
Nuclear structure from Nuclei to Neutron stars

M. Hjorth-Jensen

Department of Physics, University of Oslo, Norway

Twelfth Summer School in Nuclear Physics
University of California, Santa Cruz
July 3 - July 14, 2000

Preface

These notes include details on the nucleon-nucleon interaction which are not fully discussed during the lectures. In addition, I include two programs, one for solving the equation for the scattering matrix and phase shifts and one program for calculating the radius and mass of a neutron star, using a selected equation of state. Feel free to use these programs as you wish and please don't hesitate to ask in case something is unclear.

The scattering equation program can easily be extended to study medium corrections as well. Since I've chosen F90 as programming language, a brief intro to F90 is included in the appendix.

In these notes you will not find much on the finite nuclei part included in the oral presentation, or about neutron star details. If you wish to read more, you could look up the following references

1. M. Hjorth-Jensen, T.T.S. Kuo and E. Osnes, *Physics Reports* **261** (1995) 125-270.
2. H. Heiselberg and M. Hjorth-Jensen, *Physics Reports* **328** (2000) 237-330.

The last reference can be fetched in postscript format from

http://www.uio.no/~mhjensen/summer_2000/lecture.html

These lecture notes (postscript format) and the two programs discussed in the lectures can also be downloaded from the same URL address.

Enjoy, and remember, every question is welcome!

Contents

1	Essentials of nuclear many-body theory	1
1.1	Introduction	1
1.2	Brief survey of various features of the nucleon-nucleon interaction	4
1.2.1	The one-boson-exchange interaction and the Bethe-Salpeter equation	5
1.3	Numerical solution of the scattering equations	13
1.3.1	Program example	16
1.4	How to include in-medium correlations	23
1.4.1	The separation and reference-spectrum methods for calculating the G -matrix	24
1.4.2	The G -matrix for nuclear matter	29
1.5	Numerical evaluation of the G -matrix	32
2	Numerical studies of neutron stars	35
2.1	Introduction to the physics of neutron stars	35
2.2	Non-relativistic equilibrium equations	36
2.3	Numerical solution	37
2.4	TOV equation	40
2.5	Code example for a simple EoS	41
A	A Fortran 90/95 primer	47
A.1	Introduction to F90/F95	47
A.1.1	Introduction	47
A.1.2	DO construct	48
A.1.3	Logical constructs	48
A.1.4	Subprograms	49
A.1.5	Arrays	50
A.1.6	Allocate statement and mathematical operations on arrays	50
A.1.7	Complex variables	53

Chapter 1

Essentials of nuclear many-body theory

1.1 Introduction

The Brueckner G -matrix is probably the most important ingredient in many-body calculations of nuclear systems. In this section, we will briefly survey the philosophy behind the G -matrix.

Historically, the G -matrix was developed in microscopic nuclear matter calculations using realistic nucleon-nucleon (NN) interactions. It is an ingenious as well as an interesting method to overcome the difficulties caused by the strong, short-range repulsive core contained in all modern models for the NN interaction. The G -matrix method was originally developed by Brueckner [1], and further developed by Goldstone [2] and Bethe, Brandow and Petschek [3]. In the literature it is generally referred to as the Brueckner theory or the Brueckner-Bethe-Goldstone theory.

Suppose we want to calculate the nuclear matter ground-state energy E_0 using the non-relativistic Schrödinger equation

$$H \Psi_0(A) = E_0(A) \Psi_0(A), \quad (1.1)$$

with $H = T + V$ where A denotes the number of particles, T is the kinetic energy and V is the nucleon-nucleon (NN) potential. The corresponding unperturbed problem is

$$H_0 \psi_0(A) = W_0(A) \psi_0(A). \quad (1.2)$$

Here H_0 is just kinetic energy T and ψ_0 is a Slater determinant representing the Fermi sea, where all orbits through the Fermi momentum k_F are filled. We write

$$E_0 = W_0 + \Delta E_0, \quad (1.3)$$

where ΔE_0 is the ground-state energy shift. If we know how to calculate ΔE_0 , then we know E_0 , since W_0 is easily obtained. In the limit $A \rightarrow \infty$, the quantities E_0 and ΔE_0 themselves are not well defined, but the ratios E_0/A and $\Delta E_0/A$ are. The nuclear-matter binding energy per nucleon is commonly denoted by BE/A , which is just $-E_0/A$. In passing, we note that the empirical value for symmetric nuclear matter (proton number Z =neutron number N) is ≈ 16 MeV. There exists a formal theory for the calculation of ΔE_0 . According to the well-known Goldstone linked-diagram theory [2], the energy shift ΔE_0 is given exactly by the diagrammatic expansion shown in fig. 1.1. This theory, is a linked-cluster perturbation expansion for the ground state energy of a many-body system, and applies equally well to both nuclear matter and closed-shell nuclei such as the doubly magic nucleus ^{40}Ca . We will not discuss the Goldstone expansion, but rather discuss briefly how it is used in calculations. For systems with particles outside closed shells, a similar theory exists, namely the so-called folded-diagram theory.

Using the standard diagram rules the various diagrams contained in Fig. 1.1 can be readily calculated, viz. (in an uncoupled scheme) we have

$$(i) = \frac{(-)^{n_h+n_i}}{2^{n_{ep}}} \sum_{k_i, k_j \leq k_F} \langle k_i k_j | \tilde{V} | k_i k_j \rangle_{AS}, \quad (1.4)$$

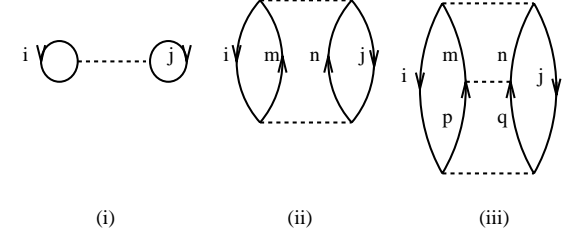


Figure 1.1: Diagrams which enter the definition of the ground-state shift energy ΔE_0 . Diagram (i) is first order in the interaction V , while diagrams (ii) and (iii) are examples of contributions to second and third order, respectively.

with $n_h = m = 2$ and $n_{ep} = 1$. n_h denotes the number of hole lines, n_i the number of closed fermion loops and n_{ep} is the number of so-called equivalent pairs. The factor $1/2^{n_{ep}}$ is needed since we want to count a pair of particles only once. We will carry this factor $1/2$ with us in the equations below. The subscript AS denotes the antisymmetrized and normalized matrix element

$$\langle k_i k_j | \tilde{V} | k_i k_j \rangle_{AS} = \langle k_i k_j | V | k_i k_j \rangle - \langle k_j k_i | V | k_i k_j \rangle. \quad (1.5)$$

Similarly, diagrams (ii) and (iii) read

$$(ii) = \frac{(-)^{2+2}}{2^2} \sum_{k_i, k_j \leq k_F} \sum_{k_m, k_n > k_F} \frac{\langle k_i k_j | \tilde{V} | k_m k_n \rangle_{AS} \langle k_m k_n | \tilde{V} | k_i k_j \rangle_{AS}}{\varepsilon_i + \varepsilon_j - \varepsilon_m - \varepsilon_n}, \quad (1.6)$$

and

$$(iii) = \frac{(-)^{2+2+2}}{2^3} \sum_{k_i, k_j \leq k_F} \sum_{k_m, k_n > k_F} \sum_{k_p, k_q > k_F} \frac{\langle k_i k_j | \tilde{V} | k_m k_n \rangle_{AS} \langle k_m k_n | \tilde{V} | k_p k_q \rangle_{AS} \langle k_p k_q | \tilde{V} | k_i k_j \rangle_{AS}}{(\varepsilon_i + \varepsilon_j - \varepsilon_m - \varepsilon_n)(\varepsilon_i + \varepsilon_j - \varepsilon_p - \varepsilon_q)}$$

In the above, ε denotes the sp energies defined by H_0 . The steps leading to the above expressions for the various diagrams are rather straightforward. Though, if we wish to compute the matrix elements for the potential V , a serious problem arises. Typically, the matrix elements will contain a term $V(|\mathbf{r}|)$, which represents the interaction potential V between two nucleons, where \mathbf{r} is the internucleon distance. All modern models for V have a strong short-range repulsive core. Hence, matrix elements involving $V(|\mathbf{r}|)$, will result in large (or infinitely large for a potential with a hard core) and repulsive contributions to the ground-state energy. Thus, the diagrammatic expansion for the ground-state energy in terms of the potential $V(|\mathbf{r}|)$ becomes meaningless.

The resolution of this problem is provided by the well-known Brueckner theory or the Brueckner G -matrix, or just the G -matrix. In fact, the G -matrix is an indispensable tool in almost every microscopic nuclear structure calculation. Its main idea may be paraphrased as follows. Suppose we want to calculate the function $f(x) = x/(1+x)$. If x is small, we may expand the function $f(x)$ as a power series $x + x^2 + x^3 + \dots$ and it may be adequate to just calculate the first few terms. In other words, $f(x)$ may be calculated using a low-order perturbation method. But if x is large (or infinitely large), the above power series is obviously

meaningless. However, the exact function $x/(1+x)$ is still well defined in the limit of x becoming very large.

These arguments suggest that one should sum up the diagrams (i), (ii), (iii) in fig. 1.1 and the similar ones to all orders, instead of computing them one by one. Denoting this all-order sum as $1/2 \tilde{G}_{ijij}$, where we have introduced the shorthand notation $\tilde{G}_{ijij} = \langle k_i k_j | \tilde{G} | k_i k_j \rangle_{AS}$ (and similarly for \tilde{V}), we have that

$$\begin{aligned} \frac{1}{2} \tilde{G}_{ijij} = & \frac{1}{2} \tilde{V}_{ijij} + \sum_{mn > k_F} \frac{1}{2} \tilde{V}_{ijmn} \frac{1}{\varepsilon_i + \varepsilon_j - \varepsilon_m - \varepsilon_n} \\ & \times \left[\frac{1}{2} \tilde{V}_{mni j} + \sum_{pq > k_F} \frac{1}{2} \tilde{V}_{mnpq} \frac{1}{\varepsilon_i + \varepsilon_j - \varepsilon_p - \varepsilon_q} \frac{1}{2} V_{pqij} + \dots \right]. \end{aligned} \quad (1.8)$$

The factor $1/2$ is the same as that discussed above, namely we want to count a pair of particles only once. The quantity inside the brackets is just $1/2 \tilde{G}_{mni j}$ and the above equation can be rewritten as an integral equation

$$\tilde{G}_{ijij} = \tilde{V}_{ijij} + \sum_{mn > k_F} \frac{1}{2} \tilde{V}_{ijmn} \frac{1}{\varepsilon_i + \varepsilon_j - \varepsilon_m - \varepsilon_n} \tilde{G}_{mni j}. \quad (1.9)$$

Note that \tilde{G} is the antisymmetrized G -matrix since the potential \tilde{V} is also antisymmetrized. This means that \tilde{G} obeys

$$\tilde{G}_{ijij} = -\tilde{G}_{jiij} = -\tilde{G}_{ijji}. \quad (1.10)$$

The \tilde{G} -matrix is defined as

$$\tilde{G}_{ijij} = G_{ijij} - G_{jiij}, \quad (1.11)$$

and the equation for G is

$$G_{ijij} = V_{ijij} + \sum_{mn > k_F} V_{ijmn} \frac{1}{\varepsilon_i + \varepsilon_j - \varepsilon_m - \varepsilon_n} G_{mni j}, \quad (1.12)$$

which is the familiar G -matrix equation. The above matrix is specifically designed to treat a class of diagrams contained in ΔE_0 , of which typical contributions were shown in fig. 1.1. In fact the sum of the diagrams in fig. 1.1 is equal to $1/2(G_{ijij} - G_{jiij})$.

Let us now define a more general G -matrix as

$$G_{ijij} = V_{ijij} + \sum_{mn > 0} V_{ijmn} \frac{Q(mn)}{\omega - \varepsilon_m - \varepsilon_n} G_{mni j}, \quad (1.13)$$

which is an extension of eq. (1.12). Note that eq. (1.12) has $\varepsilon_i + \varepsilon_j$ in the energy denominator, whereas in the latter equation we have a general energy variable ω in the denominator. Furthermore, in eq. (1.12) we have a restricted sum over mn , while in eq. (1.13) we sum over all mn and we have introduced a weighting factor $Q(mn)$. In eq. (1.13) $Q(mn)$ corresponds to the choice

$$Q(k_m, k_n) = \begin{cases} 1, & \min(k_m, k_n) > k_F \\ 0, & \text{else.} \end{cases} \quad (1.14)$$

where $Q(mn)$ is usually referred to as the G -matrix Pauli exclusion operator. The role of Q is to enforce a selection of the intermediate states allowed in the G -matrix equation. The above Q requires that the momenta of the intermediate particles m and n must be both above the Fermi surface defined by k_F . We may enforce a different requirement by using a summation over intermediate states different from that in eq. (1.13).

Before ending this section, let us rewrite the G -matrix equation in a more compact form. The sp energies ε and wave functions are defined by the unperturbed hamiltonian H_0 as

$$H_0 |\psi_m\rangle = \varepsilon_m |\psi_m\rangle, \quad (1.15)$$

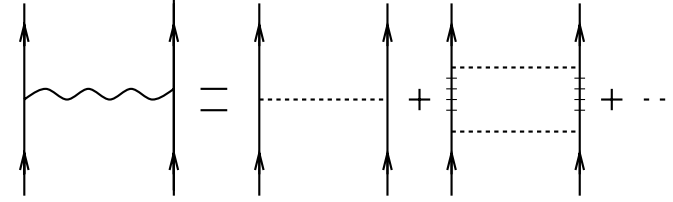


Figure 1.2: Schematic representation of the ladder renormalization of the free NN potential. The wavy line may represent either the scattering matrix T or the medium dependent reaction matrix G . The latter differs from the scattering matrix since it contains the Pauli exclusion operator Q and medium dependent single-particle energies. The physical interpretation of either T or G is that the particles must interact virtually with each other an arbitrary number of times in order to produce a finite interaction matrix element.

and similarly

$$H_0 |\psi_m \psi_n\rangle = (\varepsilon_m + \varepsilon_n) |\psi_m \psi_n\rangle. \quad (1.16)$$

The G -matrix equation can then be rewritten in the following compact form

$$G(\omega) = V + V \frac{Q}{\omega - H_0} G(\omega), \quad (1.17)$$

with $Q = \sum_{mn} |\psi_m \psi_n\rangle Q(mn) \langle \psi_m \psi_n|$. In terms of diagrams, G corresponds to an all-order sum of the "ladder-type" interactions between two particles with the intermediate states restricted by Q , as schematically shown in fig. 1.2.

The G -matrix equation has a very simple form. But its calculation is rather complicated, particularly for finite nuclear systems such as the nucleus ^{18}O . There are a number of complexities. To mention a few, the Pauli operator Q may not commute with the unperturbed hamiltonian H_0 and we have to make the replacement

$$\frac{Q}{\omega - H_0} \rightarrow Q \frac{1}{\omega - Q H_0 Q} Q.$$

The determination of the starting energy ω is also another problem.

1.2 Brief survey of various features of the nucleon-nucleon interaction

Since quantum chromodynamics (QCD) is commonly accepted as the theory of the strong interaction, the NN interaction V is completely determined by the underlying quark-quark dynamics in QCD. However,

due to the non-perturbative character of QCD at low energies, one is still far from a quantitative understanding of the NN interaction from the QCD point of view. This problem is circumvented by introducing models containing some of the properties of QCD, such as confinement and chiral symmetry breaking. One of the most used models is the so-called bag model, where a crucial question is the size of the radius (R) of the confining bag. If the size of the bag radius is chosen as in the "Little bag" ($R \leq 0.5$ fm) [7], then low-energy nuclear physics phenomena can be fairly well described in terms of hadrons like nucleons, isobars and various mesons, which are to be understood as effective descriptions of complicated multi-quark interactions. However, other models which seek to approximate QCD also indicate that an effective theory in terms of hadronic degrees of freedom may very well be the most appropriate picture for low energy nuclear physics. Although there is no unique prescription for how to construct a free NN interaction, a description of the NN interaction in terms of various meson exchanges is presently the most quantitative representation of the NN interaction [4, 8] in the energy regime of nuclear physics. We will assume that meson-exchange is an appropriate picture at low and intermediate energies. Further, we will restrict the attention to one-boson-exchange (OBE) models only.

In the next subsection we discuss the three-dimensional reduction of the Bethe-Salpeter equation and the phenomenological lagrangians which define the interactions between the various baryons and mesons. Thereafter, we briefly review the general properties of the nucleon-nucleon (NN) potential. To avoid unnecessary overlaps with already existing review articles and books on the subject [5, 4, 9, 10], only a brief exposition of the underlying theory is given. The reader is referred to the above extensive works for the more sophisticated details.

1.2.1 The one-boson-exchange interaction and the Bethe-Salpeter equation

To describe the interaction between the various baryons and mesons of table 1.1, we choose the following phenomenological lagrangians¹ for spin 1/2 baryons

$$\mathcal{L}_{ps} = g^{ps} \bar{\Psi} \gamma^5 \Psi \phi^{(ps)}, \quad (1.18)$$

$$\mathcal{L}_s = g^s \bar{\Psi} \Psi \phi^{(s)}, \quad (1.19)$$

and

$$\mathcal{L}_v = g^v \bar{\Psi} \gamma_\mu \Psi \phi_\mu^{(v)} + g^t \bar{\Psi} \sigma^{\mu\nu} \Psi \left(\partial_\mu \phi_\nu^{(v)} - \partial_\nu \phi_\mu^{(v)} \right), \quad (1.20)$$

for pseudoscalar (ps), scalar (s) and vector (v) coupling, respectively. The factors g^v and g^t are the vector and tensor coupling constants, respectively. Similarly the factor g^s is the phenomenological coupling coefficient for scalar mesons while g^{ps} is the corresponding coupling constant for pseudoscalar meson exchanges. These coupling constants may be constrained by e.g. the nucleon-nucleon scattering data. In the above equations, we have defined Ψ to be the baryon field for spin 1/2 baryons, while $\phi^{(ps)}$, $\phi^{(s)}$ and $\phi^{(v)}$ are the corresponding meson fields for pseudoscalar, scalar and vector mesons, respectively. In table 1.1² we list selected baryons with masses below 1400 MeV, though in our calculations we will only deal with those mesons which define the NN potential. Note that the above equations are for isoscalar mesons, however, for isovector mesons, the fields ϕ trivially modify to $\tau \phi$ with τ the familiar isospinor Pauli matrices. For spin 1/2 baryons, the fields Ψ are expanded in terms of the Dirac spinors (positive energy solution shown here with $\bar{u}u = 1$)

$$u(k\sigma) = \sqrt{\frac{E(k) + m}{2m}} \begin{pmatrix} \chi \\ \frac{\sigma \mathbf{k}}{E(k) + m} \chi \end{pmatrix}, \quad (1.21)$$

with χ the familiar Pauli spinor and $E(k) = \sqrt{m^2 + |\mathbf{k}|^2}$. The positive energy part of the field Ψ reads

$$\Psi(x) = \frac{1}{(2\pi)^{3/2}} \sum_{\mathbf{k}\sigma} u(k\sigma) e^{-ikx} a_{\mathbf{k}\sigma}, \quad (1.22)$$

¹We will also employ a pseudovector lagrangian for the pseudoscalar mesons. In this case we have $\mathcal{L}_{pv} = g^{pv} \bar{\Psi} \gamma_5 \gamma^\mu \Psi \partial_\mu \phi^{(ps)}$, where g^{pv} is the pseudovector coupling constant. The relativistic notations follow Itzykson and Zuber [11]. We set $c = \hbar = 1$.

²Note that we keep the name δ for the scalar isovector meson with mass 980 MeV, in order to abide to the usage in previous works [4]. The recent label is $a_0(980)$.

Table 1.1: List over selected baryons and mesons. We also include strange hadrons since the non-relativistic transition potential we will present applies to these hadrons as well.

Baryons	Mass (MeV)	Mesons	Mass (MeV)
p, n	938.926	π	138.03
Λ	1116.0	η	548.8
Σ	1197.3	σ	≈ 550.0
Δ	1232.0	ρ	770
Σ^*	1385.0	ω	782.6
		δ	983.0
		K	495.8
		K^*	895.0

with a a fermion annihilation operator. Similarly, the spin 3/2 field operator Ψ_μ are represented in terms of the Rarita-Schwinger spinors u_μ , a spin 1/2 spinor coupled to a four-vector, which in the rest-frame reads

$$u_\mu(0, j) = \sum_{\sigma_z, r} C_{\frac{1}{2} \frac{1}{2} \frac{3}{2}}^{\frac{1}{2} \frac{1}{2} \frac{3}{2}} \epsilon_\mu(r) u(0, \sigma_z), \quad j = -\frac{3}{2}, -\frac{1}{2}, \frac{1}{2}, \frac{3}{2}, \quad (1.23)$$

where C is a Clebsch-Gordan coefficient, ϵ_μ a spin-1 spinor and $u(0, \sigma_z)$ the free Dirac spinor in the rest frame with spin projections $\sigma_z = \pm \frac{1}{2}$. The $u_\mu(0, j)$ can be boosted to an arbitrary frame such that the Rarita-Schwinger field is

$$\Psi_\mu(x) = \frac{1}{(2\pi)^{3/2}} \sum_{\mathbf{k}s} u_\mu(k s) e^{-ikx} a_{\mathbf{k}s}. \quad (1.24)$$

The lagrangians which describe e.g. the interaction between a spin 1/2 and a spin 3/2 baryon are

$$\mathcal{L}_{ps} = g^{ps'} \bar{\Psi} \Psi_\mu \partial^\mu \phi^{(ps)}, \quad (1.25)$$

for the exchange of a pseudoscalar meson and

$$\mathcal{L}_v = ig^v \bar{\Psi} \gamma_5 \gamma_\mu \Psi^\nu \left(\partial_\mu \psi_\nu^{(v)} - \partial_\nu \psi_\mu^{(v)} \right) + h.c., \quad (1.26)$$

for the exchange of a vector meson.

Employing the above lagrangians, it is then possible to construct a one-boson-exchange potential model. Typically, a contribution V_s^{OBE} arising from the exchange of a scalar meson between two spin 1/2 baryons with equal masses is given by

$$\langle p'_1 p'_2 | V_s^{OBE} | p_1 p_2 \rangle = g_s^2 \frac{\bar{u}(p'_1) \bar{u}(p'_2) u(p_1) u(p_2)}{(p'_1 + p'_2 - p_1 - p_2)^2 - m_s^2}, \quad (1.27)$$

where m_s is the mass of the exchanged scalar meson³. The diagrammatic structure of this equation is shown in fig. 1.3.

In order to obtain the parameters which define an NN potential derived from OBE models, the Bethe-Salpeter equation is used as the starting point for most calculations. This equation serves to define a two-particle interaction \mathcal{T} , meant to reproduce properties like low-energy scattering data. A self-consistent comparison with the data, may in turn define the parameters of the NN interaction, such as meson cutoff parameters, or coupling constants not constrained by the data. The fully covariant Bethe-Salpeter equation reads (suppressing spin and isospin) in an arbitrary frame

$$\langle p'_1 p'_2 | \mathcal{T} | p_1 p_2 \rangle = \langle p'_1 p'_2 | \mathcal{V} | p_1 p_2 \rangle + \frac{i}{(2\pi)^4} \int d^4k \langle p'_1 p'_2 | \mathcal{V} | P + k, P - k \rangle \times S_{(1)}(P + k) S_{(2)}(P - k) \langle P + k, P - k | \mathcal{T} | p_1 p_2 \rangle, \quad (1.28)$$

³Detailed expressions for the one-boson-exchange contributions can be found in e.g. refs. [5, 4].

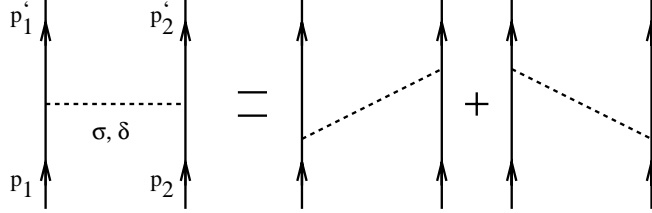


Figure 1.3: Diagrammatic representation of the exchange of a scalar boson between two equal mass and spin baryons. The leftmost picture is a shorthand for the two time-orderings to the right.

where we have defined P to be *half* the total four-momentum, i.e. $P = \frac{1}{2}(p_1 + p_2)$, and k to be the relative four-momentum. The term $S_{(i)}$ is the fermion propagator, which for e.g. positive energy spin 1/2 baryons reads

$$S_{(i)}(p) = (\not{p} - m_i + i\epsilon)^{-1}, \quad (1.29)$$

with the subscript i referring to baryon i .

In principle \mathcal{V} is supposed to represent all kinds of irreducible two-particle interactions, though it is commonly approximated by the lowest order two-particle diagram. With this prescription we obtain the familiar ladder approach to the Bethe-Salpeter equation, similar to the approach discussed in connection with the G -matrix. The schematic structure of the ladder equation, representative for both the scattering matrix and the reaction matrix G , is shown in fig. 1.2. Eq. (1.28) is a four-dimensional integral equation, which is rather tedious to solve numerically. It is therefore commonly replaced by a three-dimensional quasi-potential equation, where the time components of the four-momenta of the incoming and outgoing particles have been fixed by some adequate choice. Eq. (1.28) becomes, still in an arbitrary frame,

$$\langle \mathbf{P}'_1 \mathbf{P}'_2 | T | \mathbf{P}_1 \mathbf{P}_2 \rangle = \langle \mathbf{P}'_1 \mathbf{P}'_2 | V | \mathbf{P}_1 \mathbf{P}_2 \rangle + \frac{1}{(2\pi)^3} \int d^3k \langle \mathbf{P}'_1 \mathbf{P}'_2 | V | \mathbf{P} + \mathbf{k}, \mathbf{P} - \mathbf{k} \rangle g(\mathbf{k}, s) \langle \mathbf{P} + \mathbf{k}, \mathbf{P} - \mathbf{k} | T | \mathbf{P}_1 \mathbf{P}_2 \rangle. \quad (1.30)$$

V is now the so-called “quasi-potential”, with fixed time components of the in- and outgoing particle momenta. As such, it is no longer an independent quantity. A much favored choice for the time components of the four-momenta is to fix $p_1^0 = p_2^0 = \frac{1}{2}\sqrt{s}$, $s = (p_1 + p_2)^2$ being one of the familiar Mandelstam invariants. This prescription puts the two particles symmetrically off-shell, and is used in connection with the Blankenbecler-Sugar [12] and the Thompson [13] equations. The latter are two possible of several three-dimensional reductions of the Bethe-Salpeter equation. The quantity g is related to the baryon propagators $S_{(1)}S_{(2)}$ through

$$g(\mathbf{k}, s) = -i \int dk_0 S_{(1)}(k) S_{(2)}(-k). \quad (1.31)$$

The Blankenbecler-Sugar choice for g is (assuming $m_1 = m_2$ and spin 1/2 fermions)

$$g(\mathbf{k}, s) = \frac{m^2 \Lambda_{(1)}^+(\mathbf{k}) \Lambda_{(2)}^+(-\mathbf{k})}{E_k \frac{1}{4}s - E_k^2 + i\epsilon}, \quad (1.32)$$

and the Thompson choice

$$g(\mathbf{k}, s) = \frac{m^2 \Lambda_{(1)}^+(\mathbf{k}) \Lambda_{(2)}^+(-\mathbf{k})}{2E_k^2 \frac{1}{2}\sqrt{s} - E_k + i\epsilon}. \quad (1.33)$$

Note that the above equations are in the two-baryon center-of-mass frame with $\mathbf{P} = 0$. Here $\Lambda_i^+(\mathbf{k})$ is the projection operator for positive energy states with spin 1/2

$$\Lambda^+(\mathbf{k}) = \sum_{\sigma} |u(\mathbf{k}, \sigma)\rangle \langle u(\mathbf{k}, \sigma)| = \frac{\gamma \cdot \mathbf{k} + m}{2m}. \quad (1.34)$$

With a pseudovector coupling for the pion one obtains also a strong reduction of negative energy states compared with the pseudoscalar choice [14]. By construction, the operator g has the same imaginary cut as the propagators in the full four-dimensional Bethe-Salpeter equation and preserves therefore the unitarity relations satisfied by \mathcal{T} . The mere fact that we have fixed the time component of the four-momenta of the incoming baryons to be equally off-shell, implies that the exchanged bosons transfer three-momentum only. The meson propagators then reduce to

$$\frac{i}{-\mathbf{k}^2 - m_{\alpha}^2}, \quad (1.35)$$

with m_{α} the mass of the exchanged boson. This prescription for the meson propagators yields a so-called static interaction. The one-boson interaction then takes the form (omitting spin and isospin assignments)

$$V = \bar{u}(\mathbf{p}'_1) \bar{u}(\mathbf{p}'_2) \left(\sum_{j=1}^5 v_j(\mathbf{k}) F_j \right) u(\mathbf{p}_1) u(\mathbf{p}_2). \quad (1.36)$$

Here F_j ($j = 1, \dots, 5$) denote the Fermi invariants defined as:

$$S = 1^{(1)}1^{(2)}, \quad V = \gamma_{\mu}^{(1)} \gamma_{\mu}^{(2)}, \quad T = \frac{1}{2} \sigma_{\mu\nu}^{(1)} \sigma_{\mu\nu}^{(2)}, \quad (1.37)$$

$$A = i\gamma_5^{(1)} \gamma_{\mu}^{(1)} i\gamma_5^{(2)} \gamma_{\mu}^{(2)}, \quad P = \gamma_5^{(1)} \gamma_5^{(2)}, \quad (1.38)$$

or if we wish to have a pseudovector coupling for the pseudoscalar mesons one has to replace P with the on-shell equivalent pseudovector invariant

$$P' \propto (\gamma_{\mu}^{(1)} \cdot \partial^{\mu}) \gamma_5^{(1)} (\gamma_{\nu}^{(2)} \cdot \partial^{\nu}) \gamma_5^{(2)}, \quad (1.39)$$

with the labels (1, 2) meaning the interacting baryons 1 and 2. The v_j term refers to the appropriate meson propagator defined in eq. (1.35), including the appropriate coupling constants. In general, the coefficients $v_j(\mathbf{k})$ are functions of the three Mandelstam invariants s , t and u , though we have chosen the interaction to depend on the momentum transfer. It ought also to be stressed that neither the Blankenbecler-Sugar nor the Thompson approximations are unique approaches. However, the extensive compilations of Tjon and co-workers [14] indicate that the three-dimensional Blankenbecler-Sugar reduction of the Bethe-Salpeter equation gives only small differences compared with the results obtained by solving the full four-dimensional eq. (1.28). This conclusion applies to the Thompson choice as well [4, 6].

Expanding the free Dirac spinors in terms of $1/m$ (m is here the mass of the relevant baryon) results, to lowest order, in the familiar non-relativistic expressions for baryon-baryon potentials. Explicitly, in momentum space, the non-relativistic version of eq. (1.36) reads

$$V(k) = \sum_{j=1}^5 \tilde{v}_j(k) \tilde{F}_j. \quad (1.40)$$

Here the operators \tilde{F}_j are given by the familiar expressions

$$\tilde{F}_1 = 1, \quad \tilde{F}_2 = \frac{1}{2} i(\boldsymbol{\sigma}_1 + \boldsymbol{\sigma}_2) \cdot \mathbf{k} \times \mathbf{P}, \quad \tilde{F}_3 = \boldsymbol{\sigma}_1 \cdot \boldsymbol{\sigma}_2 k^2 - 3\boldsymbol{\sigma}_1 \cdot \mathbf{k} \boldsymbol{\sigma}_2 \cdot \mathbf{k}, \quad (1.41)$$

$$\vec{F}_4 = \sigma_1 \sigma_2 \quad \vec{F}_5 = \sigma_1 \cdot \mathbf{k} \times \mathbf{P} \sigma_2 \cdot \mathbf{k} \times \mathbf{P}, \quad (1.42)$$

In certain models also operators like \mathbf{L}^2 or an antisymmetric spin-orbit term appear [4, 15], but these can be expressed as linear combinations of the above set on the energy shell [16]. Velocity dependent terms are also omitted and σ is the spin operator for spin 1/2 particles. We end this subsection by displaying the expression for a baryon-baryon interaction in configuration space. The configuration space version of the interaction can be obtained through Hankel transformations of the above momentum-space components [16]. The following expressions listed below are for a general baryon-baryon interaction for spin 1/2 fermions where the baryons at each vertex may have different masses. Hereafter, baryon masses are always represented with a capital M , with the nucleon mass given by the average of the proton and neutron masses M_N . Following Greenberg and Lomon [15] we have the time-honored expression (omitting isospin)

$$V(\mathbf{r}) = \left\{ C_C^0 + C_C^1 + C_\sigma \sigma_1 \cdot \sigma_2 + C_T \left(1 + \frac{3}{m_\sigma r} + \frac{3}{(m_\sigma r)^2} \right) S_{12}(\hat{r}) + C_{SL} \left(\frac{1}{m_\sigma r} + \frac{1}{(m_\sigma r)^2} \right) \mathbf{L} \cdot \mathbf{S} \right\} \frac{e^{-m_\sigma r}}{m_\sigma r}, \quad (1.43)$$

where m_α is the mass of the relevant meson and S_{12} is the familiar tensor term. The coefficients are as follows. For a scalar meson exchange (σ or δ):

$$C_C^0 = -\frac{g_1 g_2}{4\pi} m_\alpha, \quad (1.44)$$

and

$$4C_C^1 = -C_{SL} = \frac{g_1 g_2}{4\pi} \frac{m_\alpha^3}{4} \left(\frac{1}{M_1^2 M_1} + \frac{1}{M_2^2 M_2} \right), \quad (1.45)$$

$$C_\sigma = C_T = 0. \quad (1.46)$$

For a pseudoscalar meson exchange (π , η , K):

$$C_\sigma = C_T = \frac{g_1 g_2}{4\pi} \frac{m_\alpha^3}{48} \left(\frac{1}{M_1^2} + \frac{1}{M_2^2} \right) \left(\frac{1}{M_1^2} + \frac{1}{M_2^2} \right), \quad (1.47)$$

$$C_C^0 = C_C^1 = C_{SL} = 0. \quad (1.48)$$

For a vector meson exchange (ρ , ω , K^*):

$$C_C^0 = \frac{m_\alpha}{4\pi} \left[G_1 G_2 - \frac{G_1 g_2^t}{2M} (M_2' + M_2) - \frac{G_2 g_1^t}{2M} (M_1' + M_1) + \frac{g_1^t g_2^t}{4M^2} (M_1' + M_1) (M_2' + M_2) \right], \quad (1.49)$$

and

$$C_C^1 = \frac{m_\alpha^3}{4\pi} \left[\frac{G_1 G_2}{16} \left\{ \left(\frac{1}{M_1} - \frac{1}{M_1'} \right) \left(\frac{1}{M_2} - \frac{1}{M_2'} \right) - \frac{1}{M_1' M_1} - \frac{1}{M_2' M_2} \right\} - \frac{G_1 g_2^t}{16M} \left\{ \frac{(M_2' + M_2)}{2} \left(\frac{1}{M_2' M_2} - \frac{1}{M_1' M_1} \right) + \left(\frac{1}{M_2} + \frac{1}{M_2'} \right) \right\} - \frac{G_2 g_1^t}{16M} \left\{ \frac{(M_1' + M_1)}{2} \left(\frac{1}{M_1' M_1} - \frac{1}{M_2' M_2} \right) + \left(\frac{1}{M_1} + \frac{1}{M_1'} \right) \right\} + \frac{g_1^t g_2^t}{32M^2} \left\{ \frac{(M_1' + M_1)}{2} (M_2' + M_2) \left(\frac{1}{M_1' M_1} + \frac{1}{M_2' M_2} \right) + (M_1' + M_1) \left(\frac{1}{M_2} + \frac{1}{M_2'} \right) + (M_2' + M_2) \left(\frac{1}{M_1} + \frac{1}{M_1'} \right) \right\} \right], \quad (1.50)$$

and

$$C_\sigma = -2C_T = \frac{m_\alpha^3}{4\pi} \frac{G_1 G_2}{24} \left(\frac{1}{M_1'} + \frac{1}{M_1} \right) \left(\frac{1}{M_2'} + \frac{1}{M_2} \right), \quad (1.51)$$

and

$$C_{SL} = -\frac{m_\alpha^3}{32\pi} \left[2G_1 G_2 \left(\frac{1}{M_1' M_1} + \frac{1}{M_2' M_2} + \frac{2}{M_2' M_1} + \frac{2}{M_1' M_2} \right) - \frac{G_1 g_2^t}{M} \left(\frac{M_2' + M_2}{M_1' M_1} + \frac{2}{M_1} + \frac{2}{M_1'} - \frac{1}{M_2} - \frac{1}{M_2'} \right) - \frac{G_2 g_1^t}{M} \left(\frac{M_1' + M_1}{M_2' M_2} + \frac{2}{M_2} + \frac{2}{M_2'} - \frac{1}{M_1} - \frac{1}{M_1'} \right) - \frac{g_1^t g_2^t}{2M^2} (M_1' + M_1) (M_2' + M_2) \left(\frac{1}{M_1' M_1} + \frac{1}{M_2' M_2} \right) \right] \quad (1.52)$$

The g_i are the standard scalar and pseudovector coupling constants of the meson-baryon-baryon vertex, the g_i^t are the vector meson exchange tensor coupling constants, and

$$G_i = g_i^v + \frac{M_i' + M_i}{2M_N} g_i^t \quad (1.53)$$

where the g_i^v are the vector meson exchange vector coupling constants.

As an example, consider the NN potential. Here we may have e.g., the exchange of π , η , ρ , ω , σ and δ mesons. Then we have $M_1' = M_1 = M_2' = M_2 = M_N$ and the coefficients for the exchange of a π meson become

$$C_\sigma = C_T = \frac{g_{NN\pi}^2}{4\pi} \frac{m_\pi^3}{12M_N^2}, \quad (1.54)$$

and

$$C_C^0 = C_C^1 = C_{SL} = 0. \quad (1.55)$$

For the exchange of a δ meson we have

$$C_C^0 = -\frac{g_{NN\delta}^2}{4\pi} m_\delta, \quad (1.56)$$

and

$$4C_C^1 = -C_{SL} = \frac{g_{NN\delta}^2}{4\pi} \frac{m_\delta^3}{2M_N}, \quad (1.57)$$

and

$$C_\sigma = C_T = 0. \quad (1.58)$$

Further, for the vector mesons one often uses the relations

$$\frac{g_{\rho NN}^t}{g_{\rho NN}^v} = 6.1, \quad (1.59)$$

and

$$\frac{g_{\omega NN}^t}{g_{\omega NN}^v} = 0.0. \quad (1.60)$$

In so doing we follow reference [4]. The fact that we choose such a strong factor for the relation between g^t and g^v for the ρ meson, leads to a tensor force for the NN interaction which is weak. Recall that the other important contribution to the tensor force stems from the exchange of the π meson. Although the coupling constants for the various mesons in different potential models vary (the variation is in general rather small), the $g_{\rho NN}^t/g_{\rho NN}^v$ relation of e.g. the Bonn potentials [4] employed here remains fixed. The strength of the tensor force is of central importance in explaining how a free NN potential is modified in the nuclear medium.

In terms of the relative and center-of-mass momenta \mathbf{k} and \mathbf{K} , the potential in momentum space is related to the nonlocal operator $V(\mathbf{r}, \mathbf{r}')$ by

$$\langle \mathbf{k}'\mathbf{K}' | V | \mathbf{k}\mathbf{K} \rangle = \int d\mathbf{r} d\mathbf{r}' e^{-i\mathbf{k}'\mathbf{r}'} V(\mathbf{r}', \mathbf{r}) e^{i\mathbf{k}\mathbf{r}} \delta(\mathbf{K}, \mathbf{K}'), \quad (1.61)$$

We will assume that the interaction is spherically symmetric and use the partial wave expansion of the plane waves in terms of spherical harmonics. This means that we can separate the radial part of the wave function from its angular dependence. The wave function of the relative motion is described in terms of plane waves as

$$e^{i\mathbf{k}\mathbf{r}} = \langle \mathbf{r} | \mathbf{k} \rangle = 4\pi \sum_{lm} i^l j_l(kr) Y_{lm}^*(\hat{\mathbf{k}}) Y_{lm}(\hat{\mathbf{r}}), \quad (1.62)$$

where j_l is a spherical Bessel function and Y_{lm} the spherical harmonic. This partial wave basis is useful for defining the operator for the nucleon-nucleon interaction, which is symmetric with respect to rotations, parity and isospin transformations. These symmetries imply that the interaction is diagonal with respect to the quantum numbers of total angular momentum J , spin S and isospin T . Using the above plane wave expansion, and coupling to final J , S and T we get

$$\langle \mathbf{k}' | V | \mathbf{k} \rangle = (4\pi)^2 \sum_{STll'mJ} i^{4+l} Y_{lm}^*(\hat{\mathbf{k}}) Y_{l'm'}(\hat{\mathbf{k}}') C_{m'M_S M}^{l'SJ} C_{m'M_S M}^{l'SJ} \langle k'l'STJM | V | k'l'STJM \rangle, \quad (1.63)$$

where we have defined

$$\langle k'l'STJM | V | k'l'STJM \rangle = \int j_{l'}(k'r') \langle l'STJM | V(r', r) | l'STJM \rangle j_l(kr) r'^2 dr' r^2 dr. \quad (1.64)$$

We have omitted the momentum of the center-of-mass motion \mathbf{K} and the corresponding orbital momentum L , since the interaction is diagonal in these variables.

In principle, all NN potentials reproduce essentially the same set of low-energy NN scattering data ($E_{lab} \leq 300$ MeV) and properties of the deuteron. These are referred to as the ‘‘on-shell’’ properties of an NN potential, since all potential models result in a roughly similar on-shell scattering matrix T . As discussed by Machleidt and Li [17], the on-shell properties of various potentials yield only rather small differences in nuclear structure observables. The crucial point is then the differing off-shell behavior of the NN potentials. Since the body of scattering data is conventionally given in terms of partial waves, the Blankenbecler-Sugar approach to the four-dimensional Bethe-Salpeter equation reads in the center-of-mass system (omitting angular momentum, isospin, spin etc. assignments)

$$T(\mathbf{k}, \mathbf{k}') = V(\mathbf{k}, \mathbf{k}') + \int_0^\infty \frac{d^3q}{(2\pi)^3} V(\mathbf{k}, \mathbf{q}) \frac{M_N^2}{E_q} \frac{\Lambda_{(1)}^+(\mathbf{q}) \Lambda_{(2)}^+(-\mathbf{q})}{\mathbf{k}^2 - \mathbf{q}^2 + i\epsilon} T(\mathbf{q}, \mathbf{k}'), \quad (1.65)$$

where $E_q = \sqrt{M_N^2 + \mathbf{q}^2}$. Since we are only interested in the matrix elements for positive-energy spinors, we obtain

$$T(\mathbf{k}, \mathbf{k}') = V(\mathbf{k}, \mathbf{k}') + \int_0^\infty \frac{d^3q}{(2\pi)^3} V(\mathbf{k}, \mathbf{q}) \frac{M_N^2}{E_q} \frac{1}{\mathbf{k}^2 - \mathbf{q}^2 + i\epsilon} T(\mathbf{q}, \mathbf{k}'). \quad (1.66)$$

If we define

$$\hat{T}(\mathbf{k}, \mathbf{k}') = \sqrt{\frac{M_N}{E_{k'}}} T(\mathbf{k}, \mathbf{k}') \sqrt{\frac{M_N}{E_k}}, \quad (1.67)$$

and

$$\hat{V}(\mathbf{k}, \mathbf{k}') = \sqrt{\frac{M_N}{E_{k'}}} V(\mathbf{k}, \mathbf{k}') \sqrt{\frac{M_N}{E_k}}, \quad (1.68)$$

we can rewrite eq. (1.66) as

$$\hat{T}(\mathbf{k}, \mathbf{k}') = \hat{V}(\mathbf{k}, \mathbf{k}') + \int_0^\infty \frac{d^3q}{(2\pi)^3} \hat{V}(\mathbf{k}, \mathbf{q}) \frac{1}{\mathbf{k}^2 - \mathbf{q}^2 + i\epsilon} \hat{T}(\mathbf{q}, \mathbf{k}'), \quad (1.69)$$

which has the same form as the non-relativistic Lippmann-Schwinger equation, and serves therefore as a starting point for non-relativistic nuclear structure calculations.

With the Thompson choice, we obtain

$$T(\mathbf{k}, \mathbf{k}') = V(\mathbf{k}, \mathbf{k}') + \int_0^\infty \frac{d^3q}{(2\pi)^3} V(\mathbf{k}, \mathbf{q}) \frac{M_N^2}{2E_q^2} \frac{1}{E_k - E_q + i\epsilon} T(\mathbf{q}, \mathbf{k}'), \quad (1.70)$$

and defining

$$\tilde{T}(\mathbf{k}, \mathbf{k}') = \frac{M_N}{E_{k'}} T(\mathbf{k}, \mathbf{k}') \frac{M_N}{E_k}, \quad (1.71)$$

and

$$\tilde{V}(\mathbf{k}, \mathbf{k}') = \frac{M_N}{E_{k'}} V(\mathbf{k}, \mathbf{k}') \frac{M_N}{E_k}, \quad (1.72)$$

we can rewrite eq. (1.70) as

$$\tilde{T}(\mathbf{k}, \mathbf{k}') = \tilde{V}(\mathbf{k}, \mathbf{k}') + \int_0^\infty \frac{d^3q}{(2\pi)^3} \tilde{V}(\mathbf{k}, \mathbf{q}) \frac{1}{E_k - E_q + i\epsilon} \tilde{T}(\mathbf{q}, \mathbf{k}'), \quad (1.73)$$

which has the form of the Lippmann-Schwinger equation, but with relativistic energies. We will use the potentials defined through the Thompson choice in our nuclear matter and neutron matter calculations, where relativistic effects become of importance at large momenta.

Let us now assign the relevant quantum numbers to the T -matrix. Recalling the structure of eq. (1.63), the general structure of the T -matrix is

$$T_{ll'}^\alpha(kk'K\omega) = V_{ll'}^\alpha(kk') + \frac{2}{\pi} \sum_{l''mM_S} \int_0^\infty d\mathbf{q} (C_{mM_S M}^{l''S\mathcal{J}})^2 \frac{Y_{l''m}^*(\hat{\mathbf{q}}) Y_{l''m}(\hat{\mathbf{q}}) V_{ll'}^\alpha(kq) T_{ll'}^\alpha(qk'K\omega)}{\omega - H_0}, \quad (1.74)$$

where we let the denominator $\omega - H_0$ be an abbreviation for one of the denominators in eqs. (1.69) and (1.73). Further, the shorthand notation

$$T_{ll'}^\alpha(kk'K\omega) = \langle kKlL\mathcal{J}ST | T(\omega) | k'Kl'L\mathcal{J}ST \rangle, \quad (1.75)$$

is introduced to denote the T -matrix with momenta k and k' and orbital momenta l and l' of the relative motion, and K is the corresponding momentum of the center-of-mass motion. Further, L , \mathcal{J} , S and T are the orbital momentum of the center-of-mass motion, the total angular momentum, spin and isospin, respectively. They are all represented by the label α . This notation applies to the potential V and the G -matrix as well.

Eq. (1.74) can be further simplified by using the orthogonality properties of the Clebsch-Gordan coefficients and the spherical harmonics. Eq. (1.74) then reduces to the well-known one-dimensional angle independent integral equation

$$T_{ll'}^\alpha(kk'K\omega) = V_{ll'}^\alpha(kk') + \frac{2}{\pi} \sum_{l''} \int_0^\infty dq q^2 \frac{V_{ll''}^\alpha(kq) T_{ll''}^\alpha(qk'K\omega)}{\omega - H_0}. \quad (1.76)$$

Inserting the denominators for the Blankenbecler-Sugar and Thompson reductions of the full Bethe-Salpeter equation, eqs. (1.69) and (1.73) can be rewritten as

$$\hat{T}_{ll'}^\alpha(kk'K) = \hat{V}_{ll'}^\alpha(kk') + \frac{2}{\pi} \sum_{l''} \int_0^\infty dq q^2 \hat{V}_{ll''}^\alpha(kq) \frac{1}{k^2 - q^2 + i\epsilon} \hat{T}_{ll''}^\alpha(qk'K), \quad (1.77)$$

for eq. (1.69) and

$$\tilde{T}_{ll'}^\alpha(kk'K) = \tilde{V}_{ll'}^\alpha(kk') + \frac{2}{\pi} \sum_{l''} \int_0^\infty dq q^2 \tilde{V}_{ll''}^\alpha(kq) \frac{1}{E_k - E_q + i\epsilon} \tilde{T}_{ll''}^\alpha(qk'K) \quad (1.78)$$

for eq. (1.73).

The reader should also observe that these equations are similar to that for the G -matrix, the only difference being the omission of the Pauli operator Q and the medium dependence of the sp energies. For intermediate states in the two latter equations with $k \neq q$, energy is not conserved and the nucleons are off their energy shell. It is this aspect of various potentials which becomes of importance in a nuclear medium. Consider the contribution to the above scattering matrices for the 3S_1 partial wave, exhibited in fig. 1.4 to second order in the ladder approximation. The second-order terms contain contributions from intermediate states in the 3D_1 partial wave which connects only through the tensor force of the NN interaction. The 3S_1 - 3D_1 matrix elements account for the largest second-order contribution. Since various potentials yield essentially the same on-shell T -matrix, contributions to the 3S_1 partial wave may be obtained from rather different first and second- or higher-order terms to the T -matrix. As an example, the tensor force contribution from the 3D_1 intermediate state shown in fig. 1.4 may be rather weak as compared to the first contribution arising from the central part of the potential. The converse may also be the case, but the outcome may still yield the same on-shell T -matrix. In other words, the bulk of the T -matrix can be expressed as

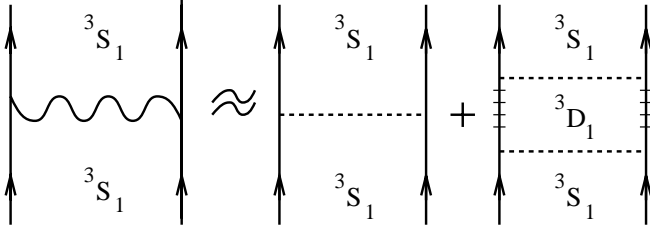


Figure 1.4: Contributions through second order in the interaction V to the T -matrix for the 3S_1 partial wave.

$$T \approx V_C + V_T \frac{1}{\omega - H_0} V_T, \quad (1.79)$$

where V_C is the central part of the NN interaction while V_T is the tensor force. Moreover, ω is the energy of the incoming nucleons and H_0 is the unperturbed hamiltonian representing the intermediate scattering states. Thus, if the tensor force is weak (strong), a stronger (weaker) central force is needed to arrive at the same on-shell T -matrix. A similar mechanism is present when we evaluate the G -matrix for either nuclear matter or finite nuclei, though, in these cases we must also account for medium effects such as the modification of the energy denominator and the inclusion of the Pauli principle.

1.3 Numerical solution of the scattering equations

We are now going to solve the non-relativistic Lippman-Schwinger equation for the neutron-proton system in momentum space for positive energies E in order to obtain the phase shifts. This corresponds to solving the Schrödinger equation in momentum space.

Following the discussion above, we need first then transform the nucleon-nucleon interaction V , which

we assume depends only on the interparticle distance r , in momentum space using the Fourier-Bessel transform (Hankel transform)

$$V_i(k, k') = \int j_l(kr)V(r)j_l(k'r)r^2 dr, \quad (1.80)$$

where j_l is the spherical Bessel function. We will just study the case $l = 0$, which means that $j_0(kr) = \sin(kr)/kr$.

For scattering states, the energy is positive, $E > 0$. The Lippman-Schwinger equation, which is the non-relativistic version of the Bethe-Salpeter equation discussed above, is an integral equation where we have to deal with the amplitude $R(k, k')$ (reaction matrix, which is the real part of the full complex T -matrix discussed in the previous section) defined through the integral equation

$$R_i(k, k') = V_i(k, k') + \frac{2}{\pi} \mathcal{P} \int_0^\infty dq q^2 V_i(k, q) \frac{1}{E - q^2/m} R_i(q, k'), \quad (1.81)$$

where the total kinetic energy of the two incoming particles in the center-of-mass system is

$$E = \frac{k_0^2}{m}. \quad (1.82)$$

The symbol \mathcal{P} indicates that Cauchy's principal-value prescription is used in order to avoid the singularity arising from the zero of the denominator. We will discuss below how to solve this problem. Eq. (1.81) represents then the problem you will have to solve numerically.

The matrix $R_i(k, k')$ relates to the phase shifts through its diagonal elements as

$$R_i(k_0, k_0) = -\frac{\tan \delta_i}{mk_0}. \quad (1.83)$$

From now on we will drop the subscript l in all equations.

In order to solve the Lippman-Schwinger equation in momentum space, we need first to write a function which sets up the mesh points. We need to do that since we are going to approximate an integral through

$$\int_a^b f(x) dx \approx \sum_{i=1}^N w_i f(x_i), \quad (1.84)$$

where we have fixed N lattice points through the corresponding weights w_i and points x_i . Start writing your main program by setting up the mesh points and the corresponding weights. Fix first the number of mesh points N . Use the function GAULEG in the program example below to set up the weights ω_i and the points k_i . Before you go on you need to observe that GAULEG uses Legendre polynomials to fix the mesh points and weights. This means that the integral is for the interval $[-1, 1]$. Your integral is for the interval $[0, \infty]$. You will need to map the weights from GAULEG to your interval. To do this, call first $GAULEG(a, b, x, w, N)$, with $a = -1$, $b = 1$. It returns the mesh points x_i and weights w_i . You map these points over to the limits in your integral. You can then use the following mapping

$$k_i = \text{const} \times \tan \left\{ \frac{\pi}{4} (1 + x_i) \right\}, \quad (1.85)$$

and

$$w_i = \text{const} \frac{\pi}{4} \frac{w_i}{\cos^2 \left(\frac{\pi}{4} (1 + x_i) \right)}. \quad (1.86)$$

If you choose units fm^{-1} for k , set $\text{const} = 1$. If you choose to work with MeV, set $\text{const} \sim 200$ ($\hbar c = 197$ MeVfm).

The next step is to write a function which calculates the potential in momentum space. The potential we choose to employ here is a parametrized potential between a proton and neutron for the partial wave 1S_0 , i.e., spin $S = 0$ and orbital momentum $l = 0$, a singlet S-state. This state does not have a bound state for the deuteron (only the triplet S-state has). The parametrized version of this potential fits the experimental phase-shifts. It is given by

$$V(r) = V_a \frac{e^{-ax}}{x} + V_b \frac{e^{-bx}}{x} + V_c \frac{e^{-cx}}{x} \quad (1.87)$$

with $x = \mu r$, $\mu = 0.7 \text{ fm}^{-1}$ (the inverse of the pion mass), $V_a = -10.463 \text{ MeV}$ and $a = 1$, $V_b = -1650.6 \text{ MeV}$ and $b = 4$ and $V_c = 6484.3 \text{ MeV}$ and $c = 7$. As an exercise, find the potential in momentum space using Eq. (1.80) with $j_0(kr) = \sin(kr)/kr$. The transform of a potential on the form $V(r) = V_0 \exp(-\mu r)/r$ is

$$V(k', k) = \frac{V_0}{4k'k} \ln \left(\frac{(k' + k)^2 + \mu^2}{(k' - k)^2 + \mu^2} \right). \quad (1.88)$$

The principal value in Eq. (1.81) is rather tricky to evaluate numerically, mainly since computers have limited precision. We will here use a subtraction trick often used when dealing with singular integrals in numerical calculations. We introduce first the calculus relation

$$\int_{-\infty}^{\infty} \frac{dk}{k - k_0} = 0. \quad (1.89)$$

It means that the curve $1/(k - k_0)$ has equal and opposite areas on both sides of the singular point k_0 . If we break the integral into one over positive k and one over negative k , a change of variable $k \rightarrow -k$ allows us to rewrite the last equation as

$$\int_0^{\infty} \frac{dk}{k^2 - k_0^2} = 0. \quad (1.90)$$

We can use this to express a principal values integral as

$$\mathcal{P} \int_0^{\infty} \frac{f(k)dk}{k^2 - k_0^2} = \int_0^{\infty} \frac{(f(k) - f(k_0))dk}{k^2 - k_0^2}, \quad (1.91)$$

where the right-hand side is no longer singular at $k = k_0$, it is proportional to the derivative df/dk , and can be evaluated numerically as any other integral.

We can then use the trick in Eq. (1.91) to rewrite Eq. (1.81) as

$$R(k, k') = V(k, k') + \frac{2}{\pi} \int_0^{\infty} dq \frac{q^2 V(k, q) R(q, k') - k_0^2 V(k, k_0) R(k_0, k')}{(k_0^2 - q^2)/m}. \quad (1.92)$$

This is the equation you are going to solve numerically in order to calculate the phase shifts of Eq. (1.83). We are interested in obtaining $R(k_0, k_0)$.

How do we proceed in order to solve Eq. (1.92)?

- Using the mesh points k_j and the weights ω_j , we can rewrite Eq. (1.92) as

$$R(k, k') = V(k, k') + \frac{2}{\pi} \sum_{j=1}^N \frac{\omega_j k_j^2 V(k, k_j) R(k_j, k')}{(k_0^2 - k_j^2)/m} - \frac{2}{\pi} k_0^2 V(k, k_0) R(k_0, k') \sum_{n=1}^N \frac{\omega_n}{(k_0^2 - k_n^2)/m}. \quad (1.93)$$

This equation contains now the unknowns $R(k_i, k_j)$ (with dimension $N \times N$) and $R(k_0, k_0)$.

- We can turn Eq. (1.93) into an equation with dimension $(N + 1) \times (N + 1)$ with a mesh which contains the original mesh points k_j for $j = 1, N$ and the point which corresponds to the energy k_0 . Consider the latter as the 'observable' point. The mesh points become then k_j for $j = 1, n$ and $k_{N+1} = k_0$.

- With these new mesh points we define the matrix

$$A_{i,j} = \delta_{i,j} - V(k_i, k_j) u_j, \quad (1.94)$$

where δ is the Kronecker δ and

$$u_j = \frac{2}{\pi} \frac{\omega_j k_j^2}{(k_0^2 - k_j^2)/m} \quad j = 1, N \quad (1.95)$$

and

$$u_{N+1} = -\frac{2}{\pi} \sum_{j=1}^N \frac{k_0^2 \omega_j}{(k_0^2 - k_j^2)/m}. \quad (1.96)$$

The first task is then to set up the matrix A for a given k_0 . This is an $(N + 1) \times (N + 1)$ matrix. It can be convenient to have an outer loop which runs over the chosen observable values for the energy k_0^2/m . Note that all mesh points k_j for $j = 1, N$ must be different from k_0 . Note also that $V(k_i, k_j)$ is an $(N + 1) \times (N + 1)$ matrix. Write a small function which sets up A .

- With the matrix A we can rewrite Eq. (1.93) as a matrix problem of dimension $(N + 1) \times (N + 1)$. All matrices R , A and V have this dimension and we get

$$A_{i,l} R_{l,j} = V_{i,j}, \quad (1.97)$$

or just

$$AR = V. \quad (1.98)$$

- Since you already have defined A and V (these are stored as $(N + 1) \times (N + 1)$ matrices) Eq. (1.98) involves only the unknown R . We obtain it by matrix inversion, i.e.,

$$R = A^{-1}V. \quad (1.99)$$

Thus, to obtain R , you will need to set up the matrices A and V and invert the matrix A . To do that you must call the function *matinv* in the program below. With the inverse A^{-1} , performing a matrix multiplication with V results in R .

With R you can then evaluate the phase shifts by noting that

$$R(k_{N+1}, k_{N+1}) = R(k_0, k_0) = -\frac{\tan \delta}{mk_0}, \quad (1.100)$$

where δ are the phase shifts.

1.3.1 Program example

```
! here we constants like the mass of the nucleon and hbar c
MODULE constants
  DOUBLE PRECISION , PUBLIC :: p_mass, hbarc
  PARAMETER (p_mass = 938.926D0, hbarc = 197.327D0)
END MODULE constants

! definition of mesh points -k- and weights -weight-
! n_rel is the number of mesh points. It could also be read
! in as a variable
! Units are MeV

MODULE mesh_variables
  INTEGER, PUBLIC :: n_rel
  PARAMETER(n_rel=40)
  DOUBLE PRECISION, ALLOCATABLE, PUBLIC :: k(:), weight(:)
END MODULE mesh_variables
```

```

! Main program
! labelling of variables follows the discussion in the text above
PROGRAM phase
  USE mesh_variables
  USE constants
  IMPLICIT NONE
  INTEGER i,j, eloop, info
  DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:,:) :: vkk, r, a, identity
  DOUBLE PRECISION, ALLOCATABLE,DIMENSION(:) :: umatrix, k
  DOUBLE PRECISION :: pih, k0, delta

  pih=2.D0/ACOS(-1.D0)

! dynamic allocation of arrays

  ALLOCATE(identity (n_rel+1,n_rel+1))
  ALLOCATE( k(n_rel), wra (n_rel))

! set up mesh points and weights

  CALL rel_mesh

! identity matrix

  identity=0.
  DO i=1,n_rel+1
    identity(i,i)=1.
  ENDDO

! loop over incoming relative momenta, units MeV

  DO eloop=1,1000, 25
    ALLOCATE( k (n_rel+1), umatrix(n_rel+1))
    k=0.
    k0=eloop; k(n_rel+1)=k0; k(1:n_rel)=k(1:n_rel)
    ALLOCATE(vkk (n_rel+1,n_rel+1), r(n_rel+1,n_rel+1), a(n_rel+1,n_rel+1))
    vkk=0.; r=0.; a=0.

! set up the potential in k-space, units MeV-2, Eq. (1.87)

    CALL v_pot_yukawa(vkk,k)

! set up the array u, Eq. (1.95)

    umatrix=0.
    DO i=1,n_rel
      umatrix(i)=p_mass*pih*weight(i)*(k(i)**2)/(k(i)**2-k0*k0)
      umatrix(n_rel+1)=umatrix(n_rel+1)- &
        p_mass*pih*k0*k0*weight(i)/(k(i)**2-k0*k0)
    ENDDO

! set up the matrix A to be inverted, Eq. (1.94)

    a=0.

```

```

  DO i=1,n_rel+1
    DO j=1,n_rel+1
      a(i,j)=vkk(i,j)*umatrix(j)
    ENDDO
  ENDDO
  a=identity+a
  CALL matinv(a,n_rel+1)

! then evaluate the reaction matrix R, Eq. (1.99)

  r=MATMUL(a,vkk)

! calculate phase shifts, Eq. (1.100)

  delta=ATAN(-k0*p_mass*r(n_rel+1,n_rel+1))

! write out momentum in fm-1, kinetic energy in MeV and phase shifts

  write(6,'(E12.6,2X,E12.6,2X,E12.6)') k0/hbarc, 2*k0*k0/p_mass, &
    delta*180./acos(-1.d0)

  DEALLOCATE(vkk , r, a)
  DEALLOCATE( k, umatrix)
ENDDO
DEALLOCATE(wra , ra) ; DEALLOCATE (identity)

END PROGRAM phase

! set up mesh points and weights according to Eqs. (1.85) and (1.86)

SUBROUTINE rel_mesh
  USE mesh_variables
  IMPLICIT NONE
  INTEGER i
  DOUBLE PRECISION pih,u,s,xx,c
  PARAMETER (c=200.)
  DIMENSION u(n_rel), s(n_rel)

  pih=ACOS(-1.D0)/4.D0
  CALL gauleg (-1.D0,1.D0,u,s,n_rel)
  DO i=1,n_rel
    xx=pih*(u(i)+1.D0)
    k(i)=DTAN(xx)*c
    weight(i)=pih*c/DCOS(xx)**2*s(i)
  ENDDO

END SUBROUTINE rel_mesh

! set up potential in k-space, units of MeV-2, see Eq. (1.87)

SUBROUTINE v_pot_yukawa(vkk,k)
  USE mesh_variables

```

```

USE constants
IMPLICIT NONE
INTEGER i,j
DOUBLE PRECISION mu1, mu2, mu3, v_1, v_2, v_3, a, b, fac
DOUBLE PRECISION, DIMENSION(n_rel+1), INTENT(IN) :: k
DOUBLE PRECISION, DIMENSION(n_rel+1,n_rel+1), INTENT(INOUT) :: vkk
PARAMETER(mu1=0.49d0*hbarc*hbarc,v_1=-3.7368/hbarc)
PARAMETER(mu2=7.84d0*hbarc*hbarc,v_2=-589.5/hbarc)
PARAMETER(mu3=24.01d0*hbarc*hbarc,v_3=2315.8214/hbarc)

DO i=1,n_rel+1      ! set up the free potential
  DO j=1,i
    a=(k(j)+k(i))*2
    b=(k(j)-k(i))*2
    fac=1./(k(i)*k(j))
    vkk(j,i)=v_1*fac*DLOG((a+mu1)/(b+mu1))+ &
              v_2*fac*DLOG((a+mu2)/(b+mu2))+ &
              v_3*fac*DLOG((a+mu3)/(b+mu3))
    vkk(i,j)=vkk(j,i)
  ENDDO
ENDDO

END SUBROUTINE v_pot_yukawa

!
!   Routines to do mtx inversion, from Numerical
!   Recipes, Teukolsky et al. Routines included
!   below are matinv, lu_decompose
!   and lu_linear_equation. See chap 2
!   of Numerical Recipes for further details
!   Included is also the routine GAULEG for setting
!   up mesh points according to the gauss-legendre
!   method. All recoded by M. Hjorth-Jensen in F90,
!   morten.hjorth-jensen@fys.uio.no. Last upgrade
!   oct 1996.
!

SUBROUTINE matinv(a,n)
IMPLICIT NONE
INTEGER, INTENT(IN) :: n
INTEGER :: i, j
DOUBLE PRECISION, DIMENSION(n,n), INTENT(INOUT) :: a
DOUBLE PRECISION, ALLOCATABLE :: y(:, :)
DOUBLE PRECISION :: d
INTEGER, ALLOCATABLE :: indx(:)

ALLOCATE (y( n, n)) ; ALLOCATE ( indx (n))
y=0.
! setup identity matrix
DO i=1,n
  y(i,i)=1.
ENDDO

```

```

!   LU decompose the matrix just once
CALL lu_decompose(a,n,indx,d)

!   Find inverse by columns
DO j=1,n
  CALL lu_linear_equation(a,n,indx,y(:,j))
ENDDO
!   The original matrix a was destroyed, now we equate it with the inverse y
a=y

DEALLOCATE ( y ); DEALLOCATE ( indx )

END SUBROUTINE matinv

!   Given an NxN matrix A(N,N), this routine replaces it by the LU
!   decomposed one, where the matrix elements are stored in the same
!   matrix A. The array indx is an output vector which records the row
!   permutation effected by the partial pivoting. d is the determinant
!

SUBROUTINE lu_decompose(a,n,indx,d)
IMPLICIT NONE
INTEGER :: n, i, j, k, imax
DOUBLE PRECISION :: sum, tiny, aamax, dum, d
DOUBLE PRECISION, DIMENSION(n,n) :: a
INTEGER, DIMENSION(n) :: indx
DOUBLE PRECISION, ALLOCATABLE :: vv(:)

tiny=1.0e-20
ALLOCATE ( vv(n) )
D=1.
DO i=1,n
  aamax=0.
  DO j=1,n
    IF (ABS(a(i,j)) > aamax) aamax=ABS(a(i,j))
  ENDDO
!   Zero is the largest element
  IF (aamax == 0.) STOP 'Singular matrix.'
!   No nonzero largest element
  vv(i)=1./aamax
ENDDO
!   loop over columns
DO j=1,n
!   solves equation 2.3.12 except for i=j of Numerical Recipes
  IF (j > 1) THEN
    DO i=1,j-1
      sum=a(i,j)
      IF (i > 1) THEN
        DO k=1,i-1
          sum=sum-a(i,k)*a(k,j)
        ENDDO
      a(i,j)=sum
    ENDDIF
  ENDDO
ENDDIF

```

```

! start searching for largest pivot element
aamax=0.
DO i=j,n
  sum=a(i,j)
  IF (j > 1)THEN
    DO k=1,j-1
      sum=sum-a(i,k)*a(k,j)
    ENDDO
    a(i,j)=sum
  ENDF
  dum=vv(i)*ABS(sum)
  IF (dum >= aamax) THEN
    imax=i
    aamax=dum
  ENDF
ENDDO
! interchange of rows
IF (j /= imax)THEN
  DO k=1,n
    dum=a(imax,k)
    a(imax,k)=a(j,k)
    a(j,k)=dum
  ENDDO
! change of parity for determinant
d=-d
vv(imax)=vv(j)
ENDIF
indx(j)=imax
IF(j /= n) THEN
  IF(a(j,j) == 0.) a(j,j)=tiny
  dum=1./a(j,j)
  DO i=j+1,n
    a(i,j)=a(i,j)*dum
  ENDDO
ENDIF
ENDDO
IF(a(n,n) == 0.) a(n,n)=tiny
DEALLOCATE ( vv)

END SUBROUTINE lu_decompose

! Solves set of linear equations Ax=b, A is input as an LU decomposed
! matrix and indx keeps track of the permutations of the rows. b is input
! as the right-hand side vector b and returns the solution x. A, n and indx
! are not modified by this routine. This function takes into that b can contain
! many zeros and is therefore suitable for matrix inversion

SUBROUTINE lu_linear_equation(a,n,indx,b)
IMPLICIT NONE
INTEGER :: n, ii, ll, i, j
DOUBLE PRECISION :: sum
DOUBLE PRECISION, DIMENSION(n,n) :: a
DOUBLE PRECISION, DIMENSION(n) :: b

```

```

INTEGER, DIMENSION(n) :: indx

ii=0
! First we solve equation 2.3.6 of numerical recipes
DO i=1,n
  ll=indx(i)
  sum=b(ll)
  b(ll)=b(i)
  IF (ii /= 0)THEN
    DO j=ii,i-1
      sum=sum-a(i,j)*b(j)
    ENDDO
  ELSEIF (sum /= 0.) THEN
    ii=i
  ENDF
  b(i)=sum
ENDDO
! then we solve equation 2.3.7
DO i=n,1,-1
  sum=b(i)
  IF (i < n) THEN
    DO j=i+1,n
      sum=sum-a(i,j)*b(j)
    ENDDO
  ENDF
! store a component of the solution x in the same place as b
  b(i)=sum/a(i,i)
ENDDO

END SUBROUTINE lu_linear_equation

SUBROUTINE gauleg(x1,x2,x,w,n)
IMPLICIT NONE
INTEGER :: i, j, m, n
DOUBLE PRECISION :: eps, x1, x2, x, w
DIMENSION :: x(n), w(n)
PARAMETER (eps=3.D-14)
DOUBLE PRECISION :: p1,p2,p3,pp,xl,xm,z,z1

m=(n+1)/2
xm=0.5d0*(x2+x1)
x1=0.5d0*(x2-x1)
DO i=1,m
  z1=0.
  z=COS(3.141592654d0*(i-.25d0)/(n+.5d0))
  DO WHILE ( ABS(z-z1) > EPS)
    p1=1.
    p2=0.
    DO j=1,n
      p3=p2
      p2=p1
      p1=((2.*j-1.)*z*p2-(j-1.)*p3)/j
    ENDDO
  ENDDO

```

```

ENDDO
pp=n*(z*p1-p2)/(z*z-1.)
z1=z
z=z-p1/pp
ENDDO
x(i)=xm-x1*z
x(n+1-i)=xm+x1*z
w(i)=2.*x1/((1.-z*z)*pp*pp)
w(n+1-i)=w(i)
ENDDO

END SUBROUTINE gauleg

```

Background literature

For a background on scattering theory, consult E. Merzbacher, Quantum Mechanics, (3rd edition), sections 13.1-13.6 and 20.1-20.4.

1.4 How to include in-medium correlations

In nuclear structure and nuclear matter calculations one has to face the problem that any realistic nucleon-nucleon (NN) potential V exhibits a strong short-range repulsion, which in turn makes a perturbative treatment of the nuclear many-body problem prohibitive. If the potential has a so-called hard core, the matrix elements of such a potential $\langle \psi | V | \psi \rangle$ evaluated for an uncorrelated two-body wave function $\psi(r)$ diverge, since the uncorrelated wave function is different from zero also for relative distances r smaller than the hard-core radius. Similarly, even if one uses potentials with softer cores, the matrix elements of the potential become very large at short distances. The above problem was however overcome by introducing the reaction matrix G (displayed by the summation of ladder type of diagrams in fig. 1.2) which accounts for the effects of two-nucleon correlations.

The matrix elements of the potential V then become

$$\langle \psi | G | \psi \rangle = \langle \psi | V | \Psi \rangle \quad (1.101)$$

where Ψ is now the correlated wave function. By accounting for the correlations in the two-body wave function Ψ , the matrix elements of the potential become finite, even for a hard-core potential V . Moreover, as will be discussed below, compared with the uncorrelated wave function, the correlated wave function enhances the matrix elements of V at distances for which the interaction is attractive. The G -matrix applies to bound states, while for scattering of free particles, the G -matrix is replaced by the T -matrix. The difference between the matrices resides in the medium dependence provided by the Pauli operator and the energy ω of the interacting particles. We have defined the G -matrix by either

$$G(\omega) = V + V \frac{Q}{\omega - H_0} G(\omega), \quad (1.102)$$

or

$$G(\omega) = V + V Q \frac{1}{\omega - Q H_0 Q} Q G(\omega). \quad (1.103)$$

The former equation applies if the Pauli operator Q commutes with the unperturbed hamiltonian H_0 , whereas the latter is needed if $[H_0, Q] \neq 0$. Similarly, the correlated wave function Ψ is given as

$$|\Psi\rangle = |\psi\rangle + \frac{Q}{\omega - H_0} G |\psi\rangle, \quad (1.104)$$

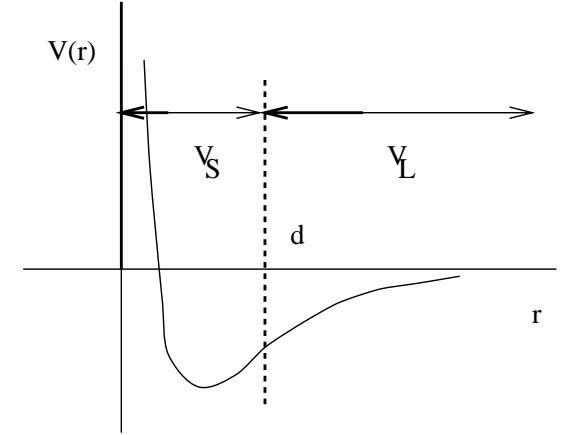


Figure 1.5: Schematic plot of the nucleon-nucleon potential for the 1S_0 partial wave. See text for further details.

or

$$|\Psi\rangle = |\psi\rangle + Q \frac{1}{\omega - Q H_0 Q} Q G |\psi\rangle. \quad (1.105)$$

These equations look rather simple, but their actual calculation is in fact rather complicated and difficult. Approximate methods were devised in the "early days", amongst these methods were the famous separation and reference spectrum methods. These methods present a physically intuitive picture of the G -matrix, and it is one of the scopes of this section to briefly review those methods here. Further, we will only discuss how to calculate the G -matrix for infinite matter.

1.4.1 The separation and reference-spectrum methods for calculating the G -matrix

Fig. 1.5 displays a schematic plot of the nucleon-nucleon potential for the 1S_0 partial wave. The internucleon distance is r . As indicated in the figure, we divide the potential into two parts, the short-range part V_S and the long range part V_L . Formally, this may written as

$$V(r) \approx V_S(r)\Theta(d-r) + V_L(r)\Theta(r-d), \quad (1.106)$$

where Θ is the familiar step function, and d is known as the separation distance. The separation distance is chosen so that the attractive part of V_S balances that of the repulsive core. A remarkable result is then to a good approximation

$$G \propto V_L(r), \quad (1.107)$$

based on the separation method. This is quite an appealing result. The whole potential $V(r)$ is "singular" in the sense that it has a very strong repulsive core. In contrast, $V_L(r)$ is obviously no longer singular and hence the G -matrix is also no longer singular. We have not yet shown how to obtain this result, nor have we discussed what d is and how to determine d^4 .

We will now derive the above result and also try to motivate for the physics behind the separation and reference-spectrum methods. First we define a useful identity following Bethe, Brandow and Petschek [3].

⁴ d is typically about 1 fm.

Suppose we have two different G -matrices, defined by

$$G_1 = V_1 + V_1 \frac{Q_1}{e_1} G_1, \quad (1.108)$$

and

$$G_2 = V_2 + V_2 \frac{Q_2}{e_2} G_2, \quad (1.109)$$

where Q_1/e_1 and Q_2/e_2 are the propagators of either eq. (1.102) or eq. (1.103). G_1 and G_2 are two different G -matrices having two different interactions and/or different propagators. We aim at an identity which will enable us to calculate G_1 in terms of G_2 , or vice versa. Defining the wave operators

$$\Omega_1 = 1 + \frac{Q_1}{e_1} G_1, \quad (1.110)$$

and

$$\Omega_2 = 1 + \frac{Q_2}{e_2} G_2, \quad (1.111)$$

we can rewrite the above G -matrices as

$$G_1 = V_1 \Omega_1, \quad (1.112)$$

and

$$G_2 = V_2 \Omega_2. \quad (1.113)$$

Using these relations, we rewrite G_1 as

$$\begin{aligned} G_1 &= G_1 - G_2^\dagger \left(\Omega_1 - 1 - \frac{Q_1}{e_1} G_1 \right) + \left(\Omega_2^\dagger - 1 - G_2^\dagger \frac{Q_2}{e_2} \right) G_1 \\ &= G_2^\dagger + G_2^\dagger \left(\frac{Q_1}{e_1} - \frac{Q_2}{e_2} \right) G_1 + \Omega_2^\dagger G_1 - G_2^\dagger \Omega_1, \end{aligned} \quad (1.114)$$

and using eqs. (1.112) and (1.113) we obtain the identity

$$G_1 = G_2^\dagger + G_2^\dagger \left(\frac{Q_1}{e_1} - \frac{Q_2}{e_2} \right) G_1 + \Omega_2^\dagger (V_1 - V_2) \Omega_1. \quad (1.115)$$

The second term on the rhs. is called the propagator-correction term; it vanishes if G_1 and G_2 have the same propagators. The third term is often referred to as the potential-correction term, and it disappears if G_1 and G_2 have the same potentials.

The reader may now ask what is the advantage of the above identity. If we assume that by some physical reasoning we are able to calculate G_2 (we will see examples of this below both for nuclear matter and for finite nuclei), and that the expression for G_2 can be calculated easily, and further that G_2 is a good approximation to the original G -matrix, then we can use the above identity to perform a perturbative calculation of G_1 in terms of G_2 . Below we will use eq. (1.115) to discuss the separation and the reference-spectrum methods.

The separation method

This method was first introduced by Moszkowski and Scott for evaluating the G -matrix in nuclear matter [18, 19]. In eq. (1.106) we separated the NN potential into two parts, V_S and V_L . Let us define the G -matrices

$$G_1 = G = V + V \frac{Q}{e} G, \quad (1.116)$$

and

$$G_2 = G_S = V_S + V_S \frac{Q_2}{e_2} G_S, \quad (1.117)$$

where V is the NN potential. Eq. (1.115) gives

$$G = G_S^\dagger + G_S^\dagger \left(\frac{Q}{e} - \frac{Q_2}{e_2} \right) G + \Omega_S^\dagger V_L \Omega. \quad (1.118)$$

The idea is that we can make $G_S \approx 0$, which in turn yields the advertised relation

$$G \approx V_L.$$

We emphasize that the above result is an approximation, but the hope is that this approximation is a physically reasonable one. To proceed, let us define the correlated wave function Ψ_a

$$G |\psi_a\rangle = V |\Psi_a\rangle, \quad (1.119)$$

where ψ_a is the unperturbed wave function. Using the definition of the correlated wave function in eq. (1.104) we have

$$|\Psi_a\rangle = |\psi_a\rangle + \frac{Q}{\omega - H_0} G |\psi_a\rangle = |\psi_a\rangle + \frac{Q}{\omega - H_0} V |\Psi_a\rangle. \quad (1.120)$$

Note that we have assumed that the Pauli operator Q and the unperturbed hamiltonian H_0 commute. If we are able to obtain the correlated wave function, we get the G -matrix by

$$\langle \psi_a | G | \psi_b \rangle = \langle \psi_a | V | \Psi_b \rangle. \quad (1.121)$$

We will try to obtain G_S through this scheme. G_S was defined according to eq. (1.117) with the propagator Q_2/e_2 . We have not yet specified how to obtain Q_2 and e_2 , though the identity in eq. (1.118) is valid for any choice of Q_2 and e_2 . The potential which defines G_S is V_S , which is a short-range repulsive potential. Using the uncertainty principle, the intermediate states included by V_S should be predominantly those at high momenta. For such scattering states there is hardly any Pauli blocking, which means that the probability of finding other nucleons at such momenta is practically equal to zero. This physically intuitive reasoning leads us to choose $Q_2 = 1$. With this choice we obtain

$$G_S = V_S + V_S \frac{1}{\omega - H_0} G_S, \quad (1.122)$$

and the corresponding correlated wave function

$$|\Psi_a^S\rangle = |\psi_a\rangle + \frac{1}{\omega - H_0} V_S |\Psi_a^S\rangle. \quad (1.123)$$

As mentioned earlier we are free to make whatever choice for e_2 in the operator $\omega - H_0$ of G_S . Let us employ a specific on-shell choice, i.e. $\omega = \varepsilon_a$, and $H_0 |\psi_a\rangle = \varepsilon_a |\psi_a\rangle$. In this case we have $(\omega - H_0) |\psi_a\rangle = 0$. Using the above definition for the correlated wave function we obtain for the diagonal matrix element

$$\langle \psi_a | G_S | \psi_a \rangle = \langle \psi_a | V_S | \Psi_a^S \rangle = \langle \psi_a | (\varepsilon_a - H_0) | \Psi_a^S \rangle, \quad (1.124)$$

with H_0 acting on Ψ_a^S . We will also assume that ψ_a is a plane-wave two-particle state which can be written as

$$\psi_a = \frac{1}{\Omega} e^{i\mathbf{k}\mathbf{r}} e^{i\mathbf{K}\mathbf{R}} = \frac{1}{\sqrt{\Omega}} \psi(\mathbf{r}) e^{i\mathbf{K}\mathbf{R}}, \quad (1.125)$$

where \mathbf{k} and \mathbf{K} denote respectively the relative and center-of-mass momenta. The term Ω in the denominators serves to normalize the wave function and is the volume of the system. We approximate H_0 with the two-particle kinetic energy, denoted by the operators T_r and T_R for the relative and center-of-mass systems, respectively. The energy ε_a is then just $k^2/M_N + K^2/4M_N$, with M_N being the mass of the nucleon. Since the potential V_S is a function of r only, it has no effect on the center-of-mass wave function. The correlated wave function takes the form

$$\Psi_a^S = \frac{1}{\sqrt{\Omega}} \Psi_a^S(r) e^{i\mathbf{K}\mathbf{R}}. \quad (1.126)$$

Since $V_S(r) = 0$ for $r > d$, the matrix of eq. (1.124) is given by, using Greens theorem,

$$\langle \psi | G_S | \psi \rangle = \frac{1}{\Omega M_N} \oint_{r=d} [\Psi^S \nabla \psi^* - \psi^* \nabla \Psi^S] dS, \quad (1.127)$$

where the integral is a surface integral over a sphere with radius equal d .

The separation distance is still not yet specified. The main idea of the separation method is to choose d such that the above surface integral vanishes. This is possible for the relative $l = 0$ partial wave. In this case we write the integrand as

$$\Psi^S \nabla \psi^* - \psi^* \nabla \Psi^S = \Psi^S \frac{\partial \psi^*}{\partial r} - \psi^* \frac{\partial \Psi^S}{\partial r}$$

and the above integral vanishes if the wave function obeys the logarithmic boundary condition

$$\frac{1}{\psi} \frac{\partial \psi}{\partial r} = \frac{1}{\Psi^S} \frac{\partial \Psi^S}{\partial r} \quad \text{at} \quad r = d. \quad (1.128)$$

Here we assume that the potential has a hard core and hence the correlated wave function is identically equal to zero inside the core radius r_c . Note that the unperturbed wave function ψ is not vanishing for $r < r_c$. The correlated wave function Ψ_S is pushed out by the core, but once outside r_c it is pulled in by the exterior attraction. The separation distance is chosen such that there is just enough exterior attraction to balance the core repulsion so that the correlated wave function heals at $r = d$. By healing we mean the logarithmic condition in the above equation. This condition means that V_S results in a zero phase shift. Hence an equivalent definition of d is that the short-range potential V_S produces zero phase shifts. Note that this is not always possible, as we will discuss in the next subsection. In this case the separation distance does not exist. Hitherto we have only exposed the basic ideas on which the separation method is founded. Our original purpose is the calculation of the full G -matrix, not only that of G_S . In fact, G_S is merely an auxiliary potential, introduced as an intermediate step in the calculation of G . Recall also that we only calculated the diagonal matrix element of G_S , and that these vanish. The hope is however that the full G_S is small enough so that G may be calculated by a low-order perturbation expansion in terms of G_S . Using eq. (1.118) we may write G as

$$G \approx G_S^{\dagger} + G_S^{\dagger} \left(\frac{Q}{\epsilon} - \frac{Q_2}{\epsilon_2} \right) G_S + \left(1 + G_S^{\dagger} \frac{1}{\epsilon_2} \right) V_L \left(1 + \frac{1}{\epsilon_2} G_S \right). \quad (1.129)$$

The second term on the rhs. may be written as

$$G_S^{\dagger} \frac{Q - 1}{\epsilon} G_S + G_S^{\dagger} \left(\frac{1}{\epsilon} - \frac{1}{\epsilon_2} \right) G_S,$$

where the first term has commonly been denoted the Pauli correction term, while the second is referred to as the dispersion correction term.

Here we note that one of the appealing features of the separation method is the physically intuitive picture provided by it, namely that the long-range part V_L is the leading term of G . Nevertheless, there is the lingering question about the accuracy of this method. How accurate is G given by, for example, the low-order expansion of eq. (1.129)? Higher-order correction terms are progressively more difficult to calculate, and in most actual cases, the numerics sets a limit to how far we can go in this direction. It will indeed be very desirable if one could formulate a different approach, where one has the possibility of calculating G almost exactly. We will see how to obtain such a scheme in our calculations of the G -matrix for finite nuclei and nuclear matter. Below, we will however briefly revisit the so-called reference-spectrum method introduced by Bethe, Brandow and Petschek [3], an excellent presentation of this method can be found in the review article of Day [20], which in spite of its age still reads well.

The reference-spectrum method

This method is also based on the identity of eq. (1.115). Here we choose an auxiliary G -matrix G_R , the reference-spectrum G -matrix, defined by

$$G_R = V + V \frac{1}{\epsilon_R} G_R. \quad (1.130)$$

Using eq. (1.115), the G -matrix can be written as

$$G = G_R^{\dagger} + G_R^{\dagger} \left(\frac{Q}{\epsilon} - \frac{1}{\epsilon_R} \right) G. \quad (1.131)$$

Unlike the separation method, we note that we have the whole potential V in the definition of G_R . Moreover, for some partial-wave channels, the potential is repulsive for all r and we have no exterior attraction. In this case the separation method we discussed in the previous subsection, can not be used, as there is no exterior attraction to heal the wave function. It is mainly this difficulty which motivated the introduction of the reference-spectrum method. In this method, one first evaluates the defect wave function χ and thereafter one calculates G_R in terms of χ . The correlated wave function Ψ^R is defined by

$$G_R | \psi_a \rangle = V | \Psi_a^R \rangle \quad (1.132)$$

where ψ_a is the unperturbed wave function. From eq. (1.130), we have

$$| \Psi_a^R \rangle = | \psi_a \rangle + \frac{1}{\epsilon_R} V | \Psi_a^R \rangle. \quad (1.133)$$

The defect wave function is defined as the difference

$$\chi_a = \psi_a - \Psi_a^R. \quad (1.134)$$

In other words, the correlated wave function is equal to the unperturbed wave function minus the defect wave function (we will see examples of various defect wave functions in the next subsection, where we discuss nuclear matter results). Eq. (1.133) then leads to the result

$$(\epsilon_R - V) | \chi \rangle = -V | \psi \rangle, \quad (1.135)$$

where we for convenience have dropped the subscript a . The G_R -matrix is just an auxiliary G -matrix, which we need in our determination of the full G -matrix defined in eq. (1.131). We need however to define the energy denominator ϵ_R . In principle G is independent of ϵ_R , though in practice we hope that G_R alone should provide a good approximation to G . Suppose we define

$$\epsilon_R = \omega - T_r - T_R, \quad (1.136)$$

where T_r and T_R denote respectively the relative and center-of-mass kinetic energies of the two interacting particles. Since V depends on the internucleon distance r only, we can write the wave functions

$$\psi(\mathbf{r}_1 \mathbf{r}_2) = \frac{1}{\sqrt{\Omega}} \psi(\mathbf{r}) e^{i \mathbf{K} \mathbf{R}}, \quad (1.137)$$

and

$$\chi(\mathbf{r}_1 \mathbf{r}_2) = \frac{1}{\sqrt{\Omega}} \chi(\mathbf{r}) e^{i \mathbf{K} \mathbf{R}}, \quad (1.138)$$

as was done in connection with the separation method. Eq. (1.135) takes then the form (in partial waves)

$$\left(\omega - \frac{K^2}{4M_N} + \frac{1}{M_N} \frac{d^2}{dr^2} - \frac{l(l+1)}{r^2} - V \right) \chi_l(r) = -V \psi_l(r). \quad (1.139)$$

The asymptotic behaviors of the defect wave function χ can now be readily seen. We let V be a hard-core potential with core radius r_c . For $r = r_c$, V is infinite and we must therefore have $\chi_l(r) = \psi_l(r)$ inside r_c . Now comes an important point. If $(\omega - K^2/4M_N)$ is negative, we can define

$$-\frac{\gamma^2}{M_N} = \omega - \frac{K^2}{4M_N}, \quad (1.140)$$

with $\gamma > 0$. Eq. (1.139) gives then

$$\chi_l(r) \propto e^{-\gamma r} \quad \text{as} \quad r \rightarrow \infty \quad (1.141)$$

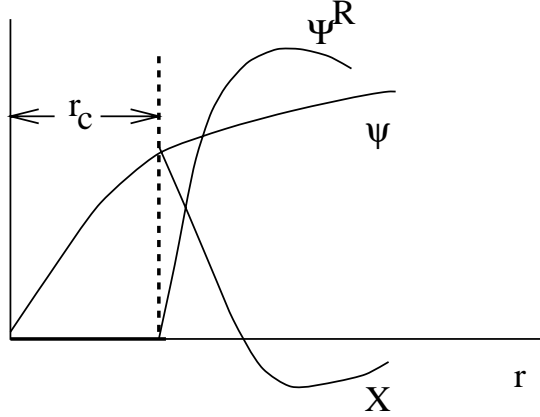


Figure 1.6: A typical defect wave function χ for a hard-core potential with core radius r_c .

Since $\Psi^R = \psi - \chi$, we see that the correlated wave function Ψ heals to ψ at large r values. A typical behavior of the defect wave function for the $l = 0$ channel is illustrated in fig. 1.6. We see that the correlated wave function Ψ^R is totally pushed out from the hard core and is zero inside r_c . At large r -values it is bound to heal to ψ . At some intermediate values of r , Ψ^R overshoots ψ in order to accommodate the wave function displaced by the core. With $\chi(r)$ obtained by solving the differential equation indicated by eq. (1.139), we can calculate the matrix element

$$\langle \psi_a | G_R | \psi_b \rangle = \langle \psi_a | V | \psi_b - \chi_b \rangle, \quad (1.142)$$

where the radial integral is now well-defined, even for a hard-core potential. For a hard-core potential $\langle \psi | V | \psi \rangle$ is not well-defined, but $\langle \psi | V | \psi - \chi \rangle$ is since $\psi - \chi = 0$ inside the hard core. A computational advantage of the reference-spectrum method is that it provides a convenient scheme, as outlined above, to obtain the reference G -matrix G_R . However, our goal is not only to obtain G_R but the full G -matrix G . The hope is that G_R is already a good approximation to G , and that G may be accurately calculated by way of a low-order perturbation expansion in terms of G_R . For example, we may approximate eq. (1.131) by

$$\begin{aligned} G &\approx G_R^\dagger + G_R^\dagger \left(\frac{Q}{\epsilon} - \frac{1}{\epsilon_R} \right) G_R^\dagger \\ &= G_R^\dagger + G_R^\dagger \left(\frac{Q}{\epsilon} - \frac{1}{\epsilon} \right) G_R^\dagger + G_R^\dagger \left(\frac{1}{\epsilon} - \frac{1}{\epsilon_R} \right) G_R^\dagger. \end{aligned} \quad (1.143)$$

Here the second and third terms are usually referred to as the Pauli- and the dispersion-correction terms, respectively. However, in spite of the nice features of the reference-spectrum method, we are still left with the problem of how to obtain a reliable estimate of higher-order terms of the full G -matrix in terms of G_R . In the next subsection we demonstrate how one can obtain (within the framework of the angle-average approximation) an “exact” solution of the G -matrix equation for nuclear matter.

1.4.2 The G -matrix for nuclear matter

In a medium such as nuclear matter we must account for the fact that certain states are not available as intermediate states in the calculation of the G -matrix. Following the discussion in the previous two subsections, this is achieved by introducing the medium dependent Pauli operator Q in eq. (1.74). Further,

the energy ω of the incoming particles, given by a pure kinetic term in eq. (1.74), must be modified so as to allow for medium corrections. How to evaluate the Pauli operator for e.g. nuclear matter is, however, not straightforward. Before discussing how to evaluate the Pauli operator for nuclear matter, we note that the G -matrix is conventionally given in terms of partial waves and the coordinates of the relative and center-of-mass motion, as in eq. (1.74). If we assume that the G -matrix is diagonal in α (α is a shorthand notation for J, S, L and T), it can then be recast in the form of a coupled-channels equation in the relative and center-of-mass system [21]

$$G_{ii'}^\alpha(kk' K \omega) = V_{ii'}^\alpha(kk') + \sum_{i''} \int \frac{d^3 q}{(2\pi)^3} V_{ii''}^\alpha(kq) \frac{Q(q, K)}{\omega - H_0} G_{i''i'}^\alpha(qk' K \omega). \quad (1.144)$$

This equation is similar in structure to the scattering equations for T , except that we now have introduced the Pauli operator Q and a medium dependent two-particle energy ω . The notations in this equation follow those of where we discuss the solution of the scattering matrix T .

Note however that the G -matrix may not be diagonal in α . This is due to the fact that the Pauli operator Q is not diagonal in the above representation in the relative and center-of-mass system. The Pauli operator depends on the angle between the relative momentum and the center of mass momentum. This angle dependence causes Q to couple states with different relative angular momentum J . A partial wave decomposition becomes therefore rather difficult. The angle dependence of the Pauli operator can be eliminated by introducing the angle-average Pauli operator, where one replaces the exact Pauli operator Q by its average \bar{Q} over all angles for fixed relative and center-of-mass momenta. The choice of Pauli operator is decisive to the determination of the sp spectrum. Basically, to first order in the reaction matrix G , there are three commonly used sp spectra, all defined by the solution of the following equations

$$\varepsilon_m = \varepsilon(k_m) = t_m + u_m = \frac{k_m^2}{2M_N} + u_m, \quad (1.145)$$

and

$$u_m = \sum_{h \leq k_F} \langle mh | G(\omega = \varepsilon_m + \varepsilon_h) | mh \rangle_{AS} \quad k_m \leq k_M, \quad (1.146)$$

$$u_m = 0, \quad k_m > k_M.$$

For notational economy, we set $|\mathbf{k}_m| = k_m$. Here we employ antisymmetrized matrix elements (AS), and k_M is a cutoff on the momentum. Further, t_m is the sp kinetic energy and similarly u_m is the sp potential. The choice of cutoff k_M is actually what determines the three commonly used sp spectra. In the conventional BHF approach one employs $k_M = k_F$, which leads to a Pauli operator Q_{BHF} (in the laboratory system) given by

$$Q_{\text{BHF}}(k_m, k_n) = \begin{cases} 1, & \min(k_m, k_n) > k_F \\ 0, & \text{else.} \end{cases}, \quad (1.147)$$

or, since we will define an angle-average Pauli operator in the relative and center-of-mass system, we have

$$\bar{Q}_{\text{BHF}}(k, K) = \begin{cases} 0, & k \leq \sqrt{k_F^2 - K^2/4} \\ 1, & k \geq k_F + K/2 \\ \frac{K^2/4 + k^2 - k_F^2}{kK}, & \text{else,} \end{cases} \quad (1.148)$$

with k_F the momentum at the Fermi surface. The difference between the Pauli operators in the lab and the relative and center-of-mass systems is shown in fig. 1.7 (a) and (b), respectively. See e.g. [21] for further details. The BHF choice sets $u_k = 0$ for $k > k_F$, which leads to an unphysical, large gap at the Fermi surface, typically of the order of 50 – 60 MeV. To overcome the gap problem, Mahaux and collaborators [22] introduced a continuous sp spectrum for all values of k . The divergencies which then may occur in eq. (1.144) are taken care of by introducing a principal value integration in eq. (1.144), to retain only the real part contribution to the G -matrix.

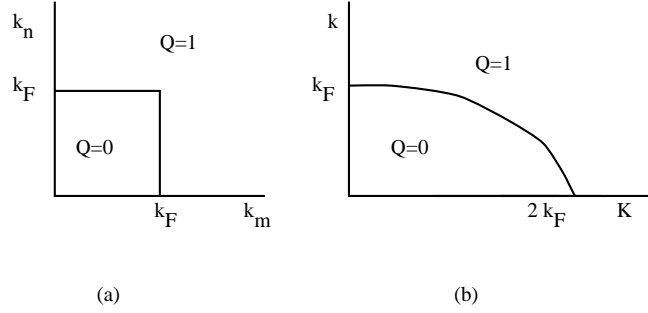


Figure 1.7: In (a) we show the Pauli operator for standard BHF theory in the lab frame, while (b) is the corresponding diagrammatic representation in the relative and center-of-mass system. Note that $\mathbf{K} = \mathbf{k}_n + \mathbf{k}_m$ and $\mathbf{k} = \frac{1}{2}(\mathbf{k}_n - \mathbf{k}_m)$.

To define the energy denominators we will also make use of the angle-average approximation. The angle dependence is handled by the so-called effective mass approximation. The single-particle energies in nuclear matter are assumed to have the simple quadratic form

$$\begin{aligned} \varepsilon(k_m) &= \frac{\hbar^2 k_m^2}{2M_N^*} + \Delta, & k_m \leq k_F \\ &= \frac{\hbar^2 k_m^2}{2M_N}, & k_m > k_F, \end{aligned} \quad (1.149)$$

where M_N^* is the effective mass of the nucleon and M_N is the bare nucleon mass. For particle states above the Fermi sea we choose a pure kinetic energy term, whereas for hole states, the terms M_N^* and Δ , the latter being an effective single-particle potential related to the G -matrix, are obtained through the self-consistent Brueckner-Hartree-Fock procedure. The sp potential is obtained through the same angle-average approximation [21]

$$\begin{aligned} U(k_m) &= \sum_{l\alpha} (2T+1)(2J+1) \left\{ \frac{8}{\pi} \int_0^{(k_F - k_m)/2} k^2 dk G_{ll}^{\alpha}(k, \bar{K}_1) \right. \\ &\quad \left. + \frac{1}{\pi k_m} \int_{(k_F - k_m)/2}^{(k_F + k_m)/2} k dk (k_F^2 - (k_m - 2k)^2) G_{ll}^{\alpha}(k, \bar{K}_2) \right\}, \end{aligned} \quad (1.150)$$

where we have defined

$$\bar{K}_1^2 = 4(k_m^2 + k^2), \quad (1.151)$$

and

$$\bar{K}_2^2 = 4(k_m^2 + k^2) - (2k + k_m - k_F)(2k + k_1 + k_F). \quad (1.152)$$

This self-consistency scheme consists in choosing adequate initial values of the effective mass and Δ . The obtained G -matrix is in turn used to obtain new values for M_N^* and Δ . This procedure continues until these parameters vary little.

1.5 Numerical evaluation of the G -matrix

In this section we outline how to calculate the G -matrix for nuclear matter. These methods can be viewed as an extension to what we discussed in connection with the solution of the Lippman-Schwinger equation. Let us first consider uncoupled channels, e.g. the 1S_0 partial wave. The one-dimensional equation for either the G -matrix can be written as

$$G(kk'K) = V(kk'K) - \int_0^{\infty} F(kqK)G(qk'K) dq, \quad (1.153)$$

where

$$F(kqK) = \frac{2}{\pi} q^2 \frac{V(kqK)Q(qK)}{q^2 - \omega - K^2}. \quad (1.154)$$

For nuclear matter we have that the Pauli operator Q is given by the angle-average operator.

For G the starting energy is negative, whereas for T it is positive. The above is an integral equation in the relative momentum k , and the G - or T -matrices are defined for a given center-of-mass momentum K . Below we will skip this variable. We assume also that G , V and Q are continuous in k and that their derivatives are also continuous. Further, we expect high-lying momenta to play a negligible role due to the denominator in F , and replace then the infinity limit with a finite value of q . In this way we can solve our problem numerically and rewrite the equation for G as

$$G(k_i, k_j) = V(k_i, k_j) - \sum_{n=1}^N F(k_i, k_n)G(k_n, k_j)w_n. \quad (1.155)$$

The numerical integration is done by using Gaussian points k_i and weights w_i

$$\begin{aligned} k_i &= C \tan\left(\frac{(1+x_i)\pi}{4}\right) \\ w_i &= C v_i \frac{\pi}{4} \cos^{-2}\left(\frac{(1+x_i)\pi}{4}\right), \end{aligned} \quad (1.156)$$

where $-1 \leq x \leq 1$ and C is a constant chosen so as to optimize the numerical grid. It is convenient to define G , F and V as $N \times N$ matrices in k -space (N is the number of mesh points) $F_{ij} = F(k_i, k_j)w_j$, $V_{ij} = V(k_i, k_j)$ and $G_{ij} = G(k_i, k_j)$. This allows us to rewrite the equation for G as a matrix equation

$$\begin{pmatrix} G_{11} & G_{12} & \cdots & G_{1N} \\ G_{21} & G_{22} & \cdots & G_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ G_{N1} & G_{N2} & \cdots & G_{NN} \end{pmatrix} = \begin{pmatrix} V_{11} & V_{12} & \cdots & V_{1N} \\ V_{21} & V_{22} & \cdots & V_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ V_{N1} & V_{N2} & \cdots & V_{NN} \end{pmatrix} - \begin{pmatrix} F_{11} & F_{12} & \cdots & F_{1N} \\ F_{21} & F_{22} & \cdots & F_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ F_{N1} & F_{N2} & \cdots & F_{NN} \end{pmatrix} \begin{pmatrix} G_{11} & G_{12} & \cdots & G_{1N} \\ G_{21} & G_{22} & \cdots & G_{2N} \\ \vdots & \vdots & \ddots & \vdots \\ G_{N1} & G_{N2} & \cdots & G_{NN} \end{pmatrix} \quad (1.157)$$

or in a more compact form

$$\begin{aligned} \mathbf{G} &= \mathbf{V} - \mathbf{F}\mathbf{G} \\ (\hat{\mathbf{1}} + \mathbf{F})\mathbf{G} &= \mathbf{V} \\ \mathbf{G} &= (\hat{\mathbf{1}} + \mathbf{F})^{-1}\mathbf{V}. \end{aligned} \quad (1.158)$$

The G -matrix can now be obtained for a given starting energy and relative momenta by inverting a matrix $(\hat{\mathbf{1}} + \mathbf{F})$ with the following matrix elements

$$\delta_{ij} + \frac{2}{\pi} V_{ij} \frac{Q_j}{k_j^2 - \omega} k_j^2 w_j.$$

Hitherto we have only considered uncoupled partial waves, though the extension to coupled channels is easily accomplished by defining a $2N \times 2N$ matrix $G_{ll'}$ (this matrix represents $G_{ll'}^{\alpha}(k; k_j K \omega)$) The couplings between various angular momenta are then accounted through the solution of

$$\begin{pmatrix} G_{ll} & G_{ll'} \\ G_{l'l} & G_{l'l'} \end{pmatrix} = \begin{pmatrix} V_{ll} & V_{ll'} \\ V_{l'l} & V_{l'l'} \end{pmatrix} - \begin{pmatrix} F_{ll} & F_{ll'} \\ F_{l'l} & F_{l'l'} \end{pmatrix} \begin{pmatrix} G_{ll} & G_{ll'} \\ G_{l'l} & G_{l'l'} \end{pmatrix}, \quad (1.159)$$

which means that we need to invert a $(2N \times 2N)$ -matrix. As an example, consider the 3S_1 - 3D_1 channel. There we have four different G -matrices, given by the partial wave combinations 3S_1 - 3S_1 , 3S_1 - 3D_1 , 3D_1 - 3S_1 and 3D_1 - 3D_1 .

A numerical stable and accurate (to 5–6 leading digits) result for both G in nuclear matter and G_F for finite nuclei is obtained using 40 – 50 mesh points. The above method is rather simple and efficient to implement numerically. This is demonstrated in the numerical code for the solution of the T -matrix.

The enclosed program for the T -matrix can easily be expanded in order to include the Pauli operator as well.

Chapter 2

Numerical studies of neutron stars

2.1 Introduction to the physics of neutron stars

The discovery of the neutron by Chadwick in 1932 prompted Landau to predict the existence of neutron stars. The birth of such stars in supernovae explosions was suggested by Baade and Zwicky 1934. First theoretical neutron star calculations were performed by Tolman, Oppenheimer and Volkoff in 1939 and Wheeler around 1960. Bell and Hewish were the first to discover a neutron star in 1967 as a *radio pulsar*. The discovery of the rapidly rotating Crab pulsar (rapidly rotating neutron star) in the remnant of the Crab supernova observed by the chinese in 1054 A.D. confirmed the link to supernovae. Radio pulsars are rapidly rotating with periods in the range $0.033 \text{ s} \leq P \leq 4.0 \text{ s}$. They are believed to be powered by rotational energy loss and are rapidly spinning down with period derivatives of order $\dot{P} \sim 10^{-12} - 10^{-16}$. Their high magnetic field B leads to dipole magnetic braking radiation proportional to the magnetic field squared. One estimates magnetic fields of the order of $B \sim 10^{11} - 10^{13} \text{ G}$. The total number of pulsars discovered sofar has just exceeded 1000 before the turn of the millenium and the number is increasing rapidly.

A distinct subclass of radio pulsars are *millisecond pulsars* with periods between $1.56 \text{ ms} \leq P \leq 100 \text{ ms}$. The period derivatives are very small corresponding to very small magnetic fields $B \sim 10^8 - 10^{10} \text{ G}$. They are believed to be recycled pulsars, i.e. old pulsars with low magnetic fields that have been spun up by accretion preserving their low magnetic field and therefore only slowly spinning down. About 20 - almost half of the millisecond pulsars - are found in binaries where the companion is either a white dwarf or a neutron star. Six double neutron stars are known sofar including the Hulse-Taylor PSR 1913+16. The first binary pulsar was found by Hulse and Taylor in 1973 and by measuring the general relativistic corrections to Newtonian gravity one could determine all parameters in the binary system as both masses, orbital periods and period derivatives, orbital distances and inclination. Parameters are overdetermined and thus provides a test of general relativity. Inward spiralling or orbital decay is an additional test of general relativity to an unprecedented accuracy. The binary neutron stars all have masses in the narrow interval $1.3 - 1.5 M_{\odot}$, which may either be due to the creation process or that heavier neutron stars are unstable. Non-rotating and non-accreting neutron stars are virtually undetectable. With the Hubble space telescope one single thermally radiating neutron star has been found. Its distance is only 160 pc from earth and its surface temperature is $T \simeq 60 \text{ eV}$. From its luminosity one deduces a radius of the neutron star $R \leq 14 \text{ km}$. In our galaxy astrophysicists expect a large abundance $\sim 10^8$ of neutron stars. At least as many supernova explosions have occurred since Big Bang which are responsible for all heavier elements present in the Universe today. The scarcity of neutron stars in the solar neighborhood may be due to a high initial velocity (asymmetric "kick") during their birth in supernovae. Recently, many neutron stars have been found far away from their supernova remnants. Future gravitational microlensing observation may determine the population of such "invisible" neutron stars as dark matter objects in the galactic halo. From the view of physicists (and mass extinctionists) supernova explosions are unfortunately rare in our and neighboring galaxies. The predicted rate is 1-3 per century in our galaxy but the most recent one was 1987A in LMC. With luck we may observe one in the near future which produces a rapidly rotating pulsar.

Light curves and neutrino counts will test supernova and neutron star models. The rapid spin down may be exploited to test the structure and possible phase transitions in the cores of neutron stars. The physics of compact objects like neutron stars offers an intriguing interplay between nuclear processes and astrophysical observables. Neutron stars exhibit conditions far from those encountered on earth; typically, expected densities ρ of a neutron star interior are of the order of 10^3 or more times the density $\rho_d \approx 4 \cdot 10^{11} \text{ g/cm}^3$ at 'neutron drip', the density at which nuclei begin to dissolve and merge together. Thus, the determination of an equation of state (EoS) for dense matter is essential to calculations of neutron star properties. The EoS determines properties such as the mass range, the mass-radius relationship, the crust thickness and the cooling rate. The same EoS is also crucial in calculating the energy released in a supernova explosion. Clearly, the relevant degrees of freedom will not be the same in the crust region of a neutron star, where the density is much smaller than the saturation density of nuclear matter, and in the center of the star, where density is so high that models based solely on interacting nucleons are questionable. Neutron star models including various so-called realistic equations of state result in the following general picture of the interior of a neutron star. The surface region, with typical densities $\rho < 10^6 \text{ g/cm}^3$, is a region in which temperatures and magnetic fields may affect the equation of state. The outer crust for $10^6 \text{ g/cm}^3 < \rho < 4 \cdot 10^{11} \text{ g/cm}^3$ is a solid region where a Coulomb lattice of heavy nuclei coexist in β -equilibrium with a relativistic degenerate electron gas. The inner crust for $4 \cdot 10^{11} \text{ g/cm}^3 < \rho < 2 \cdot 10^{14} \text{ g/cm}^3$ consists of a lattice of neutron-rich nuclei together with a superfluid neutron gas and an electron gas. The neutron liquid for $2 \cdot 10^{14} \text{ g/cm}^3 < \rho < 10^{15} \text{ g/cm}^3$ contains mainly superfluid neutrons with a smaller concentration of superconducting protons and normal electrons. At higher densities, typically 2 - 3 times nuclear matter saturation density, interesting phase transitions from a phase with just nucleonic degrees of freedom to quark matter may take place. Furthermore, one may have a mixed phase of quark and nuclear matter, kaon or pion condensates, hyperonic matter, strong magnetic fields in young stars etc.

2.2 Non-relativistic equilibrium equations

We start the discussion here using non-relativistic equations. This chapter ends however with the corresponding equations from general relativity.

If the star is in thermal equilibrium, the gravitational force on every element of volume will be balanced by a force due to the spacial variation of the pressure P . The pressure is defined by the equation of state (EoS), (recall e.g., the well-known expression $P = N k_B T$). The gravitational force which acts on an element of volume at a distance r is given by

$$F_{Grav} = -\frac{Gm}{r^2} \rho / c^2, \quad (2.1)$$

where G is the gravitational constant, $\rho(r)$ is the energy-density (also used is mass-energy density) and $m(r)$ is the total mass inside a radius r . The latter is given by

$$m(r) = \frac{4\pi}{c^2} \int_0^r \rho(r') r'^2 dr' \quad (2.2)$$

which gives rise to a differential equation for mass and density

$$\frac{dm}{dr} = 4\pi r^2 \rho(r) / c^2. \quad (2.3)$$

When the star is in equilibrium we have

$$\frac{dP}{dr} = -\frac{Gm(r)}{r^2} \rho(r) / c^2. \quad (2.4)$$

The last equations give us two coupled first-order differential equations which determine the structure of a neutron star when the EoS is known.

These are the equations to be solved numerically. The enclosed program at the end of this chapter solves these equations using the Runge-Kutta method to fourth order.

The initial conditions are dictated by the mass being zero at the center of the star, i.e., when $r = 0$, yielding $m(r=0) = 0$. The other condition is that the pressure vanishes at the surface of the star. This means that

at the point where we have $P = 0$ in the solution of the integral equations, we get the total radius R of the star and the total mass $m(r = R)$. The mass-energy density when $r = 0$ is called the central density ρ_s . Since both the final mass M and total radius R will depend on ρ_s , a variation of this quantity will allow us to study stars with different masses and radii.

In the solution of our problem, we will assume that the energy density is given by a simple parametrization from Bethe and Johnson. This parametrization gives ρ as a function of the number density $n = N/V$, with N the total number of baryons in a volume V . It reads

$$\rho(n) = 236 \times n^{2.54} + nm_n c^2, \quad (2.5)$$

where $m_n = 938.926 \text{ MeV}/c^2$, the mass of the neutron (averaged). This means that since $[n] = \text{fm}^{-3}$, we have $[\rho] = \text{MeVfm}^{-3}$. Through the thermodynamic relation

$$P = -\frac{\partial E}{\partial V}, \quad (2.6)$$

where E is the energy in units of MeV/c^2 we have

$$P(n) = n \frac{\partial \rho(n)}{\partial n} - \rho(n) = 363.44 \times n^{2.54}. \quad (2.7)$$

We see that the units of P are the same as those of the energy density, i.e., $[P] = \text{MeVfm}^{-3}$.

Here comes an important point you should observe when solving the two coupled first-order differential equations. When you obtain the new pressure given by

$$P_{new} = \frac{dP}{dr} + P_{old}, \quad (2.8)$$

this comes as a function of r . However, having obtained the new pressure, you will need to use Eq. (2.7) in order to find the number density n . This will in turn allow you to find the new value of the mass-energy density $\rho(n)$ at the relevant value of r .

2.3 Numerical solution

When we now attempt the numerical solution, we need however to rescale the equations so that we deal with dimensionless quantities only. To understand why, consider the value of the gravitational constant G and the possible final mass $m(r = R) = M_R$. The latter is normally of the order of some solar masses M_\odot , with $M_\odot = 1.989 \times 10^{30} \text{ Kg}$. If we wish to translate the latter into units of MeV/c^2 , we will have that $M_R \sim 10^{60} \text{ MeV}/c^2$. The gravitational constant is in units of $G = 6.67 \times 10^{-45} \times \hbar c (\text{MeV}/c^2)^{-2}$. It is then easy to see that including the relevant values for these quantities in our equations will most likely yield large numerical roundoff errors when we add a huge number $\frac{dP}{dr}$ to a smaller number P in order to obtain the new pressure. We list here the units of the various quantities and in case of physical constants, also their values. A bracketed symbol like $[P]$ stands for the actual units of the quantity inside the brackets.

Quantity	Units
$[P]$	MeVfm^{-3}
$[\rho]$	MeVfm^{-3}
$[n]$	fm^{-3}
$[m]$	$\text{MeV}c^{-2}$
M_\odot	$1.989 \times 10^{30} \text{ Kg} = 1.1157467 \times 10^{60} \text{ MeV}c^{-2}$
1 Kg	$= 10^{30}/1.78266270D0 \text{ MeV}c^{-2}$
$[r]$	m
G	$\hbar c 6.67259 \times 10^{-45} \text{ MeV}^{-2}c^{-4}$
$\hbar c$	197.327 MeVfm

We introduce therefore dimensionless quantities for the radius $\bar{r} = r/R_0$, mass-energy density $\bar{\rho} = \rho/\rho_s$, pressure $\bar{P} = P/\rho_s$ and mass $\bar{m} = m/M_0$.

The constants M_0 og R_0 can be determined from the requirements that the equations for $\frac{d\bar{m}}{d\bar{r}}$ og $\frac{d\bar{P}}{d\bar{r}}$ should be dimensionless. This gives

$$\frac{dM_0 \bar{m}}{dR_0 \bar{r}} = 4\pi R_0^2 \bar{\rho}_s \bar{r}, \quad (2.9)$$

yielding

$$\frac{d\bar{m}}{d\bar{r}} = 4\pi R_0^3 \bar{\rho}_s \bar{r} / M_0. \quad (2.10)$$

If these equations should be dimensionless we must demand that

$$4\pi R_0^3 \rho_s / M_0 = 1. \quad (2.11)$$

Correspondingly, we have for the pressure equation

$$\frac{d\rho_s \bar{P}}{dR_0 \bar{r}} = -GM_0 \frac{\bar{m} \rho_s \bar{r}}{R_0^3 \bar{r}^2} \quad (2.12)$$

and since this equation should also be dimensionless, we will have

$$GM_0 / R_0 = 1. \quad (2.13)$$

This means that the constants R_0 and M_0 which will render the equations dimensionless are given by

$$R_0 = \frac{1}{\sqrt{\rho_s G 4\pi}}, \quad (2.14)$$

and

$$M_0 = \frac{4\pi \rho_s}{(\sqrt{\rho_s G 4\pi})^3}. \quad (2.15)$$

However, since we would like to have the radius expressed in units of 10 km, we should multiply R_0 by 10^{-19} , since $1 \text{ fm} = 10^{-15} \text{ m}$. Similarly, M_0 will come in units of MeV/c^2 , and it is convenient therefore to divide it by the mass of the sun and express the total mass in terms of solar masses M_\odot .

The differential equations read then

$$\frac{d\bar{P}}{d\bar{r}} = -\frac{\bar{m}\bar{P}}{\bar{r}^2}, \quad \frac{d\bar{m}}{d\bar{r}} = \bar{r}^2 \bar{P}. \quad (2.16)$$

In the algorithm below, written in form of a pseudocode, we have used the labels displayed in the table here. The variables `const_1` and `const_2` are

Labels in equation	Labels in pseudocode
\bar{P}	y(1)
P	pressure
$\rho(n)$	rho
$\rho_s(n_s)$	central_energy_density
n_s	central_density
\bar{r}	star_radius
\bar{m}	y(2)

$$\text{const_1} = R_0 \times 10^{-19}, \quad (2.17)$$

and

$$\text{const_2} = M_0 / M_\odot, \quad (2.18)$$

respectively.

The algorithm for solving these equations may take the following form

```

!   loop over central densities
FOR (i=1, number_central_densities)
!   Preparation of starting values
!   central density in fm^-3
!   central energy density in units MeV/fm^3
      central_energy_density=rho(central_density)
!   central pressure, in units of MeV/fm^-3
      pressure=press(central_density)
!   Dimensionless pressure at the centre (function of central dens)
      y(1)=pressure/central_energy_density
!   Dimensionless start radius
      star_radius=step/10.
!   Dimensionless mass of the star at the centre, use e.g. Taylor
!   expansion around r=0
      y(2)=start_mass
!   Start of Runge-Kutta solution to differential equations
!   number of RK4 steps given by variable j . Iterate as long as
!   P is greater than 0
      j=0
      WHILE((pressure > 0.).AND.(j <= max_rk4_steps))
        j=j+1
!   new dimless radius
      star_radius=star_radius+step
!   get the derivatives at the given value star_radius
      CALL derivatives(star_radius,y,deriv)
!   Here we use the RK4 procedure described in the next section
      CALL runge_kutta_4(star_radius,y,ynew,deriv)
!   new values for y(1) (pressure) and y(2) (mass), dimless
      y(1)=ynew(1)
      y(2)=ynew(2)
!   new pressure, now in units of MeV/fm^-3
!   use it to test whether P=0

```

```

      pressure=y(1)*central_energy_density
      END of test
!   radius in units of km
      radius=star_radius*const_1
!   mass in units of solar masses
      mass=y(2)*const_2
      WRITE to file central_density, mass and radius
      END of loop over chosen central densities

```

2.4 TOV equation

In the above we have limited ourselves to a non-relativistic description. The theoretical description of a neutron star is however governed by conditions imposed by general relativity. General relativity has to be taken into account for the determination of the gross properties of a star with approximately one solar mass M_{\odot} and a radius R of approximately 10 km, since relativistic effects are of the order [23]

$$\frac{M}{R} \sim 0.1 - 0.2. \quad (2.19)$$

The starting point for such studies is how to determine Einstein's curvature tensor $G_{\mu\nu}$ for a massive star ($R_{\mu\nu}$, $g_{\mu\nu}$, and R denote the Ricci tensor, metric tensor, and Ricci scalar, respectively).

$$G_{\mu\nu} \equiv R_{\mu\nu} - \frac{1}{2}g_{\mu\nu}R = 8\pi T_{\mu\nu}(\rho, P(\rho)). \quad (2.20)$$

A necessary ingredient for solving this equation is the energy-momentum tensor density $T_{\mu\nu}$, for which knowledge of the EoS, i.e. pressure P as function of the energy density ρ is necessary. For a spherically symmetric and static star, the metric has the Schwarzschild form

$$ds^2 = -e^{2\phi(r)}dt^2 + e^{2\Lambda(r)}dr^2 + r^2(d\theta^2 + \sin^2\theta d\phi^2), \quad (2.21)$$

where the metric functions are given by:

$$e^{2\Lambda(r)} = (1 - \gamma(r))^{-1}, \quad (2.22)$$

$$e^{2\phi(r)} = e^{-2\Lambda(r)} = (1 - \gamma(r)) \quad \text{for } r > R_{\text{star}}, \quad (2.23)$$

with

$$\gamma(r) = \begin{cases} 2M(r)/r & , \quad r \leq R_s \\ 2M_s/r & , \quad r \geq R_s \end{cases} \quad (2.24)$$

Einstein's equations for a static star reduce then to the familiar Tolman-Oppenheimer-Volkoff equation (TOV) [23, 24, 25]:

$$\frac{dP(r)}{dr} = -\frac{1}{r^2}(\rho(r) + P(r))(M(r) + 4\pi r^3 P(r)) e^{2\Lambda(r)}, \quad (2.25)$$

where the gravitational mass $M(r)$ contained in a sphere with radius r is determined via the energy-density $\rho(r)$ by:

$$M(r) = 4\pi \int_0^r \varepsilon(r) r^2 dr. \quad (2.26)$$

The metric function $\phi(r)$ obeys the differential equation

$$\frac{d\phi}{dr} = -\frac{1}{\rho(r) + P(r)} \frac{dP}{dr}, \quad (2.27)$$

with the boundary condition

$$\phi(r = R_s) = \frac{1}{2} \ln(1 - \gamma(R_s)). \quad (2.28)$$

For a given EoS, i.e. P , one can now solve the TOV equation by integrating them for a given central energy density ε_c from the star's centre to the star's radius, defined by $P(R_s) = 0$.

More complicated is the case of rotating stars, where due to the rotation changes occur in the pressure, energy density, etc. We will omit a discussion of that here and simply limit the attention in the program enclosed below to the TOV equation.

2.5 Code example for a simple EoS

```
!
!   Program to calculate the mass (SM) and radius (r) for a
!   neutron star
!   Free format is employed, compile with e.g.,
!   f90 -c -free star.f and link with f90 -o prog.exe star.o
!
!
!   The constants defined
!   below have the following dim
!   [hbarc]=197.327 MeVfm
!   [g] = m^3Kg^-1s^-2 gravitational const
!         = 6.67259*10^-11 m^3Kg^-1s^-2
!         or 6.67259 MeV^-2\hbarc/c^4
!   [sm_solar]=1.989D+30 kg
!   The solar mass is converted into MeV. Then
!   one has the relation
!         1 kg=1.D+30/1.78266270D0 1 MeV/c^2
!   All taken from the Particle Data Group
!   (1994)
!   The program runs with dim-less variables
!   P=P/central_energy_density
!   rho=rho/central_energy_density
!   with central_energy_density the central energy-density of the star
!   r=r/r0 with r0=1.D+19 fm= 10km
!   m=m/m_solar
!
!   Prints out central density in fm^-3, radius in m,
!   neutron star mass in terms of solar masses
!
!   Global constants used to render equations dim-less
!
MODULE constants
```

```
!   gravitational constant in fmMeV^-1
!   DOUBLE PRECISION, PUBLIC, PARAMETER :: g=6.67259D-45*197.327
!   4*\pi
!   DOUBLE PRECISION, PUBLIC, PARAMETER :: fourpi=12.56637061
!   number of central densities chosen
!   INTEGER, PUBLIC, PARAMETER :: number_central_densities=30
!   the differential equations of Hartle, excluding the quadropole terms
!   INTEGER, PUBLIC, PARAMETER :: number_differential_eqs= 2
!   number of runge-kutta iterations
!   INTEGER, PUBLIC, PARAMETER :: max_rk4_steps=100000
!   Dimensionless step in RK4 procedure
!   DOUBLE PRECISION, PUBLIC, PARAMETER :: diff_eq_step = 0.001
!   Density from which one wants the central density to be calculated
!   DOUBLE PRECISION, PUBLIC, PARAMETER :: first_density = 0.1
!   Last central density
!   DOUBLE PRECISION, PUBLIC, PARAMETER :: last_density = 1.8
!   1 MeV/c^2=1.78266270D-30 kg
!   DOUBLE PRECISION, PUBLIC, PARAMETER :: xkg=1.E+30/1.78266270
!   solar mass in MeV/c^2
!   DOUBLE PRECISION, PUBLIC, PARAMETER :: sm_solar=1.989E+30*xkg
!   END MODULE constants
!
!   This module contains the parametrization of the EOS as
!   a polynomial in density.
!
MODULE eos
  USE constants
  DOUBLE PRECISION, PUBLIC :: central_energy_density
  CONTAINS
!
!   rho: energy per particle in units of MeV/fm^3
!
  DOUBLE PRECISION FUNCTION rho(x)
  IMPLICIT NONE
  DOUBLE PRECISION, INTENT(IN) :: x
!
  rho=236.*(x**(2.54))+938.926*x
!
  END FUNCTION rho
!
!   pressure in units of MeV/fm^3
!
  DOUBLE PRECISION FUNCTION press(x)
  IMPLICIT NONE
  DOUBLE PRECISION, INTENT(IN) :: x
!
  press=363.44*(x**(2.54))
!
  END FUNCTION press
!
!   dP/dr, TOV equation, dimensionless
!
```



```

        FUNCTION tov(r,e_rho,y)
        IMPLICIT NONE
        DOUBLE PRECISION :: e_rho, tov, r
        DOUBLE PRECISION, DIMENSION(number_differential_eqs), INTENT(IN) :: y
! non-relativistic
        tov=-e_rho*y(2)/(r*r)
! relativistic
        tov=-(e_rho+y(1))* &
            (y(2)+(r**3)*y(1))/(r*r-2*r*y(2))
        END FUNCTION tov

END MODULE eos

!
! Main program starts here
!
PROGRAM neutron_star
USE eos
USE constants
IMPLICIT NONE
INTEGER :: j, i
DOUBLE PRECISION, DIMENSION (number_central_densities) :: cen_dens, &
    radius, smass
DOUBLE PRECISION, DIMENSION (number_differential_eqs) :: ynew, y, ders
DOUBLE PRECISION :: star_radius, ync, density_step, pressure, const_2, const_1

! steps in choice of central densities
density_step=(last_density-first_density)/number_central_densities

! loop over central densities

DO i=1, number_central_densities

! Preparation of starting values
! central density in fm^-3

        ync=first_density+density_step*i

! central energy density in units MeV/fm^3

        central_energy_density=rho(ync)

! central pressure, in units of MeV/fm^-3

        pressure=0.
        pressure=press(ync)

! dimless const_1 const_2

        const_1=1.E-19/SQRT(fourpi*g*central_energy_density)
        const_2=fourpi*central_energy_density/ &
            (SQRT(fourpi*g*central_energy_density)**3)
! Dimensionless pressure at the centre (function of central dens)

```

```

        y(1)=pressure/central_energy_density
! Dimensionless start radius

        star_radius=diff_eq_step/10.

! Dimensionless mass of the star at the centre

        y(2)=(star_radius**3)/3.

! Start of Runge-Kutta solution to differential equations
! number of RK4 steps given by variable j

        j=0
        DO WHILE((pressure > 0.).AND.(j <= max_rk4_steps))
            j=j+1
! new dimless radius
            star_radius=star_radius+diff_eq_step
! start values for derivatives
            CALL derivatives(star_radius,y,ders)
! RK4 procedure
            CALL runge_kutta_4(star_radius,y,ynew,ders)
! new values for y(1) (pressure), y(2) (mass) and y(3) (metric function)
            y=ynew
! new pressure, now in units of MeV/fm^-3
            pressure=y(1)*central_energy_density
            ENDDO

! central density in units of fm^-3

            cen_dens(i)=ync

! radius in units of m

            radius(i)=star_radius*const_1

! mass in units of solar masses

            smass(i)=y(2)*const_2/sm_solar
            WRITE(6,'(E12.3,6E12.4)') cen_dens(i),radius(i),smass(i)
        ENDDO

END PROGRAM neutron_star

!
! 4th-Runge-Kutta solution of coupled equations
! See any textbook on numerical methods for details
!

SUBROUTINE runge_kutta_4(x,y,yout,dydx)
USE constants
IMPLICIT NONE
DOUBLE PRECISION, DIMENSION(number_differential_eqs) :: yt, dyt, dym
DOUBLE PRECISION, DIMENSION(number_differential_eqs), INTENT(IN) :: y, dydx
DOUBLE PRECISION, DIMENSION(number_differential_eqs), INTENT(OUT) :: yout

```

```

DOUBLE PRECISION :: hh, h6, xh
DOUBLE PRECISION, INTENT(IN) :: x

hh=diff_eq_step*0.5; h6=diff_eq_step/6. ; xh=x+hh
! first rk-step
yt=y+hh*dydx
CALL derivatives(xh,yt,dyt)
! second rk-step
yt=y+hh*dyt
CALL derivatives(xh,yt,dym)
! third rk-step
yt=y+diff_eq_step*dym; dym=dyt+dym
CALL derivatives(x+diff_eq_step,yt,dyt)
! fourth rk-step
yout=y+h6*(dydx+dyt+2.*dym)

END SUBROUTINE runge_kutta_4
!
! Here the expressions for the derivatives are set up
!
SUBROUTINE derivatives(r,y,ders)
USE eos
USE constants
IMPLICIT NONE
DOUBLE PRECISION :: e_rho, x
DOUBLE PRECISION, INTENT(IN) :: r
DOUBLE PRECISION, DIMENSION(number_differential_eqs), INTENT(OUT) :: ders
DOUBLE PRECISION, DIMENSION(number_differential_eqs), INTENT(IN) :: y

IF(y(1) > 0.) THEN
  x=(y(1)*central_energy_density/363.44)**(1./2.54)
  e_rho=rho(x)/central_energy_density
! TOV equation=dp/dr
  ders(1)=tov(r,e_rho,y)
! dm/dr=r*r*energydensity, gravitational mass as function of r
  ders(2)=(r**2)*e_rho
ENDIF

END SUBROUTINE derivatives

```

Appendix A

A Fortran 90/95 primer

A.1 Introduction to F90/F95

Fortran (FORMula TRANslation) was introduced in 1957 and remains the language of choice for most scientific programming. The latest standard, Fortran 90 and 95, includes extensions that are familiar to users of C. Some of the most important features of Fortran 90 include recursive subroutines, dynamic storage allocation and pointers, user defined data structures, modules, and the ability to manipulate entire arrays.

Fortran 90 is compatible with Fortran 77 and includes syntax that is no longer considered desirable. Fortran 90 does however not allow for pointer algebra. This feature is included in Fortran 95 together with other features such as the FORALL statement.

A.1.1 Introduction

In order to get started, consider the following simple Fortran 90 program which sets up Newton's second law.

```
PROGRAM newton
  IMPLICIT NONE
  DOUBLE PRECISION :: mass, acceleration, force

! write to screen and ask for the mass in kg
  WRITE(*,*) 'Give the mass in units of kg'
! read in the value
  READ(*,*) mass
! write to screen and ask for the acceleration in ms-2
  WRITE(*,*) 'Give the acceleration'
! read in the value
  READ(*,*) acceleration
  force = mass*acceleration
! write back to screen
  WRITE(*,*) mass, acceleration, force

END PROGRAM newton
```

The first statement must be a program statement; the last statement must have a corresponding end program statement.

Integer numerical variables and floating point numerical variables are distinguished. The names of all variables must be between 1 and 31 alphanumeric characters of which the first must be a letter and the last must not be an underscore.

The types of all variables must be declared. Real numbers are written as 2.0 rather than 2 and declared as DOUBLE PRECISION. In general we discourage the use of single precision in scientific computing, the achieved precision is in general not good enough. Comments begin with a ! and can be included anywhere in the program. Statements are written on lines which may contain up to 132 characters. The asterisks (**) following WRITE represent the default format for output, i.e., the output is e.g., written out on the screen. Similarly, the READ(**,*) statement means that the program is expecting a line input. Note also the IMPLICIT NONE statement which we strongly recommend the use of. In many Fortran 77 one can see statements like IMPLICIT REAL*8(a-h,o-z), meaning that all variables beginning with any of the above letters are by default floating numbers. However, such a usage makes it hard to spot eventual errors due to misspelling of variable names. With IMPLICIT NONE you have to declare all variables and therefore detect possible errors already while compiling.

A.1.2 DO construct

Fortran 90/95 use a do construct to have the computer execute the same statements more than once. An example of a do construct follows from example 4 in chapter 2. There we summed $1/n$ up to a given number, say 1000

```
PROGRAM series
  IMPLICIT NONE
  DOUBLE PRECISION :: sum
  INTEGER :: n

! Initialize the sum
  sum = 0.
  DO n = 1, 1000
    sum = sum + 1.0/FLOAT(n)
    WRITE(*,*) n,sum
  ENDDO

END PROGRAM series
```

Note that n is an integer variable. In this case the do statement specifies the first and last values of n ; n increases by unity (default). Note here that we wish to avoid a division with by an integer through the use of FLOAT(n). Moreover, Fortran does not allow floating numbers as loop variables.

A.1.3 Logical constructs

In the next program example, the do loop is exited by satisfying a test.

```
PROGRAM series_test
  IMPLICIT NONE
  DOUBLE PRECISION :: sum, newterm, relative_change
  INTEGER :: n

! initialize sum, newterm and relative_change
  sum = 0.; newterm=0.; relative_change=0.

  DO n = 1, 1000
    newterm=1.0/FLOAT(n)
    sum = sum + newterm
    relative_change = newterm/sum
    IF ( relative_change < 0.00001 ) EXIT
  ENDDO
  WRITE(*,*) n, sum, relative_change

END PROGRAM series_test
```

The features included in the above program include:

A do construct can be exited by using the EXIT statement. The IF construct allows the execution of a sequence of statements (a block) to depend on a condition. The if construct is a compound statement and begins with IF ... THEN and ends with ENDIF. Examples of more general IF constructs using ELSE and ELSEIF statements are given in the program library or in the main text. Another feature to observe is the CYCLE command, which allows the loop variable n to start at a new value. As a rule of thumb, if possible, you should avoid IF statements or calls to other functions if you operate on arrays inside loops. This may reduce the effect of optimizations gained through various compiler options.

A.1.4 Subprograms

Subprograms are called from the main program or other subprograms. Subprograms (subroutines and functions) can be included in modules. The form of a module, subroutine, and a function is similar to that of a main program. A module is accessed in the main program by the use statement. Subroutines are invoked in the main program by using the call statement. A subprogram always has access to other entities in the module. The subprograms in a module are preceded by a contains statement.

Variables and subprograms may be declared public in a module and be available to the main program (and other modules). An example follows here.

```
!
!   This module contains the parametrization of the EOS as
!   a polynomial in density. The number of terms kept in the
!   polynomial expansion is given by number_terms.
!
MODULE eos
  DOUBLE PRECISION, ALLOCATABLE, DIMENSION(:), PUBLIC :: polynom_terms
  INTEGER, PUBLIC :: number_terms

  CONTAINS

!   rho: energy per particle in units of MeV/fm^3
!   rho=\sum_{i=1}^{number of polynoms} a_i*density^{(i-1)/3}
!
  DOUBLE PRECISION FUNCTION rho(x)
  IMPLICIT NONE
  DOUBLE PRECISION, INTENT(IN) :: x
  INTEGER :: i

  rho=polynom_terms1(1)
  DO i=2,number_terms1
    rho=rho+polynom_terms1(i)*(x**(FLOAT(i-1)/3.d0))
  ENDDO

  END FUNCTION rho

!
!   pressure in units of MeV/fm^3
!   pressure = density*d rho/d density - rho
!
  DOUBLE PRECISION FUNCTION press(x)
  IMPLICIT NONE
  DOUBLE PRECISION, INTENT(IN) :: x
  INTEGER :: i

  press=-rho(x)
```

```
DO i=2,number_terms1
  press=press+(FLOAT(i-1)/3.0*polynom_terms1(i)*(x**(FLOAT(i-1)/3.)))
ENDDO

END FUNCTION press

END MODULE eos
```

INTENT(IN) means that the dummy argument cannot be changed within the subprogram. INTENT(OUT) means that the dummy argument cannot be used within the subprogram until it is given a value with the intent of passing a value back to the calling program. The statement INTENT(INOUT) means that the dummy argument has an initial value which is changed and passed back to the calling program. We recommend that you use these options when calling subprograms. The module(s) can be included in a separate file.

A.1.5 Arrays

An array is declared in the declaration section of a program, module, or procedure using the dimension attribute. Examples include

```
DOUBLE PRECISION, DIMENSION (10) :: x,y
REAL, DIMENSION (1:10) :: x,y
INTEGER, DIMENSION (-10:10) :: prob
INTEGER, DIMENSION (10,10) :: spin
```

The default value of the lower bound of an array is 1. For this reason the first two statements are equivalent to the first. The lower bound of an array can be negative. The last statement is an example of two-dimensional array.

Rather than assigning each array element explicitly, we can use an array constructor to give an array a set of values. An array constructor is a one-dimensional list of values, separated by commas, and delimited by "(/" and "/)". An example is

```
a(1:3) = (/ 2.0, -3.0, -4.0 /)
```

is equivalent to the separate assignments

```
a(1) = 2.0
a(2) = -3.0
a(3) = -4.0
```

A.1.6 Allocate statement and mathematical operations on arrays

One of the better features of Fortran 90 is dynamic storage allocation. That is, the size of an array can be changed during the execution of the program. To see how the dynamic allocation works in Fortran 90, consider the following simple example where we set up a 4×4 unity matrix.

```
.....
IMPLICIT NONE
!   The definition of the matrix, using dynamic allocation
DOUBLE PRECISION, ALLOCATABLE, DIMENSION(:,:) :: unity
!   The size of the matrix
INTEGER :: n
!   Here we set the dim n=4
n=4
!   Allocate now place in memory for the matrix
ALLOCATE ( unity(n,n) )
!   all elements are set equal zero
```

```

unity=0.
! setup identity matrix
DO i=1,n
  unity(i,i)=1.
ENDDO
DEALLOCATE ( unity)
.....

```

We always recommend to use to deallocation statement, since this frees space in memory. If the matrix is transferred to a function from a calling program, one can transfer the dimensionality n of that matrix with the call. Another possibility is to determine the dimensionality with the SIZE command, i.e.,

```
n=SIZE(unity,DIM=1)
```

will give the size of the rows, while using DIM=2 gives that of the columns.

Other useful Fortran 90 intrinsic functions are given by the following examples. Suppose we need to find the maximum absolute value of the column elements in a two-dimensional matrix. In Fortran 77 we would have to code something like

```

! loop over rows i
  DO i=1,n
    max_value=0.
! the loop over columns
    DO j=1,n
      IF (ABS(a(i,j)) > max_value) max_value=ABS(a(i,j))
    ENDDO
! then we store this number in a one-dimensional vector
    max_value_row(i)=max_value
  ENDDO

```

In Fortran 90 this statement is replaced by one line

```
max_value_row=MAXVAL(ABS(a), DIM=2)
```

where DIM=2 tells that we are searching among columns. It is understood that all rows i are evaluated simultaneously.

As another example, suppose we need to evaluate

$$w_i = \sum_{j=1}^n |x_i + x_j|,$$

which in Fortran 77 would involve a sum over i and j can be written in Fortran 90 using the intrinsic function SUM

```

DO i=1,n
  w(i)=SUM(ABS(x(i)+x))
ENDDO

```

Similarly, the product

$$w_i = \prod_{j=1, j \neq i}^n (x_i - x_j),$$

can be coded by aid of the PRODUCT function

```

DO i=1,n
  w(i)=PRODUCT(x(i)-x, MASK=(x /= x(i)) )
ENDDO

```

where the MASK argument prevents that the diagonal terms are included. Another way of writing the above MASK statement is through the WHERE statement. Consider a division of all matrix elements of a matrix B with a matrix C, i.e., we have

$$A = B/C$$

meaning that we are performing a division for i and j of b_{ij}/c_{ij} . If we wish to avoid division by zero we could write the above equation as

$$\text{WHERE} (c \neq 0) a=b/c$$

and that's all which is needed. The statement inside WHERE checks all matrix elements of the matrix C. Other useful functions are the dot products of two vectors, i.e.,

$$a=\text{DOT_PRODUCT} (b,c)$$

which means $a = b(1)c(1) + b(2)c(2) + \dots + b(n)c(n)$. This could also have been written as

$$a=\text{SUM} (b*c)$$

Here's an example of a module for matrix operations which employs the module mesh_variables through the USE statement.

```

MODULE matrix_manipulations
  USE mesh_variables
  IMPLICIT NONE

  CONTAINS

! perform the outer product of two vectors a and b

  FUNCTION outer_product(a,b)

    DOUBLE PRECISION, INTENT(IN) , DIMENSION(:) :: a, b
    DOUBLE PRECISION, DIMENSION(SIZE(a), SIZE(b)) :: outer_product

    outer_product = SPREAD( a,DIM=2, NCOPIES=SIZE(b) ) * &
      SPREAD( b,DIM=1, NCOPIES=SIZE(a) )

  END FUNCTION outer_product

! multiply a matrix with a vector

  FUNCTION matrix_vector_mult(a,b)

    DOUBLE PRECISION, INTENT(IN) , DIMENSION(:) :: b
    DOUBLE PRECISION, DIMENSION(:,), INTENT(IN) :: a
    DOUBLE PRECISION, DIMENSION(SIZE(b) ) :: matrix_vector_mult
    INTEGER :: i
    DO i=1, SIZE(b)
      matrix_vector_mult(i)=SUM(a(i,:)*b(:))
    ENDDO

  END FUNCTION matrix_vector_mult

END MODULE matrix_manipulations

```

For more functions, we recommend that you consult a F90 manual or see the link to F90 lectures on the web page.

A.1.7 Complex variables

Fortran 90 is uniquely suited to handle complex variables through variable definitions like COMPLEX and functions like AIMAG.

```

PROGRAM complex_example
IMPLICIT NONE
DOUBLE PRECISION, parameter :: pi = 3.141592654
COMPLEX :: b,bstar,f,arg, a
DOUBLE PRECISION :: c
INTEGER :: d

! A complex constant is written as two real numbers, separated by
! a comma and enclosed in parentheses.

a = (2.D0,-3.D0)
! If one of part has a kind, the other part must have same kind
b = (0.5D0,0.8D0)
WRITE(*,*) 'a =', a
WRITE(*,*) 'a*a =', a*a
WRITE(*,*) 'b =', b
WRITE(*,*) 'a*b =', a*b
! real part of b
c = REAL(b)
WRITE(*,*) 'real part of b =', c
! imaginary part of b
c = AIMAG(b)
WRITE(*,*) 'imaginary part of b =', c
arg = CMPLX(0.0D0,pi)
b = EXP(arg)
! complex conjugate of b
bstar = CONJG(b)
! absolute value of b
f = ABS(b)
WRITE(*,*) 'properties of b =', b,bstar,b*bstar,f

END PROGRAM complex_example

```

As another example, you could define your own complex operations through the following example.

```

MODULE complex_operations
IMPLICIT NONE
TYPE complex_variable
DOUBLE PRECISION :: real_part, complex_part
END TYPE complex_variable

CONTAINS

FUNCTION addition(a,b)

TYPE (complex_variable), INTENT(IN) :: a, b
TYPE (complex_variable) :: addition

addition = &
complex_variable(a%real_part+b%real_part,a%complex_part+b%complex_part )

```

```

END FUNCTION addition

FUNCTION subtraction(a,b)

TYPE (complex_variable), INTENT(IN) :: a, b
TYPE (complex_variable) :: subtraction

subtraction = &
complex_variable(a%real_part-b%real_part,a%complex_part-b%complex_part)

END FUNCTION subtraction

FUNCTION multiplication(a,b)

TYPE (complex_variable), INTENT(IN) :: a, b
TYPE (complex_variable) :: multiplication

multiplication%real_part = &
a%real_part*b%real_part-a%complex_part*b%complex_part

multiplication%complex_part = &
a%real_part*b%complex_part+a%complex_part*b%real_part

END FUNCTION multiplication

END MODULE complex_operations

```

Bibliography

- [1] K.A. Brueckner, *Phys. Rev.* 97 (1955) 1353.
- [2] J. Goldstone, *Proc. Roy. Soc. A* 293 (1957) 267.
- [3] H.A. Bethe, B.H. Brandow and A.G. Petschek, *Phys. Rev.* 129 (1963) 225.
- [4] R. Machleidt, *Adv. Nucl. Phys.* 19 (1989) 189.
- [5] R. Machleidt, K. Holinde and C. Elster, *Phys. Repts* 149 (1987) 1.
- [6] R. Brockmann and R. Machleidt, *Phys. Rev. C* 42 (1990) 1965.
- [7] G.E. Brown and M. Rho, *Phys. Lett. B* 82 (1979) 177.
- [8] see e.g. *Mesons in Nuclei*, eds. M. Rho and D.H. Wilkinson (North-Holland, Amsterdam, 1979).
- [9] K. Erkelenz, *Phys. Repts*, 13C (1974) 191.
- [10] G.E. Brown and A.D. Jackson, *The Nucleon-Nucleon Interaction*, (North-Holland, Amsterdam, 1976).
- [11] C. Itzykson and J.-B. Zuber, *Quantum Field Theory*, (McGraw-Hill, New-York, 1980).
- [12] R. Blankenbecler and R. Sugar, *Phys. Rev.* 142 (1966) 1051.
- [13] R.H. Thompson, *Phys. Rev. D* 1 (1970) 110.
- [14] M.J. Zuilhof and J.A. Tjon, *Phys. Rev. C* 24 (1981) 736; J. Fleischer and J.A. Tjon, *Phys. Rev. C* 21 (1980) 87.
- [15] W.R. Greenberg and E.L. Lomon, to appear in *Phys. Rev. D*.
- [16] P.G. Blunden and D.O. Riska, *Nucl. Phys. A* 536 (1992) 697.
- [17] R. Machleidt and G.Q. Li, *Phys. Repts* 242 (1994) 5.
- [18] B.L. Scott and S.A. Moskowski, *Ann. of Phys.* 14 (1961) 107.
- [19] B.L. Scott and S.A. Moskowski, *Nucl. Phys.* 29 (1962) 665.
- [20] B.D. Day, *Rev. Mod. Phys.* 39 (1967) 719.
- [21] M.I. Haftel and F. Tabakin, *Nucl. Phys. A* 158 (1970) 1.
- [22] C. Mahaux and R. Sartor, *Phys. Rev. C* 40 (1989) 1833; C. Mahaux, P.F. Bortignon, R.A. Broglia and C.H. Dasso, *Phys. Repts* 120 (1985) 1.
- [23] see e.g. S.L. Shapiro and S.A. Teukolsky, *Black Holes, White Dwarfs and Neutron Stars*, (Wiley, New York, 1983).
- [24] J.B. Hartle, *Astrophys. J.* 150 (1967) 1005. J.B. Hartle and M.W. Munn, *Astrophys. J.* 198 (1975) 467.
- [25] J.R. Oppenheimer and G.M. Volkoff, *Phys. Rev.* 55 (1939) 374.