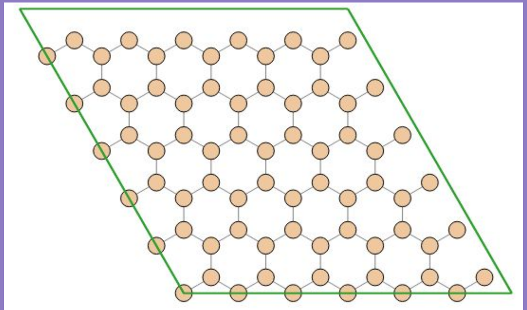
Projecte final:
Diagrama de bandes del silicà

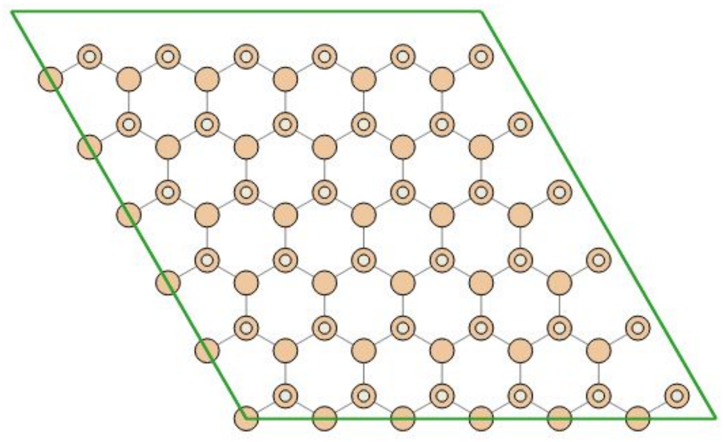
Aurembiaix Sabadell i Mercader

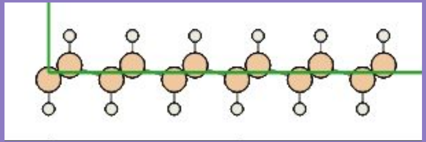
1. Introducció

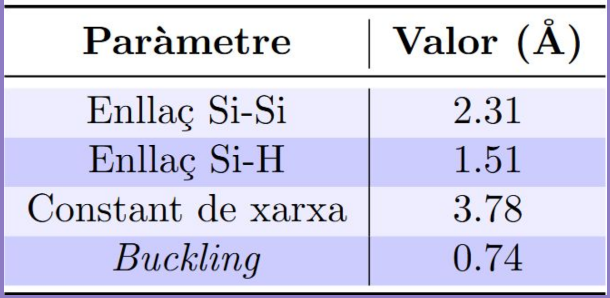
SILICÈ



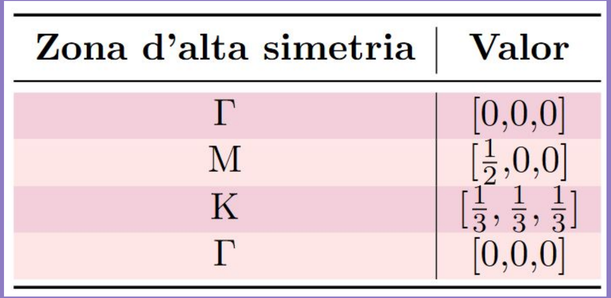
SILICÀ







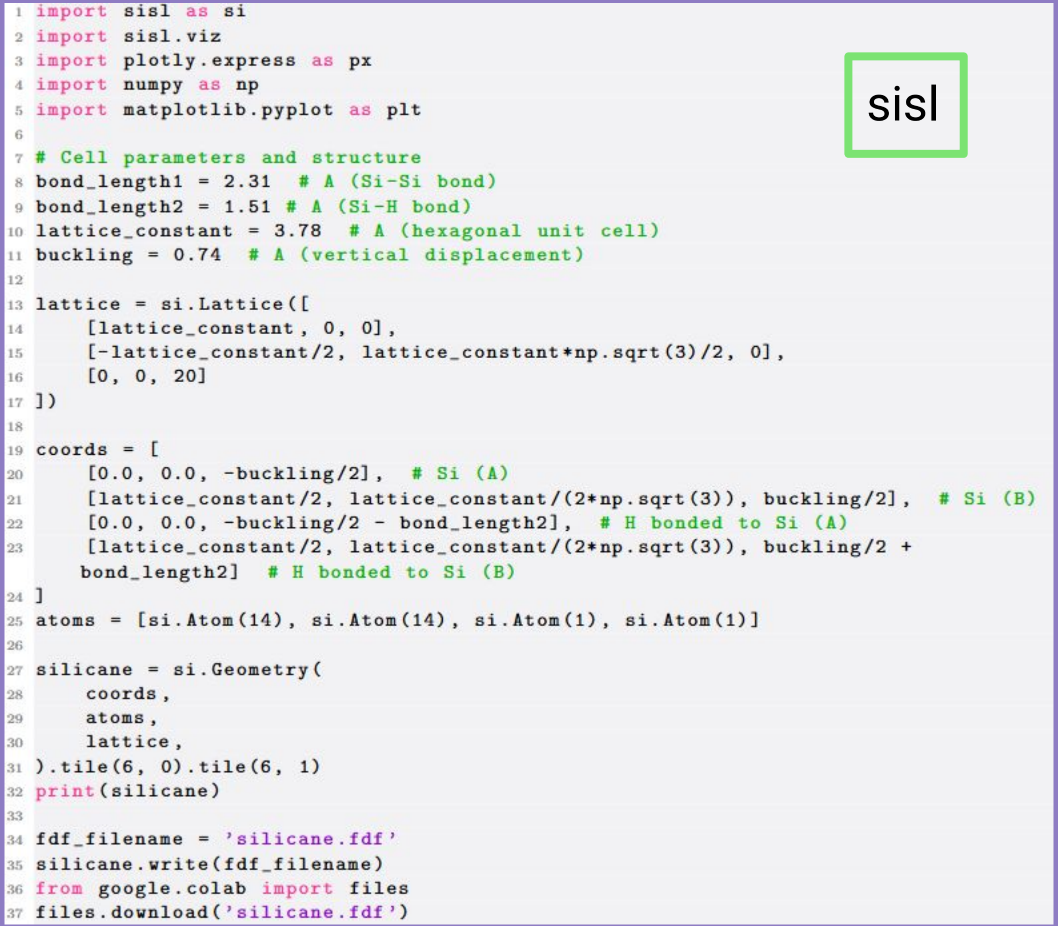
Paràmetre	Valor (Å)
Enllaç Si-Si	2.31
Enllaç Si-H	1.51
Constant de xarxa	3.78
Buckling	0.74



Zona d'alta simetria	Valor
Γ	[0,0,0]
M	$[\frac{1}{2}$,0,0]
K	$[\frac{1}{3}, \frac{1}{3}, \frac{1}{3}]$
Γ	[0,0,0]

Spin Orbit Coupling

2. Procediment

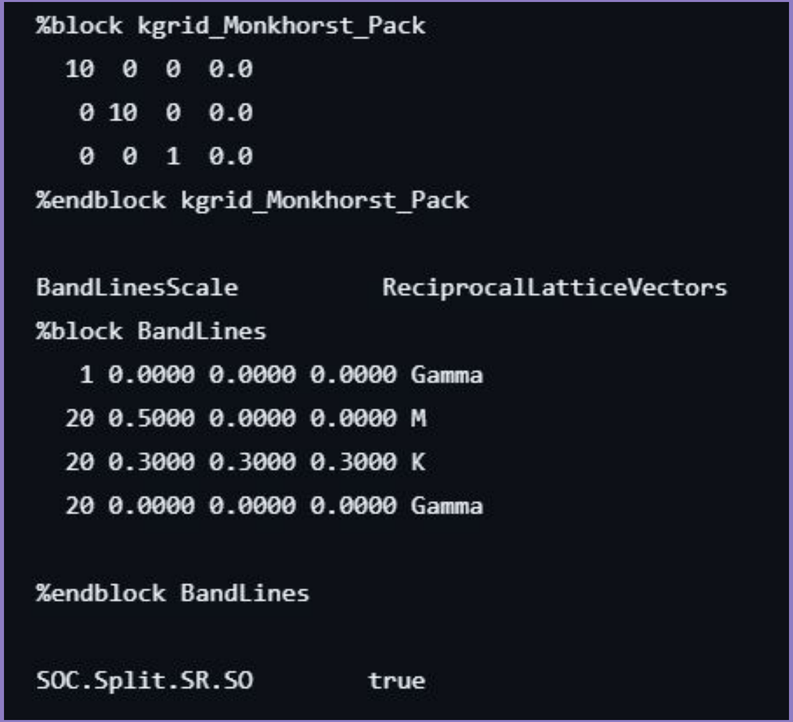


sisl

```python
import sisl as si
import sisl.viz
import plotly.express as px
import numpy as np
import matplotlib.pyplot as plt

# Cell parameters and structure
bond_length1 = 2.31   # A (Si-Si bond)
bond_length2 = 1.51 # A (Si-H bond)
lattice_constant = 3.78   # A (hexagonal unit cell)
buckling = 0.74   # A (vertical displacement)

lattice = si.Lattice([
    [lattice_constant, 0, 0],
    [-lattice_constant/2, lattice_constant*np.sqrt(3)/2, 0],
    [0, 0, 20]
])

coords = [
    [0.0, 0.0, -buckling/2],  # Si (A)
    [lattice_constant/2, lattice_constant/(2*np.sqrt(3)), buckling/2],  # Si (B)
    [0.0, 0.0, -buckling/2 - bond_length2],  # H bonded to Si (A)
    [lattice_constant/2, lattice_constant/(2*np.sqrt(3)), buckling/2 +
   bond_length2]  # H bonded to Si (B)
]
atoms = [si.Atom(14), si.Atom(14), si.Atom(1), si.Atom(1)]

silicane = si.Geometry(
    coords,
    atoms,
    lattice,
).tile(6, 0).tile(6, 1)
print(silicane)

fdf_filename = 'silicane.fdf'
silicane.write(fdf_filename)
from google.colab import files
files.download('silicane.fdf')
```

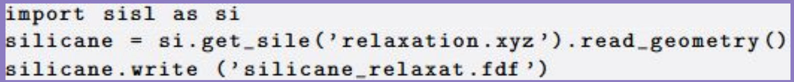
.fdf

- Relaxació de l’estructura



```
%block kgrid_Monkhorst_Pack
  10  0  0  0.0
   0 10  0  0.0
   0  0  1  0.0
%endblock kgrid_Monkhorst_Pack

BandLinesScale          ReciprocalLatticeVectors
%block BandLines
   1 0.0000 0.0000 0.0000 Gamma
  20 0.5000 0.0000 0.0000 M
  20 0.3000 0.3000 0.3000 K
  20 0.0000 0.0000 0.0000 Gamma

%endblock BandLines

SOC.Split.SR.SO        true
```

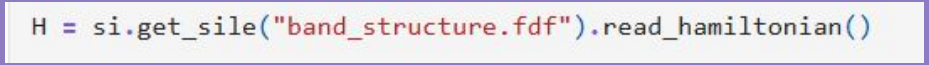
+ l’hamiltonià



```
import sisl as si
silicane = si.get_sile('relaxation.xyz').read_geometry()
silicane.write ('silicane_relaxat.fdf')
```

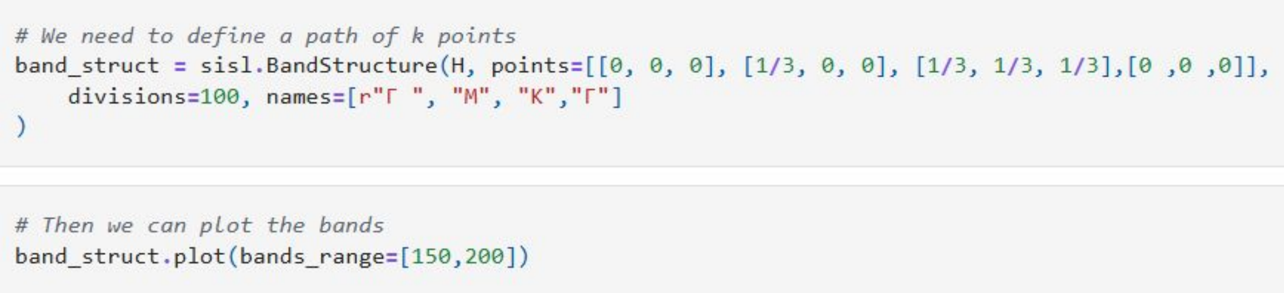
.xyz

Càlcul de l’estructura de bandes



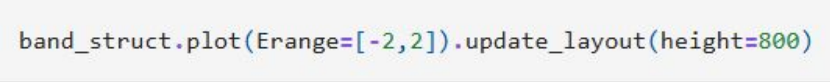
```
H = si.get_sile("band_structure.fdf").read_hamiltonian()
```

3. Resultats

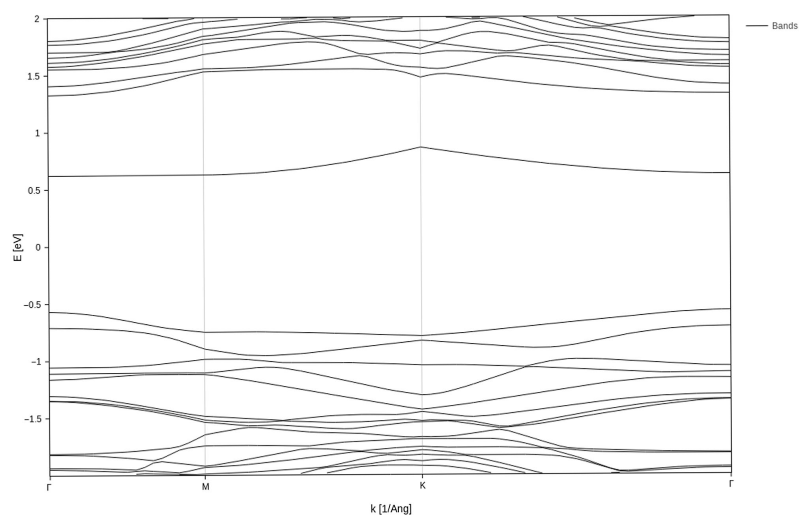


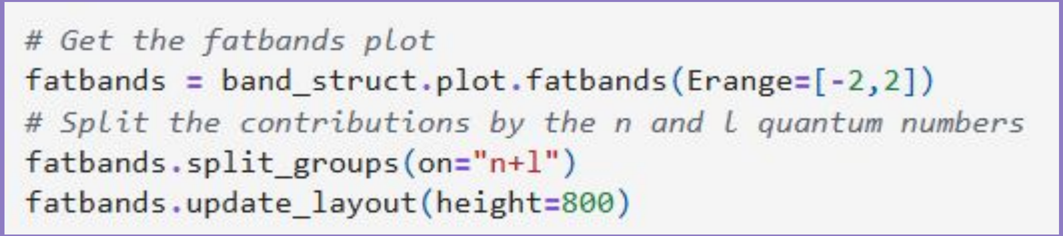
```
# We need to define a path of k points
band_struct = sisl.BandStructure(H, points=[[0, 0, 0], [1/3, 0, 0], [1/3, 1/3, 1/3],[0 ,0 ,0]],
    divisions=100, names=[r"Γ ", "M", "K","Γ"]
)
```

```
# Then we can plot the bands
band_struct.plot(bands_range=[150,200])
```

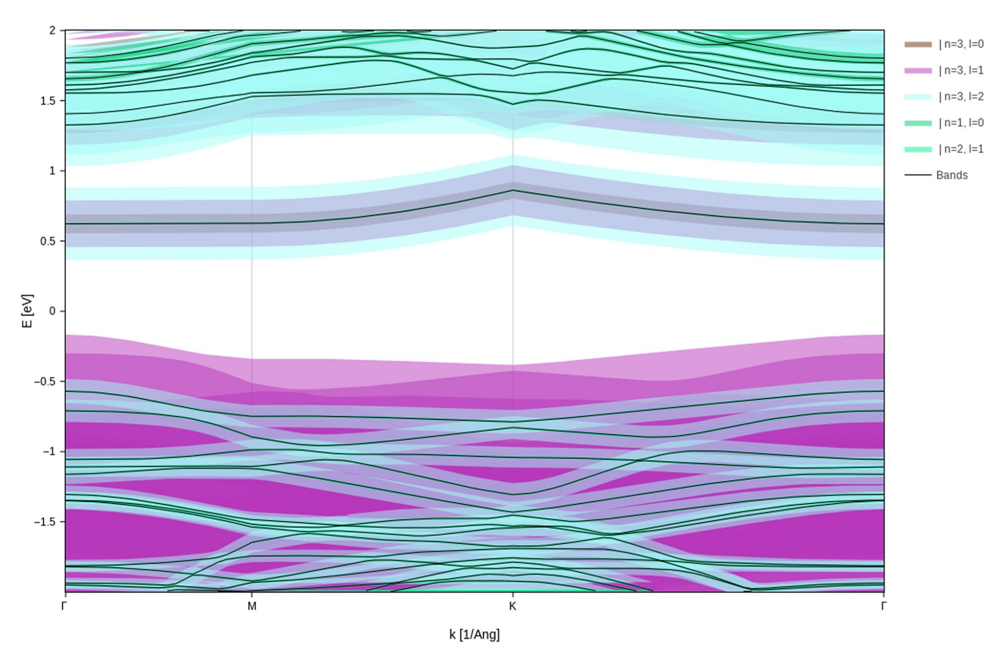


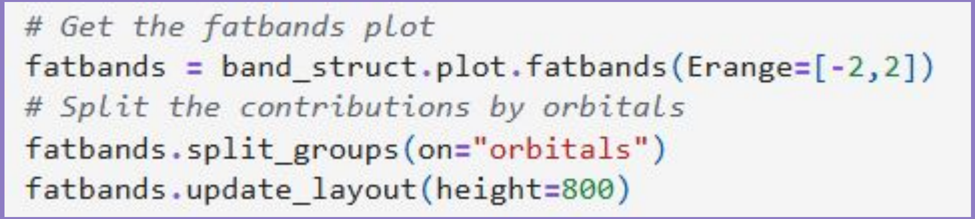
```
band_struct.plot(Erange=[-2,2]).update_layout(height=800)
```



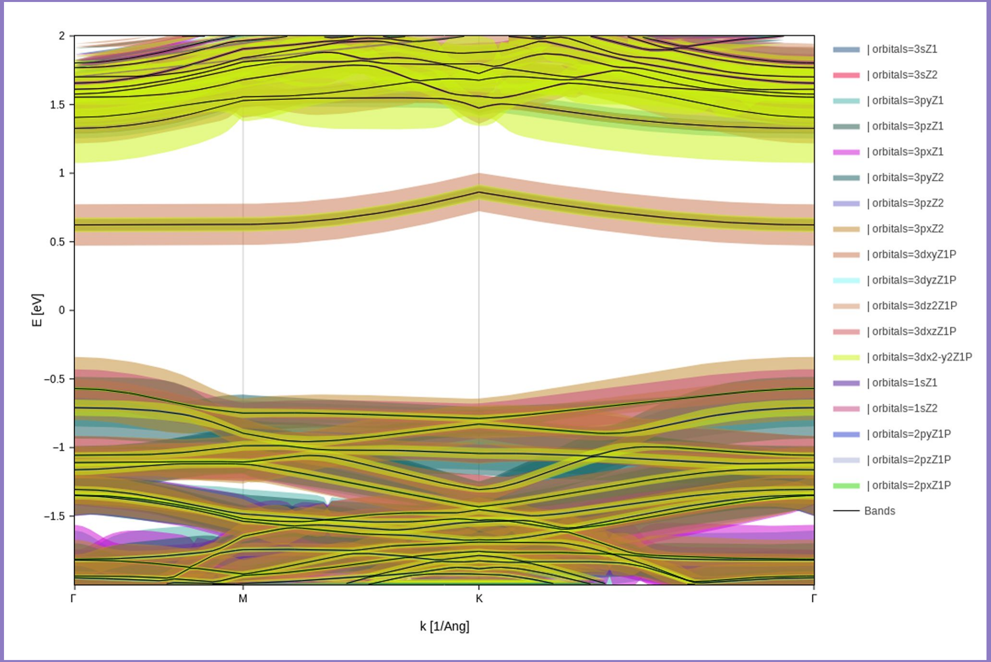


```python
# Get the fatbands plot
fatbands = band_struct.plot.fatbands(Erange=[-2,2])
# Split the contributions by the n and l quantum numbers
fatbands.split_groups(on="n+l")
fatbands.update_layout(height=800)
```





```python
# Get the fatbands plot
fatbands = band_struct.plot.fatbands(Erange=[-2,2])
# Split the contributions by orbitals
fatbands.split_groups(on="orbitals")
fatbands.update_layout(height=800)
```



4. Conclusions

- Orbitals que intervenen en les bandes
- Referències a la literatura