

My notes on model variations provided

D. L. Whittenbury

Written April, 2019 and reformatted March, 2020

Contents

Introduction	1
Models	1
1) Standard variation	1
2) $\Lambda = 1.0$ GeV	2
3) $\Lambda = 1.1$ GeV	2
4) $\Lambda = 1.2$ GeV	2
5) $\Lambda = 1.3$ GeV	2
6) $\Lambda = 1.1$ GeV, $g_{\sigma Y} \times 1.3$	2
7) $\Lambda = 1.3$ GeV, $g_{\sigma Y} \times 1.3$	2
8) $\Lambda = 2.0$ GeV, $g_{\sigma Y} \times 1.9$	2
9) Increased $f_{\rho N}/g_{\rho N}$	3
10) Fock $\delta\sigma$	3
11) Effective proton mass	3
12) Eff. Proton Mass, $\Lambda = 1.1$ GeV	3
13) Eff. Proton Mass, $\delta\sigma$	4
14) Dirac only	4
15) Hartree only	4
16) $R = 0.8$	4
17) App. $S_0 = 32.5$ MeV	4
18) App. $S_0 = 30$ MeV	4
19) $S_0 = 30$ MeV	5
20) Standard variation but nucleon only beta-equilibrium calculation	5
21) MyQMC700 (Not published)	5
Using NJL model mass parameterisations	6
1) Hartree	6
2) Standard variation but with NJL mass parameterisation	6
3) $\Lambda = 1.3$ GeV	6
4) Dirac only	6
5) $F_\sigma = 1$	6
PYTOV: Python implementation of a TOV equation integrator	6

Introduction

This is a very rough set of notes on my calculations. They can be used with my code to reproduce published results. The names of model variations are as in my thesis and published papers. You should consult these for further details. A summary of options can be found in the corresponding `model_settings.dat` files for each variation. Here I give a few additional comments on how the calculations were performed.

When re-running model variations I suggest to first set all 5 matter options (`SymmetricMatLong`, ...) to false and checking the saturation properties against what is given in `snm_saturation_properties.dat` and `model_settings.dat`. Then proceed with one of the matter calculations. Please note the code can be sensitive to the initial guess of the sigma mean field. If an error occurs, changing the guess by about an MeV or so will allow the solver to find a solution.

Models

1) Standard variation

- This is the model variation to which I compare all other variations with. The code is set up to perform this model variation when this code was shared. All other model variations are described relative to this variation.
- Sometimes the IMSL non-linear solver can be sensitive to initial guesses and a slightly different guess will allow the solver to find a solution. For reference I set the guess for the sigma mean field to 22 MeV for `SymmetricMatLong`, `NeutronMatter`, `NeutronMatterHigh` and `SymTable`. However, for `BetaEquilibrium` I needed to use 23 MeV, i.e., in `qmc_driver.f90`

`SIG_GUESS = 23.0_dp`

If you have issues with not finding a solution, first try to change SIG_GUESS. Generally, a change of an MeV or so is enough.

- For reference, BetaEquilibrium takes the longest out of all the options. In this case on my Macbook pro it took ~ 15 mins.

2) $\Lambda = 1.0$ GeV

- This model variation only differs from the default by setting

CutOff = 1000.0_dp

It simply increases the Fock term cut-off from 0.9 to 1.3 GeV.

- Guess for sigma set to 22 MeV worked.

SIG_GUESS = 22.0_dp

3) $\Lambda = 1.1$ GeV

- This model variation only differs from the default by setting

CutOff = 1100.0_dp

- Guess for sigma set to 23 MeV worked.

SIG_GUESS = 23.0_dp

4) $\Lambda = 1.2$ GeV

- This model variation only differs from the default by setting

CutOff = 1200.0_dp

- Guess for sigma set to 23 MeV worked.

SIG_GUESS = 23.0_dp

5) $\Lambda = 1.3$ GeV

- This model variation only differs from the default by setting

CutOff = 1300.0_dp

- Guess for sigma set to 23 MeV worked for all 5 options.

SIG_GUESS = 23.0_dp

6) $\Lambda = 1.1$ GeV, $g_{\sigma Y} \times 1.3$

- Firstly,

CutOff = 1100.0_dp

- This variation also phenomenologically rescales the sigma hyperon couplings. The options that must be changed to reproduce this variation are well documented in qmc_driver.f90. All matter options should all be set to false, obtain the couplings and follow the remaining instructions.

- Guess for sigma set to 23 MeV worked.

SIG_GUESS = 23.0_dp

7) $\Lambda = 1.3$ GeV, $g_{\sigma Y} \times 1.3$

- Similar to the other phenomenologically rescaled variations.
- Guess for sigma set to 21 MeV worked.

SIG_GUESS = 21.0_dp

8) $\Lambda = 2.0$ GeV, $g_{\sigma Y} \times 1.9$

- Similar to the other phenomenologically rescaled variations.
- Guess for sigma set to 22 MeV worked.

SIG_GUESS = 22.0_dp

9) Increased $f_{\rho N}/g_{\rho N}$

- Change

`kappasb = Kappa_IS(iH) * kapRescale`

to

`kappasb = Kappa_IS_Nij(iH) * kapRescale`

in `qmc_subroutines.f90` in `OmegaAngularPt()` L~1011.

- Change

`kappavb = Kappa_IV(iH) * kapRescale`

to

`kappavb = Kappa_IV_Nij(iH) * kapRescale`

in `qmc_subroutines` in `RhoAngularPt()` L~1116

- There is no option implemented for these changes.
- Guess for sigma set to 23 MeV worked.

`SIG_GUESS = 23.0_dp`

10) Fock $\delta\sigma$

- This variation includes a Fock correction to the scalar mean field arising from the dependence of the Fock terms on the mean field. This option takes quite some time to run compared to the other variations, because of the extra contribution. It is a small correction and this is why I do not implement for every variation.

`minfock = .true.`

- Everything else is as in the *Standard* variation .
- Just saturation properties ~ 1 min
- SymmetricMatLong ~ 25 mins
- SymTable ~ 13 mins
- NeutronMatter ~ 5 mins
- NeutronMatterHigh ~ 3 mins
- BetaEquilibrium ~ 5.5 hrs
- Guess for sigma set to 23 MeV worked.

`SIG_GUESS = 23.0_dp`

11) Effective proton mass

- Need to change

`kapRescale = MStar/Mass(iproton,0.0_dp)`

to

`kapRescale = MStar/Mass(iproton,Sig)`

in `qmc_subroutines.f90` in `OmegaAngularPt()` L~1008 and similarly in `RhoAngularPt()` L~1113. An option was not implemented for this.

- Guess for sigma set to 23 MeV worked.

`SIG_GUESS = 23.0_dp`

12) Eff. Proton Mass, $\Lambda = 1.1$ GeV

- Guess for sigma set to 23 MeV worked.

`SIG_GUESS = 23.0_dp`

13) Eff. Proton Mass, $\delta\sigma$

- Fock correction to the scalar field. It is a long calculation.

`minfock = .true.`

- Guess for sigma set to 27 MeV worked.

`SIG_GUESS = 27.0_dp`

Prefers larger not smaller.

14) Dirac only

- Tensor coupling is turned off.

`TensorCoupling = .false.`

- Guess for sigma set to 22 MeV worked.

`SIG_GUESS = 22.0_dp`

15) Hartree only

- Fock terms are turned off.

`incl_sigma_Fock = .false.`

`incl_omega_Fock = .false.`

`incl_rho_Fock = .false.`

`incl_pi_Fock = .false.`

- You may also want to set to the analytic expression for the symmetry energy. To use the analytic expression of the symmetry energy when fitting coupling constants set

`asym = 1`

- Note numerical derivatives will still be performed for other nuclear matter properties, e.g., L_0 . Note that when using the SymTable option, you should be aware that numerical approximations are always used to calculate the symmetry energy here. There is an analytic formula implemented in the code, but it is not used in the SymTable option. Only the approximate expressions are used. It would be a trivial modification to use the analytic expression, but I didn't think it was worth implementing.

- Guess for sigma set to 19 MeV worked.

`SIG_GUESS = 19.0_dp`

16) $R = 0.8$

- The free nucleon radius is set to 0.8 fm, this is used in the bag model mass parameterisation.

`RNFree = 0.8_dp`

- Guess for sigma set to 23 MeV worked.

`SIG_GUESS = 23.0_dp`

17) App. $S_0 = 32.5$ MeV

- Approximate expression used for the symmetry energy (Guichon/Stone papers)

`asym = 0`

- Guess for sigma set to 23 MeV worked.

`SIG_GUESS = 23.0_dp`

18) App. $S_0 = 30$ MeV

- Guess for sigma set to 20 MeV worked.

`SIG_GUESS = 20.0_dp`

- Values of S_0 is changed. It is the same value used by (Guichon/Stone)

`as = guichon_sat_sym_en`

19) $S_0 = 30$ MeV

- Numerical differentiation is used for the symmetry energy.

asym = 2

- Guess for sigma set to 21 MeV worked.

SIG_GUESS = 21.0_dp

as = guichon_sat_sym_en

20) Standard variation but nucleon only beta-equilibrium calculation

incl_hyperons = .false.

Guess for sigma set to 23 MeV worked.

SIG_GUESS = 23.0_dp

21) MyQMC700 (Not published)

To most closely reproduce the scenario QMC700 in Stone et al. Nucl. Phys. A 792 (2007) 341–369 one can calculate MyQMC700. I have calculated it in both Fortran and Mathematica and they produce the same results. There are small difference between MyQMC700 Stone et al.'s QMC700, they are: - A different mass parameterisation is used. I use a slightly newer mass parameterisation, see references in Appendix A of my thesis. - Symmetry energy in Stone et al. is evaluated using a difference formula. I use a finite difference approximation for the derivative to calculate the symmetry energy by default, but there is an option in the Fortran code to do the difference approximation that Stone et al. uses. However, the finite difference approximation seems to best approximate QMC700. - The numerical values for various constants, i.e., baryon and meson masses may be slightly different than what was used by Stone et al. - The effective sigma meson mass found in the Fock term of Stone et al can be included using the Fortran code. Stone et al. don't explicitly say how then calculate it, but in an earlier paper they appear to use a non-relativistic approximation (ρ_s is approximately ρ_v). It should be ρ_s , either one can be used in the Fortran code.

The key options to set in qmc_driver.f90 to get MyQMC700:

- There is no pion contribution in QMC700

incl_sigma_Fock = .true.

incl_omega_Fock = .true.

incl_rho_Fock = .true.

incl_pi_Fock = .false.

- The structure of the Fock terms are simple, in particular the angular integrals can be done analytically.

Sigma_Basic = .true.

Omega_Basic = .true.

Rho_Basic = .true.

Pion_Basic = .true. ! Doesn't matter because pion is switched off

- The sigma meson mass changes in-medium, but only in the sigma Fock term

InMedSigMass = .true.

- Also set how you want to calculate the effective sigma meson mass, either with ρ_v or ρ_s .

! ADDITIONAL: Calculate using \rho_v (NRApprox = .true.) or \rho_s (Default)

NRApprox = .true.

- The free nucleon radius used in the bag mass parameterisation is 0.8 fm

RNFree = 0.8_dp ! the default is 1.0_dp

QMC_Bag = .true.

- The value of the symmetry energy at saturation is 30 MeV not the default value 32.5~MeV

as = guichon_sat_sym_en

- The numerical derivative seems to give the closest approximation to QMC700. In the accompanying data I used

asym = 2

Although, if you want to use the difference formula for the symmetry energy then you must change

asym = 0

- Guess for sigma set to 19 MeV worked.

SIG_GUESS = 19.0_dp

Using NJL model mass parameterisations

Minor bug in cascade mass parameterisation has been corrected.

1) Hartree

Beta-equilibrium data not published. The general setup is very much the same as the QMC bag model cases. First

- Turn off bag model mass parameterisation

```
QMC_Bag = .false.
```

and turn on the NJL mass parameterisation obtained by Manuel

```
cOpt = 5
```

```
QMC_NJL = .true.
```

Analytic formula for the symmetry energy

```
asym = 1
```

Guess for sigma set to 21 MeV worked.

```
SIG_GUESS = 21.0_dp
```

2) Standard variation but with NJL mass parameterisation

Beta-equilibrium data not published. Similar to previous cases.

- Guess for sigma set to 23 MeV worked.

```
SIG_GUESS = 23.0_dp
```

3) $\Lambda = 1.3 \text{ GeV}$

Beta-equilibrium data not published. Similar to previous cases.

- Guess for sigma set to 23 MeV worked.

```
SIG_GUESS = 23.0_dp
```

4) Dirac only

Beta-equilibrium data not published. - The tensor coupling is turned off.

```
TensorCoupling = .false.
```

- Guess for sigma set to 19 MeV worked.

```
SIG_GUESS = 19.0_dp
```

5) $F_\sigma = 1$

Because the scalar coupling already has a decreasing density dependence, we omit the form factor for this meson only. Beta-equilibrium data not published.

- This scenario is the same as 2 except in qmc_subroutines.f90 in SigmaAngularPt() change (line ~943)

```
Fs = 1.0_dp/(1.0_dp + q2/Lambda**2)**2
```

to

```
Fs = 1.0_dp
```

An option was not implemented for this.

- Guess for sigma set to 19 MeV worked.

```
SIG_GUESS = 18.0_dp
```

PYTOV: Python implementation of a TOV equation integrator

See the README.md file in the PYTOV folder. All the above model variations have been used with this integrator and their resulting data can be found in the TOV folder.