Fortran Module for Density Dependent Parts of EDF

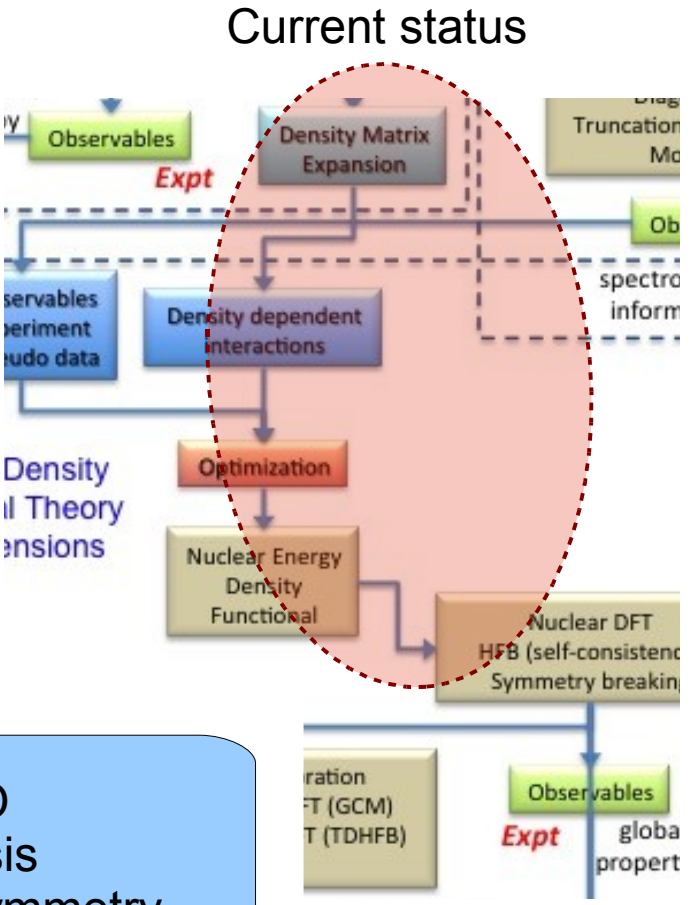
Markus Kortelainen

Department of Physics and Astronomy, UT, Knoxville, TN 37996, USA
Oak Ridge National Laboratory, Oak Ridge, TN 37831, USA

W. Nazarewicz, M. Stoitsov

General philosophy

- Main idea is to separate details of EDF and HFB solver
- All functional details coded in fortran module UNEDF: The same module for different HFB solvers
- At the moment HFBRAD and HFBTHO works with the module



UNEDF DME module (f90)
- contains DME and Skyrme functionals
- calculates numerical values of CCs and their derivatives
- inm. properties included
- i/o for functional parameters

HFBTHO
- HO basis
- axial-symmetry

HFBRAD
- coordinate space
- spherical symmetry

Used energy density

- Skyrme EDF is constructed from bilinear terms of density matrices and their derivatives to the 2nd order
- Each term multiplied by a constant coupling constant (except density dependent C^{ρ})

$$H_t^{even}(r) = C_t^{\rho}\rho_t^2 + C_t^{\tau}\rho_t\tau_t + C_t^{\Delta\rho}\rho_t\Delta\rho_t + C_t^{\nabla J}\rho_t\nabla J_t + C_t^{J}J_t^2$$

$$C_t^{\rho} = C_{t0}^{\rho} + \rho_0^{\gamma}C_{tD}^{\rho} \quad , \quad t = 0,1$$

- Generalize time-even H(r) to density dependent coupling constants

$$H_{tt'}^{even}(r) = U_{tt'}^{\rho\rho}\rho_t\rho_{t'} + U_{tt'}^{\rho\tau}\rho_t\tau_{t'} + U_{tt'}^{\rho\Delta\rho}\rho_t\Delta\rho_{t'} + U_{tt'}^{\nabla\rho\nabla\rho}\nabla\rho_t\cdot\nabla\rho_{t'}$$
$$+ U_{tt'}^{\rho\nabla J}\rho_t\nabla J_{t'} + U_{tt'}^{J\nabla\rho}J_t\cdot\nabla\rho_{t'} + U_t^{JJ}J_tJ_{t'}$$

- Amplitudes $U_{tt'}$ are functions of densities

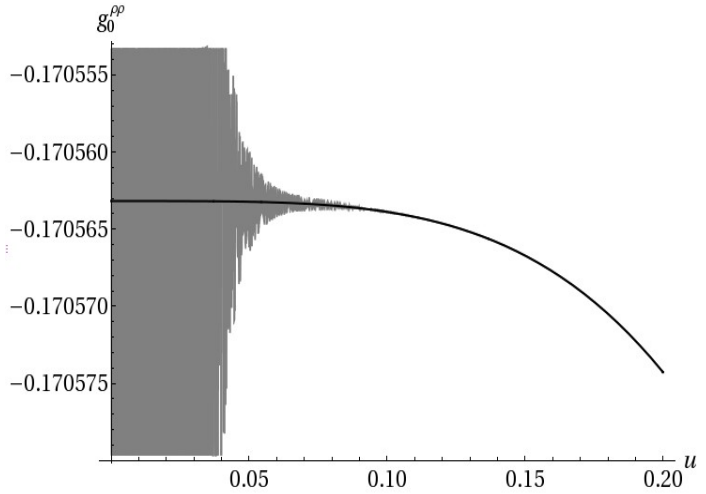
Used energy density

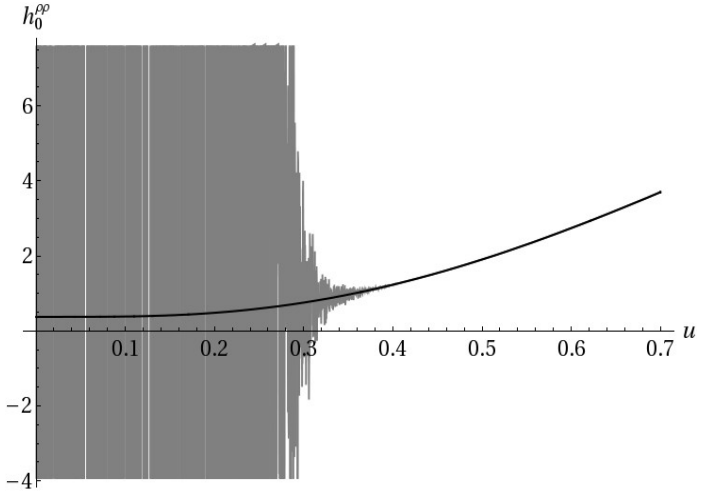
- For cross terms ($t \neq t'$) the fulfillment of the isospin invariance of the whole energy density term must be set to U function
- Due to density dependence $\rho\Delta\rho$ and $(\nabla\rho)^2$ no longer connected by partial integration (also $J\nabla\rho$ and $\rho\nabla J$)
- Time-odd terms not yet considered
- U amplitudes in module are public variables for hfb solver to use

```
!
! === PUBLIC VARIABLES ===
!
! Use pointers to prevent conflicts with UNEDF public variabes
! Example: Use UNEDF, pr=>my_pr, ipr=>my_ipr, Crho=>my_Crho ...
!
Integer, Parameter, Public :: pr=Kind(1.0D0), ipr=Kind(1)                ! to set the precision of the DFT solv
Logical, Public :: use_charge_density, use_cm_cor
Real(pr), Public, Dimension(0:3,0:7) :: Urhorho,Urhotau,UrhoDrho,Unablarho  ! ph DME amplitudes
Real(pr), Public, Dimension(0:3,0:7) :: UJnablarho,UrhonablaJ,UJJ
Real(pr), Public, Dimension(0:3,0:7) :: Urhorhopr                        ! pp amplitudes
Real(pr), Public, Dimension(0:1) :: UEnonstdr,UFnonstdr,URnonstdr        ! Other amplitudes
Real(pr), Public :: hbzero,sigma,e2charg                                 ! hbr^2/2m, DD sigma, e^2 charge
Real(pr), Public, Dimension(0:1) :: Crho,Cdrho,Ctau,CrDr,CrdJ,CJ,CpV0,CpV1  ! basic coupling constants
Real(pr), Public :: E_NM,K_NM,SMASS_NM,RHO_NM,ASS_NM,LASS_NM,VMASS_NM,P_NM,KA_NM
Real(pr), Public :: CHrho                                                ! Crho(0) from the Hartree term in NM
Logical, Public :: use_DME3N_terms,use_j2terms
Integer(ipr), Public :: DMEorder,DMElda
!
! === PRIVATE VARIABLES ===
!
Real(pr), Private, Dimension(0:1) :: nuCrho,nuCdrho,nuCtau,nuCrDr        ! basic coupling constants in natural
Real(pr), Private, Dimension(0:1) :: nuCrdJ,nuCJ,nuCpV0,nuCpV1           !
Real(pr), Private :: t0,t1,t2,t3,x0,x1,x2,x3,b4,b4p,te,to
Real(pr), Private :: nuLambda,nufpi                                      ! parameters associated to natural uni
```

Module for DME functional

- Based on EDF of B. Gebremariam, S. K. Bogner, and T. Duguet, Comput. Phys. Comm. 181 (2010) 1167
- Contains all expressions for U-amplitudes (and their derivatives) coming from the DME of chiral perturbation potential up to N2LO (2N & 3N terms) $\Rightarrow$ U is a function of ρ_0
- For very small ρ_0 Taylor expansion used to handle numerical noise





$u=k_F/m_\pi$

Conclusions & Outlook

- A fortran module for EDF coming from the DME of chiral perturbation potential has been developed
- The structure of module allows flexible way to code EDF for different HFB solvers
- Similar module also coded for Fayans functional, DME of Minnesota potential and NV DME of Gogny
- DME module to be published in CPC