



Fakultät für Naturwissenschaften

Honorarprofessur für Computergestützte Quantenchemie

Low-Rank Tensor Approximation in post Hartree-Fock Methods

von der Fakultät für Naturwissenschaften der Technischen Universität Chemnitz
genehmigte Dissertation zur Erlangung des akademischen Grades

doctor rerum naturalium

(Dr. rer. nat.)

vorgelegt von Dipl.-Chem. Udo Benedikt
geboren am 20.07.1983 in Annaberg-Buchholz
eingereicht am 15.05.2013

Gutachter:

Prof. Dr. Alexander A. Auer
Prof. Dr. Sibylle Gemming

Tag der Verteidigung: 21.01.2014

Benedikt, Udo

Low-Rank Tensor Approximation

in post Hartree-Fock Methods

Dissertation, Fakultät für Naturwissenschaften

Technische Universität Chemnitz, Mai 2013

Bibliografische Beschreibung und Referat

Udo Benedikt

Low-Rank Tensor Approximation in post Hartree-Fock Methods

Technische Universität Chemnitz, Fakultät für Naturwissenschaften

Dissertation 2013, 145 Seiten

Die vorliegende Arbeit beschäftigt sich mit der Anwendung neuartiger Tensorzerlegungs- und Tensorrepresentationstechniken in hochgenauen post Hartree-Fock Methoden um das hohe Skalierungsverhalten dieser Verfahren mit steigender Systemgröße zu verringern und somit den “Fluch der Dimensionen” zu brechen. Nach einer vergleichenden Betrachtung verschiedener Representationsformate wird auf die Anwendung des “canonical polyadic” Formates (CP) detailliert eingegangen. Dabei stehen zunächst die Umwandlung eines normalen, indexbasierten Tensors in das CP Format (Tensorzerlegung) und eine Methode der Niedrigrang Approximation (Rangreduktion) für Zweielektronenintegrale in der AO Basis im Vordergrund. Die entscheidende Größe für die Anwendbarkeit ist dabei das Skalierungsverhalten des Ranges mit steigender System- und Basissatzgröße, da der Speicheraufwand und die Berechnungskosten für Tensormanipulationen im CP Format zwar nur noch linear von der Anzahl der Dimensionen des Tensors abhängen, allerdings auch mit der Expansionslänge (Rang) skalieren. Im Anschluss wird die AO-MO Transformation und der MP2 Algorithmus mit zerlegten Tensoren im CP Format diskutiert und erneut das Skalierungsverhalten mit steigender System- und Basissatzgröße untersucht. Abschließend wird ein Coupled-Cluster Algorithmus vorgestellt, welcher ausschließlich mit Tensoren in einer Niedrigrang CP Darstellung arbeitet. Dabei wird vor allem auf die sukzessive Tensorkontraktion während der iterativen Bestimmung der Amplituden eingegangen und die Fehlerfortpflanzung durch Anwendung des Rangreduktions-Algorithmus analysiert. Abschließend wird die Komplexität des gesamten Verfahrens bewertet und Verbesserungsmöglichkeiten der Reduktionsprozedur aufgezeigt.

Stichworte: Niedrigrang Tensor Approximation, Tensorzerlegung, Elektronenstrukturmethoden, post Hartree-Fock Methoden, Coupled-Cluster

Abstract

Udo Benedikt

Low-Rank Tensor Approximation in post Hartree-Fock Methods

Technische Universität Chemnitz, Fakultät für Naturwissenschaften

PhD thesis 2013, 145 Seiten

In this thesis the application of novel tensor decomposition and tensor representation techniques in highly accurate post Hartree-Fock methods is evaluated. These representation techniques can help to overcome the steep scaling behaviour of high level ab-initio calculations with increasing system size and therefore break the “curse of dimensionality”. After a comparison of various tensor formats the application of the “canonical polyadic” format (CP) is described in detail. There, especially the casting of a normal, index based tensor into the CP format (tensor decomposition) and a method for a low rank approximation (rank reduction) of the two-electron integrals in the AO basis are investigated. The decisive quantity for the applicability of the CP format is the scaling of the rank with increasing system and basis set size. The memory requirements and the computational effort for tensor manipulations in the CP format are only linear in the number of dimensions but still depend on the expansion length (rank) of the approximation. Furthermore, the AO-MO transformation and a MP2 algorithm with decomposed tensors in the CP format is evaluated and the scaling with increasing system and basis set size is investigated. Finally, a Coupled-Cluster algorithm based only on low-rank CP representation of the MO integrals is developed. There, especially the successive tensor contraction during the iterative solution of the amplitude equations and the error propagation upon multiple application of the reduction procedure are discussed. In conclusion the overall complexity of a Coupled-Cluster procedure with tensors in CP format is evaluated and some possibilities for improvements of the rank reduction procedure tailored to the needs in electronic structure calculations are shown.

keywords: *low-rank tensor approximation, tensor decomposition, electronic structure methods, post Hartree-Fock methods, Coupled-Cluster*

Die vorliegende Arbeit wurde in der Zeit von September 2008 bis Dezember 2009 unter Leitung von Prof. Dr. Alexander A. Auer in der Juniorprofessur für Theoretische Chemie der Technischen Universität Chemnitz begonnen, von Januar 2010 bis Februar 2012 am Max-Planck-Institut für Eisenforschung in Düsseldorf fortgeführt und von März 2012 bis Mai 2013 am Max-Planck-Institut für Chemische Energieumwandlung in Mülheim an der Ruhr abgeschlossen.

Herrn Prof. Dr. Alexander A. Auer danke ich für die Einführung in die Tiefen der ab-initio Methoden mit allen Vorzügen und Klippen der zugrunde liegenden Physik. Für die gewährte Freiheit bei der Bearbeitung des Themas, die zahlreichen, anregenden Diskussionen, die großzügige Unterstützung dieser Arbeit und die Möglichkeit auch an anderen, praktisch orientierten Projekten zu arbeiten möchte ich mich ganz besonders bedanken.

Prof. Dr. Sibylle Gemming und Prof. Dr. Alexander A. Auer gilt mein Dank für die Begutachtung dieser Arbeit.

Prof. Dr. Dr. h.c. Wolfgang Hackbusch und besonders Dr. Mike Espig danke ich für die fruchtbare Zusammenarbeit und die ausführliche Einführung in das Gebiet der Tensorrepresentationstechniken. Auch wenn der Dialog am Anfang schwer fiel, konnte mir dennoch das Verständnis für verschiedene Tensorformate und die Funktionsweise des Rangreduktionsalgorithmus nahe gebracht werden. Dabei möchte ich vor allem Dr. Mike Espig für das zur Verfügung stellen der Routinen zur Rangreduktion und die stetige Weiterentwicklung der **TensorCalculus** Bibliothek sowie für die Einführung in die objektorientierte C++ Programmierung danken.

Für die Nutzung verschiedener Großrechner möchte ich mich beim Team des Chemnitzer Hochleistungs-Linux-Cluster (CHiC), vor allem bei Frank Mietke und René Oertel für die technische Unterstützung bei Compiler und Queueing-System Problemen, sowie beim Team des Rechenzentrums am MPI für Eisenforschung und am MPI für Chemische Energieumwandlung recht herzlich bedanken.

Bei meinen Kollegen Wolfgang Schneider und Karl-Heinz Böhm bedanke ich mich für die angenehme Arbeitsatmosphäre im Büro sowie die zahllosen Diskussionen zu den verschiedensten, nicht zwingend arbeitsrelevanten Themen.

Ein ganz besonderer Dank gilt meiner Familie, vor allem meinen Eltern und meinen Geschwistern, die immer zu mir gehalten, mich auch in schwierigen Zeiten unterstützt und mir die wichtigen Dinge im Leben nahe gebracht haben.

Abschließend möchte ich mich bei meiner Frau Annemarie Benedikt ganz herzlich bedanken. Sie stand mir in den letzten Jahren immer treu zur Seite, auch wenn die Zeiten in Düsseldorf bzw. Mülheim und die damit verbundenen zahllosen, oft viel zu langen Bahnfahrten zwischen Chemnitz und Düsseldorf sehr hart waren.

*"Was wir mathematisch festlegen,
ist nur zum kleinen Teil ein objektives Faktum,
zum größeren Teil eine Übersicht über Möglichkeiten."*

Werner Heisenberg (1901 – 1976)

Meiner Familie

Contents

List of Figures	iii
List of Tables	v
Nomenclature	vii
1 Introduction	1
2 Theoretical Background	7
2.1 Molecular Electronic Structure Theory	7
2.1.1 Hartree-Fock Method (HF)	8
2.1.2 Perturbation Theory	14
2.1.3 Configuration Interaction (CI)	16
2.1.4 Coupled-Cluster Theory (CC)	19
2.1.4.1 Coupled-Cluster Doubles (CCD)	23
2.1.4.2 Coupled-Cluster Singles and Doubles (CCSD)	24
2.2 Tensor Representation Techniques	25
2.2.1 Canonical Polyadic Format (CP)	26
2.2.2 Tucker Format	29
2.2.3 Hierarchical Format	30
2.2.4 Tensor Train or Matrix Product State Format	31
2.2.5 Comparison of Tensor Formats	32
3 Application of the CP Format in post Hartree-Fock Methods	35
3.1 Casting of Indexed Based Tensors to CP Format	35
3.1.1 Trivial Decomposition	36
3.1.2 Decomposition using Singular Value Decomposition	37
3.1.3 Casting Two-Electron Integrals using Resolution of the Identity (RI) .	40
3.2 Rank Reduction Algorithm	42
3.2.1 Rank Reduction for a Test-Matrix	44
3.3 Adaptation of Reduction Algorithm for Quantum Chemistry	47
3.3.1 First Examples: Water using STO-3G and 6-31G Basis Set	47
3.3.2 Improved Reduction Scheme by Slicing Procedure	52
3.3.3 Comparison of Sliced and Full Reduction Procedure	54
3.3.4 Lowering of Initial Ranks by SVD-based Casting Procedure	56
3.3.5 Slicing Scheme based on SVD Casting Procedure	58

3.4	Benchmark: Scaling of Reduced Ranks	60
3.4.1	Violation of Permutational Symmetry	64
3.5	AO-MO Transformation with Integrals in CP Format	65
3.6	MP2 Algorithm with Integrals in CP Format	74
3.6.1	Definition of MP2 Amplitudes	74
3.6.2	MP2 Energy Equation	79
3.7	LCCD Algorithm with Tensors in Low-Rank CP Representation	84
3.7.1	Amplitude Update	87
3.7.2	Second and Third Term: $\sum_{ef} t_{ij}^{ef} \langle ab ef \rangle$ and $\sum_{mn} t_{mn}^{ab} \langle mn ij \rangle$	88
3.7.3	Fourth Term: $P(ij)P(ab) \sum_{me} t_{im}^{ae} \langle mb ej \rangle$	91
3.7.4	Fifth and Sixth Term: $P(ij) \sum_m f_j^m t_{im}^{ab}$ and $P(ab) \sum_e f_e^b t_{ij}^{ae}$	92
3.7.5	Final Residual Tensor	94
3.7.6	Example: H ₂ O using 6-31G Basis Set	95
3.7.7	Rank Reduction during Iterative Procedure	101
3.8	CCD Algorithm with Tensors in Low-Rank CP Representation	103
3.9	Improved Rank Reduction Algorithm	108
4	Conclusions and Outlook	111
4.1	Outlook	115
A	Appendix	117
A.1	Computational Details	117
A.1.1	Software	117
A.1.2	Automatic Code Generation	117
A.1.3	Workstation	118
A.1.4	Compute node	118
A.1.5	Molecular Geometries	119
A.1.6	Used Basis Sets	119
A.2	Derivation of Tensor Contractions in CCD using Tenors in CP Format	120
A.2.1	Second Term	120
A.2.2	Third Term	121
A.2.3	Sixth Term	121
A.2.4	Increase of Rank during Tensor Contractions	122
A.3	Derivation of CI Method based on CP Tensors	124
A.4	Parameters for Reduction Algorithm	127
A.4.1	Determination of Step Size for Improved Reduction Algorithm	127
A.4.2	Parameters used in Original Implementation	127
A.4.3	Parameters used in New Implementation	128
Bibliography		129

List of Figures

2.1	Example for a third order tensor $\mathcal{T} \in \mathbb{R}^{I \times J \times K}$	26
2.2	Numerical example for the description of a third order tensor by application of the Kronecker product.	27
2.3	CP representation of third order tensor with rank R	28
2.4	Tucker decomposition of third order tensor with multilinear ranks R_1, R_2 and R_3	29
2.5	Hierarchical tensor representation for third order tensor.	30
2.6	Third order tensor in MPS format.	31
2.7	Tensor networks using various tensor formats for tensors of order 4 (for example t_2 amplitudes).	33
3.1	Trivial casting of third order tensor into CP format.	37
3.2	Graphical representation of the decomposition by SVD.	38
3.3	Numerical example for decomposition with the help of SVD.	39
3.4	Schematic representation for the Accelerated Gradient (AG) method.	43
3.5	Scheme for reduction algorithm.	44
3.6	Original grey scale picture of Lena Söderberg	44
3.7	Pictures of Lena Söderberg from approximate tensors with increasing ranks. .	45
3.8	Two-electron integrals of water in AO basis for the STO-3G basis set. . . .	47
3.9	Errors in elements during rank reduction of AO two-electron integrals of water using STO-3G basis.	48
3.10	Two-electron integrals of water in AO basis for the 6-31G basis set.	50
3.11	Errors in elements during rank reduction of AO two-electron integrals of water using 6-31G basis.	51
3.12	Schematic representation of the parallel rank reduction currently used for large examples.	52
3.13	Convergence of error ε in the ℓ^2 -norm for full reduction of AO integrals in CH ₄ , CO and HCN using 6-31G basis set.	54
3.14	Pseudocode for SVD based slicing scheme.	59
3.15	Scaling of reduced ranks for AO-integrals with system size for LiH chains using the 6-31G basis set and with number of basis functions for H ₂ O using the basis set series.	62
3.16	Scaling of reduced ranks of MO integrals with system size in LiH chains calculated using the 6-31G basis set.	70

3.17 Scaling of reduced ranks of MO integrals with system size in alkyl chains (C_nH_{2n+2}) calculated using the 6-31G basis set.	71
3.18 Scaling of reduced ranks of MO integrals with increasing basis set size for water.	72
3.19 Scaling of reduced ranks for the t_2 amplitudes with system size for $(\text{LiH})_n$ and scaling with basis set size for H_2O	77
3.20 Reduced ranks for MP2 t_2 amplitudes in O_2 (using 6-31G basis set) with increasing bond length.	78
3.21 Error in MP2 energies for different values of ε in a series of alkyl chains using the 6-31G basis set.	83
3.22 Convergence of LCCD correlation energy using CP tensors for H_2O in 6-31G basis set.	99
3.23 Convergence of CCD correlation energy using CP tensors for H_2O in 6-31G basis set.	106

List of Tables

2.1	Overview over the hierarchy of truncated Coupled-Cluster methods.	22
2.2	Comparison of various tensor formats.	32
3.1	Test calculations with various slice sizes.	53
3.2	Initial and final ranks for different accuracies for the AO-integrals in HF, H ₂ O, NH ₃ , CH ₄ , CO, N ₂ and HCN using the 6-31G basis set.	55
3.3	Comparison of memory requirements between index based, conventional tensor format and CP format.	56
3.4	Comparison of ranks for trivial casting and decomposition using SVD.	57
3.5	Comparison of accuracies for trivial casting and decomposition using SVD.	58
3.6	Ranks of AO integrals in LiH chains using the 6-31G basis set.	61
3.7	Ranks of AO integrals in H ₂ O using the basis sets of increasing size.	61
3.8	Parameters for fitted functions of AO integrals.	63
3.9	Reduced ranks for sliced reductions (<i>sr</i>) with different slice sizes for AO integrals in H ₂ O using different basis sets.	64
3.10	Reduced ranks for different types of MO integrals of molecules in testset using 6-31G basis set.	67
3.11	Reduced ranks and mean absolute deviations for MO integrals in test set using 6-31G basis set.	68
3.12	Scaling parameter for fitted functions of MO integrals and amplitudes.	73
3.13	Reduced ranks of MP ₂ <i>t</i> ₂ amplitudes and the corresponding two-electron integral for molecules in the testset using 6-31G basis set.	76
3.14	Error in calculated MP2 correlation energy using tensors in CP format (all obtained from reduced AO integrals) for 6-31G basis set.	81
3.15	Error in calculated MP2 correlation energy using tensors in CP format for 6-31G basis set.	82
3.16	Rank of MO integrals for H ₂ O using the 6-31G basis set.	95
3.17	Evolution of ranks during LCCD amplitude equation for H ₂ O using the 6-31G basis set.	96
3.18	Error in LCCD correlation energy using tensors in CP format for H ₂ O using 6-31G basis set.	98
3.19	Reduced ranks of <i>t</i> ₂ amplitudes in CP format (<i>Q</i>) and error in LCCD correlation energy for H ₂ O using 6-31G basis set during iterative solution of amplitude equations.	100

3.20	Initialization of rank reduction during iterative calculation of t_2 amplitude tensor for LCCD method using H ₂ O with 6-31G basis set.	103
3.21	Reduced ranks of t_2 amplitudes in CP format (Q) and error in CCD correlation energy for H ₂ O using 6-31G basis set during iterative solution of amplitude equations.	107
3.22	CPU time per iteration on a single workstation core. For the algorithm using tensors in CP format a threshold of 10^{-2} is applied and the reduction of MO integrals is excluded from the timing.	108
3.23	Timings for different improvements of the reduction algorithm.	110
4.1	Scaling of reduced ranks for two-electron integrals and t_2 amplitudes as obtained from MP2 with increasing system (n) and basis set size (N).	113
4.2	Scaling of computational cost for various tensor manipulations with increasing basis set size.	114

Nomenclature

$\langle pq rs \rangle$	$\iint \phi_p^*(\vec{x}_1) \phi_q^*(\vec{x}_2) \frac{1}{ \vec{x}_1 - \vec{x}_2 } \phi_r(\vec{x}_1) \phi_s(\vec{x}_2) d\vec{x}_1 d\vec{x}_2$
X (bold font)	matrix
X_{pq}	matrix at index p, q
\mathfrak{X} (fraktur)	tensor
\vec{x}	vector
ε	accuracy parameter of rank reduction
a, b, c, d	index for virtual orbitals
e, f	summation index for virtual orbitals
i, j, k, l	index for occupied orbitals
m, n	summation index for occupied orbitals
p, q, r, s	index for general orbitals
Q, R, L	expansion length or rank
$t_r^{(a)}$	representing vector for dimension a
fr	full reduction
sfr	sliced reduction followed by subsequent full reduction
sr	sliced reduction
$\mathcal{O}(N)$	scaling with increasing basis set size
$\mathcal{O}(n)$	scaling with increasing system size
ACRTR	cast AO integrals, reduce, transform to MO basis and further reduce
ACTR	cast AO integrals, transform to MO basis and further reduce
AO	atomic orbital basis
BLAS	Basic Linear Algebra Subprograms
CC	Coupled-Cluster
CCSD	Coupled Cluster Singles Doubles
CCSD(T)	Coupled-Cluster Singles Doubles with perturbative treatment of Triples
CD	Cholesky Decomposition
CI	Configuration Interaction
CP	canonical polyadic format
DF	density fitting
DFT	Density Functional Theory
DMRG	Density Matrix Renormalization Group
FCI	full CI
HOSVD	Higher Order Singular Value Decomposition
HT	Hierarchical Tucker format
MAD	mean absolute deviation

MCR	cast MO integrals and reduce
MO	molecular orbital basis
MP2	second-order Møller Plesset perturbation theory
MPS	matrix product state format
OSV	orbital specific virtuals
OVOS	optimized virtual orbital space
PAO	projected atomic orbitals
PNO	pair natural orbitals
RI	resolution of the identity
SVD	Singular Value Decomposition
THC	tensor hyper contraction
TT	Tensor Train format

1 Introduction

In the last years first principle electronic structure calculations have been developed to powerful tools that can be routinely used in many fields of modern sciences including chemical synthesis and analysis, catalysis, biochemistry, material sciences or energy research. In combination with results from spectroscopy or other experimental measurable data calculations based on quantum mechanics can help to understand the reaction mechanism of chemical reactions on an atomistic level. The accurate description of reaction energies as well as information about species that cannot be observed experimentally like short lived intermediates and the prediction of molecular properties is a key aspect of the ever increasing popularity of quantum chemical calculations. Furthermore, the knowledge of the influence of the electronic structure on chemical reactions can be used to design novel systems with properties tailored for special problems. [1–4]

Nowadays electronic structure methods like density functional theory (DFT) or second-order many-body perturbation theory (MP2) can be routinely used for calculations of systems with ever increasing size (200-1000 atoms). [5–9] Nevertheless, despite the dramatic gain of computational power over the last few decades [10–13] highly accurate post Hartree-Fock calculations can only be afforded for systems with limited size [14,15] due to the steep scaling of these methods with increasing system size. However, to make these highly accurate methods feasible for larger and larger systems the development of new approximations that reduce the computational effort by orders of magnitude, but still achieve a high accuracy, has become a very important aspect of research.

A particularly successful method to reduce the computational effort of electronic structure calculations, is based on the “resolution of the identity” (RI) [16–21] or “density fitting” (DF) approximation. [22–24] Since the 1970s these methods have been applied in the framework of DFT [25–27]. The RI approximation is also used in a variety of other quantum chemical approaches like explicit correlation methods. These have only become feasible by decomposing multi-electron, multi-centre integrals into two-electron integrals by making use of RI techniques. [28–34] Closely related to the RI/DF technique are methods based on the application of the Cholesky Decomposition (CD). [35–40] Both methods, the DF and CD, can be thought of as factorizing higher dimensional tensors into lower dimensional objects that are combined to approximate the target quantity. Through this decoupling of indices the characteristic expensive nested summations occurring in electronic structure methods can be performed as a series of summations with lower complexity.

In the early 90s Almlöf and Häser proposed another approach using a factorizable form of a multi-dimensional object: In Laplace-MP2 the energy denominator is approximated by a factorizing exponential expression. [41–44] Recently Ochsenfeld and co-workers developed this method further to its full potential using linear scaling techniques. [9, 45, 46] With the help of these techniques calculations on systems with 2000 atoms are feasible within a few days on 160 CPUs of a modern computer cluster. [9]

While it is evident that decomposition and factorization are key components for future developments in electronic structure theory, the true potential lies in the application of novel decomposition techniques to the wavefunction parameter tensors in post Hartree-Fock ab initio methods. There, methods based on Configuration Interaction (CI) like the hierarchy of Coupled-Cluster (CC) methods have evolved into highly efficient tools [47, 48] and their accuracy and robustness have been proven in many studies. [49–56] However, the steep scaling of computational effort with increasing system size allows routine calculations only for a limited number of systems. For CCSD(T), which is termed to be the “gold standard” in quantum chemistry [57], the computational effort scales as n^7 , where n is a measure of the system size. Thus, doubling the system size increases the computational effort by a factor of 128.

Almost all methods based on a CI like expansion of the wavefunction suffer from this so called “curse of dimensionality”. [58] This can be seen by the number of variables that are introduced in the parameterization of the wavefunction. In the CCSDT method, for example, up to triple substituted amplitudes - described as six-dimensional parameter arrays - are included. So the number of parameters grows as N^6 (N being the number of basis functions) and the computational effort for evaluating the entries in the amplitude tensor or to perform further calculations often scales even higher than the pure storage requirements.

However, there are many indications that the high scaling of traditional post Hartree-Fock electron correlation methods may well be grossly overparametrized: Electron correlation shows a $\frac{1}{r^6}$ decay behaviour and can be regarded as short ranged. Therefore, the calculation should not have exponential, but rather linear complexity with increasing system size. This is the basic assumption for the development of local correlation methods [59–64], that are pioneered by Saebø and Pulay [60–62] and developed further by Werner, Schütz and co-workers. [22, 65–72] In the framework of local approaches a local basis is constructed in a way that the wavefunction parameter tensors become sparse. Therefore, most of the parameters can be discarded and the methods often exhibit a linear scaling behaviour with increasing system size. Recent approaches based on incremental and divide-and-conquer schemes [73–78] can be regarded as related approximations in a more general sense.

Other techniques related to tensor decomposition and representation in various formats have already been established in the field of post Hartree-Fock methods. As early as in the 1950s Löwdin proposed the usage of natural orbitals in order to achieve enhanced convergence

of the CI series and thus to reduce the number of wavefunction parameters that are required to obtain a certain accuracy. [79–81] This idea has found its way into many modern theories: pair natural orbitals (PNO) [82–86] are used in efficient local approximations [14, 87–90], recently Yang *et al.* [15, 91–93] proposed the orbital specific virtual approximation (OSV) in the framework of the Pulay Saebø local approximation (non-orthogonal projected atomic orbitals PAO) [60–62] and a series of optimized virtual orbital space methods (OVOS) [94–98] minimizes the size of the virtual space.

However, all these truncation schemes are often difficult to control with respect to the achievable accuracy and not always straightforward to extend to any arbitrary kind of electronic structure methods or molecular properties. There, general schemes for tensor representation offer new possibilities: Hino *et al.* describe an ansatz in which the computational effort for the CCSD(T) method can be reduced by applying an approach related to Singular Value Decomposition (SVD) [99–101] to the t_3 amplitudes [102,103]. Bell *et al.* discuss the potential of higher order Singular Value Decomposition (HOSVD) for post Hartree-Fock methods in a recent publication. [104] In the tensor hypercontraction density fitting (THC-DF) proposed by Hohenstein *et al.* [105] three centre integrals arising in the DF procedure are decomposed with the CANDECOMP/PARAFAC procedure [100,106] in order to get a factorized representation of the two-electron integrals and to reduce the scaling of second and third order Møller-Plesset perturbation theory to a quartic scaling. [105] Despite the advantages of these schemes like a rigorous error bounds or the applicability to any kind of tensor, none of the existing schemes have been developed to their full potential. Therefore, the application of general tensor decomposition and tensor representation schemes is investigated in more detail.

General approaches for the decomposition of multi parameter problems have been discussed more intensely in the field of applied mathematics in recent years. [107–111] Using tensor representation techniques it is possible to approximate high dimensional tensors by combinations of objects with lower dimensionality. In this respect DF and CD can be regarded as special cases that are limited to certain types of tensors, like two-electron integrals. More general approaches include various tensor formats, decomposition algorithms and approximation schemes [111–115] can help to reduce the high memory requirements and also the steep scaling of computational costs of post Hartree-Fock methods by approximating also the high dimensional wavefunction parameter tensors with lower dimensional objects. One example for a factorizable approximation of high dimensional quantities is the canonical polyadic or canonical product format (CP). There, a sum of Kronecker products over representing vectors is used to represent a high dimensional tensor. [106, 107, 116–122] Besides the low storage requirements of this representation also the more efficient treatment of further tensor manipulations [123] makes this format attractive for the application in electronic structure methods. Tensor representation in the CP format is used in chemometrics [124–127], numerical linear algebra [128,129], signal processing [130,131], computer vision [132,133], data mining [134,135], neuroscience [136,137] and many others.

The key component for the representation of high dimensional tensors by objects of lower dimensionality is the decomposition procedure itself. While for tensors in two dimensions (matrices) the decomposition can be done by Singular Value Decomposition (SVD) [99], there is no straightforward generalization of the decomposition procedure to tensors of higher order ($d \geq 3$). However, some numerical schemes for the decomposition into different tensor formats exist and for $d = 3$ a decomposition in the Tucker format can be obtained by well defined and stable methods like the high order SVD (HOSVD). This format has already been successfully applied in Hartree Fock and DFT calculations. [138–140] In contrast, the decomposition and compression problem for higher dimensional tensors in the CP format is a non-trivial task. A general minimizer does not exist (ill-posed problem) [141], but recently some contributions to the numerical solution have been made. [128, 142, 143] Nowadays several schemes based on alternating least squares fit [108, 117, 119, 144], a modified Newton method [114, 121, 122] or an accelerated gradient (AG) [145, 146] algorithm are available to handle this problem. Despite the significant progress, the optimization problem remains and alternative tensor formats like the hierarchical tensor format (HT) [115, 147] or the Tree-Tucker format (TT) [148] can offer a further perspective in the representation of high dimensional tensors.

In this work the applicability of tensor representation in the CP format for tensors occurring in electronic structure methods is investigated. The effects of low-rank representation on the memory requirements and the impact on tensor contractions is studied. There, the main focus lies on the representation of wavefunction parameters in CC methods and the influence on the iterative solution of the CC amplitude equations. Especially the scaling of the rank with increasing system and basis set size for different quantities (AO integrals, MO integrals, wavefunction parameter) is assessed so that the general applicability of the CP representation can be outlined. Therefore, the robustness and performance of the rank reduction algorithm has to be investigated.

In the next chapter a short introduction and theoretical background for the used electronic structure methods is given followed by a summary of various mathematical motivated tensor representation techniques. In chapter 3 various casting procedures to obtain a representation in the CP format are assessed and the short overview of the rank reduction procedure is presented. As first example for the decomposition and rank reduction the AO integrals for water are taken and the reduction is explained in detail. Afterwards, a slicing scheme for faster rank reduction and the scaling of reduced ranks with increasing system and basis set size are evaluated. In section 3.5 the transformation from the AO to the MO basis is explained using tensors in CP representation and the scaling of reduced ranks is pointed out. In the following section the t_2 amplitude tensor is defined for the MP2 method by weighting the MO integrals with the energy denominator and the MP2 correlation energy is calculated using tensors in CP representation. Especially the correlation of the error in the low-rank approximation with the error in the energy is investigated. The following section 3.7 describes the impact of the CP representation on tensor contractions of integrals and amplitudes during

the iterative solution of the LCCD and CCD amplitude equations. Each term is evaluated in detail and the computational costs as well as necessary rank reduction steps are pointed out. Finally the convergence of the calculated correlation energy and the relation to the applied threshold for the rank reduction procedure is explained. In the last chapter (4) a summary of the results is given, some conclusions for the general applicability of low-rank tensor representation techniques in post Hartree-Fock methods are drawn and an outlook for further investigations is laid out.

2 Theoretical Background

2.1 Molecular Electronic Structure Theory

To describe the behaviour of electrons in the field of the nuclei one has to solve the time-independent non-relativistic Schrödinger equation [149]

$$\hat{H} \Psi = E \Psi, \quad (2.1)$$

that is a differential equation with the Hamilton-operator \hat{H} , the energy eigenvalue E and the wavefunction Ψ as an eigenfunction. The Hamilton-operator for an arbitrary system is the sum of the kinetic and potential energy. For a molecular system with n electrons and N nuclei the Hamiltonian in atomic units (constants $e = m = \hbar = 1$ where $-e$ and m are the charge and mass of the electron and \hbar is Planck's constant divided by 2π) can be written as

$$\hat{H} = - \sum_i^n \frac{1}{2} \nabla_i^2 - \sum_K^N \frac{1}{2M_K} \nabla_K^2 - \sum_{i,K}^{n,N} \frac{Z_K}{r_{iK}} + \sum_{i < j}^{n,n} \frac{1}{r_{ij}} + \sum_{K < L}^{N,N} \frac{Z_K Z_L}{R_{KL}}. \quad (2.2)$$

In Eqn. 2.2 the first two terms describe the kinetic energy of the electrons and the nuclei, the third stands for the attraction between the electrons and nuclei and the last two terms define the repulsion between the electrons and between the nuclei [150]. The electron-nucleus, electron-electron and nucleus-nucleus distances are denoted as r_{iK} , r_{ij} and R_{KL} while Z_K stands for the atomic number and M_K for the reduced mass of the nuclei.

Taking into account that the mass of the nuclei is much larger than the mass of the electrons, one can assume that the electrons are moving much faster and can follow the movement of the nuclei instantaneously. This leads to the Born-Oppenheimer-approximation [151]

$$\Psi(r_i, R_K) = \psi_{el}(r_i, \bar{R}_K) \psi_{Kern}(R_K), \quad (2.3)$$

where the wavefunction can be separated into an electronic and a nuclei part. The electronic wavefunction no longer depends on the coordinates of the nuclei but rather uses them as a fixed parameter \bar{R}_K . The non-relativistic, time-independent, electronic Schrödinger equation can now be written as

$$\hat{H}_{el} \psi_{el}(r_i, \bar{R}_K) = E_{el} \psi_{el}(r_i, \bar{R}_K), \quad (2.4)$$

with

$$\begin{aligned}\hat{H}_{el} &= \underbrace{-\sum_i \frac{1}{2} \nabla_i^2}_{\text{one-electron part}} - \sum_{i,K} \frac{Z_K}{r_{iK}} + \underbrace{\sum_{i < j} \frac{1}{r_{ij}}}_{\text{two-electron part}} \\ &= \sum_i \hat{h}_i + \sum_{i < j} \frac{1}{r_{ij}}.\end{aligned}\quad (2.5)$$

The full energy of the whole system can be calculated as the sum of the electronic energy E_{el} and the nuclei-interaction energy V_{KL} for fixed nuclear coordinates

$$E = E_{el} + V_{KL}. \quad (2.6)$$

Due to the electron-electron interaction (last term in Eqn. 2.5) the Schrödinger equation can only be solved analytically for systems with one electron. So one must search for an approximate numerical solution for many-electron systems. A very common approach is to use the variational principle [152] to calculate the energy expectation value \tilde{E} of a test wavefunction Φ

$$\tilde{E} = \frac{\langle \Phi | \hat{H} | \Phi \rangle}{\langle \Phi | \Phi \rangle}. \quad (2.7)$$

This energy expectation value for a normalized test function is greater than or equals the exact energy

$$\tilde{E} \geq E_{exakt}. \quad (2.8)$$

One possible way of constructing a trial wavefunction is the Hartree-Fock ansatz that will be explained in the following section.

2.1.1 Hartree-Fock Method (HF)

The main idea of the Hartree-Fock method is to solve the Schrödinger equation for a many-electron system by choosing an appropriate wavefunction. As the exact solution for one-electron systems (like the H-atom) are known, a first way for an approximate many-electron wavefunction is the Hartree-product. The total wavefunction Φ is then the simple product of one-electron hydrogen wavefunctions ϕ (orbitals)

$$\Phi(\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n) = \phi_1(\vec{x}_1) \cdot \phi_2(\vec{x}_2) \cdot \dots \cdot \phi_n(\vec{x}_n). \quad (2.9)$$

As electrons can not only vary in the three spatial coordinates \vec{r} but also have an intrinsic spin variable ω , which is either α or β , a special space-spin coordinate $\vec{x} = \{\vec{r}, \omega\}$ is used for the description of the spin orbitals $\phi(\vec{x})$. These orbitals consist of a spacial orbital $\sigma(\vec{r})$ and a spin function $\varphi(\omega)$

$$\phi(\vec{x}) = \sigma(\vec{r}) \cdot \varphi(\omega). \quad (2.10)$$

As electrons are fermions the wavefunction must be antisymmetric (change sign) under exchange of two electrons. The simple Hartree-product wavefunction (Eqn. 2.9) does not fulfil the antisymmetry principle (example with 2 electrons) in general

$$\begin{aligned}\Phi(\vec{x}_1, \vec{x}_2) &= -\Phi(\vec{x}_2, \vec{x}_1) \\ \phi_1(\vec{x}_1) \cdot \phi_2(\vec{x}_2) &\neq -\phi_1(\vec{x}_2) \cdot \phi_2(\vec{x}_1).\end{aligned}\quad (2.11)$$

To fulfil the antisymmetry principle a linear combination of the two products can be chosen

$$\Phi(\vec{x}_1, \vec{x}_2) = \frac{1}{\sqrt{2}} [\phi_1(\vec{x}_1) \cdot \phi_2(\vec{x}_2) - \phi_1(\vec{x}_2) \cdot \phi_2(\vec{x}_1)]. \quad (2.12)$$

The wavefunction is now antisymmetric and vanishes if the two electrons are in the same orbital (Pauli-principle). In a more general way Eqn. 2.12 can be written as a determinant

$$\Phi(\vec{x}_1, \vec{x}_2) = \frac{1}{\sqrt{2}} \begin{vmatrix} \phi_1(\vec{x}_1) & \phi_2(\vec{x}_1) \\ \phi_1(\vec{x}_2) & \phi_2(\vec{x}_2) \end{vmatrix}. \quad (2.13)$$

This so called Slater determinant [150,153] can easily be extended to systems with n electrons

$$\Phi(\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n) = \frac{1}{\sqrt{n!}} \begin{vmatrix} \phi_1(\vec{x}_1) & \phi_2(\vec{x}_1) & \dots & \phi_n(\vec{x}_1) \\ \phi_1(\vec{x}_2) & \phi_2(\vec{x}_2) & \dots & \phi_n(\vec{x}_2) \\ \dots & & & \dots \\ \phi_1(\vec{x}_n) & \phi_2(\vec{x}_n) & \dots & \phi_n(\vec{x}_n) \end{vmatrix}. \quad (2.14)$$

In the following a very convenient short notation of the Slater determinant for a set of spin-orbitals $\{\phi_1(\vec{x}_1), \phi_2(\vec{x}_2), \dots, \phi_n(\vec{x}_n)\}$ is used

$$\Phi(\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n) \equiv |\Phi_0\rangle = |\phi_1(\vec{x}_1) \phi_2(\vec{x}_2) \dots \phi_n(\vec{x}_n)\rangle = |\phi_1 \phi_2 \dots \phi_n\rangle. \quad (2.15)$$

The electronic energy for a given Slater determinant $|\Phi_0\rangle$ according to the Schrödinger equation can now be evaluated as a functional of the spin orbitals

$$E_0 = E_0[\{\phi_i\}] = \langle \Phi_0 | \hat{H} | \Phi_0 \rangle. \quad (2.16)$$

Substituting the Hamiltonian according to Eqn. 2.5 leads to

$$\begin{aligned}
E_0[\{\phi_i\}] &= \left\langle \Phi_0 \left| \sum_i \hat{h}_i + \sum_{i < j} \frac{1}{r_{ij}} \right| \Phi_0 \right\rangle \\
&= \left\langle \phi_1 \phi_2 \dots \phi_i \phi_j \dots \phi_n \left| \sum_i \hat{h}_i + \sum_{i < j} \frac{1}{r_{ij}} \right| \phi_1 \phi_2 \dots \phi_i \phi_j \dots \phi_n \right\rangle \\
&= \sum_i \left\langle \phi_i \left| \hat{h}_i \right| \phi_i \right\rangle + \frac{1}{2} \sum_{i,j} \langle \phi_i \phi_j | \phi_i \phi_j \rangle - \langle \phi_i \phi_j | \phi_j \phi_i \rangle. \tag{2.17}
\end{aligned}$$

There, the first part are the one-electron integrals

$$\left\langle \phi_i \left| \hat{h}_i \right| \phi_i \right\rangle = \int \phi_i^*(\vec{x}_1) \hat{h}_i \phi_i(\vec{x}_1) d\vec{x}_1, \tag{2.18}$$

with \hat{h}_i defined as in Eqn. 2.5

$$\hat{h}_i = \frac{1}{2} \nabla_i^2 - \sum_K \frac{Z_K}{r_{iK}}, \tag{2.19}$$

while the second part consists of two-electron integrals defined as

$$\langle \phi_i \phi_j | \phi_i \phi_j \rangle = \langle ij | ij \rangle = \iint \phi_i^*(\vec{x}_1) \phi_j^*(\vec{x}_2) \frac{1}{r_{12}} \phi_i(\vec{x}_1) \phi_j(\vec{x}_2) d\vec{x}_1 d\vec{x}_2, \tag{2.20}$$

with $r_{12} = |\vec{x}_1 - \vec{x}_2|$ denoting the distance between the two electrons. As stated in the variational principle [152] the energy functional can now be minimized by variation of the wavefunction. This variation is achieved by changes in the orbitals $\phi_i \rightarrow \phi_i + \delta\phi_i$ under the condition that the orbitals stay orthonormal. This can be achieved by Lagrange's method of undetermined multipliers. The resulting Lagrange functional

$$\mathcal{L}[\{\phi_i\}] = E_0[\{\phi_i\}] - \sum_{ij} \epsilon_{ij} (\langle i | j \rangle - \delta_{ij}), \tag{2.21}$$

contains the unknown multipliers ϵ_{ij} , the overlap integral between two orbitals

$$\langle i | j \rangle = \int \phi_i^*(\vec{x}_1) \phi_j(\vec{x}_1) d\vec{x}_1, \tag{2.22}$$

and the Kronecker delta δ_{ij} , that is 1 for $i = j$ and 0 otherwise. Now the derivative of the Lagrange functional $\mathcal{L}[\{\phi_i\}]$ can be set to zero. After some algebra that includes the

derivatives of all terms in Eqn. 2.17 the following equation is obtained

$$\begin{aligned} \hat{h}_i(\vec{x}_1)\phi_i(\vec{x}_1) + \sum_{j \neq i} \left[\int \phi_j^*(\vec{x}_2) \frac{1}{r_{12}} \phi_j(\vec{x}_2) d\vec{x}_2 \right] \phi_i(\vec{x}_1) \\ - \sum_{j \neq i} \left[\int \phi_j^*(\vec{x}_2) \frac{1}{r_{12}} \phi_i(\vec{x}_2) d\vec{x}_2 \right] \phi_j(\vec{x}_1) = \epsilon_i \phi_i(\vec{x}_1). \end{aligned} \quad (2.23)$$

The values of ϵ_i describe the energy eigenvalues for each orbital ϕ_i . For convenience the two terms in square brackets in Eqn. 2.23 can be reformulated with the introduction of two new operators. The first square bracket

$$\sum_{j \neq i} \left[\int \phi_j^*(\vec{x}_2) \frac{1}{r_{12}} \phi_j(\vec{x}_2) d\vec{x}_2 \right] \phi_i(\vec{x}_1) = \sum_{j \neq i} \left[\int |\phi_j(\vec{x}_2)|^2 \frac{1}{r_{12}} d\vec{x}_2 \right] \phi_i(\vec{x}_1), \quad (2.24)$$

describes the Coulomb interaction of the electron in orbital ϕ_i with the averaged charge distribution of all the other electrons. This term makes Hartree-Fock a “mean field” theory because the Coulomb interaction of an electron is always described within the averaged potential of the other electrons. The averaged potential at the point \vec{x}_1 can be described with the Coulomb operator $\mathcal{J}_j(\vec{x}_1)$

$$\mathcal{J}_j(\vec{x}_1)\phi_i(\vec{x}_1) = \left[\int \phi_j^*(\vec{x}_2) \frac{1}{r_{12}} \phi_j(\vec{x}_2) d\vec{x}_2 \right] \phi_i(\vec{x}_1). \quad (2.25)$$

The second term in square brackets in Eqn. 2.23 arises due to the antisymmetric wavefunction that is required for a proper description of fermions. It is similar to the Coulomb interaction but exchanges the orbitals ϕ_i and ϕ_j . Therefore, this term is called the exchange term and an exchange operator \mathcal{K}_j can be defined as

$$\mathcal{K}_j(\vec{x}_1)\phi_i(\vec{x}_1) = \left[\int \phi_j^*(\vec{x}_2) \frac{1}{r_{12}} \phi_i(\vec{x}_2) d\vec{x}_2 \right] \phi_j(\vec{x}_1). \quad (2.26)$$

With the help of these two effective one-electron operators the Hartree-Fock equation (2.23) can be written in a more compact form

$$\left[\hat{h}_i(\vec{x}_1) + \sum_{j \neq i} \mathcal{J}_j(\vec{x}_1) - \sum_{j \neq i} \mathcal{K}_j(\vec{x}_1) \right] \phi_i(\vec{x}_1) = \epsilon_i \phi_i(\vec{x}_1). \quad (2.27)$$

The index restriction can be removed because the Coulomb and exchange interaction cancel each other for $j = i$

$$[\mathcal{J}_i(\vec{x}_1) - \mathcal{K}_i(\vec{x}_1)] \phi_i(\vec{x}_1) = 0, \quad (2.28)$$

so that one can define a new operator called the Fock operator $f(\vec{x}_1)$ as

$$f(\vec{x}_1) = \hat{h}_i(\vec{x}_1) + \sum_j [\mathcal{J}_j(\vec{x}_1) - \mathcal{K}_j(\vec{x}_1)]. \quad (2.29)$$

Now the Hartree-Fock equation (2.27) becomes a pseudo eigenvalue equation

$$f(\vec{x}_1)\phi_i(\vec{x}_1) = \epsilon_i\phi_i(\vec{x}_1), \quad (2.30)$$

because the Fock operator still depends on all one-electron functions. Roothaan introduced the expansion of the spatial part of orbitals ϕ_i in known hydrogen like atomic orbitals χ_ν (basis functions)

$$\phi_i = \sum_\nu^K C_{\nu i} \chi_\nu, \quad (2.31)$$

so that the orbitals ϕ_i can easily be varied by changes in the coefficient matrix $C_{\nu i}$ [150]. This procedure is also called the LCAO ansatz (linear combination of atomic orbitals) and leads to a different form of the Hartree-Fock equation (2.30)

$$f(\vec{x}_1) \sum_\nu C_{\nu i} \chi_\nu(\vec{x}_1) = \epsilon_i \sum_\nu C_{\nu i} \chi_\nu(\vec{x}_1). \quad (2.32)$$

After multiplying by χ_μ^* from the left and integration one obtains a matrix equation

$$\sum_\nu C_{\nu i} \int \chi_\mu^*(\vec{x}_1) f(\vec{x}_1) \chi_\nu(\vec{x}_1) d\vec{x}_1 = \epsilon_i \sum_\nu C_{\nu i} \int \chi_\mu^*(\vec{x}_1) \chi_\nu(\vec{x}_1) d\vec{x}_1, \quad (2.33)$$

that can be further simplified by introducing the Fock matrix $F_{\mu\nu}$ and the overlap matrix $S_{\mu\nu}$ as

$$F_{\mu\nu} = \int \chi_\mu^*(\vec{x}_1) f(\vec{x}_1) \chi_\nu(\vec{x}_1) d\vec{x}_1, \quad (2.34)$$

$$S_{\mu\nu} = \int \chi_\mu^*(\vec{x}_1) \chi_\nu(\vec{x}_1) d\vec{x}_1. \quad (2.35)$$

For a closed shell system, with even number of electrons n that are all paired in $n/2$ orbitals, the Fock matrix can be expressed by substituting the Fock operator as given in Eqn. 2.29 and expansion of the orbitals in the Coulomb and exchange operator like in Eqn. 2.31

$$\begin{aligned} F_{\mu\nu} &= \int \chi_\mu^*(\vec{x}_1) \hat{h} \chi_\nu(\vec{x}_1) d\vec{x}_1 + \sum_i^{n/2} \sum_{\sigma\rho} C_{\sigma i} C_{\rho i}^* \left[\langle \mu\sigma | \nu\rho \rangle - \frac{1}{2} \langle \mu\sigma | \rho\nu \rangle \right] \\ &= H_{\mu\nu}^0 + \sum_{\sigma\rho} P_{\sigma\rho} \left[\langle \mu\sigma | \nu\rho \rangle - \frac{1}{2} \langle \mu\sigma | \rho\nu \rangle \right]. \end{aligned} \quad (2.36)$$

Here, the one-electron part $H_{\mu\nu}^0$ is the core Hamiltonian, that describes the kinetic energy of all electrons and the attraction between the nuclei and the electrons. In the two-electron

part the density matrix $P_{\sigma\rho}$

$$P_{\sigma\rho} = 2 \sum_i^{n/2} C_{\sigma i} C_{\rho i}^*, \quad (2.37)$$

is introduced and the two-electron integrals are expressed in the braket notation, c.f. Eqn. 2.20. With the Fock and the overlap matrix the Hartree-Fock equation can be transformed into the Hartree-Fock Roothaan equation

$$\sum_{\nu} F_{\mu\nu} C_{\nu i} = \epsilon_i \sum_{\nu} S_{\mu\nu} C_{\nu i}, \quad (2.38)$$

that can also be written in a more compact matrix form

$$\mathbf{F}\mathbf{C} = \mathbf{S}\mathbf{C}\boldsymbol{\epsilon}. \quad (2.39)$$

The orbital energies ϵ_i are obtained as the diagonal elements of the matrix $\boldsymbol{\epsilon}$. For an orthonormal basis ($\mathbf{S} = \mathbf{1}$) Eqn. 2.39 is an eigenvalue equation that can be solved by diagonalizing the Fock matrix \mathbf{F} to obtain the eigenvalues ϵ and the eigenvectors \mathbf{C} . Since the Fock operator depends on the orbitals, i.e. on its eigenvectors \mathbf{C} , the Hartree-Fock equations are non linear and have to be solved iteratively. This procedure is often called the self consistent field (SCF) method. The basic idea is to start with a simple initial guess for the molecular orbitals (often used: $\mathbf{C} = 0$) and set up the Fock operator that defines the averaged field of the electrons. Then the Fock matrix can be diagonalized and a new set of coefficients \mathbf{C} is obtained. With this a new Fock matrix can be constructed and the procedure is repeated. Self consistency is reached if the coefficients for constructing the Fock matrix are the same (within a given accuracy threshold) as the eigenvectors of the Fock matrix or if the Hartree-Fock energy does not change between two cycles. The Hartree-Fock energy can be obtained according to Eqn. 2.17 as

$$E_{HF} = \frac{1}{2} \sum_{\mu\nu} P_{\mu\nu} (H_{\mu\nu}^0 + F_{\mu\nu}) + V_{KL}, \quad (2.40)$$

including the nuclear repulsion energy V_{KL} .

The Hartree-Fock approximation is a very important starting point as a first approximation to solve the Schrödinger equation. It simplifies the complicated many-electron problem to effective one-electron problems by treating the electron electron interaction in an averaged way. Here a single electron only interacts with the averaged potential of the remaining electrons so that electrons do not interact directly with each other. Therefore, the Hartree-Fock energy can never reach the exact limit due to the insufficient treatment of electron correlation. The difference of the Hartree Fock energy in an infinite and complete basis set and the exact solution of the non relativistic, time independent Schrödinger equation is called correlation energy [1]

$$\Delta E_{corr} = E_{exact} - E_{HF}. \quad (2.41)$$

The electron correlation is often discussed in the context of two different components: The “dynamic correlation” describes the instantaneous interaction of two electrons while “static correlation” refers to cases in which the ground state wave function is only poorly described by a single Slater determinant. It is very difficult to clearly separate the two components from each other but there exist different approaches to treat these kinds of electron correlation. To overcome the static correlation problem one can use a multi configuration ansatz (MCSCF) to avoid near degeneracies between different configurations e.g. in open shell systems or even closed shell systems near to bond breaking. For a proper treatment of dynamic correlation there exist various “post Hartree-Fock” methods like perturbation theory, Configuration Interaction or Coupled-Cluster theory that aim at calculating the electron electron correlation directly.

2.1.2 Perturbation Theory

One ansatz that includes the description of dynamic correlation is derived from the Rayleigh-Schrödinger perturbation theory, where the Hamiltonian is partitioned into a zero-order part \hat{H}_0 and a perturbation \hat{V} , that is included with an order parameter λ classifying various contributions by their order

$$\hat{H} = \hat{H}_0 + \lambda \hat{V}. \quad (2.42)$$

Møller and Plesset [154] used this to formulate the so called Møller-Plesset perturbation theory, based upon the Hartree-Fock solution. There, the zero-order Hamiltonian can be expressed as the sum of the Fock-operators (c.f. Eqn. 2.29)

$$\hat{H}_0 = \sum_i f(i) \quad (2.43)$$

and the perturbation as the difference between the exact correlation and the mean field treatment from the Hartree-Fock method (c.f. Eqn. 2.25 and 2.26)

$$\hat{V} = \frac{1}{r_{12}} - \sum_i [\mathcal{J}_i - \mathcal{K}_i]. \quad (2.44)$$

This ansatz can be plugged into the Schrödinger equation

$$\hat{H} |\Psi_i\rangle = (\hat{H}_0 + \lambda \hat{V}) |\Psi_i\rangle = \epsilon_i |\Psi_i\rangle, \quad (2.45)$$

leading to the exact wavefunction Ψ_i of the system. The eigenvalues and eigenfunctions can now be expanded in a Taylor series with the help of the order parameter λ :

$$\epsilon_i = E_i^{(0)} + \lambda E_i^{(1)} + \lambda^2 E_i^{(2)} + \dots + E_i^{(n)} \quad (2.46)$$

$$|\Psi_i\rangle = |\Phi_i^{(0)}\rangle + \lambda |\Phi_i^{(1)}\rangle + \lambda^2 |\Phi_i^{(2)}\rangle + \dots + \lambda^n |\Phi_i^{(n)}\rangle \quad (2.47)$$

Here, $E_i^{(n)}$ denotes the n -th order energy, while $\Phi_i^{(n)}$ represents the n -th order wavefunction. Assuming intermediate normalization [47] it is possible to calculate energies up to certain order parameters:

$$n = 0 \quad \hat{H}_0 |\Phi_i^{(0)}\rangle = E_i^{(0)} |\Phi_i^{(0)}\rangle \quad (2.48)$$

$$n = 1 \quad \hat{H}_0 |\Phi_i^{(1)}\rangle + \hat{V} |\Phi_i^{(0)}\rangle = E_i^{(0)} |\Phi_i^{(1)}\rangle + E_i^{(1)} |\Phi_i^{(0)}\rangle \quad (2.49)$$

$$n = 2 \quad \hat{H}_0 |\Phi_i^{(2)}\rangle + \hat{V} |\Phi_i^{(1)}\rangle = E_i^{(0)} |\Phi_i^{(2)}\rangle + E_i^{(1)} |\Phi_i^{(1)}\rangle + E_i^{(2)} |\Phi_i^{(0)}\rangle \quad (2.50)$$

...

Projection onto the reference state $\Phi_i^{(0)}$ leads to the energy expressions for different orders

$$E_i^{(0)} = \langle \Phi_i^{(0)} | \hat{H}_0 | \Phi_i^{(0)} \rangle \quad (2.51)$$

$$E_i^{(1)} = \langle \Phi_i^{(0)} | \hat{V} | \Phi_i^{(0)} \rangle \quad (2.52)$$

$$E_i^{(2)} = \langle \Phi_i^{(0)} | \hat{V} | \Phi_i^{(1)} \rangle \quad (2.53)$$

and it can be seen, that the first order energy can be obtained from the zero order wavefunction. This way the first order wavefunction can be evaluated with the help of the Schrödinger equation for $n = 1$ (c.f. Eqn. 2.49) and the first order energy and so on. Substituting the expression for the perturbation \hat{V} , more compact equations containing two-electron integrals and orbital energies only can be written. [150]

$$E_0^{(0)} = \sum_i \epsilon_i \quad (2.54)$$

$$E_0^{(1)} = -\frac{1}{2} \sum_{ij} \langle ij || ij \rangle \quad (2.55)$$

$$E_0^{(2)} = \frac{1}{4} \sum_{abij} \frac{|\langle ab || ij \rangle|^2}{-\epsilon_a - \epsilon_b + \epsilon_i + \epsilon_j} \quad (2.56)$$

Here i, j denote occupied molecular orbitals, while a, b stand for unoccupied, virtual molecular orbitals. The used integrals are antisymmetrized two-electron integrals (c.f. Eqn. 2.20 for Bra-ket notation) defined as

$$\langle ab || ij \rangle = \langle ab | ij \rangle - \langle ab | ji \rangle. \quad (2.57)$$

It can be seen, that the sum of the zero and first order term is the Hartree-Fock energy and therefore the first correction to the Hartree-Fock solution arises for second order Møller Plesset perturbation theory (MP2).

This method yields a good cost-accuracy ratio as the computational effort scales with N^5 , where N is the number of basis functions, due to the integral transformation into the MO basis. Modern implementations using linear scaling techniques and localization schemes en-

able the routine calculation of systems with more than 1000 atoms and 10000 basis functions on a single CPU core, while on larger computer clusters calculations with 2025 atoms and 20371 basis functions are feasible within a few days. [9, 155–158] The MP2 method with sufficiently large basis sets typically has an error in calculated bond length of 2 pm [159] and approximately 20 kJ/mol for reaction enthalpies [160, 161]. Also molecular properties like NMR chemical shifts [162–165] or vibrational frequencies [2, 166] are in good agreement with experimental values.

The basic assumption in perturbation theory is the fact that the perturbation is small. However, this is not true in all cases so that there is no guarantee, that the energy for an increasing perturbation order will converge. In fact there are many studies showing a divergent behaviour for an increasing order in the MP- n methods. [167–169] Therefore, more systematic expansions of the wavefunction have to be used in order to get a hierarchy of improving methods as for example in the configuration interaction approach.

2.1.3 Configuration Interaction (CI)

The conceptually most simple but computationally hardest approach - both in terms of the implementation as well as the computational effort of the method - to treat dynamic correlation is the Configuration Interaction. There the exact wavefunction for a system with n electrons is represented as a linear combination of n -electron trial functions. [170] The Schrödinger equation can then be solved by diagonalizing the n -electron Hamiltonian in the basis of n -electron functions. For a complete basis set the exact energies for the ground state and all excited states are obtained. Therefore, CI provides the exact solution of the many-electron problem.

In CI the exact n -electron wavefunction can be constructed as a linear combination of all possible Slater determinants from a complete set of basis functions $\{\chi_i(\vec{x})\}$. [150] However, it is very convenient to express the different Slater determinants by the difference to the Hartree-Fock determinant. For example, the substitution of an occupied orbital ϕ_i by a virtual orbital ϕ_a leads to a single substituted determinant $|\Phi_i^a\rangle$, while a two fold substituted determinant is described by $|\Phi_{ij}^{ab}\rangle$. So the full CI (FCI) wavefunction can be expressed as

$$|\Phi_{CI}\rangle = c_0 |\Phi_0\rangle + \sum_{i,a} c_i^a |\Phi_i^a\rangle + \sum_{\substack{i < j \\ a < b}} c_{ij}^{ab} |\Phi_{ij}^{ab}\rangle + \sum_{\substack{i < j < k \\ a < b < c}} c_{ijk}^{abc} |\Phi_{ijk}^{abc}\rangle + \dots, \quad (2.58)$$

using expansion coefficients c for the different orders of substituted determinants. To make Eqn. 2.58 even more compact the determinants can also be expressed in second quantization. There, the creation operator \hat{a}_a^\dagger creates the orbital ϕ_a in a Slater determinant while the

annihilation operator \hat{a}_i annihilates the orbital ϕ_i :

$$\hat{a}_a^\dagger |\phi_i \phi_j \phi_k \dots\rangle = |\phi_a \phi_i \phi_j \phi_k \dots\rangle, \quad (2.59)$$

$$\hat{a}_i |\phi_i \phi_j \phi_k \dots\rangle = |\phi_j \phi_k \dots\rangle. \quad (2.60)$$

With the help of the creation and annihilation operators the substituted determinants can all be expressed relative to the reference determinant $|\Phi_0\rangle$

$$|\Phi_i^a\rangle = \hat{a}_a^\dagger \hat{a}_i |\Phi_0\rangle, \quad (2.61)$$

$$|\Phi_{ij}^{ab}\rangle = \hat{a}_a^\dagger \hat{a}_i \hat{a}_b^\dagger \hat{a}_j |\Phi_0\rangle, \quad (2.62)$$

$$|\Phi_{ijk}^{abc}\rangle = \hat{a}_a^\dagger \hat{a}_i \hat{a}_b^\dagger \hat{a}_j \hat{a}_c^\dagger \hat{a}_k |\Phi_0\rangle, \quad (2.63)$$

$$\dots = \dots,$$

so that Eqn. 2.58 can be written as

$$\begin{aligned} |\Phi_{CI}\rangle &= c_0 |\Phi_0\rangle + \sum_{ia} c_i^a \hat{a}_a^\dagger \hat{a}_i |\Phi_0\rangle + \frac{1}{4} \sum_{ijab} c_{ij}^{ab} \hat{a}_a^\dagger \hat{a}_i \hat{a}_b^\dagger \hat{a}_j |\Phi_0\rangle + \dots \\ &= \hat{C}_0 |\Phi_0\rangle + \hat{C}_1 |\Phi_0\rangle + \hat{C}_2 |\Phi_0\rangle + \dots = \hat{C} |\Phi_0\rangle. \end{aligned} \quad (2.64)$$

In an infinite basis set the infinite set of all Slater determinants $\{|\Phi_i\rangle\} = \{|\Phi_0\rangle, |\Phi_i^a\rangle, |\Phi_{ij}^{ab}\rangle, \dots\}$ spans a complete set for the expansion of any n -electron wave function. [170] Applying the linear variational theorem an eigenvalue equation is obtained

$$\mathbf{H}\mathbf{c} = E\mathbf{c}, \quad (2.65)$$

where the matrix \mathbf{c} contains the coefficients for the description of all substitutions. The energy E for the ground and all excited states can be calculated as the eigenvalues of the Hamilton matrix \mathbf{H} and the expansion coefficients in \mathbf{c} are obtained as the eigenvectors. In the Hamilton matrix all interactions of different substituted determinants are ordered in a matrix with elements in the form of $\langle \Phi_i | \hat{H} | \Phi_j \rangle$, where Φ_i and Φ_j are elements of $\{|\Phi_i\rangle\}$ (see also Eqn. 2.66, the lower triangular matrix is skipped due to the hermitian structure of the matrix).

$$\left(\begin{array}{cccccc} \langle \Phi_0 | \hat{H} | \Phi_0 \rangle & 0 & \langle \Phi_0 | \hat{H} | \Phi_{ij}^{ab} \rangle & 0 & 0 & \dots \\ \langle \Phi_i^a | \hat{H} | \Phi_i^a \rangle & \langle \Phi_i^a | \hat{H} | \Phi_{ij}^{ab} \rangle & \langle \Phi_i^a | \hat{H} | \Phi_{ijk}^{abc} \rangle & 0 & \dots \\ \langle \Phi_{ij}^{ab} | \hat{H} | \Phi_{ij}^{ab} \rangle & \langle \Phi_{ij}^{ab} | \hat{H} | \Phi_{ijk}^{abc} \rangle & \langle \Phi_{ij}^{ab} | \hat{H} | \Phi_{ijkl}^{abcd} \rangle & \dots \\ \langle \Phi_{ijk}^{abc} | \hat{H} | \Phi_{ijk}^{abc} \rangle & \langle \Phi_{ijk}^{abc} | \hat{H} | \Phi_{ijkl}^{abcd} \rangle & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{array} \right) \quad (2.66)$$

Here it can be seen, that the singly substituted determinants do not interact with the closed shell Hartree-Fock wave function, as also stated in the Brillouin theorem [150], due to vanishing Hamilton matrix elements

$$\left\langle \Phi_0 \left| \hat{H} \right| \Phi_i^a \right\rangle = 0 = \langle \phi_i | f | \phi_a \rangle. \quad (2.67)$$

This is true because the matrix element that mixes single substitutions with the reference function is equal to the off diagonal elements of the Fock matrix. Due to the block-diagonal structure of the Fock operator f (c.f. Eqn. 2.29) these elements become zero. Furthermore, each determinant can only interact with other determinants that are two orders higher or lower because the Hamiltonian includes only interactions up to two electrons. Therefore, $\langle \Phi_0 | \hat{H} | \Phi_{ij\dots}^{ab\dots} \rangle = 0$ for more than two fold substitutions so that highly substituted determinants do not interact directly with the reference determinate.

In principle, the CI method (FCI) is capable of solving the many-electron problem exactly. However, in practical applications one can only use a finite one-electron basis set so that the full functional space can not be spanned completely. Furthermore, the number of Slater determinants that have to be considered for the FCI wave function is huge and increases exponential with the number of electrons. For a system with n electrons and a one-electron basis with K basis functions $\binom{K}{n}$ n -electron determinants can be constructed. Even for small molecules (only few electrons) and small basis sets the number of Slater determinants is enormous. Therefore, the full CI method is only used for small systems as a benchmark for other methods. [166]. With modern computer technology FCI calculations for systems with 10-14 electrons are feasible. [170, 171]

However, to make the CI method applicable to larger systems the expansion of the CI wavefunction can be truncated. A very reasonable approximation is the CI singles and doubled (CISD) method, where only single and double substitutions are included in the wavefunction

$$|\Phi_{CISD}\rangle = c_0 |\Phi_0\rangle + \sum_{i,a} c_i^a |\Phi_i^a\rangle + \sum_{\substack{i < j \\ a < b}} c_{ij}^{ab} |\Phi_{ij}^{ab}\rangle. \quad (2.68)$$

The CI singles (CIS) method does not improve a closed shell Hartree-Fock wave function due to vanishing Hamilton matrix elements of the single substituted determinant and the reference wavefunction (c.f. Eqn. 2.67). Nevertheless, truncated CI methods are often used if the excited states are of interest and excitation energies need to be calculated. [170, 172, 173]

As a method for the accurate description of the correlation energy truncated CI methods suffer from the problem that they are not size extensive and size consistent. [1, 4] A method is called size extensive if the energy scales linear with the number of calculated particles. [174, 175] An example are N non interacting particles like Helium atoms far apart from each

other, for which the following equation is valid

$$E(N \cdot \text{He}) = N \cdot E(\text{He}). \quad (2.69)$$

This property can be related to thermodynamics, where extensive quantities scale with the system size.

In a size consistent method the energy of two non interacting subsystems A and B at very large distances must be the sum of the energies of the two fragments calculated separately. [176]

$$E(A \xrightarrow{d \rightarrow \infty} B) = E(A) + E(B) \quad (2.70)$$

The correct dissociation of the two fragments at the reference level is required. Therefore, restricted Hartree-Fock is size extensive, but is not necessarily size consistent, due to the lack of describing the dissociation into open shell fragments.

As truncated CI methods are not size consistent and size extensive they are not the method of choice for an accurate treatment of ground state electronic structure nowadays. There, other methods like Coupled-Cluster theory, that do not suffer from size inconsistency and have a comparable computational effort, are more common.

2.1.4 Coupled-Cluster Theory (CC)

The Coupled-Cluster method was introduced in 1966 by Cizek [177] and has been developed to a highly accurate standard method to predict molecular geometries, reaction energies and properties. [1, 49, 178–181] Due to the missing size consistency of the linear CI expansion of the wavefunction, an exponential ansatz is chosen to include dynamic correlation into the description of a system.

$$|\Phi_{CC}\rangle = e^{\hat{T}} |\Phi_0\rangle \quad (2.71)$$

Here the Coupled-Cluster wavefunction Φ_{CC} is expanded with the exponential form of Cluster operator \hat{T} acting on the reference determinate Φ_0 obtained from the Hartree-Fock method. The Cluster operator \hat{T} is defined in an analog way as shown for the CI method (c.f. Eqn. 2.64)

$$\hat{T} = \hat{T}_1 + \hat{T}_2 + \hat{T}_3 + \dots \quad (2.72)$$

$$= \sum_{ia} t_i^a \hat{a}_a^\dagger \hat{a}_i + \frac{1}{4} \sum_{ijab} t_{ij}^{ab} \hat{a}_a^\dagger \hat{a}_i \hat{a}_b^\dagger \hat{a}_j + \frac{1}{36} \sum_{ijkabc} t_{ijk}^{abc} \hat{a}_a^\dagger \hat{a}_i \hat{a}_b^\dagger \hat{a}_j \hat{a}_c^\dagger \hat{a}_k + \dots \quad (2.73)$$

The coefficient tensors $t_{ij\dots}^{ab\dots}$ are called amplitudes and a series of creation \hat{a}_a^\dagger and annihilation \hat{a}_i operators is used to construct substituted determinants. Note that here and in the following a, b, \dots denote virtual indices, while i, j, \dots denote occupied indices. If a summation is done over the index then virtual orbitals are labelled with e, f, \dots and occupied orbitals are indicated by m, n, \dots . For general indices (orbitals that belong either to the occupied or

to the virtual block) p, q, \dots are used. The general n -body cluster operator creating a n -fold substitution can be constructed as

$$\hat{T}_n = \frac{1}{(n!)^2} \sum_{\substack{ij\dots \\ ab\dots}} t_{ij\dots}^{ab\dots} \hat{a}_a^\dagger \hat{a}_i \hat{a}_b^\dagger \hat{a}_j \dots \quad (2.74)$$

Due to the exponential ansatz of the wavefunction the cluster operator does not only occur linear, but also product terms are included. This can be shown by the Taylor expansion of the exponential wave operator

$$e^{\hat{T}} = 1 + \hat{T} + \frac{1}{2!} \hat{T}^2 + \frac{1}{3!} \hat{T}^3 + \dots, \quad (2.75)$$

and the substitution in the definition of the Coupled-Cluster wavefunction in Eqn. 2.71

$$\begin{aligned} |\Phi_{CC}\rangle &= \left(1 + \hat{T}_1 + \frac{1}{2!} \hat{T}_1^2 + \frac{1}{3!} \hat{T}_1^3 + \dots \right. \\ &\quad + \hat{T}_2 + \frac{1}{2!} \hat{T}_2^2 + \frac{1}{3!} \hat{T}_2^3 + \dots \\ &\quad \left. + \hat{T}_1 \hat{T}_2 + \frac{1}{2!} \hat{T}_1^2 \hat{T}_2 + \frac{1}{2!} \hat{T}_1 \hat{T}_2^2 + \dots \right) |\Phi_0\rangle. \end{aligned} \quad (2.76)$$

Here the direct connections of \hat{T}_n with the reference function are called connected clusters, while all product terms like $\hat{T}_1 \hat{T}_2$ are called disconnected clusters. The size extensivity of the Coupled-Cluster method is a direct consequence of the disconnected clusters that occur due to the exponential form of the wave function. [182]

To obtain an equation to calculate the Coupled-Cluster energy the ansatz for the wavefunction has to be plugged into the Schrödinger Equation (c.f. Eqn. 2.1)

$$\hat{H} e^{\hat{T}} |\Phi_0\rangle = E e^{\hat{T}} |\Phi_0\rangle. \quad (2.77)$$

This equation can now be treated in a variational way to minimize the energy, but the resulting equations are very complex and no natural truncations occur. [47] A more convenient way is to use a projective technique. This means to multiply Eqn. 2.77 with the reference wavefunction from the left side

$$\langle \Phi_0 | \hat{H} e^{\hat{T}} | \Phi_0 \rangle = E \langle \Phi_0 | e^{\hat{T}} | \Phi_0 \rangle. \quad (2.78)$$

If intermediate normalization is assumed, $\langle \Phi_0 | \Phi_{CC} \rangle = 1$, this equation can be further simplified to

$$\langle \Phi_0 | \hat{H} e^{\hat{T}} | \Phi_0 \rangle = E. \quad (2.79)$$

Multiplying Eqn. 2.77 from left with with a substituted determinant

$$\langle \Phi_{ij\dots}^{ab\dots} | \hat{H} e^{\hat{T}} | \Phi_0 \rangle = E \langle \Phi_{ij\dots}^{ab\dots} | e^{\hat{T}} | \Phi_0 \rangle, \quad (2.80)$$

can be used to determine the unknown amplitudes. There the projection to determinants with different substitution levels define the equations to calculate the amplitudes of the same level. Due to the presence of $e^{\hat{T}}$ these equations are non linear and still energy dependent.

Substituting the $e^{\hat{T}}$ in the Coupled-Cluster energy expression in Eqn. 2.79 with the Taylor expanded power series one obtains

$$\langle \Phi_0 | \hat{H} (1 + \hat{T} + \frac{1}{2!} \hat{T}^2 + \frac{1}{3!} \hat{T}^3 + \dots) | \Phi_0 \rangle = E, \quad (2.81)$$

which can be written as

$$\langle \Phi_0 | \hat{H} | \Phi_0 \rangle + \langle \Phi_0 | \hat{H} \hat{T} | \Phi_0 \rangle + \langle \Phi_0 | \frac{1}{2!} \hat{H} \hat{T}^2 | \Phi_0 \rangle + \langle \Phi_0 | \frac{1}{3!} \hat{H} \hat{T}^3 | \Phi_0 \rangle + \dots = E. \quad (2.82)$$

Slater's rules show that matrix elements between the Hamiltonian and determinants that differ in more than two orbitals are zero. [150] Therefore, the above shown series terminates after the third term, so that the energy equation can be simplified to

$$\langle \Phi_0 | \hat{H} | \Phi_0 \rangle + \langle \Phi_0 | \hat{H} \hat{T} | \Phi_0 \rangle + \langle \Phi_0 | \frac{1}{2!} \hat{H} \hat{T}^2 | \Phi_0 \rangle = E. \quad (2.83)$$

From a formal point of view these equations give a good insight into the Coupled-Cluster theory. However, for a real computer implementation the equations have to be rewritten in terms of one- and two-electron integrals and the orbital energies, that are already known from the solution of the Hartree-Fock problem. To simplify this task it is convenient to use another projection with the inverse Cluster operator $e^{-\hat{T}}$ so that the energy equation (c.f. Eqn. 2.79) and the amplitude equations (c.f. Eqn. 2.80) become

$$\langle \Phi_0 | e^{-\hat{T}} \hat{H} e^{\hat{T}} | \Phi_0 \rangle = E \quad (2.84)$$

$$\langle \Phi_{ij\dots}^{ab\dots} | e^{-\hat{T}} \hat{H} e^{\hat{T}} | \Phi_0 \rangle = 0. \quad (2.85)$$

This way the amplitude equations are decoupled from the energy but they are still non linear. These equations involve the similarity transformed Hamiltonian $e^{-\hat{T}} \hat{H} e^{\hat{T}}$, which can be simplified further by the Campbell-Baker-Hausdorff-expansion [47,183] that leads to nested commutators of the amplitudes and the Hamiltonian

$$\begin{aligned} e^{-\hat{T}} \hat{H} e^{\hat{T}} = & \hat{H} + [\hat{H}, \hat{T}] + \frac{1}{2!} [[\hat{H}, \hat{T}], \hat{T}] \\ & + \frac{1}{3!} [[[[\hat{H}, \hat{T}], \hat{T}], \hat{T}] + \frac{1}{4!} [[[[[\hat{H}, \hat{T}], \hat{T}], \hat{T}], \hat{T}] + \dots \end{aligned} \quad (2.86)$$

To evaluate the commutators the Hamiltonian (see also Eqn. 2.17) also has to be represented in second quantization

$$\hat{H} = \sum_{pq} h_{pq} \hat{a}_p^\dagger \hat{a}_q + \frac{1}{4} \sum_{pqrs} \langle pq \| rs \rangle \hat{a}_p^\dagger \hat{a}_q^\dagger \hat{a}_r \hat{a}_s, \quad (2.87)$$

where $h_{pq} = \langle \phi_p | \hat{h} | \phi_q \rangle$ stands for the one-electron part, while the antisymmetrized two electron integral $\langle pq \| rs \rangle = \langle \phi_p \phi_q | \phi_r \phi_s \rangle - \langle \phi_p \phi_q | \phi_s \phi_r \rangle$ stands for the two-electron part. With the help of commutator arithmetic [182] one can show that each commutator between \hat{H} and \hat{T} transforms one general index of the Hamiltonian into a Kronecker delta function. As the Hamiltonian has only 4 different general operators, there can only be a 4 fold nested commutator with \hat{T} and therefore the Campbell-Baker-Hausdorff-expansion naturally truncates after the 5 shown terms in Eqn. 2.86. It should be noted, that the special form of the nested commutators leads to the proven term-wise size extensivity of the CC method. [182] It is now possible to insert the Campbell-Baker-Hausdorff-expansion into the amplitude and energy equation (Eqn. 2.85 and 2.84), but as the Cluster operator still contains a large order of substitutions (up to all electrons in the virtual space) it is more convenient to introduce a truncation in the Cluster operator first.

If the full Cluster operator (Eqn. 2.75) is taken into account the method would give an exact solution and be the same as full-CI [184], because all possible substitutions are considered. The effort for the calculation is very large so that for practical applications the Cluster operator is truncated. Truncation at different substitution levels leads to different methods in a hierarchy. These methods are named with the acronyms for the included substitution levels and a short summary is shown in Tab. 2.1. As already shown for the CIS (CI Singles) method, including only single substitutions does not improve the Hartree-Fock results due to the Brillouins theorem (c.f. Sec. 2.1.3). Therefore, there is also no CCS (CC Singles) method. The most important effects are captured with single and double substitutions, while the contributions of higher substitutions decrease with an increasing substitution order.

Table 2.1: Overview over the hierarchy of truncated Coupled-Cluster methods (S=singles, D=doubles, T=triples, Q=quadruples, ...). The column “scaling” shows the scaling of computational effort with the system size N and more detailed with the number of occupied (occ) and virtual (virt) orbitals.

acronym	Cluster operator	scaling	references
CCSD	$\hat{T} = \hat{T}_1 + \hat{T}_2$	N^6 (occ ² virt ⁴)	[174, 185]
CCSDT	$\hat{T} = \hat{T}_1 + \hat{T}_2 + \hat{T}_3$	N^8 (occ ³ virt ⁵)	[186, 187]
CCSDTQ	$\hat{T} = \hat{T}_1 + \hat{T}_2 + \hat{T}_3 + \hat{T}_4$	N^{10} (occ ⁴ virt ⁶)	[188, 189]
...	[190, 191]

2.1.4.1 Coupled-Cluster Doubles (CCD)

Including only double substitutions into the Cluster operator leads to the CCD method. As only double substituted amplitudes t_{ij}^{ab} have to be determined the energy and amplitude equations are not too complicated. They can be derived either by configurational-space techniques using Slater-Condon rules to evaluate the matrix elements or by algebraic techniques employing second quantization. Especially the algebraic derivation involves a lot of commutator algebra and therefore a third, very common technique can be utilized: the diagrammatic technique. All these techniques are summarized in the literature [47, 182] and the detailed derivation shall be skipped here. The final CCD energy equation contains just the reference energy and one additional term

$$E_{CCD} = E_{ref} + \frac{1}{4} \sum_{efmn} \langle ef \parallel mn \rangle t_{mn}^{ef}. \quad (2.88)$$

The two-electron integral is again written in the antisymmetrized form. The amplitude equation for t_{ij}^{ab} also called t_2 amplitudes is given in Eqn. 2.89.

$$\begin{aligned} \epsilon_{ij}^{ab} t_{ij}^{ab} = & \langle ab \parallel ij \rangle + \frac{1}{2} \sum_{ef} t_{ij}^{ef} \langle ab \parallel ef \rangle + \frac{1}{2} \sum_{mn} t_{mn}^{ab} \langle mn \parallel ij \rangle + P(ij)P(ab) \sum_{me} t_{im}^{ae} \langle mb \parallel ej \rangle \\ & - P(ij) \sum_m f_j^m t_{im}^{ab} + P(ab) \sum_e f_e^b t_{ij}^{ae} + \frac{1}{4} \sum_{mnef} t_{ij}^{ef} t_{mn}^{ab} \langle mn \parallel ef \rangle \\ & - P(ab) \frac{1}{2} \sum_{mnef} t_{ij}^{ae} t_{mn}^{bf} \langle mn \parallel ef \rangle - P(ij) \frac{1}{2} \sum_{mnef} t_{im}^{ab} t_{jn}^{ef} \langle mn \parallel ef \rangle \\ & + P(ij)P(ab) \frac{1}{2} \sum_{mnef} t_{im}^{ae} t_{nj}^{fb} \langle mn \parallel ef \rangle. \end{aligned} \quad (2.89)$$

There $\epsilon_{ij}^{ab} = \epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b$ contains the orbital energies as obtained from Hartree-Fock, $P(pq)$ is a permutation operator that fulfills the following equation

$$P(pq)f(p, q) = f(p, q) - f(q, p), \quad (2.90)$$

and f_q^p denotes one-electron contributions from the Fock matrix. As this equation is non linear it is solved in an iterative procedure. In the first iteration all terms on the right hand side of Eqn. 2.89 that are multiplied with the amplitudes are set to zero. This only leaves the first integral term and the initial $t_2^{(1)}$ amplitudes are obtained as

$$t_{ij}^{ab(1)} = \frac{\langle ab \parallel ij \rangle}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}. \quad (2.91)$$

The initial amplitudes are then used for the next iteration, where all terms from Eqn. 2.89 are included. This way the amplitudes are refined in each iteration until the changes are smaller than a given threshold and the final energy can be evaluated according to Eqn. 2.88.

2.1.4.2 Coupled-Cluster Singles and Doubles (CCSD)

To describe the dynamic correlation in practical systems the CCD method is often not accurate enough. Therefore, one can include also single substitutions leading to the CCSD method. This method has practically the same computational effort as the CCD approach but yields a better description of the electronic structure, as orbital relaxation effects are included by a proper treatment of single substituted determinants. The t_2 amplitude equation (c.f. Eqn. 2.89) is extended by contributions from single amplitudes t_1

$$\begin{aligned}
& P(ij) \sum_e \langle ab \| ej \rangle t_i^e - P(ab) \sum_m \langle mb \| ij \rangle t_m^a - P(ij) \sum_{me} f_e^m t_i^e t_{mj}^{ab} - P(ab) \sum_{me} f_e^m t_m^a t_{nj}^{eb} ij \\
& + P(ij) P(ab) \sum_{mef} \langle am \| ef \rangle t_i^e t_{mj}^{fb} - P(ij) P(ab) \sum_{mne} \langle mn \| ie \rangle t_m^a t_{nj}^{eb} \\
& - \frac{1}{2} P(ab) \sum_{mef} \langle mb \| ef \rangle t_m^a t_{ij}^{ef} + \frac{1}{2} P(ij) \sum_{mne} \langle mn \| ej \rangle t_i^e t_{mn}^{ab} + P(ab) \sum_{mef} \langle ma \| ef \rangle t_e^m t_{ij}^{fb} \\
& - P(ij) \sum_{mne} \langle mn \| ei \rangle t_m^e t_{nj}^{ab} + \sum_{ef} \langle ab \| ef \rangle t_i^e t_j^f + \sum_{mn} \langle mn \| ij \rangle t_m^a t_n^b \\
& - P(ij) P(ab) \sum_{me} \langle mb \| ej \rangle t_i^e t_m^a + \frac{1}{2} \sum_{mnef} \langle mn \| ef \rangle t_i^e t_j^f t_{mn}^{ab} + \frac{1}{2} \sum_{mnef} \langle mn \| ef \rangle t_m^a t_n^b t_{ij}^{ef} \\
& - P(ij) P(ab) \sum_{mnef} \langle mn \| ef \rangle t_i^e t_m^a t_{nj}^{fb} - P(ij) \sum_{mnef} \langle mn \| ef \rangle t_m^e t_i^f t_{nj}^{ab} \\
& - P(ab) \sum_{mnef} \langle mn \| ef \rangle t_m^e t_n^a t_{ij}^{fb} + \sum_{mnef} \langle mn \| ef \rangle t_i^e t_j^f t_m^a t_n^b \\
& + P(ab) \sum_{mef} \langle mb \| ef \rangle t_i^e t_m^a t_j^e + P(ij) \sum_{mne} \langle mn \| ej \rangle t_i^e t_m^a t_n^b
\end{aligned} \tag{2.92}$$

and an additional equation for the determination of the t_1 amplitudes (2.93) has to be solved.

$$\begin{aligned}
\epsilon_i^a t_i^a = & f_i^a + \sum_{me} f_e^m t_{im}^{ae} + \sum_{me} f_e^m t_i^e t_m^a + \frac{1}{2} \sum_{mef} \langle am \| ef \rangle t_{im}^{ef} + \sum_{mef} \langle am \| ef \rangle t_i^e t_m^f \\
& - \frac{1}{2} \sum_{mne} \langle mn \| ie \rangle t_{mn}^{ae} - \sum_{mne} \langle mn \| ie \rangle t_m^a t_n^e + \sum_e f_e^a t_i^e - \sum_m f_i^m t_m^a \\
& + \sum_{me} \langle am \| ie \rangle t_m^e - \frac{1}{2} \sum_{mnef} \langle mn \| ef \rangle t_i^e t_{mn}^{af} - \frac{1}{2} \sum_{mnef} \langle mn \| ef \rangle t_m^a t_{in}^{ef} \\
& + \sum_{mnef} \langle mn \| ef \rangle t_m^e t_{ni}^{fa} - \sum_{mnef} \langle mn \| ef \rangle t_i^e t_m^a t_n^f
\end{aligned} \tag{2.93}$$

Again, the amplitudes are obtained in an iterative solution of the given amplitude equations as already described for the CCD method and the energy can be calculated according to Eqn. 2.83 as

$$E_{CCSD} = E_{ref} + \frac{1}{4} \sum_{efmn} \langle ef \| mn \rangle t_{mn}^{ef} + \sum_{em} f_e^m t_m^e + \frac{1}{2} \sum_{efmn} \langle ef \| mn \rangle t_m^e t_n^f. \tag{2.94}$$

The CCSD method is a very robust approximation and with a sufficiently large basis a good accuracy compared to experimental results can be obtained. Typical errors for the calculation of bond length are in the order of 0.5 pm [192] while reaction enthalpies can be calculated with ≈ 10 kJ/mol deviations [160, 193]. Calculated ^{13}C -NMR chemical shifts typically deviate about 2-5 ppm [162, 194]. But the high computational cost that formally scales with N^6 , where N is the number of basis functions, often forbids the calculation of larger model systems. Therefore, some further approximations are currently developed to overcome this steep scaling behaviour and make CC methods applicable for larger and larger systems.

As already mentioned in the introduction a key component of such approximations is a decomposition of higher dimensional tensors into lower dimensional quantities in one way or another and thereby truncating the number of parameters through a given accuracy threshold. Besides the RI and CD approximations, that can be identified as obvious decomposition techniques, also schemes for defining a new set of virtual and occupied orbitals such as projected atomic orbital PAO [60–62], pair natural orbital PNO [14, 82–90] and optimized virtual space OSV [15, 91–93] methods can be seen as tensor decompositions or factorizations. In these methods the delocalized occupied Hartree-Fock orbitals are replaced by localized molecular orbitals through a unitary transformation. The virtual orbital space is handled differently in the three mentioned methods. While in the PAO method only a single, global set of virtuals is defined, the PNO approximation defines a distinct adapted set of virtuals for each electron pair. Therefore, a considerably smaller number of PNOs than PAOs per pair is needed for the same accuracy. The OSV method defines an adapted virtual set for each localized molecular orbital instead for each pair and therefore has a less compact description of the virtuals per pair but simplifies some problems associated with the PNO integral transformation. [15] Also hybrid schemes combining the advantages of different treatments of the virtual space are known [195] and can be used in the framework of local correlation methods. A pictographic representation of these factorization schemes is shown in Fig. 2.7.

However, these factorization schemes are not always straightforward to extend to arbitrary kind of electronic structure methods or molecular properties. Therefore, in the next section a general scheme for the representation of high dimensional tensors is described.

2.2 Tensor Representation Techniques

From a mathematical point of view a tensor can be seen as the expression of data in higher dimensions or in other words a multi-dimensional array of numerical values. This way a simple number can also be seen as a zero-order tensor, while a vector is a tensor of order one. Two dimensional arrays or matrices are tensors of order two. All higher dimensional arrays are called tensor of order d , where d is the number of dimensions. Here the following nomenclature is introduced: general high order tensors are denoted with capital fraktur letters if the whole tensor is addressed, while matrices are printed as bold letters. Additional sub- or

superscripts to tensors and matrices in parenthesis belong to the name of the tensor and do not denote a special index or position in the tensor. Normal index based tensors are written in latin letters with indices either given in parenthesis or as superscripts.

The normal way of representing a tensor \mathfrak{T} is index based

$$T(i_1, i_2, \dots, i_n) = \text{value}, \quad (2.95)$$

where each value can be addressed by a specific index tuple with $i_n \in \mathbb{N}$. So each value has to be stored even if it is zero and may not be necessary. As a simple example for a higher dimensional tensor a third order tensor T_{ijk} is chosen, but all following discussions also hold for higher dimensional quantities. A simple graphical representation is given in Figure 2.1. The number of elements that have to be stored can be easily calculated as $I \cdot J \cdot K$. Assuming

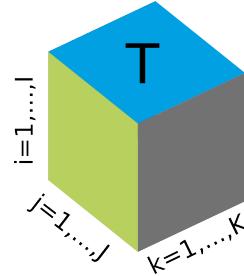


Figure 2.1: Example for a third order tensor $\mathfrak{T} \in \mathbb{R}^{I \times J \times K}$.

that the length in each dimension are the same ($I = J = K = N$) the storage amount can be calculated as N^3 . Increasing the order of the tensor to d leads to an increased number of elements of N^d . This exponential growth is often called the “curse of dimensionality” [58, 196] and makes operations with higher order tensors extremely complex.

2.2.1 Canonical Polyadic Format (CP)

To overcome this steep scaling behaviour for high order tensors they can be represented as other tensor formats using lower dimensional quantities connected by the Kronecker product \otimes . The Kronecker product of two vectors \vec{a} and \vec{b} leads to a matrix \mathbf{C} in the following way:

$$\mathbf{C} = \vec{a} \otimes \vec{b} = \begin{pmatrix} a_1 \vec{b} & a_2 \vec{b} & \dots & a_n \vec{b} \end{pmatrix} = \begin{pmatrix} a_1 b_1 & a_2 b_1 & \dots & a_n b_1 \\ a_1 b_2 & a_2 b_2 & \dots & a_n b_2 \\ \dots & \dots & \ddots & \dots \\ a_1 b_m & a_2 b_m & \dots & a_n b_m \end{pmatrix} \quad (2.96)$$

This concept of building up a higher dimensional quantity from lower dimensional objects can easily be extended to any higher order tensor by simply adding additional terms using the Kronecker product. To make this concept more clear, a small numerical example for a third order tensor is given in Figure 2.2.

$$\begin{pmatrix} 2 \\ 4 \\ 3 \end{pmatrix} \otimes \left[\begin{pmatrix} 1 \\ 3 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 2 \\ 7 \\ 5 \end{pmatrix} \right] = \begin{pmatrix} 2 \\ 4 \\ 3 \end{pmatrix} \otimes \begin{pmatrix} 2 & 6 & 2 \\ 7 & 21 & 7 \\ 5 & 15 & 5 \end{pmatrix} = \begin{pmatrix} 4 & 12 & 4 \\ 14 & 42 & 14 \\ 10 & 30 & 10 \\ 8 & 24 & 8 \\ 28 & 84 & 28 \\ 20 & 60 & 20 \\ 6 & 18 & 6 \\ 21 & 63 & 21 \\ 15 & 45 & 15 \end{pmatrix}$$

Figure 2.2: Numerical example for the description of a third order tensor by application of the Kronecker product.

This leads to a tensor format often called canonical polyadic format (CP format). In the CP format a tensor \mathfrak{T} is expressed a sum of Kronecker product over d so called representing vectors \vec{a}

$$\mathfrak{T} = \sum_{r=1}^R \vec{a}^{(1)} \otimes \vec{a}^{(2)} \otimes \dots \otimes \vec{a}^{(d)} = \sum_{r=1}^R \bigotimes_{\mu=1}^d \vec{a}_r^{(\mu)}. \quad (2.97)$$

where the expansion length R is also called rank. It should be noted, that the superscripts in parenthesis do not denote a position in the vector but are rather used to distinguish representing vectors for different dimensions. Furthermore, the representing vector for the first dimension ($\vec{a}^{(1)}$) does not necessarily have to be the same as the vector for the second dimension ($\vec{a}^{(2)}$). In the following the arrow over the representing vectors will be omitted for convenience.

To achieve such a representation for a given indexed based tensor is not straight forward and decomposition techniques have to be used in order to factorize a tensor into a sum of representing vectors [107–110, 117, 119–121, 197–199]. For example, a third-order tensor $\mathfrak{T} \in \mathbb{R}^{I \times J \times K}$ (c.f. Figure 2.1) can be written as ¹

$$\mathfrak{T} \approx \sum_{r=1}^R a_r^{(i)} \otimes b_r^{(j)} \otimes c_r^{(k)}, \quad (2.98)$$

¹This expansion is exact as long as the rank R is large enough. It has been shown that any tensor in d dimensions can be represented exactly with a rank of $R = N^{d-1}$, where N denotes the number of entries in each dimension.

with rank $R \in \mathbb{N}$, $a_r \in \mathbb{R}^I$, $b_r \in \mathbb{R}^J$ and $c_r \in \mathbb{R}^K$ and a graphical representation of this decomposition is shown in Figure 2.3. The amount of storage is now highly dependent on the

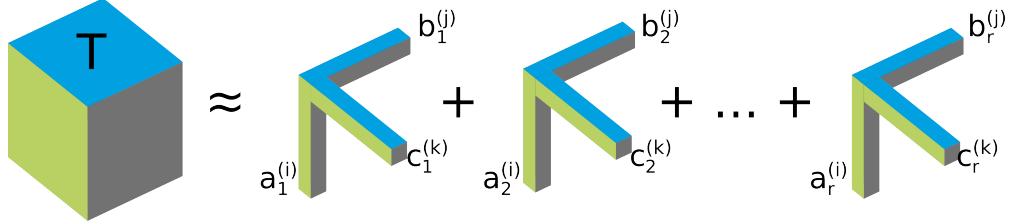


Figure 2.3: CP representation of third order tensor with rank R .

rank of the tensor and scales with $\mathcal{O}(N \cdot R \cdot d)$. As long as the rank is smaller than $\frac{1}{d}N^{d-1}$ the representation in the CP format has a lower complexity than the full indexed based representation of the tensor. If a low-rank approximation can be achieved, where the rank is significantly smaller than the given maximal value, the curse of dimensionality is broken, as for example the scaling of storage is now only linear in the number of dimensions. Also many mathematical operations dealing with high order tensors in a low-rank CP representation become favourable in terms of computational costs, as these are now also linear in the number of dimensions and dependent on the rank of the tensor [117, 119].

One way to obtain a tensor in the CP format is to reformulate the equations that define the tensor in a way, that the structure of the CP format evolves naturally [113, 114, 120, 121, 200]. This of course cannot be done in all cases and there is also no guarantee that the rank is low. An other possibility is to use an extension of the Singular Value Decomposition (SVD) [99] to higher order tensors. However, this is not straightforward and the most common way is to use parallel factor decomposition (PARAFAC) or canonical decomposition (CANDECOMP) [100, 106, 116, 118, 122, 128]. These procedures are very complex and can by no means be used as a “black-box” approximation like the SVD for matrices. Also the rank of the tensor in the CP format may still be large and not optimal so that the tensor cannot be termed a low-rank approximation. Therefore, one can try to minimize the rank by fitting a new tensor \mathfrak{V} in the CP format with a lower rank K than the original tensor \mathfrak{T} up to a given accuracy ε

$$\begin{aligned} \mathfrak{T} &= \sum_{r=1}^R \bigotimes_{\mu=1}^d t_r^{(\mu)} \\ \|\mathfrak{T} - \mathfrak{V}\| \leq \varepsilon \quad \text{with} \quad \mathfrak{V} &= \sum_{k=1}^K \bigotimes_{\mu=1}^d v_k^{(\mu)}. \end{aligned} \tag{2.99}$$

For solving this minimization problem there are different choices like an alternating least square (ALS) scheme [117, 119], a modified Newton method [121, 122] or an accelerated gradient (AG) [145, 146] algorithm. A detailed description of the used AG algorithm is given in section 3.2. Nevertheless, all these algorithms require highly nonlinear computations and the minimization procedure may possess no solution at all, which makes all close by solutions

unstable.

Due to the simple structure of the CP format it can be seen as the most general representation scheme that is able to break the curse of dimensionality. However, the decomposition into this format is often ill-posed and the ranks may be very large. Therefore, also other formats like the Tucker format or the Hierarchical format can be used.

2.2.2 Tucker Format

The Tucker decomposition uses a form of higher order component analysis to decompose a tensor into a core tensor transformed by a matrix along each node [110, 201]. It was first introduced only as a tree-mode factor analysis for tensors of order three [201], but later expanded to d -mode principle component analysis also termed Higher-order Singular Value Decomposition (HOSVD) [128, 142]. The Tucker decomposition of a third-order tensor $\mathfrak{T} \in \mathbb{R}^{I \times J \times K}$ can be written as

$$\mathfrak{T} \approx \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} g_{r_1 r_2 r_3} \cdot a_{r_1}^{(i)} \otimes b_{r_2}^{(j)} \otimes c_{r_3}^{(k)} \quad (2.100)$$

and a graphical representation is given in Figure 2.4. This format can also be extended

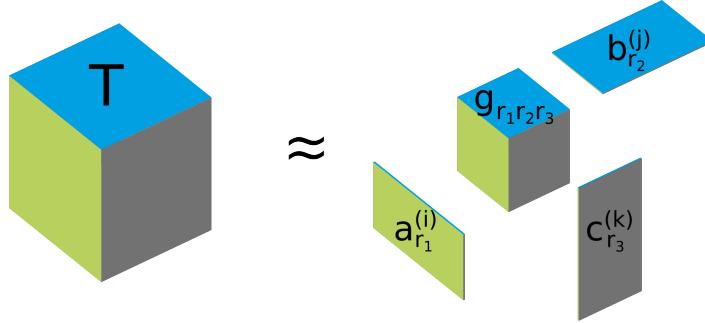


Figure 2.4: Tucker decomposition of third order tensor with multilinear ranks R_1, R_2 and R_3 .

to higher order tensors easily by an increase of the order of the core tensor and additional matrices. For a tensor of order d the Tucker representation can also be written with the help of the Kronecker product

$$\mathfrak{T} \approx \sum_{r_1=1}^{R_1} \dots \sum_{r_d=1}^{R_d} g_{r_1, \dots, r_d} \bigotimes_{\mu=1}^d t_{r_\mu}^{(\mu)} \quad \text{with} \quad g \in \mathbb{R}^{R_1 \times \dots \times R_d}. \quad (2.101)$$

Here it can be seen, that the CP format can be expressed as a special case of the Tucker format where the core tensor g is superdiagonal and all ranks are the same $R_1 = R_2 = \dots = R_d = R$.

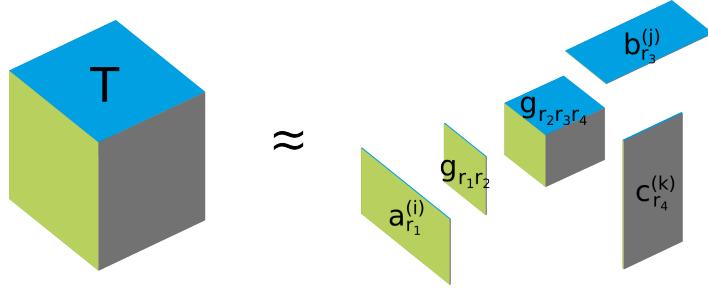


Figure 2.5: Hierarchical tensor representation for third order tensor.

The representation in the Tucker format is able to break the curse of dimensionality only if the size of the core tensor is significantly smaller than the size of the original tensor. Therefore, the multilinear rank has to be smaller than the length of each dimension of the indexed based tensor. Assuming the same length N in all of the d dimensions and a Tucker rank of $R_T = R_1 = R_2 = \dots = R_\mu$ also being constant for each dimension the number of elements to store can be decreased to $R_T^d + d \cdot N \cdot R_T$ compared to the conventional N^d memory requirements. Here, only the first term shows an exponential scaling for an increasing tensor order, while the second term is only linear in the number of dimensions. Therefore, the Tucker representation is mostly used only for tensors of order three. However, the great advantage of the Tucker format is the easy and pure algebraic decomposition algorithm that is only based on higher order SVD [128, 142].

2.2.3 Hierarchical Format

The Hierarchical format or Hierarchical Tucker format (HT) is a further representation technique commonly used for high order tensors. This format can also be achieved by application of SVD onto the matrix form of a higher order tensor [115, 202]. The decomposition of a tensor \mathfrak{T} of order tree is shown in Figure 2.5. A more general graphical representation is given in Figure 2.7 in form of a tensor network for a fourth order tensor. For the third order tensor \mathfrak{T} the first step is the separation of the first dimension from the last two by application of SVD. The remaining submatrix can then again be split up into vectors by another Singular Value Decomposition. The final result can be written as

$$\mathfrak{T} \approx \sum_{r_1}^{R_1} \cdots \sum_{r_4}^{R_4} g_1(r_1, r_2) \cdot g_2(r_2, r_3, r_4) \cdot a_{r_1}^{(i)} \otimes b_{r_3}^{(j)} \otimes c_{r_4}^{(k)}. \quad (2.102)$$

The Hierarchical format can also be extended to higher order tensors where successively more and more SVDs are needed to split the tensor up into subspaces containing the different representing vectors. For a general d order tensor the representation in the Hierarchical

format is given as

$$\mathfrak{T} \approx \sum_{r_1=1}^{R_1} \dots \sum_{r_d=1}^{R_d} \prod_{\nu=1}^P g_{J_\nu} \bigotimes_{\mu=1}^d t_{r_\mu}^{(\mu)} \quad \text{with } J_\nu \subset \{r_1, \dots, r_d\}. \quad (2.103)$$

Here, the rank R_{HT} can be defined as the maximal rank for all summations in Eqn. 2.103 $R_{HT} \geq R_\mu$ for $1 \leq \mu \leq d$ and the amount of storage can be calculated to $N \cdot R_{HT} \cdot d + (d-1) \cdot R_{HT}^3$, which is again linear scaling in the order of the tensor and therefore also breaks the curse of dimensionality.

2.2.4 Tensor Train or Matrix Product State Format

There exist also some other tensor formats that are related to the Hierarchical format like the Tensor Train (TT) format [203, 204], which can be seen as a special case of the Hierarchical decomposition separating each dimension from the remaining dimensions. In physics a linear representation also termed Matrix Product State (MPS) format [205] is commonly used in the field of the density matrix renormalization group (DMRG) method [206–211]. This representation is somehow similar to the HT or TT format and is also obtained by successive application of SVD to the matrix form of high order tensors. For a fourth order tensor \mathfrak{T} the decomposition in the MPS format yields

$$\mathfrak{T} \approx \sum_{r_1=1}^{R_1} \sum_{r_2=1}^{R_2} \sum_{r_3=1}^{R_3} t_{r_1}^{(a)} \otimes t_{r_1 r_2}^{(b)} \otimes t_{r_2 r_3}^{(i)} \otimes t_{r_3}^{(j)}. \quad (2.104)$$

A graphical representation of a third order tensor is given in Figure 2.6. The MPS format

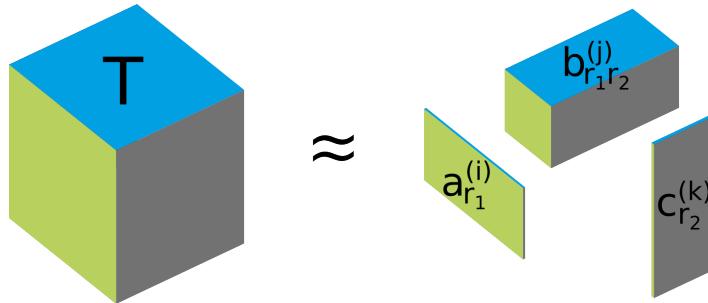


Figure 2.6: Third order tensor in MPS format.

therefore does not separate all dimensions, but keeps a link between them in a linear form. For better comparison of ranks between the representation in the MPS format and other formats an effective rank R_{MPS} is defined. For a tensor of order four this effective rank can be calculated as

$$R_{MPS} = \max(R_1, R_3) \cdot R_2. \quad (2.105)$$

The amount of storage can be calculated as $N \cdot R_{MPS} \cdot 4$, which is again only linear in the order of the tensor and shows a similar scaling as the representation of the same fourth order tensor in the CP format ($N \cdot R_{CP} \cdot 4$).

2.2.5 Comparison of Tensor Formats

To give a short overview over the presented tensor formats a comparison is given in Table 2.2. There, the storage amount and costs for numerical operation dealing with high order tensors

Table 2.2: Comparison of various tensor formats in terms of storage costs, evaluation of a value on a special multiindex i , addition of two tensors \mathfrak{T} and \mathfrak{U} and the calculation of a scalar product between two tensors. It should be noted the ranks in the formats may be different.

	storage	T_i eval.	$\mathfrak{T} + \mathfrak{U}$	$\langle \mathfrak{T}, \mathfrak{U} \rangle$
full	N^d	0	N^d	N^d
CP	$N \cdot R_{CP} \cdot d$	$R_{CP} \cdot d$	0	$N \cdot R_{CP}^2 \cdot d$
MPS	$N \cdot R_{MPS} \cdot d$	$R_{MPS} \cdot d$	$N \cdot R_{MPS}^2 \cdot d$	$N \cdot R_{MPS}^2 \cdot d$
Tucker	$N \cdot R_T \cdot d + R_T^d$	R_T^d	$R_T^{d+1} \cdot d + N \cdot R_T^2 \cdot d$	$R_T^{d+1} \cdot d + N \cdot R_T^2 \cdot d$
HT	$N \cdot R_{HT} \cdot d + R_{HT}^3 \cdot d$	$R_{HT}^3 \cdot d$	$R_{HT}^4 \cdot d + N \cdot R_{HT}^2 \cdot d$	$R_{HT}^4 \cdot d + N \cdot R_{HT}^2 \cdot d$

are considered only showing the scaling behaviour (thus omitting prefactors). It should be noted, that the rank in the different formats can differ significantly for all these formats. In general only the Tucker and the full, canonical, index based tensor format do exhibit an exponential scaling with the number of dimensions, while the CP, the MPS and the HT format depend only linearly on d . Therefore, the later should be suitable for the application in post Hartree-Fock methods, where calculations with high order tensors are required.

A general comparison of various tensor formats including already mentioned physical motivated concepts like RI or the THC format is given in Figure 2.7 using tensor networks. Here an arbitrary fourth order tensor $t(a, b, i, j)$ is taken as an example, which can be seen as a model for the t_2 amplitude tensor in the PAO, PNO, OSV and THC method or a two-electron integral for RI and CD.

As the CP format is the mathematical most general tensor format (factorizing all dimensions from each other and having only one expansion length) this work is mainly focused on representing quantities occurring in high level post HF methods in this format and the reformulation of the subsequent equations due to the new structure of the tensors. In the next section the casting routine, to bring an indexed based quantity into the CP format, and the rank reduction are described for two-electron integral tensors and the wavefunction parameters of post Hartree-Fock methods.

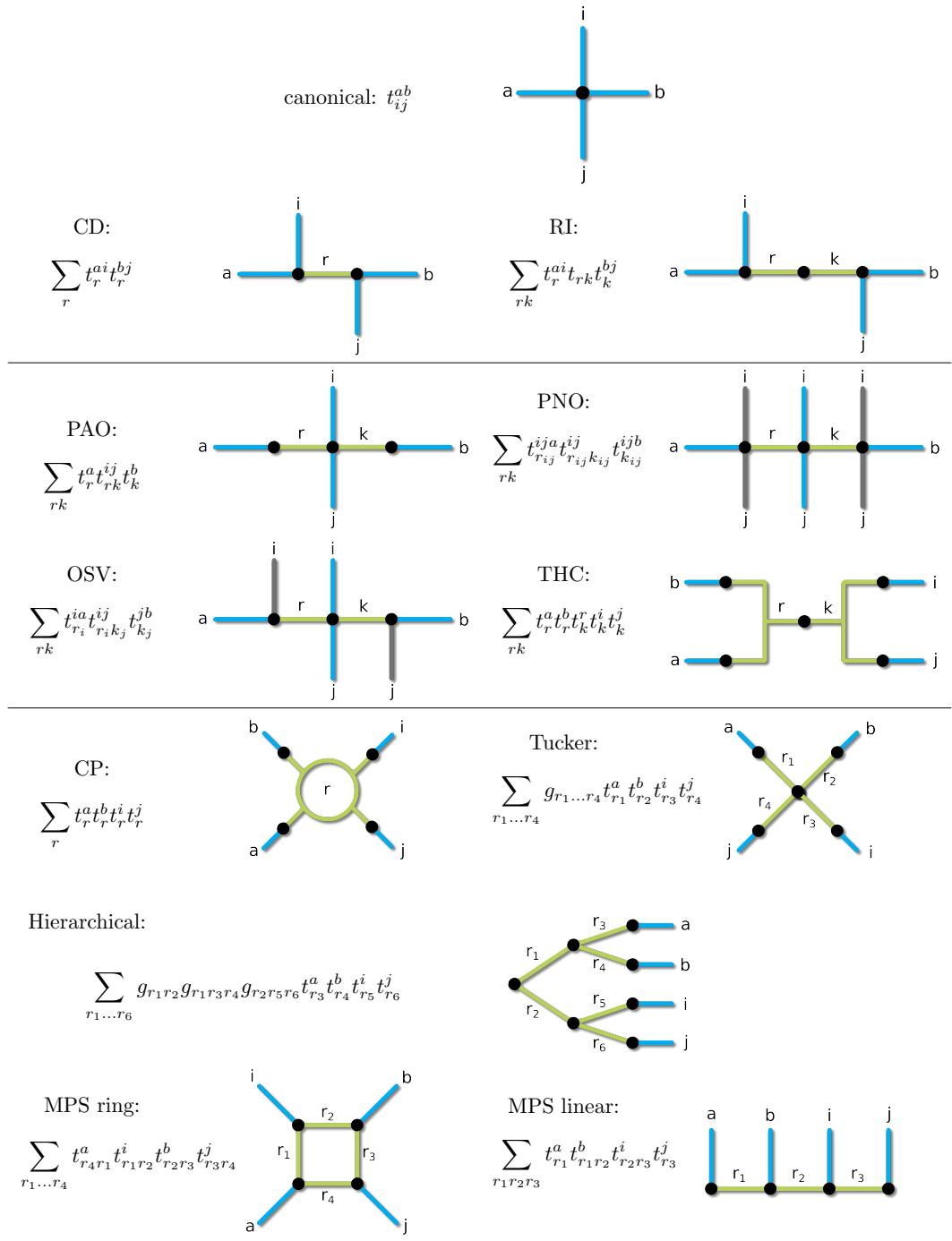


Figure 2.7: Tensor networks using various tensor formats for tensors of order 4 (for example t_2 amplitudes). Blue vertices denote external indices while internal contraction indices are shown in green. It should be noted, that the PNO and OSV approximation are special formats that use specific contractions for different values of i and j . These “auxiliary” indices are denoted in grey. Furthermore, the representation for the CP and THC format are rather schematic (common internal index). The equations are given in index based representation of the representing tensors for better comparison between the formats.

3 Application of the CP Format in post Hartree-Fock Methods

As already seen in the previous section the CP tensor format is capable of breaking the curse of dimensionality if high order tensors can be approximated by a low-rank approximation. In high level post Hartree-Fock methods such as Coupled-Cluster high order tensors are needed to describe the electron correlation properly. With an increasing level of accuracy also the complexity of the equations and the order of the occurring amplitudes is increased. Therefore, the Coupled-Cluster method is chosen as a test for the application of the CP format for higher order tensors.

The CP format can be introduced at different stages: On the one hand one can try to approximate only higher dimensional amplitudes and the fourth order tensor of the two-electron integrals in the MO basis in the CP format and perform all subsequent tensor multiplications (as for example given in Eqn. 2.89) also called tensor contractions in the CP format. This approach should reduce the scaling of the contractions drastically and the resulting quantities are also obtained in the CP format. This way the CP format can be used at all stages of the iterative procedure to obtain the amplitudes. On the other hand the two-electron integrals in the AO basis can be transferred into the CP format. This way the AO-MO transformation has to be reformulated and the integrals in the MO basis are obtained in the CP format as well. As the latter approach is more general and allows to investigate the behaviour for quantities in the AO basis too, this scheme is chosen and will be described in the following section.

The casting procedure to transform index based quantities into the CP format and a procedure for reducing the rank in the CP format are introduced. As the used tensors in electronic structure calculations (two-electron integrals and wavefunction parameters) describe the physical behaviour of the systems, some of these properties can be used to derive a specific casting and rank reduction algorithm for high order tensors occurring in post Hartree Fock methods. The reduction algorithm is the most crucial part, as all subsequent tensor contractions are only beneficial if a low-rank approximation of the initial tensors can be used.

3.1 Casting of Indexed Based Tensors to CP Format

As a first testcase for the application of the CP format for tensors occurring in first principle electronic structure calculations the two-electron integrals in the AO basis are chosen. Here, a way of transforming the tensor from the index based storage to the CP format, in the

following called casting procedure, has to be found and the size and scaling of the achievable ranks with system and basis set size are investigated.

As already mentioned in section 2.2.1 the general way to obtain a representation in the CP format is to reformulate the equations that define the tensor in a way, that a natural separable representation can be achieved. In the case of the two-electron integrals \mathfrak{v}_{AO} (c.f. Eqn. 2.20)

$$\mathfrak{v}_{AO} = \langle \mu\nu | \sigma\rho \rangle = \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} \frac{\phi_\mu^*(\vec{x}_1) \phi_\nu^*(\vec{x}_2) \phi_\sigma(\vec{x}_1) \phi_\rho(\mathbf{r}_2)}{\|\vec{x}_1 - \vec{x}_2\|} d\vec{x}_1 d\vec{x}_2 \quad (3.1)$$

$$= \int_{\mathbb{R}} \int_{\mathbb{R}^3} \int_{\mathbb{R}^3} e^{-t^2 \|\vec{x}_1 - \vec{x}_2\|^2} \phi_\mu^*(\vec{x}_1) \phi_\nu^*(\vec{x}_2) \phi_\sigma(\vec{x}_1) \phi_\rho(\vec{x}_2) \|\vec{x}_1 - \vec{x}_2\| dt d\vec{x}_1 d\vec{x}_2, \quad (3.2)$$

they have to be evaluated as a sum of four first-order tensors by a quadrature. Here, the spatial parts of the atomic orbitals ϕ_μ are expressed by Gaussian functions expanded by spherical harmonics $Y_{l,m}$ and a normalization constant N

$$\phi_\mu = N \cdot Y_{l,m} \cdot e^{-\alpha r^2}. \quad (3.3)$$

Using a standard quadrature for the given two-electron integral would require a new implementation from scratch and the evaluation on various grid points could lead to large ranks. Furthermore, most of the currently used integral routines are highly efficient using recursion schemes and can be used without any adaption leading to tensors stored index based. Therefore, it is desirable to advice a standard casting routine that allows to transform any index based tensor into the CP representation.

3.1.1 Trivial Decomposition

The easiest way is to use a trivial decomposition: For a given index based tensor of order three the multiindex of the first two dimensions is encoded by unit representing vectors of length N , where the one is placed at the current value of the index. All corresponding N values for the third dimension are used to initialize the last representing vector. A more pictographical explanation can be given if one thinks of the third order tensor as a matrix containing vectors for each index pair. In this case the index pair is coded with the help of unit vectors, while the last representing vector contains only the values of the corresponding vector. A small numerical example for this procedure is given in Figure 3.1. Here it can be seen, that the initial rank of such a decomposition is always N^{d-1} , where N is the number of entries in each of the d dimensions. The rank can only be smaller if a whole block in the tensor is zero. In this case the block can be omitted and the rank is decreased. The hole procedure is only a casting and does not need any new calculations or approximations, which makes this procedure very attractive as an initial point for the conversion of a given tensor into the CP format quickly. Nevertheless, for dens tensors the initial rank is very large and has to be reduced. Otherwise calculations with high rank tensors in the CP format, especially

$$\begin{aligned}
 & \left(\begin{pmatrix} 3 & 4 & 2 \\ 6 & 3 & 6 \\ 4 & 7 & 4 \end{pmatrix} \right) \\
 & \left(\begin{pmatrix} 8 & 7 & 4 \\ 2 & 9 & 8 \\ 5 & 5 & 3 \end{pmatrix} \right) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 3 & 4 & 2 \\ 6 & 3 & 6 \\ 4 & 7 & 4 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 8 & 7 & 4 \\ 2 & 9 & 8 \\ 5 & 5 & 3 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 4 & 5 & 5 \\ 8 & 3 & 4 \\ 5 & 5 & 4 \end{pmatrix} \\
 & \left(\begin{pmatrix} 4 & 5 & 5 \\ 8 & 3 & 4 \\ 5 & 5 & 4 \end{pmatrix} \right) \\
 & \left(\begin{pmatrix} 3 & 4 & 2 \\ 6 & 3 & 6 \\ 4 & 7 & 4 \end{pmatrix} \right) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 3 \\ 6 \\ 4 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 4 \\ 3 \\ 7 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 2 \\ 6 \\ 4 \end{pmatrix} \\
 & \left(\begin{pmatrix} 8 & 7 & 4 \\ 2 & 9 & 8 \\ 5 & 5 & 3 \end{pmatrix} \right) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 8 \\ 2 \\ 5 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 7 \\ 9 \\ 5 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 4 \\ 8 \\ 3 \end{pmatrix} \\
 & \left(\begin{pmatrix} 4 & 5 & 5 \\ 8 & 3 & 4 \\ 5 & 5 & 4 \end{pmatrix} \right) = \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 4 \\ 8 \\ 5 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 5 \\ 3 \\ 5 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 5 \\ 4 \\ 4 \end{pmatrix} \\
 \\
 & \left(\begin{pmatrix} 3 & 4 & 2 \\ 6 & 3 & 6 \\ 4 & 7 & 4 \end{pmatrix} \right) \quad \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 3 \\ 6 \\ 4 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 4 \\ 3 \\ 7 \end{pmatrix} + \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 2 \\ 6 \\ 4 \end{pmatrix} + \\
 & \left(\begin{pmatrix} 8 & 7 & 4 \\ 2 & 9 & 8 \\ 5 & 5 & 3 \end{pmatrix} \right) = \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 8 \\ 2 \\ 5 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 7 \\ 9 \\ 5 \end{pmatrix} + \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 4 \\ 8 \\ 3 \end{pmatrix} + \\
 & \left(\begin{pmatrix} 4 & 5 & 5 \\ 8 & 3 & 4 \\ 5 & 5 & 4 \end{pmatrix} \right) \quad \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 4 \\ 8 \\ 5 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 5 \\ 3 \\ 5 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 5 \\ 4 \\ 4 \end{pmatrix}
 \end{aligned}$$

Figure 3.1: Trivial casting of third order tensor into CP format.

in an iterative procedure, would even increase the computational effort as already pointed out before. Therefore, a scheme using a Singular Value Decomposition is derived to transform any index based tensor into the CP format.

3.1.2 Decomposition using Singular Value Decomposition

One way to avoid the high initial ranks from the trivial decomposition is to apply a Singular Value Decomposition on a submatrix of the full high order tensor and use small eigenvalues to omit some of the eigenvectors used as representing vectors. For an order d tensor the indices for the first $d - 2$ dimensions are again encoded as unit vectors. The last two dimensions are used to define a submatrix $B(d - 1, d)$ for the special multiindex $\{0, 1, \dots, d - 2\}$ which can be decomposed by a standard SVD into the eigenvalues Σ and eigenvectors \mathbf{U} and \mathbf{V}^T respectively

$$\mathbf{B} = \mathbf{U} \cdot \Sigma \cdot \mathbf{V}^T. \quad (3.4)$$

The original size of the eigenvector matrices $N \times N$ can be truncated by skipping eigenvalues that are smaller than a given threshold value ε (c.f. Figure 3.2). The rows and columns of the

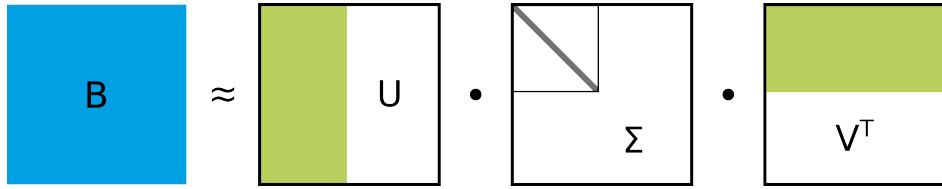


Figure 3.2: Graphical representation of the decomposition by SVD.

reduced eigenvector matrices are used to set up the last two representing vectors. This way the initial rank depends on the SVD and the given threshold parameter but in general should be $N^{d-2} \leq R \leq N^{d-1}$. To give a numerical example the third order tensor from Figure 3.1 is used but is now shown as a vector of matrices for convenience (see Figure 3.3). It can be seen that only the eigenvalue for the second submatrix can be truncated according to the given accuracy parameter of 0.1 and therefore the rank for this example can only be decreased from 9 to 8. However, for larger examples (N larger) the savings can be better, depending on the degree of near linear dependencies in the corresponding submatrices. This procedure can also be used for higher order tensors, where the last 2 dimensions define the submatrices that are decomposed via SVD. The multiindex of the remaining dimensions is encoded with a unit vector. Nevertheless, it should be mentioned that this procedure is not unique. One can also try to resort the original tensor and use other submatrices, that may have better sparsity. Furthermore, this method is not able to exploit linear dependencies in higher dimensions, as only submatrices are used for the SVD. Therefore, the obtained representation in the CP format is no real low-rank approximation but the rank should be lower than the N^{d-1} scaling obtained for the trivial decomposition.

Rather than trying to find a decomposition of an index based tensor into a low-rank CP representation one can also use the trivial or SVD based decomposition leading to a representation in the CP format with large initial ranks and afterwards try to fit a new tensor in the CP format with a lower rank. This procedure, in the following called rank reduction is the most crucial step for tensors in CP format and will be described in section 3.2.

A further possibility for casting the two electron integrals from the index based storage into the CP format arises from using the RI approximation and is discussed in the following section.

$$\begin{aligned}
 & \left(\begin{pmatrix} 3 & 4 & 2 \\ 6 & 3 & 6 \\ 4 & 7 & 4 \end{pmatrix} \right) \\
 & SVD \left(\begin{pmatrix} 3 & 4 & 2 \\ 6 & 3 & 6 \\ 4 & 7 & 4 \end{pmatrix} \right) = \left(\begin{matrix} -0.395 & 0.289 & -0.872 \\ -0.643 & -0.765 & 0.037 \\ -0.657 & 0.575 & 0.488 \end{matrix} \right) \left(\begin{matrix} 13.316 & 0.000 & 0.000 \\ 0.000 & 3.646 & 0.000 \\ 0.000 & 0.618 & 0.000 \end{matrix} \right) \left(\begin{matrix} -0.576 & -0.609 & -0.546 \\ -0.391 & 0.791 & -0.470 \\ -0.718 & 0.057 & 0.693 \end{matrix} \right) \\
 & \left(\begin{pmatrix} 8 & 7 & 4 \\ 2 & 9 & 8 \\ 5 & 5 & 3 \end{pmatrix} \right) \Rightarrow SVD \left(\begin{pmatrix} 8 & 7 & 4 \\ 2 & 9 & 8 \\ 5 & 5 & 3 \end{pmatrix} \right) = \left(\begin{matrix} -0.622 & 0.587 & -0.518 \\ -0.655 & -0.752 & -0.066 \\ -0.429 & 0.299 & 0.853 \end{matrix} \right) \left(\begin{matrix} 17.492 & 0.000 & 0.000 \\ 0.000 & 5.569 & 0.000 \\ 0.000 & 0.062 & 0.000 \end{matrix} \right) \left(\begin{matrix} -0.482 & -0.709 & -0.515 \\ 0.841 & -0.210 & -0.498 \\ -0.245 & 0.674 & -0.697 \end{matrix} \right) \\
 & \left(\begin{pmatrix} 4 & 5 & 5 \\ 8 & 3 & 4 \\ 5 & 5 & 4 \end{pmatrix} \right) SVD \left(\begin{pmatrix} 4 & 5 & 5 \\ 8 & 3 & 4 \\ 5 & 5 & 4 \end{pmatrix} \right) = \left(\begin{matrix} -0.544 & 0.591 & -0.596 \\ -0.628 & -0.757 & -0.177 \\ -0.556 & 0.278 & 0.783 \end{matrix} \right) \left(\begin{matrix} 14.486 & 0.000 & 0.000 \\ 0.000 & 3.269 & 0.000 \\ 0.000 & 0.697 & 0.000 \end{matrix} \right) \left(\begin{matrix} -0.689 & -0.510 & -0.515 \\ -0.706 & 0.634 & 0.317 \\ 0.165 & 0.582 & -0.797 \end{matrix} \right) \\
 & SVD \left(\begin{pmatrix} 3 & 4 & 2 \\ 6 & 3 & 6 \\ 4 & 7 & 4 \end{pmatrix} \right) \Rightarrow \left(\begin{pmatrix} 1 & \\ 0 & \\ 0 & \end{pmatrix} \otimes \begin{pmatrix} -5.260 \\ -8.562 \\ -8.749 \end{pmatrix} \otimes \begin{pmatrix} -0.576 \\ -0.609 \\ -0.546 \end{pmatrix} \right) + \left(\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 1.054 \\ -2.789 \\ 2.096 \end{pmatrix} \otimes \begin{pmatrix} -0.391 \\ 0.791 \\ -0.470 \end{pmatrix} \right) + \left(\begin{pmatrix} 1 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} -0.539 \\ 0.023 \\ 0.302 \end{pmatrix} \otimes \begin{pmatrix} -0.718 \\ 0.057 \\ 0.693 \end{pmatrix} \right) \left(\begin{matrix} 3.005 & 4.006 & 2.003 \\ 6.006 & 3.009 & 6.002 \\ 4.003 & 7.003 & 4.001 \end{matrix} \right) \\
 & SVD \left(\begin{pmatrix} 8 & 7 & 4 \\ 2 & 9 & 8 \\ 5 & 5 & 3 \end{pmatrix} \right) \Rightarrow \left(\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} -10.880 \\ -11.457 \\ -7.504 \end{pmatrix} \otimes \begin{pmatrix} -0.482 \\ -0.709 \\ -0.515 \end{pmatrix} \right) + \left(\begin{pmatrix} 0 \\ 1 \\ 0 \end{pmatrix} \otimes \begin{pmatrix} 3.269 \\ -4.188 \\ 1.665 \end{pmatrix} \otimes \begin{pmatrix} 0.841 \\ -0.210 \\ -0.498 \end{pmatrix} \right) \left(\begin{matrix} 7.993 & 7.027 & 3.975 \\ 2.000 & 9.002 & 7.986 \\ 5.017 & 4.971 & 3.035 \end{matrix} \right) \\
 & SVD \left(\begin{pmatrix} 4 & 5 & 5 \\ 8 & 3 & 4 \\ 5 & 5 & 4 \end{pmatrix} \right) \Rightarrow \left(\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} -7.880 \\ -9.097 \\ -8.054 \end{pmatrix} \otimes \begin{pmatrix} -0.689 \\ -0.510 \\ -0.515 \end{pmatrix} \right) + \left(\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} 1.932 \\ -2.475 \\ -0.510 \end{pmatrix} \otimes \begin{pmatrix} -0.706 \\ 0.634 \\ 0.909 \end{pmatrix} \right) + \left(\begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \otimes \begin{pmatrix} -0.415 \\ -0.123 \\ 0.546 \end{pmatrix} \otimes \begin{pmatrix} 0.165 \\ 0.582 \\ -0.797 \end{pmatrix} \right) \left(\begin{matrix} 3.997 & 5.002 & 5.001 \\ 7.995 & 2.999 & 3.998 \\ 4.998 & 5.002 & 4.001 \end{matrix} \right)
 \end{aligned}$$

Figure 3.3: Numerical example for decomposition with the help of SVD. Assuming a threshold of 0.1 all smaller singular values can be omitted (marked in red in the upper part). The lower part shows the resulting representing vectors in the CP format. The rank of this representation is 8 compared to a rank of 9 for the trivial decomposition. The final reassembled tensor (bottom right) shows that all entries can be approximate within the given accuracy of 0.1. It should be noted, that the first and the last submatrix are represented exactly. The errors are numerical errors due to rounding to third digit.

3.1.3 Casting Two-Electron Integrals using Resolution of the Identity (RI)

For the special case of casting the two-electron integrals in the atomic orbital basis into the CP format one can also use the Resolution of the Identity (RI) [16, 19, 27, 212–215] to get a representation with low rank. The RI approximation is mainly used in Density Functional Theory (DFT) as it allows a very efficient evaluation of two-electron integrals. This way modern codes recalculate the integrals each time needed so that also large systems can be calculated without storage problems. Therefore, sufficiently large fitting bases P and Q

$$\sum_{PQ} |P\rangle M_{PQ}^{-1} \langle Q| \approx 1 \quad \text{with} \quad M_{PQ} = \langle P|Q\rangle = \int \frac{P(\mathbf{x}_1)Q(\vec{\mathbf{x}}_2)}{r_{12}} d\vec{\mathbf{x}}_1 d\vec{\mathbf{x}}_2 \quad (3.5)$$

are introduced in the description of the two-electron integrals (c.f. Eqn. 2.20)

$$\langle \mu\sigma|\nu\rho \rangle \approx \sum_{PQ} \langle \mu\sigma|P\rangle M_{PQ}^{-1} \langle Q|\nu\rho \rangle. \quad (3.6)$$

Eqn. 3.6 can be also be formulated using the general tensor notation

$$\langle \mu\sigma|P\rangle = \mathfrak{R}_{\mu\sigma P} \quad (3.7)$$

$$M_{PQ} = \mathfrak{M} \quad (3.8)$$

$$\langle Q|\nu\rho \rangle = \mathfrak{R}_{Q\nu\rho} \quad (3.9)$$

as

$$\mathfrak{v}_{AO} = \mathfrak{R}_{\mu\sigma P} \mathfrak{M}^{-1} \mathfrak{R}_{Q\nu\rho}, \quad (3.10)$$

assuming summation over all internal indices. Now only three centre integrals instead of four centre integrals are needed together with the so called metric matrix M_{PQ} . These quantities can be calculated and also stored very easily as the highest tensor order is three.

The auxiliary basis has been numerically fitted to exact results and with common basis sets [216–218] the error introduced by the RI approximation is in the range of 0.5 mHartree for standard DFT calculations [20, 219].

To use the RI approximation for casting the two-electron integrals in the CP format the tree centre integrals and also the metric matrix have to be cast in the CP format. This can be done either by the trivial decomposition or by using the SVD as described before. The metric matrix can be decomposed by SVD in a very convenient way allowing for a truncation of small singular values. This leads to a description of all needed quantities in the CP format:

$$\mathfrak{R}_{\mu\sigma P} \approx \sum_{r_1}^{R_1} a_{r_1}^{(\mu)} \otimes a_{r_1}^{(\sigma)} \otimes a_{r_1}^{(P)} \quad \mathfrak{M}^{-1} \approx \sum_{r_2}^{R_2} m_{r_2}^{(P)} \otimes m_{r_2}^{(Q)} \quad \mathfrak{R}_{Q\nu\rho} \approx \sum_{r_3}^{R_3} b_{r_3}^{(Q)} \otimes b_{r_3}^{(\nu)} \otimes b_{r_3}^{(\rho)} \quad (3.11)$$

Inserting the CP tensors in the definition of the full two-electron integrals (Eqn. 3.10) leads to

$$\mathfrak{v}_{AO} \approx \sum_{PQ} \left(\sum_{r_1}^{R_1} a_{r_1}^{(\mu)} \otimes a_{r_1}^{(\sigma)} \otimes a_{r_1}^{(P)} \right) \left(\sum_{r_2}^{R_2} m_{r_2}^{(P)} \otimes m_{r_2}^{(Q)} \right) \left(\sum_{r_3}^{R_3} b_{r_3}^{(Q)} \otimes b_{r_3}^{(\nu)} \otimes b_{r_3}^{(\rho)} \right). \quad (3.12)$$

As the different representing vectors stand for the separated dimensions the summation over P and Q can be written in front of the remaining representing vectors

$$\mathfrak{v}_{AO} \approx \sum_{r_1}^{R_1} \sum_{r_2}^{R_2} \sum_{r_3}^{R_3} \left(\sum_P m_{r_2}^{(P)} a_{r_1}^{(P)} \right) \left(\sum_Q m_{r_2}^{(Q)} b_{r_3}^{(Q)} \right) a_{r_1}^{(\mu)} \otimes a_{r_1}^{(\sigma)} \otimes b_{r_3}^{(\nu)} \otimes b_{r_3}^{(\rho)}, \quad (3.13)$$

so that the last four representing vectors are independent from the expansion length R_2 . The summation over the auxiliary basis P and Q can be carried out leading to simple scalar products over the representing vectors

$$\mathfrak{v}_{AO} \approx \sum_{r_1}^{R_1} \sum_{r_2}^{R_2} \sum_{r_3}^{R_3} \langle m_{r_2}^{(P)}, a_{r_1}^{(P)} \rangle \langle m_{r_2}^{(Q)}, b_{r_3}^{(Q)} \rangle a_{r_1}^{(\mu)} \otimes a_{r_1}^{(\sigma)} \otimes b_{r_3}^{(\nu)} \otimes b_{r_3}^{(\rho)}, \quad (3.14)$$

that can be calculated and represented as coefficient matrices $\mathcal{P}(r_2, r_1)$ and $\mathcal{Q}(r_2, r_3)$

$$\mathfrak{v}_{AO} \approx \sum_{r_1}^{R_1} \sum_{r_2}^{R_2} \sum_{r_3}^{R_3} \mathcal{P} \cdot \mathcal{Q} \cdot a_{r_1}^{(\mu)} \otimes a_{r_1}^{(\sigma)} \otimes b_{r_3}^{(\nu)} \otimes b_{r_3}^{(\rho)}. \quad (3.15)$$

Here it can be seen, that only the two remaining coefficient matrices depend on the expansion length R_2 so that the summation over r_2 can be carried out as matrix matrix multiplication of \mathcal{P} and \mathcal{Q} leading to a single coefficient matrix $\mathcal{K}(r_1, r_3)$

$$\mathfrak{v}_{AO} \approx \sum_{r_1}^{R_1} \sum_{r_3}^{R_3} \mathcal{K} \cdot a_{r_1}^{(\mu)} \otimes a_{r_1}^{(\sigma)} \otimes b_{r_3}^{(\nu)} \otimes b_{r_3}^{(\rho)}. \quad (3.16)$$

The coefficient matrix connects the expansion length of the first two vectors with the expansion length of the last representing vectors and the structure can also be seen as a further tensor format combining features from the Tucker format with the CP format. This tensor format originally proposed by Auer *et al.* [220] is also used by Hohenstein *et al.* [105] and Shenvi *et al.* [221] and termed tensor hypercontraction format (THC). However, it is also possible to combine the two ranks R_1 and R_2 to a new rank R by multiplication, so that a representation in the common CP format is obtained

$$\langle \mu\sigma | \nu\rho \rangle = \mathfrak{v}_{AO} \approx \sum_r^{R=R_1 \cdot R_3} v_r^{(\mu)} \otimes v_r^{(\sigma)} \otimes v_r^{(\nu)} \otimes v_r^{(\rho)}. \quad (3.17)$$

Here the new representing vectors v_r are just copied for the respective values of r_1 and r_2 while the entries of the coefficient matrix are multiplied into one of the representing vector. It

should be noted, that the rank R as the product of the two ranks of the decomposed RI tensors has to be sufficiently small in order to get a low-rank approximation. Actual numerical tests for water using the def2-SVP basis set (24 basis functions, 71 auxiliary functions) show, that the third order RI-tensor $\mathfrak{R}_{\mu\sigma P}$ has an initial rank of 1704 after trivial decomposition. This can be reduced by the reduction algorithm (see 3.2) to 243 using an accuracy parameter of 10^{-4} leading to an initial rank after tensor contraction of 59049. Compared to the initial rank after trivial decomposition of the full two-electron integral tensor of 13824 the initial rank obtained from the RI based procedure is approximately 4 times larger.

Therefore, it is necessary to compress the high initial ranks after the casting routines by fitting a new tensor in the CP format with lower rank up to a certain accuracy. This rank reduction procedure, as described in the following section, is a mathematical procedure allowing the truncation of ranks for arbitrary tensors of arbitrary tensor order, not knowing anything about the structure or possible sparsity of the tensor. Therefore, some adaptions of the general scheme for quantities in electronic structure calculations are introduced to further improve the fitting procedure (see Section 3.9).

3.2 Rank Reduction Algorithm

To obtain a low-rank approximation from a tensor \mathfrak{T} in the CP format M. Espig developed an approximation algorithm in his dissertation. [121, 222] This algorithm fits a new tensor \mathfrak{V} up to a given accuracy ε in the ℓ^2 -norm

$$\begin{aligned} J(\mathfrak{V}) := \|\mathfrak{T} - \mathfrak{V}\| &\leq \varepsilon \quad \text{with} \quad \mathfrak{T} = \sum_{r=1}^R \bigotimes_{\mu=1}^d t_r^{(\mu)} \\ &\quad \mathfrak{V} = \sum_{k=1}^K \bigotimes_{\mu=1}^d v_k^{(\mu)}, \end{aligned} \tag{3.18}$$

where the new approximated tensor has a reduced rank K holding $K \ll R$. The algorithm is designed as a “black box” method that works for general tensors under certain conditions. The best performance is obtained for tensors in high dimensions ($d > 20$), large number of entries per dimension ($n > 5000$) and a small initial rank compared to the number of entries per dimension ($R < 500$). Recently further variants of the minimization procedure using an alternating least square (ALS) scheme [117, 119], a modified Newton method [121, 122] or an accelerated gradient (AG) [145, 146] algorithm have been established. In this work the AG algorithm (see Fig. 3.4) is used, which has certain advantages over the other methods, such as a better convergence and a complexity comparable to the ALS method. A detailed mathematical description of the algorithm is skipped here but an extensive analysis can be found in references 121 and 145. The crucial part of the AG algorithm is the computation of the exact line search parameter $\alpha_k \in \mathbb{R}_{\geq 0}$. Given a direction \mathfrak{D}^k , a solution of the one-dimensional nonlinear equation

$$p(\alpha_k) = \left\langle \nabla J(\mathfrak{V}^k + \alpha \mathfrak{D}^k), \mathfrak{D}^k \right\rangle = 0.$$

Accelerated Gradient (AG) Method

- 1: Choose initial \mathfrak{V}^0 and parameter $\varepsilon \in \mathbb{R}_{>0}$. Define new rank iterator $k := 0$, compute the gradient $\mathfrak{G}^0 := \nabla J(\mathfrak{V}^0)$ and $\mathfrak{D}^0 := -\mathfrak{G}^0$.
 - 2: **while** the gradient $\|\mathfrak{G}^k\| > \varepsilon$ **do**
 - 3: Compute the smallest root $\alpha_k \in [0, 1]$ of the polynomial
 $p(\alpha) := \langle \nabla J(\mathfrak{V}^k + \alpha \mathfrak{D}^k), \mathfrak{D}^k \rangle$, i.e.
 $\alpha_k := \min \{ \alpha \in \mathbb{R}_{\geq 0} : p(\alpha) := \langle \nabla J(\mathfrak{V}^k + \alpha \mathfrak{D}^k), \mathfrak{D}^k \rangle = 0 \}$.
 - 4: Update the representation system of \mathfrak{V} , i.e. $\mathfrak{V}^{k+1} := \mathfrak{V}^k + \alpha_k \mathfrak{D}^k$.
 - 5: Compute the gradient for the updated system, i.e. $\mathfrak{G}^{k+1} := \nabla J(\mathfrak{V}^{k+1})$.
 - 6: Compute $\beta_k := \frac{\langle \mathfrak{G}^{k+1} - \mathfrak{G}^k, \mathfrak{G}^{k+1} \rangle}{\|\mathfrak{G}^k\|^2}$, $\gamma_k := \max\{0, \beta_k\}$.
 - 7: Update the new search direction, i.e. $\mathfrak{D}^{k+1} := -\mathfrak{G}^{k+1} + \gamma_k \mathfrak{D}^k$.
 - 8: $k \mapsto k + 1$.
 - 9: **end while**
-

Figure 3.4: Schematic representation for the Accelerated Gradient (AG) method [121, 145, 146].

has to be found. The complexity of the computation of the gradient \mathfrak{G} is $\mathcal{O}(d \cdot K \cdot n \cdot (K + R))$, where d is the dimension of the tensor. Since the most expensive part in the AG method is the calculation of the gradient, the overall complexity of the AG method is

$$k_{\max} \mathcal{O}(d \cdot K \cdot n \cdot (K + R)), \quad (3.19)$$

where k_{\max} denotes the maximal number of iterations that is equivalent to the final rank of the approximated quantity \mathfrak{V} in the minimization procedure.

The native implementation [121, 122, 145, 222] of the rank-reduction scheme shown in Fig. 3.4 is organized as follows: In a first step a guess for a good low-rank approximation \mathfrak{V} can be used to initialize \mathfrak{V}^0 . This can be very helpful especially during an iterative algorithm where an updated tensor with increased rank can be reduced using an initial guess from the last iteration. If no guess is available the algorithm uses an empty tensor and approximates the tensor from scratch. Then the optimization loop starts and successively adds rank one approximations (a new set of representation vectors) in order to minimize the difference between the original tensor and the approximate low-rank representation. To define the rank one approximation a pivoting routine finds a large entry in the residual tensor $(\mathfrak{T} - \mathfrak{V})$ on a special cross over all dimensions. In a next step this new rank one approximation is optimized in order to minimize the error in the $(\mathfrak{T} - \mathfrak{V})$ residual tensor and is added to the set of ranks in \mathfrak{V} . An iterative procedure is then used to improve all representing vectors of the update \mathfrak{V} to further lower the difference between \mathfrak{T} and \mathfrak{V} . If this difference is lower than the given threshold parameter ε , the procedure is stopped and the final number of ranks has been obtained. Otherwise, the cycle starts again with the pivoting routine and adds another

rank to the approximated \mathfrak{V} . This way, the rank grows one by one during the iterations until the given accuracy is reached.

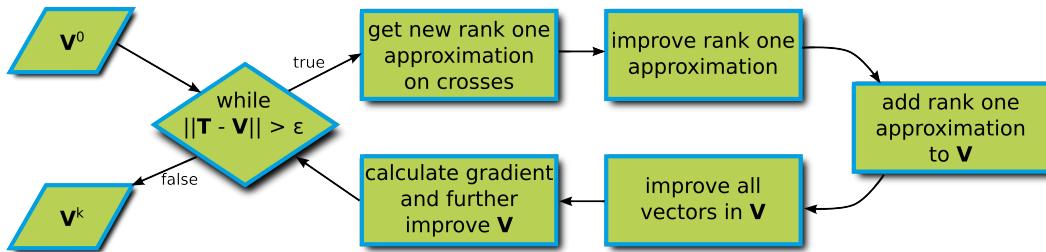


Figure 3.5: Scheme for reduction algorithm.

A very simple and general flow chart with the most important parts is given in Figure 3.5. It should be pointed out that this scheme has not been modified in any way in order to optimize it for the problem of approximating two-electron integrals or wavefunction parameters. So the AG algorithm is used as a “black box” method with only one accuracy parameter to reduce the rank of an arbitrary tensor. As a first test case a simple two dimensional matrix in form of a standard test picture is chosen. Later on also the two-electron integrals in the AO basis for a small test molecule are taken as an example from quantum chemistry.

3.2.1 Rank Reduction for a Test-Matrix

To show how the rank reduction algorithm proceeds a 512×512 grey scale picture of Lena Söderberg (see Figure 3.6), that is quite often used in the image processing community



Figure 3.6: Original 512×512 pixel grey scale picture of Lena Söderberg. [223] The grey values are used to define a second order tensor, that is cast in the CP format.



Figure 3.7: Pictures of Lena Söderberg from approximate tensors with increasing ranks. The ranks and corresponding errors in the ℓ^2 -norm are given as titles of the pictures.

[223–225], is taken as a test case. The picture is converted to a second order tensor with $N = 512$ using the integer grey values for each pixel as entries in the tensor. Casting in the CP format is done by trivial decomposition leading to an exact representation with an initial rank of 512. Now the rank reduction algorithm is used to approximate a low-rank representation with a given accuracy ($\epsilon = 1 \cdot 10^{-2}$) using the AG algorithm as obtained from the author. [121, 145, 222] To get an impression of how the algorithm is working all low-rank approximations occurring during the optimization procedure are saved and can be used to reassemble the full tensor that can be represented as an approximated picture. This way a series of approximations with increasing rank and accuracy is obtained and some representations are shown in Figure 3.7. It can be seen, that in the simple rank one approximation the algorithm first fits large (white) or small (black) values on special crosses of the picture. This

way the coarse structure of the tensor can be represented and square groups of pixels get a very rough approximation of the original values, due to the used approximation on special crosses with large values to define the initial rank one approximation. Increasing the rank to 10 the structure of the picture is represented better but most of the details are still blurred under the coarse approximation. There are still some crosses left in the picture that deviate more from the original than the rest of the picture, so that the approximation looks streaked. With a rank 35 approximation most of the strips are gone and also some details of the feather on the hat become visible. However, there are some stains especially in the uniform regions of the background. These artefacts diminish for the rank 50 and rank 75 approximation. Increasing the rank to 100 almost all features of the original picture can be approximated. There are still some blurred regions in the background where the deviations are higher, but the general approximation is already very good. The last picture in Figure 3.7 is the final result of the AG algorithm with an accuracy of $\varepsilon = 1 \cdot 10^{-2}$. No real differences compared to the original picture (c.f. Figure 3.6) can be seen with the bare eye, but from a numerical point of view, there is still an error in the individual grey values.

However, using the approximation with $r = 100$ the remaining deviation in the ℓ^2 -norm is approximately $3.2 \cdot 10^{-2}$ and the match in comparisons to the original is already very good. In the rank 100 representation $512 \cdot 100 \cdot 2 = 102400$ elements have to be stored. Compared to the full storage with $512 \cdot 512 = 262144$ elements a compression factor of ≈ 2.6 can be achieved for this simple test case while the accuracy is still good. Nevertheless, it is possible to perform manipulations of the data also in the compressed or reduced CP format. It should also be mentioned, that larger compressions can be expected for higher order tensors, due to more redundancy and near linear dependencies in the higher dimensional data.

Unfortunately, the reduction algorithm in its original implementation is not very fast and currently does not run in parallel. So the reduction of the 512×512 second order tensor with $\varepsilon = 1 \cdot 10^{-2}$ takes 588 seconds on a single core of a modern workstation computer (for details on the specifications see appendix A.1). Nevertheless, the scaling of the algorithm for higher dimensions is only linear in the number of dimensions (c.f. Eqn. 3.19). As this is only an example in two dimensions also a pure Singular Value Decomposition can be performed in order to achieve a full decomposition. This procedure takes approximately 1 second on the same system. However, this is only possible for two dimensional examples and furthermore for the given example there are only 5 eigenvalues that are smaller than the given threshold of $1 \cdot 10^{-2}$ so that for this case no real compression by SVD is possible. Despite the long reduction times also a higher order tensor from quantum chemistry calculations is used as a further example for the rank reduction.

3.3 Adaptation of Reduction Algorithm for Quantum Chemistry

3.3.1 First Examples: Water using STO-3G and 6-31G Basis Set

As a first example for high order tensors occurring in post Hartree-Fock methods the two-electron integrals ($d = 4$) for water in the AO basis using the STO-3G basis set (7 basis functions: $N = 7$) are decomposed into the CP format via the trivial decomposition. For a better visualization the structure of the original tensor is shown in figure Figure 3.8 by matrification (indices $\mu\sigma$ and $\nu\rho$ are represented as a single index on the x and the y axis respectively).

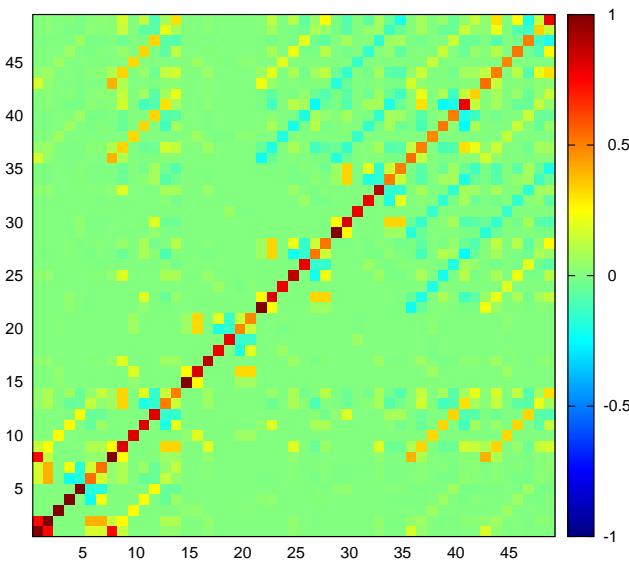


Figure 3.8: Two-electron integrals of water in AO basis for the STO-3G basis set. The indices $\mu\sigma$ and $\nu\rho$ are represented as a single index on the x and the y axis respectively. Each square denotes one integral value where red and blue squares stand for large positive or negative values, while green denotes values close to zero.

The two-electron integrals $\langle \mu\sigma | \nu\rho \rangle$ are obtained from a modified local version of the CFOUR program package [226, 227] with an accuracy of 10^{-12} , meaning that all smaller values are neglected. The integrals are stored in a symmetry adapted form (only storing $\frac{1}{8}$ of the full tensor) and the full tensor is obtained applying the full 8-fold permutation symmetry of the integrals. Throughout this work always full tensors without any index restrictions are used, so that for the water molecule in the STO-3G basis the AO integral tensor is handled as a $7 \times 7 \times 7 \times 7$ array. The trivial casting procedure is done by a simple FORTRAN program writing the tensor in the CP format on disk and leads to an initial rank of 313. This is used as an input for the reduction procedure provided by Espig *et. al* as black box C++ routines. [121, 122, 145, 222] For a given ε of 10^{-4} the initial rank of 313 can be reduced to 87. The error in each element during this procedure is depicted in logarithmic colour coded diagrams in Figure 3.9.

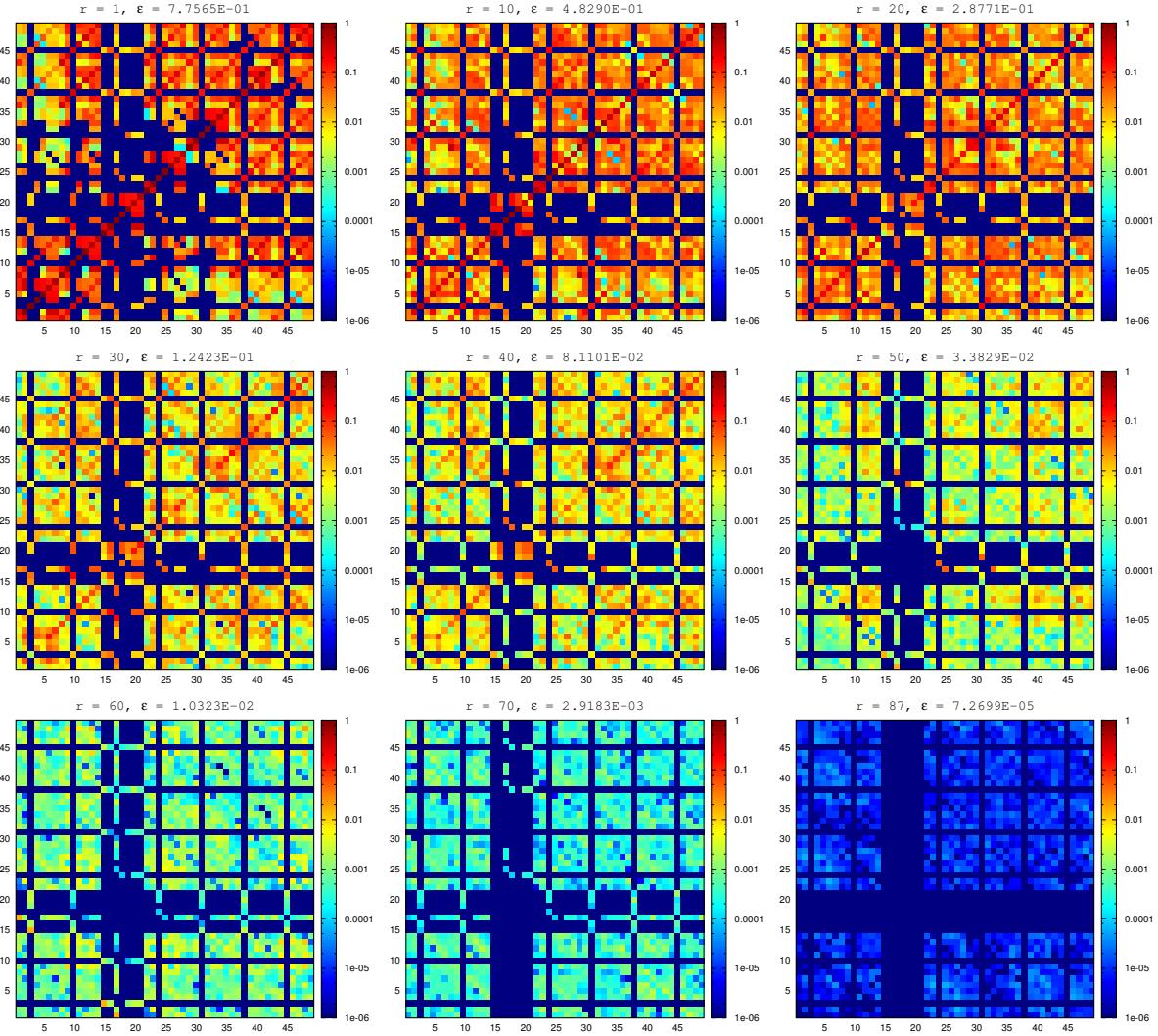


Figure 3.9: Errors in elements during rank reduction of AO two-electron integrals $\langle \mu\sigma | \nu\rho \rangle$ of water in STO-3G basis. The approximated tensor is reassembled at different stages of the reduction procedure and the error to the original AO integral is colour coded in a logarithmic scale. For the visualization the indices $\mu\sigma$ and $\nu\rho$ are represented as a single index on the x and the y axis respectively.

There the approximated tensor is reassembled into an indexed based format at different stages during the reduction procedure so that each value can be compared to the original tensor. Comparing the rank one approximation with the structure of the original tensor (c.f. Figure 3.8) it can be seen that the largest errors occur on the diagonal elements $\mu\sigma = \nu\rho$ where the original tensor has the largest integral values. Some other of the off-diagonal elements around the main diagonal also show large deviations. But there are also regions ($15 < \text{superindex} < 20$) where the error is almost zero. This is due to the fact, that also the original tensor has only very small values in this index region.

Increasing the rank to 20 improves mainly the accuracy for integral values on the main diagonal, while the errors in the other regions become more evenly distributed. For rank = 50 the large elements on the diagonals are fit almost as well as the rest of the tensor. There are some deviations especially for higher values of the superindex, but the overall error in the ℓ^2 -norm is already lower than $3.4 \cdot 10^{-2}$.

At rank 70 the errors in each element are very smooth and uniformly distributed among the whole tensor. Also the diagonal elements are well fitted and despite the cross with zero error no structure of the distribution of errors can be found. At this stage the reduction algorithm can have some problems especially in defining new rank one approximations as the first step in the optimization loop (c.f. Figure 3.5). Here the pivoting routine may need more steps to define a good guess for the new representing vectors as the error is distributed smoothly over the tensor.

Finally, with a rank of 87 the AG algorithm reaches the requested accuracy of $1 \cdot 10^{-4}$ in the ℓ^2 -norm and is even lower than this given threshold. It can be seen, that the errors are very smooth and there are no special regions with pronounced deviations. The mean absolute error averaged over all elements in the tensor is $5.5 \cdot 10^{-7}$ while the maximal error is $6.5 \cdot 10^{-5}$ so that the data in the tensor can be approximated at least with the desired accuracy.

In the reduced representation of the AO two-electron integrals with a rank of 87 an amount of $7 \cdot 876 \cdot 4 = 2436$ elements have to be stored while in the index based representation only $7^4 = 2401$ elements are needed. Modern quantum chemistry packages would furthermore exploit the permutation symmetry (8-fold), that would lower the amount in the index based storage by a factor of 8 to approximately 300 elements. The CP format itself cannot use this symmetry as a further condition, because it is the most general and flexible tensor format with now restrictions in the representing vectors. With symmetric representing vectors or a build-in symmetry the resulting ranks would even be higher than without this condition. Due to the lack of linear dependencies in this very small example it is not possible to compress the data, especially when comparing to the symmetry packed index based representation.

However, the reduction with the AG algorithm takes 25 seconds on a single core of a workstation (c.f. appendix A.1.3) while a full CCSD calculation needs only 1 second. So the rank reduction algorithm itself is very slow compared to the full Coupled-Cluster calculation for this small example. Therefore, a larger basis set is used in order to see how the reduction time increases with the size of the tensor.

As a slightly larger example the AO integrals for water in the 6-31G basis set (13 basis functions, $N = 13$) are used as further test case for the reduction using the original AG algorithm. The original structure of integrals is again shown in a matrix form using superindizes in Figure 3.10. Here, a similar structure compared to the smaller basis set (c.f. Figure 3.8)

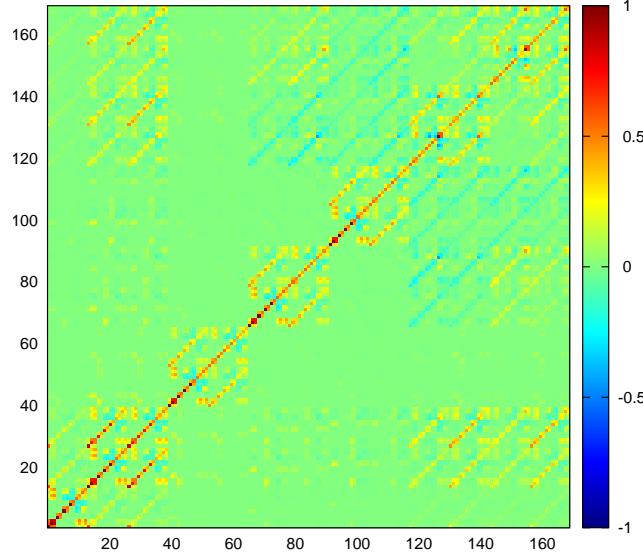


Figure 3.10: Two-electron integrals of water in AO basis for the 6-31G basis set. The indices $\mu\sigma$ and $\nu\rho$ are represented as a single index on the x and the y axis respectively. Each pixel denotes one integral value where red and blue squares stand for large positive or negative values, while green denotes values close to zero.

can be found. Again the largest absolute values are on the main diagonal or around the main diagonal, but some larger values can also be found on other diagonal elements in the corner of Figure 3.10.

The errors in each tensor element during the rank reduction procedure using the original AG-algorithm are shown in Figure 3.11 in a logarithmic scale. For a rank one approximation the errors in almost all elements are fairly large. Only some elements (superindex between 40 and 70) are approximated with high accuracy, because the values in the original tensor are close to zero. The same behaviour can also be found for the smaller basis set (c.f. Figure 3.9). Increasing the rank to 10 this structure is almost lost. Now the errors in the regions, were the two-electron integrals are zero are increasing while the errors for the other elements are decreased. In comparison to the smaller basis set the errors on the main diagonal elements are much less pronounced. Taking the rank 50 approximation as an example one can see that the errors are almost evenly distributed. There are still some regions left where the errors are larger, but increasing the rank to 100 a very homogenous structure of the errors can be seen. The aspired accuracy of 10^{-4} is reached with a rank of 341 and it can be seen, that the error in each element is below the given threshold.

To store the reduced tensor in the CP format with a rank of 341 only $13 \cdot 314 \cdot 4 = 17732$ elements are necessary compared to $13^4 = 28561$ elements for the full tensor stored in an indexed based array. So the representation in the CP format leads to a compression of 1.6.

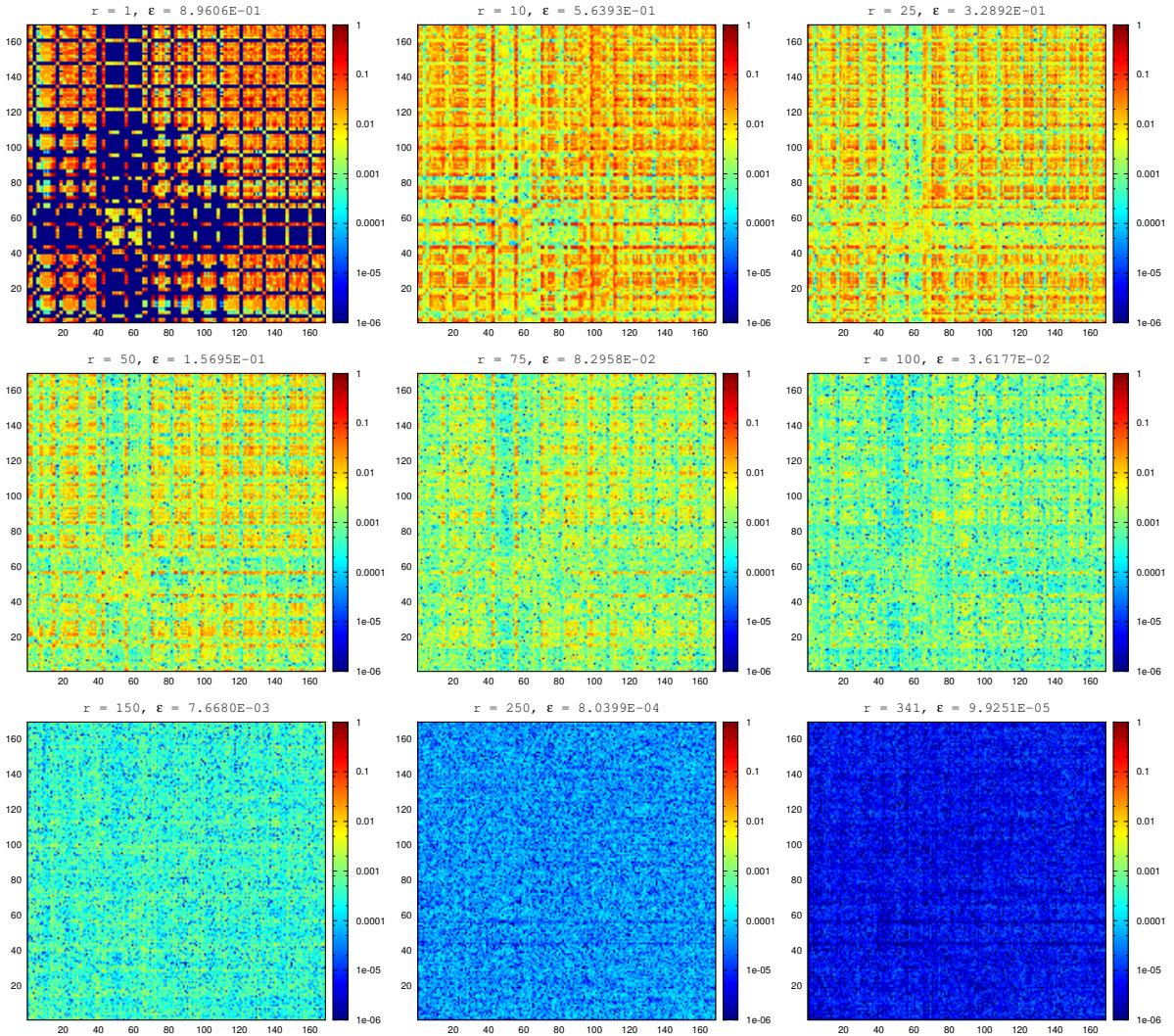


Figure 3.11: Errors in elements during rank reduction of AO two-electron integrals of water in 6-31G basis. The approximated tensor is reassembled at different stages of the reduction procedure and the error to the original AO integral is colour coded in a logarithmic scale.

Nevertheless, if the 8-fold symmetry of the two-electron integrals in the AO basis is exploited, the number of elements in the conventionally stored tensor can be decreased by a factor of 8 so that the representation in the CP format still would be larger. For larger examples the compression in the CP format should be even better, so that the amount of storage is decreased further compared to the normal tensor representation even using the symmetry properties of the tensor.

However, it should be mentioned that the rank reduction takes 917 s CPU time on a single core of a workstation, while a conventional CCSD calculation for the same system only takes 3 s CPU time. Thus the rank reduction algorithm itself is the bottleneck for further calculations due to the fact, that the algorithm is very general and used as a “black box”.

There are several possibilities to improve the performance of the algorithm like exploiting some special properties of the tensors (see Section 3.9) or to use a slicing schemes in order to avoid too large initial ranks and therefore speed up the rank reduction procedure.

3.3.2 Improved Reduction Scheme by Slicing Procedure

As already described in the previous sections the usage of the trivial casting leads to very large initial ranks and the complexity of the AG rank reduction algorithm scales with the size of the initial rank. Therefore, this algorithm is only useful if the rank of the tensor that should be approximated is already moderate. For larger systems and larger basis sets the number of basis functions increases rapidly so that also the initial ranks get larger and larger and the reduction algorithm needs more CPU time. However, the CP format allows for an efficient parallel procedure by treating large quantities in a different way: The large initial tensor in CP format is split up into slices including a fixed number of ranks. The rank reduction is then applied to the individual slices, so that reduced slices are obtained, which can then again be merged (c.f. Figure 3.12) to get the full tensor with reduced rank. In order to obtain a further compression with even lower ranks this procedure can be repeated until the rank does not change any more or a full rank reduction can be carried out for the merged tensor representation. This procedure can trivially be distributed to multiple processes and also lends itself to a distributed integral direct algorithm, where different nodes calculate different integral batches, that can be reduced on the same node on the fly. Furthermore, the memory requirements for each reduction step are also reduced, as only small pieces of the tensor have to be kept in the memory of the computer.

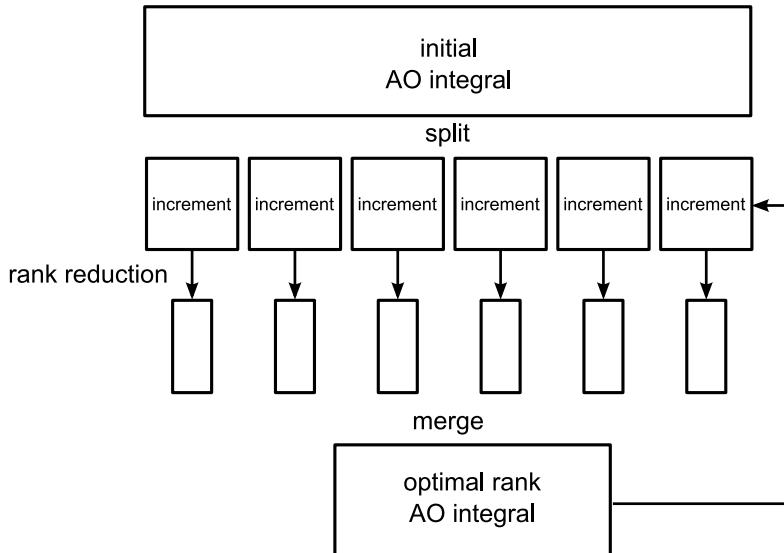


Figure 3.12: Schematic representation of the parallel rank reduction currently used for large examples.

Table 3.1: Test calculations with various slice sizes using water with cc-pVDZ (24 basis functions, full initial rank = 13824) and aug-cc-pVDZ (41 basis functions, full initial rank = 68.921) basis set. Shown are the number of slices and the final merged rank after one single sliced reduction (one cycle from the scheme shown in Figure 3.12) with an accuracy of $\varepsilon = 10^{-4}$.

slice size	cc-pVDZ		aug-cc-pVDZ	
	# of slices	merged rank	# of slices	merged rank
250	56	6923	276	39093
500	28	5207	138	30436
1000	14	4180	69	21466
2000	7	3257	35	15493

To determine a good size of the slices, various test calculations for water using the cc-pVDZ (24 basis functions) and the aug-cc-pVDZ (41 basis functions) basis sets have been carried out. The results for increasing slices are listed in Table 3.1 and it can be seen, that the final merged rank is always lower for larger slice sizes due to the fact, that more linear or nearly linear dependent data is contained in the same slice. This way the algorithm can efficiently compress the original data and decrease the final rank. Also the number of slices that have to be reduced gets smaller for larger slice sizes. On the other hand, the reduction of a smaller size is of course much faster because the initial as well as the final ranks are very small and the algorithm needs only a few iterations to obtain a low-rank approximation.

As a good low-rank approximation is vital for the following mathematical operations dealing with high order tensors in the CP format the best scheme is to use as large as possible slices. Here and in the following a typical slice size of 1000 is used, as this size offers best compression at affordable computational effort.

A typical full rank reduction of AO two-electron integrals with an initial rank of 5000 to $\varepsilon = 10^{-4}$ takes 8.6 h of CPU time on one core of a workstation. The parallelized sliced reduction with slices of 1000 ranks and the same threshold takes $5 \cdot 0.4$ CPU h on five CPUs. A subsequent full reduction to $\varepsilon = 10^{-4}$ of the prereduced slices takes 3.7 h on a single CPU, so the sliced reduction followed by a full reduction leads to good rank reduction and reduced CPU time.

Another problem of the original reduction algorithm in a full reduction of AO integrals to high accuracies is the slow convergence of the algorithm. Furthermore, especially for larger examples the expected final ranks are large, so that the algorithm needs a lot of iterations to collect a suitable amount of representing vectors without increasing the accuracy much. The typical convergence of the error ε in the ℓ^2 -norm is shown in Figure 3.13. Here the AO integrals for CH₄, CO and HCN using the 6-31G basis set are reduced to a final accuracy

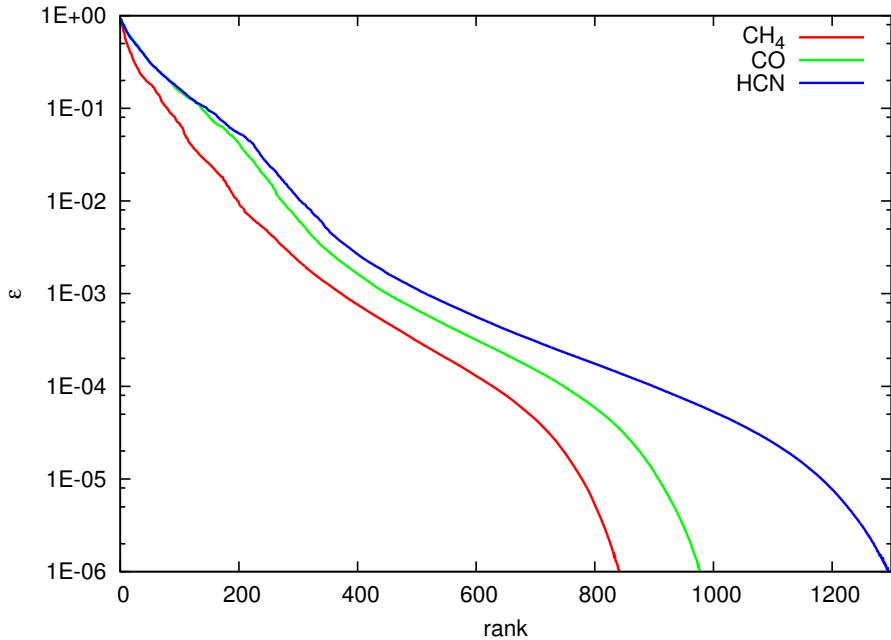


Figure 3.13: Convergence of error ε in the ℓ^2 -norm for full reduction of AO integrals in CH_4 , CO and HCN using 6-31G basis set.

of 10^{-6} with a full reduction. It can be seen, that especially for an error in the range of 10^{-3} to 10^{-4} the convergence is very slow and additional representation vectors improve the accuracy only slightly. Therefore, it is desirable to adapt the rank reduction scheme in a way, that more representing vectors can be collected in a single iteration and the convergence of the algorithm can be improved. However, for testing the applicability of the CP format for tensors in post Hartree-Fock methods the original implementation of the reduction algorithm together with the slicing of tensors for larger initial ranks is sufficient. Further improvements of the reduction algorithm are discussed in Section 3.9.

3.3.3 Comparison of Sliced and Full Reduction Procedure

In this section the decomposition of AO integrals using the trivial casting procedure and the rank reduction to obtain a low-rank approximation for a small test-set of molecules (see Table 3.2) are discussed. Here, different schemes for the rank reduction are tested: A single sliced rank reduction (*sr*), a single sliced reduction followed by a full rank reduction (*sfr*) and a full rank reduction (*fr*) of the AO integral tensor. From Table 3.2 it can be seen that the single sliced rank reduction does not lead to a good reduction for the AO integrals. The compression of ranks¹ in CH_4 for example is between 53 % for the highest accuracy ($\varepsilon = 10^{-6}$) and 92 % for the lowest accuracy ($\varepsilon = 10^{-2}$). If the remaining representing vectors are combined to one slice and reduced (*sfr*) the compression can be improved further. The final

¹The compression is calculated as: compression = $100 \cdot \left(1 - \left(\frac{\text{reduced rank}}{\text{initial rank}} \right) \right)$

Table 3.2: Initial and final ranks for different accuracies for the AO-integrals in HF, H₂O, NH₃, CH₄, CO, N₂ and HCN using the 6-31G basis set. The used geometries are listed in the appendix A.1.5. N denotes the number of elements per dimension that is equivalent to the number of basis functions. Note that for the LiH and HF no sliced reduction is necessary due to the small initial ranks.

			sliced reduction (<i>sr</i>) (size=1000)			sliced + full (<i>sfr</i>) reduction (size=1000)			full reduction (<i>fr</i>)		
molecule	N	initial rank	ε			ε			ε		
			10 ⁻²	10 ⁻⁴	10 ⁻⁶	10 ⁻²	10 ⁻⁴	10 ⁻⁶	10 ⁻²	10 ⁻⁴	10 ⁻⁶
LiH	11	1019	—	—	—	—	—	—	94	221	264
HF	11	1019	—	—	—	—	—	—	113	225	263
H ₂ O	13	2005	205	598	852	148	337	403	141	343	409
NH ₃	15	3375	307	966	1409	183	488	603	171	484	603
CH ₄	17	4793	398	1397	2233	210	648	844	199	628	842
CO	18	4872	493	1593	2488	286	781	984	274	748	979
N ₂	18	4872	490	1575	2428	283	747	987	270	729	978
HCN	20	6848	642	2104	3419	318	956	1311	304	899	1295

rank for $\varepsilon = 10^{-6}$ can be reduced by 82 % while the rank for $\varepsilon = 10^{-2}$ can be decreased by 96 %. Furthermore, even from these examples it can be seen that the compression for larger examples is better.

If a full reduction (*fr*) is carried out for the AO integrals the results are very similar to those obtained by the sliced reduction followed by a full reduction (*sfr*). Here, the final ranks are almost identical and deviate at most about 5 % for $\varepsilon = 10^{-2}$ and 1 % for $\varepsilon = 10^{-6}$. Therefore, the best balance between computational time and reduction of rank is obtained with a sliced reduction followed by a full reduction (*sfr*). This scheme is used for all larger examples (initial rank > 5000) for which a full rank reduction is not feasible due to the limitations of the initial implementation of the rank reduction algorithm. With this pilot implementation, that is a general-purpose, BLAS-level 1 and 2 (Basic Linear Algebra Subprograms) based C++ development platform, only initial ranks up to 10000 can be afforded. For an example of this size, the rank reduction algorithm takes in the order of days using a single core on a modern workstation computer.

To compare the resulting low-rank approximation in the CP format with the conventional index based tensor in terms of memory requirements Table 3.3 lists the number of elements that have to be stored in these formats. It can be seen, that for all given molecules and all tested accuracies in the CP format the amount of stored elements is smaller than in the full conventional index based tensor representation. Even though for the smaller molecules the number of elements is almost the same especially for large accuracy ($\varepsilon = 10^{-6}$), the compression for larger molecules and larger values of ε is quite good. Nevertheless, it is still

Table 3.3: Comparison of memory requirements between index based, conventional tensor format and CP format obtained by a full rank reduction using the 6-31G basis set. Shown are the number of elements that have to be stored in the different formats. For the conventional storage it is distinguished between the full storage (N^4 elements) and the symmetry adapted storage ($\approx \frac{N^4}{8}$ elements). It should be noted, that the systems and basis sets are very small and a better compression in the CP format can be expected for larger molecules and basis sets.

molecule	N	conventional		ε		
		full	sym	10^{-2}	10^{-4}	10^{-6}
LiH	11	14641	1830	4136	9724	11616
HF	11	14641	1830	4972	9900	11572
H ₂ O	13	28561	3570	7384	17524	20956
NH ₃	15	50625	6328	10320	28560	35940
CH ₄	17	83521	10440	13532	42976	57120
CO	18	104976	13122	19728	53856	70488
N ₂	18	104976	13122	19440	52488	70416
HCN	20	160000	20000	24320	71920	103600

not possible to reach the limit of symmetry adapted storage in the conventional format due to the fact, that the representing vectors do not contain any symmetry information of the target tensor. However, for the larger molecules the overhead in memory requirements of the CP format with smallest accuracy decreases. So it could be expected, that for larger molecules and larger basis sets the amount of storage in the CP format will decrease further and at some point the CP format becomes preferential over the symmetry adapted conventional format.

3.3.4 Lowering of Initial Ranks by SVD-based Casting Procedure

So far only the trivial casting procedure is used to get a decomposed representation of the AO integrals in the CP format. However, as already mentioned before, the initial rank for this procedure is very large and scales approximately with N^3 , where N is the number of basis functions. It is also possible to apply the decomposition based on the SVD of submatrices of the full tensor (see Section 3.1.2). In order to compare the results from the two different casting procedures, the AO integrals for the molecules in the testset using the 6-31G basis set are chosen as an example. The three given accuracies (10^{-2} , 10^{-4} and 10^{-6}) are used as a cutoff threshold for the SVD (smaller eigenvalues are omitted) as well as the threshold for the rank reduction procedure. The resulting initial ranks after the decomposition (R_{init}) and the final ranks after applying the rank reduction algorithm (R_{final}) are shown in Table 3.4. Here it can be seen, that the initial ranks of SVD cast tensors especially for high values of ε are significantly lower than for the trivial decomposition. On average the initial ranks for

Table 3.4: Comparison of ranks for trivial casting (triv.) and decomposition using SVD for molecules in testset using the 6-31G basis set. R_{init} denotes the initial rank after the decomposition procedure, while R_{final} describes the rank after application of the rank reduction procedure to the cast tensors. It should be noted that the same accuracy parameter ε used for the SVD casting procedure is utilized for the rank reduction in the CP format.

molecule	triv.	R_{init}			R_{final}					
		SVD ε			$\varepsilon = 10^{-2}$		$\varepsilon = 10^{-4}$		$\varepsilon = 10^{-6}$	
		10 ⁻²	10 ⁻⁴	10 ⁻⁶	triv.	SVD	triv.	SVD	triv.	SVD
LiH	1019	545	833	835	94	94	221	223	264	265
HF	1019	755	831	835	113	111	225	226	263	263
H ₂ O	2005	1405	1667	1677	141	146	343	342	409	409
NH ₃	3375	2321	3033	3063	171	181	484	484	603	604
CH ₄	4793	3044	4157	4181	199	211	628	633	842	842
CO	4872	2736	3810	3908	274	284	748	741	979	976
N ₂	4872	2782	3794	3906	270	279	729	736	978	976
HCN	6848	3557	5122	5300	304	316	899	894	1295	1294

$\varepsilon = 10^{-2}$ can be reduced by 40 % of the initial ranks obtained by the trivial decomposition. For the larger examples the reduction is even more pronounced than for the molecules with fewer basis functions and reaches up to 50 % for the HCN molecule. However, the initial rank for larger accuracy (10^{-6}) is almost the same as the full rank and deviates only about 17 % on average.

The final ranks after applying the rank reduction in the CP format with the same accuracy parameter as chosen for the SVD based casting procedure are almost the same as the ranks obtained by reducing the trivial cast integral tensor. There, the deviations are more pronounced for the largest threshold parameter (10^{-2}) and can be up to 5 % while for the higher accuracies the final ranks are almost identical and deviate less than 0.1 % from the ranks obtained from trivial cast integrals.

The accuracy for each element can be compared to the conventional tensor by a reassembly of the low-rank representation and a statistical analysis. The mean absolute deviations (MAD) between each element in the reduced CP representation obtained by trivial and SVD based casting are listed in Table 3.5. Comparing the accuracies of the final reduced representation one can see almost no deviation between the two decomposition procedures. For $\varepsilon = 10^{-2}$ the deviations are a bit larger than for higher accuracies, but the order of magnitude in the mean absolute deviation stays the same. It should also be mentioned, that the MAD is almost two orders of magnitude smaller than the given accuracy parameter in the

Table 3.5: Comparison of accuracies for trivial casting (triv.) and decomposition using SVD with the same accuracy parameter as for the rank reduction in the CP format. The integral tensors are calculated using the 6-31G basis set. The mean absolute deviation (MAD) between all elements in the integral tensor are shown for low-rank representations after applying the rank reduction with three given threshold parameters (10^{-2} , 10^{-4} and 10^{-6}).

molecule	MAD					
	$\varepsilon = 10^{-2}$		$\varepsilon = 10^{-4}$		$\varepsilon = 10^{-6}$	
	triv.	SVD	triv.	SVD	triv.	SVD
LiH	$2.776 \cdot 10^{-4}$	$4.158 \cdot 10^{-4}$	$3.576 \cdot 10^{-6}$	$3.610 \cdot 10^{-6}$	$3.724 \cdot 10^{-8}$	$3.410 \cdot 10^{-8}$
HF	$8.034 \cdot 10^{-4}$	$8.385 \cdot 10^{-4}$	$9.168 \cdot 10^{-6}$	$9.069 \cdot 10^{-6}$	$7.884 \cdot 10^{-8}$	$9.227 \cdot 10^{-8}$
H ₂ O	$6.064 \cdot 10^{-4}$	$6.768 \cdot 10^{-4}$	$7.184 \cdot 10^{-6}$	$7.203 \cdot 10^{-6}$	$7.156 \cdot 10^{-8}$	$6.876 \cdot 10^{-8}$
NH ₃	$5.144 \cdot 10^{-4}$	$5.809 \cdot 10^{-4}$	$5.964 \cdot 10^{-6}$	$6.062 \cdot 10^{-6}$	$5.861 \cdot 10^{-8}$	$5.683 \cdot 10^{-8}$
CH ₄	$4.347 \cdot 10^{-4}$	$5.276 \cdot 10^{-4}$	$5.185 \cdot 10^{-6}$	$5.163 \cdot 10^{-6}$	$5.064 \cdot 10^{-8}$	$4.899 \cdot 10^{-8}$
CO	$3.466 \cdot 10^{-4}$	$4.126 \cdot 10^{-4}$	$4.530 \cdot 10^{-6}$	$4.563 \cdot 10^{-6}$	$4.258 \cdot 10^{-8}$	$4.477 \cdot 10^{-8}$
N ₂	$3.558 \cdot 10^{-4}$	$4.193 \cdot 10^{-4}$	$4.571 \cdot 10^{-6}$	$4.605 \cdot 10^{-6}$	$4.453 \cdot 10^{-8}$	$4.418 \cdot 10^{-8}$
HCN	$2.962 \cdot 10^{-4}$	$3.477 \cdot 10^{-4}$	$3.975 \cdot 10^{-6}$	$4.026 \cdot 10^{-6}$	$3.978 \cdot 10^{-8}$	$4.092 \cdot 10^{-8}$
average	$4.544 \cdot 10^{-4}$	$5.274 \cdot 10^{-4}$	$5.519 \cdot 10^{-6}$	$5.538 \cdot 10^{-6}$	$5.297 \cdot 10^{-8}$	$5.385 \cdot 10^{-8}$

ℓ^2 -norm, so that only the maximal error in a single element reaches the threshold parameter. From this it can be concluded, that the decomposition by SVD can be used without a loss of accuracy, as long as the same or a lower threshold parameter is chosen for the SVD compared to the rank reduction in the CP format. The SVD based decomposition leads to significantly smaller initial ranks especially for small accuracies and therefore reduces the computational effort for the following rank reduction in the CP format, as the reduction algorithm scales with the initial rank.

3.3.5 Slicing Scheme based on SVD Casting Procedure

The SVD based casting procedure offers the possibility to modify the sliced reduction scheme to an adaptive slice size. In the casting procedure for the AO integral $v_{AO} = \langle \mu\sigma | \nu\rho \rangle$ one can collect all σ submatrices $B(\nu\rho)$ for a fixed value of μ . After performing the N (number of basis functions) Singular Value Decompositions the representing vectors are set up as unity vectors for the first two dimensions: the fixed μ and the variable σ , while the last two representing vectors are initialized by the eigenvectors obtained from the SVD. Again, already at this stage the number of eigenvectors can be truncated by the size of the corresponding eigenvalues. This way the initial rank for each value of μ is determined by the SVD and the rank reduction in the CP format can be carried out N times for the prereduced slices. A pseudocode for this procedure is given in Figure 3.14. This procedure allows for a good prereduction by omitting small eigenvalues identified with the SVD and offers a somehow

SVD based slicing scheme

```
1: index based AO integral tensor  $\langle \mu\sigma|\nu\rho \rangle$  as  $\chi(\mu, \sigma, \nu, \rho)$ 
2: for  $\mu$  do
3:   for  $\sigma$  do
4:     for  $\nu$  do
5:       for  $\rho$  do
6:         set up  $B(\nu, \rho)$  for fixed  $[\mu, \sigma]$ 
7:       end for
8:     end for
9:   SVD( $\mathbf{B}$ )  $\rightarrow \mathbf{B} = \mathbf{U} \cdot \Sigma \cdot \mathbf{V}^T$ 
10:  set up rows in  $(\mathbf{U} \cdot \Sigma)$  and columns in  $\mathbf{V}^T$  as representing vectors
11:  collect all representing vectors and add unity vectors coding index  $\mu$  and  $\sigma$ 
12: end for
13: use rank reduction of collected increment
14: end for
15: add all reduced increments together to prerduced tensor in CP format
```

Figure 3.14: Pseudocode for SVD based slicing scheme.

natural slicing with the number of basis functions. The initial ranks for the sliced reduction in the CP format scale at most with $\mathcal{O}(N^2)$, but are often lower especially when larger threshold values are applied. Then most of the eigenvalues are smaller than the given accuracy parameter and do not contribute to the initial ranks so that the rank reduction in the CP format can be done very efficiently. The sliced reduction also allows for a very efficient parallelization of the reduction as all slices can in principle be done in parallel without any communication between the single reduction procedures. There, only the initial data and the final representing vectors have to be communicated and distributed.

With the help of the sliced reduction based on a SVD driven casting it is also possible to decompose AO integrals for somewhat larger molecules. The largest so far tested molecule is furoxane in a 6-31G basis set yielding 44 electrons (22 doubly occupied orbitals) described by 58 basis functions. This molecule is chosen, as it is known to be a problematic case when trying to approximate electron correlation by methods like a dynamical thresholding algorithm or other localization schemes. [228, 229]

The SVD based decomposition reduces the initial rank of the AO integral tensor of furoxane from 192,460 to 121,778 for an accuracy parameter of 10^{-2} so that only 63 % of the full rank is needed. The initial ranks for a subsequent sliced reduction with a variable slice size determined by the number of small eigenvalues in the SVD for the corresponding subblock of the tensor varies in the range of 831 to 2864. This leads to 58 reductions of slices of the prerduced AO tensor, that can all be done in parallel. Nevertheless, these 58 reductions to

an accuracy of 10^{-2} take only 4241 seconds on a single core of a compute note (c.f. A.1.4) leading to an averaged reduction time of 73 seconds per slice. The reduced rank after this single sliced reduction (*sr*) is 4651, which is only 2.4 % of the initial rank based upon the trivial decomposition. The rank can be compressed by a factor of 41.4 which corresponds to a compression rate of 97.6 %. Already at this state (only a single sliced reduction passed) the amount of storage in the CP format ($58 \cdot 4651 \cdot 4 = 1,079,032$ elements) is one order of magnitude lower than storing $58^4 = 11,316,496$ elements in the index based tensor representation. Even if the symmetry adapted storage is assumed, that reduces the number of elements roughly by a factor of 8, the representation in the CP format still needs only 76 % of the number of elements. It should be noted, that this example is still a small molecule in a small basis set and the compression through rank reduction in the CP format can be expected to be even larger for larger molecules and larger basis sets.

Furthermore, a full reduction can be carried out for the prereduced tensor, which resembles the already mentioned *sfr* reduction scheme. The rank reduction yields a final rank of 2414. So overall the rank can be compressed by 98.7 % which corresponds to a factor of 79.7, while maintaining a certain accuracy in each element. The mean absolute deviation between the entries in the index based tensor representation and the low-rank approximation is $5.927 \cdot 10^{-5}$ while the maximal error is 0.01 and therefore in the order of the chosen threshold. So the chosen ℓ^2 -norm criterion to asses the quality of the low-rank approximation seems to be very good. Nevertheless, there is no easy way to correlate the ℓ^2 -error with an error in the calculated energy as only the two-electron integrals in the AO basis are approximate. As the scope of this work are high level post Hartree-Fock methods it is not intended to use the AO integral tensor in the low-rank CP format to reformulate the Hartree-Fock equation and approximate the Hartree-Fock energy. The Hartree-Fock procedure only scales with N^3 and involves up to 4 dimensional quantities so that the CP format is not well suited for reducing the complexity of the HF-SCF cycle especially as the more complex rank reduction procedure is needed to obtain a low-rank approximation. The AO integrals are merely used as an initial test of the rank reduction algorithm and as a starting point for the following integral transformation and the calculation of correlation energy in CI based methods.

3.4 Benchmark: Scaling of Reduced Ranks

Now that an efficient scheme for the casting of a conventional tensor in the CP format followed by a sliced rank reduction has been devised the decisive quantity for the application of the CP representation in high level post Hartree-Fock calculations is investigated. In the CP format the most important and size determining quantity is the rank and especially the scaling of the rank with the system or basis set size. In order to assess the scaling of the final rank after reduction of the two-electron integrals, a LiH chain (for details on the used geometries see Appendix A.1.5) in a 6-31G basis and a H₂O molecule using different basis sets are studied. [146] As the largest molecules and basis sets contain more than 40 basis

Table 3.6: Initial ranks using the SVD based casting procedure (R_{init} SVD), the reduced ranks for the adaptive sliced reduction (R_{sr}) and ranks obtained from a subsequent full reduction (R_{sfr}) of AO integrals in LiH chains using the 6-31G basis set. It should be noted, that some full reductions (marked with $-$) are omitted due to too large initial ranks.

molecule	N	R_{init} SVD			R_{sr}			R_{sfr}		
		10^{-2}	10^{-4}	10^{-6}	10^{-2}	10^{-4}	10^{-6}	10^{-2}	10^{-4}	10^{-6}
(LiH) ₁	11	545	833	835	191	419	633	96	225	268
(LiH) ₂	22	2867	6094	6770	700	2171	3994	362	1065	1678
(LiH) ₃	33	6391	16005	20089	1401	4792	10392	747	2301	3396
(LiH) ₄	44	10845	29578	39786	2247	8038	19391	1264	2301	$-$
(LiH) ₅	55	16043	46465	65249	3252	11844	30428	1885	3345	$-$
(LiH) ₆	66	21851	66352	96316	4363	16052	42627	2605	$-$	$-$

functions only single sliced rank reductions (*sr*) and a following full reduction (*sfr*) can be performed. Here it should be noted, that for the largest molecules the final full reduction in the *sfr* scheme for $\varepsilon = 10^{-6}$ has still too large initial ranks and therefore is omitted in the following discussion. The initial ranks using the SVD based casting procedure as well as the reduced ranks using the adapted slicing scheme based on the SVD are listed in Table 3.6 and Table 3.7 and a graphical representation of the scaling is presented in Figure 3.15.

Table 3.7: Initial ranks using the SVD based casting procedure (R_{init} SVD), the reduced ranks for the adaptive sliced reduction (R_{sr}) and ranks obtained from a subsequent full reduction (R_{sfr}) of AO integrals in H₂O using the STO-3G, 6-31G, 6-31G*, cc-pVDZ, aug-cc-pVDZ and cc-pVTZ basis set. It should be noted, that some full reductions are omitted due to to large initial ranks.

basis set	N	R_{init} SVD			R_{sr}			R_{sfr}		
		10^{-2}	10^{-4}	10^{-6}	10^{-2}	10^{-4}	10^{-6}	10^{-2}	10^{-4}	10^{-6}
STO-3G	7	247	261	261	101	140	165	50	75	92
6-31G	13	1405	1667	1677	326	675	989	151	343	410
6-31G*	18	3604	4264	4290	624	1745	2540	229	761	979
cc-pVDZ	24	8100	10828	10920	971	3573	5577	319	1277	2145
aug-cc-pVDZ	41	33367	52867	54999	2607	12110	23011	674	$-$	$-$
cc-pVTZ	58	110722	157440	161058	4498	24799	59085	827	$-$	$-$

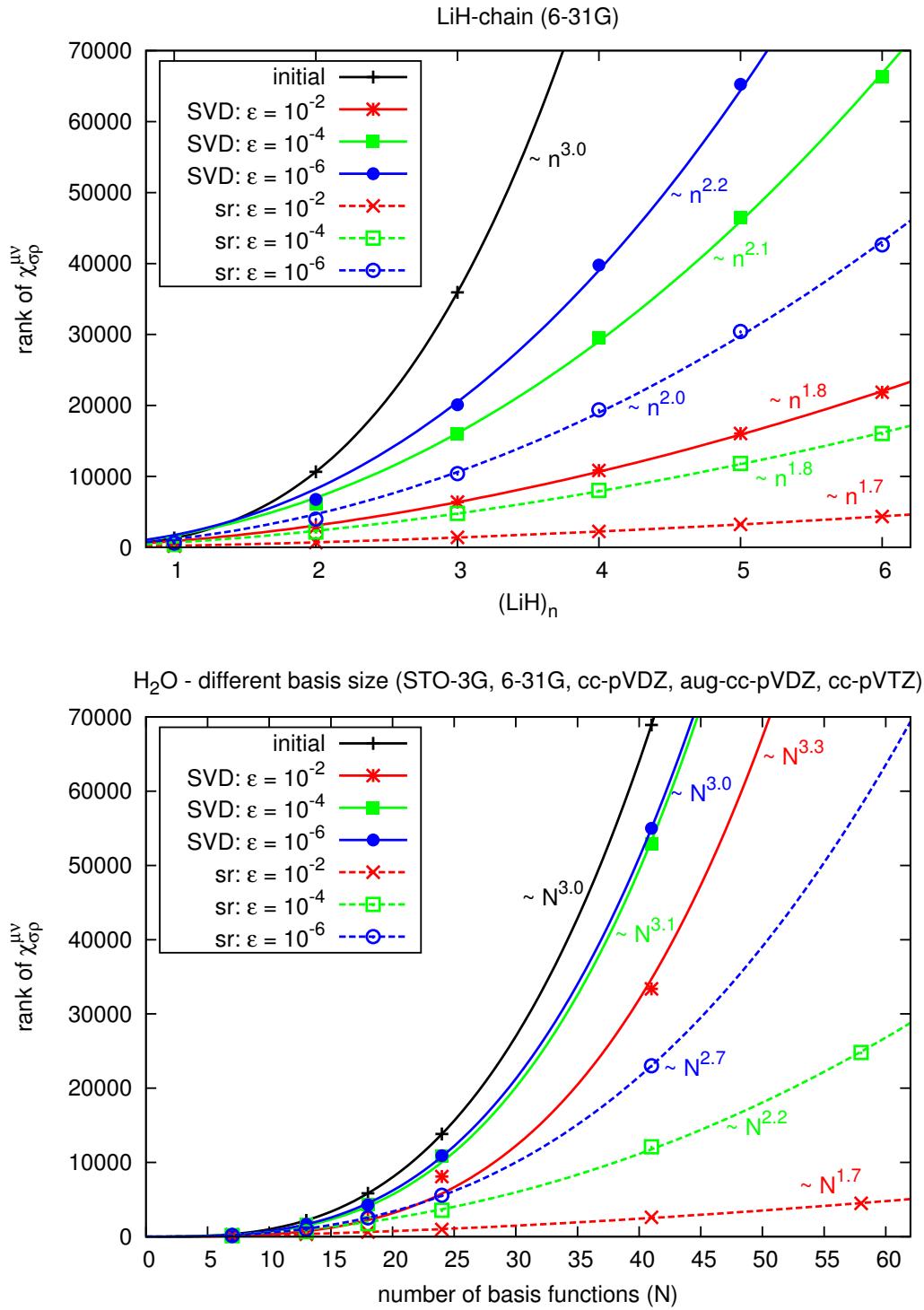


Figure 3.15: Scaling of reduced ranks for AO-integrals with system size for LiH chains using the 6-31G basis set and with number of basis functions for H₂O using the basis set series: STO-3G, 6-31G, 6-31G*, cc-pVDZ, aug-cc-pVDZ, cc-pVTZ. Both plots show results from a SVD prerduction and a single sliced (*sr*) reduction. For both systems the slice size is adapted by the SVD.

There, the scaling of the ranks is evaluated by fitting an exponential function $f(x) = a \cdot x^b$ to the plotted functions. The most important quantity is the exponent, which determines the actual scaling of ranks with increasing basis (N) or system size (n). For the simple prereduction of the initial ranks with the SVD based casting procedure a scaling of ranks of $\mathcal{O}(n^{1.8})$ with system size and $\mathcal{O}(N^{3.3})$ with basis set size is obtained using a threshold of 10^{-2} . The even higher scaling with basis set size (compared to the initial ranks from a trivial decomposition) can be attributed to fitting errors for the given functional form. Therefore, the SVD based prereduction can only be seen as a procedure to lower the initial ranks for a following rank reduction in the CP format. Although the scaling with system size is already lower than the scaling of the corresponding full initial rank, the SVD procedure mostly changes the prefactor of the fitted exponential function.

In a following adapted sliced reduction (*sr*) procedure the scaling with system size can be further decreased to $\mathcal{O}(n^{1.7})$ and a scaling with basis set size of $\mathcal{O}(N^{1.7})$, respectively. If larger accuracies are desired ($\varepsilon = 10^{-4}$), the scaling increases slightly to $\mathcal{O}(n^{1.8})$ and $\mathcal{O}(N^{2.2})$. Nevertheless, the scaling behaviour with increasing system size is almost independent of the chosen ε value, while for increasing basis set size the scaling depends more on the threshold parameter. In summary, one can assume a scaling of reduced ranks in the order of $n^{1.8}$ and $N^{2.2}$, which corresponds to a scaling of memory requirements in the order of N^3 and therefore is approximately one order lower than for index based fourth order tensors.

Performing a final full reduction of the already approximated integral tensors mainly changes the prefactor of the fitted function. The actual scaling is only slightly influenced, except for the scaling with basis set size for the largest threshold (10^{-2}), that can be lowered to $\mathcal{O}(N^{1.1})$. As a summary the parameters for the fitted exponential function of all three reduction procedures are shown in Table 3.8.

Table 3.8: Parameters for fitted functions $f(x) = a \cdot x^b$ of AO integrals using three different reduction procedures: SVD based prescreening, adapted slice size reduction and a following full reduction. Note, that for the LiH chain it could only be afforded to perform a full reduction with the smallest threshold value for 3 data points (up to $(\text{LiH})_3$), which makes the fit unreliable so that it is skipped here.

ε	SVD prereduction				adapted slice size reduction				following full reduction			
	system size		basis size		system size		basis size		system size		basis size	
	a	b	a	b	a	b	a	b	a	b	a	b
10^{-2}	894.44	1.8	0.14	3.3	226.55	1.7	4.77	1.7	104.39	1.8	10.02	1.1
10^{-4}	1686.68	2.1	0.50	3.1	687.06	1.8	3.80	2.2	264.49	2.0	1.91	2.1
10^{-6}	1741.50	2.2	0.68	3.0	1156.15	2.0	1.15	2.7	—	—	0.40	2.7

For comparison of the adapted slice size procedure based upon the SVD with a fixed slice size using the trivial decomposition technique H₂O with different basis sets is taken as an example. Here, slices with ranks of 500, 1000 and 2000 are reduced using the *sr* procedure and the final ranks can be compared to the results obtained from SVD based reduction with a variable initial rank (see Table 3.9). Generally, larger fixed slice sizes lead to better reduction of ranks and also the scaling with basis set size can be reduced by choosing larger slice sizes. Nevertheless, the *sr* procedure using the SVD based decomposition to get a variable initial rank for different slices leads to comparable or even lower ranks especially if larger basis sets are used. It should be noted that a subsequent full rank reduction would yield practically the rank of the *fr* procedure, no matter what slice size has been chosen in the initial sliced prereduction (c.f. Table 3.2 or Table 3.5). In conclusion, the optimal performance of the rank reduction algorithm is achieved with the adaptive slice size of the SVD based decomposition due to lower initial ranks. Although the ranks can be somewhat larger than for bigger fixed slice sizes, a good balance between reduction time and compression is found.

Table 3.9: Reduced ranks for sliced reductions (*sr*) with different slice sizes for AO integrals in H₂O using different basis sets.

basis set	slice = 500			slice = 1000			slice = 2000			adaptive size SVD		
	10 ⁻²	10 ⁻⁴	10 ⁻⁶	10 ⁻²	10 ⁻⁴	10 ⁻⁶	10 ⁻²	10 ⁻⁴	10 ⁻⁶	10 ⁻²	10 ⁻⁴	10 ⁻⁶
6-31G	253	698	1000	205	598	852	149	346	474	326	675	989
cc-pVDZ	1610	5207	7066	1107	4180	6914	782	3257	5555	971	3573	5577
aug-cc-pVDZ	9029	30436	41108	5380	21466	34449	3501	15493	31350	2607	12110	23011
cc-pVTZ	27293	95480	127789	15562	68872	109808	9219	45369	87386	4498	24799	–

3.4.1 Violation of Permutational Symmetry

The general structure of the CP format combined with the native implementation of the rank reduction procedure does not allow to impose any symmetric properties of the representing vectors. Therefore, the representing vectors in the reduced rank representation of the AO two-electron integrals do not necessarily contain the permutational symmetry of the full tensor. In order to assess the error that arises due to this missing permutational symmetry in the decomposed format, the values of integrals that should be identical by permutational symmetry have been compared after reassembly. This has been done for the AO integrals as well as for the MO integrals for several test cases and different values of ε . In all cases the difference δ of two equivalent integrals

$$\delta = \langle pq \| rs \rangle - \langle qp \| sr \rangle \leq \varepsilon \cdot 10^{-2}, \quad (3.20)$$

is at least two orders of magnitude smaller than the error in the ℓ^2 -norm introduced by the low-rank approximation and can thus safely be neglected. One can also try to modify the reduction procedure to ensure permutational symmetry with a symmetric structure of the representing vectors. However, this is beyond the scope of this work.

3.5 AO-MO Transformation with Integrals in CP Format

In order to calculate the correlation energy in the framework of perturbation theory or with CI based wave function methods the two-electron integrals have to be transformed to the MO basis. During the Hartree-Fock SCF procedure the AO-MO transformation coefficients (denoted as $C_{\nu i}$ in Eqn. 2.31) are optimized and the transformation to the MO basis can be done using the index based AO integral tensor $\langle \mu\sigma|\nu\rho \rangle$ as

$$\langle ab|cd \rangle = \sum_{\mu\sigma\nu\rho} C_{\mu}^a C_{\sigma}^b C_{\nu}^c C_{\rho}^d \langle \mu\sigma|\nu\rho \rangle. \quad (3.21)$$

It should be noted, that this formal N^8 procedure can be carried out in 4 steps with approximately N^5 complexity. Eqn. 3.21 shows the example for the transformation into the full virtual space. The other transformations into the all occupied space and also mixed transformations are done in the same way with different transformation coefficients. Assuming a representation of the AO integral tensor in a low-rank CP format

$$\langle \mu\sigma|\nu\rho \rangle = \mathbf{v}_{AO} \approx \sum_{r=1}^R \chi_r^{(\mu)} \otimes \chi_r^{(\sigma)} \otimes \chi_r^{(\nu)} \otimes \chi_r^{(\rho)}, \quad (3.22)$$

one can rewrite Eqn. 3.21 as

$$\langle ab|cd \rangle = \mathbf{v}_{cd}^{ab} \approx \mathbf{C}^{(a)} \mathbf{C}^{(b)} \mathbf{C}^{(c)} \mathbf{C}^{(d)} \sum_{r=1}^R \chi_r^{(\mu)} \otimes \chi_r^{(\sigma)} \otimes \chi_r^{(\nu)} \otimes \chi_r^{(\rho)}, \quad (3.23)$$

using the matrix or tensor based notation with implicit summation over all common AO indices, or more convenient in index based formulation

$$\langle ab|cd \rangle = \mathbf{v}_{cd}^{ab} \approx \sum_{\mu\sigma\nu\rho} C_{\mu}^a C_{\sigma}^b C_{\nu}^c C_{\rho}^d \sum_{r=1}^R \chi_r^{\mu} \chi_r^{\sigma} \chi_r^{\nu} \chi_r^{\rho}. \quad (3.24)$$

Here it can be seen, that the summation over the AO indices (μ, ν, σ and ρ) can be done separately, as the four representing vectors are separated from each other. Rewriting the transformation, so that the summation over same indices can be carried out efficiently as matrix-matrix multiplication, leads to

$$\langle ab|cd \rangle = v_{cd}^{ab} \approx \sum_{r=1}^R \underbrace{\left(\sum_{\mu} C_{\mu}^a \chi_r^{\mu} \right)}_{:=v_r^a} \cdot \underbrace{\left(\sum_{\sigma} C_{\sigma}^b \chi_r^{\sigma} \right)}_{:=v_r^b} \cdot \underbrace{\left(\sum_{\nu} C_{\nu}^c \chi_r^{\nu} \right)}_{:=v_r^c} \cdot \underbrace{\left(\sum_{\rho} C_{\rho}^d \chi_r^{\rho} \right)}_{:=v_r^d}, \quad (3.25)$$

so that the MO integral is obtained in the CP format as well

$$\langle ab|cd \rangle = v_{cd}^{ab} \approx \sum_{r=1}^R v_r^a v_r^b v_r^c v_r^d \quad \text{or} \quad \mathbf{v}_{cd}^{ab} \approx \sum_{r=1}^R v_r^{(a)} \otimes v_r^{(b)} \otimes v_r^{(c)} \otimes v_r^{(d)}. \quad (3.26)$$

It should be noted, that it is not intended to reassemble the full 4 dimensional quantity in any of the following equations so that all calculated tensors based on MO integrals in a low-rank representation in the CP format are also obtained in the CP format. So the summation over the expansion length is never carried out and Eqn. 3.26 denotes a definition of the tensor representation in the CP format rather than a procedure to calculate the full integral tensor.

From Eqn. 3.23 the integral transformation can also be seen as simple matrix-vector multiplications of the MO coefficient matrices \mathbf{C} with the corresponding representing vectors of the decomposed two-electron integrals in the AO-basis. This procedure can be done even more efficiently as matrix-matrix multiplication with all representing vectors for one dimension. The complexity of this transformation, which is usually done in a step wise procedure with the index based tensor to avoid the formal N^8 scaling of the transformation described in Eqn. 3.21, can be reduced from $\mathcal{O}(N^4 \cdot \text{virt} + N^3 \cdot \text{virt}^2 + N^2 \cdot \text{virt}^3 + N \cdot \text{virt}^4)$ for the conventional step wise algorithm to a scaling of $\mathcal{O}(N \cdot \text{virt} \cdot R)$ in the CP tensor format, were virt denotes the number of virtual orbitals. One very important fact is that the rank does not change during the transformation. This means that the compression in the CP tensor format is independent of the basis chosen to represent the two-electron integrals. If a low-rank approximation is found starting from the canonical orbitals it should have the same rank as for localized or natural orbitals. Furthermore, after a low-rank approximation has been found in the AO basis, also a low-rank representation of the integrals in the MO basis can be obtained.

All other types of MO integrals, as they for example occur in CC calculations, can be obtained in the same way using other transformation coefficients, but the CP format allows also to construct other types of MO integrals in a very convenient way. If only the MO integrals with four occupied and four virtual indices are transformed, all other types of integrals can then be composed from the representing vectors of these two objects by simply copying them. In Eqn. 3.27 this is demonstrated for the two occupied, two virtual index integrals.

$$\left. \begin{aligned} \mathfrak{v}_{cd}^{ab} &= \sum_{r=1}^R \mathfrak{v}_r^{(a)} \otimes \mathfrak{v}_r^{(b)} \otimes v_r^{(c)} \otimes v_r^{(d)} \\ \mathfrak{v}_{ij}^{kl} &= \sum_{r=1}^R v_r^{(k)} \otimes v_r^{(l)} \otimes \mathfrak{v}_r^{(i)} \otimes \mathfrak{v}_r^{(j)} \end{aligned} \right\} \Rightarrow \mathfrak{v}_{ij}^{ab} = \sum_{r=1}^R \mathfrak{v}_r^{(a)} \otimes \mathfrak{v}_r^{(b)} \otimes \mathfrak{v}_r^{(i)} \otimes \mathfrak{v}_r^{(j)} \quad (3.27)$$

While the rank of the integral representation does not change during transformation, the occupied and the virtual space are only subspaces of the AO basis, so that the physical information the integrals carry is reduced upon transformation. Due to this fact, it should be possible to reduce the ranks of the different MO integrals even further. Nevertheless, reducing ranks of the MO integrals further may change the ranks of \mathfrak{v}_{cd}^{ab} and \mathfrak{v}_{ij}^{kl} differently, so that the convenient substitution feature to obtain mixed space integrals is lost. The mixed \mathfrak{v}_{ij}^{ab} integral can only be build up from the two-electron integrals with the same reduced AO rank and can then be compressed further by application of the rank reduction algorithm.

In order to estimate the further reduction of ranks in the MO basis different types of MO integrals are calculated from already reduced AO integrals in CP format and the rank reduction algorithm is applied to the transformed MO integral tensor. The final ranks for the molecules in the testset using the 6-31G basis set are listed in Table 3.10. Compared to the

Table 3.10: Reduced ranks (fr) for different types of MO integrals of molecules in testset using 6-31G basis set. Three different accuracies for the rank reduction are shown. For comparison the AO integrals \mathbf{v}_{AO} are also listed.

	\mathbf{v}_{AO}			\mathbf{v}_{kl}^{ij}			\mathbf{v}_{cd}^{ab}			\mathbf{v}_{ab}^{ij}		
	10^{-2}	10^{-4}	10^{-6}	10^{-2}	10^{-4}	10^{-6}	10^{-2}	10^{-4}	10^{-6}	10^{-2}	10^{-4}	10^{-6}
LiH	94	221	264	4	5	6	84	142	203	17	22	25
HF	113	225	263	27	35	45	38	54	71	45	59	66
H ₂ O	141	343	409	27	35	47	66	106	134	62	86	97
NH ₃	171	484	603	27	35	45	105	186	240	83	113	124
CH ₄	199	628	842	26	36	45	137	299	349	103	138	153
CO	274	748	979	49	75	99	136	238	269	144	206	229
N ₂	270	729	978	50	76	136	155	239	268	141	207	229
HCN	304	899	1295	50	77	103	173	367	416	169	253	285

ranks of the integrals in the AO basis it can be seen, that especially the ranks of the four occupied integral tensor (\mathbf{v}_{kl}^{ij}) can be reduced. Nevertheless, also the other two types of MO integrals have lower ranks than the corresponding AO integral. The actual reduction in the MO basis depends on the type of the integral as well as on the number of occupied and virtual orbitals.

An alternative to the scheme of casting the AO integrals into the CP format, performing the rank reduction, transform the reduced tensor to the MO basis and further compress the representation by application of the rank reduction procedure (ACRTR: AO integral casting, reduction, transformation, reduction) is to first transform the decomposed but not rank reduced AO integrals to the MO basis and only reduce the rank of the MO integrals (ACTR: AO integral casting, transformation, reduction). A further possibility is to perform the decomposition and subsequent reduction directly for the MO integrals after they have been obtained from any quantum chemistry software package (MCR: MO integral casting, reduction). Furthermore, it might be advantageous to derive schemes in which the starting point for the decomposition and rank reduction are representations of the two-electron integrals as obtained from more efficient schemes like RI or DF techniques (c.f. Sec. 3.1.3).

A comparison of the ranks and the accuracies between the transformed MO integrals build up from already reduced AO integrals (ACRTR) and the low-rank representations obtained

by decomposition and reduction of the already transformed MO integrals (MCR) for the test set of molecules using the 6-31G basis set is given in Table 3.11 for three different kinds of MO integrals. Comparing the ranks of reduced representation obtained by the MCR procedure

Table 3.11: Reduced ranks and mean absolute deviations (MAD) for (a) \mathfrak{v}_{kl}^{ij} , (b) \mathfrak{v}_{cd}^{ab} and (c) \mathfrak{v}_{ab}^{ij} in test set using 6-31G basis set. Two different schemes are compared: MCR - casting MO integrals from CFOUR and perform the rank reduction and ACRTR - cast AO integrals, reduce, transform and reduce rank in MO basis further. Note, that the same accuracy is used for the AO and MO reduction.

	(a) \mathfrak{v}_{kl}^{ij}											
	rank						MAD					
	MCR			ACRTR			MCR			ACRTR		
	10 ⁻²	10 ⁻⁴	10 ⁻⁶	10 ⁻²	10 ⁻⁴	10 ⁻⁶	10 ⁻²	10 ⁻⁴	10 ⁻⁶	10 ⁻²	10 ⁻⁴	10 ⁻⁶
LiH	4	5	5	4	5	6	5.86·10 ⁻⁴	1.72·10 ⁻⁷	1.72·10 ⁻⁷	6.05·10 ⁻⁴	1.32·10 ⁻⁵	1.19·10 ⁻⁷
HF	27	35	46	27	35	45	1.46·10 ⁻³	1.41·10 ⁻⁵	2.00·10 ⁻⁷	1.56·10 ⁻³	1.57·10 ⁻⁵	1.87·10 ⁻⁷
H ₂ O	27	36	47	27	35	47	1.35·10 ⁻³	8.61·10 ⁻⁶	1.36·10 ⁻⁷	1.32·10 ⁻³	1.73·10 ⁻⁵	1.76·10 ⁻⁷
NH ₃	27	35	47	27	35	45	1.15·10 ⁻³	1.29·10 ⁻⁵	1.16·10 ⁻⁷	7.04·10 ⁻⁴	1.47·10 ⁻⁵	1.32·10 ⁻⁷
CH ₄	26	36	45	26	36	45	6.83·10 ⁻⁴	5.59·10 ⁻⁶	1.14·10 ⁻⁷	8.03·10 ⁻⁴	6.98·10 ⁻⁶	9.80·10 ⁻⁸
CO	50	75	103	49	75	99	8.00·10 ⁻⁴	9.62·10 ⁻⁶	1.07·10 ⁻⁷	9.36·10 ⁻⁴	9.41·10 ⁻⁶	1.08·10 ⁻⁷
N ₂	51	77	93	50	76	109	7.92·10 ⁻⁴	8.93·10 ⁻⁶	1.05·10 ⁻⁷	7.94·10 ⁻⁴	9.64·10 ⁻⁶	1.16·10 ⁻⁷
HCN	50	76	97	50	77	103	6.31·10 ⁻⁴	8.71·10 ⁻⁶	9.96·10 ⁻⁸	7.32·10 ⁻⁴	7.64·10 ⁻⁶	1.03·10 ⁻⁷
avg.							9.32·10 ⁻⁴	8.59·10 ⁻⁶	1.31·10 ⁻⁷	9.32·10 ⁻⁴	1.18·10 ⁻⁵	1.30·10 ⁻⁷

	(b) \mathfrak{v}_{cd}^{ab}											
	rank						MAD					
	MCR			ACRTR			MCR			ACRTR		
	10 ⁻²	10 ⁻⁴	10 ⁻⁶	10 ⁻²	10 ⁻⁴	10 ⁻⁶	10 ⁻²	10 ⁻⁴	10 ⁻⁶	10 ⁻²	10 ⁻⁴	10 ⁻⁶
LiH	89	141	169	84	142	203	1.79·10 ⁻⁴	1.86·10 ⁻⁶	1.87·10 ⁻⁸	3.97·10 ⁻³	7.15·10 ⁻⁵	6.39·10 ⁻⁷
HF	39	55	78	38	54	71	5.43·10 ⁻⁴	5.00·10 ⁻⁶	5.98·10 ⁻⁸	8.23·10 ⁻³	7.82·10 ⁻⁶	9.14·10 ⁻⁷
H ₂ O	72	108	133	66	107	139	4.19·10 ⁻⁴	3.85·10 ⁻⁶	4.32·10 ⁻⁸	8.93·10 ⁻³	2.08·10 ⁻⁵	1.57·10 ⁻⁶
NH ₃	114	186	221	105	186	240	3.10·10 ⁻⁴	3.17·10 ⁻⁶	3.26·10 ⁻⁸	1.69·10 ⁻³	2.22·10 ⁻⁵	3.15·10 ⁻⁶
CH ₄	165	296	337	137	299	349	2.35·10 ⁻⁴	2.23·10 ⁻⁶	2.44·10 ⁻⁸	4.04·10 ⁻³	5.35·10 ⁻⁵	4.92·10 ⁻⁶
CO	148	236	263	136	238	269	2.98·10 ⁻⁴	2.95·10 ⁻⁶	2.72·10 ⁻⁸	3.27·10 ⁻³	1.54·10 ⁻⁵	1.46·10 ⁻⁶
N ₂	155	238	264	136	239	268	2.95·10 ⁻⁴	2.78·10 ⁻⁶	2.44·10 ⁻⁸	4.69·10 ⁻³	4.65·10 ⁻⁵	5.64·10 ⁻⁶
HCN	205	365	410	173	367	416	2.23·10 ⁻⁴	2.22·10 ⁻⁶	2.32·10 ⁻⁸	4.78·10 ⁻³	6.12·10 ⁻⁵	1.06·10 ⁻⁶
avg.							3.13·10 ⁻⁴	3.01·10 ⁻⁶	3.17·10 ⁻⁸	4.95·10 ⁻³	3.74·10 ⁻⁵	2.42·10 ⁻⁶

	(c) \mathfrak{v}_{ab}^{ij}											
	rank						MAD					
	MCR			ACRTR			MCR			ACRTR		
	10 ⁻²	10 ⁻⁴	10 ⁻⁶	10 ⁻²	10 ⁻⁴	10 ⁻⁶	10 ⁻²	10 ⁻⁴	10 ⁻⁶	10 ⁻²	10 ⁻⁴	10 ⁻⁶
LiH	15	21	24	17	22	25	2.99·10 ⁻⁵	7.92·10 ⁻⁸	8.75·10 ⁻¹⁰	5.97·10 ⁻⁴	1.02·10 ⁻⁵	9.60·10 ⁻⁸
HF	39	49	66	45	59	66	1.53·10 ⁻⁴	1.19·10 ⁻⁶	1.65·10 ⁻⁸	7.34·10 ⁻⁴	1.55·10 ⁻⁵	1.94·10 ⁻⁷
H ₂ O	55	85	95	62	86	97	1.11·10 ⁻⁴	8.68·10 ⁻⁷	7.47·10 ⁻⁹	1.13·10 ⁻⁴	2.26·10 ⁻⁵	2.11·10 ⁻⁷
NH ₃	74	111	123	83	113	124	8.58·10 ⁻⁵	6.95·10 ⁻⁷	6.43·10 ⁻⁹	1.15·10 ⁻⁴	2.31·10 ⁻⁵	2.15·10 ⁻⁷
CH ₄	84	137	153	103	138	153	6.47·10 ⁻⁵	6.30·10 ⁻⁷	5.54·10 ⁻⁹	8.03·10 ⁻⁴	2.82·10 ⁻⁵	2.44·10 ⁻⁷
CO	119	207	232	144	206	229	6.91·10 ⁻⁵	6.48·10 ⁻⁷	6.31·10 ⁻⁹	6.10·10 ⁻⁴	1.99·10 ⁻⁵	1.82·10 ⁻⁷
N ₂	121	206	230	142	207	229	7.05·10 ⁻⁵	6.74·10 ⁻⁷	5.83·10 ⁻⁹	6.25·10 ⁻⁴	3.80·10 ⁻⁵	3.40·10 ⁻⁷
HCN	137	254	284	169	253	285	5.61·10 ⁻⁵	5.25·10 ⁻⁷	5.46·10 ⁻⁹	5.00·10 ⁻⁴	3.32·10 ⁻⁵	4.11·10 ⁻⁷
avg.							8.00·10 ⁻⁵	6.64·10 ⁻⁷	6.81·10 ⁻⁹	5.12·10 ⁻⁴	2.38·10 ⁻⁵	2.37·10 ⁻⁷

and by the ACRTR method one can see, that the ranks are more or less the same. There are some deviations especially for the lowest accuracy ($\varepsilon = 10^{-2}$), where depending on the type of integral the ranks obtained by the ACRTR procedure are a little bit lower or higher. Nevertheless, for higher accuracies these deviations vanish and the obtained ranks are almost the same. Comparing the accuracies of the low-rank approximation (measured as the mean absolute deviation for all elements in the tensor) it can be seen, that the integral tensors obtained by the MCR procedure are a bit more accurate those from the ACRTR method, where the accuracy is typically one order of magnitude lower. However, the MAD is still approximately two orders of magnitude lower than the given threshold for the CP rank reduction. Therefore, it can be concluded, that there is virtually no difference in decomposing and reducing already transformed MO integrals or a further reduction of CP transformed MO integrals based upon a low-rank approximation of the AO integrals.

As mentioned before the ranks do not change due to the transformation but it is nevertheless possible to reduce the rank of the MO integrals further. As already seen for the AO integrals the most important and size determining quantity is the reduced rank and especially interesting is the scaling of the ranks with system and basis set size. In order to estimate the scaling behaviour different types of MO integrals for growing LiH chains are taken as an example. The MO integrals are cast and reduced according to the MCR procedure and the scaling with system size is shown in Figure 3.16.

Here it can be seen, that approximate low-rank representations for different kinds of MO integrals exhibit almost linear up to almost cubic scaling of reduced ranks with system size. Especially the \mathbf{v}_{ij}^{ab} integral tensor exhibits low scaling with increasing system size for larger threshold values. Other types of integrals show a scaling between $\mathcal{O}(n^{2.8})$ for \mathbf{v}_{cd}^{ab} with $\varepsilon = 10^{-6}$ and $\mathcal{O}(n^{1.2})$ for \mathbf{v}_{ij}^{ab} with $\varepsilon = 10^{-2}$. To check for consistency, also a series of alkyl chains is calculated that show practically the same behaviour (see Fig. 3.17). There the scaling is between $\mathcal{O}(n^{2.6})$ for \mathbf{v}_{ij}^{ab} with $\varepsilon = 10^{-6}$ and $\mathcal{O}(n^{1.6})$ for \mathbf{v}_{ij}^{ab} with $\varepsilon = 10^{-2}$. The scaling of ranks for the four occupied integral is in between and varies between $\mathcal{O}(n^{1.9})$ and $\mathcal{O}(n^{2.3})$.

To investigate also the scaling with basis set size the water molecule is again taken as an example. The resulting scaling for two types of MO integrals is shown in Figure 3.18. It can be seen, that especially the \mathbf{v}_{ij}^{ab} type of integral shows almost linear scaling with increasing basis set size. On the other hand the four virtual integral tensor exhibits a much steeper scaling in the order of $\mathcal{O}(n^{2.5})$. It should be noted, that for the described reductions only the small cases are treated with a full rank reduction, while for larger examples (with initial ranks larger than 5000) a sliced reduction with adaptive slice size based on SVD is performed. After the sliced reductions a subsequent full reduction is performed if the ranks are smaller than 5000. For all larger cases (especially \mathbf{v}_{cd}^{ab} tensors or when using approximations with $\varepsilon = 10^{-6}$) it is currently not possible to consider a full reduction so that only the results from sliced reductions are considered. Therefore, the higher scaling for cases with high accuracy and for

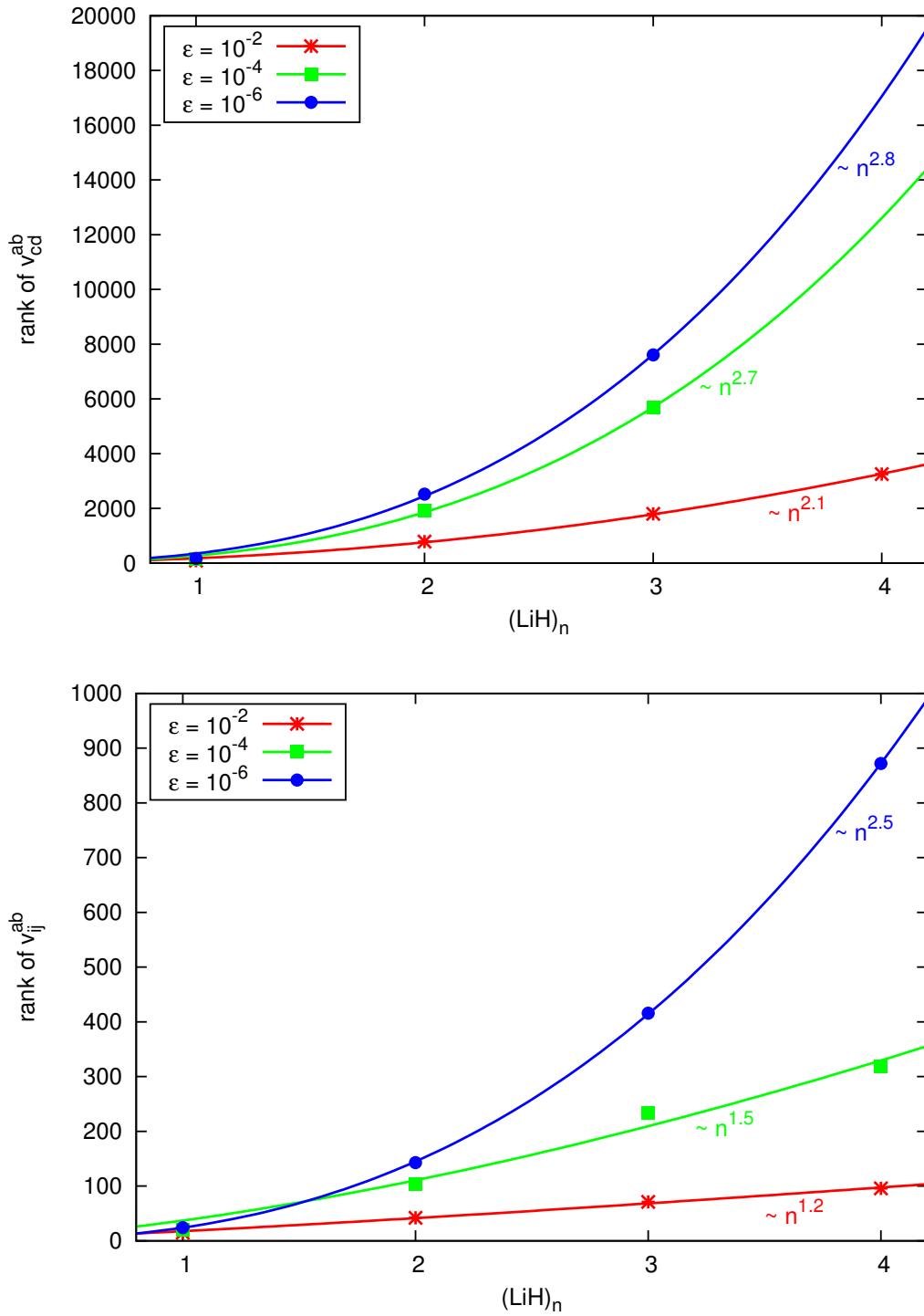


Figure 3.16: Scaling of reduced ranks of MO integrals with system size in LiH chains calculated using the 6-31G basis set. For the small examples ($n = 1 - 2$) a single full reduction (fr) is used, while the larger chains are reduced in slices (size determined by SVD casting procedure) followed by a subsequent full reduction (sfr).

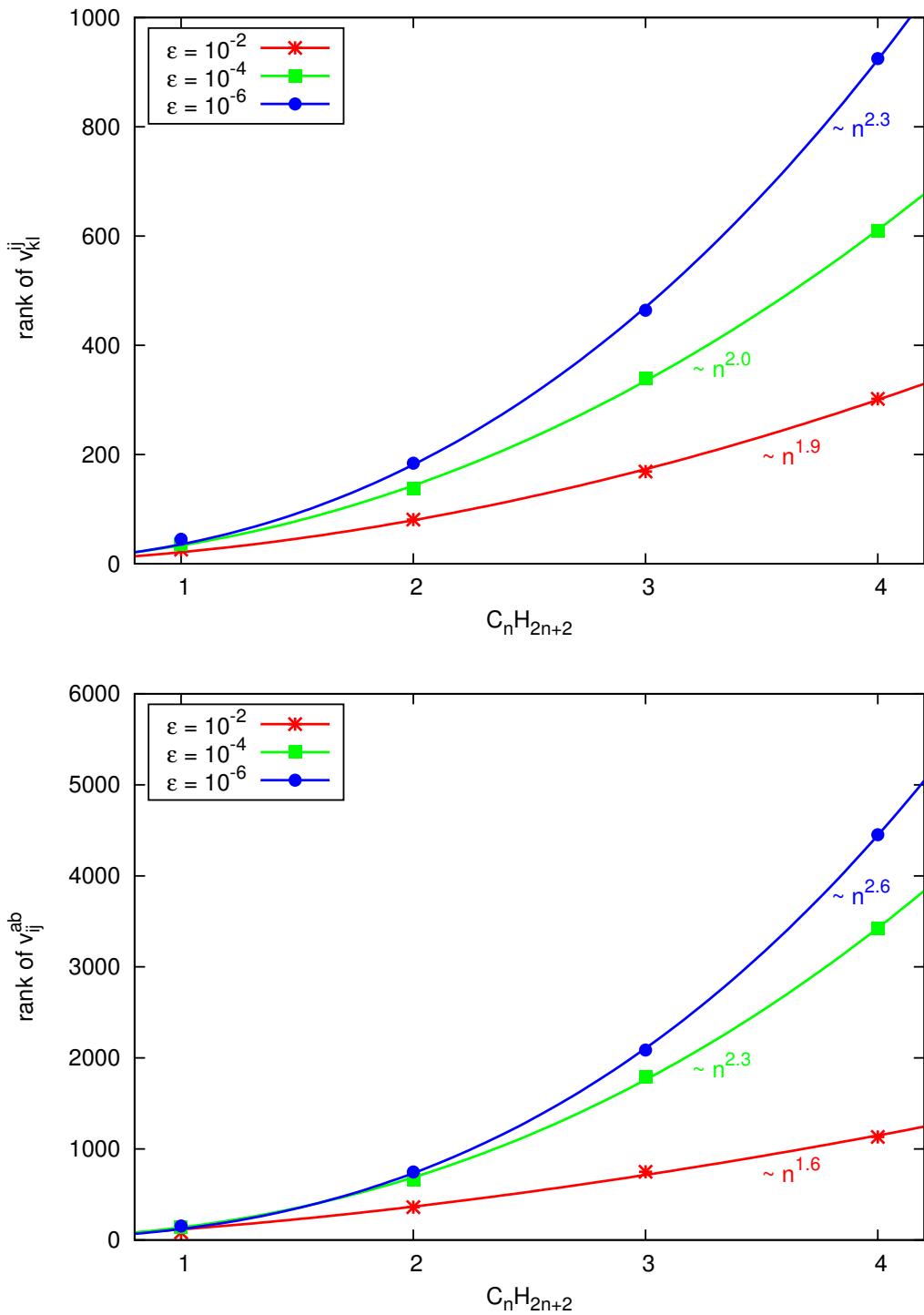


Figure 3.17: Scaling of reduced ranks of MO integrals with system size in alkyl chains ($C_n H_{2n+2}$) calculated using the 6-31G basis set. For the small examples ($n = 1 - 2$) a single full reduction (fr) is used, while the larger chains are reduced in slices (size determined by SVD) followed by a subsequent full reduction (sfr).

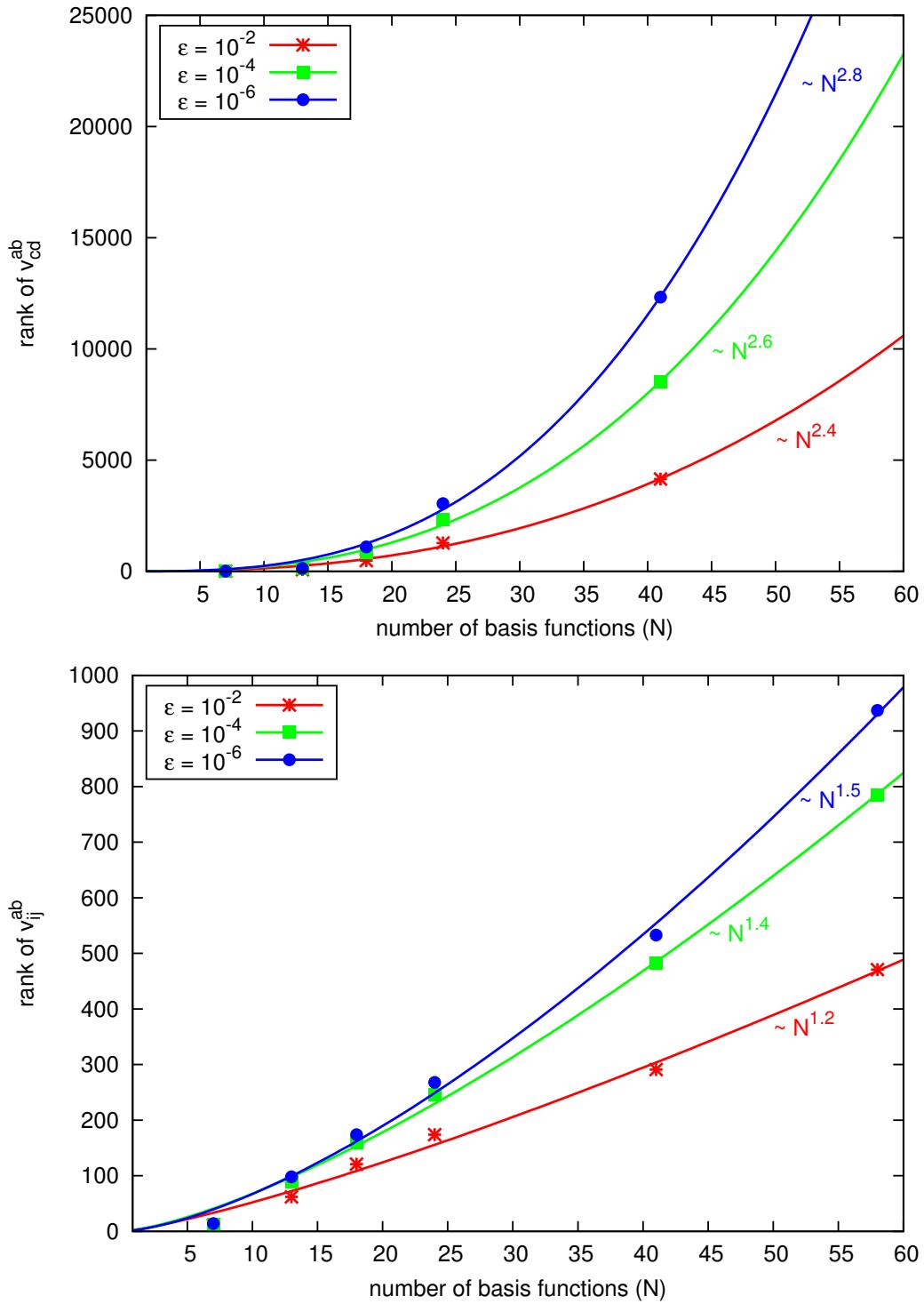


Figure 3.18: Scaling of reduced ranks of MO integrals with increasing basis set size for water using the basis set series: STO-3G, 6-31G, 6-31G*, cc-pVDZ, aug-cc-pVDZ, cc-pVTZ. For v_{ij}^{ab} full reductions are carried out, while for the larger basis sets the v_{cd}^{ab} integral tensors are reduced with the *sr* procedure (slice size determined by SVD). There the ranks are still too large for a subsequent full reduction.

the v_{cd}^{ab} tensor can be expected to have a somehow lower scaling of the full reduced ranks, although it has been shown that a final full reduction mainly reduces the fitted prefactor (see Tab. 3.8).

It can be summarized (see also Tab. 3.12) that the reduced ranks in the CP format for the MO integrals scale roughly between $\mathcal{O}(n^{1.2})$ to $\mathcal{O}(n^{2.8})$ with system size and approximately also between $\mathcal{O}(N^{1.2})$ to $\mathcal{O}(N^{2.8})$ with increasing basis set size, depending on the type of MO integral and the chosen accuracy threshold. There, especially the lowest accuracy ($\varepsilon = 10^{-2}$) leads to almost linear scaling for some type of integrals. From this it can be concluded that the amount of storage for the MO integrals can be decreased from $\mathcal{O}(N^4)$ scaling to optimally $\mathcal{O}(N^{2.2})$. Therefore, especially the bottleneck in storing the v_{cd}^{ab} can be moderated by using the new CP tensor format in a low-rank approximation, although the scaling is a bit steeper for this type of integral leading to a scaling of memory requirements in the order of $\mathcal{O}(N^{3.4})$. It should be noted that the overall scaling of the number of parameters for representing the two-electron integrals with approximately $\mathcal{O}(N^2)$ to $\mathcal{O}(N^3)$. This is similar to RI/DF and CD, where the representing matrices are of size $N^2 \cdot R$. For CD, for example, it has been found that the rank scales approximately linear with systems size if a stable error is maintained [230]. For the CP format on the other hand, the size of the representing vectors are with $N \cdot R$ of lower dimensionality with respect to N , while the rank itself exhibits a steeper scaling with system size.

Table 3.12: Scaling parameter b for fitted functions $f(x) = a \cdot x^b$ of MO integrals and amplitudes (c.f. Sec. 3.6) using adapted slice size reduction followed by full reduction (*sfr*) or simple full reduction (*fr*) when possible.

	ε	v_{ij}^{ab}	v_{cd}^{ab}	v_{kl}^{ij}	t_{ij}^{ab}
system size n	10^{-2}	1.2-1.6	2.1	1.9	1.3
	10^{-4}	1.5-2.3	2.7	2.0	1.9
	10^{-6}	2.5-2.6	2.8	2.3	2.4
basis set size N	10^{-2}	1.2	2.4	–	1.3
	10^{-4}	1.4	2.6	–	1.9
	10^{-6}	1.5	2.8	–	2.4

3.6 MP2 Algorithm with Integrals in CP Format

3.6.1 Definition of MP2 Amplitudes

While discussing a MP2 algorithm in the following, this should be regarded as a first step towards the application of tensor representation in the CP format for CI or CC theory. The greatest benefit of low-dimensional, low-rank approximations is expected for higher order CC methods or FCI, for which the computational effort is prohibitive for larger applications due to the manipulation of tensors with six, eight or more dimensions.

In the framework of CC theory MP2 can be considered to yield a first-order estimate of the t_2 amplitudes [175]. Therefore, the MP2 energy equation is reformulated in order to obtain a definition for t_2 amplitudes

$$E_{\text{MP2}} = \frac{1}{4} \sum_{efmn} \frac{|\langle ef \parallel mn \rangle|^2}{-\epsilon_e - \epsilon_f + \epsilon_m + \epsilon_n} \quad (3.28)$$

$$= \frac{1}{4} \sum_{efmn} \langle ef \parallel mn \rangle \cdot \underbrace{\frac{\langle ef \parallel mn \rangle}{-\epsilon_e - \epsilon_f + \epsilon_m + \epsilon_n}}_{:= t_{ij}^{ab}} \quad (3.29)$$

$$= \frac{1}{4} \sum_{efmn} \langle ef \parallel mn \rangle \cdot t_{mn}^{ef}. \quad (3.30)$$

There, the antisymmetrized two-electron integrals are weighted by the energy denominator containing the orbital energies obtained from a previous Hartree-Fock calculation. This ansatz is convenient from a formal point of view and also allows to estimate the performance of the CP tensor format for post Hartree-Fock methods like CC.

As presented in the previous section, the MO integrals can be obtained in a low-rank CP representation if the low-rank representation of the AO integrals is transformed to the MO basis. As this transformation can also be seen as a projection into a subspace the ranks can be further reduced although the rank stays constant upon the integral transformation

$$\langle ab \parallel ij \rangle = v_{ij}^{ab} \approx \sum_{r=1}^R v_r^{(a)} \otimes v_r^{(b)} \otimes v_r^{(i)} \otimes v_r^{(j)}. \quad (3.31)$$

It should be noted, that here the antisymmetrized two-electron integrals are used in a low-rank representation. To get also a CP representation of the t_2 amplitudes according to Eqn. 3.29 the energy denominator $D_{ij}^{ab} = \frac{1}{-\epsilon_a - \epsilon_b + \epsilon_i + \epsilon_j}$ has to be cast into a product format as well. For this purpose the approximation of $1/x$ through Laplace transformation [41, 42] or by approximation with exponential sums [231] is applied²

$$\frac{1}{-\epsilon_a - \epsilon_b + \epsilon_i + \epsilon_j} = \int_0^\infty e^{-(\epsilon_a + \epsilon_b - \epsilon_i - \epsilon_j) \cdot y} dy. \quad (3.32)$$

²Strictly speaking the energy denominator has to be multiplied with -1, as the Laplace transformation is only valid in the range $[0, \infty[$.

The resulting integral can be evaluated by a numerical quadrature scheme [231, 232]

$$\int_0^\infty e^{-(\epsilon_a - \epsilon_b + \epsilon_i + \epsilon_j) \cdot y} dy \approx \sum_{s=1}^S \omega_s \cdot e^{(-\alpha_s(\epsilon_a - \epsilon_b + \epsilon_i + \epsilon_j))} \quad (3.33)$$

$$\approx \sum_{s=1}^S \omega_s \cdot e^{(\alpha_s \epsilon_a)} \cdot e^{(\alpha_s \epsilon_b)} \cdot e^{(-\alpha_s \epsilon_i)} \cdot e^{(-\alpha_s \epsilon_j)}, \quad (3.34)$$

and the factors ω_s and α_s are available from the literature [231]. The accuracy of this approximation, like in the Laplace MP2 method [40, 42–44, 231–233], can be adjusted by choosing an appropriate threshold which results in a fixed decomposition rank for the denominator expression [231]. Furthermore, it has been shown that the number of quadrature points and therefore the expansion length S is almost independent of the system size and typically smaller than 10 [42].

From Eqn. 3.34 it can immediately be seen, that the energy denominator is obtained in the CP format as well

$$D_{ij}^{ab} \approx \sum_{s=1}^S \underbrace{\omega_s \cdot e^{(\alpha_s \epsilon_a)}}_{:=\varepsilon_s^a} \cdot \underbrace{e^{(\alpha_s \epsilon_b)}}_{:=\varepsilon_s^b} \cdot \underbrace{e^{(-\alpha_s \epsilon_i)}}_{:=\varepsilon_s^i} \cdot \underbrace{e^{(-\alpha_s \epsilon_j)}}_{:=\varepsilon_s^j} \quad (3.35)$$

$$\approx \sum_{s=1}^S \varepsilon_s^a \varepsilon_s^b \varepsilon_s^i \varepsilon_s^j, \quad (3.36)$$

and can be written with the help of Kronecker products, when the representation of the whole tensor \mathfrak{D}_{ij}^{ab} is needed

$$\mathfrak{D}_{ij}^{ab} \approx \sum_{s=1}^S \varepsilon_s^{(a)} \otimes \varepsilon_s^{(b)} \otimes \varepsilon_s^{(i)} \otimes \varepsilon_s^{(j)}. \quad (3.37)$$

Now the decomposed amplitudes can be combined with the decomposed energy denominator according to Eqn. 3.29 as

$$t_{ij}^{ab} = \langle ab || ij \rangle \cdot D_{ij}^{ab}. \quad (3.38)$$

Using the decomposed representation of the integrals and the energy denominator leads to

$$\mathfrak{t}_{ij}^{ab} \approx \mathfrak{v}_{ij}^{ab} \mathfrak{D}_{ij}^{ab} \quad (3.39)$$

$$\approx \left(\sum_{r=1}^R v_r^{(a)} \otimes v_r^{(b)} \otimes v_r^{(i)} \otimes v_r^{(j)} \right) \cdot \left(\sum_{s=1}^S \varepsilon_s^{(a)} \otimes \varepsilon_s^{(b)} \otimes \varepsilon_s^{(i)} \otimes \varepsilon_s^{(j)} \right). \quad (3.40)$$

It is now possible to arrange all representing vectors for the same dimension (same superscript) together

$$\mathfrak{t}_{ij}^{ab} \approx \sum_{r=1}^R \sum_{s=1}^S \underbrace{\left(v_r^{(a)} \cdot \varepsilon_s^{(a)} \right)}_{:=t_q^{(a)}} \otimes \underbrace{\left(v_r^{(b)} \cdot \varepsilon_s^{(b)} \right)}_{:=t_q^{(b)}} \otimes \underbrace{\left(v_r^{(i)} \cdot \varepsilon_s^{(i)} \right)}_{:=t_q^{(i)}} \otimes \underbrace{\left(v_r^{(j)} \cdot \varepsilon_s^{(j)} \right)}_{:=t_q^{(j)}}. \quad (3.41)$$

Table 3.13: Reduced ranks of MP₂ t_2 amplitudes and the corresponding two-electron integral for molecules in the testset using 6-31G basis set.

	\mathfrak{v}_{ij}^{ab}			\mathfrak{t}_{ij}^{ab}		
	10^{-2}	10^{-4}	10^{-6}	10^{-2}	10^{-4}	10^{-6}
LiH	17	22	25	15	21	24
HF	45	59	66	32	59	67
H ₂ O	62	86	97	47	84	96
NH ₃	83	113	124	65	111	124
CH ₄	103	138	153	78	138	155
CO	144	206	229	99	203	232
N ₂	141	207	229	98	203	232
HCN	169	253	285	113	249	288

In order to get the normal representation of the amplitudes in CP format the two ranks have to be combined to a single rank. Therefore, the summation over r and s are combined by multiplication to the new expansion length q and the representing vectors for each dimension have to be multiplied element by element with each other. This way the final representation has an increased rank $Q = R \cdot S$

$$\mathfrak{t}_{ij}^{ab} \approx \sum_{q=1}^{Q=R \cdot S} t_q^{(a)} \otimes t_q^{(b)} \otimes t_q^{(i)} \otimes t_q^{(j)}, \quad (3.42)$$

and the amplitudes are obtained in the CP representation of the tensor. The initial rank Q of the amplitudes is only determined by the rank of the MO integrals R and the rank of the energy denominator S , which is practically a constant for all given molecular systems under consideration. In consequence, the initial rank for the amplitudes is always larger than the rank of the two-electron integrals by a constant factor. However, due to the fact that the t_2 amplitudes are obtained from the MO integrals by weighting with the energy denominator, the optimal rank of t_2 should be approximately the same or even smaller than the rank of the MO integrals. Therefore, it should be possible to reduce the initial rank of t_2 by applying the rank-reduction algorithm presented in Sec. 3.2.

For the approximation of the energy denominator currently a very high accuracy is used and a rank of 42 is chosen, for which an error in the ℓ^2 -norm of the energy denominator of $\varepsilon = 7 \cdot 10^{-12}$ is obtained [231]. As already mentioned the initial rank of the t_2 amplitudes is always larger than the rank of the corresponding \mathfrak{v}_{ij}^{ab} integral tensor due to the construction of this object, so it must be possible to reduce the initial rank of \mathfrak{t}_{ij}^{ab} to similar sizes. For the molecules in the testset the reduced ranks of the t_2 amplitude tensor and for comparison also the ranks of the corresponding two-electron integral are listed in Table 3.13.

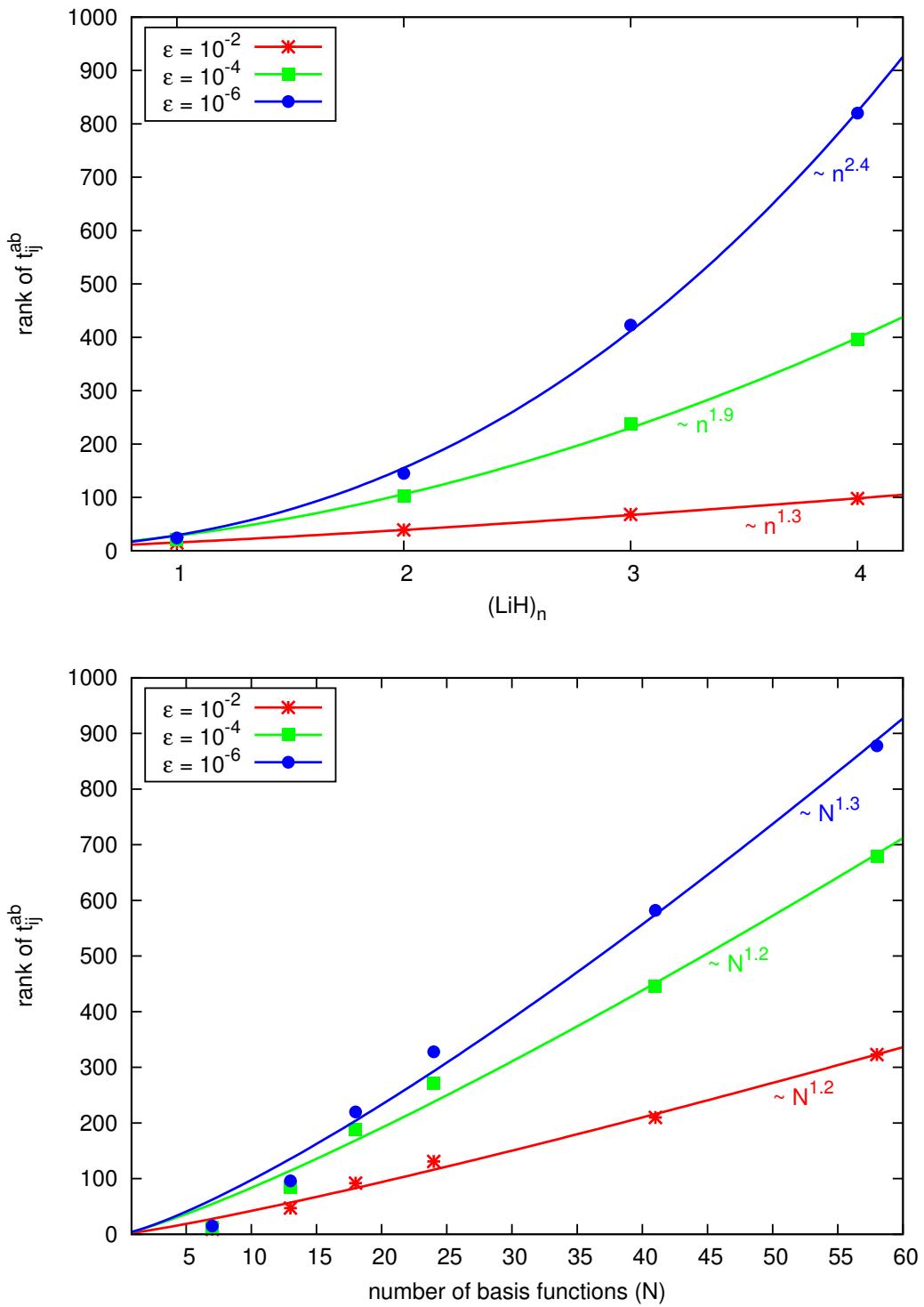


Figure 3.19: Scaling of reduced ranks for the t_2 amplitudes (obtained from a standard calculation as weighted integrals) with system size for $(\text{LiH})_n$ and scaling with basis set size for H_2O . For the small examples (initial rank < 5000) a full reduction (*fr*) is used, while the larger examples are reduced in slices (size based on SVD) followed by a subsequent full reduction (*sfr*).

Here it can be seen that the ranks of t_{ij}^{ab} are practically the same as the ranks of v_{ij}^{ab} especially when high accuracies are used for the rank reduction procedure. Using a higher threshold parameter ($\varepsilon = 10^{-2}$) the ranks of the t_2 amplitudes can be reduced even further compared to the ranks of the integral tensor, due to the fact that the entries in t_2 may become very small and therefore be represented less accurate for too large threshold values.

In Figure 3.19 the scaling of the final ranks of MP2 amplitudes with system size for growing LiH chains and with the basis set size for the H₂O molecule are given.

As already mentioned, the rank of the t_2 amplitudes can be reduced to similar or even lower values than the ranks of the corresponding MO integrals. For the lowest accuracy ($\varepsilon = 10^{-2}$) the scaling with system and basis set size is almost linear ($\mathcal{O}(n^{1.3})$ and $\mathcal{O}(N^{1.2})$, respectively). This means that the number of parameters for the t_2 tensor can be decreased from a format scaling with $\mathcal{O}(N^4)$ to $\mathcal{O}(N^{2.2})$. Especially the low scaling with basis set size should be pointed out, as other approximations, for example local correlation methods, that exhibit low scaling with system size often do not show advantageous scaling with basis set size.

To test if a reduced CP representation of the amplitude tensor captures local features of the electron correlation a dissociating O₂ molecule is taken as an example. It should be noted, that MP2 behaves erratic upon bond breaking and thus is only regarded as a simple

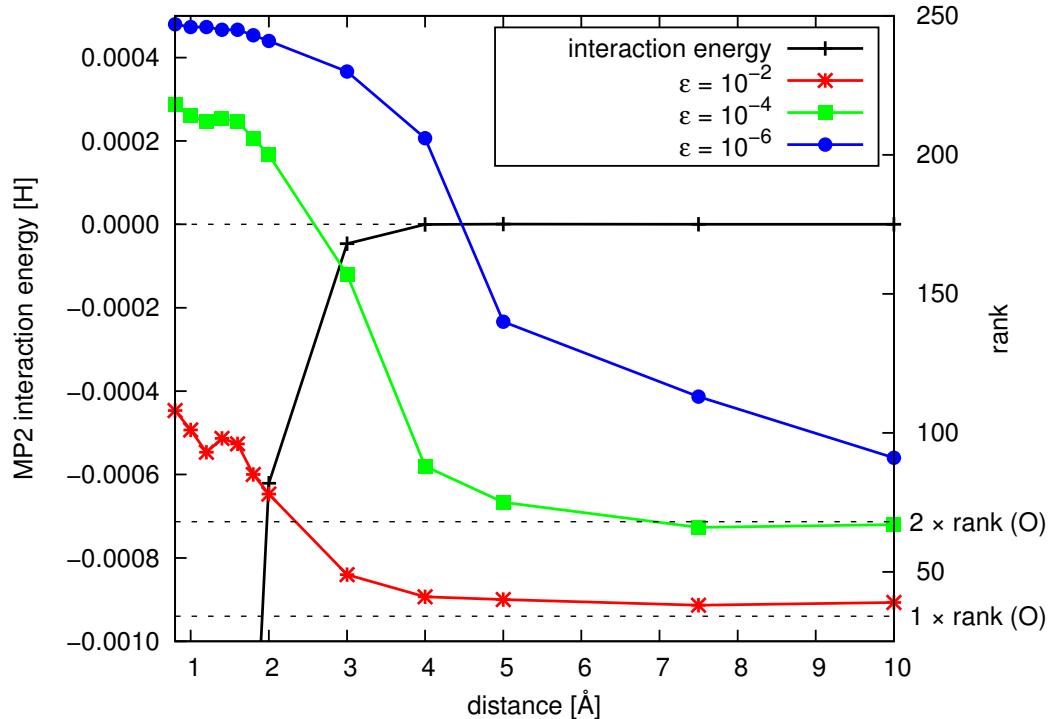


Figure 3.20: Reduced ranks for MP2 t_2 amplitudes in O₂ (using 6-31G basis set) with increasing bond length.

computational example. The t_2 amplitude tensor as obtained from MP2 (based on UHF wave function) calculated using 6-31G basis set are cast in the CP format and the ranks are reduced. Figure 3.20 shows the reduced rank for increasing distance between the two oxygen molecules. For comparison also the interaction energy is plotted. It can be seen, that the reduced ranks of the MP2 t_2 amplitudes immediately decrease as the interaction energy tends to zero. For the largest chosen threshold ($\varepsilon = 10^{-2}$) the rank can be decreased to values similar to the rank for a single oxygen atom for distances larger than 4 Å, while for the larger accuracy (10^{-4}) two times the rank for a single oxygen atom is obtained. It should be noted, that the rank of the MP2 t_2 amplitude tensor for a single oxygen atom is almost independent of the chosen reduction threshold. For the largest tested accuracy (10^{-6}) one can still see a significant decrease in the rank but even at a distance of 10 Å the rank still seems not to be converged with respect to further increasing distance. Nevertheless, one can summarize that the rank reduction algorithm is able to identify the local features of electron correlation and it seems that also the symmetry is partly recognized, as the two oxygen atoms at distances fare apart can be described with approximately the same number of parameters as a single oxygen atom.

3.6.2 MP2 Energy Equation

The MP2 energy using t_2 and the MO-integrals in the CP tensor format can be calculated according to Eqn. 3.30 as inner product of t_{ij}^{ab} and v_{ij}^{ab} (denoted as t_{mn}^{ef} and v_{ef}^{mn} to account for the internal summation indices)

$$E_{\text{MP2}} = \frac{1}{4} t_{mn}^{ef} v_{mn}^{ef} \quad (3.43)$$

$$= \frac{1}{4} \left(\sum_{q=1}^Q t_q^{(e)} \otimes t_q^{(f)} \otimes t_q^{(m)} \otimes t_q^{(n)} \right) \cdot \left(\sum_{r=1}^R v_r^{(e)} \otimes v_r^{(f)} \otimes v_r^{(m)} \otimes v_r^{(n)} \right). \quad (3.44)$$

Now representing vectors for the same internal contraction dimension (denoted with same superscript) can be grouped together and written as scalar products over the representing vectors

$$E_{\text{MP2}} = \frac{1}{4} \sum_{q=1}^Q \sum_{r=1}^R \left(t_q^{(e)} \cdot v_r^{(e)} \right) \otimes \left(t_q^{(f)} \cdot v_r^{(f)} \right) \otimes \left(t_q^{(m)} \cdot v_r^{(m)} \right) \otimes \left(t_q^{(n)} \cdot v_r^{(n)} \right) \quad (3.45)$$

$$= \frac{1}{4} \sum_{q=1}^Q \sum_{r=1}^R \langle t_q^{(e)}, v_r^{(e)} \rangle \otimes \langle t_q^{(f)}, v_r^{(f)} \rangle \otimes \langle t_q^{(m)}, v_r^{(m)} \rangle \otimes \langle t_q^{(n)}, v_r^{(n)} \rangle. \quad (3.46)$$

Thus, only scalar products of the different representing vectors of t_{ij}^{ab} and v_{ij}^{ab} have to be evaluated. If the representing vectors in Eqn. 3.45 are written in matrix notation one can immediately see, that the evaluation of the MP2 energy can be done by four matrix-matrix

multiplications and the final summation over the two expansion lengths

$$E_{\text{MP2}} = \frac{1}{4} \sum_{q=1}^Q \sum_{r=1}^R \underbrace{\left(\sum_e t_q^e \cdot v_r^e \right)}_{:=E_{qr}} \cdot \underbrace{\left(\sum_f t_q^f \cdot v_r^f \right)}_{:=F_{qr}} \cdot \underbrace{\left(\sum_m t_q^m \cdot v_r^m \right)}_{:=M_{qr}} \cdot \underbrace{\left(\sum_n t_q^n \cdot v_r^n \right)}_{:=N_{qr}} \quad (3.47)$$

$$= \frac{1}{4} \sum_{q=1}^Q \sum_{r=1}^R E_{qr} \cdot F_{qr} \cdot M_{qr} \cdot N_{qr} \quad (3.48)$$

Equations 3.46 or 3.48 denote the first steps where the summation over the expansion length is actually carried out, because the energy is evaluated as all inner products of the amplitude and integral tensor. All other needed intermediate tensors, like the energy denominator, are kept in the CP format and a reassembly of the corresponding full four dimensional objects into the index based representation is not necessary. Even more, the reassembly step is prohibitive especially for iterative procedures dealing with high order tensors (like solving non linear CC amplitude equations), as the resulting high dimensional quantities have to be stored and would have to be decomposed into the CP format again for further usage.

Evaluating the MP2 energy expression in the CP format scales formally as

$$\mathcal{O}(Q \cdot R \cdot (\text{virt} + \text{virt} + \text{occ} + \text{occ})) \approx \mathcal{O}(Q \cdot R \cdot N). \quad (3.49)$$

Overall, the complexity of the outlined MP2 algorithm in the CP format scales linear with system size and linear with the rank of the two involved tensors. As a consequence, the scaling of the rank with system and basis set size and the scaling of the rank reduction algorithm itself determine the actual scaling of the algorithm.

For any application of the tensor representation technique presented here the error in the total energy and its dependence on the value of the ε -threshold is important. This parameter has an impact on the overall scaling of the ranks of involved tensors and therefore also on the overall scaling of the method itself. Thus, the MP2 energy is calculated with all tensors in the CP format as explained before. Table 3.14 shows the absolute errors in the MP2 energy for the test molecules calculated using the 6-31G basis set. Therefore, the reduced AO integrals are transformed to the MO basis and the rank of the v_{ij}^{ab} integral is reduced again. With a decomposed energy denominator (fixed rank = 42) also the amplitudes are obtained in the CP format and the rank is reduced further. So the energies in Table 3.14 are calculated from multiple reduced quantities. It should be noted, that the threshold for each rank reduction - from AO integral to MO integral and the final reduction of the t_2 amplitudes - is always the same.

From Table 3.14 it can be seen, that the averaged error in the correlation energy is always lower than the given accuracy parameter for the rank reduction procedures. Especially for

Table 3.14: Error in calculated MP2 correlation energy using tensors in CP format ($\Delta E_{\text{MP2-CP}}$) for 6-31G basis set. The used tensors are obtained from reduced AO integrals by transformation and further reduction in the MO basis. Also the amplitudes are recompressed. Note that for all reduction procedures the same accuracy parameters are used. For comparison the MP2 energy (E_{MP2}) calculated with index based tensors is listed. As a summary the mean deviation, the mean absolute deviation (MAD) and the root mean square error (RMS) are given.

	standard MP2 E_{MP2}	$\Delta E_{\text{MP2-CP}}$		
		$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-6}$
LiH	$-1.26022296 \cdot 10^{-2}$	$8.194 \cdot 10^{-4}$	$-9.750 \cdot 10^{-8}$	$6.822 \cdot 10^{-9}$
HF	$-1.28683539 \cdot 10^{-1}$	$-2.114 \cdot 10^{-3}$	$-2.681 \cdot 10^{-6}$	$-3.175 \cdot 10^{-8}$
H ₂ O	$-1.28795560 \cdot 10^{-1}$	$-1.688 \cdot 10^{-3}$	$-8.102 \cdot 10^{-7}$	$6.219 \cdot 10^{-9}$
NH ₃	$-1.16883445 \cdot 10^{-1}$	$-1.331 \cdot 10^{-3}$	$8.330 \cdot 10^{-7}$	$-2.583 \cdot 10^{-8}$
CH ₄	$-9.96461521 \cdot 10^{-2}$	$-1.049 \cdot 10^{-3}$	$1.327 \cdot 10^{-6}$	$-3.997 \cdot 10^{-9}$
CO	$-2.12227542 \cdot 10^{-1}$	$-1.469 \cdot 10^{-3}$	$4.424 \cdot 10^{-6}$	$4.627 \cdot 10^{-8}$
N ₂	$-2.38694295 \cdot 10^{-1}$	$1.134 \cdot 10^{-3}$	$-1.790 \cdot 10^{-6}$	$1.426 \cdot 10^{-8}$
HCN	$-2.14006155 \cdot 10^{-1}$	$-8.368 \cdot 10^{-4}$	$-4.772 \cdot 10^{-6}$	$7.883 \cdot 10^{-8}$
mean		$-8.168 \cdot 10^{-4}$	$-4.457 \cdot 10^{-7}$	$1.135 \cdot 10^{-8}$
MAD		$1.305 \cdot 10^{-3}$	$2.092 \cdot 10^{-6}$	$2.675 \cdot 10^{-8}$
RMS		$1.370 \cdot 10^{-3}$	$2.642 \cdot 10^{-6}$	$3.594 \cdot 10^{-8}$

highly accurate approximations ($\varepsilon = 10^{-6}$) the error in the correlation energy is approximately two orders of magnitude lower than the given error in the ℓ^2 -norm. For the larger threshold ($\varepsilon = 10^{-2}$) the error is still approximately one order of magnitude lower than the error introduced by the reduction and the mean absolute deviation of the energy is in the order of one μ Hartree. Furthermore the successive application of the rank reduction seems to be fairly stable, but there might be some error propagation.

To prove the stability of multiple application of the rank reduction algorithm the MP2 correlation energy is calculated in a second way: the MO integrals and MP2 t_2 amplitudes are obtained by a standard calculation, decomposed by trivial decomposition and the ranks are reduced in two distinct reduction procedures (MCR). This way the error especially in the amplitudes should be smaller, as the exact amplitude tensor is approximated instead of a tensor that is build from already approximated tensors. The resulting errors in the MP2 correlation energy are listed in Table 3.15. There it can be seen that for the case of reducing conventionally obtained amplitudes and integrals the accuracy relative to the full MP2 result is slightly better than for successive application of the rank reduction for quantities build from reduced AO integrals. The mean absolute deviations in the MP2 correlation energy are now always two orders of magnitude lower than the given accuracy of the rank reduction

Table 3.15: Error in calculated MP2 correlation energy using tensors in CP format ($\Delta E_{\text{MP2-CP}}$) for 6-31G basis set. The used tensors are obtained by trivial decomposition and rank reduction of the quantities v_{ij}^{ab} and t_{ij}^{ab} (equivalent to MCR procedure). For comparison the MP2 energy (E_{MP2}) calculated with index based tensors is listed. As a summary the mean deviation, the mean absolute deviation (MAD) and the root mean square error (RMS) are given.

	standard MP2 E_{MP2}	$\Delta E_{\text{MP2-CP}}$		
		$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-6}$
LiH	$-1.26022296 \cdot 10^{-2}$	$1.081 \cdot 10^{-5}$	$-7.739 \cdot 10^{-8}$	$2.996 \cdot 10^{-11}$
HF	$-1.28683539 \cdot 10^{-1}$	$9.094 \cdot 10^{-5}$	$2.471 \cdot 10^{-7}$	$-4.140 \cdot 10^{-9}$
H ₂ O	$-1.28795560 \cdot 10^{-1}$	$9.166 \cdot 10^{-5}$	$-4.486 \cdot 10^{-7}$	$3.029 \cdot 10^{-9}$
NH ₃	$-1.16883445 \cdot 10^{-1}$	$5.804 \cdot 10^{-5}$	$8.204 \cdot 10^{-8}$	$-3.034 \cdot 10^{-9}$
CH ₄	$-9.96461521 \cdot 10^{-2}$	$4.611 \cdot 10^{-5}$	$9.612 \cdot 10^{-8}$	$1.631 \cdot 10^{-9}$
CO	$-2.12227542 \cdot 10^{-1}$	$1.306 \cdot 10^{-4}$	$8.751 \cdot 10^{-8}$	$3.676 \cdot 10^{-10}$
N ₂	$-2.38694295 \cdot 10^{-1}$	$1.426 \cdot 10^{-4}$	$-2.426 \cdot 10^{-7}$	$7.341 \cdot 10^{-10}$
HCN	$-2.14006155 \cdot 10^{-1}$	$9.534 \cdot 10^{-5}$	$-1.863 \cdot 10^{-8}$	$3.228 \cdot 10^{-10}$
mean		$8.326 \cdot 10^{-5}$	$-3.430 \cdot 10^{-8}$	$-1.324 \cdot 10^{-10}$
MAD		$8.326 \cdot 10^{-5}$	$1.625 \cdot 10^{-7}$	$1.661 \cdot 10^{-9}$
RMS		$9.271 \cdot 10^{-5}$	$2.095 \cdot 10^{-7}$	$2.207 \cdot 10^{-9}$

procedure. For the smallest ε value the deviations in the calculated energy are almost three orders of magnitude lower than the error in the ℓ^2 -norm of the corresponding tensors. The differences between the results obtained from decomposed/reduced quantities (c.f. Table 3.15, equivalent to MCR procedure) and the usage of tensors that are build from already approximated quantities (see Table 3.14) are in the order of one magnitude and the error for both methods is still lower than the given accuracy of the rank reduction procedure. So it can be concluded, that the reduction algorithm is very robust and the error in the ℓ^2 -norm is really an upper bound to the error in the correlation energy. Nevertheless, the error propagation during multiple approximations may be considered more carefully for an iterative procedure involving tensors in CP format and the ε -threshold of the approximation may have to be adjusted to smaller values.

As the usage of decomposed/reduced integral tensors in the MO basis is also a possible starting point for following Coupled-Cluster type calculations based on tensors in a low-rank CP representation the error in the MP2 correlation energy is taken as a measure for the quality of the error in the ℓ^2 -norm. Figure 3.21 shows the deviations in the MP2 energy for a series of alkyl chains calculated using the 6-31G basis set. As the quantities used here are obtained from a conventional calculation and only one rank reduction has to be applied to the amplitudes and integrals also larger values of ε are tested in the approximation. For values

of ε between 10^{-1} and 10^{-2} the error in the MP2 energy is in the range of 1 mHartree. If μ Hartree accuracy is required or multiple rank reductions are applied the value of ε should be decreased to 10^{-3} or 10^{-4} .

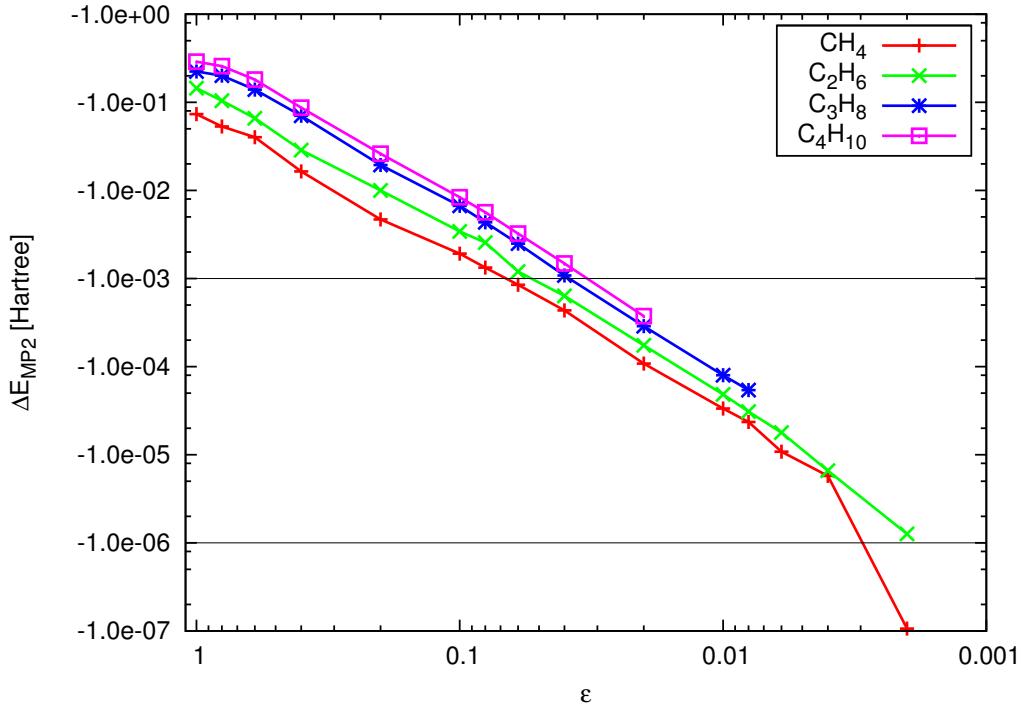


Figure 3.21: Error in MP2 energies for different values of ε in a series of alkyl chains using the 6-31G basis set. The amplitudes and MO integrals are obtained from a conventional calculation. The tensors are transformed into the CP format and reduced by a full rank-reduction.

3.7 LCCD Algorithm with Tensors in Low-Rank CP Representation

In the following sections an outline of a Coupled-Cluster algorithm using only tensors in a low-rank CP representation is given. The iterative procedure for solving the amplitude equations (set up residual tensor, rank reduction, do amplitude update, calculate energy, check convergence) is described and the different tensor contractions for the contributions to the residual tensor according to Eqn. 3.50 are explained in detail. First, all equations and the formal scaling are evaluated followed by a detailed example including the ranks of all involved tensors.

As a first attempt for using tensors in CP format in CC calculations the Linear Coupled-Cluster Doubles LCCD method is chosen. This method contains only double excitations (described by t_2 amplitudes) and only linear combinations of the parameters are allowed. General derivations of the CC amplitude and energy equations are given in Section 2.1.4 and the LCCD t_2 amplitude equation can be obtained from Eqn. 2.89 by skipping all higher order terms. In the index based format the amplitudes t_{ij}^{ab} are calculated according to

$$\begin{aligned} \epsilon_{ij}^{ab} t_{ij}^{ab} = & \langle ab\|ij\rangle + \frac{1}{2} \sum_{ef} t_{ij}^{ef} \langle ab\|ef\rangle + \frac{1}{2} \sum_{mn} t_{mn}^{ab} \langle mn\|ij\rangle + P(ij)P(ab) \sum_{me} t_{im}^{ae} \langle mb\|ej\rangle \\ & - P(ij) \sum_m f_j^m t_{im}^{ab} + P(ab) \sum_e f_e^b t_{ij}^{ae}, \end{aligned} \quad (3.50)$$

where $\epsilon_{ij}^{ab} = \epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b = (D_{ij}^{ab})^{-1}$ contains the orbital energies as obtained from Hartree-Fock, $P(pq)$ is a permutation operator and f_q^p denotes one electron contributions from the Fock matrix. This Equation is solved iteratively by setting up a residual value r_{ij}^{ab} as

$$\begin{aligned} r_{ij}^{ab} = & \underbrace{\langle ab\|ij\rangle}_{[r_1]_{ij}^{ab}} + \underbrace{\frac{1}{2} \sum_{ef} t_{ij}^{ef} \langle ab\|ef\rangle}_{[r_2]_{ij}^{ab}} + \underbrace{\frac{1}{2} \sum_{mn} t_{mn}^{ab} \langle mn\|ij\rangle}_{[r_3]_{ij}^{ab}} + \underbrace{P(ij)P(ab) \sum_{me} t_{im}^{ae} \langle mb\|ej\rangle}_{[r_4]_{ij}^{ab}} \\ & - P(ij) \underbrace{\sum_m f_j^m t_{im}^{ab}}_{[r_5]_{ij}^{ab}} + P(ab) \underbrace{\sum_e f_e^b t_{ij}^{ae}}_{[r_6]_{ij}^{ab}}, \end{aligned} \quad (3.51)$$

that contains incremental contributions $[r_n]_{ij}^{ab}$ for each tensor contraction. With the full residual tensor an amplitude update (c.f. Eqn. 3.38) is performed after each iteration

$$t_{ij}^{ab} = \frac{r_{ij}^{ab}}{\epsilon_i + \epsilon_j - \epsilon_a - \epsilon_b}. \quad (3.52)$$

In the first (or more convenient zeroth) iteration t_{ij}^{ab} is set to zero, so that only the first term

in Eqn. 3.51 contributes to the residuum so that

$$r_{ij}^{ab(0)} = \langle ab \| ij \rangle. \quad (3.53)$$

The amplitudes can now be calculated according to Eqn. 3.52 and it can be seen that this initial amplitude are the same as the already used MP2 amplitudes (see Eqn. 3.29). Now the first real iteration starts with the initial values of the amplitudes and a new residual tensor r_{ij}^{ab} can be calculated followed by the amplitude update.

In each iteration the correlation energy can be calculated in the same way as for the CCD method (c.f. Eqn. 2.88)

$$E_{\text{LCCD}} = \frac{1}{4} \sum_{efmn} \langle ef \| mn \rangle t_{mn}^{ef}, \quad (3.54)$$

and the iterative procedure is terminated if the change in the energy is below a given threshold. This threshold is typically in the order of 10^{-6} .

To use tensors in the CP format in the evaluation of the t_2 amplitudes it is necessary to provide all needed quantities like different types of MO integrals and the Fock matrix elements already in a low-rank CP format. The MO integrals can either be obtained by transformation of reduced AO integrals and a further reduction in the MO basis or by trivial decomposition of the transformed MO integrals. The Fock matrix elements can simply be cast into the CP format using the formalism of the trivial decomposition for this matrix or by a Singular Value Decomposition (where the eigenvalues are multiplied with one set of the eigenvectors). Both methods lead to a representation in the CP format with a maximal rank of N (number of basis functions)³, so that a reduction of the ranks is not necessary.

In order to keep the notation clear the following form of the integrals and Fock matrix elements with the corresponding ranks will be used for the derivation of the algorithm using tensors in CP format.

$$\begin{aligned} \mathfrak{v}_{ij}^{ab} &\approx \sum_{r_1=1}^{R_1} v_{r_1}^{(a)} \otimes v_{r_1}^{(b)} \otimes v_{r_1}^{(i)} \otimes v_{r_1}^{(j)} & \mathfrak{v}_{cd}^{ab} &\approx \sum_{r_2=1}^{R_2} v_{r_2}^{(a)} \otimes v_{r_2}^{(b)} \otimes v_{r_2}^{(c)} \otimes v_{r_2}^{(d)} \\ \mathfrak{v}_{kl}^{ij} &\approx \sum_{r_3=1}^{R_3} v_{r_3}^{(i)} \otimes v_{r_3}^{(j)} \otimes v_{r_3}^{(k)} \otimes v_{r_3}^{(l)} & \mathfrak{v}_{bj}^{ia} &\approx \sum_{r_4=1}^{R_4} v_{r_4}^{(i)} \otimes v_{r_4}^{(a)} \otimes v_{r_4}^{(b)} \otimes v_{r_4}^{(j)} \\ \mathfrak{f}_j^i &\approx \sum_{u_1=1}^{U_1} f_{u_1}^{(i)} \otimes f_{u_1}^{(j)} & \mathfrak{f}_b^a &\approx \sum_{u_2=1}^{U_2} f_{u_2}^{(a)} \otimes f_{u_2}^{(b)} \end{aligned}$$

It should again be noted, that the representing vectors $v_{r_1}^{(a)}$ in \mathfrak{v}_{ij}^{ab} and $v_{r_2}^{(a)}$ in \mathfrak{v}_{cd}^{ab} are different

³In principle the rank depends on the type of Fock matrix elements that are needed and is equal to the number of occupied orbitals for f_j^i or to the number of virtual orbitals for f_b^a , respectively.

from each other if the rank of the corresponding MO integral is reduced in the MO basis. For this case the ranks R_1 and R_2 are also different. Nevertheless, it is also possible to transform reduced AO integrals into the MO basis without further reduction. In that case the ranks would be equal and the two mentioned representing vectors are exactly the same, so that the additional subscripts of the expansion length can be skipped.

As there are six different contributions to the full residual tensor (c.f. Eqn. 3.51) that have to be expressed by tensors in the CP format one needs to distinguish the representing vectors arising from the different tensor contractions. Therefore, the expansion lengths are labelled with an additional index indicating the term from that it is constructed. So the residual contribution from the first term in Eqn. 3.51 is written as

$$[\mathfrak{r}_1]_{ij}^{ab} = \sum_{l_1=1}^{L_1} \tau_{l_1}^{(a)} \otimes \tau_{l_1}^{(b)} \otimes \tau_{l_1}^{(i)} \otimes \tau_{l_1}^{(j)}, \quad (3.55)$$

using τ instead of r as label for the representing vectors to avoid confusions with the summation index over the rank of MO integrals. The full residual tensor in the CP format will be indicated by the rank L and is the sum of six different contributions arising from the six tensor contractions in Eqn. 3.51

$$\mathfrak{r}_{ij}^{ab} = [\mathfrak{r}_1]_{ij}^{ab} + [\mathfrak{r}_2]_{ij}^{ab} + [\mathfrak{r}_3]_{ij}^{ab} + [\mathfrak{r}_4]_{ij}^{ab} - [\mathfrak{r}_5]_{ij}^{ab} + [\mathfrak{r}_6]_{ij}^{ab}. \quad (3.56)$$

The zeroth iteration, with no initial guess for the amplitude can be done in the same way as described for the MP2 amplitudes in Section 3.6. There only the first term in Eqn. 3.51 contributes to the residual tensor. Starting from a low-rank representation of the \mathfrak{v}_{ij}^{ab} integral

$$\mathfrak{v}_{ij}^{ab} \approx \sum_{r_1=1}^{R_1} v_{r_1}^{(a)} \otimes v_{r_1}^{(b)} \otimes v_{r_1}^{(i)} \otimes v_{r_1}^{(j)}, \quad (3.57)$$

one immediately obtains a low-rank approximation of the residual increment as a simple copy of the tensor

$$[\mathfrak{r}_1]_{ij}^{ab} = \mathfrak{v}_{ij}^{ab} \quad (3.58)$$

$$= \sum_{r_1=1}^{R_1} \underbrace{v_{r_1}^{(a)}}_{:=\tau_{l_1}^{(a)}} \otimes \underbrace{v_{r_1}^{(b)}}_{:=\tau_{l_1}^{(b)}} \otimes \underbrace{v_{r_1}^{(i)}}_{:=\tau_{l_1}^{(i)}} \otimes \underbrace{v_{r_1}^{(j)}}_{:=\tau_{l_1}^{(j)}} \quad (3.59)$$

$$= \sum_{l_1=1}^{L_1=R_1} \tau_{l_1}^{(a)} \otimes \tau_{l_1}^{(b)} \otimes \tau_{l_1}^{(i)} \otimes \tau_{l_1}^{(j)}. \quad (3.60)$$

As all other terms do not contribute to the full residual tensor it can be obtained as a copy

of the first increment

$$\tau_{ij}^{ab} = [\mathbf{t}_1]_{ij}^{ab} = \sum_{l_1=1}^{L_1} \underbrace{\tau_{l_1}^{(a)}}_{:=\tau_l^{(a)}} \otimes \underbrace{\tau_{l_1}^{(b)}}_{:=\tau_l^{(b)}} \otimes \underbrace{\tau_{l_1}^{(i)}}_{:=\tau_l^{(i)}} \otimes \underbrace{\tau_{l_1}^{(j)}}_{:=\tau_l^{(j)}} \quad (3.61)$$

$$= \sum_{l=1}^{L=L_1} \tau_l^{(a)} \otimes \tau_l^{(b)} \otimes \tau_l^{(i)} \otimes \tau_l^{(j)}. \quad (3.62)$$

This way the residual tensor is obtained also in the CP format.

3.7.1 Amplitude Update

In order to perform the amplitude update in a way that the resulting amplitudes are obtained in the CP format as well, the energy denominator needs to be approximated in the CP format. This is again (like for MP2) done using Laplace transformation and the numerical quadrature procedure

$$\frac{1}{-\epsilon_a - \epsilon_b + \epsilon_i + \epsilon_j} = D_{ij}^{ab} \rightarrow \mathfrak{D}_{ij}^{ab} \approx \sum_{s=1}^S \varepsilon_s^{(a)} \otimes \varepsilon_s^{(b)} \otimes \varepsilon_s^{(i)} \otimes \varepsilon_s^{(j)}. \quad (3.63)$$

Now that the residual tensor and the energy denominator are in the CP format the amplitude update can be performed as the tensor product of the two tensors:

$$\mathbf{t}_{ij}^{ab} \approx \mathbf{r}_{ij}^{ab} \cdot \mathfrak{D}_{ij}^{ab} \quad (3.64)$$

$$\approx \left(\sum_{l=1}^L \tau_l^{(a)} \otimes \tau_l^{(b)} \otimes \tau_l^{(i)} \otimes \tau_l^{(j)} \right) \cdot \left(\sum_{s=1}^S \varepsilon_s^{(a)} \otimes \varepsilon_s^{(b)} \otimes \varepsilon_s^{(i)} \otimes \varepsilon_s^{(j)} \right) \quad (3.65)$$

$$\approx \sum_{l=1}^L \sum_{s=1}^S \underbrace{\left(\tau_l^{(a)} \cdot \varepsilon_s^{(a)} \right)}_{:=t_q^{(a)}} \otimes \underbrace{\left(\tau_l^{(b)} \cdot \varepsilon_s^{(b)} \right)}_{:=t_q^{(b)}} \otimes \underbrace{\left(\tau_l^{(i)} \cdot \varepsilon_s^{(i)} \right)}_{:=t_q^{(i)}} \otimes \underbrace{\left(\tau_l^{(j)} \cdot \varepsilon_s^{(j)} \right)}_{:=t_q^{(j)}} \quad (3.66)$$

$$\approx \sum_{q=1}^{Q=L \cdot S} t_q^{(a)} \otimes t_q^{(b)} \otimes t_q^{(i)} \otimes t_q^{(j)}. \quad (3.67)$$

The two summations over l and s are merged together into one summation q in order to maintain a representation in the normal CP format with four representing vectors. The multiplications between vectors for the same dimension are done as Hadamard or entrywise products and the initial amplitudes are obtained in the CP format as well but the rank increases upon the formation of the tensor product. As already mentioned the rank of the approximate energy denominator is a constant and does not scale with the system size. The actual expansion length depends on the desired accuracy and currently is set to 42. In order to perform the tensor contractions in the following iterations the rank of the amplitudes therefore has to be reduced by application of the rank reduction algorithm. This algorithm is always mandatory if a contraction increases the rank of the resulting tensor.

With the initial amplitudes in the CP format all needed quantities for an iterative solution of Eqn. 3.51 are available in a low-rank CP representation and the tensor contractions can be done term by term in order to build up the residual tensor in CP format. The first term can again be seen as an initialization of the \mathbf{r}_1 tensor and is a simple copy operation of the corresponding integral.

3.7.2 Second and Third Term: $\sum_{ef} t_{ij}^{ef} \langle ab||ef \rangle$ and $\sum_{mn} t_{mn}^{ab} \langle mn||ij \rangle$

The second term is a contraction of the amplitudes with the four virtual MO integral

$$r_{ij}^{ab} \leftrightarrow \frac{1}{2} \sum_{ef} t_{ij}^{ef} \langle ab||ef \rangle, \quad (3.68)$$

where the arrow denotes that the term is only one contribution to the residual tensor and can be seen as an increment tensor. To calculate the increment tensor using the CP representation of the integrals and amplitudes the equation can be written as

$$[\mathbf{r}_2]_{ij}^{ab} = \frac{1}{2} \mathbf{t}_{ij}^{ef} \mathbf{v}_{ef}^{ab}. \quad (3.69)$$

It should be noted, that there is only one t_2 amplitude tensor and the indices are only used to distinguish external indices/dimensions from internal contraction indices/dimensions. The Kronecker product notation (for a derivation using matrix notation see appendix A.2.1 and A.2.2) for the two contracted tensors leads to

$$[\mathbf{r}_2]_{ij}^{ab} = \frac{1}{2} \mathbf{t}_{ij}^{ef} \mathbf{v}_{ef}^{ab} = \frac{1}{2} \left(\sum_{q=1}^Q \mathbf{t}_q^{(e)} \otimes \mathbf{t}_q^{(f)} \otimes t_q^{(i)} \otimes t_q^{(j)} \right) \cdot \left(\sum_{r_2=1}^R v_{r_2}^{(a)} \otimes v_{r_2}^{(b)} \otimes \mathbf{v}_{r_2}^{(e)} \otimes \mathbf{v}_{r_2}^{(f)} \right). \quad (3.70)$$

There, representing vectors for the same index/dimension can be multiplied together in the form of scalar products while the remaining representing vectors have to be written in the appropriate order to match the dimensions of the residual tensor

$$[\mathbf{r}_2]_{ij}^{ab} = \frac{1}{2} \sum_{q=1}^Q \sum_{r_2=1}^{R_2} \underbrace{\left\langle \mathbf{t}_q^{(e)}, \mathbf{v}_{r_2}^{(e)} \right\rangle \left\langle \mathbf{t}_q^{(f)}, \mathbf{v}_{r_2}^{(f)} \right\rangle}_{:=\tau_{l_2}^{(a)}} v_{r_2}^{(a)} \otimes \underbrace{\mathbf{v}_{r_2}^{(b)}}_{:=\tau_{l_2}^{(b)}} \otimes \underbrace{t_q^{(i)}}_{:=\tau_{l_2}^{(i)}} \otimes \underbrace{t_q^{(j)}}_{:=\tau_{l_2}^{(j)}}. \quad (3.71)$$

This way one ends up with a representation of the residual increment in the THC format (the two scalar products can be seen as a coefficient matrix). [105] In order to add this THC tensor to the existing residual tensor in CP representation the format has to be converted. Therefore, the two summations are consolidated into one summation over l_2 that is obtained as a product of q and r_2 . The corresponding matrix elements of the coefficient matrix can be multiplied into the representation system for the first dimension and the remaining vectors are simply copied. The prefactor of $\frac{1}{2}$ can also be multiplied into the representing system so

that finally one ends up with a representation in the CP format but with an increased rank of $L_2 = Q \cdot R_2$

$$[\mathfrak{r}_2]_{ij}^{ab} = \sum_{l_2=1}^{L_2=Q \cdot R_2} \tau_{l_2}^{(a)} \otimes \tau_{l_2}^{(b)} \otimes \tau_{l_2}^{(i)} \otimes \tau_{l_2}^{(j)}. \quad (3.72)$$

This rank can be reduced with the reduction algorithm and a rank similar to that of the t_2 amplitudes can be expected for the low-rank approximation of the increment to the residual. The increment tensor can then be added to the residual tensor by simple addition of the representation vectors of the increment to the set of representation vectors in the residual tensor. This way the rank increases and one can reduce the rank again using the reduction algorithm.

The third term can be done in analogy to the second term using a different type of MO integral but also in a low-rank CP approximation. Note that the rank of this type of integral does not necessarily have to be the same as the rank of the previous used four virtual index integral. As already pointed out before, these ranks are only the same if the rank in the MO basis is not further compressed after transformation of the AO integrals. However, it is possible to have a much lower rank for the four occupied MO integral, due to the fact that the occupied space is usually much smaller than the virtual space, especially if larger basis sets are used. Using the notation with Kronecker products (implicit summation over internal dimensions is assumed) the contraction yields

$$[\mathfrak{r}_3]_{ij}^{ab} = \frac{1}{2} \mathfrak{t}_{mn}^{ab} \mathfrak{v}_{ij}^{mn} \quad (3.73)$$

$$= \frac{1}{2} \left(\sum_{q=1}^Q t_q^{(a)} \otimes t_q^{(b)} \otimes \textcolor{blue}{t}_q^{(m)} \otimes \textcolor{red}{t}_q^{(n)} \right) \left(\sum_{r_3=1}^{R_3} \textcolor{blue}{v}_{r_3}^{(m)} \otimes \textcolor{red}{v}_{r_3}^{(n)} \otimes v_{r_3}^{(i)} \otimes v_{r_3}^{(j)} \right). \quad (3.74)$$

As all dimensions of the tensors are factorized the internal contraction of common indices/dimensions can be done as scalar products of the corresponding representing vectors

$$[\mathfrak{r}_3]_{ij}^{ab} = \frac{1}{2} \sum_{q=1}^Q \sum_{r_3=1}^{R_3} \langle t_q^{(m)}, v_{r_3}^{(m)} \rangle \langle \textcolor{blue}{t}_q^{(n)}, \textcolor{red}{v}_{r_3}^{(n)} \rangle t_q^{(a)} \otimes t_q^{(b)} \otimes v_{r_3}^{(i)} \otimes v_{r_3}^{(j)}. \quad (3.75)$$

The two summations over the two ranks can be merged together into one rank $l_3 = q \cdot r_3$ and the scalar products as well as the prefactor are multiplied into one of the remaining (simply copied) representing vectors. This way the residual increment is again obtained in the CP format but with an increased rank $L_3 = Q \cdot R_3$ that can be reduced through application of the rank reduction procedure

$$[\mathfrak{r}_3]_{ij}^{ab} = \sum_{l_3=1}^{L_3=Q \cdot R_3} \tau_{l_3}^{(a)} \otimes \tau_{l_3}^{(b)} \otimes \tau_{l_3}^{(i)} \otimes \tau_{l_3}^{(j)}. \quad (3.76)$$

In general the computational cost for evaluating a tensor contraction in the CP format scales as $\mathcal{O}(N \cdot Q \cdot R)$, where N denotes the number of basis functions, Q stands for the

rank of the amplitudes and R for the rank of the corresponding integral tensor. As already pointed out before, the rank for the involved four virtual MO integral scales approximately with $\mathcal{O}(N^{2.6})$ and the corresponding t_2 amplitude tensor scales as $\mathcal{O}(N^{1.4})$ for an accuracy parameter of 10^{-4} . Therefore, the overall tensor contraction for the second term exhibits an scaling of $\mathcal{O}(N^5)$ which is approximately 1 order of magnitude lower than the scaling of the conventional tensor contraction, using tensors in index based representation ($\mathcal{O}(\text{virt}^4 \cdot \text{occ}^2) \approx \mathcal{O}(N^6)$). For larger values of ε the scaling can even be decreased further to approximately $\mathcal{O}(N^4)$. The scaling of the computational cost for the third term is approximately $\mathcal{O}(N \cdot Q \cdot R_3)$ compared to the conventional scaling of $\mathcal{O}(\text{occ}^4 \cdot \text{virt}^2)$. As presented in the previous sections the four occupied integral scales approximately with $\mathcal{O}(N^2)$ while the amplitudes exhibit a typical scaling of $\mathcal{O}(N^{1.4})$. So the overall contraction shows a scaling of $\mathcal{O}(N^{4.4})$.

Nevertheless, one should keep in mind that the rank of the increments to the residual tensor are increased and have to be reduced in order to keep a low-rank structure that allows for the efficient calculation of amplitudes in a low-rank tensor format. Otherwise the ranks would increase rapidly during the iterative procedure so that the scaling becomes highly unfavourable compared to the scaling of the conventional algorithm. Therefore, the scaling of the rank reduction procedure itself has to be added to the pure contraction costs.

As the rank reduction procedure itself scales with the initial rank (product of the ranks of the amplitudes and the integrals) times the final rank, that can be expected to be reduced to a rank that is similar to that of the t_2 amplitudes (see Eqn. 3.19), the reduction procedure for the second term would scale approximately with

$$\mathcal{O}(N \cdot R_{\text{init}} \cdot R_{\text{final}}) \quad (3.77)$$

$$\approx \mathcal{O}(N \cdot (Q \cdot R_2) \cdot Q) \quad (3.78)$$

$$\approx \mathcal{O}(N \cdot (N^{1.4} \cdot N^{2.6}) \cdot N^{1.4}) \approx \mathcal{O}(N^{7.4}). \quad (3.79)$$

This is even higher than the contraction cost in the index based tensor format. Therefore, the rank reduction procedure has to be done in slices in order to keep the initial rank fix and allow for an efficient rank reduction of small increments of the residual tensor. This procedure lends itself to a parallel implementation and the overall reduction procedure scales as

$$\# \text{ of reductions} \cdot \mathcal{O}(N \cdot R_{\text{init}}(\text{fix}) \cdot R_{\text{final}}) \approx \# \text{ of reductions} \cdot \mathcal{O}(N \cdot \text{const.} \cdot N^{1.4}). \quad (3.80)$$

The slice sizes can be chosen similar to the values obtained from the sliced reduction of AO integrals and this way, each rank reduction scales only approximately with $\mathcal{O}(N^{2.4})$. Nevertheless, one has to keep in mind that the number of slices still is proportional to the initial rank so that for larger examples more and more reductions with a fixed initial rank occur. It should be mentioned that this work does not primarily aim at improving the rank

reduction itself, but rather at the applicability of the low-rank CP format for the iterative solution of CC amplitude equations based on low-rank representation of the MO integrals. Therefore, in the following only the contraction costs are evaluated and the rank reduction is only mentioned for quantities with increased ranks. Further improvements of the rank reduction procedure specially for quantities occurring in electronic structure methods are given in Section 3.9.

3.7.3 Fourth Term: $P(ij)P(ab) \sum_{me} t_{im}^{ae} \langle mb||ej \rangle$

The contraction for the fourth term is done in the same way as presented for term two and three. For simplicity the permutation operators are skipped in the derivation of the contraction and introduced later in the final definition of the residual increment

$$[\mathfrak{r}_4]_{ij}^{ab} = P(ij)P(ab) [\mathfrak{r}_4]_{ij}^{ab} \quad \text{with} \quad [\mathfrak{r}_4]_{ij}^{ab} = \mathfrak{t}_{im}^{ae} \mathfrak{v}_{ej}^{mb}. \quad (3.81)$$

The two vectors in low-rank CP representation allow for the contraction of internal indices in form of scalar products. In order to maintain a representation in the CP format the two ranks in Eqn. 3.83 have to be combined into one rank leading to an increased rank of the intermediate tensor $[\mathfrak{r}_4]_{ij}^{ab}$ in CP representation.

$$[\mathfrak{r}_4]_{ij}^{ab} = \mathfrak{t}_{im}^{ae} \mathfrak{v}_{ej}^{mb} \quad (3.82)$$

$$= \left(\sum_{q=1}^Q t_q^{(a)} \otimes \mathfrak{t}_q^{(e)} \otimes t_q^{(i)} \otimes \mathfrak{t}_q^{(m)} \right) \left(\sum_{r_4=1}^{R_4} \mathfrak{v}_{r_4}^{(m)} \otimes v_{r_4}^{(b)} \otimes \mathfrak{v}_{r_4}^{(e)} \otimes v_{r_4}^{(j)} \right) \quad (3.83)$$

$$= \sum_{q=1}^Q \sum_{r_4=1}^{R_4} \langle \mathfrak{t}_q^{(e)}, \mathfrak{v}_{r_4}^{(e)} \rangle \langle \mathfrak{t}_q^{(m)}, \mathfrak{v}_{r_4}^{(m)} \rangle t_q^{(a)} \otimes v_{r_4}^{(b)} \otimes t_q^{(i)} \otimes v_{r_4}^{(j)} \quad (3.84)$$

$$= \sum_{w_4=1}^{W_4=Q \cdot R_4} x_{w_4}^{(a)} \otimes x_{w_4}^{(b)} \otimes x_{w_4}^{(i)} \otimes x_{w_4}^{(j)} \quad (3.85)$$

It should be noted, that the order of the representation vectors in Eqn. 3.83 is important and has to match the order of the dimensions in the target tensor. As the rank of this intermediate tensor is increased it has to be reduced in order to maintain a low-rank representation in the CP format.

The final permutation of indices to build the residual increment can be done by simply changing the order of the representing vectors in the intermediate tensor

$$P(ij)P(ab) [\mathfrak{r}_4]_{ij}^{ab} = [\mathfrak{r}_4]_{ij}^{ab} - [\mathfrak{r}_4]_{ij}^{ba} - [\mathfrak{r}_4]_{ji}^{ab} + [\mathfrak{r}_4]_{ji}^{ba}. \quad (3.86)$$

This way all possible combinations are accessible by permutations of the representing vectors. The summations or subtractions can be done in the CP format as well and lead to an addition of the corresponding ranks. For subtractions the tensor is multiplied with -1 and can be

treated like a normal addition. So the final increment to the residual tensor is obtained as

$$[\mathbf{r}_4]_{ij}^{ab} = P(ij)P(ab) [\mathbf{x}_4]_{ij}^{ab} \quad (3.87)$$

$$= \sum_{w_4=1}^{W_4} x_{w_4}^{(a)} \otimes x_{w_4}^{(b)} \otimes x_{w_4}^{(i)} \otimes x_{w_4}^{(j)} - \sum_{w_4=1}^{W_4} x_{w_4}^{(b)} \otimes x_{w_4}^{(a)} \otimes x_{w_4}^{(i)} \otimes x_{w_4}^{(j)} \quad (3.88)$$

$$= \sum_{w_4=1}^{W_4} x_{w_4}^{(a)} \otimes x_{w_4}^{(b)} \otimes x_{w_4}^{(j)} \otimes x_{w_4}^{(i)} + \sum_{w_4=1}^{W_4} x_{w_4}^{(b)} \otimes x_{w_4}^{(a)} \otimes x_{w_4}^{(j)} \otimes x_{w_4}^{(i)} \quad (3.89)$$

Here the rank is again increased and has to be reduced using the rank reduction algorithm to obtain a low-rank approximation of the residual in the CP format.

3.7.4 Fifth and Sixth Term:

$P(ij) \sum_m f_j^m t_{im}^{ab}$ and $P(ab) \sum_e f_e^b t_{ij}^{ae}$

These terms contract different Fock matrix elements with the t_2 amplitudes and do not involve two-electron integrals. Therefore, also a representation of the Fock matrix elements in the CP format is needed. As the Fock matrix is only a two dimensional object the casting can be done either by trivial decomposition (c.f. Section 3.1.1) or by a Singular Value Decomposition (see Section 3.1.2). In both cases the initial rank U is at most N (number of basis functions) or even lower, if the SVD based casting is used and the Fock matrix has eigenvalues that are below the given threshold

$$f_j^i = \sum_{u_1=1}^{U_1} f_{u_1}^i f_{u_1}^j \quad \text{or} \quad \mathbf{f}_j^i = \sum_{u_1=1}^{U_1} f_{u_1}^{(i)} \otimes f_{u_1}^{(j)} \quad (3.90)$$

$$f_b^a = \sum_{u_2=1}^{U_2} f_{u_2}^a f_{u_2}^b \quad \text{or} \quad \mathbf{f}_b^a = \sum_{u_2=1}^{U_2} f_{u_2}^{(a)} \otimes f_{u_2}^{(b)}. \quad (3.91)$$

In the following only the fifth term is explained explicitly as the sixth term can be treated in the same way (a detailed derivation is given in the appendix A.2.3). For convenience the permutation operator is again skipped in the derivation of the contraction but has to be applied for constructing the final residual increment in CP format before adding the increment to the set of already obtained representing vectors

$$[\mathbf{r}_5]_{ij}^{ab} = P(ij)[\mathbf{x}_5]_{ij}^{ab} \quad \text{with} \quad [\mathbf{x}_5]_{ij}^{ab} = \mathbf{f}_j^m \mathbf{t}_{im}^{ab}. \quad (3.92)$$

Now the contraction can be done in the same way as for the previous terms by performing internal contractions as scalar products of the corresponding representing vectors

$$[\mathfrak{x}_5]_{ij}^{ab} = \mathfrak{f}_j^m \mathfrak{t}_{im}^{ab} \quad (3.93)$$

$$= \left(\sum_{u_1=1}^{U_1} f_{u_1}^{(m)} \otimes f_{u_1}^{(j)} \right) \left(\sum_{q=1}^Q t_q^{(a)} \otimes t_q^{(b)} \otimes t_q^{(i)} \otimes t_q^{(m)} \right) \quad (3.94)$$

$$= \sum_{u_1=1}^{U_1} \sum_{q=1}^Q \underbrace{\langle f_{u_1}^{(m)}, t_q^{(m)} \rangle}_{\mathbf{M}_{u_1 q}} t_q^{(a)} \otimes t_q^{(b)} \otimes t_q^{(i)} \otimes f_{u_1}^{(j)}. \quad (3.95)$$

As only the last representing vector and all scalar products depend on the first rank, this summation can be seen as a kind of transformation from the expansion length u_1 to q with the help of all scalar products. This becomes even more clear, if all scalar products in Eqn. 3.95 are written as a matrix $\mathbf{M}_{u_1 q}$ and the summation over u is put in front of the last representing vector

$$[\mathfrak{x}_5]_{ij}^{ab} = \sum_{q=1}^Q \underbrace{t_q^{(a)} \otimes t_q^{(b)} \otimes t_q^{(i)}}_{:=x_{w_5}^{(a)} \otimes x_{w_5}^{(b)} \otimes x_{w_5}^{(i)}} \otimes \underbrace{\left(\sum_{u_1=1}^{U_1} \mathbf{M}_{u_1 q} f_{u_1}^{(j)} \right)}_{:=x_{w_5}^{(j)}} \quad (3.96)$$

$$= \sum_{w_5=1}^{W_5=Q} x_{w_5}^{(a)} \otimes x_{w_5}^{(b)} \otimes x_{w_5}^{(i)} \otimes x_{w_5}^{(j)}. \quad (3.97)$$

This way the last representing vector can be calculated through a matrix-matrix multiplication of the scalar product matrix \mathbf{M} and all representing vectors written as a matrix. It should be noted, that the rank for this contraction does not change and the residual increment is obtained in the CP format with the same rank as the corresponding amplitude tensor.

The residual increment is obtained by permuting the third and fourth dimension in the incremental tensor

$$[\mathfrak{r}_5]_{ij}^{ab} = P(ij)[\mathfrak{x}_5]_{ij}^{ab} = [\mathfrak{x}_5]_{ij}^{ab} - [\mathfrak{x}_5]_{ji}^{ab} \quad (3.98)$$

$$= \sum_{w_5=1}^{W_5=Q} x_{w_5}^{(a)} \otimes x_{w_5}^{(b)} \otimes x_{w_5}^{(i)} \otimes x_{w_5}^{(j)} - \sum_{w_5=1}^{W_5=Q} x_{w_5}^{(a)} \otimes x_{w_5}^{(b)} \otimes x_{w_5}^{(j)} \otimes x_{w_5}^{(i)} \quad (3.99)$$

$$= \sum_{l_5=1}^{L_5=2 \cdot W_5=2 \cdot Q} \tau_{l_5}^{(a)} \otimes \tau_{l_5}^{(b)} \otimes \tau_{l_5}^{(i)} \otimes \tau_{l_5}^{(j)}. \quad (3.100)$$

This way the rank of the residual increment is again increased and can be reduced by the application of the reduction algorithm. It should be noted, that the ranks only increase by a constant factor compared to the increased ranks for the previous terms, where the new rank is the product of the ranks of the two contracted tensors.

In summary this contraction can be done as a matrix-matrix multiplication of the two representing vectors that have to be contracted, followed by a matrix-matrix multiplication of the resulting coefficient matrix with the remaining representing vectors of the Fock matrix. So the rank of the intermediate tensor does not change and the first three representing vectors are just copied while the last vector is obtained from the described transformation. Only the final permutation of dimensions increases the rank of the residual increment by a factor of two. However, one should be able to reduce the increased rank to approximately the rank of the corresponding t_2 amplitudes by application of the rank reduction procedure.

3.7.5 Final Residual Tensor

The final residual tensor can be set up by adding all representing vectors of the incremental tensors together so that the rank of the residual tensors is the sum of the ranks of all increments.

$$\tau_{ij}^{ab} = [\tau_1]_{ij}^{ab} + [\tau_2]_{ij}^{ab} + [\tau_3]_{ij}^{ab} + [\tau_4]_{ij}^{ab} - [\tau_5]_{ij}^{ab} + [\tau_6]_{ij}^{ab} \quad (3.101)$$

$$\begin{aligned} &= \sum_{l_1=1}^{L_1} \tau_{l_1}^{(a)} \otimes \tau_{l_1}^{(b)} \otimes \tau_{l_1}^{(i)} \otimes \tau_{l_1}^{(j)} + \sum_{l_2=1}^{L_2} \tau_{l_2}^{(a)} \otimes \tau_{l_2}^{(b)} \otimes \tau_{l_2}^{(i)} \otimes \tau_{l_2}^{(j)} \\ &\quad + \sum_{l_3=1}^{L_3} \tau_{l_3}^{(a)} \otimes \tau_{l_3}^{(b)} \otimes \tau_{l_3}^{(i)} \otimes \tau_{l_3}^{(j)} + \sum_{l_4=1}^{L_4} \tau_{l_4}^{(a)} \otimes \tau_{l_4}^{(b)} \otimes \tau_{l_4}^{(i)} \otimes \tau_{l_4}^{(j)} \end{aligned} \quad (3.102)$$

$$\begin{aligned} &\quad - \sum_{l_5=1}^{L_5} \tau_{l_5}^{(a)} \otimes \tau_{l_5}^{(b)} \otimes \tau_{l_5}^{(i)} \otimes \tau_{l_5}^{(j)} + \sum_{l_6=1}^{L_6} \tau_{l_6}^{(a)} \otimes \tau_{l_6}^{(b)} \otimes \tau_{l_6}^{(i)} \otimes \tau_{l_6}^{(j)} \\ &= \sum_{l=1}^{L=\sum_{\mu=1}^6 L_\mu} \tau_l^{(a)} \otimes \tau_l^{(b)} \otimes \tau_l^{(i)} \otimes \tau_l^{(j)} \end{aligned} \quad (3.103)$$

This enlarged rank can be reduced using the rank reduction procedure as described in Section 3.2 and a low-rank approximation is obtained. Now the amplitude update as described in Section 3.7.1 can be performed leading to updated amplitudes, again with increased rank that can be reduced to values similar to the rank of the previous iteration. The next iteration can be done using the updated amplitudes and so on and so forth.

The evaluation of the correlation energy can be done according to Eqn. 3.54 using tensors in CP format as well. This is the same as the evaluation of the MP2 correlation as presented in Section 3.6

$$E_{\text{LCCD}} = \frac{1}{4} \sum_{efmn} \langle ef \| mn \rangle t_{mn}^{ef} = \frac{1}{4} \mathfrak{v}_{mn}^{ef} \mathfrak{t}_{mn}^{ef}. \quad (3.104)$$

The energy can be calculated as a sum of all inner products of the amplitudes and the MO

integrals in CP format

$$E_{\text{LCCD}} = \frac{1}{4} \left(\sum_{q=1}^Q t_q^{(e)} \otimes t_q^{(f)} \otimes t_q^{(m)} \otimes t_q^{(n)} \right) \cdot \left(\sum_{r_1=1}^{R_1} v_{r_1}^{(e)} \otimes v_{r_1}^{(f)} \otimes v_{r_1}^{(m)} \otimes v_{r_1}^{(n)} \right) \quad (3.105)$$

$$= \frac{1}{4} \sum_{q=1}^Q \sum_{r_1=1}^{R_1} \left(t_q^{(e)} \cdot v_{r_1}^{(e)} \right) \otimes \left(t_q^{(f)} \cdot v_{r_1}^{(f)} \right) \otimes \left(t_q^{(m)} \cdot v_{r_1}^{(m)} \right) \otimes \left(t_q^{(n)} \cdot v_{r_1}^{(n)} \right) \quad (3.106)$$

$$= \frac{1}{4} \sum_{q=1}^Q \sum_{r_1=1}^{R_1} \langle t_q^{(e)}, v_{r_1}^{(e)} \rangle \otimes \langle t_q^{(f)}, v_{r_1}^{(f)} \rangle \otimes \langle t_q^{(m)}, v_{r_1}^{(m)} \rangle \otimes \langle t_q^{(n)}, v_{r_1}^{(n)} \rangle. \quad (3.107)$$

Now the sum over the two expansion lengths can be carried out as the Kronecker products can be substituted by normal products between the different scalars. Thus, only scalar products between the different representing vectors of t_{ij}^{ab} and v_{ij}^{ab} have to be evaluated and summed up.

3.7.6 Example: H₂O using 6-31G Basis Set

As an example of how the ranks change during the contractions and the subsequent rank reductions in the first iterations of the CCD amplitude equations, water with a 6-31G basis set is chosen. The two-electron integrals in the MO basis are taken from a standard calculation and cast into the CP format by trivial decomposition followed by rank reduction to an accuracy of $\varepsilon = 10^{-4}$. The resulting ranks are given in Table 3.16. It should be noted that

Table 3.16: Rank of MO integrals for H₂O using the 6-31G basis set. The tensors are obtained from standard calculations and cast into the CP format by trivial decomposition. The rank reduction is done with a threshold of 10^{-4} .

tensor	CP representation	reduced rank (R_μ)
v_{ij}^{ab}	$\sum_{r_1}^{R_1} v_{r_1}^{(a)} \otimes v_{r_1}^{(b)} \otimes v_{r_1}^{(i)} \otimes v_{r_1}^{(j)}$	85
v_{cd}^{ab}	$\sum_{r_2}^{R_2} v_{r_2}^{(a)} \otimes v_{r_2}^{(b)} \otimes v_{r_2}^{(c)} \otimes v_{r_2}^{(d)}$	108
v_{kl}^{ij}	$\sum_{r_3}^{R_3} v_{r_3}^{(i)} \otimes v_{r_3}^{(j)} \otimes v_{r_3}^{(k)} \otimes v_{r_3}^{(l)}$	36
v_{bj}^{ia}	$\sum_{r_4}^{R_4} v_{r_4}^{(i)} \otimes v_{r_4}^{(a)} \otimes v_{r_4}^{(b)} \otimes v_{r_4}^{(j)}$	61

one can also use the reduced AO integrals and perform the AO-MO transformation in the CP format as described in Section 3.5. This would lead to the same rank for all MO integrals of $R = 343$ and in principle only the v_{cd}^{ab} and v_{kl}^{ij} are needed. The two other tensors can then be constructed from these representations by simply copying the needed representation vectors

in the right order. Nevertheless, it is also possible to construct all four types of integrals by transformation and reduce the rank in the MO basis further, which leads to almost the same ranks (c.f. Table 3.11).

Starting with the zeroth iteration, for which the amplitudes are zero, the residual is just a copy of the \mathbf{v}_{ij}^{ab} integral. So the rank of the first residual tensor is equal to that of the amplitude $L = R_1 = 85$. Performing the amplitude update is done with a fairly large expansion of the denominator (rank $S = 42$) in order to achieve a very high accuracy. Therefore, the rank of the amplitudes grows to $Q = 3570$ and can be reduced by a sliced reduction (slice size is chosen as 200 due to the size of Q) followed by a full reduction with the same accuracy parameter ($\varepsilon = 10^{-4}$) to a rank of $Q = 84$. This is similar to the rank of the corresponding residual tensor or the \mathbf{v}_{ij}^{ab} integral, respectively. For practical reasons the rank of the denominator can be decreased to values around 15 still maintaining an accuracy of the numerical integration of the Laplace transformed denominator of $6.311 \cdot 10^{-7}$ in an interval from zero to $2 \cdot 10^6$ [231].

The first iteration starts with a rank of the t_2 amplitudes of $Q = 84$ and all needed contractions are done as described in the previous Sections. A summary of the resulting ranks upon tensor contraction and possible further rank reduction is given in Table 3.17. Here it should be noted, that the increased ranks are reduced in a sliced reduction “on the fly”. That means, that only a few of the scalar products are evaluated and multiplied with one representing vector while the remaining representing vectors are just copied until the new rank reaches a certain size. The rank reduction is then applied only to this small slice of the full tensor and only the reduced vectors are saved for a final full reduction. This way the initial ranks for the rank reduction procedure are kept small and the reduction is much faster than the full reduction of the full residual increment.

From Table 3.17 it can be seen, that the reduced ranks of the residual increments (including all necessary permutations) are almost the same for all given terms and in the same order

Table 3.17: Evolution of ranks during LCCD amplitude equation for H₂O using the 6-31G basis set. All increased residual ranks are reduced using the *sfr* procedure with a slice size of 200. The permutations are done with the already reduced residual tensors followed by another reduction leading to the same rank as before the permutation.

term	\mathbf{v}_{ij}^{ab}	$\frac{1}{2}\mathbf{t}_{ij}^{ef}\mathbf{v}_{ef}^{ab}$	$\frac{1}{2}\mathbf{t}_{mn}^{ab}\mathbf{v}_{ij}^{mn}$	$P(ij)P(ab)\mathbf{t}_{im}^{ae}\mathbf{v}_{ej}^{mb}$	$P(ij)\mathbf{f}_j^m\mathbf{t}_{im}^{ab}$	$P(ab)\mathbf{f}_e^b\mathbf{t}_{ij}^{ae}$
rank t_2	—	84	84	84	84	84
rank 2. tensor	85	108	36	61	5	8
rank residual	85	9072	3024	5124	84	84
reduced residual rank	85	84	86	84	84	84

as for the zeroth iteration. To get the full residual tensor in CP format all increments have to be summed up. This is done by simply adding all representing vectors for one dimension together and thus summing up the final rank. The so increased rank of $L = 507$ can again be reduced using the reduction procedure and a final low-rank representation of the residual tensor with a rank of $L = 90$ can be achieved. This rank is a little bit larger than the rank in the zeroth iteration but still quite similar.

The amplitude update is done with the new residual tensor leading to an increased rank of the amplitudes of $Q = 3780$, that can again be reduced to $Q = 87$. So the next iteration can start with almost the same expansion length for the t_2 amplitudes and yield similar ranks for the residual increments (after reduction). For one iteration the total procedure consists of 5 tensor contractions, 8 rank reductions, 3 permutations, 1 summation of increments, 1 Hadamard product for the amplitude update and 1 evaluation of all inner products for the energy, so that the rank reduction becomes the most important and time determining part of the algorithm.

In order to estimate the resulting error in the LCCD correlation energy the difference between the energy calculated with CP tensors and the result from a calculation using index based tensors after the first iteration can be taken as an example. The error in the correlation energy is $-3.92 \cdot 10^{-7}$ and so almost 3 orders of magnitude lower than the given threshold of 10^{-4} . So the error in the ℓ^2 -norm can be considered as a very robust fitting criterion. Furthermore, multiple application of the rank reduction for different quantities does not introduce any artefacts (error propagation) but seems to be a rather robust procedure.

Table 3.18 lists the LCCD correlation energy calculated with NWchem (no convergence acceleration like DIIS used; modified `tce` module) [234–236], from tensors in CP format using a very large rank of the energy denominator ($S = 42$) and E_{LCCD} calculated from CP tensors with a lower accuracy for the energy denominator ($S = 15$). The two CP based methods use an accuracy of 10^{-4} for the rank reduction and MO integrals obtained from a MCR type procedure using the same threshold. There it can be seen, that error in the LCCD correlation energy does not change much during the iterations for both values of S . The averaged error over all iterations for $S = 42$ is with $-5.16 \cdot 10^{-7}$ still almost 3 orders of magnitude smaller than the threshold for the rank reduction of 10^{-4} . Also changing the rank of the energy denominator from 42 to 15 does not change the error significantly. The averaged error compared to the standard calculation is then in the order of $-6.54 \cdot 10^{-7}$. Taking the convergence of the calculated energy into account (assuming a convergence of 10^{-6}) the difference between the calculated energies with the two values of S is in the order of $8 \cdot 10^{-7}$. Therefore, it can be concluded that the expansion of the energy denominator can be calculated with $S = 15$ still maintaining the desired accuracies that are applied for the rank reduction procedure (currently up to 10^{-6}). All following calculations are done with the smaller rank of the energy denominator.

Table 3.18: Error in LCCD correlation energy using tensors in CP format for H₂O using 6-31G basis set. For comparison two different accuracies for the energy denominator are listed: $S = 42$ and $S = 15$. In the CP format an accuracy for all reductions of $\varepsilon = 10^{-4}$ is used.

iter.	NWchem E_{LCCD}	CP $S = 42$		CP $S = 15$	
		E_{LCCD}	ΔE_{LCCD}	E_{LCCD}	ΔE_{LCCD}
1	-0.1303767113	-0.1303771036	$-3.923 \cdot 10^{-7}$	-0.1303771884	$-4.771 \cdot 10^{-7}$
2	-0.1332701143	-0.1332699013	$2.131 \cdot 10^{-7}$	-0.1332705322	$-4.179 \cdot 10^{-7}$
3	-0.1341805880	-0.1341807520	$-1.640 \cdot 10^{-7}$	-0.1341813991	$-8.111 \cdot 10^{-7}$
4	-0.1345495975	-0.1345500406	$-4.431 \cdot 10^{-7}$	-0.1345503466	$-7.491 \cdot 10^{-7}$
5	-0.1347009820	-0.1347021191	$-1.137 \cdot 10^{-6}$	-0.1347014769	$-4.949 \cdot 10^{-7}$
6	-0.1347655018	-0.1347660783	$-5.765 \cdot 10^{-7}$	-0.1347664370	$-9.352 \cdot 10^{-7}$
7	-0.1347936526	-0.1347938040	$-1.514 \cdot 10^{-7}$	-0.1347935426	$1.100 \cdot 10^{-7}$
8	-0.1348061702	-0.1348070214	$-8.512 \cdot 10^{-7}$	-0.1348073171	$-1.147 \cdot 10^{-6}$
9	-0.1348118196	-0.1348125987	$-7.791 \cdot 10^{-7}$	-0.1348128735	$-1.054 \cdot 10^{-6}$
10	-0.1348144004	-0.1348150179	$-6.174 \cdot 10^{-7}$	-0.1348145016	$-1.011 \cdot 10^{-7}$
11	-0.1348155915	-0.1348159227	$-3.312 \cdot 10^{-7}$	-0.1348160267	$-4.352 \cdot 10^{-7}$
12	-0.1348161460	-0.1348166906	$-5.447 \cdot 10^{-7}$	-0.1348167466	$-6.006 \cdot 10^{-7}$
13	-0.1348164061	-0.1348172655	$-8.594 \cdot 10^{-7}$	-0.1348168535	$-4.474 \cdot 10^{-7}$
14	-0.1348165289	-0.1348163776	$1.513 \cdot 10^{-7}$	-0.1348171679	$-6.390 \cdot 10^{-7}$
15	-0.1348165873	-0.1348170911	$-5.039 \cdot 10^{-7}$	-0.1348173176	$-7.303 \cdot 10^{-7}$
16	-0.1348166152	-0.1348173004	$-6.852 \cdot 10^{-7}$	-0.1348175825	$-9.673 \cdot 10^{-7}$
17	-0.1348166286	-0.1348174101	$-7.816 \cdot 10^{-7}$	-0.1348171873	$-5.588 \cdot 10^{-7}$
18	-0.1348166350	-0.1348168948	$-2.597 \cdot 10^{-7}$	-0.1348176410	$-1.006 \cdot 10^{-6}$
19	-0.1348166381	-0.1348175497	$-9.115 \cdot 10^{-7}$	-0.1348176958	$-1.058 \cdot 10^{-6}$
20	-0.1348166397	-0.1348173356	$-6.959 \cdot 10^{-7}$	-0.1348172084	$-5.687 \cdot 10^{-7}$

More interesting is the convergence of the LCCD algorithm using only tensors in CP format and multiple rank reduction steps, which is given in Figure 3.22 for three different thresholds. From Figure 3.22 it can be seen that the convergence using tensors in the CP format converges not smoothly as the standard LCCD algorithm as implemented in NWchem for that no convergence acceleration like DIIS is used. Especially the large threshold of 10^{-2} introduces wiggles after some iterations and the calculated energy is even below the reference correlation energy obtained from the standard calculation. Nevertheless, it is still possible to converge the correlation energy to energy changes between two iterations that are still 2 orders of magnitude lower (10^{-4}). This is also true for the larger values of ε . Also the convergence gets much smoother if thresholds smaller than 10^{-4} are used for the rank reduction of all quantities.

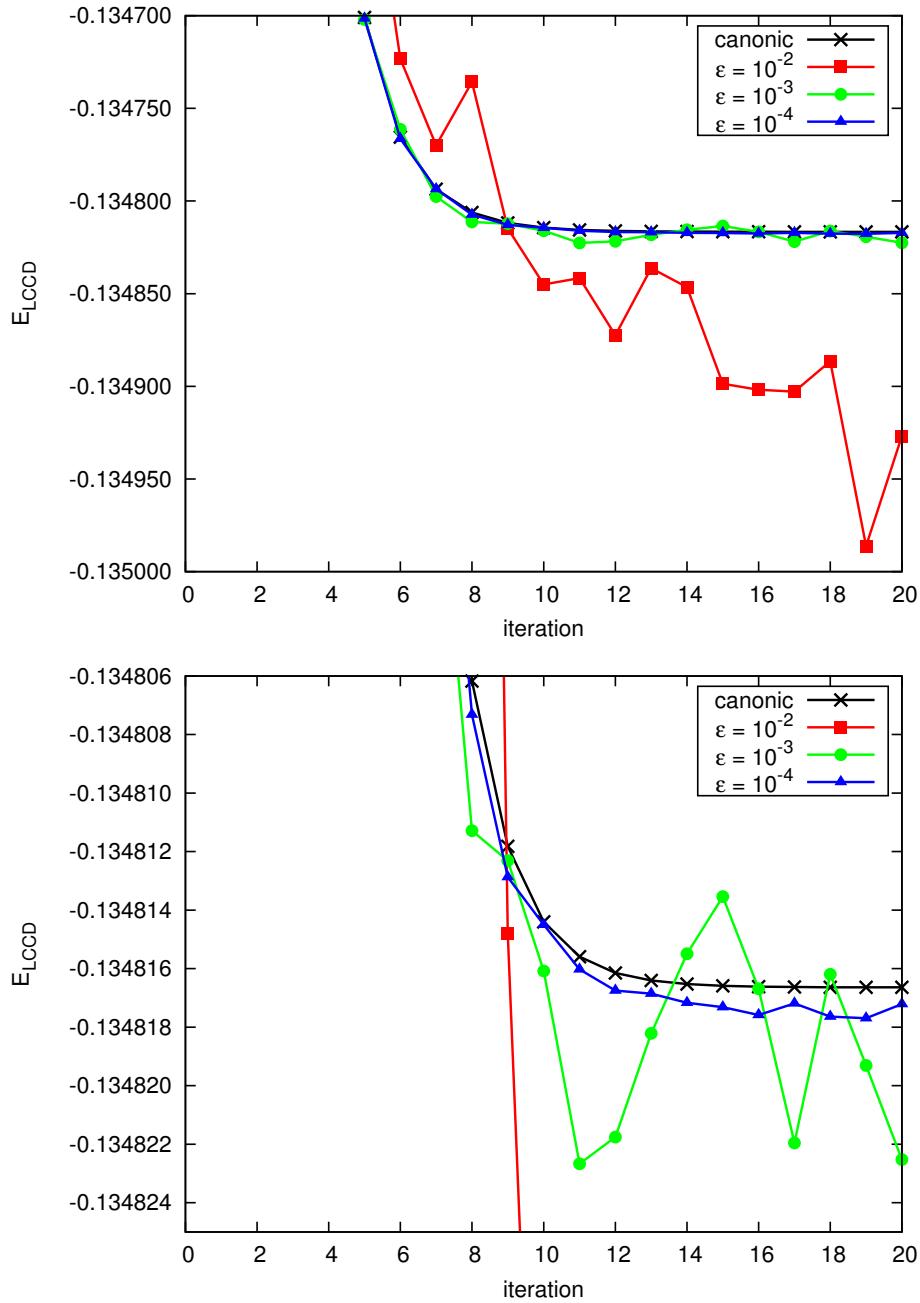


Figure 3.22: Convergence of LCCD correlation energy using CP tensors for H₂O in 6-31G basis set. Bottom diagrams shows a zoom in to see the convergence for $\epsilon = 10^{-3}$.

Another important quantity is the rank of the t_2 amplitude tensor and how it evolves during the iterative process. The ranks for H₂O using 6-31G basis set are listed in Table 3.19 together with the resulting errors in the correlation energy for three different accuracies. One can clearly see, that the reduced rank of the amplitudes is almost constant during the iterations. So the rank reduction is always capable of reducing the increased ranks of the residual

Table 3.19: Reduced ranks of t_2 amplitudes in CP format (Q) and error in LCCD correlation energy for H_2O using 6-31G basis set during iterative solution of amplitude equations.

iter.	$\varepsilon = 10^{-2}$		$\varepsilon = 10^{-3}$		$\varepsilon = 10^{-4}$	
	Q	ΔE_{LCCD}	Q	ΔE_{LCCD}	Q	ΔE_{LCCD}
1	51	$-1.565 \cdot 10^{-4}$	73	$7.092 \cdot 10^{-6}$	85	$-4.771 \cdot 10^{-7}$
2	51	$1.446 \cdot 10^{-4}$	74	$-1.771 \cdot 10^{-7}$	86	$-4.179 \cdot 10^{-7}$
3	52	$1.391 \cdot 10^{-4}$	74	$3.583 \cdot 10^{-6}$	86	$-8.111 \cdot 10^{-7}$
4	52	$1.059 \cdot 10^{-4}$	75	$-5.699 \cdot 10^{-6}$	85	$-7.491 \cdot 10^{-7}$
5	53	$2.145 \cdot 10^{-5}$	75	$-1.051 \cdot 10^{-6}$	86	$-4.949 \cdot 10^{-7}$
6	54	$4.474 \cdot 10^{-5}$	75	$4.192 \cdot 10^{-6}$	86	$-9.352 \cdot 10^{-7}$
7	53	$1.952 \cdot 10^{-5}$	73	$-3.917 \cdot 10^{-6}$	85	$1.100 \cdot 10^{-7}$
8	54	$-2.662 \cdot 10^{-5}$	73	$-5.113 \cdot 10^{-6}$	85	$-1.147 \cdot 10^{-6}$
9	53	$-3.460 \cdot 10^{-5}$	74	$-4.924 \cdot 10^{-7}$	87	$-1.054 \cdot 10^{-6}$
10	53	$2.632 \cdot 10^{-5}$	73	$-1.681 \cdot 10^{-6}$	86	$-1.011 \cdot 10^{-7}$
11	54	$-5.751 \cdot 10^{-5}$	76	$-7.075 \cdot 10^{-6}$	86	$-4.352 \cdot 10^{-7}$
12	53	$-2.314 \cdot 10^{-5}$	75	$-5.609 \cdot 10^{-6}$	86	$-6.006 \cdot 10^{-7}$
13	53	$-6.151 \cdot 10^{-5}$	72	$-1.809 \cdot 10^{-6}$	86	$-4.474 \cdot 10^{-7}$
14	54	$-8.192 \cdot 10^{-6}$	70	$1.035 \cdot 10^{-6}$	85	$-6.390 \cdot 10^{-7}$
15	54	$7.378 \cdot 10^{-6}$	74	$3.051 \cdot 10^{-6}$	86	$-7.303 \cdot 10^{-7}$
16	52	$-8.870 \cdot 10^{-5}$	74	$-7.380 \cdot 10^{-8}$	86	$-9.673 \cdot 10^{-7}$
17	51	$-1.240 \cdot 10^{-4}$	74	$-5.329 \cdot 10^{-6}$	86	$-5.588 \cdot 10^{-7}$
18	52	$-6.211 \cdot 10^{-5}$	74	$4.398 \cdot 10^{-7}$	86	$-1.006 \cdot 10^{-6}$
19	54	$-1.159 \cdot 10^{-4}$	73	$-2.669 \cdot 10^{-6}$	85	$-1.058 \cdot 10^{-6}$
20	55	$-1.061 \cdot 10^{-4}$	75	$-5.882 \cdot 10^{-6}$	85	$-5.687 \cdot 10^{-7}$

increments and also the larger rank of the amplitudes due to the denominator weighting to a similar low-rank representation. Furthermore the error propagation during the iterative procedure, where rank reduction is performed for already fitted quantities from the last iteration, seems to be smooth and the errors are cancelling each other out.

In summary, the iterative procedure can also be done using tensors in CP format. There, the ranks increase upon tensor contraction (product of the ranks of the two contracted tensors) or upon tensor summation (sum of the ranks) and the increased ranks have to be reduced using the reduction algorithm to fit a new tensor in CP format with lower rank. In general tensor contraction costs scale only with the ranks of the two corresponding tensors that are contracted and the contraction can be evaluated as simple scalar products between the representing vectors. Therefore, any contraction between the t_2 amplitudes (rank Q) and the two-electron integrals (rank R_μ) would scale approximately with $\mathcal{O}(N \cdot Q \cdot R_\mu)$, where N is

a measure of system or basis set size. Due to the scaling of the rank of the amplitudes and integrals as described in Sections 3.5 and 3.6 the overall scaling of the tensor contraction is approximately $\mathcal{O}(N^4)$ to $\mathcal{O}(N^5)$. Additional to the computational costs for the tensor contraction also the costs for the rank reduction have to be considered for the scaling of the full iterative procedure. The rank reduction procedure, in the original implementation, scales also approximately as $\mathcal{O}(N^5)$ if a sliced reduction is used but has a large prefactor. So the time determining step is now the rank reduction procedure itself, that has to be used during the iterative procedure multiple times in order to maintain a low-rank representation of the tensors.

For some type of contractions the tensor rank does not change, due to the fact that a more convenient transformation form can be used. There, only one representing vector of the target tensor depends on one expansion length, while the remaining three representing vectors depend on the other rank. This way the coefficient matrix arising from scalar products of the contracted tensor dimensions can be used to transform one of the representing vectors to match the expansion length of the other vectors by matrix-matrix multiplication. In this cases no further reduction steps are necessary and the computational costs are purely from the tensor contraction in the CP format. The numerical tests show, that the reduced ranks stay almost constant during the iterative procedure and the overall error in the calculated correlation energy is approximately two orders of magnitude lower than the desired deviations in the ℓ^2 -norm of all used tensor approximations in the CP format. Furthermore, the iterative procedure for obtaining amplitudes can be converged to changes in energy that are two orders of magnitude lower than the given threshold for the reduction procedure. Therefore, the ε parameter should be chosen two orders of magnitude larger than the desired convergence k in the energy

$$\varepsilon \approx k \cdot 10^2 \quad (3.108)$$

3.7.7 Rank Reduction during Iterative Procedure

As already mentioned before, the rank reduction during the iterative process is a crucial step to obtain a low-rank representation of all involved quantities in the CP format. Only with low ranks the scaling of computational costs for tensor contractions in CP format is advantageous. Furthermore, also the storage of intermediate tensors is only low scaling for a reduced representation in the CP format. So the rank reduction procedure is used after almost all tensor contractions (except if the ranks do not change upon contraction) and becomes the most important and time determining step of the whole algorithm. The overall scaling of computational costs is mainly governed by the scaling of the rank reduction itself. As pointed out in Section 3.2 the original implementation of the rank reduction fits a new tensor in an iterative procedure, where the ranks are increasing successively one by one, and one step scales approximately with the initial rank of the tensor and the current rank of the fitted object. Therefore, the reduction of large initial ranks is done in slices as presented in Figure 3.12, so that the initial rank can be kept fix and the whole rank reduction procedure can be done in parallel.

The iterative procedure of solving the amplitude equations allows also for a more efficient reduction of increased CP ranks. As for example the entries in the amplitudes change only little during the iterations and also the general structure of the tensor stays the same the tensor from the last iteration can be taken as an initial guess for the rank reduction. The algorithm then does not start with a rank one approximation but uses all representing vectors from the previous iteration to set up the residual tensor $\mathfrak{T} - \mathfrak{V}$ (c.f. 3.18). By adjusting all representing vectors the error in the ℓ^2 -norm can be decreased significantly and the desired accuracy can be already reached so that only a few new representing vectors need to be added in order to reach the given error in the ℓ^2 -norm. Thus, the rank is only increased slightly but the computational time for the rank reduction can be lowered significantly, as a large part of the iterative rank reduction procedure is skipped.

A numerical example of the initialization of the rank reduction procedure with tensors from the last iteration is given below for H₂O using the 6-31G basis set. There, the t_2 amplitudes are initialized by the CP representation from the last iteration and this way the rank of the amplitudes is a bit larger then for the reduction from scratch. It is also possible, that the ranks are always increasing during the iterative process especially if the changes compared to the last iteration are larger. To prevent this growth, one can apply a full rank reduction from time to time starting from scratch (rank one approximation) or starting from the initial amplitudes obtained after the zeroth iteration. A numerical example in Table 3.20 lists the ranks of t_2 amplitudes and the deviation of the LCCD correlation energy obtained for reductions from scratch or with initialization from the last iteration. There, one can see that the rank stays constant during the iterative procedure. Compared to the reduction from scratch, the ranks are larger in the case of initialization with the representation from the last iteration. Nevertheless, the ranks converge after a few iterations to a value that stays constant during the following iterative procedure. The error in the calculated correlation energy is again almost three orders of magnitude lower than the given accuracy of the rank reduction algorithm and comparable to the errors obtained with rank reductions from scratch. Also the convergence of the correlation energy is relatively smooth. It can be concluded, that the reduction algorithm is capable of adjusting small changes in the tensor by adjusting all representation vectors. So the rank reduction during the iterative solution of the amplitude equation can be done efficiently using an initial guess from the previous iteration. This way, the computationally most expensive part of the reduction (iterative growth of ranks one by one) can be skipped and only a single or a few iterations are necessary to obtain a low-rank representation of rank increased quantities. First tests for small molecules (water, methane) show, that the time for the rank reduction can be reduced by factor 2. For larger molecules even larger factors can be expected, as a large part of the fitting procedure (guessing new representing vectors and adjust them one by one) can be skipped.

This is not only true for the t_2 amplitudes but can also be applied to intermediate quantities or the residual increments. There, the low-rank approximation from the previous iteration

Table 3.20: Initialization of rank reduction during iterative calculation of t_2 amplitude tensor for LCCD method using H₂O with 6-31G basis set. For comparison the normal reduction (starting with a rank one approximation) is given. Only results for $\epsilon = 10^{-4}$ are shown. The initial guess of the first iteration are the corresponding two-electron integrals v_{ij}^{ab} .

iter.	reduction from scratch			initialization from last iteration		
	Q	E_{LCCD}	ΔE_{LCCD}	Q	E_{LCCD}	ΔE_{LCCD}
1	85	-0.1303771884	$-4.771 \cdot 10^{-7}$	99	-0.1303769160	$-2.047 \cdot 10^{-7}$
2	86	-0.1332705322	$-4.179 \cdot 10^{-7}$	106	-0.1332696961	$4.182 \cdot 10^{-7}$
3	86	-0.1341813991	$-8.111 \cdot 10^{-7}$	109	-0.1341808038	$-2.158 \cdot 10^{-7}$
4	85	-0.1345503466	$-7.491 \cdot 10^{-7}$	110	-0.1345501947	$-5.972 \cdot 10^{-7}$
5	86	-0.1347014769	$-4.949 \cdot 10^{-7}$	111	-0.1347011763	$-1.943 \cdot 10^{-7}$
6	86	-0.1347664370	$-9.352 \cdot 10^{-7}$	111	-0.1347662990	$-7.972 \cdot 10^{-7}$
7	85	-0.1347935426	$1.100 \cdot 10^{-7}$	111	-0.1347938401	$-1.875 \cdot 10^{-7}$
8	85	-0.1348073171	$-1.147 \cdot 10^{-6}$	111	-0.1348066672	$-4.970 \cdot 10^{-7}$
9	87	-0.1348128735	$-1.054 \cdot 10^{-6}$	111	-0.1348119762	$-1.565 \cdot 10^{-7}$
10	86	-0.1348145016	$-1.011 \cdot 10^{-7}$	111	-0.1348148935	$-4.930 \cdot 10^{-7}$
11	86	-0.1348160267	$-4.352 \cdot 10^{-7}$	111	-0.1348161831	$-5.916 \cdot 10^{-7}$
12	86	-0.1348167466	$-6.006 \cdot 10^{-7}$	111	-0.1348163294	$-1.834 \cdot 10^{-7}$
13	86	-0.1348168535	$-4.474 \cdot 10^{-7}$	111	-0.1348169946	$-5.885 \cdot 10^{-7}$
14	85	-0.1348171679	$-6.390 \cdot 10^{-7}$	111	-0.1348168440	$-3.150 \cdot 10^{-7}$
15	86	-0.1348173176	$-7.303 \cdot 10^{-7}$	111	-0.1348167277	$-1.405 \cdot 10^{-7}$

could also be used as an initial guess in the rank reduction of the next iteration. As storage in the CP format is not too expensive (only short representing vectors are stored) one could save all intermediate quantities and use them in the next iteration. However, this is not part of the current implementation.

3.8 CCD Algorithm with Tensors in Low-Rank CP Representation

From the LCCD method it is straight forward to extend the amplitude equations in order to derive the CCD method. There, the following higher order terms (contracting more than 2 tensors) have to be added to the residual equation (c.f. Eqn. 3.51)

$$\begin{aligned} & \frac{1}{4} \sum_{mnef} t_{ij}^{ef} t_{mn}^{ab} \langle mn||ef \rangle - P(ab) \frac{1}{2} \sum_{mnef} t_{ij}^{ae} t_{mn}^{bf} \langle mn||ef \rangle \\ & - P(ij) \frac{1}{2} \sum_{mnef} t_{im}^{ab} t_{jn}^{ef} \langle mn||ef \rangle + P(ij)P(ab) \frac{1}{2} t_{im}^{ae} t_{nj}^{fb} \langle mn||ef \rangle, \end{aligned} \quad (3.109)$$

leading to the CCD residual equation

$$\begin{aligned}
 r_{ij}^{ab} = & \langle ab\|ij\rangle + \frac{1}{2} \sum_{ef} t_{ij}^{ef} \langle ab\|ef\rangle + \frac{1}{2} \sum_{mn} t_{mn}^{ab} \langle mn\|ij\rangle + P(ij)P(ab) \sum_{me} t_{im}^{ae} \langle mb\|ej\rangle \\
 & - P(ij) \sum_m f_j^m t_{im}^{ab} + P(ab) \sum_e f_e^b t_{ij}^{ae} + \frac{1}{4} \sum_{mnef} t_{ij}^{ef} t_{mn}^{ab} \langle mn\|ef\rangle \\
 & - P(ab) \frac{1}{2} \sum_{mnef} t_{ij}^{ae} t_{mn}^{bf} \langle mn\|ef\rangle - P(ij) \frac{1}{2} \sum_{mnef} t_{im}^{ab} t_{jn}^{ef} \langle mn\|ef\rangle \\
 & + P(ij)P(ab) \frac{1}{2} t_{im}^{ae} t_{nj}^{fb} \langle mn\|ef\rangle. \tag{3.110}
 \end{aligned}$$

The additional terms can be solved in analogy to the other terms presented for the LCCD method leading to tensor contractions of three tensors in CP format. These can be calculated as scalar products over contracted indices/dimensions and copy operations of the remaining representing vectors. An example for the first term in Eqn. 3.109 is given below.

$$\mathfrak{r}_7^{ab} = \mathfrak{t}_{ij}^{ef} \mathfrak{t}_{mn}^{ab} \mathfrak{v}_{ef}^{mn} \tag{3.111}$$

$$\begin{aligned}
 &= \left(\sum_{q_1=1}^Q t_{q_1}^{(e)} \otimes t_{q_1}^{(f)} \otimes t_{q_1}^{(i)} \otimes t_{q_2}^{(j)} \right) \left(\sum_{q_2=1}^Q t_{q_2}^{(a)} \otimes t_{q_2}^{(b)} \otimes t_{q_2}^{(m)} \otimes t_{q_2}^{(n)} \right) \left(\sum_{r_1=1}^{R_1} v_{r_1}^{(m)} \otimes v_{r_1}^{(n)} \otimes v_{r_1}^{(e)} \otimes v_{r_1}^{(f)} \right) \\
 &\tag{3.112}
 \end{aligned}$$

$$\begin{aligned}
 &= \sum_{q_1=1}^Q \sum_{q_2=1}^Q \sum_{r_1=1}^{R_1} \left\langle t_{q_2}^{(m)}, v_{r_1}^{(m)} \right\rangle \left\langle t_{q_2}^{(n)}, v_{r_1}^{(n)} \right\rangle \left\langle t_{q_1}^{(e)}, v_{r_1}^{(e)} \right\rangle \left\langle t_{q_1}^{(f)}, v_{r_1}^{(f)} \right\rangle t_{q_2}^{(a)} \otimes t_{q_2}^{(b)} \otimes t_{q_1}^{(i)} \otimes t_{q_1}^{(j)}. \tag{3.113}
 \end{aligned}$$

This term contracts all indices/dimensions of the MO integral with the amplitudes and therefore the summation over rank R_1 can be carried out. At the same time the two expansion lengths q_1 and q_2 are merged together leading to a new expansion w_7 and the final residual increment is obtained as

$$[\mathfrak{r}_7]_{ij}^{ab} = \sum_{w_7=1}^{W_7=Q \cdot Q} \tau_{l_7}^{(a)} \otimes \tau_{l_7}^{(b)} \otimes \tau_{l_7}^{(i)} \otimes \tau_{l_7}^{(j)}. \tag{3.114}$$

Therefore, also for this contraction the rank is increased and the reduction algorithm is needed to obtain a low-rank representation of the increment. All further additional terms in Eqn. 3.110, that are higher order in t_2 , can be evaluated in the same way. But it is also possible to reduce the computational complexity by factorizing the equations. [237] In this approach similar contractions that use the same amplitude are combined to intermediates, that are finally contracted to the common tensor. Therefore, the tensor contractions can be factorized and a more complex structure of the CC amplitude equations is obtained.

For the CCD algorithm this procedure leads to the following intermediates

$$\mathcal{F}_e^b = f_e^b - \frac{1}{2} \sum_{mnf} t_{mn}^{bf} \langle mn\|ef \rangle \quad (3.115)$$

$$\mathcal{F}_j^m = f_j^m - \frac{1}{2} \sum_{nef} t_{jn}^{ef} \langle mn\|ef \rangle \quad (3.116)$$

$$\mathcal{W}_{ij}^{mn} = \langle mn\|ij \rangle + \frac{1}{2} \sum_{ef} t_{ij}^{ef} \langle mn\|ef \rangle \quad (3.117)$$

$$\mathcal{W}_{ef}^{ab} = \langle ab\|ef \rangle \quad (3.118)$$

$$\mathcal{W}_{ej}^{mb} = \langle mb\|ej \rangle + \frac{1}{2} \sum_{fn} t_{nj}^{fb} \langle mn\|ef \rangle, \quad (3.119)$$

that can be used in the modified amplitude equation defining the residual tensor as

$$\begin{aligned} r_{ij}^{ab} &= \langle ab\|ij \rangle + P(ab) \sum_e \mathcal{F}_e^b t_{ij}^{ae} - P(ij) \sum_m \mathcal{F}_j^m t_{im}^{ab} \\ &\quad + \frac{1}{2} \sum_{mn} t_{mn}^{ab} \mathcal{W}_{ij}^{mn} + \frac{1}{2} \sum_{ef} t_{ij}^{ef} \mathcal{W}_{ef}^{ab} + P(ij)P(ab) \sum_{me} t_{im}^{ae} \mathcal{W}_{ej}^{mb}. \end{aligned} \quad (3.120)$$

It should be noted that this concept can be easily extended to higher order CC methods [238] also using effective two particle excitation operators (τ as a combination of t_2 and products of t_1).

Using a low-rank CP representation for the amplitudes and integrals leads also to a CP representation of the intermediate tensors in Eqns. 3.115–3.119. As the ranks increase upon summation and tensor contraction (except for contractions in second terms in Eqns. 3.115 and 3.116 that can be written as a rank transformation; c.f. Sec. 3.7.4) a rank reduction is necessary for the intermediate tensors as well. However, using the factorization approach one can minimize the number of tensor contractions needed, or more specifically one can reduce the initial ranks for a reduction as the summation of two tensors only adds the two ranks, while two separate tensor contractions each multiply the ranks. Therefore, the actual implementation of the CCD algorithms is done according to Eqn. 3.120.

The first test implementation is done by hand (coding each tensor contraction separately) but the simple structure of tensor contractions in the CP format - evaluation of scalar products over contracted dimensions and copy operation for the remaining vectors followed by a rank reduction - also allows for a more efficient treatment of the contractions. There, only a simple contraction kernel is needed to treat various types of tensor contractions controlled by an outer routine that classifies the contraction. This makes further modification in the contraction and rank reduction procedure, such as using an initial guess from last iteration or a sliced reduction on the fly while contraction two tensors, much easier. For the implementation of the outer classification routine, that takes care of the contracted dimensions and possible

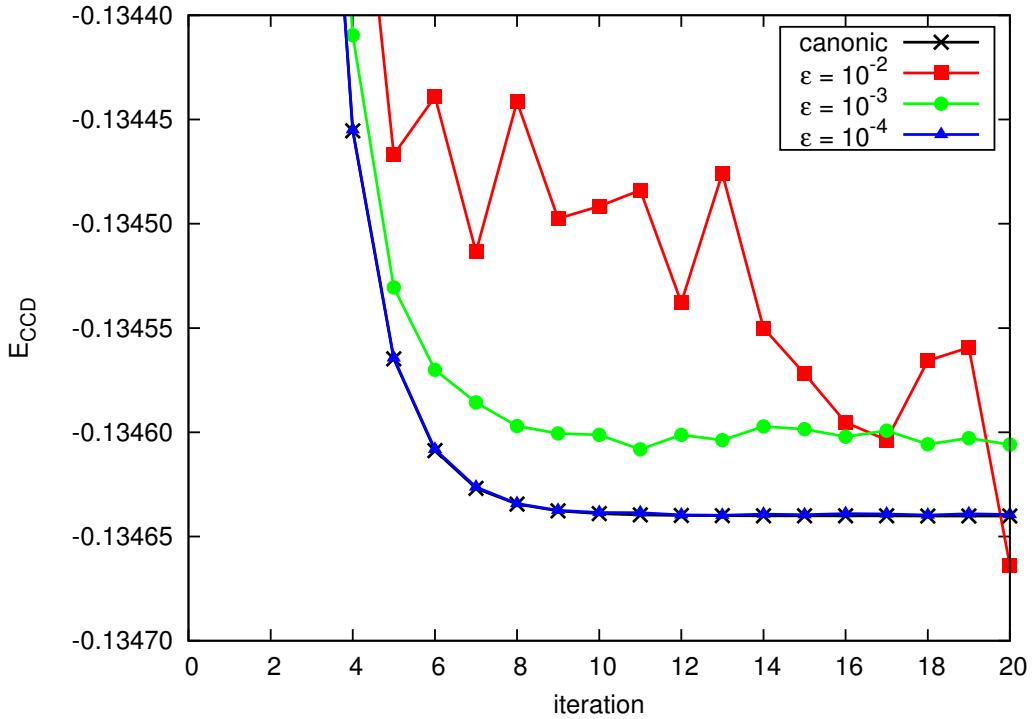


Figure 3.23: Convergence of CCD correlation energy using CP tensors for H₂O in 6-31G basis set. Note that the error in the correlation energy for $\epsilon = 10^{-3}$ is in the order of 35 μ Hartree.

permutations, the technique of automatic code generation is used in order to provide simple and reliable computer code for a large number of tensor contractions. [190, 191, 235, 239, 240] Especially for further implementations of higher order CC methods such a code generator is vital, as the number of tensor contractions increases rapidly. For this purpose, a simple `python` program is developed that is able to translate simple mathematical equations into FORTRAN or C++ code using tensors in different representation formats (for details see A.1.2). With this tool the CCD equations are generated and the structure of contractions or the definition of various intermediates can easily be varied by simple changes in the mathematical equations given as input. So also the implementation of higher order CC methods becomes straight forward.

Results obtained with the automatic generated contraction routines for the CCD method are shown in Fig. 3.23 and Tab. 3.21 for water in 6-31G basis using 3 different approximation thresholds. From Fig. 3.23 it can be seen that the convergence of the CCD correlation energy is more systematic than for the LCCD method. Only for $\epsilon = 10^{-2}$ the convergence is not so smooth and shows some jumps. Nevertheless, it is still possible to converge the energy to a threshold that is almost 2 orders of magnitude lower than the accuracy of the rank reduction. The errors are comparable to the deviations in the LCCD method but the energy approaches the reference value from the standard calculation more systematically from above.

Table 3.21: Reduced ranks of t_2 amplitudes in CP format (Q) and error in CCD correlation energy for H₂O using 6-31G basis set during iterative solution of amplitude equations. The rank reduction is not initialized with a guess from the previous iteration and starts from scratch.

iter.	$\epsilon = 10^{-2}$		$\epsilon = 10^{-3}$		$\epsilon = 10^{-4}$	
	Q	ΔE_{CCD}	Q	ΔE_{CCD}	Q	ΔE_{CCD}
1	49	$1.9700 \cdot 10^{-4}$	73	$9.2779 \cdot 10^{-5}$	86	$8.6482 \cdot 10^{-7}$
2	52	$2.7167 \cdot 10^{-4}$	73	$5.9433 \cdot 10^{-5}$	86	$9.2519 \cdot 10^{-7}$
3	50	$1.7844 \cdot 10^{-4}$	74	$4.7391 \cdot 10^{-5}$	86	$3.8862 \cdot 10^{-7}$
4	54	$1.7470 \cdot 10^{-4}$	73	$4.5864 \cdot 10^{-5}$	85	$4.8779 \cdot 10^{-7}$
5	53	$9.7989 \cdot 10^{-5}$	73	$3.4316 \cdot 10^{-5}$	86	$4.7725 \cdot 10^{-7}$
6	53	$1.6975 \cdot 10^{-4}$	74	$3.8737 \cdot 10^{-5}$	86	$5.6659 \cdot 10^{-7}$
7	53	$1.1338 \cdot 10^{-4}$	74	$4.1138 \cdot 10^{-5}$	86	$5.7919 \cdot 10^{-7}$
8	54	$1.9303 \cdot 10^{-4}$	73	$3.7389 \cdot 10^{-5}$	85	$2.7125 \cdot 10^{-7}$
9	53	$1.4020 \cdot 10^{-4}$	75	$3.7147 \cdot 10^{-5}$	85	$2.1594 \cdot 10^{-7}$
10	53	$1.4742 \cdot 10^{-4}$	76	$3.7798 \cdot 10^{-5}$	86	$5.4043 \cdot 10^{-7}$
11	53	$1.5580 \cdot 10^{-4}$	73	$3.1502 \cdot 10^{-5}$	86	$1.0582 \cdot 10^{-6}$
12	53	$1.0220 \cdot 10^{-4}$	73	$3.8705 \cdot 10^{-5}$	86	$3.1286 \cdot 10^{-7}$
13	54	$1.6410 \cdot 10^{-4}$	74	$3.6268 \cdot 10^{-5}$	86	$2.6024 \cdot 10^{-7}$
14	54	$9.0011 \cdot 10^{-5}$	73	$4.2865 \cdot 10^{-5}$	86	$7.6721 \cdot 10^{-7}$
15	52	$6.8163 \cdot 10^{-5}$	74	$4.1554 \cdot 10^{-5}$	86	$5.5069 \cdot 10^{-7}$
16	53	$4.4873 \cdot 10^{-5}$	74	$3.7948 \cdot 10^{-5}$	86	$1.0763 \cdot 10^{-6}$

To give just a small impression on the computational effort of the tensor contractions and the rank reduction compared to the standard calculation the timings for a single CCD iteration on a single core of a workstation are listed in Tab. 3.22. For the CP format the full timing is further divided into the reduction time and the pure tensor contraction part. There it can be seen that the rank reduction in the current implementation becomes the most dominant part, while the actual time for the pure tensor contraction is barely visible at all. The first implementation using a full rank reduction for tensors with increased ranks (after contraction) is much too slow compared to the standard calculation. There the increased initial ranks lead to the higher CPU time as the ranks reduction algorithm scales with the initial rank. So a similar procedure as for the reduction of the integrals is used: the tensor contraction, that increases the rank, is only done in slices so that a fixed initial rank for the reduction can be chosen. After this sliced reduction the merged rank is still larger than one would expect from the full reduction so that a further full reduction of the prereduced tensor can be done. This leads to similar ranks as obtained from the single full reduction and also the accuracy in the energy is comparable, but the computational costs can be low-

Table 3.22: CPU time per iteration on a single workstation core. For the algorithm using tensors in CP format a threshold of 10^{-2} is applied and the reduction of MO integrals is excluded from the timing.

method	time per iteration [s]
canonical	< 1
CP; full reduction	677 (reduction: 676; contraction: < 1)
CP; sliced reduction (250)	118 (reduction: 117 ; contraction: < 1)

ered significantly. Nevertheless, the algorithm using sliced reductions is more than 100 times slower than the standard calculation due to the high perfactor of the rank reduction algorithm.

Therefore, in the following section some modifications of the rank reduction procedure are presented. These mainly utilize some physical properties of the tensors, like the general structure and dominant parts of the tensor, and can also be used for the contraction of two-electron integrals, which is taken as an example.

3.9 Improved Rank Reduction Algorithm

As presented in the last section a low-rank representation can help to reduce the effective scaling of tensor contractions. Nevertheless, the ranks can increase upon the contraction procedure so that a rank reduction is inevitable, especially in an iterative solution of the amplitude equations. The computational costs for the successive rank reduction become time determining, while the pure contractions in the CP format is barely seen. Therefore, it is necessary to improve the general order rank reduction procedure and to adapt the algorithm to tensors occurring in post Hartree-Fock methods.

For this purpose, the original object oriented C++ implementation is profiled in order to determine the main bottlenecks of the reduction procedure. It turned out that some time is wasted during memory allocation, deallocation and copy operations due to the general order structure of the algorithm, that is often hidden in the structure of the tensor objects. Furthermore, some parts of the algorithm, as the optimization of the already existing vectors or the evaluation of the gradient, are done using low level BLAS (Basic Linear Algebra Subprograms) routines dealing only with vectors. Therefore, the general order C++ code is rewritten as in plain FORTRAN90 dealing only with fourth order tensors. This makes the memory management and the adaption of the reduction algorithm much easier.

The first attempt keeps the structure of the original implementation of the reduction algorithm (c.f. Fig. 3.5): choose a good rank one approximation, optimize it further, add the representing vector to the set of the already existing vectors, improve all vectors in a numer-

ical procedure and finally calculate the gradient to further improve the approximation. The main difference is the substitution of low level BLAS routines with high level matrix-matrix or matrix-vector multiplications. Furthermore, the fixed tensor order allows for a more efficient memory allocation and less copy operations during the approximation procedure. This way the increasing rank during the iterative procedure can be easily handled.

With these simple improvements it is already possible to speed up the reductions by a factor of two to three (see also Tab. 3.23), but the reduction is still too slow compared to a standard calculation itself. The actual ranks are not exactly the same due to the usage of a slightly different routine to calculate the inverse of a matrix, but nevertheless the values are comparable.

The next point for improving the algorithm is the successive increasing rank (one by one) during the iterative process. There, a step size parameter is introduced allowing the algorithm to collect more rank one approximations before improving the vectors and calculating the gradient. This way the number of iterations can be decreased significantly, while the final ranks for a given accuracy are slightly increased (see Tab. 3.23). If the step size is chosen too large the final ranks can be much too large (compared to successive rank one approximations) leading to problems with tensor contractions where the rank of the target tensor is the product of the ranks of the two contracted tensors. On the other hand, a lower step size parameter leads to an increased number of iterations and therefore increases the computational effort for the reduction. Numerical tests show that for the two-electron integrals a step size in the order of the number of basis functions leads to a good compromise between the increase in the final rank and a decrease of the computational time. The same results can be obtained for MO integrals or amplitudes using a step size that is approximately the number of virtual orbitals.

Also the guess for the rank one approximation can be optimized. There, the original algorithm used a random initial guess to define an index tuple, that defines a special cross in all dimensions (cross approximation). Afterwards, it searches for large entries on this cross and defines a new tuple with a new cross for the maximum value. This procedure is repeated until a large entry in the tensor is found and a final cross for the rank one approximation is obtained. To accelerate the procedure for finding large values in the tensor one can exploit the knowledge of the structure of the tensor. The two-electron integrals in the AO basis, for example, have a block diagonal structure (c.f. Fig. 3.8). Therefore, it is convenient to replace the random initial guess of the tuple with an element from $\langle ii||jj\rangle$, $\langle ij||ij\rangle$ or $\langle ij||ji\rangle$. These can be further sorted in size so that the initial guess should always be a large element in the tensor. If the following cross approximation finds no larger entry (which should be the case) the new rank one approximation is immediately found and one does not need to evaluate more crosses. Therefore, the computational time can again be lowered without further affecting the accuracy or the final ranks (c.f. Tab. 3.23; method termed diagonal guess).

Table 3.23: Timings for different improvements of the reduction algorithm: AO two-electron integrals in H₂O using 6-31G and 6-31G* basis with $\varepsilon = 10^{-4}$. The CPU time is measured on a single core of a workstation (c.f. A.1.3).

algorithm	6-31G		6-31G*	
	final rank	time [s]	final rank	time [s]
original implementation	343	1666	732	12547
new implementation	349	621	815	7652
step size = 10	390	155	910	1299
step size = 20	400	144	940	853
step size = 20, diagonal guess	400	61	960	560
step size = 20, diagonal guess, SVD	440	52	880	332
comparison: CCSD calculation		3		4

Furthermore it is also possible to combine these improvements with the SVD based casting procedure (see Sec. 3.1.2) or even use the sliced reduction based on the SVD casting (c.f. Sec. 3.3.5) to lower the initial rank or keep it fix for a selected slice size. This again can increase the final ranks slightly while reducing the computational time further. In the given numerical examples (see Tab. 3.23) the SVD procedure itself reduces the initial ranks only slightly, so that the actual gain in time especially for the smaller basis set is rather low. For larger molecules the SVD based casting procedure together with the slicing scheme can have much larger effects, as the computational effort for the rank reduction, especially for calculating the gradient, scales with the initial rank of the tensor.

Altogether the improvements in the reduction procedure sum up to a decrease of computational time by a factor of approximately 35 for the given examples. It can be expected that the reduction of computational effort is even larger for larger examples or larger basis sets, where more and more iterations are needed for the approximation procedure. These improvements can be seen as a first test to advise useful modifications for the design of a “next generation” rank reduction procedure tailored for the needs in electronic structure methods.

The rank reduction algorithm uses some internal threshold parameters that determine for example the quality of the improved vectors but on the other hand also can increase the computational time. These have been adjusted to suitable values for two-electron integrals and t_2 amplitudes. The actual values are listed in the appendix (see A.4) together with some numerical tests for the accuracy and the computational time.

4 Conclusions and Outlook

In this work the application of tensor representation in the low-rank canonical polyadic (CP) format has been investigated in the context of post Hartree-Fock methods. After an introduction into the theory of electronic structure calculations especially focusing on CI based correlation methods, various tensor formats were compared in terms of the decomposition type, storage and manipulation costs. Some formats, like the MPS or Tucker format, are easier to obtain in a straightforward algorithm using SVD. But these tensor formats do not offer a full factorization of all dimensions. Therefore, all following tensor manipulations become more difficult and higher scaling in terms of the expansion length. On the other hand, the fully factorized low-rank CP representation of higher order tensors can be obtained by a more complex casting procedure. There, the ranks can be high and a rank reduction step involving nonlinear optimizations (not comparable to the simple SVD) is necessary for a low-rank representation. Nevertheless, the compact and fully factorized form of the CP format makes it attractive for application in post Hartree-Fock methods. In a low-rank CP representation tensor contraction costs scale only linear with the number of dimensions of the tensor compared to the exponential scaling for tensors in an index based representation.

In the CP representation the most important quantity is the actual length of the expansion. By full factorization of all dimensions of a given tensor one is able to break the “curse of dimensionality”. Nevertheless, the computational costs of mathematical operations dealing with tensors in CP representation still depend linearly on the number of dimensions and also on the rank of the expansion. Therefore, the reduced ranks for CP representation of tensors occurring in post Hartree-Fock methods were taken as an example to investigate the applicability of these new decomposition/representation technique.

The two-electron integrals in the AO basis were chosen as an initial example and various casting procedures using the full four dimensional tensor, RI integrals and a scheme for treating larger quantities based on a SVD have been devised and the resulting expansion lengths have been investigated. In order to obtain a low rank approximation of the cast tensor in CP format a rank reduction procedure developed by M. Espig [121, 122, 145] has been used as a “black box” tool. As this rank reduction in the original implementation only allows for small initial ranks a slicing scheme has been introduced, which makes the treatment of larger examples with a fixed initial rank possible. Furthermore, this procedure circumvents the steep scaling of the reduction procedure with the initial rank. At the same time the sliced reduction also allows for a more efficient parallel reduction scheme of different slices on different cores of a modern workstation computer or the distribution of reduction task on a larger PC cluster.

The scaling of the ranks with increasing basis and system size has been investigated as this finally determines the storage requirements and also the computational cost for further tensor manipulations. For the AO two-electron integrals the scaling for the rank with system and basis set size is found to be $\mathcal{O}(N^{1.8})$ if a sliced reduction followed with $\varepsilon = 10^{-2}$ is applied. A subsequent full reduction reduces the rank further but does not influence the scaling behaviour of the tested systems (LiH chain and water molecule) much. In contrast to the full AO integral tensor, for which the rank scales as $\mathcal{O}(N^3)$, storage requirements are decreased from N^4 to $\approx 4 \cdot N^{2.8}$. For better accuracies the scaling is a bit larger ($\mathcal{O}(N^2)$) but nevertheless the memory requirements can be reduced in the low-rank CP representation.

The AO MO transformation of integrals in CP format has been done by simple matrix vector multiplications of the coefficient matrices with the corresponding representing vectors. Although this transformation does not change the rank, it is possible to compress the rank of the MO integrals further. The scaling of the rank with system size can be reduced to $\mathcal{O}(n^{1.2})$ for v_{ij}^{ab} with $\varepsilon = 10^{-2}$ in the LiH chain or slightly higher depending on the type of MO integral and applied ε -parameter. As a consequence, the memory required for storage of the MO integrals is reduced to $\approx 4 \cdot N^{2.2}$. Furthermore, the complexity of the transformation operation itself is decreased from the formal $\mathcal{O}(N^5)$ scaling of the canonical transformation to approximately $\mathcal{O}(N^{3.7})$, assuming a $\mathcal{O}(N^{1.7})$ scaling of the AO integrals.

As a first estimate for the t_2 amplitudes in CC methods the amplitudes as obtained from MP2 have been investigated. The amplitudes in CP representation are obtained by weighting the already transformed and reduced MO integrals v_{ij}^{ab} with the energy denominator. Therefore, the energy denominator is converted to the CP format in a Laplace-like decomposition with a fixed rank. The multiplication of the integral and the denominator tensor increases the rank and the reduction procedure is needed to obtain a low-rank representation. After the reduction the rank of the amplitudes is found to be almost linear scaling with system and basis set size: $\mathcal{O}(n^{1.3})$ or $\mathcal{O}(N^{1.2})$ for $\varepsilon = 10^{-2}$. A summary for the scaling of ranks with system and basis set size is given in Table 4.1. Especially the low scaling of ranks for the wavefunction parameters with increasing basis set size should be mentioned as other approximations like local correlation methods often do not exhibit this advantageous scaling with basis set size.

Once the amplitudes and integrals are available in the CP format, the MP2 energy can be calculated by simple scalar products of the corresponding representing vectors. Using amplitudes and integrals that have been reduced with ε values between 10^{-1} to 10^{-2} still yield mHartree accuracy in the MP2 energy. With slightly smaller values of ε even μ Hartree accuracy can be achieved. The complexity of the evaluation of the energy expression in the CP format scales approximately as $\mathcal{O}(N^{3.4})$ if a scaling of $\mathcal{O}(N^{1.2})$ for the integrals and $\mathcal{O}(N^{1.2})$ for the amplitudes is assumed ($\varepsilon = 10^{-2}$). However, as the conventional algorithm based on an index based representation of the integrals scales with $\mathcal{O}(N^4)$, the complexity for the MP2

Table 4.1: Scaling of reduced ranks for two-electron integrals and t_2 amplitudes as obtained from MP2 with increasing system (n) and basis set size (N). Only the exponents of the fitted functions are shown. It should be noted that the four occupied integrals have a constant rank with increasing basis set size.

ε	\mathfrak{v}_{ij}^{ab}		\mathfrak{v}_{cd}^{ab}		\mathfrak{v}_{kl}^{ij}		\mathfrak{t}_{ij}^{ab}	
	n	N	n	N	n	N	n	N
10^{-2}	1.2	1.2	2.1	2.4	1.9	-	1.3	1.2
10^{-4}	1.5	1.4	2.7	2.6	2.0	-	1.9	1.2
10^{-6}	2.5	1.5	2.8	2.8	2.3	-	2.4	1.3

energy equation can not be reduced much by the introduction of decomposed tensors in the CP format.

Furthermore, it is possible to use the reduced CP representation of the MO integrals in iteratively solving CC amplitude equations. There, increments to the residual tensor are build up by tensor contractions of integrals and amplitudes in CP representation. The contraction itself can be evaluated by simple scalar products over inner contraction variables and copy operations for the remaining representing vectors. The computational effort for the contractions scales as $\mathcal{O}(N \cdot R \cdot Q)$ leading to an overall scaling of approximately $\mathcal{O}(N^4)$ to $\mathcal{O}(N^6)$ assuming a scaling of ranks for the amplitudes of $\mathcal{O}(N)$ to $\mathcal{O}(N^2)$ and for the integrals of $\mathcal{O}(N^2)$ to $\mathcal{O}(N^3)$. Compared to the formal N^6 scaling of the CCD method this is only an improvement if the lower boundaries are considered (for larger values of ε). For higher accuracies the formal scaling of the tensor contraction stays the same. A short overview over the scaling of computational costs for some example tensor contractions is given in Table 4.2. Nevertheless, in principle also contractions of higher order tensors should exhibit a similar scaling behaviour as long as the scaling of ranks for the contracted tensors is low. For example, the perturbative treatment of triple substituted determinants in the famous CCSD(T) approach has a conventional scaling of $\mathcal{O}(N^7)$, while in the CP format exactly the same contraction costs as for the CCD method (approximately $\mathcal{O}(N^4)$ to $\mathcal{O}(N^6)$) arise. Therefore, the CP format becomes more advantageous for higher order methods from the CC hierarchy.

Upon tensor contraction the rank of the target tensor can increase depending on the type of contraction. As these target tensors are needed for further manipulations (amplitude update) and enter the next iteration, the rank reduction procedure is vital to obtain a low-rank representation of the residual tensor. The increased ranks can be very large so that the slicing scheme for a faster rank reduction can be applied “on the fly” during the contraction procedure. A subsequent full rank reduction can reduce the rank further and it is convenient to use the tensor from the previous iteration as an initial guess, which again speeds up the calculation. Altogether it is possible to converge the correlation energy to values that are

Table 4.2: Scaling of computational cost for various tensor manipulations with increasing basis set size. The prefactors are shown to distinguish the second and third contraction. For comparison the conventional scaling using tensors in index based representation is listed.

contraction	$\varepsilon = 10^{-2}$	$\varepsilon = 10^{-4}$	$\varepsilon = 10^{-6}$	conventional
$r_{ij}^{ab} \leftarrow \sum_{ef} t_{ij}^{ef} v_{ef}^{ab}$	$2 \cdot N^{4.6}$	$2 \cdot N^{4.8}$	$2 \cdot N^{5.1}$	$virt^4 \cdot occ^2 \approx N^6$
$x_{ij}^{mn} \leftarrow \sum_{ef} t_{ij}^{ef} v_{ef}^{mn}$	$2 \cdot N^{3.4}$	$2 \cdot N^{3.6}$	$2 \cdot N^{3.8}$	$virt^2 \cdot occ^4 \approx N^6$
$E \leftarrow \sum_{efmn} t_{mn}^{ef} v_{ef}^{mn}$	$4 \cdot N^{3.4}$	$4 \cdot N^{3.6}$	$4 \cdot N^{3.8}$	$virt^2 \cdot occ^2 \approx N^4$

approximately two orders of magnitude lower than the given accuracy for the rank reduction procedure. However, the convergence is not smooth and the calculated correlation energy does not approach the reference value systematically from above or below.

Although the computational cost for tensor contraction itself can be low scaling with increasing system or basis set size one still needs the rank reduction procedure if the rank increases upon the contraction. This rank reduction then becomes the time determining step as the original implementation is a factor of approximately 1000 slower than time needed for the contraction (for small examples). Therefore, one can try to adapt the reduction procedure to the requirements for tensors in post Hartree-Fock methods. With some simple improvements in the memory structure, the rigorous usage of higher level BLAS routines, the substitution of the random initial guess by a diagonal guess and an adaption of the step size in the iterative approximation the performance of the reduction procedure can be improved by approximately a factor of 30 (for H₂O in 6-31G basis). For larger examples the improvements can be expected to be even larger as also a sliced decomposition can be used together with a parallelization scheme.

In conclusion, tensor decomposition techniques¹ and their rigorous application bear the potential to overcome the “curse of dimensionality” for post Hartree-Fock ab-initio methods. If all quantities are expressed as decomposed representing vectors the high scaling of tensor contractions and storage requirements are drastically reduced through tensor decomposition and low-rank approximation. There, any tensor contraction only scales with the rank of the two tensors and no longer exponentially with the number of dimensions. First numerical tests show an overall scaling of contractions with $\mathcal{O}(N^4)$ to $\mathcal{O}(N^6)$ for fourth order tensors but a

¹It should be mentioned, that “decomposition” and “representation” are used interchangeably throughout this work, although a decomposition implies some kind of analysis, while representation implies a more general usage for further operations.

similar scaling can be expected also for contractions of higher order tensors. Ultimately, the rank reduction algorithm will become the time determining step with a scaling of approximately $\mathcal{O}(N \cdot \text{initial rank} \cdot \text{final rank})$. Taking a simple example of contracting integrals and t_2 amplitudes can lead to an increased rank of the residual tensor that scales approximately as $\mathcal{O}(N^3)$. A rank reduction of this residual tensor than exhibits a scaling of $\mathcal{O}(N^5)$, if the scaling with the number of iterations of the current algorithm is eliminated and the final quantity has a similar rank as the amplitudes. This high scaling of the reduction procedure can be further decreased if the initial tensor is sliced into smaller parts. Treating the representing vectors by slices of ranks also allows for an efficient parallelization of the reduction and contraction procedures.

4.1 Outlook

As the main bottleneck is the rank reduction procedure itself, further studies can help to improve the reduction algorithm by adapting the procedure to the special requirements in post Hartree-Fock methods. For example, the iterative improvement of the representing vectors and the exact gradient step can be united in a single procedure of lower complexity. One can also try to impose a special structure of the representing vectors in order to map special symmetry elements of the original tensors. With an improved rank reduction procedure it is straightforward to implement also higher order CC methods especially with the help of automatic code generation.

The increase in ranks upon tensor contractions can further be reduced by a intermediate reassembly step and a decomposition of the reassembled quantity into the CP format. This is only advantageous if the dimension of the reassembled tensor is two. In this case the reassembled matrix is decomposed with a SVD and the eigenvalues can be used to truncate the number of eigenvectors used as new representing vectors. This way the rank of the target tensor is still increased but with N times the smallest rank of the two contracted tensors lower than the product of the two ranks (see also A.2.4). A similar procedure can also be used for a reassembled quantity in higher dimensions using a trivial decomposition into the CP format followed by a rank reduction.

A further possibility is to use the CP format for full CI type calculations. There very high dimensional quantities need to be processed, while the length in each dimension may be short. This situation is well suited for a representation in the CP format as only short representing vectors have to be stored and handled. For CI type methods one can chose a similar formulation as known for CC type calculations leading to a linear set of equations with 13 different tensor contractions for each order of coefficient parameters (for a detailed analysis see Appendix A.3). These tensor contractions can be done in analogy to the method presented for the CC iterations only including multiple index/dimension permutations. This

way it should in principle be possible to solve the full CI equation without exponential scaling of the computational effort with increasing number of electrons.

Furthermore, other tensor formats can be tested for the application in CI or CC based methods. For example, the MPS format exhibits a similar scaling behaviour of the effective ranks like the scaling of ranks in the CP format. [241] Also the increase of rank during multiple tensor contractions can be prevented if different orders of dimensions for the same tensor are considered. The great advantage of the MPS format is the easy decomposition and rank reduction procedure based on SVD, but the simple addition of two tensors in MPS format with different expansion lengths is a bit more complicated. The Tucker format can also be evaluated for tensor contractions of amplitudes and integrals. Although the Tucker representation does not break the “curse of dimensionality” (still contains d -th order core tensor) it should still be possible to find a low-rank representation with lower complexity than in the canonical, index based representation. The advantage of the Tucker format for evaluating contractions is the fact, that the ranks do not change upon contraction and only manipulations of the core tensor have to be considered, while the actual representing vectors stay constant. Nevertheless, additions of two Tucker representations are again not straightforward and can increase the rank of the representation.

A Appendix

A.1 Computational Details

A.1.1 Software

All FORTRAN and C++ code has been compiled using the Intel® Fortran Intel® 64 Compiler Professional, Version 11.0.069 and the Intel® C++ Intel® 64 Compiler Professional, Version 11.0.069, respectively. Basic linear algebra operations were done using Intel® MKL library Version 10.1. For profiling the GNU gprof as well as an evaluation copy of Intel® VTune™ Amplifier XE 2011 (build 206420) have been used. Short conversion scripts or automatic analysis of data has been done by small `python` scripts using Python 2.7.3.

This manuscript has been written using L^AT_EX and all graphs were made by `gnuplot` while schemes were drawn using `inkscape` or `gimp`.

The original implementation of the rank reduction algorithm using the accelerated gradient method has been provided by the open source library `TensorCalculus` from Mike Espig [114, 121, 122, 145] as C++ routines that have been further modified for the treatment of quantities from quantum chemical methods. The two electron integrals and other data related to post Hartree-Fock calculations were obtained from a local modified version of CFOUR [226, 227] or a modified version of the TCE [234] from NWChem Version 5.8 [236]. The RI integrals and the metric matrix were obtained from a modified version of ORCA 2.8 [242–244].

A.1.2 Automatic Code Generation

An automatic code generator has been developed in `python` using object oriented techniques. As input simple mathematical equations defining tensor contractions with proper indexes are needed. The input can further be structured into sub-groups allowing for efficient treatment of intermediate tensors and structure of the tensor contraction. The tool evaluates the contraction by decomposing the expression in `tensor` objects that are linked through common indices. All memory management and possible resorting of tensors can be done by the code generator. As output a simple FORTRAN code using BLAS level 3 matrix-matrix multiplication routines whenever possible (otherwise a simple loop structure is use) for index based tensors is obtained. Currently it is not intended to automatically resort tensors in an optimal way for highly efficient matrix-matrix multiplication routines.

Furthermore, a C++ code can be obtained for tensors using the CP format, where a simple contraction kernel - evaluation of scalar products, scaling of vectors, copy operations and a possible rank reduction - is currently generated for each type of contraction separately. For future application this can be changed to a fixed contraction kernel driven by an outer generated routine that takes care of contraction indices and permutations. Currently the permutations are not implicitly implemented in the code generator but can be given explicit through a modified input.

The generator can also produce L^AT_EX output for tensor contractions in various formats: canonical, index based tensors, CP format, MPS format and a variant of the THC format. There the complexity and a possible rank increasing can easily be evaluated.

A.1.3 Workstation

A workstation with exclusive usage of the system and the following specifications:

- CPU: 2× Quad-Core AMD OpteronTM Processor 2354 @ 2200 MHz
- RAM: 32 GB
- Disk: 1 TB scratch disk space
- OS: openSUSE 10.3 (X86-64) with Linux Kernel 2.6.22.17

A.1.4 Compute node

A larger server with exclusive usage of the system and the following specifications:

- CPU: 4× 12-Core AMD OpteronTM Processor 6176 SE @ 2300 MHz
- RAM: 256 GB
- Disk: 4 × 500 GB disks in RAID0 ⇒ 2 TB scratch disk space
- OS: CentOS release 5.5 (Final, X86-64) with Linux Kernel 2.6.18-194

A.1.5 Molecular Geometries

The following geometries were used for the molecules (distance R [pm], angle θ [degree]):

LiH: $R_{LiH} = 159.500$ [245]

HF: $R_{HF} = 91.680$ [245]

H_2O : $R_{OH} = 95.720$, $\theta_{HOH} = 104.52$ [246]

NH_3 : $R_{NH} = 101.100$, $\theta_{HNH} = 106.70$ [247]

CH_4 : $R_{CH} = 108.580$, $\theta_{HCH} = 109.47$ [248]

CO: $R_{CO} = 112.832$ [245]

N_2 : $R_{NN} = 109.768$ [245]

HCN: $R_{HC} = 106.449$, $R_{CN} = 115.378$, $\theta_{HCN} = 180.00^1$

The LiH chain is build up as a linear chain LiH–LiH–… using a distance of 300 pm between the H-atom of one molecule and the Li-atom of the next.

xyz data for furoxan (optimized at RI-MP2//def2-SVP level of theory)

O	-0.072316	0.000000	0.065378
N	0.003653	0.000000	1.375968
N	1.401873	0.000000	-0.502727
C	1.283154	0.000000	1.736924
C	2.137772	0.000000	0.626272
H	3.223284	0.000000	0.538875
H	1.550387	0.000000	2.794106
O	1.571934	0.000000	-1.682250

A.1.6 Used Basis Sets

Table A.1: Used basis sets.

basis set	citation
STO-3G	[249]
6-31G	[250]
6-31G*	[251]
def2-SVP	[216, 252, 253]
cc-pVDZ	[254]
aug-cc-pVDZ	[254, 255]
cc-pVTZ	[254]

¹Geometry optimized at conventional CCSD(T) level of theory using cc-pVQZ basis set.

A.2 Derivation of Tensor Contractions in CCD using Tenors in CP Format

A.2.1 Second Term: $\sum_{ef} t_{ij}^{ef} \langle ab||ef \rangle$

From a formal point of view it is more convenient to examine the contraction in an index based representation of the representing vectors. This way Eqn. 3.69 can be written as

$$[r_2]_{ij}^{ab} = \frac{1}{2} \sum_e \sum_f \left(\sum_{q=1}^Q t_q^e t_q^f t_q^i t_q^j \right) \cdot \left(\sum_{r_2=1}^{R_2} v_{r_2}^a v_{r_2}^b v_{r_2}^e v_{r_2}^f \right). \quad (\text{A.1})$$

Here, the summation over the two internal indices e and f can be done as the different matrices are factorized with respect to these dimensions.

$$[r_2]_{ij}^{ab} = \frac{1}{2} \sum_{q=1}^Q \sum_{r_2=1}^{R_2} \underbrace{\left(\sum_e t_q^e v_{r_2}^e \right)}_{:=\mathbf{E}_{qr_2}} \underbrace{\left(\sum_f t_q^f v_{r_2}^f \right)}_{:=\mathbf{F}_{qr_2}} v_{r_2}^a v_{r_2}^b t_q^i t_q^j. \quad (\text{A.2})$$

The contraction over the internal indices can be performed as matrix-matrix multiplication and the two resulting matrices can be multiplied into one matrix element-wise:

$$[r_2]_{ij}^{ab} = \frac{1}{2} \sum_{q=1}^Q \sum_{r_2=1}^{R_2} \underbrace{\mathbf{E}_{qr_2} \mathbf{F}_{qr_2}}_{:=\mathbf{X}_{qr_2}} v_{r_2}^a v_{r_2}^b t_q^i t_q^j = \frac{1}{2} \sum_{q=1}^Q \sum_{r_2=1}^{R_2} \mathbf{X}_{qr_2} v_{r_2}^a v_{r_2}^b t_q^i t_q^j. \quad (\text{A.3})$$

This way one ends up with a representation of the residual increment in the THC format. In order to add this to the existing residual tensor in CP representation the format has to be converted. Therefore, the two summations are consolidated into one summation over l_2 that is obtained as a product of q and r_2 . The corresponding matrix elements of \mathbf{X} can be multiplied into the representation system for the first dimension and the remaining vectors are simply copied. The prefactor can also be multiplied into the representing system so that finally one ends up with a representation in the CP format but with an increased rank of $L_2 = Q \cdot R_2$

$$[r_2]_{ij}^{ab} = \frac{1}{2} \sum_{q=1}^Q \sum_{r_2=1}^{R_2} \underbrace{\mathbf{X}_{qr_2} v_{r_2}^a}_{:=\tau_{l_2}^a} \underbrace{v_{r_2}^b}_{:=\tau_{l_2}^b} \underbrace{t_q^i}_{:=\tau_{l_2}^i} \underbrace{t_q^j}_{:=\tau_{l_2}^j} = \sum_{l_2=1}^{L_2=Q \cdot R_2} \tau_{l_2}^a \tau_{l_2}^b \tau_{l_2}^i \tau_{l_2}^j. \quad (\text{A.4})$$

A.2.2 Third Term: $\sum_{mn} t_{mn}^{ab} \langle mn||ij \rangle$

For comparison the derivation of this contraction is also written in the index based format of the representing vectors where the common dimensions are contracted yielding coefficient matrices

$$[\mathbf{r}_3]_{ij}^{ab} = \sum_{mn} \left(\sum_{q=1}^Q t_q^a t_q^b t_q^m t_q^n \right) \left(\sum_{r_3=1}^{R_3} v_{r_3}^m v_{r_3}^n v_{r_3}^i v_{r_3}^j \right) \quad (\text{A.5})$$

$$= \sum_{q=1}^Q \sum_{r_3=1}^{R_3} \underbrace{\left(\sum_m t_q^m v_{r_3}^m \right)}_{\mathbf{M}_{qr_3}} \underbrace{\left(\sum_n t_q^n v_{r_3}^n \right)}_{\mathbf{N}_{qr_3}} t_q^a t_q^b v_{r_3}^i v_{r_3}^j. \quad (\text{A.6})$$

These can be merged together as a Hadamard or point wise product leading to a single coefficient matrix \mathbf{X}_{qr_3} . This way, a representation of the residual tensor in the THC format is obtained and it can be transformed into the CP format by merging the two summations into a single one

$$[\mathbf{r}_3]_{ij}^{ab} = \sum_{q=1}^Q \sum_{r_3=1}^{R_3} \underbrace{\mathbf{X}_{qr_3} t_q^a}_{:=\tau_{l_3}^a} \underbrace{t_q^b}_{:=\tau_{l_3}^b} \underbrace{v_{r_3}^i}_{:=\tau_{l_3}^i} \underbrace{v_{r_3}^j}_{:=\tau_{l_3}^j} \quad (\text{A.7})$$

$$= \sum_{l_3=1}^{L_3=Q \cdot R_3} \tau_{l_3}^a \tau_{l_3}^b \tau_{l_3}^i \tau_{l_3}^j. \quad (\text{A.8})$$

A.2.3 Sixth Term: $P(ab) \sum_e f_e^b t_{ij}^{ae}$

The sixth and last term combines again Fock matrix elements with the t_2 amplitudes and can be done in a similar way as presented for the fifth term (see 3.7.4). The needed part of the Fock matrix can be cast into the CP format by trivial or SVD based casting procedure leading to

$$f_b^a = \sum_{u_2=1}^{U_2} f_{u_2}^a f_{u_2}^b \quad \text{or} \quad \mathbf{f}_b^a = \sum_{u_2=1}^{U_2} f_{u_2}^{(a)} \otimes f_{u_2}^{(b)}. \quad (\text{A.9})$$

The contraction (omitting the index permutation) can be done as matrix-matrix multiplication of the representing vectors for the contracted dimensions leading to a coefficient matrix \mathbf{E} . This can be used to transform the second remaining representing vector to the expansion

length Q .

$$[\mathfrak{x}_6]_{ij}^{ab} = \mathfrak{f}_e^b \mathfrak{t}_{ij}^{ae} \quad (\text{A.10})$$

$$= \left(\sum_{u_2=1}^{U_2} f_{u_2}^{(b)} \otimes f_{u_2}^{(e)} \right) \left(\sum_{q=1}^Q t_q^{(a)} \otimes t_q^{(e)} \otimes t_q^{(i)} \otimes t_q^{(j)} \right) \quad (\text{A.11})$$

$$= \sum_{u_2=1}^{U_2} \sum_{q=1}^Q \underbrace{\langle f_{u_2}^{(e)}, t_q^{(e)} \rangle}_{:=\mathbf{E}_{u_2 q}} t_q^{(a)} \otimes f_{u_2}^{(b)} \otimes t_q^{(i)} \otimes t_q^{(j)} \quad (\text{A.12})$$

$$= \sum_{q=1}^Q \underbrace{t_q^{(a)}}_{:=x_{w_6}^{(a)}} \otimes \underbrace{\left(\sum_{u_2=1}^{U_2} \mathbf{E}_{u_2 q} f_{u_2}^{(b)} \right)}_{:=x_{w_6}^{(b)}} \otimes \underbrace{t_q^{(i)}}_{:=x_{w_6}^{(i)}} \otimes \underbrace{t_q^{(j)}}_{:=x_{w_6}^{(j)}} \quad (\text{A.13})$$

$$= \sum_{w_6=1}^{W_6=Q} x_{w_6}^{(a)} \otimes x_{w_6}^{(b)} \otimes x_{w_6}^{(i)} \otimes x_{w_6}^{(j)} \quad (\text{A.14})$$

The rank of the intermediate tensor is again the same as the rank of the amplitudes and therefore no rank reduction is needed after this contraction. Only applying the permutation of indices to build the residual increment doubles the rank

$$[\mathfrak{x}_6]_{ij}^{ab} = P(ab)[\mathfrak{x}]_{ij}^{ab} = [\mathfrak{x}_6]_{ij}^{ab} - [\mathfrak{x}_6]_{ij}^{ba} \quad (\text{A.15})$$

$$= \sum_{w_6=1}^{W_6=Q} x_{w_6}^{(a)} \otimes x_{w_6}^{(b)} \otimes x_{w_6}^{(i)} \otimes x_{w_6}^{(j)} - \sum_{w_6=1}^{W_6=Q} x_{w_6}^{(b)} \otimes x_{w_6}^{(a)} \otimes x_{w_6}^{(i)} \otimes x_{w_6}^{(j)} \quad (\text{A.16})$$

$$= \sum_{l_6=1}^{L_6=2 \cdot W_6=2 \cdot Q} \tau_{l_6}^{(a)} \otimes \tau_{l_6}^{(b)} \otimes \tau_{l_6}^{(i)} \otimes \tau_{l_6}^{(j)}, \quad (\text{A.17})$$

but the increased rank can again be reduced by application of the reduction algorithm.

A.2.4 Increase of Rank during Tensor Contractions

To prevent the large increase of ranks upon tensor contractions one can use reassembly step to build up an intermediate tensor that is again decomposed into the CP format but with a lower rank than the original tensors. To illustrate this procedure the second term of the LCCD t_2 amplitude equations (c.f. 3.7.2)

$$r_{ij}^{ab} \leftarrow \frac{1}{2} \sum_{ef} t_{ij}^{ef} \langle ab || ef \rangle, \quad (\text{A.18})$$

is chosen as an example. In CP format this contraction can be done by the scalar products over contracted dimensions e and f and copy operations of the remaining representing vectors. There, the rank of the target tensor increases to $L_2 = Q \cdot R_2$ in order to maintain a

representation in CP format

$$[\mathbf{r}_2]_{ij}^{ab} = \frac{1}{2} \sum_{q=1}^Q \sum_{r_2=1}^{R_2} \underbrace{\left\langle t_q^{(e)}, v_{r_2}^{(e)} \right\rangle \left\langle t_q^{(f)}, v_{r_2}^{(f)} \right\rangle v_{r_2}^{(a)}}_{:=\tau_{l_2}^{(a)}} \otimes \underbrace{v_{r_2}^{(b)}}_{:=\tau_{l_2}^{(b)}} \otimes \underbrace{t_q^{(i)}}_{:=\tau_{l_2}^{(i)}} \otimes \underbrace{t_q^{(j)}}_{:=\tau_{l_2}^{(j)}}, \quad (\text{A.19})$$

To prevent this increase in rank one can introduce an intermediate reassembly step by performing the summation over r_2 for each value of q and thereby generating a matrix $\mathbf{X}_q^{(a,b)}$

$$[\mathbf{r}_2]_{ij}^{ab} = \frac{1}{2} \sum_{q=1}^Q \sum_{r_2=1}^{R_2} \underbrace{\left\langle t_q^{(e)}, v_{r_2}^{(e)} \right\rangle \left\langle t_q^{(f)}, v_{r_2}^{(f)} \right\rangle v_{r_2}^{(a)} \otimes v_{r_2}^{(b)} \otimes t_q^{(i)} \otimes t_q^{(j)}}_{:=\mathbf{X}_q^{(a,b)}} \quad (\text{A.20})$$

$$= \frac{1}{2} \sum_{q=1}^Q \mathbf{X}_q^{(a,b)} \otimes t_q^{(i)} \otimes t_q^{(j)}. \quad (\text{A.21})$$

This matrix can than be decomposed by a SVD

$$\mathbf{X}_q^{(a,b)} \stackrel{\text{SVD}}{=} \sum_k^N (U_k^a)_q (\Sigma_k)_q (V_b^k)_q, \quad (\text{A.22})$$

and the eigenvectors (one multiplied by the eigenvalues leading to \tilde{U}) can be used as new representing vectors for the first and second dimension. Therefore, for each value of q at most N new representing vectors are obtained and so the rank only increases by a factor of N leading to $L_2 = Q \cdot N$

$$[\mathbf{r}_2]_{ij}^{ab} = \frac{1}{2} \sum_{q=1}^Q \left(\sum_{k=1}^N (\tilde{U}_k^{(a)})_q \otimes (V_k^{(b)})_q \right) \otimes t_q^{(i)} \otimes t_q^{(j)} \quad (\text{A.23})$$

Furthermore, the eigenvalues of the SVD can be used to truncate the summation by skipping values below the accuracy threshold. In order to maintain the full accuracy in the ℓ^2 -norm, the eigenvalues have to be scaled by the norm of the two remaining representing vectors.

This procedure is especially useful for four dimensional tensors where at most two representing vectors from each contracted tensor can remain (for three remaining vectors the one vector from the other tensor can be transformed into the other rank using the contraction matrix containing the scalar products). For tensors in higher dimensions a slightly different procedure can be used: the reassembly leads to a tensor in higher dimensions that can be decomposed using the SVD based casting into the CP format (c.f. 3.1.2) followed by a rank reduction. For a reassembled tensor in three dimensions this procedure increases the rank by a factor of approximately N^2 depending on the number of skipped eigenvalues and the quality of the low rank approximation.

A.3 Derivation of CI Method based on CP Tensors

Instead of deriving the CI equations as presented in Sec. 2.1.3 leading to a diagonalization of the Hamiltonian matrix in order to obtain the coefficient tensors and energy eigenvalues one can also describe CI in analogy to CC (c.f. Sec 2.1.4). Stating point is the definition of the CI wavefunction as

$$|\Phi_{CI}\rangle = \hat{C} |\Phi_0\rangle, \quad (\text{A.24})$$

using a reference ground state wavefunction Φ_0 as obtained from the Hartree-Fock method. The CI operator \hat{C} is a linear operator

$$\hat{C} = 1 + \hat{C}_1 + \hat{C}_2 + \hat{C}_3 + \dots, \quad (\text{A.25})$$

where each term generates substitutions from the occupied to the virtual orbital space:

$$\hat{C}_1 = \sum_{ia} c_i^a \hat{a}_a^\dagger \hat{a}_i \quad (\text{A.26})$$

$$\hat{C}_2 = \sum_{ijab} c_{ij}^{ab} \hat{a}_a^\dagger \hat{a}_i^\dagger \hat{a}_b \hat{a}_j \quad (\text{A.27})$$

$$\hat{C}_3 = \sum_{ijkabc} c_{ijk}^{abc} \hat{a}_a^\dagger \hat{a}_i^\dagger \hat{a}_b^\dagger \hat{a}_j^\dagger \hat{a}_c^\dagger \hat{a}_k \quad (\text{A.28})$$

...

Using the CI wavefunction in the Schrödinger Equation

$$\hat{H} |\Phi_{CI}\rangle = E |\Phi_{CI}\rangle \quad (\text{A.29})$$

$$\hat{H} \hat{C} |\Phi_0\rangle = E \hat{C} |\Phi_0\rangle, \quad (\text{A.30})$$

and projection technique (left projection onto the reference state)

$$\langle \Phi_0 | \hat{H} \hat{C} | \Phi_0 \rangle = E \langle \Phi_0 | \hat{C} | \Phi_0 \rangle \quad (\text{A.31})$$

$$\langle \Phi_0 | \hat{H} \hat{C} | \Phi_0 \rangle = E \underbrace{\langle \Phi_0 | \Phi_{CI} \rangle}_{=1} \quad (\text{A.32})$$

$$\langle \Phi_0 | \hat{H} \hat{C} | \Phi_0 \rangle = E, \quad (\text{A.33})$$

the CI energy equation is obtained. The same technique can be used to derive equations for the coefficients by projecting onto substituted determinants:

$$\langle \Phi_i^a | \hat{H} \hat{C} | \Phi_0 \rangle = E \quad (\text{A.34})$$

$$\langle \Phi_{ij}^{ab} | \hat{H} \hat{C} | \Phi_0 \rangle = E \quad (\text{A.35})$$

$$\langle \Phi_{ijk}^{abc} | \hat{H} \hat{C} | \Phi_0 \rangle = E \quad (\text{A.36})$$

...

Contrary to the CC method the coefficient equations are not decoupled from the energy. Using the second quantization formulation of the Hamiltonian

$$\hat{H} = \sum_{pq} \langle p|\hat{h}|q\rangle \hat{a}_p^\dagger \hat{a}_q + \sum_{pqrs} \langle pq||rs\rangle \hat{a}_p^\dagger \hat{a}_r \hat{a}_q^\dagger \hat{a}_s \quad (\text{A.37})$$

one can now evaluate all possible contractions between the Hamiltonian and the linear CI operator using commutator algebra or the more convenient diagrammatic technique. The detailed derivation is skipped here. The energy equation is similar to the equation in CC theory but only includes linear combinations of the coefficients and the Hamiltonian elements

$$E = \frac{1}{4} \sum_{efmn} \langle ef||mn\rangle c_{mn}^{ef} + \sum_{em} f_e^m c_m^e. \quad (\text{A.38})$$

The equations for the coefficients are

$$\begin{aligned} E \cdot c_i^a = & f_{ia} - \sum_{em} \langle am||ie\rangle c_m^e + \sum_e f_{ae} c_i^e - \sum_m f_{mi} c_m^a - \frac{1}{2} \sum_{mne} \langle mn||ei\rangle c_{mn}^{ea} + \\ & \frac{1}{2} \sum_{efm} \langle ma||ef\rangle c_{mi}^{ef} + \sum_{em} f_{mb} c_{im}^{ae} + \frac{1}{4} \sum_{efmn} \langle mn||ef\rangle c_{efa}^{mni} \end{aligned} \quad (\text{A.39})$$

$$\begin{aligned} E \cdot c_{ij}^{ab} = & P(a|b)P(i|j)f_{ia}c_j^b + \langle ij||ab\rangle + P(i|j) \sum_e \langle ab||ie\rangle c_j^e - P(a|b) \sum_m \langle am||ij\rangle c_m^b + \\ & P(i|j)P(a|b) \sum_{em} \langle am||ie\rangle c_{mj}^{eb} + \frac{1}{2} \sum_{ef} \langle ab||ef\rangle c_{ij}^{ef} + \frac{1}{2} \sum_{mn} \langle mn||ij\rangle c_{mn}^{ab} + \\ & P(a|b) \sum_e f_{ae} c_{ij}^{eb} - P(i|j) \sum_m f_{mi} c_{mj}^{ab} - \frac{1}{2} P(i|j) \sum_{mne} \langle mn||ei\rangle c_{mnj}^{eab} + \\ & \frac{1}{2} P(a|b) \sum_{efm} \langle ma||ef\rangle c_{mij}^{efb} + \sum_{em} f_{em} c_{mij}^{eab} + \frac{1}{4} \sum_{efmn} \langle mn||ef\rangle c_{mnij}^{efab} \end{aligned} \quad (\text{A.40})$$

$$\begin{aligned} E \cdot c_{ijk}^{abc} = & P(ab|c)P(ij|k)f_{ia}c_{jk}^{bc} + P(ab|c)P(ij|k)\langle ij||ab\rangle c_k^c + P(ij|k)P(a|bc) \sum_e \langle ab||ie\rangle c_{jk}^{ec} - \\ & P(ab|c)P(i|jk) \sum_m \langle am||ij\rangle c_{mk}^{bc} + P(ij|k)P(ab|c) \sum_{em} \langle am||ie\rangle c_{mjk}^{ebc} + \\ & \frac{1}{2} P(ab|c) \sum_{ef} \langle ab||ef\rangle c_{ijk}^{efc} + \frac{1}{2} P(ij|k) \sum_{mn} \langle mn||ij\rangle c_{mnk}^{abc} + P(ab|c) \sum_e f_{ae} c_{ijk}^{ebc} - \\ & P(ij|k) \sum_m f_{mi} c_{mjk}^{abc} - \frac{1}{2} P(ij|k) \sum_{mne} \langle mn||ei\rangle c_{mnjk}^{eabc} + \frac{1}{2} P(ab|c) \sum_{efm} \langle ma||ef\rangle c_{mijk}^{efbc} + \\ & \sum_{em} f_{em} c_{mijk}^{eabc} + \frac{1}{4} \sum_{efmn} \langle mn||ef\rangle c_{mnijk}^{efabc} \end{aligned} \quad (\text{A.41})$$

and one can see that there are only 13 different contributions to each coefficient equation. The different terms are similar but use different kinds of coefficients and more and more permutations for higher order tensors. This becomes even more clear from looking at the diagrammatic representation given in Fig. A.1. So in general only 13 terms using coefficients

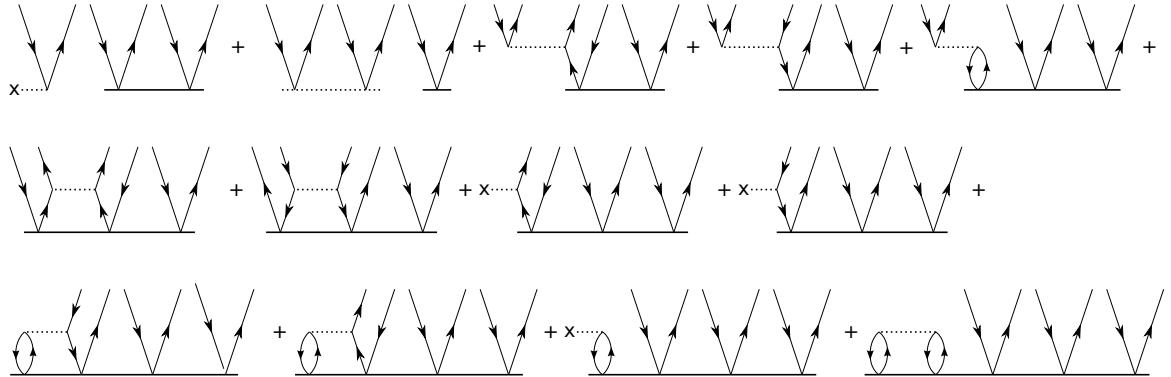


Figure A.1: Diagrammatic representation of c_3 CI coefficient equations.

that are up to two orders higher or lower and Hamiltonian elements (Fock matrix and two-electron integrals) are needed to construct the coefficient equations. These contractions can also be done using tensors in CP format following the procedures outlined in Sec. 3.7. This way the contractions can be performed by evaluating scalar products of the representing vectors and copying the remaining representing vectors. However, some contractions lead to an increase of the rank in the representation of the target tensor and the reduction procedure is needed to maintain a low rank representation. Nevertheless, the simple structure of the equations that consist more or less of the same tensor contractions allow for an efficient implementation using only a contraction and permutation kernel. Such scheme is currently under investigation.

A.4 Parameters for Reduction Algorithm

A.4.1 Determination of Step Size for Improved Reduction Algorithm

Table A.2: Tests for optimal step size (collecting Z ranks before improvement and gradient step) using water in 6-31G* basis set (18 basis functions). An accuracy parameter of 10^{-4} is applied. The CPU time is measured on a single core of a workstation.

algorithm	final rank	CPU time [s]
original C++ implementation	732	12547
Z=100	1200	627
Z=50	1100	712
new FORTRAN code	Z=20	940
	Z=10	853
	Z=5	910
	Z=1	2031
	Z=1	7652

A.4.2 Parameters used in Original Implementation

The most important internal parameters for the original implementation are given below:

- maximal steps for cross approximation: 10
- two loop parameter for improvement of rank: outer loop = 10, inner loop = 2
- accuracy and number of maximal iterations during gradient evaluation:
accuracy = 10^{-2} ; number of iterations = 50

Especially the two loop parameters for improving the ranks have been optimized through numerical testing (see Tab. A.3). Although for this example a lower amount of inner and outer loops seems sufficient for a good rank reduction, still some larger values are chosen as standard throughout this work. Especially for larger examples and for collecting rank one approximations up to a given number before performing the improvement algorithm, the larger number of outer loops is necessary to get a good reduction of ranks.

Furthermore, one can weight computational effort for the improvement part and the calculation of the gradient: If a lower number of loops is performed in the improvement of representing vectors (which corresponds to lower CPU time) the computational effort the gradient part increases in order to reach the desired accuracy. On the other hand, a large number of improvement loops decreases the time for the gradient calculation, as the accuracy is achieved within less iterations. Therefore, the presented parameters are a compromise between these two parts of the reduction algorithm and the effort for both parts is more or less balanced. This also helps to accelerate the general convergence of the algorithm.

Table A.3: Numerical tests for loop parameters of rank improving part of reduction algorithm. The CPU time is measured on a single core of a workstation.

outer loops	inner loops	final rank	CPU time [s]
10	4	71	206
2	4	70	80
20	2	70	187
15	2	70	149
10	2	70	114
8	2	70	97
6	2	70	81
4	2	70	75
2	2	71	53
1	2	72	53
10	1	70	84
5	1	71	63

A.4.3 Parameters used in New Implementation

As the new implementation of the rank reduction algorithm is still based on the original one, also the key parameters are included. The standard values using the rank one approximation are:

- maximal steps for cross approximation: 5
- two loop parameter for improvement of rank: outer loop = 5, inner loop = 2
- accuracy and number of maximal iterations during gradient evaluation:
accuracy = 10^{-2} ; number of iterations = 20

For approximations with larger step sizes (collecting more ranks before each improvement step) it turns out that the parameters have to be changed in order to achieve reductions to lower ranks and higher accuracy:

- maximal steps for cross approximation: 5
- two loop parameter for improvement of rank: outer loop = 10, inner loop = 5
- accuracy and number of maximal iterations during gradient evaluation:
accuracy = 10^{-2} ; number of iterations = 10

The change in parameters mainly reduces the CPU time, while the actual ranks are only slightly increased.

Bibliography

- [1] T. Helgaker, P. Jørgenson, J. Olsen, *Molecular Electronic Structure Theory*, 1. Auflage, Wiley and Sons, **2000**.
- [2] D. Young, *Computational Chemistry: A Practical Guide for Applying Techniques to Real World Problems*, John Wiley & Sons, **2004**.
- [3] D. B. Cook, *Handbook of Computational Quantum Chemistry*, Dover Publications, Dover, **2005**.
- [4] F. Jensen, *Introduction to Computational Chemistry*, 2. Auflage, Wiley, Chichester, **2007**.
- [5] W. Koch, M. C. Holthausen, *A Chemist's Guide to Density Functional Theory*, Wiley-VCH, Weinheim, **2001**.
- [6] T. Schwabe, R. Huenerbein, S. Grimme, *Synlett*, **10**, 1431–1441, **2010**.
- [7] C. D. Sherrill, *J. Chem. Phys.*, **132**, 110902–7, **2010**.
- [8] J. Dziedzic, S. J. Fox, T. Fox, C. S. Tautermann, C.-K. Skylaris, *Int. J. Quantum Chem.*, **113**, 771–785, **2013**.
- [9] S. A. Maurer, D. S. Lambrecht, J. Kussmann, C. Ochsenfeld, *J. Chem. Phys.*, **138**, 014101–12, **2013**.
- [10] I. Ufimtsev, T. Martinez, *Computing in Science & Engineering*, **10**, 26–34, **2008**.
- [11] A. E. DePrince, J. R. Hammond, *J. Chem. Theory Comput.*, **7**, 1287–1295, **2011**.
- [12] A. Asadchev, M. S. Gordon, *J. Chem. Theory Comput.*, **8**, 4166–4176, **2012**.
- [13] A. V. Titov, I. S. Ufimtsev, N. Luehr, T. J. Martinez, *J. Chem. Theory Comput.*, **9**, 213–221, **2012**.
- [14] C. Ripplinger, F. Neese, *J. Chem. Phys.*, **138**, 034106–18, **2013**.
- [15] M. Schütz, J. Yang, G. K.-L. Chan, F. R. Manby, H.-J. Werner, *J. Chem. Phys.*, **138**, 054109–10, **2013**.
- [16] O. Vahtras, J. Almlöf, M. Feyereisen, *Chem. Phys. Lett.*, **213**, 514–518, **1993**.
- [17] F. Weigend, M. Häser, H. Patzelt, R. Ahlrichs, *Chem. Phys. Lett.*, **294**, 143–152, **1998**.
- [18] C. Hättig, F. Weigend, *J. Chem. Phys.*, **113**, 5154–5161, **2000**.

- [19] R. Ahlrichs, *Phys. Chem. Chem. Phys.*, **6**, 5119–5121, **2004**.
- [20] F. Weigend, M. Kattannek, R. Ahlrichs, *J. Chem. Phys.*, **130**, 164106–8, **2009**.
- [21] H. J. Werner, M. Schütz, *J. Chem. Phys.*, **135**, 144116–15, **2011**.
- [22] H. J. Werner, F. R. Manby, P. J. Knowles, *J. Chem. Phys.*, **118**, 8149–8160, **2003**.
- [23] M. Schütz, F. R. Manby, *Phys. Chem. Chem. Phys.*, **5**, 3349–3358, **2003**.
- [24] F. R. Manby, *J. Chem. Phys.*, **119**, 4607–4613, **2003**.
- [25] F. P. Billingsley, J. E. Bloor, *J. Chem. Phys.*, **55**, 5178–5190, **1971**.
- [26] E. J. Baerends, D. E. Ellis, P. Ros, *Chem. Phys.*, **2**, 41–51, **1973**.
- [27] J. L. Whitten, *J. Chem. Phys.*, **58**, 4496–4501, **1973**.
- [28] W. Kutzelnigg, *Theor. Chim. Acta.*, **68**, 445–469, **1985**.
- [29] E. F. Valeev, H. F. Schaefer, *J. Chem. Phys.*, **113**, 3990–3995, **2000**.
- [30] J. Rychlewski, *Explicitly correlated wave functions in chemistry and physics : theory and applications*, Kluwer Academic Publishers, Dordrecht, Boston, **2003**.
- [31] S. Ten-No, *Chem. Phys. Lett.*, **398**, 56–61, **2004**.
- [32] W. Klopper, F. R. Manby, S. Ten-No, E. F. Valeev, *Int. Rev. Phys. Chem.*, **25**, 427–468, **2006**.
- [33] H. J. Werner, T. B. Adler, G. Knizia, F. R. Manby, *Recent progress in coupled cluster methods : theory and applications*, chapter Efficient Explicitly Correlated Coupled-Cluster Approximations, Springer, London, 573–620, **2010**.
- [34] H. J. Werner, G. Knizia, F. R. Manby, *Mol. Phys.*, **109**, 407–417, **2011**.
- [35] N. H. F. Beebe, J. Linderberg, *Int. J. Quantum Chem.*, **12**, 683–705, **1977**.
- [36] I. Røeggen, E. Wisløff-Nilssen, *Chem. Phys. Lett.*, **132**, 154–160, **1986**.
- [37] S. Wilson, *Comput. Phys. Commun.*, **58**, 71–81, **1990**.
- [38] H. Koch, A. S. de Merás, T. B. Pedersen, *J. Chem. Phys.*, **118**, 9481–9484, **2003**.
- [39] F. Aquilante, T. B. Pedersen, R. Lindh, *J. Chem. Phys.*, **126**, 194106–11, **2007**.
- [40] J. Zienau, L. Clin, B. Doser, C. Ochsenfeld, *J. Chem. Phys.*, **130**, 204112–4, **2009**.
- [41] J. Almlöf, *Chem. Phys. Lett.*, **181**, 319–320, **1991**.
- [42] M. Häser, J. Almlöf, *J. Chem. Phys.*, **96**, 489–494, **1992**.
- [43] M. Häser, *Theor. Chim. Acta.*, **87**, 147–173, **1993**.

- [44] A. K. Wilson, J. Almlöf, *Theor. Chim. Acta.*, **95**, 49–62, **1997**.
- [45] B. Doser, D. S. Lambrecht, J. Kussmann, C. Ochsenfeld, *J. Chem. Phys.*, **130**, 064107–14, **2009**.
- [46] B. Doser, J. Zienau, L. Clin, D. S. Lambrecht, C. Ochsenfeld, *Z. Phys. Chem.*, **224**, 397–412, **2010**.
- [47] T. D. Crawford, H. F. Schaefer, *Rev. Comp. Chem.*, **14**, 33–136, **2000**.
- [48] T. Helgaker, W. Klopper, D. P. Tew, *Mol. Phys.*, **106**, 2107–2143, **2008**.
- [49] R. J. Bartlett, *J. Phys. Chem.*, **93**, 1697–1708, **1989**.
- [50] H. Larsen, J. Olsen, C. Hättig, P. Jørgensen, O. Christiansen, J. Gauss, *J. Chem. Phys.*, **111**, 1917–1925, **1999**.
- [51] D. Feller, J. A. Sordo, *J. Chem. Phys.*, **112**, 5604–5610, **2000**.
- [52] J. Pittner, P. Hobza, *Chem. Phys. Lett.*, **390**, 496–499, **2004**.
- [53] A. Tajti, P. G. Szalay, A. G. Csaszar, M. Kallay, J. Gauss, E. F. Valeev, B. A. Flowers, J. Vazquez, J. F. Stanton, *J. Chem. Phys.*, **121**, 11599–11613, **2004**.
- [54] M. Heckert, M. Kállay, J. Gauss, *Mol. Phys.*, **103**, 2109–2115, **2005**.
- [55] G. Rauhut, G. Knizia, H. J. Werner, *J. Chem. Phys.*, **130**, 054105–10, **2009**.
- [56] C. Hättig, D. P. Tew, A. Köhn, *J. Chem. Phys.*, **132**, 231102–4, **2010**.
- [57] R. J. Bartlett, *Modern Electronic Structure Theory, Vol. II*, chapter Coupled Cluster Theory: An Overview of Recent Developments, World Scientific, Singapore, 1047–1131, **1995**.
- [58] R. Bellman, *Adaptive control processes: A guided tour.*, Princeton University Press, University of Michigan, **1961**.
- [59] G. Stollhoff, P. Fulde, *J. Chem. Phys.*, **73**, 4548–4561, **1980**.
- [60] P. Pulay, *Chem. Phys. Lett.*, **100**, 151–154, **1983**.
- [61] P. Pulay, S. Sæbo, *Theor. Chim. Acta.*, **69**, 357–368, **1986**.
- [62] S. Saebø, P. Pulay, *J. Chem. Phys.*, **86**, 914–922, **1987**.
- [63] S. Saebø, P. Pulay, *Annu. Rev. Phys. Chem.*, **44**, 213–236, **1993**.
- [64] H. Stoll, *Phys. Rev. B*, **46**, 6700–6704, **1992**.
- [65] C. Hampel, H. J. Werner, *J. Chem. Phys.*, **104**, 6286–6297, **1996**.
- [66] M. Schütz, G. Hetzer, H. J. Werner, *J. Chem. Phys.*, **111**, 5691–5705, **1999**.

- [67] M. Schütz, H. J. Werner, *Chem. Phys. Lett.*, **318**, 370–378, **2000**.
- [68] M. Schütz, *J. Chem. Phys.*, **113**, 9986–10001, **2000**.
- [69] M. Schütz, H. J. Werner, *J. Chem. Phys.*, **114**, 661–681, **2001**.
- [70] S. Li, J. Ma, Y. Jiang, *J. Comput. Chem.*, **23**, 237–244, **2002**.
- [71] M. Schütz, *Phys. Chem. Chem. Phys.*, **4**, 3941–3947, **2002**.
- [72] D. Kats, D. Usyat, M. Schütz, *Phys. Chem. Chem. Phys.*, **10**, 3430–3439, **2008**.
- [73] N. Flocke, R. J. Bartlett, *J. Chem. Phys.*, **121**, 10935–10944, **2004**.
- [74] B. Paulus, *Physics Reports*, **428**, 1–52, **2006**.
- [75] J. Friedrich, M. Hanrath, M. Dolg, *J. Chem. Phys.*, **126**, 154110–7, **2007**.
- [76] M. Kobayashi, H. Nakai, *J. Chem. Phys.*, **129**, 044103–9, **2008**.
- [77] J. Friedrich, M. Dolg, *J. Chem. Theory Comput.*, **5**, 287–294, **2009**.
- [78] J. Friedrich, K. Walczak, *J. Chem. Theory Comput.*, **9**, 408–417, **2012**.
- [79] P. O. Löwdin, *Phys. Rev.*, **97**, 1474–1489, **1955**.
- [80] P. O. Löwdin, *Phys. Rev.*, **97**, 1490–1508, **1955**.
- [81] P. O. Löwdin, *Phys. Rev.*, **97**, 1509–1520, **1955**.
- [82] W. Kutzelnigg, *Theor. Chim. Acta.*, **1**, 327–342, **1963**.
- [83] R. Ahlrichs, W. Kutzelnigg, W. A. Bingel, *Theor. Chim. Acta.*, **5**, 289–304, **1966**.
- [84] R. K. Nesbet, *Phys. Rev.*, **175**, 2–9, **1968**.
- [85] W. Meyer, *Int. J. Quantum Chem.*, **5**, 341–348, **1971**.
- [86] W. Meyer, *J. Chem. Phys.*, **58**, 1017–1035, **1973**.
- [87] F. Neese, A. Hansen, D. G. Liakos, *J. Chem. Phys.*, **131**, 064103–15, **2009**.
- [88] F. Neese, F. Wennmohs, A. Hansen, *J. Chem. Phys.*, **130**, 114108–18, **2009**.
- [89] F. Neese, A. Hansen, F. Wennmohs, S. Grimme, *Acc. Chem. Res.*, **42**, 641–648, **2009**.
- [90] A. Anoop, W. Thiel, F. Neese, *J. Chem. Theory Comput.*, **6**, 3137–3144, **2010**.
- [91] J. Yang, Y. Kurashige, F. R. Manby, G. K. L. Chan, *J. Chem. Phys.*, **134**, 044123–13, **2011**.
- [92] J. Yang, G. K.-L. Chan, F. R. Manby, M. Schütz, H.-J. Werner, *J. Chem. Phys.*, **136**, 144105–16, **2012**.

- [93] Y. Kurashige, J. Yang, G. K. L. Chan, F. R. Manby, *J. Chem. Phys.*, **136**, 124106–7, **2012**.
- [94] L. Adamowicz, R. J. Bartlett, *J. Chem. Phys.*, **86**, 6314–6324, **1987**.
- [95] P. Neogrády, M. Pitoňák, M. Urban, *Mol. Phys.*, **103**, 2141–2157, **2005**.
- [96] M. Pitoňák, P. Neogrády, V. Kellö, M. Urban, *Mol. Phys.*, **104**, 2277–2292, **2006**.
- [97] M. Pitoňák, F. Holka, P. Neogrády, M. Urban, *J. Mol. Struct. Theochem*, **768**, 79–89, **2006**.
- [98] L. Adamowicz, *Mol. Phys.*, **108**, 3105 – 3112, **2010**.
- [99] C. Eckart, G. Young, *Psychometrika*, **1**, 211–218, **1936**.
- [100] J. D. Carroll, J. J. Chang, *Psychometrika*, **35**, 283–319, **1970**.
- [101] G. Golub, *Matrix computations*, Johns Hopkins University Press, Baltimore, **1996**.
- [102] T. Kinoshita, O. Hino, R. J. Bartlett, *J. Chem. Phys.*, **119**, 7756–7762, **2003**.
- [103] O. Hino, T. Kinoshita, R. J. Bartlett, *J. Chem. Phys.*, **121**, 1206–1213, **2004**.
- [104] F. Bell, D. S. Lambrecht, M. Head-Gordon, *Mol. Phys.*, **108**, 2759 – 2773, **2010**.
- [105] E. G. Hohenstein, R. M. Parrish, T. J. Martinez, *J. Chem. Phys.*, **137**, 044103–10, **2012**.
- [106] R. A. Harshman, M. E. Lundy, *Computational Statistics and Data Analysis*, **18**, 39–72, **1994**.
- [107] F. Hitchcock, *J. Math. Phys.*, **6**, 164–189, **1927**.
- [108] T. Zhang, G. H. Golub, *SIAM J. Matrix Anal. Appl.*, **23**, 534–550, **2001**.
- [109] M. Ishteva, L. De Lathauwer, P. A. Absil, S. Van Huffel, *Numer. Algor.*, **51**, 179–194, **2009**.
- [110] T. G. Kolda, B. W. Bader, *SIAM Review*, **51**, 455–500, **2009**.
- [111] W. Hackbusch, *Tensor Spaces and Numerical Tensor Calculus*, Springer Series in Computational Mathematics, Springer, Berlin Heidelberg, **2012**.
- [112] W. Hackbusch, B. N. Khoromskij, *Computing*, **76**, 177–202, **2006**.
- [113] W. Hackbusch, B. Khoromskij, *J. Complex.*, **23**, 697–714, **2007**.
- [114] M. Espig, L. Grasedyck, W. Hackbusch, *Constr. Approx.*, **30**, 557–597, **2009**.
- [115] W. Hackbusch, S. Kühn, *J. Fourier Anal. Appl.*, **15**, 706–722, **2009**.
- [116] R. Bro, *Chemom. Intell. Lab. Syst.*, **38**, 149–171, **1997**.

- [117] G. Beylkin, M. J. Mohlenkamp, *P. Natl. Acad. Sci. USA*, **99**, 10246–10251, **2002**.
- [118] R. Bro, H. A. L. Kiers, *J. Chemom.*, **17**, 274–286, **2003**.
- [119] G. Beylkin, M. J. Mohlenkamp, *SIAM J. Sci. Comput.*, **26**, 2133–2159, **2005**.
- [120] S. R. Chinnamsetty, M. Espig, B. N. Khoromskij, W. Hackbusch, H. J. Flad, *J. Chem. Phys.*, **127**, 084110, **2007**.
- [121] M. Espig, Effiziente Bestapproximation mittels Summen von Elementartensoren in hohen Dimensionen, Ph.D. thesis, Universität Leipzig, **2008**.
- [122] M. Espig, W. Hackbusch, *Numer. Math.*, **122**, 489–525, **2012**.
- [123] C. F. Van Loan, *J. Comput. Appl. Math.*, **123**, 85–100, **2000**.
- [124] C. J. Appelof, E. R. Davidson, *Anal. Chem.*, **53**, 2053–2056, **1981**.
- [125] R. Henrion, *J. Chemom.*, **7**, 477–494, **1993**.
- [126] A. K. Smilde, Y. D. Wang, B. R. Kowalski, *J. Chemom.*, **8**, 21–36, **1994**.
- [127] R. Bro, *Crit. Rev. Anal. Chem.*, **36**, 279–293, **2006**.
- [128] L. De Lathauwer, B. De Moor, J. Vandewalle, *SIAM J. Matrix Anal. Appl.*, **21**, 1253–1278, **2000**.
- [129] A. N. Langville, W. J. Stewart, *Numer. Linear Algebra Appl.*, **11**, 723–752, **2004**.
- [130] L. De Lathauwer, J. Vandewalle, *Linear Algebra Appl.*, **391**, 31–55, **2004**.
- [131] L. De Lathauwer, J. Castaing, *Signal Processing*, **87**, 322–336, **2007**.
- [132] M. Vasilescu, D. Terzopoulos, in *Pattern Recognition, 2002. Proceedings. 16th International Conference on*, volume 2, 511–514, **2002**.
- [133] D. Vlasic, M. Brand, H. Pfister, J. Popovic, *ACM Transactions On Graphics*, **24**, 426–433, **2005**.
- [134] E. Acar, S. A. Camtepe, M. S. Krishnamoorthy, B. Yener, *Intelligence and Security Informatics, Proceedings*, **3495**, 256–268, **2005**.
- [135] L. Omberg, G. H. Golub, O. Alter, *P. Natl. Acad. Sci. USA*, **104**, 18371–18376, **2007**.
- [136] E. Martínez-Montes, P. A. Valdés-Sosa, F. Miwakeichi, R. I. Goldman, M. S. Cohen, *Neuroimage*, **22**, 1023–1034, **2004**.
- [137] F. Miwakeichi, E. Martínez-Montes, P. A. Valdés-Sosa, N. Nishiyama, H. Mizuhara, Y. Yamaguchia, *Neuroimage*, **22**, 1035–1045, **2004**.
- [138] B. N. Khoromskij, V. Khoromskaia, *Cent. Eur. J. Math.*, **5**, 523–550, **2007**.

- [139] B. N. Khoromskij, V. Khoromskaia, S. R. Chinnamsetty, H. J. Flad, *J. Comput. Phys.*, **228**, 5749–5762, **2009**.
- [140] B. N. Khoromskij, V. Khoromskaia, H.-J. Flad, *MPI MIS Preprint*, **44**, **2009**.
- [141] V. de Silva, L.-H. Lim, *SIAM J. Matrix Anal. Appl.*, **30**, 1084–1127, **2008**.
- [142] L. De Lathauwer, B. De Moor, J. Vandewalle, *SIAM J. Matrix Anal. Appl.*, **21**, 1324–1342, **2000**.
- [143] L. De Lathauwer, *SIAM J. Matrix Anal. Appl.*, **28**, 642–666, **2006**.
- [144] C. F. Van Loan, N. Pitsianis, *NATO Adv. Sci. Ser. E Appl. Sci. vol. 232*, chapter Approximation with Kronecker products, Kluwer, Dordrecht, 293–314, **1993**.
- [145] M. Espig, W. Hackbusch, T. Rohwedder, R. Schneider, *Numer. Math.*, **122**, 469–488, **2012**.
- [146] U. Benedikt, A. A. Auer, M. Espig, W. Hackbusch, *J. Chem. Phys.*, **134**, 054118–12, **2011**.
- [147] L. Grasedyck, *MPI MIS Preprint (submitted to SIMAX)*, **27**, 1, **2009**.
- [148] I. V. Oseledets, E. E. Tyrtyshnikov, *INM RAS Preprint*, **2**, **2009**.
- [149] E. Schrödinger, *Naturwissenschaften*, **14**, 664–666, **1926**.
- [150] A. Szabo, N. S. Ostlund, *Modern Quantum Chemistry*, Dover Publications, Mineola, New York, **1996**.
- [151] M. Born, R. Oppenheimer, *Ann. Physik*, **389**, 457–484, **1927**.
- [152] I. N. Levine, *Quantum Chemistry*, 5. Auflage, Prentice Hall, New Jersey, **2000**.
- [153] B. O. Roos, *Lecture Notes in Quantum Chemistry II*, Springer Verlag, Berlin, Heidelberg, New York, **1994**.
- [154] C. Møller, M. S. Plesset, *Phys. Rev.*, **46**, 618–622, **1934**.
- [155] T. B. Adler, H. J. Werner, F. R. Manby, *J. Chem. Phys.*, **130**, 054106–13, **2009**.
- [156] D. Cremer, *WIREs Comput Mol Sci*, **1**, 509–530, **2011**.
- [157] D. R. Bowler, T. Miyazaki, *Rep. Prog. Phys.*, **75**, 036503–43, **2012**.
- [158] S. A. Maurer, D. S. Lambrecht, D. Flaig, C. Ochsenfeld, *J. Chem. Phys.*, **136**, 144107–15, **2012**.
- [159] K. L. Bak, J. Gauss, P. Jørgensen, J. Olsen, T. Helgaker, J. F. Stanton, *J. Chem. Phys.*, **114**, 6548–6556, **2001**.
- [160] K. L. Bak, P. Jørgensen, J. Olsen, T. Helgaker, W. Klopper, *J. Chem. Phys.*, **112**, 9229–9242, **2000**.

- [161] J. Řezáč, K. E. Riley, P. Hobza, *J. Chem. Theory Comput.*, **7**, 2427–2438, **2011**.
- [162] A. A. Auer, J. Gauss, J. F. Stanton, *J. Chem. Phys.*, **118**, 10407–10417, **2003**.
- [163] E. Prochnow, A. A. Auer, K. Banert, *J. Phys. Chem. A*, **111**, 9945–9951, **2007**.
- [164] A. A. Auer, *J. Chem. Phys.*, **131**, 024116–7, **2009**.
- [165] E. Prochnow, A. A. Auer, *J. Chem. Phys.*, **132**, 064109–7, **2010**.
- [166] A. P. Scott, L. Radom, *J. Phys. Chem.*, **100**, 16502–16513, **1996**.
- [167] J. Olsen, O. Christiansen, H. Koch, P. Jørgensen, *J. Chem. Phys.*, **105**, 5082–5090, **1996**.
- [168] J. Olsen, P. Jørgensen, T. Helgaker, O. Christiansen, *J. Chem. Phys.*, **112**, 9736–9748, **2000**.
- [169] F. H. Stillinger, *J. Chem. Phys.*, **112**, 9711–9715, **2000**.
- [170] C. D. Sherrill, H. F. Schaefer, *Adv. Quantum Chem.*, **34**, 143–269, **1999**.
- [171] J. Olsen, F. Jørgensen, H. Koch, A. Balkova, R. J. Bartlett, *J. Chem. Phys.*, **104**, 8007–8015, **1996**.
- [172] M. Head-Gordon, R. Rico, M. Oumi, T. Lee, *Chem. Phys. Lett.*, **219**, 21–29, **1994**.
- [173] S. Hirata, M. Head-Gordon, R. Bartlett, *J. Chem. Phys.*, **111**, 10774–10786, **1999**.
- [174] R. J. Bartlett, G. D. Purvis, *Int. J. Quantum Chem.*, **14**, 561–581, **1978**.
- [175] R. J. Bartlett, *Ann. Rev. Phys. Chem.*, **32**, 359–401, **1981**.
- [176] J. A. Pople, J. S. Binkley, R. Seeger, *Int. J. Quantum Chem.*, **10**, 1–19, **1976**.
- [177] J. Čížek, *J. Chem. Phys.*, **45**, 4256–4266, **1966**.
- [178] J. Čížek, *Adv. Chem. Phys.*, **14**, 35–89, **1969**.
- [179] J. Čížek, J. Paldus, *Int. J. Quantum Chem.*, **5**, 359–379, **1971**.
- [180] R. F. Bishop, H. G. Kümmel, *Physics Today*, **40**, 52–60, **1987**.
- [181] H. G. Kümmel, *Int. J. Mod. Phys. B*, **17**, 5311–5325, **2003**.
- [182] I. Shavitt, R. Bartlett, *Many-body methods in chemistry and physics: MBPT and coupled-cluster theory*, Cambridge University Press, **2009**.
- [183] R. J. Bartlett, M. Musiał, *Rev. Mod. Phys.*, **79**, 291–352, **2007**.
- [184] H. J. Monkhorst, *Int. J. Quantum Chem.*, **S11**, 421–432, **1977**.
- [185] G. D. Purvis, R. J. Bartlett, *J. Chem. Phys.*, **76**, 1910–1918, **1982**.

- [186] J. Noga, R. J. Bartlett, *J. Chem. Phys.*, **86**, 7041–7050, **1987**.
- [187] G. E. Scuseria, H. F. Schaefer, *Chem. Phys. Lett.*, **152**, 382–386, **1988**.
- [188] S. A. Kucharski, R. J. Bartlett, *Theor. Chim. Acta.*, **80**, 387–405, **1991**.
- [189] S. A. Kucharski, R. J. Bartlett, *J. Chem. Phys.*, **97**, 4282–4288, **1992**.
- [190] M. Kállay, P. R. Surján, *J. Chem. Phys.*, **113**, 1359–1365, **2000**.
- [191] M. Kállay, P. G. Szalay, P. R. Surján, *J. Chem. Phys.*, **117**, 980–990, **2002**.
- [192] T. Helgaker, J. Gauss, P. Jørgensen, J. Olsen, *J. Chem. Phys.*, **106**, 6430–6440, **1997**.
- [193] W. Klopper, K. L. Bak, P. Jørgensen, J. Olsen, T. Helgaker, *J. Phys. B: At. Mol. Opt. Phys.*, **32**, R103–R130, **1999**.
- [194] J. Gauss, J. F. Stanton, *J. Chem. Phys.*, **103**, 3561–3577, **1995**.
- [195] C. Krause, H.-J. Werner, *Phys. Chem. Chem. Phys.*, **14**, 7591–7604, **2012**.
- [196] R. Bellman, *Dynamic Programming*, Dover Publications, Neuauflage, 2003, **1975**.
- [197] T. G. Kolda, *SIAM J. Matrix Anal. Appl.*, **23**, 243–255, **2001**.
- [198] M. Ishteva, L. De Lathauwer, P. A. Absil, S. Van Huffel, *Numerical Analysis and Applied Mathematics*, **1048**, 274–277, **2008**.
- [199] E. Acar, B. Yener, *IEEE T. Knowl. Data. En.*, **21**, 6–20, **2009**.
- [200] W. Hackbusch, B. N. Khoromskij, *SIAM J. Matrix Anal. Appl.*, **30**, 1233–1253, **2008**.
- [201] L. R. Tucker, *Psychometrika*, **31**, 279–279, **1966**.
- [202] L. Grasedyck, *SIAM. J. Matrix Anal. & Appl.*, **31**, 2029–2054, **2010**.
- [203] I. V. Oseledets, E. E. Tyrtyshnikov, *Linear Algebra Appl.*, **432**, 70–88, **2010**.
- [204] I. Oseledets, *SIAM J. Sci. Comput.*, **33**, 2295–2317, **2011**.
- [205] D. Perez-Garcia, F. Verstraete, M. Wolf, J. Cirac, *Quantum Inf. Comput.*, **7**, 401–430, **2007**.
- [206] S. R. White, R. M. Noack, *Phys. Rev. Lett.*, **68**, 3487–3490, **1992**.
- [207] S. R. White, *Phys. Rev. Lett.*, **69**, 2863–2866, **1992**.
- [208] S. Rommer, S. Östlund, *Phys. Rev. B*, **55**, 2164–2181, **1997**.
- [209] K. H. Marti, M. Reiher, *Z. Phys. Chem.*, **224**, 583–599, **2010**.
- [210] K. H. Marti, B. Bauer, M. Reiher, M. Troyer, F. Verstraete, *New J. Phys.*, **12**, 103008–16, **2010**.

- [211] U. Schollwöck, *Annals of Physics*, **326**, 96–192, **2011**.
- [212] R. Kendall, H. Früchtli, *Theor. Chem. Acc.*, **97**, 158–163, **1997**.
- [213] F. Weigend, *Phys. Chem. Chem. Phys.*, **4**, 4285–4291, **2002**.
- [214] H. J. Werner, F. R. Manby, *J. Chem. Phys.*, **124**, 054114–12, **2006**.
- [215] A. M. Burow, M. Sierka, F. Mohamed, *J. Chem. Phys.*, **131**, 214101–6, **2009**.
- [216] K. Eichkorn, O. Treutler, H. Öhm, M. Häser, R. Ahlrichs, *Chem. Phys. Lett.*, **240**, 283–289, **1995**.
- [217] K. Eichkorn, O. Treutler, H. Öhm, M. Häser, R. Ahlrichs, *Chem. Phys. Lett.*, **242**, 652–660, **1995**.
- [218] K. Eichkorn, F. Weigend, O. Treutler, R. Ahlrichs, *Theor. Chem. Acc.*, **97**, 119–124, **1997**.
- [219] F. Neese, F. Wennmohs, A. Hansen, U. Becker, *Chem. Phys.*, **356**, 98–109, **2009**.
- [220] A. Auer, U. Benedikt, M. Espig, W. Hackbusch, in G. Friesecke, P. Gill, editors, *Oberwolfach Reports: Mathematical Methods in Quantum Chemistry*, volume 8, 1808–1812, **2011**.
- [221] N. Shenvi, H. Van Aggelen, W. Yang, *arXiv:1209.2935*, **2012**.
- [222] TensorCalculus library: This library provides several classes representing tensors in different formats and algorithms to deal with them. For current version see: <http://gitorious.org/tensorcalculus>.
- [223] The original coloured picture taken from the data base of the Signal & Image Processing Institute, University of Southern California (<http://sipi.usc.edu/database/database.php?volume=misc&image=12>) is converted to a grey scale picture using gimp. The picture has been originally cropped from a portrait of Lena Söderberg in the November issue of the Playboy magazine 1972. For details on the story see <http://www.cs.cmu.edu/~chuck/lennapg/>.
- [224] D. C. J. Munson, *IEEE Transactions on Image Processing*, **5**, **1996**.
- [225] J. Hutchison, *IEEE Professional Communication Society Newsletter*, **45**, 1–7, **2001**.
- [226] CFOUR, Coupled Cluster techniques for Computational Chemistry, a quantum-chemical program package by J.F. Stanton, J. Gauss, M.E. Harding, P.G. Szalay with contributions from A.A. Auer, R.J. Bartlett, U. Benedikt, C. Berger, D.E. Bernholdt, O. Christiansen, M. Heckert, O. Heun, C. Huber, D. Jonsson, J. Jusélius, K. Klein, W.J. Lauderdale, D. Matthews, T. Metzroth, D.P. O'Neill, D.R. Price, E. Prochnow, K. Ruud, F. Schiffmann, S. Stopkowicz, A. Tajti, M.E. Varner, J. Vázquez, F. Wang, J.D. Watts and the integral packages MOLECULE (J. Almlöf and P.R. Taylor), PROPS (P.R. Taylor), ABACUS (T. Helgaker, H.J. Aa. Jensen, P. Jørgensen, and J. Olsen),

- and ECP routines by A. V. Mitin and C. van Wüllen. For the current version, see <http://www.cfour.de>.
- [227] J. F. Stanton, J. Gauss, J. D. Watts, W. J. Lauderdale, R. J. Bartlett, *Int. J. Quantum Chem.*, 879–894, **1992**.
 - [228] A. A. Auer, M. Nooijen, *J. Chem. Phys.*, 125, 024104–14, **2006**.
 - [229] A. A. Auer, *Z. Phys. Chem.*, 224, 293–309, **2010**.
 - [230] F. Aquilante, T. B. Pedersen, *Chem. Phys. Lett.*, 449, 354–357, **2007**.
 - [231] D. Braess, W. Hackbusch, *IMA J. Numer. Anal.*, 25, 685–697, **2005**.
 - [232] A. Takatsuka, S. Ten-No, W. Hackbusch, *J. Chem. Phys.*, 129, 044112–4, **2008**.
 - [233] P. Y. Ayala, G. E. Scuseria, *J. Chem. Phys.*, 110, 3660–3671, **1999**.
 - [234] S. Hirata, *J. Phys. Chem. A*, 107, 9887–9897, **2003**.
 - [235] A. A. Auer, G. Baumgartner, D. E. Bernholdt, A. Bibireata, V. Choppella, D. Cociorva, X. Y. Gao, R. Harrison, S. Krishnamoorthy, S. Krishnan, C. C. Lam, Q. D. Lu, M. Nooijen, R. Pitzer, J. Ramanujam, P. Sadayappan, A. Sibiryakov, *Mol. Phys.*, 104, 211–228, **2006**.
 - [236] M. Valiev, E. Bylaska, N. Govind, K. Kowalski, T. Straatsma, H. Van Dam, D. Wang, J. Nieplocha, E. Apra, T. Windus, W. de Jong, *Comput. Phys. Commun.*, 181, 1477–1489, **2010**.
 - [237] J. F. Stanton, J. Gauss, J. D. Watts, R. J. Bartlett, *J. Chem. Phys.*, 94, 4334–4345, **1991**.
 - [238] M. Kállay, P. R. Surján, *J. Chem. Phys.*, 115, 2945–2954, **2001**.
 - [239] D. Lyakh, V. Ivanov, L. Adamowicz, *J. Chem. Phys.*, 122, 1–13, **2005**.
 - [240] A. Köhn, G. W. Richings, D. P. Tew, *J. Chem. Phys.*, 129, 201103–5, **2008**.
 - [241] U. Benedikt, H. Auer, M. Espig, W. Hackbusch, A. A. Auer, submitted to *Mol. Phys.*, **2013**.
 - [242] ORCA - An ab initio, DFT and semiempirical SCF-MO package - by F. Neese and F. Wennmohs with contributions from: U. Becker, D. Bykov, D. Ganyushin, A. Hansen, R. Izsak, D. G. Liakos, C. Kollmar, S. Kossmann, D. A. Pantazis, T. Petrenko, C. Reimann, C. Ripplinger, M. Roemelt, B. Sandhöfer, I. Schapiro, K. Sivalingam, B. Wezisla and contributions from collaborators: M. Kállay, S. Grimme, E. Valeev, for current version see: <http://cec.mpg.de/forum/>.
 - [243] F. Neese, *J. Comput. Chem.*, 24, 1740–1747, **2003**.
 - [244] F. Neese, *WIREs Comput. Mol. Sci.*, 2, 73–78, **2012**.

- [245] K. P. Huber, G. H. Herzberg, *Molecular Spectra and Molecular Structure IV: Constants of Diatomic Molecules*, Van Nostrand-Reinhold, New York, **1979**.
- [246] A. R. Hoy, I. M. Mills, G. Strey, *Mol. Phys.*, **24**, 1265–1290, **1972**.
- [247] J. L. Duncan, I. M. Mills, *Spectrochimica Acta*, **20**, 523–546, **1964**.
- [248] D. L. Gray, A. G. Robiette, *Mol. Phys.*, **37**, 1901–1920, **1979**.
- [249] W. J. Hehre, R. F. Stewart, J. A. Pople, *J. Chem. Phys.*, **51**, 2657–2664, **1969**.
- [250] W. J. Hehre, R. Ditchfie, J. A. Pople, *J. Chem. Phys.*, **56**, 2257–2261, **1972**.
- [251] P. C. Hariharan, J. A. Pople, *Theor. Chim. Acta*, **28**, 213–222, **1973**.
- [252] A. Schäfer, H. Horn, R. Ahlrichs, *J. Chem. Phys.*, **97**, 2571–2577, **1992**.
- [253] F. Weigend, R. Ahlrichs, *Phys. Chem. Chem. Phys.*, **7**, 3297–3305, **2005**.
- [254] T. H. Dunning, *J. Chem. Phys.*, **90**, 1007–1023, **1989**.
- [255] R. A. Kendall, T. H. Dunning, R. J. Harrison, *J. Chem. Phys.*, **96**, 6796–6806, **1992**.

Curriculum Vitae

Persönliche Daten

Name	Udo Benedikt
Adresse	Eduard-Bilz-Str. 39, 01445 Radebeul
Geburtsdatum	20.07.1983
Geburtsort	Annaberg-Buchholz
Nationalität	deutsch

Schul- und Berufsbildung

- 09/2008–03/2013 Promotionsstudium, Honorarprofessur für Computergestützte Quantenchemie, Technische Universität Chemnitz, “Low-Rank Tensor Approximation in post Hartree-Fock Methods”
- 11/2007–08/2008 Diplomarbeit, Juniorprofessur für Theoretische Chemie, Technische Universität Chemnitz, “Optimierung von Basissätzen und hochgenaue Berechnung von indirekten Spin-Spin Kopplungskonstanten”
(Prädikat: sehr gut)
- 07/2007 Auslandsaufenthalt an der “University of Southern Denmark”, Dänemark
- 10/2003–08/2008 TU Chemnitz, Studiengang Diplom Chemie, Vertiefungsrichtung Katalyse und Grenzflächenchemie
- 08/1994–06/2002 Abitur, Landkreisgymnasium Annaberg (Note: 1,3)
- 09/1990–07/1994 Grundschule Cranzahl

Publikationen

- 2013 U. BENEDIKT, M. ESPIG, W. HACKBUSCH, and A. A. AUER, Tensor decomposition in post-Hartree–Fock methods. II. LCCD and CCD implementation, in preparation for *J. Chem. Phys.* (2013).
- 2013 U. BENEDIKT, H. AUER, M. ESPIG, W. HACKBUSCH, and A. A. AUER, Tensor Representation Techniques in post-Hartree-Fock Methods: Matrix Product State Tensor Format, in press *Mol. Phys.* (2013).

- 2013 W. B. SCHNEIDER, U. BENEDIKT, and A. A. AUER, Interaction of Platinum Nanoparticles with Graphitic Carbon Structures - A Computational Study, accepted *Chem. Phys. Chem.* (2013).
- 2013 I. KATSOUNAROS, W. B. SCHNEIDER, J. C. MEIER, U. BENEDIKT, P. U. BIEDERMANN, A. CUESTA, A. A. AUER, and K. J. J. MAYRHOFER, The impact of spectator species on the interaction of H₂O₂ with platinum – implications for the oxygen reduction reaction pathways, *Phys. Chem. Chem. Phys.* **15**, 8058–8068 (2013).
- 2013 U. BENEDIKT, W. B. SCHNEIDER, and A. A. AUER, Modelling electrified interfaces in quantum chemistry: constant charge vs. constant potential, *Phys. Chem. Chem. Phys.* **15**, 2712–2724 (2013).
- 2012 I. KATSOUNAROS, W. B. SCHNEIDER, J. C. MEIER, U. BENEDIKT, P. U. BIEDERMANN, A. A. AUER, and K. J. J. MAYRHOFER, Hydrogen peroxide electrochemistry on platinum: towards understanding the oxygen reduction reaction mechanism, *Phys. Chem. Chem. Phys.* **14**, 7384–7391 (2012).
- 2011 A. A. AUER, U. BENEDIKT, M. ESPIG, and W. HACKBUSCH, Tensor Decomposition in Electronic Structure Theory: Canonical Product Format and Coupled Cluster Theory, *Oberwolfach Reports: Mathematical Methods in Quantum Chemistry* **8**, 1808–1812, (2011).
- 2011 U. BENEDIKT, M. ESPIG, W. HACKBUSCH, and A. A. AUER, Tensor decomposition in post-Hartree–Fock methods. I. Two-electron integrals and MP2, *J. Chem. Phys.* **134**, 054118 (2011).
- 2010 D. POLSTER, H. GRAAF, T. BAUMGÄRTEL, C. VON BORCZYSKOWSKI, U. BENEDIKT, and A. A. AUER, Beyond Cassie's law - a theoretical and experimental study of mixed alkyl monolayers, *Langmuir* **26**, 8301 (2010).
- 2009 M. LEHMANN, J. SELTMANN, A. A. AUER, E. PROCHNOW and U. BENEDIKT, Synthesis and mesomorphic properties of new V shaped shape persistent nematogens containing a thiazole or a thiadiazole bending unit, *J. Mater. Chem.* **19**, 1978 (2009).
- 2008 U. BENEDIKT, A. A. AUER, and F. JENSEN, Optimization of Augmentation Functions for Correlated Calculations of Spin-Spin Coupling Constants and Related Properties, *J. Chem. Phys.* **129**, 064111 (2008).
- 2007 U. BENEDIKT, A. A. AUER, and F. JENSEN, Optimized Basis Sets for Calculating Spin-Spin Coupling Constants, *AIP Conference Proceedings* **963**, 183–186 (2007).

Vorträge auf Tagungen und Konferenzen

- 12/2012 Developers Meeting, MPI CEC Mülheim, *Tensor Representation Techniques in Electronic Structure Methods*, U. BENEDIKT, M. ESPIG, W. HACKBUSCH, and A. A. AUER.
- 03/2012 Developers Meeting, MPI BAC Mülheim, *Low Rank Tensor Representation in Post HF Methods*, U. BENEDIKT, M. ESPIG, W. HACKBUSCH, and A. A. AUER.
- 01/2012 CES Symposium, RUB Bochum, *Constant Charge vs. Constant Potential*, U. BENEDIKT, W. B. SCHNEIDER, and A. A. AUER.
- 07/2011 CES Minisymposium: Theory meets Experiment, MPIE Düsseldorf, *Nanoparticles in Electrocatalysis - Oxygen Reduction Reaction and Carbon Support Interactions*, U. BENEDIKT, W. B. SCHNEIDER, M. KETTNER and A. A. AUER.
- 08/2010 CES Minisymposium: Challenges for Theory in Electrochemistry, RUB Bochum, *Oxygen Reduction Reaction on Pt-Nanoparticles: A Density-functional based Study*, U. BENEDIKT, W. B. SCHNEIDER, and A. A. AUER.
- 05/2010 Gruppenseminar Theoretische Chemie, Johannes Gutenberg-Universität Mainz, *Tensor decomposition in post-Hartree-Fock methods*, U. BENEDIKT, M. ESPIG, W. HACKBUSCH, and A. A. AUER.
- 04/2010 Tensor Decomposition Workshop, MPIE Düsseldorf, *Recent progress in tensor decomposition for post-Hartree-Fock methods*, U. BENEDIKT, and A. A. AUER.
- 09/2009 Tensor Decomposition Workshop, MPI-MIS Leipzig, *Tensor Decomposition in post-HF Methods: First Results*, U. BENEDIKT, and A. A. AUER.
- 06/2009 Doktorandenseminar, TU Chemnitz, *Theoretische Chemie in Synthese, Spektroskopie & Materialwissenschaften: Zwischen Labor und Supercomputer-Cluster*, U. BENEDIKT.
- 12/2008 CHiC Symposium, TU Chemnitz, *Optimierung von Basissätzen und hochgenaue Berechnung von Spin-Spin Kopplungskonstanten*, U. BENEDIKT, and A. A. AUER.

Poster auf Tagungen und Konferenzen

- 09/2012 48. Symposium für Theoretische Chemie (STC), KIT Karlsruhe, *Low Rank Tensor Representation in Post Hartree-Fock Methods*, U. BENEDIKT, M. ESPIG, W. HACKBUSCH, and A. A. AUER.
- 09/2012 GDCH Electrochemistry 2012, München, *Oxygen Reduction Reaction on Pt-Nanoparticles*, U. BENEDIKT, W. B. SCHNEIDER, and A. A. AUER.
- 10/2011 Modern developments in the ab initio description of charged systems for semiconductors and electrochemistry (ai-WISE 2011), Ringberg, *Oxygen*

- Reduction Reaction on Pt-Nanoparticles: A Density-functional based Study , U. BENEDIKT, W. B. SCHNEIDER, and A. A. AUER.
- 09/2011 47. Symposium für Theoretische Chemie (STC), Sursee, *Low Rank Tensor Representation in Coupled Cluster Theory*, U. BENEDIKT, M. ESPIG, W. HACKBUSCH, and A. A. AUER.
- 09/2010 GDCH Electrochemistry 2010, Bochum, *Oxygen Reduction Reaction on Pt-Nanoparticles: A Density-functional based Study I*, U. BENEDIKT, W. B. SCHNEIDER, and A. A. AUER.
- 09/2010 46. Symposium für Theoretische Chemie (STC), Münster, *Oxygen Reduction Reaction on Pt-Nanoparticles: A Density-functional based Study*, U. BENEDIKT, W. B. SCHNEIDER, and A. A. AUER.
- 04/2010 First Principles in Quantum Chemistry, Bad Herrenalb, *Tensor decomposition in post-HF methods*, U. BENEDIKT, M. ESPIG, W. HACKBUSCH, and A. A. AUER.
- 04/2010 Abschlusssymposium SPP 114, Bad Herrenalb, *Tensor decomposition in post-HF methods*, U. BENEDIKT, M. ESPIG, W. HACKBUSCH, and A. A. AUER.
- 02/2010 International Symposium on Theoretical and Computational Chemistry (TCC), Max-Planck-Institut für Kohlenforschung, Mülheim an der Ruhr, *Tensor decomposition in post-HF methods*, U. BENEDIKT, M. ESPIG, W. HACKBUSCH, and A. A. AUER.
- 09/2009 45. Symposium für Theoretische Chemie (STC), Neuss, *Tensor Product Approximation Techniques in Coupled-Cluster Theory*, U. BENEDIKT, M. ESPIG, W. HACKBUSCH, and A. A. AUER.
- 06/2009 13. International Congress of Quantum Chemistry (ICQC), Helsinki, *Code Generation for Local Correlation Methods and Tensor Decomposition Techniques*, U. BENEDIKT, M. ESPIG, W. HACKBUSCH, and A. A. AUER.
- 03/2009 Workshop SPP 1145, Bensberg, *Exploratory Investigations: Code Generation for Local Correlation Methods and Tensor Decomposition Techniques*, U. BENEDIKT, M. ESPIG, W. HACKBUSCH, and A. A. AUER.
- 09/2008 44. Symposium für Theoretische Chemie (STC), Ramsau am Dachstein, *Optimization of Augmentation Functions for Correlated Calculations of Spin-spin Coupling Constants and Related Properties*, U. BENEDIKT, A. A. AUER, and F. JENSEN.

Selbstständigkeitserklärung

Hiermit erkläre ich, daß ich die vorliegende Arbeit selbstständig angefertigt, nicht anderweitig zu Prüfungszwecken vorgelegt und keine anderen als die angegebenen Hilfsmittel verwendet habe. Sämtliche wissentlich verwendete Textausschnitte, Zitate oder Inhalte anderer Verfasser wurden ausdrücklich als solche gekennzeichnet.

Chemnitz, den 15.05.2013

Udo Benedikt