High-performance computing and its applications to atomic structure physics

Emily V Kahl

Supervisor

A. Prof. Julian Berengut

School of Physics
Faculty of Science

This thesis is submitted in requirement for the degree of Doctor of Philosophy in physics

February 13, 2021

Thesis/Dissertation Sheet

Surname/Family Name	:	**Kahl**
Given Name/s	:	**Emily Violet**
Abbreviation for degree as give in the University calendar	:	**PhD**
Faculty	:	**Science**
School	:	**Physics**
Thesis Title	:	**High-performance computing and its applications to atomic structure physics**

Abstract 350 words maximum: (PLEASE TYPE)

Modern atomic physics is increasingly dependent on the results of high-precision calculations to guide experiments and applications, especially in complex atoms with dense spectra. Many cutting-edge applications and experiments use atoms with open-shell electronic structure and strong configuration mixing, which require extremely large numerical bases to treat with any level of accuracy. At the same time, supercomputing clusters have seen huge increases in computational power, driven by increasingly large-scale parallelism across many distributed compute nodes. Consequently, modern atomic structure code must be designed to fully utilise massively-parallel computing resources if they are to keep up with the increasing demands of experimental studies.

In this thesis, I present the results of work to modernise the AMBiT atomic structure software, which implements the configuration interaction with many-body perturbation theory (CI+MBPT) method, to take advantage of modern supercomputers. I present a detailed outline of the software engineering processes in converting AMBiT from an MPI-only model of parallelism to a hybrid MPI+OpenMP model, as well as the performance gains resulting from doing so. I show that the increased parallelism allows us to explore numerical saturation of the CI+MBPT method in open-shell atoms for the first time ever --- an investigation would not have been possible without the increased performance capabilities of modern supercomputers.

I have applied the new AMBiT to calculations of atomic systems with a variety of electronic structures. Calculations of the highly-charged ions Sn^{7+}-Sn^{10+}, which are of experimental interest for their applications in extreme ultraviolet photolithography for semiconductor fabrication, show that AMBiT is highly efficient for ions with open d-shells. We achieve very close agreement with experimental spectra: CI+MBPT calculations differ from experiments by an average error of less than 1%. Additionally, calculations for two- and three-valent Lr^{+} and Lr demonstrate that CI+MBPT implemented in AMBiT can accurately calculate the spectra of superheavy elements - systems in which relativistic and QED effects are significant. My calculations for Lr and Lr^{+} also show that our CI+MBPT implementation is competitive with other cutting-edge methods for relativistic atomic structure calculations.

This research shows that the accuracy of the CI+MBPT method, when scaled out to numerical saturation, is competitive with best-in-class methods for atomic structure calculations and can continue to serve as a workhorse for next-generation atomic structure calculations. Following its release, AMBiT has been used by multiple research groups for calculations of highly-charged ion clocks and precision tests for physics beyond the standard model. Furthermore, this research should put AMBiT and CI+MBPT in a strong position to scale up and capitalise on future gains in high-performance computing technology.

Declaration relating to disposition of project thesis/dissertation

……………………………………………………………… ……………………………………….

Signature Date

ORIGINALITY STATEMENT

‘I hereby declare that this submission is my own work and to the best of my knowledge it contains no materials previously published or written by another person, or substantial proportions of material which have been accepted for the award of any other degree or diploma at UNSW or any other educational institution, except where due acknowledgement is made in the thesis. Any contribution made to the research by others, with whom I have worked at UNSW or elsewhere, is explicitly acknowledged in the thesis. I also declare that the intellectual content of this thesis is the product of my own work, except to the extent that assistance from others in the project's design and conception or in style, presentation and linguistic expression is acknowledged.’

Signed ……………………………………………….............

Date ……………………………………………….............

COPYRIGHT STATEMENT

Signed ……………………………………………..........................

Date ……………………………………………...........................

AUTHENTICITY STATEMENT

‘I certify that the Library deposit digital copy is a direct equivalent of the final officially approved version of my thesis.’

Signed ……………………………………………..........................

Date ……………………………………………...........................

INCLUSION OF PUBLICATIONS STATEMENT

UNSW is supportive of candidates publishing their research results during their candidature as detailed in the UNSW Thesis Examination Procedure.

Publications can be used in their thesis in lieu of a Chapter if:

- The candidate contributed greater than 50% of the content in the publication and is the "primary author", ie. the candidate was responsible primarily for the planning, execution and preparation of the work for publication
- The candidate has approval to include the publication in their thesis in lieu of a Chapter from their supervisor and Postgraduate Coordinator.
- The publication is not subject to any obligations or contractual agreements with a third party that would constrain its inclusion in the thesis

Please indicate whether this thesis contains published material or not:

□ This thesis contains no publications, either published or submitted for publication
(if this box is checked, you may delete all the material on page 2)

☑ Some of the work described in this thesis has been published and it has been documented in the relevant Chapters with acknowledgement
(if this box is checked, you may delete all the material on page 2)

□ This thesis has publications (either published or submitted for publication) incorporated into it in lieu of a chapter and the details are presented below

CANDIDATE'S DECLARATION

I declare that:

- I have complied with the UNSW Thesis Examination Procedure
- where I have used a publication in lieu of a Chapter, the listed publication(s) below meet(s) the requirements to be included in the thesis.

Candidate's Name	Signature	Date (dd/mm/yy)

Abstract

Modern atomic physics is increasingly dependent on the results of high-precision calculations to guide experiments and applications, especially in complex atoms with dense spectra. Many cutting-edge applications and experiments use atoms with open-shell electronic structure and strong configuration mixing, which require extremely large numerical bases to treat with any level of accuracy. At the same time, supercomputing clusters have seen huge increases in computational power, driven by increasingly large-scale parallelism across many distributed compute nodes. Consequently, modern atomic structure code must be designed to fully utilise massively-parallel computing resources if they are to keep up with the increasing demands of experimental studies.

In this thesis, I present the results of work to optimise the AMBiT atomic structure software, which implements the configuration interaction with many-body perturbation theory (CI+MBPT) method, to take advantage of modern supercomputers. I present a detailed outline of the software engineering processes in converting AMBiT from an MPI-only model of parallelism to a hybrid MPI+OpenMP model, as well as the performance gains resulting from doing so. I show that the increased parallelism allows us to explore numerical saturation of the CI+MBPT method in open-shell atoms for the first time ever — an investigation would not have been possible without the increased performance capabilities of modern supercomputers.

I have applied the new AMBiT to calculations of atomic systems with a variety of electronic structures. Calculations of the highly-charged ions Sn^{7+}–Sn^{10+}, which are of experimental interest for their applications in extreme ultraviolet photolithography for semiconductor fabrication, show that AMBiT is highly efficient for ions with open *d*-shells. We achieve very close agreement with experimental spectra: CI+MBPT calculations differ from experiments by an average error of less than 1%. Additionally, calculations for two- and three-valent Lr^{+} and Lr demonstrate that CI+MBPT implemented in AMBiT can accurately calculate the spectra of superheavy elements — systems in which relativistic and QED effects are significant. My calculations for Lr and Lr^{+} also show that our CI+MBPT implementation is competitive with other cutting-edge methods for relativistic atomic structure calculations.

This research shows that the accuracy of the CI+MBPT method, when scaled out to numerical saturation, is competitive with best-in-class methods for atomic structure calculations and can continue to serve as a workhorse for next-generation atomic structure calculations. Following its release, AMBiT has been used by multiple research groups for calculations of highly-charged ion clocks and precision tests for physics beyond the standard model. Furthermore, this research should put AMBiT and CI+MBPT in a strong position to scale up and capitalise on future gains in high-performance computing technology.

Contents

1 Introduction and motivation 3

1.1 Motivation . . . 3

1.2 Highly-charged ions . . . 4

1.3 Superheavy elements . . . 8

1.4 Differential measurements and precision tests for physics beyond the standard model 10

1.4.1 Variation of the fundamental constants . . . 11

1.4.2 Searches for dark matter and exotic particles . . . 13

1.5 Aims and outlines of this thesis . . . 14

2 Modelling atomic structure: CI+MBPT 17

2.1 Overview of methods for atomic structure . . . 17

2.1.1 Other codes . . . 17

2.2 CI+MBPT Theory . . . 19

2.2.1 Single particle basis . . . 19

2.2.2 CI Basis . . . 21

2.2.3 CI+MBPT . . . 23

2.2.4 Emu CI . . . 26

3 High-performance computing — OpenMP 29

3.1 Glossary . . . 29

3.2 Introduction . . . 30

3.3 Why do we need OpenMP? . . . 33

3.4 A primer on computer architecture . . . 34

3.5 Profiling . . . 36

3.6 OpenMP . . . 42

3.7 Parallelism and optimisation methodology . . . 44

3.7.1 Two-body MBPT integrals . . . 44

3.7.2 Generating the Hamiltonian Matrix . . . 49

3.7.3 Many-body matrix elements . . . 54

3.7.4 Slater Integrals . . . 56

3.7.5 Choice of data structures - associative arrays . . . 58

3.8 Final performance analysis . . . 63

4 Calculations: Benchmarks — Cr^+ and Db **69**
4.1 Cr^+ 69
4.1.1 Dirac-Fock and B-spline basis 69
4.1.2 Large-scale calculation 70
4.2 Db – CI+MBPT convergence 72
4.3 Conclusion 76

5 Calculations: Tin highly-charged ions **77**
5.1 Experimental motivation 77
5.2 Theory and Calculations 78
5.3 Comparison with experimental results 92

6 Calculations: Lr^+ and Lu^+ **93**
6.1 Introduction 93
6.2 Methods and Computational Details 94
6.3 FSCC 95
6.4 CI+MBPT 96
6.5 Results 97
6.6 Summary and conclusion 98

7 Calculations: Neutral Lr and Lu **103**
7.1 Introduction 103
7.2 Methods and computational Details 104
7.2.1 CI+MBPT 105
7.3 Results 106
7.4 Summary of results 110

8 Conclusion **119**
8.1 Perspectives for the future 119
8.1.1 Computational improvements – GPU computing 119
8.1.2 CI+MBPT implementation improvements 121
8.2 Summing up 123

Preface

Some of the content of this thesis was published in the following papers:

- Kahl, Berengut, Raeder, Eliav and Borschevsky, *Calculations of the spectrum and transitions of Lr*, **To be submitted**
- Kahl, Berengut, Laatiaoui, Eliav and Borschevsky, *High-precision* ab initio *calculations of the spectrum of* Lr^{+},
  Phys. Rev. A. **100**, 062505 (2019)
- Kahl and Berengut, *AMBiT: A program for high-precision relativistic atomic structure calculations*,
  Computer Physics communications **238**, 232 – 243 (2018)
- Geddes, Czapski, Kahl and Berengut, *Saturated configuration interaction calculations for five-valent Ta and Db*,
  Phys. Rev. A **98** 042508 (2018)
- Toretti, Windberger, Ryabtsev, Dobrodey, Bekker, Ubachs, Hoekstra, Kahl, Berengut, Crespo López-Urrutia, Versolato , *Optical spectroscopy of complex open-4d-shell ions* Sn^{7+} – Sn^{10+},
  Phys. Rev. A **95**, 042503 (2017)

Acknowledgements

Firstly, I want to thank my PhD supervisor A. Prof. Julian Berengut, who is the gold-standard for being both a good supervisor and a good boss. Julian has consistently gone above and beyond the call of duty in supervising, teaching and supporting me and without his guidance I wouldn't be half the scientist and technologist I am now. He's also a top bloke.

Secondly, I'd like to thank my officemates Amy, Bryce and Daniel, as well as the rest of the UNSW Department of Theoretical Physics, for insightful conversations and company, invaluable feedback on my writing and software, and early morning cups of coffee. Extra thanks are due to Daniel for digging up the raw data and scripts for the figures in chapter 4. I also really want to thank Sue, Seda and the rest of the UNSW Physics admin team — they've been endlessly supportive of me and are very good people.

Thirdly, I would like to thank Prof. Anastasia Borschevsky for fruitful collaborations, introducing me to all sorts of interesting things in atomic and molecular science, and hosting me at her institute for three months in 2018.

Next up, a big "thank you" to Dr Michael Bromley and Dr Jacinda Ginges. Our technical discussions on matters physical and computational were incredibly informative, and changed the way I think about a lot of things in computational physics.

I would be remiss not to acknowledge the wonderful system administrators for the HPC clusters I used during this thesis: Martin Thompson at UNSW Science, Ian Mortimer at UQ Physics, Bob Dröge at the RUG Center for Information Technology, and the user support team at NCI. Without their assistance in getting AMBiT running on their clusters I would have many more grey hairs than at present.

And last but not least, my friends and family. A huge "thank you" to my life-partner and wife-partner Zoe, who is wonderful beyond words. I must also thank my parents (for obvious reasons), my dear friend Duck for enlightening conversations and support going back many years, Isaac and Bruce for being top blokes, Merryn and Rohan for keeping me grounded, and Stuart for joining me in yelling about computers. They're all good folks and I wouldn't have made it to this point without 'em.

Chapter One

Introduction and motivation

This thesis details the development of high-performance parallel algorithms for relativistic many-body atomic structure calculations. Throughout this thesis, I will outline the current state of the art in many-electron atomic structure methods, as well as the modified CI+MBPT techniques employed by the AMBiT atomic structure code, developed by the Berengut group at UNSW. I will then present high-performance, massively-parallel implementations of these algorithms and demonstrate their performance and scalability modern high-performance computing clusters. Finally, I will present several calculations of challenging atomic and ionic systems via these algorithms and demonstrate their top-of-class accuracy and performance.

1.1 Motivation

This work is motivated by the increasing need for precision calculations to support new experiments in atomic physics. Due to the complicated mathematics involved in atomic structure physics (for example, there is no analytical solution to the Dirac equation for atoms with more than one electron), numerical calculations play an important role in supporting experiments, as well as providing accurate predictions to guide the development of new applications. In recent years, atomic experiments and calculations have made key contributions to disparate areas in physics, such as:

- Astrophysics, including stellar, solar and coronal dynamics [1, 2, 3], thermal and optical properties of kilonovae [4], and diagnostics of nebulae and stellar atmospheres [5, 6, 7].
- Precision metrology: optical atomic clocks are extremely stable, with Yb^+ ion clocks [8] and Sr lattice clocks [9] reaching fractional systematic uncertainties at the level of 10^{-18}. At this level of precision, atomic clocks are sensitive to the effects of physics beyond the standard model (which is discussed in more detail below) and can serve applications in precision quantum sensing and geodesy [10, 11].
- Photon-atom scattering and polarisabilities: quantifying the effects of environmental perturbations and black-body radiation shifts on atomic clocks [12, 13, 14], polarisability and Stark shifts in the spectra of complex atoms [15, 16], and modelling Van der Waals forces

in weakly bound molecules [17].

Many of these applications rely on atoms and ions with complex electronic structures, especially open-shell systems (i.e. atoms or ions with partially filled valence shells) and strong configuration mixing. These properties present serious difficulties for precision calculations, due to the large calculation sizes required to accurately model complex configurations. Additionally, photon-atom applications typically require treatment of the energy-levels and transition matrix elements for both bound and continuum states, which are difficult for current-generation codes to calculate in tandem to the desired level of accuracy.

Narrowing down, this thesis will focus on atomic structure calculations motivated by two recent experimental breakthroughs in precision atomic spectroscopy: precision spectroscopy of highly-charged ions and experimental spectroscopy of superheavy elements. These breakthroughs have enabled atomic experiments with a degree of accuracy which would have been unfeasible or outright impossible a decade ago. Both highly-charged ions and superheavy elements present considerable theoretical and computational challenges when computing spectra to the level of accuracy required by experiments, necessitating the development of a new generation of atomic structure codes.

Calculations for open-shell atomic systems are also critically important for experiments aiming to provide precision tests for physics beyond the standard model, which serve as an important search for new physics independent of high-energy experiments particle accelerator experiments. These three applications (highly-charged ions, superheavy elements, and precision searches for new physics), as well as their computational challenges, are described in more detail below.

1.2 Highly-charged ions

“Clock-like” precision spectroscopy of highly-charged ions (HCI) has only recently become experimentally feasible due to three chief breakthroughs:

1. Accessibility of transitions: Early studies of highly-charged ions typically focused on few-electron systems which could be produced in accelerator-synchrotron experiments, such as the hydrogen-like systems U^{91+} [18, 19, 20], C^{5+} [21], and O^{7+} [22], as well as helium-like U^{90+} [20]. Hydrogen- and helium-like highly-charged ions have been used to perform precision tests of QED via electron g-factor experiments [21, 22] and lamb-shift measurements [20], since the size of atomic QED corrections are much stronger in ions with high ionic charge. The review article by Kozlov *et al* [23] provides a detailed overview of historical HCI experiments.

   The above-mentioned highly-charged ions have few electrons, so the electronic binding energies are dominated by extremely strong nuclear charge contributions, leading to spectra in the far UV or X-ray regions. This makes precision spectroscopy experiments difficult, although work carried out at NIST and elsewhere [24, 25, 26] has allowed few-electron HCI systems to be used for tests of QED with accuracy at the parts-per-million level [27, 28].

   PPM accuracy is not sufficient for precision clocks, however, so we must seek out optically-accessible transitions to achieve “clock-precision” spectroscopy. Berengut *et al* [29] demonstrated the existence of optical, near-degenerate transitions in many-electron highly-charged ions. These transitions exist near so-called "level crossings" due to cancellations between large electron-electron contributions to binding energies. The existence of optical transitions not only allows for highly-precise spectroscopic experiments, but also opens the possibility of using HCI systems as optical frequency atomic clocks. The high binding energies of HCIs result in relatively small electronic polarisabilities and thus high-stability, as well as sensitivity to physics beyond the standard model, such as variation of the fundamental constants [23, 30].

2. Production and trapping: the production and trapping of stationary HCIs is an important prerequisite to precision HCI spectroscopy. Early methods of production such as accelerators or the even earlier beam-foil method [31, 32] produced HCIs with large average velocities (order of mm/ns) and thus only spend extremely short amounts of time in a given experimental apparatus. This meant that only transitions with short lifetimes could be observed in early experiments; the transition rate of the long-lived transitions required for optical clocks is far too low to be observed at high velocities [23].

   The development of electron-beam ion traps (EBITs) were key in overcoming the problems of earlier methods of producing highly-charged ions. An EBIT consists of a high-intensity, magnetically focused and confined beam of electrons with currents in the mA to A range, which interacts with a dilute sample of the target (neutral) system. The electrons simultaneously ionise the sample (with the maximum charge state tunable via the kinetic energy of the electrons) and provide a confining electric potential, thus combining HCI production and trapping into one experimental apparatus [33]. EBITs were originally developed by the group of Levine *et al* to study the atomic structure of highly-charged plasmas under controlled, stationary conditions [33]. Further work [34] showed that EBITs were highly-effective means of studying hydrogen- and helium-like ions of heavy elements. EBITs are now used to study HCIs across all charge states at a fractional uncertainty in frequency measurements of $< 10^{-6}$ [35, 26].

   However, while EBITs opened up significant new areas of experimental applications in highly-charged systems, the ions produced and trapped in EBITs typically have temperatures of the order of 10^4 K. These high temperatures introduce significant thermal broadening of spectral lines, requiring either extensive cooling or sophisticated experimental techniques to compensate for thermal effects (see Refs. [36, 24] for an overview of experimental developments in this area).

3. Cooling and state production: HCIs are usually produced at very high temperatures ($\sim 10^4 - 10^6$ K) whether in an EBIT or otherwise, which, as previously discussed, makes precision spectroscopy difficult due to large Doppler/thermal broadening in the sample. Cooling of near-neutral ions typically relies on some combination of evaporative- and laser-cooling techniques, the latter of which is unsuited to most highly-charged ions. Laser cooling relies on the presence of fast-cycling optical transitions, but these are not present in HCIs — low-lying optical transitions in HCIs are strongly forbidden and thus have lifetimes

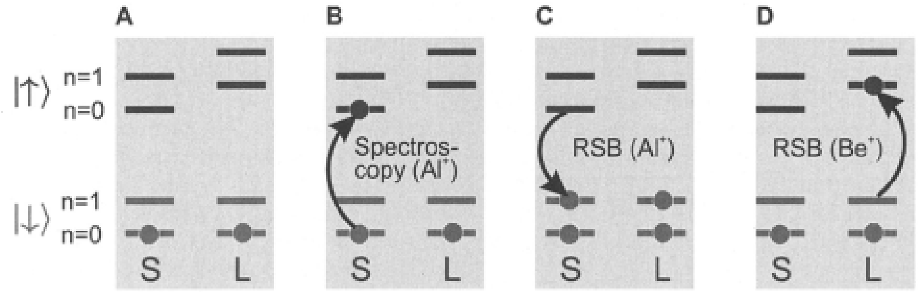
too long to be useful for laser cooling, while the dipole-allowed transitions are all in the EUV or X-ray ranges [37]. Consequently, the last key development in the spectroscopy of highly-charged ions is the development of cooling and trapping schemes to reduce the temperature of an HCI sample to the mK regime, with the more recent breakthroughs coming via sympathetic cooling.

The sympathetic cooling approach relies on embedding HCIs (produced in an EBIT) into a sample of cold ions which have already undergone Coulomb crystallisation. The earliest attempts at this technique built on the work of Levine [33], and were carried out for Xe^{44+} in Refs. [38, 39] by using a dilute Be^{+} buffer gas in a cryogenic Penning trap. In this scheme, Be^{+} and Be^{2+} ions are produced by electrically ionising a metal vapour cloud, before the Be^{+} ions are magnetically separated, decelerated and captured in the Penning trap. The Be^{+} ions are then laser cooled to ~ 100 mK and the cooling laser is left to run continuously throughout the experiment. The hot Xe ions (produced from an EBIT) are then injected into the Penning trap, where they are cooled by Coulomb interaction with the cryogenic Be^{+} ions, which are continuously re-cooled via the active laser. The combined plasma in the Penning trap tends to an equilibrium temperature of ≤ 1 K, with the exact temperature depending on the laser cooling rate, Coulomb interaction strength and rates of external heating (e.g. trap imperfections).

The Penning trap method is relatively simple and effective at trapping highly-charged ions due to their large charge-to-mass ratios, and has proven to be effective at cooling systems for which laser cooling is impossible or impractical. However, the strong magnetic fields required by Penning traps tends to introduce field noise when used for laser spectroscopy [37], which reduces the precision of the spectral measurements. Consequently, more recent work such as that in Ref. [40, 41] has focused on using radio-frequency (RF) Paul traps for precision spectroscopy. Both Penning traps and RF traps employ hyperbolic electrodes to produce a quadrupole electric field, but where Penning traps achieve three-dimensional confinement via a static magnetic field, an RF trap oscillates the quadrupole field to achieve the trapping [42].

RF trapping employs a system of pre-cooling and trapping similar to that in ref [38, 39], but using an RF trap for the spectroscopy chamber. Additionally, *quantum logic spectroscopy* can be used to compensate for ions which lack suitable cooling/trapping transitions by mapping the target *spectroscopy ion*'s internal state to that of a *logic ion* with more favorable characteristics via their Coulomb interactions and collective motion in the trap [43, 44]. This allows for sympathetic cooling of the spectroscopic ion to very low temperatures, as well as noise-suppressed measurements of the target clock-frequency.

The coupled motion of the trap and clock ions also allows for precision spectroscopy of ions like Ar^{13+} which lack convenient transitions for electronic state preparation and detection via *quantum logic spectroscopy* [43, 45]. A schematic representation of this process for an Al^{+} spectroscopy ion and Be^{+} logic ion is shown below (figure reproduced with permission from Ref. [43]), and proceeds as follows:



A The system, consisting of the spectroscopy ion (S) and the logic ion (L) are prepared in their respective electronic ground states and in the coupled mode of the trap. Although only the ground and first excited states of this mode are shown, it will have a full excitation spectrum which is superimposed over the internal electronic energy spectrum. After this step, both ions are in the electronic ground state $|{\downarrow}\rangle$ and the ground state of the coupled mode ($n = 0$).

B The spectroscopy ion is illuminated by a probe laser with energy near the clock transition. If the probe frequency is not equal to the clock frequency, then the spectroscopy ion remains in the ground state and no further excitations occur. If the spectroscopy ion *does* get excited to the clock state, then the system proceeds to phase **C**. After this step, the spectroscopy ion is in $|{\uparrow}\rangle, n = 0$, while the logic ion remains unchanged.

C The spectroscopy ion is illuminated by a red sideband π-pulse (RSB(Al^+) in the diagram) to de-excite it to the electronic ground-state, while simultaneously exciting the trap's coupled mode. The Coulomb interaction between the two ions ensures that the logic ion is also promoted into the coupled-mode's excited state. After this step, both ions are in their respective $|{\downarrow}\rangle, n = 1$ states if the spectroscopy pulse in phase **B** was successful, otherwise the red sideband pulse does not affect the system.

D A red sideband pulse, this time near the logic ion's more well-characterised transition, is then applied to the logic ion to excite it to $|{\uparrow}\rangle, n = 0$.

The energy of this pulse is only sufficient to excite the logic ion if the spectroscopy pulse in phase **B** had the correct frequency to excite the spectroscopy ion. The Coulomb interaction between the two ions ensures that the spectroscopy ion also transfers to the coupled-mode ground state. After this step, the spectroscopic ion will be in the $|{\downarrow}\rangle, n = 0$ state, while the logic ion will be in the $|{\uparrow}\rangle, n = 0$ state.

The key to this process is that each successive laser pulse is only able to change the state of the system if the previous phase of the process was successful. Consequently, the logic ion will absorb the second red sideband pulse by transferring to its electronic excited state in phase **D** if and only if the initial pulse in phase **B** matched the clock frequency of the spectroscopy ion. Consequently, repeated measurement the absorption profile (or subsequent fluorescence through de-excitation) by the logic ion while tuning the spectroscopy pulse provides a means to measure the clock frequency with an extremely low fractional uncertainty.

> Quantum logic spectroscopy allows for precision spectroscopy of atomic [44] or molecular ions [45, 46] with otherwise challenging experimental characteristics, such as a lack of suitable cooling sidebands, and is a promising tool for precision spectroscopy of highly-charged ions. For example, in Ref. [35], sympathetic cooling via a Be^{+} ion (the logic ion) cools the Ar^{13+} spectroscopy ion, initially produced at megakelvin temperatures, down to the 10 mK level. Quantum logic spectroscopy was then used to measure the $^{40}\mathrm{Ar}^{13+}\ {}^{2}P_{1/2} \rightarrow^{2} P_{3/2}$ fine-structure transition with a fractional uncertainty of 3×10^{-15} — the first such measurement to be carried out in a highly-charged ion. The experiment also measured the excited-state lifetime of the target transition, as well as its Landé g-factor, all of which are necessary precursors to the development of highly-charged ion clocks.

These relatively recent developments are applicable to any species of HCI and, for the first time ever, open up the possibility of precision highly-charged ion spectroscopy.

The combination of experimental developments has allowed the experimental realisation of high-precision HCI spectroscopy and opened-up applications such as ultra stable highly-charged ion clocks [47, 48], precision searches for physics beyond the standard model [29, 23]. However, many of the systems of interest have have little to no available experimental data, so high-precision atomic structure calculations are critically important to guide experiments and applications.

Assessing the difficulty of atomic structure calculations for highly-charged ions is not completely straightforward. Almost by definition, the electronic structure of HCIs has large contributions from the nuclear potential, which can speed up the convergence of electronic structure calculations. However, in order to get electronic transitions in the optical regime, highly-charged ionic systems of interest, such as highly-charged tin ions [49, 50] and Ir^{17+} [29, 48], have complicated, open-shell electronic structures and large cancellations of electronic binding energies. This large cancellation is computationally challenging regardless of choice of approximation due to the introduction of large numerical instabilities and uncertainties. For example, in the case of Ir^{17+} it is extremely difficult to predict even the ground-state configuration, which has so-far prevented its experimental realisation as a precise clock to test for new physics.

1.3 Superheavy elements

As with highly-charged ions, experimental spectroscopy of superheavy elements has historically been out of reach due to the inherent difficulties posed by their physical properties. There is no standard definition of what counts as a "superheavy" element, but in this thesis I will use the term to refer to elements with a nuclear charge $Z > 100$, also known as the transfermium elements. Elements with $Z < 100$ are typically produced by neutron capture and successive beta-decay in lighter elements, which have the dual advantages of large reaction cross-sections and fluxes, as well as relatively simple experimental apparatuses [51]. The decay-rates due to alpha-decay and spontaneous fission drastically increase for elements beyond fermium ($Z = 100$).

The "liquid-drop" model of the nucleus (a macroscopic model which treats nucleons like molecules in a liquid drop, where the stability is solely determined by the balance of the coulomb and strong nuclear forces) predicts that nuclei cannot exist for $Z \gtrapprox 100$ [52], so the existence of even short-lived superheavy nuclei is solely due to the effects of nuclear shell structure and correlation (the so-called "microscopic picture"). This instability prevents the use of neutron capture for superheavy element production [51, 52] — the two-step process of neutron-capture to increase the nuclear mass followed by beta decay to increase the atomic number Z will not have time to complete before the nucleus decays by spontaneous fission.

As outlined in Refs. [53, 52], the only viable method for producing transfermium elements is through the complete fusion of lighter nuclei, which allows for the inclusion of large numbers of neutrons to stabilise the nuclei. Cold fusion is typically achieved via bombarding a thin foil (less than 1 mg/cm^2 [51]) composed of a heavy element (the "target") with heavy ions from an accelerator (the "projectile") [54]. Greater stability of the product nuclei can be achieved by using projectile and/or target nuclei with magic numbers of protons and neutrons, where stability from nuclear shell effects is maximised [54]; in particular, the "doubly magic" ^{48}Ca nuclei has been used to produce superheavy elements up to $Z = 118$ (Oganesson) with high enough production yields to allow the reaction pathways to be fully characterised [55, 53]. A comprehensive treatment of the nuclear physics involved in the synthesis of superheavy elements is presented in Refs. [53, 52, 56, 57].

Even given the recent advances outlines above, precision spectroscopy of superheavy elements remains challenging due to their low production yields and short half-lives. Because of this, the most successful experiments in optical spectroscopy in superheavy elements, as well as measurements of their ionisation potentials, have required "atom-at-a-time" measurements. Early work in lawrencium (Lr, Z = 103) showed that surface-ionisation measurements could be carried out at an atom-at-a-time scale, producing accurate measurements of the first ionisation potential of a superheavy element [58] — the measured value of $4.96^{-0.07}_{+0.08}$eV agrees closely with the value of 4.963(15)eV predicted by Fock-Space coupled cluster calculations. Further work in ref [59] showed that the same experimental approach could also be applied to fermium, mendelevium and nobelium ($100 \leq Z \leq 102$), also achieving close agreement with relativistic atomic calculations.

Additionally, high-resolution optical spectroscopy of superheavy atoms has been demonstrated in atom-at-a-time experiments with nobelium (Z = 102): the ionisation potential [60], ground-state $^1S_0 \to ^1P_1$ transition [61], and measurement of the hyperfine splitting and isotope shift to probe nuclear size and shape [62]. These experimental measurements were carried out via the SHIP heavy ion separator located at GSI Darmstadt — the nobelium atoms and ions were embedded and allowed to thermalise in an argon buffer gas to reduce thermal and collision broadening to $\sim$ 4 GHz level [62, 63]. Further enhancements to this apparatus aim to further reduce the collisional broadening by embedding the target atoms/ions in a supersonic gas jet, which then reduces the temperature and pressure compared to a static buffer gas cell. It is estimated that the next generation of heavy ion separators using supersonic buffer gas jets will reduce the line width due to gas collisions to $\sim$ 100 Mhz [63].

Atomic structure calculations of superheavy elements

The aforementioned short lifetimes and low production-rates of superheavy elements, combined with the gas-jet method of thermalisation and buffering, means that each atom is only available for spectroscopic analysis for short periods of time. Consequently, spectroscopic searches must be guided by accurate calculations so to precisely target the transition lines of interest.

This requirement imposes two restrictions on atomic structure codes if they are to be useful in the spectroscopy of superheavy atoms and ions:

1. Accuracy: often transitions of interest are quite narrow and, by the nature of superheavy elements, strongly dependent on relativistic effects. As a rule of thumb, the accuracy of atomic structure calculations should be $\sim 1\%$ (which often works out to ~ 500 cm^{-1}) to provide a useful guide to experiments.

2. Fast turnaround: while not as crucially important as accuracy, the challenges of designing and running experiments with superheavy elements means that accurate theoretical predictions are required well before the experiment can actually commence. Additionally, accurate spectra of several ionisation levels of the same element are often required to guide successive experiments. Consequently, we want electronic structure calculations to be as fast as possible to maximise scientific throughput, so it is especially important to take advantage of modern high-performance computing architectures.

Computationally, the largest challenges stem from the strong relativistic effects in superheavy elements, which must be "baked in" at every level of the calculation (as opposed to being treated perturbatively or semi-empirically after the fact). Additionally, QED corrections are much larger in superheavy atoms than for more conventional systems, and can reach several hundred cm^{-1}; comparable to the level of accuracy required to guide experimental design [64, 65]. These corrections must therefore be included in atomic structure calculations as well if they are to produce useful results.

Currently, QED corrections cannot be treated rigorously in many-electron atoms, with the current state of the art theoretical methods being limited to approximate methods such as the radiative potential method of Ginges and Flambaum [66, 67, 68], the model Lamb shift operator formalism of Shabaev *et al* [69] and the self-energy screening approximation of Lowe *et al* [70]. These methods are only implemented in a small number of atomic structure packages, limiting their reach — an open-source, user-friendly atomic structure code such as AMBiT can thus greatly improve access to atomic structure calculations of superheavy elements (along with other systems with strong QED contributions).

Finally, many superheavy elements display strong correlation effects and open-shell electronic configurations; both of which are very computationally intensive to treat accurately [64, 65]. Consequently, modern atomic structure theory and HPC techniques are increasingly important in maximising scientific throughput when studying superheavy elements.

1.4 Differential measurements and precision tests for physics beyond the standard model

In addition to the aforementioned experimental motivation for high-precision calculations of highly-charged ions and superheavy elements, there are also myriad applications for "conventional" atomic systems (i.e. non-highly-charged, pre-actinide atoms and ions) in the search for new physics beyond the standard model. These tests rely on the extremely high-precision achievable by atomic, molecular and optical (AMO) experiments as an alternative to the high-energy frontier explored by, for example, TeV-scale particle accelerators and cosmological processes. The lack of unambiguous detection of new physics at the energy scales probed by the Large Hadron Collider implies that, if they exists at all, deviations from the standard model most likely occur at energy scales much larger than is feasible to reach with particle accelerators in the near future [71]. AMO experiments can provide an alternative since, despite the high energy scale of potential physics beyond the standard model, the extremely high precision reachable by AMO experiments make them potentially sensitive to signals of physics beyond the standard model, while also being much smaller and cheaper than particle accelerators [71, 72].

Except for the simplest systems, atomic and molecular structure calculations cannot reach the relative precision of 10^{-18} required to directly simulate the effects of physics beyond the standard model, due to the fast-growing computational complexity of accurately modelling many-electron effects (for example, the size of a full CI calculation grows exponentially with the number of atomic electrons). However, this limitation can be worked around by exploiting *differential measurements.* Rather than directly comparing theory and experiment, we can instead calculate how observable quantities *change* due to the effects of new physics, which are less sensitive to numerical limitations and uncertainties [73] .

There are a huge number of such differential measurements, including the search for variations in the fundamental constants (such as the fine-structure constant α [74] and the electron-proton mass ratio μ [75, 76]), atomic parity violation (APV) [77, 78], tests for symmetry-violating neutron and electron electric dipole moments (EDMs) [77, 79, 80], precision tests of QED via electronic or hyperfine structure spectroscopy [27, 81], and isotope shift spectroscopy [73, 82, 83, 84].

In the rest of this section, I will review a few interesting examples of atomic experiments and tests for physics beyond the standard model, as well as some of the challenges motivating the need for high-performance atomic structure codes. A comprehensive study of molecular, nuclear and optical tests for new physics can be found in Refs. [72, 27, 36].

1.4.1 Variation of the fundamental constants

As previously mentioned, highly-charged ion clocks can be used as sensitive test for variation in the fundamental physical constants, such as spatial or temporal variation in the fine structure constant α [29, 23]. These variations in the fundamental constants are also potentially observable by more conventional optical atomic clocks, which, despite their potentially lower sensitivity and stability compared to HCI clocks, have the advantage of more mature technology and theoretical predictions.

In order to detect variation in α, at least two clock transitions with different sensitivities to α-variation must be used [29]. Mathematically, we can quantify α-sensitivity by calculating the change the energy of a clock transition which would be induced by a change in the value of α

[72, 85]. For a sufficiently small change $\alpha_0 \to \alpha = \alpha_0 + \delta\alpha$ (where α_0 is the current value of the fine-structure constant), the energy dependence of a level can be parameterised by the coefficient q:

$$E(\alpha) = E_0 + q\left[\left(\frac{\alpha}{\alpha_0}\right)^2 - 1\right] \tag{1.1}$$

where E_0 is the measured energy of the transition. Since q is essentially the derivative of the rate of change of E with respect to α, it can be determined by calculating the upper and lower energy of the clock transition for a number of different values of α, and then calculating the numerical derivative for each level [29, 85]. This will provide two values q_f and q_i for the corresponding energy levels i and f, so the q-factor for the transition is $q = q_f - q_i$ [29]. The α-dependence can also be represented in a unitless form $K \equiv 2q/E_0$ [29, 72], so larger values of K indicate greater sensitivity to variation in α and can be compared like-for-like between different clock candidates (in practice both q and K are used throughout the literature).

Highly-sensitive transitions have three major characteristics [30]:

1. Small transition energy E_0, which will amplify the relative sensitivity K. There is a tradeoff between sensitivity and increased experimental and theoretical complexity, however: the transition must be accessible to modern lasers; and nearly degenerate transitions are more challenging to accurately calculate, since they result from large numerical cancellations in many-body systems.

2. The transition must be highly stable to ensure the high precision required to detect small changes in α. The current best upper-bound on α-variation is of the order of 10^{-17} [86], and ionic clocks such as Yb^+ can now reach fractional accuracies of $\sim 10^{-18}$ [8]; a level where even the smallest perturbations such as gravitational effects and thermal blackbody radiation shifts become significant [72]. Consequently, we need clock transitions with very narrow linewidths and long lifetimes — the clock transition should be a "forbidden" transition (e.g. the $E3$ electric octopole transition in Yb^+), the upper level should have few decay channels, and the atom or ion itself should be insensitive to environmental perturbations (e.g. a large electric polarisability will reduce the transition's sensitivity to blackbody radiation).

3. High ionic charge, increasing the size of relativistic corrections to the energy levels of the system, which scales as $\sim \alpha^2$ causing a large sensitivity q. Generally speaking, we can expect large ionic charges to increase the spacing between levels, thus making the clock transition inaccessible to modern lasers [72]. However, the fortuitous level-crossings due to the convergence between many-body effects and nuclear potential in the HCI systems explored in Refs. [30, 29, 48] can serve as near-optical clock transitions while maintaining a high ionic charge.

All of the points above generally pose theoretical challenges when calculating the spectrum of the clock system. The most sensitive candidates, such as Yb$^+$ [72], Ir^{17+} [48], and microwave transitions between nearly-degenerate states in Dy [87], have open f-shell configurations with strong configuration mixing, which require huge calculations to even identify the ground- and clock-state, let alone accurately resolve the entire spectrum (to identify, for example, side-channel transitions for cooling and trapping). The large cancellations necessary to achieve transitions with small energies also cause slow numerical convergence, which further increases the size of the atomic calculations needed to resolve them. Finally, the lifetimes and sensitivity to external perturbations depend on the electromagnetic multipole transition matrix elements, which require accurately calculated atomic wavefunctions. In particular, electric polarisabilities, required to quantify the sensitivity to blackbody radiation shifts, need accurate wavefunctions for large numbers of bound and continuum states, even if those states are not directly involved in the proposed experiment (this is also true for experiments using more conventional atomic clocks with simpler electronic structure such as Sr lattice clocks). This is currently difficult for open-shell systems and highly-charged ions, further highlighting the need for greater utilisation of computational resources.

1.4.2 Searches for dark matter and exotic particles

Precision atomic experiments can also be used to search for dark matter and other exotic particles through their coupling to standard model fields. In this section, I will discuss two types of atomic experiment which can potentially detect new particles: precision clocks and measurement of couplings between electrons and nucleons due to interactions with new particles.

The experiments in section 1.4.1 are focused on detecting steady spatial or temporal drift in the fundamental constants, but are in principle also sensitive to transient effects due to coupling with dark matter. For example, transient interactions with large, localised topological defects (TD) can appear as short time-scale variations in the fine-structure constant via frequency shifts in atomic clocks [88, 89]. These frequency shifts would be observable in networks of atomic clocks, such as Sr lattice clocks or the network of Cs and Rb clocks in the Global Positioning System, which have been used in conjunction with astrophysical observations to place limits on the density and coupling strength of TD dark matter [88, 89].

Similarly, precision atomic clocks are also sensitive to possible interactions with light dark matter (LDM), which at high enough densities would behave more like a diffuse, oscillating field than as particles [87, 90]. One class of LDM arises from scalar fields such as the dilaton, which arises in certain classes of high-energy theories [72]. The dilaton field is expected to interact with atomic clocks as an oscillation in the apparent value of the fine structure constant over long time-scales [72], and would therefore be detectable by the experiments outlined in section 1.4.1. Refs. [87, 91] used two near-degenerate transitions in neutral Dy, the splitting of which is strongly dependent on α, to set bounds on the electron-dilaton coupling. These bounds could in turn be improved by more sensitive experiments, such as those in Yb$^+$ [8] or highly-charged ion clocks [23].

An alternative approach in the search for new particles is to probe electron-nucleon couplings beyond those predicted by the standard model through isotope-shift spectroscopy. Many-electron atomic structure calculations do not even get close to the 10^{-18} level of precision reached by the most stable atomic clocks; one way around this problem is to use isotope-shift experiments, which are much less sensitive to experimental uncertainties [73]. Different isotopes of the same element will have qualitatively similar spectra, but slightly different energies for their electronic transitions, due to the difference between the nuclear mass (the so-called mass shift) and finite nuclear size (the field shift) for the isotopes [92].

The chief advantage of this approach is that it relies on measurements which are less sensitive to systematic uncertainties in electronic structure calculations due to inexact treatments of electron correlation [73]. The relatively small changes in atomic spectra due to nuclear effects (e.g. changes in nuclear size and mass in different isotopes of the same element) are difficult to accurately calculate from first principles in many-electron atoms, but the *relationship* between the mass- and field-shifts (due to the nuclear mass and nuclear size, respectively) is less dependent on numerical uncertainties [82, 73, 92].

Isotope shift measurements can place tight bounds on new interactions between electrons and nucleons, such as those arising from vector-boson LDM fields using conceptually simple atomic spectroscopy repeated across multiple isotopes of the same element. Analysis of existing data carried out in Ref. [73] already provided complementary constraints on potential new bosons, and isotope shift measurements of Yb$^+$ carried out in Ref. [83], while not sensitive enough to detect new particles, provide a strong foundation for future searches. Furthermore, isotope shift measurements can serve as sensitive probes of higher-order nuclear properties within the standard model which would be difficult to directly measure, such as nuclear charge distribution [93].

Although isotope shift measurements are not directly relevant to the work in this thesis, atomic structure calculations are still necessary to extract the mass- and field-shift coefficients for the target isotopes, which in the case of Yb$^+$ is extremely computationally challenging. Consequently, this class of precision tests for new physics will still benefit from improved atomic structure code.

1.5 Aims and outlines of this thesis

This thesis is a mixture of physics and software engineering. A large chunk of the work in this thesis was in modernising and optimising the AMBiT atomic structure code, primarily through overhauling the parallel elements of the code to take better advantage of modern supercomputing architecture. This work greatly improved the performance of our atomic structure calculations, as well as effectively removing limitations on memory usage (by effectively taking advantage of the huge memory pools in distributed systems); optimisations large enough that none of the calculations in this thesis would have been feasible without these software optimisations.

The physics component, on the other hand, consists of relativistic calculations of the electronic structure of several highly-charged ions and superheavy elements. The bulk of this work was directly motivated by experimental needs, and undertaken alongside regular communication with experimentalist colleagues.

The chapters in this thesis are organised as follows:

Chapter 2 gives an overview of computational methods for atomic structure calculations and describes, in detail, the mathematical basis of the CI+MBPT method used throughout this thesis. Chapter 3 discusses high-performance computing, the performance issues with the old version of AMBiT which necessitated my OpenMP performance optimisations, and provides an in-depth discussion of these optimisations and their effects on AMBiT's overall performance. The remaining chapters present the results of atomic structure calculations using the new version of AMBiT, firstly to benchmark the code's accuracy in chapter 4, and then to support ongoing experimental studies in chapters 5, 6 and 7. Chapter 5 discusses calculations of the optical spectra of the tin highly-charged ions Sn^{7+} – Sn^{10+}, and chapters 6 and 7 present calculations of the spectra of the superheavy element lawrencium and its ion (Lr and Lr^{+}) and compares results with independent calculations using other atomic structure codes.

These calculations show that AMBiT and the CI+MBPT method achieve top-of-class accuracy for complex atomic systems, and serve as a powerful tool for cutting-edge atomic structure calculations. Finally, the conclusion in chapter 8 discusses future improvements for both AMBiT and the CI+MBPT method and how they may be used to guide future advances in atomic physics.

Chapter Two

Modelling atomic structure: CI+MBPT

2.1 Overview of methods for atomic structure

Disclaimer: The contents of this chapter are adapted from the previously published work in Ref. [94]. I was lead author on this paper and this review is my original work.

2.1.1 Other codes

The combination of configuration interaction with many-body perturbation-theory (CI+MBPT) is one of the workhorses of *ab initio* high-precision atomic structure calculations and is known to provide highly accurate results for many-electron atoms [95, 96, 49, 97]. Initially developed by Dzuba, Flambaum and Kozlov [95] to calculate atomic energy spectra, CI+MBPT is also capable of providing other atomic properties such as electronic transitions data (e.g. electric- and magnetic-multipole transition matrix elements) [98].

The accuracy of CI+MBPT calculations lies in the ability to partition atomic structure calculations and take advantage of the complementary strengths of CI and MBPT. The electrons are partitioned into "core" electrons, which are treated as inert, and valence electrons, which display the dynamics of interest. CI provides a highly accurate treatment of valence-valence electron interactions (see, for example [99]), while MBPT treats the core-valence correlations in a computationally efficient manner [95, 96]. This combination of techniques allows for the treatment of three [95], four [100, 101] and five [49, 102] valence electrons, with agreements with experimental spectra and transition matrix elements to better than a few percent.

Despite these advantages, standard implementations, such as the `CI+MBPT` code of Kozlov *et al* [97], still require infeasibly large computational resources for $\gtrsim 4$ valence electrons [96]. Additionally, three-body MBPT corrections must be included to accurately treat systems with many valence electrons in this formalism [95, 103, 102]. The number of these three-body MBPT diagrams grows extremely rapidly with MBPT basis size and can significantly increase computation time [100, 102], and are not included in Kozlov *et al*'s `CI+MBPT` [97], for example.

Alternatively, pure configuration interaction (without the introduction of MBPT) is a frequently used approach in atomic structure software packages and is able to treat simple, few-electron systems. CI-only software packages include the `RELCI` package by S. Fritzsche *et al* [99] (also part of the `GRASP` package), and the `PATOM` code of Bromley and co-workers [104]. Of these packages, `PATOM` is only capable of calculating the spectra of one- and two-valence electron atoms, while `RELCI` only supports *restricted* calculations for more than two valence electrons [99].

Another common method is multiconfigurational Dirac-Fock (MCDF), as implemented in the `GRASP` series of relativistic atomic structure packages [105, 106], the MCDF code of Desclaux [107] and the Jena Atomic Calculator (`JAC`) [108]. The Flexible Atomic Code (`FAC`) software [109] has a similar rationale, calculating atomic spectra using Dirac-Fock and CI optimised for each configurations, which can also be combined with MBPT corrections [110, 3]. `FAC` accurately calculates continuum processes and is commonly used for modelling plasma dynamics such as electron-impact ionization, excitation, and recombination processes.

Although pure CI and MCDF can provide a high degree of accuracy for atoms with few valence electrons atoms, the number of many-electron configurations increases exponentially with the number of electrons [111], making a direct solution with these techniques computationally infeasible for systems with $\gtrsim 4$ valence electrons [96, 97]. The size of CI+MBPT calculations is typically reduced by partitioning the electrons into core electrons, which are typically treated with a self-consistent field method such as Dirac-Fock, and "valence" electrons which are directly included in the CI or MCDF procedure. While the partitioning reduces the computational bottleneck from the total number of electrons to the number of valence electrons, it is still infeasible to treat core-valence correlations without the introduction of MBPT [96, 95]. Calculations by Kozlov, et al [97] suggest that CI+MBPT provides approximately an order-of-magnitude greater accuracy than pure CI due to the ability to treat core-valence interactions without dramatically increasing the size of the CI problem.

Our atomic structure code, AMBiT, has several advantages over existing CI+MBPT software. First, we implement three-body MBPT corrections, providing a significant increase in accuracy for many-electron systems. Second, we can undertake the CI+MBPT procedure in either the electron-only (as in most CI/CI+MBPT packages), or in the particle-hole formalism as presented in [96]. This allows us to form open-shell (i.e. partially-filled) configurations from either all electrons or the corresponding number of positively charged "holes" in an otherwise filled shell. For example, the electron-only configuration $5\text{d}^9 6\text{s}$ with the Fermi level below the 5d shell is equivalent to the particle-hole configuration $5\text{d}^{-1} 6\text{s}$ where the 5d shell is included in the core. The electron-only and particle-hole approaches are formally equivalent at the CI level, but the particle-hole formalism can provide significantly more accurate MBPT corrections by reducing the contribution of so-called "subtraction diagrams" [100, 112], which can seriously degrade the accuracy of open-shell calculations.

In addition to the standard core-valence MBPT corrections, we can also use MBPT to treat high-lying valence-valence correlations [96]. Valence-valence MBPT can significantly reduce the size of the CI problem, especially for systems where the core and valence electrons are separated by a relatively large energy gap (such as highly-charged ions). Additionally, we have developed a new addition to the standard CI+MBPT procedure, which we refer to as emu CI. This technique allows for a significant reduction in the computational size of a CI+MBPT problem without significant reductions in accuracy [113] and is discussed further in section 2.2.4 and chapter 3.

2.2 CI+MBPT Theory

Our CI+MBPT calculations consist of three conceptual stages. First, we treat the relatively inert core electrons using the Dirac-Fock (DF) method, the relativistic generalisation of the Hartree-Fock self-consistent field method. Second, we treat the valence electrons and holes with configuration interaction using a set of many-electron basis-functions built from single-electron B-spline orbitals. Finally, the effects of core-valence correlations and virtual core-excitations are included via many-body perturbation theory by modifying the matrix elements used in the CI problem.

The full details of this process have been extensively discussed elsewhere (see, for example [95, 96, 100, 49, 114, 113]), so we will only present details relevant to our implementation of CI+MBPT. All calculations and formulae in this section are presented in atomic units ($\hbar = e = m_e = 1$).

2.2.1 Single particle basis

First, we perform a Dirac-Fock calculation, typically in the V^N, V^{N-1} or V^{N-M} approximations, where N is the number of electrons and M is the number of valence electrons. That is to say, we include either all N electrons of the ion, or some subset of them in the Dirac-Fock procedure. The choice of potential has significant consequences for the convergence of the calculation, with a V^N potential producing "spectroscopic" core orbitals, which are optimised for a particular configuration, while the V^{N-M} potential (i.e. only including a subset of electrons in DF) potentially provides a better basis for the convergence of MBPT, by avoiding large contributions from so-called "subtraction diagrams" [112], which are discussed further below.

In either choice of potential, the resulting one-electron Dirac-Fock operator is (see, e.g. [114]):

$$h_{\mathrm{DF}} = c\,\boldsymbol{\alpha}\cdot\mathbf{p} + (\beta - 1)c^2 - \frac{Z}{r} + V^{N_{\mathrm{DF}}} \tag{2.1}$$

where $\boldsymbol{\alpha}$ and β are Dirac matrices. We write the wavefunction as

$$\psi(\mathbf{r}) = \frac{1}{r}\begin{pmatrix} f_{n\kappa}(r)\ \Omega_{\kappa,m}(\hat{r}) \\ i g_{n\kappa}(r)\ \Omega_{-\kappa,m}(\hat{r}) \end{pmatrix} \tag{2.2}$$

where $\kappa = (-1)^{j+l+1/2}(j+1/2)$ and $\Omega_{\kappa,m}$ are the usual spherical spinors. The eigenvalue equation $h_{\mathrm{DF}}\,\psi_i = \epsilon_i\psi_i$ can be written in the form of coupled ODEs:

$$\frac{df_i}{dr} = -\tfrac{\kappa}{r} f_i(r) + \tfrac{1}{c}\left(\epsilon_i + \tfrac{Z}{r} - V^{N_{\mathrm{DF}}} + 2c^2\right) g_i(r) \tag{2.3}$$

$$\frac{dg_i}{dr} = -\tfrac{1}{c}\left(\epsilon_i + \tfrac{Z}{r} - V^{N_{\mathrm{DF}}}\right) f_i(r) + \tfrac{\kappa}{r} g_i(r) \tag{2.4}$$

for each orbital ψ_i. Numerical methods for solving these equations may be found in [114]. The resulting wavefunctions are used for orbitals in the Dirac-Fock core, as well as valence orbitals included in $V^{N_{\mathrm{DF}}}$ potential, while other basis orbitals are constructed using B-splines as described below.

At this stage, we may modify the Dirac-Fock operator to incorporate the effects of finite nuclear size [103], nuclear mass-shift [100, 102], and the Breit interaction (including both Gaunt and retardation terms in the frequency-independent limit) [114]:

$$B_{ij} = -\frac{1}{2r_{ij}} \left(\boldsymbol{\alpha}_i \cdot \boldsymbol{\alpha}_j + \left(\boldsymbol{\alpha}_i \cdot \boldsymbol{r}_{ij} \right) \left(\boldsymbol{\alpha}_j \cdot \boldsymbol{r}_{ij} \right) / r_{ij}^2 \right) \tag{2.5}$$

We may also include Lamb shift corrections, which are calculated via the radiative potential method originally developed by Flambaum and Ginges [66]. The more recent formulation employed in this code includes the self-energy [67] and vacuum polarisation [68] corrections, collectively referred to in this paper as the QED corrections. These corrections are propagated through the rest of the calculation via modification of the MBPT and radial CI (Slater) integrals, or the residual two-electron Coulomb operator in the case of the Breit interaction.

We construct the remaining valence and virtual orbitals (pseudostates) as a linear combination of B-spline basis functions. B-splines are piecewise polynomials defined on a lattice of “knot-points”, which accurately approximate the single-particle wavefunctions using fewer functions and with fewer cancellation issues than some analytic approximations such as Slater- or Gaussian-type spinors [115]. B-splines have proven to be an effective choice of basis function for the calculation of atomic properties which are sensitive to both relativistic corrections and nuclear effects, including parity violation [116, 117, 118], polarisability and tune-out wavelengths [119, 120], and relativistic effects in heavy ions [121].

We expand the large and small radial components, $f_{n\kappa}(r)$ and $g_{n\kappa}(r)$ of the virtual orbitals as linear combinations of two sets of B-splines $\{l_i\}$ and $\{s_i\}$:

$$\begin{aligned} f_{n\kappa}(r) &= \sum_i p_i l_i(r) \\ g_{n\kappa}(r) &= \sum_i p_i s_i(r) \end{aligned} \tag{2.6}$$

Each component of the wavefunction has the same set of expansion coefficients, which are obtained variationally by solving the generalised eigenvalue problem [115, 116]:

$$A\mathbf{p} = \varepsilon S\mathbf{p} \tag{2.7}$$

where $A_{ij} = \langle i| h_{\mathrm{DF}} |j\rangle$ is the matrix representation of the Dirac-Fock operator in the B-spline basis, $S_{ij} = \langle i|j\rangle$ is the overlap matrix, $|i\rangle = \begin{pmatrix} l_i(r) \\ s_i(r) \end{pmatrix}$ are the B-spline basis functions, and ε is the single-particle energy of the virtual orbital.

There is some freedom when choosing the exact form of the sets $\{l\}$ and $\{s\}$, as well as the boundary conditions of the resulting B-spline basis functions. By default, AMBiT uses the Dual Kinetic-Balance (DKB) splines developed in Ref. [116] due to their superior accuracy for atomic properties at small distances from the nucleus and robustness against the effects of so-called “spurious states”. However, alternative approaches can also be used, which in the terminology of

AMBiT are called “Notre Dame” [115] and “Vanderbilt” [122] splines. The resulting basis set is then ordered by energy and used for both CI and MBPT procedures.

2.2.2 CI Basis

The one-particle basis functions are then used to construct a set of many-particle “projections”, which are (properly anti-symmetrised) configurations with definite angular momentum projection m_j for every electron or hole. AMBiT’s representation of projections is tightly coupled to their corresponding relativistic configurations — combinations of single-particle orbitals which also specify total angular momentum for each constituent orbital (e.g. $1\mathrm{s}^2\ 2\mathrm{s}_{1/2}\ 2\mathrm{p}_{3/2}$).

projections corresponding to a particular relativistic configuration are represented as arrays of $2m_j$ angular momentum projections (which are always integer valued) of the orbitals in the configuration.

All projections corresponding to a relativistic configuration in the CI-space form a basis from which to build many particle Configuration State Functions (CSFs) $\{|I\rangle\}$. CSFs are eigenfunctions of the $\hat{J}^2$ and $\hat{J}_z$ operators and are formed as a linear combination of projections:

$$|I\rangle = \sum_n c_n \left|\mathrm{proj}_n\right\rangle . \tag{2.8}$$

We create $|I\rangle$ within the stretched state $M_J = J$, therefore only projections with $\sum m_j = J$ are included in the expansion. The coefficients $\{c_n\}$ are determined variationally by diagonalising the $\hat{J}^2$ operator in the projection basis:

$$\hat{J}^2 \left|I\right\rangle = J(J+1)\left|I\right\rangle \tag{2.9}$$

The projections and CSF expansion coefficients for each configuration with the same number of electrons, J^π symmetry and projection M_J are stored to disk. This allows the initial cost of diagonalising the $\hat{J}^2$ operator to be amortised across all calculations with the same angular components, dramatically reducing the overall computational cost.

The CI-space of CSFs are formed by taking electron excitations from a set of “leading configurations” (reference configurations) that are also used to determine which three-body MBPT diagrams to include. Leading configurations can contain any number of valence electrons and holes – the only limit is the computational resources available when running the software. We construct CI configurations and CSFs by taking excitations from these leading configurations up to some maximum principal quantum number, n, and orbital angular momentum, l. These limits are represented in the software and throughout the rest of this paper using a shorthand representation of called a “basis string”. For example, we can specify orbitals with $0 \leq l \leq 3$ and $n \leq 10$ for each partial wave via the string 10spdf, or s- and p-orbitals with $n \leq 10$ and d-orbitals with $n \leq 7$ with the string 10sp7d.

AMBiT can construct CSFs to use in CI calculations with an arbitrary number of electron excitations, but finite computational resources usually limit the CI basis to single- (often referred to as CIS) or single- and double-excitations (CISD). However, important triple or quadruple excitations should also be included. The CSFs can also include valence-holes in otherwise filled shells, which can lie between the Fermi level of the system and some minimum n and l, the latter of which is referred to as the "frozen core".

The atomic level wavefunctions Ψ for a given total angular momentum and parity J^π are then constructed as a linear combination of CSFs $|I\rangle$:

$$\Psi = \sum_{I \in P} C_I \, |I\rangle \tag{2.10}$$

where P is the subspace of configurations included in CI and the coefficients C_I are obtained from the matrix eigenvalue problem of the CI Hamiltonian:

$$\sum_J H_{IJ} C_J = E C_I \tag{2.11}$$

In the particle-hole formalism, the CI Hamiltonian is [96]:

$$\hat{H} = \sum_i c\boldsymbol{\alpha} \cdot \mathbf{p} + (\beta - 1)c^2 + \frac{Ze_i}{r_i} - e_i V^{N_{\mathrm{core}}} + \sum_{i<j} \frac{e_i e_j}{|\mathbf{r}_i - \mathbf{r}_j|} \tag{2.12}$$

where $e_i = -1$ for valence electron states and $+1$ for holes. It is important to note that the one-body potential $V^{N_{\mathrm{core}}}$ in the CI Hamiltonian only includes contributions from the core electrons, since valence-valence correlations are included directly via the two-body Coulomb operator.

The Hamiltonian matrix elements depend not only on the angular (CSF) components, but also on the two-body radial Slater-Coulomb integrals:

$R(ab, cd)$ (where a, b, c, d are single-particle orbitals derived from either the DF or B-Spline parts of the calculation), which is the multipole expansion of the corresponding Coulomb integral.

Each Slater integral $R^k(ab, cd)$ (where k is the order of the multipole moment) appears in multiple Hamiltonian matrix elements, so to remove redundant computational work, we calculate all valid Slater integrals once at the start of the calculation and re-use their values throughout the rest of the CI+MBPT process.

As a further optimisation, we first calculate the Hartree-Y operator (sometimes called the Hartree screening operator) for the orbitals b and d, which is defined as:

$$Y^k_{bd}(r) = \int \frac{r_<^k}{r_>^{k+1}} \psi_c^\dagger(r') \psi_d(r') dr' \cdot \xi(k + J_b + J_d) \cdot \Delta(k, J_b, J_d) \tag{2.13}$$

Where ξ and Δ ensure the correct angular momentum conditions $|J_b - J_d| \leq k \leq J_b + J_d$ and $\xi(n) = (1 + (-1)^n)/2$. The Hartree Y operator is defined such that:

$$R^k(ab, cd) = \langle a | \, Y^k_{bd} \, | c \rangle \tag{2.14}$$

So by calculating the Hartree-Y operator and then iterating over pairs of orbitals we only calculate components which are shared between integrals once, further reducing redundant work.

2.2.3 CI+MBPT

The size of the CI matrix grows extremely rapidly as additional orbitals are included, so it is computationally infeasible to include core-valence correlations or core excitations directly in the CI procedure. Instead, we treat these interactions as a small perturbation and include their contributions to the total energy in second-order MBPT by modifying the CI matrix elements [95, 100]. The final matrix eigenvalue problem is then:

$$\sum_{J \in P} \left(H_{IJ} + \sum_{M \in Q} \frac{\langle I | \, \hat{H} \, | M \rangle \langle M | \, \hat{H} \, | J \rangle}{E - E_M} \right) C_J = E C_I \tag{2.15}$$

where the subspace Q includes all orbitals not in the CI procedure and is complementary to P. For computational efficiency, we do not directly modify the CI matrix elements as suggested by equation (2.15), due to the large number of configurations in Q. Instead, we modify the radial integrals via the Slater-Condon rules for calculating matrix elements (see ref [123] for a formal discussion of this process).

The subspace Q is formally infinite, but we only include corrections from a finite, truncated subset of orbitals in MBPT. In AMBiT, the P and Q subspaces are divided at the level of orbitals. The CI space P includes any configuration with all single-particle orbitals drawn from the valence basis or holes outside the frozen core; these in turn are defined by two basis strings. Similarly, the MBPT space Q is bounded by a separate orbital limit, also expressed as a basis string (e.g. 30spdfg for all orbitals with $n \leq 30$ and $l \leq 4$).

Consequently, an excitation is only included in MBPT if at least one of the orbitals involved are not included in the CI space P. This prevents double-counting of configurations and ensures that diagrams are independent of the number of electron- and hole-excitations in CI.

The number of terms in the MBPT corrections grows rapidly but the diagrammatic technique [95] greatly simplifies the calculation of these terms. In this formalism, each contribution to the MBPT expansion is represented by a Goldstone diagram, with the number of external lines corresponding to the number of valence electrons included in the interaction [100]. Figure 2.1 shows an example of a one-body MBPT diagram describing the self-energy correction arising from core-valence interactions (left) and a subtraction diagram involving an interaction with an external field (right) [100]. These subtraction diagrams enter the MBPT expansions with a negative sign and increase in magnitude with $V^{N_{\text{core}}} - V^{N_{\text{DF}}}$ [112]. Explicit formulas for one- and two-body core-valence diagrams implemented in AMBiT can be found in Ref. [100].

Subtraction diagrams are partially cancelled out by some two- and three-body diagrams in the

MBPT expansion [102], necessitating the systematic inclusion of all one-, two- and three-body MBPT diagrams in the CI+MBPT procedure to ensure accurate spectra. Even given this cancellation, subtraction diagrams can grow large enough to be non-perturbative in open-shell systems, which can significantly impact the accuracy of the resulting spectra [102]. Consequently, there is a tradeoff between the more "spectroscopic" orbitals produced by calculations in a V^N potential and potentially large subtraction diagrams when $V^{N_{\mathrm{DF}}} \neq V^{N_{\mathrm{core}}}$; the optimal choice will depend on the specifics of the target system. This is not a hard constraint though – the formulation of MBPT used in AMBiT can, in principle, treat systems with any number of valence electrons or holes subject to available computational resources.

An additional complexity is that the energy denominators of (2.15) include the energy eigenvalue E in the Brillouin-Wigner perturbation theory formalism. In practice we approximate the energy denominators using the valence orbital energies [100]. See Refs. [95, 123] for further discussion of this subtle point. Finally, the diagrammatic technique allows us to eliminate terms corresponding to unlinked diagrams, as they represent valence electron interactions not included in MBPT [95]. This greatly reduces the computational expense of including MBPT corrections.

In addition to the standard core-valence MBPT, AMBiT can also include MBPT corrections to valence-valence integrals, as introduced in [96]. In this approach, the MBPT expansion in equation (2.15) includes additional diagrams representing correlations between highly-excited valence states (i.e. outside the upper-bounds of the CI-space P), as shown in figure 2.2.

Valence-valence MBPT is significantly computationally cheaper than including the orbitals directly in the full CI subspace. However, as with all MBPT techniques, care must be taken to ensure that there are no non-perturbative diagrams in the MBPT expansion. Specifically, including orbitals which are far from spectroscopic (such as orbitals with high orbital angular momentum) or orbitals which are close in energy to those in P can produce non-perturbative diagrams with small energy-denominators. These diagrams can significantly reduce the accuracy of CI+MBPT calculations and are easy to inadvertently include in the MBPT expansion, so this approach should only be used with a carefully constructed MBPT subspace Q.

The energies for each calculated eigenstate are presented by AMBiT in ascending order of energy, grouped by total angular momentum J and parity π. The solutions also contain the CI expansion coefficients and Landé g-factors, to aid in identifying levels. It is important to note that the absolute energies do not represent ionisation energies or any other physically meaningful quantity: the atom is effectively in a box due to the finite extent of the basis orbitals. Rather only the *relative* energies of the eigenstates (and the resulting atomic spectrum) represent physically meaningful quantities.

Finally, the resulting CI+MBPT wavefunctions are used to calculate transition matrix elements for electric and magnetic multipole operators, which we refer to as "external field" operators, as well as hyperfine dipole and quadrupole operators. AMBiT can calculate either reduced matrix elements T:

$$T_{if} = \langle f|\,\hat{O}\,|i\rangle \tag{2.16}$$

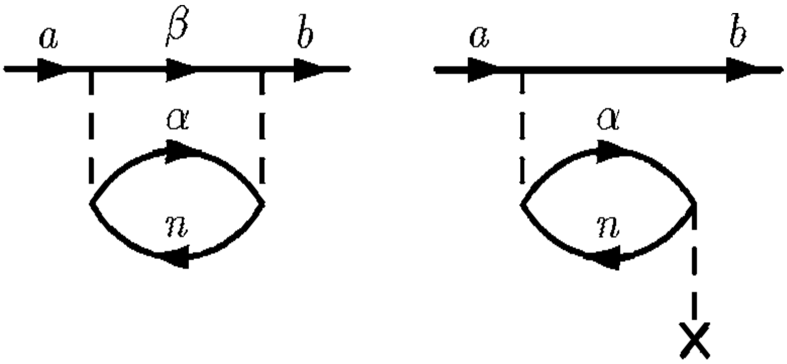


Figure 2.1: Some Goldstone diagrams representing a one-body core-valence correlation (left) and one-body subtraction diagram (right). Lines running left to right represent electrons, while lines running right to left are holes. $|a\rangle$ and $|b\rangle$ are valence orbitals, $|\alpha\rangle$ and $|\beta\rangle$ are virtual, and $|n\rangle$ is a hole in the core [100].

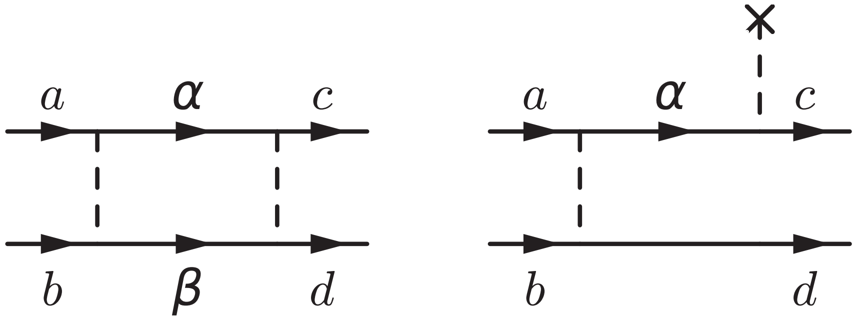


Figure 2.2: Two-body valence-valence diagram (left) and valence-valence subtraction diagram (right). External lines a, b, c, d correspond to valence or hole orbitals in the CI subspace P, while the interior lines α and β are virtual electron orbitals, at least one of which must not be valence for the intermediate state to be in Q [96].

for some operator $\hat{O}$, initial state $|i\rangle$ and final state $|f\rangle$, or line-strengths:

$$S_{if} = |T_{if}|^2 \tag{2.17}$$

Transition matrix element calculations may additionally include frequency-dependent random-phase approximation (RPA) corrections [124, 125, 126]; detailed equations implemented in AMBiT are presented in Ref. [127].

2.2.4 Emu CI

The CI method outlined in section 2.2 relies on constructing and diagonalising the Hamiltonian matrix over a set of many-electron CSFs. The number of CSFs, and consequently the size of the CI matrix, scales exponentially with the number of electrons included in the CI problem subspace, resulting in prohibitively large CI matrices for systems with more than three valence electrons. Additionally, CI is slowly converging even for relatively simple systems with few valence electrons [128], making saturation in open-shell systems infeasible with current computational methods.

AMBiT implements an approach that greatly reduces the computational difficulties associated with CI, which we refer to as emu CI [113] (as the structure of the resulting CI matrix resembles an emu's footprint). This approach is especially well-suited to the common case where we are only interested in calculating a few of the lowest-lying energy levels and allows for the use of significantly larger CI basis sizes than would otherwise be possible. A schematic representation of this approach is shown in figure 2.3.

Emu CI relies on the fact that the CI expansion (2.10) is dominated by relatively few large contributions from off-diagonal CI matrix elements. Other CSFs contribute less strongly, and so interactions between these may be neglected. The shaded region of the matrix shown in figure 2.3 is formed as the Cartesian product of the N_{CI} CSFs in the CI-space P, which we refer to as the "large side" of the matrix, and a smaller set of $N_{\text{small}} < N_{\text{CI}}$ CSFs, which we refer to as the "small side" of the matrix. The small side contains a subset of the N_{CI} CSFs that make the largest contribution to the CI expansion. Perturbation theory estimates performed in [129] show that the remaining off-diagonal terms, shown in white, produce a negligible contribution for the small number of states of interest, and can be set to zero without serious loss of accuracy.

The small-side CSFs are formed by allowing electron and/or hole-excitations from a set of leading configurations (which is not necessarily the same as used when forming the main CI-space) up to some maximum principal quantum number n and orbital angular momentum l. This limit is specified using the same format of basis string used when forming the main CI-space. Finally, the small-side can include an arbitrary number of electron- and hole-excitations (which also do not have to be the same as in the main CI-space). For example, we may want to incorporate contributions from triple-excitation configurations by include single- and double-excitations in N_{small}, and single- double- and triple-excitations in N_{CI}, since including triples in the full CI procedure would require prohibitively large matrix sizes. The specifics need to be considered on a case-by-case basis and are provided in full for each calculation in this thesis.

We can then construct the CI matrix such that the significant off-diagonal terms are grouped together in a block, producing the structure shown in figure 2.3. These elements are further sorted such that the configurations with the largest number of corresponding projections appear first in the matrix to provide better performance when constructing and diagonalising the matrix in parallel.

Emu CI is conceptually similar to the approach described in Ref. [129]. The key difference is that the `CIPT` approach in Ref. [129] (now referred to as *Fast CI* or `FCI` [130]) treats contributions from matrix elements with one high-lying configuration via perturbation theory, rather than constructing the full matrix. A similar method of limiting the CI matrix to only the largest contributions has also been implemented in the `GRASP` software [106], but uses preliminary, first-order calculations to limit the final CI matrix to only include contributions from CSFs with approximate mixing coefficients larger than a specified cutoff threshold. Emu CI differs from the existing approaches by the structure of the matrix and the procedure used to construct it.

The dramatically reduced number of non-zero elements in the emu CI matrix compared to standard CI significantly reduces the computational resources required to obtain accurate atomic spectra. Recent calculations of the spectra of neutral tantalum and dubnium [113] shows that emu CI is capable of producing highly accurate atomic spectra for five-electron systems despite its relatively small resource usage. Spectra from these calculations were within $\sim 1-10\%$ of experimental values, and convergence tests showed that the CI expansion was close to saturation. Similarly, applying this technique to the Cr^+ calculations presented in this paper reduces the number of non-zero elements in the effective CI by a large factor. Even larger reductions in matrix size were used in [113]. The full matrices would be far to large to store in memory, even on modern high-performance computing clusters. Emu CI, combined with modern parallel programming techniques enables the use of extremely large CI bases, even for challenging open-shell systems with strong correlations.

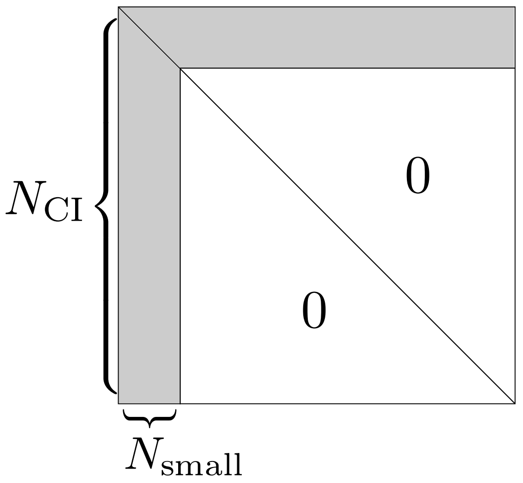


Figure 2.3: Structure of the CI matrix under the emu CI approximation. The $N_{\mathrm{CI}} \times N_{\mathrm{small}}$ nonzero off-diagonal terms are shaded in light grey, diagonal terms are represented by the black line, terms neglected in the approximation are shown in white.

Chapter Three

High-performance computing — OpenMP

3.1 Glossary

Definitions of terms used in this chapter:

1. *Cluster*: a computer system consisting of multiple smaller, tightly interlinked computers, which are capable coordinating to carry out large, computationally intensive calculations in parallel. Often referred to as a supercomputer

2. *Core*: The basic unit of computational resources on clusters. Performs arithmetic, logic and generally executes the instructions in a program. Sometimes colloquially referred to as a processor or CPU.

3. *Socket*: a grouping of cores, generally sharing some kind of fast cache memory and/or connection (*bus*) to main memory. Sometimes referred to as a processor, CPU or package.

4. *Node*: a singular, self-contained computer, many instances of which are interlinked to form a cluster. Nodes contain one or more sockets, can only communicate with one another via message-passing and all sockets and cores in a node draw from a common pool of resources (memory, disk access, networking, etc.).

5. *MPI*: Short for *Message Passing Interface*. A software library facilitating parallelism across multiple, potentially heterogeneous computational resources such as nodes on a cluster or cores on a socket.

6. *OpenMP*: software library facilitating shared-memory parallelism (e.g. within a single server or computer) by using threads.

7. *Process*: basic unit of parallelism employed by MPI (and operating systems, more generally) Processes operate as independent, persistent instances of a given program and do not share resources.

8. *Thread*: basic unit of parallelism employed by OpenMP. By default, OpenMP threads share resources such as memory address space.

9. *Mapping*: scheme by which MPI processes are distributed among computational resources. Processes can be mapped to either cores, sockets or nodes.

10. *Binding*: used interchangeably with mapping.

11. *Affinity*: scheme by which OpenMP threads are assigned to cores on a node.

12. *Lock*: programming construct which synchronises access to some shared resource, such as a shared variable or file. Locks are used in OpenMP to ensure threads do not attempt to modify a resource at the same time, which would cause undefined behaviour and bugs. OpenMP has explicit locks, as well as the closely related *critical section*, which ensures a section of code is only executed by one thread at a time.

13. *Mutex*: short for *MUTual EXclusion*, synonymous with lock.

14. *Memory address space*: the range of discrete memory addresses available to a process or thread. Modern operating systems use *virtual memory*, which abstracts away the hardware implementation details to give processes the illusion of a large, contiguous memory address space.

15. *API*: short for *Application Programming Interface*. The set of conventions, abstractions and methods specifying how multiple software applications may interact. This is commonly used to describe the set of function calls and data types exposed by a library to allow its use by external software programs.

3.2 Introduction

CI+MBPT is a standard workhorse of computational atomic structure physics, and, as discussed in chapter 2 is an effective technique for complex, open-shell atomic systems. However, the computational resources required by CI+MBPT grow extremely rapidly with the number of atomic electrons, and especially valence electrons. For example, the number of possible configuration state functions (CSFs) which can be formed from valence electrons grows exponentially with the number of valence electrons and holes in the atomic system. This is especially problematic in open-shell systems such as certain superheavy elements, as well as those used in precision tests for new physics. Accurate calculations often require terabytes of memory and hundreds of CPU hours to complete.

For example, in chapter 4 we present large-scale calculations of the spectra of Ta and Db presented in ref [113]. These calculations were the first to saturate the CI basis for Ta ($Z = 73$) and Db ($Z = 105$), both of which have five valence electrons, and agree with experimental values at the level of 1-10%. In order to reach saturation of the CI basis for Ta, we required matrices up to 952112×20462 elements (for $J = 5/2$, $\Pi = \text{odd}$) and has a peak memory usage of ~ 240GB. This calculation is too large to fit within the memory of a single compute node on *Gadi* (the National Computational Infrastructure's high-performance computing cluster), and thus requires parallelism that scales across multiple nodes. The calculations for Db required similar matrix sizes.

In order to produce a clean set of comparisons between methods of parallelism, I'll be using a smaller version of the Db calculation in [113] which I have tailored to fit within a single node on *Gadi*. The input file for this calculation is shown in listing 1. The calculation produces the five lowest energy solutions with odd parity and $J = 5/2$, using a CI basis which includes single- and double-electron excitations up to 19spdf on the large side of the emu CI matrix, and single- and double-excitations up to 9spd on the small side. It also calculates the Landeé g-factors and M1 transition matrix elements for all combination of atomic eigenstates.

The largest CI matrix from this example calculation has 731868×11783 elements, and (under the current OpenMP+MPI branch of AMBiT) the calculation requires 125GB of memory — enough to fit on a single *Gadi* compute node. Furthermore, the Db calculation requires a walltime of approximately 30min when distributed across 48 cores with OpenMP, but has a CPU time of 25hr 04min, where CPU time is defined as $T_{\text{CPU}} = T_{\text{wall}} \times$ Avg.CPU utilisation.

The CPU time can serve as a rough estimate of the length of time required to run the code on one core (in serial); so a serial calculation of Db of the same size as in Ref. [113] would require more than a day to complete. Run-times on the order of days are clearly undesirable when attempting to prototype calculations or produce rapid turnaround of results, such as when collaborating on currently active experiments.

Modern high-performance computing clusters can easily meet memory and compute requirements demanded by open-shell calculations, but codes must be efficiently parallelised to make use of them. The exponential increase in CPU clock frequencies which defined much of computing in the 20th century (encapsulated by the well-known *Moore's Law*) stalled over a decade ago [131] and almost all recent increases in computational power have derived from increasingly sophisticated (and complicated) parallelism, both in individual CPUs and in computing systems more broadly.

For example, a single core on an HPC cluster such as the National Computational Infrastructure's (NCI's) *Gadi* [132] is not much more powerful than a consumer-grade PC; rather, the power of HPC systems lies in the ability to link hundreds or thousands of nodes (each with multiple cores) together and distribute the computational workload by running software in parallel. Such large-scale parallelism is difficult to achieve, and requires considerable domain-specific programming knowledge to fully exploit modern HPC resources. Consequently, accurate simulation of the properties of next-generation experiments and applications in atomic physics may not be feasible without effective use of parallelism.

AMBiT originally employed some degree of parallelism via the *Message Passing Interface* (MPI) library and runtime for multiprocessing. While MPI provides excellent inter-node scaling, it was not sufficient to fully utilise the large numbers of compute cores present within modern compute nodes. With that in mind, this thesis presents a description of the process and rationale for overhauling AMBiT to employ a hybrid method of parallelism, consisting of MPI for inter-node parallelism and OpenMP for intra-node parallelism.

In this chapter, I will first outline why the MPI-only method of parallelism was insufficient to meet the computational needs of this thesis. Then, I will detail the changes and work required to implement the hybrid MPI+OpenMP parallelism, and finish with a quantitative investigation

```
ID=DbI
Z = 105

NuclearRadius=9.5

[Lattice]
NumPoints=1000.
StartPoint=1.e-6
EndPoint=50.

[HF]
N = 104
Configuration='1s2 2s2 2p6 3s2 3p6 3d10 4s2 4p6 4d10 4f14 5s2 5p6 5d10 6s2 6p6 5f14: 6d3 7s1'

[Basis]
--bspline-basis
ValenceBasis=19spdf

[CI]
LeadingConfigurations='6d3 7s2, 6d4 7s1, 6d3 7s1 7p1, 6d5, 6d2 7s2 7p1'
ElectronExcitations=2
OddParityTwoJ='5'
NumSolutions=5

[CI/SmallSide]
LeadingConfigurations='6d3 7s2, 6d4 7s1, 6d3 7s1 7p1, 6d5, 6d2 7s2 7p1'
ElectronExcitations='1, 9spd, 2, 9spd'

[MBPT]
Basis=30spdfgh

[Transitions]
M1/AllBelow=0.0
```

Listing 1: AMBiT input file used to generate Db emu CI spectra while profiling with Arm Forge. The calculation produces the five lowest energy solutions with odd parity and $J = 5/2$, using a CI basis which includes single- and double-electron excitations up to 19spdf on the large side of the emu CI matrix, and single- and double-excitations up to 9spd on the small side.

into the performance and parallel scaling of the new version of AMBiT.

3.3 Why do we need OpenMP?

At the start of this project, the AMBiT software had been under continuous development by Berengut *et al* at UNSW since 2005, and was a relatively mature codebase with some degree of parallelism via MPI, which is a library that allows HPC code (C, C++ or Fortran) to be run as a series of independent, loosely linked processes, which can be spread across multiple nodes of a cluster. MPI implements the *message passing* model of parallelism: the program is parallelised by distributing the workload between processes which duplicate the program's executable code, and explicitly communicate by sending each other chunks of data to collect and synchronise results between processes. This communication takes place via network interconnect for processes located on separate nodes, or through manipulating some shared memory location for processes on the same node. However, even though multiple MPI processes may run on a single node/machine, processes still have independent *virtual* address spaces and views of the system's resources (as opposed to the multithreading model, where all threads have access to the same address space by default). This means that even though processes on a single node share the same *physical* memory, processes running on the same node must still explicitly pass around data to communicate, and shared data must be duplicated between all processes which require it.

The duplication of shared resources between processes presents serious limitations on the extent of parallelism available to AMBiT when running with a pure-MPI approach. In a typical CI+MBPT calculation, the CI matrix is distributed between MPI processes, with each process only generating and operating on a few “chunks” of the matrix. However, the orbitals, one- and two-body MBPT integrals, Slater (Coulomb) integrals, and angular momentum data are required when generating the CI matrix elements, so must be duplicated in full between MPI processes. This imposes significant memory overhead when running with a large numbers of MPI processes on a single node.

These memory characteristics are problematic, as it sometimes necessitates deliberately underutilising a node by requesting less than its full complement of cores in order to avoid exhausting the available memory. This results in wasted computational resources and sub-optimal run-times for calculations.

The duplication of memory is especially pronounced on newer compute nodes, such as the nodes on *Gadi*, which contain 48 cores and 190GB of memory per node. As a specific example, the input file in listing 1 for CI+MBPT calculations of Db results in a CI matrix with $952112 \times 11783 = 1.1218735696 \times 10^{10}$ matrix elements, which, assuming 8 bytes per element for double-precision floating point numbers, should take at least:

$$952112 \times 11783 \text{ doubles} \times 8 \text{ bytes/double} \approx 90GB \tag{3.1}$$

The memory estimate in equation 3.1 is a lower-bound to the true memory usage, since it only considers the memory consumed by the CI matrix and ignores the orbitals, two-body Slater and MBPT integrals and angular momentum data (among other objects). It does, however, demonstrate that the CI matrix (which is approximately evenly distributed between MPI processes) is not sufficient by itself to exhaust the 190GB of memory on the *Gadi* nodes. This means that any memory exhaustion issues must come from the objects which are duplicated between MPI processes.

This calculation barely fits in memory when using 4 MPI processes, but increasing the number of processes beyond this exhausts the 190GB memory on the node. In fact going to the full 48 MPI processes blows up the memory usage to $\sim 900GB$, meaning that it is only possible fully utilise the CPUs on special-purpose, large memory nodes, which are usually expensive and limited in number.

OpenMP can provide a solution to this problem by allowing all threads within a process to share the same memory address space. OpenMP is an API specification for shared-memory parallelism, which consists of a set of library functions and compiler directives [133]. OpenMP is essentially a high-level abstraction on top of standard, operating-system threading tools (such as UNIX `pthreads`) to distribute parallel workloads between multiple threads of execution within a node. In the hybrid MPI+OpenMP mode I have implemented in AMBiT, each MPI process spawns multiple threads, with all threads belonging to the same process sharing a virtual memory address space, and thus variables and data. This shared-memory behaviour is key to the performance gains I have realised by adding OpenMP to AMBiT. By scheduling fewer MPI processes per node and using multithreading to parallelise across a node's cores, the shared components of the calculation only need to be stored once per node (or per socket, depending on the binding), thus significantly reducing the wasted memory.

The dubnium calculation runs to completion when using all 48 cores with OpenMP, with a maximum memory usage of 125GB. Clearly, overhauling the AMBiT to use OpenMP has dramatically increased the size of atomic calculations we can carry out, as the OpenMP version is able to employ 12× as many cores, with a corresponding reduction in walltime. Furthermore, the memory usage is almost independent of the number of OpenMP threads used by the calculation. This allows for AMBiT calculations to efficiently utilise all the compute power available on a compute node, while still maintaining the multi-node scalability afforded by MPI.

Good parallel algorithm design requires a detailed knowledge of the hardware on which the code will be running. Before I describe the process of parallelising and optimising AMBiT, I will present a brief overview of the hardware architecture of modern HPC systems, as relates to the work in this thesis.

3.4 A primer on computer architecture

Rather than consisting of a single, very fast computer, modern supercomputers are almost always organised as clusters — huge arrays of interlinked, often off-the-shelf servers which derive their computing power from the ability to execute massively parallel workloads by splitting the workload up between computational resources and having those resources communicate the results to one another. This section will focus on the high-level, conceptual details of cluster architecture; more technical points can be found in the footnotes, although the low-level details will differ between individual clusters. A comprehensive treatment of computer architectures can be found in Ref. [131].

Conceptually, the computing resources of a cluster are organised into a hierarchy of cores, sockets and nodes, in order of increasingly coarse grain. A schematic example of this architecture is shown in 3.1 — a simple model of a cluster which has two nodes, with two sockets per node and four CPUs per socket. I will use this simplified cluster to illustrate different methods of parallelism throughout the rest of this section.

At the lowest level are *cores*, or *processors* (although somewhat confusingly, this latter terminology can also be used to refer to sockets), which for the purposes of this thesis are essentially chips that can do basic arithmetic, manage branching logic, and store and retrieve values in memory. Cores are the basic unit of computing resources and are what actually executes computer programs.

Processor architectures have become increasingly complex as a result of efforts to increase the performance of serial software in spite of stalled growth in clock-speed and transistor density. In order to preserve the illusion of purely serial execution (an assumption which is baked into the design of programming languages such as C), compute cores dedicate considerable resources to exploiting so-called *instruction-level parallelism* (ILP). ILP relies on the fact that independent CPU instructions (i.e. instructions which use different parts of a core and do not depend on each other's result) can often be re-ordered or issued in parallel to increase instruction throughput without changing the semantics of the program. Aggressively exploiting ILP can result in significant speedups (even for inherently sequential code), but requires special effort from application developers to maximise performance, as the re-ordering of instructions happens at the hardware level and is often deliberately opaque (again, to preserve the illusion that programs are executed purely sequentially).

Another area of hidden complexity is memory accesses. Accessing the main memory of a computer (often generically referred to as RAM) is extremely slow compared to CPU arithmetic and branching operations, so modern computer architectures attempt to reduce the average latency by caching memory accesses. Memory caching relies on the principle of *locality of reference*, which takes two chief forms [1]:

1. Temporal locality: if a memory address is accessed, it is likely to be accessed again in the near future (for example, the counter variable in a loop or a frequently read array). To optimise for temporal locality, the contents of memory locations are stored in small, fast caches close to CPU cores when accessed, thus speeding up future accesses.

2. Spatial locality: if a memory address is accessed, the addresses "nearby" are also likely to be required in the near future (for example, when sequentially accessing elements in an

[1] These details are specific to x86 architecture processors, but almost all mainstream processors have roughly the same caching behaviour

array). To optimise for spatial locality, memory addresses are read in contiguous blocks or "cache lines" and stored in CPU caches to speed up future accesses to adjacent memory locations.

Consequently, a core does not directly access main memory when it needs some data — instead, it checks the caches (most x86 CPUs have a hierarchy of three caches of increasing size and latency) and only accesses main memory if the desired data is not in any of the caches.

The assumption of locality of reference is a heuristic which does not apply to all codes (or even all sections of a code), but is common enough that CPU optimisations which assume locality provide large enough performance gains on average to have become nearly ubiquitous. Consequently, design patterns which are memory cache-friendly is extremely important for performance-sensitive software.

The next logical level up from cores are *sockets*. Sockets are groups of cores which are "close" to one another in the machine[2], so communication within a socket is much faster than communication between sockets. The number of cores on a socket varies considerably between clusters and even individual nodes on the same cluster. NCI's *Gadi* cluster, which I used for the bulk of this thesis' work, has 24 cores per socket on its standard nodes, each of which has two sockets for a total of 48 cores per node.

Finally, one or more sockets are grouped together into *nodes*: self-contained servers containing one or more sockets, with all cores on a node sharing a common pool of memory (RAM), and so can communicate with one another relatively quickly. Separate nodes in a cluster do not share resources and can only communicate and coordinate over a local area network, which is significantly slower than communication between cores on the same node.

The different characteristics of inter- vs intra-node communication necessitates the use of different techniques in parallelising code. AMBiT employs two software libraries to achieve this: *OpenMP* for distributing the workload among cores within a node (which may or may not cross socket boundaries) and *MPI* for distributing the workload between sockets and nodes. These two methods of parallelism can be used in isolation or combined for a hybrid approach as I have done in this thesis.

3.5 Profiling

Generally speaking, parallelism and performance optimisation are difficult to reason about, and there is always the danger of spending a lot of time on optimsations which end up not helping the overall performance very much. I have only parallelised the sections of code which take up the largest percentage of the execution time; referred to as *hotspots*. The easiest way to measure and visualise code hotspots is through profiling the program's execution, which are presented here for two "typical" atomic calculations. While there is no obvious definition of "typical", I have chosen to use two calculations which stress different subsystems of AMBiT.

The first representative calculation is a Db CI+MBPT calculation with approximately compa-

[2] They share a memory bus and usually at least one cache

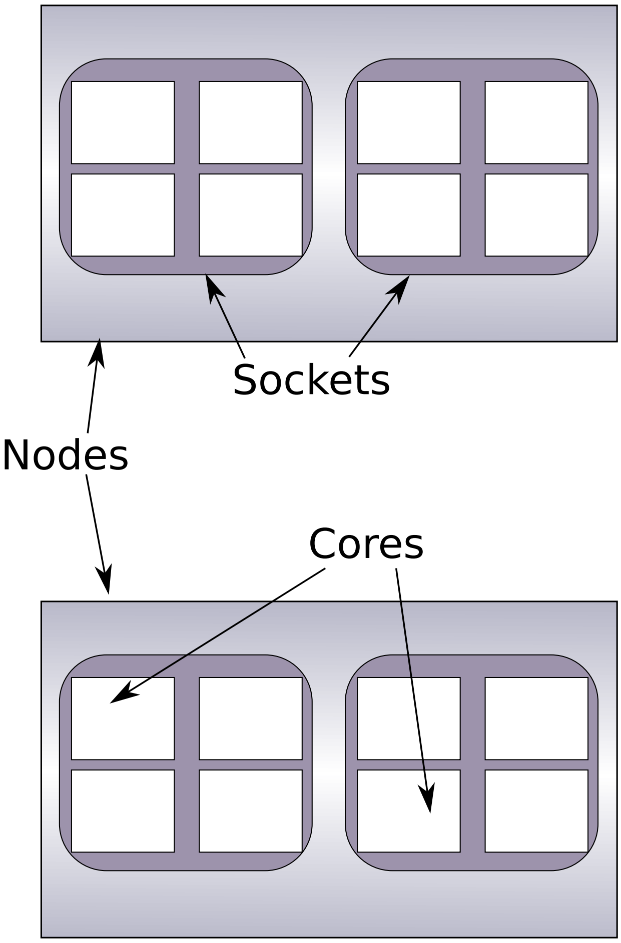


Figure 3.1: Schematic of a sample cluster architecture. This sample cluster consists of two nodes (represented by the outermost rectangles), each of which has two sockets (rounded rectangles) with four cores (small, white rectangles) per socket for a total of eight cores per node.

rable CI and MBPT bases, since open-shell systems tend to generate large numbers of CSFs (and thus CI matrices) with relatively few Slater integrals. Second, I have also profiled a CI-only calculation of singly-ionised lawrencium (Lr$^+$, $Z = 103$), which has two valence electrons above closed shells; the input file for this calculation is shown in listing 2. Systems with few valence electrons tend to generate a lot of MBPT integrals but relatively small CI-matrices, so Lr$^+$ provides complementary performance tests to the CI-heavy Db calculations.

```
ID = LrII

Z = 103

NuclearRadius=7.2
NuclearThickness=2.3

-m
-s12

[Lattice]
NumPoints = 1000
StartPoint = 1.e-6
EndPoint = 50.

[HF]
N = 101
Configuration = '1s2 2s2 2p6 3s2 3p6 3d10 4s2 4p6 4d10 4f14 5s2 5p6
    5d10 6s2 6p6 5f14: 7s1'

// Basis properties
[Basis]
--bspline-basis
ValenceBasis = 10spdf

[CI]
LeadingConfigurations = '7s2'
ElectronExcitations = 2
HoleExcitations = 0
NumSolutions = 10
EvenParityTwoJ = '0'

[MBPT]
--use-valence
Basis=12spdf
```

Listing 2: AMBiT input file used to generate the callgrind profiles for Lr$^+$, with CI+MBPT.

Figures 3.2 and 3.3 show the call-graph profiles for small Db and Lr$^+$ calculations, respectively. These calculations were generated by the *callgrind* tool, as part of the *Valgrind* suite of profiling and debugging tools [3]. These profiles are "call-graph" style, meaning that they represent the

[3] Valgrind imposes a very large performance overhead, so it was only feasible to use callgrind for very small

program hierarchy as a tree, with each branch of the tree representing a call-stack — each subroutine is called by its parent node and so on up to the root node. The topmost three nodes in both graphs are the subroutines required to start-up the Valgrind runtime and AMBiT's main loop, and can be ignored for the purposes of this analysis. Only subroutines taking a significant fraction ($> 1\%$) of the program's run time are shown, and multiple instances of the same call path are aggregated to show the time spent across all instances of that path, given as a percentage of the total program runtime.

Starting with the Db profile in figure 3.2, we can see that the bulk of the runtime is spent in three parts of the code:

1. `HamiltonianMatrix::GenerateMatrix` (60.79%) – Generating the Hamiltonian (CI) matrix from pre-calculated radial integrals and angular data. This subroutine does not include the actual diagonalisation of the matrix.
2. `TransitionCalculator::CalculateTransitions` (22.31%) – Calculating the transition matrix elements, in this case the electric multipole elements. This is implemented via the overload of ManyBodyOperator::GetMatrixElement() taking two LevelVector objects (an array of eigenstates calculated by diagonalising the CI matrix element).
3. `ManyBodyOperator<...<SzOperator> > GetMatrixElement()` (14.79%) – Calculation of the S_z matrix elements as part of calculating g-factors.
4. `SlaterIntegrals<std::map<> >::GetTwoElectronIntegral()` (11.02%) – A non-trivial portion of the time spent generating the Hamiltonian matrix comes from simply accessing two-body Slater (Coulomb) integrals, which are pre-calculated before the matrix is generated and are stored in C++'s `std::map` data structure. `std::map` is an associative array which is implemented as a binary search tree, which is optimised for a wide variety of applications and usage patterns rather than any one specific usage. Even though this is not strictly related to OpenMP, we can still gain large performance improvements by using data structures which are better suited to our usage patterns.

The g-factors and transition matrix elements are implemented as special cases of a more general class called `ManyBodyOperator` via C++ templates — a design pattern which allows us a great deal of flexibility when implementing quantum mechanical operators (e.g. the spin projection S_z or electric dipole operator). Operators can use the same set of optimised subroutines, while specialised code for each operator is automatically generated by the compiler. Consequently, parallelising the *GetMatrixElement* functions in `ManyBodyOperator` will both optimise the hotspots (g-factors and transition matrix elements) in the current version of AMBiT and provide some future-proofing with regards to adding operators for new physical quantities.

Next, the Lr^+ profile in figure 3.3 shows one major bottleneck which is not present in the Db benchmarks, namely the two-body MBPT integrals. This makes sense, since Lr^+ is a two-electron system and thus will have many more two-electron integrals than CSFs for most possible ways of carrying out the calculations. In this specific case, we have 137531 two-body MBPT integrals

calculations

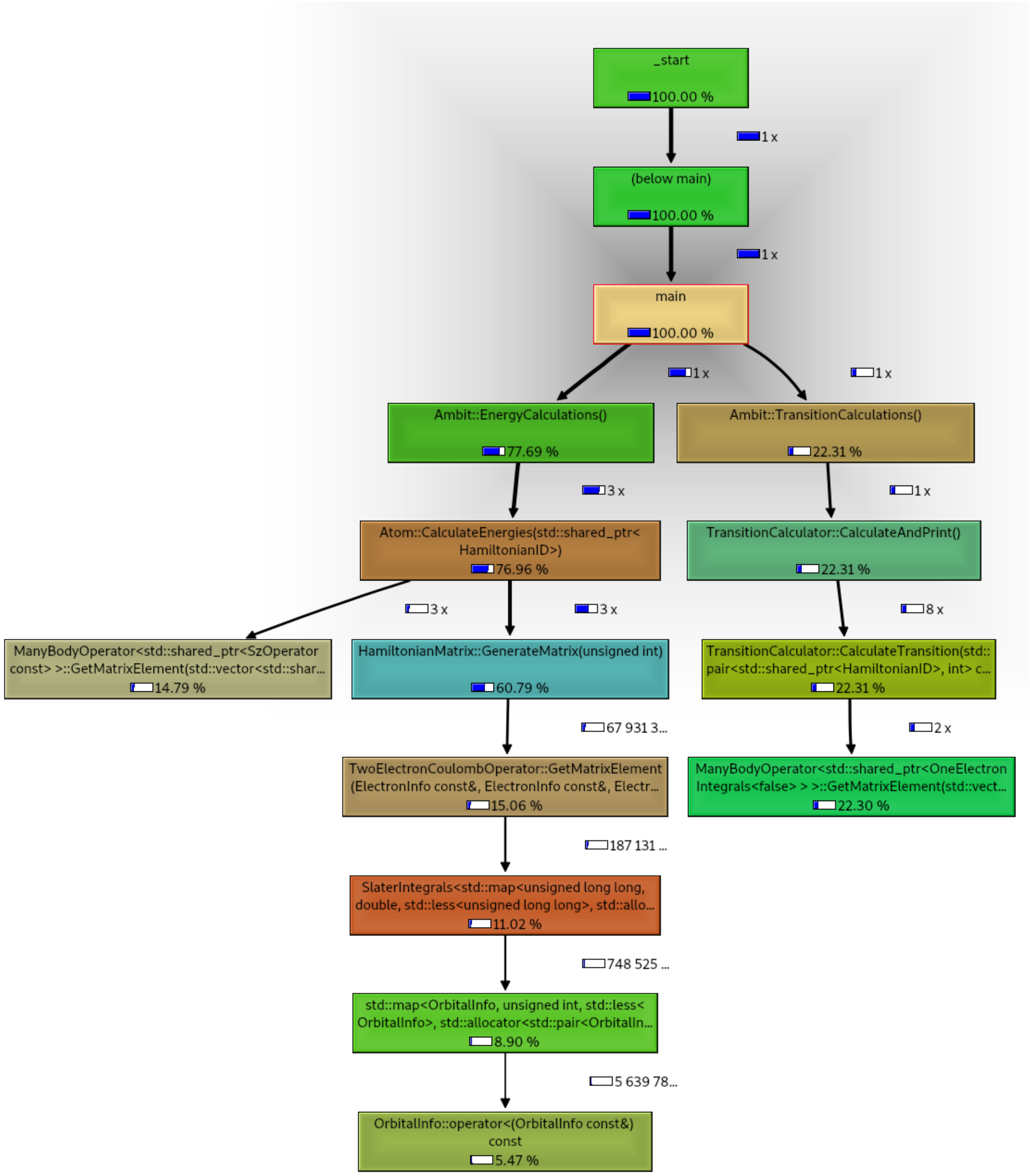
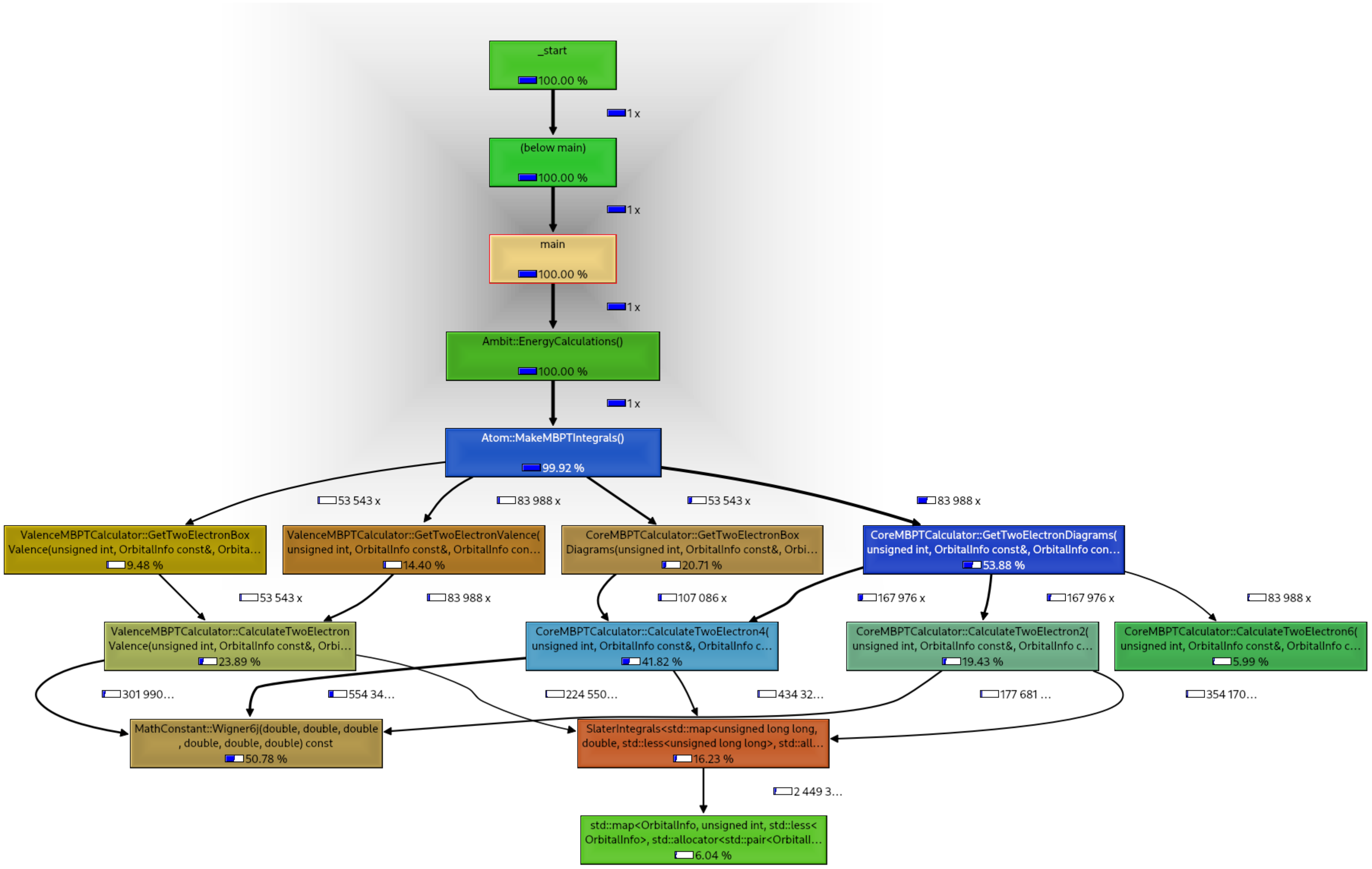

_start
100.00 %
1 x
(below main)
100.00 %
1 x
main
100.00 %
1 x
1 x
Ambit::EnergyCalculations()
77.69 %
Ambit::TransitionCalculations()
22.31 %
3 x
1 x
Atom::CalculateEnergies(std::shared_ptr< HamiltonianID>)
76.96 %
TransitionCalculator::CalculateAndPrint()
22.31 %
3 x
3 x
8 x
ManyBodyOperator<std::shared_ptr<SzOperator const> >::GetMatrixElement(std::vector<std::shar...
14.79 %
HamiltonianMatrix::GenerateMatrix(unsigned int)
60.79 %
TransitionCalculator::CalculateTransition(std:: pair<std::shared_ptr<HamiltonianID>, int> c...
22.31 %
67 931 3...
2 x
TwoElectronCoulombOperator::GetMatrixElement (ElectronInfo const&, ElectronInfo const&, Electr...
15.06 %
ManyBodyOperator<std::shared_ptr<OneElectron Integrals<false> > >::GetMatrixElement(std::vect...
22.30 %
187 131 ...
SlaterIntegrals<std::map<unsigned long long, double, std::less<unsigned long long>, std::allo...
11.02 %
748 525 ...
std::map<OrbitalInfo, unsigned int, std::less< OrbitalInfo>, std::allocator<std::pair<OrbitalIn...
8.90 %
5 639 78...
OrbitalInfo::operator<(OrbitalInfo const&) const
5.47 %

Figure 3.2: Call graph with relative execution time for Db CI+MBPT MPI-only calculation, with CI basis of 10spdf and MBPT basis of 12spdf


_start
100.00 %
1 x
(below main)
100.00 %
1 x
main
100.00 %
1 x
Ambit::EnergyCalculations()
100.00 %
1 x
Atom::MakeMBPTIntegrals()
99.92 %
53 543 x
83 988 x
53 543 x
83 988 x
ValenceMBPTCalculator::GetTwoElectronBox
Valence(unsigned int, OrbitalInfo const&, Orbita…
9.48 %
ValenceMBPTCalculator::GetTwoElectronValence(
unsigned int, OrbitalInfo const&, OrbitalInfo con…
14.40 %
CoreMBPTCalculator::GetTwoElectronBox
Diagrams(unsigned int, OrbitalInfo const&, Orbi…
20.71 %
CoreMBPTCalculator::GetTwoElectronDiagrams(
unsigned int, OrbitalInfo const&, OrbitalInfo con…
53.88 %
53 543 x
83 988 x
107 086 x
167 976 x
167 976 x
83 988 x
ValenceMBPTCalculator::CalculateTwoElectron
Valence(unsigned int, OrbitalInfo const&, Orbi…
23.89 %
CoreMBPTCalculator::CalculateTwoElectron4(
unsigned int, OrbitalInfo const&, OrbitalInfo c…
41.82 %
CoreMBPTCalculator::CalculateTwoElectron2(
unsigned int, OrbitalInfo const&, OrbitalInfo c…
19.43 %
CoreMBPTCalculator::CalculateTwoElectron6(
unsigned int, OrbitalInfo const&, OrbitalInfo c…
5.99 %
301 990…
554 34…
224 550…
434 32…
177 681 …
354 170…
MathConstant::Wigner6j(double, double, double
, double, double, double) const
50.78 %
SlaterIntegrals<std::map<unsigned long long,
double, std::less<unsigned long long>, std::all…
16.23 %
2 449 3…
std::map<OrbitalInfo, unsigned int, std::less<
OrbitalInfo>, std::allocator<std::pair<OrbitalI…
6.04 %

but only 90 CSFs, which is very unbalanced but not unrealistic for few-electron calculations.

I have focused my efforts on the "hot spots" listed above, which are the sections of code which have the largest potential performance gains. The rest of this chapter contains an explanation of the performance engineering and design choices I made when parallelising the expensive subroutines, followed by an exploration of the resulting gains in performance.

Unless otherwise stated, the profiles in the rest of this chapter were generated by profiling with the *Arm Map* profiler (part of the Arm Forge suite of HPC tools [134]), using the input file shown in listing 1. Even though this calculation requires less than the 190GB of memory available on *Gadi* compute nodes, I also wanted to make sure that calculations fit within the 128GB of memory on a single, standard compute node on the UNSW Faculty of Science's on-premises cluster *Katana*, to serve as an extra point of comparison. I have also only included solutions for odd-parity states with $J = 5/2$, as these were the largest matrices in the calculations. This chapter only analyses the OpenMP performance of AMBiT, and structuring the benchmarks to not use any MPI parallelism ensures clean measurements of only the OpenMP sections.

3.6 OpenMP

The utility of OpenMP comes from its relatively simple, high-level syntax. The C++ syntax of OpenMP is largely inherited from C and has two main components: a set of library functions for interacting with and querying the OpenMP runtime, and a set of compiler directives to specify which regions of code should be parallelised. These compiler directives have the form `#pragma omp <directive>`, where the directive is placed before the parallel section and specifies either a parallel execution environment or a synchronisation/thread control construct. The compiler then uses the instructions in the `pragma` to generate multithreaded code without the programmer needing to know the underlying implementation.

OpenMP follows the fork-join model, where the program executes in serial (i.e. single-threaded) until a parallel construct is reached, at which point it forks into multiple threads of execution, according to the compiler directive. Once all threads have completed execution, execution is then resumed by the master thread until either another parallel section is reached, or the program terminates. A full discussion of all OpenMP constructs and functions can be found in the OpenMP standard [133].

Despite the relatively simple syntax, there are three chief difficulties in obtaining maximum parallel performance from OpenMP. Firstly, shared-memory parallelism opens up the possibility of race-conditions and non-deterministic software bugs. The order in which different OpenMP threads execute a particular instruction is undefined, so multiple threads attempting to modify the contents of shared-memory without explicit synchronisation constructs will cause unpredictable behaviour, known as a race condition [133, 135]. This is especially true in the idiomatic modern C++ used in AMBiT due to our use of standard library containers, many of which do not guarantee thread-safety of write operations [136]. In order to ensure code correctness, it is necessary to either make heavy use of synchronisation constructs (such as mutexes or barriers), or refactor the parallel functions to use a *lockless algorithm*.

The use of synchronisation necessarily introduces performance slowdowns: if one thread obtains an exclusive lock on a variable to prevent multiple concurrent writes causing race conditions, then all other threads attempting to access this variable must wait until the original thread is finished and releases the lock. In the worst case scenario, every thread must wait for every other thread to finish writing to the variable before execution can resume, eliminating any benefit from using multiple threads. Consequently, I attempted to minimise the use of explicit locks throughout the OpenMP sections of AMBiT, which required significant refactoring of some sections of code.

Another challenge when writing OpenMP code is that the OpenMP interface is very high-level tends to hide implementation- and hardware-specific details from the programmer. This is by design, and often makes OpenMP much more easier to write than corresponding code using other threading libraries. However, there are many performance pitfalls which can arise from low-level details like cache-coherence [131, 135] and thread-scheduling policies [137]. Furthermore, even relatively small performance overheads can have a large effect on the parallel scaling for very large numbers of threads, so I had to use the Linux `perf` tool to profile low-level performance characteristics which Arm Map does not measure.

Finally, I have loosely followed two parallel programming "rules of thumb", which provide useful heuristics when choosing between competing parallel designs. Both of these rules of thumb relate to synchronisation between units of parallelism (e.g. OpenMP critical sections or MPI barriers):

1. Whenever possible, don't synchronise — by definition, synchronisation imposes restrictions on parallel execution flow, which in turn impacts parallel performance. A locking construct such as a mutex or OpenMP critical section imposes some ordering on threads attempting to access some resource (e.g. a variable), so any threads attempting to access the resource while it is in use must stall their execution and wait until the resource becomes available again. During this waiting period, the threads are not doing work, thus reducing the overall performance of the program.

   Although they are more difficult to design and can often have worse serial performance, a well-implemented lockless algorithm (i.e. one without any synchronisation) will often have better parallel performance and can scale out to more compute resources than a comparable algorithm which does use synchronisation.

2. If synchronisation is unavoidable, do it as infrequently as possible — very few real computational problems are amenable to a completely lockless parallel algorithm. A more realistic constraint is to ensure that synchronisation happens as infrequently as possible throughout parallel sections. The biggest performance trap that this rule deals with is the anti-pattern where some synchronisation construct is embedded inside a loop, or worse, multiple nested loops, so the overhead from locking is incurred many times in quick succession.

   Throughout AMBiT, I have tried to ensure that synchronisation is left until the last possible moment in each parallel subroutine. Often, this takes the form of storing thread-local copies of intermediate calculation results which would otherwise be stored into some mutex-protected data structure. The intermediate, thread-local copies can then be merged at the very end of the parallel function, so the cost of synchronisation is only incurred once.

It is worth noting that the use of temporary copies of thread-local variables is different to the duplication of objects between MPI processes which originally motivated the move to OpenMP in two key ways: the intermediate objects created to avoid synchronisation are much smaller than the duplicated orbitals and integrals, and they are very short-lived (the resources assigned to intermediate objects are always freed after the end of their subroutine). This means they do not significantly impact the maximum, long-term memory usage and do not affect our ability to scale out calculations to larger numbers of threads.

3.7 Parallelism and optimisation methodology

In this section, I will describe my methodology in parallelising and optimising the AMBiT code, including the computational rationale behind these design decisions. Wherever possible, I have included partial profiles and performance data, as I used a primarily data-driven approach to optimisation, often including multiple iterations and profiles to find the highest-performing designs. I will also include brief discussions on some failed attempts at optimisation, including why their outcomes were unsatisfactory, what they demonstrate about high-performance code and hardware, and how I rectified them in the final version of the software.

3.7.1 Two-body MBPT integrals

Prior to my OpenMP overhaul, the two-body MBPT diagrams already had node-level parallelism via MPI: the list of valid diagrams (which correspond to tuples of valence orbitals and the multipole moment of the integral) gets divided up evenly between MPI processes, calculated in parallel across MPI processes, then written to disk for use in future calculations and re-read to ensure each MPI node has all the calculated integrals in memory. I have left this level of parallelism intact, the existing MPI workload distribution is already efficient and balanced.

As a first attempt, I parallelised the two-body diagram loops with the OpenMP *reduction* construct. A reduction uses a binary operation (such as addition or boolean logic operations) to collapse all elements of a collection into a single value — the operator is applied to successive pairs in the sequence of elements, then recursively applied to the results of the previous step until only one value remains. A reduction can be parallelised if its operator is associative (allowing for the operations to be carried out in arbitrary order by threads), and ideally commutative (allowing for the elements of the collection to be re-ordered for performance if necessary) [133].

Reductions in OpenMP are conceptually implemented by extensions to the standard parallel loop construct: each thread is assigned a chunk of loop iterations (just as in a standard parallel loop) and carries out the reduction operation on its subset of the loop domain. The partial reductions for each thread are stored in a private variable, which are then in turn reduced into a single, final answer once all threads have finished their work.

The sum rules used to calculate MBPT corrections in AMBiT, the general form of which is shown in equation 3.2 (using the terminology of chapter 2), naturally translate into reduction constructs since addition is both associative and commutative, so I tried to use the OpenMP reduction construct to all two-body MBPT diagrams.

$$\Sigma^{(\mathrm{MBPT})} = \sum_{M \in Q} \frac{\langle I | \hat{H} | M \rangle \langle M | \hat{H} | J \rangle}{E - E_M} \tag{3.2}$$

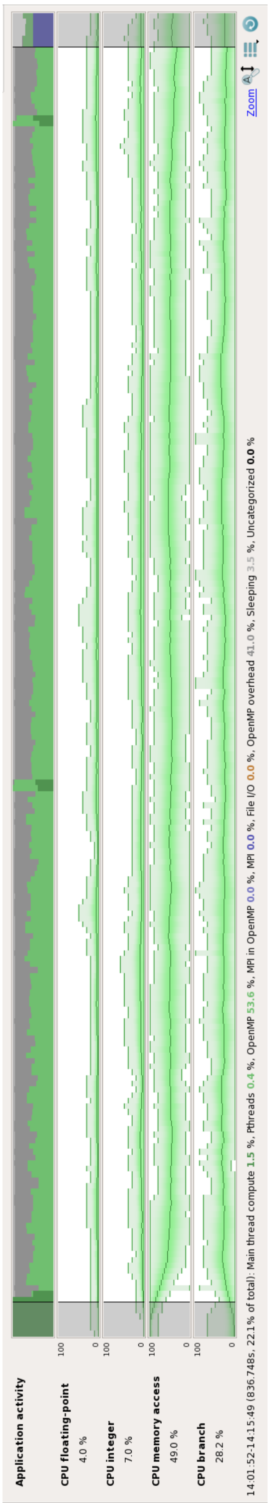
This approach has some performance limitations. Although reductions give good performance *within* the loops while requiring almost no changes to the overall code structure, it incurs a startup cost every time the execution enters an OpenMP parallel loop (i.e. once per set of external orbitals). This means we need each parallel loop to carry out a large number of iterations, otherwise any performance gain from parallelism will by overwhelmed by overhead from spinning up the parallel sections. Consequently, the best performance will be achieved when each MBPT diagram sum rule is expensive to calculate, but we need relatively few of them. Unfortunately, this approach is rarely the case in real CI+MBPT calculations, and the overhead from spinning up parallel regions limits the overall scalability of the code. As shown in figure 3.4a, the two-body MBPT diagrams achieve $\sim 50\%$ parallel utilisation, with the remaining CPU time split between OpenMP overhead and a small amount of serial code, which is clearly non-optimal and will limit parallel scalability.

The large overhead from using reductions means I needed to do more extensive re-factoring in order to extract better performance. To get an idea of the required work, let's first look at the (serial) structure of the highest-level subroutine which we can parallelise for MBPT. In (highly abstracted) pseudocode [4]:

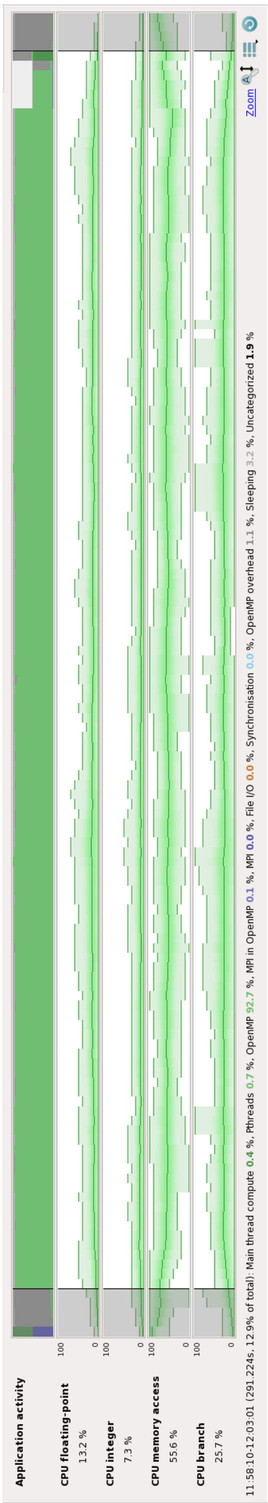
```
for((orb1, orb3, orb2, orb4) in valence)
{
  for(k) // Multipole moment. Actual range will depend on calculation parameters
  {
    if(correct_parity(orbs, k))
    {
      // Maps orbitals and k to unsigned integer key
      key = GetKey(k, orb1, orb2, orb3, orb4)

      if(key not already in two_body_integrals)
      {
        // Calculate core-valence and/or valence-valence diagrams
        integral = calculate_MBPT_diagrams(k, orb1, orb2, orb3, orb4)
        two_body_integrals[key] = integral // Associative array
      }
    }
  }
}
```

[4] The real code is found in the file `MBPT/CoreValenceIntegrals.cpp`



14:01:52-14:15:49 (836.748s, 22.1% of total): Main thread compute 1.5 %, Pthreads 0.4 %, OpenMP 53.6 %, MPI in OpenMP 0.0 %, MPI 0.0 %, File I/O 0.0 %, OpenMP overhead 41.0 %, Sleeping 3.5 %, Uncategorized **0.0** %



11:58:10-12:03:01 (291.224s, 12.9% of total): Main thread compute 0.4 %, Pthreads 0.7 %, OpenMP 92.7 %, MPI in OpenMP 0.1 %, MPI 0.0 %, File I/O 0.0 %, Synchronisation 0.0 %, OpenMP overhead 1.1 %, Sleeping 3.2 %, Uncategorized **1.9** %

(a) Arm Forge profile of the first attempt at parallelising MBPT integrals via OpenMP reductions.

(b) Arm Forge profile of the faster version of MBPT integrals, via splitting the calculation into separate serial and parallel loops.

Figure 3.4: Arm Forge profiles of the slow and fast parallel implementations of two-body MBPT integrals. The topmost bar in each profile shows the program's performance within the two-body MBPT subroutines, averaged over all threads and shown as a function of time: light green sections indicate time spent in OpenMP parallel regions, dark green represents serial code, grey represents OpenMP overhead. The bottom four bar charts show the time spent in the four most common types of CPU instructions in this section: floating-point and integer arithmetic, memory accesses, and branching logic.

The naive approach would be to parallelise the loop over orbitals since that will have a large number of iterations, and each iteration will have a large amount of work. However, this subroutine has some limitations which mean that we can't simply add an OpenMP directive and be done with it. There's some freedom in how we arrange/order the orbitals in a particular integral, which we account for in the function which converts orbitals to keys. Consequently, if we don't have some kind of coordination between threads, then we'll end up with a lot of duplicated work due to threads calculating redundant diagrams. This means there must be some inherently serial component of the MBPT integrals subroutine, which we can deal with in one of two ways:

1. Encase the conditional to check whether this integral has already been calculated, as well as storing the integral in `two_body_integrals` in a critical section. This will force them to be executed by one thread at a time, or
2. Split the serial and parallel components of the workload into two separate loops.

The first approach synchronises threads at every iteration of the loop over orbitals, which causes a lot of time wasted due to thread-contention over the `two_body_integrals` map. Preliminary tests showed that this approach does not scale well, and can give worse performance than serial execution due to the presence of critical sections inside the tight inner loop over orbitals.

Instead, I decided to split the calculation into two loops as follows:

1. Loop over orbitals and k-values in serial, and check whether each combination satisfies parity rules.
2. If we haven't yet saved this orbital, make a tuple of the orbitals plus k and append it to a vector. Next, calculate the unsigned integer key for this set of orbitals and store that in a separate vector such that `orbitals[i]` and `keys[i]` represent the same integral. We can't just store the key, since the ordering of orbitals is semi-important: appropriate permutations might correspond to the same physical quantity, but certain orderings are more numerically stable than others. Converting to and from the numeric key does not preserve ordering of orbitals, so we want to explicitly store the most stable tuples of orbitals.

   The value of the MBPT integrals is not actually calculated at this stage, so this loop is relatively cheap even when executed in serial.
3. Pre-allocate storage to hold the corresponding integral values. Pre-allocating storage before entering the parallel region means that modifying elements of the vector is thread-safe, as long as no two threads attempt to access the same element.
4. Loop over the pre-calculated orbital tuples in parallel, calculate the corresponding integrals and store them in the pre-allocated vector. After this stage, both vectors should be full of key-value pairs such that the key at `keys[i]` corresponds to the integral at `values[i]`.
5. In serial, the root MPI process collects the key-value vectors from all processes and writes the to disk. All processes then read the complete integrals from disk and store them in an associative array.

In pseudocode, the new algorithm has the form:

```
for((orb1, orb3, orb2, orb4) in valence)
{
  for(k) // Multipole moment. Actual range will depend on calculation parameters
  {
    if(correct_parity(orbs, k))
    {
      key = GetKey(k, orb1, orb2, orb3, orb4) // Maps orbitals and k to
                                              // unsigned integer key
      if(key not in two_body_integrals)
      {
        // Append this key to the end of the array, but don't calculate the integrals
        keys.append(key)
        expanded_keys.append(k, orb1, orb2, orb3, orb4)
      }
    }
  }
}

values.resize(number_of_keys)

#pragma omp parallel for
for(i = 0 to number_of_keys)
{
  // Calculate core-valence and/or valence-valence diagrams.
  // We get the orbitals by "unpacking" the elements of expanded_keys
  k, orb1, orb2, orb3, orb4 = expanded_keys[i]
  integral = calculate_MBPT_diagrams(k, orb1, orb2, orb3, orb4)
  values[i] = integral
}
```

where `#pragma omp parallel for` indicates that OpenMP should execute the loop over `i` in parallel.

This approach is somewhat more intricate than my first attempt at parallelising two-body MBPT, but provides significantly higher performance and scalability: there is only one OpenMP parallel region when calculating the integrals, and the parallel region contains no synchronisation or locking constructs so should scale much better. This is borne out in the profile shown in figure 3.4b, where the MBPT calculation is faster than my first attempt using reductions by a factor of 2.9 ($T_{\text{wall}} = 291$s for the fast MBPT subroutine, vs $T_{\text{wall}} = 837$s for the initial attempt. The profile also shows the subroutines achieve $> 90\%$ parallel utilisation.

3.7.2 Generating the Hamiltonian Matrix

Parallelising the CI matrix generation was made simpler by exploiting the way the CI matrix is stored in memory. CI matrix is already split into so-called “chunks”, each of which which contains a fixed number of relativistic configurations, each of which correspond to a variable number of projections and CSFs. In the MPI-only version, these chunks are then parallelised by distributing them between MPI processes, so each process gets a subset of chunks to iterate over (rather than the full matrix). We can conceptually represent the process of generating the CI matrix with the pseudocode:

```
// chunks_array contains this MPI process's set of chunks, so this
// loop will have different iterations for each process
for(current_chunk in chunks_array)
{
  for(config_i, config_j in current_chunk)
  {
    for(proj_i in config_i)
    {
      for(proj_j in config_j)
      {
        if(do_three_body)
        {
          operatorH = H_three_body->GetMatrixElement(proj_i, proj_j);
        }
        else
        {
          operatorH = H_two_body->GetMatrixElement(proj_i, proj_j);
        }

        for(CSF_i in proj_i)
        {
          for(CSF_j in proj_j)
          {
            // M is the Hamiltonian matrix
            M[CSF_i, CSF_j] += f(CSF_i, CSF_j) * operatorH;
          }
        }
      } // End of loop over projection_j
    } // End of loop over projection_i
  } // End of loop over configs
49 } // End of loop over chunks
```

The MPI parallelism (which is implicit in the loop over each process's subset of chunks on line 21) is already efficient, so I have kept the MPI distribution as is, and added an extra level of intra-node parallelism via OpenMP. As a first approach, I added OpenMP parallelism by splitting up the loop over matrix chunks between threads by using the OpenMP `parallel for` construct, as shown in the pseudocode:

```
// A pragma is used to pass a special instruction to the compiler
// In this case, we're telling the compiler to parallelise this
// loop with OpenMP
#pragma omp parallel for
for(current_chunk in chunks_array)
{
  // Generate the chunk's matrix elements...
} // End of loop over chunks
```

Moving away from pseudocode, the C++ implementation of the code makes chunk-level parallelism relatively easy to implement. MPI processes are assigned a roughly equal number of chunks to evaluate, the "metadata" for which is stored in a C++ `std::vector`. Each chunk contains the rows corresponding to a fixed number of relativistic configurations. Different relativistic configurations have different numbers of CSFs, so the chunks potentially contain a different number of CSFs (rows). The matrix elements are themselves stored in a contiguous, row-major ordered matrix M while the diagonal elements (for emu CI calculations) are stored in a separate array D [5]. This means, individual elements of the Hamiltonian matrix are never directly manipulated but are accessed through the abstraction of the matrix chunks; individual chunks cannot access the matrix elements belonging to other chunks.

Since the chunks are stored in a contiguous, random-access container, the loop over chunks can be implemented by looping over a range of integer indices in parallel, and then indexing into the current chunk as shown below:

```
#pragma omp parallel for
for(unsigned int chunk_index = 0; chunk_index < chunks.size(); chunk_index++)
{
   MatrixChunk& current_chunk = chunks[chunk_index];
   Eigen::Matrix<double ... Eigen::RowMajor>& M = current_chunk.chunk;
   Eigen::Matrix<double ... Eigen::RowMajor>& D = current_chunk.diagonal;
   // Generate the chunk's matrix elements
} // End of loop over chunks
```

[5] Internally, the matrices are stored in memory as a contiguous, one-dimensional array of numbers, with the *Eigen* library serving as a compatibility layer to allow the higher-level code to treat the data *as if* it were stored in a 2D matrix

This approach has very small overhead, since each chunk belongs to exactly one thread, so there is very little copying or sharing of data between threads. Furthermore, all large data chunks and matrices are accessed via C++ references, so there is very little in the way of memory allocation or manipulation.

The biggest factor in the performance of this section of code is *load-balancing*. Ideally, we want all threads to take roughly the same length of time to complete their work in a parallel region. The program cannot proceed to solving the Hamiltonian matrix until the matrix has been fully generated, so if the threads have drastically different amounts of work to complete, then the threads which finish early will be sitting idle while they wait for the rest to catch up.

Additionally, it is far worse for a small number of threads to take longer to complete than it is for that small number to finish early. As an example, if one thread completes its work 20% quicker than all other threads, then that one core will be idle for 20% of the run-time of the parallel section. However, if one thread takes 20% longer to complete its work than the rest of the threads, then *all* $N - 1$ remaining threads (which can be as many as 24 or 47 on *Gadi*) will be idle for 20% of the run time. Clearly the second option will result in far worse scaling and CPU utilisation than having a single thread finish early. Consequently, improper load-balancing schemes can drastically reduce the performance of parallel sections of code; a problem which generating the Hamiltonian matrix is especially susceptible to due to the variable number of projections and CSFs belonging to each matrix chunk.

The simplest way to ensure at least *some* kind of load-balancing among chunks is to modify the `#pragma` directive of the OpenMP for-loop to use one of the OpenMP scheduling schemes, via the *scheduling clause*:

```
#pragma omp parallel for schedule(<scheduling_policy>)
```

where `<scheduling_policy>` specifies the scheme by which the iterations of the loop are distributed between threads [133]. The two key schemes of interest here are *static* and *dynamic* scheduling. Under static scheduling, each thread gets a fixed number of loop iterations to execute: the default static scheduling will provide each thread with an approximately equal sized “chunk” of iterations, the actual size of which is implementation-defined [133]. Alternatively, multiple fixed-size chunks can be assigned to the threads as an extra parameter to the scheduling clause, so the directive:

```
#pragma omp parallel for schedule(static, 10)
```

will divide the loop domain into chunks of 10 iterations, which are then distributed to threads in a round-robin fashion [133]. There is very little performance overhead to static scheduling, as the size of the chunks, as well as the threads they’re assigned to, is calculated once, either at compile time (if the number of iterations is an integer constant expression) or upon entering the parallel loop. However, static scheduling does not allow for more than the most rudimentary load-balancing: the chunk size must be constant, and there is no way to specify *which* iterations should belong to which chunk. As a result, static scheduling is unable to properly handle the unbalanced loads caused by unequal numbers of projections for different chunks of configurations in the Hamiltonian matrix.

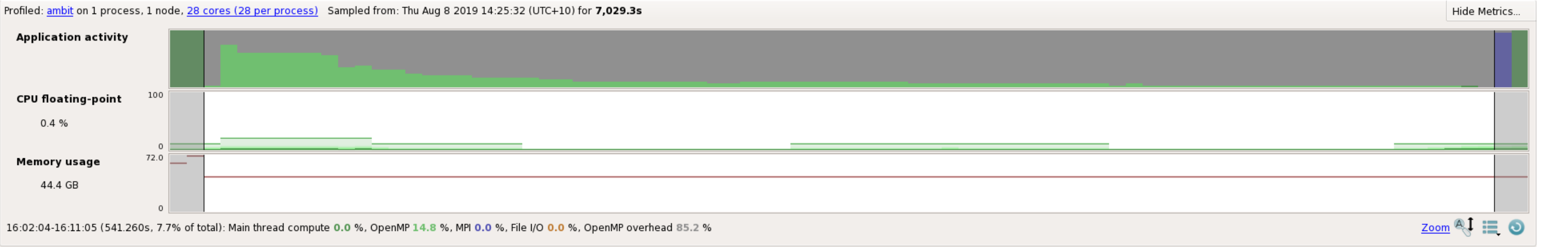
Instead, we must turn to dynamic scheduling. The dynamic scheduling scheme distributes iteration chunks to threads at run time on a more flexible basis: threads execute a chunk of iterations (the default size is one iteration per chunk, but this can be overridden in the scheduling clause), then obtain a new chunk from the OpenMP run-time and continue executing until there are no remaining chunks. This way, threads with large chunks are less likely to hold up execution, as the other threads can pick up extra work while the large chunks are being generated. The downside is that dynamic scheduling has a larger performance overhead at run-time than static scheduling, since the blocks of chunks must be distributed between threads every time a thread finishes its current block of iterations.

Figures 3.5a and 3.5b show profiles for the same Db CI+MBPT calculation using static and dynamic scheduling, respectively, with a focus on the section corresponding to generating the Hamiltonian matrix. These profiles do not contain any further optimsations beyond adding OpenMP to the Hamiltonian matrix subroutine. The static scheduling calculation requires 6hr 58min on a single compute node, while using dynamic scheduling cuts the required walltime to 3hr 24min — dynamic scheduling clearly provides better OpenMP performance in this case. The performance difference between static and dynamic scheduling remains consistent across all atomic systems presented in this thesis.

The highlighted region of the profile in both figures shows the reason for the difference in performance between scheduling schemes. The topmost bar in figure 3.5a shows that most of the CPU time required to generate the matrix is spent in OpenMP overhead when using static scheduling; specifically, this is the time the threads spend waiting at the implicit barrier at the end of the OpenMP parallel region. Furthermore, the pattern of CPU usage (shown in light green) displays the characteristic behaviour of a workload imbalance: the CPU utilisation starts relatively high, but falls off in a “step-like” pattern. This indicates that certain threads reach the OpenMP barrier earlier than others, so the time spent in OpenMP overhead gradually increases (or conversely, the CPU usage decreases) as each thread finishes and sits idle waiting for the threads with large workloads to complete.

The profile in figure 3.5b shows much more balanced CPU usage patterns. The subroutine spends all of its time with almost full CPU utilisation, indicating that all threads finish their work at approximately the same time. Dynamic scheduling thus makes far more efficient utilisation of CPU resources, leading to a much shorter absolute walltime to complete.

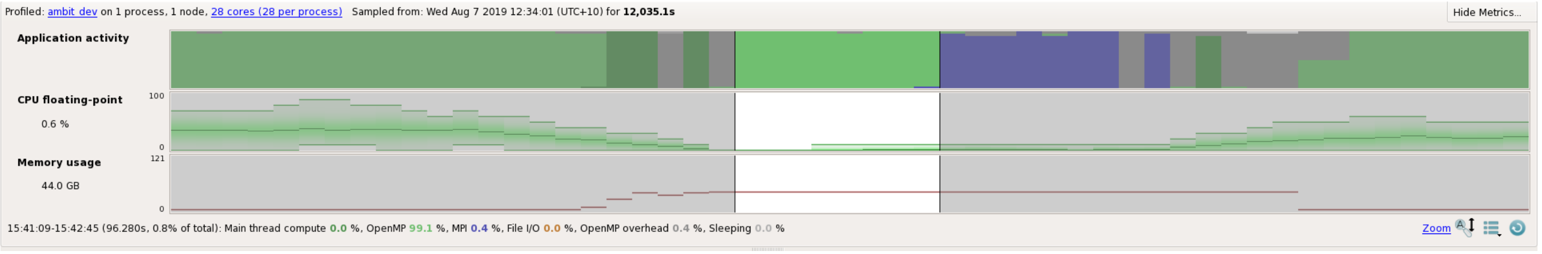
This kind of dynamic workload balancing has drawbacks, however. Firstly, it incurs extra overhead in setting up the work-queue and distributing chunks between threads when compared to static scheduling (where the workload distribution can be done at compile-time). Secondly, the approach of dynamically sharing chunks will not be able to produce balanced workloads in the case where a subset of chunks ($N < N_{threads}$) takes longer to complete than the rest of the matrix combined. If the atomic system contains a small number of configurations with e.g. high-angular momentum, orbitals in half-full shells, then chunks containing the open-shell configs will contain extremely large numbers of CSFs and projections. In this case, the majority of threads (and pos-



Profiled: ambit on 1 process, 1 node, 28 cores (28 per process) Sampled from: Thu Aug 8 2019 14:25:32 (UTC+10) for **7,029.3s**

16:02:04-16:11:05 (541.260s, 7.7% of total): Main thread compute 0.0 %, OpenMP 14.8 %, MPI 0.0 %, File I/O 0.0 %, OpenMP overhead 85.2 %

(a) Profile of calculation with static scheduling. The CPU utilisation (shown in light green) displays the descending "step pattern" characteristic of a workload imbalance.



Profiled: ambit dev on 1 process, 1 node, 28 cores (28 per process) Sampled from: Wed Aug 7 2019 12:34:01 (UTC+10) for **12,035.1s**

15:41:09-15:42:45 (96.280s, 0.8% of total): Main thread compute 0.0 %, OpenMP 99.1 %, MPI 0.4 %, File I/O 0.0 %, OpenMP overhead 0.4 %, Sleeping 0.0 %

(b) Profile of calculation with dynamic scheduling. The CPU utilisation (shown in light green in the highlighted region) displays a much more balanced usage pattern than that shown in figure 3.5a.

Figure 3.5: Profiles of subroutine to CI matrix for Db calculations, using different OpenMP scheduling policies. The profiles were generated via Arm Map.

sibly even the majority of MPI processes) will run through the entire work queue and be stuck waiting at the end of the parallel region for the threads with "big chunks" to finish generating their work, which wastes CPU resources.

OpenMP loop iterations are atomic (i.e. cannot be dynamically split into smaller tasks), so the pathological case where the matrix contains a few extremely large chunks will result in most of the threads sitting idle (sometimes for very long periods of time), thus wasting compute resources and limiting parallel scalability. I have yet to find a solution for this within the framework of OpenMP which would not require a substantial re-write of AMBiT (for example, shifting to a framework where the matrix is broken into smaller, more uniform units such as projections), so I have had to leave this problem as it is. The dynamic load-balancing only fails in very narrow circumstances, so this is not a "show-stopper" performance bug as of yet but may need to be fixed in the future.

As a final note, the actual process of solving the CI matrix eigenvalue problem is very easy to parallelise. The Davidson algorithm [138] is amenable to MPI parallelism as it relies on matrix-vector multiplication, which can be trivially split into blocks and calculated in parallel, with each process getting an even-sized chunk of work. This MPI parallelism was already implemented with effective scaling characteristics, so I have left it untouched. However, the individual matrix-vector multiplications can be further parallelised via OpenMP-style multithreading without any actual code re-factoring, as our linear algebra operations are all implemented via the Eigen C++ libraries [139]. Eigen acts as a template-based wrapper around lower-level numerical libraries (which can be chosen at compile time), several of which support multithreading for primitive linear algebra operations. By default, Eigen supports multithreading, with no changes to the code, for general dense matrix multiplication, but can also be configured to use other linear algebra engines with more efficient parallelism such as OpenBLAS and Intel's MKL [139]. This allows us to parallelise the Davidson eigensolver routine for effectively zero development cost.

3.7.3 Many-body matrix elements

This section of the code consists of subroutines which calculate the matrix elements for many-body operators $\langle I|\,\widehat{O}\,|J\rangle$, where $\widehat{O}$ is a one- two- or three-body operator, and $|I\rangle$ and $|J\rangle$ are solutions to the CI eigenvalue problem.

There are two main expensive cases/prototypes to consider when optimising the many body operator:

- GetMatrixElement(LevelVector&), which is used when calculating the expectation value of an operator, most commonly the spin projection operator $\langle I|\,S_z\,|I\rangle$ as part of the g-factors.
- GetMatrixElement(LevelVector&, LevelVector&) for calculating transition matrix elements $\langle I|\,\widehat{O}\,|J\rangle$, such as for electric/magnetic multipole transitions.

`LevelVector` is an internal class that holds the solutions to the CI eigenproblem. Both of these cases need parallelisation, but the g-factor calculations in particular can take a very long time in many-electron calculations, such as the dubnium profile shown in figure 3.2.

The standard OpenMP parallel for loop is not a perfect fit for this section of the code, as it relies on loops over non-random access iterators. In both `ManyBodyOperator` subroutines in this section, we iterate over a custom object which holds the list of relativistic configurations in each CI solution.

When iterating over the configurations, we need to keep track the corresponding projections and CSFs for the current configuration. CSFs are represented as a linear combination of projections, so the code only stores the expansion coefficients, rather than the whole wavefunction — conceptually, these coefficients are stored in a matrix-like structure, with each row corresponding to a new CSF. The iterators must also keep track of which CSFs belong to the current projection involves querying the the stored angular momentum data to calculate the number of CSFs corresponding to the current configuration and updating a separate iterator into the CSFs. This offset will be different for each configuration and must be done every time the iterator over configurations is incremented. As a result, the configuration iterators take $O(n)$ time to advance by n elements, which means that they cannot (by default) be used as the loop variable in OpenMP directives.

The textbook solution to this kind of problem is to use the OpenMP *task* construct, which is outlined in section 2.11 of the OpenMP 4.0 standard [133]. The tasking construct generates a series of small workloads and places them in a priority queue. The task queue is shared between all threads in a parallel region, and all threads can take tasks to execute if they have no other work to do. Tasks are executed in a first-in, first-out (FIFO) manner by default, but optional priorities can be assigned to tasks, as well as requirements that certain tasks be executed before others can begin. The chief advantage of tasks in this particular use case is the ability to generate tasks via an asynchronous pattern:

1. A single thread runs through the loop over configurations and generates (but doesn't yet execute) tasks containing the numerically expensive parts of the matrix-element calculation.
2. The remaining threads begin executing tasks from the work queue.
3. If the generator thread finishes generating tasks while there is still work in the queue, then it begins executing tasks along with the consumer threads, until the work is finished.

On paper, tasks work well for loops with an unknown length at run-time, recursive algorithms, and algorithms which rely on non-random access iterators (e.g. traversing the elements of a linked-list); all algorithms which are more difficult to parallelise using standard worksharing directives like parallel for-loops. However, despite these theoretical advantages, the actual implementations of tasking have noticeable overhead which increases non-linearly with both the number of tasks and the number of threads [137]. This overhead is especially bad for workloads consisting of large numbers of tasks with unequal size. This means that, in practice, codes which trigger this edge-case behaviour end up spending most of the execution time in OpenMP overhead, even in the ideal case of using multiple threads to produce tasks outlined in Ref. [137], as any benefit from running with multiple threads is cancelled out by the increased OpenMP overhead.

Unfortunately, my testing showed that the tasking-based parallelisation of `ManyBodyOperator` subroutines suffers from this performance pathology. Profiling with Arm Map shows that only $\sim 30\%$ of the CPU time when calculating g-factors is spent on arithmetic operations (the section of code has very few branching instructions so this will be the majority of the productive work), while the remaining time is spent in OpenMP overhead, which is a combination of tasks waiting at the end of the region and the task scheduler. This is obviously sub-optimal and limits parallel scalability.

Instead, we turn to a for-loop with an integer loop variable which runs over the size of the configuration list, while using `std::advance` to manually advance the configuration list iterator to the appropriate point. Since the iterator is not random-access, advancing it by an arbitrary amount adds some overhead each time it is advanced. This overhead is linear in the number of projections but, crucially, is constant in the number of threads. Furthermore, this overhead is split between worker threads and processes, since it is incurred once per parallel loop, meaning that its impact is not as bad as it may at first seem. While not ideal for scalability, the loop-based approach is far better than the equivalent implementation using tasks, where the nonlinear overhead eventually leads to anti-scaling behaviour and severely reduces performance.

This performance improvement is borne out when comparing the run time of the two methods: using task-based parallelism for Db calculations takes 1044s to calculate the g-factors for the $J = 1/2$ levels, but for-loop based parallelism takes 558s for the same g-factors. Transition matrix elements have similar performance benefits, as will be shown in the final profiles in section 3.8.

3.7.4 Slater Integrals

The subroutine to calculate two-body Slater (Coulomb) integrals did not appear as a hotspot in figures 3.2 and 3.3, but I've included it here since it is similar to the two-body MBPT calculations, but with a couple of extra constraints which make it more difficult to effectively parallelise:

- Thread-safety: The integrals are stored in a hash table, which is not thread-safe to write to. Inserting new keys occasionally causes the whole map to grow in size (this is necessary to maintain performance) [140], so requires synchronisation between threads to avoid corrupting the data structure. This synchronisation is currently achieved in AMBiT via an OpenMP `critical` section, and necessarily limits the parallel speedup.

- Work duplication: As with the two-body MBPT integrals, there's a certain amount of freedom in how we arrange the orbitals in a particular Slater integral which results in some duplication of workload. Consequently, we can't just iterate over all orbitals without some kind of synchronisation between threads to avoid duplication.

  In addition, the way we calculate the Slater integrals imposes extra restrictions on the ordering of the calculations. For a given Slater integral $R^k(ab, cd)$ (i.e. the k-th order element in the multipole expansion of the corresponding Coulomb integral $\left\langle ab|\frac{1}{|r_1-r_2|}|cd\right\rangle$), we first calculate the Hartree-Y operator (sometimes called the Hartree screening operator) for the orbitals b and d, which is defined as:

$$Y^k_{bd}(r) = \int \frac{r^k_<}{r^{k+1}_>} \psi^\dagger_c(r')\psi_d(r')dr' \cdot \xi(k + L_c + L_d) \cdot \Delta(k, J_c, J_d) \tag{3.3}$$

Where ξ and Δ ensure the correct angular momentum conditions $|L_c - L_d| \leq 2k \leq L_c + L_d$ and $k + L_c + L_d =$ even. The Hartree Y operator is defined such that:

$$R^k(ac, bd) = \langle a| Y^k_{cd} |b\rangle \tag{3.4}$$

By calculating the Hartree Y operator first, we can factor out the shared components for all integrals with the same c and d, thus preventing a significant amount of duplication. The downside is that the orbitals must be looped over in order such that all integrals with the same c and d are grouped together. This makes parallel execution trickier to implement, since we ideally want to be able to loop over integrals in an arbitrary order.

With these constraints in mind, I had originally used the OpenMP *task* construct to parallelise the Slater integrals calculation. Using the producer-consumer model, we can have one thread iterate over the c and d orbitals and generate tasks containing all Slater integrals which correspond to that cd pair, thus avoiding workload duplication. The problem with this, as shown in section 3.7.3, is that OpenMP tasks incur a large overhead which grows with both the number of tasks and nonlinearly with the number of threads. This overhead becomes large enough to cause significant bottlenecks when calculating very large numbers of Slater integrals — a condition which is almost always met in few-electron systems with many more Slater integrals than CSFs in the Hamiltonian matrix.

Given these performance characteristics, I have instead used the much simpler `parallel for` construct [6]. Parallelising the outermost loop over orbitals results in coarser-grained parallelism than generating tasks, but the reduced overhead makes `parallel for` the better performing design.

Even without the overhead from task parallelism, there is still considerable overhead from synchronising access to the two-body integrals store/map. The map is written to inside the innermost body of five tightly nested loops (the orbitals a, b, c, d and multipole moment k), leading to high-levels of contention when large numbers of threads are sitting idle waiting for a mutex, in turn limiting the parallel scaling.

The most obvious optimisation would be to take a similar approach to the one I used for the two-body MBPT integrals: split the loop into an inherently serial component where we calculate valid sets of orbitals, and then a parallel loop where we calculate the corresponding integrals. This approach would also require thread-safe intermediate containers to hold the orbital-integral pairs (to eliminate the mutex), and a follow-up serial component to transfer the integrals into the

[6] this was only possible after changing the underlying structure of AMBiT's OrbitalMap to an associative array with random-access iterators, which I describe in section 3.7.5

final associative array. Unfortunately, this approach is complicated somewhat by the necessity of iterating over orbitals in a specific order. Unlike the two-body MBPT integrals where the tuples of orbitals can be calculated in an arbitrary order (which is ideal for parallel sections), the grouping of the Slater integrals necessary for deduplicating work calculating the Hartree Y operator means that we can't simply add a parallel for loop and be done with it.

OpenMP offers no way to impose ordering or grouping on parallel loop iterations [7], so the necessary grouping will need to be achieved through the intermediate container we use to store the sets of valid integrals to be calculated.

I attempted to ensure the orbitals were grouped appropriately by storing them in an associative array - the keys in the map are c, d orbital pairs, which map to an array containing the corresponding (a, b, k) triples. Unfortunately, the extra overhead in constructing the map and then accessing the elements, plus the somewhat cache-unfriendly data layout (which I tested for all three types of maps in section 3.7.5) cancels out the benefits from increased parallelism by splitting the loops.

Consequently, I have kept the Slater integrals subroutine as one loop, and parallelised it using the OpenMP `parallel for` construct. This approach does not provide optimal scaling behaviour, but it would likely require a prohibitively large re-factor (relative to the computational benefit) of the way the Slater integrals are calculated to achieve optimal strong scaling. The effects of the relatively-large serial overhead of this approach on the code's overall performance will be discussed further in section 3.8.

3.7.5 Choice of data structures - associative arrays

While it doesn't strictly fall under the category of parallel programming, the choice of data structures can have significant impacts on software performance. AMBiT makes heavy use of associative arrays, making them a good candidate for optimisation by careful analysis and choice of implementation.

An associative array (also sometimes known as a map or a dictionary) is a data structure where elements are accessed by a key (which can be any data type) rather than an index/position as in arrays or vectors. Logically, an associative array consists of a set of key-value pairs, where each key appears at most once in the array, and which may or may not have some ordering imposed on them. There are many ways to implement this abstract definition in software, but I will focus on three implementations in this sections: binary-search trees, hash tables, and flat maps.

In all major C++ compilers (as of the time of writing this thesis), the standard template library (STL) defines APIs for two types of associative arrays. Firstly, `std::map` is an associative array with the following properties [136]:

- Keys are stored in order,

[7] It is possible to assign each thread a certain number of iterations in a chunk (e.g. each thread might evaluate four loop iterations under dynamic scheduling before retrieving four more from the work queue), chunk size must be a loop-invariant constant, which is not useful here as each pair of c, d orbitals will have a variable number of corresponding integrals.

- Inserting, deleting, and searching for a key must have $O(\log n)$ complexity (where n is the number of elements in the map),
- Iterators must not be invalidated by inserting or deleting element (except if the iterator pointed to the deleted element).

In theory, the C++ standard does not specify how a `std::map` should be implemented, but in practice the above requirements limit implementations to a binary-search tree [136]. All major C++ implementations (i.e. GCC, LLVM and Intel) use a specific kind of self-balancing binary-search tree called a red-black tree. A full overview of red-black trees are beyond the scope of this section (see, e.g. [141] for an in-depth discussion), but for the purposes of this section, a binary search tree (of which red-black trees are a special case) is a tree which fulfills the following two properties [141]:

1. Each node has at most two children, except for leaf nodes which have no children,
2. For each node, all elements in the left subtree are "less than" the element in the current node, while elements in the right subtree are "greater than" the element in the current node (for type-appropriate definitions of "greater than" and "less than").

Due to this ordering of nodes, searching for an element in a binary-search tree has $O(\log n)$ average complexity, where n is the number of nodes in the tree [8]. Strictly speaking, the complexity of searching through a binary-search tree is proportional to the height of the tree — the length of the longest path from root to leaf. In the worst case performance, the all elements in the tree are consistently in the same subtree, in which case the tree is said to be "maximally unbalanced" and the complexity of searching for an element reduces to $O(n)$. Red-black trees have the additional property that they are self-balancing, and so avoid the worst-case linear behaviour.

Internally, a red-black tree implementation consists of a series of "node" objects, where each node contains at least the data in the node, plus a pointer to its two child nodes. The pointer-based structure of trees makes inserting new data relatively cheap, since these operations only require finding the appropriate position in the tree (which is $O(\log n)$), allocating memory for the data in the new node, and updating some pointers. The same analysis holds true for deleting a node.

However, iterating over the tree requires stepping sequentially through all nodes by following the pointers from parent to child node — referred to as "pointer chasing" or "walking the tree". Even though this operation is only linearly proportional to the number of nodes in the tree (since all nodes are visited exactly once), it tends to perform badly on modern CPU architectures due to the effects of memory caching. Modern CPU architectures attempt to speed up memory accesses by storing the results of recent memory requests in small, fast caches, so that subsequent accesses to the same addresses do not need to go through the (relatively slow) main memory (RAM). Cache memory accesses have approximately an order of magnitude lower latency than RAM, so proper utilisation of CPU caching is critical to achieving good performance in memory-heavy code such as AMBiT.

[8] This is conceptually identical to a binary search algorithm. Each time we step down a level in the tree, the size of the search space reduces (on average) by half, leading to logarithmic scaling behaviour.

The data in a tree is not guaranteed to be stored contiguously in memory (in fact, if other operations are performed in between tree updates it is very likely to be non-contiguous), which hinders the ability of memory caching to speed up memory accesses. Memory caching requires spatial locality of data to be useful (since the data is transferred from RAM to cache in contiguous blocks called cache-lines), so walking the tree will incur cache-misses and subsequent expensive instructions to fetch the data from RAM on most steps through the tree. This is true of almost all pointer-chasing algorithms, and can seriously limit performance on modern CPU architectures [131].

Contrast this with another common implementation of an associative array: the hash table. Briefly, hash tables store key-value pairs in a flat array, where the index of a particular record is calculated by computing the "hash" of its key. A hash function maps a (potentially complex) data type to an integer. A good hash function should be fast to compute, have few collisions (i.e. be unlikely to map two keys to the same index) and give good locality of reference. Since the hash of the key is converted to an index in a flat array, searching, inserting, and deleting an element can all be done in constant ($O(1)$) time [141].

Hash functions usually do not preserve relative ordering of keys, so hash tables are only feasible when the ordering of elements is not important. Additionally, hash tables do not always benefit from memory caching, but their $O(1)$ complexity means they will usually outperform binary search trees for usage cases which are predominantly random-access. Finally, the performance of hash tables is more variable than for binary search trees, particularly when it comes to inserting data. If the hash table gets too full, then the table will need to be re-sized and the elements re-ordered, which will have $O(n)$ complexity [9]. Consequently, if large numbers of elements are inserted, and the final size of the table is not known ahead of time, then multiple expensive re-size operations will be required, increasing the time taken to build the hash table. There is therefore a trade-off between (potentially) reduced performance when constructing the hash table and its superior speed when it comes to random accesses.

Even though they all share $O(1)$ complexity for searching and insertion/deletion, not all hash tables have the same overall performance: "Big-O complexity" only describes *asymptotic* dependency of run-time on problem size, but two different data structures can have different pre-factors to the complexity analysis or extra overhead from setting up or tearing down the data, leading to large real-world performance differences. Hash table design is an area of ongoing research in software engineering, but to improve compatibility across computer systems, we have settled on Google's `dense_hash_map` [140] as our hash table implementation. `dense_hash_map` generally has better performance than the C++ standard library's hash table implementation, `std::unordered_map` (see, for example, Ref. [142]) for small, frequently accessed associative arrays, while still being installed on many HPC systems. The wide availability and ease of installation make `dense_hash_map` a good fit for the requirements of AMBiT, even though faster and more advanced hash table implementations exist.

[9] Hash table implementations such as Google's `dense_hash_map` do this re-ordering before the table becomes full to optimise lookup time

The last associative array implementation I have used in AMBiT is the flat map [10], so called because it essentially a "flattened" version of a binary search tree. A flat map consists of two arrays: one holding the set of keys, the other holding their corresponding values. The two arrays are sorted, with their relative order maintained, so the i-th key corresponds to the i-th value. Since the arrays are always sorted, searching for an element can be achieved via binary-search, which has $O(\log n)$ complexity. However, inserting an element requires all elements to the right of its position to be shifted one position to the right, which requires $O(n)$ memory operations. The same logic applies to deletion, except the positions are shifted to the left. Consequently, a flat map has the same search performance as a binary search tree, but can potentially be much slower for insertion and deletions, especially when constructing the map.

The chief advantage of a flat map is that, once the map has been constructed, the data is stored contiguously in memory, meaning iteration over the map is extremely fast. The improved data locality means that memory caching provides significant performance boosts, and the predictable memory access patterns mean that the CPU can relatively easily predict which elements will be needed and pre-fetch them, allowing for significant performance boosts from modern CPU hardware optimisations such as pipelining and speculative execution. A detailed explanation of these hardware optimisations can be found in chapter 2 of Ref. [131], but I will now provide a brief outline as they apply to AMBiT.

As previously stated, memory caching relies on the principle of *locality of reference*, which takes two chief forms [11]:

1. Temporal locality: if a memory address is accessed, it is likely to be accessed again in the near future (for example, the counter variable in a loop or a frequently read array). To optimise for temporal locality, the contents of memory locations are stored in small, fast caches close to CPU cores when accessed, thus speeding up future accesses.

2. Spatial locality: if a memory address is accessed, the addresses "nearby" are also likely to be required in the near future (for example, when sequentially accessing elements in an array). To optimise for spatial locality, memory addresses are read in contiguous blocks or "cache lines" and stored in CPU caches to speed up future accesses to adjacent memory locations.

The assumption of locality of reference is a heuristic which does not apply to all codes (or even all sections of a code), but is common enough that CPU optimisations which assume locality provide large enough performance gains on average to have become nearly ubiquitous. Consequently, optimising code to maximise locality can provide huge performance benefits. Spatial locality of data is greatly improved by the use of flat maps. When iterating over a flat map (for example, iterating over an associative array of orbitals to calculate two-electron integrals), the memory accesses are almost entirely sequential, which allows for significantly larger speedups due to memory caching when compared to the more scattered pointer-chasing accesses of a binary-search tree.

[10] Specifically, the flat_map implementation provided by the Boost C++ libraries as described in Ref. [143]

[11] These details are specific to x86 architecture processors, but almost all mainstream processors have roughly the same caching behaviour

Speculative execution, on the other hand, is a technique to exploit *instruction-level parallelism* (ILP), where multiple independent CPU instructions (e.g. arithmetic operations which use independent registers of a CPU) are re-ordered so they can be executed in parallel by different parts of an individual core to increase the instruction throughput. Speculative execution extends this model by executing instructions before it is known whether they are needed: upon encountering a conditional statement, the CPU "guesses" which branch is most likely to be taken and executes the program as if that guess is correct, while the actual conditional is evaluated in parallel. Conditionals often expensive to evaluate (for example, they may depend on the values of multiple complex variables), so executing these instructions in parallel with the speculated branch can save considerable time, especially inside tight loops. Even if the CPU's guess is incorrect, the speculated work is discarded and the correct branch is then executed, but this is no worse for performance than if the conditional and work were to be executed in serial (i.e. without speculation). A more detailed explanation of the principles and implementation of speculative execution can be found in chapter 3 and appendix C of Ref. [131].

The simple structure of a flat map when compared to other associative arrays can expose large performance gains from speculative execution when accessed in a predictable, contiguous manner. Furthermore, modern CPUs can speculatively execute quite complex instructions, such as dereferencing pointers and accessing fields of complex data structures [131], so flat maps of the data types used in AMBiT can still provide large performance gains.

Finally, the contiguous memory layout of a flat map means that their iterators satisfy the C++ constraint of being "random access": namely, their iterators can be incremented and decremented, and support accessing arbitrary elements by index in constant time. This allows them to be used efficiently with OpenMP looping constructs, such as the nested loops over orbitals when calculating two-body Slater integrals, and allows for much cleaner parallel sections [133]. This efficient and clean integration with OpenMP is not possible when using `std::map`, since it is implemented as a red-black tree and has linear complexity when searching for a random element by index.

Given the above characteristics, a flat map will provide superior performance to a binary-search tree or hash table if it is sequentially read more often than it is written to. Additionally, if the flat map is constructed once and used frequently, then the relatively expensive startup cost can be amortised over the entire runtime of the program, thus reducing the effect of the sorting and resizing which necessarily accompanies inserting new keys.

Concurrently with adding OpenMP parallelism to AMBiT, I replaced all frequently accessed instances of `std::map` with more appropriate data structures. The features of the C++ programming language provide great benefits in this area: the combination of object-oriented design and *templates* for polymorphism [144] means that library authors can expose a high-level application interface, independent of the underlying implementation. We can therefore switch between associative array implementations with minimal or no code rewrites.

There are three associative containers in AMBiT which are important for performance:

1. OrbitalMap: consists of key-value pairs mapping OrbitalInfo objects (essentially metadata

for single-particle orbitals) to Orbitals (data and radial wavefunctions for single-particle orbitals). This structure mainly acts as the domain for (usually nested) loops over orbitals when generating integrals (either Slater or MBPT), so is well suited to a flat map.

2. SlaterIntegralsMap: consists of key value pairs mapping tuples of orbital-multipole moments for a particular two-electron integral to the value of the two-electron integral [12]. Most of the accesses to this data structure come from either writes while calculating two-electron integrals or reads when generating the Hamiltonian matrix. These reads and writes occur in effectively random order (from the perspective of memory caching and speculative execution), so is well suited to the `dense_hash_map` hash table.

3. Wigner 3j symbols: Wigner 3j symbols are calculated extremely frequently, and the same combinations of angular momenta are needed multiple times throughout a given calculation. In order to minimise unnecessary work, we cache results so that subsequent requests for the same symbol need to only look up the result in a hash table rather than do the full calculation. This container is a good fit for `dense_hash_map`, since it contains a relatively small number of terms and is accessed frequently.

After replacing the three most expensive instances of `std::map`, the walltime of the Db calculation described by listing 1 reduces from 7311s to 2876s, for a speedup of 2.54. Calculations in Lr$^+$, Lu$^+$ and Lu and Lr (the results of which are shown later in this thesis) show similar speedups of magnitude ~ 1.5 to ~ 2. This speedup is essentially “free” from a software complexity perspective, as the common API shared by the three associative array implementations means there is almost no re-factoring of the source code when switching between data structures.

3.8 Final performance analysis

The Arm Map profile of the Db calculation given by listing 1 incorporating all of the performance enhancements in this chapter is shown in figure 3.6.

First of all, we can see from the CPU activity panel that AMBiT now has 71.1% OpenMP utilisation, 5.9% OpenMP overhead, 15% serial execution, with the remaining being non-OpenMP overhead. The bottom-most panel below the CPU activity breakdown shows which sections of code correspond to the individual chunks of time in the profile, which I have limited to only show the most important functions. From earliest to latest in the calculation run-time, the functions which dominate the CPU activity are:

1. Dirac-Fock calculation of single-particle orbitals: serial execution, accounts for $< 1\%$ of the total program run-time.

2. Two-electron Slater integrals: accounts for 2% of the total program run-time, but only has $\sim 5\%$ OpenMP operations, with the rest of the CPU time taken up by OpenMP overhead from thread synchronisation.

[12] We incorporate MBPT corrections via modification of the Slater-Coulomb integrals, so there is only one data structure which holds all two-body integrals. This is also true for one-body integrals, but the total time spent accessing one-electron integrals is small enough that there is minimal benefit from optimisation

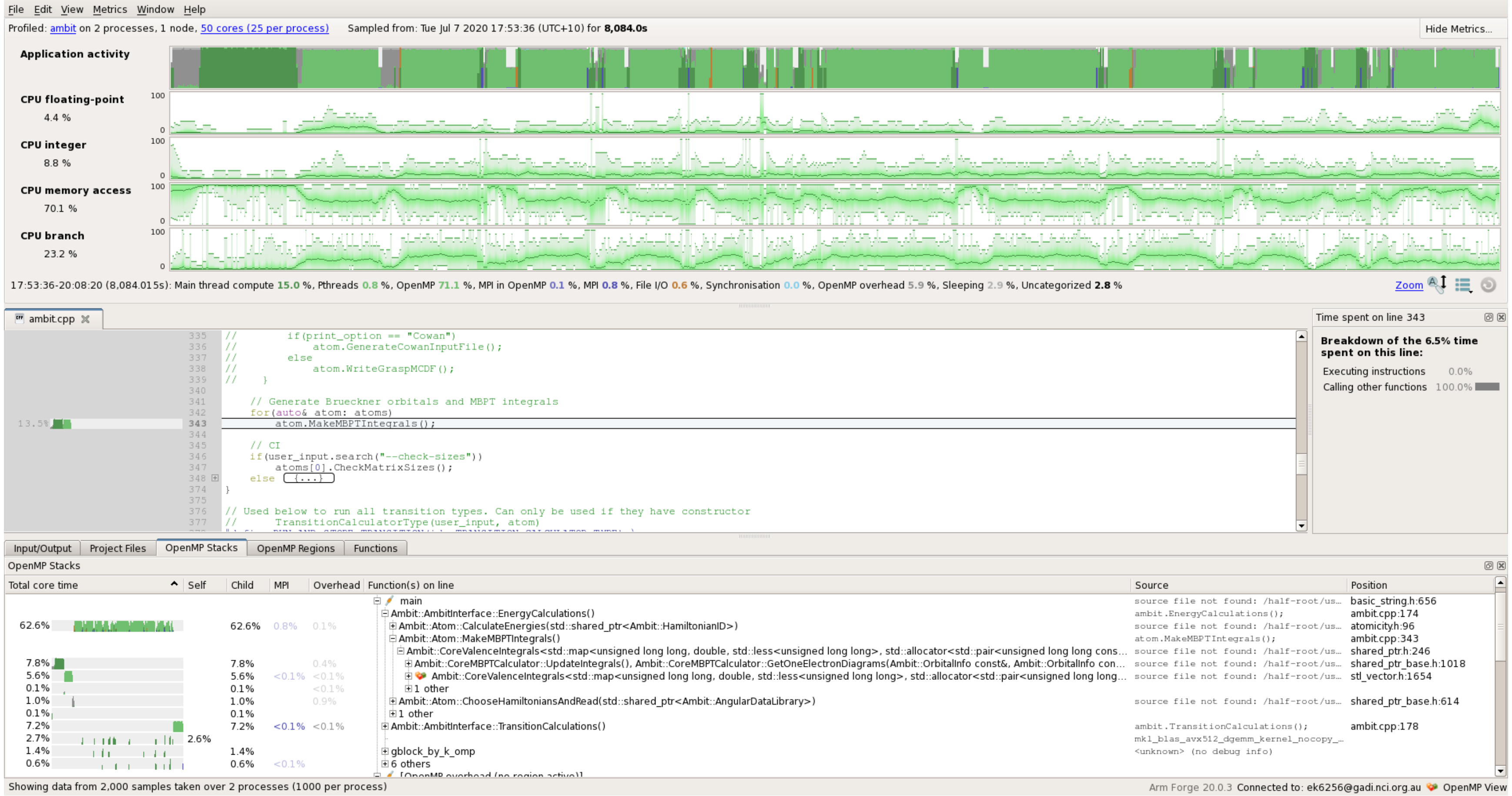
3. One-electron MBPT integrals: serial execution, accounts for 7% of the total program run-time.
4. Two-electron MBPT integrals: 88% OpenMP parallel operations, with the remaining CPU time coming from various forms of overhead. Accounts for 7% of the total program run-time.
5. Generating and solving the CI matrices: sections of $\sim$ 100% OpenMP utilisation while generating the CI matrices, interspersed with striations of relatively low parallel efficiency while solving the CI eigenvalue problem with the Davidson algorithm. The total time spent in Davidson is very low compared to the time taken to actually generate the matrix (even for very large matrices), so I have not optimised it beyond the original MPI-enabled implementation. There is, however, some automatic multithreading of the matrix-vector multiplications when AMBiT is configured to use Intel MKL as its linear algebra engine.

   In the rare case where we need to calculate the full spectrum of a relatively large matrix, which would result in a lot of time being spent in the Davidson algorithm, we can instead turn to the MPI-parallelised ScaLAPACK library, which is more efficient for direct, full diagonalisation of large matrices.
6. Transition matrix elements: close to 100% OpenMP utilisation, accounts for $\sim$ 5% of the calculation's total run-time.

Figure 3.6 also shows a breakdown of *where* the CPU time in a typical AMBiT calculation gets spent. The four bars below the "Application Activity" timeline show a breakdown of the amount of time spent in floating-point and integer arithmetic, memory accesses, and branching logic operations. The green lines within the bar representing the average activity per thread for a particular operation, while the broader green regions show the per-thread minimum and maximum activities. The percentages to the side of the bars show the total percentage of the run-time spent in each type of activity (sampling uncertainty means the percentages sum to slightly more than 100%, this is a limitation of the Arm tools).

We can immediately see that for most of the calculation, very little time is spent in arithmetic operations, with floating-point and integer operations making up a combined total of 13% of the total operations. While this is not necessarily a bad thing for performance on standard HPC nodes, it does limit our ability to take advantage of advanced vector arithmetic operations present in modern CPUs. This leaves a significant fraction of the performance improvements in recent CPU design unutilised, but is especially problematic if we ever want to port AMBiT to work on GPUs, where it is vital that calculations be dominated by arithmetic operations (referred to as *compute-bound* programs). The bulk of our operations are memory and branching logic, which, as mentioned in section 3.7.5, I have optimised by to use more appropriate data structures.

Finally, figure 3.7 shows the parallel speedup and scaling for one MPI process and up to 48 OpenMP threads on NCI's newer *Gadi* cluster. These nodes are x86_64 architecture, and consist of two sockets with 24 cores per socket (see, for example `https://nci.org.au/our-systems/hpc-systems`); the relatively large number of cores per node provides a strong test of the parallel scalability of my OpenMP additions.



Profiled: ambit on 2 processes, 1 node, 50 cores (25 per process) Sampled from: Tue Jul 7 2020 17:53:36 (UTC+10) for **8,084.0s**

17:53:36-20:08:20 (8,084.015s): Main thread compute 15.0 %, Pthreads 0.8 %, OpenMP 71.1 %, MPI in OpenMP 0.1 %, MPI 0.8 %, File I/O 0.6 %, Synchronisation 0.0 %, OpenMP overhead 5.9 %, Sleeping 2.9 %, Uncategorized **2.8** %

```
//        if(print_option == "Cowan")
//            atom.GenerateCowanInputFile();
//        else
//            atom.WriteGraspMCDF();
//    }

    // Generate Brueckner orbitals and MBPT integrals
    for(auto& atom: atoms)
13.5%     atom.MakeMBPTIntegrals();

    // CI
    if(user_input.search("--check-sizes"))
        atoms[0].CheckMatrixSizes();
    else {...}
}

// Used below to run all transition types. Can only be used if they have constructor
//      TransitionCalculatorType(user_input, atom)
```

Time spent on line 343

Breakdown of the 6.5% time spent on this line:

Executing instructions	0.0%
Calling other functions	100.0%

Total core time	Self	Child	MPI	Overhead	Function(s) on line	Source	Position
					main	source file not found: /half-root/us...	basic_string.h:656
					Ambit::AmbitInterface::EnergyCalculations()	ambit.EnergyCalculations();	ambit.cpp:174
62.6%		62.6%	0.8%	0.1%	Ambit::Atom::CalculateEnergies(std::shared_ptr<Ambit::HamiltonianID>)	source file not found: /half-root/us...	atomicity.h:96
					Ambit::Atom::MakeMBPTIntegrals()	atom.MakeMBPTIntegrals();	ambit.cpp:343
					Ambit::CoreValenceIntegrals<std::map<unsigned long long, double, std::less<unsigned long long>, std::allocator<std::pair<unsigned long long cons...	source file not found: /half-root/us...	shared_ptr.h:246
7.8%		7.8%		0.4%	Ambit::CoreMBPTCalculator::UpdateIntegrals(), Ambit::CoreMBPTCalculator::GetOneElectronDiagrams(Ambit::OrbitalInfo const&, Ambit::OrbitalInfo con...	source file not found: /half-root/us...	shared_ptr_base.h:1018
5.6%		5.6%	<0.1%	<0.1%	Ambit::CoreValenceIntegrals<std::map<unsigned long long, double, std::less<unsigned long long>, std::allocator<std::pair<unsigned long long...	source file not found: /half-root/us...	stl_vector.h:1654
0.1%		0.1%		<0.1%	1 other		
1.0%		1.0%		0.9%	Ambit::Atom::ChooseHamiltoniansAndRead(std::shared_ptr<Ambit::AngularDataLibrary>)	source file not found: /half-root/us...	shared_ptr_base.h:614
0.1%		0.1%			1 other		
7.2%		7.2%	<0.1%	<0.1%	Ambit::AmbitInterface::TransitionCalculations()	ambit.TransitionCalculations();	ambit.cpp:178
2.7%	2.6%					mkl_blas_avx512_dgemm_kernel_nocopy_...	
1.4%		1.4%			gblock_by_k_omp	<unknown> (no debug info)	
0.6%		0.6%	<0.1%		6 others		

Showing data from 2,000 samples taken over 2 processes (1000 per process)

Arm Forge 20.0.3 Connected to: ek6256@gadi.nci.org.au

Figure 3.6: Final profile of AMBiT with OpenMP and new data structures, as generated via Arm Map. The calculation used the input file in listing 1, and used a single node with 48 threads.

I have deliberately not included MPI parallelism in this specific benchmark to ensure it only measures the efficacy of my OpenMP modifications. The calculation displays approximately linear scaling up to 8 threads, but the speedup drops off after $\sim$ 24 threads, and flattens out as we approach a full 48 threads per node. This graph is somewhat noisy due to the effects of running on shared-user HPC systems [13], but the overall trends in performance are clear. The diminishing returns likely result from memory contention among large numbers of threads, especially once they spread across socket boundaries.

To alleviate possible memory contention and performance hits from accessing memory across socket-boundaries, we can instead allocate two MPI processes per node to ensure that each MPI process's threads are confined to one socket (I will refer to this as per-socket binding). The OpenMPI installation on *Gadi* is configured to only allow for per-socket binding when using all available CPUs on a socket, so this case is limited to 48 total threads. The speedup for this per-socket binding, also shown in figure 3.7, shows increased performance when compared to using one MPI process per node, as expected. With this in mind, I have employed the per-socket topology of MPI+OpenMP parallelism for the calculations in the rest of the thesis. Even though future nodes may have different quantitative scaling behaviour, the basic scaling for large numbers of OpenMP threads will likely still hold, so running with one MPI process per socket will still be a good default binding.

Looking at the parallel scaling as we move to multiple nodes, we see approximately linear speedup for up to four nodes (192 cores), with diminishing returns for larger numbers of nodes, before levelling off at a maximum speedup of 86.8 for nine nodes (432 cores). This maximum speedup is currently more than sufficient for our needs (for the Dubnium example in this chapter, it reduces the calculation walltime from $2.2 \times 10^5 s \approx 2$ days in serial to $2.5 \times 10^3 s \approx 30$ minutes), in addition to allowing access to a pool of $>$ 1TB of memory across 9 nodes. Furthermore, a rough analysis via Amdahl's law indicates that a large fraction of the code is parallelised. If we take the speedup for 9 nodes as the asymptotic maximum S_∞, then Amdahl's law indicates that the parallel fraction p will be at most [145]:

$$\begin{aligned} p &= 1 - \frac{1}{S_\infty} \\ &= 1 - \frac{1}{86.8} \\ &= 0.98 \end{aligned} \tag{3.5}$$

This is obviously a rough approximation, but shows that I have achieved a relatively high level of parallelism, even given the limitations I have previously outlined. In order to achieve larger speedups it will be necessary to parallelise the remaining serial sections of the code, but the parallelism in the current version still provides large performance benefits (even for much larger calculations than those in this thesis).

[13] If a calculation requires less than a full node's worth of resource then the job scheduling algorithm on Gadi will attempt to fit multiple (potentially unrelated) jobs on a single node. This can cause unpredictable performance impacts, resulting in a slightly noisy measurement of parallel speedup.

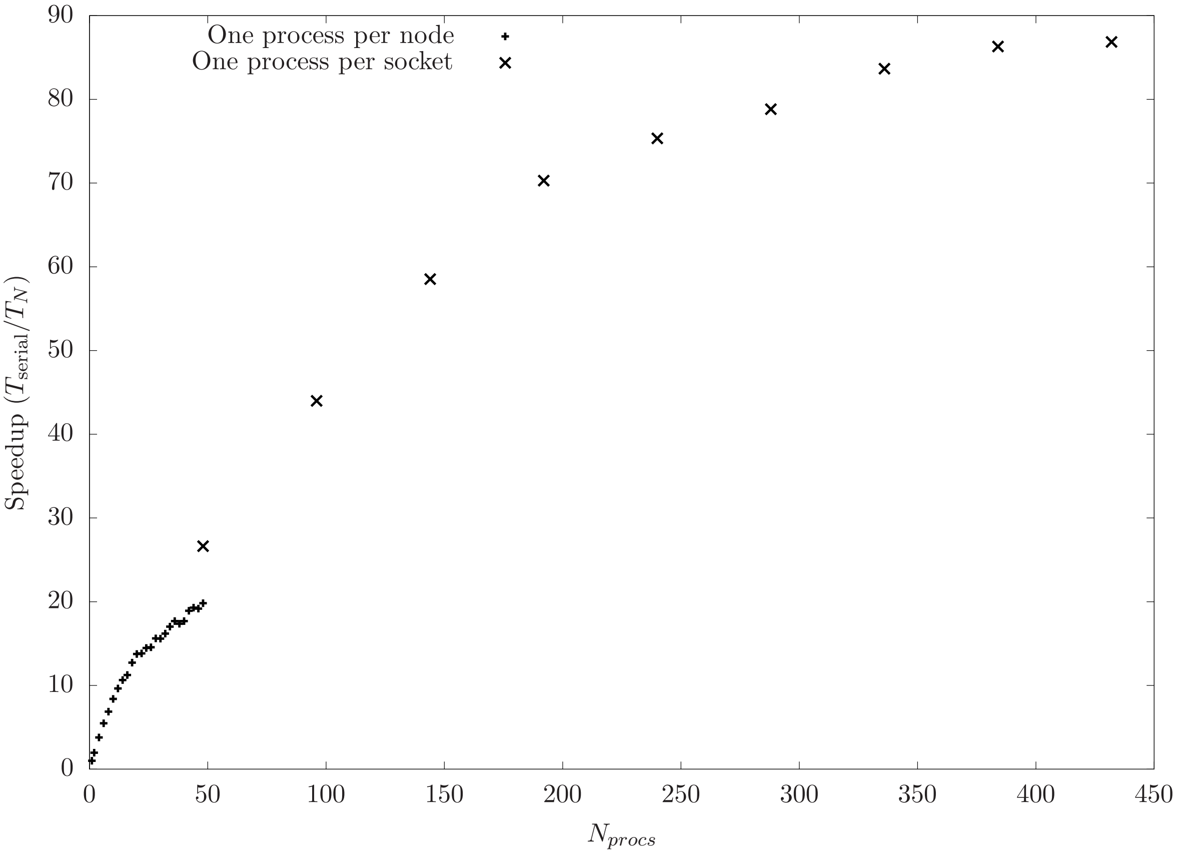


Figure 3.7: Parallel speedup of CI+MBPT calculations for Db for multiple threads on a single node on *Gadi*. The calculation was run with one MPI process (binding per node) and multiple OpenMP processes. Here, we use the formula Speedup $= \frac{T_{\text{serial}}}{T_N}$ for N OpenMP threads.

The profile in figure 3.6 shows two obvious scalability bottlenecks. First, the one-body MBPT integrals do not currently have any OpenMP parallelism. Initially, this function did not seem to take up a large enough percentage of the execution time to warrant adding OpenMP; it was a small enough fraction to not even appear in the profiles in figures 3.2 and 3.3. However, the proportion of the run-time taken up by the obvious hotspots (such as two-body MBPT integrals) decreased after they were parallelised, thus shifting the bottlenecks to these smaller, serial sections of the code.

Secondly, the OpenMP overhead when calculating the two-body Slater integrals is quite large *within* that subroutine, and can potentially become significant for systems which spend a large proportion of their time calculating Slater integrals compared to generating the CI matrix (for example, two-electron systems where MBPT corrections are less important, such as helium). Even in dubnium, where the Slater integrals are a relatively small percentage of the calculation, this overhead still limits scalability when dealing with a large number of OpenMP threads.

There is an additional limiting factor to the parallelism, which is not immediately obvious from the profiling: duplicated work across MPI processes. By definition, any work which is not in some way split between MPI processes will not contribute to the speedup when running across multiple nodes, even if it is parallelised *within* the node via OpenMP. Currently, the subroutine to calculate Slater integrals has no MPI parallelism, so must be calculated independently and in full by every node. The fact that it is serial from the perspective of multi-node parallelism, combined with its large OpenMP overhead, indicates that the Slater integrals subroutine is probably responsible for the parallel speedup levelling off at 86 for the relatively small calculation size of nine nodes.

The rest of this thesis consists of atomic structure calculations using AMBiT, both to validate the numerical accuracy and convergence of the new, parallel version of AMBiT, as well as calculations carried out to assist current and future experimental projects.

Chapter Four

Calculations: Benchmarks — Cr^+ and Db

4.1 Cr^+

Disclaimer: This section is taken mostly verbatim from the Computer Physics Communications paper on AMBiT *[94], on which I am the primary author.*

In this section, I will present results of a full CI+MBPT calculation for the spectrum of Cr^+ using the MPI+OpenMP version of AMBiT. This calculation first appeared in Ref. [94] as an example of the usage of AMBiT, but also serves as a powerful benchmark for the accuracy and scalability of our code, since Cr^+ is a five-electron system which requires a large CI basis and both core- and valence-MBPT corrections in order to produce an accurate spectrum.

These calculations utilise emu CI and demonstrate that this technique allows for significantly higher accuracy than the previous best calculations, using the MPI-only version of AMBiT [102].

Energy levels from these calculations are shown in table 4.1, along with comparison with experimental spectra from [146].

4.1.1 Dirac-Fock and B-spline basis

The calculation presented in this section was undertaken using a V^N Dirac-Fock potential, including all $3d^5$ valence electrons in the DF potential. This choice of potential necessarily introduces subtraction diagrams to MBPT, but these did not significantly degrade the accuracy of the calculations due to the large-CI basis employed throughout. We generated a frozen core consisting of filled 1s, 2s, 2p, 3s and 3p shells and included 3d valence orbitals above the Fermi level, as described in chapter 2. Additionally, we generate all single-particle B-Spline orbitals up to 16spdf for pure CI and 30spdfg for MBPT using the V^N Dirac-Fock operator.

The choice of V^N potential produces 3d orbitals which are spectroscopic for the $3d^5$ configuration, but which are less accurate at treating the $3d^4$ 4s configurations. Forming 3d orbitals with five electrons results in a Dirac-Fock wavefunction which is less tightly bound to the nucleus and thus has higher energy than the corresponding 3d^4 DF orbital. Although this means our calculations tend to underestimate the energy of the levels with 3d^4 4s configuration, we find that use of the V^{N-1} potential is much more slowly convergent for the ground-state 3d^5 wavefunction, resulting in significantly reduced accuracy compared to our choice of V^N potential.

4.1.2 Large-scale calculation

The large-scale calculation targets even-parity states with $J = 1/2, 3/2, 5/2, 7/2, 9/2$. First, we choose the leading-configurations (from which the electrons will be excited to form the CI basis) according to the following rules of thumb:

1. Start with all configurations of experimental/theoretical interest.
2. Add all configurations which can be reached by one-electron excitations from the reference configurations such that $\Delta l = \pm 1$ (i.e. dipole-allowed transitions) and $n = 0, \pm 1$.
3. Add all configurations which can be reached by two-electron excitation from the reference configurations such that $\Delta l = \pm 1$ (i.e. dipole allowed transitions) and $n = 0, \pm 1$.

This process is a rough heuristic, but nonetheless tends to ensure that the most important configurations are included when building the CI matrix, due to the large contributions from dipole-accessible states. However, each set of leading configurations introduced by this procedure increases the computational cost of the CI-calculation (sometimes significantly when including orbitals with large orbital angular momentum l), so we check the change in energy for small CI-basis calculations after each step, and only keep the leading configurations which make large contributions to the energy.

In the case of Cr$^+$, we form the large-side from all single- and double- excitations from the 3d^5, 3d^{4}4s and 3d^{4}4p leading configurations up to 15spdf. We shall refer to this kind of calculation as emu CI. Including 3d^{4}4f in the leading configurations only changes the energy by a small amount but significantly increases the size of the resulting CI matrix, so we have not included it.

We then form the small-side from all single-excitations up to 15spdf and single- and double-excitations up to 5spdf, corresponding to all dipole-accessible transitions from the leading configurations. Our tests indicate that increasing the limit on the maximum principal quantum number n beyond 5 does not significantly improve the accuracy of the calculation.

See listing 3 for the full input file used to generate the emu CI-only spectrum.

Our approach accurately captures the important configurations in the CI expansion at the expense of significantly larger matrix sizes. The largest matrix ($J = 5/2$) has 374944 × 109779 non-zero elements and requires approximately 550GB of memory, which, while large, is still much less than CI with a "full" matrix and is still within the capabilities of modern HPC clusters.

Table 4.1 shows the calculated spectra from these large-scale CI-only calculations, as well as the effects of core-valence MBPT with a basis of 30spdfg. The CI-only calculations in table 4.1 give excellent agreement with experiment, with average errors at the 1% level. However, somewhat unexpectedly, the inclusion of MBPT slightly degrades the calculation's accuracy. Further in-

```
ID=CrII
Z=24

[Lattice]
NumPoints=1000
StartPoint=1.0e-6
EndPoint=60.0

[HF]
N=23
Configuration='1s2 2s2 2p6 3s2 3p6 : 3d5'

[Basis]
--bspline-basis
ValenceBasis=15spdf
FrozenCore=3sp
BSpline/Rmax=60.0

[CI]
LeadingConfigurations='3d5, 3d4 4s1, 3d4 4p1'
ElectronExcitations=2
HoleExcitations=0
EvenParityTwoJ='1, 3, 5, 7, 9, 11'
NumSolutions=6

[CI/SmallSide]
LeadingConfigurations='3d5, 3d4 4s1, 3d4 4p1'
ElectronExcitations='1,15spdf, 2,5spdf'
```

Listing 3: AMBiT input file used to generate Cr^+ emu CI.

creasing the size of the MBPT basis via the *MBPT/Basis* does not increase the accuracy beyond that of the CI-only calculation.

This behaviour suggests that CI+MBPT has qualitatively different convergence at regimes close to saturation of the CI expansion for open-shell atoms, although we have not yet fully characterised this phenomenon or its numerical cause. It could be the result of CI being more robust against residual uncertainties in DF wavefunctions than the MBPT expansion, or of "large" (but still < 1) MBPT diagrams hindering convergence, but a more detailed analysis of the convergence properties will be needed to determine the cause and potential solution to this unexpected numerical behaviour.

4.2 Db – CI+MBPT convergence

Disclaimer: This section includes some results and diagrams from Ref. [113]. I have not included any of the written content from the paper, and new calculations using holes and triple-excitations are entirely my own work.

Similarly to Cr$^+$, neutral dubnium (Db, Z = 105) and tantalum (Ta, Z = 73) serve as an effective benchmark of the computational scaling and accuracy of AMBiT – Db and Ta are both open-shell systems with five valence electrons, but are neutral systems, and thus have relatively higher contributions from electron correlation than the Cr$^+$ ion, in addition to larger relativistic effects due to their higher nuclear charge. Unless otherwise specified, the results in this section were previously published in Ref. [113] - where we showed that the combination of emu CI+MBPT with my work overhauling the parallelism in AMBiT allowed us to reach full saturation of the CI basis in these complicated systems.

Figures 4.1 and 4.2 show the convergence of CI+MBPT calculations in Ta for even- and odd-parity states, respectively. In both figures, the limit for single- and double-excitations on the large-side of the CI-matrix, as well as single-excitations on the small-side, was raised from 11spdf to 21spdf, while the small-side was set to include only double-excitations up to 6sp5d. The MBPT basis was also kept constant at 30spdfgh.

For both parities, we see that the CI+MBPT energies for low-lying states monotonically converge to the "saturated value" with the relative difference between the energies for successive matrix sizes (e.g. moving from 19spdf to 21spdf) rapidly converging to below 1%, indicating that further increasing the maximum PQN in the basis would not increase the accuracy of the calculations any further. Finally, the CI-energies change by less than 1% when including the full CI matrix (i.e. without the emu CI approximation) [1], indicating that the approach to constructing the small-side outlined here and in the previous section effectively captures the most important configurations in the CI-expansion.

The convergence graphs in figures 4.1 and 4.2 demonstrate that we can actually reach saturation of the CI+MBPT expansion for single- and double-excitations without exhausting our computational resources. Reaching saturations would not have been possible without the OpenMP work

[1] The term "full CI" here does not imply that we have included all atomic electrons and excitations, like it does in computational chemistry texts

Table 4.1: Energy levels of Cr^+ (in cm^{-1}) for large-scale emu CI calculation. The "large-side" of the matrix contains single- and double-excitations up to 16spdf, while the "small-side" contains single-excitations up to 15spdf and single- and double-double-excitations up to 5spdf. The first and second columns give the nonrelativistic configuration and approximate LS-coupling term for each calculated level. The E_{CI} and Δ_{CI} columns give the (excitation) energies and difference from experimental values as calculated using a CI-only approach. The E_{MBPT} and Δ_{MBPT} columns show the same comparison for CI+MBPT. The experimental values [146] are shown under E_{Expt}.

Configuration	Term	E_{CI}	Δ_{CI} (%)	$\text{E}_{\text{CI+MBPT}}$	$\Delta_{\text{CI+MBPT}}$ (%)	E_{Expt}
$3d^5$	$^6S_{5/2}$	0	–	0	–	0
$3d^4$ 4s	$^6D_{1/2}$	11956	0.05	11237	6.1	11962
$3d^4$ 4s	$^6D_{3/2}$	12072	-0.3	11341	5.7	12033
$3d^4$ 4s	$^6D_{5/2}$	12265	-1.0	11512	5.2	12148
$3d^4$ 4s	$^6D_{7/2}$	12531	-1.8	11750	4.5	12304
$3d^4$ 4s	$^6D_{9/2}$	12867	-3.0	12048	3.6	12496
$3d^4$ 4s	$^4D_{1/2}$	19441	0.4	19605	-0.4	19528
$3d^4$ 4s	$^4D_{3/2}$	19624	0.04	19771	-0.7	19631
$3d^4$ 4s	$^4D_{5/2}$	19921	-0.6	20043	-1.2	19798
$3d^4$ 4s	$^4D_{7/2}$	20320	1.5	20398	1.9	20024

I described in section 3: these calculations required $\sim 200GB$ or memory distributed across multiple nodes using hybrid MPI+OpenMP parallelism, but would require in excess of 900GB per node to utilise all allocated CPUs if we were to attempt this using only MPI.

The convergence and saturation of the CI basis when using single- and double-excitations does not necessarily imply that we have perfectly characterised the system, however. The energy levels for Ta in Ref. [113] disagree with experimentally determined spectra from Ref. [147] by an average of -1500 cm^{-1} (or $\sim$ 14.5%) for even-parity levels and 1773 cm^{-1} (or $\sim$ 14%) for odd-parity states. However, this is only the average accuracy — some states, notably the even-parity levels in the fine-structure splitting of the $6s^2 5d^3$ ground-state configuration, achieve accuracies on the level of 1%. This indicates that even though the CI energy might not change appreciably by the addition of more configurations (to either the small or large side), it's still possible that we have missed important contributions from, say, triple- or quadruple-excitations, or that we may need to open up the core to hole-excitations out of the 5f shell. As I will show in the rest of this chapter, such effects are marginal when compared to the increased cost of including them in the calculation. My estimates of these two effects show that they are not likely to have a large enough effect to be worth the extra cost of including, however.

Small-scale calculations where I have only included orbitals up to 11spdf indicate that the inclusion of triple-excitations when construction the large-side of the emu CI matrix changes the energy by < 1% for both even and odd parity states in both Ta and Db, while blowing up the memory usage to 480GB — more than double that of the saturated CI calculations with excitations up to 21spdf. On the other hand, including triple excitations up to 9spd on the small-side of the matrix changes the CI energy levels by an average of $\sim$ 1% compared to CI with singles and doubles, but causes the memory usage to increase even further to 587GB, necessitating the use of either specialised node types with large pools of memory or four full normal nodes on NCI's *Gadi* cluster (each normal node has 190GB of memory).

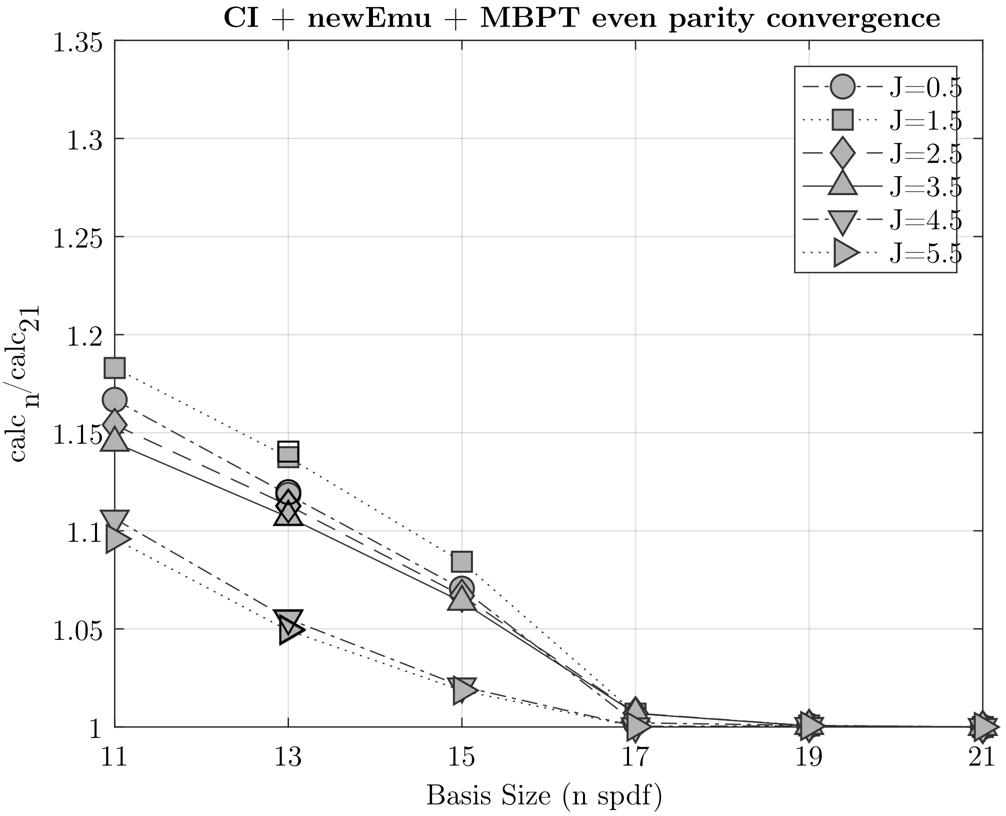


Figure 4.1: CI+MBPT convergence behaviour of low-lying even-parity states in neutral Ta as the CI-basis is increased from 11spdf to 21spdf. Open shapes denote the results of the largest non-Emu CI (i.e. full-matrix) calculations. Figure originally published in Ref. [113] and reproduced with permission.

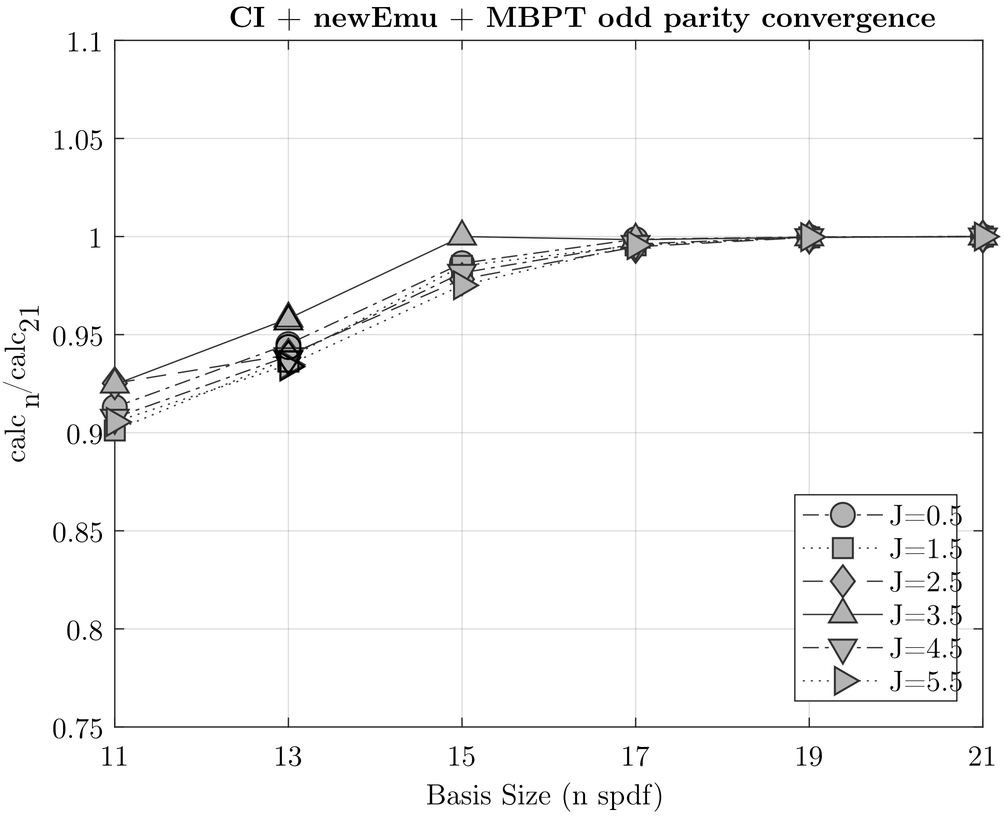


Figure 4.2: CI+MBPT convergence behaviour of low-lying odd-parity states in neutral Ta as the CI-basis is increased from 11spdf to 21spdf. Open shapes denote the results of the largest non-Emu CI (i.e. full-matrix) calculations. Figure originally published in Ref. [113] and reproduced with permission.

Given the large memory requirements, it was not feasible to test the accuracy of triple-excitation calculations with larger basis sets, but given the relatively small contribution of low-lying triple-excitations it is unlikely that including triples in a full-size calculation would significantly improve the accuracy beyond that of the calculations in Ref. [113].

Quantifying the effect of including valence-holes on CI convergence is more difficult, as it causes the CI matrix to rapidly reach unmanageable sizes, even with the OpenMP improvements. The nf shell is the closest filled shell to the Fermi level and is going to have the largest contribution to the CI energy, but will have a large number of CSFs for even single-hole excitations due to the large number of angular momentum sub-levels in orbitals with large l.

Small single-hole CI+MBPT calculations in Db and Ta, where I allowed excitations up to 11spdf on the large side of the matrix, change the energy by an average of $\sim 2\%$ for even parity-states and $\sim 5\%$ for odd-parity states. This indicates that including core-excitations does marginally increase the accuracy beyond the closed-core CI+MBPT, but even this relatively small hole-CI calculation required 2.4TB of memory, so a larger calculation would be extremely challenging to perform. Consequently, it is only worth including hole-excitations in the CI expansion if extremely high-precision is required. They were not necessary to calculate the field-shift coefficients in Ref. [113] to a satisfactory accuracy.

However, it is worth stressing that, in principal, the MPI+OpenMP parallelism in AMBiT allows these expensive hole-CI calculations to be carried out on HPC clusters. Terabytes of memory consumption is large, but not outside the computational resources available on modern HPC clusters. It would have been effectively impossible to carry out these triple-excitation and hole-CI calculations under the MPI-only paradigm, as the code could not scale to the large numbers of nodes necessary to treat the extremely large bases.

4.3 Conclusion

The benchmark calculations in this chapter show that not only does AMBiT scale very effectively to large CI+MBPT calculations across multiple nodes, it can handle basis sizes large enough to numerically saturate CI calculations in complicated, open-shell systems. Furthermore, the close accuracy between calculations and tabulated experimental values of the spectra of Cr^+ and Ta show that we are able to treat challenging five-valent systems at the sub-10% and even sub-1% accuracy level necessary to support experimental and industrial applications. The remaining chapters in this thesis will detail calculations which are more directly motivated by experimental concerns, but which still require large-scale parallel calculations.

Chapter Five

Calculations: Tin highly-charged ions

5.1 Experimental motivation

A rapidly advancing industrial application of highly-charged ions is the use of atomic plasmas for photolithography in computer chip fabrication. Current chip manufacturing pipelines use the process of photolithography, where device features are etched into the chip via UV light. A detailed description of the mechanics of photolithography can be found in Refs. [148, 149] (the broad details of the process have not changed much since the early days of computing), but for the purposes of this chapter the key points are as follows:

- Features on the chip are delineated by a "mask", which selectively blocks the incident UV light from reaching a light-sensitive chemical layered on top of the semiconductor substrate. The photosensitive chemical can then be treated to either selectively deposits a layer of additional material in the mask pattern, or etch the mask pattern into the chip's material when exposed to the UV light.

- The masking process is limited by diffraction, so the minimum size of the etched features is proportional to the wavelength of UV light.

- Smaller features are usually desirable, as it allows more logical functions to fit in the same area of semiconductor, which, in very broad terms, roughly corresponds to a "faster" chip.

Consequently, it is desirable to UV light with the smallest wavelength possible when etching the chip features [1].

One successful approach to extreme ultraviolet (EUV) photolithography relies on the fact that tin plasmas (Sn, Z = 50) have a very strong emission peak at around 13.5 nm, which in turn is the shortest wavelength of light which can be easily focused using multilayer mirror optics [150]. This relatively strong emission peak is the result of a fortuitous clustering of transitions with the form $4p^6 4d^m \rightarrow \left(4p^6 4d^{m-1} 4f + 4p^5 4d^{m+1} + 4p^6 4d^{m-1} 5p\right)$ [50], which results in intense unresolved

[1] This analysis ignores limiting factors such as energy consumption, heat dissipation, dependability and cost of computer systems, which impose their own limits on the chip feature size beyond the fundamental optical considerations outlined here. Further discussion of such limits can be found in chapter 1 of ref [131].

transition arrays in the sequence of Sn^{8+} to Sn^{14+} charge states. This charge state consistency leads to highly-efficient production of EUV light suitable for photolithography [150, 50, 151].

The Mo/Si multilayer mirrors employed in EUV setups only have significant reflectivity for a narrow < 0.5 nm band around the 13.5 nm target [152, 153], so any light from the plasma source outside this wavelength is effectively wasted, as it will not be focused onto the lithography setup. Consequently, it is crucial that the plasma process be tuned to maximise the production of light within this narrow band of wavelengths. The plasma dynamics which govern EUV light production are extremely complicated, depending on radiation-hydrodynamics over a large range of time-scales; characteristics which require enormous computational resources to provide useful modeling. Leaving aside general plasma dynamics (which are well outside the scope of this thesis), atomic structure effects play an important role in determining emission profiles (both in and out of band), as well as the opacity of the plasma [150]. The important transitions in highly-ionised species of tin tend to have partially-filled 4d shells, as well as valence hole transitions in the 4p shell — both of which are computationally challenging to model with spectroscopic accuracy.

In this chapter, I will describe work I carried out with experimentalist colleagues at the Advanced Center for Nanolithography (ARCNL) and elsewhere which constituted the first optical spectroscopy of the level structure of Sn^{7+} to Sn^{10+} ions, demonstrating the power of the CI+MBPT method for these open-shell systems, as well as its close agreement with experimental results.

5.2 Theory and Calculations

Disclaimer: This section is taken in modified form from Ref. [49]. This is a multi-author paper in collaboration with experimental colleagues at ARCNL and other institutes, however the calculations and comparison to experiment are my own original work so I have included those sections here in more-or-less unmodified form.

The detailed electronic structure of Sn^{7+}–Sn^{10+} was calculated using the *ab initio* AMBiT code which combines configuration interaction and many-body perturbation theory (CI+MBPT). Full details of this method have been presented previously [100, 102, 96], here we explain some of the physics and details relevant to the current calculations of tin ions. A more formal discussion, including mathematical details, may be found in [100]. Atomic units ($\hbar = m_e = e = 1$) are used throughout this section.

In all cases we start with a Dirac-Fock (relativistic Hartree-Fock) calculation in the V^N approximation. In this approximation all N electrons of the tin ion are included in the self-consistency procedure, creating a Dirac-Fock potential and electron orbitals that are optimised for the $[\mathrm{Kr}]\,4d^{\mathrm{m}}$ ground-state configuration. This is particularly important for this study because between $m = 4$ and 7, the $4d$ orbitals pass through the half-filled shell ($4d^5$), in which the exchange contribution is maximal. We will use Sn^{9+} ($\mathrm{m} = 5$) as a working example.

A large orbital basis is then formed by diagonalising a set of B-splines [115, 116] on the Dirac-Fock operator

$$\hat{h}_{\mathrm{DF}} = c\,\boldsymbol{\alpha}\cdot\mathbf{p} + (\beta - 1)m_e c^2 - \frac{Z}{r} + V^N(r) \tag{5.1}$$

The resulting basis is ordered by energy. The lowest few valence orbitals in each wave are close to "spectroscopic", while the higher energy orbitals, so-called pseudostates, include large contributions from the continuum.

We now form a set of many-body configurations for the CI method. The CI basis includes all configurations formed by allowing single and double excitations from the $4d^5$ ground-state configuration up to $8spdf$ orbitals (i.e. including $5s - 8s$, $5p - 8p$, $4d - 8d$, and $4f - 8f$ orbitals). The configurations included in CI are within the subspace P; all others are within the subspace Q. For each configuration we generate a complete set of projections (specifying the total angular momentum and projection of each electron in the configuration), which we diagonalise over the $\hat{J}^2$ operator to obtain configuration state functions (CSFs). The CSFs are diagonal in total angular momentum, projection, and relativistic configuration, and they form the CI basis which we denote $|I\rangle$. All CSFs corresponding to configurations in the subspace P are included in CI.

We express the many-electron wavefunction ψ as a linear combination of CSFs from the subspace P,

$$\psi = \sum_{I\in P} C_I \left|I\right\rangle .$$

where the C_I are obtained from the matrix eigenvalue problem. The Hamiltonian for the CI problem is

$$\hat{H} = E_{\mathrm{core}} + \sum_i \hat{h}_{\mathrm{CI}} + \sum_{i<j} \frac{1}{|\mathbf{r}_i - \mathbf{r}_j|} \tag{5.2}$$

where the indices i and j run over the valence electrons only. Note that the one-body operator $\hat{h}_{\mathrm{CI}}$ is not equal to the Dirac-Fock operator: $\hat{h}_{\mathrm{CI}}$ has a potential term $V^{N_{\mathrm{core}}}$ due to the core electrons only. Therefore the basis orbitals are not eigenvalues of the one-body CI operator, which must then be included explicitly.

Because the size of the CI matrix grows rapidly with the inclusion of additional orbitals, we must account for these configurations using many-body perturbation theory. The matrix-eigenvalue equation for the combined CI+MBPT method in second-order of perturbation theory is

$$\sum_{J\in P} \left(H_{IJ} + \sum_{M\in Q} \frac{\left\langle I\right| \hat{H} \left|M\right\rangle \left\langle M\right| \hat{H} \left|J\right\rangle}{E - E_M} \right) C_J = EC_I \tag{5.3}$$

where the CSFs $|M\rangle$ belong to configurations outside of the subspace P (they are in the subspace Q, complementary to P).

Because of the extremely large number of CSFs in the subspace Q, it is prohibitively expensive computationally to modify all matrix elements H_{IJ} directly. Instead, the CI+MBPT method includes (5.3) by modifying the radial integrals of the one and two-body matrix elements [95]. The Slater-Condon rules for calculating matrix elements of Slater determinants ensure that this is equivalent to (5.3), except for the energy denominator (this is a complex discussion beyond the scope of this work, see [95, 123, 100]). Because in this work $\hat{h}_{\mathrm{DF}} \neq \hat{h}_{\mathrm{CI}}$ we must include so-called 'subtraction diagrams' with terms proportional to $\hat{h}_{\mathrm{CI}} - \hat{h}_{\mathrm{DF}}$. These can become very large when there are many valence electrons (since $V^{N_{\mathrm{core}}} - V^N$ is large) but there is cancellation between some of the largest subtraction diagrams and the three-body MBPT operator [102]. For this reason it is important to include three-body operators when calculating these tin ions. An alternative is to calculate the orbitals in the V^{N-m} approximation (equal to $V^{N_{\mathrm{core}}}$) as suggested in [112], however in this case the orbitals are much further from spectroscopic and the CI basis must be made much larger to correct them. In this work all one, two, and three-body second-order diagrams are included.

Until recently, only core-valence correlations were taken into account using MBPT; these include the effects of configurations $|M\rangle$ which include an excitation from the N_{core} electrons. Recently [96] showed that valence-valence correlations could also be included in the same manner. In the current work we include valence-valence correlations with excited orbitals up to $30spdfg$; this includes the effect of configurations that have one or two pseudo-orbitals above $8spdf$, but which have no core excitations. In this work for the first time we include the valence-valence subtraction diagrams presented in [96] (they vanished in that work because $\hat{h}_{\mathrm{DF}}$ was the same as $\hat{h}_{\mathrm{CI}}$).

Finally, Breit and Lamb shift corrections are included. The latter include the vacuum polarization (Uehling) [68] and self-energy [67] corrections in the radiative potential formulation of Flambaum and Ginges [66]. Because both of these effects originate near the nucleus, they have a fairly constant ratio for all the levels we calculated.

Table 5.1: Energy levels of the Sn^{9+} $4d^5$ configuration (in cm^{-1}) calculated by AMBiT CI+MBPT code. The first column give the approximate LS-term of the calculated energy levels. The CI values give the energy as calculated using only configuration interaction, while the Σ^{core}, Σ^{val}, Breit, and QED are the successive corrections to the CI energy by including core-valence MBPT, valence-valence MBPT, Breit, and QED contributions, respectively. The total energy including all corrections is also presented, as are the available experimentally determined values (see main text).

	Energy (cm^{-1})						
Level	CI	Σ^{core}	$\Sigma^{\mathrm{val.}}$	Breit	QED	Total	Exp
$^6S_{5/2}$	0	0	0	0	0	0	0
$^4G_{5/2}$	39469	-4203	-2141	284	-28	33381	33784
$^4G_{7/2}$	42840	-4593	-1833	11	-2	36421	36874
$^4G_{11/2}$	43706	-4676	-1756	-132	3	37145	37535
$^4G_{9/2}$	44212	-4606	-1734	-120	7	37759	38170
$^4P_{5/2}$	43692	-3649	-2067	60	-4	38032	38315
$^4P_{3/2}$	44398	-3174	-2316	138	-12	39035	39190
$^4P_{1/2}$	47021	-2711	-2281	32	-1	42060	
$^4D_{7/2}$	51789	-4612	-2351	-98	8	44737	44915
$^4D_{5/2}$	55276	-3752	-2521	-106	10	48907	
$^4D_{1/2}$	55286	-3812	-2310	-190	22	48996	
$^4D_{3/2}$	56627	-3340	-2319	-241	25	50753	
$^2I_{11/2}$	62330	-7093	-2270	-110	5	52863	53692

Table 5.1: (continued)

				Energy (cm^{-1})			
Level	CI	Σ^{core}	$\Sigma^{\text{val.}}$	Breit	QED	Total	Exp
$^2I_{13/2}$	65768	-7344	-2186	-318	18	55937	56792
$^4F_{7/2}$	66988	-5849	-3102	60	-9	58088	58487
$^2D_{5/2}$	65152	-3732	-2808	-150	18	58479	58756
$^4F_{3/2}$	65795	-4189	-3004	-17	4	58588	58891
$^4F_{9/2}$	67897	-5777	-3104	-33	-2	58981	59417
$^4F_{5/2}$	71298	-4933	-2901	-193	20	63291	63643
$^2H_{9/2}$	74999	-5532	-3005	-207	17	66273	66824
$^2G_{7/2}$	75308	-4572	-3146	-292	27	67325	67698
$^2D_{3/2}$	76386	-4767	-3007	-325	34	68321	
$^2F_{7/2}$	80012	-6786	-3048	-351	32	69859	70199
$^2F_{5/2}$	81165	-5771	-3713	-170	19	71529	71806
$^2H_{11/2}$	82714	-5812	-2746	-527	45	73674	74311
$^2F_{7/2}$	85363	-6347	-3651	-323	31	75073	75470
$^2G_{9/2}$	85188	-6283	-3135	-465	42	75347	75795
$^2F_{5/2}$	90363	-7289	-4328	-145	16	78616	78700
$^2S_{1/2}$	87288	-5338	-2910	-647	64	78457	
$^2D_{3/2}$	99595	-6555	-4503	-149	18	88405	88649
$^2D_{5/2}$	102913	-6472	-4465	-373	37	91640	91927
$^2G_{9/2}$	111086	-8615	-4736	-273	23	97485	98217
$^2G_{7/2}$	112328	-8403	-4689	-341	32	98927	99649

Sn^{9+} has a half-filled $4d$-shell, and for this ion the results are presented broken down into different contributions (Table 5.1). The MBPT corrections are separated into core-valence contributions, Σ^{core} (which correspond to unfreezing of the $4sp3d$ core), and valence-valence contributions, Σ^{val} (introduced in Ref. [96]), which account for configurations that include orbitals above $8spdf$. The column marked QED shows the vacuum polarization and self-energy corrections.

The Sn^{9+} and Sn^{10+} ions were treated with CI+MBPT calculations using only electron excitations (the approach of Refs. [95, 100]). However, as the number of valence electrons increases, this electron-only approach becomes inaccurate due to very large contributions from the subtraction diagrams. To avoid this inaccuracy, the particle-hole CI+MBPT calculations are instead used for the Sn^{7+} and Sn^{8+} ions. This approach, described in Ref. [96], places the Fermi level above the $4d$ shell and treats the $4d^m$ ground-state configuration as a corresponding number of valence holes in an otherwise filled shell. That is, the one-body CI operator includes the potential due to a completely filled $4d$ shell, $V^{N_{\text{core}+10}}$.

At the CI level, the electron-only and particle-hole calculations produce identical spectra. However, the accuracy of the two approaches diverges substantially once MBPT correlations are included. Electron-only CI+MBPT calculations are significantly less accurate for Sn^{8+} and Sn^{7+} ions due to the much larger contributions from subtraction diagrams than in the particle-hole calculations [96]. Consequently, all CI+MBPT results for Sn^{7+} and Sn^{8+} are calculated in the particle-hole framework.

Table 5.2: Mean differences and standard deviation between our CI+MBPT calculations and experiment for measured transitions in different Sn ions.

Ion	$\Delta E_{\text{th-exp}}$ (eV)
Sn^{7+}	-0.004 ± 0.014
Sn^{8+}	-0.005 ± 0.023
Sn^{9+}	-0.010 ± 0.026
Sn^{10+}	-0.008 ± 0.034

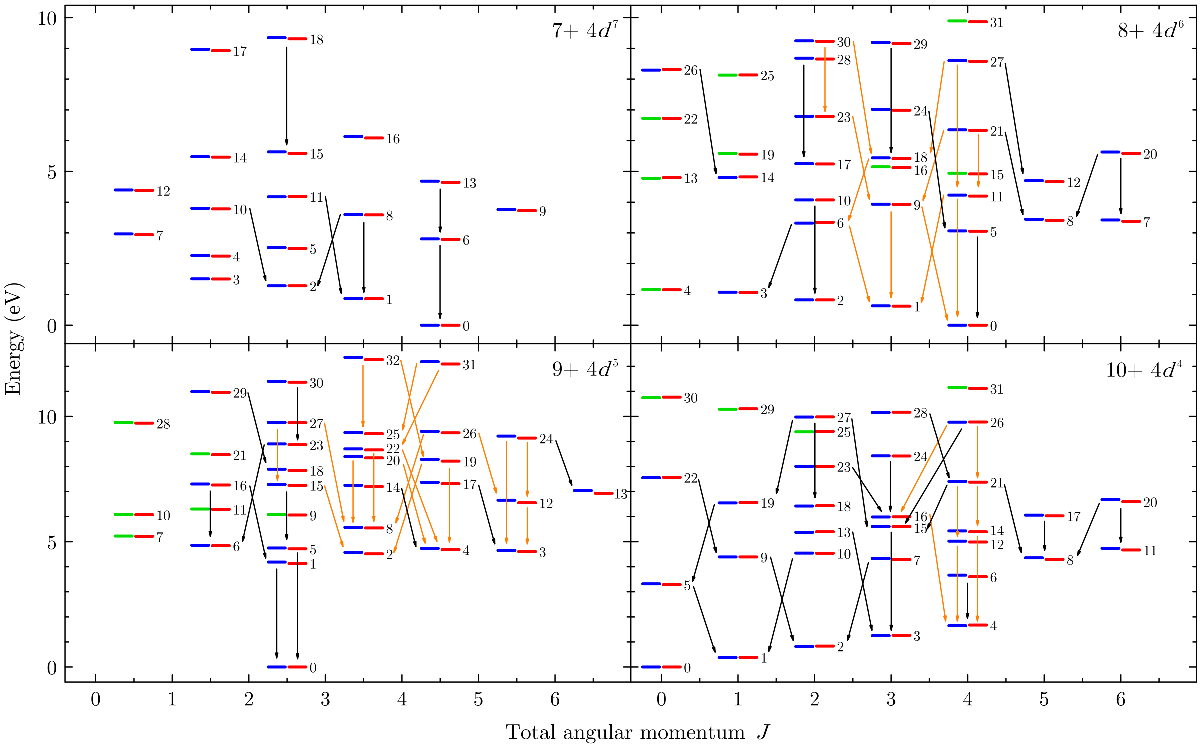


Figure 5.1: Grotrian diagrams for the ions Sn^{7+}–Sn^{10+}. Sn^{7+} energy levels are taken from [154], whereas Sn^{8+}–Sn^{10+} energy levels are results of this work (see Table 5.5). Red lines represent results from AMBiT and blue levels represent experimentally determined levels. Some levels were not experimentally accessible, so green levels represent levels which were obtained via the method of orthogonal parameters (a semi-empirical fitting procedure). The arrows indicate the transition that have been observed experimentally with optical spectroscopy. Red arrows indicate transitions that are part of one or more Ritz combinations, which further confirm the line identifications. [49]

Table 5.3: Experimental vacuum wavelengths λ_{exp} and line intensities of the emissions of Sn^{7+} within its ground electronic configuration [Kr]$4d^7$. The spectra were recorded at the acceleration potential V_{max} of 157 V which yielded the maximum of the fluorescence. The intensities of the experimental lines are given by the integral of the Gaussian fit, corrected for the grating efficiency. The wavelengths λ_{Ritz} are determined from the energy levels of Sn^{7+} given in Ref. [154], and the transition probabilities $gA_{ij,\mathrm{COWAN}}$ are calculated by the COWAN code using the aforementioned experimental energy levels. The wavelengths $\lambda_{\mathrm{CI+MBPT}}$ are calculated by the AMBiT CI+MBPT code. The "Transition" column reports the lower and upper levels of the transitions as indicated in Fig. 5.1. The approximate LS-terms of lower and uppers levels are given in the last column. The numbers between brackets are used to distinguish between the different levels with the same LSJ values. The superscript *bl* marks lines that are affected by blends with other transitions.

λ_{exp}	Intensity	λ_{Ritz}	$gA_{ij,\mathrm{COWAN}}$	$\lambda_{\mathrm{CI+MBPT}}$	Transition	Terms
(nm)	(arb. units)	(nm)	(s^{-1})	(nm)	(see Fig. 5.1)	
333.9^{bl}	132	334.2	167	333.3	15-18	${}^2F_{5/2}$-${}^2D_{5/2}(1)$
374.5	271	374.8	375	372.9	1-11	${}^4F_{7/2}$-${}^2D_{5/2}(2)$
386.0	68	386.1	57	384.6	16-18	${}^2F_{7/2}$-${}^2D_{5/2}(1)$
441.9	142	441.8	262	444.0	0-6	${}^4F_{9/2}$-${}^2G_{9/2}$
454.3	12	454.2	66	455.0	1-8	${}^4F_{7/2}$-${}^2G_{7/2}$
492.2	69	492.2	44	495.8	2-10	${}^4F_{5/2}$-${}^4P_{3/2}$
536.0^{bl}	35	535.9	32	537.5	2-8	${}^4F_{5/2}$-${}^2G_{7/2}$
660.9	116	661.6	170	668.3	6-13	${}^2G_{9/2}$-${}^2H_{9/2}$

Table 5.4: Vacuum wavelengths and line intensities of the emission from the ions Sn^{8+}–Sn^{10+}, determined at the acceleration potential V$_{\text{max}}$ that maximizes the ion fluorescence (found to be the same for Sn^{8+} and Sn^{9+}). The intensities of the experimental lines are given by the integral of the Gaussian fit, corrected for the grating efficiency. The transition wavelengths λ_{orth} are determined from the level energies calculated with the orthogonal parameters method (Table 5.5). The transition probabilities $gA_{ij,\text{COWAN}}$ are determined with the COWAN code using the same level energies. The transition wavelengths $\lambda_{\text{CI+MBPT}}$ are calculated by the AMBiT CI+MBPT code. The "Transition" column refers to the Grotrian diagrams in Fig. 5.1, indicating the lower and upper levels of the transition. The configuration and the approximate LS-coupling terms of the transitions are given in the last two columns (the numbers in brackets distinguish between different levels with the same LSJ values, supplementing a sequential index as defined by Nielson and Koster [155]). The superscript bl indicates lines that are affected by blends with other transitions, while the superscript D marks doubly assigned lines.

Ion	V$_{\text{max}}$	λ_{exp}	Intensity	λ_{orth}	$gA_{ij,\text{COWAN}}$	$\lambda_{\text{CI+MBPT}}$	Transition	Config.	Term symbol
	(V)	(nm)	(arb. units)	(nm)	(s^{-1})	(nm)	(see Fig. 5.1)		
8+	137	283.4	18	284	104	283.1	11-27	[Kr]4d^6	$^3F_4(2)$-$^3F_4(1)$
		293.3	42	293	226	295.2	0-11		$^5D_4$-$^3F_4(2)$
		313.5	17	314	54	315.1	5-24		$^3H_4$-$^1F_3$
		315.0	16	315	60	315.4	0-9		$^5D_4$-$^3F_3(2)$
		317.6	45	318	192	316.6	12-27		$^3G_5$-$^3F_4(1)$
		326.0	15	326	105	325.2	18-30		$^3D_3$-$^3P_2(1)$
		330.4	70	330	279	331.6	18-29		$^3D_3$-$^3F_3(1)$
		344.6	25	344	31	346.5	1-11		$^5D_3$-$^3F_4(2)$
		354.8	33	355	102	352.3	14-26		$^3P_1(2)$-$^3P_0(1)$
		360.2^{bl}	43	360	120	363.5	17-28		$^3D_2$-$^3F_2(1)$
		374.5^{bl}	271	374	131	374.6	1-9		$^5D_3$-$^3F_3(2)$
		381.2	36	381	65	381.3	2-10		$^5D_2$-$^3F_2(2)$
		392.1	24	392	78	392.2	18-27		$^3D_3$-$^3F_4(1)$
		404.6	60	404	239	405.7	0-5		$^5D_4$-$^3H_4$
		426.6	54	428	186	424.0	8-21		$^3H_5$-$^1G_4(2)$
		434.1	36	434	107	434.2	9-23		$^3F_3(2)$-$^1D_2(2)$
		460.9	138	461	315	454.5	1-6		$^5D_3$-$^3P_2(2)$
		505.7	38	505	107	507.3	23-30		$^1D_2(2)$-$^3P_2(1)$
		513.2^{bl}	85	515	89	516.1	9-21		$^3F_3(2)$-$^1G_4(2)$

Table 5.4: (continued)

Ion	V$_{\mathrm{max}}$	λ_{exp}	Intensity	λ_{orth}	$gA_{ij,\mathrm{COWAN}}$	$\lambda_{\mathrm{CI+MBPT}}$	Transition	Config.	Term symbol
	(V)	(nm)	(arb. units)	(nm)	(s^{-1})	(nm)	(see Fig. 5.1)		
		551.8	40	552	65	542.9	3-6		$^5D_1$-$^3P_2(2)$
		560.9	149	560	185	568.9	7-20		$^3H_6$-$^1I_6$
		566.8	91	565	87	576.5	8-20		$^3H_5$-$^1I_6$
		584.4^{D}	118	585	62	599.1	6-18		$^3P_2(2)$-$^3D_3$
		584.4^{D}	118	585	291	581.0	11-21		$^3F_4(2)$-$^1G_4(2)$
9+	137	261.0	66	261	304	262.9	0-5	[Kr]$4d^5$	$^6S_{5/2}$-$^4G_{5/2}$
		272.0	53	272	418	273.8	3-24		$^4G_{11/2}$-$^2H_{11/2}$
		276.6	30	276	209	278.4	4-24		$^4G_{9/2}$-$^2H_{11/2}$
		296.0^{D}	46	296	115	295.2	8-27		$^4D_{7/2}$-$^2F_{5/2}(1)$
		296.0^{D}	46	296	149	299.6	0-1		$^6S_{5/2}$-$^4G_{5/2}$
		304.8	25	303	69	306.2	19-32		$^2H_{9/2}$-$^2G_{7/2}(1)$
		306.6^{bl}	34	306	16	307.7	6-23		$^4P_{3/2}$-$^2F_{5/2}(2)$
		312.1	62	312	151	311.5	4-22		$^4G_{9/2}$-$^2F_{7/2}(1)$
		323.9	50	324	105	326.7	8-26		$^4D_{7/2}$-$^2G_{9/2}(2)$
		333.9^{bl}	70	333	132	335.0	2-19		$^4G_{7/2}$-$^2H_{9/2}$
		338.7^{bl}	219	339	222	338.2	4-20		$^4G_{9/2}$-$^2G_{7/2}(2)$
		349.1^{bl}	271	349	297	350.7	4-19		$^4G_{9/2}$-$^2H_{9/2}$
		356.8	41	354	57	362.0	22-31		$^2F_{7/2}(1)$-$^2G_{9/2}(1)$
		395.5	64	394	131	398.1	8-22		$^4D_{7/2}$-$^2F_{7/2}(1)$
		398.3	68	398	86	396.7	1-16		$^4G_{5/2}$-$^4F_{3/2}$
		399.9	26	401	42	398.2	18-29		$^4F_{5/2}$-$^2D_{3/2}(2)$
		413.5	43	415	133	419.2	25-32		$^2F_{7/2}(2)$-$^2G_{7/2}(1)$
		438.9	46	439	155	442.7	8-20		$^4D_{7/2}$-$^2G_{7/2}(2)$
		439.7	34	441	109	446.2	25-31		$^2F_{7/2}(2)$-$^2G_{9/2}(1)$
		452.4	32	453	105	444.8	12-26		$^2I_{11/2}$-$^2G_{9/2}(2)$

Table 5.4: (continued)

Ion	$V_{\max}$	λ_{exp}	Intensity	λ_{orth}	$gA_{ij,\textsc{cowan}}$	$\lambda_{\mathrm{CI+MBPT}}$	Transition	Config.	Term symbol
	(V)	(nm)	(arb. units)	(nm)	(s^{-1})	(nm)	(see Fig. 5.1)		
		457.0^D	58	456	99	453.4	2-15		$^4G_{7/2}$-$^2D_{5/2}(3)$
		457.0^D	58	456	82	458.0	3-17		$^4G_{11/2}$-$^4F_{9/2}$
		485.0^{bl}	358	484	472	480.5	12-24		$^2I_{11/2}$-$^2H_{11/2}$
		489.2	414	489	562	489.1	5-15		$^4G_{5/2}$-$^2D_{5/2}(3)$
		492.2	107	492	69	491.9	4-14		$^4G_{9/2}$-$^4F_{7/2}$
		497.0	55	496	107	497.3	23-30		$^2F_{5/2}(2)$-$^2D_{5/2}(2)$
		501.4	167	502	339	496.6	15-27		$^2D_{5/2}(3)$-$^2F_{5/2}(1)$
		507.6	105	508	103	511.4	6-16		$^4P_{3/2}$-$^4F_{3/2}$
		570.8	76	569	151	563.8	13-24		$^2I_{13/2}$-$^2H_{11/2}$
		618.9	137	621	119	636.2	3-12		$^4G_{11/2}$-$^2I_{11/2}$
10+	217	283.7	13	283	278	283.0	15-27	[Kr]4d^4	$^3F_3(2)$-$^3F_2(1)$
		284.7	18	285	114	286.3	3-15		$^5D_3$-$^3F_3(2)$
		286.0	33	286	163	287.4	4-16		$^5D_4$-$^3D_3$
		297.4^D	15	297	110	298.8	1-10		$^5D_1$-$^3F_2(2)$
		297.4^D	15	298	91	296.7	15-26		$^3F_3(2)$-$^3F_4(1)$
		300.5	15	300	73	300.2	3-13		$^5D_3$-$^3P_2(2)$
		328.1^D	112	328	137	333.1	4-14		$^5D_4$-$^3G_4$
		328.1^D	112	328	261	327.4	16-26		$^3D_3$-$^3F_4(1)$
		346.8	91	347	278	348.2	2-9		$^5D_2$-$^3P_1(2)$
		349.1^{bl}	271	349	93	349.7	18-27		$^3D_2$-$^3F_2(1)$
		353.0	38	353	82	359.0	2-7		$^5D_2$-$^3G_3$
		361.9	14	361	95	363.1	19-27		$^3D_1$-$^3F_2(1)$
		367.7	145	368	227	374.6	4-12		$^5D_4$-$^3F_4(2)$
		383.7	28	384	84	378.0	5-19		$^3P_0(2)$-$^3D_1$
		392.7	68	392	124	390.6	9-22		$^3P_1(2)$-$^1S_0(2)$

Table 5.4: (continued)

Ion	V$_{\max}$	$\lambda_{\rm exp}$	Intensity	$\lambda_{\rm orth}$	$gA_{ij,\textsc{cowan}}$	$\lambda_{\rm CI+MBPT}$	Transition Config.	Term symbol
	(V)	(nm)	(arb. units)	(nm)	(s^{-1})	(nm)	(see Fig. 5.1)	
		407.4	38	408	87	402.4	8-21	$^3H_5$-$^1G_4(2)$
		421.8	26	421	139	428.3	1-5	$^5D_1$-$^3P_0(2)$
		450.5	18	450	88	444.7	21-28	$^1G_4(2)$-$^3F_3(1)$
		508.2	127	508	193	510.3	16-24	$^3D_3$-$^1F_3$
		520.7	84	521	310	518.6	12-21	$^3F_4(2)$-$^1G_4(2)$
		524.2	17	525	45	516.5	21-26	$^1G_4(2)$-$^3F_4(1)$
		534.4	117	535	234	538.4	8-20	$^3H_5$-$^1I_6$
		614.1^D	95	614	102	643.9	4-6	$^5D_4$-$^3H_4$
		614.1^D	95	616	106	616.1	16-23	$^3D_3$-$^1D_2(2)$
		628.3	69	630	116	626.8	14-21	$^3G_4$-$^1G_4(2)$
		639.9	157	642	201	642.9	11-20	$^3H_6$-$^1I_6$
		689.5^{bl}	85	690	164	697.0	15-21	$^3F_3(2)$-$^1G_4(2)$
		728.1	39	727	103	713.1	8-17	$^3H_5$-$^3G_5$

Table 5.5: Energy levels of the Sn^{8+} $4d^6$, Sn^{9+} $4d^5$, and Sn^{10+} $4d^4$ configurations (in cm^{-1}), ordered according to their total energy, optimised with Kramida's LOPT algorithm [156] from the measured magnetic dipole transitions. The levels are labeled by their approximate LS-coupling terms; the numbers in brackets are used to differentiate between the different levels with the same LSJ values, supplementing a sequential index as defined by Nielson and Koster [155]. The uncertainties x_j ($j = 1-5$) and y are estimated to be the root mean square of the deviation of the orthogonal parameters fitting for the respective configuration: $x_{1,2,3} = \pm 16\,\mathrm{cm}^{-1}$; $x_4 = \pm 41\,\mathrm{cm}^{-1}$; $x_5 = y = \pm 14\,\mathrm{cm}^{-1}$. The dispersive energy uncertainty D_1 is close to the minimum uncertainty of separation from other levels, and the energy uncertainty D_2 is that relative to the ground level of the configuration (for the exact definition, see [156]). N is the total number of lines connected to the level. $E_{\rm orth}$ values are the semi-empirical energy levels, described further in Ref. [49]. The $E_{\textsc{AMBiT}}$ values are the energy levels calculated using the AMBiT CI+MBPT code. The differences between the experimental energy levels and the calculated values are presented in columns $\Delta E_{\rm orth}$ ($E_{\rm exp} - E_{\rm orth}$) and $\Delta E_{\textsc{AMBiT}}$ ($E_{\rm exp} - E_{\rm CI+MBPT}$). The energies determined from previous measurements on vacuum sparks [157] are reported in the column $E_{\rm vs}$, along with the difference $\Delta E_{\rm vs} = E_{\rm exp} - E_{\rm vs}$. The uncertainties in the systematic common shifts of the identified level groups z_i ($i = 1-8$) are quoted in [157] to be of the order of several hundreds of cm^{-1} (see main text for further details). The uncertainty of the level energies within each of the groups was estimated at $10\,\mathrm{cm}^{-1}$.

Ion	Level	Term	$E_{\rm exp}$	D_1	D_2	N	$E_{\rm orth}$	$\Delta E_{\rm orth}$	$E_{\textsc{AMBiT}}$	$\Delta E_{\textsc{AMBiT}}$	$E_{\rm vs}$	$\Delta E_{\rm vs}$
8+ $4d^6$	0	$^5D_4$	0	30	0	2	-5	5	0	0	0	0
	1	$^5D_3$	5 075	13	30	3	5 064	11	5 011	64	5 050	25

Table 5.5: (continued)

Ion	Level	Term	E_{exp}	D_1	D_2	N	E_{orth}	ΔE_{orth}	E_{AMBIT}	ΔE_{AMBIT}	E_{vs}	ΔE_{vs}
	2	$^5D_2$	6 634+ x_2	0	0	0	6 634	0	6 626	8	6 670+ z_1	-36
	3	$^5D_1$	8 648	13	40	1	8 636	12	8 593	55	8 670+ z_1	-22
	4	$^5D_0$					9 345		9 307			
	5	$^3H_4$	24 716	24	24	1	24 726	10	24 651	65	24 685+ z_2	31
	6	$^3P_2(2)$	26 785	16	40	2	26 771	14	27 011	-226		
	7	$^3H_6$	27 592	13	43	1	27 604	-12	27 270	322	27 610+ z_2	-18
	8	$^3H_5$	27 778	22	39	1	27 781	-3	27 503	275	27 710+ z_2	68
	9	$^3F_3(2)$	31 740	12	30	4	31 736	4	31 709	31	31 747+ z_3	-7
	10	$^3F_2(2)$	32 847+ x_2	28	28	1	32 847	0	32 855	-8	33 028+ z_3	-181
	11	$^3F_4(2)$	34 103	11	30	4	34 102	1	33 873	230	34 220+ z_2	-117
	12	$^3G_5$	37 908	40	59	1	37 930	-22	37 616	292	37 950+ z_2	-42
	13	$^1S_0(4)$					38 532		38 684			
	14	$^3P_1(2)$	38 694+ x_1	0	0	0	38 694	0	38 903	-209		
	15	$^3G_4$					39 872		39 674		39 609+ z_2	
	16	$^3G_3$					41 548		41 310			
	17	$^3D_2$	42 340+ x_3	0	0	0	42 340	0	42 287	53	41 787+ z_2	553
	18	$^3D_3$	43 879	17	40	3	43 887	-8	43 704	175		
	19	$^3D_1$					45 061		44 847			
	20	$^1I_6$	45 421	13	41	1	45 399	22	45 032	389	45 440+ z_2	-19
	21	$^1G_4(2)$	51 219	11	30	2	51 217	2	51 085	134	50 840+ z_4	379
	22	$^3P_0(2)$					54 202		54 250			
	23	$^1D_2(2)$	54 777	13	40	2	54 795	-18	54 742	35		
	24	$^1F_3$	56 613	41	47	1	56 586	27	56 385	228		
	25	$^3P_1(1)$					65 561		65 611			
	26	$^3P_0(1)$	66 874+ x_1	32	32	1	66 875	-1	67 067	-193		
	27	$^3F_4(1)$	69 394	23	40	2	69 401	-7	69 198	196	68 566+ z_2	828
	28	$^3F_2(1)$	70 006+ x_3	31	31	1	70 006	1	69 800	206		
	29	$^3F_3(1)$	74 146	37	54	1	74 144	2	73 860	286	73 385+ z_2	761
	30	$^3P_2(1)$	74 552	14	40	2	74 548	4	74 454	98		
	31	$^1G_4(1)$					79 767		79 565		79 186+ z_2	
	32	$^1D_2(1)$					101 675		101 319		99 838+ z_4	
	33	$^1S_0(1)$					131 838		131 874		130 008+ z_4	
9+ $4d^5$	0	$^6S_{5/2}$	0	59	0	1	-17	17	0	0	0	0

Table 5.5: (continued)

Ion	Level	Term	E_{exp}	D_1	D_2	N	E_{orth}	ΔE_{orth}	E_{AMBIT}	ΔE_{AMBIT}	E_{vs}	ΔE_{vs}
	1	${}^4G_{5/2}$	33 784	61	61	1	33 748	36	33 382	402	33 582+z_5	202
	2	${}^4G_{7/2}$	36 874	21	70	2	36 834	40	36 422	452	36 610+z_5	264
	3	${}^4G_{11/2}$	37 535	10	90	2	37 576	-41	37 145	390	37 399+z_5	136
	4	${}^4G_{9/2}$	38 170	20	80	4	38 173	-3	37 759	411	37 958+z_5	212
	5	${}^4G_{5/2}$	38 315	16	59	1	38 282	33	38 031	284	38 110+z_5	205
	6	${}^4P_{3/2}$	39 190	16	68	1	39 183	7	39 035	155	39 010+z_5	180
	7	${}^4P_{1/2}$					42 159		42 061			
	8	${}^4D_{7/2}$	44 915	15	80	4	44 958	-43	44 737	178	44 470+z_5	445
	9	${}^4D_{5/2}$					49 065		48 907			
	10	${}^4D_{1/2}$					49 104		48 996			
	11	${}^4D_{3/2}$					50 861		50 753			
	12	${}^2I_{11/2}$	53 692	8	80	3	53 685	7	52 862	830	53 554+z_5	138
	13	${}^2I_{13/2}$	56 792	12	83	1	56 765	27	55 937	855	56 660+z_5	132
	14	${}^4F_{7/2}$	58 487	16	77	1	58 491	-4	58 088	399	58 300+z_5	187
	15	${}^2D_{5/2}$(3)	58 756	14	60	2	58 721	35	58 479	277	58 370+z_5	386
	16	${}^4F_{3/2}$	58 891	25	66	1	58 848	43	58 589	302		
	17	${}^4F_{9/2}$	59 417	28	87	1	59 469	-52	58 982	435	58 850+z_5	567
	18	${}^4F_{5/2}$	63 643+x_4	0	0	0	63 643	0	63 291	352		
	19	${}^2H_{9/2}$	66 824	22	70	3	66 846	-22	66 273	551	66 427+z_5	397
	20	${}^2G_{7/2}$(2)	67 698	18	80	2	67 687	11	67 326	372	66 975+z_5	723
	21	${}^2D_{3/2}$(3)					68 584		68 322			
	22	${}^2F_{7/2}$(1)	70 199	18	80	3	70 228	-29	69 858	341	70 185+z_5	14
	23	${}^2F_{5/2}$(2)	71 806	43	80	1	71 837	-31	71 530	276		
	24	${}^2H_{11/2}$	74 311	16	80	3	74 338	-27	73 673	638	74 195+z_5	116
	25	${}^2F_{7/2}$(2)	75 470	16	80	2	75 423	47	75 073	397	74 385+z_5	1 085
	26	${}^2G_{9/2}$(2)	75 795	18	80	2	75 816	-21	75 347	448	75 345+z_5	450
	27	${}^2F_{5/2}$(1)	78 700	16	60	2	78 654	46	78 617	83		
	28	${}^2S_{1/2}$					78 719		78 456			
	29	${}^2D_{3/2}$(2)	88 649+x_4	25	25	1	88 702	-53	88 405	244		
	30	${}^2D_{5/2}$(2)	91 927	16	81	1	91 976	-49	91 640	287	90 911+z_5	1 016
	31	${}^2G_{9/2}$(1)	98 217	18	80	2	98 228	-11	97 484	733	96 800+z_5	1 417
	32	${}^2G_{7/2}$(1)	99 649	21	80	2	99 568	81	98 926	723	98 277+z_5	1 372

Table 5.5: (continued)

Ion	Level	Term	E_{exp}	D_1	D_2	N	E_{orth}	ΔE_{orth}	E_{AMBIT}	ΔE_{AMBIT}	E_{vs}	ΔE_{vs}
	33	$^2P_{3/2}$					114 830		114 351			
	34	$^2P_{1/2}$					117 607		117 122			
	35	$^2D_{5/2}(1)$					128 906		128 281			
	36	$^2D_{3/2}(1)$					130 802		130 180			
10+ $4d^4$	0	$^5D_0$	$0+y$	0	0	0	0	0	0	0	0	0
	1	$^5D_1$	3 043	22	0	1	3 043	0	3 141	-98	3 035	8
	2	$^5D_2$	6 590+ x_5	0	0	0	6 590	0	6 717	-127	6 545	45
	3	$^5D_3$	10 073	49	84	1	10 054	19	10 213	-140	10 005	68
	4	$^5D_4$	13 300	24	70	3	13 315	-15	13 516	-216	13 280	20
	5	$^3P_0(2)$	26 752	27	23	1	26 750	2	26 490	262		
	6	$^3H_4$	29 584	15	75	1	29 589	-5	29 046	538	29 380+ z_6	204
	7	$^3G_3$	34 918+ x_5	32	32	1	34 899	19	34 573	345	34 630+ z_6	288
	8	$^3H_5$	35 147	24	73	1	35 143	4	34 639	508	34 814+ z_7	333
	9	$^3P_1(2)$	35 425+ x_5	33	33	1	35 429	-4	35 438	-13	35 048+ z_6	377
	10	$^3F_2(2)$	36 669	61	61	1	36 666	3	36 613	56	36 297+ z_6	372
	11	$^3H_6$	38 232	10	70	1	38 226	6	37 656	576	37 890+ z_7	342
	12	$^3F_4(2)$	40 490	13	70	2	40 475	15	40 208	282	40 130+ z_7	360
	13	$^3P_2(2)$	43 351	44	95	1	43 377	-26	43 530	-179	42 898+ z_6	453
	14	$^3G_4$	43 777	10	70	2	43 765	12	43 539	238	43 710+ z_7	67
	15	$^3F_3(2)$	45 197	8	70	2	45 196	1	45 146	51	44 766+ z_7	431
	16	$^3D_3$	48 279	35	80	2	48 263	16	48 310	-31	47 850+ z_7	429
	17	$^3G_5$	48 881	8	73	1	48 893	-12	48 663	218	48 480+ z_7	401
	18	$^3D_2$	51 801	30	60	1	51 808	-7	51 885	-84		
	19	$^3D_1$	52 814	31	35	1	52 806	8	52 945	-131		
	20	$^1I_6$	53 860	14	74	1	53 867	-7	53 211	649	53 475+ z_7	385
	21	$^1G_4(2)$	59 693	6	70	4	59 684	9	59 493	200		
	22	$^1S_0(2)$	60 890+ x_5	26	42	1	60 870	20	61 041	-151		
	23	$^1D_2(2)$	64 563	15	80	1	64 549	14	64 542	21		
	24	$^1F_3$	67 957	16	80	1	67 958	-1	67 907	50	66 757+ z_8	1 200
	25	$^3P_2(1)$					75 662		75 823			
	26	$^3F_4(1)$	78 771	14	70	3	78 777	-6	78 854	-83		
	27	$^3F_2(1)$	80 446	50	47	1	80 445	1	80 484	-38		
	28	$^3F_3(1)$	81 891	20	71	1	81 881	10	81 982	-91	80 207+ z_8	1 684
	29	$^3P_1(1)$					82 941		83 107			
	30	$^3P_0(1)$					86 664		86 851			
	31	$^1G_4(1)$					89 965		89 627			

Table 5.5: (continued)

Ion	Level	Term	E_{exp}	D_1	D_2	N	E_{orth}	ΔE_{orth}	E_{AMBIT}	$\Delta E_{\mathrm{AMBIT}}$	E_{vs}	ΔE_{vs}
	32	$^1D_2(1)$					112 401		112 544			
	33	$^1S_0(1)$					144 549		145 002			

5.3 Comparison with experimental results

The final results are presented in Tables 5.2, 5.3, 5.4, and 5.5, where predicted energy levels and transition wavelengths calculated via AMBiT are compared to experimentally measured values (as presented in [49]). These results are also presented graphically in figure 5.1, which contains Grotrian diagrams for the theoretical and experimental levels of the ions Sn^{7+} – Sn^{10+}. We restricted our calculations to the fine-structure levels of the ground-state configuration for each ions, of which there were a huge number due to the open d-shells in each ion.

Table 5.2 shows the average differences between the CI+MBPT and experimentally measured energy levels. Calculations for all Sn ions differ by less than 1% on average, showing that we have both correctly identified the levels and included all of the important terms in the CI and MBPT expansions. This can be seen much more neatly in the Grotrian diagrams in figure 5.1, where we can clearly see that our CI+MBPT calculations did not miss a single experimentally accessible level, and we have very close agreement between theory and experiment for all levels (not just on average).

The very good agreement between theory and experiment in this section validates the predictive power of CI+MBPT, particularly the valence-valence MBPT method implemented in AMBiT. Valence-valence MBPT provides a high level of accuracy for relatively little computational cost, particularly in highly-charged ions important for tin plasma dynamics, where the large spacing between atomic orbitals ensures rapid convergence of the MBPT expansion. AMBiT has also been used further studies of the electronic structure of tin ions in, for example, Refs. [158, 159]

Chapter Six

Calculations: Lr^+ and Lu^+

Disclaimer: This chapter was previously published in Ref. [160], as part of a collaboration to guide experimental studies of the spectrum of lawrencium. I was lead author on the paper and performed the CI+MBPT calculations and data analysis. The FSCC and QEDMOD calculations were carried out by A. Borschevsky, the outline of the proposed experiment was provided by M. Laatiaoui, and my co-authors also contributed edits and small amounts of content to the paper. I have include it in this chapter, slightly modified for readability and consistency with the rest of this thesis.

6.1 Introduction

The study of the transfermium elements (Z>100) lies at the frontier of contemporary nuclear and atomic physics research. The element synthesis itself provides a fertile terrain for studying effective interactions and nuclear matter under extreme conditions. Experimental shell gaps and single particle energies can be obtained from nuclear spectroscopy, which helps to improve model predictions for next spherical shell closures in the nuclear map: the location of the island of stability of superheavy elements.

Optical spectroscopy gives access to the atomic structure and provides insights into fundamental physics such as relativistic, correlation, and quantum electrodynamic (QED) effects. In addition, it can provide complementary information on single-particle and collective properties of atomic nuclei via hyperfine structure measurements. Such studies are continuously applied to ever heavier elements and are penetrating territories of the map that were previously inaccessible [161]. A good example of this is the recent laser spectroscopy of the element nobelium ($Z = 102$) [61], which demonstrated the technical feasibility despite a complete lack of tabulated spectral lines and production yields from nuclear fusion reactions of about one atom per second. These experiments have clearly shown how atomic modelling can efficiently support and guide atomic structure investigations and, in particular, that experiments and theory have to be pursued hand in hand. Current developments target the next heavier element, lawrencium ($Z = 103$), in its neutral and singly charged states from both theory and experimental view points.

The planned experiments will attempt to optically excite Lr in a supersonic gas-jet: the Lr atoms are produced with high-energy from fusion reactions and are stopped and thermalised in a buffer gas cell. The gas-jet method enables to accelerate the lawrencium-buffer gas mixture into a low-pressure and low-temperature jet. This in turn allows to reduce collisional broadening and thus to increase the experimental resolution [161, 162]. Previous experiments proved that the gas mixture contains both atomic species, neutral as well as singly ionised Lr, wherein the fraction of the ions substantially dominates the sample composition under typical experimental conditions [163, 164]. For both species, due to the extremely low production yields, highly precise theoretical predictions of the spectral lines are required to develop excitation schemes and to narrow down the search window to be able to pinpoint the ground-state transitions. Moreover, predictions of lifetimes and branching ratios are needed to quantify experimental parameters such as required detector sensitivities and beam times.

In this work we provide high accuracy prediction of the energies and the g-factors of the low-lying excited states of Lr$^+$, along with transition rates and branching ratios between the different states. The calculations are performed within two complementary state-of-the-art relativistic approaches: the Fock space coupled cluster (FSCC) method [165, 166], and the configuration interaction approach combined with many-body perturbation theory method (CI+MBPT) [95]. In order to estimate the accuracy of our predictions for Lr$^+$, analogous calculations were performed for its lighter homologue, Lu$^+$, where we can compare the results of our calculations to experimental values.

While numerous predictions were reported for neutral Lr, to the best of our knowledge, no experimental and only three prior theoretical studies of atomic properties of Lr$^+$ are available. Dzuba *et al* [167] calculated the first to the third ionization potentials of Lr using a linearised CI + all-order approach, while Cao and Dolg [168] calculated the first to the fourth ionization potentials of Lr using relativistic *ab initio* pseudopotentials combined with the complete active space self-consistent field method and corrected for spin-orbit effects. In a much earlier publication, Fraga presented a Hartree-Fock investigation of this system [169]; however, in that work, the ground state of Lr$^+$ was misidentified as $6d^2$.

6.2 Methods and Computational Details

All the calculations were carried out within the framework of the projected Dirac-Coulomb-Breit (DCB) Hamiltonian [170] (atomic units $\hbar = m_e = e = 1$ are used throughout this work),

$$H_{DCB} = \sum_i h_D(i) + \sum_{i<j} (1/r_{ij} + B_{ij}). \tag{6.1}$$

Here, h_D is the one electron Dirac Hamiltonian,

$$h_D(i) = c\,\boldsymbol{\alpha}_i \cdot \mathbf{p}_i + c^2(\beta_i - 1) + V_{\text{nuc}}(i), \tag{6.2}$$

where $\boldsymbol{\alpha}$ and β are the four-dimensional Dirac matrices. The nuclear potential V_{nuc} takes into account the finite size of the nucleus. The two-electron term includes the nonrelativistic electron repulsion and the frequency independent Breit operator,

$$B_{ij} = -\frac{1}{2r_{ij}}[\boldsymbol{\alpha}_i \cdot \boldsymbol{\alpha}_j + (\boldsymbol{\alpha}_i \cdot \mathbf{r}_{ij})(\boldsymbol{\alpha}_j \cdot \mathbf{r}_{ij})/r_{ij}^2], \tag{6.3}$$

and is correct to second order in the fine structure constant α.

6.3 FSCC

We have calculated the transition energies of Lr$^+$ and its lighter homologue Lu$^+$ using the relativistic multireference valence universal FSCC method, described in detail in Refs. [165, 166] and implemented in TRAFS-3C [171]. This approach has demonstrated a high degree of accuracy when treating heavy atomic systems; see, for example, Ref. [172]. Its particular advantage is the possibility of obtaining a large number of energy levels; it is therefore very well suited for calculating excitation spectra.

Our calculations start by solving the relativistic Hartree-Fock equations and correlating the closed-shell reference states for Lr^{3+} and Lu^{3+}, which correspond to closed shell configurations. After the first stage of the calculation, two electrons were added, one at a time, to obtain the singly ionised atoms. At each stage of the calculations the appropriate coupled cluster equations were solved iteratively. To achieve optimal accuracy, large model spaces were used, going up to $13s11p9d8f6g5h$ for Lu$^+$ and $14s12p10d9f6g5h$ for Lr$^+$, and the convergence of transition energies with respect to the model space size was verified. In order to allow the use of such large model spaces without encountering convergence difficulties in the coupled cluster iterations, the FSCC calculations were augmented by the extrapolated intermediate Hamiltonian approach (XIH) [173].

The uncontracted universal basis set [174] was used, consisting of even-tempered Gaussian type orbitals, with exponents given by

$$\begin{aligned} \xi_n &= \gamma\delta^{(n-1)}, \quad \gamma = 106\ 111\ 395.371\ 615 \\ \delta &= 0.486\ 752\ 256\ 286. \end{aligned} \tag{6.4}$$

The basis set used for both ions consists of 37 s (n=1–37), 31 p (n=5–35), 26 d (n=9–34), 21 f (n=13–33), 16 g (n=17–32), 11 h (n=21–31), and 6 i (n=25–30) functions. The outer 60 electrons of Lu$^+$ and 74 electrons of Lr$^+$ were correlated, and virtual orbitals with energies over 200 a.u. were omitted. The FSCC calculations were performed using the Tel-Aviv Relativistic Atomic Fock Space coupled cluster code (TRAFS-3C), written by E. Eliav, U. Kaldor and Y. Ishikawa.

To account for the QED corrections to the transition energies we applied the model Lamb shift operator (MLSO) of Shabaev *et al* [175] to the atomic no-virtual-pair many-body DCB Hamiltonian as implemented into the QEDMOD program [175]. Our implementation of the MLSO formalism into the Tel Aviv atomic computational package allows us to obtain the vacuum polarization and self energy contributions beyond the usual mean-field level, namely at the Dirac-

Coulomb-Breit Fock-Space Coupled Cluster with Single- and Double-excitations (DCB-FSCCSD) level.

6.4 CI+MBPT

Our calculations of the transition lifetimes and branching ratios, as well as the Landè g-factors for the excited states of Lr^+ and Lu^+ were performed using the relativistic configuration interaction approach augmented with many-body perturbation theory method, via the AMBiT atomic structure software [94]. We also present the transition energies calculated via this approach. The full details of this process have been extensively discussed elsewhere (see, for example Refs. [94, 95, 96, 100, 49, 113]), so we will only present a brief outline of the method here.

We start with a Dirac-Fock calculation in the V^{N-1} potential [176]; that is, all but one electron in the atom are included in the self-consistency calculations. This results in a set of Dirac-Fock orbitals which are optimised for states with a single electron-excitation (i.e. $6snl$ or $7snl$ for Lu^+ and Lr^+, respectively). Small-scale CI-only and CI+MBPT calculations showed that this choice of potential produces closer agreement to experimental and FSCC energy levels than including all N electrons in Dirac-Fock.

We generate a large basis of one-particle orbitals by diagonalising a set of B-splines over the one-electron Dirac-Fock operator [115, 116]. We modify the operator to incorporate Lamb shift corrections via the radiative potential method developed by Flambaum and Ginges [66], which includes the self-energy [67] and vacuum polarisation [68] contributions (finite nuclear-size effects are included using a Fermi distribution for nuclear charge). These corrections are propagated throughout the rest of the calculation by modification of the radial CI (Slater) and MBPT integrals.

Next, we use the B-spline basis functions to construct a set of many-electron configurations for the CI expansion. We form the many-body functions by allowing all single and double excitations from the $6s^2/7s^2$ ground-state up to $16spdfg$ (i.e. excitations with $n < 16$, and $0 < l < 4$). We then take the Slater determinants with a given M_J corresponding to these excitations and diagonalise the J^2 operator to form configuration state functions (CSFs), which are used to form the CI wavefunction via the standard CI eigenvalue problem [96].

We employ the emu CI method [113, 94] to significantly reduce the size of the CI eigenproblem by exploiting the fact that the CI expansion is typically dominated by contributions from N_{dominant} low-lying, dominant configurations. We divide the CI Hamiltonian matrix elements into three classes: leading diagonal elements; off-diagonal matrix elements containing at least one dominant configuration; and off-diagonal elements with no dominant configurations. The contributions from the high-lying off-diagonal terms to the low-energy levels are small compared to the dominant terms, and so can be set to zero without significant loss of accuracy [113, 129]. Typically $N_{\text{dominant}} \ll N_{\text{CI}}$, so emu CI can significantly reduce the size of the CI matrix and thus computational load when compared to standard CI.

For both Lr^+ and Lu^+, we construct the dominant configurations from all single excitations up to 16*spdfg* and single and double excitations up to 12*spdfg*; further increasing N_{dominant} changes the energy levels by less than 0.01%, suggesting this threshold captures all important configurations. In both systems, increasing the basis size beyond 16*spdfg* changes the energy by $\sim 1\%$, indicating that the CI component of our calculations are well converged.

Additionally, we include corrections from core-valence correlations to second-order via the diagrammatic MBPT technique described in Refs. [95, 100]. We have included all one- and two-body diagrams with orbitals up to 35*spdfghi* ($n \leq 35$, $0 \leq l \leq 6$). The MBPT corrections are rapidly convergent as more partial waves are added, and adding orbitals with $l \geq 7$ to the MBPT basis changes the energy by less than ~ 50 cm^{-1}. Consequently, the MBPT component of our calculation is also well-converged.

The resulting CI+MBPT wavefunctions are used to calculate the Landè g-factors and transition matrix elements. We include only the lowest-order matrix element of the complete correlated wavefunction. Transition lifetimes and branching ratios are derived from these matrix elements. Our use of MBPT to account for core-valence correlations means that in principle the electromagnetic transition operators should be modified [177, 178]. The resulting effective operators may be approximated by including higher-order corrections such as random-phase approximation, but these effects generally affect the matrix elements at well below our estimated precision of 30% [98].

For Lu^{+} the experimental transition energy was used in the expression for Einstein coefficients, while for Lr^{+} we used the calculated energies (our recommended values obtained from averaging the FSCC and the CI+MBPT results, see Section III for further details).

6.5 Results

Table 6.1 contains the calculated ionization potential and transition energies of Lu^{+}, obtained with both approaches, along with the experimental energies. While many states are obtained in the calculations, here we present only the 8 lowest levels (from the $5d6s$ and the $6s6p$ configurations) that correspond to experimentally relevant transitions in Lr^{+}. Generally, the results are in good agreement with experimental values, with average differences between theory and experiment of $-263\,(348)$ cm^{-1} (where the number in brackets is the standard deviation of the difference) for the FSCC approach, and $16\,(389)$ cm^{-1} for CI+MBPT. The two methods are also in good agreement with each other (average absolute difference of $278\,(496)$ cm^{-1}). We expect similar accuracy for the calculated transition energies of the heavier homologue of Lu^{+}, Lr^{+}, where no experimental data is yet available. The Breit interaction effect lowers the excitation energies by $20 - 150$ cm^{-1}, depending on the level. The QED corrections from both the MLSO formalism (for FSCC) and radiative potential method (for CI+MBPT) contribute between 100 – 200 cm^{-1}, also lowering the energies. Table 6.1 also contains the calculated g-factors, which are overall in good agreement with experiment, indicating that the CI+MBPT approach successfully reproduces the character of the electronic wavefunction. A notable exception is the g-factor of the $^{3}P_{2}$ state, which is predicted to be 1.5, while the experimental value is reported as 1.66 [179] (an assignment that may be erroneous).

Table 6.2 contains the calculated ionization potential, excitation energies, and g-factors of the lawrencium ion. In all cases the energies are significantly higher than the corresponding levels in Lu$^+$ (see Grotrian energy-level diagram for both Lu$^+$ and Lr$^+$ in Figure 1). This is due to the relativistic stabilisation of the valence $7s$ shell in the heavier ion, which makes this system more inert. The effect of the Breit interaction is higher in Lr$^+$ than in Lu$^+$, but the signs remain the same. Similarly, QED corrections in Lr$^+$ are slightly larger than in Lu$^+$, but remain on the order of 300 cm^{-1}, and are negative for all the considered states. The order of levels obtained using CI+MBPT and FSCC is the same, and the average difference between the two methods is $-47\,(747)$ cm^{-1}.

Table 6.2 also contains the recommended values for the excitation energies for the Lr$^+$ ion. The FSCC and CI+MBPT calculations have a comparable accuracy for Lu$^+$, often bracketing the experimental values. We use this observation to propose recommended transition energies which we calculate as the mean of the FSCC and CI+MBPT results. A conservative uncertainty estimate is given by either the difference between the two calculated energies or the standard deviation of the difference between the CI+MBPT and experimental energy levels for Lu$^+$ (389 cm^{-1}), whichever is larger.

Einstein A coefficients (transition probabilities) for electric-dipole allowed (E1) transitions and branching-ratios for the transitions between the 8 lowest states in Lu$^+$ and for a number of other transitions where experimental results are available are shown in Table 6.3.

Our calculated A values are mostly larger than experimental values tabulated in [180] by 10% – 30%, but the relative strengths are very well reproduced, and the strongest transitions are identified correctly. The results of our CI+MBPT calculations for Lr$^+$ transitions are shown in Table 6.4. We expect a similar accuracy for the predicted Einstein coefficients and branching ratios to that obtained for the lighter homologue Lu$^+$. The $7s7p$ configurations can decay via electric dipole transitions, however the even-parity $6d7s$ states can only decay via $M1$ or $E2$ transitions to other even-parity states, for which the Einstein A coefficients are shown in Table 6.5.

The lifetimes of the Lr$^+$ levels, calculated via the Einstein A-coefficients, are presented in Table 6.2. Because $M1$ and $E2$ transitions are slow, even-parity states have significantly longer lifetimes than states which can decay via E1 transitions. In particular, the $6d7s\ {}^3D_1$ state can only decay to the ground-state via a suppressed M1 transition, and so it has a lifetime of 2.2×10^6 seconds, or ~25 days, which is several orders of magnitude longer than any of the other levels.

6.6 Summary and conclusion

We have calculated energies, g-factors, and lifetimes of several low-lying atomic levels in Lr$^+$. A good agreement between the calculated FSCC and CI+MBPT energies is achieved. Similar calculations for the lighter homologue Lu$^+$ support the high accuracy of both approaches. In view of the prospects opened up by the forthcoming experiments, we identified two strong ground-state transitions in Lr$^+$, leading to $7s7p\ {}^3P_1$ and $7s7p\ {}^1P_1$ states at 31540 cm^{-1} and 47295 cm^{-1},

Table 6.1: Ionization potential (top row), excitation energies, and g-factors of Lu$^+$. The CI+MBPT and FSCC calculations include the Breit and the QED corrections, the latter of which is shown in a separate column for comparison between the two calculations. Only levels relevant to the proposed Lr$^+$ experiment are presented here.

State		g-factor		Energy (cm^{-1})				
		Exp.	CI+MBPT	FSCC	Δ QED	CI+MBPT	Δ QED	Exp. [179]
$6s^2$	$^1S_0$ IP	–	–	112696	-100	–		111970
$5d6s$	$^3D_1$	0.5	0.52	12354	-158	11664	-144	11796
	$^3D_2$	1.16	1.14	12985	-156	12380	-143	12432
	$^3D_3$	1.33	1.41	14702	-148	14267	-134	14199
	$^1D_2$	1.01	1.09	17892	-157	17875	-160	17332
$6s6p$	$^3P_0$	–	–	27091	-103	27303	-105	27264
	$^3P_1$	1.47	1.51	28440	-105	28520	-106	28503
	$^3P_2$	1.50	1.66	32294	-89	32603	-97	32453
	$^1P_1$	1.02	0.99	38464	-155	37385	-129	38223

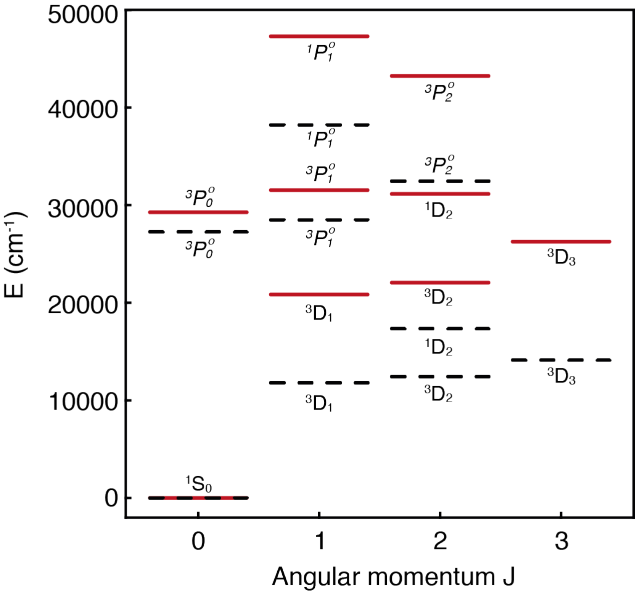


Figure 6.1: Grotrian diagram of experimental energy levels for Lu$^+$ (dashed, black) and recommended calculated energy levels for Lr$^+$ (solid, red). Levels are labeled by their approximate LS-coupling term symbol.

Table 6.2: Ionization potential (top row) and excitation energies of Lr$^+$. The CI+MBPT and FSCC columns include the Breit and the QED corrections, the latter of which is shown in a separate column for comparison between the two calculations. The recommended values are obtained as the mean of FSCC and CI+MBPT results. Lifetimes and and g-factors derived from CI+MBPT calculations are also included.

		g-factors			Energies (cm^{-1})			Lifetime (s)
State		CI+MBPT	FSCC	ΔQED	CI+MBPT	ΔQED	Recommended	
$7s^2$	$^1S_0$	–	116949	-219	–	–	116949 ± 389	–
$6d7s$	$^3D_1$	0.5	20265	-342	21426	-374	20846 ± 1161	2.23×10^{6}
	$^3D_2$	1.15	21623	-344	22507	-373	22065 ± 884	8.26×10^{-2}
	$^3D_3$	1.33	26210	-326	26313	-355	26262 ± 389	2.97×10^{-2}
	$^1D_2$	1.02	31200	-373	30942	-397	31071 ± 389	1.53×10^{-3}
$7s7p$	$^3P_0$	–	29487	-167	29059	-306	29273 ± 428	2.56×10^{-7}
	$^3P_1$	1.42	31610	-179	31470	-314	31540 ± 389	1.45×10^{-8}
	$^3P_2$	1.50	43513	-240	42860	-308	43186 ± 653	2.43×10^{-8}
	$^1P_1$	1.08	47819	-260	46771	-376	47295 ± 1048	1.11×10^{-9}

Table 6.3: Einstein coefficients ($A_{\text{CI+MBPT}}$) for dipole-allowed E1 transitions in Lu$^+$, calculated within the CI+MBPT approach using experimental transition energies, and compared to experimental values (E_{NIST}, A_{NIST}) where available [180]. Note that levels which are not relevant to the proposed Lr$^+$ experiment and are not included in [180] have been omitted, so branching ratios may not sum to 100% for all levels.

Upper level		Lower level		E_{NIST}	$A_{\text{CI+MBPT}}$(s^{-1})	A_{NIST} (s^{-1})	Branching ratio
$6s6p$	$^3P_0$	$5s5d$	$^3D_1$	–	2.19×10^{7}	–	1.00
$6s6p$	$^3P_1$	$6s^2$	$^1S_0$	28503	1.62×10^{7}	1.25×10^{7}	0.41
$6s6p$	$^3P_1$	$5d6s$	$^3D_1$	–	6.84×10^{6}	–	0.17
$6s6p$	$^3P_1$	$5d6s$	$^3D_2$	16707	1.60×10^{7}	9.90×10^{6}	0.40
$6s6p$	$^3P_1$	$5d6s$	$^1D_2$	–	7.39×10^{4}	–	0.18×10^{-3}
$6s6p$	$^3P_2$	$5d6s$	$^3D_1$	–	5.64×10^{5}	–	0.016
$6s6p$	$^3P_2$	$5d6s$	$^3D_2$	–	6.20×10^{6}	–	0.17
$6s6p$	$^3P_2$	$5d6s$	$^3D_3$	–	2.88×10^{7}	–	0.80
$6s6p$	$^3P_2$	$5d6s$	$^1D_2$	–	3.56×10^{5}	–	9.91×10^{-3}
$6s6p$	$^1P_1$	$6s^2$	$^1S_0$	38223	5.21×10^{8}	4.53×10^{8}	0.96
$6s6p$	$^1P_1$	$6s^2$	$^3D_1$	–	9.60×10^{3}	–	1.77×10^{-5}
$6s6p$	$^1P_1$	$6s^2$	$^3D_2$	–	9.86×10^{6}	–	0.02
$6s6p$	$^1P_1$	$6s^2$	$^1D_2$	–	1.07×10^{7}	–	0.02
$5d6p$	$^3D_1$	$6s^2$	$^1S_0$	45532	4.78×10^{7}	7.14×10^{7}	0.13
$5d6p$	$^3D_3$	$5d6s$	$^3D_3$	36298	1.82×10^{8}	1.66×10^{8}	0.56
$5d6p$	$^3D_3$	$5d6s$	$^3D_2$	34534	1.09×10^{8}	9.20×10^{7}	0.33

Table 6.4: Einstein coefficients ($A_{\text{CI+MBPT}}$) for dipole-allowed E1 transitions in Lr^+, calculated within the CI+MBPT approach and using our recommended calculated energies (E_{calc}). Branching ratios for each transition are also shown. We estimate the uncertainty in the A-coefficients at 30% (see text).

Upper level		Lower level		E_{calc} (cm^{-1})	$A_{\text{CI+MBPT}}(s^{-1})$	Branching ratio
$7s7p$	$^3P_0$	$7s6d$	$^3D_1$	8515	5.44×10^6	1.00
$7s7p$	$^3P_1$	$7s^2$	$^1S_0$	31540	6.36×10^7	0.900
$7s7p$	$^3P_1$	$7s6d$	$^3D_1$	10694	2.42×10^6	0.034
$7s7p$	$^3P_1$	$7s6d$	$^3D_2$	9475	4.66×10^6	0.066
$7s7p$	$^3P_1$	$7s6d$	$^1D_2$	608	4.54×10^{-1}	6.4×10^{-9}
$7s7p$	$^3P_2$	$7s6d$	$^3D_1$	22391	9.41×10^5	0.021
$7s7p$	$^3P_2$	$7s6d$	$^3D_2$	21172	9.70×10^6	0.214
$7s7p$	$^3P_2$	$7s6d$	$^3D_3$	16967	3.43×10^7	0.758
$7s7p$	$^3P_2$	$7s6d$	$^1D_2$	12304	3.19×10^5	0.007
$7s7p$	$^1P_1$	$7s^2$	$^1S_0$	47295	8.34×10^8	0.960
$7s7p$	$^1P_1$	$7s6d$	$^3D_1$	26449	1.36×10^6	0.002
$7s7p$	$^1P_1$	$7s6d$	$^3D_2$	25230	1.63×10^7	0.019
$7s7p$	$^1P_1$	$7s6d$	$^1D_2$	16363	1.68×10^7	0.019

Table 6.5: Einstein coefficients ($A_{\text{CI+MBPT}}$) for M1 and E2 transitions in Lr^+, calculated within the CI+MBPT approach and using our recommended calculated energies. We estimate the uncertainty in the A-coefficients at 30% (see text).

Upper level		Lower level		E_{calc} (cm^{-1})	$A_{M1}(s^{-1})$	$A_{E2}(s^{-1})$
$7s6d$	$^3D_1$	$7s^2$	$^2S_0$	20846	4.48×10^{-7}	–
$7s6d$	$^3D_2$	$7s^2$	$^2S_0$	22065	–	10.82
$7s6d$	$^3D_2$	$7s6d$	$^3D_1$	1219	0.79	2.78×10^{-5}
$7s6d$	$^3D_3$	$7s6d$	$^3D_1$	5416	–	0.0061
$7s6d$	$^3D_3$	$7s6d$	$^3D_2$	4197	33.61	0.015
$7s6d$	$^1D_2$	$7s^2$	$^2S_0$	31552	–	806.55
$7s6d$	$^1D_2$	$7s6d$	$^3D_1$	10306	49.81	0.0694
$7s6d$	$^1D_2$	$7s6d$	$^3D_2$	9087	5.72	0.0623
$7s6d$	$^1D_2$	$7s6d$	$^3D_3$	4890	5.67	0.0034

respectively, that should in principle be amenable for experimental verification. The level searches are likely to focus on the $^3P_1$ state, which is predicted with a relatively higher accuracy and exhibits an experimentally more convenient transition wavelength for optical probing.

Chapter Seven

Calculations: Neutral Lr and Lu

7.1 Introduction

The optical spectroscopy of transfermium elements ($Z > 100$) provides a unique test-bed for atomic and nuclear physics. Experimental and theoretical studies of the electronic structure of these elements can serve to probe relativistic effects, QED corrections, and configuration-mixing effects, as well as nuclear properties via hyperfine structure measurements. Recent measurements of the spectrum [61, 62] and ionisation potential [60] of nobelium ($Z = 102$) demonstrate that precision optical spectroscopy of transfermium elements is possible even when nuclear production rates are very low (on the order of one atom per second). The combination of state-of-the-art atomic structure calculations with new experimental methods has opened up new regions of the periodic table to precision optical spectroscopy. Lawrencium ($Z = 103$) provides a natural extension of investigations into the electronic structure of transfermium elements, with work already underway on optical spectroscopy of Lr and its ions [160].

These calculations were undertaken as part of a long-running collaboration with experimentalist researchers at GSI Helmholtzzentrum für Schwerionenforschung, utilising the SHIP facility [62, 63] for optical spectroscopy of superheavy elements (the work in chapter 6 and Ref. [160] was also part of this collaboration). The choice of energy levels, transition rates and g-factors was directly motivated by the requirements of planned studies of neutral lawrencium at this facility. Preliminary experiments using the RADRIS method developed for the spectroscopy of nobelium [61] have been performed to determine optimal materials and configurations for spectroscopy of lawrencium [181]; careful choice of filament material is important to compensate for the lower first ionization potential of lawrencium. Current production methods for lawrencium have a reduced cross section of a factor of ten compared to nobelium, making reliable theoretical predictions even more important if spectroscopy is to be possible within a realistic beam time period.

Previous theoretical studies of the electronic spectrum of Lr report incomplete spectra or only calculated a limited set of properties. Borschevsky *et al* [182] used the relativistic Fock space coupled cluster approach to calculate ionisation potentials and energy levels of Lr, but that study only reported energies for levels which can be reached by exciting a single electron from the ground-state ($7s^27p$) configuration. Dzuba *et al* [167] report the transition energies and the *g*-factors of Lr obtained using the configuration interaction method combined with the all-order single-double coupled-cluster technique (CI+all order). Multi-configurational Dirac-Fock (MCDF) was also employed [183] for calculations of both singly- and multiply-excited states. However, the estimated uncertainties in those predicted excitation energies are between 1200 cm^{-1} and 2400 cm^{-1}, which is too large to serve as a guide for precision spectroscopic measurements. Zou and Froese Fischer also carried out MCDF calculations of the energy levels and transition rates for Lu and Lr, but only calculated the three lowest lying states [184]. Finally, Sato *et al* performed experimental measurements of the ionisation potential of Lr [58, 59], along with corresponding coupled-cluster calculations.

In this chapter, I will present predictions of energies and Landé g-factors for low-lying states of neutral lawrencium, which will be probed in the planned measurements, as well as transition strengths and branching ratios for transitions between those states. I have used the configuration interaction with many-body perturbation theory (CI+MBPT) approach to calculate the g-factors and the transition rates, and to treat states that can not be handled by previous studies using the coupled cluster approach. As there are currently no experimental data of the Lr excitation spectrum, I have benchmarked the accuracy of our calculations by performing analogous calculations for the lighter homologue Lu, for which experimental values are available.

7.2 Methods and computational Details

The calculations in this chapter were carried out within the framework of the projected Dirac-Coulomb-Breit Hamiltonian [170] (atomic units $\hbar = m_e = e = 1$ are used throughout this chapter),

$$H_{DCB} = \sum_i h_D(i) + \sum_{i<j} (1/r_{ij} + B_{ij}). \tag{7.1}$$

Here, h_D is the one electron Dirac Hamiltonian,

$$h_D(i) = c\,\boldsymbol{\alpha}_i \cdot \mathbf{p}_i + c^2(\beta_i - 1) + V_{\text{nuc}}(i), \tag{7.2}$$

where $\boldsymbol{\alpha}$ and β are the four-dimensional Dirac matrices. The nuclear potential V_{nuc} takes into account the finite size of the nucleus, using the Fermi two-parameter charge distribution model. We also modify this operator to include the effects of QED corrections at this stage. The precise choice of the finite nucleus model was shown to have negligible effect on the calculated electronic properties [185]. The two-electron term includes the nonrelativistic electron repulsion and the frequency independent Breit operator,

$$B_{ij} = -\frac{1}{2r_{ij}}[\boldsymbol{\alpha}_i \cdot \boldsymbol{\alpha}_j + (\boldsymbol{\alpha}_i \cdot \mathbf{r}_{ij})(\boldsymbol{\alpha}_j \cdot \mathbf{r}_{ij})/r_{ij}^2], \tag{7.3}$$

and is correct to second order in the fine structure constant α.

7.2.1 CI+MBPT

The CI+MBPT calculations for both Lu and Lr were carried out using the AMBiT atomic structure software [94]. The full details of this process have been extensively discussed in chapter 2 of this thesis, so I will only present a brief outline of the method here.

We start with a Dirac-Fock calculation in the V^{N-1} potential [176]; that is, all but one electron in the atom are included in the self-consistency calculations. The resulting Dirac-Fock potential corresponds to a closed-shell, ns^2 configuration (where $n = 6$ for Lu and $n = 7$ for Lr), which reduces the impact of so-called subtraction diagrams on the accuracy of MBPT — see [94] for a discussion of the role of subtraction diagrams in CI+MBPT calculations. Small-scale CI+MBPT calculations showed that the choice of V^{N-1} potential produces closer agreement to experimental Lu energies.

We generate a large basis of one-particle orbitals by diagonalising a set of B-splines over the one-electron Dirac-Fock operator [115, 116]. We modify the operator to incorporate Lamb shift corrections via the radiative potential method developed by Flambaum and Ginges [66], which includes the self-energy [67] and vacuum polarisation [68] contributions. These corrections are propagated throughout the rest of the calculation by modification of the radial CI (Slater) and MBPT integrals.

Next, we use the B-spline basis functions to construct a set of many-electron configurations for the CI expansion. We form the many-body functions by allowing all single and double excitations from the $6s^25d/7s^27p$ ground-state for Lu and Lr, respectively, up to $22spdfg$ (i.e. excitations with $n < 22$, and $0 < l < 4$). We then take the Slater determinants with a given M_J corresponding to these excitations and diagonalise the J^2 operator to form configuration state functions (CSFs), which are used to form the CI wavefunction via the standard CI eigenvalue problem [96].

The emu CI method [113, 94] significantly reduces the size of the CI eigenproblem by exploiting the fact that the CI expansion is typically dominated by contributions from N_{small} low-lying, dominant configurations. We divide the CI Hamiltonian matrix elements into three classes: leading diagonal elements; off-diagonal matrix elements containing at least one dominant configuration; and off-diagonal elements with no dominant configurations. The contributions from the high-lying off-diagonal terms to the low-energy levels are small compared to the dominant terms, and so can be set to zero without significant loss of accuracy [113, 129]. Typically $N_{\text{small}} \ll N_{\text{CI}}$, so emu CI can significantly reduce the size of the CI matrix and thus computational load when compared to standard CI.

For both Lr and Lu, the CI basis consists of all single excitations from the ground-state configuration up to $22spdfg$ and single and double excitations up to $12spdfg$; further increasing N_{small} changes the energy levels by an average of $6\ \text{cm}^{-1}$, suggesting this threshold captures all important configurations. In both systems, increasing the basis size beyond $22spdfg$ changes the energy by $\sim 1\ \text{cm}^{-1}$, indicating that the CI component of our calculations are well converged.

Additionally, we include corrections from core-valence correlations to second-order via the diagrammatic MBPT technique described in Refs. [95, 100]. We have included all one- and two-body diagrams with orbitals up to $35spdfgh$ ($n \leq 35$, $0 \leq l \leq 5$). The MBPT corrections are rapidly convergent as more partial waves are added, and adding orbitals with $l \geq 6$ to the MBPT basis changes the energy by an average of 108 cm^{-1}.

The resulting CI+MBPT wavefunctions are used to calculate the Landé g-factors and transition matrix elements, including only the lowest-order matrix element of the complete correlated wavefunction. Transition lifetimes and branching ratios are derived from these matrix elements. Our use of MBPT to account for core-valence correlations means that in principle the electromagnetic transition operators should be modified [177, 178]. The resulting effective operators may be approximated by including higher-order corrections such as random-phase approximation, but these effects generally affect the matrix elements at well below our estimated precision (derived from comparison with experimentally derived transitions in Lu) of 40% [98]

7.3 Results

CI+MBPT results for Lu (shown in Table 7.1) agree closely with experimental values from [179]; the average disagreement between CI+MBPT and experimental energy levels is 141(294) cm $^{-1}$ (the number in brackets is the standard deviation of the difference between theory and experimental energies). We assume similar computational accuracy for Lr (Table 7.2) as we obtain for the lighter homologue Lu. In the absence of experimental data to compare against, we (conservatively) estimate the uncertainty in our CI+MBPT energy levels for Lr as the standard deviation of the differences between theory and experimental energy levels for Lu. We also compare our Lr results against previous calculations from Refs. [182, 167, 183, 184], which are presented in Table 7.3. Our results are generally in good agreement, except for the 7s^{2}6d $^2D_{3/2}$ level, where the CI+MBPT energy is smaller than results from previous calculations by $\sim$ 700 cm^{-1}. The reason for this disagreement is unclear, but the homologous 6s^{2}5d $^2D_{3/2}$ level in Lu likely does not have the same issue: the 6s^{2}5d $^2D_{3/2}$ level is the ground-state, so we should expect to see all excitation energies to be off by a large amount if our CI+MBPT calculations had errors of several hundred cm^{-1} in the ground-state energy.

We find different ground states for Lu (6s^{2}5d) and Lr (7s^{2}7p) due to the relativistic stabilisation of the 7p orbital and anti-contraction of the 6d orbital, in agreement with earlier studies [182, 167, 183, 184]. Additionally, the 7s^{2}8s level in Lr, which is the main target state in currently planned experiments, is lower in energy by $\sim$ 4000 cm^{-1} than the analogous 6s^{2}7s level in Lu; again due to the relativistic contraction of the 8s orbital.

Figures 7.1 – 7.3 show the effects of relativistic corrections on the radial wavefunctions of the 7s and 8s, 7p, and 6d orbitals in lawrencium. These orbitals are taken from the Dirac-Fock step of our CI+MBPT calculations; the relativistic orbitals are the ones we have used throughout the full CI+MBPT calculations, while the non-relativistic orbitals were obtained by setting the fine structure constant $\alpha \to 0$ (it was not possible within the structure of the AMBiT code to do a formally non-relativistic calculation, so the non-relativistic orbitals presented here are an approximation). We can clearly see the relativistic contraction of the 7s, 8s and 7p orbitals,

Figure 7.1: Comparison between relativistic and non-relativistic one-electron particle density $|\Psi|^2$ for the 7s and 8s orbital in Lr

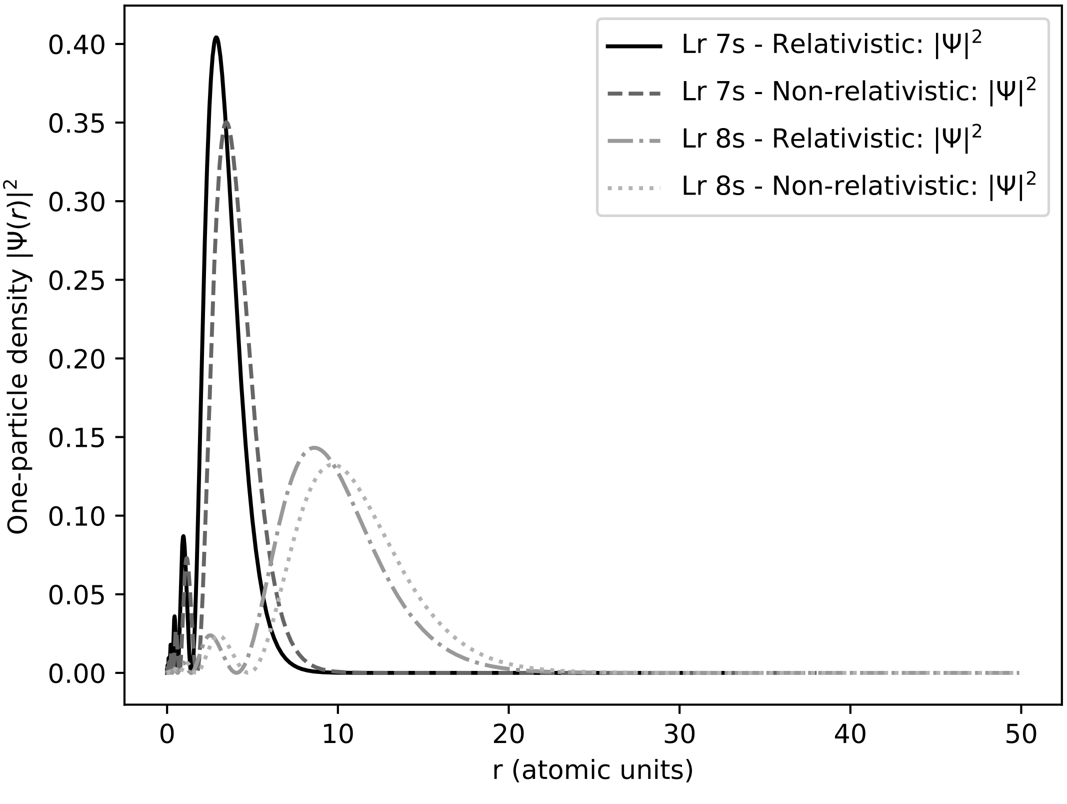


Figure 7.2: Comparison between relativistic and non-relativistic one-electron particle density $|\Psi|^2$ for the 7p orbital in Lr

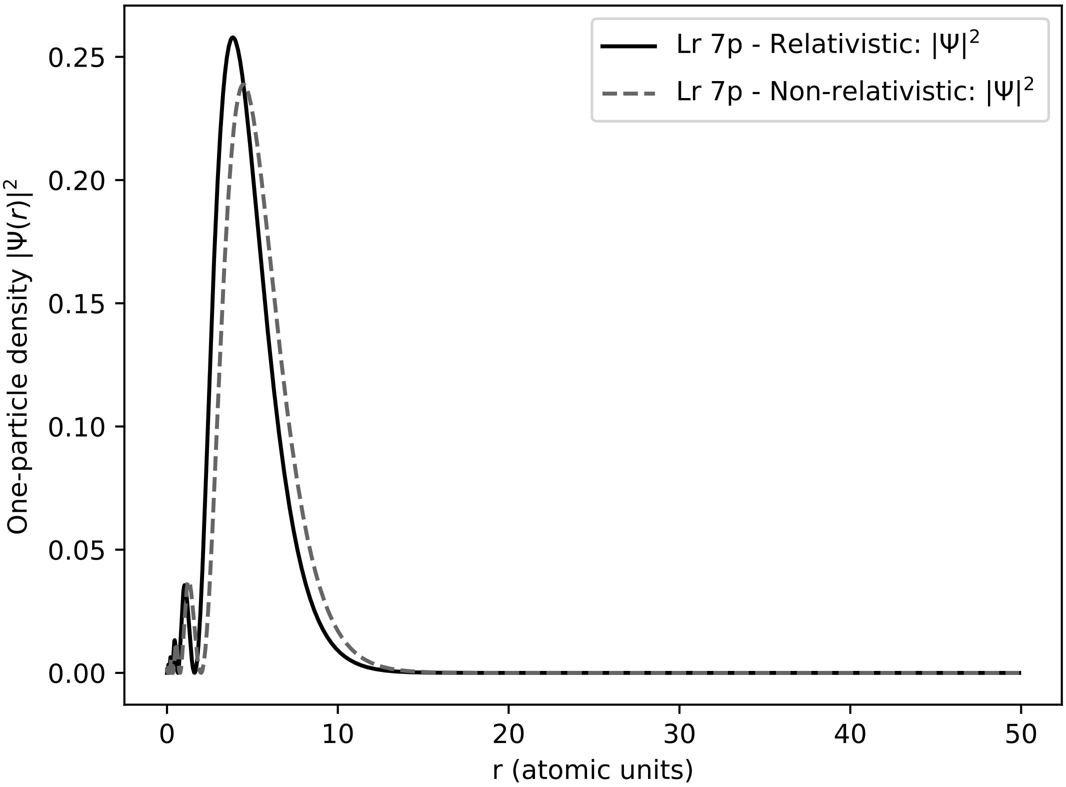
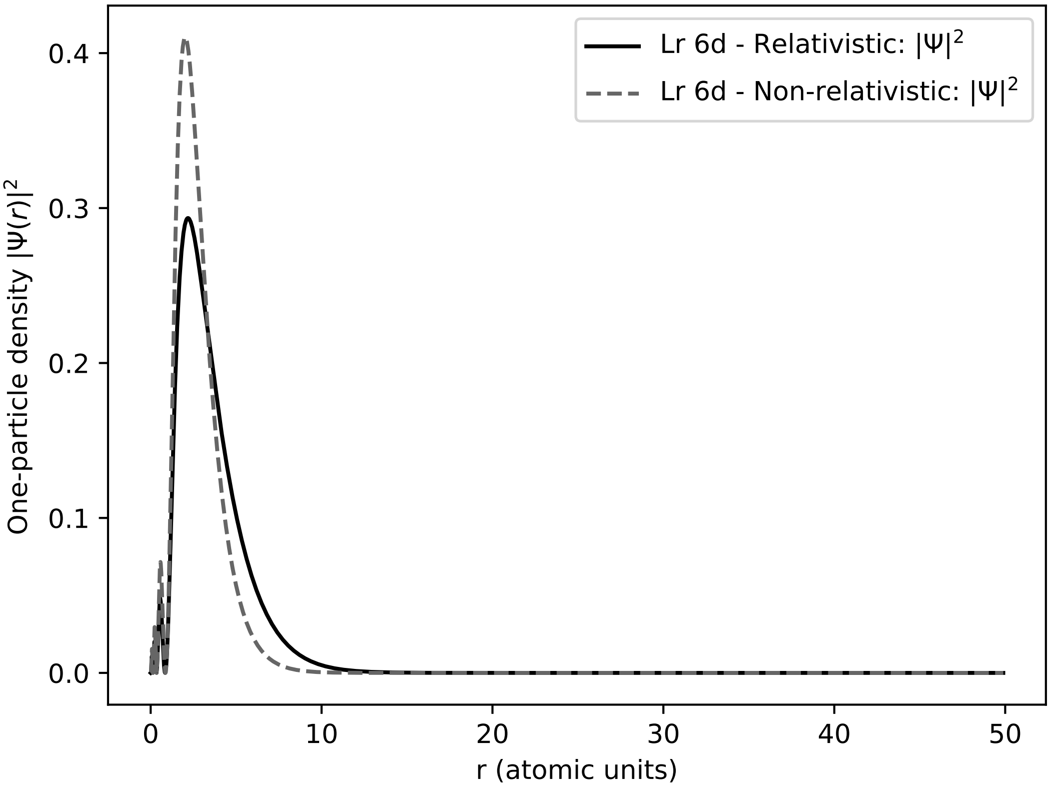


Figure 7.3: Comparison between relativistic and non-relativistic one-electron particle density $|\Psi|^2$ for the 6d orbital in Lr



which in turn result in the different ordering of configurations we observe between Lu and Lr. The effect of relativity on the 6d orbital is less pronounced than in the other orbitals shown here.

As a result of the relativistic corrections to the orbitals in lawrencium, the 7s^{2}8s level has fewer decay channels and thus a longer lifetime of 1.46$\times$10^{-8}s than the 6s^{2}7s level in Lu, which has a lifetime of 2.42$\times$10^{-17}s. The lifetimes of all Lr levels we have calculated are shown in Table 7.2. We included the contributions of the forbidden M1 and E2 transitions in the lifetimes of each state, but these have a negligible effect on the total lifetimes, with the exception of the 7s^{2}6d $^2D_{5/2}$ level. This level can only decay via the "forbidden" M1 transition to the 7s^{2}6d $^2D_{3/2}$ level, resulting a significantly longer lifetime of 0.25s compared to all other levels in Table 7.2.

Table 7.4 shows the Einstein A-coefficients (transition probabilities) for the lowest-lying electric dipole (E1) transitions in lutetium. We used the CI+MBPT wavefunctions in conjunction with the experimentally tabulated energy levels when calculating the transition rates, as this provides a more direct test of the uncertainty in our E1 transition matrix elements compared to using CI+MBPT energies. The spectra of both Lu and Lr are relatively dense, so to reduce the size of the tables we have only included Lu transitions which fulfill two criteria: they must have experimentally derived A-coefficients tabulated in Ref. [146], and at least one state in the transition must be analogous to the target states in Lr. Our A-coefficients are mostly larger than the experimentally derived values and differ from experimental results by an average of 40%, although transitions involving states with $J \geq 2.5$ tend to have worse accuracy. The relative strengths of the different transitions are reproduced and we can reliably identify the strongest transitions.

We present electric-dipole transition energies and A-coefficients between low-lying and target levels of lawrencium in table 7.5, where we expect similar accuracy to the E1 transitions in Lu. As in table 7.4, we have only included results for the levels which are of experimental interest, as well as states lower in energy than the experimental targets. The forbidden M1 and E2 transitions contribute a negligible amount compared to the dominant E1 transitions (except for the aforementioned 7s^{2}6d $^2D_{5/2}$ level, which can only decay via M1 transition), so we have not included them in the tables.

7.4 Summary of results

In this chapter, I have have calculated energies, g-factors, and lifetimes of several low-lying atomic levels in Lr. CI+MBPT energies are in good agreement with previously published computational studies, while similar calculations for the lighter homologue Lu are in good agreement with experimentally measured spectra, and support the accuracy the CI+MBPT approach. The transition rates for Lu agree with experimentally tabulated results at the level of $\sim$ 40%, and Lr calculations should have similar accuracy.

I have identified three transitions from the atomic ground state with a suitable transition strengths above 10^7 s^{-1}: the $7s^2\,8s\,^2S_{1/2}$ level at 20 485 cm^{-1}, the $7s\,7p^2\,^4P_{1/2}$ level at 25 381 cm^{-1} and the $7s^2\,7d\,^2D_{3/2}$ level at 28 580 cm^{-1}. The combination of dipole-allowed transition and large transition rate is required to ensure an efficient transfer of the population, making them attrac-

Table 7.1: Energy levels of lutetium. The CI+MBPT column gives the energy in cm^{-1} and includes the Breit and QED corrections, the latter of which is also presented separately in the ΔQED column. Experimental values are from results tabulated in [179].

Configuration	Term	J	g-factor	g-factor (expt.)	CI+MBPT (a.u.)	ΔQED (cm^{-1})	expt.(cm^{-1})
$6s^2 5d$	^2D	3/2	0.80	0.79	0	–	0
		5/2	1.20	1.20	2174	11	1994
$6s^2 6p$	$^2P^o$	1/2	0.66	0.66	4361	50	4136
		3/2	1.33	1.33	7734	56	7476
$5d6s6p$	$^4F^o$	3/2	0.45	0.5	17577	-70	17427
		5/2	1.06	1.07	18695	-69	18505
		7/2	1.24	1.22	20754	-61	20433
		9/2	1.33	1.3	23016	-55	22609
$6s5d^2$	^4F	3/2	0.41	–	18324	-98	18851
		5/2	1.03	1.04	18943	-95	19403
		7/2	1.24	–	19884	-91	20247
		9/2	1.33	–	21002	-85	21242
$5d6s6p$	$^4D^o$	1/2	0.04	0.00	20783	-72	20762
		3/2	1.16	1.19	21254	-72	21195
		5/2	1.38	1.39	22359	-65	22222
		7/2	1.42	1.41	23720	-63	23524
$5d6s6p$	$^2D^o$	5/2	1.21	1.23	21663	-72	21462
		3/2	0.86	0.87	22368	-67	22124
$6s5d^2$	^4P	1/2	2.60	–	21621	-68	21472
		3/2	1.68	1.73	22618	-73	22467
		5/2	1.43	–	22842	-81	22802
$5d6s6p$	$^4P^o$	1/2	2.61	–	24218	-66	24108
		3/2	1.64	1.67	24469	-65	24308
		5/2	1.51	1.53	25501	-60	25191
$6s^2 7s$	^2S	1/2	2.02	2.05	24396	79	24126
$6s5d^2$	^2D	3/2	0.85	–	24549	-83	24518
		5/2	1.13	–	24764	-98	24711
$6s5d^2$	^2F	5/2	1.09	1.6	25999	-92	25861
		7/2	1.06	–	26691	-106	26570
		9/2	1.11	–	27822	-90	26671
$5d6s6p$	$^2F^o$	5/2	0.89	0.88	28194	-83	28020
		7/2	1.14	–	29897	-79	29487

Table 7.2: Energy levels of lawrencium. The CI+MBPT column gives the calculated energy in cm^{-1} and includes the Breit and QED corrections, the latter of which is also presented separately in the ΔQED column.

Configuration	Term	J	g-factor	CI+MBPT (cm^{-1})	ΔQED (cm^{-}1)	Lifetime (s)
$7s^27p$	$^2P^o$	1/2	0.67	0	–	–
		3/2	1.33	8606	8	1.79 $\times 10^{-6}$
$7s^26d$	^2D	3/2	0.80	712	-77	3.69 $\times 10^{-4}$
		5/2	1.20	5252	-54	0.25
$7s^28s$	^2S	1/2	2.00	20485	40	1.456$\times 10^{-8}$
$7s7p6d$	$^4F^o$	3/2	0.48	20985	-189	1.10 $\times 10^{-7}$
		5/2	1.07	23289	-188	5.89 $\times 10^{-8}$
		7/2	1.25	28574	-174	2.07 $\times 10^{-7}$
		9/2	1.33	34758	-169	
$7s7p6d$	Odd (Ambiguous)	1/2	0.44	25887	-2	3.94 $\times 10^{-8}$
$7s7p6d$	$^4D^o$	3/2	1.29	26808	-76	4.89 $\times 10^{-8}$
		5/2	1.37	28708	-181	6.86 $\times 10^{-8}$
		7/2	1.35	33549	-161	2.87 $\times 10^{-7}$
$7s7p^2$	^4P	1/2	2.45	25381	-131	3.47 $\times 10^{-8}$
$7s6d^2$	^4F	3/2	0.43	24742	-151	9.27 $\times 10^{-7}$
		5/2	1.04	26165	-49	1.02 $\times 10^{-5}$
		7/2	1.23	28290	-222	5.2 $\times 10^{-4}$
		9/2	1.31	30754	-212	1.36 $\times 10^{-3}$
$7s^27d$	^2D	3/2	0.80	28580	31	1.34 $\times 10^{-8}$
		5/2	1.20	28725	-88	1.81 $\times 10^{-8}$
$7s^28p$	$^2P^o$	1/2	0.39	26996	-161	1.26 $\times 10^{-7}$
		3/2	1.30	28307	-82	6.19 $\times 10^{-8}$
$7s^29s$	^2S	1/2	2.00	30621	47	4.47 $\times 10^{-8}$
$7s^29p$	$^2P^o$	1/2	0.86	32307	-132	7.46 $\times 10^{-8}$
		3/2	1.33	33473	-167	1.12 $\times 10^{-7}$
$7s^26f$	$^2F^o$	5/2	0.86	31755	-167	5.78 $\times 10^{-8}$
		7/2	1.15	32560	21	5.69 $\times 10^{-8}$

Table 7.3: Comparison of Lr energy levels with previous calculations. All energies are given in cm^{-1}; energy levels not present in a particular source are left blank.

Configuration	Term	J	g-factor	CI+MBPT	FSCC[182]	CI + all order [167]	MCDF [183]	MCDF[184]
$7s^27p$	$^2P^o$	1/2	2.01	0	0	0	0	0
		3/2	0.80	8606	8413	8495	8138	7807
$7s^26d$	^2D	3/2	0.80	712	1436	1555	1331	1127
		5/2	1.20	5251	5106	5423	4187	
$7s^28s$	^2S	1/2	2.01	20484	20118	20253	20405	
$7s7p6d$	Odd	1/2	2.44	25886			27904	
$7s7p6d$	$^4F^o$	3/2	0.43	20985		21288	20886	
		5/2	1.20	23289		23530	23155	
		7/2	1.23	28574		28320	27276	
		9/2	1.31	34757		34212	32775	
$7s7p6d$	$^4D^o$	3/2	0.83	26808				
		5/2	1.04	28707				
		7/2	0.88	33549				
$7s7p^2$	^4P	1/2	2.44	25380				
$7s6d^2$	^4F	3/2	0.43	24742			25409	
		5/2	1.04	26165			27397	
		7/2	1.23	28290				
		9/2	1.31	30754			34807	
$7s^27d$	^2D	3/2	0.83	28580	28118			
		5/2	1.22	28725	28385			
$7s^28p$	$^2P^o$	1/2	2.00	26995	26111	25912	25246	
		3/2	1.61	28306	27508	27079	26902	
$7s^29s$	^2S	1/2	2.00	30620	30119			
$7s^29p$	$^2P^o$	1/2	2.21	32306	32295			
		3/2	0.80	33473	32840			
$7s^26f$	$^2F^o$	5/2	1.37	31754	32949			
		7/2	0.92	32560	32950			

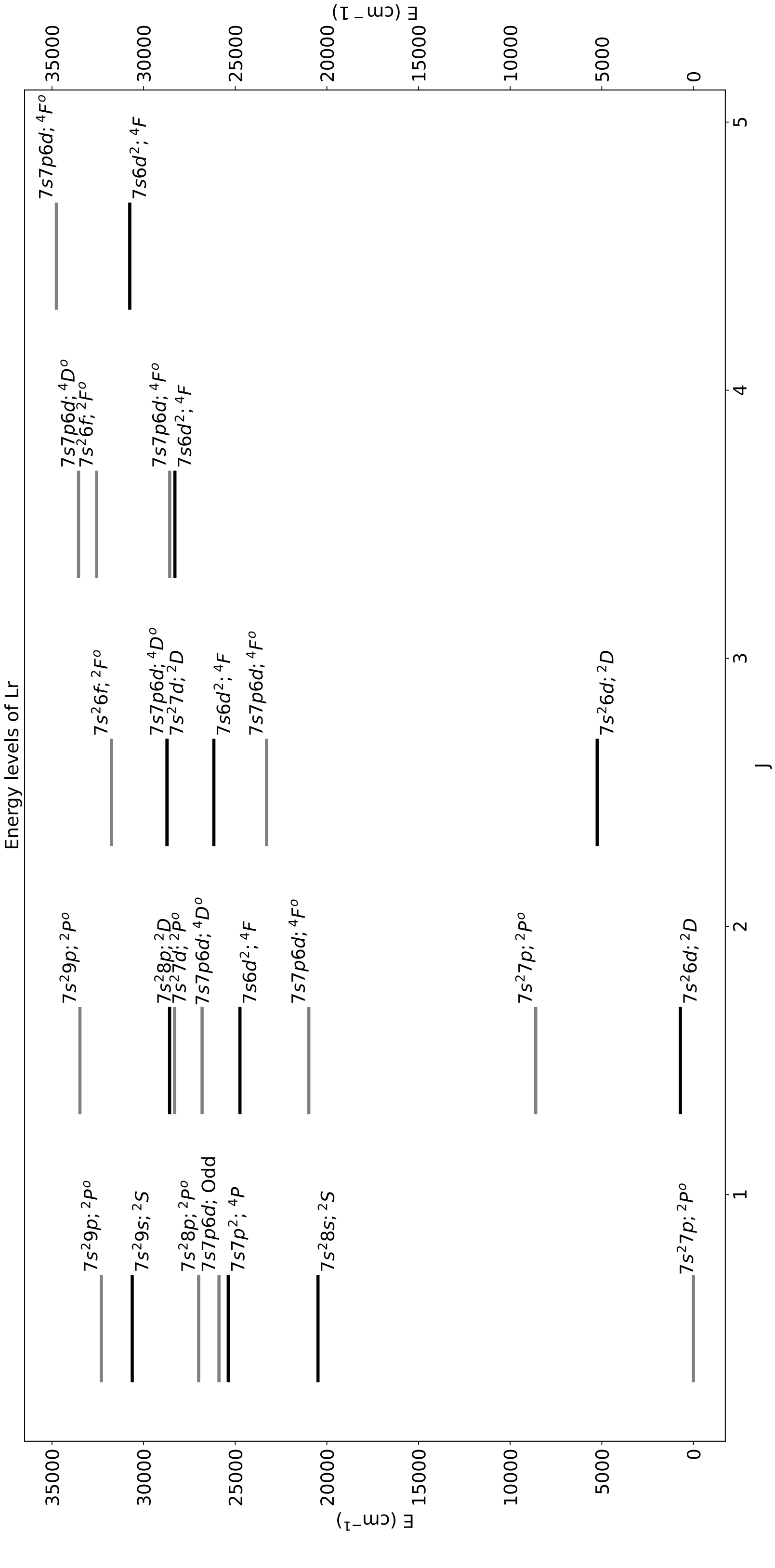


Figure 7.4: Diagram of the energy level scheme of Lr, as calculated by the CI+MBPT method and including Breit and QED corrections.

tive first targets for level searches. There are also three low-lying states which are very close in energy and have similar lifetimes: the $7s7d6d\ ^4F^\circ_{7/2}$ level at 28 574 cm^{-1}, the $7s^28p\ ^2P^\circ_{3/2}$ level at 28 307 cm^{-1} and the $7s6d^2\ ^4F^\circ_{7/2}$ level at 29 290 cm^{-1}. The lifetimes of these levels are all extremely short ($< 10^{-4}$), so it will be difficult to distinguish the levels by successive, delayed excitation or ionization. A similar difficulty was observed during the spectroscopy of nobelium in Refs. [61, 186] for similarly small level spacings, so future experiments will need to compensate for this degeneracy to achieve the desired level of accuracy.

These comprehensive results will serve as a useful guide for planned, future experimental studies of the spectrum of Lr, following on the work on its ion Lr$^+$ in chapter 6 and Ref. [160].

Table 7.4: Einstein A-coefficients for electric-dipole (E1) transitions in lutetium, as calculated by using CI+MBPT wavefunctions and experimental transition energies. We compare our calculated values with experimentally derived A-coefficients (A_{NIST}) tabulated in Ref. [146].

Upper level		lower level		E (cm^{-1})	$A_{\text{CI+MBPT}}$(s^{-1})	A_{NIST} (s^{-1})
5d6s6p	$^4F^o_{3/2}$	6s^2 5d	$^2D_{3/2}$	17427	2.54×10^6	1.73×10^6
5d6s6p	$^4F^o_{3/2}$	6s^2 5d	$^2D_{5/2}$	15433	9.07×10^4	7.2×10^4
5d6s6p	$^4F^o_{5/2}$	6s^2 5d	$^2D_{3/2}$	18505	1.81×10^6	1.20×10^6
5d6s6p	$^4F^o_{5/2}$	6s^2 5d	$^2D_{5/2}$	16511	1.49×10^6	9.2×10^5
5d6s6p	$^4D^o_{1/2}$	6s^2 5d	$^2D_{3/2}$	20762	1.06×10^6	9.0×10^5
5d6s6p	$^4D^o_{3/2}$	6s^2 5d	$^2D_{3/2}$	21195	5.58×10^4	1.13×10^5
5d6s6p	$^4D^o_{3/2}$	6s^2 5d	$^2D_{5/2}$	19201	2.57×10^5	2.5×10^5
5d6s6p	$^2D^o_{5/2}$	6s^2 5d	$^2D_{3/2}$	21462	4.01×10^6	3.15×10^6
5d6s6p	$^2D^o_{5/2}$	6s^2 5d	$^2D_{5/2}$	19468	1.15×10^7	9.1×10^6
5d6s6p	$^2D^o_{3/2}$	6s^2 5d	$^2D_{3/2}$	22125	3.22×10^7	2.26×10^7
5d6s6p	$^2D^o_{3/2}$	6s^2 5d	$^2D_{5/2}$	20131	7.35×10^4	3.3×10^4
5d6s6p	$^4D^o_{5/2}$	6s^2 5d	$^2D_{3/2}$	22222	3.37×10^5	1.28×10^5
5d6s6p	$^4D^o_{5/2}$	6s^2 5d	$^2D_{5/2}$	20228	2.21×10^6	9.7×10^5
6s^{2}7s	$^2S_{1/2}$	6s^2 6p	$^2P^o_{1/2}$	19990	2.78×10^7	3.20×10^7
6s^{2}7s	$^2S_{1/2}$	6s^2 6p	$^2P^o_{3/2}$	16650	4.86×10^7	4.9×10^7
5d6s6p	$4P^o_{3/2}$	6s^2 5d	$^2D_{3/2}$	24308	1.07×10^6	4.3×10^5
5d6s6p	$4P^o_{3/2}$	6s^2 5d	$^2D_{5/2}$	22314	9.13×10^4	8.8×10^4
5d6s6p	$4P^o_{5/2}$	6s^2 5d	$^2D_{3/2}$	25192	3.98×10^6	2.21×10^6
5d6s6p	$4P^o_{5/2}$	6s^2 5d	$^2D_{5/2}$	23198	1.22×10^6	6.6×10^5
5d6s6p	$^2F^o_{5/2}$	6s^2 5d	$^2D_{3/2}$	28020	1.05×10^8	6.9×10^7
5d6s6p	$^2F^o_{5/2}$	6s^2 5d	$^2D_{5/2}$	26026	3.52×10^7	2.65×10^7
5d6s6p	$^2F^o_{7/2}$	6s^2 5d	$^2D_{5/2}$	27493	1.18×10^7	6.4×10^6
5d6s6p	$^2F^o_{5/2}$	6s^2 5d	$^2D_{3/2}$	30184	1.31×10^8	1.85×10^8
6s^{2}6d	$^2D_{3/2}$	6s^2 6p	$^2P^o_{3/2}$	24066	1.70×10^7	1.68×10^7
6s^{2}6d	$^2D_{5/2}$	6s^2 6p	$^2P^o_{3/2}$	24237	9.08×10^7	8.9×10^7
5d6s6p	$^2F^o_{7/2}$	6s^2 5d	$^2D_{5/2}$	29757	2.77×10^8	2.44×10^8

Continued next page...

Upper level		Lower level		E (cm^{-1})	$A_{\mathrm{CI+MBPT}}$(s^{-1})	A_{NIST} (s^{-1})
6s^{2}5f	$^2F^o_{5/2}$	6s^2 5d	$^2D_{3/2}$	36633	1.87×10^7	2.32×10^7
6s^{2}5f	$^2F^o_{7/2}$	6s^2 5d	$^2D_{5/2}$	34650	2.52×10^7	2.31×10^7

Table 7.5: Einstein A-coefficients for electric-dipole (E1) transitions in lawrencium, as calculated within the CI+MBPT framework. States in a transition are identified by the same labels as in table 7.2. Branching ratios for each transition are also shown. We estimate the uncertainty in the coefficients as 40% (see text).

Upper level		Lower level		E (cm^{-1})	A(s^{-1})	Branching ratio
7s^{2}6d	$^2D_{3/2}$	7s^{2}7p	$^2P^o_{1/2}$	712	1.26×10^3	1
7s^{2}8s	$^2S_{1/2}$	7s^{2}7p	$^2P^o_{3/2}$	11878	3.57×10^7	0.52
7s^{2}8s	$^2S_{1/2}$	7s^{2}7p	$^2P^o_{1/2}$	20485	3.31×10^7	0.48
7s 6d2	$^4F_{3/2}$	7s7p6d	$^4F^o_{5/2}$	1453	3.59×10^3	2.92×10^{-3}
7s 6d2	$^4F_{3/2}$	7s7p6d	$^4F^o_{3/2}$	3757	2.58×10^5	0.24
7s 6d2	$^4F_{3/2}$	7s^{2}7p	$^2P^o_{3/2}$	16136	6.73×10^3	0.01
7s 6d2	$^4F_{3/2}$	7s^{2}7p	$^2P^o_{1/2}$	24742	7.96×10^5	0.75
7s 7p2	$^4P_{1/2}$	7s7p6d	$^4F^o_{3/2}$	4396	2.27×10^3	8.06×10^{-5}
7s 7p2	$^4P_{1/2}$	7s^{2}7p	$^2P^o_{3/2}$	16775	3.22×10^6	0.11
7s 7p2	$^4P_{1/2}$	7s^{2}7p	$^2P^o_{1/2}$	25381	2.51×10^7	0.89
7s 6d2	$^4F_{5/2}$	7s7p6d	$^4F^o_{5/2}$	2876	9.60×10^4	0.85
7s 6d2	$^4F_{5/2}$	7s7p6d	$^4F^o_{3/2}$	5180	1.36×10^4	0.12
7s 6d2	$^4F_{5/2}$	7s^{2}7p	$^2P^o_{3/2}$	17559	3.00×10^3	0.03
7s^{2}7d	$^2D_{3/2}$	7s^{2}8p	$^2P^o_{3/2}$	273	2.51×10^2	3.38×10^{-6}
7s^{2}7d	$^2D_{3/2}$	7s^{2}8p	$^2P^o_{1/2}$	1585	1.73×10^5	1.39×10^{-3}
7s^{2}7d	$^2D_{3/2}$	7s7p6d	$^4D^o_{3/2}$	1772	2.43×10^4	3.26×10^{-4}
7s^{2}7d	$^2D_{3/2}$	7s7p6d	Odd ($J = 1/2$)	2693	7.09×10^5	0.01
7s^{2}7d	$^2D_{3/2}$	7s7p6d	$^4F^o_{5/2}$	5291	3.30×10^3	4.44×10^{-5}
7s^{2}7d	$^2D_{3/2}$	7s7p6d	$^4F^o_{3/2}$	7585	2.52×10^2	3.40×10^{-6}
7s^{2}7d	$^2D_{3/2}$	7s^{2}7p	$^2P^o_{3/2}$	19974	1.21×10^7	0.16
7s^{2}7d	$^2D_{3/2}$	7s^{2}7p	$^2P^o_{1/2}$	28580	6.14×10^7	0.83
7s^{2}7d	$^2D_{5/2}$	7s7p6d	$^4D^o_{5/2}$	18	1.72×10^{-3}	3.18×10^{-11}
7s^{2}7d	$^2D_{5/2}$	7s7p6d	$^4F^o_{7/2}$	151	5.44×10^{-1}	1.01×10^{-8}
7s^{2}7d	$^2D_{5/2}$	7s^{2}8p	$^2P^o_{3/2}$	419	5.47×10^3	1.01×10^{-4}
7s^{2}7d	$^2D_{5/2}$	7s7p6d	$^4D^o_{3/2}$	1917	1.97×10^5	3.66×10^{-3}
7s^{2}7d	$^2D_{5/2}$	7s7p6d	$^4F^o_{5/2}$	5436	3.97×10^3	7.32×10^{-5}
7s^{2}7d	$^2D_{5/2}$	7s7p6d	$^4F^o_{3/2}$	7740	4.18×10^3	7.73×10^{-5}
7s^{2}7d	$^2D_{5/2}$	7s^{2}7p	$^2P^o_{3/2}$	20119	5.39×10^7	0.99

Continued next page...

Upper level		Lower level		E (cm^{-1})	A(s^{-1})	Branching ratio
7s^{2}7p	$^2P^o_{3/2}$	7s^{2}6d	$^2D_{5/2}$	3355	3.13×10^5	0.49
7s^{2}7p	$^2P^o_{3/2}$	7s^{2}6d	$^2D_{3/2}$	7894	3.01×10^5	0.51
7s7p6d	$^4F^o_{3/2}$	7s^{2}8s	$^2S_{1/2}$	501	1.53×10^{-1}	5.74×10^{-8}
7s7p6d	$^4F^o_{3/2}$	7s^{2}6d	$^2D_{5/2}$	15734	1.56×10^5	0.02
7s7p6d	$^4F^o_{3/2}$	7s^{2}6d	$^2D_{3/2}$	20273	8.80×10^6	0.98
7s7p6d	$^4F^o_{5/2}$	7s^{2}6d	$^2D_{5/2}$	18038	3.26×10^6	0.19
7s7p6d	$^4F^o_{5/2}$	7s^{2}6d	$^2D_{3/2}$	22577	1.35×10^7	0.81
7s7p6d	Odd ($J = 1/2$)	7s7p^2	$^4P_{1/2}$	506	5.39×10^2	2.11×10^{-5}
7s7p6d	Odd ($J = 1/2$)	7s6d^2	$^4F_{3/2}$	987	1.20×10^4	4.70×10^{-4}
7s7p6d	Odd ($J = 1/2$)	7s^{2}8s	$^2S_{1/2}$	5402	3.58×10^6	0.14
7s7p6d	Odd ($J = 1/2$)	7s^{2}6d	$^2D_{3/2}$	25175	2.21×10^7	0.86
7s7p6d	$^4D^o_{3/2}$	7s6d^2	$^4F_{5/2}$	643	2.41×10^3	1.19×10^{-4}
7s7p6d	$^4D^o_{3/2}$	7s7p^2	$^4P_{1/2}$	1427	7.41×10^3	3.66×10^{-4}
7s7p6d	$^4D^o_{3/2}$	7s6d^2	$^4F_{3/2}$	2066	9.36×10^3	4.62×10^{-4}
7s7p6d	$^4D^o_{3/2}$	7s^{2}8s	$^2S_{1/2}$	6324	3.22×10^6	0.16
7s7p6d	$^4D^o_{3/2}$	7s^{2}6d	$^2D_{5/2}$	21557	2.59×10^6	0.13
7s7p6d	$^4D^o_{3/2}$	7s^{2}6d	$^2D_{3/2}$	26283	1.44×10^7	0.71
7s^{2}8p	$^2P^o_{1/2}$	7s7p^2	$^4P_{1/2}$	1615	5.95×10^2	8.43×10^{-5}
7s^{2}8p	$^2P^o_{1/2}$	7s6d^2	$^4F_{3/2}$	2253	5.27×10^4	0.007
7s^{2}8p	$^2P^o_{1/2}$	7s^{2}8s	$^2S_{1/2}$	6511	6.42×10^6	0.91
7s^{2}8p	$^2P^o_{1/2}$	7s^{2}6d	$^2D_{3/2}$	26283	5.95×10^5	0.08
7s^{2}8p	$^2P^o_{3/2}$	7s6d^2	$^4F_{5/2}$	2141	8.12×10^3	5.30×10^{-4}
7s^{2}8p	$^2P^o_{3/2}$	7s7p^2	$^4P_{1/2}$	2926	2.73×10^4	1.79×10^{-3}
7s^{2}8p	$^2P^o_{3/2}$	7s6d^2	$^4F_{3/2}$	3564	1.35×10^4	8.82×10^{-4}
7s^{2}8p	$^2P^o_{3/2}$	7s^{2}8s	$^2S_{1/2}$	7822	1.14×10^7	0.75
7s^{2}8p	$^2P^o_{3/2}$	7s^{2}6d	$^2D_{5/2}$	23055	6.14×10^4	3.80×10^{-3}
7s^{2}8p	$^2P^o_{3/2}$	7s^{2}6d	$^2D_{3/2}$	27595	3.76×10^6	0.24
7s7p6d	$^4F^o_{7/2}$	7s6d^2	$^4F_{7/2}$	284	1.135×10^2	2.39×10^{-5}
7s7p6d	$^4F^o_{7/2}$	7s6d^2	$^4F_{5/2}$	2409	1.17×10^4	2.00×10^{-3}
7s7p6d	$^4F^o_{7/2}$	7s^{2}6d	$^2D_{5/2}$	23323	4.74×10^6	0.998
7s7p6d	$^4D^o_{5/2}$	7s6d^2	$^4F_{7/2}$	417	1.07×10^2	3.48×10^{-6}
7s7p6d	$^4D^o_{5/2}$	7s^{2}7d	$^2D_{3/2}$	127	5.53×10^{-3}	5.79×10^{-11}
7s7p6d	$^4D^o_{5/2}$	7s6d^2	$^4F_{5/2}$	2542	1.14×10^4	3.69×10^{-4}
7s7p6d	$^4D^o_{5/2}$	7s6d^2	$^4F_{3/2}$	3965	4.99×10^3	1.62×10^{-4}
7s7p6d	$^4D^o_{5/2}$	7s^{2}6d	$^2D_{5/2}$	23456	2.03×10^7	0.66
7s7p6d	$^4D^o_{5/2}$	7s^{2}6d	$^2D_{3/2}$	27966	1.05×10^7	0.34

Chapter Eight

Conclusion

In this thesis, I have presented highly-parallel implementations of the CI+MBPT method for atomic structure calculations. Starting with the AMBiT codebase, I have described, in detail, the process of converting existing MPI-enabled scientific code to use a hybrid MPI+OpenMP model. I have also overhauled key data structures in the AMBiT code base to take advantage of performance improvements in modern CPU hardware due to memory caching and instruction-level parallelism. These changes have allowed us to scale our atomic calculations up to hundreds of processors, which in turn allowed us to reach numerical saturation in CI+MBPT calculations for the first time.

I have applied AMBiT to both benchmark systems, like Db and Cr^+, as well as systems relevant to cutting-edge experimental applications (Lr, Lr^+ and tin highly-charged ions). These calculations consistently reached the 1% or sub-1% level of accuracy, even for challenging open-shell systems, which simply would not have been possible without the work I performed to overhaul the AMBiT's parallelism. This further proves AMBiT's utility in guiding experimental searches and applications in complex atoms and ions — a role in which AMBiT is at the cutting-edge of available techniques.

8.1 Perspectives for the future

8.1.1 Computational improvements – GPU computing

The biggest, most obvious software engineering which may be considered is to overhaul AMBiT to take advantage of GPU computing.

As high-performance computing moves into the exaflop scale [1], supercomputing clusters are increasingly moving towards heterogeneous architectures including application-specific accelerators [187]. For most of the 20th century, gains in computational power had come from the trend of exponential increase in clock frequency of individual CPU cores, which in turn depended on *Dennard scaling* (the trend that the power density of transistors stays constant as their size decreases) [131], so CPUs could be expected to double in power every 18 months — the famous *Moore's law*. However, Dennard scaling ended in approximately 2004 as CPU manufacturers

[1] 1 EFLOP of computing power corresponds to 10^{18} floating point operations per second

could no longer reduce transistor operating voltages, and smaller processor features became vulnerable to current leakages [187]. Consequently, subsequent increases in computational power came largely from more aggressively exploiting parallelism (e.g. increasingly large multicore processors) rather than raw clock speed. More recently, power consumption issues in traditional general-purpose CPUs have forced HPC architects to focus on manycore architecture consisting of large numbers of relatively slow and "unsophisticated" processing units, of which the GPU is the most well-established example.

Programming for a GPU is fundamentally different to programming for traditional CPU architectures. As previously stated, a GPU can exploit massive levels of parallelism, with a typical GPU unit containing thousands of arithmetic units, and can thus reach extremely high levels of numerical performance (as of the time of writing, cutting-edge GPUs offer theoretical peak performances of several teraflops).

The tradeoff is that typical constructs in CPU computing such as extensive branching logic or non-consecutive memory accesses (including common pointer-chasing algorithms) are extremely slow when naively transferred to a GPU, which often requires extensive code re-factoring in order to even run (let alone reach peak performance). Additionally, GPU applications must be much more deliberate with memory access patterns, for two reasons: first, device memory (located on the GPU) is limited to a few GB, so any data which doesn't fit in memory must be shuffled back and forth between the host (main node) and device (the GPU), which is extremely slow. Additionally, GPU compute capacity (in FLOPS) vastly exceeds their memory bandwidth (even for device memory), so GPU code must be strongly *compute-bound* in order to keep the device arithmetic units fully utilised. For the interested reader, a more detailed breakdown of GPU computing can be found in, e.g. Refs. [188, 189].

So where does this leave AMBiT? I believe that the underlying CI+MBPT algorithm could be adapted to GPU computing, and would result in large performance improvements over the current MPI+OpenMP implementation, but this would require substantial re-writing of the codebase. Firstly, as we saw in the OpenMP profiles in chapter 3, the most expensive part of a typical calculation with AMBiT is generating the CI matrix, which is strongly memory bound — for Db $\sim 70\%$ of the run time is spent in memory accesses, and a further $\sim 25\%$ is spent in branching logic. This behaviour is partially because we front-load the calculation of the one- and two-electron integrals, but is also because generating the individual matrix elements in the CSF/J^2 representation requires some degree of fiddling with angular momentum data (which we also pre-calculate) in both memory accesses and branching logic. Neither of these properties is ideal for GPU performance.

I believe that the best way forward would be to refactor AMBiT so that integrals and matrix elements are calculated only when needed and then thrown away, which I'll refer to as "on-the-fly" generation. The approach would probably look similar to that employed by the `BIGSTICK` CI code [190, 191] or earlier work by Olsen *et al* [192] — the CI matrix elements are calculated during the matrix-vector products in the Davidson algorithm, and only persist inside the loop iteration for that particular matrix vector product. At first glance, this may seem like an inefficient approach, as it duplicates the numerical work of calculating the matrix elements (although

BIGSTICK somewhat gets around it by requiring the Coulomb integrals to be pre-calculated by an external program and supplied in a file, reducing the components which need to be calculated at runtime), but it has several potential advantages when it comes to GPU computing. First, the complete CI matrix is never stored in memory, which means that the limited device memory of most GPUs would no-longer be an issue (no need to shuffle data back and forth between device and host). The second is that on-the-fly calculation of the CI-matrix shifts the performance from being memory-bound to being compute bound, which is vastly preferable for GPU performance. Shan *et al* showed that this approach provides very good scaling on the (now-discontinued) *Intel Xeon Phi co-processor* [191], which is conceptually somewhere between a CPU and a GPU in architecture and performance characteristics, so the kinds of programming patterns which perform well on the Phi (compute-bound algorithms, heavy use of vector operations, minimal branching logic) transfer to the GPU as well [193, 194].

Overhauling AMBiT for GPU computing is a much bigger endeavour than adding OpenMP support was, and is too large of a project for this thesis. The entire subprocess of generating and diagonalising the CI matrix needs to be replaced with one which essentially merges the Davidson algorithm with the code to generate the matrix elements (along with the necessary GPU specific code). Additionally, the current CSF basis of the matrix is likely too dependent on branching logic to achieve good GPU performance (although this would need to be properly profiled), and so it may be necessary to shift to representing the CI matrix in terms of projections (the so-called *M-basis representation*, since projections are Slater determinants with a well-defined angular momentum projection M); in turn posing problems from degeneracy and isolating desired solutions, which are not present in the CSF basis. It is also not clear if it is feasible to include MBPT integrals on-the-fly, or if they must be pre-calculated (thus increasing the complexity of data-movement and GPU offloading) due to the very large number of MBPT diagrams needed for each integral.

All of this probably amounts to several years' worth of work to get to the point where the new version is production ready. It would, however, allow us to tackle significantly more complex systems in much less time than even current MPI+OpenMP calculations, so would ultimately be a worthwhile investment.

8.1.2 CI+MBPT implementation improvements

There are also some potential improvements to our implementation of CI+MBPT which would improve the accuracy and the range of applications AMBiT could support.

Firstly, even though B-Splines are fast and highly accurate for the calculations in this thesis, they are not the only possible choice of basis functions and have some shortcomings. B-Spline basis functions give approximately spectroscopic wavefunctions for low-lying, bound states and incorporate relativistic effects in a mathematically elegant way, but they do not give as good of a representation of continuum wavefunctions. This limitation has not been a problem for the work in this thesis, but there are a number of photon-atom interactions, such as photon-atom scattering, atomic polarisability and nonlinear optics, which require an accurate treatment of the continuum to calculate. For example, consider the atomic dipole polarisability α, which determines the second-order shift to atomic energy levels due to an external electric field. For the ground-state of atomic hydrogen, $\sim 20\%$ of the polarisability is due to contributions from the continuum, while for heavier systems the continuum typically contributes $\sim 1\%$ to the polarisability [195]. Consequently, if CI+MBPT is to be useful for these calculations, or related calculations such as photon-atom scattering cross-sections, it will be necessary to implement new types of basis functions to capture continuum dynamics.

One approach which would work well with the structure of AMBiT is to use Laguerre spinors (L-spinors) [196, 104] and/or the closely related Coulomb Sturmian basis functions [197], as the one-particle basis functions. L-spinors have a similar functional form to the eigenfunctions of the hydrogen atom, but have a tunable exponential parameter which allows for the spectrum to be "tuned" to give a discretised representation of both the bound and continuum states. The discretised continuum wavefunctions are known as *pseudostates*, which are not individually physically meaningful, but can be summed over to give a good approximation of the total contribution from the whole continuum [198]. Jiang *et al* used L-spinors as the basis for a large relativistic CI calculation of the spectrum and polarisability of Sr^+ [104] and Ca^+ [199], giving results which agree to within 1% of more sophisticated perturbation theory methods. Thus far their implementation is limited to systems with a single valence electron above closed shells, so it will take some work to extend the L-spinor formalism to many-electron CI+MBPT calculations. However, it represents a promising way forwards for applying CI+MBPT to photon-atom calculations.

Another possible target for improvement is the accuracy of our transition matrix element calculations. Strictly speaking, we should modify the electric and magnetic multipole operators when using MBPT to account for the modified effective Hamiltonian in the CI+MBPT equations [177]. AMBiT does implement the random-phase approximation (RPA) equations to account for this effect (full details of the equations are provided in Ref. [127]), and there are input file options to control their use in transition matrix element calculations. Unfortunately, my experiences with the calculations in this thesis have shown that these equations (or at least their implementation in AMBiT) are not unconditionally stable and can sometimes rapidly and unpredictably diverge for many-electron systems. Consequently, it would be necessary to properly investigate the convergence properties of RPA and potentially re-cast the equations in AMBiT into a more stable form (for example, an implicit, matrix-based formulation) if we want to use AMBiT to carry out precision photon-atom and transition calculations.

Finally, there are still some numerical convergence issues in our Dirac-Fock (DF) implementation, which is based on the algorithm in Ref. [114]. This is less of a pressing concern than the other improvements in this section, as it is usually possible to "twiddle" the parameters of the DF calculation until it converges, but the fact that we cannot always guarantee that the DF basis will be well-behaved for all calculations does impact the code's usability. One possible route forward is to take cues from efforts to develop unconditionally stable Hartree-Fock algorithms for molecular structure calculations [200, 201]. Molecular structure calculations have different requirements to atomic structure, however: for example, most codes for large molecules are strictly non-relativistic, and typically focus on closed-shell configurations and light atoms. We therefore wouldn't be able to plug in these approaches without some modifications to the numerics, but it at least presents a possible starting point for overhauling AMBiT's DF subroutines to be more reliable.

8.2 Summing up

This perspective on future work should not be taken to mean that AMBiT is prohibitively slow or resource intensive in its current state — there are limitations and anomalies in all methods for atomic structure theory, and AMBiT fulfils an important niche in precision, many-electron calculations. Furthermore, AMBiT is still scalable to large numbers of conventional CPUs and will remain performant on future supercomputers even if no additional software engineering is done.

In this thesis, I have shown that AMBiT can deliver highly accurate calculations of open-shell systems, including highly-charged ions superheavy elements, transition metals and lanthanides. The calculations in this thesis cover a wide-range of typical electronic structures and performance requirements, so AMBiT can easily be applied in future to problems in astrophysics, atomic clocks, precision searches for new physics or industrial applications. I am confident that AMBiT will remain a scalable, cutting-edge tool for relativistic atomic structure calculations, competitive with any other publicly available atomic code, and can continue to guide theory and experiment for years to come.

Bibliography

[1] S. Gustafsson, P. Jönsson, C. Froese Fischer, and I. Grant. Combining multiconfiguration and perturbation methods: perturbative estimates of core–core electron correlation contributions to excitation energies in Mg-like iron. *Atoms*, **5**, 1:3 (2017).

[2] H. Bekker, C. Hensel, A. Daniel, A. Windberger, T. Pfeifer, and J. R. Crespo López-Urrutia. Laboratory precision measurements of optical emissions from coronal iron. *Phys. Rev. A*, **98**:062514 (2018). URL `http://dx.doi.org/10.1103/PhysRevA.98.062514`.

[3] C. Shah, J. R. C. López-Urrutia, M. F. Gu, T. Pfeifer, J. Marques, F. Grilo, J. P. Santos, and P. Amaro. Revisiting the Fe XVII Line Emission Problem: Laboratory Measurements of the 3s–2p and 3d–2p Line-formation Channels. *The Astrophysical Journal*, **881**, 2:100 (2019). URL `http://dx.doi.org/10.3847/1538-4357/ab2896`.

[4] M. Tanaka, D. Kato, G. Gaigalas, and K. Kawaguchi. Systematic opacity calculations for kilonovae. *Monthly Notices of the Royal Astronomical Society*, **496**, 2:1369–1392 (2020). ISSN 0035-8711. URL `http://dx.doi.org/10.1093/mnras/staa1576`.

[5] C. X. Song, K. Wang, G. D. Zanna, P. Jönsson, R. Si, M. Godefroid, G. Gaigalas, L. Radžiūtė, P. Rynkun, X. H. Zhao, J. Yan, and C. Y. Chen. Large-scale Multiconfiguration Dirac–Hartree–Fock Calculations for Astrophysics: n = 4 Levels in P-like Ions from Mn XI to Ni XIV. *The Astrophysical Journal Supplement Series*, **247**, 2:70 (2020). URL `http://dx.doi.org/10.3847/1538-4365/ab7cc6`.

[6] K. Wang, P. Jönsson, G. D. Zanna, M. Godefroid, Z. B. Chen, C. Y. Chen, and J. Yan. Large-scale Multiconfiguration Dirac–Hartree–Fock Calculations for Astrophysics: Cl-like Ions from Cr viii to Zn xiv. *The Astrophysical Journal Supplement Series*, **246**, 1:1 (2019). URL `http://dx.doi.org/10.3847/1538-4365/ab5530`.

[7] A. Amarsi, N. Grevesse, J. Grumer, M. Asplund, P. Barklem, and R. Collet. The 3D non-LTE solar nitrogen abundance from atomic lines. *Astronomy & Astrophysics*, **636**:A120 (2020).

[8] N. Huntemann, C. Sanner, B. Lipphardt, C. Tamm, and E. Peik. Single-Ion Atomic Clock with 3×10^{-18} Systematic Uncertainty. *Phys. Rev. Lett.*, **6**:063001 (2016). URL `http://dx.doi.org/10.1103/PhysRevLett.116.063001`.

[9] T. L. Nicholson, S. L. Campbell, R. B. Hutson, G. E. Marti, B. J. Bloom, R. L. McNally, W. Zhang, M. D. Barrett, M. S. Safronova, G. F. Strouse, W. L. Tew, and J. Ye. Systematic evaluation of an atomic clock at 2×10^{-18} total uncertainty. *Nat. Comm.*, **6**:6896 (2015). URL `http://dx.doi.org/10.1038/ncomms7896`.

[10] A. D. Ludlow, M. M. Boyd, J. Ye, E. Peik, and P. O. Schmidt. Optical atomic clocks. *Rev. Mod. Phys.*, **87**:637 (2015). URL `http://dx.doi.org/10.1103/RevModPhys.87.637`.

[11] H. Katori. Optical lattice clocks and quantum metrology. *Nature Photonics*, **5**:203–210 (2011). URL `http://dx.doi.org/10.1038/nphoton.2011.45`.

[12] F.-F. Wu, Y.-B. Tang, T.-Y. Shi, and L.-Y. Tang. Magic-intensity trapping of the Mg lattice clock with light shift suppressed below 10^{-19}. *Physical Review A*, **101**, 5:053414 (2020).

[13] F.-F. Wu, Y.-B. Tang, T.-Y. Shi, and L.-Y. Tang. Dynamic multipolar polarizabilities and hyperpolarizabilities of the Sr lattice clock. *Phys. Rev. A*, **100**:042514 (2019). URL `http://dx.doi.org/10.1103/PhysRevA.100.042514`.

[14] J. Cheng, K. Jiang, and L. He. Suppress Blackbody Radiation Shift with Two Clock Ytterbium Clock Transition. *Wuhan University Journal of Natural Sciences*, **24**, 5:427–430 (2019).

[15] A. A. Golovizin, E. S. Kalganova, D. D. Sukachev, G. A. Vishnyakova, D. O. Tregubov, K. Y. Khabarova, V. N. Sorokin, and N. N. Kolachevsky. Methods for determining the polarisability of the fine structure levels in the ground state of the thulium atom. *Quantum Electronics*, **47**, 5:479 (2017).

[16] V. Dzuba, A. Kozlov, and V. Flambaum. Scalar static polarizabilities of lanthanides and actinides. *Physical Review A*, **89**, 4:042507 (2014).

[17] J. Dhiflaoui, M. Bejaoui, M. Farjallah, and H. Berriche. Investigation of the electronic structure of Be^{2+} He and Be^{+} He, and static dipole polarisabilities of the helium atom. *Molecular Physics*, **116**, 10:1347–1357 (2018).

[18] T. Beier, L. Dahl, H.-J. Kluge, C. Kozhuharov, and W. Quint. Trapping ions of hydrogen-like uranium: The HITRAP project at GSI. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, **235**, 1:473 – 478 (2005). ISSN 0168-583X. URL `http://dx.doi.org/https://doi.org/10.1016/j.nimb.2005.03.193`. The Physics of Highly Charged Ions.

[19] H.-J. Kluge, T. Beier, K. Blaum, L. Dahl, S. Eliseev, F. Herfurth, B. Hofmann, O. Kester, S. Koszudowski, C. Kozhuharov, G. Maero, W. Noertershaeuser, J. Pfister, W. Quint, U. Ratzinger, A. Schempp, R. Schuch, T. Stoehlker, R. Thompson, M. Vogel, G. Vorobjev, D. Winters, and G. Werth. HITRAP: A facility at GSI for highly charged ions. *arXiv e-print*, page 0710.5595v1 (2007).

[20] A. Gumberidze, T. Stöhlker, D. Banaś, K. Beckert, P. Beller, H. F. Beyer, F. Bosch, X. Cai, S. Hagmann, C. Kozhuharov, D. Liesen, F. Nolden, X. Ma, P. H. Mokler, M. Steck,

D. Sierpowski, S. Tashenov, A. Warczak, and Y. Zou. Precision tests of QED in strong fields: experiments on hydrogen- and helium-like uranium. *Journal of Physics: Conference Series*, **58**:87–92 (2007). URL `http://dx.doi.org/10.1088/1742-6596/58/1/013`.

[21] H. Häffner, T. Beier, N. Hermanspahn, H.-J. Kluge, W. Quint, S. Stahl, J. Verdú, and G. Werth. High-Accuracy Measurement of the Magnetic Moment Anomaly of the Electron Bound in Hydrogenlike Carbon. *Phys. Rev. Lett.*, **85**:5308–5311 (2000). URL `http://dx.doi.org/10.1103/PhysRevLett.85.5308`.

[22] J. Verdú, S. Djekić, S. Stahl, T. Valenzuela, M. Vogel, G. Werth, T. Beier, H.-J. Kluge, and W. Quint. Electronic g Factor of Hydrogenlike Oxygen $^{16}O^{7+}$. *Phys. Rev. Lett.*, **92**:093002 (2004). URL `http://dx.doi.org/10.1103/PhysRevLett.92.093002`.

[23] M. G. Kozlov, M. S. Safronova, J. R. Crespo López-Urrutia, and P. O. Schmidt. Highly charged ions: Optical clocks and applications in fundamental physics. *Rev. Mod. Phys.*, **90**:045005 (2018). URL `http://dx.doi.org/10.1103/RevModPhys.90.045005`.

[24] A. Payne, C. Chantler, M. Kinnane, J. Gillaspy, L. Hudson, L. Smale, A. Henins, J. Kimpton, and E. Takacs. Helium-like titanium x-ray spectrum as a probe of QED computation. *Journal of Physics B: Atomic, Molecular and Optical Physics*, **47**, 18:185001 (2014).

[25] S. Sanders, R. Silwal, B. Rudramadevi, A. Gall, and E. Takacs. High-resolution X-ray spectroscopy of highly charged tungsten EBIT plasma. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, **431**:47–54 (2018).

[26] R. Silwal, E. Takacs, J. M. Dreiling, J. D. Gillaspy, and Y. Ralchenko. Identification and plasma diagnostics study of extreme ultraviolet transitions in highly charged yttrium. *Atoms*, **5**, 3:30 (2017).

[27] C. T. Chantler, A. T. Payne, J. D. Gillaspy, L. T. Hudson, L. F. Smale, A. Henins, J. A. Kimpton, and E. Takacs. X-ray measurements in helium-like atoms increased discrepancy between experiment and theoretical QED. *New Journal of Physics*, **16**, 12:123037 (2014). URL `http://dx.doi.org/10.1088/1367-2630/16/12/123037`.

[28] C. Chantler, M. Kinnane, J. Gillaspy, L. Hudson, A. Payne, L. Smale, A. Henins, J. Pomeroy, J. Tan, J. Kimpton *et al.* Testing three-body quantum electrodynamics with trapped Ti 20+ ions: Evidence for a Z-dependent divergence between experiment and calculation. *Physical review letters*, **109**, 15:153001 (2012).

[29] J. C. Berengut, V. A. Dzuba, V. V. Flambaum, and A. Ong. Hole transitions in multiply-charged ions for precision laser spectroscopy and searching for α-variation. *Phys. Rev. Lett.*, **106**:210802 (2011).

[30] J. C. Berengut, V. A. Dzuba, and V. V. Flambaum. Enhanced Laboratory Sensitivity to Variation of the Fine-Structure Constant using Highly Charged Ions. *Phys. Rev. Lett.*, **105**:120801 (2010). URL `http://dx.doi.org/10.1103/PhysRevLett.105.120801`.

[31] S. Bashkin. Beam Foil Spectroscopy. *Appl. Opt.*, **7**, 12:2341–2350 (1968). URL http://dx.doi.org/10.1364/AO.7.002341.

[32] H. G. Berry. Beam-foil spectroscopy. *Reports on Progress in Physics*, **40**, 2:155–217 (1977). URL http://dx.doi.org/10.1088/0034-4885/40/2/002.

[33] M. A. Levine, R. E. Marrs, J. R. Henderson, D. A. Knapp, and M. B. Schneider. The Electron Beam Ion Trap: A New Instrument for Atomic Physics Measurements. *Physica Scripta*, **T22**:157–163 (1988). URL http://dx.doi.org/10.1088/0031-8949/1988/t22/024.

[34] B. M. Penetrante, J. N. Bardsley, D. DeWitt, M. Clark, and D. Schneider. Evolution of ion-charge-state distributions in an electron-beam ion trap. *Phys. Rev. A*, **43**:4861–4872 (1991). URL http://dx.doi.org/10.1103/PhysRevA.43.4861.

[35] P. Micke, T. Leopold, S. King, E. Benkler, L. Spieß, L. Schmoeger, M. Schwarz, J. C. López-Urrutia, and P. Schmidt. Coherent laser spectroscopy of highly charged ions using quantum logic. *Nature*, pages 1–6 (2020).

[36] P. Indelicato. QED tests with highly charged ions. *Journal of Physics B: Atomic, Molecular and Optical Physics*, **52**, 23:232001 (2019). URL http://dx.doi.org/10.1088/1361-6455/ab42c9.

[37] O.-O. Versolato, M. Schwarz, A. Windberger, J. Ullrich, P.-O. Schmidt, M. Drewsen, and J.-R. Crespo López-Urrutia. Cold highly charged ions in a cryogenic Paul trap. *Hyperfine Interactions*, **214**, 1:189–194 (2013). ISSN 1572-9540. URL http://dx.doi.org/10.1007/s10751-013-0806-9.

[38] L. Gruber, J. P. Holder, J. Steiger, B. R. Beck, H. E. DeWitt, J. Glassman, J. W. McDonald, D. A. Church, and D. Schneider. Evidence for Highly Charged Ion Coulomb Crystallization in Multicomponent Strongly Coupled Plasmas. *Phys. Rev. Lett.*, **86**:636–639 (2001). URL http://dx.doi.org/10.1103/PhysRevLett.86.636.

[39] L. Gruber, J. P. Holder, and D. Schneider. Formation of Strongly Coupled Plasmas from MultiComponent Ions in a Penning Trap. *Physica Scripta*, **71**, 1:60–107 (2005). URL http://dx.doi.org/10.1088/0031-8949/71/1/010.

[40] L. Schmöger, O. O. Versolato, M. Schwarz, M. Kohnen, A. Windberger, B. Piest, S. Feuchtenbeiner, J. Pedregosa-Gutierrez, T. Leopold, P. Micke, A. K. Hansen, T. M. Baumann, M. Drewsen, J. Ullrich, P. O. Schmidt, and J. R. C. López-Urrutia. Coulomb crystallization of highly charged ions. *Science*, **347**, 6227:1233–1236 (2015). ISSN 0036-8075. URL http://dx.doi.org/10.1126/science.aaa2960.

[41] J. Dreiling, A. Naing, J. Tan, J. Hanson, S. Hoogerheide, and S. Brewer. Capture of highly charged ions in a pseudo-hyperbolic Paul trap. *Journal of Applied Physics*, **126**, 2:024501 (2019).

[42] R. E. March. Quadrupole ion traps. *Mass Spectrometry Reviews*, **28**, 6:961–989 (2009). URL http://dx.doi.org/10.1002/mas.20250.

[43] P. O. Schmidt, T. Rosenband, C. Langer, W. M. Itano, J. C. Bergquist, and D. J. Wineland. Spectroscopy using quantum logic. *Science*, **309**, 5735:749–752 (2005). ISSN 0036-8075. URL `https://science.sciencemag.org/content/309/5735/749`.

[44] S. M. Brewer, J.-S. Chen, A. M. Hankin, E. R. Clements, C. W. Chou, D. J. Wineland, D. B. Hume, and D. R. Leibrandt. $^{27}\mathrm{Al}^{+}$ Quantum-Logic Clock with a Systematic Uncertainty below 10^{-18}. *Phys. Rev. Lett.*, **123**:033201 (2019). URL `http://dx.doi.org/10.1103/PhysRevLett.123.033201`.

[45] F. Wolf, Y. Wan, J. C. Heip, F. Gebert, C. Shi, and P. O. Schmidt. Non-destructive state detection for quantum logic spectroscopy of molecular ions. *Nature*, **530**, 7591:457–460 (2016).

[46] M. Shi, P. Herskind, M. Drewsen, and I. Chuang. Microwave quantum logic spectroscopy and control of molecular ions. *New Journal of Physics*, **15**, 11:113019 (2013).

[47] A. Derevianko, V. A. Dzuba, and V. V. Flambaum. Highly Charged Ions as a Basis of Optical Atomic Clockwork of Exceptional Accuracy. *Phys. Rev. Lett.*, **109**:180801 (2012). URL `http://dx.doi.org/10.1103/PhysRevLett.109.180801`.

[48] A. Windberger, J. R. Crespo López-Urrutia, H. Bekker, N. S. Oreshkina, J. C. Berengut, V. Bock, A. Borschevsky, V. A. Dzuba, E. Eliav, Z. Harman, U. Kaldor, S. Kaul, U. I. Safronova, V. V. Flambaum, C. H. Keitel, P. O. Schmidt, J. Ullrich, and O. O. Versolato. Identification of the Predicted $5s{-}4f$ Level Crossing Optical Lines with Applications to Metrology and Searches for the Variation of Fundamental Constants. *Phys. Rev. Lett.*, **114**:150801 (2015). URL `http://dx.doi.org/10.1103/PhysRevLett.114.150801`.

[49] F. Torretti, A. Windberger, A. Ryabtsev, S. Dobrodey, H. Bekker, W. Ubachs, R. Hoekstra, E. V. Kahl, J. C. Berengut, J. R. C. López-Urrutia, and O. O. Versolato. Optical spectroscopy of complex open-$4d$-shell ions Sn^{7+} - Sn^{10+}. *Phys. Rev. A*, **95**:042503 (2017). URL `http://dx.doi.org/10.1103/PhysRevA.95.042503`.

[50] V. Y. Banine, K. N. Koshelev, and G. H. P. M. Swinkels. Physical processes in EUV sources for microlithography. *Journal of Physics D: Applied Physics*, **44**, 25:253001 (2011). URL `http://dx.doi.org/10.1088/0022-3727/44/25/253001`.

[51] S. Hofmann and G. Münzenberg. The discovery of the heaviest elements. *Reviews of Modern Physics*, **72**, 3:733 (2000).

[52] Y. T. Oganessian and V. K. Utyonkov. Super-heavy element research. *Reports on Progress in Physics*, **78**, 3:036301 (2015). URL `http://dx.doi.org/10.1088/0034-4885/78/3/036301`.

[53] Y. Oganessian. Heaviest nuclei from ^{48}Ca-induced reactions. *Journal of Physics G: Nuclear and Particle Physics*, **34**, 4:R165–R242 (2007). URL `http://dx.doi.org/10.1088/0954-3899/34/4/r01`.

[54] V. Y. Denisov and S. Hofmann. Formation of superheavy elements in cold fusion reactions. *Physical Review C*, **61**, 3:034606 (2000).

[55] Y. T. Oganessian, V. K. Utyonkov, Y. V. Lobanov, F. S. Abdullin, A. N. Polyakov, R. N. Sagaidak, I. V. Shirokovsky, Y. S. Tsyganov, A. A. Voinov, G. G. Gulbekian, S. L. Bogomolov, B. N. Gikal, A. N. Mezentsev, S. Iliev, V. G. Subbotin, A. M. Sukhov, K. Subotic, V. I. Zagrebaev, G. K. Vostokin, M. G. Itkis, K. J. Moody, J. B. Patin, D. A. Shaughnessy, M. A. Stoyer, N. J. Stoyer, P. A. Wilk, J. M. Kenneally, J. H. Landrum, J. F. Wild, and R. W. Lougheed. Synthesis of the isotopes of elements 118 and 116 in the ^{249}Cf and $^{245}\mathrm{Cm} +^{48}$ Ca fusion reactions. *Phys. Rev. C*, **74**:044602 (2006). URL `http://dx.doi.org/10.1103/PhysRevC.74.044602`.

[56] S. Hofmann, S. Heinz, R. Mann, J. Maurer, G. Münzenberg, S. Antalic, W. Barth, H. Burkhard, L. Dahl, K. Eberhardt *et al.* Review of even element super-heavy nuclei and search for element 120. *The European Physical Journal A*, **52**, 6:180 (2016).

[57] H. Haba. A new period in superheavy-element hunting. *Nature chemistry*, **11**, 1:10–13 (2019).

[58] T. Sato, M. Asai, A. Borschevsky, T. Stora, N. Sato, Y. Kaneya, K. Tsukada, C. E. Düllmann, K. Eberhardt, E. Eliav *et al.* Measurement of the first ionization potential of lawrencium, element 103. *Nature*, **520**, 7546:209–211 (2015).

[59] T. K. Sato, M. Asai, A. Borschevsky, R. Beerwerth, Y. Kaneya, H. Makii, A. Mitsukai, Y. Nagame, A. Osa, A. Toyoshima *et al.* First ionization potentials of Fm, Md, No, and Lr: verification of filling-up of 5f electrons and confirmation of the actinide series. *Journal of the American Chemical Society*, **140**, 44:14609–14613 (2018).

[60] P. Chhetri, D. Ackermann, H. Backe, M. Block, B. Cheal, C. Droese, C. E. Düllmann, J. Even, R. Ferrer, F. Giacoppo, S. Götz, F. P. Heßberger, M. Huyse, O. Kaleja, J. Khuyagbaatar, P. Kunz, M. Laatiaoui, F. Lautenschläger, W. Lauth, N. Lecesne, L. Lens, E. Minaya Ramirez, A. K. Mistry, S. Raeder, P. Van Duppen, T. Walther, A. Yakushev, and Z. Zhang. Precision Measurement of the First Ionization Potential of Nobelium. *Phys. Rev. Lett.*, **120**:263003 (2018). URL `http://dx.doi.org/10.1103/PhysRevLett.120.263003`.

[61] M. Laatiaoui, W. Lauth, H. Backe, M. Block, D. Ackermann, B. Cheal, P. Chhetri, C. E. Düllmann, P. van Duppen, J. Even, R. Ferrer, F. Giacoppo, S. Götz, F. P. Heŝberger, M. Huyse, O. Kaleja, J. Khuyagbaatar, P. Kunz, F. Lautenschläger, A. K. Mistry, S. Raeder, E. M. Ramirez, T. Walther, C. Wraith, and A. Yakushev. Atom-at-a-time laser resonance ionization spectroscopy of nobelium. *Nature*, **538**, 7626 (2016). ISSN 00280836.

[62] S. Raeder, D. Ackermann, H. Backe, R. Beerwerth, J. C. Berengut, M. Block, A. Borschevsky, B. Cheal, P. Chhetri, C. E. Düllmann, V. A. Dzuba, E. Eliav, J. Even, R. Ferrer, V. V. Flambaum, S. Fritzsche, F. Giacoppo, S. Götz, F. P. Heßberger, M. Huyse, U. Kaldor, O. Kaleja, J. Khuyagbaatar, P. Kunz, M. Laatiaoui, F. Lautenschläger, W. Lauth, A. K. Mistry, E. Minaya Ramirez, W. Nazarewicz, S. G. Porsev, M. S. Safronova, U. I. Safronova, B. Schuetrumpf, P. Van Duppen, T. Walther, C. Wraith, and A. Yakushev. Probing Sizes and Shapes of Nobelium Isotopes by Laser Spectroscopy. *Phys. Rev. Lett.*, **120**:232503 (2018). URL `http://dx.doi.org/10.1103/PhysRevLett.120.232503`.

[63] R. Ferrer, A. Barzakh, B. Bastin, R. Beerwerth, M. Block, P. Creemers, H. Grawe, R. de Groote, P. Delahaye, X. Fléchard, S. Franchoo, S. Fritzsche, L. P. Gaffney, L. Ghys, W. Gins, C. Granados, R. Heinke, L. Hijazi, M. Huyse, T. Kron, Y. Kudryavtsev, M. Laatiaoui, N. Lecesne, M. Loiselet, F. Lutton, I. D. Moore, Y. Martínez, E. Mogilevskiy, P. Naubereit, J. Piot, S. Raeder, S. Rothe, H. Savajols, S. Sels, V. Sonnenschein, J.-C. Thomas, E. Traykov, C. Van Beveren, P. Van Den Bergh, P. Van Duppen, K. Wendt, and A. Zadvornaya. Towards high-resolution laser ionization spectroscopy of the heaviest elements in supersonic gas jet expansion. *Nature communications*, **8** (2017). ISSN 2041-1723.

[64] E. Eliav, U. Kaldor, and A. Borschevsky. *Electronic Structure of the Transactinide Atoms*, pages 1–16. American Chemical Society (2018). ISBN 9781119951438. URL `http://dx.doi.org/10.1002/9781119951438.eibc2632`.

[65] P. Schwerdtfeger, L. F. Pašteka, A. Punnett, and P. O. Bowman. Relativistic and quantum electrodynamic effects in superheavy elements. *Nuclear Physics A*, **944**:551 – 577 (2015). ISSN 0375-9474. URL `http://dx.doi.org/https://doi.org/10.1016/j.nuclphysa.2015.02.005`. Special Issue on Superheavy Elements.

[66] V. V. Flambaum and J. S. M. Ginges. Radiative potential and calculations of QED radiative corrections to energy levels and electromagnetic amplitudes in many-electron atoms. *Phys. Rev. A*, **72**:052115 (2005).

[67] J. S. M. Ginges and J. C. Berengut. Atomic many-body effects and Lamb shifts in alkali metals. *Phys. Rev. A*, **93**:052509 (2016).

[68] J. S. M. Ginges and J. C. Berengut. QED radiative corrections and many-body effects in atoms: vacuum polarization and binding energy shifts in alkali metals. *J. Phys. B*, **49**:095001 (2016).

[69] V. M. Shabaev, I. I. Tupitsyn, and V. A. Yerokhin. Model operator approach to the Lamb shift calculations in relativistic many-electron atoms. *Phys. Rev. A*, **88**:012513 (2013). URL `http://dx.doi.org/10.1103/PhysRevA.88.012513`.

[70] J. Lowe, C. Chantler, and I. Grant. Self-energy screening approximations in multi-electron atoms. *Radiation Physics and Chemistry*, **85**:118 – 123 (2013). ISSN 0969-806X. URL `http://dx.doi.org/https://doi.org/10.1016/j.radphyschem.2013.01.004`.

[71] J. Beacham, C. Burrage, D. Curtin, A. De Roeck, J. Evans, J. Feng, C. Gatto, S. Gninenko, A. Hartin, I. Irastorza *et al.* Physics beyond colliders at CERN: beyond the Standard Model working group report. *Journal of Physics G: Nuclear and Particle Physics*, **47**, 1:010501 (2019).

[72] M. Safronova, D. Budker, D. DeMille, D. F. J. Kimball, A. Derevianko, and C. W. Clark. Search for new physics with atoms and molecules. *Reviews of Modern Physics*, **90**, 2:025008 (2018).

[73] J. C. Berengut, D. Budker, C. Delaunay, V. V. Flambaum, C. Frugiuele, E. Fuchs, C. Grojean, R. Harnik, R. Ozeri, G. Perez, and Y. Soreq. Probing New Long-Range Interactions by Isotope Shift Spectroscopy. *Phys. Rev. Lett.*, **120**:091801 (2018). URL `http://dx.doi.org/10.1103/PhysRevLett.120.091801`.

[74] J. K. Webb, V. V. Flambaum, C. W. Churchill, M. J. Drinkwater, and J. D. Barrow. Search for Time Variation of the Fine Structure Constant. *Phys. Rev. Lett.*, **82**:884–887 (1999). URL `http://dx.doi.org/10.1103/PhysRevLett.82.884`.

[75] W. Ubachs, J. Koelemeij, K. Eikema, and E. Salumbides. Physics beyond the Standard Model from hydrogen spectroscopy. *Journal of Molecular Spectroscopy*, **320**:1 – 12 (2016). ISSN 0022-2852. URL `http://dx.doi.org/https://doi.org/10.1016/j.jms.2015.12.003`.

[76] M. G. Kozlov and S. A. Levshakov. Microwave and submillimeter molecular transitions and their dependence on fundamental constants. *Annalen der Physik*, **525**, 7:452–471 (2013). URL `http://dx.doi.org/10.1002/andp.201300010`.

[77] J. Ginges and V. V. Flambaum. Violations of fundamental symmetries in atoms and tests of unification theories of elementary particles. *Physics Reports*, **397**, 2:63–154 (2004).

[78] B. Roberts, V. Dzuba, and V. Flambaum. Parity and time-reversal violation in atomic systems. *Annual Review of Nuclear and Particle Science*, **65**:63–86 (2015).

[79] O. Sushkov, V. Flambaum, and I. Khriplovich. Possibility of investigating P-and T-odd nuclear forces in atomic and molecular experiments. *Sov. Phys. JETP*, **60**:873 (1984). URL `http://www.jetp.ac.ru/cgi-bin/dn/e_060_05_0873.pdf`.

[80] T. Fukuyama. Searching for new physics beyond the standard model in electric dipole moment. *International Journal of Modern Physics A*, **27**, 16:1230015 (2012). URL `https://doi.org/10.1142/S0217751X12300153`.

[81] J. D. Gillaspy. Precision spectroscopy of trapped highly charged heavy elements: pushing the limits of theory and experiment. *Physica Scripta*, **89**, 11:114004 (2014). URL `http://dx.doi.org/10.1088/0031-8949/89/11/114004`.

[82] J. C. Berengut, C. Delaunay, A. Geddes, and Y. Soreq. Generalized King linearity and new physics searches with isotope shifts. *arXiv preprint arXiv:2005.06144* (2020).

[83] I. Counts, J. Hur, D. P. Craik, H. Jeon, C. Leung, J. Berengut, A. Geddes, A. Kawasaki, W. Jhe, and V. Vuletić. Observation of 3σ Nonlinear Isotope Shift in Yb^+ Search for New Boson. *arXiv preprint arXiv:2004.11383* (2020).

[84] R. Silwal, A. Lapierre, J. D. Gillaspy, J. M. Dreiling, S. Blundell, A. Borovik Jr, G. Gwinner, A. Villari, Y. Ralchenko, E. Takacs *et al.* Measuring the difference in nuclear charge radius of Xe isotopes by EUV spectroscopy of highly charged Na-like ions. *Physical Review A*, **98**, 5:052502 (2018).

[85] V. A. Dzuba, V. V. Flambaum, and J. K. Webb. Space-Time Variation of Physical Constants and Relativistic Corrections in Atoms. *Phys. Rev. Lett.*, **82**:888–891 (1999). URL `http://dx.doi.org/10.1103/PhysRevLett.82.888`.

[86] M. S. Safronova. The search for variation of fundamental constants with clocks. *Annalen der Physik*, **531**, 5:1800364 (2019).

[87] K. Van Tilburg, N. Leefer, L. Bougas, and D. Budker. Search for Ultralight Scalar Dark Matter with Atomic Spectroscopy. *Phys. Rev. Lett.*, **115**:011802 (2015). URL `http://dx.doi.org/10.1103/PhysRevLett.115.011802`.

[88] A. Derevianko and M. Pospelov. Hunting for topological dark matter with atomic clocks. *Nature Physics*, **10**, 12:933–936 (2014).

[89] B. M. Roberts, G. Blewitt, C. Dailey, M. Murphy, M. Pospelov, A. Rollings, J. Sherman, W. Williams, and A. Derevianko. Search for domain wall dark matter with atomic clocks on board global positioning system satellites. *Nature communications*, **8**, 1:1–9 (2017).

[90] A. Derevianko. Detecting dark-matter waves with a network of precision-measurement tools. *Phys. Rev. A*, **97**:042506 (2018). URL `http://dx.doi.org/10.1103/PhysRevA.97.042506`.

[91] A. Hees, J. Guéna, M. Abgrall, S. Bize, and P. Wolf. Searching for an oscillating massive scalar field as a dark matter candidate using atomic hyperfine frequency comparisons. *Physical review letters*, **117**, 6:061301 (2016).

[92] V. V. Flambaum, A. J. Geddes, and A. V. Viatkina. Isotope shift, nonlinearity of King plots, and the search for new particles. *Phys. Rev. A*, **97**:032510 (2018). URL `http://dx.doi.org/10.1103/PhysRevA.97.032510`.

[93] V. Flambaum and V. Dzuba. Sensitivity of the isotope shift to the distribution of nuclear charge density. *Physical Review A*, **100**, 3:032511 (2019).

[94] E. Kahl and J. Berengut. AMBiT: A programme for high-precision relativistic atomic structure calculations. *Computer Physics Communications*, **238**:232 – 243 (2019). ISSN 0010-4655. URL `http://dx.doi.org/https://doi.org/10.1016/j.cpc.2018.12.014`.

[95] V. A. Dzuba, V. V. Flambaum, and M. G. Kozlov. Combination of the many-body perturbation theory with the configuration-interaction method. *Phys. Rev. A*, **54**:3948 (1996).

[96] J. C. Berengut. Particle-hole configuration interaction and many-body perturbation theory: Application to Hg^{+}. *Phys. Rev. A*, **94**:012502 (2016).

[97] M. G. Kozlov, S. G. Porsev, M. S. Safronova, and I. I. Tupitsyn. CI-MBPT: A package of programs for relativistic atomic calculations based on a method combining configuration interaction and many-body perturbation theory. *Computer Physics Communications*, **195**:199 – 213 (2015). ISSN 0010-4655. URL `http://dx.doi.org/doi:10.1016/j.cpc.2015.05.007`.

[98] S. G. Porsev, M. G. Kozlov, Y. G. Rakhlina, and A. Derevianko. Many-body calculations of electric-dipole amplitudes for transitions between low-lying levels of Mg, Ca, Sr. *Phys. Rev. A*, **64**:012508 (2001).

[99] S. Fritzsche, C. Fischer, and G. Gaigalas. RELCI: A program for relativistic configuration interaction calculations. *Computer Physics Communications*, **148**, 1:103 – 123 (2002). ISSN 0010-4655. URL `http://dx.doi.org/http://dx.doi.org/10.1016/S0010-4655(02)00463-0`.

[100] J. C. Berengut, V. V. Flambaum, and M. G. Kozlov. Calculation of isotope shifts and relativistic shifts in CI, CII, CIII, and CIV. *Phys. Rev. A*, **73**:012504 (2006).

[101] I. M. Savukov. Configuration-interaction many-body-perturbation-theory energy levels of four-valent Si I. *Phys. Rev. A*, **91**:022514 (2015). URL `http://dx.doi.org/10.1103/PhysRevA.91.022514`.

[102] J. C. Berengut. Isotope shifts and relativistic shifts of CrII for the study of α variation in quasar absorption spectra. *Phys. Rev. A*, **84**:052520 (2011). URL `http://dx.doi.org/10.1103/PhysRevA.84.052520`.

[103] J. C. Berengut, V. V. Flambaum, and M. G. Kozlov. Isotope shift calculations in Ti II. *Journal of Physics B: Atomic, Molecular and Optical Physics*, **41**, 23:235702 (2008). URL `http://dx.doi.org/10.1088/0953-4075/41/23/235702`.

[104] J. Jiang, J. Mitroy, Y. Cheng, and M. W. J. Bromley. Relativistic semiempirical-core-potential calculations of Sr^+ using Laguerre and Slater spinors. *Phys. Rev. A*, **94**:062514 (2016). URL `http://dx.doi.org/10.1103/PhysRevA.94.062514`.

[105] P. Jönsson, G. Gaigalas, J. Bieron, C. Fischer, and I. Grant. New version: GRASP2K relativistic atomic structure package. *Computer Physics Communications*, **184**, 9:2197–2203 (2013). ISSN 0010-4655.

[106] P. Jönsson, X. He, C. Froese Fischer, and I. P. Grant. The GRASP2K relativistic atomic structure package. *Comp. Phys. Comm.*, **177**:597–622 (2007). URL `http://dx.doi.org/10.1016/j.cpc.2007.06.002`.

[107] J. Desclaux. A multiconfiguration relativistic DIRAC-FOCK program. *Computer Physics Communications*, **9**, 1:31 – 45 (1975). ISSN 0010-4655. URL `http://dx.doi.org/https://doi.org/10.1016/0010-4655(75)90054-5`.

[108] S. Fritzsche. A fresh computational approach to atomic structures, processes and cascades. *Computer Physics Communications*, **240**:1–14 (2019).

[109] M. F. Gu. The flexible atomic code. *Canadian Journal of Physics*, **86**, 5:675–689 (2008). URL `http://dx.doi.org/10.1139/p07-197`.

[110] M. Gu. New benchmark of X-ray line emission models of Fe XVII. *arXiv preprint arXiv:0905.0519* (2009).

[111] V. A. Dzuba and V. V. Flambaum. Exponential Increase of Energy Level Density in Atoms: Th and Th II. *Phys. Rev. Lett.*, **104**:213002 (2010).

[112] V. A. Dzuba. V^{N-M} potential for atomic calculations. *Phys. Rev. A*, **71**:032512 (2005). URL `http://dx.doi.org/10.1103/PhysRevA.71.032512`.

[113] A. J. Geddes, D. A. Czapski, E. V. Kahl, and J. C. Berengut. Saturated-configuration-interaction calculations for five-valent Ta and Db. *Phys. Rev. A*, **98**:042508 (2018). URL `http://dx.doi.org/10.1103/PhysRevA.98.042508`.

[114] W. R. Johnson. *Atomic Structure Theory : lectures on atomic physics.* Springer, Berlin ; London (2007). ISBN 3540680101.

[115] W. R. Johnson, S. A. Blundell, and J. Sapirstein. Finite basis sets for the Dirac equation constructed from B splines. *Phys. Rev. A*, **37**:307 (1988).

[116] K. Beloy and A. Derevianko. Application of the dual-kinetic-balance sets in the relativistic many-body problem of atomic structure. *Comp. Phys. Commun.*, **179**:310 (2008).

[117] S. Porsev, K. Beloy, and A. Derevianko. Precision determination of weak charge of Cs 133 from atomic parity violation. *Physical Review D*, **82**, 3:036008 (2010).

[118] V. A. Dzuba, J. C. Berengut, V. V. Flambaum, and B. Roberts. Revisiting Parity Non-conservation in Cesium. *Phys. Rev. Lett.*, **109**:203003 (2012).

[119] Y.-H. Zhang, F.-F. Wu, P.-P. Zhang, L.-Y. Tang, J.-Y. Zhang, K. G. H. Baldwin, and T.-Y. Shi. QED and relativistic nuclear recoil corrections to the 413-nm tune-out wavelength for the 2 $^3S_1$ state of helium. *Phys. Rev. A*, **99**:040502 (2019). URL `http://dx.doi.org/10.1103/PhysRevA.99.040502`.

[120] Y.-B. Tang, B.-Q. Lou, and T.-Y. Shi. Relativistic multireference coupled-cluster theory based on a *B*-spline basis: Application to atomic francium. *Phys. Rev. A*, **96**:022513 (2017). URL `http://dx.doi.org/10.1103/PhysRevA.96.022513`.

[121] J. Berengut, V. Dzuba, and V. Flambaum. Transitions in Zr, Hf, Ta, W, Re, Hg, Ac, and U ions with high sensitivity to variation of the fine-structure constant. *Physical Review A*, **84**, 5:054501 (2011).

[122] C. F. Fischer and F. A. Parpia. Accurate spline solutions of the radial Dirac equation. *Physics Letters A*, **179**, 3:198 – 204 (1993). ISSN 0375-9601. URL `http://dx.doi.org/https://doi.org/10.1016/0375-9601(93)91138-U`.

[123] M. G. Kozlov and S. G. Porsev. Effective Hamiltonian for valence electrons of an atom. *Opt. Spectrosc.*, **87**:352 (1999).

[124] W. R. Johnson, C. D. Lin, K. T. Cheng, and C. M. Lee. Relativistic Random-Phase Approximation. *Physica Scripta*, **21**, 3-4:409 (1980). URL `http://iopscience.iop.org/1402-4896/21/3-4/029`.

[125] V. A. Dzuba, V. V. Flambaum, and O. P. Sushkov. Relativistic many-body calculations of the hyperfine-structure intervals in caesium and francium atoms. *Journal of Physics*

B: Atomic and Molecular Physics, **17**, 10:1953 (1984). URL `http://stacks.iop.org/0022-3700/17/i=10/a=005`.

[126] V. Dzuba, V. Flambaum, P. Silvestrov, and O. Sushkov. Shielding of an external electric field in atoms. *Physics Letters A*, **118**, 4:177 – 180 (1986). ISSN 0375-9601. URL `http://dx.doi.org/https://doi.org/10.1016/0375-9601(86)90251-3`.

[127] V. A. Dzuba, J. C. Berengut, J. S. M. Ginges, and V. V. Flambaum. Screening of an oscillating external electric field in atoms. *Phys. Rev. A*, **98**:043411 (2018). URL `http://dx.doi.org/10.1103/PhysRevA.98.043411`.

[128] M. W. J. Bromley and J. Mitroy. Convergence of partial wave expansion of the He ground state. *Int. J. Quantum Chem.*, **107**:1150 (2007).

[129] V. A. Dzuba, J. C. Berengut, C. Harabati, and V. V. Flambaum. Combining configuration interaction with perturbation theory for atoms with a large number of valence electrons. *Physical Review A*, **95**, 1:012503 (2017).

[130] V. Dzuba, V. Flambaum, and M. Kozlov. Fast configuration-interaction calculations for nobelium and ytterbium. *Physical Review A*, **99**, 3:032501 (2019).

[131] J. L. Hennessy and D. A. Patterson. *Computer Architecture: A Quantitative Approach*. Elsevier (2019).

[132] National Computational Infrastructure. HPC Systems at NCI (2020). URL `http://nci.org.au/our-systems/hpc-systems`.

[133] OpenMP Architecture Review Board. *OpenMP Application Program Interface, Version 4.0* (2013).

[134] Arm Ltd. Arm Forge (2019). URL `https://www.arm.com/products/development-tools/server-and-hpc/forge`.

[135] P. E. McKenney. Is Parallel Programming Hard, And, If So, What Can You Do About It? (v2017.01.02a). *CoRR*, **abs/1701.00854** (2017). URL `http://arxiv.org/abs/1701.00854`.

[136] B. Stroustrup. *The C++ programming language*. Pearson Education (2013).

[137] J. LaGrone, A. Aribuki, and B. Chapman. A set of microbenchmarks for measuring OpenMP task overheads. In *Proceedings of the International Conference on Parallel and Distributed Processing Techniques and Applications (PDPTA)*, page 1. Citeseer (2011).

[138] A. Stathopoulos and C. Froese Fischer. A Davidson program for finding a few selected extreme eigenpairs of a large, sparse, real, symmetric matrix. *Comput. Phys. Commun.*, **79**:268 (1994).

[139] G. Guennebaud, B. Jacob *et al.* Eigen v3. http://eigen.tuxfamily.org (2010).

[140] Google Inc. sparsehash: C++ associative containers. GitHub repository. URL `https://github.com/sparsehash/sparsehash`.

[141] M. T. Goodrich, R. Tamassia, and M. H. Goldwasser. *Data structures and algorithms in Java*. John Wiley & Sons (2014).

[142] M. Ankerl. Hashmaps Benchmarks - Finding the Fastest, Memory Efficient Hashmap. Blog Post (2019). URL `https://martin.ankerl.com/2019/04/01/hashmap-benchmarks-01-overview/`.

[143] Boost C++ Libraries. *boost: Class Template flat_ map*, 1.72.0 edition (2019). URL `https://www.boost.org/doc/libs/1_72_0/doc/html/boost/container/flat_map.html`.

[144] D. Vandevoorde and N. M. Josuttis. *C++ Templates: The Complete Guide, Portable Documents*. Addison-Wesley Professional (2002).

[145] M. D. Hill and M. R. Marty. Amdahl's Law in the Multicore Era. *Computer*, **41**, 7:33–38 (2008).

[146] A. Kramida, Y. Ralchenko, J. Reader, and NIST ASD Team. NIST Atomic Spectra Database (v5.4) (2016). URL `http://physics.nist.gov/asd`.

[147] B. Arcimowicz, J. Dembczyński, P. Głowacki, J. Ruczkowski, M. Elantkowska, G. H. Guthöhrlein, and L. Windholz. Progress in the analysis of the even parity configurations of tantalum atom. *The European Physical Journal Special Topics*, **222**, 9:2085–2102 (2013). ISSN 1951-6401. URL `http://dx.doi.org/10.1140/epjst/e2013-01988-6`.

[148] V. A. Dhaka, E. R. Clark, and A. Pinck. Photochemical fabrication process (1971). US Patent 3,573,975.

[149] J. Lof, A. T. A. M. Derksen, C. A. Hoogendam, A. Kolesnychenko, E. R. Loopstra, T. M. Modderman, J. C. H. Mulkens, R. A. S. Ritsema, K. Simon, J. T. De Smit *et al.* Lithographic apparatus and device manufacturing method (2005). US Patent 6,952,253.

[150] O. O. Versolato. Physics of laser-driven tin plasma sources of EUV radiation for nanolithography. *Plasma Sources Science and Technology*, **28**, 8:083001 (2019). URL `http://dx.doi.org/10.1088/1361-6595/ab3302`.

[151] G. O'Sullivan, P. Dunne, T. Higashiguchi, D. Kos, O. Maguire, T. Miyazaki, F. O'Reilly, J. Sheil, E. Sokell, and D. Kilbane. Spectroscopy of highly charged ions for extreme ultraviolet lithography. *Nuclear Instruments and Methods in Physics Research Section B: Beam Interactions with Materials and Atoms*, **408**:3 – 8 (2017). ISSN 0168-583X. URL `http://dx.doi.org/https://doi.org/10.1016/j.nimb.2017.03.114`. Proceedings of the 18th International Conference on the Physics of Highly Charged Ions (HCI-2016), Kielce, Poland, 11-16 September 2016.

[152] Q. Huang, V. Medvedev, R. van de Kruijs, A. Yakshin, E. Louis, and F. Bijkerk. Spectral tailoring of nanoscale EUV and soft x-ray multilayer optics. *Applied Physics Reviews*, **4**, 1:011104 (2017). URL `http://dx.doi.org/10.1063/1.4978290`.

[153] S. Bajt, J. B. Alameda, T. W. B. Jr., W. M. Clift, J. A. Folta, B. B. Kaufmann, and E. A. Spiller. Improved reflectance and stability of Mo/Si multilayers. *Optical Engineering*, **41**, 8:1797 – 1804 (2002). URL `http://dx.doi.org/10.1117/1.1489426`.

[154] V. I. Azarov and Y. N. Joshi. Analysis of the $4d^7$ - $4d^6$ 5p transition array of the eighth spectrum of tin: Sn VIII. *J. Phys. B*, **26**, 20:3495–3514 (1993). ISSN 0953-4075.

[155] C. Nielson and G. F. Koster. *Spectroscopic Coefficients for the p^n, d^n, and f^n Configurations.* MIT press (1963).

[156] A. Kramida. The program LOPT for least-squares optimization of energy levels. *Computer Physics Communications*, **182**, 2:419–434 (2011).

[157] S. S. Churilov and A. N. Ryabtsev. Analyses of the Sn IX–Sn XII spectra in the EUV region. *Phys. Scr.*, **73**, 6:614 (2006).

[158] J. Scheers, A. Ryabtsev, A. Borschevsky, J. Berengut, K. Haris, R. Schupp, D. Kurilovich, F. Torretti, A. Bayerle, E. Eliav *et al.* Energy-level structure of Sn^{3+} ions. *Physical Review A*, **98**, 6:062503 (2018).

[159] J. Scheers, C. Shah, A. Ryabtsev, H. Bekker, F. Torretti, J. Sheil, D. Czapski, J. Berengut, W. Ubachs, J. C. López-Urrutia *et al.* EUV spectroscopy of highly charged Sn^{13+}–Sn^{15+} ions in an electron-beam ion trap. *Physical Review A* (2020).

[160] E. V. Kahl, J. C. Berengut, M. Laatiaoui, E. Eliav, and A. Borschevsky. High-precision ab initio calculations of the spectrum of Lr^+. *Phys. Rev. A*, **100**:062505 (2019). URL `http://dx.doi.org/10.1103/PhysRevA.100.062505`.

[161] M. Laatiaoui and S. Raeder. Laser Spectroscopy of the Heaviest Elements: One Atom at a Time. *Nuclear Physics News*, **29**, 1:21–25 (2019). URL `http://dx.doi.org/https://doi.org/10.1080/10619127.2019.1571804`.

[162] A. Zadvornaya, P. Creemers, K. Dockx, R. Ferrer, L. P. Gaffney, W. Gins, C. Granados, M. Huyse, Y. Kudryavtsev, M. Laatiaoui, E. Mogilevskiy, S. Raeder, S. Sels, P. Van den Bergh, P. Van Duppen, M. Verlinde, E. Verstraelen, M. Nabuurs, D. Reynaerts, and P. Papadakis. Characterization of Supersonic Gas Jets for High-Resolution Laser Ionization Spectroscopy of Heavy Elements. *Phys. Rev. X*, **8**:041008 (2018). URL `http://dx.doi.org/10.1103/PhysRevX.8.041008`.

[163] O. Kaleja, B. Andelić, K. Blaum, M. Block, P. Chhetri, C. Droese, C. E. Düllmann, M. Eibach, S. Eliseev, J. Even, S. Götz, F. Giacoppo, N. Kalantar-Nayestanaki, E. Minaya Ramirez, A. Mistry, T. Murböck, S. Raeder, and L. Schweikhard. The performance of the cryogenic buffer-gas stopping cell of SHIPTRAP. *Nuclear Instruments and Methods in Physics Research Section B*, page in press (2019). URL `http://dx.doi.org/https://doi.org/10.1016/j.nimb.2019.05.009`.

[164] F. Lautenschläger, P. Chhetri, D. Ackermann, H. Backe, M. Block, B. Cheal, A. Clark, C. Droese, R. Ferrer, F. Giacoppo, S. Götz, F.-P. Heßberger, O. Kaleja, J. Khuyagbaatar, P. Kunz, A. K. Mistry, M. Laatiaoui, W. Lauth, S. Raeder, T. Walther, and C. Wraith.

Developments for resonance ionization laser spectroscopy of the heaviest elements at SHIP. *Nuclear Instruments and Methods in Physics Research Section B*, **383**:115–122 (2016). URL `http://dx.doi.org/https://doi.org/10.1016/j.nimb.2016.06.001`.

[165] E. Eliav, U. Kaldor, and Y. Ishikawa. Open-shell relativistic coupled-cluster method with Dirac-Fock-Breit wave functions: Energies of the gold atom and its cation. *Phys. Rev. A*, **49**:1724–1729 (1994). URL `http://dx.doi.org/10.1103/PhysRevA.49.1724`.

[166] E. Eliav, U. Kaldor, and Y. Ishikawa. Ionization potentials and excitation energies of the alkali-metal atoms by the relativistic coupled-cluster method. *Phys. Rev. A*, **50**:1121–1128 (1994). URL `http://dx.doi.org/10.1103/PhysRevA.50.1121`.

[167] V. A. Dzuba, M. S. Safronova, and U. I. Safronova. Atomic properties of superheavy elements No, Lr, and Rf. *Phys. Rev. A*, **90**:012504 (2014). URL `http://dx.doi.org/10.1103/PhysRevA.90.012504`.

[168] X. CAO and M. DOLG. Theoretical prediction of the second to fourth actinide ionization potentials. *Molecular Physics*, **101**, 7:961–969 (2003). URL `http://dx.doi.org/10.1080/0026897021000046807`.

[169] S. Fraga. Electronic structure of the transactinides with 6d^N 7s^2, 7p^N, and 8s^N configurations. *Anales de Fisica*, **70**:249 (1974).

[170] J. Sucher. Foundations of the relativistic theory of many-electron atoms. *Phys. Rev. A*, **22**:348–362 (1980). URL `http://dx.doi.org/10.1103/PhysRevA.22.348`.

[171] E. Eliav, U. Kaldor, and Y. Ishikawa. Tel-Aviv Relativistic Atomic Fock Space coupled cluster code (TRAFS-3C). Available on request.

[172] E. Eliav, A. Borschevsky, and U. Kaldor. *High-Accuracy Relativistic Coupled-Cluster Calculations for the Heaviest Elements*, pages 819–849. Springer-Verlag Berlin Heidelberg (2017).

[173] E. Eliav, M. J. Vilkas, Y. Ishikawa, and U. Kaldor. Extrapolated intermediate Hamiltonian coupled-cluster approach: Theory and pilot application to electron affinities of alkali atoms. *J. Chem. Phys.*, **122**, 22:224113 (2005). URL `http://dx.doi.org/10.1063/1.1929727`.

[174] G. L. Malli, A. B. F. Da Silva, and Y. Ishikawa. Universal Gaussian basis set for accurate *ab initio* /P *relat ivistic* D *irac - Fock calculations. Phys. Rev. A*, **47**:143–146 (1993). URL `http://dx.doi.org/10.1103/PhysRevA.47.143`.

[175] V. Shabaev, I. Tupitsyn, and V. Yerokhin. QEDMOD: Fortran program for calculating the model Lamb-shift operator. *Computer Physics Communications*, **189**:175 – 181 (2015). ISSN 0010-4655. URL `http://dx.doi.org/https://doi.org/10.1016/j.cpc.2014.12.002`.

[176] W. R. Johnson. *Lectures on Atomic Physics*. Dept. of Physics, University of Notre Dame, South Bend, IA (1994).

[177] V. A. Dzuba, V. V. Flambaum, M. G. Kozlov, and S. G. Porsev. Using effective operators in calculating the hyperfine structure of atoms. *Journal of Experimental and Theoretical Physics*, **87**:885 (1998).

[178] S. G. Porsev, Y. G. Rakhlina, and M. G. Kozlov. Electric-dipole amplitudes, lifetimes, and polarizabilities of the low-lying levels of atomic ytterbium. *Phys. Rev. A*, **60**:2781–2785 (1999). URL `http://dx.doi.org/10.1103/PhysRevA.60.2781`.

[179] W. C. Martin, R. Zalubas, and L. Hagan. *Atomic energy levels - The rare-Earth elements* (1978).

[180] J. E. Sansonetti and W. C. Martin. Handbook of Basic Atomic Spectroscopic Data. *Journal of Physical and Chemical Reference Data*, **34**, 4:1559–2259 (2005). ISSN 0047-2689.

[181] T. Murböck, S. Raeder, P. Chhetri, K. Diaz, M. Laatiaoui, F. Giacoppo, and M. Block. Filament studies for laser spectroscopy on lawrencium. *Hyperfine Interactions*, **241**, 1:1–9 (2020).

[182] A. Borschevsky, E. Eliav, M. Vilkas, Y. Ishikawa, and U. Kaldor. Transition energies of atomic lawrencium. *The European Physical Journal D*, **45**, 1:115–119 (2007).

[183] S. Fritzsche, C. Dong, F. Koike, and A. Uvarov. The low-lying level structure of atomic lawrencium (Z= 103): energies and absorption rates. *The European Physical Journal D*, **45**, 1:107–113 (2007).

[184] Y. Zou and C. Froese Fischer. Resonance Transition Energies and Oscillator Strengths in Lutetium and Lawrencium. *Phys. Rev. Lett.*, **88**:183001 (2002). URL `http://dx.doi.org/10.1103/PhysRevLett.88.183001`.

[185] L. Visscher and K. G. Dyall. Fully relativistic *ab initio* calculations of the energies of chiral molecules including parity-violating weak interactions. *Atom. Data Nucl. Data Tabl.*, **67**:207 (1997). URL `http://dx.doi.org/10.1103/PhysRevA.60.4439`.

[186] P. Chhetri, D. Ackermann, H. Backe, M. Block, B. Cheal, C. E. Düllmann, J. Even, R. Ferrer, F. Giacoppo, S. Götz *et al.* Impact of buffer gas quenching on the $^1S_0 \rightarrow^1 P_1$ ground-state atomic transition in nobelium. *Europ. Phys. J. D*, **71**, 7:195 (2017). URL `http://dx.doi.org/10.1140/epjd/e2017-80122-x`.

[187] J. S. Vetter, R. Brightwell, M. Gokhale, P. McCormick, R. Ross, J. Shalf, K. Antypas, D. Donofrio, T. Humble, C. Schuman *et al.* Extreme heterogeneity 2018-productive computational science in the era of extreme heterogeneity: Report for DOE ASCR workshop on extreme heterogeneity. Technical report, USDOE Office of Science (SC), Washington, DC (United States) (2018).

[188] H. Nguyen. *Gpu Gems 3.* Addison-Wesley Professional (2007).

[189] R. Farber. *CUDA application design and development.* Elsevier (2011).

[190] C. W. Johnson, W. E. Ormand, K. S. McElvain, and H. Shan. BIGSTICK: A flexible configuration-interaction shell-model code. *arXiv preprint arXiv:1801.08432* (2018).

[191] H. Shan, S. Williams, C. Johnson, and K. McElvain. A locality-based threading algorithm for the configuration-interaction method. In *2017 IEEE International Parallel and Distributed Processing Symposium Workshops (IPDPSW)*, pages 1178–1187. IEEE (2017).

[192] J. Olsen, P. Jørgensen, and J. Simons. Passing the one-billion limit in full configuration-interaction (FCI) calculations. *Chemical Physics Letters*, **169**, 6:463–472 (1990).

[193] J. Jeffers, J. Reinders, and A. Sodani. *Intel Xeon Phi Processor High Performance Programming: Knights Landing Edition.* Morgan Kaufmann (2016).

[194] R. Rahman. *Intel Xeon Phi Coprocessor Architecture and Tools: The Guide for Application Developers.* Apress (2013).

[195] J. Mitroy, M. S. Safronova, and C. W. Clark. Theory and applications of atomic and ionic polarizabilities. *J. Phys. B*, **43**:202001 (2010). URL `http://dx.doi.org/10.1088/0953-4075/43/20/202001`.

[196] I. P. Grant and H. M. Quiney. Rayleigh-Ritz approximation of the Dirac operator in atomic and molecular physics. *Phys. Rev. A*, **62**:022508 (2000). URL `http://dx.doi.org/10.1103/PhysRevA.62.022508`.

[197] I. Tupitsyn and A. Loginov. Use of sturmian expansions in calculations of the hyperfine structure of atomic spectra. *Optics and Spectroscopy*, **94**, 3:319–326 (2003). ISSN 0030-400X.

[198] S. J. Grunefeld, M. W. J. Bromley, and Y. Cheng. The photon scattering cross-sections of atomic hydrogen. *arxiv*, page 1606.00939 (2016).

[199] J. Jiang, L. Jiang, X. Wang, D.-H. Zhang, L.-Y. Xie, and C.-Z. Dong. Magic wavelengths of the Ca$^+$ ion for circularly polarized light. *Physical Review A*, **96**, 4:042503 (2017).

[200] E. Cancès and C. Le Bris. Can we outperform the DIIS approach for electronic structure calculations? *International Journal of Quantum Chemistry*, **79**, 2:82–90 (2000).

[201] C. Le Bris. Computational chemistry from the perspective of numerical analysis. *Acta Numerica*, **14**:363 (2005).