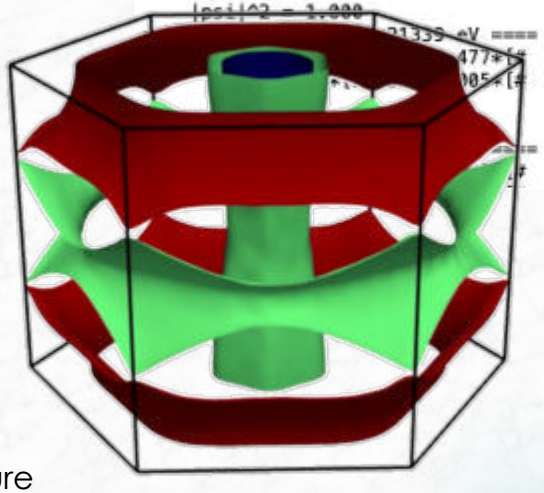
Li3Mn1O5

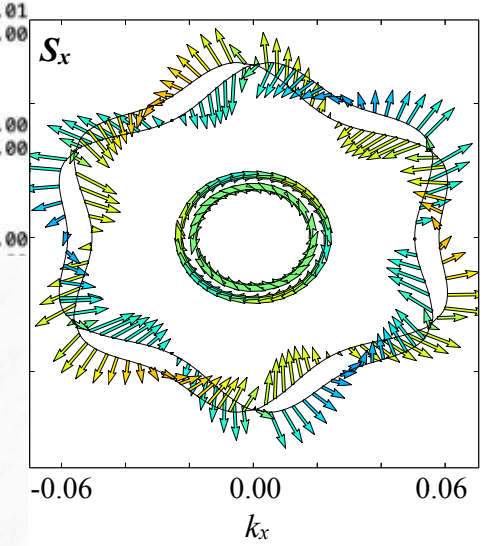
- Simple, single line commands

```
$pyprocar.bandsplot('PROCAR',mode='parametric',
orbitals=[4,5,6],code='vasp'…)
```

- Supports multiple DFT codes

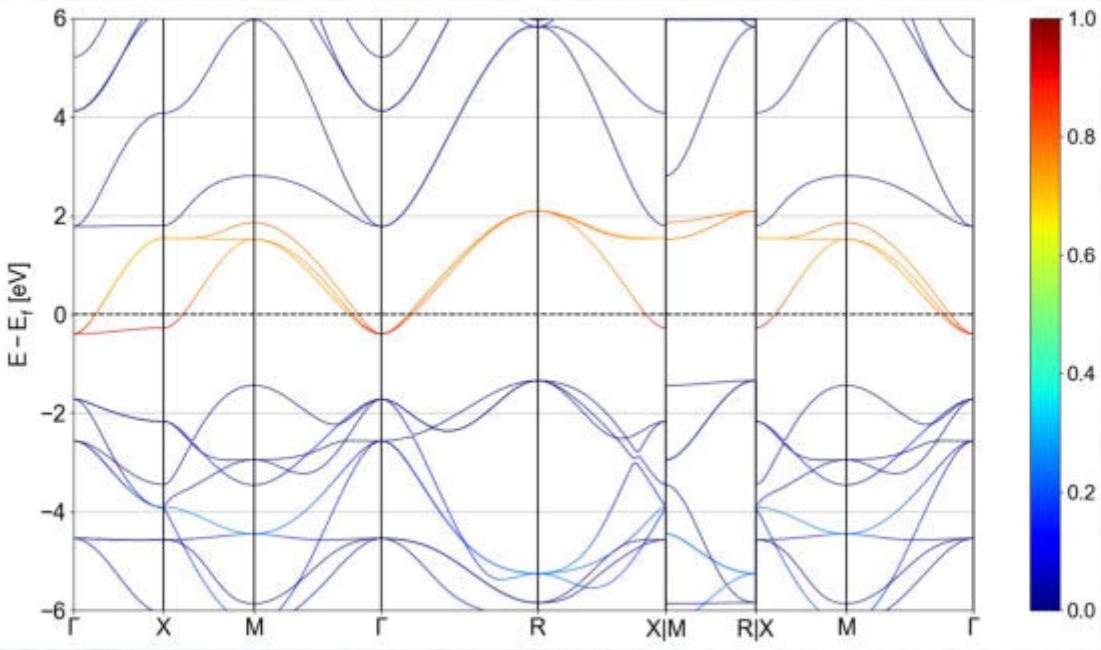
VASP, Quantum Espresso, Abinit, Elk, Lobster
Siesta*, DFTB+*





Herath, Uthpala, et al. "PyProcar: A Python library for electronic structure pre/post-processing." *Computer Physics Communications* 251 (2020): 107080.

Plotting band structures



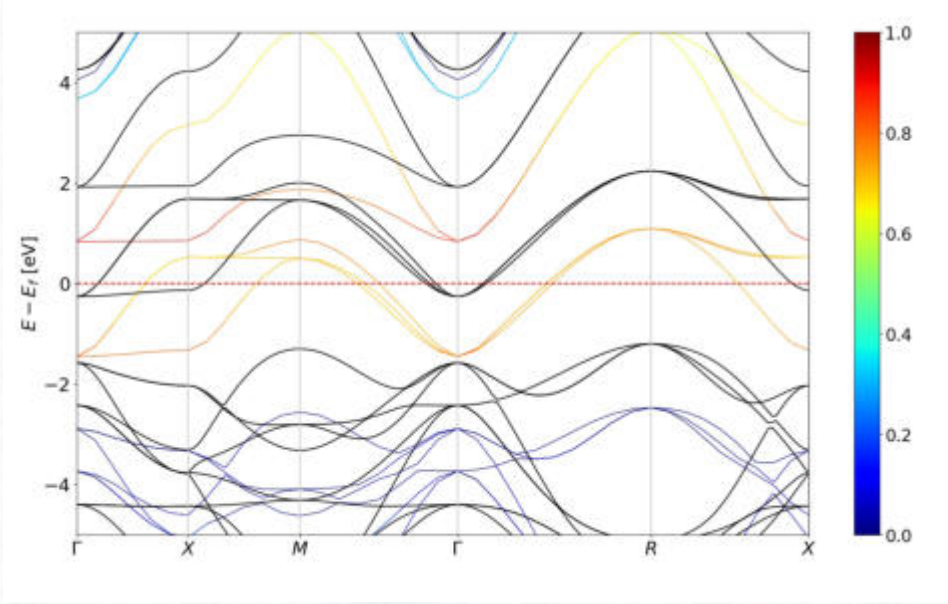
Perform a combination of one or more projections.

V – d_{t2g}

```
pyprocar.bandsplot(..,mode='parametric',atoms=[1],
                                orbitals=[4,5,7])
```

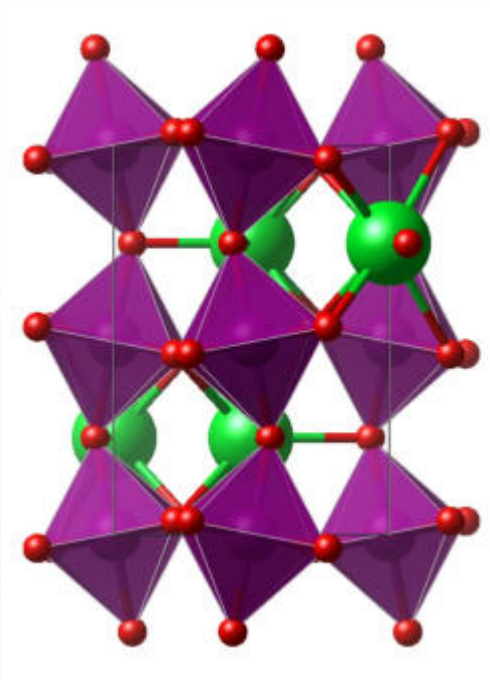
Export plots as matplotlib objects for further processing

```
ax=pyprocar.bandsplot(..,mode='plain',code='elk')
pyprocar.bandsplot(..,mode='parametric',code='vasp',ax=ax)
```

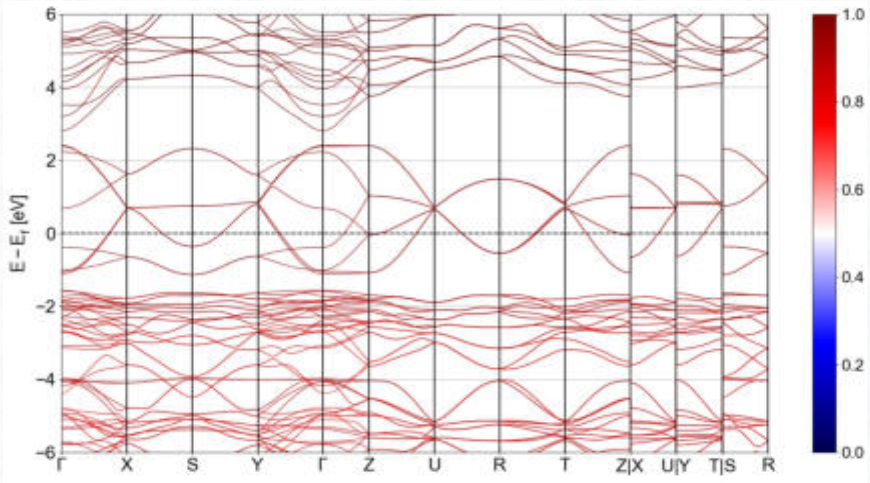


Plotting band structures

Collinear spin calculations

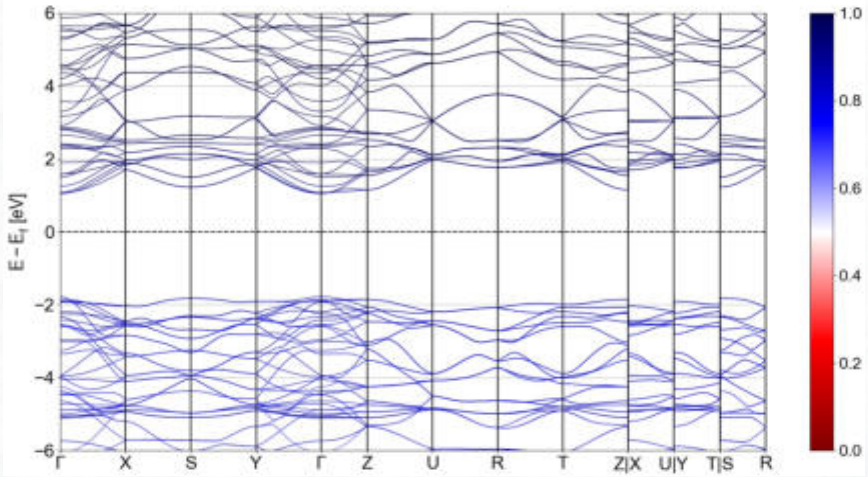


$La_{0.7}Sr_{0.3}MnO_3$



Spin up

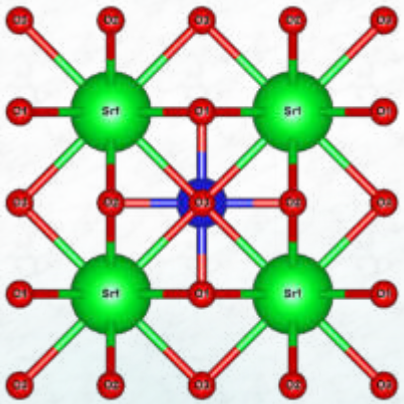
```
pyprocar.bandsplot(.., spin=1 , separate=True)
```

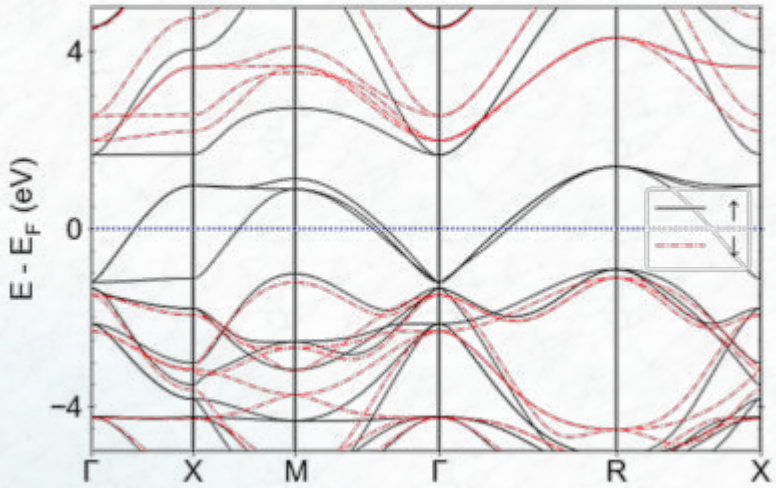


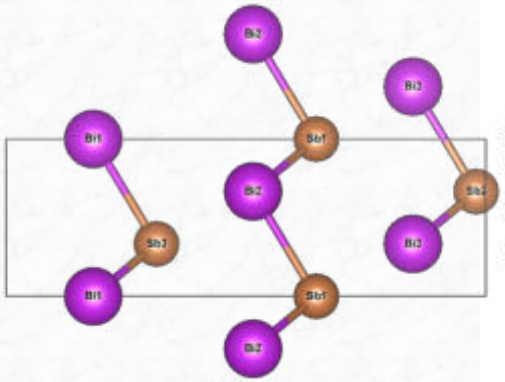
Spin down

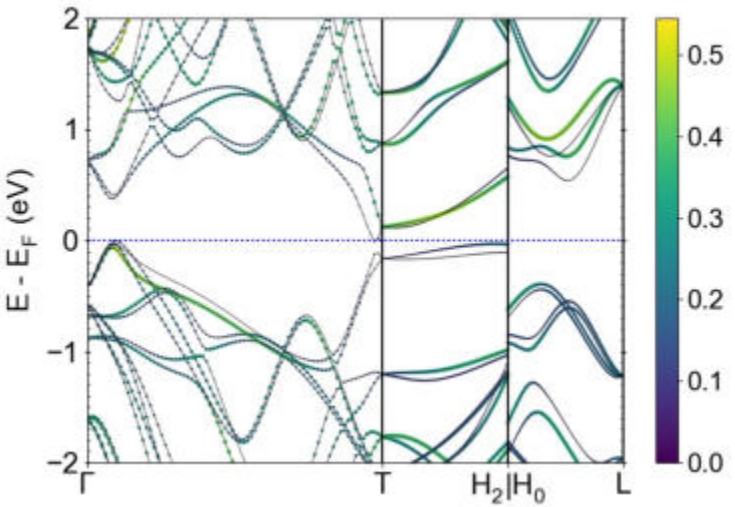
Plotting band structures

scatter + plain







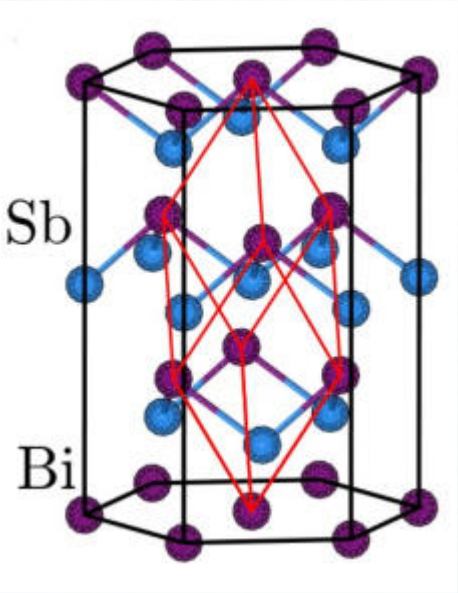



```
pyprocar.bandsplot(mode='plain’,
     linestyles=['solid', 'dashdot’],
     colors=["black", "red"],
     elimit=[-5, 5],
     savefig="SrVO3-spin-pol.png",
     show=True,
     legend=True,
     )
```

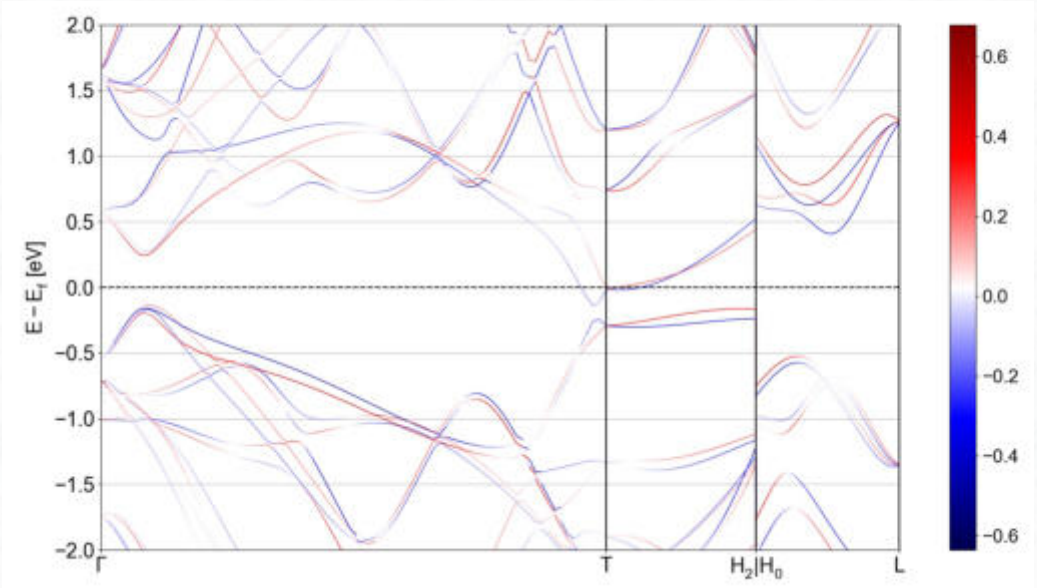
```
ebs_plot = pyprocar.bandsplot(mode='scatter',
                              legend=False,
                              spins=[0, 1],
                              orbitals=['p'],
                              atoms=['Bi'],
                              weighted_color=True,
                              weighted_width=True,
                              elimit=[-2,2]
                              )
pyprocar.bandsplot(mode='plain',
                              legend=False,
                              opacities=[0.5],
                              elimit=[-2,2],
                              ax=ebs_plot.ax)
```

Plotting band structures

Non-collinear spin calculations

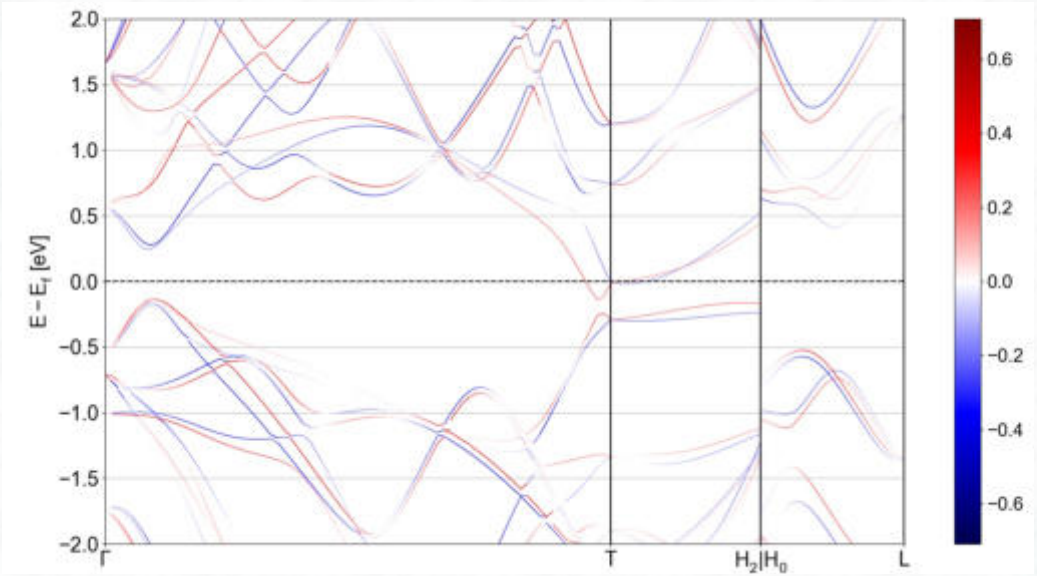


BiSb



S_x

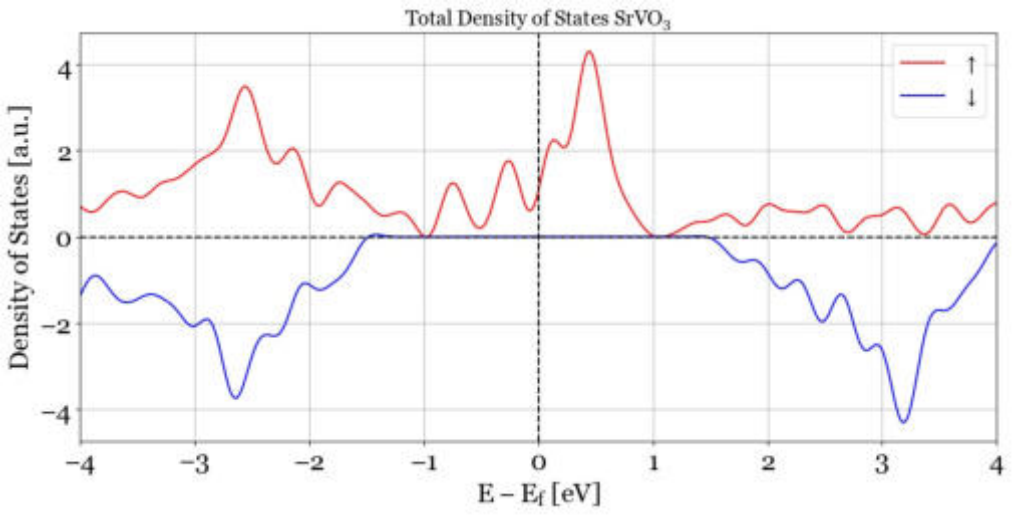
```
pyprocar.bandsplot(.., spin=1)
```



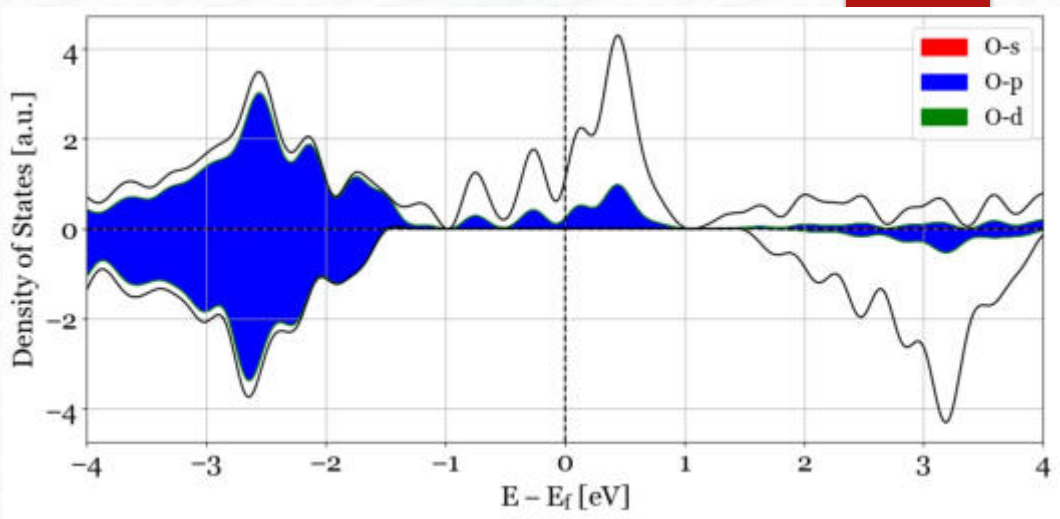
S_y

```
pyprocar.bandsplot(.., spin=2)
```

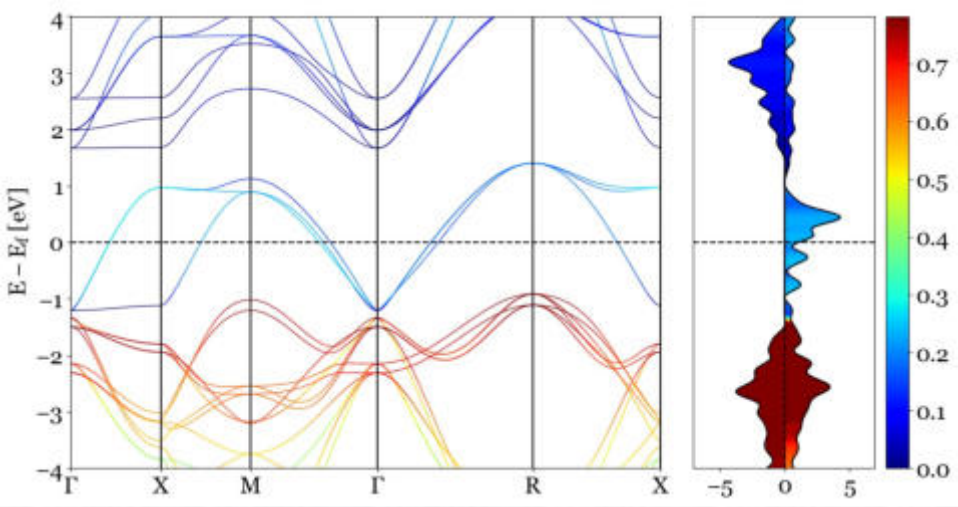
Density of States




```
pyprocar.dosplot(mode='plain')
```




```
pyprocar.dosplot(mode='stack_orbitals')
```



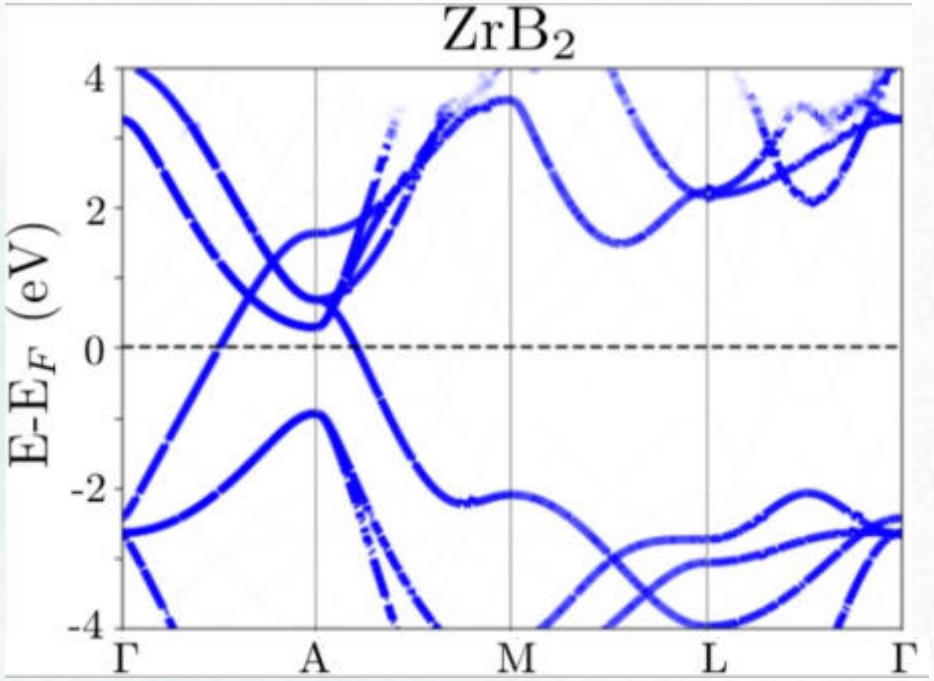
Partial Density of States + Band structure

```
pyprocar.bandsdosplot(vasprunfile='vasprun.xml',
bands_file='PROCAR',bands_mode='parametric',dos_mode='parametric,'
orbitals=[1,2,3],atoms=[2,3,4],..)
```

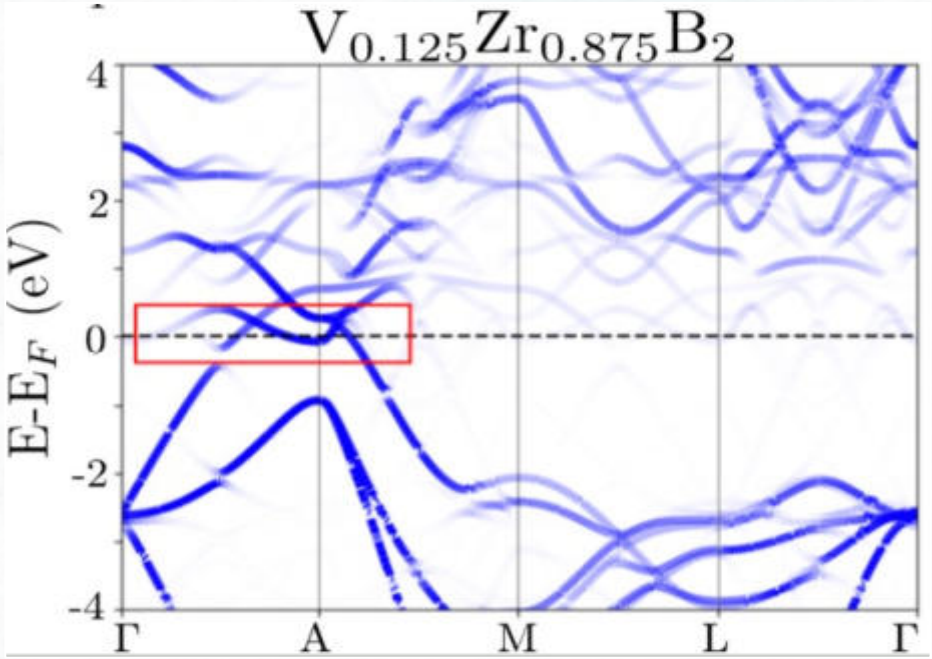
Band unfolding

- Useful to unfold bands from supercells to compare with primitive cell.

E.g. - 10% of V doping can induce superconductivity in ZrB2. A 2 x 2 x 2 supercell was used to see the changes in the electronic structure.



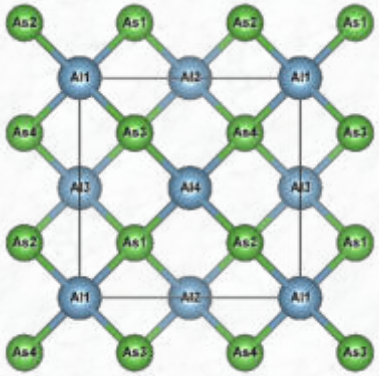
Primitive cell band structure



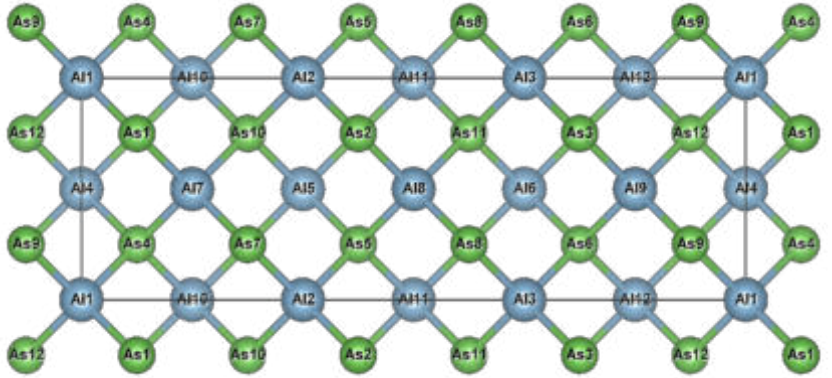
Unfolded band structure of supercell

```
pyprocar.unfold( fname='PROCAR', supercell_matrix=np.diag([2, 2, 2]), ispin=None, show_band=True, width=4, ...)
```

```
[[-1, 1, 1],
 [1, -1, 1],
 [1, 1, -1]]
```



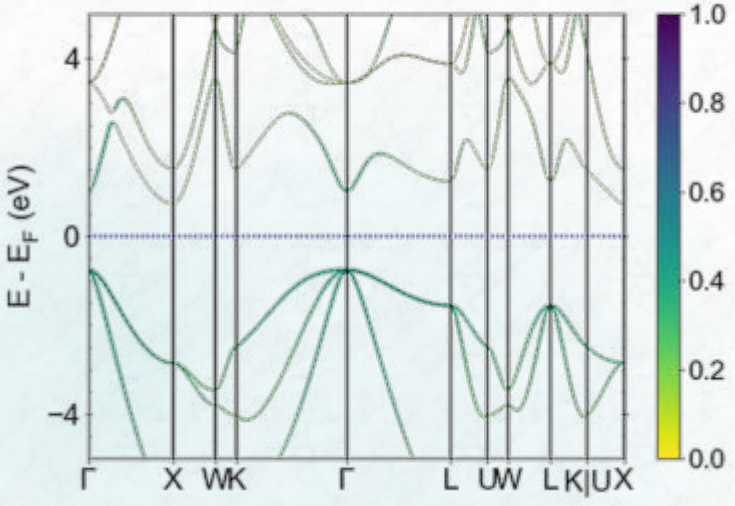
```
[[1, 0, 0],
 [0, 1, 0],
 [0, 0, 3]]
```

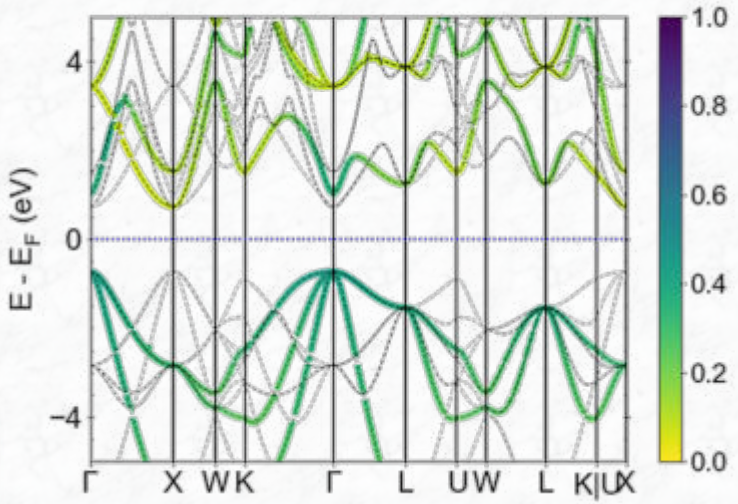


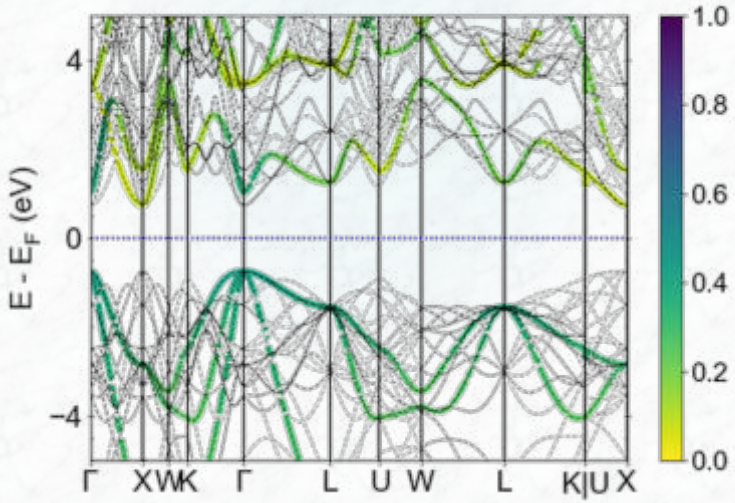
Primitive

Conventional

Conventional 1x1x3



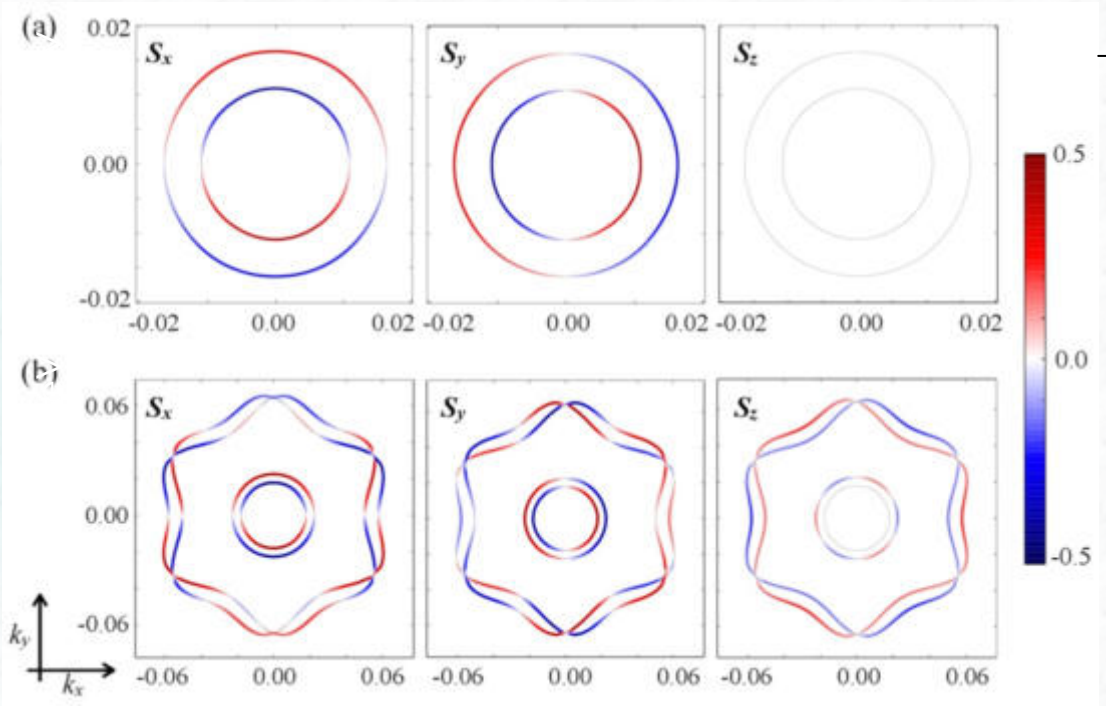




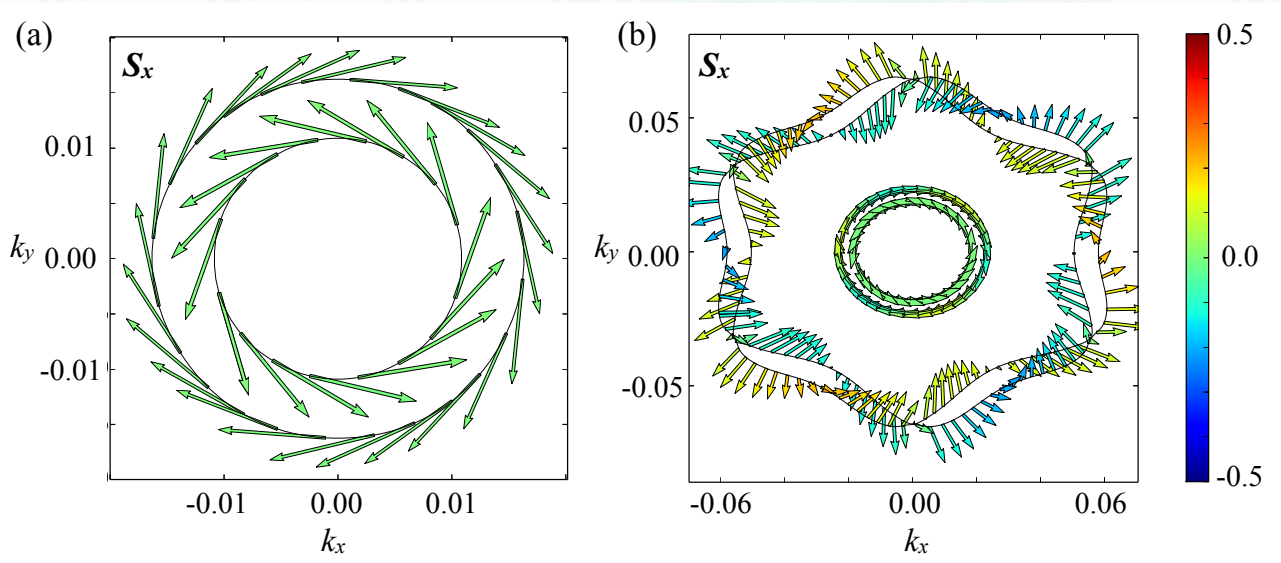

```
ebs_plot = pyprocar.bandsplot(procar="PROCAR", poscar="POSCAR", outcar="OUTCAR", kpoints="KPOINTS",
                              mode='parametric',
                              atoms=['As'],
                              unfold_mode=unfolding_mode,
                              transformation_matrix= np.dot(np.diag([1, 1, 3]),M),
                              weighted_width=True,
```

2D spin texture

- Visualize the constant energy surface spin textures in a system.
- Spin texture in BiSb monolayer. Useful to investigate Rashba and Dresselhaus type spin-splitting effects.



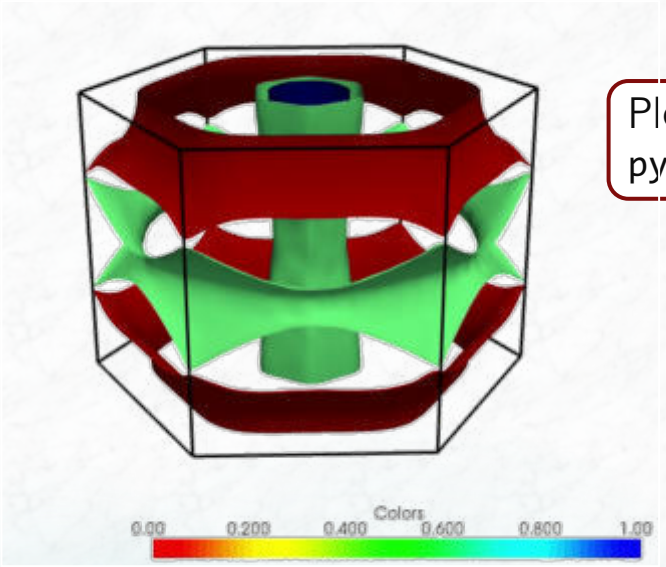
Represent with arrows




```
pyprocar.fermi2D('PROCAR',st=True,energy=1,
spin=1,noarrow=True)
```

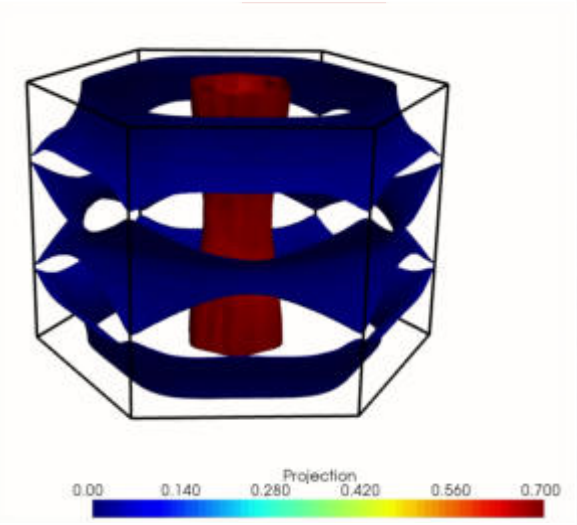
```
pyprocar.fermi2D('PROCAR',st=True,energy=1,
spin=1,noarrow=False)
```

3D Fermi surface



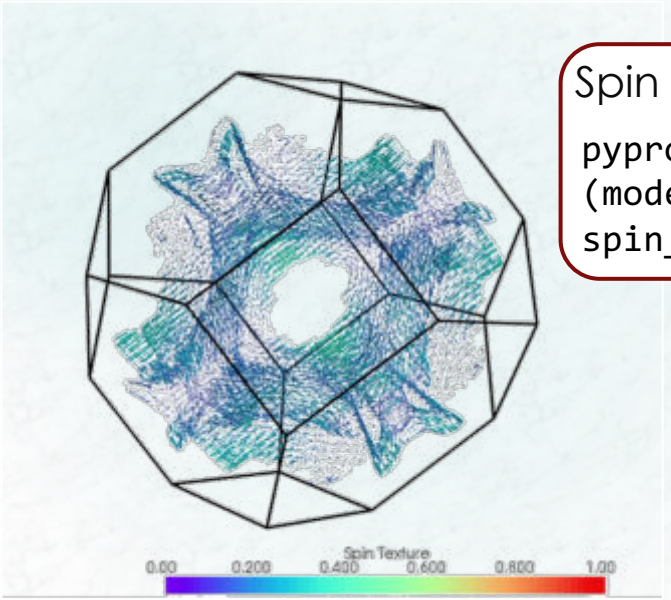
Plain Fermi surface – MgB_2

```
pyprocar.fermi3D(mode='plain'...)
```



Orbital projected Fermi surface- MgB_2: B-p_z

```
pyprocar.fermi3D(mode='parametric',
orbitals=[3],atoms=[1,2])
```

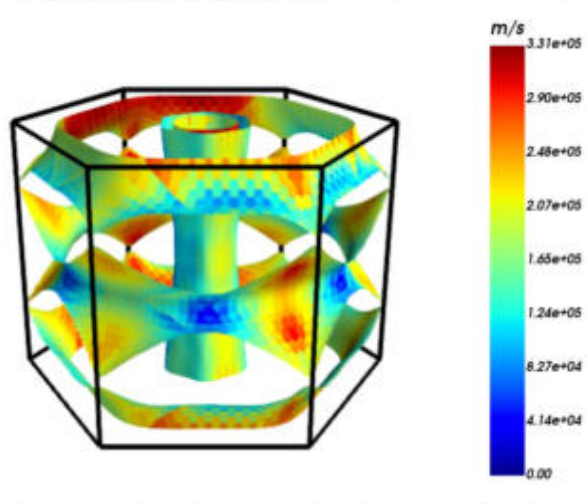


Spin texture surface- BiSb

```
pyprocar.fermi3D
(mode='parametric',
spin_texture=True)
```

Plotting from external data: Fermi velocity projection

```
pyprocar.fermi3D(mode='external'
,color_file='data.dat'...)
```

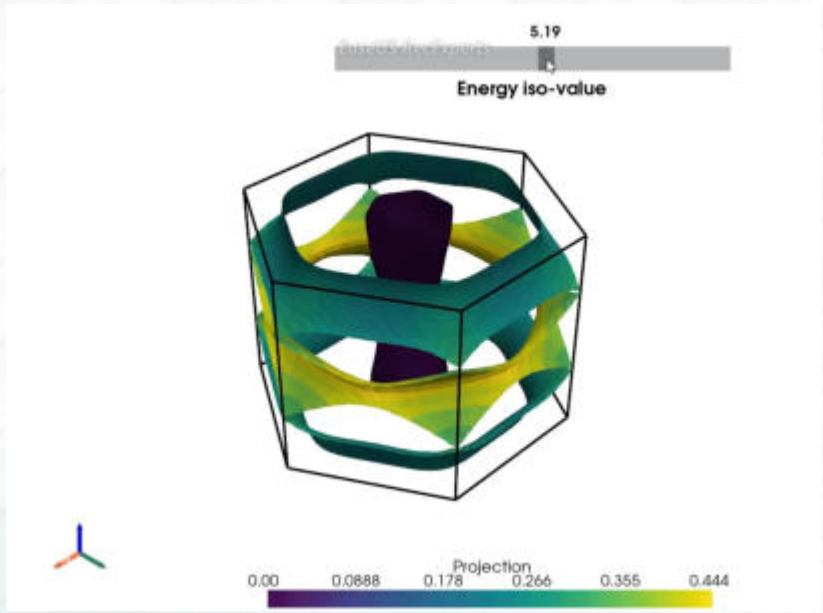


other: effective mass, electron-phonon coupling

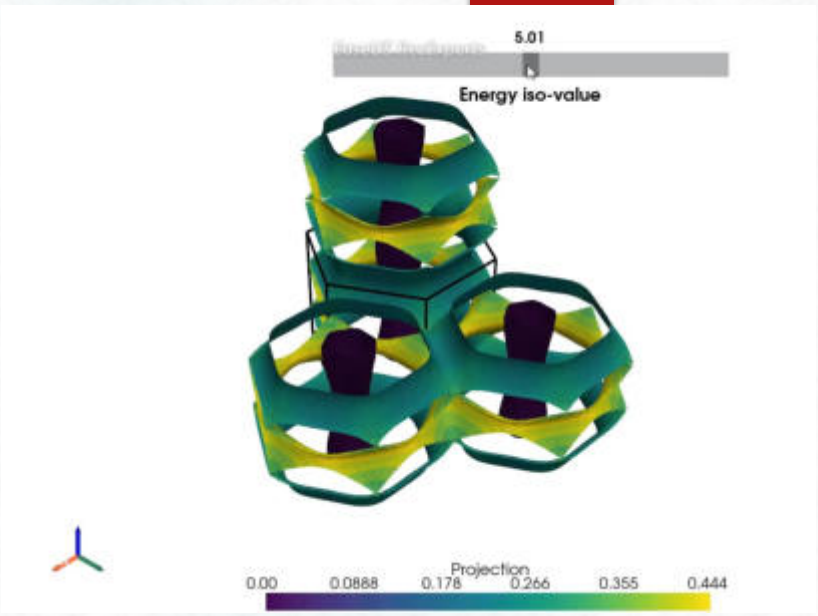
Input formats
- PROCAR format
- bxsf format

Export formats
- 2D : png, pdf, ps, …
- 3D : obj, ply, glb, xml…
- other : gif, mp4

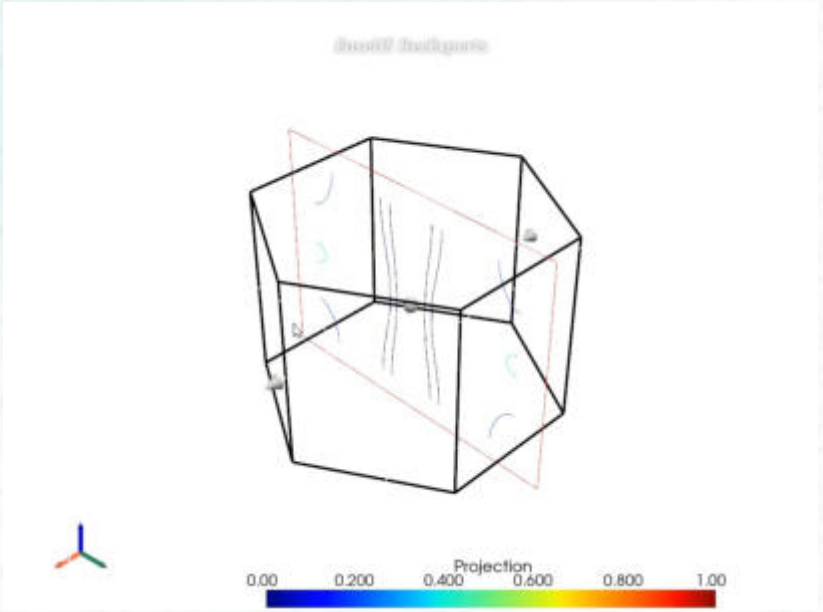
3D Fermi surface



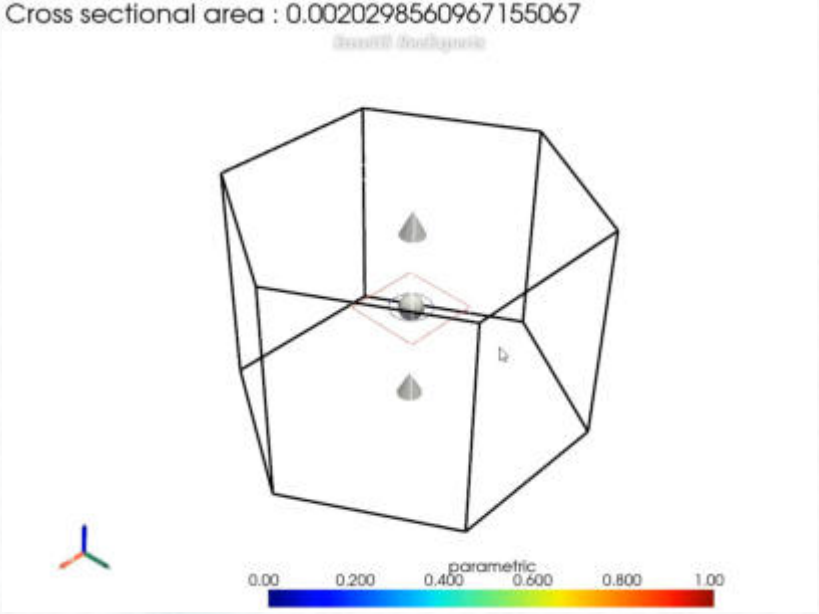

```
pyprocar.fermi3D(...,
                 iso_slider = True,
                 iso_range = 5,
                 iso_surfaces = 5,)
```




```
pyprocar.fermi3D(...,
extended_zone_directions=
[[1,0,0],[0,1,0],[0,0,1]])
```




```
pyprocar.fermi3D(...,
show_slice=True,
slice_origin = (0,0,0)
slice_normal = (1,0,0))
```




```
pyprocar.fermi3D(…
                 bands = [5],
                 show_slice=True,
                 slice_normal = (0,0,1),
                 slice_origin = (0,0,0),
show_cross_section_area= True)
```

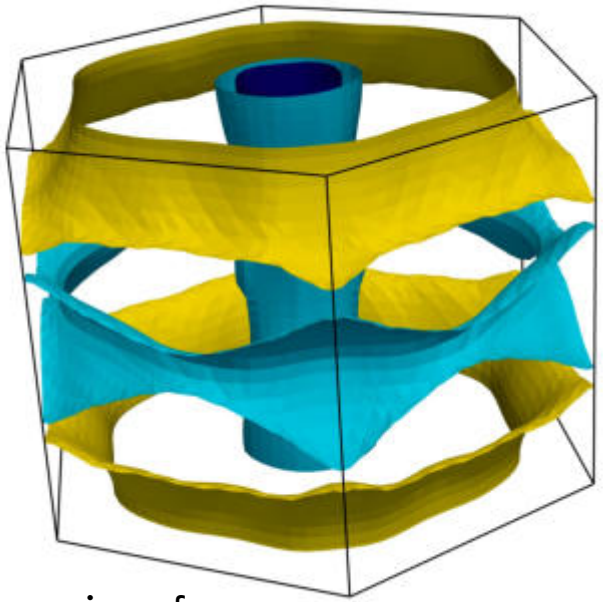
Fermi Surface - Abinit

```
pyprocar.fermi3D(
    “PROCAR”,
    abinit_output=“abinit.out”,
    code="abinit",
    mode="plain",
    interpolation_factor=4,
```

With Abinit v9+ PROCAR is generated when prtprocar is set in the .in file.

However, due to parallelization, the PROCAR is split into multiple files. To fix this:

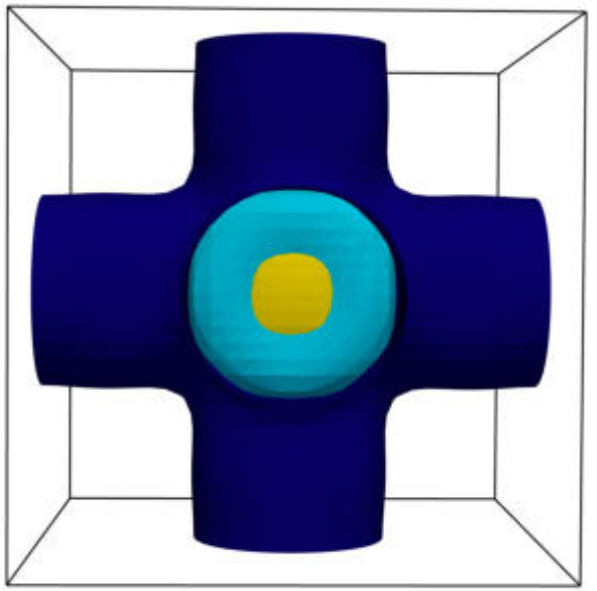
Pyprocar.cat(mergeparallel=True, code=“abinit”) in the directory where the PROCAR_ files are. In a future release this step will be automated.



MgB2 Fermi surface

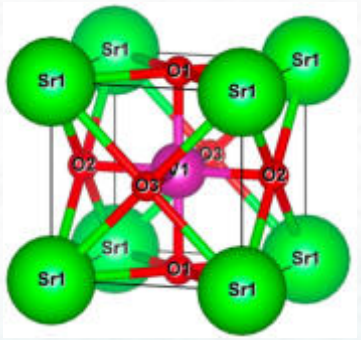
Another option:

Subtract Fermi surface of different spin polarizations



SrVO3 Fermi surface

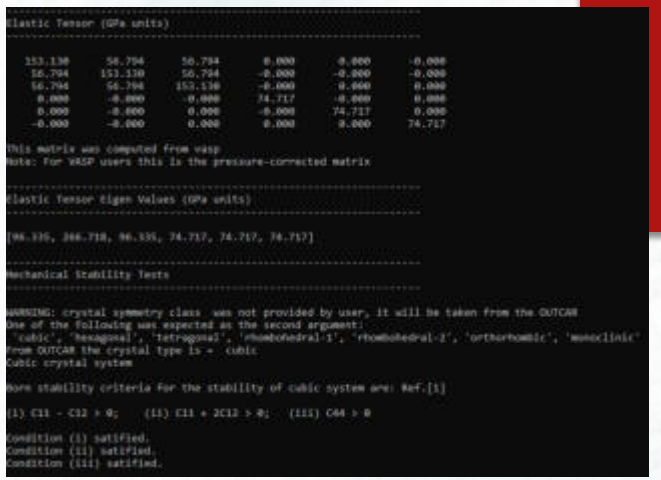
MechElastic

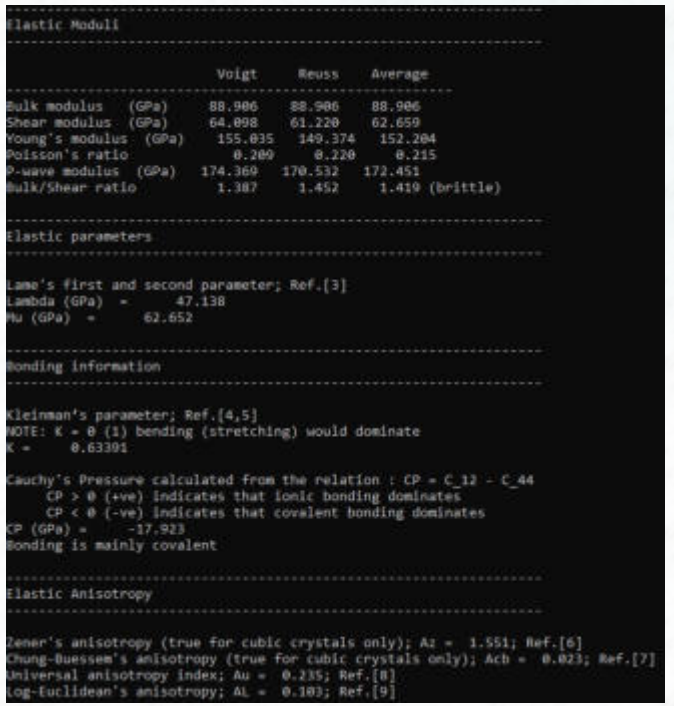


DFT

Quantum Espresso

Numerical data





- Simple, single line commands

```
mechelastic.calculate_elastic(code="vasp",
                              dim="3D",
                              infile="OUTCAR-Si_bulk")
```

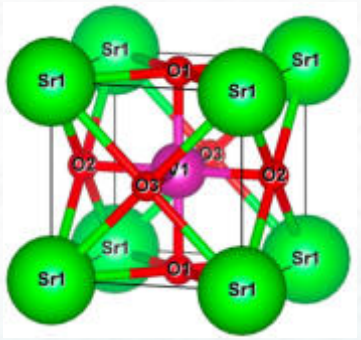
- Supports multiple DFT codes

  `VASP, Quantum Espresso, Abinit,`

- Elastic constants and EOS

Singh, Sobhit, et al. "MechElastic: A Python Library for Analysis of Mechanical and Elastic Properties of Bulk and 2D Materials." *arXiv preprint arXiv:2012.04758* (2020).

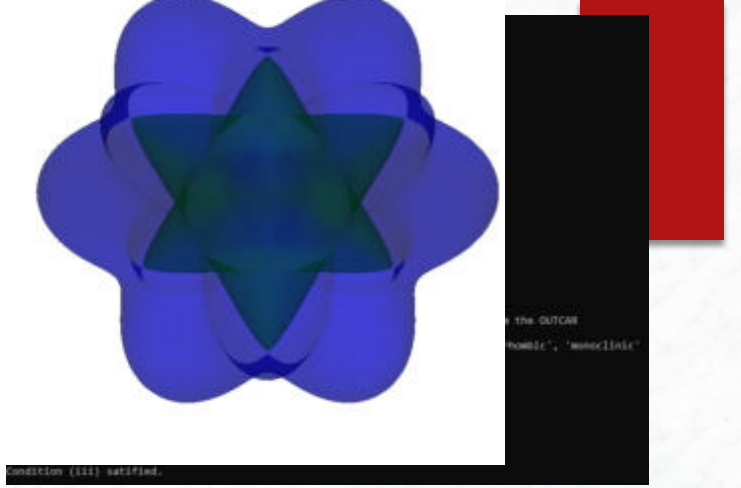
MechElastic

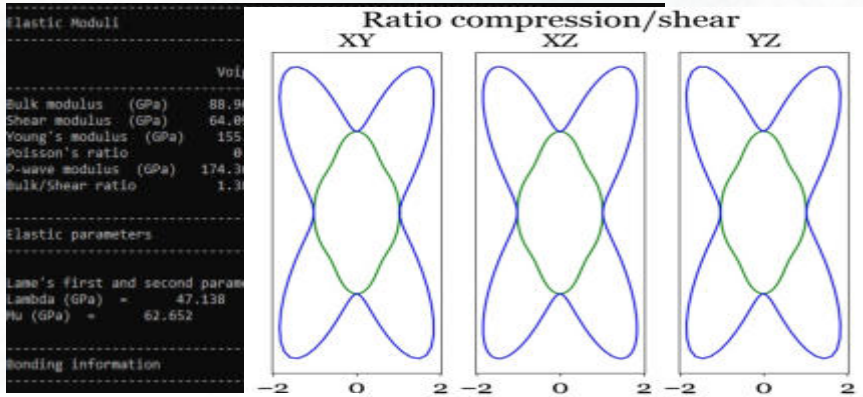


DFT

Quantum Espresso

Numerical data





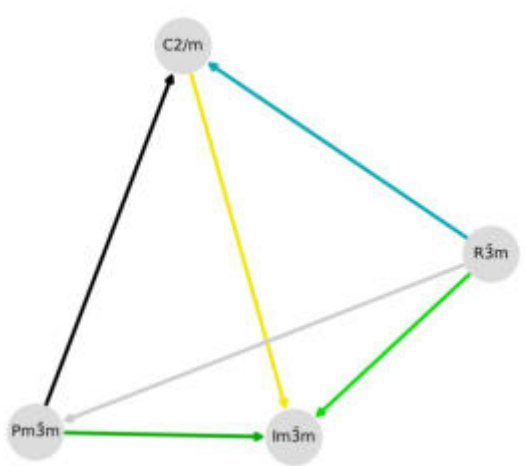
- Simple, single line commands

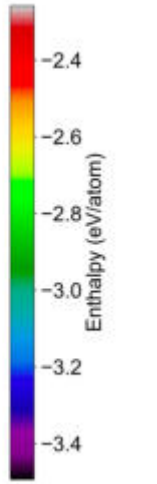
```
mechelastic.calculate_elastic(code="vasp",
                              dim="3D",
                              infile="OUTCAR-Si_bulk")
```

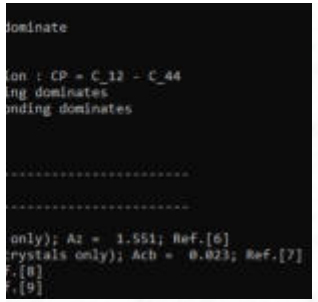
- Supports multiple DFT codes

  `VASP, Quantum Espresso, Abinit,`

- Elastic constants and EOS







Singh, Sobhit, et al. "MechElastic: A Python Library for Analysis of Mechanical and Elastic Properties of Bulk and 2D Materials." *arXiv preprint arXiv:2012.04758* (2020).

MechElastic 3D Analysis

```
mechelastic.calculate_elastic(code="vasp",
                              dim="3D",
                              infile="OUTCAR-Si_bulk")
```

```
-------------------------------------------------------------
Elastic Tensor (GPa units)
-------------------------------------------------------------

   153.130      56.794      56.794       0.000       0.000      -0.000
    56.794     153.130      56.794      -0.000      -0.000      -0.000
    56.794      56.794     153.130      -0.000       0.000       0.000
     0.000      -0.000      -0.000      74.717      -0.000       0.000
     0.000      -0.000       0.000      -0.000      74.717       0.000
    -0.000      -0.000       0.000       0.000       0.000      74.717

This matrix was computed from vasp
Note: For VASP users this is the pressure-corrected matrix

-------------------------------------------------------------
Elastic Tensor Eigen Values (GPa units)
-------------------------------------------------------------

[96.335, 266.718, 96.335, 74.717, 74.717, 74.717]

-------------------------------------------------------------
Mechanical Stability Tests
-------------------------------------------------------------

WARNING: crystal symmetry class  was not provided by user, it will be taken from the OUTCAR
One of the following was expected as the second argument:
 'cubic', 'hexagonal', 'tetragonal', 'rhombohedral-1', 'rhombohedral-2', 'orthorhombic', 'monoclinic'
From OUTCAR the crystal type is =  cubic
Cubic crystal system

Born stability criteria for the stability of cubic system are: Ref.[1]

(i) C11 - C12 > 0;    (ii) C11 + 2C12 > 0;   (iii) C44 > 0

Condition (i) satified.
Condition (ii) satified.
Condition (iii) satified.
```

MechElastic 3D Analysis

```
---------------------------------------------------------------
Elastic Moduli
---------------------------------------------------------------

                        Voigt     Reuss    Average
-------------------------------------------------------
Bulk modulus   (GPa)    88.906    88.906    88.906
Shear modulus  (GPa)    64.098    61.220    62.659
Young's modulus  (GPa)   155.035   149.374   152.204
Poisson's ratio            0.209     0.220     0.215
P-wave modulus  (GPa)   174.369   170.532   172.451
Bulk/Shear ratio           1.387     1.452     1.419 (brittle)
```

```
---------------------------------------------------------------
Elastic parameters
---------------------------------------------------------------

Lame's first and second parameter; Ref.[3]
Lambda (GPa)  =      47.138
Mu (GPa)  =      62.652
```

```
---------------------------------------------------------------
Bonding information
---------------------------------------------------------------

Kleinman's parameter; Ref.[4,5]
NOTE: K = 0 (1) bending (stretching) would dominate
K =     0.63391

Cauchy's Pressure calculated from the relation : CP = C_12 - C_44
     CP > 0 (+ve) indicates that ionic bonding dominates
     CP < 0 (-ve) indicates that covalent bonding dominates
CP (GPa) =     -17.923
Bonding is mainly covalent
```

```
---------------------------------------------------------------
Elastic Anisotropy
---------------------------------------------------------------

Zener's anisotropy (true for cubic crystals only); Az =  1.551; Ref.[6]
Chung-Buessem's anisotropy (true for cubic crystals only); Acb =  0.023; Ref.[7]
Universal anisotropy index; Au =  0.235; Ref.[8]
Log-Euclidean's anisotropy; AL =  0.103; Ref.[9]
```

```
---------------------------------------------------------------
Elastic Wave Velocities and Debye Temperature
---------------------------------------------------------------

Longitudinal wave velocity (vl) :   8694.638 m/s; Ref.[10]
Transverse wave velocity (vt) :   5240.943 m/s; Ref.[10]
Average wave velocity (vm) :   5795.129 m/s; Ref.[10]
Debye temperature  :    630.982 K; Ref.[10]

WARNING: The  Debye model for the atomic displacement is based on a monoatomic crystal approximation.
 Here we consider an averaged mass, in case your crystal has several species.
```

```
---------------------------------------------------------------
Melting Temperature
---------------------------------------------------------------

Melting temperature calculated from the empirical relation: Tm = 607 + 9.3*Kvrh \pm 555 (in K); Ref.[11]
Tm =    1433.825 K (plus-minus 555 K)


WARNING: This is a crude approximation and its validity needs to be checked!
```

```
---------------------------------------------------------------
Hardness Analysis
---------------------------------------------------------------

Hardness (H1a) = 9.24 GPa;   Ref.[12]
Hardness (H1b) = 9.24 GPa;   Ref.[12]
Hardness (H2)  = 8.19 GPa;   Ref.[13]
Hardness (H3)  = 9.66 GPa;   Ref.[14]
Hardness (H4)  = 6.96 GPa;   Ref.[15]
Hardness (H5)  = 11.94 GPa;   Ref.[16]

Hardness recommendation model:
       ************************************************************************
                    Cubic  Hexagonal  Orthorhombic  Rhombohedral  General
       ************************************************************************
       Insulator      H2      H1b          H2            H2          H2
       Semiconductor  H5    H1b, H3                     H2          H5
       Metal          H1a     H4           H4            H4          H4
       ************************************************************************
       Insulator     : bandgap > 2 eV
       Semiconductor : bandgap < 2 eV
       Metal         : bandgap = 0
```

The MechElastic 2D Analysis

```
import mechelastic
mechelastic.calculate_elastic(code="vasp",
                              dim="2D",
                              infile="OUTCAR-graphene"
                              lattice_type = 'hexagonal')
```

```
 Elastic tensor for two-dimensional system in N/m units

  334.1501     89.8781      0.8525     -0.0015     -0.0169      0.0009
   89.8781    334.2872      0.8606      0.0018      0.0115      0.0013
    0.8525      0.8606     -0.7386      0.0170     -0.0437     -0.0004
   -0.0015      0.0018      0.0170     -1.6537      0.0009     -0.0073
   -0.0169      0.0115     -0.0437      0.0009     -0.4477     -0.0042
    0.0009      0.0013     -0.0004     -0.0073     -0.0042    122.0850
```

```
                Mechanical Stability Test

Hexagonal lattice

Stability criteria for the stability of hexagonal system are:

(i) C11 + C12 > 0;    (ii) C11 - C12 > 0;

Condition (i) satisfied.
Condition (ii) satisfied.
```

```
 Elastic properties in two-dimensions

[Useful refs. Andrew et al.;  Phys. Rev. B 85, 125428 (2012), Peng et al., Acta Mechanica 223 (2012), 2591-2596; Comput. Mater. Sci. 68, 320 (2013);  Mech. Mater. 64, 135 (2013)  ]

-----------------------------------------------------
2D layer modulus (N/m)          :      212.048
2D Young's modulus Y[10] (N/m) :      309.985
2D Young's modulus Y[01] (N/m) :      310.112
2D Shear modulus G (N/m)        :      122.085
2D Poisson ratio v[10]          :        0.269
2D Poisson ratio v[01]          :        0.269
-----------------------------------------------------
```

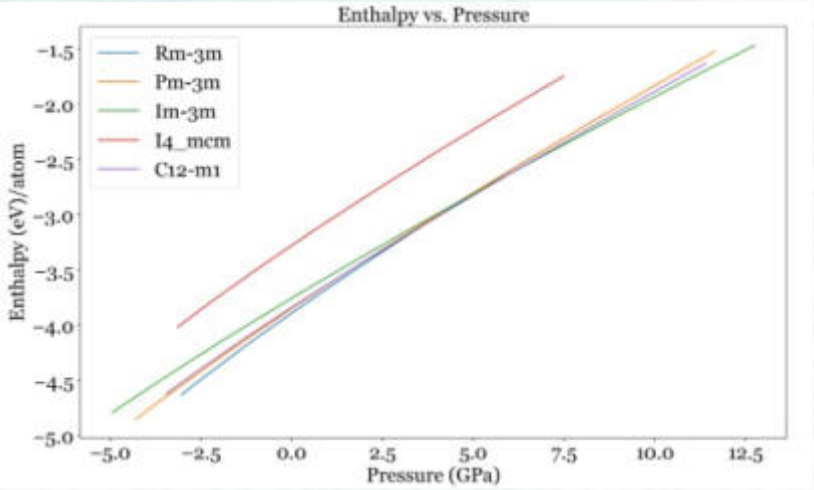
EOS module

Enthalpy Curves

H=U+PV

```
eos_object = EOS()

infiles = ["Rm-3m", "Pm-3m",
        "Im-3m", "I4_mcm",
        "C12-m1"]
natoms = [6, 1, 2, 9, 4]
eos_object.plot_enthalpy_curves(infiles,
eostype = 'energy',natoms, au=False)
```

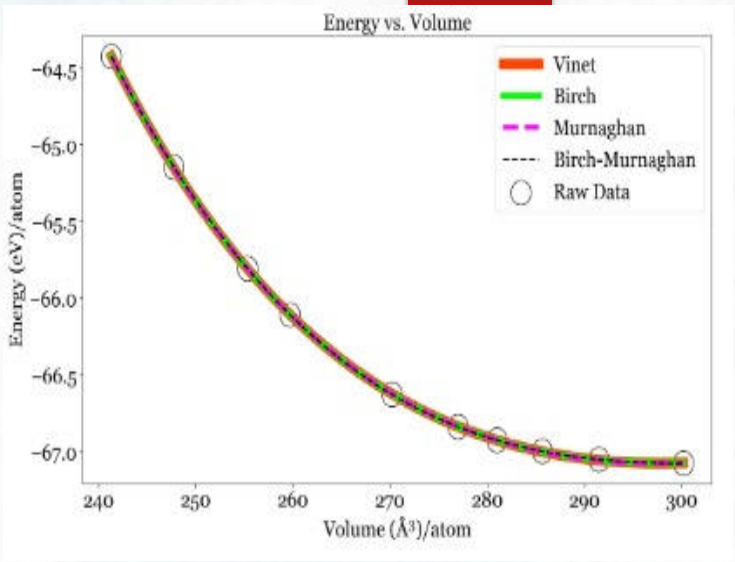


EvsV.dat

```
300.21 -67.07549985
291.50 -67.05105419
285.67 -66.99700837
281.02 -66.92530138
277.00 -66.83788211
270.22 -66.62790736
259.71 -66.10926114
255.34 -65.80844965
247.75 -65.14777164
241.31 -64.42659671
```

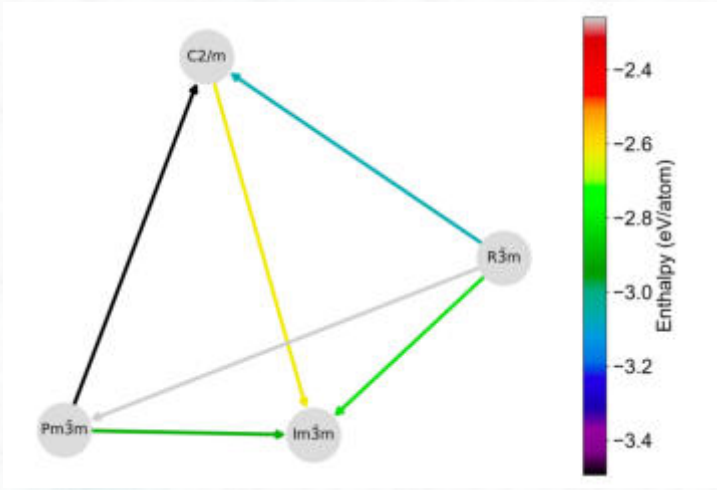
```
eos = EOS("EvsV.dat")
eos.plot_eos(au=False)
```

Curve Fitting

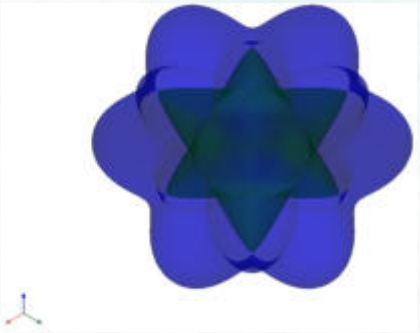


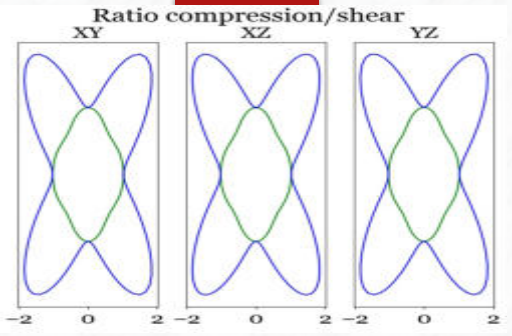
Phase transition paths

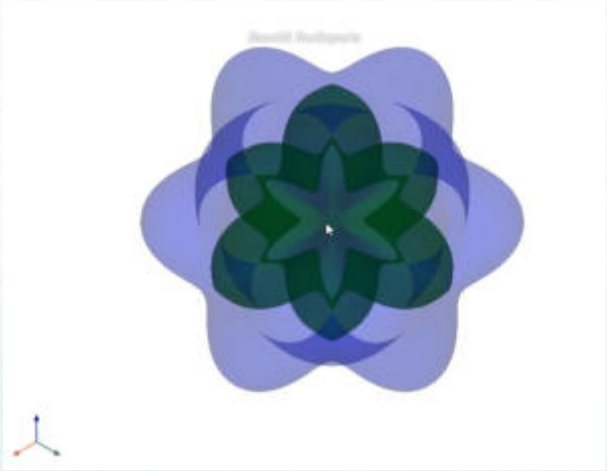
A network plot of the phase transition paths for different phases of Bi. The color of the arrows represents the relative enthalpy for different phase transitions.



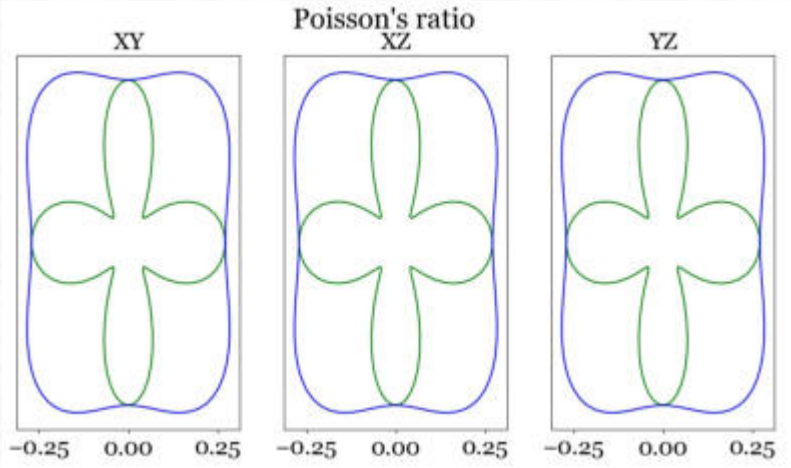
2D and 3D Visualizations of Elastic Properties



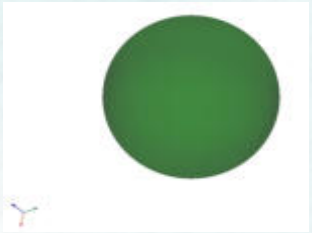


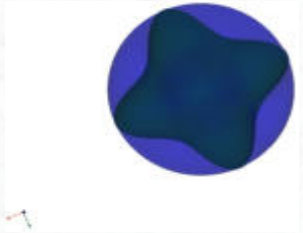


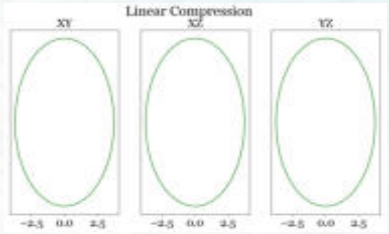
```
Output1,meshes= mechelastic.calculate_elastic(…,
                                              plot= ‘’3D”
                                              elastic_calc= “POISSON”
                                              show = True)
```

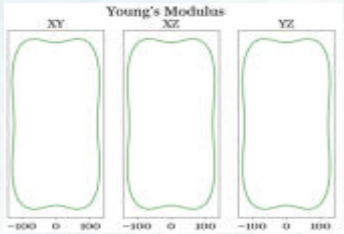


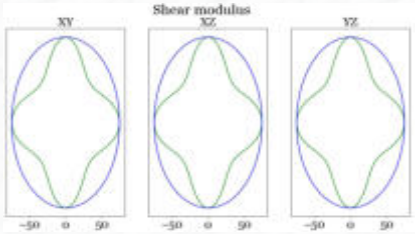

```
Output2,fig =  mechelastic.calculate_elastic(… ,
                                              plot= ‘’2D”
                                              elastic_calc= “POISSON”
                                              show=True )
```

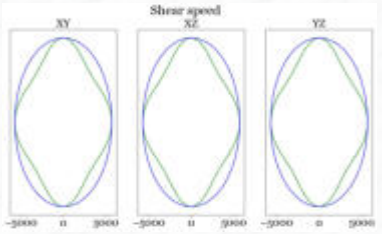


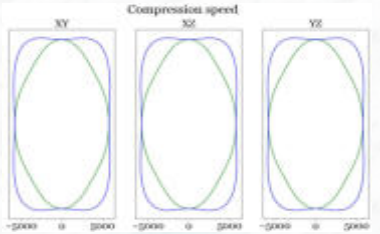


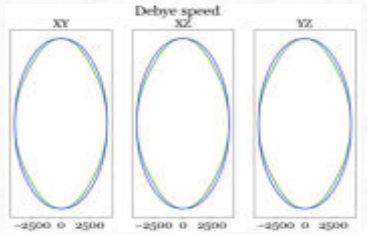












Perspectives

Development of two packages to perform electronic structure and elastic analysis of electronic structure calculations.

Perspectives:

Improve the Fermi surface topology.

Correlate Fermi area cross section with the De Haas-van Alphen Effect (Verstraete).

Compute the Gyrotropic Magnetic Effect (we need extra terms, Fermi curvature).

Extend the package to perform Wavefunction or density analysis.

Some ideas from the community are also considered.