[10]
```
import numpy as np
import matplotlib.pyplot as plt
import read_upf
import importlib
```

[33]
```
importlib.reload(read_upf)
upf = read_upf.read_upf('C.pbe-tm-gipaw-dc.UPF')
```

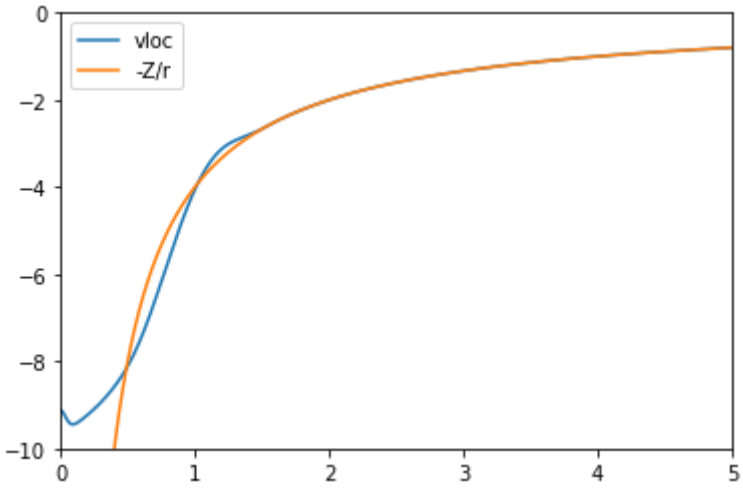
[34]
```
upf.__dict__
```

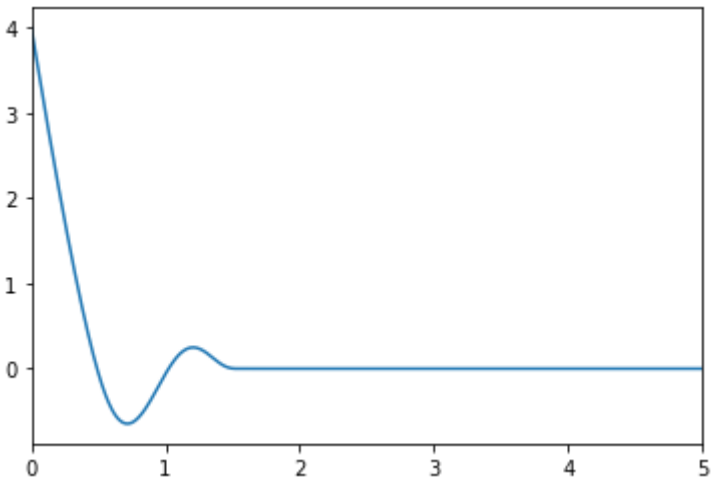
```
{'version': 1,
 'info': '\nGenerated using "atomic" code by A. Dal Corso  (espresso
distribution)     \nAuthor: D.C.   Generation date: 14Sep2009
\nC
\n    1        The Pseudo was generated with a Scalar-Relativistic
Calculation\n    1 1.5400000E+00    L component and cutoff radius for
Local Potential\nnl pn  l   occ               Rcut            Rcut US
E pseu\n2S  1  0  2.00      1.54000000000      1.54000000000
-1.01066959746\n2P  2  1  2.00      1.54000000000      1.54000000000
-0.38848154851\n',
 'element': 'C',
 'type': 'NC',
 'nlcc': False,
 'qexc': 'SLA PW PBX PBC',
 'val': 4.0,
 'lmax': 1,
 'npoints': 1441,
 'nwfc': 4,
 'nproj': 1,
 'r': array([5.59104380e-05, 5.64723472e-05, 5.70399038e-05, ...,
        9.83212846e+01, 9.93094299e+01, 1.00307506e+02]),
 'rab': array([5.59104380e-07, 5.64723472e-07, 5.70399038e-07, ...,
        9.83212846e-01, 9.93094299e-01, 1.00307506e+00]),
 'vloc': array([-9.12512778, -9.1251278 , -9.12512782, ..., -0.04068295,
        -0.04027815, -0.03987737]),
 'pswfc': [{'label': '2S',
   'occ': 2.0,
   'wfc': array([4.25017538e-05, 4.29289035e-05, 4.33603462e-05, ...,
          0.00000000e+00, 0.00000000e+00, 0.00000000e+00])},
  {'label': '3S',
   'occ': 0.0,
   'wfc': array([ 8.78871551e-06,  8.87704357e-06,  8.96625934e-06, ...,
          -7.18452658e-06, -6.36791124e-06, -5.63700275e-06])},
  {'label': '2P',
   'occ': 2.0,
   'wfc': array([7.18567007e-09, 7.33083024e-09, 7.47892283e-09, ...,
          1.63716670e-26, 8.91331845e-27, 4.82294082e-27])},
  {'label': '3P',
   'occ': 0.0,
   'wfc': array([ 9.40887783e-09,  9.59894977e-09,  9.79286141e-09, ...,
```

```
          -8.26973434e-03, -8.20436274e-03, -8.13987494e-03])}],
 'atrho': array([3.61279826e-09, 3.68578163e-09, 3.76023936e-09, ...,
        5.36062959e-52, 1.58894491e-52, 4.65215162e-53])}
```

[52]
```
plt.plot(upf.r, upf.vloc, label='vloc')
plt.plot(upf.r, -upf.val/upf.r, label='-Z/r')
plt.legend()
plt.xlim(0,5)
plt.ylim(-10,0)
plt.show()

diff = upf.vloc*upf.r + upf.val
plt.plot(upf.r, diff)
plt.xlim(0,5)
plt.show()
for i in range(upf.npoints-1,0,-1):
    if abs(diff[i]) > 1e-3:
        print('rcloc =', upf.r[i])
        break
```

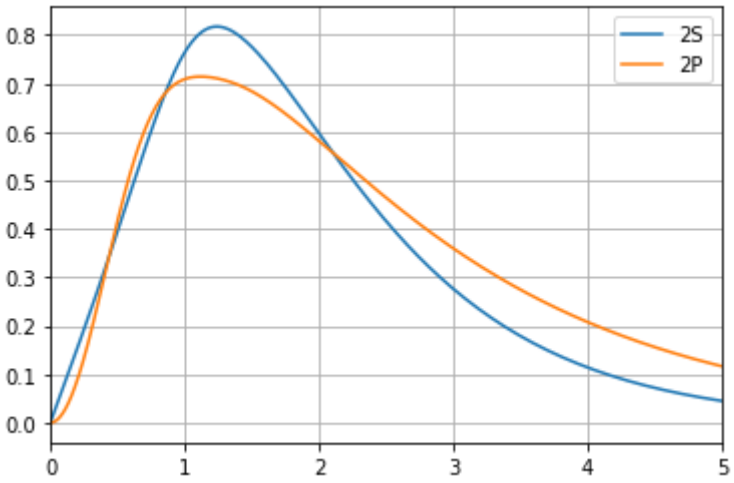





```
rcloc = 1.50416891657
```

[54]
```
for orb in upf.pswfc:
    if orb['occ'] > 0:
```

```
            plt.plot(upf.r, orb['wfc'], label=orb['label'])
    plt.legend()
    plt.xlim(0,5)
    plt.grid()
    plt.show()
```




```
[ ]
```