nucleardatapy

September 2, 2025

1 nucleardatapy toolkit for simple access to experimental nuclear data, astrophysical observations, and theoretical predictions

J. Margueron, IRL-NPA, USA

paper: https://arxiv.org/abs/2506.20434

documentation: https://nucleardatapy.readthedocs.io/en/latest/

tutorials: https://jeromemargueron.github.io/nucleardatapy

[ ]:
```
!pip install nucleardatapy
```

[6]:
```
import matplotlib.pyplot as plt
%matplotlib inline
import numpy as np
import nucleardatapy as nuda
```

1.1 Warm up

[9]:
```
nuda.hello()
```

```
hello world!
```

[10]:
```
help(nuda)
```

```
Help on package nucleardatapy:

NAME
    nucleardatapy

DESCRIPTION
    This toolkit is a community-driven tool providing data for nuclear and
astro-nuclear research community. It contains microscopic, phenomenological,
experimental and observational data for the understanding of neutron stars.
    To get some help, please contact: nucleardatapy@gmail.com.

PACKAGE CONTENTS
```

```
    astro (package)
    corr (package)
    create_folder
    crust (package)
    cst
    env
    eos (package)
    fig (package)
    hello
    hnuc (package)
    matter (package)
    nuc (package)
    param

SUBMODULES
    eos_setupAMBeq_fig
    eos_setupAMLeq_fig
    eos_setupAM_asy_lep_fig
    eos_setupAM_asy_nuc_fig
    eos_setupAM_asy_tot_fig
    eos_setupAM_fig
    eos_setupCC_fig
    matter_all_fig
    matter_setupFFGNuc_fig
    matter_setupMicro_fig
    matter_setupMicro_gap_fig
    matter_setupPheno_fig
    nuc_setupBEExp_chart_fig
    nuc_setupBEExp_fig
    nuc_setupBETheo_fig
    setup_EsymDen
    setup_EsymLsym
    setup_KsatQsat
    setup_am
    setup_am_Beq
    setup_am_Leq
    setup_be_exp
    setup_be_theo
    setup_check
    setup_crust
    setup_ffg
    setup_gw
    setup_hic
    setup_isgmr_exp
    setup_masses
    setup_micro
    setup_micro_band
    setup_micro_effmass
```

```
    setup_micro_esym
    setup_micro_gap
    setup_micro_lp
    setup_mr
    setup_mtov
    setup_mup
    setup_nep
    setup_nep_stat_model
    setup_nep_stat_models
    setup_pheno
    setup_pheno_esym
    setup_rch_exp
    setup_rch_theo
    setup_re1L_exp
    setup_re1Xi_exp
    setup_re2L_exp
    setup_rnp_exp
    setup_rnp_theo

DATA
    CST_AtmMass = 931.494028
    CST_dmHc2 = 1e-07
    CST_dmnc2 = 5e-07
    CST_mHc2 = 7.2889706
    CST_mnc2 = 8.0713171
    G_cgs = 6.67259e-08
    G_si = 6.67259e-11
    HThsl = 0.001
    HTsl = 86400
    HTvsl = 1.0
    ILt = 3.155692608e+37
    ISt = 1e-30
    MODEL_STYLES = {'ddrh': {'color': 'green', 'label': 'DDRH', 'marker': ...
    MODEL_TABLES_DIR = '/home/sudhanvalalit/Documents/Research/Jerome/nucl...
    MeV2J = 1.6021766339999998e-13
    MeV2kg = 1.7826619216278976e-30
    MeVfm32dyncm2 = 1.602176634e+33
    MeVfm32gcm3 = 1782661921627.8977
    Msol_cgs = 1.99e+33
    Msol_si = 1.99e+30
    Nc = 3
    Rsol_cgs = 69600000000.0
    Rsol_si = 696000000.0
    SOURCE_LABELS_LATEX = {'208Pb': r'$^{208}\mathrm{Pb}$', '48Ca': r'$^{4...
    c = 299792458.0
    c2 = 8.987551787368176e+16
    col = ['blue', 'orange', 'green', 'red', 'purple', 'brown', 'pink', 'g...
    day = 86400
```

```
    e = 1.602176634e-19
    elements = array(['H', 'He', 'Li', 'Be', 'B', 'C', 'N', 'O'...'Nh', 'F...
    five = 5.0
    fiveThird = 1.6666666666666667
    four = 4.0
    h2m = 41.471043279826425
    half = 0.5
    hbc = 197.3269804
    hour = 3600
    mec2 = 0.510998949
    minute = 60
    mmuc2 = 105.6583745
    mnc2 = 939.565346
    mnuc2 = 939.0
    mnuc2_approx = 939.0
    month = 2592000
    mpc2 = 938.272013
    ns = 1e-09
    nsat = 0.16
    nucleardatapy_tk = None
    one = 1.0
    path_ame = '/home/sudhanvalalit/Documents/Research/Jerome/nucleardatap...
    path_data = '/home/sudhanvalalit/Documents/Research/Jerome/nucleardata...
    pi = 3.141592653589793
    pi2 = 9.869604401089358
    rshsol_si = 2954.85453973408
    third = 0.3333333333333333
    three = 3.0
    threeFifth = 0.6
    two = 2.0
    twoThird = 0.6666666666666666
    verb = False
    verb_latex = True
    verb_output = False
    year = 31556926.080000002
    yearMin = 1890

FILE
    /home/sudhanvalalit/Documents/Research/Jerome/nucleardatapy/version-
1.0/nucleardatapy/__init__.py
```

Load the energy per particle from APR equation of state and instantiate the object apr:

[11]:
```python
apr = nuda.matter.setupMicro( model = '1998-VAR-AM-APR' )
print('den in SM:',apr.sm_den)
print('energy per particle in SM:',apr.sm_e2a_int)
```

```
model ->  1998-VAR-AM-APR
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
den in SM: [0.04 0.08 0.12 0.16 0.2  0.24 0.32 0.4  0.48 0.56 0.64 0.8  0.96]
energy per particle in SM: [ -6.48 -12.13 -15.04 -16.   -15.09 -12.88  -5.03
2.13  15.46  34.39
  58.35 121.25 204.02]
```

[12]:
```
apr.print_outputs()
```

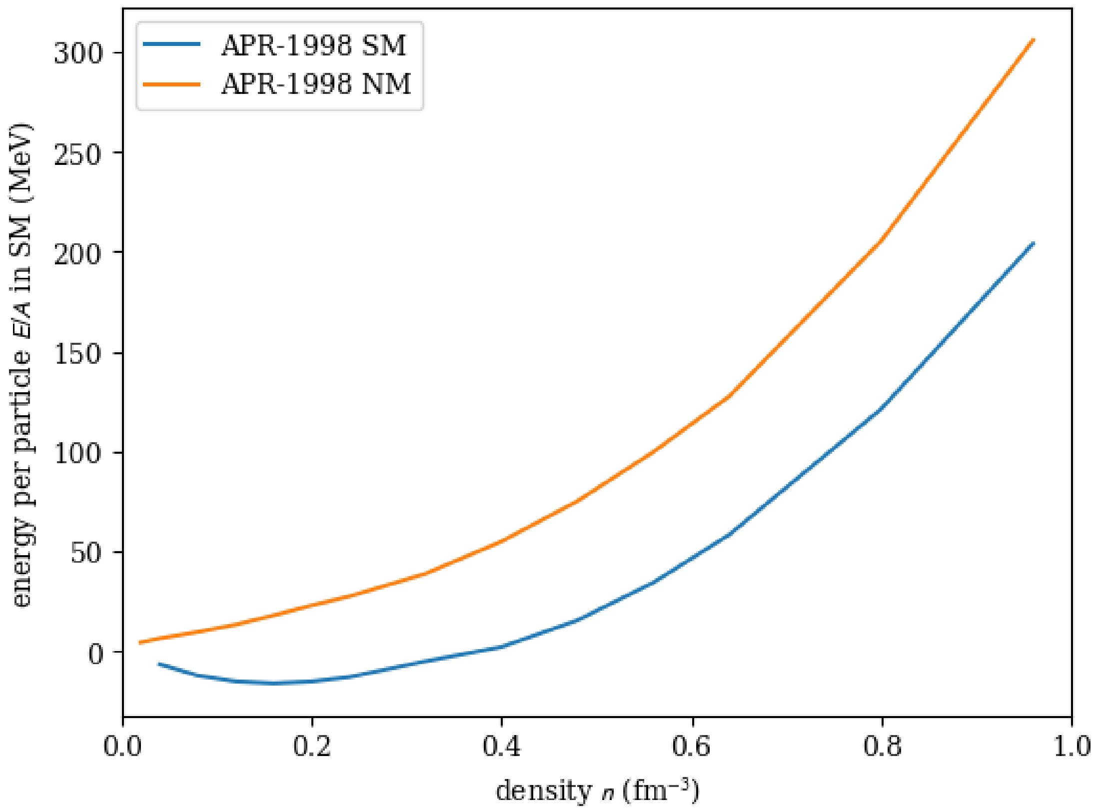
```
- Print output:
   model: 1998-VAR-AM-APR
   ref:   Akmal, Pandharipande and Ravenhall, Phys. Rev. C 58, 1804 (1998)
   label: APR-1998
   note:  write here notes about this EOS.
   self.sm_den:  [0.04 0.08 0.12 0.16 0.2  0.24 0.32 0.4  0.48 0.56 0.64 0.8
0.96]
   self.sm_effmass:  None
   den: [0.01  0.031 0.051 0.072 0.092 0.113 0.133 0.154 0.174 0.195 0.215 0.236
 0.256 0.277 0.297 0.318 0.338 0.359 0.379 0.4  ] in fm$^{-3}$
   kfn: [0.01  0.031 0.051 0.072 0.092 0.113 0.133 0.154 0.174 0.195 0.215 0.236
 0.256 0.277 0.297 0.318 0.338 0.359 0.379 0.4  ] in fm$^{-1}$
   asy: 0.0
   sm_den: [0.04 0.08 0.12 0.16 0.2  0.24 0.32 0.4  0.48 0.56 0.64 0.8  0.96] in
fm$^{-3}$
   sm_kfn: [0.84  1.058 1.211 1.333 1.436 1.526 1.68  1.809 1.923 2.024 2.116
2.279
 2.422] in fm$^{-1}$
   sm_e2a: [ 932.439  926.789  923.879  922.919  923.829  926.039  933.889
941.049
  954.379  973.309  997.269 1060.169 1142.939] in MeV
   sm_e2a_err: [ 0.113  0.425  0.79   1.12   1.32   1.352  0.704  0.373  3.247
8.426
 16.338 42.438 85.688] in MeV
   sm_eps: [  37.298   74.143  110.865  147.667  184.766  222.249  298.844
376.419
  458.102  545.053  638.252  848.135 1097.221] in MeV fm$^{-3}$
   sm_eps_err: [5.0000e-03 3.4000e-02 9.5000e-02 1.7900e-01 2.6400e-01
3.2500e-01
 2.2500e-01 1.4900e-01 1.5580e+00 4.7180e+00 1.0456e+01 3.3950e+01
 8.2261e+01] in MeV fm$^{-3}$
   sm_pre: [-2.61000e-01 -6.94000e-01 -6.57000e-01  1.60000e-02  1.57600e+00
  4.36500e+00  9.46300e+00  1.87940e+01  4.74880e+01  8.39950e+01
  1.35537e+02  2.91381e+02  5.33861e+02] in MeV fm$^{-3}$
   sm_cs2: [-0.013 -0.007  0.009  0.028  0.061  0.077  0.06   0.24   0.397
0.475
```

```
 0.62   0.858  1.079]
   nm_den: [0.02 0.04 0.08 0.12 0.16 0.2  0.24 0.32 0.4  0.48 0.56 0.64 0.8
0.96] in fm$^{-3}$
   nm_kfn: [0.84  1.058 1.333 1.526 1.68  1.809 1.923 2.116 2.279 2.422 2.55
2.666
 2.872 3.052] in fm$^{-1}$
   nm_e2a: [ 944.015  946.015  949.215  952.855  957.505  962.485  967.055
978.385
  994.515 1014.695 1039.315 1067.145 1144.905 1245.435] in MeV
   nm_e2a_err: [3.90000e-02 1.13000e-01 3.38000e-01 6.98000e-01 1.25600e+00
2.00600e+00
 2.88600e+00 5.43500e+00 9.61600e+00 1.57770e+01 2.44390e+01 3.57220e+01
 7.18690e+01 1.28465e+02] in MeV
   nm_eps: [  18.88    37.841   75.937  114.343  153.201  192.497  232.093
313.083
  397.806  487.054  582.017  682.973  915.924 1195.618] in MeV fm$^{-3}$
   nm_eps_err: [1.00000e-03 5.00000e-03 2.70000e-02 8.40000e-02 2.01000e-01
4.01000e-01
 6.93000e-01 1.73900e+00 3.84700e+00 7.57300e+00 1.36860e+01 2.28620e+01
 5.74950e+01 1.23327e+02] in MeV fm$^{-3}$
   nm_pre: [5.60000e-02 1.27000e-01 5.28000e-01 1.49100e+00 3.20100e+00
4.74200e+00
 6.73700e+00 1.75560e+01 3.63200e+01 6.50160e+01 1.01687e+02 1.58788e+02
 3.65277e+02 6.19583e+02] in MeV fm$^{-3}$
   nm_cs2: [0.003 0.005 0.016 0.037 0.044 0.037 0.075 0.178 0.275 0.348 0.45
0.697
 0.969 0.801]
```

Plot the binding energy as function of the particle density:

[13]:
```
#import matplotlib as mpl
#mpl.use("Agg")
import matplotlib.pyplot as plt
%matplotlib inline
plt.xlabel(r'density $n$ (fm$^{-3}$)')
plt.ylabel(r'energy per particle $E/A$ in SM (MeV)')
plt.xlim(0.0,1.0)
plt.plot( apr.sm_den, apr.sm_e2a_int, label = apr.label+' SM' )
plt.plot( apr.nm_den, apr.nm_e2a_int, label = apr.label+' NM' )
plt.legend(loc='upper left')
```

[13]:
```
<matplotlib.legend.Legend at 0x70986e738d70>
```



List the variety of many-body approaches (mbs):

[14]:
```
mbs, mbs_lower = nuda.matter. micro_mbs()
print('mbs available:',mbs)
```

```
mbs available: ['VAR', 'AFDMC', 'BHF2', 'BHF23', 'QMC', 'MBPT', 'NLEFT']
```

Select a many-body approach and list the number of models:

[15]:
```
mb = 'VAR'
models, models_lower=nuda.matter.micro_models_mb( mb=mb )
print('for mb:',mb,' models:',models)
```

```
for mb: VAR  models: ['1981-VAR-AM-FP', '1998-VAR-AM-APR', '1998-VAR-AM-APR-
fit']
```

Select some microscopic models and show their predictions:

[16]:
```
models = [ '1981-VAR-AM-FP', '1998-VAR-AM-APR' ]
print('models:',models)
plt.xlabel(r'density $n$ (fm$^{-3}$)')
plt.xlim(0.0,1.0)
plt.ylabel(r'energy per particle $E/A$ in SM (MeV)')
```

```
plt.ylim(-20.0,300.0)
for model in models:
    mic = nuda.matter.setupMicro( model = model )
    mic.print_outputs()
    plt.plot( mic.sm_den, mic.sm_e2a_int, label = mic.label+' SM' )
    plt.plot( mic.nm_den, mic.nm_e2a_int, label = mic.label+' NM' )
plt.legend(loc='upper left')
```

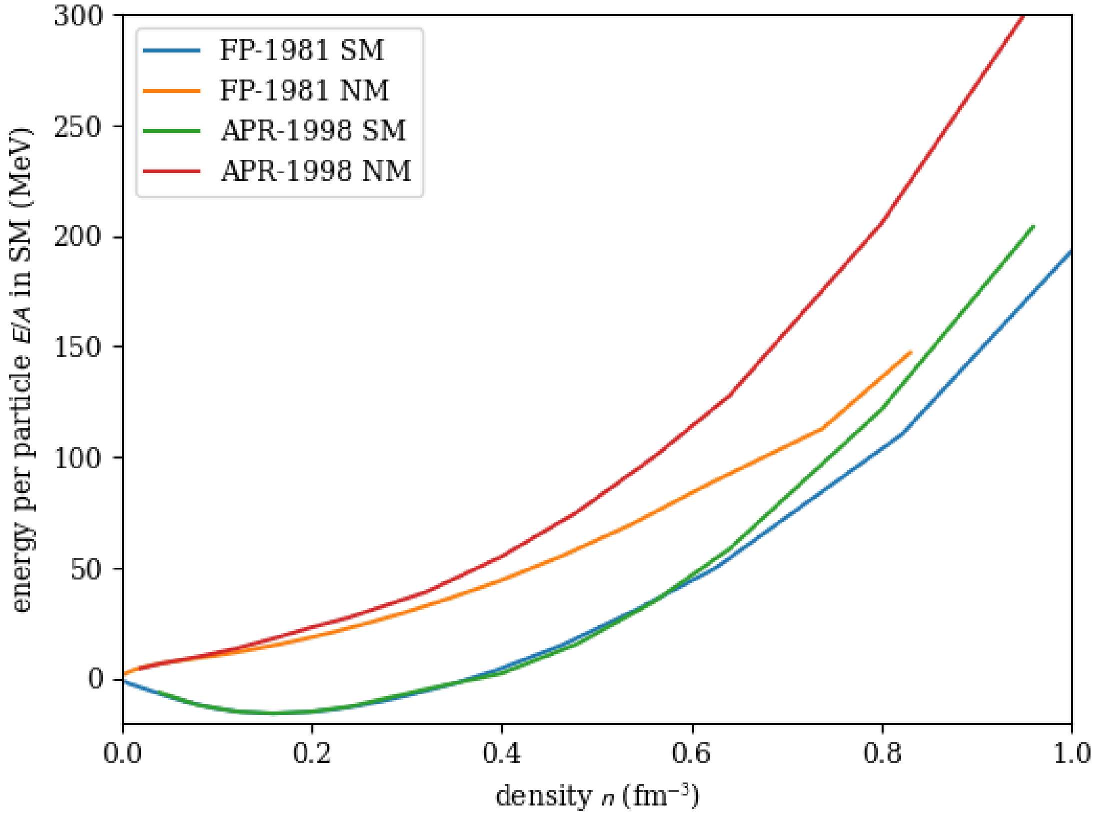
```
models: ['1981-VAR-AM-FP', '1998-VAR-AM-APR']
model ->  1981-VAR-AM-FP
flag_nm: True
flag_sm: True
flag_kf: True
flag_den: False
- Print output:
   model: 1981-VAR-AM-FP
   ref:   Friedman and Pandharipande, Nucl. Phys. A. 361, 502 (1981)
   label: FP-1981
   note:  write here notes about this EOS.
   self.sm_den:  [0.004323  0.0084434 0.01459   0.023169  0.034584  0.049242
0.067547
 0.097464  0.1257    0.15891   0.19752   0.22344   0.25154   0.27667
 0.30343   0.34974   0.39393   0.46331   0.54038   0.62555   0.82185
 1.0554    1.4048    1.8238   ]
   self.sm_effmass:  None
   den: [0.01  0.031 0.051 0.072 0.092 0.113 0.133 0.154 0.174 0.195 0.215 0.236
 0.256 0.277 0.297 0.318 0.338 0.359 0.379 0.4  ] in fm$^{-3}$
   kfn: [0.01  0.031 0.051 0.072 0.092 0.113 0.133 0.154 0.174 0.195 0.215 0.236
 0.256 0.277 0.297 0.318 0.338 0.359 0.379 0.4  ] in fm$^{-1}$
   asy: 0.0
   sm_den: [0.004 0.008 0.015 0.023 0.035 0.049 0.068 0.097 0.126 0.159 0.198
0.223
 0.252 0.277 0.303 0.35  0.394 0.463 0.54  0.626 0.822 1.055 1.405 1.824] in
fm$^{-3}$
   sm_kfn: [0.4   0.5   0.6   0.7   0.8   0.9   1.    1.13 1.23 1.33 1.43 1.49 1.55 1.6
 1.65 1.73 1.8   1.9   2.    2.1   2.3   2.5   2.75 3.   ] in fm$^{-1}$
   sm_e2a: [ 937.243   936.334   935.517   934.272   932.654   930.627   928.319
925.319
  923.619   922.919   923.599   924.789   926.639   928.779   931.373   936.418
  942.383   953.719   969.469   988.659 1049.119 1157.419 1378.819 1725.619] in MeV
   sm_e2a_err: [3.00000e-03 1.00000e-02 2.20000e-02 4.70000e-02 9.50000e-02
1.79000e-01
 3.13000e-01 5.80000e-01 8.41000e-01 1.11200e+00 1.32400e+00 1.38100e+00
 1.35100e+00 1.22700e+00 1.00200e+00 3.83000e-01 5.97000e-01 3.00000e+00
 7.22300e+00 1.36130e+01 3.96230e+01 1.00890e+02 2.70363e+02 6.27718e+02] in MeV
   sm_eps: [    4.052     7.906    13.649    21.646    32.255    45.826    62.705
90.185
```

```
  116.099  146.661  182.429  206.635  233.087  256.965  282.606  327.503
  371.233  441.867  523.881  618.455  862.218 1221.54  1936.964 3147.183] in MeV
fm$^{-3}$
   sm_eps_err: [0.000000e+00 0.000000e+00 0.000000e+00 1.000000e-03 3.000000e-03
 9.000000e-03 2.100000e-02 5.700000e-02 1.060000e-01 1.770000e-01
 2.610000e-01 3.090000e-01 3.400000e-01 3.400000e-01 3.040000e-01
 1.340000e-01 2.350000e-01 1.390000e+00 3.903000e+00 8.515000e+00
 3.256500e+01 1.064790e+02 3.798050e+02 1.144832e+03] in MeV fm$^{-3}$
   sm_pre: [-6.000000e-03 -1.100000e-02 -2.900000e-02 -7.800000e-02
-1.680000e-01
 -3.250000e-01 -5.380000e-01 -7.590000e-01 -6.520000e-01 -4.700000e-02
  1.388000e+00  2.772000e+00  4.850000e+00  7.061000e+00  9.232000e+00
  1.491400e+01  2.257100e+01  3.984800e+01  6.263300e+01  9.503500e+01
  2.573570e+02  5.987120e+02  1.433611e+03  3.071787e+03] in MeV fm$^{-3}$
   sm_cs2: [-1.000e-03 -2.000e-03 -4.000e-03 -7.000e-03 -1.000e-02 -1.300e-02
 -1.200e-02 -3.000e-03  1.100e-02  2.900e-02  5.000e-02  6.700e-02
  9.000e-02  8.700e-02  9.500e-02  1.510e-01  2.050e-01  2.680e-01
  2.890e-01  4.320e-01  8.290e-01  1.047e+00  1.259e+00  1.428e+00]
   nm_den: [0.004 0.007 0.012 0.017 0.025 0.034 0.045 0.058 0.074 0.093 0.114
0.138
 0.166 0.192 0.222 0.257 0.298 0.345 0.399 0.462 0.535 0.619 0.738 0.83 ] in
fm$^{-3}$
   nm_kfn: [0.5   0.6   0.7   0.8   0.9   1.    1.1   1.2   1.3   1.4   1.5
1.6
 1.7   1.785 1.874 1.968 2.066 2.17  2.278 2.392 2.512 2.637 2.796 2.908] in
fm$^{-1}$
   nm_e2a: [ 941.622  942.312  943.068  943.887  944.742  945.592  946.438
947.309
  948.264  949.387  950.785  952.575  954.825  957.255  960.295  964.235
  969.155  975.605  983.725  994.505 1008.715 1027.525 1052.365 1086.665] in MeV
   nm_e2a_err: [4.0000e-03 9.0000e-03 1.8000e-02 3.3000e-02 5.6000e-02
8.9000e-02
 1.3500e-01 1.9800e-01 2.8200e-01 3.9800e-01 5.6000e-01 7.8700e-01
 1.1080e+00 1.4870e+00 2.0160e+00 2.7780e+00 3.8550e+00 5.4410e+00
 7.7130e+00 1.1110e+01 1.6194e+01 2.3832e+01 3.6430e+01 5.3448e+01] in MeV
   nm_eps: [  3.976   6.874  10.921  16.32   23.26   31.933  42.542  55.285
70.361
  87.98  108.39  131.741 158.406 183.889 213.474 248.194 288.614 336.681
 392.703 459.66  539.965 636.346 776.856 902.476] in MeV fm$^{-3}$
   nm_eps_err: [0.0000e+00 0.0000e+00 0.0000e+00 1.0000e-03 1.0000e-03
3.0000e-03
 6.0000e-03 1.2000e-02 2.1000e-02 3.7000e-02 6.4000e-02 1.0900e-01
 1.8400e-01 2.8600e-01 4.4800e-01 7.1500e-01 1.1480e+00 1.8780e+00
 3.0790e+00 5.1350e+00 8.6690e+00 1.4759e+01 2.6893e+01 4.4388e+01] in MeV
fm$^{-3}$
   nm_pre: [5.00000e-03 1.10000e-02 2.10000e-02 3.90000e-02 6.30000e-02
9.50000e-02
 1.40000e-01 2.10000e-01 3.29000e-01 5.35000e-01 9.03000e-01 1.47100e+00
```

```
 2.41100e+00 3.55700e+00 5.25200e+00 7.72200e+00 1.14110e+01 1.69440e+01
 2.55490e+01 3.82030e+01 6.25490e+01 7.81060e+01 1.46661e+02 3.39173e+02] in MeV
fm$^{-3}$
   nm_cs2: [2.000e-03 2.000e-03 3.000e-03 3.000e-03 4.000e-03 4.000e-03
5.000e-03
 6.000e-03 9.000e-03 1.500e-02 2.100e-02 2.900e-02 4.000e-02 5.100e-02
 6.300e-02 8.000e-02 1.000e-01 1.360e-01 1.570e-01 2.640e-01 2.460e-01
 1.610e-01 9.800e-01 2.075e+00]
model ->  1998-VAR-AM-APR
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
- Print output:
   model: 1998-VAR-AM-APR
   ref:   Akmal, Pandharipande and Ravenhall, Phys. Rev. C 58, 1804 (1998)
   label: APR-1998
   note:  write here notes about this EOS.
   self.sm_den:  [0.04 0.08 0.12 0.16 0.2  0.24 0.32 0.4  0.48 0.56 0.64 0.8
0.96]
   self.sm_effmass:  None
   den: [0.01  0.031 0.051 0.072 0.092 0.113 0.133 0.154 0.174 0.195 0.215 0.236
 0.256 0.277 0.297 0.318 0.338 0.359 0.379 0.4  ] in fm$^{-3}$
   kfn: [0.01  0.031 0.051 0.072 0.092 0.113 0.133 0.154 0.174 0.195 0.215 0.236
 0.256 0.277 0.297 0.318 0.338 0.359 0.379 0.4  ] in fm$^{-1}$
   asy: 0.0
   sm_den: [0.04 0.08 0.12 0.16 0.2  0.24 0.32 0.4  0.48 0.56 0.64 0.8  0.96] in
fm$^{-3}$
   sm_kfn: [0.84  1.058 1.211 1.333 1.436 1.526 1.68  1.809 1.923 2.024 2.116
2.279
 2.422] in fm$^{-1}$
   sm_e2a: [ 932.439  926.789  923.879  922.919  923.829  926.039  933.889
941.049
  954.379  973.309  997.269 1060.169 1142.939] in MeV
   sm_e2a_err: [ 0.113  0.425  0.79   1.12   1.32   1.352  0.704  0.373  3.247
8.426
 16.338 42.438 85.688] in MeV
   sm_eps: [  37.298   74.143  110.865  147.667  184.766  222.249  298.844
376.419
  458.102  545.053  638.252  848.135 1097.221] in MeV fm$^{-3}$
   sm_eps_err: [5.0000e-03 3.4000e-02 9.5000e-02 1.7900e-01 2.6400e-01
3.2500e-01
 2.2500e-01 1.4900e-01 1.5580e+00 4.7180e+00 1.0456e+01 3.3950e+01
 8.2261e+01] in MeV fm$^{-3}$
   sm_pre: [-2.61000e-01 -6.94000e-01 -6.57000e-01  1.60000e-02  1.57600e+00
  4.36500e+00  9.46300e+00  1.87940e+01  4.74880e+01  8.39950e+01
  1.35537e+02  2.91381e+02  5.33861e+02] in MeV fm$^{-3}$
   sm_cs2: [-0.013 -0.007  0.009  0.028  0.061  0.077  0.06   0.24   0.397
```

```
0.475
  0.62   0.858  1.079]
   nm_den: [0.02 0.04 0.08 0.12 0.16 0.2  0.24 0.32 0.4  0.48 0.56 0.64 0.8
0.96] in fm$^{-3}$
   nm_kfn: [0.84  1.058 1.333 1.526 1.68  1.809 1.923 2.116 2.279 2.422 2.55
2.666
 2.872 3.052] in fm$^{-1}$
   nm_e2a: [ 944.015  946.015  949.215  952.855  957.505  962.485  967.055
978.385
  994.515 1014.695 1039.315 1067.145 1144.905 1245.435] in MeV
   nm_e2a_err: [3.90000e-02 1.13000e-01 3.38000e-01 6.98000e-01 1.25600e+00
2.00600e+00
 2.88600e+00 5.43500e+00 9.61600e+00 1.57770e+01 2.44390e+01 3.57220e+01
 7.18690e+01 1.28465e+02] in MeV
   nm_eps: [  18.88    37.841   75.937  114.343  153.201  192.497  232.093
313.083
  397.806  487.054  582.017  682.973  915.924 1195.618] in MeV fm$^{-3}$
   nm_eps_err: [1.00000e-03 5.00000e-03 2.70000e-02 8.40000e-02 2.01000e-01
4.01000e-01
 6.93000e-01 1.73900e+00 3.84700e+00 7.57300e+00 1.36860e+01 2.28620e+01
 5.74950e+01 1.23327e+02] in MeV fm$^{-3}$
   nm_pre: [5.60000e-02 1.27000e-01 5.28000e-01 1.49100e+00 3.20100e+00
4.74200e+00
 6.73700e+00 1.75560e+01 3.63200e+01 6.50160e+01 1.01687e+02 1.58788e+02
 3.65277e+02 6.19583e+02] in MeV fm$^{-3}$
   nm_cs2: [0.003 0.005 0.016 0.037 0.044 0.037 0.075 0.178 0.275 0.348 0.45
0.697
 0.969 0.801]
```

```
[16]: <matplotlib.legend.Legend at 0x70986e6e0690>
```



1.2 Module: Matter

Get the full list of microscopic models:

[17]:
```
models, models_lower = nuda.matter.micro_models()
print('models:',models)
```

```
models: ['1981-VAR-AM-FP', '1998-VAR-AM-APR', '1998-VAR-AM-APR-fit',
'2012-AFDMC-NM-RES-1', '2012-AFDMC-NM-RES-2', '2012-AFDMC-NM-RES-3',
'2012-AFDMC-NM-RES-4', '2012-AFDMC-NM-RES-5', '2012-AFDMC-NM-RES-6',
'2012-AFDMC-NM-RES-7', '2012-AFDMC-NM-FIT-1', '2012-AFDMC-NM-FIT-2',
'2012-AFDMC-NM-FIT-3', '2012-AFDMC-NM-FIT-4', '2012-AFDMC-NM-FIT-5',
'2012-AFDMC-NM-FIT-6', '2012-AFDMC-NM-FIT-7', '2022-AFDMC-NM', '2024-BHF-AM-2BF-
Av18', '2024-BHF-AM-2BF-BONN', '2024-BHF-AM-2BF-CDBONN', '2024-BHF-AM-2BF-
NSC97a', '2024-BHF-AM-2BF-NSC97b', '2024-BHF-AM-2BF-NSC97c', '2024-BHF-AM-2BF-
NSC97d', '2024-BHF-AM-2BF-NSC97e', '2024-BHF-AM-2BF-NSC97f', '2006-BHF-AM-Av18',
'2024-BHF-AM-23BF-Av18', '2024-BHF-AM-23BF-BONN', '2024-BHF-AM-23BF-CDBONN',
'2024-BHF-AM-23BF-NSC97a', '2024-BHF-AM-23BF-NSC97b', '2024-BHF-AM-23BF-NSC97c',
'2024-BHF-AM-23BF-NSC97d', '2024-BHF-AM-23BF-NSC97e', '2024-BHF-AM-23BF-NSC97f',
'2008-QMC-NM-swave', '2010-QMC-NM-AV4', '2009-DLQMC-NM', '2014-AFQMC-NM',
```

'2016-QMC-NM', '2018-QMC-NM', '2024-QMC-NM', '2013-MBPT-NM', '2016-MBPT-AM',
'2019-MBPT-AM-L59', '2019-MBPT-AM-L69', '2020-MBPT-AM', '2024-NLEFT-AM']

Select a few microscopic models:

[18]:
```
models = [ '2016-QMC-NM', '2016-MBPT-AM', '2016-MBPT-AM' ]
```

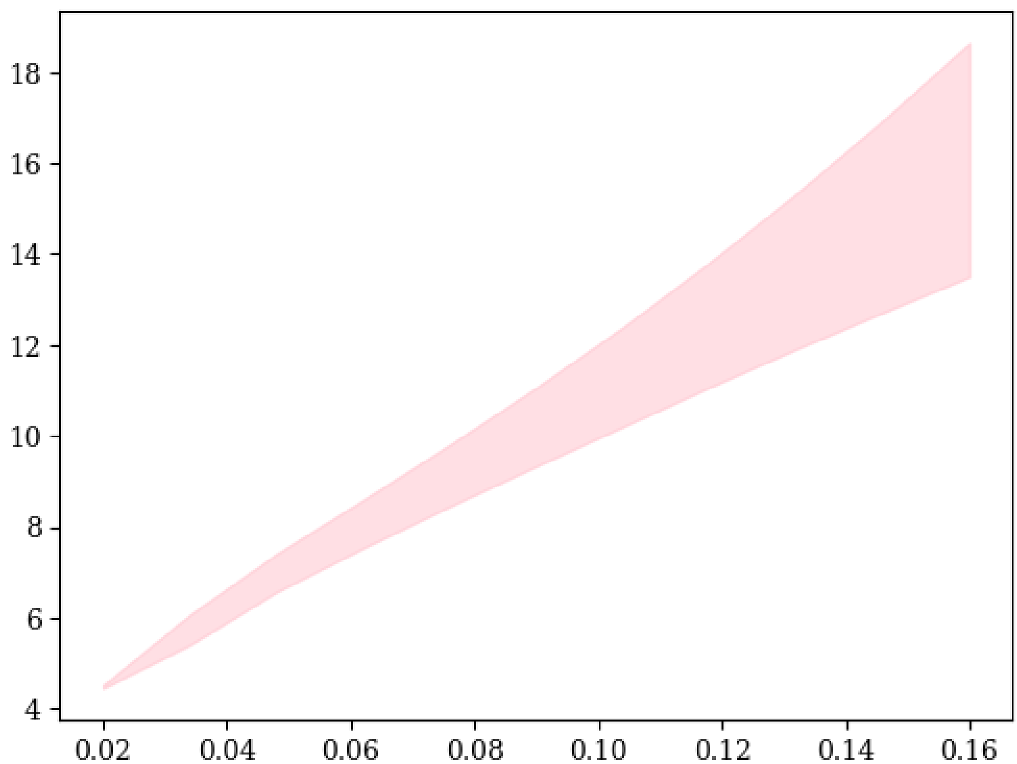
The reference band in NM is then calculated in the following way:

[19]:
```
band = nuda.matter.setupMicroBand( models, matter='NM' )
#band.print_outputs( )
plt.fill_between( band.den, y1=band.e2a_int - band.e2a_std, y2=band.e2a_int +
↪band.e2a_std, color=band.color, alpha=band.alpha )
```

```
Band in NM
model ->  2016-QMC-NM
flag_nm: True
flag_sm: False
flag_kf: True
flag_den: False
model ->  2016-MBPT-AM
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
model ->  2016-MBPT-AM
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
model ->  2016-QMC-NM
flag_nm: True
flag_sm: False
flag_kf: True
flag_den: False
model ->  2016-MBPT-AM
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
model ->  2016-MBPT-AM
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
model ->  2016-QMC-NM
flag_nm: True
flag_sm: False
```

```
flag_kf: True
flag_den: False
model ->  2016-MBPT-AM
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
model ->  2016-MBPT-AM
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
```

[19]: `<matplotlib.collections.FillBetweenPolyCollection at 0x70986e73a660>`



Plot:

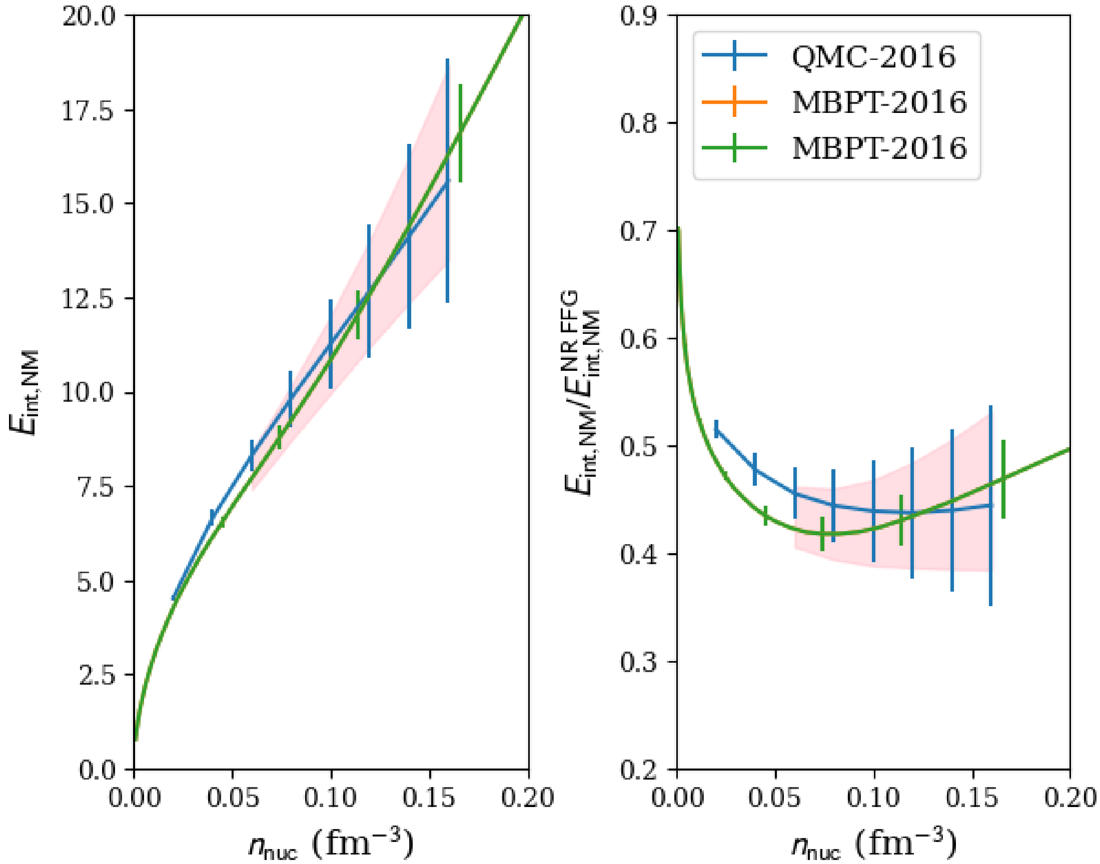
[20]:
```
den = np.linspace(0.06,0.16,num=6)
print('den:',den)
nuda.fig.matter_setupMicro_band_fig( None, models, den=den, matter='NM' )
```

```
den: [0.06 0.08 0.1  0.12 0.14 0.16]
Plot name: None
```

```
model ->  2016-QMC-NM
flag_nm: True
flag_sm: False
flag_kf: True
flag_den: False
model: 2016-QMC-NM
model ->  2016-MBPT-AM
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
model: 2016-MBPT-AM
model ->  2016-MBPT-AM
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
model: 2016-MBPT-AM

Band in NM
model ->  2016-QMC-NM
flag_nm: True
flag_sm: False
flag_kf: True
flag_den: False
model ->  2016-MBPT-AM
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
model ->  2016-MBPT-AM
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
model ->  2016-QMC-NM
flag_nm: True
flag_sm: False
flag_kf: True
flag_den: False
model ->  2016-MBPT-AM
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
model ->  2016-MBPT-AM
flag_nm: True
flag_sm: True
```

```
flag_kf: False
flag_den: True
```



Check if a model is inside the reference band in NM:

[21]:
```python
# define the reference band:
den = np.linspace(0.06,0.16,num=6)
models = [ '2016-MBPT-AM', '2016-QMC-NM', '2020-MBPT-AM' ]
band = nuda.matter.setupMicroBand( models, den=den, matter='NM' )
# define the model:
model = '1998-VAR-AM-APR'
micro = nuda.matter.setupMicro( model = model )
# check if the model is inside the reference band:
check = nuda.matter.setupCheck( eos = micro, band = band )
print('eos inside band?',check.isInside)
print('eos outside band?',check.isOutside)
```

```
Band in NM
model ->  2016-MBPT-AM
```

```
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
model ->  2016-QMC-NM
flag_nm: True
flag_sm: False
flag_kf: True
flag_den: False
model ->  2020-MBPT-AM
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
model ->  2016-MBPT-AM
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
model ->  2016-QMC-NM
flag_nm: True
flag_sm: False
flag_kf: True
flag_den: False
model ->  2020-MBPT-AM
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
model ->  1998-VAR-AM-APR
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
eos inside band? True
eos outside band? False
```

plot energy per particle in NM:

[22]:
```
micro_mbs, micro_mbs_lower = nuda.matter.micro_mbs()
pheno_models = [ 'Skyrme', 'NLRH', 'DDRH', 'DDRHF' ]
nuda.fig.matter_all_e2a_fig( None, micro_mbs, pheno_models, band, band, matter =␣
↪'NM' )
```

```
Plot name: None
model ->  1981-VAR-AM-FP
flag_nm: True
flag_sm: True
```

```
flag_kf: True
flag_den: False
mb: VAR model: 1981-VAR-AM-FP
model ->  1998-VAR-AM-APR
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: VAR model: 1998-VAR-AM-APR
model ->  2012-AFDMC-NM-RES-1
flag_nm: True
flag_sm: False
flag_kf: False
flag_den: True
mb: AFDMC model: 2012-AFDMC-NM-RES-1
model ->  2012-AFDMC-NM-RES-2
flag_nm: True
flag_sm: False
flag_kf: False
flag_den: True
mb: AFDMC model: 2012-AFDMC-NM-RES-2
model ->  2012-AFDMC-NM-RES-3
flag_nm: True
flag_sm: False
flag_kf: False
flag_den: True
mb: AFDMC model: 2012-AFDMC-NM-RES-3
model ->  2012-AFDMC-NM-RES-4
flag_nm: True
flag_sm: False
flag_kf: False
flag_den: True
mb: AFDMC model: 2012-AFDMC-NM-RES-4
model ->  2012-AFDMC-NM-RES-5
flag_nm: True
flag_sm: False
flag_kf: False
flag_den: True
mb: AFDMC model: 2012-AFDMC-NM-RES-5
model ->  2012-AFDMC-NM-RES-6
flag_nm: True
flag_sm: False
flag_kf: False
flag_den: True
mb: AFDMC model: 2012-AFDMC-NM-RES-6
model ->  2012-AFDMC-NM-RES-7
flag_nm: True
flag_sm: False
```

```
flag_kf: False
flag_den: True
mb: AFDMC model: 2012-AFDMC-NM-RES-7
model ->  2012-AFDMC-NM-FIT-1
flag_nm: True
flag_sm: False
flag_kf: False
flag_den: False
mb: AFDMC model: 2012-AFDMC-NM-FIT-1
model ->  2012-AFDMC-NM-FIT-2
flag_nm: True
flag_sm: False
flag_kf: False
flag_den: False
mb: AFDMC model: 2012-AFDMC-NM-FIT-2
model ->  2012-AFDMC-NM-FIT-3
flag_nm: True
flag_sm: False
flag_kf: False
flag_den: False
mb: AFDMC model: 2012-AFDMC-NM-FIT-3
model ->  2012-AFDMC-NM-FIT-4
flag_nm: True
flag_sm: False
flag_kf: False
flag_den: False
mb: AFDMC model: 2012-AFDMC-NM-FIT-4
model ->  2012-AFDMC-NM-FIT-5
flag_nm: True
flag_sm: False
flag_kf: False
flag_den: False
mb: AFDMC model: 2012-AFDMC-NM-FIT-5
model ->  2012-AFDMC-NM-FIT-6
flag_nm: True
flag_sm: False
flag_kf: False
flag_den: False
mb: AFDMC model: 2012-AFDMC-NM-FIT-6
model ->  2012-AFDMC-NM-FIT-7
flag_nm: True
flag_sm: False
flag_kf: False
flag_den: False
mb: AFDMC model: 2012-AFDMC-NM-FIT-7
model ->  2022-AFDMC-NM
flag_nm: True
flag_sm: False
```

```
flag_kf: False
flag_den: True
mb: AFDMC model: 2022-AFDMC-NM
model ->  2024-BHF-AM-2BF-Av18
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: BHF2 model: 2024-BHF-AM-2BF-Av18
model ->  2024-BHF-AM-2BF-BONN
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: BHF2 model: 2024-BHF-AM-2BF-BONN
model ->  2024-BHF-AM-2BF-CDBONN
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: BHF2 model: 2024-BHF-AM-2BF-CDBONN
model ->  2024-BHF-AM-2BF-NSC97a
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: BHF2 model: 2024-BHF-AM-2BF-NSC97a
model ->  2024-BHF-AM-2BF-NSC97b
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: BHF2 model: 2024-BHF-AM-2BF-NSC97b
model ->  2024-BHF-AM-2BF-NSC97c
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: BHF2 model: 2024-BHF-AM-2BF-NSC97c
model ->  2024-BHF-AM-2BF-NSC97d
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: BHF2 model: 2024-BHF-AM-2BF-NSC97d
model ->  2024-BHF-AM-2BF-NSC97e
flag_nm: True
flag_sm: True
```

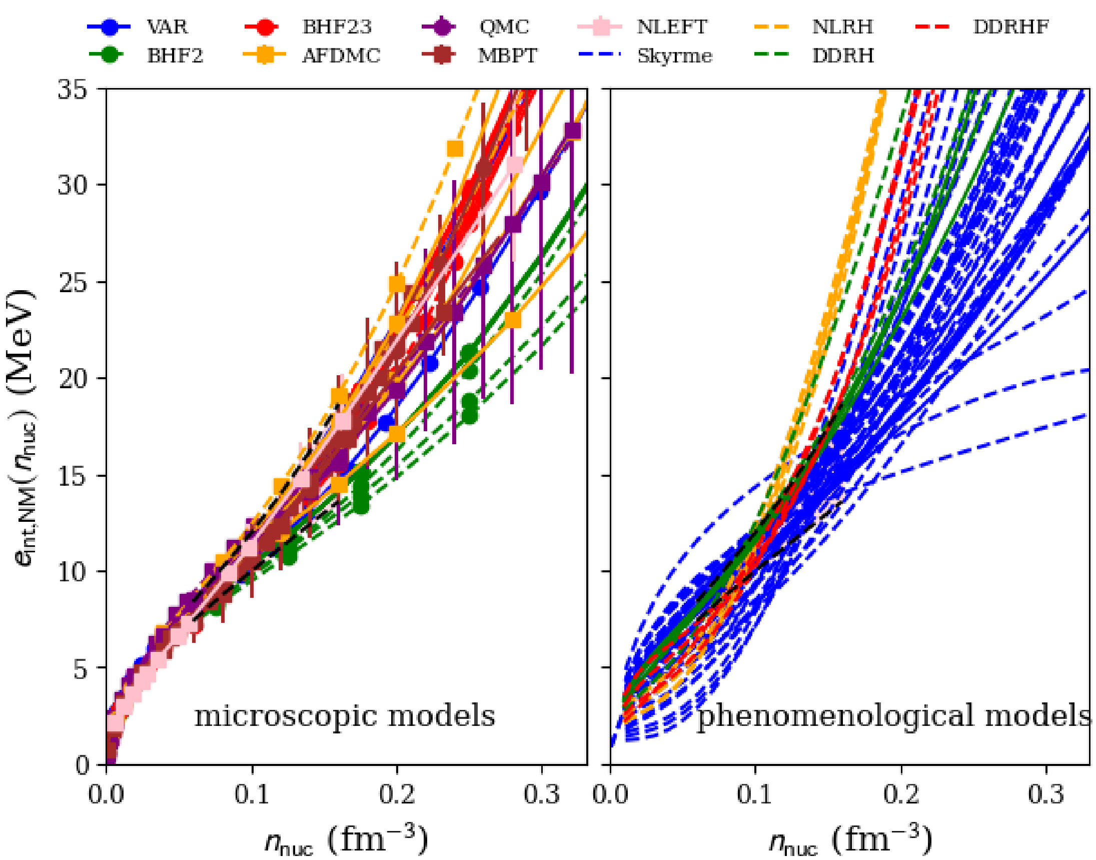
```
flag_kf: False
flag_den: True
mb: BHF2 model: 2024-BHF-AM-2BF-NSC97e
model ->  2024-BHF-AM-2BF-NSC97f
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: BHF2 model: 2024-BHF-AM-2BF-NSC97f
model ->  2006-BHF-AM-Av18
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: BHF23 model: 2006-BHF-AM-Av18
model ->  2024-BHF-AM-23BF-Av18
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: BHF23 model: 2024-BHF-AM-23BF-Av18
model ->  2024-BHF-AM-23BF-BONN
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: BHF23 model: 2024-BHF-AM-23BF-BONN
model ->  2024-BHF-AM-23BF-CDBONN
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: BHF23 model: 2024-BHF-AM-23BF-CDBONN
model ->  2024-BHF-AM-23BF-NSC97a
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: BHF23 model: 2024-BHF-AM-23BF-NSC97a
model ->  2024-BHF-AM-23BF-NSC97b
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: BHF23 model: 2024-BHF-AM-23BF-NSC97b
model ->  2024-BHF-AM-23BF-NSC97c
flag_nm: True
flag_sm: True
```

```
flag_kf: False
flag_den: True
mb: BHF23 model: 2024-BHF-AM-23BF-NSC97c
model ->  2024-BHF-AM-23BF-NSC97d
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: BHF23 model: 2024-BHF-AM-23BF-NSC97d
model ->  2024-BHF-AM-23BF-NSC97e
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: BHF23 model: 2024-BHF-AM-23BF-NSC97e
model ->  2024-BHF-AM-23BF-NSC97f
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: BHF23 model: 2024-BHF-AM-23BF-NSC97f
model ->  2008-QMC-NM-swave
flag_nm: True
flag_sm: False
flag_kf: True
flag_den: False
mb: QMC model: 2008-QMC-NM-swave
model ->  2010-QMC-NM-AV4
flag_nm: True
flag_sm: False
flag_kf: True
flag_den: False
mb: QMC model: 2010-QMC-NM-AV4
model ->  2009-DLQMC-NM
flag_nm: True
flag_sm: False
flag_kf: True
flag_den: False
mb: QMC model: 2009-DLQMC-NM
model ->  2014-AFQMC-NM
flag_nm: True
flag_sm: False
flag_kf: True
flag_den: False
mb: QMC model: 2014-AFQMC-NM
model ->  2016-QMC-NM
flag_nm: True
flag_sm: False
```

```
flag_kf: True
flag_den: False
mb: QMC model: 2016-QMC-NM
model ->  2018-QMC-NM
flag_nm: True
flag_sm: False
flag_kf: True
flag_den: False
mb: QMC model: 2018-QMC-NM
model ->  2024-QMC-NM
flag_nm: True
flag_sm: False
flag_kf: False
flag_den: True
mb: QMC model: 2024-QMC-NM
model ->  2013-MBPT-NM
flag_nm: True
flag_sm: False
flag_kf: False
flag_den: True
mb: MBPT model: 2013-MBPT-NM
model ->  2016-MBPT-AM
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: MBPT model: 2016-MBPT-AM
model ->  2019-MBPT-AM-L59
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: MBPT model: 2019-MBPT-AM-L59
model ->  2019-MBPT-AM-L69
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: MBPT model: 2019-MBPT-AM-L69
model ->  2020-MBPT-AM
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: True
mb: MBPT model: 2020-MBPT-AM
model ->  2024-NLEFT-AM
sm_popt: [-9.17066389 10.01180306 -3.0695997 ]
sm_pcov: [[ 0.50462836 -0.83636896  0.34040806]
```

```
 [-0.83636896  1.40263142 -0.5763757 ]
 [ 0.34040806 -0.5763757   0.23873678]]
nm_popt: [-1.18794133  0.71930878 -0.11294907]
nm_pcov: [[ 0.01360533 -0.01702127  0.00515285]
 [-0.01702127  0.02165016 -0.00663766]
 [ 0.00515285 -0.00663766  0.00205604]]
flag_nm: True
flag_sm: True
flag_kf: False
flag_den: False
mb: NLEFT model: 2024-NLEFT-AM
model: Skyrme  param: BSK14
model: Skyrme  param: BSK16
model: Skyrme  param: BSK17
model: Skyrme  param: BSK27
model: Skyrme  param: BSkG1
model: Skyrme  param: BSkG2
model: Skyrme  param: F-
model: Skyrme  param: F+
model: Skyrme  param: F0
model: Skyrme  param: FPL
model: Skyrme  param: LNS
model: Skyrme  param: LNS1
model: Skyrme  param: LNS5
model: Skyrme  param: NRAPR
model: Skyrme  param: RATP
model: Skyrme  param: SAMI
model: Skyrme  param: SGII
model: Skyrme  param: SIII
model: Skyrme  param: SKGSIGMA
model: Skyrme  param: SKI2
model: Skyrme  param: SKI4
model: Skyrme  param: SKMP
model: Skyrme  param: SKMS
model: Skyrme  param: SKO
model: Skyrme  param: SKOP
model: Skyrme  param: SKP
model: Skyrme  param: SKRSIGMA
model: Skyrme  param: SKX
model: Skyrme  param: Skz2
model: Skyrme  param: SLY4
model: Skyrme  param: SLY5
model: Skyrme  param: SLY230A
model: Skyrme  param: SLY230B
model: Skyrme  param: SV
model: Skyrme  param: T6
model: Skyrme  param: T44
model: Skyrme  param: UNEDF0
```

```
model: Skyrme  param: UNEDF1
model: NLRH  param: NL-SH
model: NLRH  param: NL3
model: NLRH  param: NL3II
model: NLRH  param: PK1
model: NLRH  param: PK1R
model: NLRH  param: TM1
model: DDRH  param: DDME1
model: DDRH  param: DDME2
model: DDRH  param: DDMEd
model: DDRH  param: PKDD
model: DDRH  param: TW99
model: DDRHF  param: PKA1
model: DDRHF  param: PKO1
model: DDRHF  param: PKO2
model: DDRHF  param: PKO3
```



1.3 Module: Crust

The complete list of available crust models:

[23]:
```python
models, models_lower = nuda.crust.crust_models( )
print(models)
```

```
['1973-Negele-Vautherin', '2018-PCPFDDG-BSK22', '2018-PCPFDDG-BSK24',
'2018-PCPFDDG-BSK25', '2018-PCPFDDG-BSK26', '2020-MVCD-D1S', '2020-MVCD-D1M',
'2020-MVCD-D1MS', '2022-GMRS-BSK14', '2022-GMRS-BSK16', '2022-GMRS-DHSL59',
'2022-GMRS-DHSL69', '2022-GMRS-F0', '2022-GMRS-H1', '2022-GMRS-H2',
'2022-GMRS-H3', '2022-GMRS-H4', '2022-GMRS-H5', '2022-GMRS-H7', '2022-GMRS-
LNS5', '2022-GMRS-RATP', '2022-GMRS-SGII', '2022-GMRS-SLY5']
```

Select a crust models: '2020-MVCD-D1S'

[24]:
```python
crust = nuda.crust.setupCrust( model = '2020-MVCD-D1S' )
print('model:',crust.model)
print('ref:',crust.ref)
print('notes:',crust.note)
```

```
model: 2020-MVCD-D1S
ref: C. Mondal, X. Viñas, M. Centelles, and J.N. De, Phys. Rev. C 102, 015802
(2020).
notes: semiclassical variational Wigner-Kirkwood method along with shell and
pairing corrections calculated with the Strutinsky integral method and the BCS
approximation.
```

[25]:
```python
crust = nuda.crust.setupCrust( model = '2022-GMRS-H4' )
print('model:',crust.model)
print('ref:',crust.ref)
print('notes:',crust.note)
```

```
model: 2022-GMRS-H4
ref: G. Grams, J. Margueron, R. Somasundaram, and S. Reddy, EPJA 58, 56 (2022).
notes: .
```

1.4 Module: EOS

[26]:
```python
models , models_lower = nuda.matter.micro_esym_models()
print(models)
```

```
['1981-VAR-AM-FP', '1998-VAR-AM-APR', '1998-VAR-AM-APR-fit', '2024-BHF-AM-2BF-
Av18', '2024-BHF-AM-2BF-BONN', '2024-BHF-AM-2BF-CDBONN', '2024-BHF-AM-2BF-
NSC97a', '2024-BHF-AM-2BF-NSC97b', '2024-BHF-AM-2BF-NSC97c', '2024-BHF-AM-2BF-
NSC97d', '2024-BHF-AM-2BF-NSC97e', '2024-BHF-AM-2BF-NSC97f', '2006-BHF-AM-Av18',
'2024-BHF-AM-23BF-Av18', '2024-BHF-AM-23BF-BONN', '2024-BHF-AM-23BF-CDBONN',
'2024-BHF-AM-23BF-NSC97a', '2024-BHF-AM-23BF-NSC97b', '2024-BHF-AM-23BF-NSC97c',
```

```
'2024-BHF-AM-23BF-NSC97d', '2024-BHF-AM-23BF-NSC97e', '2024-BHF-AM-23BF-NSC97f',
'2019-MBPT-AM-L59', '2016-MBPT-AM', '2019-MBPT-AM-L69', '2020-MBPT-AM',
'2024-NLEFT-AM']
```

2 Module: Astro

Masses of neutron stars:

[27]:
```
sources, sources_lower = nuda.astro.masses_sources( )
print('sources:', sources )
```

```
sources: ['J1614-2230', 'J0348+0432', 'J2215+5135', 'J0740+6620', 'J1600+3053']
```

3 Conclusions

- The goal of this numerical tool is to provide an easy way to access most of the available theoretical, experimental, and astrophysical data.
- The toolkit offers modular functions that average over several predictions, for which the user is free to decide which one to consider.
- Please share with us missing data. We will be very pleased to add them to the toolkit.

[ ]:
```

```