TECHNISCHE
UNIVERSITÄT
DARMSTADT

Angular Momentum Projection and Three-Body Forces in the No-Core Shell Model

Master Thesis

Sven Binder

Darmstadt 2010

Contents

Chapter 1

Introduction

Theoretical nuclear structure physics describes atomic nuclei on the quantum level in order to obtain a theoretical understanding of the known properties of nuclei, such as binding energies, nucleon densities or electromagnetic transition amplitudes.

From the conceptual point of view, the nucleus is regarded as a self-bound system of A nucleons, with strong correlations induced by the nuclear force. For these reasons, the nuclear many-body problem defies easy theoretical treatment. On the one hand, the interaction among the nucleons is not completely determined yet, leaving grounds for some theoretical uncertainties even in this rather essential ingredient of the nuclear problem. On the other hand, the solution of the large-scale eigenvalue problem corresponding to the A-body Schrödinger equation is a challenge on its own due to computational expense, rather than formal complexity, even for small nucleon numbers A. This work will deal with both the many-body problem and the nuclear interaction within the ab initio no-core shell model (NCSM) framework.

As in almost all applications of many-body quantum mechanics the underlying Hilbert spaces have very large (infinite) dimensions and thus one is confronted with large eigenvalue problems in order to compute observables as eigenvalues of the corresponding operators. This is particularly true for nuclear structure physics, since strong correlations among the nucleons naturally demand large model spaces in order to obtain converged results. Two ways of dealing with large Hilbert spaces are considered:

First, when matrices become too large to solve the full eigenvalue problem using a direct method one can resort to the more practicable approximative solution using iterative methods like the Lanczos algorithm for instance. These methods allow approximative partial diagonalizations of the matrices, i.e. approximations of subsets of the full eigenspectrum only, but at much lower computational cost. This is highly welcome since usually only few eigenvalues are of real interest, such as the ground state energy and the first excitations, for the case of the Hamiltonian. Thus obtaining only few eigenvalues instead of the full spectrum is no real loss at all as long as the approximation method targets the right range of the spectrum. In practice Lanczos-type algorithms are frequently used because of their distinct convergence behavior to the extreme eigenvalues of a spectrum, leading to ground state approximations at comparatively low computational cost. In order to get access to interior energy eigenvalues the usability of Lanczos-type algorithms as angular momentum projection tools is investigated.

A second approach to cope with Hilbert spaces too large for practical treatment is the reduction of its dimensionality before the calculation of observables. Many truncation schemes are designed for calculations of energetic ground states by removing such states from the basis of the Hilbert space that contribute mostly to high energy eigenstates, maintaining this way the lower end of the energy eigenvalue spectrum. The Importance Truncation [18] scheme employed here, however, does not chose basis states to reject by focusing on energy arguments but rather on their capability to improve an initial approximation of a certain Hamiltonian eigenvector. Consequently, the Hilbert spaces obtained from this truncation are tailored to describe a specific Hamiltonian eigenvector, regardless of its location in the eigenspectrum. Of course, in general the partial diagonalization can be combined with the Hilbert space truncation, which works fine for the computation of energy ground states. However, the truncation will severely affect the angular momentum projection approach mentioned above since a loss of completeness of the Hilbert basis usually implies a loss of symmetries.

This truncation-induced loss of total angular momentum will cause the projection method to fail as a Krylov method and even in non-truncated Hilbert spaces the loss of total angular momentum due to numerical errors demands a careful treatment.

Regarding the nuclear interaction, there is evidence that three-nucleon forces must be included in nuclear structure calculations in addition to the traditional two-nucleon interactions. Unlike the electromagnetic interaction that governs the physics in the atomic shell, the interactions among the nucleons are more complicated and far less understood. It is known that the nuclear force emerges as residual interaction between nucleons in the low-energy limit of quantum chromodynamics (QCD). Consequently, one should ultimately be able to derive the nuclear force from QCD. However, attempts to derive it directly from QCD have failed due to its non-perturbative character at low energies. At this point the chiral effective field theory (χEFT), in which the chiral symmetry is spontaneously broken and where the pions then emerge as the corresponding Goldstone bosons, helps to connect nuclear structure with QCD. The chiral perturbative expansion naturally leads to NNN interactions which will be employed in order to investigate their relevance in nuclear structure calculations.

This work is organized as follows. Chapter 2 introduces the No-Core-Shell Model and Chapter 3 the Importance Truncation. The Lanczos algorithm is reviewed in Chapter 4. Since the Lanczos algorithm is regarded as being numerically instable, Chapter 5 focuses on numerical issues and their treatment. How the algorithm can be employed as an angular momentum projection tool is illustrated in Chapter 6. Independent from the angular momentum discussion, in Chapter 7, 8 electromagnetic form factors and transition probabilities are calculated within the IT-NCSM framework. Three-body forces are discussed in Chapter 9, where the foundations are set to employ available three-body matrix elements in NCSM computations, and where general Slater-Condon rules are derived.

Chapter 2

No-Core Shell Model

The no-core shell model (NCSM) is a more fundamental approach than the traditional nuclear shell model in the sense that one starts with the nucleonic degrees of freedom and treats the nucleus as a full A-body problem, interacting via some basic nucleonic interaction

$$H_A = \frac{1}{A}\sum_{i<j}^{A}\frac{(\boldsymbol{p}_i-\boldsymbol{p}_j)^2}{2m} + \sum_{i<j}^{A} V_{ij}^{\mathrm{NN}} + \sum_{i<j<k}^{A} V_{ijk}^{\mathrm{NNN}}. \tag{2.1}$$

It is convenient to add a center-of-mass harmonic oscillator potential $H_{CM} = \frac{1}{2Am}\boldsymbol{P}_{CM}^2 + \frac{1}{2}Am\Omega^2\boldsymbol{R}$ ($\boldsymbol{R} = \sum_i \boldsymbol{r}_i$) to (2.1) that does not influence the actually interesting intrinsic properties of the nucleus but constrains the center-of-mass movement.

Although arbitrary basis sets could be employed, the use of the harmonic oscillator basis has crucial advantages. The harmonic oscillator single-particle wave functions

$$\phi_{nlm}(\boldsymbol{r}) = g_{nl}(r)\; Y_{lm}(\theta,\varphi) \tag{2.2}$$

are given in terms of the solutions $g_{nl}(r)$ of the radial one-particle Schroedinger equation with the harmonic potential $U(r) = \frac{1}{2}m_N\omega^2 r^2$,

$$\left[\frac{-\hbar^2}{2m_N r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}\right) + \frac{l(l+1)\hbar^2}{2m_N r^2} + \frac{1}{2}m_N\omega^2 r^2\right] g_{nl}(r) = \epsilon_{nl} g_{nl}(r), \tag{2.3}$$

and the spherical harmonics, carrying the angular dependencies. The con-

ventions for the quantum numbers used here are according to [22]

$$\begin{aligned} n &= 0, 1, 2, \ldots && (2.4)\\ 2n + l &= 0, 1, 2, \ldots && (2.5)\\ \epsilon_{nl} &= (2n + l + \tfrac{3}{2})\hbar\omega. && (2.6) \end{aligned}$$

The $g_{nl}(r)$ exist as analytical solutions of (2.3) in terms of associated Laguerre polynomials $L_n^{(k)}(x)$ [22]

$$g_{nl}(r) = \sqrt{\frac{2n!}{b^3\Gamma(n + l + \frac{3}{2})}} \left(\frac{r}{b}\right)^l e^{-r^2/2b^2} L_n^{(l+\frac{1}{2})}(r^2/b^2) \qquad (2.7)$$

where $b = \sqrt{\hbar/m_N\omega}$ is the oscillator length. So a single-particle harmonic oscillator wave function including spin and isospin degrees of freedom is given by

$$\begin{aligned} \phi_{nlm_lm_sm_t}(\boldsymbol{x}, \boldsymbol{\sigma}, \boldsymbol{\tau}) &= g_{nl}(r)\ Y_{lm_l}(\theta, \varphi)\ \chi_{m_s}^{(s=1/2)}(\boldsymbol{\sigma})\ \chi_{m_t}^{(\tau=1/2)}(\boldsymbol{\tau}) \\ &= \langle \boldsymbol{x}, \boldsymbol{\sigma}, \boldsymbol{\tau} | nlm_lm_sm_t \rangle. \qquad (2.8) \end{aligned}$$

This analyticity comes at hand at various occasions: In Chapter 7 where electromagnetic transition amplitudes are calculated, analytical expressions for the the radial integrals of g_{nl} that enter the calculations can be found. Furthermore, coordinate transformations of oscillator functions result in finite, rather than infinite, expansions

$$\phi'_{nlm}(\boldsymbol{r}') = \sum_{\mathcal{NLM}}^{<\infty} c_{\mathcal{NLM}}\ \phi_{\mathcal{NLM}}(\boldsymbol{r}) \qquad (2.9)$$

in the oscillator basis in the original coordinates, which is exploited in Chapter 9. Regarding translationally invariance, the harmonic oscillator basis is the basis of choice since it - when appropriately truncated by a chosen maximum excitation energy - allows the use of single-particle coordinates while preserving translational invariance.

Because of the conservation of total angular momentum it is more convenient to employ single-particle states that have good total angular momentum that can be obtained from (2.8) by coupling the orbital angular momentum and the spin

$$\begin{aligned} \phi_{njm_jm_t}(\boldsymbol{x}, \boldsymbol{\sigma}, \boldsymbol{\tau}) &= \sum_{m_lm_s} \begin{pmatrix} l & \frac{1}{2} & j \\ m_l & m_s & m_j \end{pmatrix}_{CG} \phi_{nlm_lm_sm_t}(\boldsymbol{x}, \boldsymbol{\sigma}, \boldsymbol{\tau}) \\ &= \langle \boldsymbol{x}, \boldsymbol{\sigma}, \boldsymbol{\tau} | n(ls)jm_jm_t \rangle. \qquad (2.10) \end{aligned}$$

Chapter 3

Importance Truncation

Truncation schemes are naturally used in almost all applications of quantum theory. Usually an energy argument of some kind is used to truncate the basis such that the low-energy states are maintained. In contrary to truncations that are insensitive to the actual problem that is to be solved, the Importance Truncation [18, 19] is a method that takes the actual problem into account in more detail.

By the means of Importance Truncation such states shall be removed from the basis $|\Phi_\nu\rangle$ of the full model space $\mathcal{M}_{\text{full}}$ that are of minor importance for the description of a specific Hamiltonian eigenstate, called the target state. The solution of the Hamiltonian eigenvalue problem in this truncated model space shall at least provide a reasonable approximation for the target state.

To determine the importance of a basis state one needs a pre-computed zeroth-order approximation of the target state, the reference state $|\Psi_{\text{ref}}\rangle$. This reference state is obtained from a diagonalization in a subspace of the full model space, the reference space $\mathcal{M}_{\text{ref}}$. Other basis states of $\mathcal{M}_{\text{full}}$ that do not lie in the reference space will lead to corrections of $|\Psi_{\text{ref}}\rangle$. It is reasonable to characterize the importance of a basis vector outside the reference space through the amount of correction it introduces to $|\Psi_{\text{ref}}\rangle$. Estimates of these corrections can be obtained using perturbation theory, even without explicit knowledge of the unperturbed and perturbation part of the Hamiltonian. In the language of perturbation theory, $|\Psi_{\text{ref}}\rangle$ corresponds to the unperturbed state, that is an eigenstate of the unperturbed Hamiltonian H_0, for which the corrections are computed. Therefore, the unperturbed Hamiltonian can

formally be defined as the operator which obeys the eigenvalue equation

$$H_0|\Psi_{\text{ref}}\rangle = \epsilon_{\text{ref}}|\Psi_{\text{ref}}\rangle \tag{3.1}$$

with $\epsilon_{\text{ref}} = \langle\Psi_{\text{ref}}|H|\Psi_{\text{ref}}\rangle$ where H is the full Hamiltonian. Following [18], the unperturbed Hamiltonian can be expressed as

$$H_0 = \epsilon_{\text{ref}}|\Psi_{\text{ref}}\rangle\langle\Psi_{\text{ref}}| + \sum_{\nu\notin\mathcal{M}_{\text{ref}}} \epsilon_\nu|\Phi_\nu\rangle\langle\Phi_\nu|, \tag{3.2}$$

where the energies ϵ_ν have to be chosen in some way in order to fix the operator. Then the perturbation part W of the full Hamiltonian is given by

$$W = H - H_0 \tag{3.3}$$

and with this the first-order correction to $|\Psi_{\text{ref}}\rangle$ from many-body perturbation theory reads

$$|\Psi^{(1)}\rangle = -\sum_{\nu\notin\mathcal{M}_{\text{ref}}} \frac{\langle\Phi_\nu|W|\Psi_{\text{ref}}\rangle}{\epsilon_\nu - \epsilon_{\text{ref}}}|\Phi_\nu\rangle. \tag{3.4}$$

Since H_0 does not connect states from $\mathcal{M}_{\text{ref}}$ with basis states from the outside, instead of W the full Hamiltonian can be inserted, yielding

$$|\Psi^{(1)}\rangle = -\sum_{\nu\notin\mathcal{M}_{\text{ref}}} \frac{\langle\Phi_\nu|H|\Psi_{\text{ref}}\rangle}{\epsilon_\nu - \epsilon_{\text{ref}}}|\Phi_\nu\rangle. \tag{3.5}$$

In this way, an importance measure κ_ν for a basis state $|\Phi_\nu\rangle$, which solely depends on H, $|\Psi_{\text{ref}}\rangle$ and $|\Phi_\nu\rangle$ is defined by

$$\kappa_\nu = -\frac{\langle\Phi_\nu|H|\Psi_{\text{ref}}\rangle}{\epsilon_\nu - \epsilon_{\text{ref}}}. \tag{3.6}$$

In practice, all basis states with importance measure smaller than a threshold value, $\kappa_\nu < \kappa_{\text{min}}$, are discarded from the basis and the eigenvalue problem of H is solved in the remaining model space.

Since the Importance Truncation is clearly a variational approach, changes in κ_{min} will result in more or less continuous departures of the results from the exact values. Therefore, one can perform a series of calculations with varying κ_{min} and try to extrapolate the results to the $\kappa_{\text{min}} = 0$ (the exact) case.

Chapter 4

The Simple Lanczos Algorithm

"Could anything be more simple?"
Beresford Parlett [10].

The Lanczos process was introduced by Cornelius Lanczos in 1950 [1]. In his paper he presented the Method of Minimized Iterations as a process that expands each eigenvector of a symmetric $n \times n$ matrix into a convergent series with at most n terms, so that after n steps the exact results are obtained. In comparison, from the power method, the simplest method to address the eigenvector to the dominant eigenvalue, one usually does not get exact results for finite iterations. Although this is a nice feature of the Lanczos process this is not the way it has been used in the past nor the way it is used in computations today.

At first the Lanczos process has been used to reduce a symmetric matrix A to tridiagonal form T. Using exact arithmetics, for a $n \times n$ matrix A this reduction can be achieved after n steps. One ends up with an orthonormal matrix $Q = (q_1, ..., q_n)$ that transforms A into a tridiagonal representation

$$Q^T A Q = T. \tag{4.1}$$

Because of the orthonormality of Q ($Q^T = Q^{-1}$) this transformation is a similarity transformation and so A and T have the same set of eigenpairs. Computing eigenpairs from tridiagonal matrices is a comparatively easy task and so the Lanczos algorithm together with the QR method for finding the eigenvalues of tridiagonal matrices have been used as a direct method for the

solution of eigenvalue problems. In theory the algorithm has some remarkable properties such as simplicity and memory efficiency in computations.

However, in practice almost all of these properties vanish due to inevitable rounding errors and the method can not reliably be used as a direct method for larger problems unless very expensive measures are employed. For that reason more numerically stable direct methods such as Householder transformations are used instead which have been developed shortly after the Lanczos process. Although the Lanczos process fails as a direct method it is still very useful when used as an iterative one.

Instead of performing all n steps, the Lanczos process can be halted after $k < n$ steps and T_k, the $k \times k$ submatrix of T, can be diagonalized. Lanczos himself was aware that usually after a few iterations the extreme eigenvalues of T_k already provide good approximations of the extreme eigenvalues of the full tridiagonal representation T (and so of A). A first detailed analysis of the efficiency of the Lanczos process can be found in C.C. Paige's doctoral thesis [5]. Therefore, the Lanczos algorithm is the method of choice when only a few extreme eigenvalues are needed. It is well-suited for large sparse matrices A since in every iteration step A only occurs in one matrix-vector multiplication so that the sparsity of A can easily be exploited.

But even when used as an iterative method the Lanczos algorithm suffers from numerical errors. There are many variations of the algorithm that represent different ways to deal with the effects of finite precision arithmetics. The most common variants will be presented in the following and will be used and compared in the calculations in Section 5.2.

There are two major ways to look at and to derive the Lanczos algorithm. The first one

$$\text{Lanczos} = \text{Krylov} + \text{Rayleigh-Ritz} \tag{4.2}$$

reveals the physical significance of the method and the second one

$$\text{Lanczos} = \text{Reduction to Tridiagonal Form} \tag{4.3}$$

is a fast way to derive the algorithm. Since on the one hand the first way is essential to understand the physics and on the other hand the second approach is usually used in the mathematical treatment of the algorithm both ways are presented in the following.

4.1 The Krylov Space Approach

The Lanczos method is an orthogonal projection method onto a Krylov space $\mathcal{K}_m$ [2, 8]. Therefore, some basic facts about projection methods and Krylov spaces will be compiled below. In the nuclear context, one will be able to identify angular momentum subspaces of Hilbert spaces with such Krylov spaces.

The algorithm iteratively generates an orthonormal basis $Q = (q_1, \ldots, q_n)$ of the Krylov space. Then the matrix $A_m = Q_m A Q_m$ is the projection of A onto this Krylov space. From the matrix A_m approximations of the eigenvalues of A can be extracted.

4.1.1 Projection Methods

In order to find the eigenpairs (ν_i, ξ_i) of a linear operator

$$A : \mathcal{V} \to \mathcal{V}$$

one usually has to resort to the full vector space $\mathcal{V}$, where $\mathcal{V} \equiv \mathbb{C}^n$ if $A \in \mathbb{C}^{n\times n}$ or, for some quantum operator $\hat{A}$, $\mathcal{V} \equiv \mathcal{H}$, the underlying Hilbert space spanned by the eigenstates of A.

A projection method provides approximations $(\tilde{\nu}_i, \tilde{\xi}_i)$ of the exact eigenpairs (ν_i, ξ_i) from a subspace $\mathcal{K} \subseteq \mathcal{V}$. Therefore, since

$$\tilde{\nu}_i \in \mathcal{K} \quad (\tilde{\xi}_i \in \mathbb{C}),$$

$\mathcal{K}$ is referred to as the subspace of candidate approximants or the right subspace.

To find an approximation $\tilde{\nu}$ from $\mathcal{K}$ (the index i will be dropped in the following) one has to impose $m = \dim \mathcal{K}$ constraints. Usually, and so in this case, one demands the residual vector

$$s := A\tilde{\nu} - \tilde{\xi}\tilde{\nu} \tag{4.4}$$

to be orthogonal to m linear independent vectors $l_1, \ldots, l_m$. These vectors define another subspace

$$\mathcal{L} = \operatorname{span}\{l_1, \ldots, l_m\}, \tag{4.5}$$

called the subspace of constraints or the left subspace.

In general $\mathcal{L}$ may be equal to $\mathcal{K}$ but it does not have to. Thus two classes of projection methods arise, the orthogonal ($\mathcal{L} = \mathcal{K}$) and the oblique ($\mathcal{L} \neq \mathcal{K}$).

The Lanczos procedure is an orthogonal projection method and so in the following $\mathcal{L}$ will be set equal to $\mathcal{K}$ and the orthogonality condition mentioned above becomes

$$A\tilde{\nu} - \tilde{\xi}\tilde{\nu} \perp \mathcal{K}. \tag{4.6}$$

The constraints (4.6) are called Galerkin conditions.

$\mathcal{K}$ has to be specified by some basis that, because of the finite dimension of $\mathcal{K}$, can be assumed to be orthonormal. Let $\{k_1, \ldots, k_m\}$ be this basis and let

$$K := \Big(k_1, \ldots, k_m\Big) \tag{4.7}$$

denote the matrix build from the column vectors k_i. The approximative problem can be solved by translating it into the basis $\{k_i\}$.

Set

$$\begin{aligned} \tilde{\nu} &= Ky \qquad (4.8)\\ (\dim \tilde{\nu} &= \dim \mathcal{V}, \quad \dim y = \dim \mathcal{K}). \end{aligned}$$

With this the Galerkin condition becomes

$$\langle AKy - \tilde{\xi}Ky, u\rangle = 0 \quad \forall u \in \mathcal{K}. \tag{4.9}$$

It is $u \in \mathcal{K}$ and so u can be expanded in $\{k_i\}$, $u = \sum_{i=1}^{m} c_i k_i$, and therefore

$$\sum_{i=1}^{m} c_i \langle AKy - \tilde{\xi}Ky, k_j\rangle = 0, \quad j = 1, \ldots, m. \tag{4.10}$$

This equation holds for arbitrary $(c_1, \ldots, c_m) \in \mathbb{C}^m$ and must therefore be true for each basis vector k_i

$$\begin{aligned} \langle AKy - \tilde{\xi}Ky, k_j\rangle &= 0, \quad j = 1, \ldots, m \qquad (4.11)\\ (AKy - \tilde{\xi}Ky)K &= 0. \qquad (4.12) \end{aligned}$$

Using the orthonormality of K one gets

$$\begin{aligned} K^T(AKy - \tilde{\xi}Ky)KK^T &= K^T 0 K^T && (4.13)\\ K^T AKy - \tilde{\xi}y &= 0 && (4.14)\\ K^T AKy &= \tilde{\xi}y. && (4.15) \end{aligned}$$

$A_m(K) := K^T AK$ is the constraint or the projection of A onto the space $\mathcal{K}$, represented in the orthonormal basis $\{k_i\}$. Therefore, demanding the residual vector of the approximative eigenpair to be orthogonal to $\mathcal{K}$, the subspace of approximants, leads to the eigenvalue problem

$$A_m(K)y = \tilde{\xi}y. \tag{4.16}$$

Eigenvalues $\tilde{\xi}$ of $A_m(K)$ already are the desired subspace approximations of A's exact eigenvalues and from the eigenvectors y of $A_m(K)$ one gets the approximations of the eigenvectors of A via

$$\tilde{\nu} = Ky. \tag{4.17}$$

This method, computing $A_m(K) = K^T AK$ from an orthonormal basis $\{k_i\}$ and computing approximative eigenpairs $(\tilde{\nu}, \tilde{\xi})$ from $A_m(K)$ is called the Rayleigh-Ritz procedure, which is summarized in Table 4.1 and consequently $(\tilde{\nu}, \tilde{\xi})$ are called Ritz vectors and Ritz values (resp. Ritz pairs).

R :	The Rayleigh-Ritz Procedure
R(1):	Compute an orthonormal basis $\{k_i\}_{i=1,\ldots,m}$ of the subspace $\mathcal{K}$. Let $K = (k_1, \ldots, k_m)$.
R(2):	Compute $A_m = K^T AK$.
R(3):	Compute the eigenvalues of A_m and select the k desired ones $\tilde{\xi}_i, i = 1, \ldots, k$, where $k \leq m$
R(4):	Compute the eigenvectors $y_i, i = 1, \ldots, k$, of A_m associated with $\tilde{\xi}_i, i = 1, \ldots, k$, and the corresponding approximate eigenvectors of A, $\tilde{\nu}_i = Vy_i, i = 1, \ldots, k$

Table 4.1: The Rayleigh-Ritz procedure [2]

.

Because of the orthonormality of K its inverse is given by its transpose

$$K^T = K^{-1} \tag{4.18}$$

and so

$$A_m(K) = K^{-1}AK. \tag{4.19}$$

Therefore, the transformation $A \to A_m(K)$ resembles an similarity transformation from which is known that it preserves the eigenvalues of A. $A \to A_m(K)$ is of course not a similarity transformation since usually dim $K <$ dim A but this resemblance indicates why one gets useful approximations from $A_m(K)$. Indeed, the Rayleigh-Ritz approximations are optimal for the information provided by the subspace $\mathcal{K}$:

Proposition 4.1.1
The minimum of $||AK - KR||_2$ over all $k \times k$ symmetric matrices R is attained by $R = A_m(K)$ [4].

An important feature of orthogonal projection methods is the fact that if $\mathcal{K}$ is an invariant subspace under A the approximative eigenpairs obtained from $A_m(K)$ are exact.

Definition 4.1.1
If

$$Ak \in \mathcal{K}, \ \forall k \in \mathcal{K}$$

then $\mathcal{K}$ is called an invariant subspace under A.

Proposition 4.1.2
If $\mathcal{K}$ is invariant under A then every approximate eigenpair obtained from the orthogonal projection method onto $\mathcal{K}$ is exact [2].

Instead of a mathematical proof (which is eminently short but of no interest here) this statement is motivated physically.

If total angular momentum is conserved, a Hilbert space $\mathcal{H}$ decomposes into a direct sum of subspaces $\mathcal{J}^{(i)}$ that are invariant under the total angular momentum operator J^2

$$\mathcal{H} = \mathcal{J}^{(0)} \oplus \cdots \oplus \mathcal{J}^{(n)} \oplus \ldots \ . \tag{4.20}$$

Let $A = H$ be a Hamilton operator that, due to the conservation of total angular momentum, commutes with J^2

$$[H, J^2] = 0. \tag{4.21}$$

Then the subspaces $\mathcal{J}^{(j)}$ are also invariant under H and there is a simultaneous eigenbasis $|E_k^{(j)}\rangle$ for H and J^2 with

$$\begin{aligned} H|E_k^{(j)}\rangle &= E_k^{(j)}|E_k^{(j)}\rangle & (4.22)\\ J^2|E_k^{(j)}\rangle &= j(j+1)|E_k^{(j)}\rangle. & (4.23) \end{aligned}$$

An arbitrary state $|\psi\rangle$ can be expanded in this basis

$$\begin{aligned} |\psi\rangle &= \sum_j \sum_{k=1}^{k_j} \langle E_k^{(j)}|\psi\rangle |E_k^{(j)}\rangle \\ |\psi\rangle &= \sum_j \sum_{k=1}^{k_j} \psi_k^{(j)} |E_k^{(j)}\rangle, & (4.24) \end{aligned}$$

where k_j denotes the number of basis states with $J^2|E_k^{(j)}\rangle = j(j+1)|E_k^{(j)}\rangle$. Set $\mathcal{K} \equiv \mathcal{J}^{(j)}$ and let $|\psi^{(j)}\rangle \in \mathcal{K}$ be an approximation of an eigenvector of H. Then the Galerkin condition (4.6) becomes

$$H|\psi^{(j)}\rangle - \tilde{\xi}|\psi^{(j)}\rangle \perp \mathcal{K}. \tag{4.25}$$

Since $|\psi^{(j)}\rangle$ is a state with well-defined total angular momentum $j(j+1)$ its expansion reads

$$|\psi^{(j)}\rangle = \sum_{k=1}^{k_j} \psi_k^{(j)} |E_k^{(j)}\rangle. \tag{4.26}$$

When the Hamiltonian acts on it

$$H|\psi^{(j)}\rangle = \sum_{k=1}^{k_j} \psi_k^{(j)} E_k^{(j)} |E_k^{(j)}\rangle \tag{4.27}$$

one just gets another linear combination of the basis states that built $|\psi^{(j)}\rangle$ and so

$$H|\psi^{(j)}\rangle \in \mathcal{K}. \tag{4.28}$$

Obviously

$$\tilde{\xi}|\psi^{(j)}\rangle \in \mathcal{K}, \tag{4.29}$$

and, therefore, the Galerkin condition (4.25) can not be satisfied for $H|\psi^{(j)}\rangle \neq \tilde{\xi}|\psi^{(j)}\rangle$. For this reason it is $H|\psi^{(j)}\rangle = \tilde{\xi}|\psi^{(j)}\rangle$ which means that the approximative eigenvalue $\tilde{\xi}$ is actually exact. This is because $\mathcal{K}$ is the full invariant subspace of H. If $\tilde{\mathcal{K}} \subset \mathcal{K}$ then also $H|\psi^{(j)}\rangle \in \tilde{\mathcal{K}}$ and $\tilde{\xi}|\psi^{(j)}\rangle \in \tilde{\mathcal{K}}$ but (4.25) may be satisfied.

4.1.2 Krylov Spaces

According to (4.2)

$$\text{Lanczos} = \text{Krylov} + \text{Rayleigh-Ritz} \tag{4.30}$$

Krylov spaces play an important role in the Lanczos algorithm.

Definition 4.1.2
The space $\mathcal{K}_m(A, q)$ spanned by q and the first $m-1$ iterates of the simple power method applied to a matrix A and vector q,

$$\mathcal{K}_m(A, q) = span\left\{q, Aq, \ldots, A^{m-1}q\right\},$$

is called Krylov space of dimension m with respect to A and q. Consequently the vectors $A^k q$ are referred to as Krylov vectors.

The Lanczos and Arnoldi algorithms are orthogonal projection methods onto Krylov spaces, i.e. they search for approximative solutions within these subspaces. At a first glance this could easily be achieved by calculating the Krylov vectors by iterative application of A onto q. After an orthonormalization of the Krylov vectors the Rayleigh-Ritz procedure could be applied to obtain approximations from the Krylov space. However, although this works in principle, this approach is not practicable. This is because the Krylov vectors are the iterates of the power method which, of course, converges to the eigenvector of the dominant eigenvalue of A

$$\lim_{k\to\infty} A^k q = \nu_{\text{dom}}. \tag{4.31}$$

Because of this convergence that comes along with an increase of linear dependence of the generated Krylov vectors a reliable orthonormalization can not be performed numerically since severe cancellations take place. Avoiding this linear dependence while generating and orthonormalizing the Krylov vectors will be the main task of the Arnoldi and Lanczos algorithm.

There is obviously a natural connection between Krylov spaces and polynomials since every $k \in \mathcal{K}_m(A, q)$ can be expressed as a linear combination of Krylov vectors

$$k = \sum_{i=0}^{m-1} c_i A^i q, \quad k \in \mathcal{K}_m(A, q),\ c_i \in \mathbb{C}. \tag{4.32}$$

Proposition 4.1.3
The Krylov space $\mathcal{K}_m$ is the subspace of all vectors in $\mathbb{C}^n$ which can be written as $x = p(A)q$, where $p \in \prod_{m-1}$ is a polynomial of degree not exceeding $m-1$ [2].

Since according to Proposition (4.1.2) Rayleigh-Ritz approximations of eigenvalues from invariant subspaces turn out to be exact there is a particular interest in these subspaces. One interesting statement can be made in terms of the degree of the minimal polynomial of q.

Definition 4.1.3

(1) The minimal polynomial of a vector q is the nonzero monic polynomial p of lowest degree such that $p(A)q = 0$.

(2) The degree $\mu = \mu(A, q)$ of the minimal polynomial of q is called the grade of q with respect to A.

Proposition 4.1.4
Let μ be the degree of the minimal polynomial of q. Then $\mathcal{K}_\mu$ is invariant under A and $\mathcal{K}_m = \mathcal{K}_\mu$ for all $m \geq \mu$ [2].

Krylov spaces of Hermitian matrices have a nice feature when one calculates an orthonormal basis of it. In general, every finite set of vectors can be orthonormalized by the well-known Gram-Schmidt procedure. The effort for this rises fast with a growing number of vectors to be orthonormalized.

However, for Krylov vectors of Hermitian matrices the situation simplifies drastically.

Let $Q_m = (q_1, \ldots, q_m)$ be orthonormal with column vectors q_i that arise from the Gram-Schmidt procedure applied to the set of Krylov vectors $\{q_1, Aq_1, \ldots, A^{m-1}q_1\}$.

Consider $\mathrm{span}\{q_1, q_2, q_3, Aq_3\}$. It is $q_3 \in \mathcal{K}_3(q_1, A)$ and so

$$\begin{aligned} \mathrm{span}\{q_1, q_2, q_3, Aq_3\} &= \mathrm{span}\{q_1, q_2, q_3, A(\gamma_2 A^2 + \gamma_1 A + \gamma_0 \mathbb{1})q_1\} \\ &= \mathrm{span}\{q_1, q_2, q_3, A^3 q_1\} \\ &= \mathcal{K}_4(q_1, A). \end{aligned} \tag{4.33}$$

So the space spanned by q_1, q_2, q_3 and Aq_3 is identical to the Krylov space $\mathcal{K}_4(q_1, A)$. The advantage in using Aq_3 instead of $A^3 q_1$ to span the space lies in the fact that for Hermitian matrices Aq_3 is already orthogonal to q_1:

$$\langle Aq_3, q_1 \rangle = \langle q_3, A^\dagger q_1 \rangle = \langle q_3, Aq_1 \rangle. \tag{4.34}$$

Since $Aq_1 \in \mathcal{K}_2(A, q_1)$ it takes the general form

$$Aq_1 = c_0 q_1 + c_1 q_2 \tag{4.35}$$

and one gets

$$\langle Aq_3, q_1 \rangle = \langle q_3, c_0 q_1 + c_1 q_2 \rangle. \tag{4.36}$$

By definition q_1, q_2, q_3 are already orthogonal and for that reason $\langle Aq_3, q_1 \rangle = 0$.

This can be generalized to the case of more vectors: Instead of orthonormalizing a given set of m Krylov vectors for a subsequent Rayleigh-Ritz procedure one can in the case of Hermitian matrices also generate recursively an orthonormal basis where the new generated Krylov vectors only have to be orthonormalized against the two precedent basis vectors (Table 4.2). This is exactly what the Lanczos algorithm does.

ONB :		Generation of an Orthonormal Krylov Basis in the Case of Hermitian Matrices
ONB(1):	Start :	Choose q_1 with $\|\|q_1\|\| = 1$
		Set $q_0 = q_{-1} = 0$
	Iterate :	$i = 1, \ldots, m-1$
ONB(2):		$p = Aq_i$
ONB(3):		$q_{i+1} = p - \langle p, q_i \rangle q_i - \langle p, q_{i-1} \rangle q_{i-1}$

Table 4.2: Generation of an orthonormal Krylov basis in the case of an Hermitian matrix A.

4.1.3 The Arnoldi Algorithm

The (Hermitian) Lanczos algorithm emerges from the Arnoldi algorithm as a straightforward simplification when Arnoldi is applied to a Hermitian matrix. The basic ideas are best understood by studying the Arnoldi algorithm for (non-Hermitian) matrices and so it will be discussed before constraining to Hermitian matrices and the Lanczos algorithm.

According to Chapter 4.1.1, in order to calculate the projection of eigenpairs onto an subspace $\mathcal{K}$ one requires an orthonormal basis of this subspace. From now on this subspace will be a Krylov space $\mathcal{K}_m(A, q)$.

As mentioned in Chapter 4.1.2 it is not recommended to orthonormalize the set of Krylov vectors directly via the Gram-Schmidt procedure or other methods because of their increasing linear dependence and, at least in the case of Hermitian matrices, because of the possibility to simplify the orthonormalization process.

A central insight of Chapter 4.1.2 is that if $Q_m = (q_1, \ldots, q_m)$ is an orthonormal basis of the Krylov space $\mathcal{K}_m(A, q_1)$ and if $K_m(q_1, A) = (q_1, Aq_1, \ldots,$

$A^{m-1}q_1$) is the set of Krylov vectors then

$$\begin{aligned} \mathcal{K}_{m+1}(q_1, A) &\overset{\text{Def. (4.1.2)}}{=} \text{span}\{K_m, A^m q_1\} \\ &\overset{(4.33)}{=} \text{span}\{K_m, Aq_m\} \\ &= \text{span}\{K_m, q_{m+1}\}, \end{aligned} \tag{4.37}$$

where q_{m+1} denotes the vector Aq_m orthonormalized against all $q_i, i = 1, \ldots, m$ (in the case of Hermitian matrices only orthonormalizing against q_m and q_{m-1} would be necessary). The fact (4.37) is exploited by the Arnoldi-Algorithm, summarized in Table 4.3.

A :	The Arnoldi Algorithm		
A(1):	Start	:	Choose q_1 with $\|\|q_1\|\| = 1$
	Iterate :		$j = 1, \ldots, m$:
A(2):			$h_{ij} = \langle q_i, Aq_j \rangle, \quad i = 1, \ldots, j$
A(3):			$w_j = Aq_j - \sum_{i=1}^{j} h_{ij} q_i$
A(4):			$h_{j+1,j} = \|\|w_j\|\|$
A(5):			$q_{j+1} = w_j / h_{j+1,j}$

Table 4.3: The Arnoldi algorithm.

In steps $A(2)$ and $A(3)$ one recognizes the classical Gram-Schmidt procedure, where the vector Aq_j is orthogonalized to all precedent q_i. Steps $A(4)$ and $A(5)$ are for normalization of the newly generated basis vector.

Proposition 4.1.5
The vectors $q_1, \ldots, q_m$ form an orthonormal basis of the subspace $\mathcal{K}_m = span\ \{q_1, Aq_1, \ldots, A^{m-1}q_1\}$ [2].

After m steps the orthonormal matrix Q_m is generated and in order to calculate approximate eigenvalues, the Rayleigh-Ritz procedure now demands the transformation of A to $A_m(Q_m)$ via

$$A_m(Q_m) = Q^\dagger A Q. \tag{4.38}$$

As a matter a of fact these matrix multiplications do not have to be performed since the matrix elements of A_m arises as byproducts of the Arnoldi algorithm. Indeed, these matrix elements are given by the overlaps h_{ij} calculated in step $A(2)$ and the norms $h_{j+1,j}$, see $A(4)$. This is easy to understand, because from

$$A(4): \quad w_j = Aq_j - \sum_{i=1}^{j} h_{ij} q_i \tag{4.39}$$

and

$$A(6): \quad q_{j+1} = w_j / h_{j+1,j} \tag{4.40}$$

follow by inserting (4.40) into (4.39) the equations

$$Aq_j = \sum_{i=1}^{j+1} h_{ij} q_i, \qquad j = 1, \ldots, m. \tag{4.41}$$

This system of equations takes in matrix representation the form

$$AQ_m = Q_m H_m + h_{m+1,m} q_{m+1} e_m^\dagger \tag{4.42}$$

with the upper Hessenberg matrix

$$H_m = \begin{pmatrix} h_{11} & h_{12} & h_{13} & \ldots & h_{1m} \\ h_{21} & h_{22} & h_{23} & \ldots & h_{2m} \\ 0 & h_{32} & h_{33} & \ldots & h_{3m} \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & \ldots & 0 & h_{m,m-1} & h_{mm} \end{pmatrix} \tag{4.43}$$

and the Hermitian conjugate of the m-th unit vector $e_m^\dagger$. The last term in (4.42), $h_{m+1,m} v_{m+1} e_m^\dagger$, takes into account that in (4.41) the element $h_{m+1,m}$ appears under the sum that however is not incorporated in H_m. This term vanishes for $h_{m+1,m} = 0$ as it would be for instance in the case dim $Q =$ dim A (the common similarity transformation) so that then one would have $AQ = QH$.

Multiplication of (4.42) with $Q_m^\dagger$ yields

$$Q_m^\dagger A Q_m = H_m \tag{4.44}$$

since q_{m+1} is orthogonal to the columns of $Q_m^\dagger$. Therefore, according to (4.44), the matrix H_m build from the coefficients h_{ij} of the Arnoldi algorithm is the projection of A to the subspace $\text{span}\{Q\} = \mathcal{K}_m(q_1, A)$.

Proposition 4.1.6
Denote by Q_m the $n \times m$ matrix with column vectors $q_1, \ldots, q_m$ and by H_m the $m \times m$ Hessenberg matrix whose nonzero entries are defined by the algorithm. Then the following relations hold:

$$AH_m = Q_m H_m + h_{m+1,m} q_{m+1} e_m^\dagger \tag{4.45}$$

$$Q_m^\dagger A Q_m = H_m \tag{4.46}$$

[2].

4.1.4 The Lanczos Algorithm

From (4.43) and (4.44) follows that $A_m = H_m$ is an upper $m \times m$ Hessenberg matrix with matrix elements h_{ij} calculated in steps $A(2)$ and $A(4)$ of the Arnoldi algorithm Table 4.3. Since

$$h_{ij} = \langle Aq_j, q_i \rangle \tag{4.47}$$

for a Hermitian matrix A follows

$$h_{ij} = \langle q_j, A^\dagger q_i \rangle = \langle q_j, Aq_i \rangle \tag{4.48}$$

and thus

$$h_{ij}^* = \langle Aq_i, q_j \rangle = h_{ji} \tag{4.49}$$

or, more specifically for $h_{ij} \in \mathbb{R}$,

$$h_{ij} = h_{ji}. \tag{4.50}$$

For an upper Hessenberg matrix it is $h_{ij} = 0$ for $j < i - 1$ and therefore the projection A_m of a Hermitian matrix A takes a tridiagonal form because of (4.50). This is of course highly welcome since there are powerful algorithms for solving eigenproblems of tridiagonal matrices, the QR algorithm for instance.

Proposition 4.1.7
Assume that Arnoldi's method is applied to a Hermitian matrix A. Then the coefficients h_{ij} generated by the algorithm are real and such that

$$h_{ij} = 0, \text{ for } 1 \leq i < j-1 \tag{4.51}$$

$$h_{j,j+1} = h_{j+1,j}, \ j = 1, \ldots, m. \tag{4.52}$$

In other words, the matrix $H_m = A_m$ obtained from the Arnoldi process is real, tridiagonal, and symmetric [2].

Of course, $h_{ij} \neq 0$ only for elements at the diagonal and subdiagonals simply reflects the fact that a newly generated vector Aq_j is already orthogonal to all but its two preceding basis vectors, as mentioned in Chapter 4.1.2. Therefore, besides the matrix A and the new vector Aq_j one only has to store the two preceding basis vectors in memory during computations (beside the h_{ij}). This is what makes the Lanczos algorithm fast and memory efficient.

Since most h_{ij} vanish a new notation is convenient

$$\begin{aligned} \alpha_j &:= h_{jj} \\ \beta_j &:= h_{j-1,j}, \end{aligned} \tag{4.53}$$

in which A_m becomes

$$A_m = \begin{pmatrix} \alpha_1 & \beta_2 & & & & \\ \beta_2 & \alpha_2 & \beta_3 & & & \\ & \ddots & \ddots & \ddots & & \\ & & & & \beta_m & \\ & & & \beta_m & \alpha_m & \end{pmatrix}. \tag{4.54}$$

In this notation the Lanczos algorithm reads:

L :	The Lanczos Algorithm		
L(1):	Start :	Choose q_1 with $\|\|q_1\|\| = 1$	
		Set	$\beta_1 := 0$
		Set	$q_0 := 0$
	Iterate :	$j = 1, \ldots, m$:	
L(2):		$\alpha_j = \langle q_j, Aq_j \rangle$	
L(3):		$w_j = Aq_j - \alpha_j q_j - \beta_j q_{j-1}$	
L(4):		$\beta_{j+1} = \|\|w_j\|\|$	
L(5):		$q_{j+1} = w_j / \beta_{j+1}$	

Table 4.4: The hermitian Lanczos algorithm.

So the Lanczos algorithm is nothing but the Arnoldi algorithm (Table 4.3) applied to an Hermitian matrix.

4.2 The Tridiagonal Approach

Chapter 4.1.4 revealed that the Lanczos algorithm creates a tridiagonal representation $T_m := A_m$ of a Hermitian matrix A. With this knowledge one can easily derive the Lanczos algorithm not as a Rayleigh-Ritz procedure on a Krylov space as starting point but as a method to reduce a Hermitian matrix to tridiagonal form. This approach leads to a three-term recurrence for the Lanczos vectors that defines the algorithm. The way the Lanczos algorithm is presented in this approach will later make it easier to study its behavior in the presence of rounding errors.

For an Hermitian $n \times n$ matrix A, a tridiagonal representation

$$T := T_n = \begin{pmatrix} \alpha_1 & \beta_2 & & & \\ \beta_2 & \alpha_2 & \beta_3 & & \\ & \beta_3 & \ddots & \ddots & \\ & & \ddots & & \beta_n \\ & & & \beta_n & \alpha_n \end{pmatrix} \tag{4.55}$$

can be always found that is generated by an unitary matrix $Q = \Big(q_1, \ldots, q_n\Big)$ via

$$T = Q^\dagger A Q. \tag{4.56}$$

T and Q are completely determined by A and q_1.

Proposition 4.2.1
If equation (4.56) holds and Q is written by columns as $Q = (q_1, \ldots, q_m)$ then both T and Q are completely determined by A and q_1 (or by A and q_n) [10].

Since Q is orthonormal, (4.56) can be rewritten as

$$AQ = TQ. \tag{4.57}$$

Equating columns on each side gives

$$\begin{aligned} Aq_1 &= \alpha_1 q_1 + \beta_2 q_2 && (4.58)\\ Aq_i &= \beta_i q_{i-1} + \alpha_i q_i + \beta_{i+1} q_{i+1} \ , \qquad i = 2, \ldots, n-1 && (4.59)\\ Aq_n &= \beta_n q_{n-1} + \alpha_n q_n. && (4.60) \end{aligned}$$

Since q_i are orthonormal the coefficients a_k follow from

$$\alpha_i = q_i^\dagger A q_i, \qquad i = 1, \ldots, n. \tag{4.61}$$

To determine β_k rearrange (4.58) and (4.59)

$$\begin{aligned} \beta_2 q_2 &= Aq_1 - \alpha_1 q_1 && (4.62)\\ \beta_{i+1} q_{i+1} &= Aq_i - \beta_i q_{i-1} - \alpha_i q_i \ , \qquad i = 2, \ldots, n-1 && (4.63) \end{aligned}$$

or

$$\beta_{i+1} q_{i+1} = Aq_i - \beta_i q_{i-1} - \alpha_i q_i \ , \qquad i = 1, \ldots, n-1 \tag{4.64}$$

if $q_0 \equiv 0$, and take the norm, making use of $||q_i|| = 1$

$$\beta_{i+1} = ||Aq_i - \beta_i q_{i-1} - \alpha_i q_i|| \ , \qquad i = 1, \ldots, n-1. \tag{4.65}$$

Choosing the β_k as positive quantities does not pose a constraint. From (4.59) also follows that in iteration step i the next Lanczos vector q_{i+1} is determined by the three-term recurrence

$$\beta_{i+1} q_{i+1} = Aq_i - \beta_i q_{i-1} - \alpha_i q_i. \tag{4.66}$$

Therefore, the equations that determine the tridiagonal representation and the Lanczos vectors can be summarized by

$$\begin{aligned} \alpha_i &= q_i^\dagger A q_i && (4.67)\\ \beta_{i+1} &= ||Aq_i - \beta_i q_{i-1} - \alpha_i q_i|| && (4.68)\\ q_{i+1} &= (Aq_i - \beta_i q_{i-1} - \alpha_i q_i)/\beta_{i+1}. && (4.69) \end{aligned}$$

By comparing these expressions with (4.4) it is easy to see that they just define the Lanczos algorithm. As promised, the tridiagonal approach is a fast way to derive the algorithm but it does not provide much insight in its relationship to Krylov spaces and its physical significance.

The starting point for the investigation of the effects of finite arithmetics precision are the matrix equations

$$\begin{aligned} Q_j^\dagger Q_j &= \mathbb{1}_j && (4.70)\\ AQ_j - Q_j T_j &= q_{j+1}\beta_{j+1} e_j^\dagger && (4.71) \end{aligned}$$

that specify the output of the algorithm at each iteration step j. Eq. (4.70) represents the orthogonality among the Lanczos vectors and (4.71) is the three-term recurrence (that is nothing but (4.42)).

4.3 Breakdowns

At least in exact arithmetic the Lanczos algorithm can break down. It turns out that this happens when the Lanczos basis spans an invariant subspace under A so that the eigenpairs obtained from this subspace are the exact ones.

In practice the algorithm usually does not break down due to rounding errors that lead to a quick deviation from the theoretical behavior. However, when applied to highly degenerate Hilbert spaces, where the full Hilbert space fragments into many invariant subspaces of small dimensions, after a few iterations the Lanczos basis will span such an invariant subspace almost exactly and the algorithm will almost break down because the impact of rounding errors on the algorithm is limited after only a few iterations. Therefore, it is worthwhile to examine the breakdown of the algorithm more closely.

Proposition 4.3.1
The Lanczos algorithm breaks down at step j (i.e. $w_j = 0$ or equivalent $\beta_{j+1} = 0$) if and only if the minimal polynomial of q_1 is of degree j. Moreover, in this case the subspace $\mathcal{K}_j$ is invariant and the approximate eigenvalues and eigenvectors are exact [2].

During the j-th iteration a vanishing w_j in step L(3) of the Lanczos algorithm (Table 4.4) causes the algorithm to break down: Step L(4) then produces $\beta_{j+1} = 0$ and therefore L(5) fails to create the new Lanczos vector q_{j+1}. However, this does not happen accidently: The algorithm stops in the j-th step if and only if the degree $\mu(q_1, A)$ of the minimal polynomial of q_1 with respect to A is equal to j. Then $\mathcal{K}_m(q_1, A)$ is invariant under A and the eigenvalues obtained from this subspace are exact. To prove this remarkable statement ($\Rightarrow$) set $\mu(q_1, A) = j$ and assume that $w_j \neq 0$. Then a new Lanczos vector q_{j+1} could be generated and the Krylov space spanned by $q_1, \ldots, q_{j+1}$ would have dimension $j + 1$. This would contradict Proposition 4.1.4 that the dimension of a Krylov space can not exceed the degree of the minimal polynomial. On the other hand ($\Leftarrow$), if $w_j = 0$ then one has found a polynomial $p(A)$ such that $p(A)q_1 = 0$. Therefore, from the definition of μ, $\mu(q_1, A) \leq j$. Since the algorithm did not break down in a previous step one is left with $\mu(q_1, A) = j$. Since $\mathcal{K}_j$ is invariant under A the Ritz pairs are according to Proposition 4.1.2 exact. $\square$

After a breakdown the algorithm can be restarted. One has to choose a new start vector, which in typical calculations is just another random vector, that has to be orthogonalized against each Lanczos vector to ensure the orthogonality of the Lanczos basis and to avoid this vector to have components in the already spanned invariant subspace. This guarantees that the algorithm will span another invariant subspace. For this purpose all generated Lanczos vectors have to be stored at least in secondary memory. After having orthogonalized the new vector against the Lanczos basis the algorithm can be continued as before.

Since $\beta_{j+1} = 0$ if the algorithm breaks down in the j-th step, the Hessen-

berg matrix H_m not surprisingly takes a block diagonal form

$$\left(\begin{array}{ccccc|ccc} \alpha_1 & \beta_2 & & & & & & \\ \beta_2 & \ddots & & & & & & \\ & & \alpha_{j-1} & \beta_j & & & & \\ & & \beta_j & \alpha_j & 0 & & & \\ \hline & & & 0 & \alpha_{j+1} & \beta_{j+2} & & \\ & & & & \beta_{j+2} & \ddots & \beta_m & \\ & & & & & \beta_m & \alpha_m \end{array}\right). \tag{4.72}$$

This simply reflects the well-known fact that eigenvalues from independent subspaces can be calculated separately.

The degree of the minimal polynomial has an descriptive interpretation. It can be regarded as the number of eigenvectors that build the start vector q_1. Let $\{\nu_1, \ldots, \nu_n\}$ be the eigenbasis of A, then the expansion of q_1 in this (properly ordered) basis reads

$$q_1 = \sum_{i=1}^{m \leq n} c_i \nu_i \in \mathcal{V}_m. \tag{4.73}$$

This vector lies in the space $\mathcal{V}_m$ of dimension m spanned by $\nu_1, \ldots, \nu_m$. This is also true for the Krylov vectors because operating with A on this vector only yields another linear combination of these eigenvectors

$$\begin{aligned} q_k &= A^{k-1} q_1 \\ &= \sum_{i=1}^{m \leq n} c_i A^{k-1} \nu_i \\ &= \sum_{i=1}^{m \leq n} c_i \xi_i^{k-1} \nu_i \in \mathcal{V}_m. \end{aligned} \tag{4.74}$$

After m steps of the Lanczos algorithm one has generated m linear independent (orthogonal) vectors that span $\mathcal{V}_m$. So the algorithm has to break down because a $m+1$-th Krylov vector can not lie in $\mathcal{V}_m$ and be orthogonal to the full basis of $\mathcal{V}_m$ at the same time.

4.4 Convergence in Exact Arithmetics

"It is too good to be true - as we shall soon see."
Beresford Parlett [10].

In his original paper [1] Lanczos already mentioned the observation that during the Lanczos procedure the roots of the characteristic polynomials of the tridiagonal matrices T_m do not change much from the beginning and that the same could be said about the vibrational modes associated with these roots (i.e. the eigenvalues). The Lanczos algorithm converges fast and monotone to a few extreme eigenvalues what makes it suitable for the calculation of physical ground states and low-lying excited states. There are some theorems about the convergence of the eigenvalues obtained from the Lanczos algorithm ([13]) but only the few listed below are of practical interest, which makes the algorithm a variational method.

The Lanczos algorithm generates a series $(H_m)_m$ of Hessenberg matrices with growing dimension m from which in every iteration step Ritz values $\tilde{\xi}_i^{(m)}$ can be computed. Usually, only a few iteration steps are necessary to get good approximations of the extreme eigenvalues and so $\dim H_m \ll \dim A$. Therefore, the eigenvalues of H_m can almost effortless be computed in each iteration step and their convergence monitored. Obviously, monotone convergence is expected since the Ritz approximations of eigenvalues are the best approximations from the subspace $\mathcal{K}_m$ whose dimension increases in every iteration step.

Proposition 4.4.1
As long as $\beta_{m+1} \neq 0$ the spectra $\tilde{\xi}_i^{(m)}$ of H_m are non-degenerate and interlaced in the following way:

$$\begin{array}{lccccccc}
\xi_{min} \leq & & & \tilde{\xi}_1^{(1)} & & & & \leq \xi_{max} \\
\xi_{min} \leq & & \tilde{\xi}_1^{(2)} & & \tilde{\xi}_2^{(2)} & & & \leq \xi_{max} \\
\xi_{min} \leq & \tilde{\xi}_1^{(3)} & & \tilde{\xi}_2^{(3)} & & \tilde{\xi}_3^{(3)} & & \leq \xi_{max} \\
& & & \cdots & & & &
\end{array}$$

where ξ_{min} and ξ_{max} denote the minimum and maximum exact eigenvalues [13].

This Proposition is actually the Cauchy Interlace Theorem, because the matrices $(H_j)_j \leq m$ are submatrices of H_m. For an example see Figure 4.1.

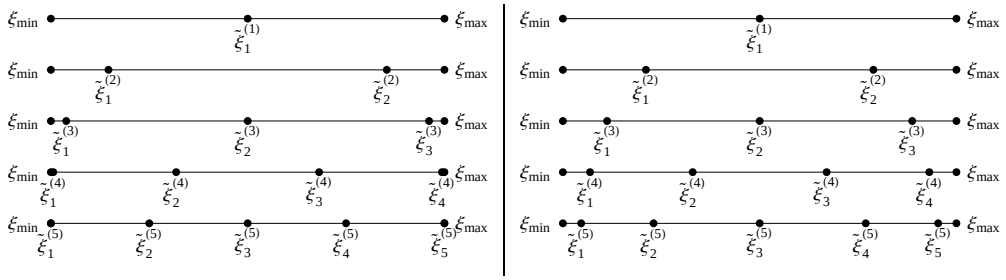


Figure 4.1: *Monotone convergence of the Ritz values* $\tilde{\xi}_i^{(Iteration)}$ *obtained from the first 5 iterations of the Simple Lanczos algorithm to the extreme (exact) values* ξ_{min} *and* ξ_{max} *of the matrix* $A = diag\ (1, 2, \ldots, n)$ *with start vector* $q_1 = \mathcal{N}(1, 1, \ldots, 1)$. *Left:* $n = 5$ *(final results are exact). Right:* $n = 1000$.

In exact arithmetics after n steps the Lanczos algorithm will generate a tridiagonal representation T of A that obeys (4.57), or $||AQ - TQ|| = 0$. This tridiagonal representation is the rotation of A into the full Lanczos basis. A first estimate of the quality of the approximations in iteration step m can be obtained by considering this norm when A is only rotated into the Lanczos basis generated so far,

$$||AQ_m - H_m Q_m|| \overset{(4.42)}{=} ||\beta_{m+1} q_{m+1} e_m^\dagger|| = \beta_{m+1}. \tag{4.75}$$

So this norm is just the last computed β.

Proposition 4.4.2
There are m *eigenvalues of* A, *call them* $\xi_{1'}, \ldots, \xi_{m'}$, *such that*

$$|\xi_{i'} - \tilde{\xi}_i^{(m)}| \leq \beta_{m+1}, \quad i = 1, \ldots, m$$

[9].

β_{m+1} is a global bound for all the Ritz values. With this an estimate of the quality of the approximations is completely effortless since β_{m+1} is computed by the algorithm anyway but it is rather rough. For a more precise estimate

that also discriminates among the Ritz values one considers the norm of the residual (4.4) of the i-th Ritz pair in the m-th iteration step

$$||A\tilde{\nu}_i^{(m)} - \tilde{\nu}_i^{(m)}\tilde{\xi}_i^{(m)}|| \overset{(4.17)}{=} ||AQ_m y_i^{(m)} - \tilde{\xi}_i^{(m)} Q_m y_i^{(m)}|| \tag{4.76}$$

$$= ||(AQ_m - Q_m\tilde{\xi}_i^{(m)})y_i^{(m)}|| \tag{4.77}$$

$$\overset{(4.42)}{=} ||(h_{m+1,m}q_{m+1}e_m^\dagger)y_i^{(m)}||. \tag{4.78}$$

In the Hermitian case H_m is symmetric, $h_{m+1,m} = h_{m,m+1}$, and from (4.53) follows $h_{m,m+1} = \beta_{m+1}$. Then the norm becomes

$$||A\tilde{\nu}_i^{(m)} - \tilde{\nu}_i^{(m)}\tilde{\xi}_i^{(m)}|| = ||(\beta_{m+1}q_{m+1}e_m^\dagger)y_i^{(m)}|| \tag{4.79}$$

$$\overset{||q_{m+1}||=1}{=} \beta_{m+1}|e_m^\dagger y_i^{(m)}| \tag{4.80}$$

$$= \beta_{m+1}(y_i^{(m)})_m \tag{4.81}$$

$$=: \beta_{im}, \tag{4.82}$$

where $(y_i^{(m)})_m$ is the bottom element of the Ritz vector $y_i^{(m)}$. This means that one can compute the residual norm $||A\tilde{\nu}_i^{(m)} - \tilde{\nu}_i^{(m)}\tilde{\xi}_i^{(m)}||$ without computing the (n-dimensional!) $\tilde{\nu}_i^{(m)}$ and without performing the matrix-vector multiplication $A\tilde{\nu}_i^{(m)}$. All one needs is provided by the algorithm and H_m.

Proposition 4.4.3

To each $\tilde{\xi}_i^{(m)}$ there is a corresponding eigenvalue of A, call it $\xi_{i'}$, such that

$$|\xi_{i'} - \tilde{\xi}_i^{(m)}| \leq \beta_{im}, \quad i = 1, \ldots m$$

[9].

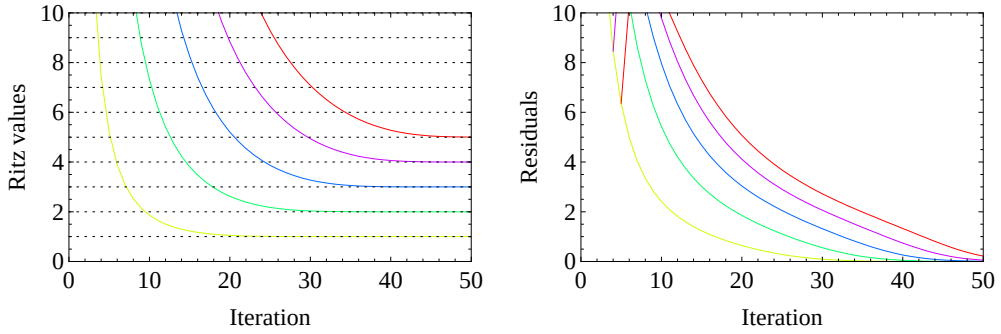


Figure 4.2: *The 5 lowest Ritz values $\tilde{\xi}_i^{(m)}$ and their residual norms β_{im}.*

There is also a gap theorem:

Proposition 4.4.4
Let the gap be $\gamma_i^{(m)} = \min_{k \neq i'} |\xi_k - \tilde{\xi}_i^{(m)}|$. *Then*

$$|\xi_{i'} - \tilde{\xi}_i^{(m)}| \leq \beta_{im}^2 / \gamma_i^{(m)}, \ \ i = 1, \ldots, m.$$

[9].

The convergence of a Ritz value depends on the gaps between the eigenvalues of A. The larger the ratio

$$\min\{\xi_{i+1} - \xi_i, \xi_i - \xi_{i-1}\}/(\xi_{i+1} - \xi_i) \tag{4.83}$$

the more does a Ritz value settle on to ξ_i [10], see Figure 4.3.

Furthermore, q_1 needs to have a non-vanishing component into the direction of an eigenvector ν_k one wants to approximate. It is easy to verify that if q_1 is orthogonal to ν_k then $\nu_k \notin \mathcal{K}$.

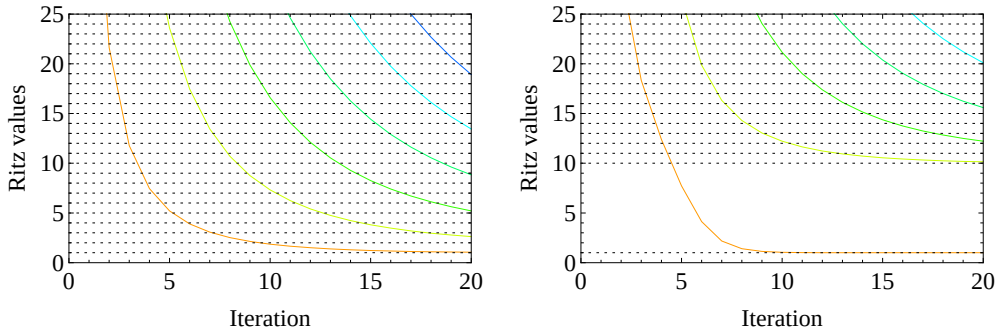


Figure 4.3: *Typical convergence of the leftmost Ritz values to the exact eigenvalues (dotted lines). Left:* $A = diag(1, 2, \ldots, n)$. *Right:* $A = diag(1, 10, 11, \ldots, n+8)$, $n = 100$, $q_1 = (1, 1, \ldots)$. *The convergence to the smallest eigenvalue is faster in the right figure because of the larger gap to the second one.*

4.5 Simple Ritz Restart

Restarting the Lanczos algorithm makes sense even if there is no breakdown. On the one hand, since in practice all Lanczos vectors have to be stored, after a certain number of iterations one can run out of memory without having experienced any convergence so far. On the other hand, as the iterations

go by, rounding errors will accumulate and contaminate the results. An inaccuracy of α_k or β_k in iteration k will affect the quality of the eigenvalue approximations in all further iterations.

To avoid these problems the algorithm can be halted after a certain number of iterations, or when any other halting criterion is fulfilled, and can be restarted using some of the information one has already gathered about the eigenvalue problem that is to be solved.

The simplest way for a reasonable restart scheme is to completely restart the algorithm with a new start vector that is closer to the eigenvector one wants to approximate. Because of the convergence pattern of Fig. 4.1, the restarted algorithm then continues converging from this value further on to the lowest exact eigenvalue. If one is interested in the lowest eigenvalue, the natural choice for this start vector is simply the first Ritz vector of the preceding Lanczos run. Using this Ritz vector, the first computed α in the restarted algorithm will be the corresponding Ritz value which will be the single eigenvalue of the Hessenberg matrix in the first iteration step.

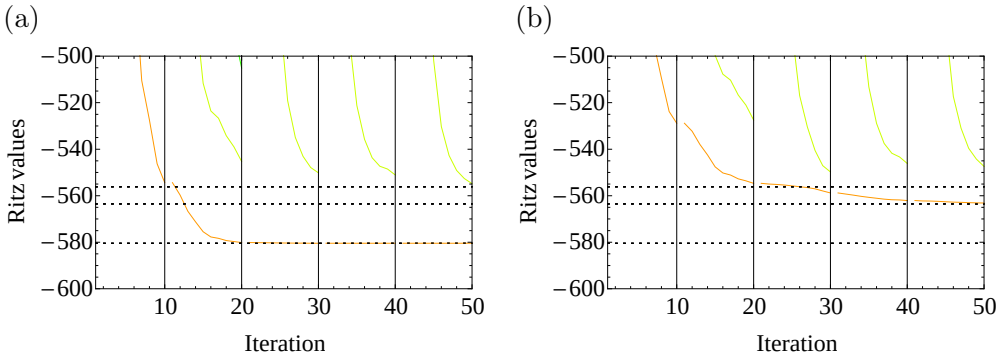


Figure 4.4: *Restarted Lanczos algorithm applied to A with $a_{ij} = 10^{\cos(ij)}$, $n = 2000$, $q_1 = (1, 1, \dots)$. The vertical lines indicate the iterations at which the algorithm was stopped for restart, the dotted lines are the 3 lowest exact eigenvalues. (a) : Convergence to the lowest eigenvalue. (b) : With a start vector orthogonal to the converged vector from (a) the lowest Ritz value converges to the second exact eigenvalue and so on.*

To obtain more eigenvalues than the lowest one, this iteration scheme can be performed until the lowest Ritz vector has converged. Then the algorithm

can be restarted with a start vector orthogonal to the converged Ritz vector. In exact arithmetics, if the converged Ritz vector is the exact eigenvector, the Krylov space generated in this algorithm will be orthogonal to the converged Ritz vector. In actual computations the Ritz vector approximation will not be exact and so the newly generated Lanczos vectors have to be kept orthogonal to the converged vector in every iteration step.

Chapter 5

Numerical Issues

In theory the Lanczos algorithm is a fast, efficient and, at least after n steps, a direct method to obtain the eigenvalues of an Hermitian matrix A. In practice, however, it was long considered as unstable and, therefore, inferior to other methods. It turns out that already small rounding errors have a destructive effect on the orthogonality of the Lanczos basis. With this loss of orthogonality comes the loss of all the remarkable theoretical properties of the algorithm.

5.1 Loss of Orthogonality

Why the Lanczos algorithm is cherished can be understood from a look at Figure 5.1 (a). Although the matrix is quite large ($n = 5000$) after 4 iterations the approximation to its smallest eigenvalue is already almost exact. After 10 iterations about 5 excellent (to the eye) approximations of eigenvalues at the lower end have emerged from the algorithm. This is what is expected from Chapter 4.4: Every Ritz value converges to a distinct eigenvalue of A and usually the convergence to the extreme ones is fastest (and there are large gaps at the lower end of the spectrum). However, using the Simple Lanczos algorithm of Table 4.4 in the presence of rounding errors, the results look quite different, as illustrated in Figure 5.1 (c). While the results of the first iterations look the same, thereafter the algorithm seems to produce copies of approximations of eigenvalues already computed. After 40 iterations there are already 4 copies of the smallest eigenvalue where only

one should be.

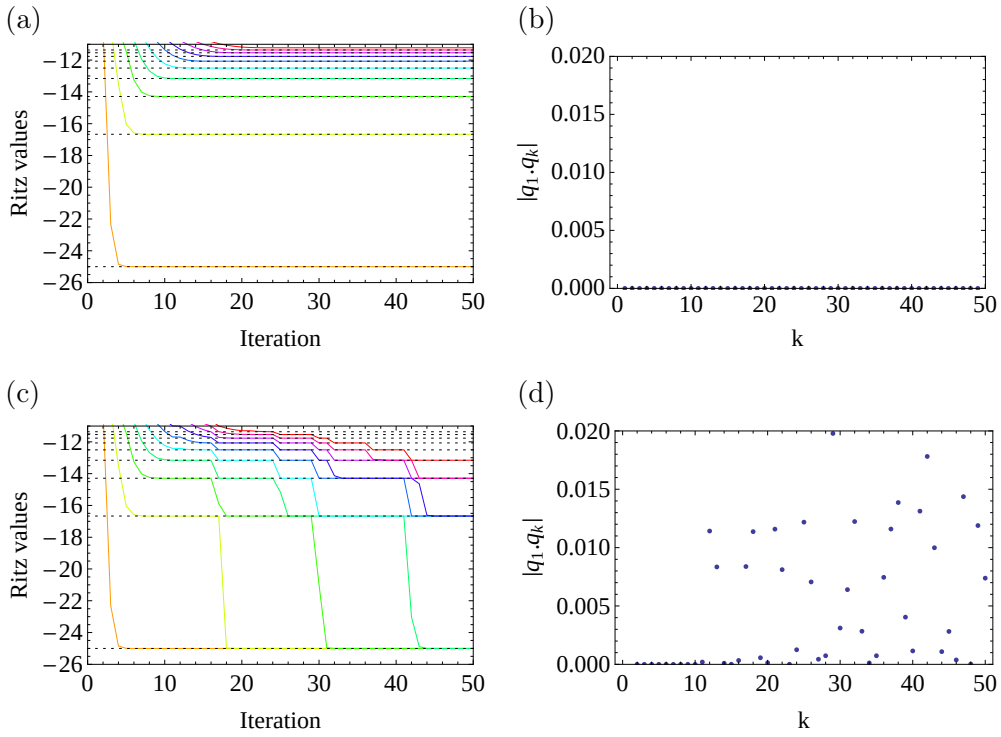


Figure 5.1: *Results of the Simple Lanczos algorithm (c) and the Lanczos algorithm with full reorthogonalization (a) for the matrix* $A = diag(10k/(1.2-k))$, $k = 1, \ldots, 5000$ *with* $q_1 = \mathcal{N}(1, 1, \ldots)$ *and round-off* $\epsilon = 10^{-16}$. *To obtain (a) the Lanczos basis was kept orthogonal by full reorthogonalization in each iteration step while for (c) the new Lanczos vector was orthogonalized against the two preceding Lanczos vectors what leads to non-vanishing inner products among the Lanczos basis (d).*

Figure 5.1 (a) was obtained using the Lanczos algorithm with full reorthogonalization. In every iteration step the new Lanczos vector was orthogonalized against the full Lanczos basis and not only against the two preceding Lanczos vectors. So the orthogonality among the Lanczos vectors was kept to working precision. Figure 5.1 (b) shows that the inner product of q_1 with the rest of the Lanczos basis $\{q_i, i \neq 1\}$ vanishes. When the expensive full reorthogonalization is not performed, as in Figs. 5.1 (c) and (d), this situation changes. After about 10 iteration the orthogonality is already lost and the algorithm becomes unreliable. Since the matrix is diagonal and,

therefore, extremely sparse the orthogonality loss is still quite small. Usually orthogonality is lost totally after around 20 iterations.

In the presence of rounding errors the two basic equations of the Lanczos algorithm (4.70) and (4.71) fail and have to be adapted to the new non-ideal situation

$$Q_j^\dagger Q_j = C_j^\dagger + \mathbb{1}_j + C_j \tag{5.1}$$

$$AQ_j - Q_j T_j = q_{j+1}\beta_{j+1}e_j^\dagger + F_j. \tag{5.2}$$

From now on Q_j and T_j lose their meaning from the ideal picture drawn in the discussion of exact arithmetic. They denote the error-prone quantities one works with that are stored in the computer. C_j is a strictly upper triangular matrix that measures the orthogonality loss among the Lanczos vectors (since $Q_j^\dagger Q_j$ is symmetric and all q_i are normalized perfectly to working precision, all information about orthogonality loss beyond round-off level is encoded in C_j). Obviously, its matrix elements are given by the inner products of the Lanczos vectors, $(C_j)_{ik} = q_i.q_k$. Similarly, the matrix F_j indicates how much the three-term-recurrence fails due to round-off in iteration step j. Therefore, F_j remains at round-off level, at least for sparse matrices, satisfying an inequality of the form $||F_j|| \leq \phi(n)\epsilon||A||$ with round-off ϵ and some linear function ϕ [9]. However, C_j grows rapidly with the iterations, see Figure 5.2.

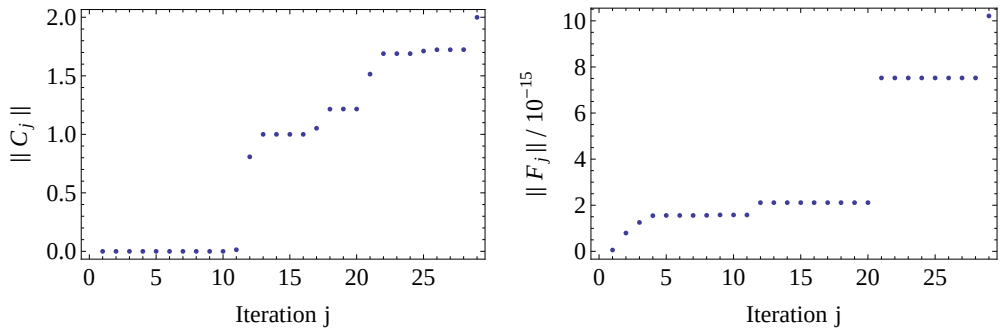


Figure 5.2: *Deviations from the theoretical behavior for the example given in Figure 5.1. The orthogonality loss among the Lanczos vectors is strong (left) while the three-term-recurrence in every iteration step only fails at round-off level (right).*

5.1.1 Systematics in the Orthogonality Loss

Since the orthogonality loss is the crucial point in understanding the numerical Lanczos behavior it is worthwhile to examine it further. At a first glance the orthogonality loss, since it is due to round-off errors, seems to be random and uncontrollable. However, already Figure 5.1 (c) indicates that there is more to say about the Lanczos algorithm in finite precision arithmetic: Although the numerical behavior of the Lanczos algorithm deviates from the behavior expected in exact arithmetic, the obtained Rayleigh-Ritz approximations are obviously not arbitrarily. As mentioned before, the numerical Lanczos algorithm produces copies of Ritz values. Therefore, there should also be copies of Ritz vectors and it turns out that the appearance of such copies is connected to the orthogonality loss among the Lanczos vectors. To see this, in addition to the Lanczos picture, one takes a look at the Ritz picture, noticing that the Ritz vectors also represent a basis of the Krylov space. As for the Lanczos basis the Ritz basis also obeys in exact arithmetic the equation $\tilde{V}_j^H \tilde{V}_j = \mathbb{1}_j$ where V_j is the matrix built from column vectors $\tilde{\nu}_i^{(j)}$. So the orthogonality loss among the Ritz basis can be measured by an upper triangular matrix G_j with matrix elements $(G_j)_{ik} = \tilde{\nu}_i^{(j)}.\tilde{\nu}_k^{(j)}$ satisfying

$$\tilde{V}_j^H \tilde{V}_j = G_j^H + \mathbb{1} + G_j. \tag{5.3}$$

It is worth noticing that the once computed Lanczos basis is fixed while the Ritz basis changes in every iteration step.

Figures 5.3 - 5.5 show how orthogonality is lost in the Lanczos and Ritz picture. In each basis orthogonality is maintained until iteration step 9. There one first notices a loss of orthogonality in both bases that even increases in the next iteration step. Since the Lanczos basis is fixed this orthogonality loss pattern remains unchanged in the following iterations while the one for the Ritz basis changes. In step 10 more Lanczos vectors lose orthogonality but orthogonality in the Ritz picture is completely restored except for one Ritz vector that is a copy of another Ritz vector since their scalar product equals 1. More precisely, it is Ritz vector $\tilde{\nu}_{10}^{(11)}$ that is a copy of $\tilde{\nu}_{11}^{(11)}$. This is expected from Figure 5.1, for instance, since it is the extreme eigenvalues the Lanczos algorithm first produces copies for. This convergence also indicates that $\tilde{\nu}_{11}^{(11)}$ is already close to the dominant eigenvalue of A. So there is a graphic picture behind the orthogonality loss in the Lanczos basis that can

be revealed from a look at the Ritz picture [8]:

$$\text{Convergence (Ritz basis)} \;\Rightarrow\; \text{Loss of Orthogonality (Lanczos Basis)}.$$

From iteration 11 on, no additional orthogonality loss can be observed until iteration 18 when two additional Ritz vectors begin to converge (Figure 5.4). Once again, after these Ritz vectors have converged in iteration step 20, there is no additional orthogonality loss until iteration 26, where another Ritz vector begins to converge and completes its convergence in iteration 29 (Figure 5.5).

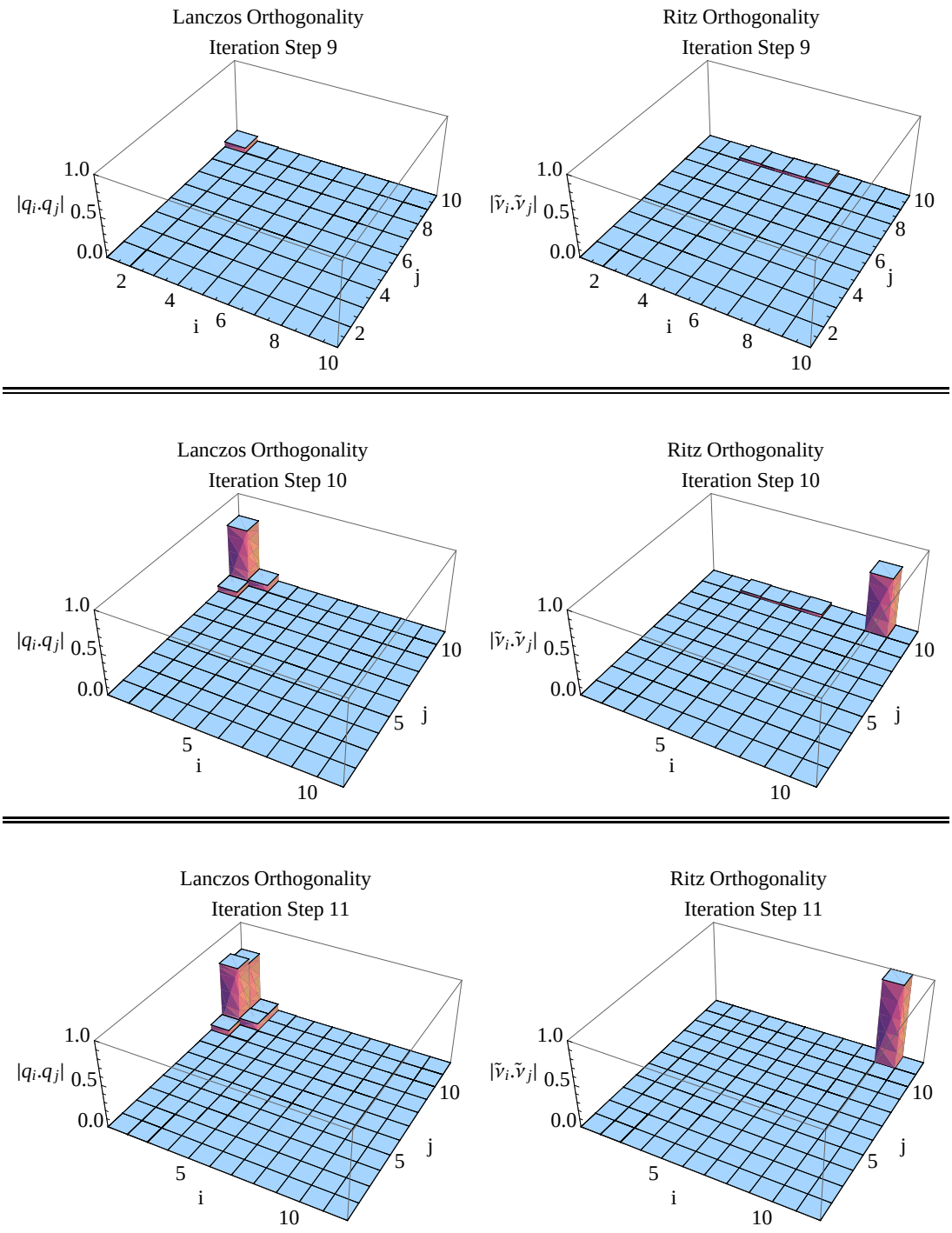


Figure 5.3: *Orthogonality loss in the Lanczos and Ritz picture, measured by* C_j *and* Y_j*, from the Simple Lanczos algorithm applied to* A *with* $a_{ij} = 10^{5\cos(i\times j)}$*,* $n = 5000$*,* $q_1 = \mathcal{N}(1, 1, \dots)$*. Depicted for* $i < j$*. Round-off:* $\epsilon = 10^{-8}$*.*

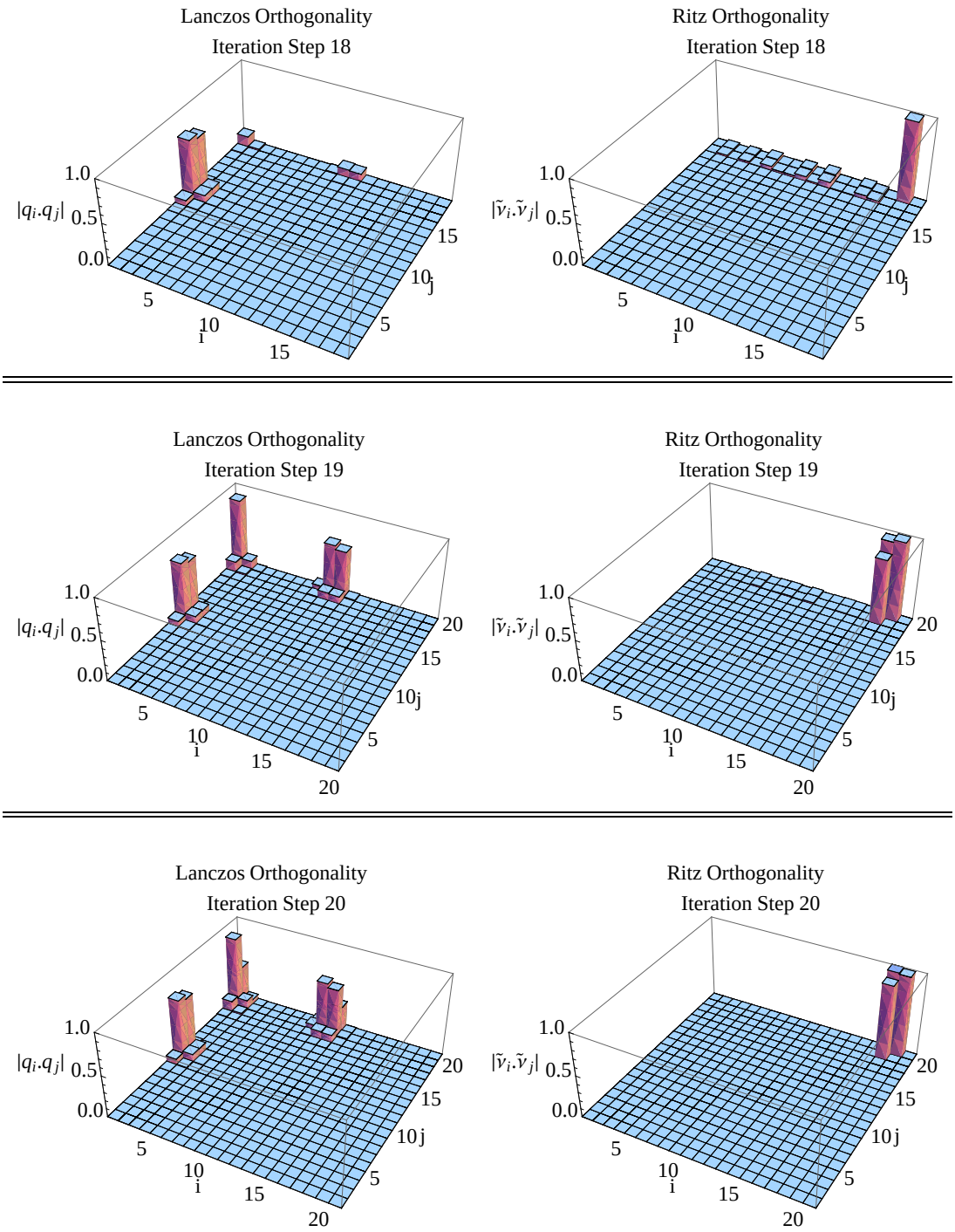


Figure 5.4: *Orthogonality loss in the Lanczos and Ritz picture, measured by* C_j *and* Y_j*, from the Simple Lanczos algorithm applied to* A *with* $a_{ij} = 10^{5\cos(i\times j)}$*,* $n = 5000$*,* $q_1 = \mathcal{N}(1, 1, \dots)$*. Depicted for* $i < j$*. Round-off:* $\epsilon = 10^{-8}$*.*

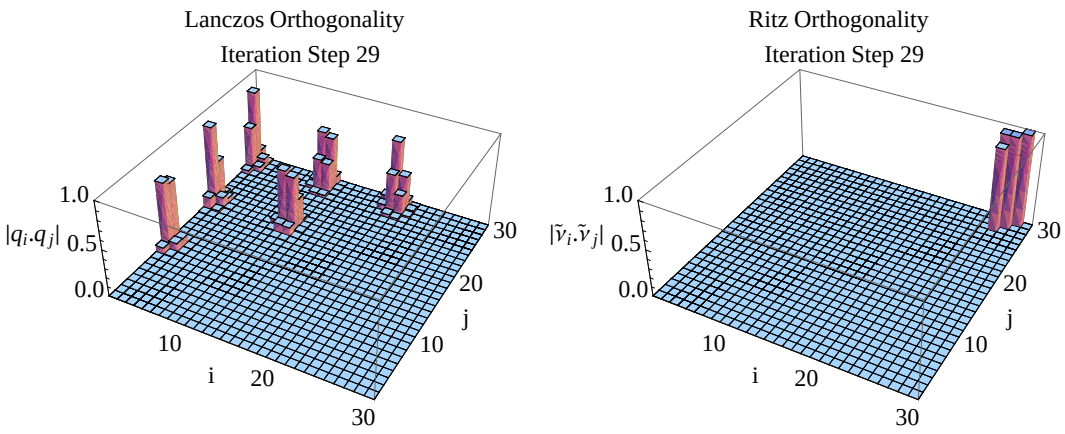


Figure 5.5: *Orthogonality loss in the Lanczos and Ritz picture, measured by C_j and Y_j, from the Simple Lanczos algorithm applied to A with $a_{ij} = 10^{5\cos(i\times j)}$, $n = 5000$, $q_1 = \mathcal{N}(1, 1, \dots)$. Depicted for $i < j$. Round-off: $\epsilon = 10^{-8}$.*

5.1.2 Paige's Theorem

A series of pictures equivalent to Figures 5.3 - 5.5 can be found in [10]. In this paper the discussion goes further by investigating the angles $|\tilde{\nu}_i^{(j)}.q_{j+1}|$ between the Ritz vectors $\tilde{\nu}_i^{(j)}$ and the last Lanczos vectors q_{j+1}. These scalar products vanish in exact arithmetic because

$$\text{span}\{\tilde{\nu}_1^{(j)}, \dots, \tilde{\nu}_j^{(j)}\} = \text{span}\{q_1, \dots, q_j\} = \mathcal{K}_j \perp q_{j+1}. \tag{5.4}$$

It turns out that q_{j+1} is tilted mostly into directions of converged Ritz vectors, i.e. Ritz vectors that are good approximations of eigenvectors.

This and the results from Section 5.1.1 are backed by Paige's Theorem:

Proposition 5.1.1 (Paige's Theorem)
If local orthogonality is maintained in the Lanczos process, governed by (5.2), i.e. if $q_{k+1}^H q_k = 0$ for all k, then for each $i \le j$,

$$(a) \quad \langle \tilde{\nu}_i^{(j)}, q_{j+1} \rangle = \frac{\gamma_{ij}^{(j)}}{\beta_{j+1}(y_i^{(j)})_j}$$

$$(b) \quad \text{for } k \le j, i \ne k,$$
$$(\tilde{\xi}_i - \tilde{\xi}_k)\tilde{\nu}_i^H \tilde{\nu}_k = \gamma_{ii}^{(j)} \left(\frac{(y_k^{(j)})_j}{(y_i^{(j)})_j} \right) - \gamma_{kk}^{(j)} \left(\frac{(y_i^{(j)})_j}{(y_k^{(j)})_j} \right) + (\gamma_{ki}^{(j)} - \gamma_{ik}^{(j)})$$

where

$$\Gamma^{(j)} = (\gamma_{ik}^{(j)}) = \tilde{V}_j^H [\textit{strictly upper triangle } (Q_j^H F_j - F_j^H Q_j)] \tilde{V}_j \tag{5.5}$$

and $(y_k^{(j)})_j$ *denotes the bottom element (*j*-th component) of* $y_k^{(j)}$ *[10].*

The interesting part of Paige's Theorem is Proposition 5.1.1 (a). The matrix $\Gamma^{(j)}$ looks quite unhandy but according to [8] its matrix elements $\gamma_{ik}^{(j)}$ satisfy the inequality

$$|\gamma_{ik}^{(j)}| \leq \epsilon ||A||, \quad \forall i, k, j, \tag{5.6}$$

and that is all one needs to know. The connection between converging Ritz vectors and the orthogonality loss of the Lanczos vectors is best explained in the words of Parlett and Scott (in adapted notation, Y_j denotes the matrix built from column vectors $y_i^{(j)}$) [9]:

> *The bottom elements of the* $y_i^{(j)}$ *appear in a special way. With any good program,* Y_j *will be orthonormal (to working accuracy) so that* $\sum_{i=1}^{j} (y_i^{(j)})_j^2 = 1$.

(The sum $\sum_{i=1}^{j} (y_i^{(j)})_j^2 = 1$ is not the norm of the i-th vector, which would equal 1 as well – it is a sum running over the bottom elements of all $y_i^{(j)}$. The matrix built from all $y_i^{(j)}$,

$$Y_j = \begin{pmatrix} \uparrow & \uparrow & & \uparrow \\ y_1^{(j)} & y_2^{(j)} & \dots & y_j^{(j)} \\ \downarrow & \downarrow & & \downarrow \end{pmatrix},$$

is orthonormal and so for every column it is $||(Y_j)_i|| = ||y_i^{(j)}|| = 1$. From orthonormality of Y_j follows that

$$Y_j^T = \begin{pmatrix} \leftarrow & y_1^{(j)} & \rightarrow \\ \leftarrow & y_2^{(j)} & \rightarrow \\ & \vdots & \\ \leftarrow & y_j^{(j)} & \rightarrow \end{pmatrix}$$

is also orthonormal and for every column one has $1 = ||(Y_j^T)_i|| = \sum_{i=1}^{j} (y_i^{(j)})_j^2$.)

If

$$|(y_k^{(j)})_j|\tilde{|}(y_i^{(j)})_j|\tilde{j}^{-1/2}, \quad |\tilde{\xi}_i - \tilde{\xi}_k| > ||T_j||/100, \tag{5.7}$$

then the error bounds (Proposition 4.4.3) on $\tilde{\xi}_i$ and $\tilde{\xi}_k$ indicate that they are poor eigenvalue approximations while Paige's Theorem shows that $\tilde{\nu}_i$ and $\tilde{\nu}_k$ are orthogonal to working accuracy.

A Ritz vector is considered unconverged if it has a bottom element of the order $j^{-1/2}$ because that is the order of elements of the normalized vector $\mathcal{N}(1, 1, \dots)$ that shows no indications for any convergence. Note that

$$\tilde{\nu}_i^{(j)} = Qy_i^{(j)} = (y_1^{(j)})_1 q_1 + \cdots + (y_1^{(j)})_j q_j.$$

Therefore, for unconverged Ritz vectors their bottom elements, absolute values are large and so $\beta_{j+1}(y_i^{(j)})_j$ will be large too. This is nothing but β_{ij} from Proposition 4.4.3 from which then the large error bounds follow. On the other hand from Paige's Theorem Proposition 5.1.1 (a) follows that the i-th Ritz vector is orthogonal to q_{j+1} since $\gamma_{ij}^{(j)}\tilde{\epsilon}||A||$ is divided by a large number.

Conversly, if

$$|(y_i^{(j)})_j| < 10^{-3}, \tag{5.8}$$

say, then $\tilde{\xi}_i$ (if isolated) is a good eigenvalue approximation, $\tilde{\nu}_i$ is good too, and $\tilde{\nu}_i$ will not be orthogonal to any unconverged $\tilde{\nu}_k$ (indicated by $(y_k^{(j)})_j\tilde{j}^{-1/2}$).

On the other hand, a Ritz vector with bottom element $\tilde{1}0^{-3}$ is considered as converged. This number is independent of n since $y_i^{(j)} \in \mathcal{K}_j$, and so the general condition for convergence can be formulated as $(y_i^{(j)})_j \ll j^{-1/2}$. With a small bottom element then the statements of Propositions 4.4.3 and 5.1.1 reverse.

Since Y_j is orthogonal to working accuracy, it is Q_j which must have lost orthonormality. The better the approximations $\tilde{\xi}_i$ and $\tilde{\nu}_i^{(j)}$ the greater the departure of Q_j from orthogonality. □

Actually, Proposition 4.4.3 can not be used since it is valid for exact arithmetics. There is, however, a very similar theorem for the case of finite precision arithmetic, that Parlett and Scott worked with and that also can be found in [9].

The conclusion that should be drawn from Section 5.1 is the following: Although the Lanczos algorithm is a remarkably powerful method in theory, numerical reality diminishes its capabilities drastically. The reason for the loss of its power is the loss of orthogonality in the Lanczos basis but the properties of exact arithmetics can most widely be restored by manually orthogonalizing the Lanczos basis – at a certain cost. Since the Gram-Schmidt procedure becomes too expensive for large matrices and many iterations, more efficient methods for orthogonalizing the Lanczos basis are to be developed. Here one can exploit the fact that the orthogonality loss happens not randomly but rather with a systematics behind it.

5.2 Variants of the Lanczos Algorithm

There are many variants of the Lanczos algorithm. Some of them, the Lanczos algorithm with full or selective reorthogonalization for instance, represent more or less sophisticated ways to tackle the orthogonality loss problem, and some are simply other formulations of the Simple Lanczos algorithm with different numerical stability properties.

5.2.1 The Simple Lanczos Algorithm

The Simple Lanczos is the cheapest way to compute eigenvalues using the Lanczos algorithm. However, the denotation "Simple Lanczos" is not uniquely defined because there are (in exact arithmetics) several equivalent formulations of the Simple Lanczos algorithm based on two different ways to compute α_j and β_j. The resulting algorithms have been analyzed by Paige [6], [7]. In the following the term Simple Lanczos (LS) algorithm will denote the algorithm found to be the most stable, Table 5.1, and that is also presented in [8].

The equivalence of LS (Table 5.1) and L (Table 4.4) is not obvious and there is no source known for citation so there will be a few words to motivate

this statement. It is easy to show that the quantities q_j, α_j and β_j fulfill the same equations in algorithms LS and L but this does not prove their equivalence yet. It also has to be shown that these quantities denote the same things in both algorithms, namely Lanczos vectors q_j and matrix elements of the Hessenberg matrix α_j and β_j. However, this is only shown for the simple case of quantities in the first iteration since a proof via induction turns out to be somewhat tricky.

LS :		A More Stable Simple Lanczos Algorithm
LS(1):	Start :	Choose r with $\|\|r\|\| = 1$
		Set $\beta_1 := 1$
		Set $q_0 := 0$
	Iterate :	$j = 1, \ldots, m$:
LS(2):		$q_j = r/\beta_j$
LS(3):		$p_j = Aq_j$
LS(4):		$r = p_j - \beta_j q_{j-1}$
LS(5):		$\alpha_j = \langle q_j, r \rangle$
LS(6):		$r = r - \alpha_j q_j$
LS(7):		$\beta_{j+1} = \|\|r\|\|$

Table 5.1: A more stable variant of the Simple Lanczos algorithm.

In L, $q_1^{(L)}$ is the manually chosen start vector while in LS this is $r^{(LS)}$ and $q_1^{(LS)}$ is given by $q_1^{(LS)} = r^{(LS)}/\beta_1^{(LS)}$. Since $\beta_1^{(LS)} = 1$ in LS, $q_1^{(L)}$ and $q_1^{(LS)}$ are equivalent. With this the equivalence of $\alpha_1^{(L)}$ and $\alpha_1^{(LS)}$ can be shown. In L, $\alpha_1^{(L)} = \langle Aq_1^{(L)}, q_1^{(L)} \rangle$. In LS $\alpha_1^{(LS)}$ is given by $\alpha_1^{(LS)} = \langle q_1^{(LS)}, r^{(LS)} \rangle$. At the point where $\alpha_1^{(LS)}$ is computed $r^{(LS)}$ denotes no longer the start vector but $r^{(LS)} = p_1^{(LS)} - \beta_1^{(LS)} q_0^{(LS)}$. Since $q_0^{(LS)} = 0$ and $p_1^{(LS)} = Aq_1^{(LS)}$ one arrives at

$\alpha_j^{(LS)} = \langle q_1^{(LS)}, Aq_1^{(L)} \rangle$. $\beta_2^{(L)}$ is computed in L(4) via

$$\begin{aligned} L(4) \ : \quad \beta_2^{(L)} &= ||w_1^{(L)}|| \\ &= ||Aq_1^{(L)} + \alpha_1^{(L)} q_1^{(L)} - \beta_1^{(L)} q_0^{(L)}|| \\ &= ||Aq_1^{(L)} - \alpha_1^{(L)} q_1^{(L)}||. \end{aligned}$$

In LS one finds the same expression

$$\begin{aligned} LS(7) \ : \quad \beta_2^{(LS)} &= ||r^{(LS)}|| \\ &= ||p_1^{(LS)} - \beta_1^{(LS)} q_0 - \alpha_1^{(LS)} q_1^{(LS)}|| \\ &= ||Aq_1^{(LS)} - \alpha_1^{(LS)} q_1^{(LS)}|| \end{aligned}$$

in which $q_1^{(LS)}$ and $\alpha_1^{(LS)}$ have the same meaning as in L. Therefore, $\beta_2^{(LS)}$ is equivalent to $\beta_2^{(L)}$. To see that in both algorithms the Lanczos vectors are generated from the same equations compare

$$\begin{aligned} L(5) \ : \quad q_{j+1}^{(L)} &= w_j^{(L)} / \beta_{j+1}^{(L)} \\ &= \frac{Aq_j^{(L)} - \alpha_j^{(L)} q_j^{(L)} - \beta_j^{(L)} q_{j-1}^{(L)}}{\beta_{j+1}^{(L)}} \end{aligned}$$

with

$$\begin{aligned} LS(2) \ : \quad q_{j+1}^{(LS)} &= r / \beta_{j+1}^{(LS)} \\ &= \frac{p_j^{(LS)} - \beta_j^{(LS)} q_{j-1}^{(LS)} - \alpha_j^{(LS)} q_j^{(LS)}}{\beta_{j+1}^{(LS)}} \\ &= \frac{Aq_j^{(LS)} - \alpha_j^{(LS)} q_j^{(LS)} - \beta_j^{(LS)} q_{j-1}^{(LS)}}{\beta_{j+1}^{(LS)}} \end{aligned}$$

which is the same. In L the general formula to compute $\beta_{j+1}^{(L)}$ is

$$L(5) \ : \quad \beta_{j+1}^{(L)} = ||w_j^{(L)}|| = ||Aq_j^{(L)} - \alpha_j^{(L)} q_j^{(L)} - \beta_j^{(L)} q_{j-1}^{(L)}||$$

and this is the same as in LS

$$\begin{aligned} LS(7) \ : \quad \beta_{j+1}^{(LS)} &= ||r^{(LS)}|| \\ &= ||p_j^{(LS)} - \beta_j^{(LS)} q_{j-1}^{(LS)} - \alpha_j^{(LS)} q_j^{(LS)}|| \\ &= ||Aq_j^{(LS)} - \alpha_j^{(LS)} q_j^{(LS)} - \beta_j^{(LS)} q_{j-1}^{(LS)}||. \end{aligned}$$

Analogously, the generating equation for $\alpha_j^{(L)}$

$$L(2) \ : \ \alpha_j^{(L)} = \langle Aq_j^{(L)}, q_j^{(L)} \rangle$$

can be shown to be the same as in LS

$$\begin{aligned} LS(5) \ : \ \alpha_j^{(LS)} &= \langle q_j^{(LS)}, r^{(LS)} \rangle \\ &= \langle q_j^{(LS)}, p_j^{(LS)} - \beta_j^{(LS)} q_{j-1}^{(LS)} \rangle \\ &= \langle q_j^{(LS)}, p_j^{(LS)} \rangle - \beta_j^{(LS)} \langle q_j^{(LS)}, q_{j-1}^{(LS)} \rangle \\ &= \langle q_j^{(LS)}, Aq_j^{(LS)} \rangle. \end{aligned}$$

r carries no label j since its meaning varies during an iteration. Furthermore an index j would suggest to store all r_j, but this is not necessary since it only serves as a container.

In practice it turns out to be quite hard to discover significant differences between L and LS, but nevertheless, LS will be used as the basic Lanczos algorithm in the following.

5.2.2 The Lanczos Algorithm with full Reorthogonalization

The Lanczos algorithm with full reorthogonalization has already been proposed by Lanczos himself and was a first attempt to prevent the orthogonality loss among the Lanczos vectors. It is the most straight-forward and also the most expensive method for maintaining the orthogonality, since its recipe for keeping orthogonality is simply to orthogonalize the new Lanczos vector q_{j+1} against all the other vectors $\{q_1, \ldots, q_j\}$ in every iteration step. For large dimensions this is conveniently done by the Gram-Schmidt orthogonalization procedure

$$q_{j+1} := q_{j+1} - \sum_{i=1}^{j} \langle q_{j+1}, q_i \rangle q_i.$$

For the Gram-Schmidt procedure there is a more stable variant that should be used in practical implementations that projects the component on q_i of q_{j+1} after already having orthogonalized q_{j+1} against $\{q_1, \ldots, q_{i-1}\}$

$$\text{For } i = 1, \ldots, j \ : \ q_{j+1} := q_{j+1} - \langle q_{j+1}, q_i \rangle q_i.$$

This orthogonalization is placed between steps LS(6) and LS(7) so that β_{j+1} is computed from the norm of a perfectly orthogonalized r.

Figure 5.1 already showed the superiority of the Lanczos algorithm with full reorthogonalization over the Simple Lanczos. By maintaining orthogonality among the Lanczos basis it is avoided to produce copies of already computed Ritz values so that every different Ritz value represents an approximation to a different eigenvalue.

LFR :	Lanczos Algorithm with Full Reorthogonalization	
LFR(1):	Start :	Choose r with $\|\|r\|\| = 1$
		Set $\beta_1 := 1$
		Set $q_0 := 0$
	Iterate :	$j = 1, \ldots, m$:
LFR(2):		$q_j = r/\beta_j$
LFR(3):		$p_j = Aq_j$
LFR(4):		$r = p_j - \beta_j q_{j-1}$
LFR(5):		$\alpha_j = \langle q_j, r \rangle$
LFR(6):		$r = r - \alpha_j q_j$
	For :	$i = 1, \ldots, j$:
LFR(7) :		$r = r - \langle r, q_i \rangle q_i$
	End For	
LFR(8):		$\beta_{j+1} = \|\|r\|\|$

Table 5.2: The Lanczos algorithm with full reorthogonalization.

5.2.3 The Lanczos Algorithm with Selective Reorthogonalization

The main reason for the discussion of Paige's Theorem is its usefulness in the development of a more sophisticated way to avoid the orthogonality loss than

the Lanczos with full reorthonormalization does. Since there is systematics in the orthogonality loss there should also be a way to exploit this knowledge [8], [9].

In exact arithmetics, the scalar product of the Ritz vectors $\tilde{\nu}_i^{(j)}$ and the last computed Lanczos vector vanishes, (5.4). But in finite precision arithmetics q_{j+1} will have components in the directions of $\tilde{\nu}_i^{(j)}$. Orthogonalizing q_{j+1} against the Lanczos basis is equivalent to removing all its components into directions of the Ritz vectors. Of course this can be achieved by

$$q_{j+1} := q_{j+1} - \sum_{i=1}^{j} \langle \tilde{\nu}_i^{(j)}, q_{j+1} \rangle \tilde{\nu}_i^{(j)}, \tag{5.9}$$

but this is obviously no better way than the full reorthogonalization. However, from Paige's Theorem one knows that q_{j+1} has its strongest components into directions of converged Ritz vectors and one can decide which vectors have converged from a look at the j-dimensional vectors $y_i^{(j)}$. So there is a cheap way to determine the set of Ritz vectors against that q_{j+1} has to be orthogonalized. Once more the stable variant of the Simple Lanczos algorithm Table 5.1 will be the basis for the algorithm. Then, this additional orthonormalization will be performed between steps LS(6) and LS(7), before r's norm is computed yielding β_{j+1}. To distinguish the different values of r during an iteration step, r' denotes the original r from Table 5.1 until step LS(6) (inclusively) and r denotes the final form, according to Table 5.3.

$\vdots$	
LS(6):	$r' = r' - \alpha_j q_j$
LS(*):	Intermediate step that generates r from r' by orthogonalizing r' against converged Ritz vectors $r = r' - \sum_{\text{converged } \tilde{\nu}_i^{(j)}} \langle r', \tilde{\nu}_i^{(j)} \rangle \tilde{\nu}_i^{(j)}$
LS(7):	$\beta_{j+1} = \|\|r\|\|$

Table 5.3: Modification of the stable Lanczos algorithm of Table 5.1 and the notation used for r.

The question when a Ritz vector should be considered as converged can be answered in the following way: According to [8], orthogonality among the

Lanczos basis will be maintained at a preset κ-level if r' is orthogonalized against all Ritz vectors with

$$|\cos\angle(\tilde{\nu}_i^{(j)}, r')| \geq \kappa/\sqrt{j} \tag{5.10}$$

because then the inequality holds

$$||Q_j^\dagger q_{j+1}|| = ||Y_j^\dagger q_{j+1}|| < \kappa. \tag{5.11}$$

A computable expression for $\cos\angle(\tilde{\nu}_i^{(j)}, r')$ can easily be derived using Paige's Theorem 5.1.1 (a):

$$\begin{aligned} \langle \tilde{\nu}_i^{(j)}, r' \rangle &= \frac{\gamma_{ij}^{(j)}}{\beta'_{j+1}(y_i^{(j)})_j} \\ &\Rightarrow \; ||\tilde{\nu}_i^{(j)}||\; ||r'|| \cos\angle(\tilde{\nu}_i^{(j)}, r') = \frac{\gamma_{ij}^{(j)}}{\beta'_{ij}} \end{aligned} \tag{5.12}$$

with $\beta'_{j+1} = ||r'||$ and $\beta'_{j+1}(y_i^{(j)})_j \equiv \beta'_{ij}$. The vectors $\tilde{\nu}_i^{(j)}$ and r' stay normalized even in finite precision arithmetics,

$$\Rightarrow \cos\angle(\tilde{\nu}_i^{(j)}, r') = \frac{\gamma_{ij}^{(j)}}{\beta'_{ij}}. \tag{5.13}$$

According to (5.10) the absolute value of $\gamma_{ij}^{(j)}/\beta'_{ij}$ is larger than $\kappa/\sqrt{j}$,

$$\left|\frac{\gamma_{ij}^{(j)}}{\beta'_{ij}}\right| > \frac{\kappa}{\sqrt{j}} \;\Rightarrow\; |\beta'_{ij}| < \frac{|\gamma_{ij}^{(j)}|}{\kappa}\sqrt{j}. \tag{5.14}$$

The $\gamma_{ij}^{(j)}$ are unknown but Equation (5.6) stated that $|\gamma_{ij}^{(j)}| \leq \epsilon||A||$ and so one finds an upper bound for β'_{ij}

$$|\beta'_{ij}| < \frac{\epsilon||A||}{\kappa}\sqrt{j}. \tag{5.15}$$

Equation (5.15) tells which β_{ij} indicate converged Ritz vectors depending on the required degree of orthogonality κ. The next question is for which κ one will obtain useful results. According to [12], if the Lanczos vectors are semiorthogonal, i.e.

$$Q_j^\dagger Q_j = \mathbb{1}_j + E, \quad ||E|| < \sqrt{\epsilon}, \tag{5.16}$$

then for T_j holds

$$T_j = A_j + G, \quad ||G|| = \mathcal{O}(\epsilon||A||) \tag{5.17}$$

where A_j stands for the exact projection onto the exact Krylov space. Therefore, if orthogonality of the Lanczos basis is maintained at $\sqrt{\epsilon}$-level the resulting Ritz values still are the best one can get due to machine precision. So $\kappa = \sqrt{\epsilon}$ is a reasonable choice and Equation (5.15) becomes

$$|\beta'_{ij}| < \sqrt{\epsilon}\sqrt{j}||A||. \tag{5.18}$$

However, this expression is useless unless there is some estimate for $||A|| = \max_i |\xi_i|$.

Some tests with low-dimensional ^{4}He Hamiltonians suggest that the spectral norm is usually of the order of the largest absolute value of the diagonal elements

$$\max_i |a_{ii}| \doteq \max_i |\xi_i|. \tag{5.19}$$

so that one can set

$$\gamma \cdot \max_i |a_{ii}| > \max_i |\xi_i| \tag{5.20}$$

with a number $\gamma < 100$. With this one arrives at the final bound for β'_{ij} that will be used in the calculations

$$|\beta'_{ij}| < \gamma\sqrt{j\epsilon}\ \max_i |a_{ii}| =: \tau(\gamma). \tag{5.21}$$

LSRa :		Lanczos Algorithm with Selective Reorthogonalization Version (a)
LSRa(1):	Start :	Choose r with $\|\|r\|\| = 1$
		Set $\beta_1 := 1$
		Set $q_0 := 0$
	Iterate :	$j = 1, \ldots, m$:
LSRa(2):		$q_j = r/\beta_j$
LSRa(3):		$p_j = Aq_j$
LSRa(4):		$r = p_j - \beta_j q_{j-1}$
LSRa(5):		$\alpha_j = \langle q_j, r \rangle$
LSRa(6):		$r = r - \alpha_j q_j$
LSRa(7):		$\beta_{j+1} = \|\|r\|\|$
LSRa(8) :		Compute all β_{ij}
	For :	$i = 1, \ldots, j$:
	If :	$\beta_{ij} < \tau$
LSRa(9):		Compute $\tilde{\nu}_i$
LSRa(10):		$r := r - \langle r, \tilde{\nu}_i \rangle \tilde{\nu}_i$
	End If	
	End For	
	If :	$\exists\, i : \beta_{ij} < \kappa$
LSRa(11):		$\beta_{j+1} = \|\|r\|\|$
	End If	

Table 5.4: An uneconomic variant of the Lanczos algorithm with selective reorthogonalization.

A first, but still uneconomic, selective reorthogonalization Lanczos algorithm is shown in Table 5.4 and an example for its results is shown in Figure

5.6: As for the full reorthogonalization variant, no copies of Ritz values are generated (Figure 5.6 (a)). From Figure 5.6 (c) one gets confirmation that selective reorthogonalization keeps the Lanczos orthogonality at $\kappa \approx \sqrt{\epsilon}$-level. From Figure 5.6 (b) one can see when the residual of one of the 10 lowest Ritz values falls below τ (black line) so that selective reorthogonalization against the according Ritz vector has to be performed. Finally, Figure 5.6 (d) shows against which Ritz vector r' has been orthogonalized. For this, the Ritz values have been ordered in the way

$$\tilde{\xi}_1^{(j)} < \tilde{\xi}_2^{(j)} < \cdots < \tilde{\xi}_j^{(j)}$$

so that $\tilde{\nu}_j^{(j)}$ denotes the Ritz vector to the highest Ritz value. Thus, Figure 5.6 (d) shows that around iteration 6 the Ritz vector to the highest Ritz value converges but its residual crossing the τ boundary does not appear in (b) because there only the residuals to the lowest Ritz values are plotted.

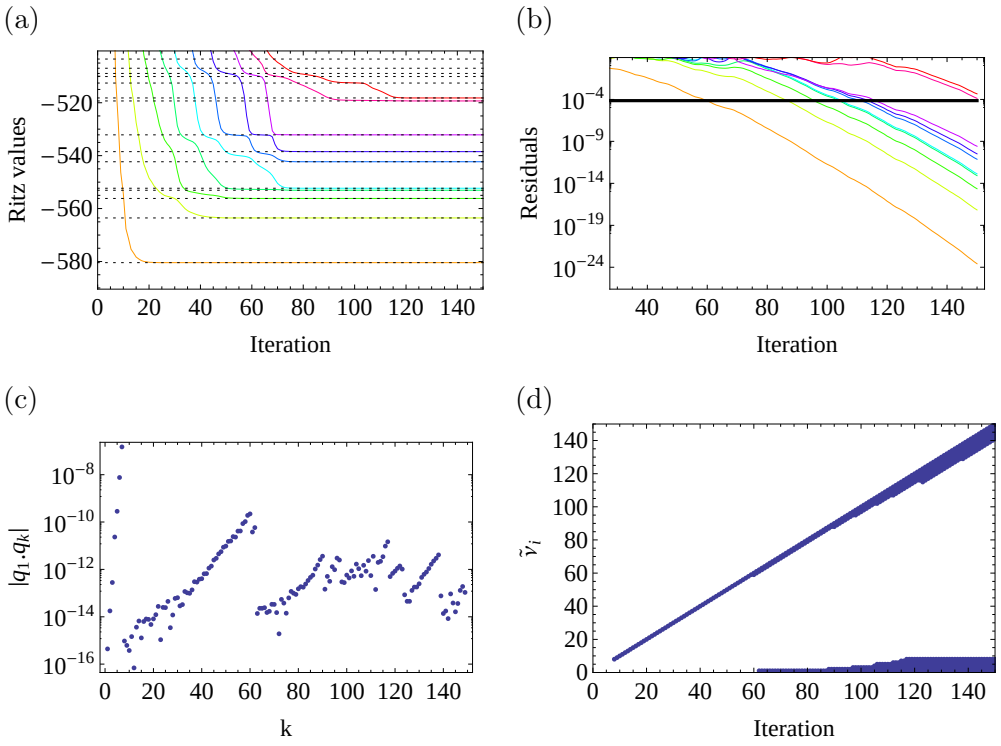


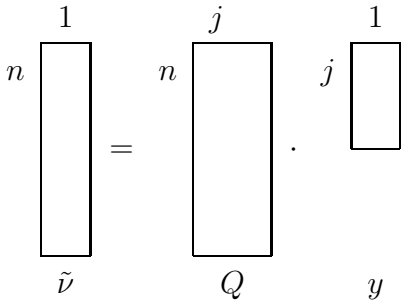
Figure 5.6: *Results of LSRa for A with $a_{ij} = 10^{\cos(ij)}$, $n = 2000$, $q_1 = \mathcal{N}(1, 1, \dots)$: Convergence of Ritz values (a), Residuals for the lowest Ritz values (b), Lanczos orthogonality (c), Ritz vectors against which r' has been orthogonalized (d).*

5.2.4 A more economic Lanczos Algorithm with Selective Reorthogonalization

At a first glance, Figure 5.6 (d) looks like a great improvement over the Lanczos with full reorthogonalization: Instead of orthogonalizing q_{j+1} against j vectors $q_1, \dots, q_j$ in iteration step j, orthogonality can be maintained by orthogonalizing r' against only very few converged Ritz vectors. The drawback is that, unlike the Lanczos vectors that can be brought back from secondary memory when needed, these Ritz vectors have be to computed via $\tilde{\nu}_i^{(j)} = Qy_i^{(j)}$. For this purpose all the Lanczos vectors have to be retrieved back from memory secondary too, and for each Ritz vector a $(n \times j)$ - $(j \times 1)$ matrix-vector multiplication has to be done with the dense matrix Q.

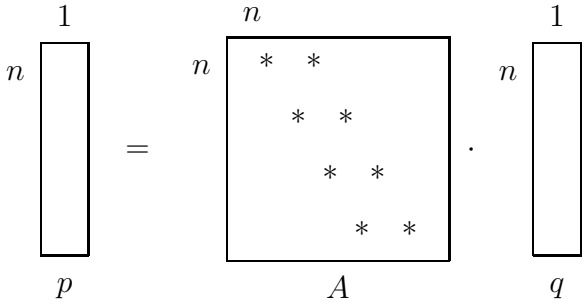
The cost of computing a single Ritz vector (ignoring the cost of retrieving

the Lanczos vectors) is



$$= jn \otimes + jn \oplus = 2jn \text{ flops.} \tag{5.22}$$

This can roughly be compared to the cost of a single Lanczos step (ignoring the cost of retrieving the matrix from memory). The main part is the matrix-vector multiplication Aq_j. If γ is the average number of matrix elements per row then the cost for computing Aq_j is



$$= \gamma n \otimes + \gamma n \oplus = 2\gamma n \text{ flops.} \tag{5.23}$$

For the case of a sparse matrix with $n = 10^6$ and $\gamma = 10^3$ one gets

$$\frac{\text{cost(compute 1 Ritz vector)}}{\text{cost(perform 1 Lanczos step)}} \approx \frac{2jn}{2\gamma n} = \frac{j}{\gamma n} = \begin{cases} 10^{-3} & , j = 1 \\ 1 & , j = 1000. \end{cases}$$

Thus, computing a Ritz vector is not always a cheap task relative to a Lanczos step. Furthermore, the computation of a single Ritz vector is as expensive as a full reorthogonalization in an iteration step. So there is no gain at all in using selective reorthogonalization unless it can be avoided to compute all converged Ritz vectors every iteration step.

Indeed, there is no need in computing all converged Ritz vectors at every step. If a Ritz vector has converged at some iteration, indicated by $\beta_{ij} < \tau$,

it will usually be even more converged in the following iterations, Figure 5.6 (c) and (d), and selective reorthogonalization then demands computing them in all of these iterations. But since these Ritz vectors are converged, they differ only very little, so that they are basically the same. Thus, instead of computing a converged Ritz vector again, a previously computed one can be used. Two problems arise:

1. Since the Ritz basis changes in every iteration step it is not easy to identify Ritz vectors and to assign converged Ritz vectors to their counterparts of a previous iteration.

2. The τ boundary may be too large for a Ritz vector as approximate solution of the eigenproblem to be considered as conveniently converged, especially if τ is larger than the accuracy or the distance between two eigenvalues one is interested in. Then it is surely not recommended to orthogonalize against Ritz vectors with residuals not smaller than $\approx \tau$.

Both problems can be solved by introducing a new boundary $\delta \leq \tau$ and a classification of Ritz vectors:

$$\begin{aligned}
\beta_{ij} > \tau \quad &\Rightarrow \quad \tilde{\nu}_i^{(j)} \quad &&: \text{"unconverged" Ritz vector} \\
\beta_{ij} < \tau \quad &\Rightarrow \quad \tilde{\nu}_i^{(j)} =: \tilde{\nu}_i^{(\text{thres},j)} \quad &&: \text{"threshold" Ritz vector} \\
\beta_{ij} < \delta \quad &\Rightarrow \quad \tilde{\nu}_i^{(j)} =: \tilde{\nu}_i^{\text{good}} \quad &&: \text{"good" Ritz vector}
\end{aligned}$$

A "threshold" Ritz vector is a Ritz vector that is considered converged in the sense of selective reorthogonalization but that is not conveniently converged as an approximation to an eigenvector. Therefore, threshold Ritz vectors have to be computed in every iteration step. A "good" Ritz vector is enough converged to use this vector instead of the following Ritz vectors approximating the same eigenvector. So when a residual indicates the appearance of a good Ritz vector it only has to be computed once more and then stored in secondary memory. This solves the second problem mentioned above but still the identification problem remains.

If a Ritz value that approximates the i-th exact eigenvalue, $\tilde{\xi}_i^{(j)}$, has a residual of $\text{res}(\tilde{\xi}_i^{(j)}) = \delta$ - which is the maximum possible distance to the exact eigenvalue - at some iteration j and smaller residuals in the following

iterations, $\mathrm{res}(\tilde{\xi}_i^{(k>j)}) \leq \delta$, then all these Ritz values will lie in the interval $[\tilde{\xi}_i^{(j)} - 2\delta, \tilde{\xi}_i^{(j)} + 2\delta]$, see Figure 5.7.

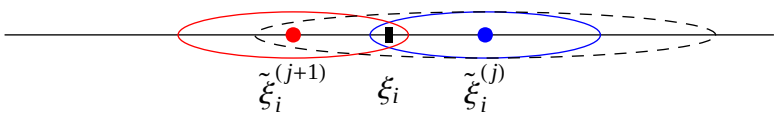


Figure 5.7: *If a Ritz value has a certain residual δ in some iteration j (red) then the Ritz value in the next iteration with smaller residuals (blue) will lie in the interval $[\tilde{\xi}_i^{(j)} - 2\delta, \tilde{\xi}_i^{(j)} + 2\delta]$ (dashed).*

This can be used to identify Ritz vectors as such that approximate certain eigenvalues. If the distance of the eigenvalues (either of all or just of those one is interested in) is larger than 4δ then two Ritz vectors of different iterations can be identified as the Ritz vectors approximating the same eigenvalue if their residuals are smaller than δ and if Ritz value 2 lies within the interval [Ritz value $1 - 2\delta$, Ritz value $1 + 2\delta$]. It is not possible for the Ritz value approximating another exact eigenvalue to lie within this interval if its residual is smaller than δ.

All this considerations lead to a economic variant of the selective reorthogonalization Lanczos as given in Table 5.5. An example of how the number of computed Ritz vectors is diminished is shown in Figure 5.8.

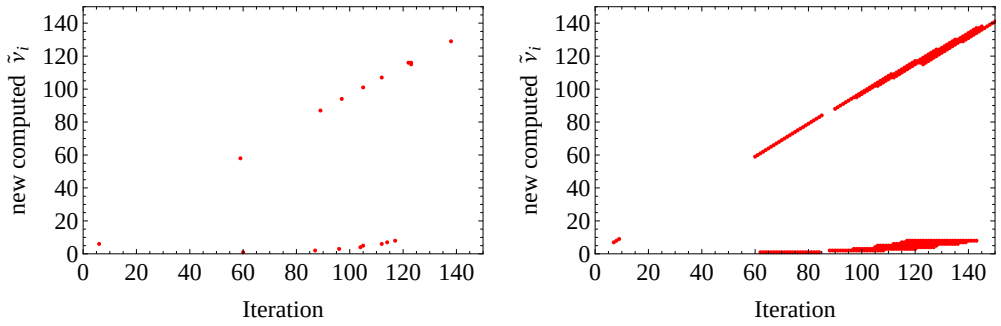


Figure 5.8: *The selective reorthogonalization applied to the example of Figure 5.6. The set of Ritz vectors r' had to be orthogonalized against is the same as Figure 5.6 (d). Shown are the Ritz vectors that had to be computed. Left: $\delta = \tau$. Right: $\delta = 10^{-4}\tau$.*

LSR

Lanczos Algorithm with Selective Reorthogonalization

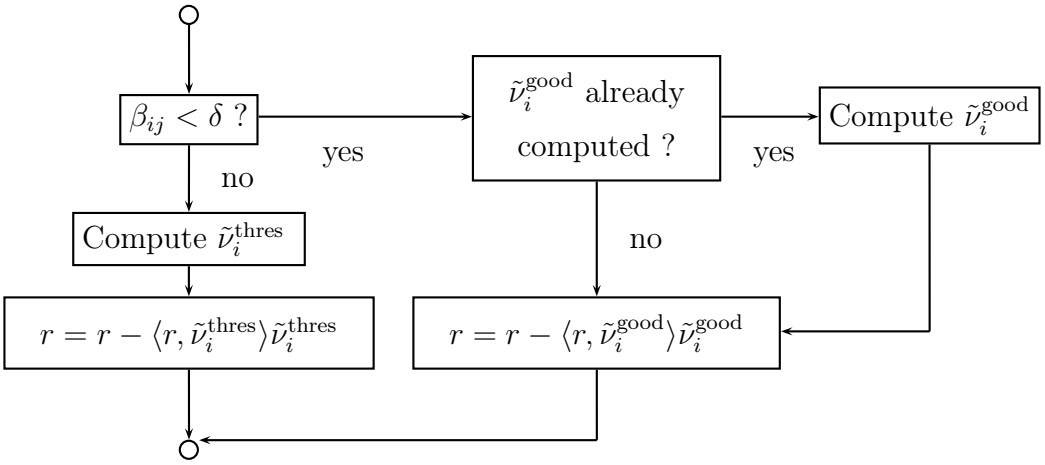
LSR(1-7): Perform : LS(1-7)

LSR(8) : Compute all β_{ij}

For : $i = 1, \ldots, j$:

If : $\beta_{ij} < \tau$

LSR(9) :



End If

End For

If : $\exists\, i : \beta_{ij} < \kappa$

LSR(10): $\beta_{j+1} = ||r||$

End If

Table 5.5: The Lanczos algorithm with selective reorthogonalization.

Chapter 6

Angular Momentum Projection in Shell Model Spaces

The Lanczos algorithm is well suited for an application to nuclear energy calculations in a shell-model framework. This problem can be formulated as the eigenvalue problem of the nuclear Hamiltonian. Usually one is only interested in the ground state energy and a few additional excitations and so the convergence properties of Lanczos algorithm make it the favorable method. However, there is more to be said about the Lanczos algorithm in nuclear structure calculations. Usually, the total angular momentum is a conserved quantity and as a consequence Krylov spaces of the form $\mathcal{K}_m(|j\rangle, H)$, where $|j\rangle$ is some J^2 eigenvector with eigenvalue $j(j+1)$, are subspaces of $\mathcal{J}^{(j)}$. Therefore, when started with $|j\rangle$ as start vector, the Lanczos algorithm will project the Hamiltonian onto the total angular momentum subspace spanned by the Krylov vectors. Thus, the eigenvalue approximations obtained from the algorithm will only belong to energy states with total angular momentum specified by $|j\rangle$. However, in practical computations various error sources will contaminate the angular momentum purity of the Krylov space and the eigenvalue approximations will no longer be constrained to eigenvectors with the desired angular momentum. Reasons for this are rounding errors during the algorithm and a non-conserved total angular momentum. Since the matrix elements of the Hamiltonian and total angular momentum operator are determined numerically, they will lead to slightly non-commuting H and J^2 matrices from the beginning which causes small angular momentum im-

purities. Another reason for a non-conserved total angular momentum is a truncation of the Hilbert space that produces a non-vanishing commutator of the Hamiltonian and total angular momentum, already in the theory. Impurities caused by a Hilbert space truncation are the dominating ones.

From now on, the Dirac Bra-Ket notation will be used, where $|\cdot\rangle$ denotes a vector $\in \mathcal{H}$ and the scalar product reads $\langle\cdot|\cdot\rangle$.

6.1 Subspace Iteration in Exact Arithmetics

Nuclear energies corresponding to Hamiltonian eigenstates with a certain total angular momentum $\hat{j} = j(j+1)$ can be obtained from the Hamiltonian projected onto the subspace of the desired total angular momentum. The Lanczos algorithm provides an elegant way to obtain such a projection.

Exact arithmetics and total angular momentum conservation, i.e.

$$[H, J^2] = 0, \tag{6.1}$$

is assumed. Equation (6.1) implies that there is a simultaneous eigenbasis of H and J^2. As in Section 4.1.1, $\{|E_k^{(j)}\rangle\}$ denote these eigenstates, where $\{|E_k^{(j)}\rangle \ : k = 1, \ldots, k_j\}$ is the subset of all k_j eigenstates with total angular momentum quantum number j and the eigenvalue equations for H and J^2 read

$$H|E_k^{(j)}\rangle = E_k^{(j)}|E_k^{(j)}\rangle \tag{6.2}$$

$$J^2|E_k^{(j)}\rangle = j(j+1)|E_k^{(j)}\rangle. \tag{6.3}$$

An eigenvector $|j\rangle$ of J^2 to the eigenvalue $\hat{j}$ can be expanded in the simultaneous eigenbasis of H and J^2

$$|j\rangle = \sum_{k=1}^{k_j} c_k |E_k^{(j)}\rangle. \tag{6.4}$$

As shown in Section 4.1.1, H acting on this vector will produce another eigenvector to J^2 to the same eigenvalue. As a consequence the Krylov space $\mathcal{K}_m(H, |j\rangle)$ is a total angular momentum subspace of dimension $\min(m, \mu(H, |j\rangle))$. If $|j\rangle$ is used as the start vector for a Lanczos algorithm applied to the Hamilton matrix, the algorithm will compute eigenvalue approximations from exactly this subspace.

An example for such an subspace iteration is shown in Figure 6.1. The Ritz values clearly converge only to eigenvalues of eigenvectors with the total angular momentum of the start vector.

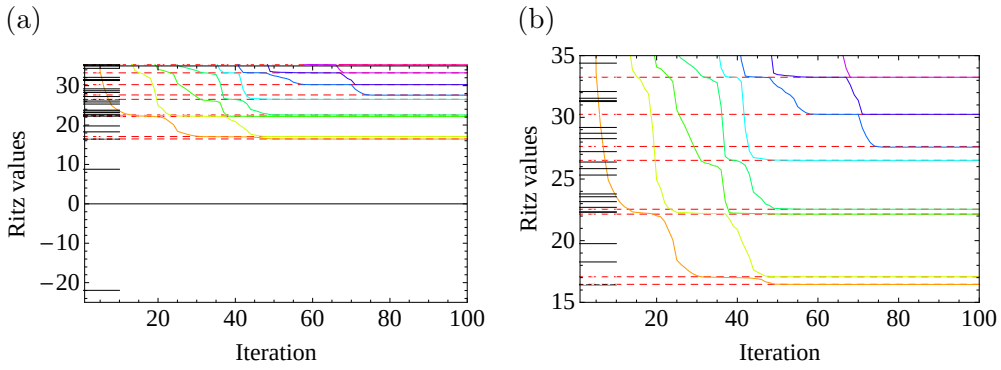


Figure 6.1: *Subspace iteration for a start vector* $|j\rangle$ *with* $\hat{j} = 2$ *and a Hamilton matrix for* ^{4}*He in a* $N_{max} = 4$ *model space (dim* ≈ 1000*). The red (dashed) lines show the positions of the exact energy eigenvalues to eigenvectors with total angular momentum* $\hat{j}$*, the remaining eigenvalues are plotted as black lines at the left margin.*

6.2 Angular Momentum Projection

When using the Lanczos algorithm for a subspace iteration as in the preceding section, the problem of determining the start vector with good total angular momentum remains. Once more the Lanczos algorithm is the method of choice because it allows to exploit the highly degenerated structure of the total angular momentum operator.

6.2.1 Angular Momentum Projection in Exact Arithmetics

In general, when total angular momentum is conserved, the Hilbert space decomposes into a direct sum of subspaces with total angular momentum quantum number $j = 0, 1, \ldots, j_{\max}$,

$$\mathcal{H} = \mathcal{J}^{(0)} \oplus \mathcal{J}^{(1)} \oplus \cdots \oplus \mathcal{J}^{(j)} \oplus \cdots \oplus \mathcal{J}^{(j_{\max})}. \tag{6.5}$$

In the shell model context the $N\hbar\omega$ model space is higly degenerated and so $j_{\max} \ll \dim\ \mathcal{H}$. In the basis (6.2), (6.3) an arbitrary state $|\phi\rangle$ reads

$$|\phi\rangle = \sum_{j=0}^{j_{\max}} \sum_{k=1}^{k_j} \phi_k^{(j)} |E_k^{(j)}\rangle. \tag{6.6}$$

This can be rewritten as a linear combination of single vectors $|j\rangle$ with total angular momenta $\hat{j}$ contained in $|\phi\rangle$,

$$|\phi\rangle = \sum_{j=0}^{j_{\max}} c_j |j\rangle, \tag{6.7}$$

where the $|j\rangle$ are defined as the normalized J^2 eigenstates

$$|j\rangle = \frac{\sum\limits_{k=1}^{k_j} \phi_k^{(j)} |E_k^{(j)}\rangle}{||\sum\limits_{k=1}^{k_j} \phi_k^{(j)} |E_k^{(j)}\rangle||} \tag{6.8}$$

and the expansion coefficients are given by

$$c_j = ||\sum_{k=1}^{k_j} \phi_k^{(j)} |E_k^{(j)}\rangle||. \tag{6.9}$$

Every action of J^2 on (6.7) produces another linear combination of the J^2 eigenvectors $|j\rangle$,

$$|\phi_n\rangle \equiv (J^2)^n |\phi\rangle = \sum_{j=0}^{j_{\max}} c_j \hat{j}^n |j\rangle. \tag{6.10}$$

Since there are at most $j_{\max}+1$ different eigenvectors contained in $|\phi\rangle$, there is a $k \leq j_{\max}+1$ for which the vectors $|\phi_0\rangle, \ldots, |\phi_k\rangle$ span an invariant subspace of $\mathcal{H}$. By construction, this space is the Krylov space $\mathcal{K}_k(J^2, |\phi\rangle)$ for which the Lanczos algorithm breaks down and delivers the exact eigenvalues and, therefore, the start vector with good total angular momentum needed for the subspace iteration for the Hamiltonian. Obviously, the number of different $|j\rangle$ contained in $|\phi\rangle$ can be identified with the degree of the minimal polynomial $\mu(J^2, |\phi\rangle)$.

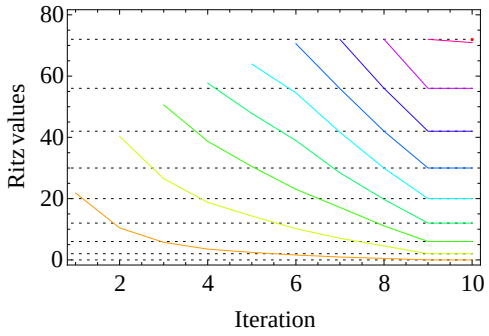


Figure 6.2: *Spanning an invariant total angular momentum subspace under the J^2 matrix of ^6Li in a $N_{max} = 4$ model space. There are 9 different total angular momenta in this space, so after 9 iterations the Ritz values become exact.*

6.2.2 Angular Momentum Projection from Converged Ritz Vectors

In exact arithmetics, it is guaranteed the angular momentum projection procedure from the previous section returns the component with the target angular momentum that is contained in the initial start vector. However, in finite precision arithmetics this is no longer guaranteed. Consider a Hilbert space that contains N different total angular momenta. Numerics can prevent the algorithm to compute the target angular momentum to the desired accuracy after N steps. Then one has to perform more iterations than there are distinct total angular momentum eigenvalues. As a consequence the algorithm starts to compute approximations of different eigenvectors to the same eigenvalue, as it is depicted in Fig. 6.3 for a Hilbert space with $N = 7$. As is can be seen from plots (a) - (d), β_{j+1} and the sum over all residuals $\sum_i \beta_{ij}$ almost vanish for $j = kN, k = 1, 2, \ldots$, when new invariant subspaces are almost spanned in these iterations.

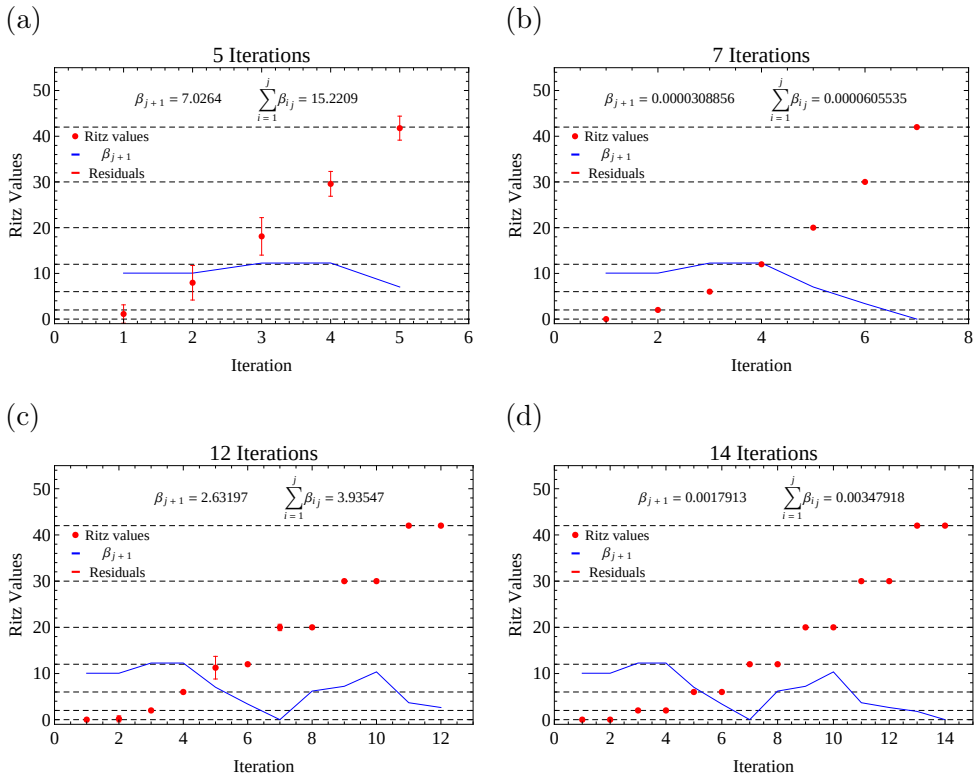


Figure 6.3: *Cyclic spanning of invariant angular momentum subspaces for a Hilbert space with $N = 7$ distinct total angular momenta.*

If an arbitrary start vector for the subspaces iteration is searched for, i.e. the only constraint on the vector is to have target angular momentum, then one can iterate until one of the approximations have converged to the desired accuracy.

The vector obtained this way after $m > N$ iterations will have only little to do with the initial vector given to the J^2 Lanczos. If, as start vector for a H Lanczos, one wants to utilize a precomputed approximation $|\tilde{q}\rangle$ that has to be projected on target angular momentum, the J^2 Lanczos is supposed to return its $\hat{j}_{\text{target}}$ component $|\tilde{q}^{j_{\text{target}}}\rangle$ or else the information from the precomputed approximation is lost. After N iterations one has not perfect, but still fairly good approximations of the angular momentum eigenvectors contained in $|\tilde{q}\rangle$ and so it might be assumed that these vectors will preferentially converge further with the iterations, rather than the newly introduced approximations to the same eigenvalue. Denoting Ritz vectors

whose Ritz values have converged to a certain accuracy to j_{target} as $\{|\tilde{\nu}_i^{j_{\text{target}}}\rangle\}$, this would lead to Ritz vectors $\{|\tilde{\nu}_i^{j_{\text{target}}}\rangle\}$ with one Ritz vector having the maximum overlap with $|\tilde{q}\rangle$ and with vanishing overlaps with all the other Ritz vectors. That this is not the case can be seen from Fig. 6.4, where the bars indicate how much overlap a Ritz vector has with the start vector. Obviously, in general the $|\tilde{q}^{j_{\text{target}}}\rangle$ direction is distributed to all converged Ritz vectors $\{|\tilde{\nu}_i^{j_{\text{target}}}\rangle\}$ and not only to one.

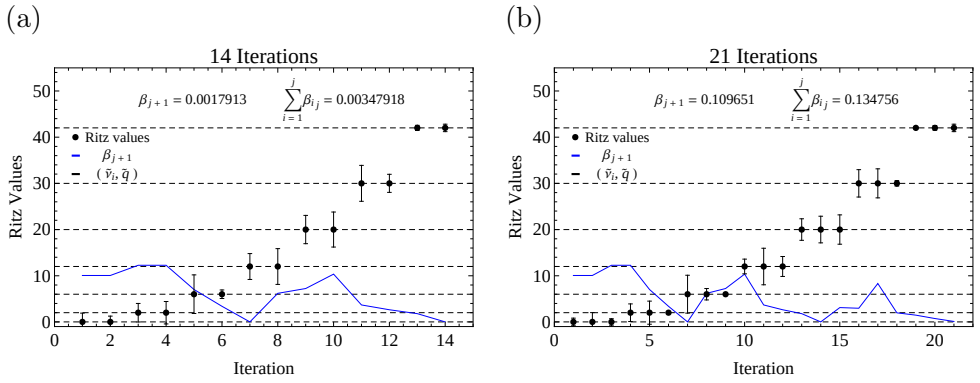


Figure 6.4: *Overlap distribution of the start vector among the converged Ritz vectors.*

A first attempt to extract $|\tilde{q}^{j_{\text{target}}}\rangle$ from these Ritz vectors, under the assumption that $\{|\tilde{q}^{j_{\text{target}}}\rangle\}$ are capable to build $|\tilde{q}^{j_{\text{target}}}\rangle$ good enough, is to find the normalized linear combination $|q\rangle := \sum_i c_i |\tilde{\nu}_i^{j_{\text{target}}}\rangle$ of these Ritz vectors that have maximum overlap with $|\tilde{q}\rangle$, i.e., to maximize the overlap function

$$\sigma(c_1, \ldots, c_k) := \sum_{i=1}^{k} c_i \ \langle \tilde{q} | \tilde{\nu}_i^{j_{\text{target}}} \rangle \stackrel{!}{=} \max, \tag{6.11}$$

over $c_1, \ldots, c_k$ under the constraint (normalization)

$$h(c_1, \ldots, c_k) = \sum_{i=1}^{k} c_i^2 - 1 = 0. \tag{6.12}$$

This can be done in terms of a Lagrange multiplier λ, leading to a system of equations that the coefficients $c_1, \ldots, c_k$ have to obey

$$\begin{cases} \text{grad } \sigma(c_1, \ldots, c_k) + \lambda \text{ grad } h(c_1, \ldots, c_k) &= 0 \\ h(c_1, \ldots, c_k) &= 0. \end{cases} \tag{6.13}$$

Evaluating these expressions gives

$$\begin{cases} \langle\tilde{q}|\tilde{\nu}_1^{j_{\text{target}}}\rangle + 2\lambda\ c_1 &= 0 \\ \qquad\vdots & \\ \langle\tilde{q}|\tilde{\nu}_k^{j_{\text{target}}}\rangle + 2\lambda\ c_k &= 0 \\ \sum_{i=1}^{k} c_i^2 - 1 &= 0 \end{cases} \tag{6.14}$$

where the first k lines are decoupled. Therefore, one finds simple expressions for c_l,

$$c_l = -\langle\tilde{q}|\tilde{\nu}_l^{j_{\text{target}}}\rangle/2\lambda, \tag{6.15}$$

which can be inserted into the last line in order to determine λ,

$$\lambda = \pm\frac{1}{2}\sqrt{\sum_{i=1}^{k}\left(\langle\tilde{q}|\tilde{\nu}_i^{j_{\text{target}}}\rangle\right)^2}. \tag{6.16}$$

The coefficients c_l that maximizes σ are then given by

$$c_l = \text{sign}(\lambda)\frac{\langle\tilde{q}|\tilde{\nu}_l^{j_{\text{target}}}\rangle}{\sqrt{\sum_{i=1}^{k}\left(\langle\tilde{q}|\tilde{\nu}_i^{j_{\text{target}}}\rangle\right)^2}} \tag{6.17}$$

and the optimal result $|q\rangle$ of the angular momentum projection becomes

$$|q\rangle = \zeta\sum_{l=1}^{k}\frac{\langle\tilde{q}|\tilde{\nu}_l^{j_{\text{target}}}\rangle}{\sqrt{\sum_{i=1}^{k}\left(\langle\tilde{q}|\tilde{\nu}_i^{j_{\text{target}}}\rangle\right)^2}}|\tilde{\nu}_l^{j_{\text{target}}}\rangle, \tag{6.18}$$

where $\zeta = \pm 1$ denotes the sign for that $\langle\tilde{q}|q\rangle$ becomes positive. The result (6.18) is not really surprising and in fact it could have been written down directly without explicitly maximizing the overlap, remembering that angular momentum subspaces are orthogonal and the well-known fact that in Hilbert spaces the overlaps of an orthonormal system with a vector lead to the best approximation within the space spanned by the orthonormal system.

Proposition 6.2.1

Let (e_i) *be an orthonormal system within the Hilbert space* $\mathcal{H}$ *over* $\mathbb{K} = \mathbb{R}$ *or* $\mathbb{K} = \mathbb{C}$ *and* $x \in \mathcal{H}$*. Then,* $\forall c_1, \ldots, c_n \in \mathbb{K}$

$$\left\| x - \sum_{i=1}^{n} c_i e_i \right\| \geq \left\| x - \sum_{i=1}^{n} \langle x, e_i \rangle e_i \right\| .$$

Consequently, in order to find the best approximation of $|\tilde{q}_j^{j_{\text{target}}}\rangle$, instead of maximizing the overlap function σ one can directly use $\langle \tilde{q}_j | \tilde{\nu}_l^{j_{\text{target}}} \rangle$ as expansion coefficients and normalize the resulting $|q_j\rangle$ afterwards. The reason why this approach fails in practice can be seen in Fig. 6.5.

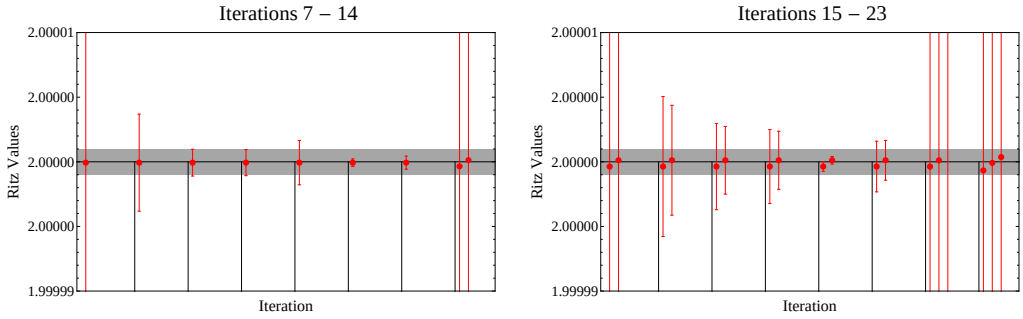


Figure 6.5: *Eigenvalue and residual evolution during a* J^2 *Lanczos in a* $N = 7$ *Hilbert space.*

A Ritz vector $|\tilde{\nu}_i\rangle$ is usually considered as converged to the eigenvalue $\hat{j}$ if $\tilde{\xi}_i \pm \text{res}(\tilde{\xi}_i)$ lies in a certain interval around the eigenvalue $\hat{j}$. But due to the total angular momentum degeneracy a converged Ritz vector in some iteration may no longer be considered converged in a later iteration. Thus, in general one never has enough converged Ritz vectors at hand for a proper construction of $|\tilde{q}_j^{j_{\text{target}}}\rangle$.

6.2.3 Angular Momentum Projection from Arbitrary Bases

Section 6.2.2 already suggests that it is difficult to extract an angular momentum component from a given vector more accurately than the Lanczos algorithm does in the iteration when the invariant subspace is approximately spanned, even if one does many additional iteration steps. Thus, in practice

when confronted with large systems where matrix-vector multiplications represent the upper end one's computational abilities, one will have to live with this accuracy. Nevertheless, a different way that does not depend on converged Ritz vectors, and that at a first glance could be capable to overcome some problems encountered in truncated Hilbert spaces (but is not), shall at least be presented although it neither will be applied in the following nor its practical value is further investigated.

If the Lanczos algorithm spans the k-dimensional invariant subspace of the vector $|\tilde{q}\rangle$ not accurately enough accurate, then one can perform more Lanczos steps or try to compute this invariant subspace a second (third etc.) time a different way and orthonormalize the basis vectors, obtaining a n-dimensional basis $\{|q_j\rangle : j = 1, \ldots, n\}$. It has to be assumed that in this basis all the k angular momentum components $|\tilde{q}_i^{(j)}\rangle$ of $|\tilde{q}\rangle$ can accurately be approximated by linear combinations $|k_i\rangle$, so that there exists a set of coefficients

$$\{\xi_l^{(i)} \ : i = 1, \ldots, n, \ l = 1, \ldots, k\} \tag{6.19}$$

with

$$|\tilde{q}^{(j_i)}\rangle \equiv |k_i\rangle = \sum_{l=1}^{n} \xi_l^{(i)} |q_l\rangle, \quad i = 1, \ldots, k, \tag{6.20}$$

where the $\xi_l^{(i)}$ obey the normalization conditions

$$\sum_{l=1}^{n} (\xi_l^{(i)})^2 - 1 = 0, \quad i = 1, \ldots, k. \tag{6.21}$$

From these $|k_i\rangle$ the approximation $|q\rangle$ for $|\tilde{q}\rangle$ is constructed with coefficients $c_i, i = 1, \ldots, k$,

$$|q\rangle = \sum_{i=1}^{k} c_i |k_i\rangle, \tag{6.22}$$

for which the normalization condition reads

$$\sum_{m=1}^{k} c_m^2 - 1 = 0. \tag{6.23}$$

Thus, linear combinations (6.20) and (6.22) are sought that have the following properties:

1. $|q\rangle$ maximizes the overlap with $|\tilde{q}\rangle$,

$$\begin{aligned} |\langle q|\tilde{q}\rangle| &= |\sum_{m=1}^{n} c_m \langle q|k_m\rangle| = \sigma_{\max} \\ |\langle q|\tilde{q}\rangle| &= |\sum_{m=1}^{n}\sum_{l=1}^{n} c_m \xi_l^{(m)} \langle q|q_l\rangle| = \sigma_{\max} \end{aligned} \tag{6.24}$$

with $\sigma_{\max} = 1$ under the assumption made above. It is convenient to formulate this equation for $\xi_l^{(i)}$ as a minimization problem

$$\begin{aligned} \sigma(\vec{\xi}, \vec{c}) &:= \sigma(\xi_1^{(1)}, \ldots, \xi_n^{(1)}, \xi_1^{(2)}, \ldots, \xi_n^{(2)}, \ldots, \xi_1^{(k)}, \ldots, \xi_n^{(k)}, c_1, \ldots, c_k) \\ &= |\sum_{m=1}^{k}\sum_{l=1}^{n} c_m \xi_l^{(m)} \langle q|q_l\rangle - \sigma_{\max}| \to \min = 0 \end{aligned} \tag{6.25}$$

where the minimum is actually known to be 0. This assures the $\{|k_i\rangle\}$ to be a set of k vectors from which $|\tilde{q}\rangle$ can be reconstructed. This does by no means assure $|k_i\rangle$ to be the i-th angular momentum component of $|\tilde{q}\rangle$.

2. Therefore, the additional k constraints that all $|k_m\rangle$ are J^2 eigenvectors to the eigenvalue $\hat{m} = m(m+1)$ have to be imposed, again as a minimization problem for the $\xi_l^{(i)}$ with known minimum 0

$$\begin{aligned} \gamma_m(\xi_1^{(m)}, \ldots, \xi_n^{(m)}) &:= ||J^2|k_m\rangle - \hat{m}|k_m\rangle|| \\ &= ||\sum_{l=1}^{n} \xi_l^{(m)} \left(|p_l\rangle - \hat{m}|q_l\rangle\right)|| \\ &\to \min = 0, \quad i = 1, \ldots, k. \end{aligned} \tag{6.26}$$

Here the abbreviation $|p_l\rangle := J^2|q_l\rangle$ was introduced for quantities that already have been computed if the basis $|q_j\rangle$ was obtained from the Lanczos algorithm.

Minimizing $k+1$ coupled functions simultaneously is a hard task, and so, using the fact that all minima are equal to 0, they are condensed into a single

function η to be minimized,

$$\begin{aligned}\eta(\vec{\xi},\vec{c}) &:= \eta(\xi_1^{(1)},\ldots,\xi_n^{(1)},\xi_1^{(2)},\ldots,\xi_n^{(2)},\ldots,\xi_1^{(k)},\ldots,\xi_n^{(k)},c_1,\ldots,c_k)\\ &:= \left[\sum_{m=1}^{k}\gamma_m(\xi_1^{(m)},\ldots,\xi_n^{(m)})\right]+\sigma(\vec{\xi},\vec{c})\\ &\to \min=0. \end{aligned} \tag{6.27}$$

However, η contains a modulus of σ and several vector norms of γ_m that are not differentiable in the minimum. Therefore, the squares of these functions (for which the minimization conditions remain unchanged) are used instead which makes the modulus and norms obsolete. So one minimizes the new function η',

$$\eta'(\vec{\xi},\vec{c}) := \sum_{m=1}^{k}\left[\gamma_m^2(\xi_1^{(m)},\ldots,\xi_n^{(m)})\right]+\sigma^2(\vec{\xi},\vec{c})\to\min=0,$$

with

$$\begin{aligned}\gamma_m^2(\xi_1^{(m)},\ldots,\xi_n^{(m)}) &= ||\sum_{l=1}^{n}\xi_l^{(m)}\left(|p_l\rangle-\hat{m}|q_l\rangle\right)||^2\\ &= \sum_{l=1}^{n}\sum_{l'=1}^{n}\xi_l^{(m)}\xi_{l'}^{(m)}\Big\{\langle p_l|p_{l'}\rangle-\hat{m}\langle p_l|q_{l'}\rangle-\hat{m}\langle q_l|p_{l'}\rangle+\hat{m}^2\delta_{ll'}\Big\}. \end{aligned} \tag{6.28}$$

From Hermitecity of J^2 follows that $\langle p_l|q_{l'}\rangle=\langle q_l|p_{l'}\rangle$ and so one obtains

$$\gamma_m^2(\xi_1^{(m)},\ldots,\xi_n^{(m)}) = \sum_{l=1}^{n}\sum_{l'=1}^{n}\xi_l^{(m)}\chi_{ll'}^{(m)}\xi_{l'}^{(m)} \tag{6.29}$$

with the symmetric matrix $\chi_{ll'}^{(m)}$ given by

$$\chi_{ll'}^{(m)} := \langle p_l|p_{l'}\rangle-2\hat{m}\langle q_l|p_{l'}\rangle+\hat{m}^2\delta_{ll'}. \tag{6.30}$$

With this, one arrives at

$$\eta'(\vec{\xi},\vec{c}) = \sum_{m=1}^{k}\Big[\sum_{l=1}^{n}\sum_{l'=1}^{n}\xi_l^{(m)}\chi_{ll'}^{(m)}\xi_{l'}^{(m)}\Big]+\Big[\sum_{m=1}^{k}\sum_{l=1}^{n}c_m\xi_l^{(m)}\langle q|q_l\rangle-\sigma_{\max}\Big]^2 \tag{6.31}$$

which is to be minimized under the constraints

$$\phi(c_1,\ldots,c_k) \;:=\; \sum_{m=1}^{k} c_m^2 - 1 = 0 \tag{6.32}$$

$$\phi_i(\xi_1^{(i)},\ldots,\xi_n^{(i)}) \;:=\; \sum_{l=1}^{n} (\xi_l^{(i)})^2 - 1 = 0, \quad i = 1,\ldots,k. \tag{6.33}$$

Using Lagrange multipliers this leads to a system of equations f,

$$f(\vec{\xi},\vec{c},\vec{\lambda}) := \begin{cases} 0 = \text{grad } \eta' + \lambda \text{ grad } \phi + \sum_{i=1}^{k} \lambda_i \text{ grad } \phi_i \\ 0 = \phi \\ 0 = \phi_1 \\ \vdots \\ 0 = \phi_k \end{cases} . \tag{6.34}$$

To write this down explicitly one needs the gradient

$$\text{grad} = \Big(\frac{\partial}{\partial \xi_1^{(1)}},\ldots,\frac{\partial}{\partial \xi_n^{(k)}},\frac{\partial}{\partial c_1},\ldots,\frac{\partial}{\partial c_k}\Big)^T \tag{6.35}$$

of the function

$$\begin{aligned} \frac{\partial}{\partial \xi_j^{(i)}}\eta' &= 2\sum_{l=1}^{n} \xi_l^{(i)} \chi_{lj}^{(i)} + 2c_i\langle q|q_j\rangle \left[\sum_{m=1}^{k}\sum_{l=1}^{n} c_m \xi_l^{(m)}\langle q|q_l\rangle - \sigma_{\max}\right] \\ \frac{\partial}{\partial c_i}\eta' &= 2\left[\sum_{m=1}^{k}\sum_{l=1}^{n} c_m \xi_l^{(m)}\langle q|q_l\rangle - \sigma_{\max}\right]\left[\sum_{l=1}^{n} \xi_l^{(i)}\langle q|q_l\rangle\right] \end{aligned} \tag{6.36}$$

and the constraints

$$\begin{aligned} &\tfrac{\partial}{\partial c_i}\phi_m = 0 \qquad && \tfrac{\partial}{\partial \xi_j^{(i)}}\phi_m = \delta_{im} 2\xi_j^{(i)} \\ &\tfrac{\partial}{\partial c_i}\phi = 2c_i && \tfrac{\partial}{\partial \xi_j^{(i)}}\phi = 0, \end{aligned} \tag{6.37}$$

leading to

$$f_\alpha(\vec{\xi},\vec{c},\vec{\lambda}) = \begin{cases} 0 = 2\sum_{l=1}^{n} \xi^{(i)}\chi_{lj}^{(i)} + 2\Gamma_2 c_i\langle q|q_j\rangle + 2\lambda_i\xi_j^{(i)} \\ \qquad\qquad : \alpha \in [1,kn], \quad j = \mathrm{mod}[\alpha,n], \quad i = (\alpha-j)/n \\ 0 = 2\Gamma_2\Gamma_3^{(i)} + 2\lambda c_i \quad : \alpha \in [kn+1, k(n+1)], \quad i = \alpha - kn \\ 0 = \sum_{m=1}^{k} c_m^2 - 1 \qquad : \alpha = k(n+1)+1 \\ 0 = \sum_{l=1}^{n} (\xi_l^{(i)})^2 - 1 \quad : \alpha \in [k(n+1)+2, k(n+2)+1] \\ \qquad\qquad\qquad\qquad i = \alpha - k(n+1) - 1 \end{cases} \tag{6.38}$$

with the abbreviations

$$\Gamma_1^{(m)}(\xi_1^{(m)},\ldots,\xi_n^{(m)}) = \sum_{l=1}^{k}\sum_{l'=1}^{n} \xi_l^{(m)}\chi_{ll'}^{(m)}\xi_{l'}^{(m)}, \ m = 1,\ldots,k \tag{6.39}$$

$$\Gamma_2(\vec{\xi}) = \sum_{m=1}^{k}\sum_{l=1}^{n} c_m\xi_l^{(m)}\langle q|q_l\rangle - \sigma_{\max} \tag{6.40}$$

$$\Gamma_3^{(m)}(\xi_1^{(m)},\ldots,\xi_n^{(m)}) = \sum_{l=1}^{n} \xi_l^{(m)}\langle q|q_l\rangle, \ m = 1,\ldots,k. \tag{6.41}$$

f is a non-linear system of equations and has, therefore, to be solved iteratively. For this purpose the multidimensional Newton method is employed, for the following reasons: The system is only weakly non-linear and, therefore, one expects good convergence by the linear approximations done in the Newton iteration. Secondly, one needs the system and the Hesse matrix for which explicit expressions can be found. Furthermore, does the success of the Newton method depend on good start values for which one can find good estimates, as it is discussed below.

The Hesse matrix is defined as

$$f'(\vec{\xi},\vec{c},\vec{\lambda}) = \begin{pmatrix} \frac{\partial f_1}{\partial \xi_1^{(1)}} & \cdots & \frac{\partial f_1}{\partial c_k} & \frac{\partial f_1}{\partial \lambda} & \frac{\partial f_1}{\partial \lambda_1} & \cdots & \frac{\partial f_1}{\partial \lambda_k} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ \frac{\partial f_{(n+1)k}}{\partial \xi_1^{(1)}} & \cdots & \frac{\partial f_{(n+1)k}}{\partial c_k} & \frac{\partial f_{(n+1)k}}{\partial \lambda} & \frac{\partial f_{(n+1)k}}{\partial \lambda_1} & \cdots & \frac{\partial f_{(n+1)k}}{\partial \lambda_k} \end{pmatrix}$$

(6.42)

and its matrix elements follow from the form of the system (6.38) and the second order derivatives

$$\frac{\partial}{\partial \xi_{j'}^{(i')}}\left[\frac{\partial}{\partial \xi_j^{(i)}}\eta'\right] = 2\delta_{i'i}\chi_{j'j}^{(i')} + 2c_{i'}c_i\langle q|q_{j'}\rangle\langle q|q_j\rangle \tag{6.43}$$

$$\begin{aligned}\frac{\partial}{\partial c_{i'}}\left[\frac{\partial}{\partial \xi_j^{(i)}}\eta'\right] &= 2c_i\langle q|q_j\rangle\sum_{l=1}^{n}\xi_l^{(i')}\langle q|q_l\rangle \\ &+ 2\delta_{i'i}\langle q|q_j\rangle\left[\sum_{m=1}^{k}\sum_{l=1}^{n}c_m\xi_l^{(m)}\langle q|q_l\rangle - \sigma_{\max}\right]\end{aligned} \tag{6.44}$$

$$\begin{aligned}\frac{\partial}{\partial \xi_{j'}^{(i')}}\left[\frac{\partial}{\partial c_i}\eta'\right] &= 2c_{i'}\langle q|q_{j'}\rangle\sum_{l=1}^{n}\xi_l^{(i)}\langle q|q_l\rangle \\ &+2\delta_{i'i}\langle q|q_{j'}\rangle\left[\sum_{m=1}^{k}\sum_{l=1}^{n}c_m\xi_l^{(m)}\langle q|q_l\rangle - \sigma_{\max}\right].\end{aligned} \tag{6.45}$$

$$\frac{\partial}{\partial c_{i'}}\left[\frac{\partial}{\partial c_i}\eta'\right] = 2\left[\sum_{l=1}^{n}\xi_l^{(i')}\langle q|q_l\rangle\right]\left[\sum_{l=1}^{n}\xi_l^{(i)}\langle q|q_l\rangle\right] \tag{6.46}$$

$$\begin{array}{ccccccc}\frac{\partial}{\partial c_{i'}}\left[\frac{\partial}{\partial c_i}\phi_m\right] & = & \frac{\partial}{\partial c_{i'}}\left[\frac{\partial}{\partial \xi_j^{(i)}}\phi_m\right] & = & \frac{\partial}{\partial \xi_{j'}^{(i')}}\left[\frac{\partial}{\partial c_i}\phi_m\right] & = & 0 \\ \frac{\partial}{\partial c_{i'}}\left[\frac{\partial}{\partial \xi_j^{(i)}}\phi\right] & = & \frac{\partial}{\partial \xi_{j'}^{(i')}}\left[\frac{\partial}{\partial c_i}\phi\right] & = & \frac{\partial}{\partial \xi_{j'}^{(i')}}\left[\frac{\partial}{\partial \xi_j^{(i)}}\phi\right] & = & 0 \\ \frac{\partial}{\partial \xi_{j'}^{(i')}}\left[\frac{\partial}{\partial \xi_j^{(i)}}\phi_m\right] = 2\delta_{im}\delta_{i'i}\delta_{j'j} & & & & \frac{\partial}{\partial c_{i'}}\left[\frac{\partial}{\partial c_i}\phi\right] = 2\delta_{i'i} & & \end{array} \tag{6.47}$$

and trivial derivatives with respect to the Lagrange multipliers.

To find a good starting point for the Newton iteration one needs a first estimate for the coefficients $\xi_j^{(i)}$ and c_i. As in Fig. 6.6 (a), the Ritz values will more or less cluster around the exact eigenvalues. So one can take all Ritz vectors of a certain cluster and compute from them a single vector that has the maximum overlap with $|\tilde{q}\rangle$ and take this as first approximation of a angular momentum component of $|k_i\rangle$, from which the coefficient estimates

$$\xi_j^{(i)} = \mathcal{N}\sum_{l\in\mathrm{conv}(i)}\sum_{m=1}y_m^{(l)}y_j^{(l)}\langle q|q_m\rangle \tag{6.48}$$

follow, where $\mathcal{N}$ is the normalization, y denote the Hessenberg eigenvectors and conv(i) denotes the set of indices of Ritz vectors belonging to a certain cluster. From the first approximations of $|k_i\rangle$ again the linear combination with maximum overlap with $|\tilde{q}\rangle$ can be computed, leading to a first estimate for the coefficients c_i

$$c_j = \sum_{i=1}^{n} \xi_i^{(j)} \langle q|q_i\rangle. \tag{6.49}$$

This has been done in Fig. 6.6. Plots (a) and (b) show the Ritz values with their residuals and the overlaps of the corresponding Ritz vectors with a randomly chosen $|q\rangle$. Plots (c) and (d) illustrate the convergence during the Newton iteration of η' and η as well as for $\Gamma_1^{(2)}$ which is equal to γ_2^2, and therefore, measures the angular momentum deviation of $|k_2\rangle$. $\sqrt{\Gamma_1^{(2)}}$ (and so does η) does not become smaller than 10^{-7} which can be understood from Fig. 6.6 (f), where the scattering of the exact eigenvalues of the given J^2 matrix is shown around the value $\hat{j}_i = 2$, which lies in the same order. That the method is, in principle, capable to extract all angular momentum components of a given vector $|\tilde{q}\rangle$ is shown in (e), where the overlaps of the $|k_i\rangle$ with $|\tilde{q}\rangle$ (red) is compared with the exact distribution (blue) of angular momentum in the initial vector $|\tilde{q}\rangle$.

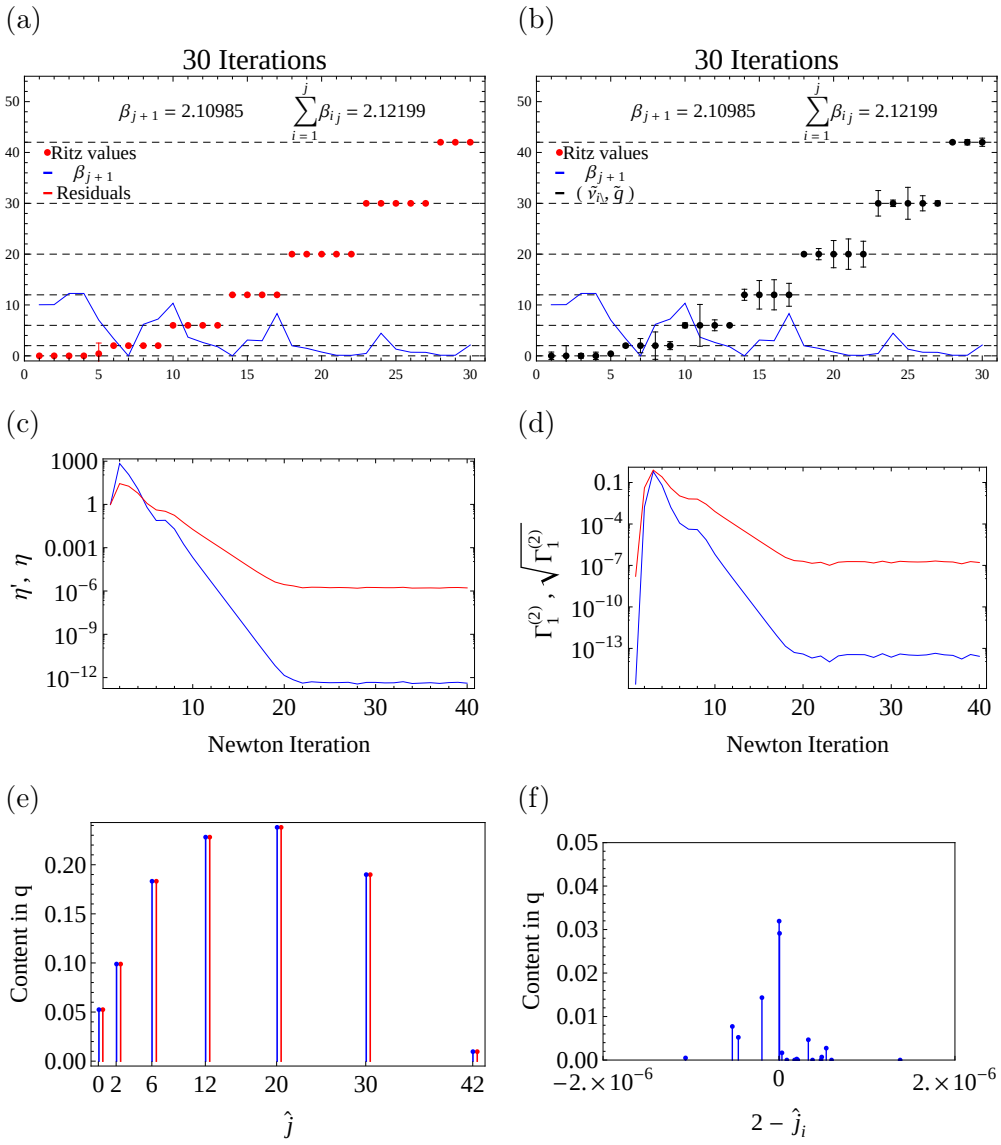


Figure 6.6: *Example for the angular momentum projection of a vector $|\tilde{q}\rangle$ using an arbitrary, not necessarily converged basis. (a) and (b) display the properties of the basis $|q_l\rangle$. (c) and (d) depict the convergence of the η and $\Gamma^{(2)}$ functions during the Newton iterations. (e) compares the overlaps of $|\tilde{q}\rangle$ with the original angular momentum components $|\tilde{q}^{(j_i)}\rangle$ contained in it (red) with the overlaps of $|\tilde{q}\rangle$ with reconstructed angular momentum components $|k_i\rangle$ (blue). (f) shows the scattering of the exact J^2 matrix eigenvalues around the integer value 2, from which the final results for η and $\Gamma^{(2)}$ in the Newton iteration can be understood.*

6.3 Subspace Iteration

This section regards the Lanczos iteration in a total angular momentum subspace of an $N\hbar\omega$ shell model space under realistic numerical conditions, i.e., in the presence of rounding errors, leading to

$$[H, J^2]\tilde{1}0^{-8}. \tag{6.50}$$

One may try to compute a couple of eigenvalues within the total angular momentum subspace simultaneously, as it was done in Fig. 6.1. However, when starting with a vector with good angular momentum, the Lanczos vectors lose the good angular momentum rapidly over the iterations, such that eigenvalues from other angular momentum subspaces are obtained eventually, see Fig. 6.7.

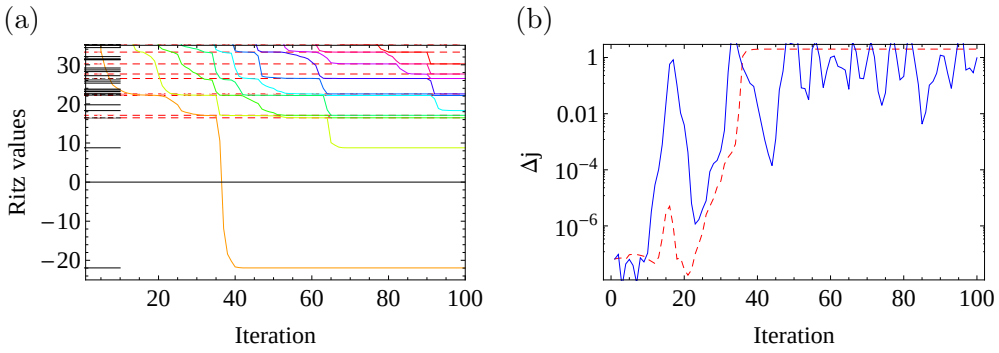


Figure 6.7: *(a) : Finite precision results of Fig. 6.1. (b) : Total angular momentum deviation of the last Lanczos (blue, solid) and first Ritz vector (red, dashed).*

Usually the Lanczos vectors lose their good angular momentum faster than their orthogonality, even if there is no reorthogonalization employed. In fact, a reorthogonalization does neither prevent the Lanczos vectors from losing their good angular momentum nor decelerate this process.

One way to deal with this problem is to correct the angular momentum of a new Lanczos vector in every iteration step, analog to its orthogonality. This can be achieved in the same way the start vectors are generated, using the new Lanczos vector as starting point for the angular momentum projection. If the angular momentum deviation of the Lanczos vector is sufficiently small, then

it has a strong component into the direction of the target angular momentum and the projection will generate the angular momentum corrected version of the Lanczos vector quickly after two or three iterations.

In practice this approach to compute more than one eigenvalue simultaneously fails. Let $\{|q_i\rangle\}$ denote the Lanczos basis that is orthogonal and has good total angular momentum. When a new Lanczos vector $|\tilde{q}_j\rangle$ is generated in LFR(2) (Table. 5.1) in iteration j, then this vector may have no good angular momentum and so it is rotated by the angular momentum projection into the correct direction in the angular momentum space, yielding the purified Lanczos vector $|q_j\rangle$ which is then stored in the Lanczos basis and is further used by the algorithm. The correction $|\Delta\tilde{q}_j\rangle$ is then given by

$$|q_j\rangle = |\tilde{q}_j\rangle + |\Delta\tilde{q}_j\rangle. \tag{6.51}$$

The algorithm given in Table. 5.1, for example, is only valid if $|\tilde{q}_j\rangle$ is used as Lanczos vector, but after the projection one works with $|q_j\rangle$ instead of $|\tilde{q}_j\rangle$. Therefore, in principle, corrections have to be introduced to cover the usage of the 'wrong' Lanczos vector. Then, for example, LFR(3) becomes

$$|p_j\rangle = A|q_j\rangle - A|\Delta\tilde{q}_j\rangle \tag{6.52}$$

and the elements of the Hessenberg matrix are determined by

$$\alpha_j = \langle q_j|r\rangle - 2\langle q_j|A|\Delta\tilde{q}_j\rangle + \langle\Delta\tilde{q}_j|A|\Delta\tilde{q}_j\rangle + \beta_j\langle\Delta\tilde{q}_j|A|q_{j-1}\rangle \tag{6.53}$$

and

$$\begin{aligned}\beta_{j+1} = \; || \; r - A|\Delta\tilde{q}_j\rangle + \Big(\langle q_j| + \langle\Delta\tilde{q}_j|\Big)A|\Delta\tilde{q}_j\rangle \\ - \beta_j\langle q_j|q_{j-1}\rangle|\Delta\tilde{q}_j\rangle - \beta_j\langle\Delta\tilde{q}_j|q_{j-1}\rangle\Big(|q_j\rangle + |\Delta\tilde{q}_j\rangle\Big) \; ||.\end{aligned} \tag{6.54}$$

Since these corrections are obviously not practicable and, therefore, worthless in actual computations, the algorithm will be employed in the usual way, leading to perturbations of the Hessenberg matrix.

The effects of such perturbations can be seen in Fig. 6.8, where a start vector with (poor) angular momentum accuracy of 10^{-4} was generated and the angular momentum deviation of all Lanczos vectors has been kept at the same level. From

$$\langle\nu_i|H|\nu_i\rangle = \langle y_i|Q_m^T H Q_m|y_i\rangle = \langle y_i|H_m|y_i\rangle = \theta_i \tag{6.55}$$

follows that the energy expectation value $\langle\nu_i|H|\nu_i\rangle$ of a Ritz vector ν_i equals its corresponding Ritz value. Fig. 6.8 (a) however shows significant deviations of the expectation value of the first Ritz vector (black dots) from the lowest eigenvalue of the Hessenberg matrix. This deviation is independent from a possible loss of good angular momentum of the Ritz vector or the Lanczos basis since (6.55) is of course not restricted to a particular Lanczos basis Q_m. Since the components of the eigenvectors of H_m are the expansion coefficients of the Ritz vectors in the Lanczos basis the deviation of the expectation and Ritz value is a manifestation of the losing connection of the Hessenberg matrix elements and the Lanczos basis.It should be noted that this is not caused by the projection returning a completely different purified vector than that one actually wanted. This can be seen from the difference of the original $|\tilde{q}_j\rangle$ and the purified $|q_j\rangle$, measured by their overlap, is at the level of the angular momentum accuracy, see Fig. 6.8 (c). Because there is no way to avoid the decoupling of α_j, β_j and the Lanczos basis, one has to stop the algorithm at some point in order to perform a restart. As a restart condition could be formulated in terms of a maximum deviation δ_{restart} of the Ritz and expectation value for the first Ritz vector,

$$\text{if} \quad |\langle\nu_1|H|\nu_1\rangle - \theta_1| > \delta_{\text{restart}} \quad \Rightarrow \quad \text{restart}. \tag{6.56}$$

Another reason for the necessity of a restart is the loss of good angular momentum. Fig. 6.8 (b) shows that although the Lanczos vectors are kept at an angular momentum deviation level around 10^{-4} the deviation of the first Ritz vector grows fast until some convergence occurs. This is not surprising since in the Ritz vector the deviations of the Lanczos vectors accumulate weighted with the elements of the corresponding Hessenberg eigenvector. One may suspect the loss of good angular momentum rather than the perturbations of α_j and β_j to be the reason for the deviating Ritz expectation values (disregarding the general argument above that this deviation has nothing to with the angular momenta of the basis vectors). That this is indeed not the case can be seen from Fig. 6.8 (d) where the blue lines at the right margin are the lowest 2 eigenvalues of the real projected matrix H_m, i.e., the matrix that has been computed by actually performing the matrix multiplications $H_m = Q_m^T H Q_m$.

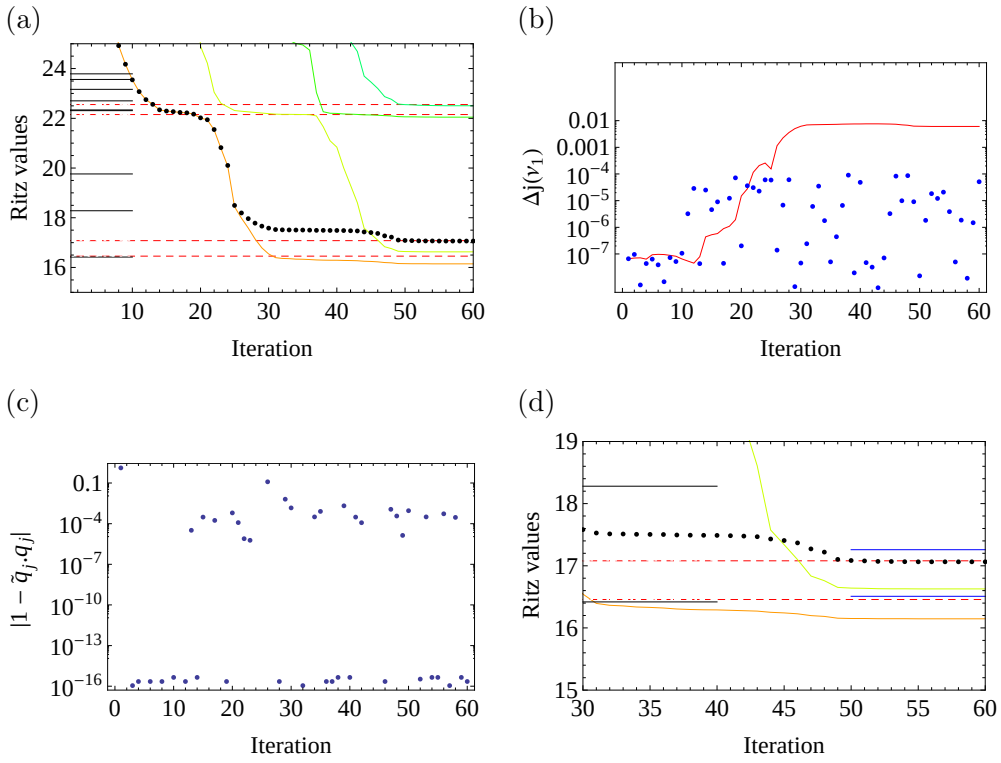


Figure 6.8: *Subspace iteration for* ^{4}*He Hamilton matrix for a* $N\hbar\omega = 4$ *model space. (a): Deviation of the Ritz values from the expectation values of the corresponding Ritz vectors in a subspace iteration using corrected the Lanczos vectors* $|\tilde{q}_j\rangle \to |q_j\rangle$*. (b): Angular momentum deviation of the Lanczos vectors (blue points) and the first Ritz vector (red line). (c): Differences of* $|\tilde{q}_j\rangle$ *and the returned* $|q_j\rangle$ *measured by their overlaps. (d): The lines at the right margin show the two lowest eigenvalues of the real projection* H_m*.*

In summary, Fig. 6.8 strongly suggests some kind of restart of the algorithm. However, it is not recommended to use the Simple Ritz Restart because this boosts the loss of good total angular momentum when the Ritz vector used for restart is nearly converged as can be shown easily: Using the simultaneous, normalized eigenbasis of H and J^2 (6.2), (6.3), the new start vector, which is assumed to be a good approximation of the ground state

$|E_0^{(j_0)}\rangle$ with total angular momentum j_0, for example, can be written as

$$|q_1\rangle = a_0|E_0^{(j_0)}\rangle + a_1 \sum_{\substack{j,k \\ c_{j_0,0}=0}} c_{j,k}|E_k^{(j)}\rangle, \tag{6.57}$$

where $\sum |c_{jk}|^2 = 1$, $||\ |q_1\rangle\ || = 1$, and $|a_0| \gg |a_1|$. Then one has

$$H|q_1\rangle = a_0 E_0^{(j_0)}|E_0^{(j_0)}\rangle + a_1 \sum_{\substack{j,k \\ c_{j_0,0}=0}} c_{j,k}E_k^{(j)}|E_k^{(j)}\rangle \tag{6.58}$$

and a straightforward calculation for $|r\rangle$ as it is given in the LFR Lanczos for iteration $i = 1$ yields

$$\begin{aligned} |r\rangle = & \left\{a_0 E_0^{(j_0)} - |a_0|^2 a_0 E_0^{(j_0)} - a_0|a_1|^2 \sum_{\substack{j,k \\ c_{j_0,0}=0}} |c_{j,k}|^2 E_k^{(j)}\right\}|E_0^{(j_0)}\rangle \\ & + \left\{a_1 - |a_0|^2 a_1 E_0^{(j_0)} - |a_1|^2 a_1 \sum_{\substack{j',k' \\ c_{j_0,0}=0}} |c_{j',k'}|^2 E_{k'}^{(j')}\right\} \\ & \times \sum_{\substack{j,k \\ c_{j_0,0}=0}} c_{j,k}|E_k^{(j)}\rangle. \end{aligned} \tag{6.59}$$

Dropping terms $\mathcal{O}(a_1^2)$ leaves

$$\begin{aligned} |r\rangle = & \left\{a_0 E_0^{(j_0)} - |a_0|^2 a_0 E_0^{(j_0)}\right\}|E_0^{(j_0)}\rangle \\ & + \left\{a_1 - |a_0|^2 a_1 E_0^{(j_0)}\right\} \sum_{\substack{j,k \\ c_{j_0,0}=0}} c_{j,k}|E_k^{(j)}\rangle \\ |r\rangle = & \ \tilde{a}_0|E_0^{(j_0)}\rangle + \tilde{a}_1 \sum_{\substack{j,k \\ c_{j_0,0}=0}} c_{j,k}|E_k^{(j)}\rangle. \end{aligned} \tag{6.60}$$

So the coefficients of $|r\rangle$ in the expansion analog to (6.57) are given by $\tilde{a}_0$ and $\tilde{a}_1$. Since $\sum_{\substack{j,k \\ c_{j_0,0}=0}} c_{j,k}|E_k^{(j)}\rangle$ contains all the unwanted angular momenta, the quantities

$$\frac{|a_1|^2}{|a_0|^2} \quad \text{resp.} \quad \frac{|\tilde{a}_1|^2}{|\tilde{a}_0|^2} \tag{6.61}$$

will serve as rough measures for the amount of bad angular momenta in the states $|q_1\rangle$ and $|r\rangle$. From the normalization of $|q_1\rangle$ follows that

$$\frac{|a_1|^2}{|a_0|^2} = \frac{1}{|a_0|^2} - 1 \ll 1 \quad (a_0 \approx 1) \tag{6.62}$$

which is much smaller than 1 for $a_0 \approx 1$ indicating that bad momenta are not strongly present in the initial start vector. From the normalization of $|r\rangle$ one obtains an analog relation for $\tilde{a}_0$ and $\tilde{a}_1$

$$\frac{|\tilde{a}_1|^2}{|\tilde{a}_0|^2} = \frac{1}{|\tilde{a}_0|^2} - 1 = \frac{1}{(E_0^{(j_0)})^2} \frac{1}{|a_0 - |a_0|^2 a_0|^2} - 1 \gg 1 \quad (a_0 \approx 1) \tag{6.63}$$

which is obviously much greater than 1 for $a_0 \approx 1$ indicating that the new Lanczos vector generated from $|q_1\rangle$ will have a strong angular momentum deviation.

6.3.1 Orthogonal Ritz Restart

As for the Simple Ritz Restart, the Thick-Restart [14] is not a convenient restart scheme for the subspace iteration. This is because the Thick-Restart utilizes information from the perturbed Hessenberg matrix from the preceding algorithm. Attempts to account for this errors lead to an non-orthogonal Y for which the Thick-Restart scheme is no longer valid.

In summary, a restart scheme is required that, on the one hand, uses the information contained in a Ritz or Ritz-like vector without employing it as a start vector for the iterations, and that, on the other hand does not resort to any other error-prone quantities from the preceding run of the algorithm. Obviously, the first criterion is violated by the Simple Ritz Restart and the second one by the Thick-Restart.

In an attempt to fulfill both criteria, the Lanczos algorithm will here in the following be modified so that it builds a space $\mathcal{K}_{m-1}^{\perp q_1}(A, q_2)$ orthogonal to a certain vector q_1 and that computes the projection of the matrix onto $\mathrm{span}(q_1, \mathcal{K}_{m-1}^{\perp q_1}(A, q_2))$. As for the standard variant of the Lanczos algorithm, it is easier to construct the Arnoldi analogue first and to make the trivial transition to the Hermitian case afterwards. The resulting algorithms will be denoted Orthogonal Ritz algorithms although they are not restricted to Ritz vectors or even to single vectors. The Orthogonal Ritz Arnoldi algorithm is presented in Table 6.1.

ORA :		The Orthogonal Ritz Arnoldi Algorithm
ORA(1):	Start :	Choose q_1 with $\|\|q_1\|\| = 1$
		Choose q_2 with $\|\|q_2\|\| = 1$
		and $q_2 \perp q_1$
ORA(2):		$k = Aq_1$
ORA(3):		$h_{11} = \langle q_1, k \rangle$
	Iterate :	$j = 2, \ldots, m$:
ORA(4):		$h_{ij} = \langle q_i, Aq_j \rangle, \quad i = 1, \ldots, j$
ORA(5):		$w_j = Aq_j - \sum_{i=1}^{j} h_{ij} q_i$
ORA(6):		$h_{j+1,j} = \|\|w_j\|\|$
ORA(7):		$q_{j+1} = w_j / h_{j+1,j}$
ORA(8):		$h_{j1} = \langle q_j, k \rangle$

Table 6.1: The Orthogonal Ritz Arnoldi algorithm.

Definition 6.3.1
The space $\mathcal{K}_{m-1}^{\perp q_1}(A, q_2)$ *is defined as*

$$\mathcal{K}_{m-1}^{\perp q_1}(A, q_2) = span\{\tilde{q}_2, \tilde{q}_3, \ldots, \tilde{q}_m\} \tag{6.64}$$

where the $\tilde{q}_i$ *are recursively determined for a given* q_1 *and* q_2 *by*

$$\begin{aligned} \tilde{q}_2 &= q_2 - \langle q_2, q_1 \rangle \; q_1 \\ \tilde{q}_3 &= A\tilde{q}_2 - \langle A\tilde{q}_2, q_1 \rangle \; q_1 \\ &\vdots \\ \tilde{q}_m &= A\tilde{q}_{m-1} - \langle A\tilde{q}_{m-1}, q_1 \rangle \; q_1. \end{aligned} \tag{6.65}$$

The tilde is used to distinguish the non-orthogonal vectors $\tilde{q}_i$ from the orthonormal basis q_i of $\mathcal{K}_{m-1}(A, q_2)$ generated by the algorithm. The convergence of the Orthogonal Ritz Arnoldi (Lanczos) will depend strongly on the

space $\mathcal{K}_{m-1}^{\perp q_1}(A, q_2)$ so it is worthwhile to examine one of its properties. In general, $\mathcal{K}_{m-1}^{\perp q_1}(A, q_2)$ is neither a Krylov space nor a subspace of $\mathcal{K}_{m-1}(A, q_2)$. However, it is a subspace of the invariant space of q_2 under A.

Proposition 6.3.1
Let $\mu := \mu(A, q_2)$ be the degree of the minimal polynomial of q_2 with respect to A and let

$$\mathcal{I}(A, q_2) := \mathcal{K}_{\mu}(A, q_2) \tag{6.66}$$

denote the invariant subspace of q_2 under A. Then $\mathcal{K}_{m-1}^{\perp q_1}(A, q_2)$ is a subset of $\mathcal{I}(A, q_2)$,

$$\mathcal{K}_{m-1}^{\perp q_1}(A, q_2) \subseteq \mathcal{I}(A, q_2). \tag{6.67}$$

Proof 6.3.1
For an arbitrary vector $k \in \mathcal{I}(A, q_2)$ one has

$$(Ak)^{\perp q_1} := Ak - \langle Ak, q_1 \rangle q_1 \ \in \ \mathcal{I}(A, q_2) : \tag{6.68}$$

If $\langle Ak, q_1 \rangle = 0$ this follows trivially from the invariance property of $\mathcal{I}(A, q_2)$. For $\langle Ak, q_1 \rangle \neq 0$ an orthonormal basis $\{q_1, q_2, \ldots, q_\mu\}$ of $\mathcal{I}(A, q_2)$ can be constructed from q_1 in which Ak can be expanded

$$Ak = \sum_{i=1}^{\mu} c_i q_i. \tag{6.69}$$

Obviously, $(Ak)^{\perp q_1}$ is then simply given by

$$(Ak)^{\perp q_1} = \sum_{i=2}^{\mu} c_i q_i \ \in \mathcal{I}(A, q_2) \tag{6.70}$$

which of course lies in $\mathcal{I}(A, q_2)$.

The basis vectors $\{\tilde{q}_2, \tilde{q}_3, \tilde{q}_4, \ldots, \tilde{q}_m\}$ of $\mathcal{K}_{m-1}^{\perp q_1}(A, q_2)$ from Def. 6.3.1 can be written as

$$\{\tilde{q}_2, \tilde{q}_i := (A\tilde{q}_{i-1})^{\perp q_1}, \ i = 3, \ldots, m-1\}. \tag{6.71}$$

Therefore, with the above considerations, if $\tilde{q}_i \in \mathcal{I}(A, q_2)$ then this is also true for $\tilde{q}_{i+1}$ and so by induction this is true for all basis vectors since the

induction assumption $\tilde{q}_i \in \mathcal{I}(A, q_2)$ *is true for* $\tilde{q}_2$. *Then, the proposition follows immediately*

$$\Rightarrow span\{\tilde{q}_2, \tilde{q}_i := (A\tilde{q}_{i-1})^{\perp q_1},\ i = 3, \ldots, m-1\} \equiv \mathcal{K}_{m-1}^{\perp q_1}(A, q_2) \subseteq \mathcal{I}(A, q_2). \tag{6.72}$$

□

Proposition 6.3.2
The Orthogonal Ritz Arnoldi algorithm, Table 6.1, computes the projection A_m *of* A *onto* $span\Big\{q_1, \mathcal{K}_{m-1}^{\perp q_1}(A, q_2)\Big\}$.

Proof 6.3.2
From steps ORA(4) and ORA(5) it is clear that by construction $\{q_2, \ldots, q_m\}$ *is a orthonormal basis of* $\mathcal{K}_{m-1}^{\perp q_1}(A, q_2)$.

There does not seem to be a quick and elegant way to prove the proposition as for the standard Arnoldi algorithm. Therefore, one has to take a closer look at the projection of A *the algorithm is supposed to compute. In general, the projection* $A_m = Q_m A Q$ *is given by*

$$Q_m^T A Q_m = \begin{pmatrix} \langle q_1, Aq_1\rangle & \langle q_1, Aq_2\rangle & \ldots & \langle q_1, Aq_m\rangle \\ \langle q_2, Aq_1\rangle & \langle q_2, Aq_2\rangle & \ldots & \langle q_2, Aq_m\rangle \\ \langle q_3, Aq_1\rangle & \langle q_3, Aq_2\rangle & \ldots & \langle q_3, Aq_m\rangle \\ \vdots & \vdots & & \vdots \\ \langle q_m, Aq_1\rangle & \langle q_m, Aq_2\rangle & \ldots & \langle q_m, Aq_m\rangle \end{pmatrix}. \tag{6.73}$$

This holds for any matrix Q_m *and it is a similarity transformation for the special case of orthonormal* Q_m.

The submatrix $A_m^{sub} := (A_m)_{i,j=2}^m$ *takes Hessenberg form as in the standard Arnoldi: Since*

$$Aq_j \in span\{q_1, q_2, \ldots, q_{j+1}\} \tag{6.74}$$

the orthogonality of $\{q_k\}$, *guaranteed by ORA(5), demands that* $(A_m^{sub})_{ij} = \langle q_i, Aq_j\rangle$ *vanish for* $i > j+1$. *The part of these non-vanishing matrix elements of* A_m^{sub} *at the diagonal and above are computed in ORA(4). The elements at the subdiagonal are computed in ORA(6):*

$$h_{j+1,j} \overset{ORA(6)}{=} ||w_j|| = \sqrt{\langle w_j, w_j\rangle} \tag{6.75}$$

$$
\begin{aligned}
h_{j+1,j} &\overset{ORA(7)}{\overset{ORA(5)}{=}} \sqrt{\langle h_{j+1,j} q_{j+1} \ , \ Aq_j - \sum_{i=1}^{j} h_{ij} q_i \rangle} \\
&= \sqrt{h_{j+1,j} \Big(\langle q_{j+1}, Aq_j \rangle - \sum_{i=1}^{j} h_{ij} \langle q_{j+1}, q_i \rangle \Big)} \\
&= \sqrt{h_{j+1,j} \langle q_{j+1}, Aq_j \rangle}
\end{aligned}
$$

$$
\begin{aligned}
\Rightarrow h_{j+1,j}^2 &= h_{j+1,j} \langle q_{j+1}, Aq_j \rangle \\
\Rightarrow h_{j+1,j} &= \langle q_{j+1}, Aq_j \rangle . \qquad (6.76)
\end{aligned}
$$

The first row of A_m is computed in steps ORA(3) and ORA(4) and the missing elements from the first column follow from ORA(8) so that after m steps A_m takes the form

$$
A_m = \begin{pmatrix}
ORA(3,1) & ORA(4,2) & ORA(4,3) & \dots & ORA(4,m) \\
ORA(8,2) & ORA(4,2) & ORA(4,3) & \dots & ORA(4,m) \\
ORA(8,3) & ORA(6,2) & ORA(4,3) & \dots & ORA(4,m) \\
ORA(8,4) & 0 & ORA(6,3) & \dots & ORA(4,m) \\
\vdots & \vdots & \vdots & & \vdots \\
ORA(8,m) & 0 & 0 & \dots & ORA(4,m)
\end{pmatrix} . \qquad (6.77)
$$

where ORA(step, iteration) denotes the step and the iteration when a matrix element is computed.

Unlike the matrix elements of A_m^{sub} below the subdiagonal the elements of the first column of A_m do not vanish since in general

$$
Aq_1 \notin span\Big\{ q_1, \mathcal{K}_{m-1}^{\perp q_1}(A, q_2) \Big\} . \qquad (6.78)
$$

In order to compute these elements the additional vector $k = Aq_1$ used in ORA(8) has to be stored in memory. □

The transition to the Hermitian case and, therefore, to the Lanczos algorithm (Table 6.2) is straightforward. Because of the Hermiticity of A

$$
\langle q_i, Aq_j \rangle = \langle q_j, Aq_i \rangle \qquad (6.79)
$$

and A_m becomes symmetric. A_m^{sub} then takes the common tridiagonal form and the elements of the first column of A_m do not have to be computed in ORA(8) since they follow from ORA(4). Therefore, there is no need to keep the vector $k = Aq_1$ in memory anymore. In a more adequate new Lanczos notation A_m becomes

$$A_m = \begin{pmatrix} \gamma_1 & \gamma_2 & \gamma_3 & \cdots & \gamma_{m-1} & \gamma_m \\ \gamma_2 & \alpha_1 & \beta_2 & & & \\ \gamma_3 & \beta_2 & \alpha_2 & \ddots & & \\ \vdots & & \ddots & \ddots & \ddots & \\ \gamma_{m-1} & & & \ddots & \alpha_{m-2} & \beta_{m-1} \\ \gamma_m & & & & \beta_{m-1} & \alpha_{m-1} \end{pmatrix}. \tag{6.80}$$

ORL :		The Orthogonal Ritz Lanczos Algorithm
ORL(1):	Start :	Choose q_1 with $\|\|q_1\|\| = 1$
		Choose q_2 with $\|\|q_2\|\| = 1$
		and $q_2 \perp q_1$
		Set $\beta_1 = 0$
ORL(2):		$\gamma_1 = \langle q_1, Aq_1 \rangle$
	Iterate :	$j = 2, \ldots, m$:
ORL(3):		$\gamma_j = \langle q_1, Aq_j \rangle$
ORL(4):		$\alpha_{j-1} = \langle q_j, Aq_j \rangle$
ORL(5):		$w_j = Aq_j - \gamma_j q_1 - \alpha_{j-1} q_j - \beta_{j-1} q_{j-1}$
ORL(6):		$\beta_j = \|\|w_j\|\|$
ORL(7):		$q_{j+1} = w_j / \beta_j$

Table 6.2: The Orthogonal Ritz Lanczos algorithm.

ORLS :		A More Stable Orthogonal Ritz Lanczos Algorithm
ORLS(1):	Start :	Choose q_1 with $\|\|q_1\|\| = 1$
		Set $\beta_1 = 1$
ORLS(2):		$r = Aq_1 - \langle Aq_1, q_1 \rangle$ with $\|\|r\|\| = 1$
ORLS(3):		$r = r/\|\|r\|\|$
ORLS(4):		$\gamma_1 = \langle q_1, Aq_1 \rangle$
	Iterate :	$j = 2, \ldots, m$:
ORLS(5):		$q_j = r/\beta_{j-1}$
ORLS(6):		$p = Aq_j$
ORLS(7):		$\gamma_j = \langle q_1, p \rangle$
ORLS(8):		$r = p - \gamma_j q_1$
ORLS(9):	If $j > 2$:	$r = r - \beta_{j-1} q_{j-1}$
	End If	
ORLS(10):		$\alpha_{j-1} = \langle q_j, r \rangle$
ORLS(11):		$r = r - \alpha_{j-1} q_j$
ORLS(12):		$\beta_j = \|\|r\|\|$

Table 6.3: The more stable version of the Orthogonal Ritz Lanczos algorithm.

As mentioned before, the convergence of the Orthogonal Ritz Lanczos strongly depends on the choice of q_2 and therefore on the space $\mathcal{K}_{m-1}^{\perp q_1}(A, q_2)$. The top (blue) curve of Fig. 6.9 shows the convergence of the algorithm when a random q_2 is employed. After the restart the convergence becomes extremely slow, indicating that there is a problem with the choice of q_2. This behavior can be motivated in terms of invariant subspaces. There are vectors $\{k_1, \ldots, k_m\},\ m \leq n$ with

$$\mathbb{R}^n = \mathcal{I}(k_1, A) \oplus \mathcal{I}(k_2, A) \oplus \cdots \oplus \mathcal{I}(k_m, A), \tag{6.81}$$

where $\mathcal{I}(k_j, A)$ denotes the invariant subspace of k_j under A. At the time

of a restart, the first Ritz vector $\tilde{\nu}_1$ trivially lies in the invariant subspace $\mathcal{I}(\tilde{\nu}_1, A)$. The more $\tilde{\nu}_1$ is converged to the first exact eigenvector ν_1, the more will $\mathcal{I}(\tilde{\nu}_1, A)$ resemble the invariant space of ν_1, $\mathcal{I}(\nu_1, A)$, and therefore the smaller will the dimension of $\mathcal{I}(\tilde{\nu}_1, A)$ since $\dim \mathcal{I}(\nu_1, A) = 1$. For that reason there is a considerable probability that a random q_2 will only have a small component in $\mathcal{I}(\tilde{\nu}_1, A)$ so that $\mathcal{I}(\tilde{\nu}_1, A) \cap \mathcal{I}(q_2, A) \approx \emptyset$. From Proposition 6.3.1 then follows that the space $\mathcal{K}_j^{\perp \tilde{\nu}_1}(A, q_2)$ will only contribute little to the approximation of ν_1, which can be seen clearly in Fig. 6.9. So there is good reason for not choosing a q_2 from a different invariant subspace than the one that contains q_1. A vector from the same subspace can be constructed by the remaining Ritz vectors: Since they are linear combinations of the Lanczos vectors that span the Krylov space $\mathcal{K}_m(A, q_1)$ that is a subset of $\mathcal{I}(q_1)$ the vector $\sum c_i \nu_i$ will lie in $\mathcal{I}(q_1)$. Therefore, the restarted algorithm will continue to span the invariant subspace of q_1. As expected, this leads to a strongly improved convergence as it can be seen from the middle (green) curve in Fig. 6.9.

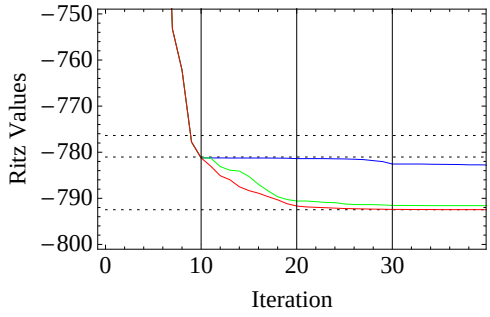


Figure 6.9: *Convergence of the Orthogonal Ritz Restart with some test matrix for different choices of start vectors. Blue (top): Arbitrary start vector. Green (mid): Start vector from Ritz vetors. Red (bottom): Start vector* $q_2 = Aq_1 - \langle Aq_1, q_1 \rangle$.

However, this convergence still is not satisfactory. The approximation comes close to the exact value but it converges to a value right above, indicating that there is something missing in the space from which one obtains the approximation. The missing part turn out to be the Aq_1 component that may strictly speaking not be missing but is suppressed by the algorithm

since for $q_2 \neq q_1$ it is never computed directly and from all the other Lanczos vectors the q_1 component is projected out before the matrix acts on it. Therefore, for the choice $q_2 = Aq_1 - \langle Aq_1, q_1 \rangle$ one expects a more complete convergence and this is confirmed by the lowest (red) curve in Fig. 6.9.

At a first glance nothing has bee gained at all with this approach since $q_2 = Aq_1 - \langle Aq_1, q_1 \rangle$ is exactly the quantity that causes all the trouble in the Simple Ritz Restart because of its large total angular momentum deviation. But the difference to the Orthogonal Ritz Restart is that there, q_1 and q_2 are allowed to be completely disconnectd. Therefore, one can even use a strongly angular momentum corrected q_2 without introducing errors in the algorithm.

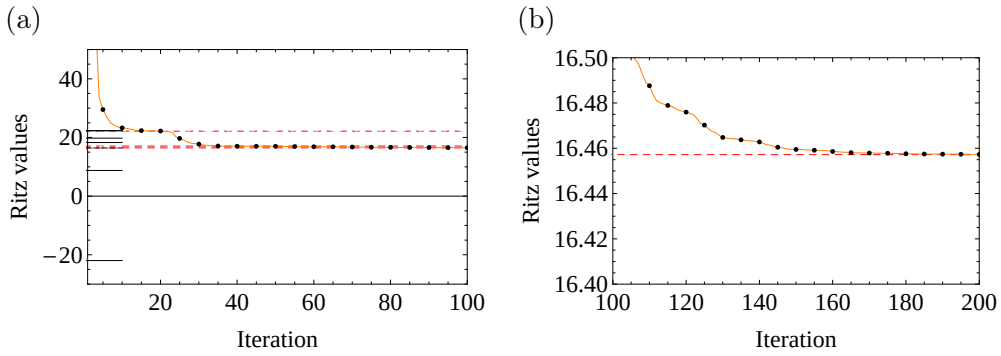


Figure 6.10: *Convergence of the Orthogonal Ritz Restart for the example of Fig. 6.8.*

6.3.2 Krylov Rayleigh-Ritz

Although the Orthogonal Ritz Lanczos finally is a converging Lanczos variant it is still far from being satisfying. This is because the Orthogonal Ritz Lanczos solves the Ritz restart problem but still does not avoid perturbations of the Hessenberg matrix. One the one hand, effects of such perturbations are reduced through frequent restarts. One the other hand, it is known that restarted algorithms converge more slowly than non-restarted ones and so clearly the Orthogonal Ritz Restart does.

At this point, after lots of effort that lead to no satisfying results it should be questioned whether the Lanczos algorithm is practical in the subspace iteration at all. In order to discuss this issue the most important properties, except for the matrix vector multiplication in every iteration step, of the

Simple Lanczos algorithm in exact arithmetics are listed below.

(1.) In every iteration step only 2 orthogonalizations are required.

(2.) Only $2-3$ vectors have to be kept in memory.

(3.) The Hessenberg matrix has tridiagonal form.

(4.) The Hessenberg matrix is calculated as a byproduct.

Property (1.) had to be sacrificed in order to maintain orthogonality among the Lanczos basis. For this reason property (2.) was also lost since the Lanczos vectors had to be at hand for the orthogonalization. Regarding properties (3.) and (4.), there is no real use for them in the subspace iteration. First, the reduction to tridiagonal form, (3.), is useful when many or all eigenvalues are approximated because one commonly does more than n iterations due to the problem of ghost eigenvalues and then the computation of eigenvalues from the Hessenberg matrix becomes a problem in itself. In the subspace iteration, however, the number of iterations is clearly below 1000 and for matrix dimensions of this order the eigenvalue problem can be attacked with direct methods easily. Likewise, (4.) is only of minor attractivity in the subspace iteration. One the one hand, because of the small dimensionality of the Hessenberg matrix it is not too important to compute it as a byproduct, as long as the computation is sufficiently cheap compared to matrix-vector multiplications. On the other hand, since the Hessenberg matrix contains perturbations, the sense of computing it at all is questionable.

Once more, one should remember Equation (4.2),

$$\text{Lanczos} = \text{Krylov} + \text{Rayleigh-Ritz}, \tag{6.82}$$

which states that the Lanczos is nothing but a convenient way to do a Rayleigh-Ritz projection onto a Krylov space. In the Rayleigh-Ritz procedure this Krylov space has to be present as some orthonormal basis Q_m and the projection of a matrix A then is $A_m = Q_m^T A Q$, or

$$Q_m^T A Q_m = \begin{pmatrix} \langle q_1, Aq_1\rangle & \langle q_1, Aq_2\rangle & \dots & \langle q_1, Aq_m\rangle \\ \langle q_2, Aq_1\rangle & \langle q_2, Aq_2\rangle & \dots & \langle q_2, Aq_m\rangle \\ \langle q_3, Aq_1\rangle & \langle q_3, Aq_2\rangle & \dots & \langle q_3, Aq_m\rangle \\ \vdots & \vdots & & \vdots \\ \langle q_m, Aq_1\rangle & \langle q_m, Aq_2\rangle & \dots & \langle q_m, Aq_m\rangle \end{pmatrix}. \tag{6.83}$$

The basis Q_m generated in the Orthogonal Ritz Lanczos algorithm, for example, is orthogonal, guaranteed by full reorthogonalization. Furthermore, by projection, it spans the angular momentum subspace to an accuracy of the accumulated angular momentum deviations of the Lanczos vectors. So the basis Q_m is good and only the computation of A_m fails to be accurate. Therefore, as it was already shown in Fig. 6.8, the Rayleigh-Ritz projection of A on span$\{Q_m\}$ delivers the real eigenvalue approximations from span$\{Q_m\}$.

In fact, doing the Rayleigh-Ritz projection does not cause sizable computational cost as long as the number of iterations is small, because the quantities Aq_j in (6.83) are computed in the Lanczos algorithm anyway when a new Lanczos vector is generated,

$$\text{Iteration step } j \ : Aq_j = \tilde{p}_j \overset{\text{projection}}{\rightarrow} p_j \overset{\text{orthogonalization}}{\rightarrow} q_{j+1} \tag{6.84}$$

and the only difference to the Lanczos algorithm is the need of computing the vector overlaps $\langle q_i, Aq_j \rangle \ , i = 1, \ldots, j$ in every iteration j. Here, the symmetry of A is already exploited so that only the upper triangular matrix elements have to be computed. Therefore, no additional storing of Aq_j is necessary. The Krylov Rayleigh-Ritz algorithm is presented in Table 6.4 with the most convenient choice of the order of projection and orthogonalization, as discussed below. It is obvious that this algorithm computes the projection of A onto Q_m by computing the matrix elements (6.83) so that a proof can be omitted.

KRR :		The Krylov Rayleigh - Ritz Algorithm
KRR(1):	Start :	Choose p with $\|\|p\|\| = 1$
		Set $\beta = 1$
	Iterate :	$j = 1, \ldots, m$:
KRR(2):		$q_j = p/\beta$
KRR(3):		$p = Aq_j$
KRR(4):	Compute :	$A_{ij} = A_{ji} = \langle q_i, p \rangle,\ i = 1, \ldots j$
	For :	$i = 1, \ldots, j$:
KRR(5):		$p = p - \langle p, q_i \rangle\ q_i$
	End For	
KRR(6):	If Necessary :	Project p on good angular momentum
		while keeping $q^{(\mathrm{proj.})}$ orthogonal to $\{q_i\}$
KRR(7):		$\beta = \|\|p\|\|$

Table 6.4: The Krylov Rayleigh-Ritz algorithm.

At first glance there is no real difference to the Arnoldi algorithm, since the crucial step $KRR(4)\acute{}$, where the matrix elements of the Rayleigh-Ritz projection are computed, looks the same as $A(2)$, where the Hessenberg matrix elements are determined. The reason why KRR is not equal to algorithm A is again the difference in the basis generated by both algorithms. The Arnoldi algorithm generates an orthonormal basis of the Krylov space $\mathcal{K}_j(q_1, A)$. In the numerical reality, where the start vector q_1 contains a certain amount of bad angular momentum and the total angular momentum is not exactly conserved, this Krylov space will contain more than the target angular momentum and, therefore, one needs the angular momentum projection. The projected basis spans the space $\mathcal{K}_j(q_1, A) \cap \mathcal{J}^{(j_{\mathrm{target}})}$ which is only a subset of the actual Krylov space. But the fact that the Arnoldi basis spans $\mathcal{K}_j(q_1, A)$ leads to the vanishing matrix elements of the projection and to the Hessenberg form, as it is explicitly used in Proof 6.3.2. For that reason will

A_m obtained from the Rayleigh-Ritz projection no longer have Hessenberg form but it will be symmetric for Hermitian matrices A.

Since the Krylov Rayleigh-Ritz is identical to the Lanczos algorithm in exact arithmetics all propositions concerning the convergence of the Lanczos algorithm still hold for the Krylov Rayleigh-Ritz algorithm, as well as the numerical considerations concerning the loss of orthogonality, for instance. For the numerical Krylov Rayleigh-Ritz algorithm the simple residual formula derived for the Lanczos algorithm does not hold any more because it is based on the validity of the three-term recurrence for the Lanczos algorithm. Besides computing the residual directly, at the cost of one matrix-vector multiplication per residual, all residuals can be calculated if one stores the quantities $|p_l\rangle = A|q_l\rangle$ during the algorithm. Then, the residuals are given by

$$\mathrm{res}\Big((\tilde{\nu}_i, \tilde{\xi}_i)\Big) = ||A\tilde{\nu}_i - \theta_i\tilde{\nu}_i|| = ||\sum_{k=1}^{m} y_k^{(i)}\Big(p_k - \theta_i q_k\Big)||. \tag{6.85}$$

Regarding the projection, there are various possibilities for where the projection can take place, as shown in Fig. 6.11.

(1.) Projecting p on good angular momentum right after the computation of the matrix elements A_{ij}. From Fig. 6.11 (1a) this choice looks reasonable but it leads to strong angular momentum deviations within the basis $\{q_i\}$. This is because of the reorthogonalization that takes place after the projection and through which the angular momentum deviations of previous q_i accumulate in the new basis vector. This accumulation of angular momentum deviation is, however, a minor effect that can not account for most of the deviations. More problematic are cancelations in the reorthogonalization that lead to amplifications of unwanted angular momenta as discussed for the Ritz Restart.

(2.) To avoid this accumulation of angular momentum deviations and the cancelation in the orthogonalization, the obvious way is to place the projection right after the orthogonalization. This guarantees the new basis vector having the desired accuracy regarding the angular momentum, 6.11 (2b), but it destroys the orthogonality of $\{q_i\}$, (2c). This loss of orthogonality leads to catastrophic results, (2a), that might be surprising at first.

Although the vectors $\{q_i\}$ have quite accurate angular momentum, the angular momentum of the first Ritz vector is totally lost. This might be understood in the following way: The loss of orthogonality induces the production of copies of Ritz vectors. Therefore, it will happen that after the projection rank $Q_m < m$ in the target angular momentum subspace, and consequently one gets eigenvector approximations from other subspaces that are contained in span$\{Q_m\}$ due to numerical errors.

(3.) In order to maintain both, more or less the desired angular momentum accuracy and orthogonality, one can place an orthogonalization of p before and after the projection. This circumvents the cancelation problem and restores the orthogonality after the projection. In this way a little angular momentum deviation is introduced in p after the projection through the orthogonalization. On the other hand the orthogonality of the basis vectors is guaranteed.

(4.) A similar possibility is to put the second orthogonalization into the Lanczos angular momentum projection algorithm, i.e. to orthogonalize the Lanczos vectors generated there against $\{q_i\}$, which causes the Ritz vector to be orthogonal to $\{q_i\}$. This orthogonalization has to take place before the reorthogonalization of the Lanczos vectors since their orthogonality has highest priority. In this way the angular momenta of $\{q_i\}$ are improved, but the degree of orthogonality is somewhat reduced.

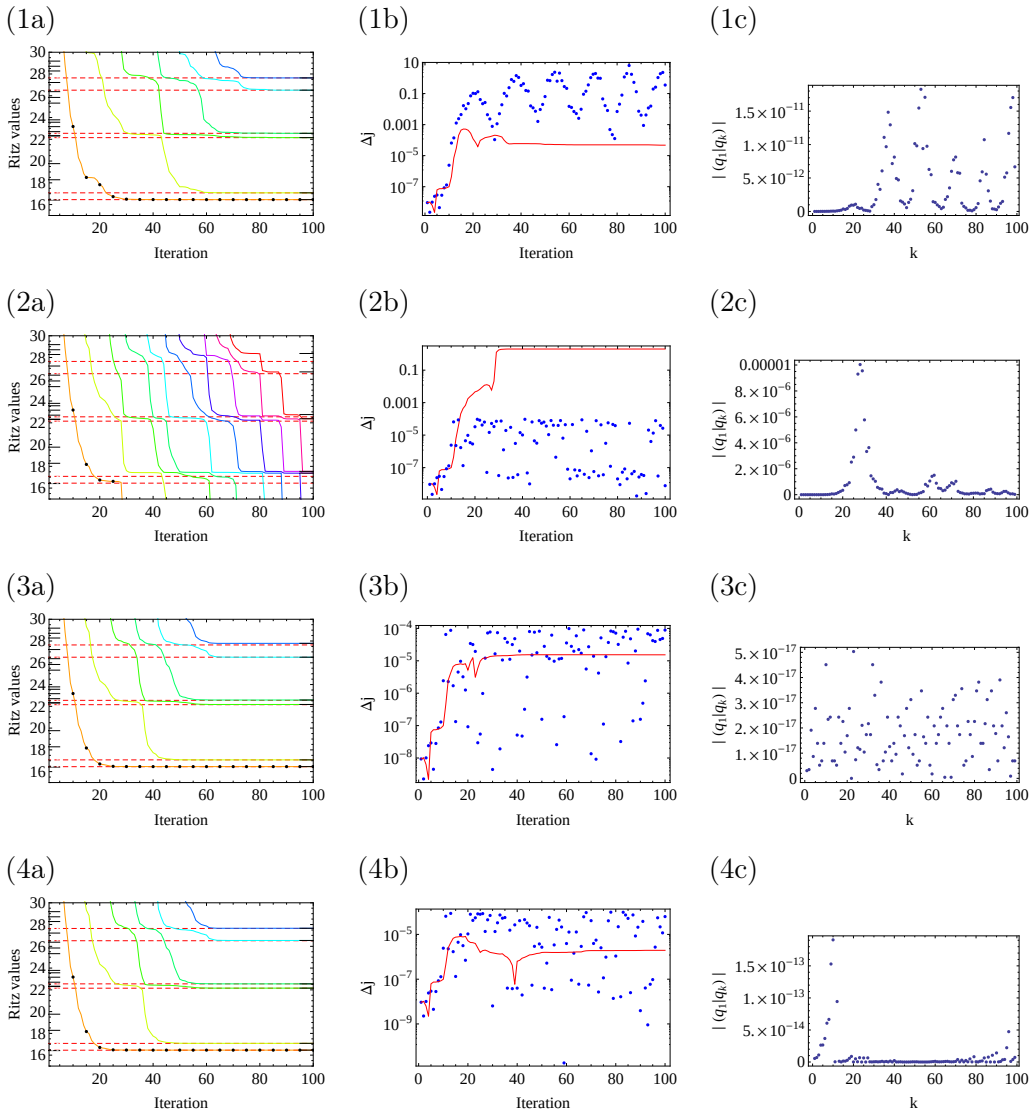


Figure 6.11: *Results of different arrangements of projection and reorthogonalization, as discussed in the text, for the example of Fig. 6.8, with total angular momentum accuracy* Δj *of* 10^{-4}. *Black dots in the convergence plots depict the energy expectation values of the first Ritz vector.*

Even for the most convenient variants (3) and (4), the convergence is not perfect. In fact, there is convergence of the Ritz values, but not to the exact eigenvalues to arbitrary accuracy. As depicted for the KRR version (4), in Fig. 6.12 (a), after a certain number of iterations the first Ritz value stays

right above the exact eigenvalue and shows no further convergence to it. One might suspect that this has something to do with the orthogonalization that in principle can prevent some directions to get included in the basis Q_m but this does not seem to occur in practice. Since the convergence of the first Ritz value remains unchanged after a restart — so that orthogonalization effects vanish — the reason for the failing convergence must be another.

In fact it is the low angular momentum accuracy that sets the limit to the convergence. If the Ritz vector is projected on angular momentum to an accuracy of 10^{-7} and with this the algorithm is restarted the Ritz value immediately comes closer to the exact eigenvalue and converges still further until again some iteration is reached after which the convergence stops without the Ritz value having reached the exact eigenvalue, Fig. 6.12 (b). As a consequence the residual norm computed via (6.85) will not become arbitrarily small and may possibly not be a good choice as convergence criterion. As in the Lanczos algorithm, where the residual is essentially determined by the last component of the Hessenberg vectors, one can do the same thing with the eigenvectors of the Rayleigh-Ritz projection in order to determine whether or not one can expect further convergence of a Ritz value by increasing the basis dimension.

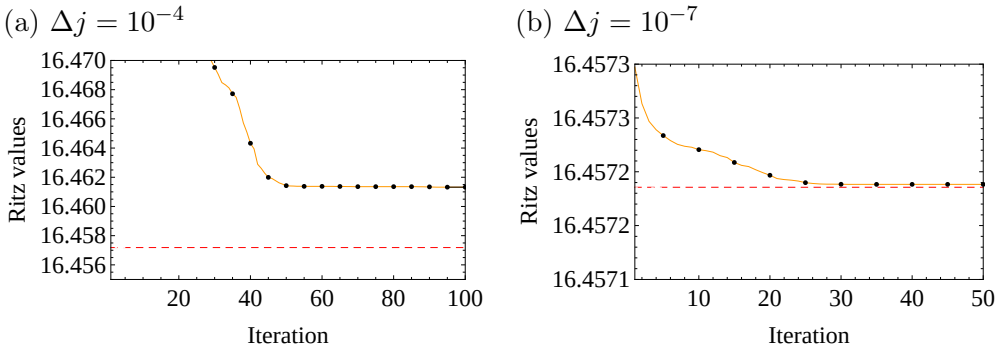


Figure 6.12: *Convergence limitations due to angular momentum accuracy., (a) With an total angular momentum accuracy* Δj *of* 10^{-4} *the first Ritz value converges but not to the exact eigenvalue. (b) After a restart with an improved angular momentum accuracy of* 10^{-7} *the Ritz value comes more closely to the exact eigenvalue but again does not come arbitrary close.*

6.4 Truncated Spaces

After considering the angular momentum projection in full $N\hbar\omega$ shell model spaces the actually interesting case of truncated spaces is considered now. Hilbert space truncations lead to non-vanishing commutators

$$[\tilde{H}, \tilde{J}^2] \geq 0 \tag{6.86}$$

of the operators represented in these truncated spaces even in exact arithmetics. As a consequence operating with $\tilde{H}$ on a $\tilde{J}^2$ eigenstate will no longer produce another $\tilde{J}^2$ eigenstate.

6.4.1 Lanczos Projection

That, due to the non-conservation of total angular momentum, starting from a $\tilde{J}^2$ eigenstate q_1 the Krylov vectors $\{q_1, \tilde{H}q_1, \tilde{H}^2q_1, \dots\}$ immediately lose the initial angular momentum seems nothing to worry about. Of course, the subspace iteration using the Lanczos algorithm will completely fail because in every iteration step large corrections to the new Lanczos vectors would have to be introduced in order to rotate them on good angular momentum, but the Krylov Rayleigh-Ritz algorithm would be insensitive to such interventions. However, this algorithm requires that the angular momentum projection still works in truncated spaces.

As seen in Section 6.2.1 the angular momentum projection of a vector using the Lanczos algorithm by spanning the invariant subspace of the vector with respect to J^2 is a powerful method as long as the number of distinct angular momentum eigenspaces is small. With every additional angular momentum eigenvalue introduced by modifying the Hilbert space an additional Lanczos iteration would be necessary in order to span such an invariant subspace.

Fig. 6.13 depicts the effect of the importance truncation (Chapter 3) on the angular momentum spectrum and its impact on the angular momentum projection using the Lanczos algorithm. (1b): In a full $N\hbar\omega$ shell model space the spectrum has its usual highly degenerate form with eigenvalues $j(j+1)$, $j = 0, \dots, j_{\max}$. Since $j_{\max} = 8$ in this example, the Lanczos algorithm provides almost exact eigenvalues after 9 iterations. (2b): Even a moderate truncation of the Hilbert space destroys the angular momentum

spectrum. New eigenvalues are introduced and consequently the Lanczos algorithm has no near-breakdown at iteration step 9 (2a). In iteration step 11 or 12 one might suspect an almost-breakdown since most Ritz values show good agreement with the old eigenspectrum but the residuals tell a different story. (3a,b): With further truncation the angular momentum spectrum gets farther away from its original shape and the angular momentum projection does not provide any converged results within few iterations.

Without spanning the invariant subspaces after few iterations there is no hope for finding an angular momentum component of a given vector, even after a number of iterations that matches the number of different angular momenta. Because the number of different angular momenta becomes large the algorithm might already produce a second (third, ...) eigenvalue approximation before there is a first approximation for another eigenvalue. So one is confronted with multiple Ritz vectors for a certain eigenvalue and Section 6.2.2 revealed that it is not easy to obtain the desired results from them.

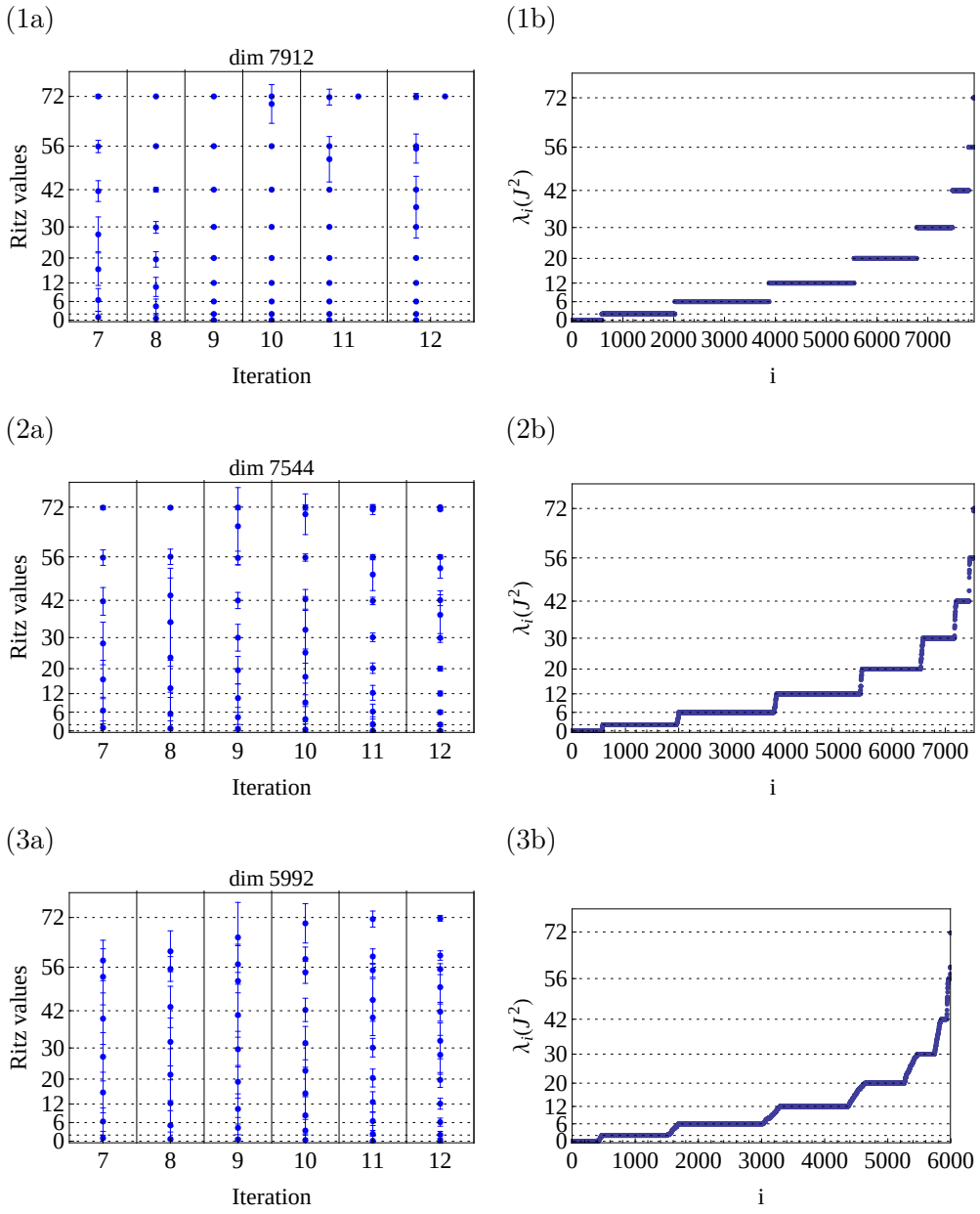


Figure 6.13: *Model space truncation effects for a ^{4}He total angular momentum spectrum ($N_{max}\hbar\Omega = 6$). The degeneracy is lost even for moderate truncations, preventing the Lanczos algorithm to span invariant subspaces within a few iterations.*

6.4.2 Shift-and-Invert Lanczos

Since the Lanczos projection fails in truncated spaces due to the destruction of the original J^2 eigenspectrum, the focus is on alternative ways to compute the projection of the Hamiltonian onto an angular momentum eigenspace. One way is to make explicit use of the knowledge of the target angular momentum and even of the loss of degeneracy through truncation.

In the untruncated case the angular momentum eigenspaces have large dimensions. So one would have to find many $\tilde{J}^2$ eigenstates to get reasonable eigenvalue approximations from an Rayleigh-Ritz projection. By truncation the eigenspace dimensions become smaller because of the increasing number of spaces. So the hope might be that the Importance Truncation reduces the dimension of the target angular momentum eigenspace (which still contains, due to the character of the Importance Truncation, the target state) so far that only a few eigenvectors already sufficiently span this space. Considered as an eigenspace of the target angular momentum is the space spanned by all eigenvectors of $\tilde{J}^2$ with eigenvalues $\in [j_{\text{target}} - \delta, j_{\text{target}} + \delta]$ with a δ not too small.

To find these eigenstates one can do a series of Lanczos algorithms, not for $\tilde{J}^2$ but for the shifted and inverted matrix

$$\left(\tilde{J}^2_\sigma\right)^{-1} := \left(\tilde{J}^2 - \sigma \mathbb{1}\right)^{-1} \tag{6.87}$$

with shift parameter $\sigma = j_{\text{target}}$. The eigenvalues $\xi_i(\tilde{J}^2)$ of $\tilde{J}^2$ are related to the ones of $\left(\tilde{J}^2_\sigma\right)^{-1}$ via [16]

$$\xi_i(\tilde{J}^2) = \sigma + \frac{1}{\xi_i((\tilde{J}^2_\sigma)^{-1})} \quad \Rightarrow \quad \xi_i((\tilde{J}^2_\sigma)^{-1}) = \frac{1}{\xi_i(\tilde{J}^2) - \sigma} \tag{6.88}$$

and the eigenvectors are identical. So the eigenvalues of $\tilde{J}^2$ that are nearest to σ become the extreme eigenvalues of $\left(\tilde{J}^2_\sigma\right)^{-1}$ and thus they are perfectly accessible to the Lanczos algorithm.

For large matrices $\left(\tilde{J}^2_\sigma\right)^{-1}$ cannot be formed explicitly. However, the only way the matrix enters the algorithm is the matrix-vector multiplication to obtain p,

$$p = \left(\tilde{J}^2_\sigma\right)^{-1} q_j \quad \Rightarrow \quad \tilde{J}^2_\sigma p = q_j. \tag{6.89}$$

So one can solve the linear system $\tilde{J}_\sigma^2 p = q_j$ using the shifted-only matrix to get p.

In practice however, the angular momentum spectrum is not enough resolved for $\left(\tilde{J}_\sigma^2\right)^{-1}$ to be sufficiently non-singular and the linear systems can not be solved reliably so that this approach also fails.

6.4.3 Hamiltonian Spectrum Shift

Another approach to target Hamiltonian eigenvalues from a specific angular momentum subspace is to shift up all the Hamiltonian eigenvalues that belong to other angular momentum subspaces. This can be achieved by adding an angular momentum dependent term to the Hamiltonian

$$H_{\lambda,j_{\text{target}}} = H + \lambda\Big(J^2 - \hat{j}_{\text{target}}\Big)^2. \tag{6.90}$$

This is not a uniform global spectrum shift as the magnitude of the shift depends on the angular momentum eigenvalue. However it is clear that the shifts are uniform in each separate J^2 subspace. $H_{\lambda,j_{\text{target}}}$ clearly commutes with J^2

$$[H_{\lambda,j_{\text{target}}}, J^2] = 0, \tag{6.91}$$

and so each angular momentum subspace can be considered separately. A simultaneous eigenstate of H and J^2 (6.2), (6.3) will still be eigenstate of $H_{\lambda,j_{\text{target}}}$ as well, satisfiying

$$H_{\lambda,j_{\text{target}}}|E_n^{(k)}\rangle = \Big(E_n + \lambda(\hat{k} - \hat{j}_{\text{target}})^2\Big)|E_n^{(k)}\rangle \tag{6.92}$$

with $\hat{k} = k(k+1)$. Thus, with $\lambda > 0$ all energy eigenvalues from angular momentum subspaces $\mathcal{J}^{(\hat{j}\neq\hat{j}_{\text{target}})}$ are shifted in the positive direction while the others are unaffected. If the eigenvalues of the original and the modified Hamiltonian are ordered such that

$$\xi_i(H_{\lambda,j_{\text{target}}}) \quad \xrightarrow{\lambda\to 0} \quad \xi_i(H) \tag{6.93}$$

and $\xi_i(J^2)$ denote the corresponding J^2 eigenvalues, then the shifted Hamiltonian spectrum is given in terms of the original spectrum by

$$\xi_i(H_{\lambda,j_{\text{target}}}) = \xi_i(H) + \lambda\Big(\xi_i(J^2) - \hat{j}_{\text{target}}\Big)^2. \tag{6.94}$$

From (6.94) a maximum shift can be estimated, by inserting $\hat{j}_{\max}$ and $\hat{j}_{\text{target}} = 0$. For $\hat{j}_{\max} = 240$ ($j_{\max} = 15$) this maximum shift is of the order

$$\lambda\Big(\xi_i(J^2) - \hat{j}_{\text{target}}\Big)^2 \quad \approx \quad \lambda \cdot 57600 \tag{6.95}$$

and the second largest shift for $\hat{j}_{\max} = 210$ ($j_{\max} = 14$) is of the order

$$\lambda\Big(\xi_i(J^2) - \hat{j}_{\text{target}}\Big)^2 \quad \approx \quad \lambda \cdot 44100. \tag{6.96}$$

The maximum of the original spectrum $\max_i |\xi_i(H)|$ usually is much smaller than these shifts. Thus, the upper end of the spectrum gets stretched far into the positive direction, while the target state energy remains at its original position. Since the upper end of the spectrum has much larger modulus than the lower end, the Lanczos algorithm will preferentially converge to these eigenvalues first. Additionally, since the shifts for $j_{\max} = 15$ and $j_{\max} = 14$ differ much, also large gaps are induced at least among the eigenvalues to high angular momenta. This will also worsen the convergence of the algorithm to the eigenvalues of interest (Fig. 6.14).

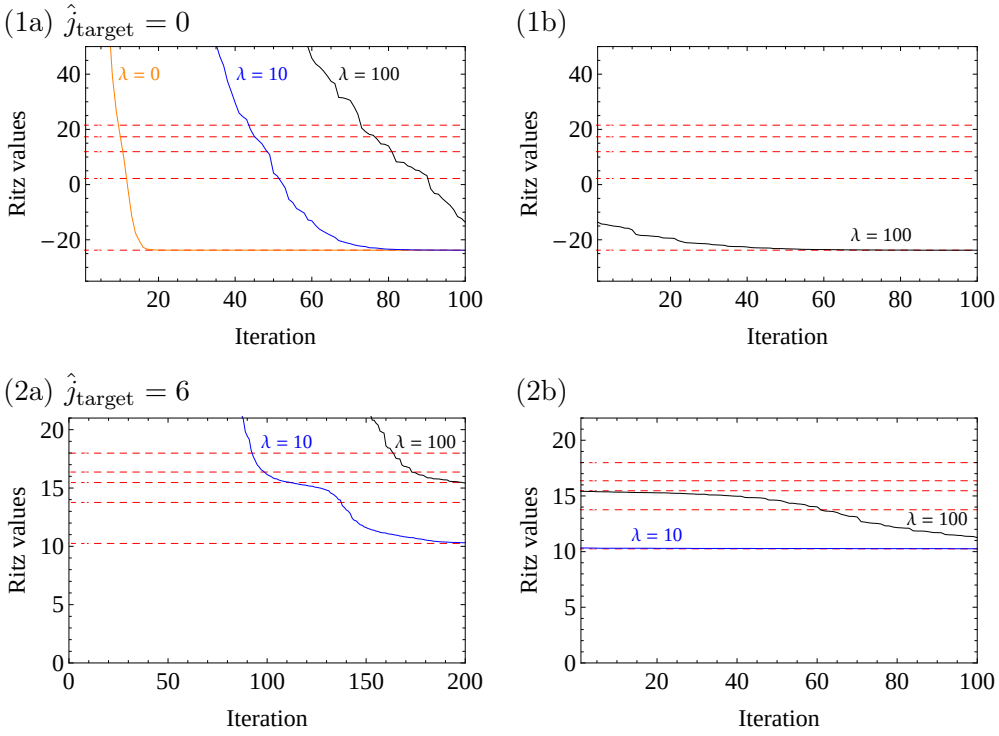


Figure 6.14: *Convergence for a shifted* ^{4}*He Hamiltonian for different* $\hat{j}_{target}$ *and* λ.

It should ne noted that even for $0 < \lambda < 1$ the spectrum is shifted up, because λ only stretches the shift induced by the term $(\hat{k}-\hat{j}_{\text{target}})^2$. Therefore, the minimum value of λ that is necessary to bring the target state to the low end of the spectrum also depends on the angular momenta of the lower lying states.

In truncated spaces a simple eigenvalue relation like (6.94) does not exist any more because of the non-commutation of $\tilde{H}$ and $\tilde{J}^2$, but it is expected that the energy spectrum from $\mathcal{J}^{(\hat{j}\neq\hat{j}_{\text{target}})}$ still will be shifted up.

Examples for the spectrum shift together with the importance truncation can be seen from Fig. 6.15 to 6.20, where a UCOM [33, 34] transformed AV18 interaction has been employed. The first part of the plots shows the exact H spectrum from the full $N_{\max}\hbar\omega$ model space with the $N_{\max}$ given in the caption. The red (solid) lines depict energy states with target angular momentum. The truncation was done by diagonalizing $H_{\lambda,j_{\text{target}}}$ in the $(N_{\max}-2)\hbar\omega$ for the lowest five states with angular momentum $\hat{j}_{\text{target}}$ and using these as reference states for generation of the truncated $N_{\max}\hbar\omega$ space. How the basis dimension is decreased due to the truncation is illustrated in the additional plot at the right margin. The second part of the figures illustrates the effect of the importance truncation on the (unshifted) Hamiltonian. From left to right the truncation parameter $\kappa_{\min}$ was varied from 0.1×10^{-3} to 1.0×10^{-3}. As one can see, except for the target states, the original spectrum is strongly perturbed by the truncation. The third section lists the angular momenta of the truncated spectrum at $\kappa_{\min} = 1.0 \times 10^{-3}$. The next part of the figures shows how the $H_{\lambda,j_{\text{target}}}$ spectrum evolves with increasing shift parameter; left for moderate shifts with $\lambda = 0, \ldots, 1$, right for strong shifts with $\lambda = 1, \ldots, 20$ or $\lambda = 1, \ldots, 5$. As expected, the states with $\hat{j}_{\text{target}}$ are nearly unaffected by the shifts, even for the largest λ. The last two sections show the isolated $\hat{j}_{\text{target}}$ spectrum at the largest calculated λ shift and its comparison to the original $\hat{j}_{\text{target}}$. In summary, nearly all of the deviation from the original spectrum originates in the truncation and not in the shift of the Hamiltonian.

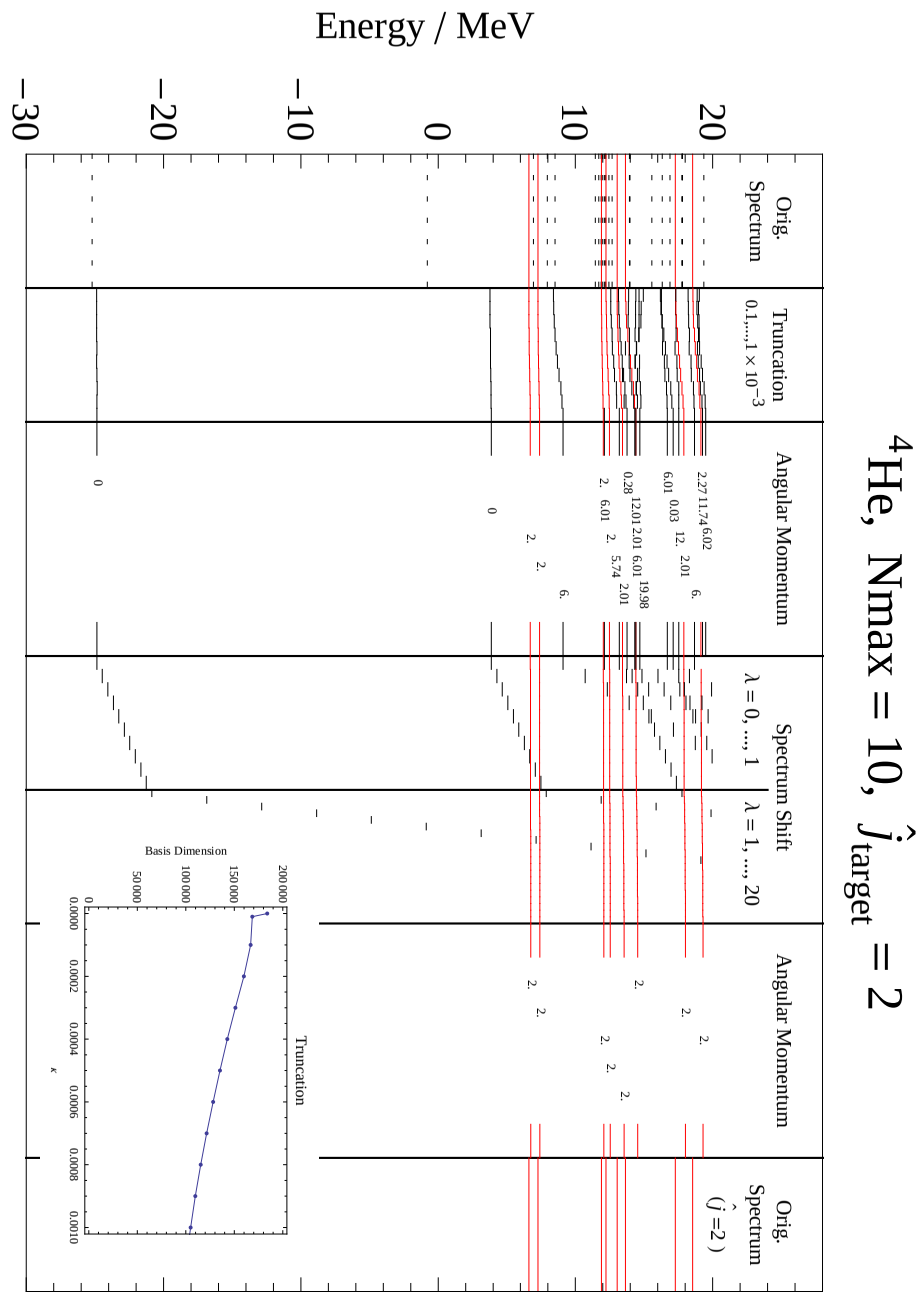


Figure 6.15: *Evolution of the energy spectrum of* ^{4}*He in a* $N_{max}\hbar\omega = 10$ *model space. Energy eigenvalues from* $\mathcal{J}^{(\hat{j}_{target})}$ *are displayed as red lines. The truncation is done with respect to the lowest five* $\hat{j}_{target} = 2$ *states.*

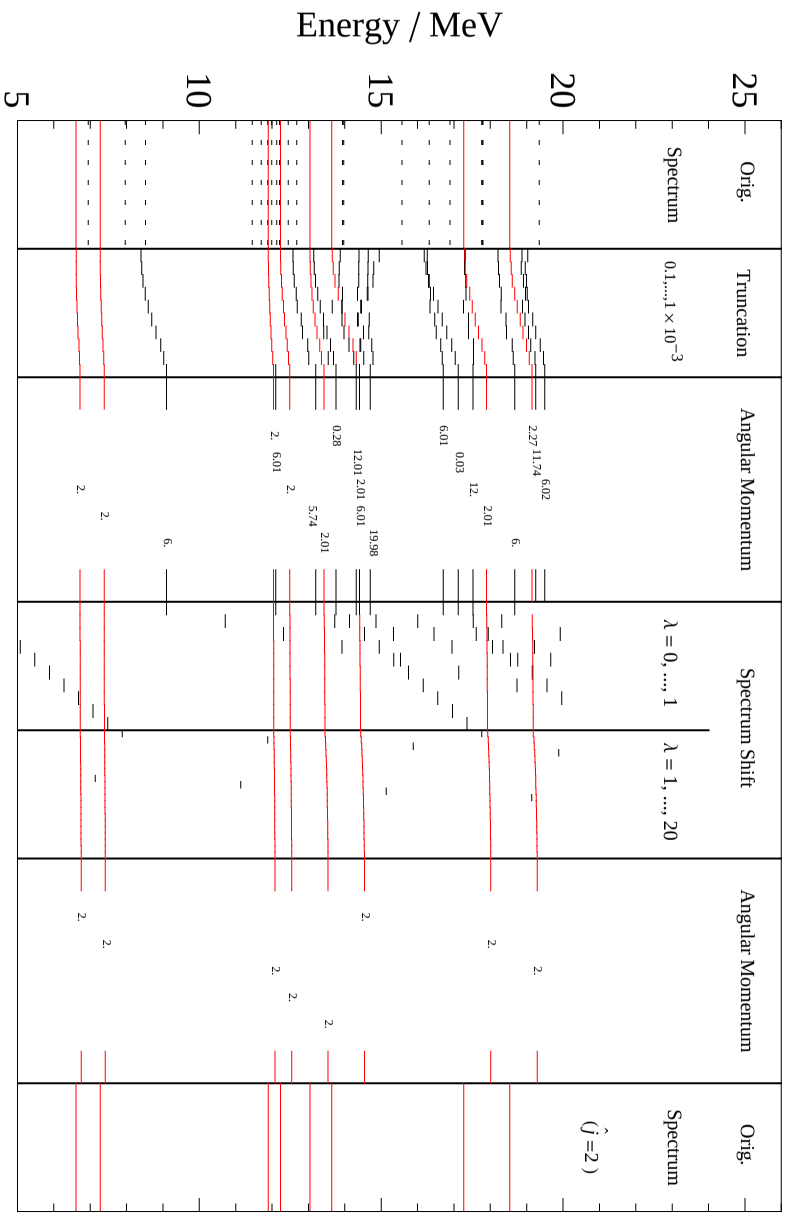


Figure 6.16: *As in Fig. 6.15, different plot range.*

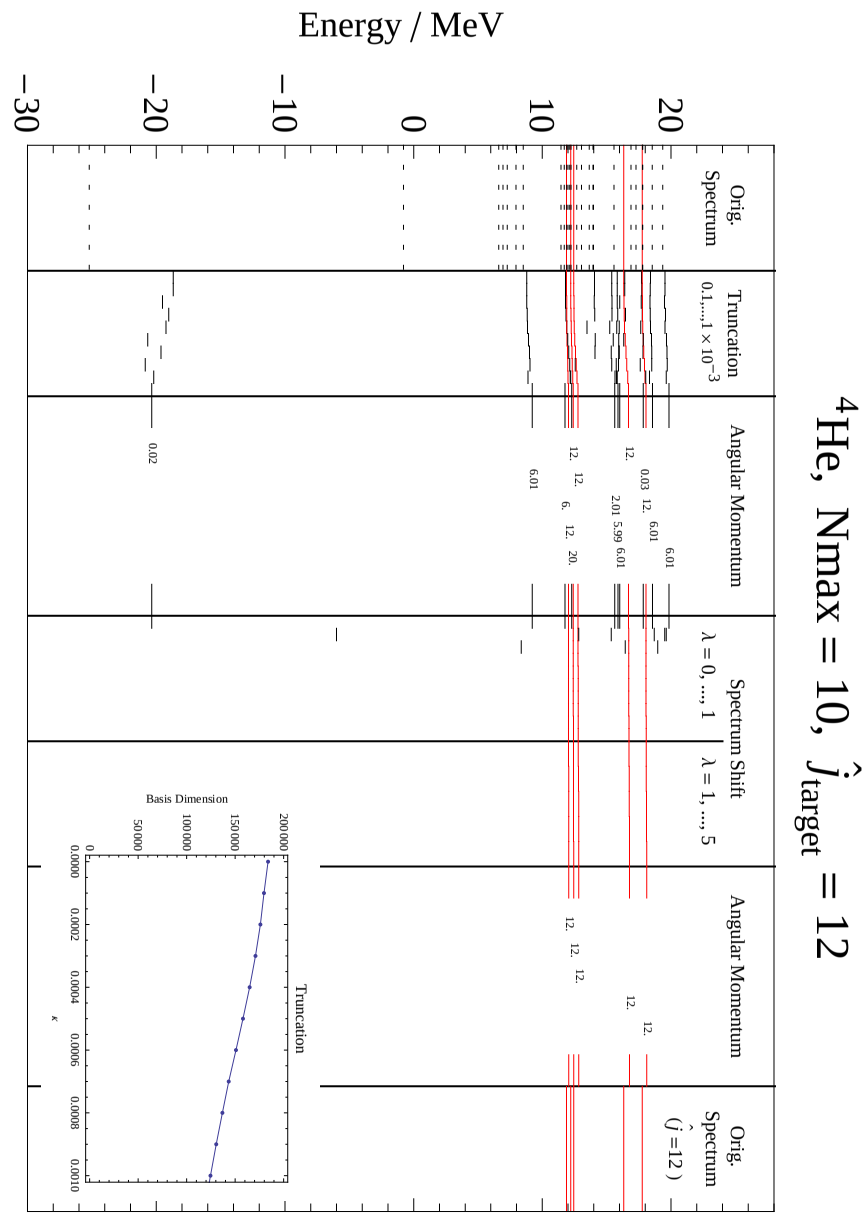


Figure 6.17: *As in Fig. 6.15, for* $\hat{j}_{target} = 12$.

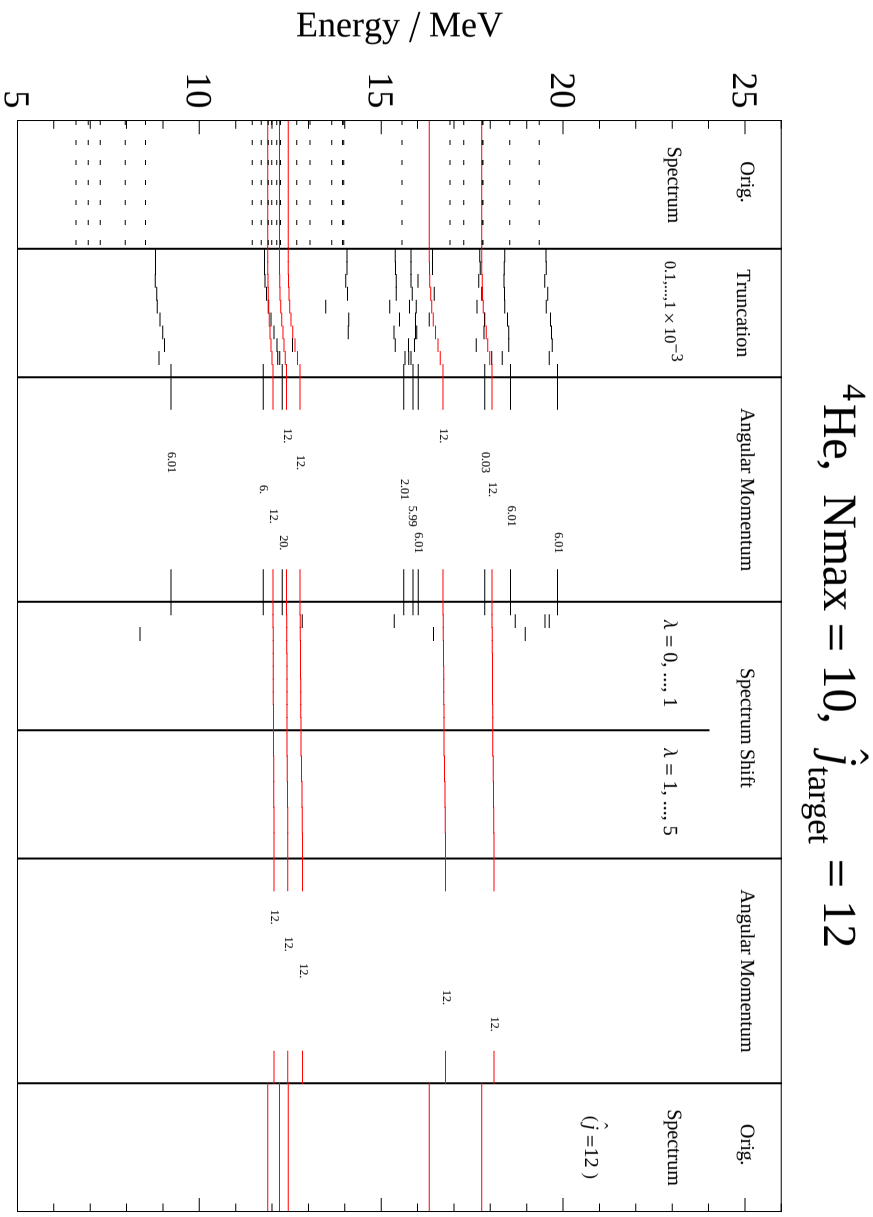


Figure 6.18: *As in Fig. 6.17, different plot range.*

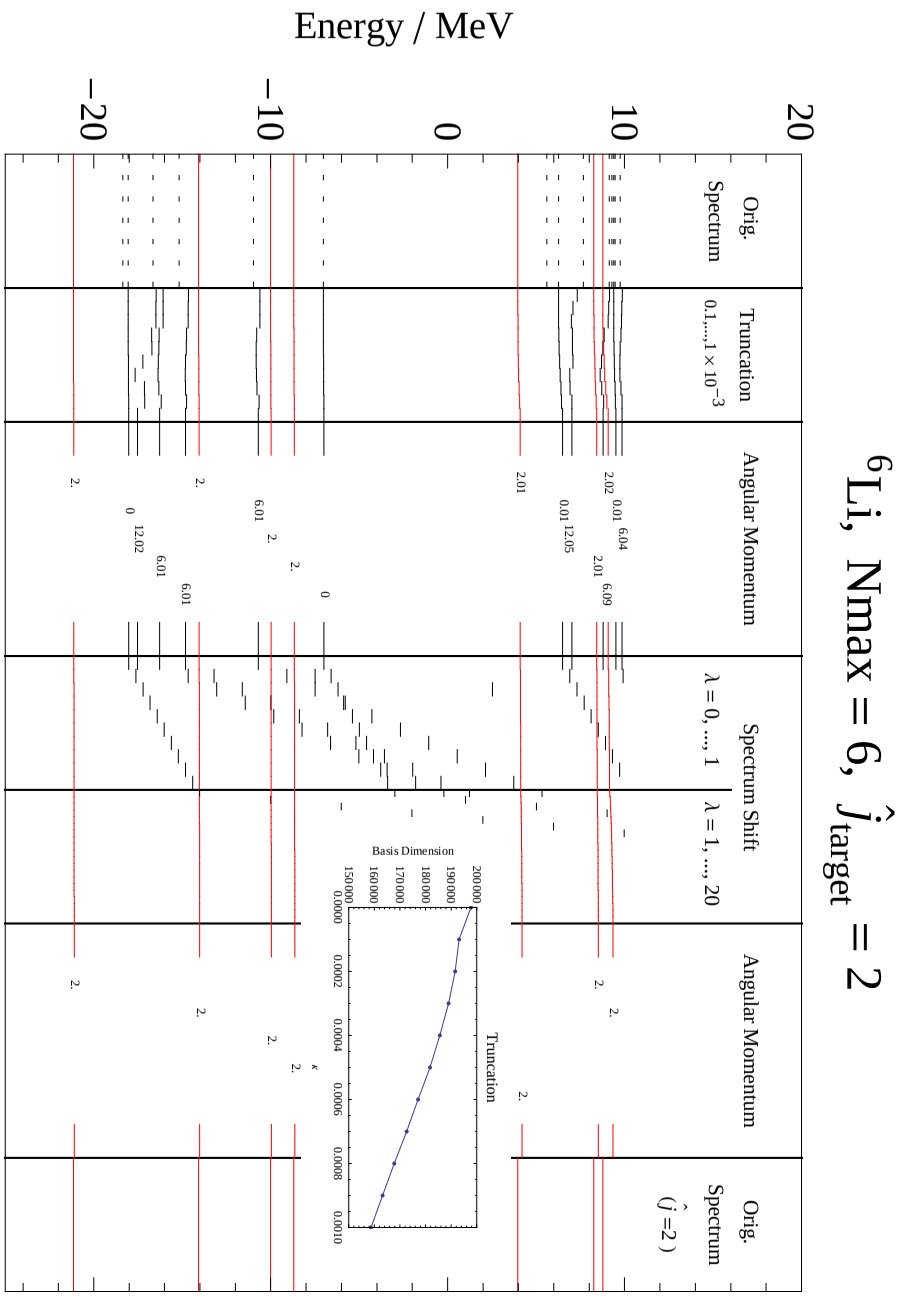
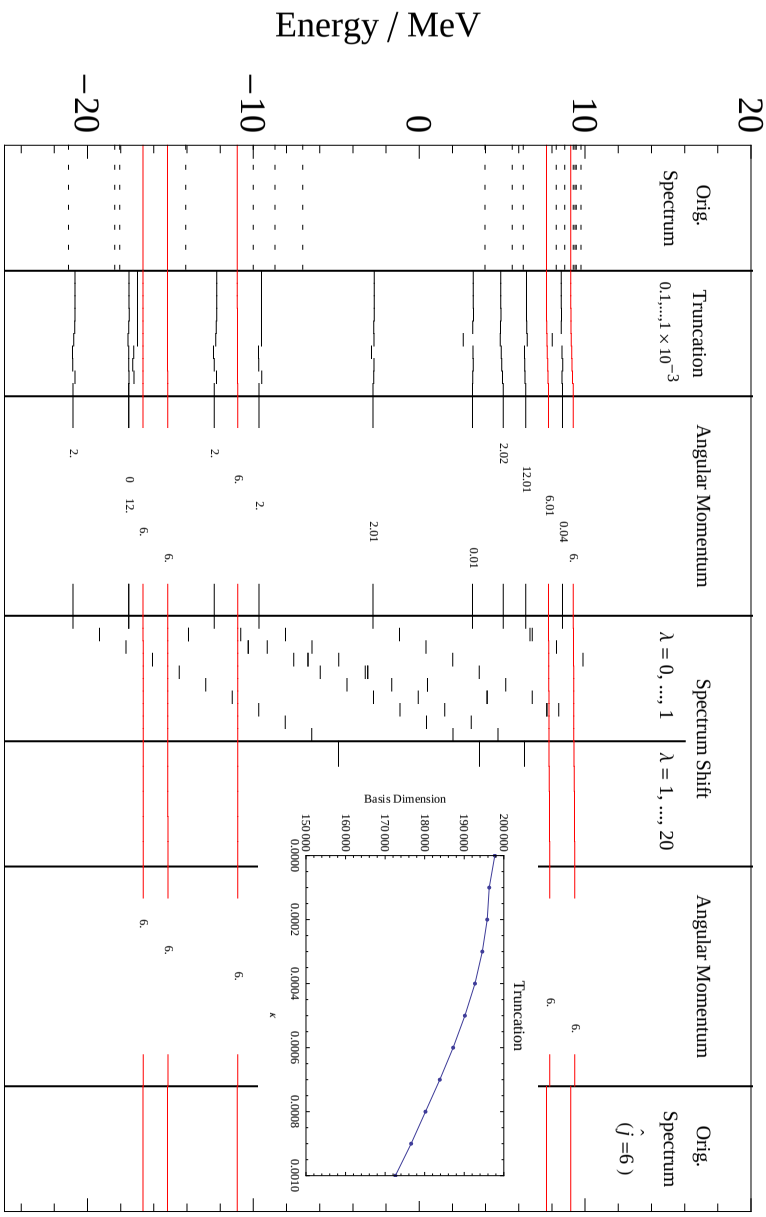


Figure 6.19: *Evolution of the energy spectrum of* ^{6}*Li in a* $N_{max}\hbar\omega = 6$ *model space. Energy eigenvalues from* $\mathcal{J}^{(\hat{j}_{target})}$ *are displayed as red lines. The truncation is done with respect to the lowest five* $\hat{j}_{target} = 2$ *states.*



^{6}Li, Nmax = 6, $\hat{j}_{\text{target}} = 6$

Figure 6.20: *As in Fig. 6.19, for* $\hat{j}_{target} = 6$.

6.5 Conclusion

In theory, the Lanczos algorithm is a powerful tool to perform a total angular momentum projection of the Hamiltonian and to extract the low end of its eigenspectrum.

In practice, besides the orthogonality loss, which can easily be dealt with by full reorthogonalization, one has to tackle the problem of loss of total angular momentum among the Lanczos basis. However, if the start vector for the algorithm is already a reasonable approximation for the ground state one can achieve converged results before the angular momentum is lost. Good angular momentum can be restored either by (expensive) explicit correction of the Lanczos vectors or simply by restarting the algorithm. It has been seen that the tridiagonal form of the matrix projection has to be sacrificed in any case, either in the Orthogonal Ritz Restart or the Krylov Rayleigh Ritz algorithm. That is, the Lanczos algorithm in its standard formulation is not convenient for angular momentum projection purposes if one takes measures to correct the angular momentum, and has to be replaced by a method that is less sensitive to perturbation from the outside of standard usage. But even with a robust alternative, like the Krylov Rayleigh Ritz, the convergence to the exact eigenvalues is limited by angular momentum impurities.

In truncated spaces the conservation of total angular momentum is lost and the degeneracy of angular momentum eigenvalues is resolved and therefore the idea of projecting a state on a certain angular momentum, exploiting the degeneracy of J^2 eigenvalues, does not work any longer. Another attempt to get projections of vectors using a shifted and inverted J^2 also fails due to singularities of this matrix. Finally, with the modification

$$H_{\lambda,j_{\text{target}}} = H + \lambda\Big(J^2 - \hat{j}_{\text{target}}\Big)^2 \tag{6.97}$$

of the Hamiltonian it is possible to directly address low-lying states of $\mathcal{J}^{(j)}$ spaces using a Lanczos algorithm even in truncated spaces. This comes at the cost of diminished convergence properties.

Chapter 7

Electromagnetic Transitions

The emission and absorption of electromagnetic radiation provides the most important source of information about the structure and lifetimes of nuclei. Electromagnetic moments and transition probabilities can accurately be measured, so they represent a valuable observable to confront theory with.

7.1 Transition Probabilities

Electromagnetic transition rates of a system from an initial state $|\Psi_i\rangle$ to a final state $|\Psi_f\rangle$ can be determined using Fermi's Golden Rule

$$T_{fi} = \frac{2\pi}{\hbar} |\langle \Psi_f | H_{\rm int} | \Psi_i \rangle|^2 \; g(E_f) \tag{7.1}$$

where $g(E_f)$ is the density of the final states and $H_{\rm int}$ is the Hamiltonian describing the interaction of the nucleus with an external electromagnetic field.

Starting from classical expressions for the Hamiltonians describing the radiation field,

$$H_{\rm field} \;=\; \frac{1}{8\pi} \int {\rm d}^3 r \, \Big(\boldsymbol{E}^2 + \boldsymbol{B}^2 \Big), \tag{7.2}$$

and the interaction of the radiation field $A^\mu = (\Phi, \boldsymbol{A})$ with the nucleus with charge density ρ and current density $\boldsymbol{j}$,

$$H_{\rm int} \;=\; \int {\rm d}^3 r \, \Big(\rho\Phi - \frac{1}{c} \boldsymbol{j} \cdot \boldsymbol{A} \Big), \tag{7.3}$$

after expanding the radiation field in multipoles it can be quantized in terms of photons.

The quantum number λ, related to orbital angular momentum and parity of the photon, is used to classify multipole radiation as electric ($\lambda = E$) or magnetic ($\lambda = M$) radiation. For the case of spontaneous emission of photons (i.e. vanishing external field) one obtains the probability for the emission of a single photon with wave number k, total angular momentum I and corresponding z-projection m

$$T_{fi}(\lambda, kIm) = \frac{8\pi(I+1)}{\hbar I((2I+1)!!)^2}\left(\frac{E^\gamma}{\hbar c}\right)^{2I+1} |\langle\Psi_f|\hat{\mathcal{M}}(\lambda kIm)|\Psi_i\rangle|^2. \quad (7.4)$$

In (7.4) the multipole transition operators

$$\begin{aligned}
\hat{\mathcal{M}}(EkIm) &= \frac{(2I+1)!!}{k^I(I+1)}\int \mathrm{d}^3r\ \left\{\rho Y_{Im}\frac{\partial}{\partial r} r j_I(kr) + i\frac{k}{c}\boldsymbol{j}\cdot\boldsymbol{r}Y_{Im}j_I(kr)\right\} \\
\hat{\mathcal{M}}(MkIm) &= \frac{-(2I+1)!!}{ck^I(I+1)}\int \mathrm{d}^3r\ \boldsymbol{j}\cdot(\boldsymbol{r}\times\boldsymbol{\nabla})\cdot(j_I(kr)Y_{Im}) \qquad (7.5)
\end{aligned}$$

are introduced. In the limit of long wavelengths the first order expansion in kr of the Bessel functions $j_I(kr)$ leads to the electric and magnetic multipole operators $\hat{Q}_{Im}, \hat{M}_{Im}$

$$\begin{aligned}
\hat{\mathcal{M}}(EkIm) &\approx \hat{Q}_{Im} \equiv \int \mathrm{d}^3r\ \rho r^I Y_{Im} \qquad (7.6) \\
\hat{\mathcal{M}}(MkIm) &\approx \hat{Q}_{Im} \equiv \frac{1}{c(I+1)}\int \mathrm{d}^3r\ (\boldsymbol{r}\times\boldsymbol{j})\cdot\boldsymbol{\nabla}(r^I Y_{Im}) \qquad (7.7)
\end{aligned}$$

which are independent of k. By inserting the charge and current density of a nucleus one obtains the final form of the electric and magnetic multipole operators in the long wavelength limit as

$$\begin{aligned}
\hat{Q}_{Im} &= e\sum_{i=1}^{A}\ \left(\tfrac{1}{2} - t_3^{(i)}\right)\ r_i^I\ Y_{Im}(\theta_i,\phi_i) \\
\hat{M}_{Im} &= \mu_N\sum_{i=1}^{A}\left\{g_s^{(i)}\boldsymbol{s}_i + \frac{2}{I+1}g_l^{(i)}\boldsymbol{l}_i\right\}\cdot\left(\triangle r^I Y_{Im}(\theta,\phi)\right)_{\boldsymbol{r}=\boldsymbol{r_i}}. \qquad (7.8)
\end{aligned}$$

If one does not distinguishes, and therefore averages over, the angular momentum projections in the initial and final states the probability for a certain multipole transition can be written as

$$T_{fi}(\lambda I) = \frac{8\pi(I+1)}{\hbar I((2I+1)!!)^2}\left(\frac{E_\gamma}{\hbar c}\right)^{2I+1} B(\lambda I, I_i \to I_f) \qquad (7.9)$$

with the reduced transition probabilities given in terms reduced matrix elements $\langle \cdot || \cdot || \cdot \rangle$

$$B\left(\begin{smallmatrix}\lambda=E\\ \lambda=m\end{smallmatrix}, I\right) = \frac{1}{2I_i+1} \cdot \begin{cases} |\langle \Psi_f || \hat{Q}_I || \Psi_i \rangle|^2 \\ |\langle \Psi_f || \hat{M}_I || \Psi_i \rangle|^2 \end{cases} . \tag{7.10}$$

In order to calculate the many-body matrix elements in the second quantization framework, the matrix elements in a single-particle basis have to be known. In the basis $|i\rangle \equiv |n_i(l_i s) j_i m_i m_{t_i}\rangle$ given in (2.10), the reduced matrix elements of the multipole operators [21, 22] are

$$\begin{aligned} \langle f || \hat{Q}_I || i \rangle &= \delta_{m_{t_i} m_{t_f}} \delta_{m_{t_i} \frac{1}{2}} \frac{e}{\sqrt{4\pi}} \frac{1+(-1)^{l_i+l_f+I}}{2} \mathcal{R}^{(I)}_{n_i l_i n_f l_f} \\ &\quad \times \sqrt{(2I+1)(2j_i+1)(2j_f+1)} \begin{pmatrix} j_f & j_i & I \\ 1/2 & -1/2 & 0 \end{pmatrix}_{3j} \end{aligned} \tag{7.11}$$

and

$$\begin{aligned} \langle f || \hat{M}_I || i \rangle &= \delta_{m_{t_i} m_{t_f}} \frac{\mu_N}{c\sqrt{4\pi}} (-1)^{j_i+I-\frac{1}{2}} \frac{1-(-1)^{l_f+l_i+I}}{2} \mathcal{R}^{(I-1)}_{n_i l_i n_f l_f} \\ &\quad \times \sqrt{(2I+1)(2j_i+1)(2j_f+1)} \begin{pmatrix} j_f & j_i & I \\ 1/2 & -1/2 & 0 \end{pmatrix}_{3j} \\ &\quad \times (I-\kappa) \left[g_l \left(1 + \frac{\kappa}{I+1} \right) - \tfrac{1}{2} g_s \right] \end{aligned} \tag{7.12}$$

with the abbreviation

$$\kappa \equiv (-1)^{l_i+j_i+\frac{1}{2}} + (-1)^{l_f+j_f+\frac{1}{2}}. \tag{7.13}$$

and the integrals $\mathcal{R}^{(\lambda)}_{n_i l_i n_f l_f}$ of the radial part of the harmonic oscillator wave function

$$\mathcal{R}^{(\lambda)}_{n_i l_i n_f l_f} \equiv \int_0^\infty g_{n_f l_f}(r)\ r^\lambda\ g_{n_i l_i}(r)\ r^2\ \mathrm{d}r. \tag{7.14}$$

For $l_i + l_f + \lambda =$ even (which are the only integrals contributing to non-vanishing matrix elements in (7.12) and (7.12)) the result can be written in closed form as [22]

$$\begin{aligned} \mathcal{R}^{(\lambda)}_{n_i l_i n_f l_f} &= (-1)^{n_i+n_f} \sqrt{\frac{n_i! n_f!}{\Gamma[n_i+l_i+\frac{3}{2}]\Gamma[n_f+l_f+\frac{3}{2}]}} \tau_i! \tau_f! \\ &\quad \times \sum_{\sigma=\sigma_{\min}}^{\sigma_{\max}} \frac{\Gamma\left[\frac{1}{2}(l_i+l_f+\lambda)+\sigma+\frac{3}{2}\right]}{\sigma!(n_i-\sigma)!(n_f-\sigma)!(\sigma+\tau_i-n_i)!(\sigma+\tau_f-n_f)!}, \end{aligned} \tag{7.15}$$

where

$$\begin{aligned}
\tau_i &= \tfrac{1}{2}(l_f - l_i + \lambda) && (7.16)\\
\tau_f &= \tfrac{1}{2}(l_i - l_f + \lambda) && (7.17)\\
\sigma_{\min} &= \max\{0, n_i - \tau_i, n_f - \tau_f\} && (7.18)\\
\sigma_{\max} &= \min\{n_i, n_f\}. && (7.19)
\end{aligned}$$

This form bears the risk of numerical overflows in actual computations. A more secure form can straightforwardly be derived using $\sum_i f_i = \sum_i \exp\ln f_i$,

$$\begin{aligned}
\mathcal{R}^{(\lambda)}_{n_i l_i n_f l_f} = (-1)^{n_i+n_f} \sum_{\sigma=\sigma_{\min}}^{\sigma_{\max}} \exp\Big\{ & -\ln[\tau_i!] + \ln[\tau_f!] \\
+ \; & \tfrac{1}{2}\Big(\ln[n_i!] + \ln[n_f!] - \ln\Gamma[n_i + l_i + \tfrac{3}{2}] - \ln\Gamma[n_f + l_f + \tfrac{3}{2}]\Big) \\
+ \; & \ln\Gamma[\tfrac{1}{2}(l_i + l_f + \lambda) + \sigma + \tfrac{3}{2}] - \ln[\sigma!] - \ln[(n_i - \sigma)!] \\
- \; & \ln[(n_f - \sigma)!] - \ln[(\sigma + \tau_i - n_i)!] - \ln[(\sigma + \tau_f - n_f)!] \Big\}, \qquad (7.20)
\end{aligned}$$

where the prefactor has at first been moved into the sum in order to keep the number of exp-calls as small as possible.

7.2 Results

Large-scale IT-NCSM calculations have been done for the positive-parity states of ^{10}B, ^{12}C and ^{16}O with results presented in this and the next section. The interaction employed is a SRG [23, 24] evolved CD-Bonn interaction where the $L = 0$ partial wave has been evolved independently from the other partial waves, with flow parameters

$$\begin{aligned}
\alpha_{S_{\text{wave}}} &= 0.015 \text{ fm}^4 \\
\alpha_{\text{other}} &= 0.001 \text{ fm}^4 \qquad (7.21)
\end{aligned}$$

adjusted such that the binding energies of ^{4}He and ^{16}O are reasonably reproduced. With this choice of parameters the interaction is expected to be suitable to appropriately describe the nuclear mass range between helium and oxygen. The harmonic oscillator frequency is fixed at $\hbar\omega = 24$ MeV.

7.2.1 ^{10}B

The calculation of the ^{10}B excitation spectrum is of particular interest since previous calculations did not reproduce the right ordering of the ground and first excited state [26, 27]. It was argued [27] that this was a general feature of two-body Hamiltonians and it was confirmed that for the chiral interaction [41] the inclusion of NNN contributions leads to the experimentally observed ground state spin [32].

It is therefore interesting to see the particular choice of the CD-Bonn parameters (7.21) reproduce the spin ordering of the first five states from (importance truncated) $N_{\max}\hbar\omega$ model spaces with $N_{\max} \geq 6$ (Fig. 7.1). Altogether, besides missing the 3^+ state, the experimental excitation spectrum is already nicely reproduced at $10\hbar\omega$. The spectrum has been calculated by ordinary partial diagonalization of the Hamiltonian for the lowest six eigenstates. IT-NCSM with sequential growing of the Hilbert spaces has been employed, where the full NCSM for $4\hbar\omega$ has been solved and from there for each $N\hbar\omega$ space the six computed states shown in Fig. 7.1 have been used as reference states from which the subsequent $(N+2)\hbar\omega$ model spaces have been built.

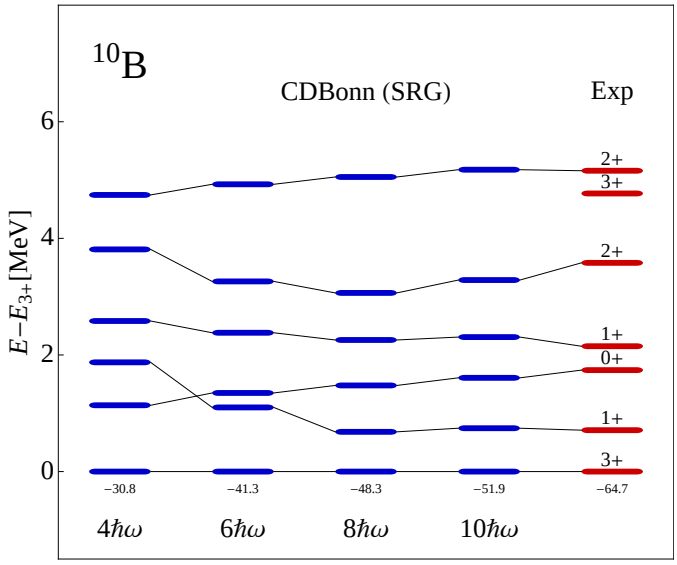


Figure 7.1: *Positive parity ^{10}B excitation spectrum.*

Usually, expectation values of observables evolve smoothly and well-behaved

under variation of the importance parameter $\kappa_{\min}$. One can therefore extrapolate the results obtained for various values of $\kappa_{\min}$ to the untruncated case $\kappa_{\min} \to 0$. It is expected that the full NCSM results are recovered at the limit $\kappa_{\min} \to 0$ within the extrapolation errors.

The total angular momentum expectation value J_{Exp} is a reasonable quantity to test the reliability of the extrapolation since the exact expectation values are known to be integers for the full NCSM. As illustrated in Fig. 7.2 for the three lowest ^{10}B states, the departure of the J expectation value from the exact value as function of $\kappa_{\min}$ can reasonably be reproduced with fit functions $j_{\text{ext}}(\kappa_{\min}) \in \prod_2$. The exact NCSM value $\lambda_{\text{full NCSM}}$ of an observable $\hat{\lambda}$ usually lies between the $\kappa_{\min} \to 0$ extrapolations $\lambda^{(1)}(\kappa_{\min}) \in \prod_1$ and $\lambda^{(2)}(\kappa_{\min}) \in \prod_2$. This is even more expected to hold for the expectation values of the intrinsic Hamiltonian since they display a more linear evolution with $\kappa_{\min}$ than J_{Exp} does. The estimate of $\lambda_{\text{full NCSM}}$ is then simply given by the average value of $\lambda^{(1)}(0)$ and $\lambda^{(2)}(0)$ while the error is determined by their distance,

$$\lambda_{\text{full NCSM}} = \frac{1}{2}\left(\lambda^{(2)}(0) + \lambda^{(1)}(0)\right) \pm \frac{1}{2}\left(\lambda^{(2)}(0) - \lambda^{(1)}(0)\right). \tag{7.22}$$

Quadrupole moments and $B(E2)$ transition probabilities show a more non-linear behavior which makes it harder to obtain a stable extrapolation. It is however still reasonable to expect the $\kappa_{\min} = 0$ value to lie between the linear and second-order extrapolation as it becomes evident in Fig. 7.3, the errors consequently just become larger.

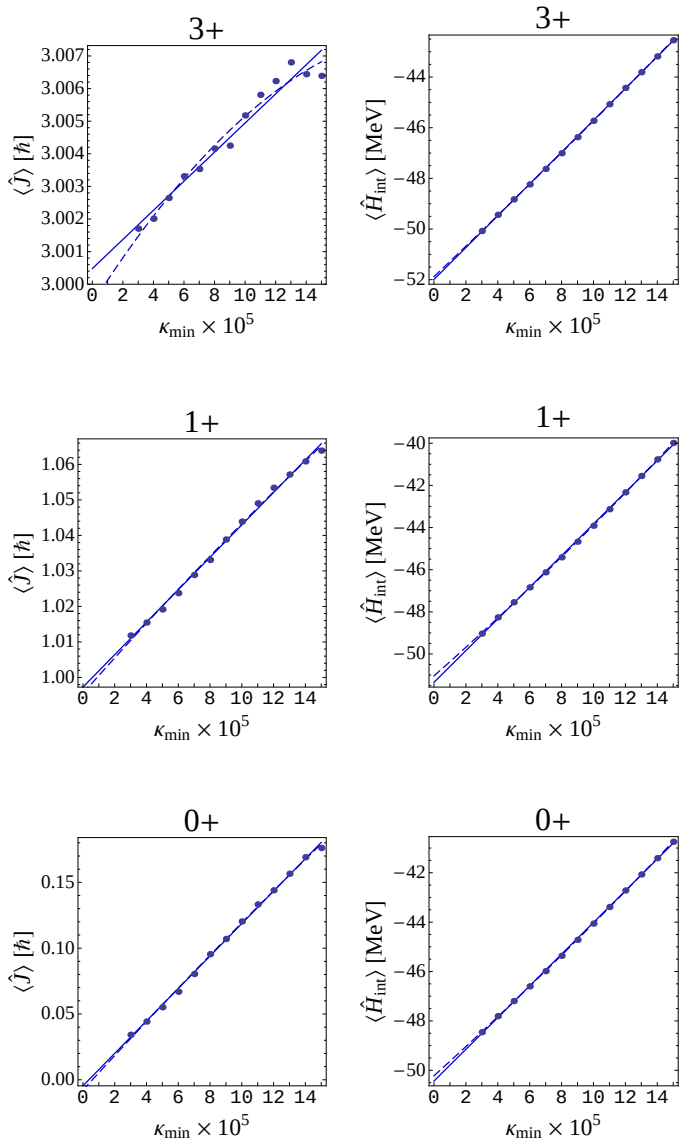


Figure 7.2: *J and H_{int} expectation values for different values of κ_{min} in the $8\hbar\omega$ model space. The expected result for the full NCSM usually lies between the linear and second-order extrapolation $\kappa_{min} \to 0$.*

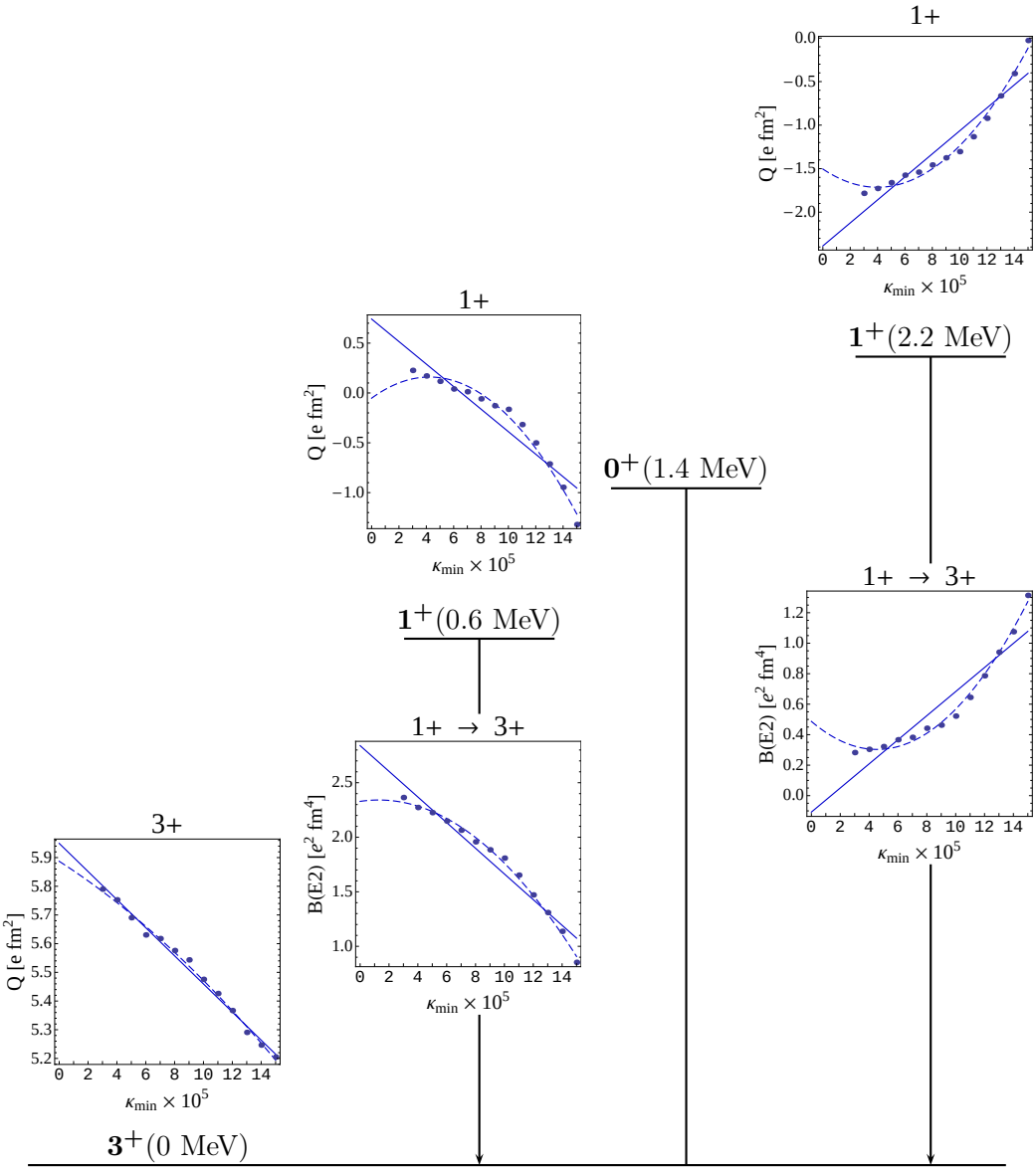


Figure 7.3: *Quadrupole moments and $B(E2)$ transition probabilities (if lowest possible transition) for the first four ^{10}B states at $N_{max}\hbar\omega = 8\hbar\omega$.*

7.2.2 ^{12}C

For the ^{12}C nucleus, excitation energies and reduced transition probabilities have been calculated for the lowest two states. As Fig. 7.4 illustrates, the predicted excitation energy is too small with a trend even to decrease as one goes to larger model spaces. This may be an indication for the lack of three-body force and assigned spin-orbit contributions in the CD-Bonn interaction. From the convergence of the excitation energy one would suspect the value of $E_{2+} - E_{0+}$ to settle at around 3.5 MeV.

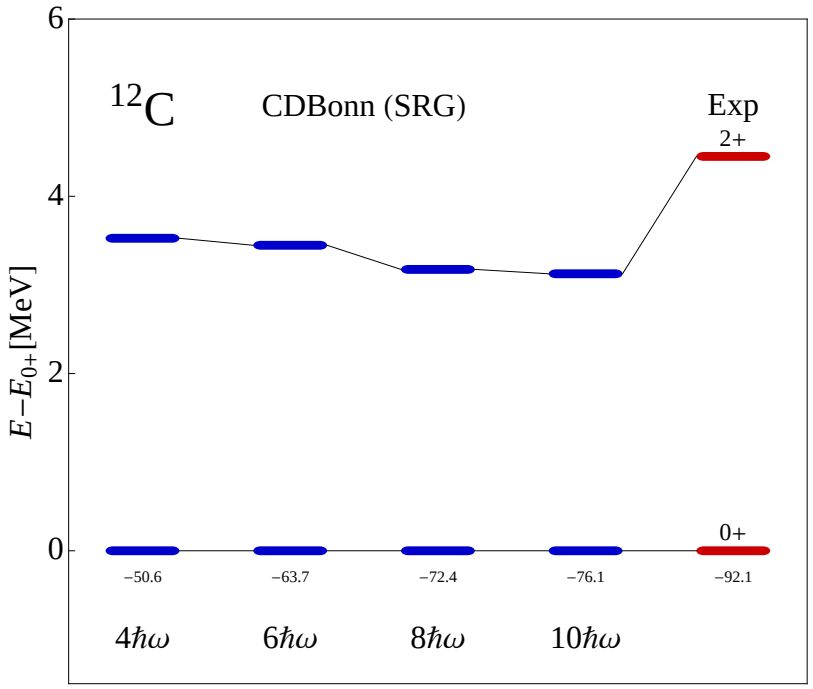


Figure 7.4: *Positive parity ^{12}C excitation spectrum.*

The evolution of the $B(E2)$ values with increasing $N_{\max}\hbar\omega$ is displayed in Fig. 7.5. The convergence is not quite clear, and the final value from the $10\hbar\omega$ model space $B(E2, 2^+ \rightarrow 0^+) = 4.3\ e^2\text{fm}^4$ clearly misses the experimental value in the range of 8.2 $e^2\text{fm}^4$ (as cited in [30]) and 7.6 $e^2\text{fm}^4$ (as cited in [31]). On the other hand, this result is similar to the ones presented in [30] (3.51 $e^2\text{fm}^4$) and [31] (4.62 $e^2\text{fm}^4$), using a effective CD-Bonn interaction at oscillator frequency $\hbar\omega = 15$ MeV.

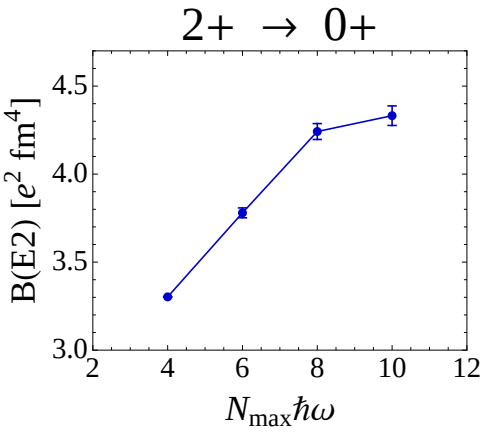


Figure 7.5: $B(E2, 2^+ \to 0^+)$ *values for the states from Fig. 7.4.*

Chapter 8

Density Distributions and Form Factors

One of the most elementary properties of nuclei is their size. Charge radii can be measured with high precision, so they constitute good observables to test theoretical predictions. Charge densities will be computed from the one-particle density matrices of Hamiltonian eigenstates. With the density matrix at hand, also electromagnetic form factors can easily be calculated.

The one-particle density matrix is defined in terms of field operators $\hat{\psi}^{\dagger}_{m_s m_t}(\mathbf{r})$, $\hat{\psi}_{m_s m_t}(\mathbf{r})$ as [21]

$$\begin{aligned} \langle \boldsymbol{r}, m_s m_t | \hat{\rho}_\Psi | \boldsymbol{r'}, m'_s m'_t \rangle &:= \rho(\boldsymbol{r} m_s m_t; \mathbf{r}' m'_s m'_t) \\ &= \langle \Psi | \hat{\psi}^{\dagger}_{m'_s m'_t}(\boldsymbol{r'}) \hat{\psi}_{m_s m_t}(\boldsymbol{r}) | \Psi \rangle. \end{aligned} \tag{8.1}$$

In the single-particle basis $\{|p\rangle\}$

$$|p\rangle \equiv |n_p \; l_p \; m_{l_p} \; s \; m_{s_p} \; m_{t_p}\rangle = \hat{c}^{\dagger}_p |0\rangle, \tag{8.2}$$

the field operator expansion reads

$$\begin{aligned} \hat{\psi}^{\dagger}_{m_s m_t}(\boldsymbol{r}) &= \left[\sum_q \; \hat{c}^{\dagger}_q \; \langle q | \right] | \boldsymbol{r}, m_s m_t \rangle \\ \hat{\psi}_{m_s m_t}(\boldsymbol{r}) &= \langle \boldsymbol{r}, m_s m_t | \left[| \sum_p | p \rangle \; \hat{c}_p \right] \end{aligned}$$

and in terms of these the density operator $\hat{\rho}_\Psi$ becomes

$$\hat{\rho}_\Psi = \sum_{pq} |p\rangle \underbrace{\langle\Psi|\hat{c}_q^\dagger \hat{c}_p|\Psi\rangle}_{\rho_{pq}} \langle q|. \tag{8.3}$$

The diagonal elements of $\hat{\rho}_\Psi$ can be interpreted as average density of particles of type m_t and with spin projection m_s

$$\langle \mathbf{r}, m_s m_t|\hat{\rho}_\Psi|\mathbf{r}, m_s m_t\rangle \equiv \rho_{m_t m_s}(\mathbf{r}). \tag{8.4}$$

Thus, integration over the space and spin spaces yields the particle number N_{m_t} for the particle species m_t

$$\sum_{m_s} \int \mathrm{d}\mathbf{r}\ \rho_{m_t m_s}(\mathbf{r}) = N_{m_t} = \left\{ \begin{smallmatrix} N_n,\ m_t{=}-\frac{1}{2} \\ N_p,\ m_t{=}+\frac{1}{2} \end{smallmatrix} \right. \tag{8.5}$$

In the following, the density will be calculated with angular and spin degrees of freedom integrated out and fixed isospin projection m_t,

$$\rho_{m_t m_t}(r) = \sum_{m_s} \int \mathrm{d}\phi \int \mathrm{d}\theta \sin\theta \sum_{pq} \langle \boldsymbol{r}, m_s m_t|p\rangle \langle\Psi|\hat{c}_q^\dagger \hat{c}_p|\Psi\rangle \langle q|\boldsymbol{r}, m_s m_t\rangle, \tag{8.6}$$

which is just the radial neutron or proton density. Using the j-coupled harmonic oscillator basis the coordinate representation of $|p\rangle$ reads

$$\langle \boldsymbol{r} m_s m_t|p\rangle = \sum_{m_{s_p} m_{l_p}} \begin{pmatrix} l_p & 1/2 & j_p \\ m_{l_p} & m_{s_p} & m_{j_p} \end{pmatrix}_{CG} R_{n_p l_p}(r)\ Y_{l_p m_{l_p}}(\theta,\phi)\ \delta_{m_s, m_{s_p}}\ \delta_{m_t, m_{t_p}}. \tag{8.7}$$

Plugging these expressions for $\langle \boldsymbol{r} m_s m_t|p\rangle$ and analogous for $\langle q|\boldsymbol{r} m_s m_t\rangle$ into (8.6) and making use of the orthogonality of spherical harmonics one obtains

$$\begin{aligned} \rho_{m_t}(r) \equiv \rho_{m_t m_t}(r) = & \sum_{m_s} \sum_{pq}\ \delta_{m_t m_{t_p}} \delta_{m_t m_{t_q}} \delta_{l_p, l_q} \delta_{m_{j_p}, m_{j_q}} \\ \times\ & \rho_{pq} \begin{pmatrix} l_p & s & j_p \\ m_{j_p}{-}m_s & m_s & m_{j_p} \end{pmatrix}_{CG} \begin{pmatrix} l_q & s & j_q \\ m_{j_q}{-}m_s & m_s & m_{j_q} \end{pmatrix}_{CG} R_{n_p l_p}(r)\ R_{n_q l_q}(r). \end{aligned} \tag{8.8}$$

From the density the root-mean-square charge radius can be calculated

$$\langle r^2\rangle = 4\pi \int \mathrm{d}r\ r^4\ \rho_p(r). \tag{8.9}$$

The proton (or charge) density distribution $\rho_p(\mathbf{r})$ can be used to derive the electromagnetic form factor $|F(\mathbf{q})|^2$ as the correction term for scattering that accounts for the spatial extension of the nucleus

$$\left(\frac{\mathrm{d}\sigma}{\mathrm{d}\Omega}\right)_{\text{ext}} = |F(\mathbf{q})|^2 \cdot \left(\frac{\mathrm{d}\sigma}{\mathrm{d}\Omega}\right)_{\text{point}}. \tag{8.10}$$

$F(\mathbf{q})$ can be obtained as the fourier transform of the charge density distribution (normalized by $4\pi e \int \mathrm{d}r\ r^2 \rho_p(r) = Ze$)

$$F(\mathbf{q}) = \int \mathrm{d}\mathbf{r}\ \rho_p(\mathbf{r})\ e^{i\mathbf{qr}}. \tag{8.11}$$

For spherical symmetric density distributions as they are used here the angular integration can be performed arriving at

$$F(q) = 4\pi \int \mathrm{d}r\ r^2\ \rho_p(r)\ \frac{\sin(qr)}{qr}. \tag{8.12}$$

In order to actually calculate the matrix representation of $\hat{\rho}_\Psi$, it is not feasible to implement the formula

$$\hat{\rho}_\Psi = \sum_{pq}\ |p\rangle\ \langle\Psi|\hat{c}_q^\dagger \hat{c}_p|\Psi\rangle\ \langle q| \tag{8.13}$$

straightforwardly, because of the large dimension of the Slater determinant basis $\{|\Phi_i\rangle\}$ in which $|\Psi\rangle$ is represented

$$|\Psi\rangle = \sum_i\ C_i\ |\Phi_i\rangle. \tag{8.14}$$

With this expansion, (8.13) reads

$$\hat{\rho}_\Psi = \sum_{ij}\sum_{pq}\ C_i^*\ C_j\ |p\rangle\ \langle\Phi_i|\hat{c}_q^\dagger \hat{c}_p|\Phi_j\rangle\ \langle q| \tag{8.15}$$

The loop over the single particle states is never performed, because from the Slater determinants $\langle\Phi_i|$ and $|\Phi_j\rangle$ the single particle states pq that can contribute to the elements ρ_{pq} of $\hat{\rho}_\Psi$ can easily be determined from Slater-Condon rules.

Symmetry of $\hat{\rho}_\Psi$ allows to compute only a triangle of its matrix representation. In (8.13) this could be accomplished by restraining the sum over

single particle states by $p \leq q$. But it is even possible in (8.15), by restraining the $|\Psi\rangle$ expansion by $i \leq j$. As a consequence, for two Slater determinants in the single non-conincidence case (Sec. 9.3)

$$\begin{aligned} |\Phi_i\rangle &= |a\Box\rangle \\ |\Phi_j\rangle &= |b\Box\rangle, \end{aligned} \tag{8.16}$$

where the Box is used to abbreviate orbitals occupied in both determinants, the restriction $i < j$ will lead to the term

$$\langle a\Box|\hat{c}_a^\dagger \hat{c}_b|b\Box\rangle, \tag{8.17}$$

yielding a contribution to ρ_{ba}, but not to the case in which the determinants are exchanged

$$\langle b\Box|\hat{c}_b^\dagger \hat{c}_a|a\Box\rangle \tag{8.18}$$

which would contribute to ρ_{ab}. But, because of the symmetry of $\hat{\rho}_\Psi$, the missing contribution to ρ_{ba} from $i > j$ is the very same than the one to ρ_{ba} from $i < j$. Thus, if one adds up all contributions one gets for ρ_{pq} for $p \geq q$ or $p < q$ in the triangle $p \leq q$ of ρ, this triangle will be complete. Analog considerations also hold for higher-body density matrices.

8.1 Results

8.1.1 ^{16}O

The IT-NCSM allows to go to model spaces beyond $N_{\max}\hbar\omega = 8\hbar\omega$, where the model space dimensions in the full NCSM become too large for an actual solution even for the ground state problem only. At $N_{\max}\hbar\omega = 14\hbar\omega$ (Fig. 8.1(a)) the binding energy seem to have reached the tail of the typical exponential convergence, so that the final value of -121.9 MeV already is in reasonable agreement with the experimental one of -127.6 MeV [28]. An exponential fit an the data predicts a ground state energy for $N_{\max} \to \infty$ of -128.84 MeV.

The charge density (Fig. 8.1(b)) and charge form factor (Fig. 8.1(c)) vary only little with increasing model spaces so that no significant improvement of the results would be expected from even larger spaces. Generally,

the charge radii are too small (2.12 fm) compared to the experimental one (2.7 fm [29]) which cause the calculated charge form factor to deviate from experiment. Thus, the failure to reproduce the experimental charge form factor may indicate a shortcoming of the employed interaction (from Sec. 7.2).

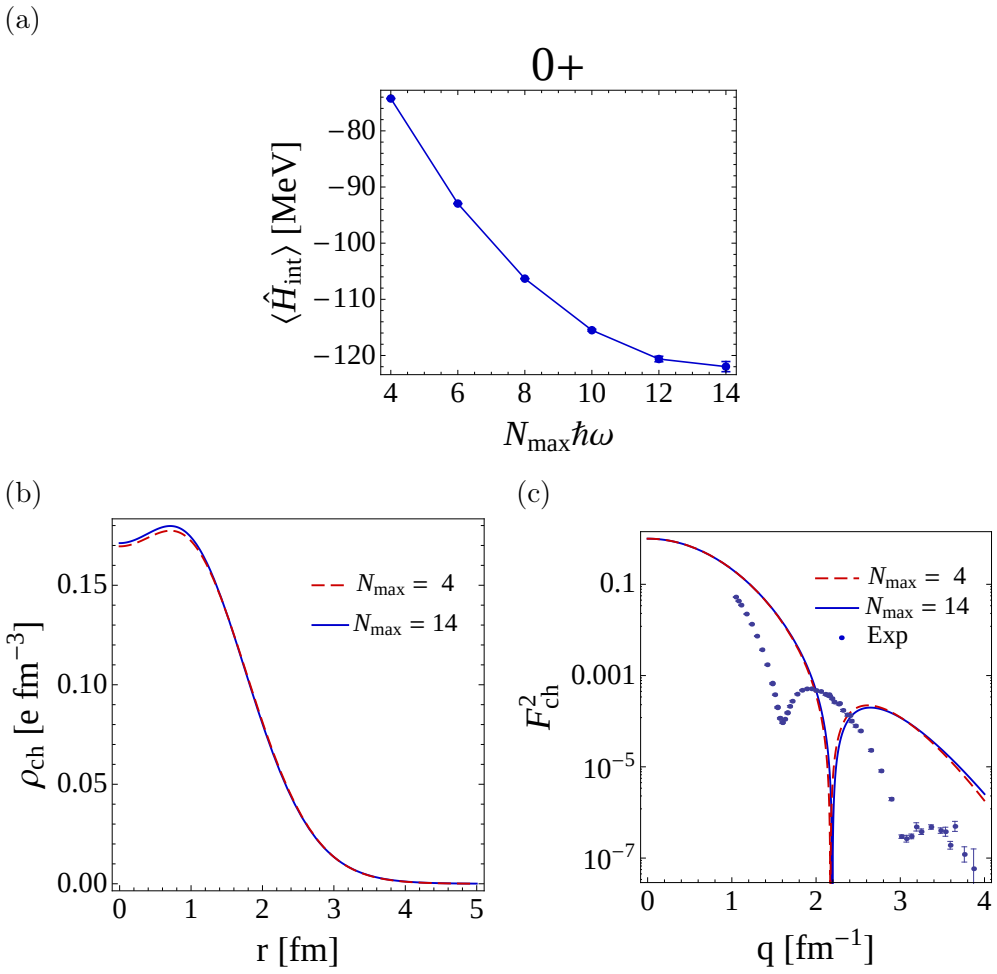


Figure 8.1: *(a) Binding energy of ^{16}O. (b) Charge density. (c) Charge form factor. Experimental values taken from [35].*

Chapter 9

Three-Body Forces

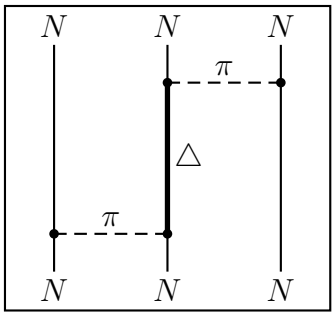
9.1 Introduction

Accurate prediction of nuclear wave functions and corresponding observables as ground state energies, excitation spectra and other is a major challenge in theoretical nuclear structure physics. With the upcoming of ever powerful computer technologies nowadays converged full NCSM calculations for light nuclei are available, offering an outstanding opportunity to confront these theoretical predictions with experiment.

Although modern phenomenological NN potentials reproduce the NN phase shifts virtually exactly, some nuclear observables calculated from these potentials may miss the experimental observations significantly. For ground state energies, the relevance of three-body forces is, for instance, reviewed in [32] where full NCSM results for ^{3}H and ^{4}He from (bare and effective) NN and NN+NNN interactions [41] obtained from chiral effective field theory , are compared. There, the NNN contribution to the ground state energy turns out to be of the order -0.6 MeV for ^{3}H and -3 MeV for ^{4}He, whereas the converged results show a significantly reduced deviation from the experimental values of around 0.01 MeV and 0.1 MeV for ^{3}H and for ^{4}He respectively.

The need of three- or even higher-body forces is not surprising since the NCSM treats the nucleon as being elementary, which one knows it is not. One consequence is the existence of excitations of the nucleon, such as the $\triangle$-isobar. If the $\triangle$ degrees of freedom are not implemented in the theory,

processes involving these excitations, such as



will naturally appear as three-nucleon interactions. The appearance of many-body interactions is a common and well-known consequence of theories in which many fundamental degrees of freedom have been reduced to a few effective ones [37].

The actual fundamental degrees of freedom for the nuclear problem are the quark and gluon fields of quantum chromodynamics. Thus, one would ultimately describe the interaction between nucleons in terms of quarks and gluons. However, nonperturbativity of QCD in the low-energy regime makes it not feasible today to derive NN interactions directly from these fundamental degrees of freedom. To connect the theoretical description of the interaction among nucleons with the underlying fundamental theory and especially its symmetries anyway, one can work in the framework of chiral effective field theory.

Chirality [38] is of great importance for the nucleon interaction. Since the masses of the up and down quark are much smaller than the relevant nucleonic mass scale - the nucleon mass - they can approximatively be treated as being massless. The free Lagrangian of massless up and down quarks

$$\mathcal{L} = i\bar{\psi}_u \partial\!\!\!/ \psi_u + i\bar{\psi}_d \partial\!\!\!/ \psi_d \tag{9.1}$$

is invariant under the chiral transformations on $\psi = (\psi_u, \psi_d)$

$$\begin{aligned} \Lambda_V : \psi &\rightarrow e^{i\frac{\boldsymbol{\tau}}{2}\boldsymbol{\Theta}}\psi \\ \Lambda_A : \psi &\rightarrow e^{-i\gamma_5\frac{\boldsymbol{\tau}}{2}\boldsymbol{\Theta}}\psi \end{aligned} \tag{9.2}$$

and so is massless two-flavor QCD. Λ_A implies an uniform meson mass spectrum, which is not observed in experiment, and, therefore, the symmetry Λ_A is spontaneously broken. The Goldstone bosons arising from this symmetry breaking turn out to be the pions. Since chiral symmetry is only approximate the pions attain mass, which, however, is still much smaller than the nucleon mass. So, with these light particles at hand, one would expect the low-energy and -momentum regime to be dominated by the pions.

Effective field theory [39, 40] allows to isolate the most relevant ingredients of a theory at a certain low energy scale and to separate these from the rest. It is defined by the most general Lagrangian involving the relevant light degrees of freedom that is consistent with the symmetries of the underlying theory. Heavy-particle exchanges between the light particles are thereby replaced by a tower of local interactions among the light field modes. The effects of the suppressed heavier degrees of freedom is then encoded in the coupling constants of the low-energy effective Lagrangian. Since chirality suggests the physics to be dominated by pions, the S-matrix elements should conveniently be expandable in pion masses and momenta. The application of this chiral perturbation theory leads to the prediction of a NNN interaction at N^2LO [41]. Together with two low-energy constants c_D, c_E one has at this order the NNN diagrams from Fig. 9.1, which have been regularized and determined in [36, 41].

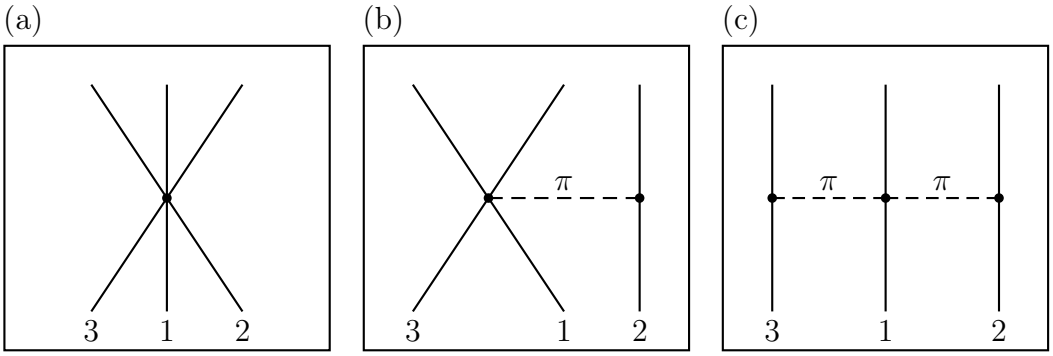


Figure 9.1: *Diagrams of the three-body nucleon interaction appearing at N^2LO of chiral effective field theory. (a) Contact interaction. (b) One-pion exchange with two-nucleon contact. (c) Two-pion exchange.*

9.2 Matrix Elements

In order to employ NNN interactions in shell model calculations using second quantization techniques, its matrix elements in the three-nucleon system

$$\langle (n(ls)jmtm_t)_{(abc)}|V^{\mathrm{NNN}}|(n(ls)jmtm_t)_{(a'b'c')}\rangle \tag{9.3}$$

have to be calculated. Translational invariance implies that the exact wave function factorize into an intrinsic part which determines the nuclear properties and a part corresponding to the dynamics of the center-of-mass

$$|\psi\rangle = |\mathrm{cm}\rangle \ \otimes \ |\mathrm{int}\rangle. \tag{9.4}$$

Since the nuclear interaction is invariant under translations its matrix elements must not depend on the particular center-of-mass state but on relative coordinates only,

$$\langle \mathrm{cm}| \ \otimes \ |\langle \mathrm{int}|V^{\mathrm{NNN}}|\mathrm{cm}'\rangle \ \otimes \ |\mathrm{int}'\rangle = \delta_{\mathrm{cm},\mathrm{cm}'} \ \langle \mathrm{int}|V^{\mathrm{NNN}}|\mathrm{int}'\rangle. \tag{9.5}$$

By formulating the three-nucleon system in a translational invariant basis, i.e. by performing a coordinate transformation from single-particle coordinates $(\boldsymbol{r}_a, \boldsymbol{r}_b, \boldsymbol{r}_c)$ to center-of-mass and relative coordinates $(\mathbf{cm}, \boldsymbol{\xi}_1, \boldsymbol{\xi}_2)$, the center-of-mass degree of freedom can explicitly be omitted in the calculations, facilitating these.

Section 9.2.1 will introduce Jacobi coordinates and the Talmi transformation, and Section 9.2.2 gives an overview over the principal procedure that will lead to the desired translational invariant formulation of the three-nucleon system, thereby introducing the notation used in the subsequent sections and therefore serving as reference. The antisymmetrization of the Jacobi basis and the transformation from the m-scheme are considered in detail in Section 9.2.3 and 9.2.4.

9.2.1 Jacobi Coordinates and Talmi Transformation

The use of Jacobi coordinates enables to separate the center-of-mass degree of freedom of a many-body system with single-particle coordinates $\boldsymbol{r}_i$ and masses m_i by introducing the center-of-mass coordinate $\mathbf{cm}$ and relative coordinates $\boldsymbol{\xi}_i$. The corresponding transformation is given in (9.6).

Set $\boldsymbol{g}_1 = \boldsymbol{r}_1,\ \mu_1 = m_1$.

$$T_k : \begin{cases} \boldsymbol{\xi}_k &= \boldsymbol{r}_k - \boldsymbol{g}_{k-1} \\ \boldsymbol{g}_k &= (m_k \boldsymbol{r}_k + \mu_{k-1}\boldsymbol{g}_{k-1})/\mu_k, \qquad k = 2, \ldots, A \\ \mu_k &= \mu_{k-1} + m_k \end{cases} \tag{9.6}$$

In this sense μ_k represents the total mass of the subsystem consisting of particles $1, \ldots, k$ and $\boldsymbol{g}_k$ is its center-of-mass position. The $\boldsymbol{\xi}_i$ are the Jacobi coordinates, where $\boldsymbol{\xi}_k$ is the position of the k-th particle relative to the center-of-mass of the previous $k-1$ particles. Using the transformation T, the original set of Cartesian coordinates $\boldsymbol{r}_1, \ldots, \boldsymbol{r}_A$ is transformed to Jacobi coordinates

$$\mathbf{cm}, \boldsymbol{\xi}_1, \ldots, \boldsymbol{\xi}_A, \tag{9.7}$$

where $\mathbf{cm} \equiv \boldsymbol{g}_A$ is the center-of-mass coordinate of the whole system.

Eq. (9.6) gives not the only possible set of Jacobi coordinates since any orthogonal transformation among the $\boldsymbol{\xi}_i$, $i = 1, \ldots, A$ yields another convenient set of coordinates. The particular choice for the following discussion is given in Fig. 9.2.

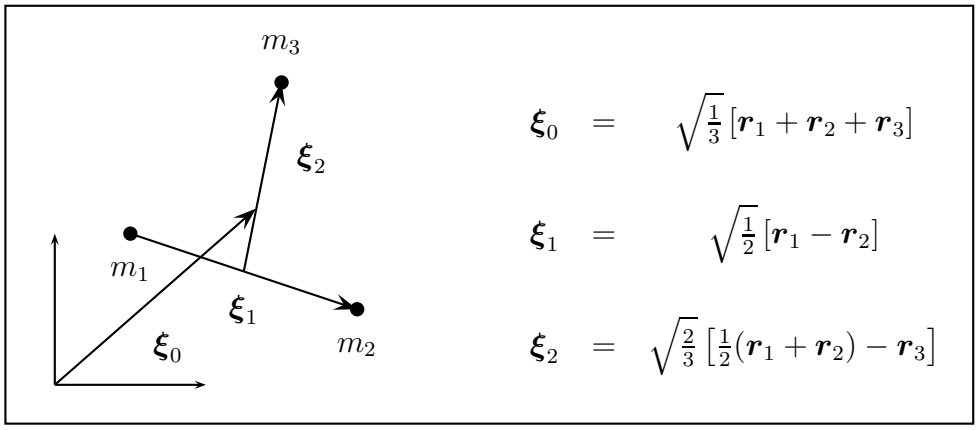


Figure 9.2: Jacobi Coordinates generated by (9.6).

For the two-nucleon system the center-of-mass degree of freedom can be separated by the well-known Talmi transformation. For this transformation

the use of harmonic oscillator wave functions is required since they have the property that a coordinate transformation $(\boldsymbol{r}_1, \boldsymbol{r}_2) \rightarrow (\boldsymbol{R}, \boldsymbol{r})$ on the two-body wave function will lead to a finite expansion of oscillator functions in the original coordinates [43],

$$|[e_1 l_1(\boldsymbol{r}_1), e_2 l_2(\boldsymbol{r}_2)]\Lambda\lambda\rangle = \sum_{EL,el} \langle [EL, el]\Lambda | [e_1 l_1, e_2 l_2]\Lambda\rangle_d \; |[EL(\boldsymbol{R}), el(\boldsymbol{r})]\Lambda\lambda\rangle \tag{9.8}$$

where the quantum numbers $e = 2n + l$ are used instead of n. Due to conservation of the total oscillator energy

$$e_1 + e_2 = E + e \tag{9.9}$$

the sum of the expansion is finite. The harmonic oscillator brackets in (9.8) that serve as expansion coefficients depend on the oscillator quantum numbers and on the particular (orthogonal) transformation of the coordinates that can be expressed by a single number d. This is because every orthogonal transformation can be represented as

$$\begin{pmatrix} \boldsymbol{R} \\ \boldsymbol{r} \end{pmatrix} = \begin{pmatrix} \sqrt{\frac{d}{1+d}} & \sqrt{\frac{1}{1+d}} \\ \sqrt{\frac{1}{1+d}} & -\sqrt{\frac{d}{1+d}} \end{pmatrix} \begin{pmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \end{pmatrix} \tag{9.10}$$

The transformation matrix M is orthogonal and symmetric, $M^{-1} = M^T = M$, and so the inverse transformation is given by the same matrix,

$$\begin{pmatrix} \boldsymbol{r}_1 \\ \boldsymbol{r}_2 \end{pmatrix} = \begin{pmatrix} \sqrt{\frac{d}{1+d}} & \sqrt{\frac{1}{1+d}} \\ \sqrt{\frac{1}{1+d}} & -\sqrt{\frac{d}{1+d}} \end{pmatrix} \begin{pmatrix} \boldsymbol{R} \\ \boldsymbol{r} \end{pmatrix}. \tag{9.11}$$

9.2.2 General Procedure

Matrix elements of the NNN interaction are available in a total antisymmetrized Jacobi basis. Since usually shell model calculations are performed in the m-scheme the goal of the preceding sections will be to develop the transformation of matrix elements between these basis sets, according to [44, 45, 46].

The transition from a three-nucleon harmonic oscillator basis with single-particle coordinates to one with Jacobi coordinates can be done by means of

two subsequent Talmi transformations. The first transformation of coordinates of particles a and b will lead from

$$|n_a l_a m_{l_a}(\boldsymbol{r}_a)\rangle \; |n_b l_b m_{l_b}(\boldsymbol{r}_b)\rangle \; |n_c l_c m_{l_c}(\boldsymbol{r}_c)\rangle \tag{9.12}$$

to

$$|\mathcal{N}_{12}\mathcal{L}_{12}\mathcal{M}_{12}(\mathbf{cm}_{12})\rangle \; |n_{12} l_{12} m_{12}(\boldsymbol{\xi}_1)\rangle \; |n_c l_c m_c(\boldsymbol{r}_c)\rangle \tag{9.13}$$

where $\mathbf{cm}_{12}$ is the center-of-mass coordinate of particles a and b and $\boldsymbol{\xi}_1$ is the first Jacobi coordinate. Single-particle coordinates are denoted by the subscript a, b, c rather than 1,2,3 since these numbers are related to Jacobi coordinates. A second Talmi transformation of the coordinates $\mathbf{cm}_{12}$ and $\boldsymbol{r}_c$ brings the desired result,

$$|n_{cm} l_{cm} m_{cm}(\mathbf{cm})\rangle \; |n_{12} l_{12} m_{12}(\boldsymbol{\xi}_1)\rangle \; |n_3 l_3 m_3(\boldsymbol{\xi}_2)\rangle, \tag{9.14}$$

with the total center-of-mass coordinate $\mathbf{cm}$ and the second Jacobi coordinate $\boldsymbol{\xi}_2$. In fact, the way these Talmi transformations are done is in accordance to T_k.

Including the spin etc. degrees of freedom one obtains a non-antisymmetrized Jacobi basis of the form

$$|n_{cm} l_{cm} m_{cm}\rangle \; |\alpha\rangle = |n_{cm} l_{cm} m_{cm}\rangle \; |n_{12} n_3 [(l_{12} s_{ab}) j_{12} (l_3 s_c) I_3] J M_J (t_{ab} t_c) T M_T\rangle \tag{9.15}$$

in which the spatial quantum numbers are as above and the (iso-) spins of particle a and b are coupled to total (iso-) spin s_{ab} (t_{ab}). This basis will be antisymmetrized by antisymmetrizing $|\alpha\rangle$ using coefficients of fractional parentage obtained from diagonalizing the antisymmetrizer χ. The result is a basis of the relative-coordinate wave function of the form $|NJTi\rangle$ so that the complete three-nucleon basis reads

$$|n_{cm} l_{cm} m_{cm}\rangle \; |NJTi\rangle \tag{9.16}$$

where $N = 2n_{12} + l_{12} + 2n_3 + l_3$ is the total oscillator energy and i is a non-physical label on the basis states.

9.2.3 Antisymmetrization

In this section the antisymmetrization of the Jacobi basis

$$|\alpha\rangle = |n_{12}n_3[(l_{12}s_{ab})j_{12}(l_3s_c)I_3]JM_J(t_{ab}t_c)TM_T\rangle \tag{9.17}$$

is discussed. The expansion coefficients are obtained by diagonalizing the antisymmetrizer $\hat{\chi}$. Since the basis (9.17) is already antisymmetric under $1 \leftrightarrow 2$ these coefficients are coefficients of fractional parentage.

The antisymmetrizer is defined as

$$\begin{aligned} \hat{\chi} &= \frac{1}{3!}\sum_{\hat{\mathcal{P}}}(-1)^P\hat{\mathcal{P}} \\ &= \frac{1}{6}\left(\hat{\mathcal{P}}_{123} - \hat{\mathcal{P}}_{132} + \hat{\mathcal{P}}_{231} - \hat{\mathcal{P}}_{213} + \hat{\mathcal{P}}_{312} - \hat{\mathcal{P}}_{321}\right). \end{aligned} \tag{9.18}$$

Antisymmetry in $1 \leftrightarrow 2$ implies $\hat{\mathcal{P}}_{ijk} = -\hat{\mathcal{P}}_{jik}$ and because of $\hat{\mathcal{P}}_{123} = \mathbb{1}$ the antisymmetizer reduces to

$$\hat{\chi} = \frac{1}{3}\left(\mathbb{1} + \hat{\mathcal{P}}_{312} + \hat{\mathcal{P}}_{231}\right). \tag{9.19}$$

The permutation operators can easily be translated into transposition operators ($[\hat{\mathcal{T}}_{ij}, \hat{\mathcal{T}}_{kl}] \neq 0$)

$$\hat{\chi} = \frac{1}{3}\left(\mathbb{1} + \hat{\mathcal{T}}_{23}\hat{\mathcal{T}}_{12} + \hat{\mathcal{T}}_{12}\hat{\mathcal{T}}_{23}\right). \tag{9.20}$$

Antisymmetry in $1 \leftrightarrow 2$ implies $\hat{\mathcal{T}}_{12} = -1$ which leads to

$$\hat{\chi} = \frac{1}{3}\left(\mathbb{1} - 2\hat{\mathcal{T}}_{23}\right). \tag{9.21}$$

So the matrix elements of $\hat{\chi}$ have essentially reduced to the ones of $\hat{\mathcal{T}}_{23}$.

In order to determine these matrix elements it is convenient to bring the Jacobi basis (9.17) into the form

$$|[n_{12}l_{12}, n_3l_3]LM_L\rangle\ |(s_{ab}s_c)SM_S\rangle\ |(t_{ab}t_c)TM_T\rangle. \tag{9.22}$$

This is because on the one hand, the action of $\hat{\mathcal{T}}_{23} \equiv \hat{\mathcal{T}}_{bc}$ on these states will be analogously for the spin and isospin part. On the other hand, the spatial part is separated what is welcome since it will be subject to a Talmi

transformation. The transformation (9.17) $\rightarrow$ (9.22) can straightforwardly be found to be

$$
\begin{aligned}
&|n_{12}n_3[(l_{12}s_{ab})j_{12}(l_3s_c)I_3]JM_J;(t_{ab}t_c)TM_T\rangle = \\
&\quad \sum_{LS}\sum_{M_LM_S} \hat{j}_{12}\ \hat{I}_3\ \hat{L}\ \hat{S} \begin{Bmatrix} l_{12} & s_{ab} & j_{12} \\ l_3 & s_c & I_3 \\ L & S & J \end{Bmatrix} \begin{pmatrix} L & S & J \\ M_L & M_S & M_J \end{pmatrix}_{CG} \\
&\quad \times |[n_{12}l_{12},n_3l_3]LM_L\rangle\ |(s_{ab}s_c)SM_S\rangle\ |(t_{ab}t_c)TM_T\rangle
\end{aligned} \tag{9.23}
$$

with $\hat{k} = \sqrt{2k+1}$and consequently the matrix elements of the transposition operator are given as

$$
\begin{aligned}
&\langle n'_{12}n'_3[(l'_{12}s'_{ab})j'_{12}(l'_3s_c)I'_3]J'M'_J;(t'_{ab}t_c)T'M'_T| \\
&\qquad \hat{\mathcal{T}}_{bc}\ \ |n_{12}n_3[(l_{12}s_{ab})j_{12}(l_3s_c)I_3]JM_J;(t_{ab}t_c)TM_T\rangle \\
&= \sum_{L'S'}\sum_{M'_LM'_S}\sum_{LS}\sum_{M_LM_S} \hat{j}'_{12}\ \hat{I}'_3\ \hat{L}'\ \hat{S}'\ \hat{j}_{12}\ \hat{I}_3\ \hat{L}\ \hat{S} \\
&\qquad \times \begin{Bmatrix} l'_{12} & s'_{ab} & j'_{12} \\ l'_3 & s_c & I'_3 \\ L' & S' & J' \end{Bmatrix} \begin{Bmatrix} l_{12} & s_{ab} & j_{12} \\ l_3 & s_c & I_3 \\ L & S & J \end{Bmatrix} \\
&\qquad \times \begin{pmatrix} L' & S' & J' \\ M'_L & M'_S & M'_J \end{pmatrix}_{CG} \begin{pmatrix} L & S & J \\ M_L & M_S & M_J \end{pmatrix}_{CG} \\
&\qquad \times \langle [n'_{12}l'_{12},n'_3l'_3]L'M'_L|\ \hat{\mathcal{T}}_{bc}\ |[n_{12}l_{12},n_3l_3]LM_L\rangle \\
&\qquad \times \langle (s'_{ab}s_c)S'M'_S|\ \hat{\mathcal{T}}_{bc}\ |(s_{ab}s_c)SM_S\rangle \\
&\qquad \times \langle (t'_{ab}t_c)T'M'_T|\ \hat{\mathcal{T}}_{bc}\ |(t_{ab}t_c)TM_T\rangle.
\end{aligned} \tag{9.24}
$$

In the following sections the isospin

$$\langle (t'_{ab}t_c)T'M'_T|\ \hat{\mathcal{T}}_{bc}\ |(t_{ab}t_c)TM_T\rangle \tag{9.25}$$

and spatial matrix element

$$\langle [n'_{12}l'_{12},n'_3l'_3]L'M'_L|\ \hat{\mathcal{T}}_{bc}\ |[n_{12}l_{12},n_3l_3]LM_L\rangle \tag{9.26}$$

are calculated.

Isospin Matrix Element

To calculate the isospin matrix element

$$\langle (t'_{ab}t_c)T'M_{T'}|\ \hat{\mathcal{T}}_{bc}\ |(t_{ab}t_c)TM_T\rangle \tag{9.27}$$

the action of $\hat{\mathcal{T}}_{bc}$ on the isospin wave function is considered,

$$
\begin{aligned}
&|[(t_a t_b)t_{ab} t_c]TM_T\rangle = \\
&\quad \sum_{m_{t_a} m_{t_b}} \sum_{m_{t_{ab}} m_{t_c}} \left(\begin{smallmatrix} t_a & t_b & t_{ab} \\ m_{t_a} & m_{t_b} & m_{t_{ab}} \end{smallmatrix} \right)_{CG} \left(\begin{smallmatrix} t_{ab} & t_c & T \\ m_{t_{ab}} & m_{t_c} & M_T \end{smallmatrix} \right)_{CG} |t_a m_{t_a} t_b m_{t_b} t_c m_{t_c}\rangle \\
&\Rightarrow \hat{\mathcal{T}}_{bc} \, |[(t_a t_b)t_{ab} t_c]TM_T\rangle = \\
&\quad \sum_{m_{t_a} m_{t_b}} \sum_{m_{t_{ab}} m_{t_c}} \left(\begin{smallmatrix} t_a & t_b & t_{ab} \\ m_{t_a} & m_{t_b} & m_{t_{ab}} \end{smallmatrix} \right)_{CG} \left(\begin{smallmatrix} t_{ab} & t_c & T \\ m_{t_{ab}} & m_{t_c} & M_T \end{smallmatrix} \right)_{CG} |t_a m_{t_a} t_c m_{t_c} t_b m_{t_b}\rangle.
\end{aligned}
\tag{9.28}
$$

From a glance on the Clebsch-Gordan coefficients it follows that it is now particles a and c (that is in the state $|t_b m_{t_b}\rangle$) that are coupled to an intermediate angular momentum, but to the same state $|t_{ab} m_{t_{ab}}\rangle$ as before.

Regarding the T-coupling, $\hat{\mathcal{T}}_{bc}$ changes the coupling order according to

$$
|[(t_a t_b)t_{ab} t_c]TM_T\rangle \quad \rightarrow \quad |[(t_a t_c)t_{ac} t_b]TM_T\rangle. \tag{9.29}
$$

The new coupling order can be expanded in the old one using $6j$-symbols,

$$
\begin{aligned}
&|[(t_a t_c)t_{ac} t_b]TM_T\rangle \\
&\quad = \sum_{\tilde{t}_{ab}} (-1)^{t_b + t_c + \tilde{t}_{ab} + t_{ac}} \, \hat{\tilde{t}}_{ab} \, \hat{t}_{ac} \, \left\{ \begin{smallmatrix} t_b & t_a & \tilde{t}_{ab} \\ t_c & T & t_{ac} \end{smallmatrix} \right\} |[(t_a t_b)\tilde{t}_{ab} t_c]TM_T\rangle \\
&\quad = \sum_{\tilde{t}_{ab}} (-1)^{t_b + t_c + \tilde{t}_{ab} + t_{ab}} \, \hat{\tilde{t}}_{ab} \, \hat{t}_{ab} \, \left\{ \begin{smallmatrix} t_b & t_a & \tilde{t}_{ab} \\ t_c & T & t_{ab} \end{smallmatrix} \right\} |[(t_a t_b)\tilde{t}_{ab} t_c]TM_T\rangle
\end{aligned}
\tag{9.30}
$$

where in the last row t_{ac} has been set to t_{ab} because of the argument above. Using (9.30) the isospin matrix element (9.27) is immediately found to be

$$
\begin{aligned}
&\langle (t'_{ab} t_c)T'M_{T'}| \, \hat{\mathcal{T}}_{bc} \, |(t_{ab} t_c)TM_T\rangle \\
&\qquad = \delta_{T'T} \, \delta_{M_{T'} M_T} \, (-1)^{1 + t'_{ab} + t_{ab}} \, \hat{t}'_{ab} \, \hat{t}_{ab} \, \left\{ \begin{smallmatrix} t_a & t_b & t'_{ab} \\ t_c & T & t_{ab} \end{smallmatrix} \right\}.
\end{aligned}
\tag{9.31}
$$

Spatial Matrix Element

For the spatial matrix element

$$
\langle [n'_{12} l'_{12}, n'_3 l'_3]L'M'_L| \, \hat{\mathcal{T}}_{bc} \, |[n_{12} l_{12}, n_3 l_3]LM_L\rangle \tag{9.32}
$$

the action of $\hat{\mathcal{T}}_{bc}$ on the spatial wave function

$$
|n_{cm} l_{cm} m_{cm}(\mathbf{cm})\rangle \, |[n_{12} l_{12}(\boldsymbol{\xi}_1), n_3 l_3(\boldsymbol{\xi}_2)]LM_L\rangle \tag{9.33}
$$

is considered. Some exercise leads to the expansion of (9.33) into the product basis of the coordinates $\boldsymbol{r}_i$

$$
\begin{aligned}
&|n_{cm}l_{cm}m_{cm}(\mathbf{cm})\rangle\ |[n_{12}l_{12}(\boldsymbol{\xi}_1), n_3l_3(\boldsymbol{\xi}_2)]LM_L\rangle \\
&= \sum_{m_{12}m_3}\sum_{\Lambda M_\Lambda}\sum_{n_cl_c\mathcal{N}_{12}\mathcal{L}_{12}}\sum_{\mathcal{M}_{12}m_c}\sum_{L_{12}M_{12}}\sum_{n_al_an_bl_b}\sum_{m_am_b} \\
&\quad\times (-1)^{l_{12}+l_3-L}\ (-1)^{\mathcal{L}_{12}+l_c-\Lambda} \\
&\quad\times \delta_{2n_{cm}+l_{cm}+2n_3+l_3,2\mathcal{N}_{12}+\mathcal{L}_{12}+2n_c+l_c} \\
&\quad\times \delta_{2n_{12}+l_{12}+2\mathcal{N}_{12}+\mathcal{L}_{12},2n_a+l_a+2n_b+l_b} \\
&\quad\times \langle[\mathcal{N}_{12}\mathcal{L}_{12}, n_cl_c]\Lambda|[n_{cm}l_{cm}, n_3l_3]\Lambda\rangle_{1/2} \\
&\quad\times \langle[n_al_a, n_bl_b]L_{12}|[n_{12}l_{12}, \mathcal{N}_{12}\mathcal{L}_{12}]L_{12}\rangle_1 \\
&\quad\times \begin{pmatrix} l_3 & l_{12} & L \\ m_3 & m_{12} & M_{12}\end{pmatrix}_{CG}\begin{pmatrix} l_{cm} & l_3 & \Lambda \\ m_{cm} & m_3 & M_\Lambda\end{pmatrix}_{CG} \\
&\quad\times \begin{pmatrix} \mathcal{L}_{12} & l_c & \Lambda \\ \mathcal{M}_{12} & m_c & M_\Lambda\end{pmatrix}_{CG}\begin{pmatrix} \mathcal{L}_{12} & l_{12} & L_{12} \\ \mathcal{M}_{12} & m_{12} & M_{12}\end{pmatrix}_{CG} \\
&\quad\times \begin{pmatrix} l_a & l_b & L_{12} \\ m_a & m_b & M_{12}\end{pmatrix}_{CG} \\
&\quad\times |n_cl_cm_c(\boldsymbol{r}_c)\rangle\ |n_al_am_a(\boldsymbol{r}_a)\rangle\ |n_bl_bm_b(\boldsymbol{r}_b)\rangle
\end{aligned}
\tag{9.34}
$$

$\hat{\mathcal{T}}_{bc}$ acts on the wave functions only,

$$
\begin{aligned}
&\hat{\mathcal{T}}_{bc}\ |n_cl_cm_c(\boldsymbol{r}_c)\rangle\ |n_al_am_a(\boldsymbol{r}_a)\rangle\ |n_bl_bm_b(\boldsymbol{r}_b)\rangle \\
&\qquad = |n_cl_cm_c(\boldsymbol{r}_b)\rangle\ |n_al_am_a(\boldsymbol{r}_a)\rangle\ |n_bl_bm_b(\boldsymbol{r}_c)\rangle.
\end{aligned}
\tag{9.35}
$$

In the product wave function, the positions of the quantum numbers have not changed and so all expansion coefficients can be used without any alteration to recast (9.34) into the form (9.32). The only difference will not be in the quantum numbers but in the coordinates since everyplace where originally a $\boldsymbol{r}_a$ occurred in (9.32) after recast there will be $\boldsymbol{r}_b$ and vice versa.

Therefore, $\hat{\mathcal{T}}_{bc}$ induces a coordinate transformation on the $\boldsymbol{\xi}_i$ defined in Fig. 9.2,

$$
\begin{aligned}
\boldsymbol{\xi}_1' &= \hat{\mathcal{T}}_{bc}\ \boldsymbol{\xi}_1 = \sqrt{\tfrac{1}{2}}[\boldsymbol{r}_a - \boldsymbol{r}_c] \\
\boldsymbol{\xi}_2' &= \hat{\mathcal{T}}_{bc}\ \boldsymbol{\xi}_2 = \sqrt{\tfrac{2}{3}}\left[\tfrac{1}{2}(\boldsymbol{r}_a + \boldsymbol{r}_c) - \boldsymbol{r}_b\right].
\end{aligned}
\tag{9.36}
$$

As in (9.10) this transformation can be represented by an orthogonal matrix and it is easily solved for d yielding $d = 1/3$. With this, the states in the

coordinates $\boldsymbol{\xi}'_i$ can be expanded in states with the original coordinates $\boldsymbol{\xi}_i$ by means of a Talmi transformation,

$$\begin{aligned}
&\hat{\mathcal{T}}_{bc}|[n_{12}l_{12}(\boldsymbol{\xi}_1), n_3l_3(\boldsymbol{\xi}_2)]LM_L\rangle \\
&\quad = \quad |[n_{12}l_{12}(\boldsymbol{\xi}'_1), n_3l_3(\boldsymbol{\xi}'_2)]LM_L\rangle \\
&\quad = \quad \sum_{\tilde{n}_{12}\tilde{l}_{12}} \sum_{\tilde{n}_3\tilde{l}_3} \delta_{2\tilde{n}_{12}+\tilde{l}_{12}+2\tilde{n}_3+\tilde{l}_3,2n_{12}+l_{12}+2n_3+l_3} \\
&\qquad\qquad \times \langle[\tilde{n}_{12}\tilde{l}_{12}, \tilde{n}_3\tilde{l}_3]L|[n_{12}l_{12}, n_3l_3]L\rangle_{1/3} \;\; |[\tilde{n}_{12}\tilde{l}_{12}, \tilde{n}_3\tilde{l}_3]LM_L\rangle. \quad (9.37)
\end{aligned}$$

Then, the matrix element (9.32) follows as

$$\begin{aligned}
&\langle[n'_{12}l'_{12}, n'_3l'_3]L'M'_L| \;\hat{\mathcal{T}}_{bc}\; |[n_{12}l_{12}, n_3l_3]LM_L\rangle \\
&\quad = \quad \delta_{L'L}\;\delta_{M'_LM_L}\;\delta_{2n'_{12}+l'_{12}+2n'_3+l'_3,2n_{12}+l_{12}+2n_3+l_3} \\
&\qquad \times \langle[n'_{12}l'_{12}, n'_3l'_3]L|[n_{12}l_{12}, n_3l_3]L\rangle_{1/3}. \qquad (9.38)
\end{aligned}$$

Total Matrix Element

With the results of the preceding sections the total matrix element of the transposition operator reads

$$\begin{aligned}
&\langle n'_{12}n'_3[(l'_{12}s'_{ab})j'_{12}(l'_3s_c)I'_3]J'M'_J(t'_{ab}t_c)T'M'_T| \\
&\qquad \hat{\mathcal{T}}_{bc} \;\; |n_{12}n_3[(l_{12}s_{ab})j_{12}(l_3s_c)I_3]JM_J(t_{ab}t_c)TM_T\rangle \\
&\quad = \quad \sum_{L'S'}\sum_{M'_LM'_S}\sum_{LS}\sum_{M_LM_S} \\
&\qquad \times \;\; \delta_{L'L}\;\delta_{M'_LM_L}\;\delta_{S'S}\;\delta_{M'_SM_S}\;\delta_{T'T}\;\delta_{M'_TM_T} \\
&\qquad \times \;\; \delta_{2n'_{12}+l'_{12}+2n'_3+l'_3,2n_{12}+l_{12}+2n_3+l_3} \\
&\qquad \times \;\; \hat{j}'_{12}\;\hat{I}'_3\;\hat{L}'\;\hat{S}'\;\hat{j}_{12}\;\hat{I}_3\;\hat{L}\;\hat{S}\;\hat{s}'_{ab}\;\hat{s}_{ab}\;\hat{t}'_{ab}\;\hat{t}_{ab} \\
&\qquad \times \;\; (-1)^{1+s'_{ab}+s_{ab}}\;(-1)^{1+t'_{ab}+t_{ab}} \\
&\qquad \times \;\; \left\{\begin{smallmatrix} s_a & s_b & s'_{ab} \\ s_c & S & s_{ab}\end{smallmatrix}\right\}\left\{\begin{smallmatrix} t_a & t_b & t'_{ab} \\ t_c & T & t_{ab}\end{smallmatrix}\right\} \\
&\qquad \times \;\; \left\{\begin{smallmatrix} l'_{12} & s'_{ab} & j'_{12} \\ l'_3 & s_c & I'_3 \\ L' & S' & J'\end{smallmatrix}\right\}\left\{\begin{smallmatrix} l_{12} & s_{ab} & j_{12} \\ l_3 & s_c & I_3 \\ L & S & J\end{smallmatrix}\right\} \\
&\qquad \times \;\; \left(\begin{smallmatrix} L' & S' & J' \\ M'_L & M'_S & M'_J\end{smallmatrix}\right)_{CG}\left(\begin{smallmatrix} L & S & J \\ M_L & M_S & M_J\end{smallmatrix}\right)_{CG} \\
&\qquad \times \;\; \langle[n'_{12}l'_{12}, n'_3l'_3]L|[n_{12}l_{12}, n_3l_3]L\rangle_{1/3}. \qquad (9.39)
\end{aligned}$$

The delta functions eliminate some summations and then the orthogonality property of the Clebsch-Gordan coefficients

$$\sum_{M_L M_S} \left(\begin{smallmatrix} L & S & J' \\ m_L & m_S & M'_J \end{smallmatrix} \right)_{CG} \left(\begin{smallmatrix} L & S & J \\ m_L & m_S & M_J \end{smallmatrix} \right)_{CG} = \delta_{J'J} \; \delta_{M'_J M_J} \tag{9.40}$$

can be used to show the diagonality in J, M_J. The final result then becomes

$$\begin{aligned}
&\langle n'_{12} n'_3 [(l'_{12} s'_{ab}) j'_{12} (l'_3 s_c) I'_3] J' M'_J (t'_{ab} t_c) T' M'_T | \\
&\qquad \hat{\mathcal{T}}_{bc} \;\; | n_{12} n_3 [(l_{12} s_{ab}) j_{12} (l_3 s_c) I_3] J M_J (t_{ab} t_c) T M_T \rangle \\
&= \; \delta_{J'J} \; \delta_{M'_J M_J} \; \delta_{T'T} \; \delta_{M'_T M_T} \; \delta_{2n'_{12}+l'_{12}+2n'_3+l'_3, 2n_{12}+l_{12}+2n_3+l_3} \\
&\qquad \times \sum_{LS} (-1)^{s'_{ab}+s_{ab}+t'_{ab}+t_{ab}} \; \hat{L}^2 \; \hat{S}^2 \; \hat{j}'_{12} \; \hat{j}_{12} \; \hat{I}'_3 \; \hat{I}_3 \; \hat{s}'_{ab} \; \hat{s}_{ab} \; \hat{t}'_{ab} \; \hat{t}_{ab} \\
&\qquad \times \left\{ \begin{smallmatrix} s_a & s_b & s'_{ab} \\ s_c & S & s_{ab} \end{smallmatrix} \right\} \left\{ \begin{smallmatrix} t_a & t_b & t'_{ab} \\ t_c & T & t_{ab} \end{smallmatrix} \right\} \left\{ \begin{smallmatrix} l'_{12} & s'_{ab} & j'_{12} \\ l'_3 & s_c & I'_3 \\ L & S & J \end{smallmatrix} \right\} \left\{ \begin{smallmatrix} l_{12} & s_{ab} & j_{12} \\ l_3 & s_c & I_3 \\ L & S & J \end{smallmatrix} \right\} \\
&\qquad \times \; \langle [n'_{12} l'_{12}, n'_3 l'_3] L | [n_{12} l_{12}, n_3 l_3] L \rangle_{1/3}.
\end{aligned} \tag{9.41}$$

Total Antisymmetric Basis

From Section 9.2.4 follows that the antisymmetrizer is diagonal in the quantum numbers T, J and $N \equiv 2n_{12} + l_{12} + 2n_3 + l_3$. Therefore, each $\boxed{\text{NJT}}$ block of dimension $\dim \boxed{\text{NJT}}$ can be diagonalized separately.

Actually, the antisymmetrizer, as well as the interaction, are also diagonal in the projection quantum numbers M_J and M_T, but since their matrix elements do not depend on these quantum numbers at all they will ultimately be represented in the basis with quantum numbers $\alpha / \{M_J, M_T\}$.

Such a $\boxed{\text{NJT}}$ block has $\dim \boxed{\text{NJT}}$ eigenvectors, but not all correspond to physical states

$$|NiJT\rangle \tag{9.42}$$

with eigenvalue 1. The other states

$$|NlJT\rangle_{\text{sp}} \tag{9.43}$$

are spurious and correspond to eigenvalues 0. The total number of physical states, or the physical dimension $\dim_p \boxed{\text{NJT}}$ of the block, can be determined

from the trace of the block since it is equal to the sum of its eigenvalues. So the physical states from a $\boxed{\text{NJT}}$ block can be written as

$$|NiJT\rangle = \sum_{k=1}^{\dim \boxed{\text{NJT}}} c_{k,i}^{(NJT)} \; |\alpha_k^{(NJT)}\rangle, \qquad i = 1, \ldots, \dim_p \boxed{\text{NJT}} \tag{9.44}$$

where $|\alpha_k^{(NJT)}\rangle$ denotes the k-th state of the basis $|\alpha/\{M_J, M_T\}\rangle$ whose quantum numbers are in accordance with N, J, T,

$$\begin{aligned} & N = 2n_{12} + l_{12} + 2n_3 + l_3 \\ & |j_{12} - I_3| \leq J \leq j_{12} + I_3 \\ & |t_{ab} - \tfrac{1}{2}| \leq T \leq t_{ab} + \tfrac{1}{2}, \end{aligned} \tag{9.45}$$

and the coefficient of fractional parentage $c_{k,i}^{(NJT)}$ is just the k-th component of the i-th physical eigenvector of the block $\boxed{\text{NJT}}$. Exploiting the diagonality of the antisymmetrizer in the quantum numbers N, J, T and the diagonality of the interaction in the quantum numbers J and T is crucial in practical computations, so that their block structure will be explicitly accounted for in the formulas by the use of notations like $\alpha_k^{(NJT)}$. A sum over the whole basis $|\alpha/\{M_J, M_T\}\rangle$, i.e., over the set of quantum numbers

$$n_{12}, l_{12}, n_3, l_3, s_{ab}, j_{12}, I_3, J, t_{ab}, T \tag{9.46}$$

can thus be represented conveniently as

$$\sum_{\alpha/\{M_J, M_T\}} f(\alpha/\{M_J, M_T\}) \;\; = \;\; \sum_{NJT} \sum_{k=1}^{\dim \boxed{\text{NJT}}} f(\alpha_k^{(NJT)}). \tag{9.47}$$

9.2.4 m-Scheme $\leftrightarrow$ Jacobi Transformation

This section considers the transformation of the interaction matrix elements from the total antisymmetrized Jacobi basis $|NJTi\rangle$ to the m-scheme

$$\langle NiJT| \; V^{\text{NNN}} \; |N'i'JT\rangle \quad \rightarrow \quad \langle abc| \; V^{\text{NNN}} \; |a'b'c'\rangle. \tag{9.48}$$

It will be convenient to couple the state

$$|abc\rangle = |n_a(l_a s_a) j_a m_a m_{t_a}\rangle |n_b(l_b s_b) j_b m_b m_{t_b}\rangle |n_c(l_c s_c) j_c m_c m_{t_c}\rangle \tag{9.49}$$

to total angular momentum $\mathcal{J}$ first,

$$|abc\rangle = \sum_{J_{12}} \sum_{\mathcal{J}} \left(\begin{smallmatrix} j_a & j_b & J_{12} \\ m_a & m_b & m_a+m_b \end{smallmatrix} \right)_{CG} \left(\begin{smallmatrix} J_{12} & j_c & \mathcal{J} \\ m_a+m_b & m_c & \mathcal{M} \end{smallmatrix} \right)_{CG} \{\{|a\rangle|b\rangle\}^{J_{12}}|c\rangle\}^{\mathcal{J}\mathcal{M}} \tag{9.50}$$

where $\mathcal{M} = m_a + m_b + m_c$. The identity

$$\mathbb{1} = \sum_{\substack{n_{cm}l_{cm} \\ m_{cm}}} \sum_{\alpha} |n_{cm}l_{cm}m_{cm}\rangle|\alpha\rangle\langle n_{cm}l_{cm}m_{cm}|\langle\alpha|, \tag{9.51}$$

where α denotes the set of quantum numbers

$$\alpha = \{n_{12}, l_{12}, n_3, l_3, s_{ab}, j_{12}, I_3, J, M_J, t_{ab}, T, M_T\}, \tag{9.52}$$

can be written as

$$\mathbb{1} = \sum_{n_{cm}l_{cm}} \sum_{\alpha} \sum_{\tilde{\mathcal{J}}\tilde{\mathcal{M}}} \{|n_{cm}l_{cm}\rangle|\alpha\rangle\}^{\tilde{\mathcal{J}}\tilde{\mathcal{M}}}\{\langle n_{cm}l_{cm}|\langle\alpha|\}^{\tilde{\mathcal{J}}\tilde{\mathcal{M}}}, \tag{9.53}$$

where α goes over into $\alpha \to \alpha/M_J$,

$$\alpha = \{n_{12}, l_{12}, n_3, l_3, s_{ab}, j_{12}, I_3, J, t_{ab}, T, M_T\}. \tag{9.54}$$

Inserting into (9.50) yields

$$\begin{aligned} |abc\rangle \;=\; \sum_{J_{12}} \sum_{\mathcal{J}} \sum_{n_{cm}l_{cm}} \sum_{\alpha} & \left(\begin{smallmatrix} j_a & j_b & J_{12} \\ m_a & m_b & m_a+m_b \end{smallmatrix} \right)_{CG} \left(\begin{smallmatrix} J_{12} & j_c & \mathcal{J} \\ m_a+m_b & m_c & \mathcal{M} \end{smallmatrix} \right)_{CG} \\ & T \left[\begin{smallmatrix} a & b & c & J_{12} \\ n_{cm} & l_{cm} & \alpha & \mathcal{J} \end{smallmatrix} \right] \{|n_{cm}l_{cm}\rangle|\alpha\rangle\}^{\mathcal{J}\mathcal{M}} \end{aligned} \tag{9.55}$$

with the overlap

$$T \left[\begin{smallmatrix} a & b & c & J_{12} \\ n_{cm} & l_{cm} & \alpha & \mathcal{J} \end{smallmatrix} \right] := \{\langle n_{cm}l_{cm}|\langle\alpha|\}^{\mathcal{J}}\{\{|a\rangle|b\rangle\}^{J_{12}}|c\rangle\}^{\mathcal{J}} \tag{9.56}$$

calculated in Section 9.2.4. $T[\cdot]$ is diagonal in the total isospin projection and so M_T from α can be fixed to be

$$M_T = \mathcal{M}_T \equiv m_{t_a} + m_{t_b} + m_{t_c}, \tag{9.57}$$

so that α goes over into $\alpha \to \alpha/M_T$,

$$\alpha = \{n_{12}, l_{12}, n_3, l_3, s_{ab}, j_{12}, I_3, J, t_{ab}, T\}. \tag{9.58}$$

Analog to (9.53) an identity can be expressed in terms of $|NiJT\rangle$,

$$\begin{aligned}
\mathbb{1} &= \sum_{\substack{\tilde{n}_{cm}\tilde{l}_{cm}\\ \tilde{m}_{cm}}} \sum_{Ni} \sum_{\tilde{J}\tilde{M}_J} \sum_{\tilde{T}} |\tilde{n}_{cm}\tilde{l}_{cm}\tilde{m}_{cm}\rangle |NiJ\tilde{M}_J\tilde{T}\rangle \langle \tilde{n}_{cm}\tilde{l}_{cm}\tilde{m}_{cm}| \langle NiJ\tilde{M}_J\tilde{T}| \\
\mathbb{1} &= \sum_{\tilde{n}_{cm}\tilde{l}_{cm}} \sum_{Ni} \sum_{\tilde{J}} \sum_{\tilde{T}} \sum_{\tilde{\mathcal{J}}\tilde{\mathcal{M}}} \{|\tilde{n}_{cm}\tilde{l}_{cm}\rangle |NiJ\tilde{T}\rangle\}^{\tilde{\mathcal{J}}\tilde{\mathcal{M}}} \{\langle \tilde{n}_{cm}\tilde{l}_{cm}| \langle NiJ\tilde{T}|\}^{\tilde{\mathcal{J}}\tilde{\mathcal{M}}}.
\end{aligned} \tag{9.59}$$

Regarding the overlap

$$\begin{aligned}
&\{\langle \tilde{n}_{cm}\tilde{l}_{cm}| \langle NiJ\tilde{T}|\}^{\tilde{\mathcal{J}}\tilde{\mathcal{M}}} \{|n_{cm}l_{cm}\rangle |\alpha\rangle\}^{\mathcal{J}\mathcal{M}} \\
&= \delta_{\tilde{\mathcal{J}}\mathcal{J}}\ \delta_{\tilde{\mathcal{M}}\mathcal{M}}\ \delta_{\tilde{J}J}\ \delta_{\tilde{n}_{cm}n_{cm}}\ \delta_{\tilde{l}_{cm}l_{cm}}\ \delta_{N,2n_{12}+l_{12}+2n_3+l_3}\ \delta_{\tilde{T}T} \\
&\quad \times\ \{\langle n_{cm}l_{cm}| \langle NiJT|\}^{\mathcal{J}\mathcal{M}} \{|n_{cm}l_{cm}\rangle |\alpha\rangle\}^{\mathcal{J}\mathcal{M}} \\
&= c_{\alpha i}\ \times\ \delta_{\tilde{\mathcal{J}}\mathcal{J}}\ \delta_{\tilde{\mathcal{M}}\mathcal{M}}\ \delta_{\tilde{J}J}\ \delta_{\tilde{n}_{cm}n_{cm}}\ \delta_{\tilde{l}_{cm}l_{cm}}\ \delta_{N,2n_{12}+l_{12}+2n_3+l_3}\ \delta_{\tilde{T}T}\ ,
\end{aligned} \tag{9.60}$$

where

$$c_{\alpha i} := \{\langle n_{cm}l_{cm}| \langle NiJT|\}^{\mathcal{J}\mathcal{M}} \{|n_{cm}l_{cm}\rangle |\alpha\rangle\}^{\mathcal{J}\mathcal{M}}, \tag{9.61}$$

inserting (9.59) into (9.55) leads to the final expansion

$$\begin{aligned}
|abc\rangle &= \sum_{J_{12}} \sum_{\mathcal{J}} \sum_{n_{cm}l_{cm}} \sum_{\alpha} \sum_{i} \left(\begin{smallmatrix} j_a & j_b & J_{12} \\ m_a & m_b & m_a+m_b \end{smallmatrix}\right)_{CG} \left(\begin{smallmatrix} J_{12} & j_c & \mathcal{J} \\ m_a+m_b & m_c & \mathcal{M} \end{smallmatrix}\right)_{CG} \\
&\qquad T \left[\begin{smallmatrix} a & b & c & J_{12} \\ n_{cm} & l_{cm} & \alpha & \mathcal{J} \end{smallmatrix}\right] c_{\alpha i}\ \{|n_{cm}l_{cm}\rangle |NiJT\rangle\}^{\mathcal{J}\mathcal{M}}
\end{aligned}$$

$$\begin{aligned}
\mathcal{M} &= m_a + m_b + m_c \\
N &= 2n_{12} + l_{12} + 2n_3 + l_3 \\
\alpha &= \{n_{12}, l_{12}, n_3, l_3, s_{ab}, j_{12}, I_3, J, t_{ab}, T\}.
\end{aligned} \tag{9.62}$$

According to [46], the overlaps

$$c_{\alpha i} := \{\langle n_{cm}l_{cm}| \langle NiJT|\}^{\mathcal{J}\mathcal{M}} \{|n_{cm}l_{cm}\rangle |\alpha\rangle\}^{\mathcal{J}\mathcal{M}}, \tag{9.63}$$

are identical to

$$\langle NiJT|\alpha\rangle \tag{9.64}$$

so that these are the coefficients of fractional parentage obtained from diagonalizing the antisymmetrizer in Section 9.2.3.

Matrix Element

Using expansion (9.62), the matrix element (9.48) reads

$$
\begin{aligned}
&\langle abc|V^{\mathrm{NNN}}|a'b'c'\rangle \\
&= \sum_{J_{12}}\sum_{\mathcal{J}}\sum_{n_{cm}l_{cm}}\sum_{\alpha}\sum_{i}\sum_{J'_{12}}\sum_{\mathcal{J}'}\sum_{n'_{cm}l'_{cm}}\sum_{\alpha'}\sum_{i'} \\
&\quad\times \begin{pmatrix} j_a & j_b & J_{12} \\ m_a & m_b & m_a+m_b \end{pmatrix}_{CG} \begin{pmatrix} J_{12} & j_c & \mathcal{J} \\ m_a+m_b & m_c & \mathcal{M} \end{pmatrix}_{CG} \\
&\quad\times \begin{pmatrix} j'_a & j'_b & J'_{12} \\ m'_a & m'_b & m'_a+m'_b \end{pmatrix}_{CG} \begin{pmatrix} J'_{12} & j'_c & \mathcal{J}' \\ m'_a+m'_b & m'_c & \mathcal{M}' \end{pmatrix}_{CG} \\
&\quad\times T\begin{bmatrix} a & b & c & J_{12} \\ n_{cm} & l_{cm} & \alpha & \mathcal{J} \end{bmatrix} T\begin{bmatrix} a' & b' & c' & J'_{12} \\ n'_{cm} & l'_{cm} & \alpha' & \mathcal{J}' \end{bmatrix} \; c_{\alpha i}\; c_{\alpha' i'} \\
&\quad\times \{\langle n_{cm}l_{cm}|\langle NiJT|\}^{\mathcal{J}\mathcal{M}}\; V^{\mathrm{NNN}}\; \{|n'_{cm}l'_{cm}\rangle|N'i'J'T'\rangle\}^{\mathcal{J}'\mathcal{M}'} \qquad (9.65)
\end{aligned}
$$

The matrix element in the last line has to be expressed in terms of the uncoupled basis,

$$
\begin{aligned}
&\{\langle n_{cm}l_{cm}|\langle NiJT|\}^{\mathcal{J}\mathcal{M}}\; V^{\mathrm{NNN}}\; \{|n'_{cm}l'_{cm}\rangle|N'i'J'T'\rangle\}^{\mathcal{J}'\mathcal{M}'} \\
&= \sum_{m_{cm}M_J}\sum_{m'_{cm}M'_J} \begin{pmatrix} l_{cm} & J & \mathcal{J} \\ m_{cm} & M_J & \mathcal{M} \end{pmatrix}_{CG} \begin{pmatrix} l'_{cm} & J' & \mathcal{J}' \\ m'_{cm} & M'_J & \mathcal{M}' \end{pmatrix}_{CG} \\
&\quad\times \langle n_{cm}l_{cm}m_{cm}|n'_{cm}l'_{cm}m'_{cm}\rangle\langle NiJM_JT|\; V^{\mathrm{NNN}}\; |N'i'J'M'_JT'\rangle \\
&= \sum_{m_{cm}M_J}\sum_{m'_{cm}M'_J} \begin{pmatrix} l_{cm} & J & \mathcal{J} \\ m_{cm} & M_J & \mathcal{M} \end{pmatrix}_{CG} \begin{pmatrix} l'_{cm} & J' & \mathcal{J}' \\ m'_{cm} & M'_J & \mathcal{M}' \end{pmatrix}_{CG} \\
&\quad\times \delta_{n_{cm}n'_{cm}}\; \delta_{l_{cm}l'_{cm}}\; \delta_{m_{cm}m'_{cm}}\; \delta_{JJ'}\; \delta_{M_JM'_J}\; \delta_{TT'} \\
&\quad\times \langle NiJM_JT|\; V^{\mathrm{NNN}}\; |N'Ji'M_JT\rangle \\
&= \delta_{n_{cm}n'_{cm}}\; \delta_{l_{cm}l'_{cm}}\; \delta_{m_{cm}m'_{cm}}\; \delta_{JJ'}\; \delta_{TT'} \\
&\quad \sum_{m_{cm}M_J} \begin{pmatrix} l_{cm} & J & \mathcal{J} \\ m_{cm} & M_J & \mathcal{M} \end{pmatrix}_{CG} \begin{pmatrix} l_{cm} & J & \mathcal{J}' \\ m_{cm} & M_J & \mathcal{M}' \end{pmatrix}_{CG} \\
&\quad\times \langle NiJM_JT|\; V^{\mathrm{NNN}}\; |N'i'JM_JT\rangle \qquad (9.66)
\end{aligned}
$$

where the diagonality of V^{NNN} in the quantum numbers J, M_J, T has been used. Furthermore, the interaction is independent from M_J, rather than only diagonal, and so the matrix element

$$
\langle NiJM_JT|\; V^{\mathrm{NNN}}\; |N'i'JM_JT\rangle = \langle NiJT|\; V^{\mathrm{NNN}}\; |N'i'JT\rangle \qquad (9.67)
$$

can be pulled in front of the sum. Then once more the orthogonality of Clebsch-Gordan coefficients eliminates the sum and the symbols,

$$
\begin{aligned}
&\{\langle n_{cm} l_{cm}|\langle NiJT|\}^{\mathcal{J}\mathcal{M}}\ V^{\rm NNN}\ \{|n'_{cm} l'_{cm}\rangle|N'i'J'T'\rangle\}^{\mathcal{J}'\mathcal{M}'} \\
&= \delta_{n_{cm}n'_{cm}}\ \delta_{l_{cm}l'_{cm}}\ \delta_{m_{cm}m'_{cm}}\ \delta_{JJ'}\ \delta_{TT'} \\
&\quad \times\ \langle NiJT|\ V^{\rm NNN}\ |N'i'JT\rangle \\
&\quad \sum_{m_{cm}M_J} \left(\begin{smallmatrix} l_{cm} & J & \mathcal{J} \\ m_{cm} & M_J & \mathcal{M} \end{smallmatrix}\right)_{CG} \left(\begin{smallmatrix} l_{cm} & J & \mathcal{J}' \\ m_{cm} & M_J & \mathcal{M}' \end{smallmatrix}\right)_{CG} \\
&= \delta_{n_{cm}n'_{cm}}\ \delta_{l_{cm}l'_{cm}}\ \delta_{m_{cm}m'_{cm}}\ \delta_{JJ'}\ \delta_{TT'}\ \delta_{\mathcal{J}\mathcal{J}'}\ \delta_{\mathcal{M}\mathcal{M}'} \\
&\quad \times\ \langle NiJT|\ V^{\rm NNN}\ |N'i'JT\rangle. \qquad (9.68)
\end{aligned}
$$

Inserting (9.66) into (9.65), and replacing $\mathcal{M}$ by $m_a + m_b + m_c$ according to (9.62), yields

$$
\begin{aligned}
&\langle abc|V^{\rm NNN}|a'b'c'\rangle \\
&= \sum_{n_{cm}l_{cm}} \sum_{\mathcal{J}} \sum_{J_{12}J'_{12}} \sum_{\alpha\alpha'} \sum_{ii'} \\
&\quad \times\ \left(\begin{smallmatrix} j_a & j_b & J_{12} \\ m_a & m_b & m_a{+}m_b \end{smallmatrix}\right)_{CG} \left(\begin{smallmatrix} J_{12} & j_c & \mathcal{J} \\ m_a{+}m_b & m_c & m_a{+}m_b{+}m_c \end{smallmatrix}\right)_{CG} \\
&\quad \times\ \left(\begin{smallmatrix} j'_a & j'_b & J'_{12} \\ m'_a & m'_b & m'_a{+}m'_b \end{smallmatrix}\right)_{CG} \left(\begin{smallmatrix} J'_{12} & j'_c & \mathcal{J} \\ m'_a{+}m'_b & m'_c & m_a{+}m_b{+}m_c \end{smallmatrix}\right)_{CG} \\
&\quad \times\ T\left[\begin{smallmatrix} a & b & c & J_{12} & J & \mathcal{J} \\ n_{cm} & l_{cm} & n_{12} & l_{12} & n_3 & l_3 \\ s_{ab} & j_{12} & I_3 & t_{ab} & T & \end{smallmatrix}\right] T\left[\begin{smallmatrix} a' & b' & c' & J'_{12} & J & \mathcal{J} \\ n_{cm} & l_{cm} & n'_{12} & l'_{12} & n'_3 & l'_3 \\ s'_{ab} & j'_{12} & I'_3 & t'_{ab} & T & \end{smallmatrix}\right] \\
&\quad \times\ c\left[\begin{smallmatrix} n_{12} & l_{12} & n_3 & l_3 \\ s_{ab} & j_{12} & I_3 & J \\ t_{ab} & T & & i \end{smallmatrix}\right]\ c\left[\begin{smallmatrix} n'_{12} & l'_{12} & n'_3 & l'_3 \\ s'_{ab} & j'_{12} & I'_3 & J \\ t'_{ab} & T & & i' \end{smallmatrix}\right] \\
&\quad \times\ \langle NiJT|\ V^{\rm NNN}\ |N'i'JT\rangle
\end{aligned}
$$

$$
\begin{aligned}
N &= 2n_{12} + l_{12} + 2n_3 + l_3 \\
N' &= 2n'_{12} + l'_{12} + 2n'_3 + l'_3 \\
\alpha &= \{n_{12}, n_3, l_{12}, s_{ab}, j_{12}, l_3, I_3, J, t_{ab}, T, M_T\} \\
\alpha' &= \{n'_{12}, n'_3, l'_{12}, s'_{ab}, j'_{12}, l'_3, I'_3, t'_{ab}\}. \qquad (9.69)
\end{aligned}
$$

Some of the delta functions in (9.66) eliminated sums over indices contained in α' so that is has been redefined.

Here two δ functions will be included that come from $T[\cdot]$, arising from the fact that there is no non-vanishing overlap between states with different

oscillator energy. Therefore, the summation range over oscillator quantum numbers in (9.51)

$$\sum_{n_{cm}l_{cm}} \sum_{n_{12}l_{12}} \sum_{n_3 l_3} \sum_{n'_{12}l'_{12}} \sum_{n'_3 l'_3} \tag{9.70}$$

is constrained by

$$\begin{aligned} 2n_{cm} + l_{cm} + 2n_{12} + l_{12} + 2n_3 + l_3 &= 2n_a + l_a + 2n_b + l_b + 2n_c + l_c \\ 2n_{cm} + l_{cm} + 2n'_{12} + l'_{12} + 2n'_3 + l'_3 &= 2n'_a + l'_a + 2n'_b + l'_b + 2n'_c + l'_c. \end{aligned} \tag{9.71}$$

In summary, the final form of the transformation of the interaction matrix elements from the total antisymmetric Jacobi basis to the m-scheme reads

$$\begin{aligned} &\langle abc|V^{\rm NNN}|a'b'c'\rangle \\ &= \sum_{n_{cm}l_{cm}} \sum_{\mathcal{J}} \sum_{J_{12}J'_{12}} \sum_{\alpha\alpha'} \sum_{ii'} \\ &\quad \times \ \delta_{2n_{cm}+l_{cm}+2n_{12}+l_{12}+2n_3+l_3,2n_a+l_a+2n_b+l_b+2n_c+l_c} \\ &\quad \times \ \delta_{2n_{cm}+l_{cm}+2n'_{12}+l'_{12}+2n'_3+l'_3,2n'_a+l'_a+2n'_b+l'_b+2n'_c+l'_c} \\ &\quad \times \ \left(\begin{smallmatrix} j_a & j_b & J_{12} \\ m_a & m_b & m_a+m_b \end{smallmatrix}\right)_{CG} \left(\begin{smallmatrix} J_{12} & j_c & \mathcal{J} \\ m_a+m_b & m_c & m_a+m_b+m_c \end{smallmatrix}\right)_{CG} \\ &\quad \times \ \left(\begin{smallmatrix} j'_a & j'_b & J'_{12} \\ m'_a & m'_b & m'_a+m'_b \end{smallmatrix}\right)_{CG} \left(\begin{smallmatrix} J'_{12} & j'_c & \mathcal{J} \\ m'_a+m'_b & m'_c & m_a+m_b+m_c \end{smallmatrix}\right)_{CG} \\ &\quad \times \ T \begin{bmatrix} a & b & c & J_{12} & J & \mathcal{J} \\ n_{cm} & l_{cm} & n_{12} & l_{12} & n_3 & l_3 \\ s_{ab} & j_{12} & I_3 & t_{ab} & T & \end{bmatrix} T \begin{bmatrix} a' & b' & c' & J'_{12} & J & \mathcal{J} \\ n_{cm} & l_{cm} & n'_{12} & l'_{12} & n'_3 & l'_3 \\ s'_{ab} & j'_{12} & I'_3 & t'_{ab} & T & \end{bmatrix} \\ &\quad \times \ c \begin{bmatrix} n_{12} & l_{12} & n_3 & l_3 & \\ s_{ab} & j_{12} & I_3 & J & \\ t_{ab} & T & & & i \end{bmatrix} \ c \begin{bmatrix} n'_{12} & l'_{12} & n'_3 & l'_3 & \\ s'_{ab} & j'_{12} & I'_3 & J & \\ t'_{ab} & T & & & i' \end{bmatrix} \\ &\quad \times \ \langle NiJT| \ V^{\rm NNN} \ |N'i'JT\rangle \end{aligned}$$

$$\begin{aligned} N &= 2n_{12} + l_{12} + 2n_3 + l_3 \\ N' &= 2n'_{12} + l'_{12} + 2n'_3 + l'_3 \\ \alpha &= \{n_{12}, n_3, l_{12}, s_{ab}, j_{12}, l_3, I_3, J, t_{ab}, T, M_T\} \\ \alpha' &= \{n'_{12}, n'_3, l'_{12}, s'_{ab}, j'_{12}, l'_3, I'_3, t'_{ab}\}. \end{aligned} \tag{9.72}$$

As mentioned in Section 9.2.3, it is advantageous to use the $|\alpha^{(NJT)}\rangle$ representation of $|\alpha\rangle$. Regarding the different sets of quantum numbers contained

in α and α' in Eq. (9.72), their sums are replaced according to

$$\sum_{\alpha}\sum_{\alpha'} f(\alpha)\, f(\alpha') \;\rightarrow\; \sum_{NJT}\sum_{k}\sum_{N'}\sum_{k'} f(\alpha_k^{(NJT)})\, f(\alpha_{k'}^{(N'JT)}). \quad (9.73)$$

$$\begin{aligned}
&\langle abc|V^{\rm NNN}|a'b'c'\rangle \\
&= \sum_{n_{cm}l_{cm}}\sum_{\mathcal{J}}\sum_{J_{12}J'_{12}}\sum_{NN'}\sum_{JT}\sum_{kk'}\sum_{ii'} \\
&\quad\times\ \delta_{2n_{cm}+l_{cm}+2n_{12}+l_{12}+2n_3+l_3,2n_a+l_a+2n_b+l_b+2n_c+l_c} \\
&\quad\times\ \delta_{2n_{cm}+l_{cm}+2n'_{12}+l'_{12}+2n'_3+l'_3,2n'_a+l'_a+2n'_b+l'_b+2n'_c+l'_c} \\
&\quad\times\ \begin{pmatrix} j_a & j_b & J_{12} \\ m_a & m_b & m_a+m_b \end{pmatrix}_{CG} \begin{pmatrix} J_{12} & j_c & \mathcal{J} \\ m_a+m_b & m_c & m_a+m_b+m_c \end{pmatrix}_{CG} \\
&\quad\times\ \begin{pmatrix} j'_a & j'_b & J'_{12} \\ m'_a & m'_b & m'_a+m'_b \end{pmatrix}_{CG} \begin{pmatrix} J'_{12} & j'_c & \mathcal{J} \\ m'_a+m'_b & m'_c & m_a+m_b+m_c \end{pmatrix}_{CG} \\
&\quad\times\ T\begin{bmatrix} a & b & c \\ n_{cm} & l_{cm} & \alpha_k^{(NJT)} \\ \mathcal{J} & J_{12} & \end{bmatrix}\ T\begin{bmatrix} a' & b' & c' \\ n_{cm} & l_{cm} & \alpha_{k'}^{(N'JT)} \\ \mathcal{J} & J'_{12} & \end{bmatrix} \\
&\quad\times\ c_{k,i}^{(NJT)}\ c_{k',i'}^{(N'JT)} \\
&\quad\times\ \langle NiJT|\ V^{\rm NNN}\ |N'i'JT\rangle. \qquad (9.74)
\end{aligned}$$

Further simplification can be obtained from defining a new symbol $\tilde{T}$ via

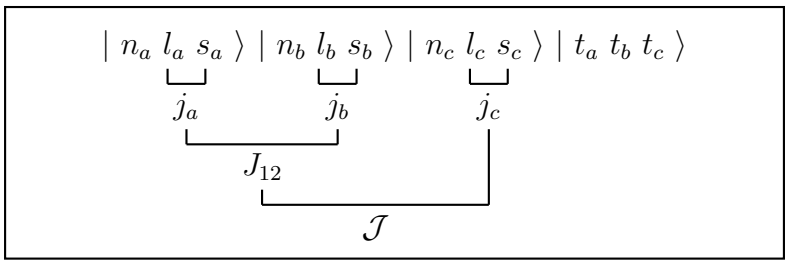
$$\begin{aligned}
\tilde{T}\begin{bmatrix} a & b & c \\ n_{cm} & l_{cm} & \alpha_k^{(NJT)} \\ \mathcal{J} & & \end{bmatrix} &= \sum_{J_{12}} \begin{pmatrix} j_a & j_b & J_{12} \\ m_a & m_b & m_a+m_b \end{pmatrix}_{CG} \begin{pmatrix} J_{12} & j_c & \mathcal{J} \\ m_a+m_b & m_c & m_a+m_b+m_c \end{pmatrix}_{CG} \\
&\quad\times\ T\begin{bmatrix} a & b & c \\ n_{cm} & l_{cm} & \alpha_k^{(NJT)} \\ \mathcal{J} & J_{12} & \end{bmatrix}. \qquad (9.75)
\end{aligned}$$

The Overlap T

The overlap

$$T\begin{bmatrix} a & b & c & J_{12} & J & \mathcal{J} \\ n_{cm} & l_{cm} & n_{12} & l_{12} & n_3 & l_3 \\ s_{ab} & j_{12} & I_3 & t_{ab} & T & \end{bmatrix} := \{\langle n_{cm}l_{cm}|\langle\alpha|\}^{\mathcal{J}}\{\{|a\rangle|b\rangle\}^{J_{12}}|c\rangle\}^{\mathcal{J}} \qquad (9.76)$$

will be evaluated by expanding the state $\{\{|a\rangle|b\rangle\}^{J_{12}}|c\rangle\}^{\mathcal{J}}$ in the basis $\{|n_{cm}l_{cm}\rangle|\alpha\rangle\}^{\mathcal{J}}$. To do so, a number of transformations have to be performed that will be discussed in the following. So, starting with $\{\{|a\rangle|b\rangle\}^{J_{12}}|c\rangle\}^{\mathcal{J}}$,

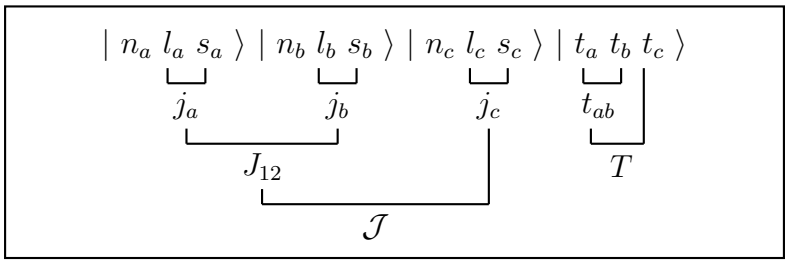


at first the isospins are coupled to a total T, according to

$$|t_a m_{t_a} t_b m_{t_b} t_c m_{t_c}\rangle \quad \rightarrow \quad |[(t_a t_b) t_{ab} t_c] T m_{t_a} + m_{t_b} + m_{t_c}\rangle \tag{9.77}$$

by means of Clebsch-Gordan coefficients,

$$\begin{aligned} &|t_a m_{t_a} t_b m_{t_b} t_c m_{t_c}\rangle \\ &\quad = \sum_{t_{ab}} \sum_{T M_T} \left(\begin{smallmatrix} t_a & t_b & t_{ab} \\ m_{t_a} & m_{t_b} & m_{t_a}+m_{t_b} \end{smallmatrix} \right)_{CG} \left(\begin{smallmatrix} t_{ab} & t_c & T \\ m_{t_a}+m_{t_b} & m_{t_c} & M_T \end{smallmatrix} \right)_{CG} |[(t_a t_b) t_{ab} t_c] T M_T\rangle \end{aligned} \tag{9.78}$$

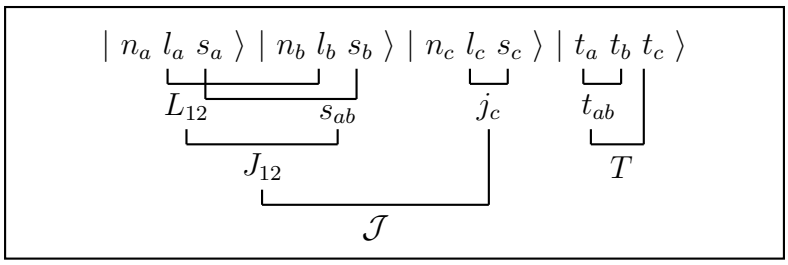


Then the single-particle spins and orbital angular momenta of particles 1 and 2 have to be decoupled in order to be able to perform the Talmi transformation, for which their orbital angular momenta have to be coupled to a total orbital angular momentum L_{12}. This de- and re-coupling can be done in one single transformation, taking advantage of the J_{12}-coupling. All what is needed is a change of the J_{12}-coupling order according to

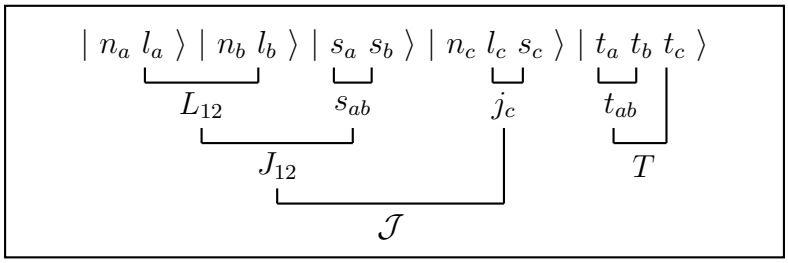
$$|[(l_a s_a) j_a (l_b s_b) j_b] J_{12}\rangle \quad \rightarrow \quad |[(l_a l_b) L_{12} (s_a s_b) s_{ab}] J_{12}\rangle \tag{9.79}$$

where the new intermediate angular momenta L_{12} and s_{ab} are introduced. This is most easily obtained using $9j$-symbols,

$$\begin{aligned} &|[(l_a s_a) j_a (l_b s_b) j_b] J_{12}\rangle = \\ &\qquad \sum_{L_{12} s_{ab}} \hat{j}_a \hat{j}_b \hat{L}_{12} \hat{s}_{12} \left\{ \begin{smallmatrix} l_a & s_a & j_a \\ l_b & s_b & j_b \\ L_{12} & s_{ab} & J_{12} \end{smallmatrix} \right\} |[(l_a l_b) L_{12} (s_a s_b) s_{ab}] J_{12}\rangle . \end{aligned} \tag{9.80}$$

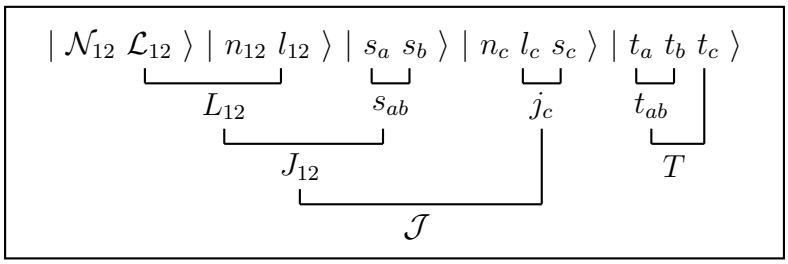


Rearrangement of wave function parts for optical convenience gives :



The first Talmi transformation transforms the single-particle coordinates $\boldsymbol{r}_a, \boldsymbol{r}_b$ into the relative (Jacobi) coordinate $\boldsymbol{\xi}_1$ and the center-of-mass coordinate for particles 1 and 2, $\mathbf{cm}_{12}$, according to (REF)

$$
\begin{aligned}
&|[n_a l_a(\boldsymbol{r}_a), n_b l_b(\boldsymbol{r}_b)] L_{12}\rangle = \\
&\sum_{\mathcal{N}_{12}\mathcal{L}_{12} n_{12} l_{12}} \quad \delta_{2\mathcal{N}_{12}+\mathcal{L}_{12}+2n_{12}+l_{12},2n_a+l_a+2n_b+l_b} \\
&\times \quad \langle [\mathcal{N}_{12}\mathcal{L}_{12}, n_{12} l_{12}] L_{12} | [n_a l_a(\boldsymbol{r}_a), n_b l_b(\boldsymbol{r}_b)] L_{12}\rangle_1 \\
&\times \quad |[\mathcal{N}_{12}\mathcal{L}_{12}(\mathbf{cm}_{12}), n_{12} l_{12}(\boldsymbol{\xi}_1)] L_{12}\rangle, \qquad (9.81)
\end{aligned}
$$



The second Talmi transformation will be about the wave functions $|\mathcal{N}_{12}\mathcal{L}_{12}\rangle$ and $|n_c l_c\rangle$. In preparation for that the $|(l_c s_c) j_c\rangle$-coupling has to be broken up and it is convenient by doing so to rearrange the $\mathcal{J}$-coupling in a way

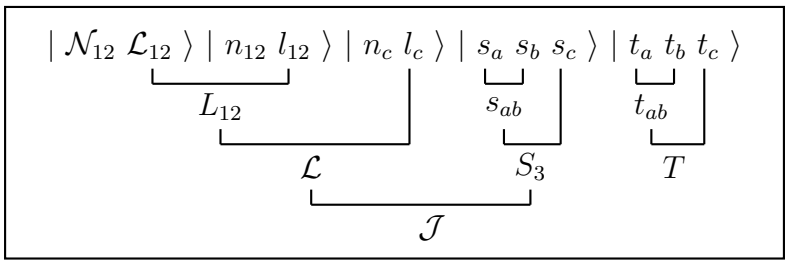
that all three spins and orbital angular momenta each are coupled to total angular momenta $\mathcal{L}$ and S_3, according to

$$|[(L_{12}s_{ab})J_{12}(l_cs_c)j_c]\mathcal{J}\rangle \quad \rightarrow \quad |[(L_{12}l_c)\mathcal{L}(s_{ab}s_c)S_3]\mathcal{J}\rangle. \tag{9.82}$$

Again, this rearrangement is easily obtained employing $9j$-symbols

$$|[(L_{12}s_{ab})J_{12}(l_cs_c)j_c]\mathcal{J}\rangle = \sum_{\mathcal{L}S_3} \hat{J}_{12}\hat{j}_c\hat{\mathcal{L}}\hat{S}_3 \left\{ \begin{smallmatrix} L_{12} & s_{ab} & J_{12} \\ l_c & s_c & j_c \\ \mathcal{L} & S_3 & \mathcal{J} \end{smallmatrix} \right\} |[(L_{12}l_c)\mathcal{L}(s_{ab}s_c)S_3]\mathcal{J}\rangle \tag{9.83}$$

arriving at:

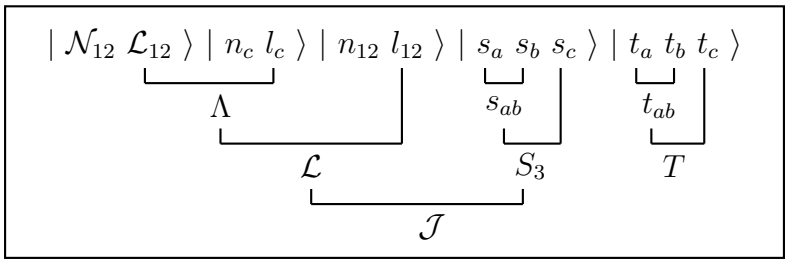


For the second Talmi transformation that transforms the coordinates $\mathbf{cm}_{12}$ and $\boldsymbol{r}_c$ into the Jacobi coordinate $\boldsymbol{\xi}_2$ and the total center-of-mass coordinate $\boldsymbol{\xi}_0$, the orbital angular momenta of $|\mathcal{N}_{12}\mathcal{L}_{12}\rangle$ and $|n_cl_c\rangle$ need to be coupled to a total orbital angular momentum Λ. For this purpose, the $\mathcal{L}$-coupling order is changed according to

$$|[(\mathcal{L}_{12}l_{12})L_{12}l_c]\mathcal{L}\rangle \quad \rightarrow \quad |[(\mathcal{L}_{12}l_c)\Lambda l_{12}]\mathcal{L}\rangle. \tag{9.84}$$

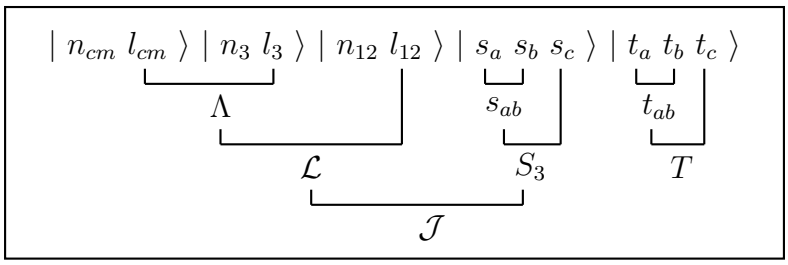
Since this is a reordering of a three-momentum coupling the transformation coefficients are given by $6j$-symbols as usual,

$$|[(\mathcal{L}_{12}l_{12})L_{12}l_c]\mathcal{L}\rangle = \sum_{\Lambda}(-1)^{l_{12}+l_c+L_{12}\Lambda}\hat{L}_{12}\hat{\Lambda} \left\{ \begin{smallmatrix} l_c & \mathcal{L}_{12} & \Lambda \\ l_{12} & \mathcal{L} & L_{12} \end{smallmatrix} \right\} |[(\mathcal{L}_{12}l_c)\Lambda l_{12}]\mathcal{L}]\rangle, \tag{9.85}$$



The second Talmi transformation separates the three-body center-of-mass from the relative-coordinate part

$$
\begin{aligned}
&|[\mathcal{N}_{12}\mathcal{L}_{12}(\mathbf{cm}_{12}), n_c l_c(\boldsymbol{r}_c)]\Lambda\rangle = \\
&\quad \sum_{n_{cm} l_{cm} n_3 l_3} \delta_{2\mathcal{N}_{12}+\mathcal{L}_{12}+2n_c+l_c, 2n_{cm}+l_{cm}+2n_3+l_3} \\
&\quad \times \quad \langle [n_{cm} l_{cm}, n_3 l_3]\Lambda | [\mathcal{N}_{12}\mathcal{L}_{12}, n_c l_c]\Lambda\rangle_2 \\
&\quad \times \quad |[n_{cm} l_{cm}(\boldsymbol{\xi}_0), n_3 l_3(\boldsymbol{\xi}_2)]\Lambda\rangle,
\end{aligned}
\tag{9.86}
$$

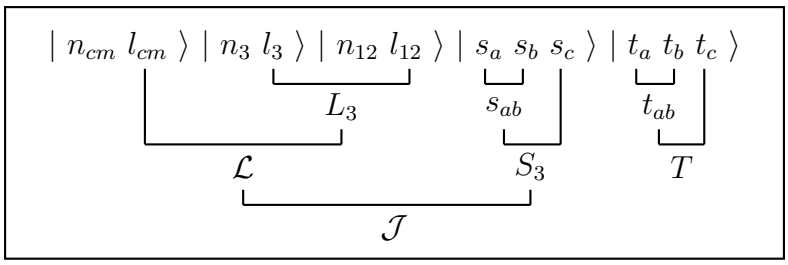


What is left after the Talmi transformations is to recast this state into the form of the left state in (9.76). Since there the center-of-mass angular momentum is not subject to a intermediate coupling in the $\mathcal{J}$-coupling, the Λ-coupling is broken up first by a reordering of the $\mathcal{L}$-coupling according to

$$
|[(l_{cm} l_3)\Lambda]\mathcal{L}\rangle \quad \rightarrow \quad |[l_{cm}(l_3 l_{12})L_3]\mathcal{L}\rangle \tag{9.87}
$$

by the use of $6j$-symbols

$$
\begin{aligned}
&|[(l_{cm} l_3)\Lambda]\mathcal{L}\rangle = \\
&\quad \sum_{L_3} (-1)^{l_{cm}+l_3+l_{12}+\mathcal{L}} \hat{\Lambda}\hat{L}_3 \left\{ \begin{smallmatrix} l_{cm} & l_3 & \Lambda \\ l_{12} & \mathcal{L} & L_3 \end{smallmatrix} \right\} |[l_{cm}(l_3 l_{12})L_3]\mathcal{L}\rangle,
\end{aligned}
\tag{9.88}
$$

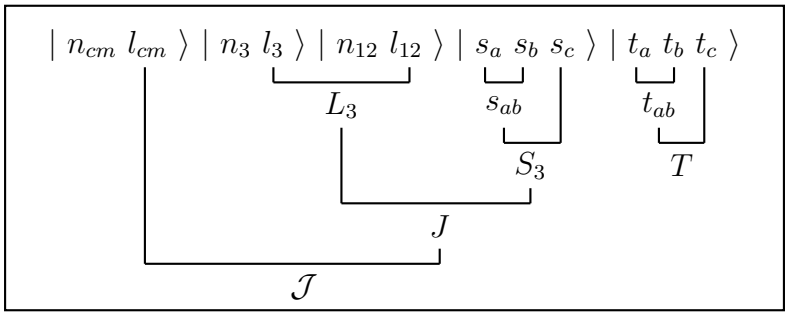


For the same reason the $\mathcal{L}$-coupling involving l_{cm} is broken up according to

$$|[(l_{cm}L_3)\mathcal{L}S_3]\mathcal{J}\rangle \quad \rightarrow \quad |[l_{cm}(L_3S_3)J]\mathcal{J}\rangle \tag{9.89}$$

since $\mathcal{L}$ serves as intermediate angular momentum in the $\mathcal{J}$-coupling,

$$|[(l_{cm}L_3)\mathcal{L}S_3]\mathcal{J}\rangle = \sum_J (-1)^{L_3+S_3+l_{cm}+\mathcal{J}} \hat{\mathcal{L}}\hat{J} \left\{ \begin{smallmatrix} l_{cm} & L_3 & \mathcal{L} \\ S_3 & \mathcal{J} & J \end{smallmatrix} \right\} |[l_{cm}(L_3S_3)J]\mathcal{J}\rangle, \tag{9.90}$$



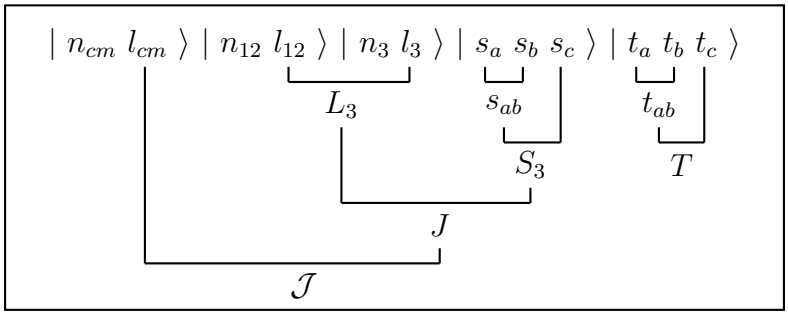
The transposition

$$|n_3l_3\rangle\ |n_{12}l_{12}\rangle \quad \rightarrow \quad |n_{12}l_{12}\rangle\ |n_3l_3\rangle \tag{9.91}$$

introduces the phase

$$(-1)^{l_{12}+l_3-L_3} \tag{9.92}$$

arising from the interchange of two columns in the L_3-coupling Clebsch-Gordan coefficient.

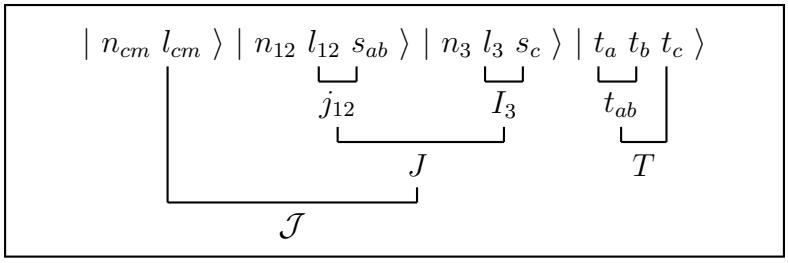


The final step is to change the J-coupling order, introducing the new intermediate angular momenta j_{12} and I_3,

$$|[(l_{12}l_3)L_3(s_{ab}\tfrac{1}{2})S_3]J\rangle \;\;\rightarrow\;\; |[(l_{12}s_{ab})j_{12}(l_3\tfrac{1}{2})I_3]J\rangle. \tag{9.93}$$

Since this is a rearrangement of the coupling of four angular momenta, one can again employ $9j$-symbols,

$$|[(l_{12}l_3)L_3(s_{ab}\tfrac{1}{2})S_3]J\rangle = \sum_{j_{12}I_3} \hat{j}_{12}\hat{I}_3\hat{L}_3\hat{S}_3 \begin{Bmatrix} l_{12} & l_3 & L_3 \\ s_{ab} & s_c & S_3 \\ j_{12} & I_3 & J \end{Bmatrix} |[(l_{12}s_{ab})j_{12}(l_3\tfrac{1}{2})I_3]J\rangle, \tag{9.94}$$



From the above considerations, the transformation coefficients for

$$\{\{|a\rangle|b\rangle\}^{J_{12}}|c\rangle\}^{\mathcal{J}} \;\;\rightarrow\;\; \{|n_{cm}l_{cm}\rangle|\alpha\rangle\}^{\mathcal{J}} \tag{9.95}$$

are obtained by patching all the transformations above together:

$$
\begin{aligned}
&\{\{|a\rangle|b\rangle\}^{J_{12}}|c\rangle\}^{\mathcal{J}} = \\
&\sum_{t_{ab}}\sum_{TM_T}\sum_{L_{12}s_{ab}}\sum_{\substack{\mathcal{N}_{12}\mathcal{L}_{12}\\ n_{12}l_{12}}}\sum_{\mathcal{L}S_3}\sum_{\Lambda}\sum_{\substack{n_{cm}l_{cm}\\ n_3l_3}}\sum_{L_3}\sum_{J}\sum_{j_{12}I_3} \\
&\times\ \delta_{2\mathcal{N}_{12}+\mathcal{L}_{12}+2n_{12}+l_{12},2n_a+l_a+2n_b+l_b} \\
&\times\ \delta_{2\mathcal{N}_{12}+\mathcal{L}_{12}+2n_c+l_c,2n_{cm}+l_{cm}+2n_3+l_3} \\
&\times\ \langle[\mathcal{N}_{12}\mathcal{L}_{12},n_{12}l_{12}]L_{12}|[n_al_a,n_bl_b]L_{12}\rangle_1 \\
&\times\ \langle[n_{cm}l_{cm},n_3l_3]\Lambda|[\mathcal{N}_{12}\mathcal{L}_{12},n_cl_c]\Lambda\rangle_2 \\
&\times\ \begin{pmatrix} t_a & t_b & t_{ab} \\ m_{t_a} & m_{t_b} & m_{t_a}+m_{t_b}\end{pmatrix}_{CG}\begin{pmatrix} t_{ab} & t_c & T \\ m_{t_a}+m_{t_b} & m_{t_c} & M_T\end{pmatrix}_{CG} \\
&\times\ \hat{j}_a\ \hat{j}_b\ \hat{L}_{12}\ \hat{s}_{12}\ \begin{Bmatrix} l_a & s_a & j_a \\ l_b & s_b & j_b \\ L_{12} & s_{ab} & J_{12}\end{Bmatrix} \\
&\times\ \hat{J}_{12}\ \hat{j}_c\ \hat{\mathcal{L}}\ \hat{S}_3\ \begin{Bmatrix} L_{12} & s_{ab} & J_{12} \\ l_c & s_c & j_c \\ \mathcal{L} & S_3 & \mathcal{J}\end{Bmatrix} \\
&\times\ (-1)^{l_{12}+l_c+L_{12}+\Lambda}\ \hat{L}_{12}\ \hat{\Lambda}\ \begin{Bmatrix} l_c & \mathcal{L}_{12} & \Lambda \\ l_{12} & \mathcal{L} & L_{12}\end{Bmatrix} \\
&\times\ (-1)^{l_{cm}+l_3+l_{12}+\mathcal{L}}\ \hat{\Lambda}\ \hat{L}_3\ \begin{Bmatrix} l_{cm} & l_3 & \Lambda \\ l_{12} & \mathcal{L} & L_3\end{Bmatrix} \\
&\times\ (-1)^{L_3+S_3+l_{cm}+\mathcal{J}}\ \hat{\mathcal{L}}\ \hat{J}\ \begin{Bmatrix} l_{cm} & L_3 & \mathcal{L} \\ S_3 & \mathcal{J} & J\end{Bmatrix} \\
&\times\ (-1)^{l_{12}+l_3-L_3}\ \hat{j}_{12}\ \hat{I}_3\ \hat{L}_3\ \hat{S}_3\ \begin{Bmatrix} l_{12} & l_3 & L_3 \\ s_{ab} & s_c & S_3 \\ j_{12} & I_3 & J\end{Bmatrix} \\
&\times\ \{|n_{cm}l_{cm}\rangle|\alpha\rangle\}^{\mathcal{J}}
\end{aligned}
\tag{9.96}
$$

Recalling the quantum numbers in $|\alpha\rangle$, the inner product of $\{|n'_{cm}l'_{cm}\rangle|\alpha'\rangle\}^{\mathcal{J}}$ and (9.96) will generate δ functions

$$
\begin{aligned}
&\delta_{n'_{cm}n_{cm}}\ \delta_{l'_{cm}l_{cm}}\ \delta_{n'_{12}n_{12}}\ \delta_{n'_3n_3}\ \delta_{l'_{12}l_{12}}\ \delta_{s'_{ab}s_{ab}}\ \delta_{j'_{12}j_{12}} \\
&\delta_{l'_3l_3}\ \delta_{I'_3I_3}\ \delta_{J'J}\ \delta_{t'_{ab}t_{ab}}\ \delta_{T'T}\ \delta_{M_{T'}M_T}
\end{aligned}
\tag{9.97}
$$

that will eliminate the corresponding summations in (9.96), leaving the sums

$$
\sum_{L_{12}}\ \sum_{\mathcal{N}_{12}\mathcal{L}_{12}}\ \sum_{\mathcal{L}}\ \sum_{S_3}\ \sum_{\Lambda}\ \sum_{L_3}. \tag{9.98}
$$

The primes can be dropped since every occurring quantum number is either be summed over or uniquely assigned to each of the states in the overlap.

The overlap (9.76) finally becomes

$$
\begin{aligned}
&\{\langle n_{cm} l_{cm}|\langle\alpha|\}^{\mathcal{J}}\{\{|a\rangle|b\rangle\}^{J_{12}}|c\rangle\}^{\mathcal{J}} = \\
&\sum_{L_{12}} \sum_{\mathcal{N}_{12}\mathcal{L}_{12}} \sum_{\mathcal{L}} \sum_{S_3} \sum_{\Lambda} \sum_{L_3} \\
&\times \quad \delta_{2\mathcal{N}_{12}+\mathcal{L}_{12}+2n_{12}+l_{12},2n_a+l_a+2n_b+l_b} \\
&\times \quad \delta_{2\mathcal{N}_{12}+\mathcal{L}_{12}+2n_c+l_c,2n_{cm}+l_{cm}+2n_3+l_3} \\
&\times \quad \langle[\mathcal{N}_{12}\mathcal{L}_{12}, n_{12}l_{12}]L_{12}|[n_a l_a, n_b l_b]L_{12}\rangle_1 \\
&\times \quad \langle[n_{cm} l_{cm}, n_3 l_3]\Lambda|[\mathcal{N}_{12}\mathcal{L}_{12}, n_c l_c]\Lambda\rangle_2 \\
&\times \quad \begin{pmatrix} t_a & t_b & t_{ab} \\ m_{t_a} & m_{t_b} & m_{t_a}+m_{t_b} \end{pmatrix}_{CG} \begin{pmatrix} t_{ab} & t_c & T \\ m_{t_a}+m_{t_b} & m_{t_c} & m_{t_a}+m_{t_b}+m_{t_c} \end{pmatrix}_{CG} \\
&\times \quad \hat{j}_a\ \hat{j}_b\ \hat{L}_{12}\ \hat{s}_{12} \begin{Bmatrix} l_a & s_a & j_a \\ l_b & s_b & j_b \\ L_{12} & s_{ab} & J_{12} \end{Bmatrix} \\
&\times \quad \hat{J}_{12}\ \hat{j}_c\ \hat{\mathcal{L}}\ \hat{S}_3 \begin{Bmatrix} L_{12} & s_{ab} & J_{12} \\ l_c & s_c & j_c \\ \mathcal{L} & S_3 & \mathcal{J} \end{Bmatrix} \\
&\times \quad (-1)^{l_{12}+l_c+L_{12}+\Lambda}\ \hat{L}_{12}\ \hat{\Lambda} \begin{Bmatrix} l_c & \mathcal{L}_{12} & \Lambda \\ l_{12} & \mathcal{L} & L_{12} \end{Bmatrix} \\
&\times \quad (-1)^{l_{cm}+l_3+l_{12}+\mathcal{L}}\ \hat{\Lambda}\ \hat{L}_3 \begin{Bmatrix} l_{cm} & l_3 & \Lambda \\ l_{12} & \mathcal{L} & L_3 \end{Bmatrix} \\
&\times \quad (-1)^{L_3+S_3+l_{cm}+\mathcal{J}}\ \hat{\mathcal{L}}\ \hat{J} \begin{Bmatrix} l_{cm} & L_3 & \mathcal{L} \\ S_3 & \mathcal{J} & J \end{Bmatrix} \\
&\times \quad (-1)^{l_{12}+l_3-L_3}\ \hat{j}_{12}\ \hat{I}_3\ \hat{L}_3\ \hat{S}_3 \begin{Bmatrix} l_{12} & l_3 & L_3 \\ s_{ab} & s_c & S_3 \\ j_{12} & I_3 & J \end{Bmatrix}. \qquad (9.99)
\end{aligned}
$$

The two δ functions can be used to eliminate the $\mathcal{N}_{12}$ and l_{cm} summation, for instance, by constraining them to be

$$
\begin{aligned}
l_{cm} &= 2n_a + l_a + 2n_b + l_b + 2n_c + l_c - 2n_{cm} - 2n_{12} - l_{12} - 2n_3 - l_3 \\
\mathcal{N}_{12} &= \tfrac{1}{2}\Big[2n_a + l_a + 2n_b + l_b - \mathcal{L}_{12} - 2n_{12} - l_{12}\Big]. \qquad (9.100)
\end{aligned}
$$

For computational purposes, the most efficient implementation of $\tilde{T}$ is of the form

$$
\begin{aligned}
\tilde{T}\left[\begin{array}{ccc} a & b & c \\ n_{cm} & l_{cm} & \alpha_k^{(NJT)} \\ \mathcal{J} & & \end{array}\right] &= (-1)^{l_c+l_{12}}\ \hat{j}_a\ \hat{j}_b\ \hat{j}_c\ \hat{j}_{12}\ \hat{s}_{12}\ \hat{I}_3\ \hat{\mathcal{J}} \\
&\times\ \left(\begin{array}{ccc} t_a & t_b & t_{ab} \\ m_{t_a} & m_{t_b} & m_{t_a}+m_{t_b} \end{array}\right)_{CG} \left(\begin{array}{ccc} t_{ab} & t_c & T \\ m_{t_a}+m_{t_b} & m_{t_c} & m_{t_a}+m_{t_b}+m_{t_c} \end{array}\right)_{CG} \\
&\times\ \sum_{J_{12}}\ \hat{J}_{12}\ \left(\begin{array}{ccc} j_a & j_b & J_{12} \\ m_a & m_b & m_a+m_b \end{array}\right)_{CG} \left(\begin{array}{ccc} J_{12} & j_c & \mathcal{J} \\ m_a+m_b & m_c & m_a+m_b+m_c \end{array}\right)_{CG} \\
&\times\ \sum_{L_{12}}\ (-1)^{L_{12}}\ (2L_{12}+1)\ \left\{\begin{array}{ccc} l_a & s_a & j_a \\ l_b & s_b & j_b \\ L_{12} & s_{ab} & J_{12} \end{array}\right\} \\
&\times\ \sum_{\mathcal{L}_{12}}\ \langle[\mathcal{N}_{12}\mathcal{L}_{12}, n_{12}l_{12}]L_{12}|[n_a l_a, n_b l_b]L_{12}\rangle_1 \\
&\times\ \sum_{\Lambda}\ (-1)^{\Lambda}\ (2\Lambda+1)\ \langle[n_{cm}l_{cm}, n_3 l_3]\Lambda|[\mathcal{N}_{12}\mathcal{L}_{12}, n_c l_c]\Lambda\rangle_2 \\
&\times\ \sum_{\mathcal{L}}\ (-1)^{\mathcal{L}}\ (2\mathcal{L}+1)\ \left\{\begin{array}{ccc} l_c & \mathcal{L}_{12} & \Lambda \\ l_{12} & \mathcal{L} & L_{12} \end{array}\right\} \\
&\times\ \sum_{S_3}\ (-1)^{\mathcal{J}+S_3}\ (2S_3+1)\ \left\{\begin{array}{ccc} L_{12} & s_{ab} & J_{12} \\ l_c & s_c & j_c \\ \mathcal{L} & S_3 & \mathcal{J} \end{array}\right\} \\
&\times\ \sum_{L_3}\ (2L_3+1)\ \left\{\begin{array}{ccc} l_{cm} & L_3 & \mathcal{L} \\ S_3 & \mathcal{J} & J \end{array}\right\} \left\{\begin{array}{ccc} l_{12} & l_3 & L_3 \\ s_{ab} & s_c & S_3 \\ j_{12} & I_3 & J \end{array}\right\} \left\{\begin{array}{ccc} l_{cm} & l_3 & \Lambda \\ l_{12} & \mathcal{L} & L_3 \end{array}\right\}. \qquad (9.101)
\end{aligned}
$$

9.3 Slater-Condon Rules

Working with a Slater determinant basis of the model space

$$\{|\Phi\rangle = \hat{\chi}\ |\phi_{\alpha_1}^{(1)}\ \ldots\ \phi_{\alpha_N}^{(N)}\rangle \equiv |\alpha_1\ \ldots\ \alpha_N\rangle\}, \qquad (9.102)$$

the calculation of expectation values of some operator requires the evaluation of its matrix elements in this basis. The Slater-Condon rules facilitate the calculation of matrix elements of M-body operators

$$\hat{O}_M = \sum_{p_1<\cdots<p_M=1}^{N} \hat{o}_M(p_1, \ldots, p_M) \qquad (9.103)$$

between such N-dimensional Slater determinants by reducing this expression into a sum of matrix elements of $\hat{o}_M$ between M-dimensional Slater determinants, symbolically written as

$$\langle\Phi'|\ \hat{O}_M\ |\Phi\rangle \propto \sum\ \langle\Phi'_{(M)}|\ \hat{O}_M\ |\Phi_{(M)}\rangle. \qquad (9.104)$$

Slater-Condon rules for one- and two-body operators (Table 9.1)

$$\begin{aligned} \hat{F} &= \sum_{p=1}^{N} \hat{f}(p) \\ \hat{G} &= \sum_{p_1<p_2=1}^{N} \hat{g}(p_1,p_2) \end{aligned} \tag{9.105}$$

and their derivation are commonly found in textbooks on nuclear physics or quantum chemistry [47], [48]. For one- and two-body operators the Slater-Condon rules are most easily proven directly, i.e. by writing all relevant cases and terms explicitly down and working with these. For three-body operators however, the number of relevant terms is already too large for a reasonable explicit treatment.

The way the one- and two-body Slater-Condon rules are derived in [49], although also essentially explicit, allows a generalization to the M-body case by replacing the explicit steps by their generalized counterparts where one can take advantage of determinants. Since along these lines in [49], also the case of non-orthogonal single-particle basis states is considered, these rules will in the following also be generalized to the M-body case. In fact, most of the effort will be about the non-orthogonal case from which the special case of orthogonal bases will then be derived. From this one gets a quite deep insight on the structure of matrix elements between Slater determinants. If one is solely interested in obtaining the practical rules for the orthogonal case only, there is surely a faster derivation.

Due to the fact that there are only non-vanishing matrix elements

$$\langle \Phi' |\, \hat{O}_M \,| \Phi \rangle \tag{9.106}$$

of a M-body operator betweens Slater determinants that differ in at most M orbitals, in calculating matrix elements one distinguished between the *no non-coincidence*

$$\langle \Box |\, \hat{O}_M \,| \Box \rangle, \tag{9.107}$$

single non-conincidence

$$\langle i' \Box |\, \hat{O}_M \,| i \Box \rangle, \tag{9.108}$$

double non-coincidence

$$\langle i'j'\Box|\ \hat{O}_M\ |ij\Box\rangle \tag{9.109}$$

case and so on, where $\Box$ is used to abbreviate the orbital labels that occur in both determinants. It is assumed that the determinants are in maximum coincidence order, i.e. the creation operators assigned to the $\Box$ orbitals have same positions in the creation operator strings generating both determinants. Using this notation the common Slater-Condon rules for one- and two-body operators read [47]:

Slater-Condon Rules for one-body operators			Slater-Condon Rules for two-body operators		
$\langle\Box\|\ \hat{F}\ \|\Box\rangle$	$=$	$\sum_{k\in\Box}\langle k\|\ \hat{f}\ \|k\rangle$	$\langle\Box\|\ \hat{G}\ \|\Box\rangle$	$=$	$\sum_{\substack{k<l\\k,l\in\Box}}\langle kl\|\ \hat{g}\ \|kl\rangle$
$\langle i'\Box\|\ \hat{F}\ \|i\Box\rangle$	$=$	$\langle i'\|\ \hat{f}\ \|i\rangle$	$\langle i'\Box\|\ \hat{G}\ \|i\Box\rangle$	$=$	$\sum_{k\in\Box}\langle i'k\|\ \hat{g}\ \|ik\rangle$
$\langle i'j'\Box\|\ \hat{F}\ \|ij\Box\rangle$	$=$	0	$\langle i'j'\Box\|\ \hat{G}\ \|ij\Box\rangle$	$=$	$\langle i'j'\|\ \hat{g}\ \|ij\rangle$

Table 9.1: *Slater-Condon rules for one- and two-body operators and orthogonal single-particle basis states.*

9.3.1 Preparations

In order to be able to generalize the Slater-Condon to higher-body operators it is necessary to review some properties of determinantal structures.

Definition 9.3.1 *The determinant of a matrix of order N is defined in terms of the permutation group S_N by the well-known formula*

$$\det A = \sum_{\pi\in S_N} \mathrm{sgn}(\pi)\cdot\prod_{i=1}^{N} A_{i,\pi(i)} \tag{9.110}$$

where $\mathrm{sgn}(\pi)=\pm 1$ *denotes the parity or signature of the permutation* π.

It is worthwhile to take a closer look at the signature function. In (9.110) the signature of a given π

$$\mathrm{sgn}(\pi) = \mathrm{sgn}\left(\begin{smallmatrix} 1 & 2 & \dots & N \\ \pi(1) & \pi(2) & \dots & \pi(N) \end{smallmatrix}\right) \tag{9.111}$$

is given by the number of transpositions needed to bring the elements of both rows into the same order. Clearly the signature depends on the reference configuration (first row) of the elements that are permuted which can be chosen arbitrarily. It is, however, possible to relate signature functions with different reference configurations by the following definition [51]:

Definition 9.3.2 *Let $i_1, \dots, i_k$ and $j_1, \dots, j_k$ be permutations of distinct N numbers. Let σ and τ be elements of S_N. Then the sign*

$$\mathrm{sgn}\begin{pmatrix} \sigma(i_1) & \dots & \sigma(i_N) \\ \tau(j_1) & \dots & \tau(j_N) \end{pmatrix} \tag{9.112}$$

is defined by

$$\mathrm{sgn}\begin{pmatrix} \sigma(i_1) & \dots & \sigma(i_N) \\ \tau(j_1) & \dots & \tau(j_N) \end{pmatrix} = \mathrm{sgn}(\sigma)\ \mathrm{sgn}(\tau)\ \mathrm{sgn}\left(\begin{smallmatrix} i_1 & \dots & i_N \\ j_1 & \dots & j_N \end{smallmatrix}\right)$$

where

$$\mathrm{sgn}(\sigma) = \mathrm{sgn}\begin{pmatrix} i_1 & \dots & i_N \\ \sigma(i_1) & \dots & \sigma(i_N) \end{pmatrix}$$

and analogous for τ.

As pointed out in [49], there is a connection between the Slater-Condon rules and the generalized Laplace expansion formula for determinants. Define R as the row set and C as the column set of a matrix A. Let $I = \{i_1, \dots, i_k\} \subset R$, $J = \{j_1, \dots, j_k\} \subset C$, then $A[R\backslash\{I\}, C\backslash\{J\}]$ is the submatrix of A where the rows I and columns J are removed. The determinant of this submatrix is called the complementary minor of A. A common term related to determinant expansions is the algebraic complement [50].

Definition 9.3.3 *The algebraic complement $A_{M}{}^{i_1,\dots,i_k}_{j_1,\dots,j_k}$ of a matrix M is defined as*

$$A_{M}{}^{i_1,\dots,i_k}_{j_1,\dots,j_k} = (-1)^{\sum_{z=1}^{k}(i_z+j_z)}\ \det\ M[R\backslash\{I\}, C\backslash\{J\}]. \tag{9.113}$$

In other words, the algebraic complement is nothing but the complementary minor multiplied with a sign which depends on which rows and columns have been removed from the original matrix.

It will be necessary to express complementary minors in terms of S_N and the full matrix A. Since the result is not too obvious and no derivation could be found in the literature, this expression will now be derived for $\dim I = \dim J = 1$ and for the general case afterwards.

Proposition 9.3.1 *The complementary minor* $\det A[R\backslash\{i\}, C\backslash\{j\}]$ *can be expressed in terms of the permutation group* S_N *as follows*

$$\det A[R\backslash\{i\}, C\backslash\{j\}] = \sum_{\substack{\pi\in S_N \\ \pi(i)=j}} (-1)^{i+j}\ \mathrm{sgn}(\pi)\ \cdot \prod_{\substack{m=1 \\ m\neq i}}^{N} A_{m,\pi(m)} \tag{9.114}$$

and the general case $\det A[R\backslash\{i_1,\ldots,i_k\}, C\backslash\{j_1,\ldots,j_k\}]$ *reads*

$$\det A[R\backslash\{i_1,\ldots,i_k\}, C\backslash\{j_1,\ldots,j_k\}] = \sum_{\substack{\pi\in S_N \\ \pi(i_1)=j_1 \\ \vdots \\ \pi(i_k)=j_k}} (-1)^{\sum_{z=1}^{k}(i_z+j_z)}\ \mathrm{sgn}(\pi)\ \cdot \prod_{\substack{m=1 \\ m\neq i_1 \\ \vdots \\ m\neq i_k}}^{N} A_{m,\pi(m)}. \tag{9.115}$$

Proof 9.3.1

(i) Since $A[R\backslash\{i\}, C\backslash\{j\}]$ *is a* $N-1\times N-1$ *square matrix, Eq. (9.110) can be used*

$$\det A[R\backslash\{i\}, C\backslash\{j\}] = \sum_{\pi'\in S_{N-1}} \mathrm{sgn}(\pi')\ \cdot \prod_{m=1}^{N-1} A[R\backslash\{i\}, C\backslash\{j\}]_{m,\pi'(m)}. \tag{9.116}$$

π' permutes the numbers $\{1,2,\ldots,N-1\}$ *and the signature is according to*

$$\mathrm{sgn}(\pi') = \mathrm{sgn}\begin{pmatrix} 1 & 2 & \ldots & N-1 \\ \pi'(1) & \pi'(2) & \ldots & \pi'(N-1) \end{pmatrix}.$$

Equation (9.116) can be written in terms of A and $\pi \in S_N$ as

$$\det A[R\backslash\{i\}, C\backslash\{j\}] = \sum_{\substack{\pi\in S_N \\ \pi(i)=j}} \phi(i,j)\ \mathrm{sgn}(\pi)\ \cdot \prod_{\substack{m=1 \\ m\neq i}}^{N} A_{m,\pi(m)} \quad (9.117)$$

which can be seen as follows: In (9.116) the product index m runs over all $N-1$ row values of the submatrix. For the product in (9.117) not to run over the excluded row i, the constraint $m \neq i$ is introduced. Furthermore, in (9.116) the column index $\pi'(m)$ trivially only belongs to columns of the submatrix. Therefore, in (9.117) no contribution may come from a term with $\pi(m) = j$. This is achieved by constraining the permutations π by $\pi(i) \equiv j$ since those contributions are then excluded by the product constraint. The permutations π now permute the full row index set of A and the signum is

$$\mathrm{sgn}(\pi) = \mathrm{sgn}\begin{pmatrix} 1 & 2 & \dots & i & \dots & j & \dots & N \\ \pi(1) & \pi(2) & \dots & \pi(i) & \dots & \pi(j) & \dots & \pi(N) \end{pmatrix}. \quad (9.118)$$

To ensure that the contributing terms in (9.117) have the same signs as in (9.116), a correction factor $\phi(i,j)$ has to be multiplied. Using Def. 9.3.2, this correction factor can be determined from the following consideration:

$$\begin{aligned} &\mathrm{sgn}(\pi) \quad (9.119)\\ &= \mathrm{sgn}\begin{pmatrix} 1 & 2 & \dots & i & \dots & j & \dots & N \\ \pi(1) & \pi(2) & \dots & \pi(i) & \dots & \pi(j) & \dots & \pi(N) \end{pmatrix} \\ &= \mathrm{sgn}(\pi)\ \mathrm{sgn}\begin{pmatrix} 1 & 2 & \dots & i & \dots & j & \dots & N \\ 1 & 2 & \dots & i & \dots & j & \dots & N \end{pmatrix} \\ &= \mathrm{sgn}(\pi)\ \mathrm{sgn}(i \to [1])\ \mathrm{sgn}(j \to [1])\ \mathrm{sgn}\begin{pmatrix} j & 1 & 2 & \dots & N \\ i & 1 & 2 & \dots & N \end{pmatrix} \\ &= \mathrm{sgn}(\pi)\ \mathrm{sgn}(j \to [1])\ \mathrm{sgn}(i \to [1])\ \mathrm{sgn}(\pi)\ \mathrm{sgn}\begin{pmatrix} j & 1 & 2 & \dots & N \\ \pi(i) & \pi(1) & \pi(2) & \dots & \pi(N) \end{pmatrix}. \end{aligned}$$

The signature $\mathrm{sgn}(j \to [1])$ is the signature of the permutation that brings j to the first position in the first row and $\mathrm{sgn}(i \to [1])$ is analogous for the second row. Since in the third line the numbers are ordered, these signatures are just $(-1)^{j-1}$ and $(-1)^{i-1}$. It is $(\mathrm{sgn}(\pi))^2 = 1$, and because of $\pi(i) = j$ the first column in the last signature is automatically in order. So, the last signature is only determined by the permutation of columns $2, \dots, N-1$, which is just the signature of π' again. Therefore,

$$\mathrm{sgn}(\pi) = (-1)^{i+j}\ \mathrm{sgn}(\pi') \quad (9.120)$$

and the correction factor $\phi(i,j)$ is determined to be

$$\phi(i,j) = (-1)^{i+j}. \tag{9.121}$$

(ii) Analog considerations lead for the general case to

$$\begin{aligned} &\det A[R\backslash\{i_1,\ldots,i_k\}, C\backslash\{j_1,\ldots,j_k\}] \\ &= \sum_{\substack{\pi\in S_N \\ \pi(i_1)=j_1 \\ \vdots \\ \pi(i_k)=j_k}} \phi(i_1,\ldots,i_k,j_1,\ldots,j_k)\ \mathrm{sgn}(\pi)\cdot \prod_{\substack{m=1 \\ m\neq i_1 \\ \vdots \\ m\neq i_k}}^{N} A_{m,\pi(m)}. \end{aligned} \tag{9.122}$$

The correction factor is determined in the same way as before:

$$\begin{aligned} \mathrm{sgn}(\pi) &= \mathrm{sgn}\begin{pmatrix} 1 & \ldots & i_1 & \ldots & i_k & \ldots & N \\ \pi(1) & \ldots & \pi(i_1) & \ldots & \pi(i_k) & \ldots & \pi(N) \end{pmatrix} \\ &= \mathrm{sgn}(\pi)\ \mathrm{sgn}\begin{pmatrix} 1 & \ldots & i_1 & \ldots & i_k & \ldots & N \\ 1 & \ldots & i_1 & \ldots & i_k & \ldots & N \end{pmatrix} \\ &= \mathrm{sgn}(\pi)\ \mathrm{sgn}(j_1\to 1)\ \ldots\ \mathrm{sgn}(j_k\to k) \\ &\quad \times \mathrm{sgn}(i_1\to 1)\ \ldots\ \mathrm{sgn}(i_k\to k)\ \mathrm{sgn}\begin{pmatrix} j_1 & \ldots & j_k & 1 & \ldots & N \\ i_1 & \ldots & i_k & 1 & \ldots & N \end{pmatrix} \\ &= (-1)^{\sum_{z=1}^{k}(i_z+j_z)}\ \mathrm{sgn}(\pi'), \end{aligned} \tag{9.123}$$

where in the last line the ordering $i_1 < \ldots i_k$, $j_1 < \ldots j_k$ allowed the identification $\mathrm{sgn}(i_n \to n) = (-1)^{i_n - n}$ and analogous for j_n. □

9.3.2 General Matrix Element Formula

In this section an expression for the matrix element

$$\langle\Phi'|\ \hat{A}\ |\Phi\rangle \tag{9.124}$$

of the three-body operator

$$\hat{A} = \sum_{\substack{p,q,r=1 \\ p<q<r}}^{N} \hat{a}(p,q,r) \tag{9.125}$$

between the Slater determinants

$$|\Phi'\rangle = \hat{\chi}\ |\phi_{\alpha_1}^{(1)}\ldots\phi_{\alpha_N}^{(N)}\rangle \equiv |\alpha_1\ \ldots\alpha_N\rangle \tag{9.126}$$

and

$$|\Phi\rangle = \hat{\chi}\ |\phi^{(1)}_{\beta_1} \dots \phi^{(N)}_{\beta_N}\rangle \equiv |\beta_1\ \dots \beta_N\rangle \tag{9.127}$$

is derived in terms of algebraic complements and matrix elements between M-dimensional Slater determinants. This expression will be the starting point for the derivation of the Slater rules for the different non-coincidence cases and single-particle bases.

The antisymmetrizer in (9.126) and (9.127) and its hermitecity and idempotency can be expressed as

$$\begin{aligned} \hat{\chi} &= \frac{1}{\sqrt{N!}} \sum_{\pi \in S_N} \mathrm{sgn}(\pi)\ \hat{\pi} \\ \hat{\chi}^\dagger &= \hat{\chi}, \qquad\qquad \hat{\chi}^2 = \sqrt{N!}\ \hat{\chi}. \end{aligned} \tag{9.128}$$

It should be noted that the action of the permutation operator $\hat{\pi}$ on a product wave function is according to

$$\hat{\pi}\ |\phi_{\beta_1}\ \dots\ \phi_{\beta_N}\rangle\ =\ |\phi_{\beta_{\pi(1)}}\ \dots\ \phi_{\beta_{\pi(N)}}\rangle. \tag{9.129}$$

Furthermore, it shall be remarked that it is well known that the overlap of two Slater determinants again has the structure of a determinant - the determinant of the overlap matrix $S(\Phi', \Phi)$,

$$S(\Phi', \Phi)_{ij} = \langle \phi_{\alpha_i} | \phi_{\beta_j} \rangle. \tag{9.130}$$

With this and using that $[\hat{a}, \hat{\chi}] = 0$, one has

$$\begin{aligned} \langle \Phi' |\ \hat{A}\ |\Phi\rangle &= \sum_{\substack{p,q,r=1 \\ p<q<r}}^{N} \langle \phi^{(1)}_{\alpha_1} \dots \phi^{(N)}_{\alpha_N} |\ \hat{\chi}^\dagger\ \hat{a}(p,q,r)\ \hat{\chi}\ |\phi^{(1)}_{\beta_1} \dots \phi^{(N)}_{\beta_N}\rangle \\ &= \sqrt{N!} \sum_{\substack{p,q,r=1 \\ p<q<r}}^{N} \langle \phi^{(1)}_{\alpha_1} \dots \phi^{(N)}_{\alpha_N} |\ \hat{a}(p,q,r)\ \hat{\chi}\ |\phi^{(1)}_{\beta_1} \dots \phi^{(N)}_{\beta_N}\rangle \\ &= \sum_{\substack{p,q,r=1 \\ p<q<r}}^{N} \sum_{\pi \in S_N} \mathrm{sgn}(\pi)\ \langle \phi^{(1)}_{\alpha_1} \dots \phi^{(N)}_{\alpha_N} |\ \hat{a}(p,q,r)\ |\phi^{(1)}_{\beta_{\pi(1)}} \dots \phi^{(N)}_{\beta_{\pi(N)}}\rangle. \end{aligned} \tag{9.131}$$

The operator $\hat{a}(p,q,r)$ acts in the Hilbert spaces of particles p,q and r only, and therefore

$$\begin{aligned}\langle\Phi'|\ \hat{A}\ |\Phi\rangle &= \sum_{\substack{p,q,r=1\\ p<q<r}}^{N}\ \sum_{\pi\in S_N} \operatorname{sgn}(\pi)\ \langle\phi^{(p)}_{\alpha_p}\ \phi^{(q)}_{\alpha_q}\ \phi^{(r)}_{\alpha_r}|\ \hat{a}(p,q,r)\ |\phi^{(p)}_{\beta_{\pi(p)}}\phi^{(q)}_{\beta_{\pi(q)}}\phi^{(r)}_{\beta_{\pi(r)}}\rangle \\ &\quad\times \prod_{\substack{m=1\\ m\neq p,q,r}}^{N} \langle\phi^{(m)}_{\alpha_m}|\phi^{(m)}_{\beta_{\pi(m)}}\rangle. \end{aligned} \tag{9.132}$$

For an one-particle operator $\hat{h}(p)$ the obvious identity

$$\sum_{p=1}^{N}\ \langle\phi^{(p)}_{\alpha_p}|\ \hat{h}(p)\ |\phi^{(p)}_{\beta_{\pi(p)}}\rangle = \sum_{p=1}^{N}\sum_{a=1}^{N}\ \delta_{a,\pi(p)}\ \langle\phi^{(p)}_{\alpha_p}|\ \hat{h}(p)\ |\phi^{(p)}_{\beta_a}\rangle \tag{9.133}$$

holds and an analogous identity for the case of higher-body operators is found straightforwardly. Using this, (9.132) becomes

$$\begin{aligned}\langle\Phi'|\ \hat{A}\ |\Phi\rangle = &\sum_{\substack{p,q,r=1\\ p<q<r}}^{N}\ \sum_{\substack{a,b,c=1\\ a\neq b\neq c}}^{N}\ \sum_{\pi\in S_N} \delta_{a,\pi(p)}\ \delta_{b,\pi(q)}\ \delta_{c,\pi(r)}\ \operatorname{sgn}(\pi) \\ &\times \langle\phi^{(p)}_{\alpha_p}\ \phi^{(q)}_{\alpha_q}\ \phi^{(r)}_{\alpha_r}|\ \hat{a}(p,q,r)\ |\phi^{(p)}_{\beta_a}\ \phi^{(q)}_{\beta_b}\ \phi^{(r)}_{\beta_c}\rangle \\ &\times \prod_{\substack{m=1\\ m\neq p,q,r}}^{N} \langle\phi^{(m)}_{\alpha_m}|\phi^{(m)}_{\beta_{\pi(m)}}\rangle \end{aligned} \tag{9.134}$$

($a\neq b\neq c$ because of the single occupancy of orbitals in Slater determinants). The delta functions can be used to constrain the permutations π such that

$$\sum_{\pi\in S_N}\ \delta_{a,\pi(p)}\ \delta_{b,\pi(q)}\ \delta_{c,\pi(r)}\cdots = \sum_{\substack{\pi\in S_N\\ \pi(p)=a\\ \pi(q)=b\\ \pi(r)=c}}\ \ldots \tag{9.135}$$

and under the product one recognizes the matrix elements of the overlap matrix. After inserting an identity of the form $1=((-1)^{a+b+c+p+q+r})^2$ one

arrives at

$$\begin{aligned}\langle\Phi'|\ \hat{A}\ |\Phi\rangle = & \sum_{\substack{p,q,r=1\\ p<q<r}}^{N}\ \sum_{\substack{a,b,c=1\\ a\neq b\neq c}}^{N}\ \langle\phi_{\alpha_p}^{(p)}\ \phi_{\alpha_q}^{(q)}\ \phi_{\alpha_r}^{(r)}|\ \hat{a}(p,q,r)\ |\phi_{\beta_a}^{(p)}\ \phi_{\beta_b}^{(q)}\ \phi_{\beta_c}^{(r)}\rangle \\ & \times\ (-1)^{a+b+c+p+q+r} \\ & \times\ \sum_{\substack{\pi\in S_N\\ \pi(p)=a\\ \pi(q)=b\\ \pi(r)=c}}\ (-1)^{a+b+c+p+q+r}\ \mathrm{sgn}(\pi)\ \cdot\ \prod_{\substack{m=1\\ m\neq p,q,r}}^{N}\ S(\Phi',\Phi)_{m,\pi(m)}\end{aligned} \tag{9.136}$$

where Proposition 9.3.1 can be used to write the last line as a complementary minor

$$\begin{aligned}\langle\Phi'|\ \hat{A}\ |\Phi\rangle = & \sum_{\substack{p,q,r=1\\ p<q<r}}^{N}\ \sum_{\substack{a,b,c=1\\ a\neq b\neq c}}^{N}\ \langle\phi_{\alpha_p}^{(p)}\ \phi_{\alpha_q}^{(q)}\ \phi_{\alpha_r}^{(r)}|\ \hat{a}(p,q,r)\ |\phi_{\beta_a}^{(p)}\ \phi_{\beta_b}^{(q)}\ \phi_{\beta_c}^{(r)}\rangle \\ & \times\ (-1)^{a+b+c+p+q+r}\ \det\ S(\Phi',\Phi)[R\backslash\{p,q,r\},C\backslash\{a,b,c\}].\end{aligned} \tag{9.137}$$

Together with the remaining sign, the complementary minor becomes the corresponding algebraic complement of the overlap matrix as defined in Def. 9.3.3,

$$\langle\Phi'|\ \hat{A}\ |\Phi\rangle = \sum_{\substack{p,q,r=1\\ p<q<r}}^{N}\ \sum_{\substack{a,b,c=1\\ a\neq b\neq c}}^{N}\ \langle\phi_{\alpha_p}^{(p)}\phi_{\alpha_q}^{(q)}\phi_{\alpha_r}^{(r)}|\hat{a}(p,q,r)|\phi_{\beta_a}^{(p)}\phi_{\beta_b}^{(q)}\phi_{\beta_c}^{(r)}\rangle A_{S}^{pqr}{}_{abc}. \tag{9.138}$$

To proceed further, the *abc* sum is rewritten according to the identity

$$\sum_{\substack{a,b,c=1\\ a\neq b\neq c}}^{N} f[a,b,c] = \sum_{\substack{a,b,c=1\\ a<b<c}}^{N}\ \sum_{\sigma\in S_3} f[\sigma(a),\sigma(b),\sigma(c)], \tag{9.139}$$

which is stated here without proof. With this, Eq. (9.138) becomes

$$\begin{aligned}\langle\Phi'|\ \hat{A}\ |\Phi\rangle\ \ = & \sum_{\substack{p,q,r=1\\ p<q<r}}^{N}\ \sum_{\substack{a,b,c=1\\ a<b<c}}^{N}\ \sum_{\sigma\in S_3}\ \langle\phi_{\alpha_p}^{(p)}\ \phi_{\alpha_q}^{(q)}\ \phi_{\alpha_r}^{(r)}|\ \hat{a}(p,q,r)\ |\phi_{\beta_{\sigma(a)}}^{(p)}\phi_{\beta_{\sigma(b)}}^{(q)}\phi_{\beta_{\sigma(c)}}^{(r)}\rangle \\ & \times\ A_{S}^{pqr}{}_{\sigma(a)\sigma(b)\sigma(c)}.\end{aligned} \tag{9.140}$$

Transpositions of columns of a determinant produce signs and as a consequence one can convince oneself that

$$A_{S^{pqr}_{\sigma(a)\sigma(b)\sigma(c)}} = \mathrm{sgn}(\sigma)\ A_{S^{pqr}_{abc}}. \tag{9.141}$$

Eq. (9.140) then can be recasted in the form

$$\begin{aligned}\langle\Phi'|\ \hat{A}\ |\Phi\rangle \ &= \ \sum_{\substack{p,q,r=1\\ p<q<r}}^{N}\ \sum_{\substack{a,b,c=1\\ a<b<c}}^{N}\ \langle\phi^{(p)}_{\alpha_p}\ \phi^{(q)}_{\alpha_q}\ \phi^{(r)}_{\alpha_r}|\ \hat{a}(p,q,r) \\ &\times\ \left[\sum_{\sigma\in S_3}\ \mathrm{sgn}(\sigma)\ |\phi^{(p)}_{\beta_{\sigma(a)}}\ \phi^{(q)}_{\beta_{\sigma(b)}}\ \phi^{(r)}_{\beta_{\sigma(c)}}\rangle\right]\ A_{S^{pqr}_{abc}}\end{aligned} \tag{9.142}$$

where one can insert the definition of the antisymmetrizer

$$\langle\Phi'|\ \hat{A}\ |\Phi\rangle \ = \ \sqrt{3!}\ \sum_{\substack{p,q,r=1\\ p<q<r}}^{N}\ \sum_{\substack{a,b,c=1\\ a<b<c}}^{N}\ \langle\phi^{(p)}_{\alpha_p}\ \phi^{(q)}_{\alpha_q}\ \phi^{(r)}_{\alpha_r}|\ \hat{a}(p,q,r)\ \hat{\chi}|\phi^{(p)}_{\beta_a}\ \phi^{(q)}_{\beta_b}\ \phi^{(r)}_{\beta_c}\rangle\ A_{S^{pqr}_{abc}}. \tag{9.143}$$

The factor $\sqrt{3!}$ can be canceled by antisymmetrizing also the left state, again by exploiting $\hat{\chi}^2 = \sqrt{3!}\hat{\chi}$ and $[\hat{a},\hat{\chi}] = 0$, arriving at

$$\langle\Phi'|\ \hat{A}\ |\Phi\rangle \ = \ \sum_{\substack{p,q,r=1\\ p<q<r}}^{N}\ \sum_{\substack{a,b,c=1\\ a<b<c}}^{N}\ A_{S^{pqr}_{abc}}\ \langle\alpha_p\ \alpha_q\ \alpha_r|\ \hat{a}\ |\beta_a\ \beta_b\ \beta_c\rangle, \tag{9.144}$$

where the occupation number representation of the Slater determinants will be used from now on and the particle coordinates on the operator have been dropped.

There has been nothing special about the three-body case so far and it is clear that the above considerations hold for arbitrary M-body operators $\hat{O}_M$ which allows to formulate the general simple matrix element formula

$$\langle\Phi'|\ \hat{O}_M\ |\Phi\rangle = \sum_{\substack{p_1,\ldots,p_M=1\\ p_1<\cdots<p_M}}^{N}\ \sum_{\substack{a_1,\ldots,a_m=1\\ a_1<\cdots<a_m}}^{N}\ A_{S^{p_1\ldots p_M}_{a_1\ldots a_M}}\ \langle\alpha_{p_1}\ldots\alpha_{p_M}|\ \hat{o}_M\ |\beta_{a_1}\ldots\beta_{a_M}\rangle. \tag{9.145}$$

9.3.3 Slater-Condon Rules for non-orthogonal Bases

Two Slater determinants $|\Phi'\rangle$ and $|\Phi\rangle$ are considered that are in maximum conincidence form and that have k non-coincidences. Thus,

$$\alpha_i = \beta_i, \quad \forall\ i > k \tag{9.146}$$

and the Slater determinants can be written as

$$\begin{aligned} |\Phi'\rangle &= |\alpha_1\ \ldots\ \alpha_k\ \alpha_{k+1}\ \ldots\ \alpha_N\rangle \\ |\Phi\rangle &= |\beta_1\ \ldots\ \beta_k\ \alpha_{k+1}\ \ldots\ \alpha_N\rangle. \end{aligned} \tag{9.147}$$

Regarding the summations over p_i and a_i in (9.145), from $p_1 < \cdots < p_M$ and $a_1 < \cdots < a_M$ it becomes clear that $a_i > k$ for $i > k$. So, because of (9.146), β_{a_i} can be replaced by α_{a_i}. However, this does not effectively simplify the structure of the matrix element formula. For orthogonal bases, the structure of the determinant in the algebraic complement helps to reduce the equation, but since for non-orthogonal bases there is no information about this determinant, the Slater rules can only be stated as below.

Proposition 9.3.2 ***(Slater Rules for Non-Orthogonal Bases)*** *Let*

$$\begin{aligned} |\Phi'\rangle &= |\alpha_1\ \ldots\ \alpha_k\ \alpha_{k+1}\ \ldots\ \alpha_N\rangle \\ |\Phi\rangle &= |\beta_1\ \ldots\ \beta_k\ \beta_{k+1}\ \ldots\ \beta_N\rangle \end{aligned} \tag{9.148}$$

be two Slater determinants of non-orthogonal single-particle basis states ϕ_i that are in maximum coincidence form. In the k non-coincidence case, the matrix element of a M-body operator $\hat{O}_M$ between these Slater determinants reads

$$\begin{aligned} \langle\Phi'|\ \hat{O}_M\ |\Phi\rangle = &\sum_{\substack{p_1,\ldots,p_M=1 \\ p_1<\cdots<p_M}}^{N}\ \sum_{\substack{a_1,\ldots,a_M=1 \\ a_1<\cdots<a_M}}^{N} A_{S}{}_{a_1\ldots a_M}^{p_1\ldots p_M} \\ &\times \langle\alpha_{p_1}\ldots\alpha_{p_k}\ \alpha_{p_{k+1}}\ldots\alpha_{p_M}|\ \hat{o}_M\ |\beta_{a_1}\ldots\beta_{a_k}\ \alpha_{a_{k+1}}\ldots\alpha_{a_M}\rangle \end{aligned} \tag{9.149}$$

where $A_{S}{}_{a_1\ldots a_M}^{p_1\ldots p_M}$ is defined as

$$A_{S}{}_{a_1\ldots a_M}^{p_1\ldots p_M} = (-1)^{\sum_{z=1}^{M}(p_z+a_z)} \det S(\Phi',\Phi)[R\backslash\{p_1,\ldots,p_M\}, C\backslash\{a_1,\ldots,a_M\}]. \tag{9.150}$$

9.3.4 Slater-Condon Rules for orthogonal Bases

In the case of orthogonal single-particle basis states the Slater rules simplify significantly because due to the orthonormality of the basis functions the overlap matrix only contains the trivial entries

$$S(\Phi', \Phi)_{ij} = \langle \phi_{\alpha_i} | \phi_{\beta_j} \rangle = \delta_{\alpha_i \beta_j} \tag{9.151}$$

which leads to the simple structure of the overlap matrix

$$S(\Phi', \Phi) = \begin{pmatrix} 0_{k\times k} & \\ & \mathbb{1}_{(N-k)\times(N-k)} \end{pmatrix}. \tag{9.152}$$

The algebraic complement $A_{S}{}^{p_1 \dots p_M}_{a_1 \dots a_M}$ essentially is the determinant of the submatrix of $S(\Phi', \Phi)$ where rows p_i and columns a_i have been removed. Clearly, for this determinant not to vanish, the first k rows and columns have to be among the removed ones - otherwise the submatrix would contain an empty row, causing the determinant to vanish. Since $p_1 < \dots < p_M$ and $a_1 < \dots < a_M$ in (9.149), it is mandatory to have $p_i = a_i = i$ for $i = 1, \dots, k$, which eliminates the corresponding summations,

$$\begin{aligned} \langle \Phi' | \, \hat{O}_M \, | \Phi \rangle = & \sum_{\substack{p_{k+1}, \dots, p_M = k+1 \\ p_{k+1} < \dots < p_M}}^{N} \sum_{\substack{a_{k+1} \dots a_M = k+1 \\ a_{k+1} < \dots < a_M}}^{N} A_{S}{}^{1 \dots k, p_{k+1} \dots p_M}_{1 \dots k, a_{k+1} \dots a_M} \\ & \times \langle \alpha_1 \dots \alpha_k \; \alpha_{p_{k+1}} \dots \alpha_{p_M} | \, \hat{O}_M \, | \beta_1 \dots \beta_k \; \alpha_{a_{k+1}} \dots \alpha_{a_M} \rangle. \end{aligned} \tag{9.153}$$

Similarly, $A_{S}{}^{1 \dots k, p_{k+1} \dots p_M}_{1 \dots k, a_{k+1} \dots a_M}$ essentially is the determinant of a submatrix of $\mathbb{1}_{(N-k)\times(N-k)}$ with rows $p_{k+1} \dots p_M$ and columns $a_{k+1} \dots a_M$ removed. For a submatrix of an identity matrix not to contain an empty row or column, the index sets of the removed rows and columns have to be equal (then the submatrix is just again a lower-order identity matrix). Again, because of $p_{k+1} < \dots < p_M$ and $a_{k+1} < \dots < a_M$, one immediately concludes $p_i = a_i$ for $i > k$, which can be used to eliminate the a_i-summations. So, the determinant in the algebraic complement is nothing but a determinant of an identity which equals 1. The factor $(-1)^{\sum_z (p_z + a_z)}$, accompanying the determinant is also trivially 1 due to the fact that $p_z = a_z$, as found above.

Proposition 9.3.3 ***(Slater Rules for Orthogonal Bases)***
Let

$$\begin{aligned} |\Phi'\rangle &= |\alpha_1 \ \ldots \ \alpha_k \ \alpha_{k+1} \ \ldots \ \alpha_N\rangle \\ |\Phi\rangle &= |\beta_1 \ \ldots \ \beta_k \ \beta_{k+1} \ \ldots \ \beta_N\rangle \end{aligned} \tag{9.154}$$

be two Slater determinants of orthogonal single-particle basis states ϕ_i that are in maximum coincidence form. In the k non-coincidence case, the matrix element of a M-body operator $\hat{O}_M$ between these Slater determinants reads

$$\langle\Phi'|\ \hat{O}_M\ |\Phi\rangle = \tag{9.155}$$
$$\sum_{\substack{p_{k+1},\ldots,p_M=k+1 \\ p_{k+1}<\cdots<p_M}}^{N} \langle\alpha_1 \ldots \alpha_k \ \alpha_{p_{k+1}} \ \ldots \alpha_{p_M}|\ \hat{O}_M\ |\beta_1 \ldots \beta_k \ \alpha_{a_{k+1}} \ldots \alpha_{a_M}\rangle.$$

In order to bring this Slater-Condon rules into the notation used in Table 9.1, again define $\square$ as the set of the coinciding orbitals of both Slater determinants,

$$\square \equiv \{\alpha_{k+1}, \ldots, \alpha_N\}. \tag{9.156}$$

Then one can write

$$\langle i'_1 \ldots i'_k \square|\ \hat{O}_M\ |i_1 \ldots i_k \square\rangle = \sum_{\substack{\triangle\in\square^{M-k} \\ \triangle_1<\cdots<\triangle_{M-k}}} \langle i'_1 \ldots i'_k \triangle|\ \hat{o}_M\ |i_1 \ldots i_k \triangle\rangle. \tag{9.157}$$

In particular, one can read off the Slater-Condon rules for three-body operators as :

Slater-Condon Rules for three-body operators		
$\langle \Box \vert \hat{A} \vert \Box \rangle$	$=$	$\sum_{\substack{p,q,r \in \Box \\ p<q<r}} \langle pqr \vert \hat{a} \vert pqr \rangle$
$\langle i' \Box \vert \hat{A} \vert i \Box \rangle$	$=$	$\sum_{\substack{p,q \in \Box \\ p<q}} \langle i'pq \vert \hat{a} \vert ipq \rangle$
$\langle i'j' \Box \vert \hat{A} \vert ij \Box \rangle$	$=$	$\sum_{p \in \Box} \langle i'j'p \vert \hat{a} \vert ijp \rangle$
$\langle i'j'k' \Box \vert \hat{A} \vert ijk \Box \rangle$	$=$	$\langle i'j'k' \vert \hat{a} \vert ijk \rangle$

Table 9.2: *Slater-Condon rules for three-body operators and orthogonal single-particle basis states.*

Bibliography

[1] *Cornelius Lanczos*
An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators
Journal of Research of the National Bureau of Standards, ***45****,4, 255-282 (1950)*

[2] *Yousef Saad*
Numerical Methods for Large Eigenvalue Problems
Manchester University Press

[3] *Yousef Saad*
Iterative Methods for Sparse Linear Systems
P.W.S.-Kent Publishing Co.

[4] *James W. Demmel*
Applied Numerical Linear Algebra
SIAM

[5] *Christopher Conway Paige*
The Computation of Eigenvalues and Eigenvectors of Very Large Sparse Matrices
PhD Thesis, University of London

[6] *Christopher Conway Paige*
Computational Variants of the Lanczos Method for the Eigenproblem
J. Inst. Maths Applics, ***10****, 373-381 (1972)*

[7] *Christopher Conway Paige*
Error Analysis of the Lanczos Algorithm for Tridiagnalizing a Symmetric Matrix
J. Inst. Maths Applics, ***18****, 341-349 (1976)*

[8] *Beresford Parlett*
The Symmetric Eigenvalue Problem
Prentice-Hall Series in Computational Mathematics

[9] *Beresford Parlett, D.S. Scott*
The Lanczos Algorithm With Selective Orthogonalization
Mathematics of Computation, ***33****, 145, 217-238, (1979)*

[10] *Beresford Parlett*
Do We Fully Understand the Symmetric Lanczos Yet?
Proceedings of the Cornelius Lanczos International Centenary Conference (Raleigh, NC, 1993)
93-107, SIAM, Philadelphia, PA, 1994. 65Y20 (65F15)

[11] *Horst D. Simon*
The Lanczos Algorithm With Partial Orthogonalization
Mathematics of Computation, ***42****, 165, 115-142 (1984)*

[12] *Z. Bai, J. Demmel, J. Dongarra, A. Ruhe, and H. van der Vorst, editors*
Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide
SIAM, Philadelphia, 2000

[13] *Gérard Meurant*
The Lanczos and Conjugate Gradient Algorithms
SIAM

[14] *Kesheng Wu, Horst Simon*
Thick-Restart Lanczos Method for Symmetric Eigenvalue Problems
J. Math. Anal. Appl., ***22****, 2000*

[15] *J. Chen, N. Nakajima, M. Okamoto*
Shift-and-invert Lanczos algorithm for ideal MHD stability analysis
*Computer Physics Communication, **113**, 1-9 (1998)*

[16] *Kesheng Wu, Horst Simon*
An Evaluation of the Parallel Shift.and-Invert Lanczos Method
Lawrence Berkely National Laboratory tech report, LBNL-43178

[17] *J. Morrison, A. Watt, R.R. Whitehead*
A novel projection technique in nuclear structure calculations
*J. Phys. A: Math., Nucl. Gen., **7**, 6 (1974)*

[18] *Robert Roth*
Importance Truncation for Large-Scale Configuration Interaction Approaches
*Phsical Review C, **79**, 064324 (2009)*

[19] *Robert Roth, Peter Navrátil*
Ab Initio Study of ^{40}Ca with an Importance-Truncated No-Core Shell Model
*Physical Review Letters, **99**, 092501 (2007)*

[20] *Kris Heyde*
The Nuclear Shell Model
Springer (1990)

[21] *Peter Ring, Peter Schuck*
The Nuclear Shell Model
Springer (2004)

[22] *Jouni Suhonen*
From Nucleons to Nucleus
Springer (2007)

[23] *S.K. Bogner, R.J. Furnstahl, R.J: Perry*
Similarity renormalization group for nucleon-nucleon interactions
Physical Review C, ***75*** *, 061001(R) (2007)*

[24] *R. Roth, S. Reinhard, H. Hergert*
Unitary correlation operator method and similarity renormalization group: Connections and differences
Physical Review C, ***77*** *, 064003 (2008)*

[25] *A.N. Antonov, M.K. Gaidarov, D.N. Kadrev, P.E. Hodgson, E. Moya de Guerra*
Charge density distributions and related form factors in neutron-rich light exotic nuclei
arXiv, nucl-th/0310063v2 (2004)

[26] *A.M. Shirokov, J.P. Vary, A.I. Mazur, T.A. Weber*
Realistic nuclear Hamiltonians: Ab exitu approach
Physical Letters B, ***644****, 1 (2007)*

[27] *Bruce Barrett, Bodgan Mihaila, Steven C. Pieper, Robert B. Wiringa*
Ab initio Calculations of Light Nuclei
Nuclear Physics News, ***13****, 1 (2003)*

[28] *G. Audi, A.H. Wapstra*
The 1995 update to the atomic mass evaluation
Nuclear Physics A, ***595****, 409-480 (1995)*

[29] *S. Raman*
Atomic Data and Nuclear Data Tables, ***36***

[30] *H. Masui, N. Itagaki*
Simplified modeling of cluster-shell competition in carbon isotopes
Physical Review C, ***75****, 054309 (2007)*

[31] *P. Navrátil, J.P. Vary, B.R. Barrett*
Large-basis ab initio no-core shell model and its application to ^{12}C
*Physical Review C, **62**, 054311 (2000)*

[32] *Petr Navrátil, Sofia Quaglioni, Ionel Stetcu, Bruce Barrett*
Recent developments in no-core shell model calculations
arXiv, nucl-th/0904.0463v1 (2009)

[33] *R. Roth*
Importance Truncation for Large-Scale Configuration Interaction Approaches
*Phys. Rev. C **79**, 064324 (2009)*

[34] *R. Roth, H. Hergert, N. Paar, P. Papakonstantinou*
Nuclear Structure in the UCOM Framework: From Realistic Interactions to CollectiveExcitations
Nucl. Phys. A, 788 (2007)

[35] *I. Sick, J.S. McCarthy*
Elastic electron scattering from 12C and 16O
*Nucl. Phys. A, **150**, 3 (1970)*

[36] *E. Epelbaum, A. Nogga, W. Glöckle, H. Kamada, Ulf-G. Meißner, H. Witala*
Three-nucleon forces from chiral effective field theory
*Physical Review C, **66**, 064001 (2002)*

[37] *J. L. Friar*
Three-Nucleon Forces and the Three-Nucleon Systems
*Czechoslovak Journal of Physics, **43**, 3/4 (1993)*

[38] *Volker Koch*
Aspects of Chiral Symmetry
arXiv, nucl-th/9706075v2 (2008)

[39] *Antonio Pich*
Effective Field Theory
arXiv, hep-ph/9806303v1 (1998)

[40] *E. Epelbaum*
Modern Theory of Nuclear Forces
arXiv, nucl-th/0811.1338v1 (2008)

[41] *Petr Navrátil*
Local three-nucleon interaction from chiral effective field theory
arXiv, nucl-th/0707.4680v1 (2008)

[42] *Kenneth Meyer, Glen Hall, Dan Offin*
Introduction to Hamiltonian Dynamical Systems and the N-Body Problem
Springer (2009)

[43] *G.P. Kamuntavicius, R.K. Kalinauskas, B.R. Barrett, S. Michevicius, D. Germanas*
The general harmonic-oscillator brackets: compact expression, symmetries, sums and Fortran code
arXiv, nucl-th/0105009v1 (2001)

[44] *Petr Navrátil, W. Erich Ormand*
Ab initio shell model with a genuine three-nucleon force for the p-shell nuclei
Physical Review C, ***68****, 034305 (2003)*

[45] *P. Navrátil, G.P. Kamuntavicius, B.R. Barrett*
Few-nucleon systems in a translationally invariant harmonic oscillator basis
Physical Review C, ***61****, 044001 (2000)*

[46] *A. Nogga, P. Navrátil, B.R. Barrett, J.P. Vary*
Spectra and binding energy predictions of chiral interactions for ^{7}Li
Physical Review C, ***73****, 064002 (2006)*

[47] *Isaiah Shavitt, Rodney J. Bartlett*
Many-Body Methods in Chemistry and Physics
Cambridge (2009)

[48] *Attila Szabo, Julie Szabo, Neil S. Ostlund*
Modern Quantum Chemistry: Introduction to Advanced Electronic Structure Theory
Dover Pubn. Inc. (1996)

[49] *István Mayer*
Simple Theorems, Proofs, and Derivations in Quantum Chemistry
Kluwer Academic, Plenum Publishers (2003)

[50] *Richard A. Brualdi, Dragos Cvetkovic*
A Combinatorial Approach to Matrix Theory and its Applications
Crc. Pr. Inc. (2008)

[51] *Yoshiaki Ueno, Kazuhisa Maehara*
An Elementary Proof of the Generalized Laplace Expansion Formula
Academic Reports, Fac. Eng. Tokyo Inst. Polytech, ***25****, 1 (2002)*

Acknowledgement

I wish to thank **Prof. Robert Roth** for giving me the opportunity of being part of the tnp^{++} group and for continually providing advice, motivation and work.

I am particularly grateful to **my parents** for the enduring support over the last years.

This world would be a much more pleasant place if each copy of Linux was supplemented by a free **Felix Schmitt** or **Markus Hild**. But since there are only two around I truly appreciate of being among the privileged ones who got their share of them.

With **Joachim Langhammer** and **Angelo Calci** came along, besides valuable physical discussions, all the things I have missed since elementary school. Well, many thanks for that guys.

And of course I will take the opportunity for my annual reminder to **Heiko Hergert**, that I still will need a job in a couple of years. Wherever you hide — my application will find you.

Erklärung zur Master Thesis gemäss §19 Abs. 6 DPO/AT

Hiermit versichere ich, die Master Thesis ohne Hilfe Dritter nur mit den angegebenen Quellen und Hilfsmitteln angefertigt zu haben. Alle Stellen, die aus diesen Quellen entnommen wurden, sind als solche kenntlich gemacht worden. Diese Arbeit hat in gleicher oder ähnlicher Form noch keiner Prüfungsbehörde vorgelegen.

Darmstadt, den 09. März 2010

Sven Binder