Computing the Spectrum of the Confined Hydrogen Atom

Faculty of Electrical Engineering, Mathematics and Computer Science

Karl Kästner

Delft University of Technology

Challenge the future

Part I

Thesis Proper

Thesis submitted to the
Delft Institute of Applied Mathematics
at the Delft University of Technology,
for obtaining the degree

Master of Science in Applied Mathematics

Public defence on

Wednesday, 29th August 2012, 10:00am

by

Karl Kästner
Dipl.-Ing. Informationstechnik
Berufsakademie Stuttgart (Germany)

born

1984 in Halle/Saale (Germany)

Thesis Committee

Kees Vuik
Chairman, Full Professor at TU Delft

Martin van Gijzen
Daily supervisor, Associate professor at TU Delft

Wim (Wilhelm) Teunis van Horssen
Associate professor at TU Delft

Domenico Giordano
Scientist at ESA/ESTEC

Faculty of Electrical Engineering,
Mathematics and Computer Science
Mekelweg 4
2628 CD Delft
The Netherlands

www.tudelft.nl

Acknowledgements

First of all I want to thank Prof. van Gijzen for the time and work he invested in the supervision of this thesis, as well as for the many interesting hints and suggestions concerning the topic. Furthermore I want to thank everyone involved in the realisation of the Erasmus Mundus Programme of Computer Simulation for Science and Engineering (COSSE), in particular those at the Kungliga Tekniska Högskolan Stockholm and at the Technische Universiteit Delft, which are the universities I attended during the course of the last two years.

Abstract

This thesis is about the numerical approximation of the spectrum of the confined hydrogen atom. The energy levels and wave functions of the hydrogen atom are modelled by the corresponding eigenvalues and eigenvectors of the Schrödinger equation with Coulomb potential. At first different discretisation techniques for the hydrogenic Schrödinger equation are analysed. These include the finite difference method and the finite element method. Then appropriate methods for the solution of the discretised eigensystem are discussed. These include the Lanczos method, the implicitly restarted Arnoldi method and the Jacobi-Davidson method. A programme for the automatic computation of eigenspectra is implemented and numerical experiments are conducted to exemplify the theory.

The programme is further on used to compute the spectrum of the hydrogen atom and several aspects of confinement are systematically investigated. These include the size and shape of the cavity as well as the position of the nucleus therein. It is shown that with increasing confinement all energy levels diverge towards positive infinity and that higher energy levels are affected first with decreasing domain size. Degeneracy is partially retained under confinement if the nucleus is positioned in the centre of the domain, but the spectrum becomes non-degenerate if the nucleus is shifted.

Table of Contents

Chapter 1

Introduction

Hydrogen is the most abundant element in the universe. Moreover most other elements eventually originate from hydrogen by means of nuclear fusion[1]. Many physical effects can therefore be described by hydrogenic atom models. These models find important applications in solid state physics and statistical thermodynamics.

Each element has a distinct set of discrete energy levels. An electron inside an atom can only reside in one of the discrete levels at any time. If an electron drops down from a higher to a lower level a single photon is instantaneously emitted. The wave length of the emitted photon corresponds to the difference in potential energy between the higher and the lower level. The emission lines of hydrogen are partially visible and appear if one uses a prism to break the light from a hydrogen gas discharge lamp. As the energy levels form a discrete set, so does the set of emission lines. This set is called the *spectrum* of the atom.

To each energy level belongs a distinct wave function which describes the propagation of the electron in the vicinity of the nucleus. The energy levels and wave functions determine together the chemical properties of an element. These properties change under pressure and under influence of external electromagnetic fields. Such conditions can be modelled with shifted confinement.

The eigenvalues and energy levels of the hydrogenic Schrödinger equation can be found analytically, if one considers the unconfined atom. If the atom is confined into a cavity and shifted to an arbitrary position therein, then the exact solution cannot be found anymore. However, different approximation techniques can be applied in such a case. This thesis is about the numerical approximation of the eigenvalues and eigenfunctions of the hydrogenic Schrödinger equation. Numerical methods are favourable as they allow an arbitrary accuracy and an application to a variety of problems.

This Master thesis was commissioned by the European Space Agency [Age12]. Which will use the results of this thesis in a model describing vehicle reentry into planetary atmospheres. The thesis was made possible by a scholarship received as part of the Erasmus Mundus Programme Computer Simulations for Science and Engineering (COSSE) [DSBE12] which is funded by the European Commission of the European Union [Com12].

[1] The elements until Fe^{56} are generated by nuclear fusion from lighter elements. Heavier elements are generate by catching of neutrons.

Chapter 2

Physical Background

Throughout this thesis the hydrogen atom is modelled by the Schrödinger equation with a Coulomb potential [Sch26]. The time independent and dimensionless form of this partial differential equation is,

$$\begin{aligned} -\tfrac{1}{2}\Delta u(\vec{x}) + \tfrac{1}{r}u(\vec{x}) &= \lambda u(\vec{x}), \qquad (2.1)\\ \vec{x} &\in \Omega. \end{aligned}$$

Here Δ is the Laplacian operator $\Delta = \sum_{i=1}^{d} \frac{\partial^2}{\partial x_i^2}$ and r the radius $r = \sqrt{\vec{x}^T\vec{x}}$. The eigenfunctions u and the eigenvalues λ are the quantities of interest which are to be computed. To confine the atom in the d dimensional domain Ω, homogeneous Dirichlet conditions are applied at the boundary Γ,

$$\begin{aligned} u(\vec{x}) &= 0, \qquad (2.2)\\ \vec{x} &\in \Gamma. \qquad (2.3) \end{aligned}$$

In this thesis the two and three dimensional cases are investigated.

Note that there exist infinitely many solutions to the eigenvalue problem (2.1). However, only solutions corresponding to the reasonably small eigenvalues of the discrete spectrum are of physical interest.

The physical values can be recovered from the computed quantities by the relation,

$$\begin{aligned} \phi_k(a_0 x, a_0 y) &= u(x, y) \qquad (2.4)\\ E_k &= E_H \lambda_k \qquad (2.5) \end{aligned}$$

Where E_k represents the energy level and ϕ_k the corresponding wave function. The two constants are the Bohr radius $a_0 = 52.918pm$ and the Hartree energy $E_H = 27.211eV$.

A detailed derivation from the time dependent Schrödinger equation with variables expressed in physical dimensions is given in the literature study [Kä12, p. L 10].

2.1 The Unconfined Hydrogen Atom

In three dimensions the eigenvalues of the hydrogen atom are given by the series,

$$\lambda_n = \frac{1}{2}\frac{1}{n^2}, n = 1, 2, 3, \ldots. \qquad (2.6)$$

The eigenvalues are n^2 degenerate in three dimensions, e.g. the eigenvalue first state exists once, the second four times, the third nine times and so on. Further on in this thesis the eigenvalues λ_k will be indexed by the letter k, including the multiplicity of the respective value.

The ground state is given by the function,

$$u_1(r) = \frac{1}{\pi}\exp(-r). \qquad (2.7)$$

The wave function of the ground state is radial symmetric. Higher wave functions are the product of polynomials with an decaying exponential.

The eigenfunctions u_k can be scaled arbitrarily and still satisfies the differential equation for the same eigenvalue, as it is the case for eigenfunctions of all eigenproblems. As the square of the wave function corresponds to the probability of the electron to be detected at a certain location, one normalises the usually the square of the wave function to one,

$$1 = \int_\Omega u_k^2 d\vec{x}. \tag{2.8}$$

Note that the Hamiltonian of the hydrogen wave function is symmetric and therefore the eigenfunctions form a set of orthogonal functions, e.g.

$$0 = \int_\Omega u_j u_k d\vec{x}, j \neq k \tag{2.9}$$

It is important to realise that the wave functions are bounded and continuous although the coefficient of the Coulomb potential is singular. However, this is not the case for higher derivatives of the eigenfunctions, see chapter 3.3.

For the unconfined case the eigenpairs can be found analytically by writing the equation in spherical coordinates and applying the technique of the separation of variables [Kä12, p. L 13]. However, if the atom is confined and the nucleus shifted from the centre, the variables cannot be separated any more. For such cases only approximations to the exact solution can be found. In this thesis numerically approximations are considered. This allows a maximum of flexibility and arbitrary accuracy.

When the domain of the hydrogen wave functions are restricted to a plane it becomes quasi two dimensional. Models with reduced dimensionality find application in solid state physics [LK55, Bas81]. An analytic solution of the unconfined two dimensional hydrogen atom is found similar to the three dimensional case by writing equation (2.1) in polar coordinates and separating the variables. The eigenvalues are given by the series,

$$\lambda_n = \frac{2}{(2n-1)^2}, n = 1, 2, 3, \dots \ . \tag{2.10}$$

The degeneracy follows the series of the odd integers, e.g. the first distinct eigenvalue is found once in the spectrum, the second three times, the third eigenvalue five times and so forth.

The wave function of the ground state is,

$$u_1(r) = \frac{1}{2\pi} \exp\left(-\frac{r}{2}\right). \tag{2.11}$$

Many papers have been written about the hydrogen atom in one and two dimensions. Solution formulae for arbitrarily many dimensions are found in [All57] and for three dimensions beyond other in [YGC$^+$91]. Note that there exists no straight forward extension to one dimension, as the Coulomb potential effectively separates the domain in two invariant subspaces. The one dimensional case is therefore not considered in this thesis, except for the radial Schrödinger equation where the Coulomb potential resides on one end of the domain.

2.2 Validity of the Hydrogenic Schrödinger Equation

The hydrogenic Schrödinger equation is not an exact model of the hydrogen atom. It neglects higher order physical effects, most prominently the influence of relativity. The eigenvalues of the hydrogenic Schrödinger equation have approximately only three physically significant digits. An overview of those effects is given in [vD11]. The higher order physical effects can be incorporated into the model, but this usually yields a more complex differential equation which is more difficult to solve. An alternative to the Schrödinger equation is the Dirac equation, which incorporates relativistic effects, see [Dir28, GYC$^+$91]. This poses the important question up to which accuracy the hydrogenic Schrödinger equation remains valid in case of confinement. However, answering this question is not part of the current thesis.

Chapter 3

Discretisation

Two different discretisation techniques for the hydrogenic Schrödinger equations are discussed here. These are the finite difference method and the finite element method. It is shown that both methods yield converging solutions. However, the rate of convergence is at most linear with respect to the step size for both methods at the origin of the Coulomb potential, even if the higher order discretisations are applied. It is furthermore shown that an arbitrarily high rate of convergence with respect to the number of mesh points can be achieved by locally refining the mesh at the origin of the Coulomb potential.

Note that there are further methods to discretise the Schrödinger equation. Those include the sparse grid technique [Gar98, GG00, GH06, GH07, Yse05] and the Lagrangian mesh method [BH86, BS08, Bay06]. Both methods seem to prove themselves superior to the finite element method under certain circumstances.

3.1 Finite Difference Discretisation

Consider the Schrödinger equation in Cartesian coordinates in d dimensions where the domain is a hypercube with side length L and the Coulomb potential located at the origin $\vec{x} = 0$. Its basic finite difference discretisation on a homogeneous grid with n points in one dimension is, cf. [Kä12, p. L 19],

$$\frac{1}{h^2}[1, -2, 1] \begin{bmatrix} u(x_{i-1}) \\ u(x_i) \\ u(x_{i+1}) \end{bmatrix} + \frac{1}{x_i} = \lambda u(x_i), \; i = 1 \dots n, \tag{3.1}$$

where h is the step size, $h = x_i - x_{i-1} = \frac{1}{n+1}$. In case of homogeneous Dirichlet boundary conditions the value at the boundary are,

$$\begin{aligned} u(-\tfrac{1}{2}L) &= 0, \; i = 0 \\ u(+\tfrac{1}{2}L) &= 0, \; i = n + 1. \end{aligned} \tag{3.2}$$

Finite difference discretisations in higher dimensions can be obtained from the one dimensional discretisation via the Kronecker product [Kä12, p. L 26]. In d dimensions the approximate solution u is then computed on n^d interior grid points. In two dimensions the basic finite difference discretisation has a five point kernel, in three dimensions it has a seven point kernel.

The eigenvalues of the unconfined hydrogen atom are approximated in the limit case $L \to \infty$. With a finite value of L the eigenvalues of the confined hydrogen atom are approximated.

Note that the finite difference discretisation is invalid in one dimension, as the Coulomb potential effectively separates the domain in two halves. However, the radial Schrödinger equation is effectively a one dimensional equation with the kernel (3.1) and the different boundary conditions,

$$u(0) = 0 \tag{3.3}$$

$$u(r) = 0$$

$$r = \tfrac{1}{2}L. \tag{3.4}$$

The radial Schrödinger equation is found by separating the Schrödinger equation in three dimensional spherical coordinates.

A detailed discussion of higher order accurate discretisations and grids with varying grid spaces is contained in the literature study [Kä12, p. L 18].

3.1.1 Discretisation Error of Finite Difference Approximations for Eigensystems

The discretisation error $u_*(x) - u_h(x)$ of the finite difference approximation u_h of the exact solution u_* can be expressed in form of higher order derivatives. The discretisation error at the point x_i in one dimension is, c.f. [Kä12, p. L 26],

$$\begin{aligned} u_*(x) - u_h(x_i) &= \frac{1}{3}(h_r - h_l)u'''(x_i) + \frac{1}{12}(h_l^2 - h_l h_r + h_r^2)u^{iv}(x_i) + O(h^3), \\ h_l &= x_i - x_{i-1}, \\ h_r &= x_{i+1} - x_i, \end{aligned} \tag{3.5}$$

where u_* is the exact solution and u_h the approximate solution obtained by solving the discrete eigensystem. For a uniform grid with constant step size h this simplifies to

$$u_*(x) - u_h(x) = \frac{1}{12}h^2 u^{vi}(x) + O(h^4). \tag{3.6}$$

In higher dimensions the errors combine with the Kronecker product as the difference quotients do.

This is an error estimate for approximations to boundary value problems. To estimate the error of approximated eigenvalues, consider at first the discretised Schrödinger equation (3.1) without potential in the unit interval $[-\frac{1}{2}, \frac{1}{2}]$. This is the discrete Laplacian operator in one dimension. The exact eigenvalues λ_k^* are found as the series,

$$\lambda_k^* = -\pi^2 k^2. \tag{3.7}$$

Conversely, the eigenvalues λ_k^h of the Laplacian discretised with a three point finite difference scheme are given by the series [Kä12, p. L 69]

$$\lambda_k^h = \frac{-4}{h^2}\sin^2\left(\frac{\pi k h}{2}\right). \tag{3.8}$$

Where $h = \frac{1}{n+1}$ is the step-width on a uniform grid with n grid points.

The discretisation error $\lambda_k^h - \lambda_k^*$ can be estimated by comparing the discrete with the analytic eigenvalues.

$$\lambda_k^h - \lambda_k^* = -\pi^2 k^2 + \frac{4}{h^2}\sin^2\left(\frac{\pi k h}{2}\right) \tag{3.9}$$

The sine is expanded as a Taylor series,

$$\begin{aligned} \lambda_k^h - \lambda_k^* &= -\pi^2 k^2 + \frac{4}{h^2}\left(\frac{\pi k h}{2} - \frac{\pi^3 k^3 h^3}{6 \cdot 8} + O(k^5 h^5)\right)^2 \\ &= -\pi^2 k^2 + \frac{4}{h^2}\left(\frac{\pi^2 k^2 h^2}{4} - \frac{\pi^4 k^4 h^4}{6 \cdot 8} + O(k^6 h^6)\right) \\ &= \frac{-\pi^4 k^4 h^2}{12} + O(k^6 h^4). \end{aligned} \tag{3.10}$$

The relative error is subsequently found as

$$\frac{\lambda_k^h - \lambda_k^*}{\lambda_k^*} = \frac{\pi^2 k^2 h^2}{12} + O(k^4 h^4). \tag{3.11}$$

Therefore the rate of convergence of the first eigenvalue equals the rate of convergence experienced in approximating the solution of linear systems, cf. (3.6). For higher eigenvalues, the relative accuracy deteriorates quadratically with increasing k. Thought, one could correct this for cases similar to the Laplacian operator where an approximation of the discretisation error itself is known [PHA81].

The eigenvectors of the discrete Laplacian match exactly the eigenfunctions of the infinite dimensional Laplacian. However, this is not the case for general functions. For eigenvectors of the Poisson type, the error estimate according to equation (3.6) is

$$\begin{aligned} u_k^*(x) - u_k^h(x) &= \frac{1}{12}h^2 u_k^{iv}(x) + O(h^4) \\ &= \frac{1}{12}h^2 (\sin(\pi k x))^{iv} + O(h^4) \\ &= O(h^2 k^4). \end{aligned} \tag{3.12}$$

The error estimate of the eigenvectors has therefore the same asymptotic behaviour as that of the eigenvalues.

This analysis shows that the discretisation error estimate obtained for functions (3.6) carries over to the error estimate of eigenvalues (3.11) and eigenfunctions (3.12) of the discrete Laplacian with respect to the influence of the step size and higher order derivatives. The accuracy of eigenpairs deteriorates furthermore with increasing k, e.g. for higher eigenvalues. However, this is of no concern to the approximation of energy levels of the hydrogen atom, as one is only interested in a few of the lowest eigenvalues.

Note that the discretisation error does not only depend on the step size, but also on higher derivatives, e.g. the smoothness of the approximated function. This is an essential problem, as the eigenfunctions of the unconfined hydrogenic Schrödinger equation are not smooth, cf. equations (2.7) and (2.11). The consequences are discussed in chapter 3.3.

3.1.2 Grid Adaption for Finite Difference Methods

Further on in section 3.3.3 it is shown that the convergence of the finite difference and finite element approximations for the eigenpairs of the hydrogenic Schrödinger equations is significantly improved by local grid refinement.

This subsection explores the practical aspects of finite difference approximations on adaptive grids. In general it is shown that finite difference approximations on adaptive grids suffer from serious drawbacks. Those drawbacks do in general not apply to finite element discretisations, as laid out in the second half of this chapter.

In principle it is possible to transform the coordinate system with some function $f(x)$ and than to use the regular finite difference discretisation on this grid,

$$\begin{aligned} (u(f(x)))'' + c(x)u(f(x)) &= \lambda u(f(x)) \\ f''(x)u'(f(x)) + (f'(x))^2 u''(f(x)) + c(x)u(f(x)) &= \lambda u(f(x)). \end{aligned} \tag{3.13}$$

This is how a differential equation in polar, cylindrical and spherical coordinates is derived. Usually symmetry is lost for other choices for $f(x)$. As this thesis focuses on discretisations in Cartesian coordinates, this approach is not considered further on.

Different degrees of adaptation can be considered. It is possible to cluster grid points while retaining the structure of the discretisation matrix by using a grid as depicted in figure 3.1.2. This leads to the discretisation scheme, c.f. [Kä12, p. L 26],

$$u''(x_i) = \left[\frac{2}{h_l h_{lr}}, \frac{-2}{h_l h_r}, \frac{2}{h_r h_{lr}}\right] \begin{bmatrix} u(x_{i-1}) \\ u(x_i) \\ u_(x_{i+1}) \end{bmatrix} + O(h), \tag{3.14}$$

with $h_l = x_i - x_{i-1}$, $h_r = x_{i+1} - x_i$ and $h_{lr} = x_{i+1} - x_{i-1}$. This scheme is asymmetric, but symmetry can be retained by a similarity transform, cf. [Kä12, p. L 43]. As the grid is still structured it is still possible to set it up by Kronecker products. The fundamental drawback of this scheme is the loss of accuracy, as this scheme is only first order accurate according to the estimate (3.5). Furthermore the discretisation matrix becomes ill conditioned if the step width is varied over several magnitudes. The grid also has many

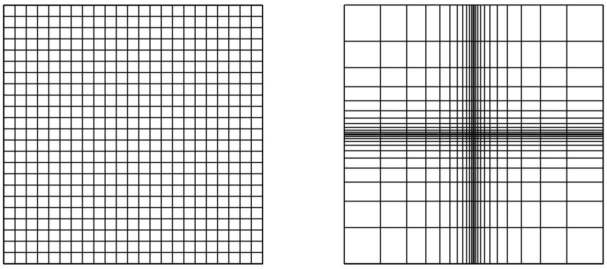


Figure 3.1: A homogeneous grid and a grid with varying step width for the finite difference method

more grid than compared to an unstructured grid. This schemes can be extended to higher accuracy by including more points [Kä12, p. L 63], but in this case the symmetry of the discretisation matrix cannot be recovered and cancellation errors become severe. Note that even on uniform grids higher order finite difference discretisation matrices are asymmetric if higher order derivatives of the solution are nonzero at the boundary, which is the case for the confined hydrogen atom.

An alternative discretisation scheme that retains the full rate of convergence is found by changing the step size only by factors of two and applying the kernel

$$\begin{bmatrix} u''(x_i-2h) \\ u''(x_i) \\ u''(x_i+h) \end{bmatrix} = \frac{1}{h^2} \begin{bmatrix} 4 & & \\ & 4 & \\ & & 1 \end{bmatrix} \begin{bmatrix} 1 & -2 & 1 & 0 & 0 \\ 0 & 1 & -2 & 0 & 1 \\ 0 & 0 & 1 & -2 & 1 \end{bmatrix} \begin{bmatrix} u(x_i-4h) \\ u(x_i-2h) \\ u_{(}x_i) \\ u_{(}x_i+h) \\ u(x_i+2h) \end{bmatrix} + O(h^2). \tag{3.15}$$

However, this discretisation matrix is not symmetric and also does not resolve the cancellation errors if the step size is varied over several magnitudes.

Note that finite difference discretisations on unstructured grids [Dur02, ST03, GA06]. However, those suffer from similar restriction with respect to the loss of symmetry and reduced speed of convergence. Furthermore the discretisation matrices cannot be set up with the Kronecker product any more in those cases.

3.2 Finite Element Discretisation

This section discusses the application of the finite element method to the hydrogenic Schrödinger equation. Many papers have been published about this topic. However, most publications contain only solutions for non-singular potentials or the radial equation. Few papers actually cover the finite element solution to the hydrogenic Schrödinger equation [AR93, FRRT78] (2D), [AER94, MIKG12] (3D).

The finite element method finds an approximate value λ_h and the corresponding function u_h in the inner product space V_h which is a subspace of the solution space V ($V_h \subset V$). The approximate solution u_h satisfies the variational or weak form of the eigenvalue equation,

$$\int_\Omega \Delta u_h v_h + \tfrac{1}{x} u_h v d\vec{x} = \lambda \int_\Omega u_h v_h d\vec{x}, \forall v \in V_h. \tag{3.16}$$

Here a is some differential operator. The approximate solution is the projection of the true solution onto the space V_h. Henceforth the subscript h is dropped for better readability.

The weak form admits infinitely many solutions, as the original eigenvalue equation does. Of particular interest is the solution,

$$\lambda = \min \frac{\int_\Omega \Delta u_h v_h + \frac{1}{x} u_h v d\vec{x}}{\int_\Omega u v d\vec{x}}. \tag{3.17}$$

The weak formulation of the hydrogenic Schrödinger equation (2.1) is found by multiplying it with the test function v and integrating once, cf. [Kä12, p. L 30],

$$-\frac{1}{2}\int_{\Omega} \Delta u \cdot v + \frac{1}{r} u \cdot v \, d\vec{x} = \lambda \int_{\Omega} u \cdot v \, d\vec{x}. \tag{3.18}$$

By the divergence theorem this becomes

$$-\frac{1}{2}\int_{\Omega} (\nabla u) \cdot (\nabla v) \, d\vec{x} + \underbrace{\frac{1}{2}\int_{\Gamma} \frac{\partial u}{\partial n} \cdot v \, d\vec{x}}_{=0} + \int_{\Omega} \frac{1}{r} u \cdot v \, d\vec{x} = \lambda \int_{\Omega} u \cdot v \, d\vec{x}, \tag{3.19}$$

where the boundary term drops out due to the homogeneous Dirichlet boundary conditions $v(x) = 0, x \in \Gamma$.

In Galerkin's method the solution is estimated by piecewise polynomial basis ϕ_i functions which are only non-zero in a subset of the domain. The solution $u(x)$ is approximated by the interpolation of the explicitly computed function values $u_i = u(x_i)$. The interpolation is obtained by multiplying with the trial functions $\phi_i(x)$,

$$u(x) = \sum_{i=1}^{m} u_i \phi_i(x), i = 1 \dots n. \tag{3.20}$$

The basis functions are substituted into equation (3.19) and the integrals are computed. The integrals are usually approximated by numerical quadrature [Kä12, p. L 33].

By substituting the approximate solution $u(x)$ into the weak formulation one attains a system of equations,

$$\begin{aligned} -\frac{1}{2}\sum_{i=1}^{n} \int_{\Omega} u_i \nabla\phi_i(x) \nabla v_j(x) dx + \sum_{i=1}^{n} \frac{1}{2} \, u_i \nabla\phi_i(x) v_j(x)\Big|_{\Gamma} \\ + \sum_{i=1}^{n} \int_{\Omega} \frac{1}{x} u_i \phi_i(x) v_j(x) dx = \lambda \sum_{i=1}^{n} \int_{\Omega} u_i \phi_i(x) v_j(x) dx. \end{aligned} \tag{3.21}$$

Note that the derivatives of the unknown function are not contained anymore in this approximation. Instead the test and trial functions are differentiated. This is a simple procedure, if one chooses polynomials as test functions. Note that one usually chooses the trial and test functions to be equal.

This conveys the differential equation over into a generalised eigensystem of the form

$$Ax = \lambda Bx, \tag{3.22}$$

where A is called the stiffness matrix and B the mass matrix. The choice of only locally non-zero test functions makes this system sparse and the choice to use the same polynomials as test and as trial functions makes the system symmetric. This allows the application of specialised eigensolvers, see chapter 4. The quadrature schemes are chosen such that the matrix B is positive definite. The matrix A is indefinite in case of the hydrogenic Schrödinger equation.

The hydrogenic Schrödinger equation discretised with the finite element method in combination with linear polynomials on in one dimension a uniform is,

$$\frac{1}{h}[1, -2, 1] \begin{bmatrix} u(x_i - h) \\ u(x_i) \\ u(x_{i+1}) \end{bmatrix} + \frac{h}{x_i} = \lambda \frac{h}{6}[1, 4, 1] \begin{bmatrix} u(x_i - h) \\ u(x_i) \\ u(x_{i+1}) \end{bmatrix}, \; i = 1, \dots, n, \tag{3.23}$$

where the potential discretised with the trapezoidal rule. The boundary conditions are

$$\begin{aligned} u(-\tfrac{1}{2}L) = 0, \; i = 0 \\ u(+\tfrac{1}{2}L) = 0, \; i = n + 1. \end{aligned} \tag{3.24}$$

Discretisation matrices for higher dimensions cannot be simply set up by Kronecker products of one dimensional matrices.

3.2.1 Discretisation Error of Finite Element Approximations for Eigensystems

The discretisation error of the finite element method corresponds to the interpolation error $u_*(x)-\pi(u_h(x))$. As before u_* denotes the exact solution and $\pi(u_h(x))$ the interpolation of the discrete approximation to any point inside the domain. As with the finite difference method the discretisation error is related to higher derivatives. If the element integrals are solved exactly, then the estimate for piecewise linear basis functions is, c.f. [Kä12, p. L 32],

$$||u_*(x) - \pi(u_h(x))|| \leq Ch^2|u''(x)| + O(h^3). \tag{3.25}$$

Where the constant C does not depend on the step size h.

For higher polynomials of order p and under certain conditions the discretisation error is, c.f. [SF88, Theorem 3.7],

$$||u_*(x) - \pi(u_h(x))|| \leq Ch^{2p}|u^{(p+1)}(x)|. \tag{3.26}$$

Where $u^{(p+1)}$ denotes the derivative of order $p+1$. If the conditions for this estimate are not met, the error is bounded at least by,

$$||u_*(x) - \pi(u_h(x))|| \leq Ch^{p+1}|u^{(p+1)}(x)|, \tag{3.27}$$

by assuming that the mesh is sufficiently adapted and not degenerated.

Strang and Fix give the condition $p \leq 1$ for equation (3.26) and $p \geq 1$ for equation (3.27) to apply for the solution of boundary value problems. However the numerical experiments 5.2 show that the approximate eigenvalues at least for the discrete Laplacian decay also for higher order approximations with $p \geq 1$ with the faster rate $2p$.

One realises that the major difference in the error estimate between finite difference and finite element approximation is the order of the derivatives which contribute to the error. On a uniform grid the order of the derivative is two degrees lower for the finite element method.

The error estimates given above are again valid for the approximation of boundary value problems. To estimate the discretisation error of approximated eigenvalues consider again the Laplacian operator in one dimension. If the Laplacian operator is discretised with the finite element method and linear polynomials on a constant grid, then the approximated eigenvalues λ_k^h are found as, cf. [Kä12, p. L 69],

$$\lambda_k^h = \frac{-4}{h^2}(\sin^2(\tfrac{1}{2}\pi kh))\frac{3}{2+\cos(\pi kh)}. \tag{3.28}$$

This is the same as (3.8), but with an additional denominator is due to the mass matrix.

Expansion in the form of a Taylor series yields (c.f. [SF88, p.225]), neglecting terms larger than h^4,

$$\begin{aligned}\lambda_k^h &= \frac{6}{h^2}\frac{1-(1-\frac{1}{2}(\pi kh)^2+\frac{1}{24}(\pi kh)^2)}{2+1-\frac{1}{2}(\pi kh)^2+\frac{1}{24}(\pi kh)^4}\\ &= \frac{2}{h^2}\frac{\frac{1}{2}(\pi kh)^2+\frac{1}{24}(\pi kh)^2}{1-\frac{1}{6}(\pi kh)^2+\frac{1}{72}(\pi kh)^4}\\ &= \frac{2}{h^2}(\tfrac{1}{2}(\pi kh)^2+\tfrac{1}{24}(\pi kh)^2)(1+\tfrac{1}{6}(\pi kh)^2+\tfrac{1}{72}(\pi kh)^4)\\ &= \frac{2}{h^2}(-\tfrac{1}{2}(\pi kh)^2-\tfrac{1}{24}(\pi kh)^4)\\ &= -\pi^2k^2-\tfrac{1}{12}\pi^4k^4h^2.\end{aligned} \tag{3.29}$$

Therefore the error estimate of the eigenvalues computed with the finite element method matches up to higher order terms the error estimate of the eigenvalues computed with the finite difference method.

The eigenvectors of the discrete Laplacian computed with the finite element method in one dimension are exact, as with the finite difference method. This does not hold in higher dimensions and also does not apply to general functions. The error estimate for eigenvectors of the Poisson type is,

$$\begin{aligned} ||u_k^*(x) - u_k^h(x)||_2 &= h^2 u_k''(x) + O(h^3) \\ &= h^2 \sin(\pi k x)'' + O(h^3) \\ &= h^2 k^2 + O(h^2 k^2) \end{aligned} \tag{3.30}$$

One realises that the approximated eigenvectors corresponding to larger eigenvalues are more accurate than the eigenvectors approximated by the finite difference method.

More on eigenvalue error estimation for finite element methods can be found in [SF88, ch.6].

3.3 Convergence at the Origin of the Coulomb Potential

Consider the error estimates for the finite difference (3.5) and finite element (3.25) methods. To apply those estimates to the wave function of the hydrogen ground state, one has to find its derivatives first. The ground state of the unconfined hydrogen (2.11) atom and its first partial derivatives with respect to x in two dimensions are (the coordinate axes and function are scaled for convenience),

$$u(x,y) = \exp(-\sqrt{x^2+y^2}) \tag{3.31}$$

$$\frac{\partial}{\partial x}u(x,y) = \left(\frac{-x}{(x^2+y^2)^{\frac{1}{2}}}\right)\exp(-\sqrt{x^2+y^2}) \tag{3.32}$$

$$\frac{\partial^2}{\partial x^2}u(x,y) = \left(\frac{x^2}{(x^2+y^2)} - \frac{y^2}{(x^2+y^2)^{\frac{3}{2}}}\right)\exp(-\sqrt{x^2+y^2}) \tag{3.33}$$

$$\frac{\partial^3}{\partial x^3}u(x,y) = \left(\frac{3xy^2 - x^5 - x^3y^2}{(x^2+y^2)^{\frac{5}{2}}} + \frac{3xy^2}{(x^2+y^2)^2}\right)\exp(-\sqrt{x^2+y^2}) \tag{3.34}$$

$$\frac{\partial^4}{\partial x^4}u(x,y) = \left(\frac{-3y^2(2x^4+2x^2y^2+4x^2-y^2)}{(x^2+y^2)^{\frac{7}{2}}} + \frac{x^6+x^4y^2-12x^2y^2+3y^4}{(x^2+y^2)^3}\right)\exp(-\sqrt{x^2+y^2}). \tag{3.35}$$

At the origin the ground state reaches its maximum and the derivatives become discontinuous.

In the vicinity of the singularity this becomes, where r is some norm of $\vec{x}$,

$$\left|\lim_{r\to 0} u(x,y)\right| = 1 \tag{3.36}$$

$$\left|\lim_{r\to 0} \frac{\partial u(x,y)}{\partial x}\right| = \lim_{r\to 0} \frac{1}{\sqrt{2}} \tag{3.37}$$

$$\left|\lim_{r\to 0} \frac{\partial^2 u(x,y)}{\partial x^2}\right| = \lim_{r\to 0} \frac{1}{2\sqrt{2}r} \tag{3.38}$$

$$\left|\lim_{r\to 0} \frac{\partial^3 u(x,y)}{\partial x^3}\right| = \lim_{r\to 0} \frac{3}{4\sqrt{2}r^2} \tag{3.39}$$

$$\left|\lim_{r\to 0} \frac{\partial^4 u(x,y)}{\partial x^4}\right| = \lim_{r\to 0} \frac{9}{8\sqrt{2}r^3}. \tag{3.40}$$

Apparently a derivative of order d diverges at the origin proportional to $\frac{1}{r^{d-1}}$. Assume now, that the grid is set up, such that the origin is located exactly at the mid-point of a finite difference interval and that in the discrete case $r_{min} = \frac{1}{2}h$.

According to the estimate (3.5) the contributions of higher derivatives to the discretisation error of the finite difference method are attenuated like $O(h^{d-2})$. The estimate therefore diverges at the origin of the Coulomb potential with the rate $O\left(h^{d-2}\left(\frac{1}{h^{d-1}}\right)\right) = O(\frac{1}{h})$, and one could prematurely conclude that the finite difference method is not converging. However, in the section 3.3.2 a refined estimate is derived, effectively showing that the finite difference method indeed converges.

According to equation (3.27) the finite element method with linear basis functions attenuates derivatives of order d at least proportional to h^d. Therefore the finite element approximation converges at the origin proportional to $\frac{h^d}{h^{d-1}} = h$. Note the rate of convergence is reduced compared to the regions where the function is sufficiently smooth. The rate of convergence of the finite element method with Lagrangian basis functions remains only linear at the origin of the Coulomb singularity, even for higher order polynomials. Therefore it is essential to apply local refinement close to the origin.

This analysis is valid for the hydrogen ground state of the unconfined atom. As wave functions of higher energy levels decay slower, their non-smoothness at the origin of the Coulomb potential is less pronounced. The ground state is therefore the wave function which is most difficult to compute and where convergence is slower than for all other wave functions. Therefore the above analysis show the dominant effect on convergence.

Note furthermore that the ground state is also the last state to become significantly effected by confinement, as the maximum of the corresponding radial probability is closest to the Coulomb potential compared to all other wave functions. Therefore the above conclusion is not only valid for the unconfined hydrogen ground state alone, but over a wide range of confinement as well.

3.3.1 Convergence Estimation via Fourier Analysis

In the section above the point wise divergence of higher derivatives of the hydrogen wave function was derived by differentiation in real space. This section complements this analysis by an analogue investigation in the frequency domain.

Define the Fourier transform $\hat{u}(t)$ of a square integrable function $u(x)$ as,

$$\hat{u}(t) = \int_{-\infty}^{\infty} \exp(i2\pi t)u(x)dx,\ i = \sqrt{-1}. \tag{3.41}$$

By applying the Fourier transform to the derivatives $\frac{d^d}{dx^d}f(x)$ of the function $f(x)$ one obtains the corresponding spectral derivatives $\hat{u}_d(t)$,

$$\hat{u}_d(t) = (2\pi i t)^d \hat{u}(t). \tag{3.42}$$

By Plancherel's theorem it is known that the L_2-norm of a function and its Fourier transform are equal,

$$||u||_2 = ||\hat{u}||_2. \tag{3.43}$$

Therefore the Fourier coefficients of a diverging function must diverge as well and vice versa. Furthermore the rate of divergence should be equal.

Consider now the ground state wave function of the unconfined hydrogen atom. It is square integrable and therefore its Fourier coefficients exist and decay to zero. The exact discrete Fourier coefficients of the hydrogen ground state are [Kä12, p. L 23],

$$\hat{u}(t) = \frac{1}{\pi n}\frac{1}{\sqrt{\pi a_0^3}}\frac{2}{4\pi^2 t^2 + 1} \tag{3.44}$$

One realises that the Fourier coefficients of the d-th derivative of the hydrogen ground state decay like t^{d-2} for $\lim_{t\to\infty}$, hence only the ground state itself and the first derivative remain bounded. Therefore all higher derivatives diverge with rate $\frac{1}{h^{d-1}}$ as their Fourier coefficients do not decay to zero.

As a matter of fact the conclusion that the Fourier transforms of the derivatives of the hydrogen ground state diverge with rate t^{d-2} for $\lim_{t\to\infty}$ coincides with the findings of the last section that the derivatives of order d diverge in real space with the rate $\frac{1}{r^{d-2}}$ for $\lim_{r\to 0}$.

3.3.2 Improved Error Estimate for the Finite Difference Approximation

The discretisation error estimate for the finite difference approximation, cf. equation (3.6) contains higher derivatives. In case of the Coulomb potential these derivatives diverge faster than they are attenuated by the estimate, as shown in the previous section. One could conclude that the finite difference approximation therefore does not converge. However, in this chapter the error estimate is refined and it is shown, that the finite difference approximation converges.

Let D_*^2 be the exact operator of the second derivative, u_* a vector of sampled values of the exact eigenfunction, λ_* the exact eigenvalue and V an arbitrary potential. Then the Schrödinger equation reads

$$D_*^2 u_* + V u_* = \lambda_* u_*. \tag{3.45}$$

The approximate finite difference solution with the step width h on a uniform grid satisfies similarly

$$D_h^2 u_h + V u_h = \lambda_h u_h. \tag{3.46}$$

Multiplying the former equation by u_h^T and the latter equation by u_*^T and subtracting both from each other yields

$$(\lambda_* - \lambda_h) u_h^T u_* = u_h^T D_*^2 u_* - u_*^T D_h^2 u_h. \tag{3.47}$$

Where the potential terms cancel out as they are diagonal. By symmetry of the differential operators one furthermore finds,

$$\lambda_* - \lambda_h = \frac{u_h^T (D_*^2 - D_h^2) u_*}{u_h^T u_*}. \tag{3.48}$$

Expanding the exact derivative via its Taylor series $D_*^2 = D_h^2 + \frac{1}{12} h^2 D_*^4 + O(h^4)$ yields, cf. [Kä12, p. L 20],

$$\lambda_* - \lambda_h = \frac{1}{12} \frac{h^2 u_h^T D_*^4 u_* + O(h^4)}{u_h^T u_*}. \tag{3.49}$$

Now assume that the solution has converged such that u_h is sufficiently close to u_* and that both u_h and u_* are normalised to one. Than the expression simplifies to

$$\lambda_* - \lambda_h = \frac{h^2}{12} u_*^T D_*^4 u_* + O(h^4). \tag{3.50}$$

Therefore the approximate eigenvalues of smooth functions converge quadratically with decreasing step size.

Observe that the estimate contains the fourth derivative of the eigenvector. This implies that if the respective eigenvector is not smooth, the approximation converges slower than with rate h^2 or not at all depending on the degree of non smoothness.

Now assume the potential V to be the Coulomb potential. In this case it is essential to realise that both the function u and its derivatives obtain their absolute maxima at the origin of the potential. As both vectors are normalised to one $u_*(0)$ is attenuated like h^{+1} and its fourth derivative diverges like h^{-2} as discussed in the last chapter. Therefore it holds, that

$$|u_*^T D_*^4 u_*| < \frac{C_1}{h}. \tag{3.51}$$

As a consequence the error of the finite difference method converges to zero,

$$|\lambda_* - \lambda_h| < \frac{C_1}{12} h. \tag{3.52}$$

And therefore the finite difference approximation to the eigenvalues of the hydrogenic Schrödinger equation converges linearly.

3.3.3 Improving Convergence by Local Mesh Refinement

The above analysis shows, that the finite difference and finite element discretisation on a uniform mesh converges at most linearly with respect to the element size at the origin of the Coulomb potential. In this section it is shown which global rate of convergence can be achieved with adaptive refinement. Figure 3.3.3 depicts uniform grids and adapted meshes in two and three dimensions. In principle one it is possible to compute the hydrogen spectrum on a uniform grid [Sta12]. However, the required resources to achieve sufficient accuracy exceed the capacity of a single work station.

Assumption 1 The initial mesh and the refinement algorithm guarantee that the number of split elements is proportional to the number of elements selected for refinement. If the mesh is refined iteratively, this holds also for the children of split elements in successive refinement steps. Furthermore the refinement routine shall guarantee that none of the recursively refined elements degenerates.

Then the mesh is sufficiently graded with a number of grid points of n_1 for the given tolerance h_{min}, if one starts with a coarse mesh and grades it by splitting iteratively only the element containing the Coulomb potential,

$$n_1 = C_1 \log\left(\frac{1}{h_{min}}\right). \tag{3.53}$$

This means that the element size at the origin of the Coulomb potential decays exponentially with the number of grid points n,

$$h_{min} = C_1 \exp(-n_1) \tag{3.54}$$

Assumption 2 If the solution is sufficiently smooth or the mesh is sufficiently graded to resolve the non-smoothness of the solution, then the solution converges with the maximum rate possible for a basis function of order p and quadrature rule of order $2p$, which is $O(h^{\frac{2p}{d}})$, (3.26).

This means, that if the mesh is uniformly refined in successive iterations, each of the initial elements is split into n_2 elements,

$$n_2 = C_2 \left(\frac{1}{h_{min}}\right)^{\frac{2p}{d}}. \tag{3.55}$$

Implication The solution to the hydrogenic Schrödinger equation can be computed by first grading the mesh and than uniformly refining it. To achieve a desired accuracy the number of grid points n is,

$$\begin{align} n &= n_1 \cdot n_2 \tag{3.56} \\ &= C_1 \log\left(\frac{1}{h_{min}}\right) \cdot C_2 \left(\frac{1}{h_{min}}\right)^{\frac{2p}{d}} \tag{3.57} \end{align}$$

One realises that the full convergence rate of $\frac{2p}{d}$ is not achieved. However, the actual converge is not considerably inferior. Furthermore one achieves close to optimal convergence rates by adaptively refining the pre-graded mesh instead of uniformly refining it.

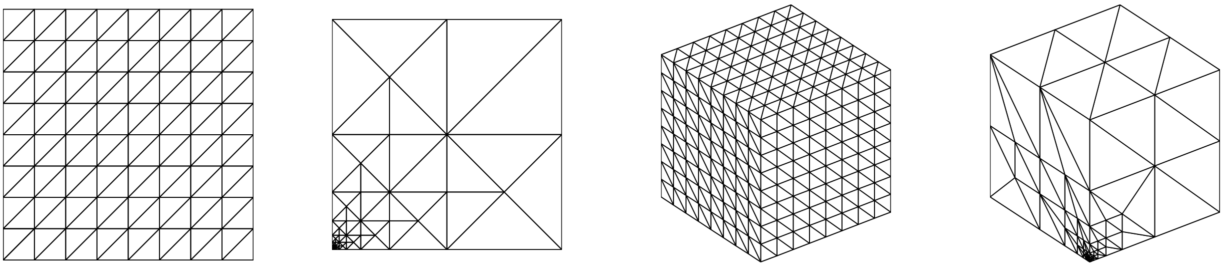


Figure 3.2: Uniform tessellations of the square and cube as well as graded meshes to resolve the Coulomb potential

3.4 Implementation of the Adaptive Finite Element Method

In the previous section it was shown that accurate approximations to Schrödinger eigenpairs can be obtained with a feasible number of mesh points by locally refining the mesh at the origin of the Coulomb potential.

The remainder of this chapter describes the essential steps of the adaptive finite element method in grater detail. These steps are the assembly of the discretisation matrices, the error estimation and the local mesh refinement. The solution of the large sparse eigensystem is addressed in the next chapter.

At first it is noted that the adaptive finite element method is superior to the finite difference methods as it does not suffer from the same limitations. So are the discretisation matrices of symmetric Hamiltonians in general symmetric, even if the mesh is locally refined and the degree of the basis function is increased. The convergence rate is moreover not reduced by local refinement and the discretisation matrix stays well conditioned even if the element size varies over several magnitudes. However, the implementation of an adaptive finite element code is considerably more complicated compared to the setup of simple difference matrices by tensor products.

This section focuses on the implementation in three dimensions. The basics of the adaptive finite element method and its peculiarities in one and two dimensions were already covered in the literature study [Kä12, p. L 29]

The finite element routine was entirely implemented from scratch. It is applicable to time independent second order partial differential equations with potential terms in one, two and three dimensions. The package was implemented in MATLAB and Java. For matrix-vector arithmetic in Java the package JAMA is used [MoSN].

Note that there exist already many finite element toolboxes, some non-commercial packages are among others PLTMG [Ban12], KASKADE [BER95] and FENICS [LMW12].

3.4.1 Definition of a Mesh

The domain is divided into triangles in 2D and tetrahedra in 3D.

Before discussing the refinement routine, it is clarified what in this thesis is referred to as a mesh.

In this thesis only meshes based on triangulations are discussed. Elements in those meshes are simplices with the least number of vertices necessary to span a hyper volume in d dimensions. In one, two and three dimensions the elements are the familiar line segments, triangles and tetrahedra. The number of vertices n_p of an element is

$$\begin{aligned} n_p &= d+1, d = 1,2,3,\dots \\ &= 2,3,4,\dots . \end{aligned} \tag{3.58}$$

The number of vertices equals the number of the $d-1$ dimensional faces. Vertices are shared between adjacent elements.

In d dimensions the number of edges per element n_e is,

$$\begin{aligned} n_e &= \frac{d(d+1)}{2}, d = 1,2,3,\dots \\ &= 1,3,6,\dots . \end{aligned} \tag{3.59}$$

In three dimensions a mesh consists of

- a set of vertices,
- a set of edges, each connecting two vertices,
- a set of tetrahedral elements, having four corner vertices connected by six edges,
- a set of triangular faces at the boundary.

For the mesh to be regular, the elements are not allowed to overlap with each other or to leave space in the domain bound by the surface faces empty. In addition no vertices are located anywhere else than the corners of the elements and edges do not cross. Note that the triangulation of a set of vertices is not necessarily unique.

Lagrangian polynomials of arbitrary degree are implemented as basis functions, see figure 3.4.1 For Lagrangian basis functions these are equally spaced points inside the element. The unit elements and the respective points supporting the Lagrangian basis functions are shown in figure 3.4.1. Note that there exist different kind of basis functions, most notably Hermite polynomials and spectral elements [Pat84].

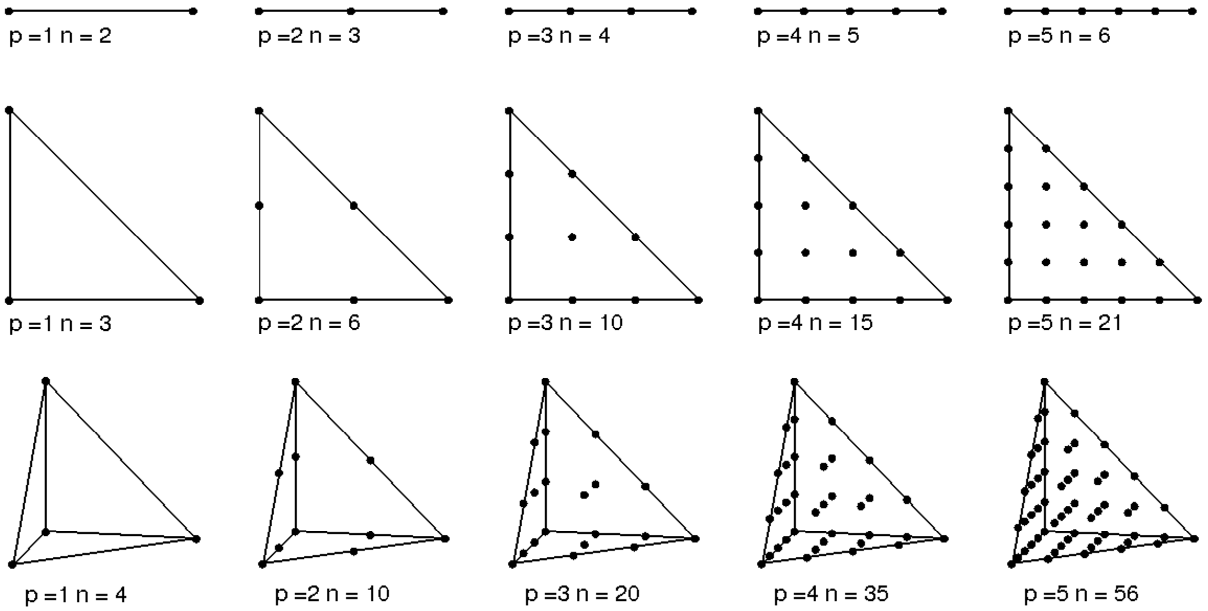


Figure 3.3: Degrees of freedom of the Lagrangian basis functions for one, two and three dimensional elements

3.4.2 Assembly of the Discretisation Matrices

The set up of the finite element matrix is more complicated than the set up of the finite difference discretisation matrix. The finite element assembly with numerical quadrature runs over each element and processes the following steps.

1. The homogeneous point coordinates of the current element are loaded into the rows of the matrix X. The corner points are always stored in the first d rows. In d dimensions an element with polynomial basis functions of degree p has n degrees of freedom,

$$n = \binom{p+d}{d}. \tag{3.60}$$

2. The matrix Q containing the coordinates of the quadrature points is computed by multiplying the matrix B with the first $d+1$ rows of X,

$$Q = B \begin{bmatrix} x_1 \\ \dots \\ x_{d+1} \end{bmatrix}. \tag{3.61}$$

   Where B contains the barycentric coordinates of the quadrature points on the unit element.

3. The hyper volume of the element is computed, this is the scaled determinant of the submatrix of size $(d+1) \times (d+1)$ of X,

$$v = \frac{1}{d!} \left| \begin{bmatrix} x_1 \\ \dots \\ x_{d+1} \end{bmatrix} \right|. \tag{3.62}$$

In one, two and three dimensions the hyper volume corresponds to length, area and volume respectively.

4. The Vandermonde matrix of the basis points is set up. In one dimensions the columns of the Vandermonde matrix are simply the powers $0 \ldots p$ of the coordinates,

$$V_x = [x^0, \ldots, x^p]. \tag{3.63}$$

5. The Vandermonde matrix is inverted, yielding the matrix of test function coefficients C,

$$C = V_x^{-1}. \tag{3.64}$$

6. In case of the stiffness matrix, the test function polynomial is differentiated once with respect to each coordinate axis, yielding d coefficient matrices, one for each partial derivative.

7. The Vandermonde matrix of the quadrature point coordinates is set up. For the stiffness matrix in one dimension the maximum power equals again p,

$$V_q = [q^0, \ldots, q^p]. \tag{3.65}$$

   For the stiffness matrix the number of columns in V_q is reduced by one.

8. The values Φ of the test functions at the quadrature points are computed by multiplying the Vandermonde matrix of the quadrature points V_q with the matrix of the test function coefficients C,

$$\Phi = V_q C. \tag{3.66}$$

   In case of the stiffness matrix the C is replaced by the matrices of the partial derivatives.

9. The values of the PDE coefficients at the quadrature points are computed. For the Coulomb potential with $r = ||x||_2$ this is,

$$f(\vec{x}) = \frac{1}{r}. \tag{3.67}$$

   For the stiffness matrix and mass matrices this is the constant $f = -\frac{1}{2}$ and $f = 1$ respectively.

   Special care has to be taken to set up the Schrödinger equation. It is necessary to place the Coulomb potential into the origin of the coordinate system to minimise cancellation errors. Furthermore one has to choose carefully the location of the mesh points in combination with the quadrature rule to avoid the evaluation of the potential at the origin. Note that it is favourable to place a triangle corner point directly at the singularity in combination with Lagrangian basis functions [SF88, p.56]. So if one applies the trapezoidal rule, the nucleus has to be shifted by an infinitesimal amount away from the origin.

10. Finally the entries into the discretisation matrix are obtained by forming the generalised inner product,

$$M = v\ \Phi^T\ \mathrm{diag}(w)\ \mathrm{diag}(f)\ \Phi. \tag{3.68}$$

   Subsequently the values M are written to the global discretisation matrix. For the stiffness matrix in higher dimensions M is the sum of d inner products of the partial derivatives.

In higher dimensions the Vandermonde matrix has to be generalised by mixed powers of the coordinates. The actual implementation pre-computes some values and exploits symmetry.

Note that there occur two set of points in the finite element assembly. The first set contains the supporting points of the basis functions, at those points values of the approximate solution are explicitly computed. The second set contains sample coordinates of the quadrature rule. One obtains a diagonal mass matrix, if both sets are equal.

For the element integral approximation an extensive set of Gaussian and Newton-Cotes rules was implemented. Newton cotes formulae where derived using the scheme laid out in the literature study [Kä12,

p. L 33]. Gaussian quadrature rules are nontrivial to derive and were taken from [Jin84, Kea86][1]. Note that the of the class Newton-Cotes rules only the trapezoidal rule in combination with linear basis functions is suitable for the solution of eigenvalue problems, as the mass matrices of higher order Newton-Cotes rules are singular.

3.4.3 A Posteriori Error Estimates for Mesh Adaption

A good error estimate of the approximate solution is essential for the fast convergence of the adaptive finite element method. In this thesis two different methods of error estimation were implemented.

The first method is based on the comparison of approximations obtained with basis functions of different degree. A Lagrangian basis polynomial of order $2p$ has the same vertices of as the basis function polynomial with order p on the once uniformly refined mesh, figure 3.4. Therefore the rows and columns of the discretisation matrices for both cases refer to the same unknowns. It is possible to extend the error estimate (3.49) to the finite element method and to compute an error estimate by subtracting the discrete operators from each other,

$$\lambda - \tilde{\lambda} = \frac{1}{\tilde{u}^T u} \left(\tilde{u}^T B^{-1} A u - u^T \tilde{B}^{-1} \tilde{A} \tilde{u} \right), \tag{3.69}$$

where in $A = -\frac{1}{2} D^2 + V$ in case of the Schrödinger equation. This error estimates does neither require to solve the discretised system a second time nor the interpolate from one solution onto another mesh.

The second method is based on an approximation via higher derivatives based on equation (3.26). The error is estimated by semi-norms of higher derivatives similar to [EJ88]. The leading term in the finite element error estimate is one degree higher than the larges non-zero derivative of the basis functions. This derivative has therefore to be estimated. This is done by computing first the derivative of order p and then by approximating the next higher derivative by differentiating between each neighbours. In case of linear basis functions this becomes in d dimensions [EJ88],

$$|D^2 u_j| \approx \max_{k=1,\dots,d} \left(\sum_{i=1}^{d+1} \left(\frac{1}{h_{ij}} \left(\frac{\partial}{\partial x_k} u_i - \frac{\partial}{\partial x_k} u_j \right) \right)^2 \right)^{1/2}, \tag{3.70}$$

where h_{ij} denotes the distance between the centres of the elements i and j.

Note that all both error estimates require the mesh to be sufficiently graded to resolve higher derivatives. Otherwise the error is underestimated.

Beside the semi-norms of the higher derivatives, one requires also a measure of the size h and the regularity σ of an element.

As measure of the element size twice the radius of the circumcircle of the element is chosen,

$$h = 2||\vec{x}_m - \vec{x}_1|| = \dots = 2||\vec{x}_m - \vec{x}_d|| \tag{3.71}$$

$$x_m = \frac{2}{d} \sum_{i=1}^{d} x_i, \tag{3.72}$$

where $\vec{x}_i$ is any of the corner vertices of the element and x_m its midpoint, e.g. the centre of the cicumcircle. There also exist other measures for the element size, such as any norm of the edges of an element. The radius of the circumcircle is an appropriate choice as at the midpoint of the circumcircle the interpolation error is maximised.

The adaptive finite element method only converges, as log as the elements do not degenerate. In d dimensions an element does not degenerate, as long as the quotient of the hyper volume v and the length h and the d-th power of the edge length h is bound away from zero,

$$0 < \sigma = \frac{v}{h^d}. \tag{3.73}$$

[1] Note that the tabulated values in [Kea86] are pre-weighted by 1/6 and that the first weight of the degree 8 formulae should be negative ($-0.039327\dots$).

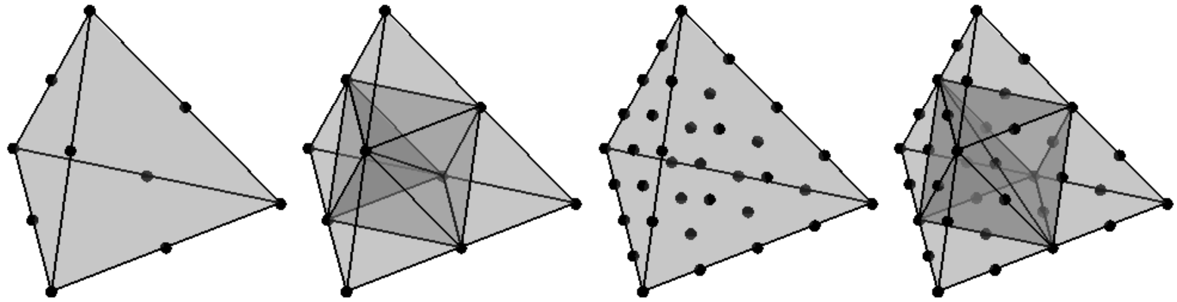


Figure 3.4: The number and coordinates of the vertices of the Lagrangian basis function of degree p and of the basis function of $\frac{1}{2}p$ on the uniformly refined mesh coincide, left pair: $p = 2$ and $p = 1$, right pair $p = 4$ and $p = 2$.

There are exist different measures for the element degeneracy, for example the minimum sine of interior angle of the element faces or the smallest singular value of the edge matrix.

Note that the refinement algorithm guarantees that all elements remain regular, e.g. do not degenerate upon iterative refinement. In this case the different measures of the element size and degeneracy are equal up to some constant.

Before the mesh is refined the eligible elements have to be selected based on the error estimate. As the edges of each refined element are halved, it is suitable to select all elements where the the local error exceeds, cf. (3.26),

$$|u_i^* - u_i^h| > \left(\frac{1}{2}\right)^{2p} ||u_* - u_h||_\infty, \tag{3.74}$$

3.4.4 Adaptive Mesh Refinement

There are different techniques to improve a finite element solution, in this the the method of increasing the degree of the basis function polynomial p and mesh refinement, e.g. the reduction of the element size h are applied. The relocation of mesh points is not considered.

Two types of refinement algorithms find wide application, those are bisection routines and red-green routines. The bisection routines are not discussed here.

In 2D a variant of the red-green-blue refinement without explicitly stored hierarchical structure is implemented [Car04]. The 3D refinement routine is based on hierarchical red-green refinement [BSW83]. The red-green refinement routine halves all edges of the elements marked for refinement and tessellate the refined elements at first. This is called regular or red refinement. In addition elements at the interface of refined and unrefined red element are only tessellated into smaller elements, without splitting additional edges. This is called closure or green refinement. Mesh degradation is avoided, as green elements are never split twice, but rather the parent element is split regularly. This implementation requires a tree-like storage structure of the elements and their children. The actual mesh is generated from the tree before assembling the discretisation matrices.

Red-green refinement allows rapid convergence upon iterative refinement, if the elements for refinement are selected appropriately, as on one hand the edge length in the refined region is halved and on the other hand the refined region can be selected very locally. Red-green refinement is superior to bisection refinement with respect to these criteria.

The Red-green refinement allows for the implementation of Multigrid solvers due to the tree like mesh hierarchy. This is of special interest for the application of the Lanczos algorithm to discretisations with non-diagonal mass matrices.

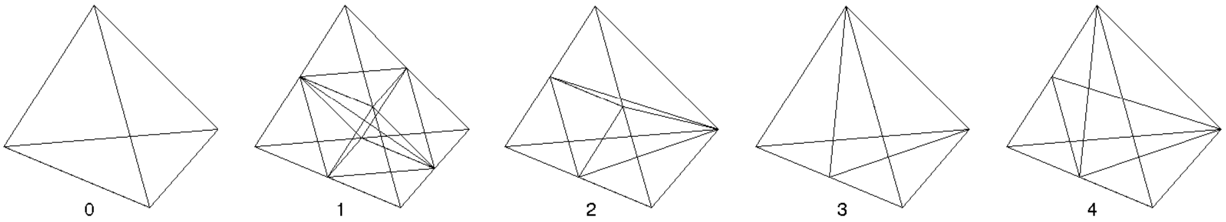


Figure 3.5: The rules 0 to 4 for the refinement of tetrahedral meshes as specified in table 3.1. To avoid mesh degradation, children of the tetrahedra split with rules 2, 3 and 4 are merged before being split again with rule 1. Note that the splitting rules for triangular meshes in tow dimensions are reflected in the faces of the tetrahedra.

Regular refinement becomes increasingly difficult in higher dimensions, as special rules have to be applied at the interface between elements which are refined and elements which are not refined. Those rules are depicted in figure 3.5

If any edge of such an element is either split or not split in two segments during refinement, than there number of all possible combinations is

$$\begin{aligned} &2^{\frac{d(d+1)}{2}}, \\ &= 1, 8, 64, \ldots. \end{aligned} \tag{3.75}$$

Note that most combinations of split edges again allows for different tessellations of the split element. Neighbouring elements have to be tessellated, such that the edges of their children coincide and do not cross.

Whilst implementing all rules in two dimensions requires a moderate effort, it becomes cumbersome in three dimensions. However, it is possible to implement just a subset of those rules. If the combination of split edges of an element is not covered by the rules, it is regularly refined by splitting its remaining edges as well.

For a regular refinement of an adaptive mesh in three dimensions following cases have been implemented, These rules are also depicted in figure 3.5. Note that the rule 4 is an extension to the minimum set of rules, as otherwise rule 5 might be applied recursively to often and yield a uniform refinement.

Rule	Combinations	Children	Child Colour	Description
0	1	0	-	no edges are split
1	1	8	red	all edges are split
2	4	4	green	three edges on one face are split
3	6	2	green	a single edge is split
4	12	3	green	two neighbouring edges are split
5	40	8	red	otherwise split all edges and apply rule 1

Table 3.1: Rules for refinement of tetrahedral meshes in three dimensions as depicted in image 3.5.

Furthermore two rules must be checked before refining each element to guarantee mesh conformity, e.g. that there are no unsplit elements where an edge has been split twice,

- Before an edge is split twice, every of the elements who share the edge have to be split.
- Before the child of an element is split, all the neighbours of this element have to be split.

This also leads to a gradual change in the element size. Note that if elements are split recursively by the above rule, the recursion terminates quickly, as the edge length in each recursion level doubles.

It is important that those rules guarantee that the mesh does not degenerated upon successive refinements.

The actual mesh is generated from the tree structure after each refinement step before the assembly. Note that the mesh refinement routines only need to be implemented for basic elements where only the corner points are specified. Additional points of required for higher basis functions are inserted in a step between mesh generation and assembly.

Chapter 4

Choice of the Eigenvalue Solver

The finite element discretisation yields an algebraic eigenvalue system of the form,[1]

$$Ax = \lambda Bx, \tag{4.1}$$

where λ is an eigenvalue and x is an eigenvector. The matrix $B^{-1}A$ acts on the vector x as if it where the scalar λ. The matrices A and B are large, sparse and symmetric. B is in addition also positive definite.

Only the eigenvalues at the lower end of the spectrum are of physical interest. The smallest algebraic eigenvalue satisfies

$$\lambda_1 = \min \frac{x^T Ax}{x^T Bx}. \tag{4.2}$$

Subsequent eigenvalues λ_i are minimise the same system with the additional constraint that the vector x_i is orthogonal to $x_1, \ldots, x_{i-1}$.

Due to the large size of the matrices it is infeasible to use standard methods like the QR algorithm to compute the eigenvalues. One has to use specialised eigensolvers which exploit the sparsity of the system. The choice of an appropriate eigensolver depends on the dimensionality of the problem and the finite element scheme used to set up the discretisation matrices. For solving the eigenvalue problem three different algorithms were considered. Those are the Lanczos method, the implicitly restarted Arnoldi's method and the Jacobi-Davidson method.

A detailed overview on eigenvalue algorithms is given in the literature study [Kä12, p. L 51]. This chapter summarise the important parts with respect to computing the hydrogen spectrum. The summary is concluded with some numerical experiments in the next chapter.

4.1 Lanczos Method

The Lanczos method factorises a symmetric matrix A in an iterative three term recurrence into an orthonormal basis V_k and into a symmetric tridiagonal matrix T_k [Lan50],

$$AV_k = V_k T_k + r_k e_k^T. \tag{4.3}$$

The residual r_k drops to zero once the factorisation is completed, e.g when k equals the dimension n of A or if an invariant subspace is explored. In that case the eigenvalues of the matrices A and T_n are equal, as both matrices are similar. If one only computes a partial factorisation, then the eigenvalues of T_k are approximations of the exterior eigenvalues of A. In two and higher dimensions the exterior eigenvalues converge quickly and the required dimension of T_k to achieve a desired accuracy is much smaller than the rank of A.

The eigenvalues of T_k can be efficiently computed using the bisection algorithm. The error can be estimated with the eigenvectors of the projected matrix T_k, without computing the approximate eigenvectors

[1] For convenience the arrow indicating vectors is omitted in this chapter.

of the matrix A. As the algorithm is only based on a three term recurrence, only two vectors of the basis V need to be kept in storage at any time. The matrix-vector multiplications are cheap, as long as A is sparse.

One realises that the advantage of the Lanczos algorithm is, that it is suitable for fast computation of exterior eigenvalues of large sparse eigensystems. The drawback is, that a linear system has to be solved at each iteration step for generalised eigenvalue problems. For very large systems the Lanczos method is therefore only efficient for discretisations with diagonal mass matrices, where the solution of the linear system is inexpensive or where the generalised eigensystem can be inexpensively casted into a standard eigenvalue problem,

$$Ax = \lambda Bx \tag{4.4}$$

$$L^T L = B \tag{4.5}$$

$$Lx = y \tag{4.6}$$

$$L^{-T} A L^{-1} y = \lambda y. \tag{4.7}$$

Therefore it is only reasonable to apply the Lanczos method to discretisations in combination with Newton-Cotes quadrature, which yield a diagonal mass matrix, but have a lower order of accuracy than Gauss-Quadrature.

Note that there exist several ways to combine the Lanczos method with higher order finite element discretisations nevertheless. The first kind is to use special basis functions yielding diagonal matrices. This is nontrivial as the mass matrix B is required to be positive definite. This approach requires more unknowns for the same accuracy compared to non diagonal discretisation schemes and becomes infeasible in higher dimensions [CJJT01]. The second approach is to solve the B-system iteratively and to recycle the Krylov subspace. The reader is referred to [Saa87]. The third way is to use the Multigrid method to solve the B-system. This requires special refinement and assembly routines [BDY87]. Finally it might also be easier to construct higher order schemes on quadrilaterals which yield diagonal mass matrices.

There is another issue with the Lanczos method which is the loss of orthogonality of the eigenvectors. The loss of orthogonality is directly connected to the convergence of exterior eigenvalues and cannot be avoided. It leads to spurious eigenvalues which are either redundant copies of converged eigenvalues or additional eigenvalues which are not related to any eigenvalue of A. There are two ways to address this issue. Firstly one may reorthogonalise the basis when the orthogonality is lost, see [Sim84]. One usually refrains from this approach, as this voids the advantages of the Lanczos method as one has to store the complete basis and additional matrix vector products are required to restore the orthogonality. Secondly one may filter out spurious eigenvalues. An inexpensive and reliable method to achieve this was discovered by Cullum and Willoughby [CW02, CW81]. They compare the eigenvalues of the matrix $T_{1..m}$ with the eigenvalues of the submatrix $T_{2..m}$ and then filter the eigenvalues according to their occurrence in both matrices.

There are two more issues, which are however less problematic. The plain Lanczos method is unable to find the correct multiplicity of the eigenvalues. This is mostly a theoretic problem, as one may choose a discretisation such that the discretisation error makes each eigenvalue unique, cf. section 5.3.1. The Lanczos method also does not compute the eigenvectors, if one does not store the complete basis V. The eigenvectors are required for the local error estimate of the adaptive mesh refinement. However, the eigenvectors can be easily computed using the shift-and-invert method once the eigenvalues are computed (using iterative method if the matrix becomes to large).

In context of this thesis the FORTRAN routines provided by Cullum and Willoughby is used, [CW]. Their FORTRAN routines were adapted to make them compile on a current Linux system and were equipped with a MATLAB interface for convenience, see the companion source code documentation.

It is important to provide the Lanczos implementation by Cullum and Willoughby with appropriately chosen parameters. Those are the dimension of the T-matrix to be computed, the number of eigenvalues to compute before post-processing, and an upper and a lower bound of the desired eigenvalues. A good choice for the size of the T matrix is about $k\sqrt[d]{n}$, where k are the number of desired eigenvalues, d is the dimension of the problem and n is the rank of the discretisation matrix. The range of eigenvalues for the quasi unconfined hydrogen atom in two dimensions following equation (6.1) is $[-2, 0]$ and for the three dimensional atom it is $[-0.5, 0]$. For the confined atom the upper bound has to be raised. It was found that the number of computed eigenvalues from the T-matrix had to be equal to its rank. This is not optimal as the implementation should be improved at this point, as otherwise eigenvalue computation by the bisection routine becomes slower then the Lanczos factorisation itself.

4.2 Implicitly Restarted Arnoldi Method

The Arnoldi method is basically Lanczos method for non Hermitian matrices [Arn51]. Instead of factorising the matrix A into an orthogonal basis V_k a symmetric tridiagonal matrix T_k with an three term recurrence, it factorises A into an orthogonal basis V_k and a Hessenberg matrix H_k with a $k+1$-term recurrence,

$$AV_k = V_k H_k + r_k e_k^T. \tag{4.8}$$

In the Hermitian case H_k will be real symmetric tridiagonal and hence equal to the tridiagonal Lanczos matrix T_k. One realises, that the Arnoldi method implicitly avoids the loss of orthogonality observed with the Lanczos method by means of a full orthogonalisation. Due to the storage of the complete basis and the full orthogonalisation Arnoldi's method is computationally rather expensive compared to the Lanczos method. Therefore a spectral transformation is introduced to let the desired eigenvalues converge faster,

$$(A - \sigma B)^{-1} B x = \nu x. \tag{4.9}$$

The eigenvectors of the transformed system coincide with those of the untransformed problem. The eigenvalues become

$$\lambda = \frac{1}{\nu} + \sigma. \tag{4.10}$$

If one computes the smallest eigenvalues of Poisson type problems like the hydrogenic Schrödinger equation, a shift close to zero will make the desired eigenvalues large and well separated in the transformed problem. Those are the values of interest. The eigenvalues will therefore converge in just a few steps compared to the untransformed problem. The number of the k-term recurrence is furthermore reduced by restarting the algorithm. However, the relatively low number of iterations comes at a price of a matrix factorisation and the additional solution of triangular systems at every step. Note that the factorisation cannot be replaced by a plain iterative solver, as the system needs to be solved many times. In such a case factorisation is the method of choice in two dimensions. Note again the aforecited papers about the successive solution of the same system with different right hand sides. The application of Multigrid is more intricate in this case, as the system is indefinite. As the matrix is factorised, there is no penalty incurred with non-diagonal B-matrices. The Arnoldi method can therefore be well combined with higher order accurate finite element discretisation.

Arnoldi's method is efficiently implemented in ArPack [Sor92, LSY98]. It is available in MATLAB via the *eigs* and *arpackc* commands. An elaborate migration and adaption of source code as for the Lanczos method was therefore not necessary.

4.3 Jacobi-Davidson Method

The Jacobi-Davidson method is an advanced eigenvalue method which computes the eigenpairs of a matrix A directly via a Newton-like iteration [SvdV00, SBFvdV96]. A small orthogonal search space Q is used to project the matrix A onto a small non-sparse matrix M. Those eigenvalues are by similarity approximations of the eigenvalues of A. The eigenvalues of the projected matrix can be readily computed by standard factorisation methods due to its small size. The most important part of the algorithm is the expansion of the subspace. The current residual r of the approximate solution x is reduced by solving the correction equation,

$$r = (A - \theta I)x \tag{4.11}$$

$$(I - QQ^T)(A - \theta I)(I - QQ^T) = -(I - QQ^T)r. \tag{4.12}$$

And the search space is subsequently expanded by the solution of the correction equation orthogonalised against the current search space. The search space is kept small by deflation. The correction equation has to be solved iteratively as it is singular.

The Jacobi-Davidson method converges in very few iterations compared to the Lanczos or to the Arnoldi method if it is provided with a reasonable start space. It is very suitable if one computes only a few eigenpairs of very large matrices and it is therefore well applicable to high dimensional problems.

Beside choosing a good initial search space it is very important to provide appropriate parameters for the Jacobi-Davidson method. So should the correction equation not be solved correctly. An accuracy of just 10^{-2} resulted in the shortest run times for the discrete Laplacian. Another important parameter is the size of the search space. It should be small and have a size between $k + 5$ and $k + 10$, where k is the number of desired eigenvalues. A profound study of the Jacobi-Davidson method, choices for the solution of the correction equation and parameters can be found in [Geu02].

A MATLAB implementation of the Jacobi-Davidson method on the inventor's homepages [Slea]. However, to study the method in greater detail it was implemented following the description in [BDJ$^+$00, SvdVB99]. The details of the implementation are given in the literature study [Kä12, p. L 56].

Chapter 5

Numerical Experiments

This chapter has two parts. In the first part the application of the finite element routine laid out in chapter 3 is demonstrated. In the second part practical aspects of the solvers for large sparse eigensystem discussed in chapter 4 is investigated.

5.1 Finite Element Routine

This section demonstrates the practical performance of the implemented adaptive finite element routine. In the first part the application to the test case of the Laplacian operator is demonstrated. In the remainder of the section the application to the hydrogenic Schrödinger equation is discussed.

5.2 Laplacian Operator

This section shortly demonstrates the convergence of the *hp*-finite element approximation to the smallest eigenvalue of the negative Laplacian operator, see figure 5.1. As the eigenfunctions of the Laplacian operator are smooth, the eigenvalues can be accurately approximated with uniform grids having relatively few grid points. In three dimensions more grid points are required to achieve the same accuracy than in two dimensions, if basis functions of the same degree are used. Note, that in two dimensions the accuracy of higher order approximations ($p = 5$ and $p = 6$) deteriorates early due to cancellation errors.

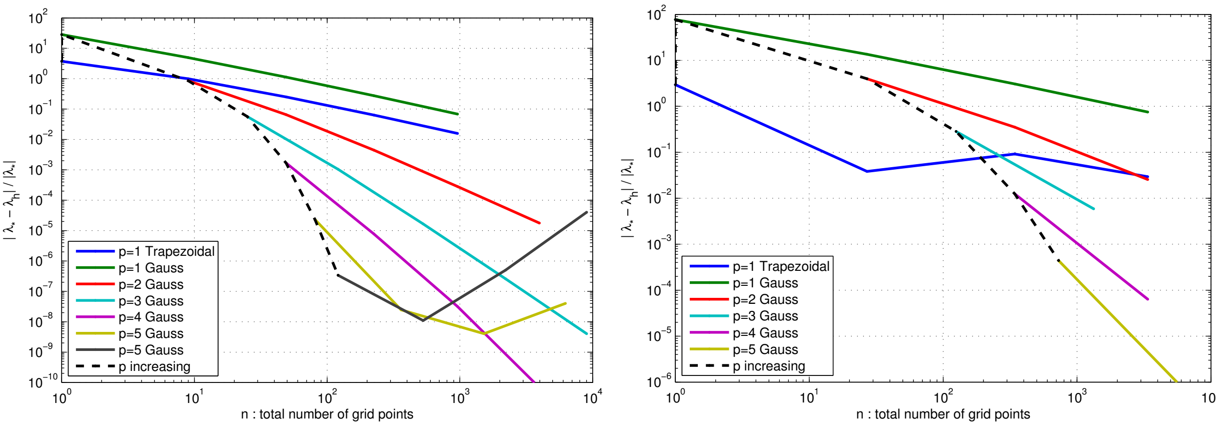


Figure 5.1: Convergence of the finite element approximation to smallest eigenvalue of the negative Laplacian operator, left 2D, right 3D, p denotes the degree of the polynomial basis functions.

5.2.1 Hydrogenic Schrödinger Equation on Uniform Grids

This example demonstrates the slow convergence of different discretisations on uniform grids to the ground state of the unconfined hydrogen atom.

Figure 5.2.1 shows the convergence of the finite difference and finite element approximation in two dimensions.[1] Note that on a uniform grid the discretisation matrices obtained with the finite difference method and the finite element method in combination with linear basis functions and the trapezoidal rule are equal.

- The observed convergence is only of order $h = \frac{1}{\sqrt{n}}$, confirming the error estimate (3.52). Note that the approximation of smooth functions would converge twice as fast with linear basis functions. Note furthermore that in three dimensions the convergence with respect to the number of grid points is even slower, e.g. $h = \frac{1}{\sqrt[3]{n}}$.
- The rate of convergence is not improved by using higher order discretisations.

The next experiment shows that better convergence is achieved by applying an adaptive finite element method in combination with higher degree basis functions.

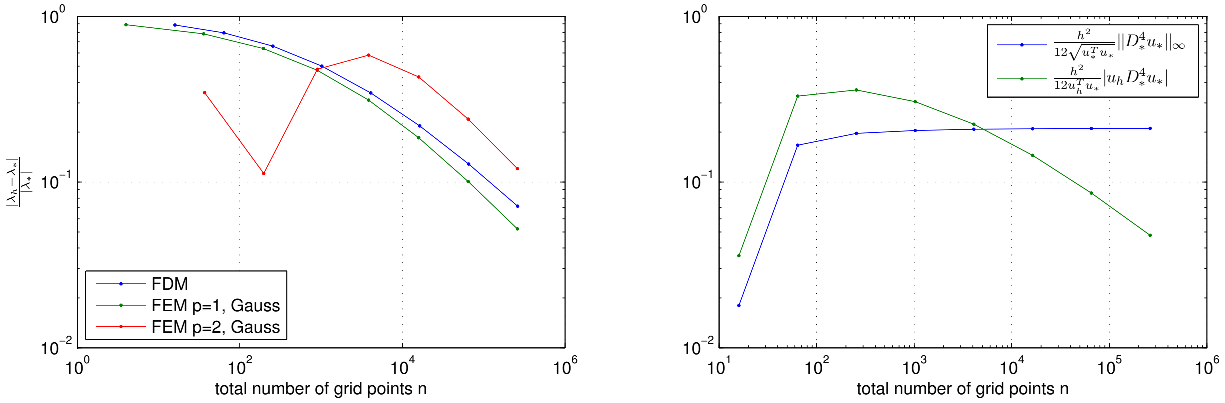


Figure 5.2: The left plot shows the convergence of the finite difference and finite element method on a uniform grid for the ground state of the hydrogen atom in the domain $L = [10, 10]a_0$. The right plot shows the non-convergent standard error estimate $\frac{h^2}{12\sqrt{u_*^T u_*}}||D^4 u_*||_\infty$, (3.6) and the refined converging error estimate $\frac{h^2}{12 u_h^T u_*} u_h^T D^4 u_*$, (3.49). Note that the refined error estimate (green graph right) is a good approximation of the observed error (green graph left).

5.2.2 Convergence of the Adaptive Finite Element Approximation in Two Dimensions

This section demonstrates the convergence of the adaptive finite element method to the ground state of the unconfined hydrogen atom in two dimensions. Figure 5.3 shows the convergence of several discretisations obtained with different basis functions degree. The eigensystem is solved by the implicitly restarted Arnoldi method in shift and invert mode. The convergence is significantly faster than that observed for discretisations on uniform grids, cf. 5.2.1. The rate of convergence does furthermore increase with increasing basis function degree and is not reduced to the rate of linear basis functions. The discretisation error drops below the model error for about $2 \cdot 10^4$ mesh points, 2.2.

Note that the convergence of higher degree polynomials is delayed, e.g. a minimum number of grid points is required before convergence sets in. Therefore it is not optimal to choose the basis function degree as high as possible. However, the advantage of higher order discretisations is more pronounced if one approximates several eigenfunctions on the same mesh or if one requires a higher accuracy. Note that higher degree polynomials do reduce the sparsity of the discretisation matrix and that for very high degrees the Vandermonde matrices become singular due to rounding errors.

[1] All timings in this chapter refer to the system:
Intel(R) Core(TM)2 Duo CPU P8700 @ 2.53GHz, 8GB Ram
MATLAB Version 7.12.0.635 (R2011a)
Debian GNU/Linux 6.0
javac 1.6.0_18
GNU Fortran (Debian 4.4.5-8)

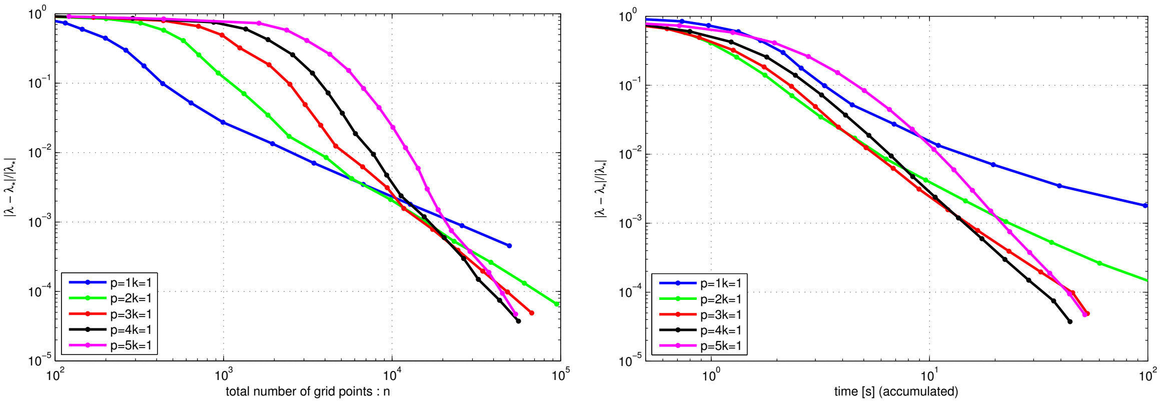


Figure 5.3: Convergence of the finite element approximation to the ground state of the hydrogen atom in two dimensions. The side length of the domain is $L = 128a_0$ (quasi unconfined).

5.2.3 Convergence of the Adaptive Finite Element Approximation in Three Dimensions

Here the convergence of the adaptive finite element method for the ground state for the unconfined hydrogen atom in three dimensions is demonstrated. At first it is observed that not number of grid points to achieve a desired accuracy is not considerably higher than in two dimensions. The second observation is, that the convergence of discretisations obtains with higher degree basis functions sets in later than in two dimensions. Thirdly the run time for a large number of unknowns becomes much larger than in two dimensions for the same number of unknowns, as the factorisation of the matrices is more expensive.

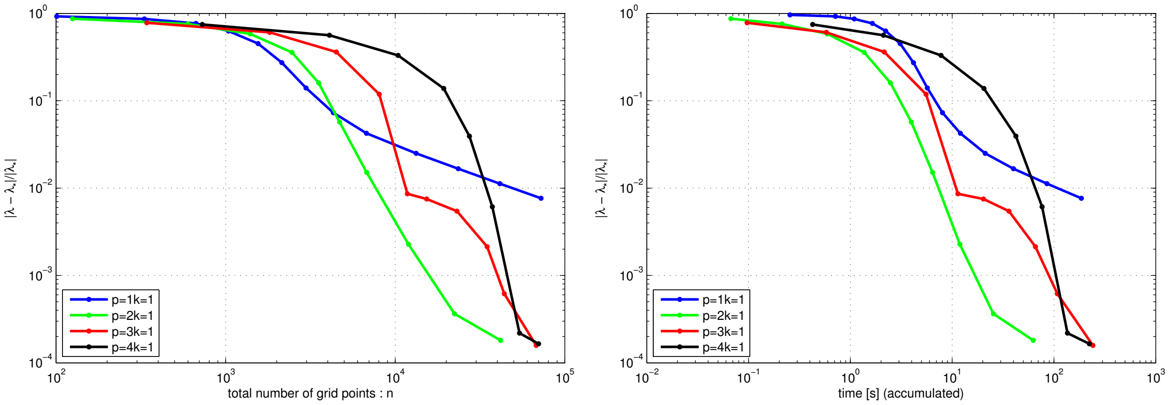


Figure 5.4: Convergence of the adaptive finite element approximation to the hydrogen ground state in three dimensions. The domain side length is $L = 128a_0$ (quasi unconfined).

5.2.4 Run Time of Individual Components of the Adaptive Finite Element Routine

The run time of the eigenvalue solver becomes the dominant component in two dimensions when several eigenvalues are computed. In three dimensions the run time of the eigenvalues solver is the dominant component even if only one eigenvalue is approximated due to the higher fill in during matrix factorisation, see figure 5.5.

5.2.5 Mesh Quality

According to equation (3.26), the approximation is expected to converge with rate $\frac{2p}{d}$ with respect to the number of grid points n. However, at the location of the Coulomb potential the convergence is expected to be just linear with respect to the element size. Figure 5.6 shows the results of a test run to verify these assumptions. The global convergence is indeed of rate $\frac{2p}{d}$ with respect to the number of grid points, whilst

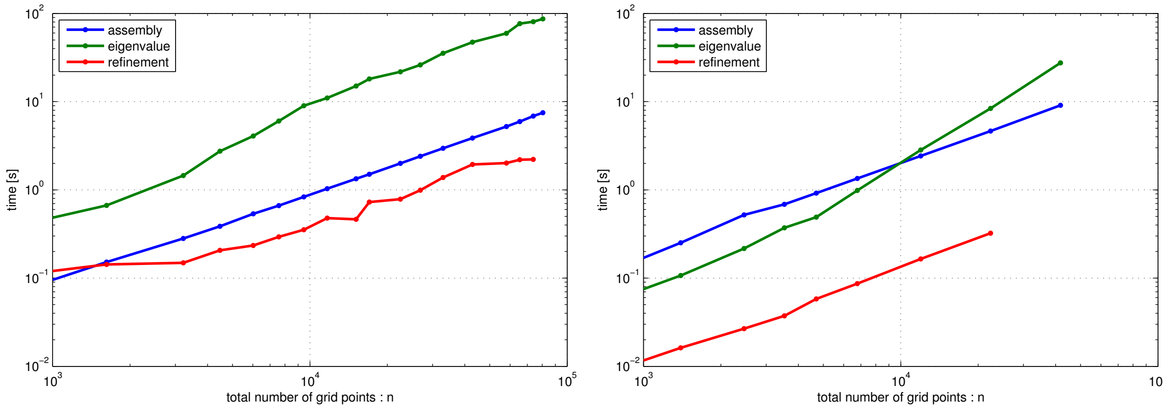


Figure 5.5: Run time contribution of the individual components of the finite element routine, left approximation with $p = 5$ of the 10 lowest eigenvalues in the two dimensional domain $L = 160a_0$, right approximation with $p = 2$ of the lowest eigenvalue in the three dimensional domain with $L = 128a_0$.

the accuracy of the eigenfunction is just linear with respect to the smallest element. This numerical experiments therefore verifies the theoretic estimate 3.3 about the convergence at the origin of the Coulomb potential.

To achieve convergence it is important to avoid mesh degeneration upon iterative local refinement, cf. equation (3.73). Figure 5.6 shows that the mesh quality is well preserved. In two dimensions the minimum angle α_{min} remains constant whilst the minimum edge length is reduced by several magnitudes. Similarly in three dimensions the ratio of the element volume and cubed element size min $\frac{v}{h^3}$ remains constant whilst the minimum edge length is reduced by several magnitudes.

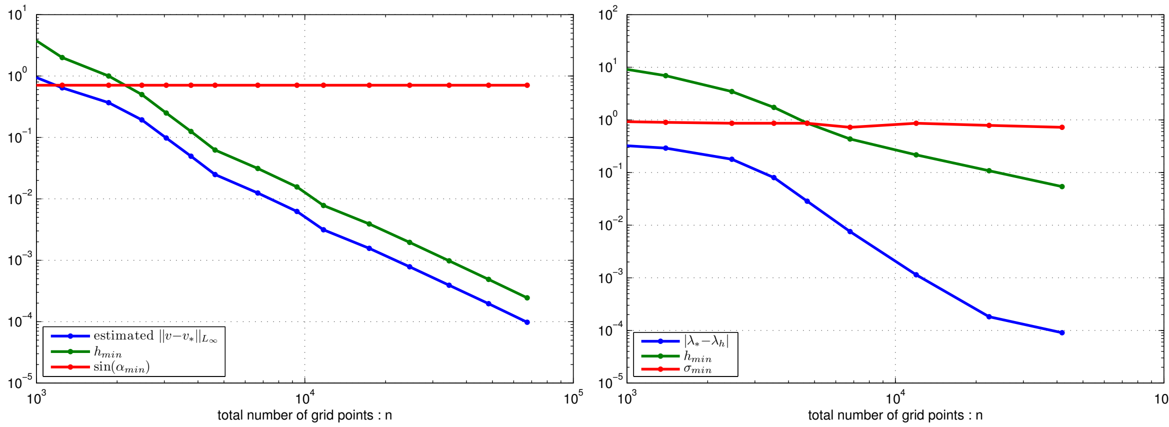


Figure 5.6: Quality of the adaptively refinement meshes in 2D (left, $p = 3$) and 3D (right, $p = 2$) in the domain with side length $L = 128a_0$. The mesh quality does not degenerate measured by the minimum angle $\sigma = \alpha_{min}$ or minimum ratio of element volume and cubed element size $\sigma = \frac{v}{h^3}$.

5.2.6 Impact of Confinement on the Run Time

The spectrum of the hydrogen atom changes upon confinement. This also influences the resources required to compute the eigenvalues. Figure 5.7 shows an example for two dimensions.

- The problem becomes easier with increasing confinement. If the confinement is increased, one can compute the eigenvalues for a fixed accuracy with fewer mesh points in a shorter time. The number of required mesh points to compute the 16 smallest eigenvalues drops linearly with decreasing domain size, if the domain is smaller $10a_0$.
- Whenever the spectrum is close to being indefinite, more mesh points and a longer run time is required. This is reasonable, as an eigenvalue in transition from being negative to being positive

has a large relative error as it is small compared to the discretisation error. The peak of the lowest eigenvalue at a domain size of about $1.4a_0$ is most pronounced.

- The number of grid points remains almost constant, once the domain is sufficiently large, e.g. larger than $10a_0$ for the smallest 16 eigenvalues where the computed eigenvectors are quasi unconfined. However, the run time of the eigensolver continues to increase.

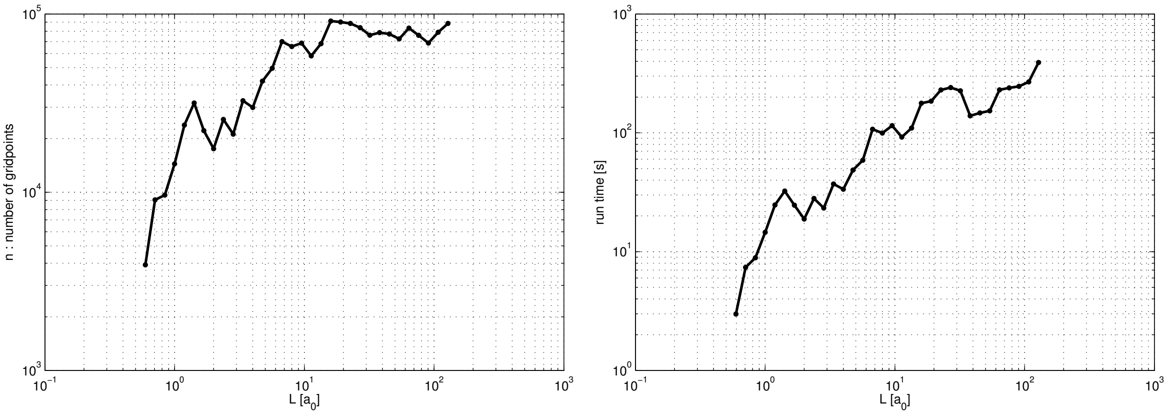


Figure 5.7: Required number of grid points (left) and run time (right) to compute 16 eigenvalues to a relative tolerance of 10^{-3} depending on the domain side length L_0.

5.3 Eigensolver

This section evaluates the performance of the solvers for large sparse eigensystems as discussed in chapter 4. The first part again deals with the application to the discrete Laplacian and the remainder of the section treats the application to the discretised hydrogenic Schrödinger equation.

5.3.1 Eigensolvers for the Discrete Laplacian

Here the performance of the eigensolvers discussed in chapter 4 are compared with respect to finding the discrete eigenvalues of the Laplacian operator discretised with the finite difference method. The Laplacian operator is well suited as benchmark as analytic solutions exists and contrary to the hydrogenic Schrödinger equation a plain uniform discretisation is sufficient. Four evaluation cases were chosen: the computation of the lowest eigenvalue in 2D and 3D and computation of the lowest 16 eigenvalues in 2D and 3D. The grid is chosen, such that the discretisation error makes the analytically degenerated eigenvalues unique. This is achieved by choosing the number of unknowns per axis to be $n \times 2n$ in 2D and $n \times 2n \times 4n$ in 3D. The measured run times are displayed in table 5.1. The Lanczos method by Cullum and Willoughby proved to be the fastest method in 3D, whilst ArPack in shift and invert mode was faster in 2D. The black-box Jacobi-Davidson method was not competitive, except for 3D when very few eigenvalues are computed. Note that the Lanczos method in the current implementation requires a rather good guess for

K	Domain Size	ArPack	JDQR	Lanczos
1	100x200	0.33	4.61	0.12
4	100x200	0.52	12.75	0.47
16	100x200	1.15	44.41	2.00
64	100x200	4.19	X	8.47
1	15x30x60	2.13	1.49	0.06
4	15x30x60	3.26	6.37	0.24
16	15x30x60	5.04	29.92	0.99
64	15x30x60	13.57	X	4.50

Table 5.1: Run times in seconds of different eigensolvers for computing the k smallest eigenvalues of the negative discrete Laplacian discretised with the finite difference method.

the bisection interval as well as the dimension k of the matrix T_k.

For the discrete Laplacian in two dimensions the factorisations necessary for the spectral transformations pay off. However, in three dimensions methods avoiding the complete factorisation of the discretisation matrices are more efficient.

5.3.2 Eigensolvers for the Hydrogenic Schrödinger Equation

The eigenfunctions of the hydrogenic Schrödinger equation are significantly harder to compute than those of the Laplacian operator. For all solvers the spectrum had been pre-shifted to make the spectrum positive definite and the mode to compute the eigenvalue with the smallest magnitude ('SM') was used.

The ArPack solver is the only one able finding the requested eigenvalues for all cases. Thought, the number of basis vectors has to be increased to ensure convergence for hard problems where the domain is large such that the eigenvectors are quasi unconfined and where there are sufficiently many unknowns to achieve a high accuracy.

For the Jacobi-Davidson method the number of outer and inner iterations had to be increased to improve convergence. However, even with an unlimited number of iterations the Jacobi-Davidson method failed to converge for hard problems in two and three dimensions. The Jacobi-Davidson method was furthermore much slower than ArPack as in the Laplacian operator example. At this point it should be noted that the convergence of the Jacobi-Davidson method can be improved by providing a good initial guess, a suitable preconditioner and by applying an implementation that exploits the symmetry of the matrices. Further improvements could be gained by adapting the FORTRAN implementation of Fokkema and van Gijzen [FvGM12] for the computation instead of using the MATLAB implementation provided by Slejpen and van der Vorst [Slea].

The run time of the Lanczos solver indicates that the matrix had to be completely factorised to find the eigenvalue with the desired accuracy. Furthermore the method did not converge for large matrices in two dimensions, even if the size of the T matrix exceeded the size of the matrix A.

The slow convergence of the eigensolvers could be due to several reasons. Firstly one observes that the wave functions of the unconfined hydrogen atom are non smooth and contain high frequency components. The orthogonal basis vectors of projection matrices can be interpreted as oscillating functions exploring each a different frequency. Therefore one concludes that one requires many basis vectors to explore the spectrum of the hydrogen wave functions with sufficient accuracy. Regarding this the choice of the starting vector should be further investigates.

Secondly this could be related to the discretisations with diagonal mass matrices obtained with the trapezoidal rule and the subsequent transformation of the generalised eigenvalue problem into a standard eigenvalue problem.

5.3.3 Structured vs. Unstructured Meshes

This benchmark shows that the performance of the matrix-vector product computation degrades if one uses an unstructured mesh, but that the penalty incurred is much lower than the benefits gained by the unstructured mesh.

The discretisation matrices of the adaptive finite element method are large and sparse. Large sparse matrix vector products do not achieve peak performance on the CPU. Both data access and operation count are of order n, the rank of the matrix. As the data access is much slower than arithmetic operations, performance of the sparse matrix vector product remains bandwidth limited. Further performance degradation is experienced for matrix vector products with unstructured matrices. Fetching a single non-cached value comes on current processors with a latency of about 50ns. This is roughly 100 times slower than an average arithmetic operations required at peak performance.
The performance can be improved by two measures which mostly remove the latency penalty

- Data elements can be pre-fetched, as the structure of the matrix is fixed.

$\log_{10}(N)$	ArPack	JD	Lanczos
0.9542	0.0194	0.0161	0.0004
1.6902	0.0704	0.0441	0.0007
2.0212	0.0821	0.0663	0.0006
2.2122	0.0602	0.1108	0.0008
2.2856	0.0584	0.0672	0.0009
2.5065	0.0924	0.1192	0.0030
2.5694	0.0532	0.1138	0.0034
2.6031	0.1048	0.1269	0.0147
2.6385	0.0929	73.8182*	0.0221
2.6712	0.1488	*	0.0265*
2.7903	0.0611	*	0.0733*
3.0175	0.0703	*	0.1400*
3.4741	0.1372	*	0.8861*
3.9145	0.2421	*	6.8537*

$\log_{10}(N)$	ArPack	JD	Lanczos
1.4314	0.0532	0.0993	0.0007
2.2227	0.0693	0.1169	0.0084
2.5340	0.0847	0.4715	0.0024
2.7356	0.1194	0.7438	0.0026
2.8463	0.1129	0.8043	0.0039
2.9138	0.0873	2.2082	0.0044
2.9809	0.0973	2.8853	0.0055
3.0410	0.1154	4.2271	0.0313
3.0927	0.1279	34.5208*	0.0724
3.1402	0.1595	*	0.0573
3.1818	0.1148	*	0.0834
3.2198	0.1341	*	0.1893
3.2548	0.1468	*	0.4098
3.5433	0.3228	*	1.1544
4.0301	1.7099	*	6.0437
4.2703	4.2547	*	13.9292
4.9054	101.6776	*	134.1197

Table 5.2: Run time of various eigensolvers for the discretised hydrogenic Schrödinger equation in seconds, left 2D, right 3D. The side length L_0 was set to 128. Linear polynomials in combination with the trapezoidal rule were applied. Asterisk' indicate the occasions where the solver did not converge, or that computation was not conducted as the previous run exceeded 30s.

- Instead of computing a single matrix vector product one can use a block of vectors. Fetching the values of several vectors with the same index simultaneously reduces the average latency, as the CPU always fetches a complete cache line.

For the test a matrix-vector products with a Poisson matrix are computed. Then the multiplication is repeated with a permuted matrix. The rows and columns are randomly permuted in a way that symmetry is preserved. The matrix rank is chosen sufficiently large compare to the cache size.

Run	Time [s]	Penalty Factor
Ab	0.47	
$A(p,p)b$	3.58	7.6x
$b^T A^T$	0.38	
$b^T A(p,p)^T$	1.80	4.7x

Table 5.3: Averaged run time for matrix vector multiplications with a 2D FDM Poisson matrix of rank $1.6 \cdot 10^7$. Transpose and permutation operations were computed before measuring.

From the measurement following conclusion can be drawn

- The MATLAB sparse matrix multiplication seems to efficiently pre-fetch data.
- The worst case measured penalty is as expected a factor of 8.
- The MATLAB sparse matrix vector multiplication implementation is not optimal,
  - it does not exploit symmetry,
  - there are differences between $b^T A^T$ and Ab, by symmetry, the run times should be equal,
  - block multiplication should increase efficiency, but it does not.

The unstructured mesh will the improve the convergence rate from being linear to almost the full rate related to the applied basis function. This benefit is much larger than the constant penalty factor. The penalty experienced for the unstructured sparse matrix vector product is with pre-fetching at most 8 and therefore much smaller than that. The advantage gained by the unstructured mesh is therefore not substantially deteriorated by the penalty incurred by the unstructured matrix vector product. The discretisation matrix of an unstructured grid is furthermore not completely random and the penalty will be lower than the worst case.

Chapter 6

The Spectrum of the Confined Two Dimensional Hydrogen Atom

In this chapter the computed eigenspectra for various cases of confinement are presented and discussed. The spectra were computed for the hydrogenic Schrödinger equation (2.1) in a two dimensional domain on a Cartesian grid. Section 6.1 covers the spectrum in case of symmetric confinement into a square domain, called quantum well. Asymmetric confinement is discussed in section 6.3. Influence of the domain shape are presented in section 6.5. A comparison to the radial Schrödinger equation is given in section 6.6. Care should be taken before conveying these results over into three dimensions. Regarding this, note also the remark on physical plausibility in chapter 2.2.

6.1 Change of Energy Levels under Symmetric Confinement

The energy levels of the hydrogen atom change under confinement. If the domain size is reduced while the nucleus stays in the centre, the energy levels change as follows:

1. The eigenvalues increase with decreasing cavity size, see figure 6.1 and table 8.1. Eventually all eigenvalues become positive and continue to diverge towards infinity while the cavity size goes to zero.

   Energy Levels in 2D Squared Domains with Central Potential
   unconfined ⇒ confined

   $$\frac{-2}{(2k-1)^2} \Rightarrow +\infty \tag{6.1}$$

2. Higher energy levels are significantly affected before changes in lower energy levels become apparent.

3. The degeneracy of the eigenvalues is partially lost. This means that some energy levels which have an equal value without confinement take on different values when the atom is confined. However, even in case of arbitrarily strong confinement some energy levels remain doubly degenerated.

   Degeneracy in 2D Squared Domains with Central Potential
   unconfined ⇒ confined
   1, 3, 5, 7, . . . ⇒ 1 or 2-fold

4. The unconfined values are separated into groups of different value but the eigenvalues of the confined atom become interleaved as the energy levels originating from different groups cross each other. This means that some eigenvalue corresponding to a specific eigenvector may increase faster than the value of corresponding to another eigenvector.

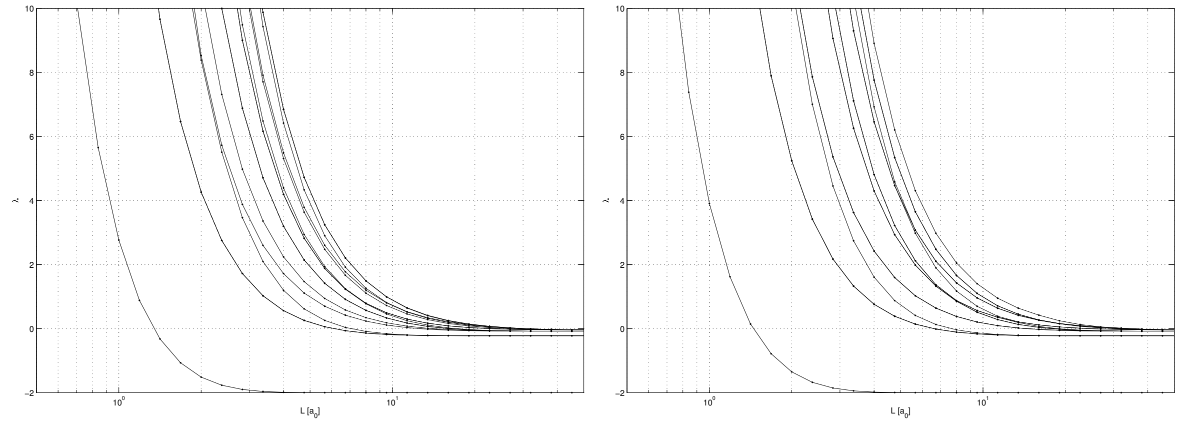


Figure 6.1: The smallest 16 eigenvalues of the two dimensional hydrogenic Schrödinger Equation confined to a square domain of side length L_0 (left) and circular domain of diameter L_0 (right). Due to degeneracy the trajections of four and respectively six eigenvalues are completely covered by others.

6.2 Interpretation of the Changes under Symmetric Confinement

1. The eigenvectors of the unconfined atom decay exponentially towards zero with increasing distance from the nucleus. The decay rate for the ground state is largest and the rate for higher energy levels decreases with each group. In case of weak confinement eigenfunctions of lower energy levels may still be zero at the boundary and are de facto unaffected, whilst the eigenfunctions of higher energy levels are already squeezed into the domain, leading to a significant change of the corresponding eigenvalue.

2. The different eigenfunctions of a specific degenerated eigenvalue can be distinguished into two types. In one group with equal shapes, where the difference is just the rotation of the eigenfunction and in others with different shapes. One realises that those eigenfunctions with different shapes are differently affected in case of confinement. The eigenvalues of those eigenvectors drift away with increasing confinement. However, those degenerated eigenvalues with equal shape are all equally effected by the confinement and hence stay degenerated, however strong the confinement becomes. One realises that an energy level affected by confinement can only be doubly degenerated in a square two dimensional domain, as there exist only two orthogonal coordinate axes.

3. The crossing of eigenvalues can also be deduced from the different shapes of the eigenvectors, e.g. there exist eigenvectors of lower energy levels which are more difficult to compress due to there shape than some eigenfunctions of higher energy levels.

6.3 Change of Energy Levels under Asymmetric Confinement

In this section the change of eigenvalues due to confinement with shift is discussed. The results are depicted in figure 6.3 and table 8.1. Only the shift parallel to one coordinate axis is considered. As an example serves a cavity with a side length of about 32 Bohr radii.

1. When the nucleus is shifted inside the cavity, the value of the energy level increases. However, the change is only small compared to the increase resulting from increasing confinement. All values remain bounded.

2. Higher energy levels are effected before lower energy levels, but lower energy levels are effected relatively stronger by the confinement than higher energy levels.

3. The degeneracy of eigenvalues is completely lost if the nucleus is shifted to a position in between the centre of the cavity of and the boundary midpoint. However, the degeneracy is regained when the atom is located exactly on the boundary.

4. The trajections of eigenvalues belonging to different eigenvalues cross each other. This crossing is more prominently pronounced than that observed with increasing confinement. When the eigenvalues are degenerated, they can be sorted into groups with identical value. Note that due to the crossing some eigenvalues are member of a different group in the centre than on the boundary.

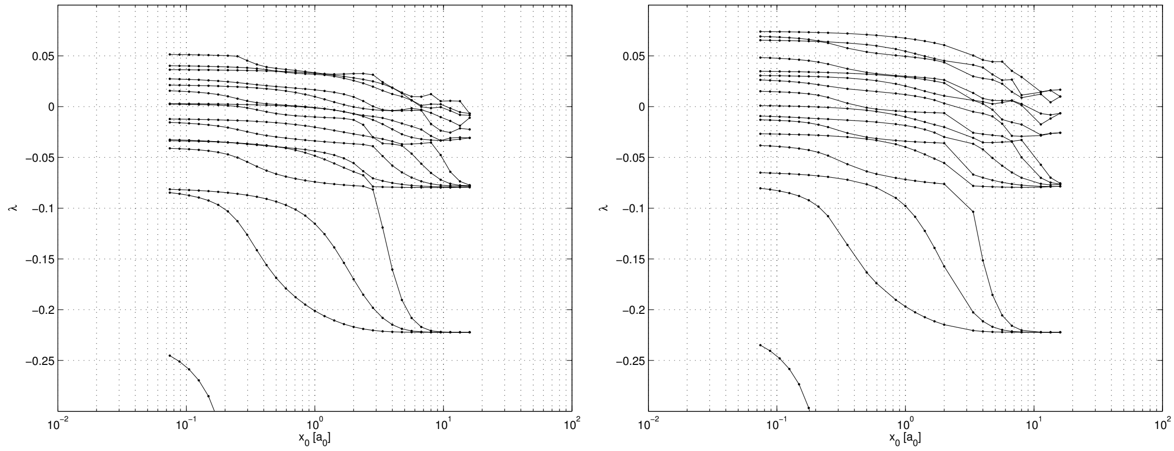


Figure 6.2: Smallest 16 eigenvalues of the two dimensional Schrödinger equation atom confined into a square domain of side length $32a_0$ (left) and circular domain of diameter $32a_0$ (right). The position x_0 is varied from the centre to the side mid-point. Note that the value are connected by magnitude and not by their respective eigenfunction.

6.4 Interpretation of the Changes under Asymmetric Confinement

Only if the nucleus resides exactly in the centre of the domain or in the centre or corner the boundary, there exist two symmetry axes. By orthogonality of the eigenfunctions follows, that only in those cases degeneracy exist. Note that in d-dimensions the nucleus must be presumably shifted to different positions on $d-1$ coordinate axes to make the eigenvalues unique.

6.5 Comparison to Confinement in a Circular Domain

The computed spectra for circular domain showed no significant difference to those computed for square domains, see the figures 6.1 and 6.3.

- If the nucleus is positioned in the centre, then more eigenvalues are degenerated.
- There is no degeneracy if the nucleus is positioned at the boundary. In a circular domain only the levels $1, 4, 9, 16, \ldots$ are not degenerated, all other states are pairwise degenerate.
- If the atom is confined into a circular domain with diameter L_0 the eigenvalues are slightly stronger affected as if it were were confined into a squared domain with side length L_0.

6.6 Comparison to the Radial Schrödinger Equation

In this section the eigenvalues of the hydrogenic Schrödinger equation in two dimensions are compared to those of the radial Schrödinger equation under confinement, giving an indicating whether the radial Schrödinger equation is still valid in case of confinement. The eigenvalues of the unconfined hydrogen atom can, beside their multiplicity, be found by solving the radial Schrödinger equation. The radial Schrödinger equation is one of the three equations obtained by separating the variables of the three dimensional equation expressed in spherical coordinates, see [Kä12, p. L 13]. The trajection of the eigenvalues of the radial Schrödinger equation is depicted in figure 6.6. The values are listed in table 7.

1. The change of the eigenvalues is qualitatively similar, e.g. the values increase with decreasing domain size.

2. A quantitative comparison is difficult, as already the eigenvalues of the unconfined two dimensional hydrogen atom differ from those of the three dimensional atom and hence from those of the radial equation.

3. Each eigenvalue of the radial Schrödinger equation corresponds to several eigenvalues of the three dimensional Schrödinger equation. When the eigenvalues of the radial Schrödinger equation are used to approximate the three dimensional system, e.g. for computing the partition function, then the values should be used in an n^2 multiplicity. Note that the degeneracy is partially removed under confinement and therefore the eigenvalues of the radial Schrödinger equation are not anymore the exact eigenvalues of the three dimensional Schrödinger equation.

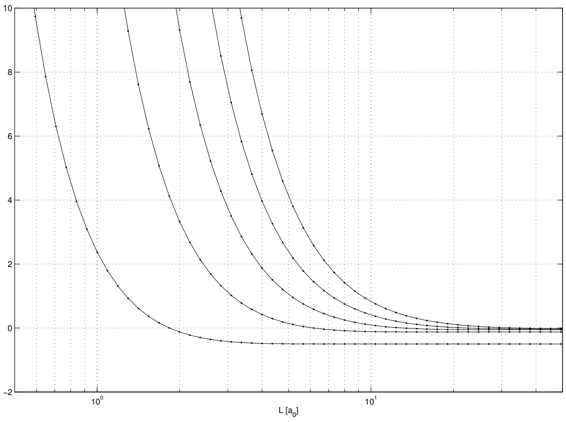


Figure 6.3: The smallest five eigenvalues of the radial Schrödinger equation under confinement. Note that each eigenvalue corresponds to several eigenvalues of the three dimensional Schrödinger equation, as specified in table 7.

Chapter 7

Conclusion

For this thesis a software package to approximate the energy levels and wave functions of the hydrogen atom was developed. The hydrogenic Schrödinger equation with the Coulomb potential was chosen as atom model. Different discretisation methods and convergence criteria with respect to the singular Coulomb potential were evaluated. An adaptive finite element framework to discretise partial differential equations in one, two and three dimensions was successfully implemented. Different solvers for large sparse eigensystems such as the implicitly restarted Arnoldi method, the Jacobi-Davidson method and the Lanczos method where analysed and applied to the discretisation matrices. As no readily available implementation of the Lanczos method without reorthogonalisation was available, a FORTRAN library was ported to a current Linux system and equipped with an MATLAB wrapper.

The implemented software was used to compute eigenspectra in of the two dimensional hydrogen atom for different confinement scenarios. The influence of several aspects of confinement were investigated, these are

- the size of the cavity,
- the position of the nucleus inside the cavity,
- shape of the cavity,
- differences between confinement in two and three dimensions as well as the one dimensional radial Schrödinger equation.

The current implementation can be used to compute arbitrary eigenpairs of the three dimensional Schrödinger equation under confinement. Thought, the mesh has to be adapted for each eigenvalue individually, to keep the discretisation matrices at a size the eigensolver can manage. This is because, the ArPack package in the shift and invert mode is used to compute the eigenpairs. This mode requires the complete factorisation of the mass and stiffness matrices. This works well in two dimension where several eigenvalues can be computed with sufficient accuracy in a reasonable time on a single work station. However, in three dimension the factorisation is more expensive even for matrices with a comparable size, due to the structure of the matrix. Therefore only few eigenvalues of the three dimensional Schrödinger equation can be approximated simultaneously at the moment.

Standard implementations of alternative eigensolvers like the Jacobi-Davidson method or Lanczos method do not perform better for the hydrogenic Schrödinger equation, albeit especially with the Lanczos method considerable improvements where observed for the Laplacian operator test case.

In the current implementation the application Lanczos method is limited to discretisations with diagonal mass matrices. It is conceivable, that the Lanczos method in combination with non diagonal mass matrices converges faster. To test this, it is firstly necessary to port or implement a proper Lanczos routine for generalised eigenvalue problems and secondly to extend the finite element implementation by a Multigrid solver to solve the mass matrix systems during the generalised Lanczos iterations efficiently.

Future research could also include relativistic effects by using the Dirac equation instead of the Schrödinger equation or the computation of energy levels of more complex atoms and molecules.

Chapter 8 Appendix

8.1 Tabulated Eigenvalues

L_0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0.707	+9.998e+00	+4.405e+01	+4.405e+01	+7.478e+01	+8.821e+01	+9.399e+01	+1.239e+02	+1.239e+02	+1.619e+02	+1.619e+02	+1.713e+02	+1.926e+02	+1.936e+02	+2.419e+02	+2.419e+02	+2.478e+02
1.000	+2.767e+00	+2.091e+01	+2.091e+01	+3.652e+01	+4.186e+01	+4.601e+01	+6.100e+01	+6.100e+01	+7.973e+01	+7.973e+01	+8.434e+01	+9.531e+01	+9.598e+01	+1.199e+02	+1.199e+02	+1.220e+02
1.414	-3.207e-01	+9.662e+00	+9.662e+00	+1.764e+01	+1.933e+01	+2.231e+01	+2.984e+01	+2.984e+01	+3.900e+01	+3.900e+01	+4.123e+01	+4.695e+01	+4.743e+01	+5.925e+01	+5.925e+01	+5.967e+01
2.000	-1.514e+00	+4.262e+00	+4.262e+00	+8.383e+00	+8.526e+00	+1.066e+01	+1.445e+01	+1.445e+01	+1.888e+01	+1.888e+01	+1.994e+01	+2.297e+01	+2.332e+01	+2.888e+01	+2.912e+01	+2.912e+01
2.828	-1.899e+00	+1.721e+00	+1.721e+00	+3.464e+00	+3.879e+00	+4.981e+00	+6.883e+00	+6.883e+00	+9.004e+00	+9.004e+00	+9.488e+00	+1.113e+01	+1.138e+01	+1.377e+01	+1.420e+01	+1.420e+01
4.000	-1.988e+00	+5.614e-01	+5.614e-01	+1.196e+00	+1.717e+00	+2.241e+00	+3.196e+00	+3.196e+00	+4.192e+00	+4.192e+00	+4.393e+00	+5.311e+00	+5.488e+00	+6.418e+00	+6.847e+00	+6.847e+00
5.657	-1.999e+00	+5.911e-02	+5.911e-02	+2.567e-01	+6.986e-01	+9.415e-01	+1.416e+00	+1.416e+00	+1.879e+00	+1.879e+00	+1.937e+00	+2.474e+00	+2.602e+00	+2.900e+00	+3.242e+00	+3.242e+00
8.000	-2.000e+00	-1.395e-01	-1.395e-01	-8.833e-02	+2.336e-01	+3.419e-01	+5.691e-01	+5.691e-01	+7.759e-01	+7.903e-01	+7.903e-01	+1.106e+00	+1.200e+00	+1.259e+00	+1.491e+00	+1.491e+00
11.314	-1.999e+00	-2.053e-01	-2.053e-01	-1.952e-01	+3.207e-02	+7.713e-02	+1.746e-01	+1.746e-01	+2.459e-01	+2.973e-01	+2.973e-01	+4.582e-01	+5.174e-01	+5.283e-01	+6.407e-01	+6.505e-01
16.000	-2.000e+00	-2.205e-01	-2.205e-01	-2.194e-01	-4.757e-02	-3.106e-02	+1.724e-03	+1.724e-03	+2.226e-02	+8.567e-02	+8.567e-02	+1.599e-01	+1.914e-01	+2.128e-01	+2.407e-01	+2.543e-01
22.627	-2.000e+00	-2.222e-01	-2.222e-01	-2.221e-01	-7.373e-02	-6.910e-02	-6.177e-02	-6.177e-02	-5.733e-02	+2.323e-04	+2.323e-04	+2.921e-02	+5.150e-02	+6.227e-02	+6.950e-02	+7.207e-02
32.000	-2.000e+00	-2.222e-01	-2.222e-01	-2.222e-01	-7.946e-02	-7.878e-02	-7.793e-02	-7.793e-02	-7.737e-02	-3.073e-02	-3.073e-02	-2.241e-02	-1.098e-02	-8.492e-03	-6.841e-03	-6.841e-03
45.255	-1.999e+00	-2.222e-01	-2.222e-01	-2.222e-01	-7.999e-02	-7.997e-02	-7.994e-02	-7.994e-02	-7.991e-02	-3.944e-02	-3.944e-02	-3.812e-02	-3.551e-02	-3.463e-02	-3.463e-02	-3.424e-02
64.000	-2.000e+00	-2.222e-01	-2.222e-01	-2.222e-01	-8.000e-02	-8.000e-02	-8.000e-02	-8.000e-02	-8.000e-02	-4.076e-02	-4.076e-02	-4.070e-02	-4.048e-02	-4.042e-02	-4.042e-02	-4.038e-02
90.510	-1.999e+00	-2.222e-01	-2.222e-01	-2.222e-01	-8.000e-02	-8.000e-02	-8.000e-02	-8.000e-02	-8.000e-02	-4.082e-02	-4.082e-02	-4.082e-02	-4.081e-02	-4.081e-02	-4.081e-02	-4.081e-02
128.000	-2.000e+00	-2.222e-01	-2.222e-01	-2.222e-01	-8.000e-02	-8.000e-02	-8.000e-02	-8.000e-02	-8.000e-02	-4.082e-02	-4.082e-02	-4.082e-02	-4.082e-02	-4.082e-02	-4.082e-02	-4.082e-02

Table 8.1: Smallest 16 eigenvalues of the two dimensional hydrogenic Schrödinger equation in the squared domain with side length $L_0 a_0$ and centred nucleus. The relative tolerance of 10^{-3} and absolute tolerance 10^{-4}.

x_0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0.088	-2.511e-01	-8.586e-02	-8.174e-02	-4.147e-02	-3.378e-02	-3.281e-02	-1.592e-02	-1.229e-02	+2.354e-03	+2.905e-03	+1.526e-02	+2.112e-02	+2.706e-02	+3.632e-02	+4.016e-02	+5.123e-02
0.125	-2.696e-01	-8.952e-02	-8.263e-02	-4.260e-02	-3.421e-02	-3.350e-02	-1.709e-02	-1.259e-02	+2.035e-03	+2.744e-03	+1.410e-02	+2.076e-02	+2.637e-02	+3.620e-02	+3.985e-02	+5.091e-02
0.177	-3.092e-01	-9.676e-02	-8.392e-02	-4.492e-02	-3.485e-02	-3.455e-02	-1.912e-02	-1.303e-02	+1.470e-03	+2.515e-03	+1.191e-02	+2.025e-02	+2.527e-02	+3.602e-02	+3.939e-02	+5.044e-02
0.250	-4.169e-01	-1.129e-01	-8.586e-02	-5.020e-02	-3.606e-02	-3.578e-02	-2.259e-02	-1.365e-02	+1.775e-04	+2.190e-03	+7.891e-03	+1.950e-02	+2.362e-02	+3.576e-02	+3.871e-02	+4.961e-02
0.354	-7.168e-01	-1.415e-01	-8.882e-02	-5.895e-02	-3.778e-02	-3.720e-02	-2.676e-02	-1.454e-02	-3.146e-03	+1.728e-03	+3.885e-03	+1.837e-02	+2.161e-02	+3.538e-02	+3.752e-02	+4.151e-02
0.500	-1.206e+00	-1.686e-01	-9.352e-02	-6.632e-02	-3.944e-02	-3.939e-02	-2.983e-02	-1.583e-02	-6.782e-03	+1.070e-03	+2.058e-03	+1.664e-02	+1.980e-02	+3.484e-02	+3.498e-02	+3.767e-02
0.707	-1.635e+00	-1.877e-01	-1.014e-01	-7.102e-02	-4.288e-02	-4.124e-02	-3.199e-02	-1.767e-02	-8.964e-03	+1.247e-04	+7.108e-04	+1.393e-02	+1.822e-02	+3.288e-02	+3.403e-02	+3.568e-02
1.000	-1.878e+00	-2.011e-01	-1.153e-01	-7.413e-02	-4.837e-02	-4.369e-02	-3.377e-02	-2.023e-02	-1.023e-02	-1.254e-03	-9.217e-04	+9.801e-03	+1.652e-02	+3.176e-02	+3.281e-02	+3.338e-02
1.414	-1.973e+00	-2.106e-01	-1.387e-01	-7.632e-02	-5.597e-02	-4.773e-02	-3.530e-02	-2.344e-02	-1.109e-02	-3.365e-03	-3.334e-03	+4.444e-03	+1.410e-02	+2.980e-02	+3.094e-02	+3.192e-02
2.000	-1.996e+00	-2.169e-01	-1.700e-01	-7.783e-02	-6.404e-02	-5.562e-02	-3.659e-02	-2.685e-02	-1.304e-02	-7.025e-03	-6.971e-03	-3.083e-04	+9.585e-03	+2.566e-02	+2.834e-02	+3.246e-02
2.828	-2.000e+00	-2.204e-01	-1.981e-01	-8.161e-02	-7.876e-02	-7.064e-02	-3.898e-02	-3.022e-02	-3.013e-02	-1.304e-02	-9.630e-03	-3.036e-03	+3.236e-04	+1.970e-02	+2.487e-02	+3.139e-02
4.000	-2.000e+00	-2.218e-01	-2.148e-01	-1.604e-01	-7.932e-02	-7.514e-02	-5.802e-02	-3.686e-02	-3.417e-02	-2.222e-02	-1.365e-02	-4.096e-03	-3.901e-03	+1.244e-02	+1.857e-02	+1.873e-02
5.657	-2.000e+00	-2.222e-01	-2.209e-01	-2.082e-01	-7.956e-02	-7.762e-02	-6.981e-02	-4.626e-02	-3.679e-02	-3.029e-02	-1.941e-02	-3.676e-03	-1.650e-03	+6.121e-03	+6.363e-03	+9.972e-03
8.000	-1.999e+00	-2.222e-01	-2.221e-01	-2.207e-01	-7.967e-02	-7.861e-02	-7.596e-02	-6.535e-02	-3.541e-02	-3.291e-02	-2.890e-02	-1.708e-02	-6.179e-03	-1.432e-03	+2.499e-03	+1.231e-02
11.314	-1.999e+00	-2.222e-01	-2.222e-01	-2.222e-01	-7.967e-02	-7.900e-02	-7.794e-02	-7.591e-02	-6.422e-02	-3.307e-02	-3.097e-02	-2.558e-02	-1.406e-02	-4.066e-03	-1.622e-03	+5.666e-03
16.000	-2.000e+00	-2.222e-01	-2.222e-01	-2.222e-01	-7.946e-02	-7.878e-02	-7.793e-02	-7.793e-02	-7.737e-02	-3.073e-02	-3.073e-02	-2.241e-02	-1.098e-02	-8.492e-03	-6.841e-03	-6.841e-03

Table 8.2: Smallest 16 eigenvalues of the two dimensional hydrogenic Schrödinger equation in the squared domain with side length $32a_0$ and nucleus position at x_0 Bohr radii away from the boundary. The relative tolerance of 10^{-3} and absolute tolerance 10^{-4}.

L_0	1	2	3	4	5
	1	2-5	6-14	15-30	31-55
0.707	+6.304e+00	+3.502e+01	+8.383e+01	+1.525e+02	+2.411e+02
1.000	+2.374e+00	+1.657e+01	+4.086e+01	+7.513e+01	+1.193e+02
1.414	+6.140e-01	+7.615e+00	+1.969e+01	+3.677e+01	+5.883e+01
2.000	-1.250e-01	+3.328e+00	+9.314e+00	+1.782e+01	+2.881e+01
2.828	-4.011e-01	+1.322e+00	+4.280e+00	+8.505e+00	+1.398e+01
4.000	-4.833e-01	+4.202e-01	+1.873e+00	+3.967e+00	+6.691e+00
5.657	-4.987e-01	+4.686e-02	+7.490e-01	+1.781e+00	+3.133e+00
8.000	-5.000e-01	-8.474e-02	+2.465e-01	+7.499e-01	+1.417e+00
11.314	-5.000e-01	-1.195e-01	+3.882e-02	+2.792e-01	+6.051e-01
16.000	-5.000e-01	-1.248e-01	-3.473e-02	+7.628e-02	+2.327e-01
22.627	-5.000e-01	-1.250e-01	-5.327e-02	-2.554e-03	+7.042e-02
32.000	-5.000e-01	-1.250e-01	-5.550e-02	-2.690e-02	+5.654e-03
45.255	-5.000e-01	-1.250e-01	-5.556e-02	-3.107e-02	-1.569e-02
64.000	-5.000e-01	-1.250e-01	-5.556e-02	-3.125e-02	-1.981e-02
90.510	-5.000e-01	-1.250e-01	-5.556e-02	-3.125e-02	-2.000e-02
128.000	-5.000e-01	-1.250e-01	-5.556e-02	-3.125e-02	-2.000e-02

Table 8.3: The smallest five eigenvalues of the radial Schrödinger equation under confinement. Each eigenvalue corresponds to several eigenvalues of the three dimensional Schrödinger equation. The correspondence is given in the second row.

8.2 Bibliography

[AER94] J. Ackermann, B. Erdmann, and R. Roitzsch. A Self-Adaptive Multilevel Finite Element Method for the Stationary Schrödinger Equation in Three Space Dimensions. *The Journal of Chemical Physics*, 101(9):7643–7650, 1994.

[Age12] European Space Agency. European Space Research and Technology Centre (ESA-ESTEC). `http://www.esa.int/esaMI/ESTEC/index.html`, 2012.

[All57] S.P. Alliluev. On the Relation Between "Accidental" Degeneracy and "Hidden" Symmetry of a System. *Soviet Journal of Experimental and Theoretical Physics*, 6:156, 1957.

[AR93] J. Ackermann and R. Roitzsch. A Two-Dimensional Multilevel Adaptive Finite Element Method for the Time-Independent Schrödinger Equation. *Chemical Physics Letters*, 214(1):109–117, 1993.

[Arn51] W.E. Arnoldi. The Principle of Minimized Iterations in the Solution of the Matrix Eigenvalue Problem. *Quarterly of Applied Mathematics*, 9:17–29, 1951.

[Ban12] R.E. Bank. PLTMG, A Software Package for Solving Elliptic Partial Differential Equations: Users' Guide 11.0, 2012.

[Bas81] G. Bastard. Hydrogenic Impurity States in a Quantum Well: A Simple Model. *Physical Review B*, 24(8):4714–4722, Oct 1981.

[Bay06] D. Baye. Lagrange-mesh method for quantum-mechanical problems. *physica status solidi (b)*, 243(5):1095–1109, 2006.

[BDJ$^+$00] Z. Bai, J. Demmel, Dongarra J., A. Ruhe, and H.A. van der Vorst. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide.* SIAM, Philadelphia, 2000.

[BDY87] R.E. Bank, T.F. Dupont, and H. Yserentant. The hierarchical basis multigrid method. *Numer. Math*, 52:427–458, 1987.

[BER95] R. Beck, B. Erdmann, and R. Roitzsch. KASKADE 3.0 - An Object-Oriented Adaptive Finite Element Code , 1995.

[BH86] D. Baye and P.-H. Heenen. Generalised Meshes for Quantum Mechanical Problems. *J. Phys. A: Math. Gen.*, 19:2041–2059, 1986.

[BS08] D. Baye and K.D. Sen. Confined hydrogen atom by the lagrange-mesh method: Energies, mean radii, and dynamic polarizabilities. *Phys. Rev. E*, 78(2):026701, Aug 2008.

[BSW83] R.E. Bank, A.H. Sherman, and A. Weiser. Refinement Algorithms and Data Structures for Regular Local Mesh Refinement. *Scientific Computing (Applications of Mathematics and Computing to the Physical Sciences)*, pages 3–17, 1983.

[Car04] C. Carstensen. Adaptive Mesh Refining Algorithm allowing for an H^1-Stable L^2-Projection onto Courant Finite Element Spaces. *Constructive Approximation*, 20(4):549–564, 2004.

[CJJT01] G. Cohen, P. Joly, Roberts J.E., and N. Tordjman. Higher Order Triangular Finite Elements with Mass Lumping for the Wave Equation. *SIAM Journal on Numerical Analysis*, 38(6):2047–2078, 2001.

[Com12] European Commission. Erasmus Mundus. `http://ec.europa.eu/education/external-relation-programmes/doc72_en.htm`, 2012.

[CW] J.K. Cullum and R.A. Willoughby. Lanczos algorithms for large symmetric eigenvalue computations, volume 2 - programs. `http://www.netlib.org/lanczos/`.

[CW81] J.K. Cullum and R.A. Willoughby. Computing Eigenvalues of Very Large Symmetric Matrices - An Implementation of a Lanczos Algorithm with no Reorthogonalization. *Journal of Computational Physics*, 44((2)):329–358, 1981.

[CW02] J.K. Cullum and R.A. Wiloughby. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations Volume 1: Theory (Classics in Applied Mathematics)*. SIAM: Society for Industrial and Applied Mathematics, 1st edition, 2002.

[Dir28] P.A.M. Dirac. The Quantum Theory of the Electron. *Proc. R. Soc. Lond. A 1*, 117(778):610–624, February 1928.

[DSBE12] TU Delft, KTH Stockhoml, TU Berlin, and FAU Erlangen. COSSE - Computer Simulations for Science and Engineering. `http://www.kth.se/en/studies/programmes/master/em/cosse`, 2012.

[Dur02] Durbin, P.A. and Iaccarino, G. Approach to Local Refinement of Structured Grids. *Journal of Computational Physics*, 181:639–653, 2002.

[EJ88] K. Eriksson and C. Johnson. An adaptive finite element method for linear elliptic problems. *Mathematics of Computation*, 50(182):361–383, 1988.

[FRRT78] M. Friedman, Y. Rosenfeld, A. Rabinovitch, and R. Thieberger. Finite Element Method for Solving the Two-Dimensional Schrödinger Equation. *Journal of Computational Physics*, 26(2):169–180, 1978.

[FvGM12] D.R. Fokkema and van Gijzen M.B. JDQZ FORTRAN Package, 2012.

[GA06] B. Grossman and Dadone A. Ghost-Cell Method with Far-Field Coarsening and Mesh Adaption for Cartesian Grids. *Computers and Fluids*, 35:676–687, 2006.

[Gar98] J. Garcke. Berechnung von Eigenwerten der Stationären Schrödingergleichung mit der Kombinationstechnik, 1998.

[Geu02] R. Geus. Jacobi-Davidson Algorithm for Solving Large Sparse Symmetric Eigenvalue Problems with Application to the Design of Accelerator Cavities, 2002.

[GG00] J. Garcke and M. Griebel. On the computation of the eigenproblems of hydrogen helium in strong magnetic and electric fields with the sparse grid combination technique. *J. Comput. Phys.*, 165(2):694–716, December 2000.

[GH06] M. Griebel and J. Hamaekers. A Wavelet Based Sparse Grid Method for the Electronic Schrödinger Equation. *Proceedings oh the International Congress of Mathematicians*, 3:1473–1506, 2006.

[GH07] M. Griebel and J. Hamaekers. Sparse Grids for the Schrödinger Equation. *ESAIM: Mathematical Modelling and Numerical Analysis*, 41:215–247, 2007.

[GYC$^+$91] S.H. Guo, X.L. Yang, F.T. Chan, K.W. Wong, and W.Y. Ching. Analytic Solution of a Two-Dimensional Hydrogen Atom. II. Relativistic Theory. *Phys. Rev. A*, 43(3):1197–1205, Feb 1991.

[Jin84] Yu Jinyun. Symmetric Gaussian Quadrature Formulae for Tetrahedronal Regions. *Computer Methods in Applied Mechanics and Engineering*, 43:349–353, May 1984.

[Kea86] P Keast. Moderate-degree tetrahedral quadrature formulas. *Comput. Methods Appl. Mech. Eng.*, 55(3):339–367, May 1986.

[Kä12] K. Kästner. Computation of Energy Levels of the Confined Hydrogen Atom - Literature Study, 2012.

[Lan50] C. Lanczos. An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators. *J. Res. Nat. Bureau Standards*, 45(4):255–282, 1950.

[LK55] J.M. Luttinger and W. Kohn. Motion of Electrons and Holes in Perturbed Periodic Fields. *Phys. Rev.*, 97(4):869–883, Feb 1955.

[LMW12] A. Logg, K-A. Mardal, and G Wells. *Automated Solution of Differential Equations by the Finite Element Method: The FEniCS Book*. Springer, 2012.

[LSY98] R.B. Lehoucq, D.C. Sorensen, and C. Yang. *ARPACK User's Guide: Solution of Large-Scale Eigenvalue Problems with Implicity Restarted Arnoldi Methods.* SIAM, 1998.

[MIKG12] P. Motamarri, M. Iyer, J. Knap, and V. Gavini. Higher-Order Adaptive Finite-Element Methods for Orbital-Free Density Functional Theory, 2012.

[MoSN] The Mathworks, National Institute of Standards, and Technology (NIST). JAMA : A Java Matrix Package. `http://math.nist.gov/javanumerics/jama/`.

[Pat84] A.T. Patera. A Spectral Element Method for Fluid Dynamics: Laminar Flow in a Channel Expansion. *Journal of Computational Physics*, 54:468–488, 1984.

[PHA81] J. Paine, F. Hoog, and R. Anderssen. On the Correction of Finite Difference Eigenvalue Approximations for Sturm-Liouville Problems. *Computing*, 26(2):123–139, 1981.

[Saa87] Y. Saad. Lanczos Method for Solving Symmetric Linear Systems with Several Right-Hand Sides. *Mathematics of Computation*, 48(178), 1987.

[SBFvdV96] G.L.G Sleijpen, A.G.L. Booten, D.R. Fokkema, and H.A. van der Vorst. Jacobi-Davidson Type Methods for Generalized Eigenproblems and Polynomial Eigenproblems. *BIT*, 36(3):595–633, 1996.

[Sch26] E. Schrödinger. Quantisierung als Eigenwertproblem (Erste Mitteilung). *Annalen der Physik*, 79:361–376, 1926.

[SF88] G. Strang and G. Fix. *An Analysis of the Finite Element Method, Second Edition.* Wellesley-Cambridge Press, 1973 [1988].

[Sim84] H.D. Simon. Analysis of the Symmetric Lanczos Algorithm with Reorthogonalization Methods. *Linear Algebra and Its Applications*, 61((C)):101–131, 1984.

[Slea] G.L.G. Sleijpen. jdqr. `http://www.staff.science.uu.nl/~sleij101/JD%5Fsoftware/JDQR.html`.

[Sor92] D.C. Sorensen. Implicit Application of Polynomial Filters in a k-Step Arnoldi Method. *SIAM J. Matrix Anal. Appl.*, pages 357–385, 1992.

[ST03] M. Sun and K. Takayama. Error Localization in Solution-adaptive Grid Methods. *Journal of Computational Physics*, 190:346–350, 2003.

[Sta12] S Stamatiadis. Solving the Molecular Schrödinger Equation in Cartesian Coordinates, 2012.

[SvdV00] G.L.G. Sleijpen and H.A. van der Vorst. A Jacobi-Davidson Iteration Method for Linear Eigenvalue Problems. *SIAM J. Matrix Anal. Appl*, 17:401–425, 2000.

[SvdVB99] G.L.G. Sleijpen, H.A. van der Vorst, and Z. Bai. Jacobi-Davidson Algorithms for various Eigenproblems: A working Document, 1999.

[vD11] L. van Dommelen. Quantum Mechanics for Engineers, 2011.

[YGC$^+$91] X.L. Yang, S.H. Guo, F.T. Chan, K.W. Wong, and W.Y. Ching. Analytic Solution of a Two-Dimensional Hydrogen Atom. I. Nonrelativistic Theory. *Phys. Rev. A*, 43(3):1186–1196, Feb 1991.

[Yse05] H. Yserentant. Sparse Grid Spaces for the Numerical Solution of the Electronic Schrödinger Equation. *Numer. Math.*, 101(2):381–389, August 2005.

Part II

Literature Study

Computing the Spectrum of the Confined Hydrogen Atom

Literature Study in Preparation for the Master Thesis

by

Karl Kästner
Delft University of Technology

Supervisors

Dr. Ir. Martin van Gijzen (TU Delft)
Dr. Domenico Giordano (ESA-ESTEC)

August 17, 2012

Chapter 1

Table of Contents

Introduction

The hydrogen atom is the most simplistic atom. Its typical isotope consists just of a single proton nucleus and a single electron shell. Hydrogen is not only the most abundant element in the universe, but all other elements are literally generated by nuclear fusion processes starting with hydrogen. Many other elements have therefore similar electronic properties and hydrogenic atom models can be used to describe them. Those models find amongst others applications in chemistry, solid state physics and plasma physics.

Each element has a distinct set of energy levels which determine its behaviour in chemical reactions. The energy levels also correspond to emission lines in the spectrum of an element. The spectrum of hydrogen is partially visible and can be probed by making the gas glow in a discharge lamp and splitting the emitted light with a prism into its spectral components.

The energy levels of hydrogen are given by the eigenvalues of the Schrödinger equation. For the unconfined hydrogen atom those eigenvalues can be computed analytically. A general formula to compute those eigenvalues was even found in the advent of quantum mechanics. However, for the arbitrarily confined atom, if at all, only analytic approximations exist.

This literature study is about the numerical approximation of the eigenvalues of a cubically confined hydrogen atom. It consists of three parts. In part one a short introduction into the physical background is given. Part two describes numerical methods used for discretising the differential equation and for computing its eigenvalues. The last part is an evaluation of the aforementioned methods with regard to their applicability to the confined hydrogen atom. Research questions are stated and a road map for the forthcoming master project is laid out.

This work was commissioned by the Space Research and Technology Centre of the European Space Agency (ESA-ESTEC) [Age12] and was facilitated by the Commission of the European Union under the Erasmus Mundus COSSE programme [Com12], [DSBE12].

Literature Overview

In the history of quantum physics the confined hydrogen atom was the topic of many publications. The first publications were by Michels and de Boer in 1937 [MdBB37] and Sommerfeld and Welker in 1938 [SW38] and focused on the determination of the ground state. Higher states and the splitting of degenerated energy levels was then described by Groot and Seldam in 1946 [dGtSC46]. In 1976 Ley-Koy and Runbinstein first considered the hydrogen atom confined in penetrable boundaries [LKR79]. These works considered only the cylindrical confined hydrogen atom. It followed solutions in other coordinate systems and domains, parabolic by Krähmer in 1998 [KSY98], cylindrical by Yurenyey in 2008 [YSP08] and spheroidal by Cruz in 2009 [CCR09]. A cubic cavity with the atom located at off-centred positions was first discussed by Kretov in 2009, [KAV09]. Most authors used analytic methods based on perturbation theory and basis functions.

The Hydrogen atom was also topic of numerical computations after Schroeder's original analytic solution. Application of finite difference methods reach furthest back, see Bolton 1956 [BSR56] and Kimball 1934 [KS34]. The finite element method was used by Levin and Shertzer in 1985 [LS85]. A very important development due to their high accuracy are the Lagrangian-mesh methods introduced by Baye and Heenen in 1986 [BH86]. Less common techniques like shooting methods were also applied (Killingbeck 1987 [Kil77]). The most recent development is the use of Sparse Grid methods, (Garcke 2000 [Gar98]). Excluding the sparse grid methods, most methods were only applied to the the radial Schrödinger equation.

So, although there are many publications about the hydrogenic Schrödinger equation, few actually tangent the numerical solution in Cartesian coordinates. Note that many time dependent solutions were also published, which are, however, not reviewed here as it is of no importance for this thesis.

Chapter 2

Physical Background

The purpose of this thesis is to compute energy levels of the confined hydrogen atom. The energy levels are required for further computations in plasma physics, as described in section 2.2. The hydrogen atom is modelled by the Schrödinger equation with Coulomb potential, the hydrogenic Schrödinger equation. The numerical approximation of the solutions of the hydrogenic Schrödinger equation is the central part of this thesis.

The hydrogenic Schrödinger equation itself is stated in section 2.3. The relation between computed values and physical quantities is derived in section 2.4 . The model error of the hydrogenic Schrödinger equation is discussed in 2.6.

2.1 Overview of Quantities Referred to throughout the Thesis

If not otherwise stated the symbols used in this thesis have the meaning as given in table 2.1. For simplicity vectors are not highlighted in bold face or indicated by arrows. The symbols given in table 2.1 and used throughout this chapter are to be interpreted as quantities commonly used in physics, e.g. h is Planck's constant, λ a wave length and E an energy levels. In later chapters symbols are to be interpreted as quantities commonly used in mathematics, e.g. λ refers to an eigenvalue and h to the step size.

2.2 Motivation: The Divergence of the Partition Function for the Unconfined Hydrogen Atom

Most of everyday scientific phenomena and engineering problems can be accurately described and solved using classical physics. However, some particular fields of physics cannot be thoroughly understood solely with classical methods, but require frequently the consideration of quantum mechanics. Such fields are thermodynamics and solid state physics.

The partition function is used to describe essential properties of gases and plasmas such as free energy, entropy and enthalpy (equation (2.1)). It forms a connection between quantum mechanics and its macroscopic effects,

$$Z = \sum g_i e^{-\beta E_i}, i = 1, 2, 3, ... \tag{2.1}$$

The value Z of the partition function is determined by the value of the quantum states, e.g. energy levels E, the Boltzmann factor β, and the degeneration factor g.

As the eigenvalues are raised to their exponential the partition function either converges with very few terms to a finite value or approaches infinity.

So does the value of the partition function not converge to a finite value in case of the unconfined hydrogen atom. As a finite value is essential to utilise the partition function in further calculations, the hydrogen model has to be adapted.

Quantity	Symbol	Dimension	
Elementary Quantities			
mass	m	kilogramme, kg	scalar
time	t	second, s	scalar
length	x	meter, m	vector
Current	I	Ampere, A	scalar
Voltage	U	Volt, V	scalar
frequency	f	Hertz, one per second, $1/s$	scalar
Derived Quantities			
Energy	E	$kg \cdot m^2/s^2$	scalar
momentum	$p = mv$	m^2/s	scalar
wave length	$\lambda = c/f$	$1/m$	scalar
velocity	$\dot{x} = \frac{\partial x}{\partial t} = v$	m/s	vector
wave number	k	$1/m$	scalar
angular frequency	$\omega = 2\pi/f$	$1/s$	scalar
radius	$r = \sqrt{\sum_{i=1}^{3} x_i}$	m	scalar
Operators			
spatial derivative in one dimension	$\phi' = \frac{d\phi}{dx}$		
Laplacian operator	$\Delta = \sum_{i=1}^{3} \frac{\partial^2}{\partial^2 x_i}$		
Euclidean vector norm	$\|\|x\|\|_2 = \sqrt{x^T x}$		
Constants			
elementary charge	e	$1.602 \cdot 10^{-19} As$	scalar
Planck's constant	h	$6.602 \cdot 10^{-31} As/V$	scalar
reduced Planck's constant	$\hbar = h/2\pi$	$1/2\pi \cdot [h]$	scalar
Pi	π	3.142	scalar
velocity of light in vacuum	c	$2.999 \cdot 10^{8} m/s$	scalar
imaginary unit	$i = \sqrt{-1}$	1	scalar
nanometre	nm	$1nm = 10^{-9}m$	scalar
picometre	pm	$1pm = 10^{-12}m$	scalar
Others			
computational domain	Ω		
boundary of the computational domain	Γ		

Table 2.1: Overview of Physical Quantities and Constants up to four digits of accuracy

A possible solution is to assume that the hydrogen atom is confined, e.g. that the empty space around the atom is limited by other atoms in a finite distance.

2.3 Formulation of the Boundary Value Problem

The boundary value problem (2.2) is referred to as the hydrogenic Schrödinger equation. It corresponds to the time independent non-dimensional [1] Schrödinger equation (2.29), if one ignores the reduced mass correction (2.19). The eigenvalue $\mathcal{E}$ and eigenfunction $\psi(\chi)$ are its solutions. Both are real valued. The eigenfunctions are furthermore square integrable. The solutions are restricted to the computational domain Ω and depend on location of the nucleus χ_0 within it,

$$\mathcal{E}\psi(\chi) = -\left(\frac{1}{2}\Delta + \frac{1}{\rho(\chi)}\right)\psi(\chi) \tag{2.2}$$

$$\rho = ||\chi - \chi_0|| \tag{2.3}$$

$$\chi \in \Omega. \tag{2.4}$$

$$\mathcal{E} = \mathbb{R} \tag{2.5}$$

$$\psi(\chi) \in \mathcal{L}^2. \tag{2.6}$$

In case of no confinement the domain Ω are the complete three natural space dimensions $\Omega = \mathbb{R}^3$ and in case of confinement it is an connected subspace of it. In particular the confinement to cuboids $\Omega \subset \mathbb{R}^3$, rectangles $\Omega \subset \mathbb{R}^2$ and closed lines $\Omega \subset \mathbb{R}^1$ is considered.

As the boundary Γ of the domain is assumed to consist of walls of infinite potential height, such that the electron cannot penetrate it. Therefore the eigenfunction $\psi(\chi)$ vanishes at the boundary. This corresponds to homogeneous Dirichlet boundary conditions (equation (2.7)),

$$\psi(\chi) = 0 \text{ on } \Gamma. \tag{2.7}$$

The eigenfunctions of $\psi(\chi)$ are square integrable and remain bounded at the location of the nucleus χ_0. Therefore no additional boundary condition is required for the point χ_0.

Note that some authors define different scaling constants and come up with the equation $-(\Delta + \frac{2}{\rho})\psi(\chi) = \tilde{\mathcal{E}}\psi(\chi)$ as hydrogenic Schrödinger equation. The eigenvalues $\tilde{\mathcal{E}} = 2\mathcal{E}$ yield of course the same energy levels E upon back-transformation into physical quantities.

2.4 Derivation of the Boundary Value Problem

The hydrogen atom can be modelled by the Schrödinger equation with a Coulomb potential. The energy levels are found as the eigenvalues of this equation. The general Schrödinger equation is time dependent (TDSE), but as the energy levels do not change as time elapses, it is beneficial to bring the equation into a time independent form. This form is known as the time independent Schrödinger equation (TISE), a second order elliptic partial differential equation which is derived in this section.

The behaviour of the hydrogen atom is described by the time dependent Schrödinger equation (2.8) [Sch26],

$$i\hbar \frac{\partial}{\partial t}\Psi(t,x) = \hat{H}\Psi(t,x). \tag{2.8}$$

Where the wave function $\Psi(x,t)$ determines the probability distribution of the electron's current location and the Hamiltonian operator $\hat{H}$ denotes the total energy of the system, e.g. the sum of the potential and the kinetic energy,

$$\hat{H} = E_{kin} + E_{pot}. \tag{2.9}$$

As the Schrödinger equation is not relativistic, space and and time can be separated from each other,

$$\Psi(t,x) = \phi(t)\psi(x) \tag{2.10}$$

$$i\hbar \frac{1}{\phi(t)}\frac{\partial\phi(t)}{\partial t} = \frac{1}{\psi(x)}\hat{H}\psi(x). \tag{2.11}$$

Assuming furthermore conservation of energy, e.g. that the Hamiltonian does not change over time, both sides of the equation remain constant. This constant is the energy level E,

$$E = \frac{1}{\psi(x)}\hat{H}\psi(x). \tag{2.12}$$

For the potential energy the notation $E_{pot} = V(r)$ with radius $r = ||x||$ is further on used,

$$E\psi(x) = (E_{kin} + V(r))\psi(x). \tag{2.13}$$

For an external observer both the proton and the electron encircle the centre of mass and hence contribute together to the energy contained in the system. Therefore the kinetic energy is given by formula (2.20),

$$E_{kin} = \frac{1}{2}m_p\dot{x}_p^2 + \frac{1}{2}m_e\dot{x}_e^2. \tag{2.14}$$

To determine the kinetic energy it is sufficient to consider only the distance x between the electron and proton instead of the absolute positions x_p and x_e,

$$x = x_p - x_e \tag{2.15}$$

$$\tag{2.16}$$

[1] The term dimension is used throughout this thesis in different contexts. Firstly the physical dimension refers to the unit of the metric system, amongst others meter, second, kilogramme. Secondly the spatial dimension refers to the number of unit vectors spanning the *space*. Thirdly the dimension referring to the length of a vector, corresponding to a function evaluated at a finite number of points. Continuous functions which are defined at an infinite number of points are referred to as infinite dimensional.

If one also chooses the centre of mass x_c as origin of the coordinate system,

$$x_c(m_p + m_e) = m_p x_p + m_e x_e \tag{2.17}$$
$$0 = m_p x_p + m_e x_e, \tag{2.18}$$

then the expression of the kinetic energy simplifies to one term

$$\begin{aligned} E_{kin} &= \frac{1}{2} m_p \left(\frac{m_e \dot{x}}{m_p + m_e} \right)^2 + \frac{1}{2} m_e \left(\frac{m_p \dot{x}}{m_p + m_e} \right)^2 \\ &= \frac{1}{2} \frac{m_p m_e}{m_p + m_e} \dot{x}^2. \end{aligned}$$

Where the constant m_r is called the reduced mass,

$$m_r = \frac{m_p m_e}{m_p + m_e}, \tag{2.19}$$

a common quantity in the solution of two body systems. Note that $m_r \approx m_e$ as the mass of the electron is much smaller as the mass of the proton,

$$E_{kin} = \frac{1}{2} m_r \dot{x}^2 = \frac{1}{2} m_r v^2. \tag{2.20}$$

Substitution of equation (2.20) into equation (2.13) yields

$$\begin{aligned} E\psi(x) &= \left(\frac{m_r v^2}{2} + V(r) \right) \psi(x) \\ &= \left(\frac{p^2}{2m_r} + V(r) \right) \psi(x). \end{aligned} \tag{2.21}$$

In quantum physics the momentum p can be interpreted as an operator acting on the wave-function [vN31],

$$p = -i\hbar\nabla. \tag{2.22}$$

The squared momentum can subsequently be replaced by the scaled Laplacian operator $\Delta = \sum \partial^2/\partial x_i^2$,

$$E\psi(x) = \left(\frac{(-i\hbar\nabla)^2}{2m_r} + V(r) \right) \psi(x) \tag{2.23}$$
$$= \left(\frac{-\hbar^2}{2m_r} \Delta + V(r) \right) \psi(x). \tag{2.24}$$

The potential energy $E_{pot} = V$ is given by the Coulomb potential,

$$i\hbar \frac{\partial}{\partial t} \psi(x) = \left(\frac{-\hbar^2}{2m_r} \Delta + \frac{-e^2}{r 4\pi\epsilon_0} \right) \psi(x). \tag{2.25}$$

To ease further computations the equation is transformed into a non-dimensional form. The Hartree energy E_H and the Bohr radius a_0 are introduced. And the variables of energy and space are then replace by their respective non-dimensional forms $E = \mathcal{E} E_H$, $x = a_0 \chi, r = a_0 \rho$, $\Delta_x = \frac{1}{a_0^2} \Delta_\chi$,

$$a_0 = \frac{4\pi\epsilon_0 \hbar^2}{m_e e^2} \tag{2.26}$$

$$E_H = \frac{m_e e^4}{(4\pi\epsilon_0\hbar)^2} = \frac{1}{a_0^2} \frac{h^2}{m_e^2} \tag{2.27}$$

$$\mathcal{E} E_H \psi(\chi) = \left(\frac{1}{a_0^2} \frac{-\hbar^2}{2m_r} - \frac{e^2}{\rho a_0 4\pi\epsilon_0} \right) \psi(\chi) \tag{2.28}$$

$$\mathcal{E} \frac{1}{a_0^2} h^2 m_e \psi(\chi) = \left(-\frac{1}{2} \frac{m_e}{m_r} \frac{1}{a_0^2} \frac{\hbar^2}{m_e} - \frac{\hbar^2}{\rho a_0^2 m_e} \right) \psi(\chi)$$

$$-\mathcal{E} \psi(\chi) = \left(\frac{1}{2} \frac{m_e}{m_r} \Delta + \frac{1}{\rho} \right) \psi(\chi). \tag{2.29}$$

This represents the boundary value problem as introduced in section 2.3.

2.5 Singular Potentials

Depending on the author the term *singular potential* is more or less strictly interpreted and used for functions which are singular at one point and finite elsewhere. Following three different definitions exist [And76].

1. The first class includes potentials who's integral remains finite,

$$\lim_{\epsilon \to 0} \int_{\epsilon}^{a} V(x)dx < \infty. \tag{2.30}$$

2. The second class contains potentials which do not satisfy the requirements for class one, but who's singularity tends slower to infinity than x tends to zero. This applies to the Coulomb potential and hence to the hydrogenic Schrödinger equation. The solutions to the Schrödinger equation with such a potential are still bound,

$$\lim_{\epsilon \to 0} \int_{\epsilon}^{a} xV(x)dx < \infty. \tag{2.31}$$

3. The third class of potentials are these who's singularities tend faster to infinity than x tends to zero. Some authors only regard these potentials as singular, as here the solutions to the Schrödinger equation themselves are unbound,

$$\lim_{\epsilon \to 0} \int_{\epsilon}^{a} xV(x)dx = \infty. \tag{2.32}$$

2.6 Physical Limitations of the Mathematical Model

The Schrödinger equation is a simplified model of the real hydrogen atom and neglects several physical effects. The true energy levels differ therefore slightly from those computed with the Schrödinger equation. However, for the unconfined hydrogen atom this error [2] is lower than 0.01%. Hence even if the eigenvalues are found exactly or are approximated with arbitrary precision, their physical significance does not exceed a few digits. The deviation of the energy levels are referred to as fine structure and is mainly due to

- the relativistic corrections to the kinetic energy,
- the spin-orbit interaction,
- the Darwin term
- and the Lamb shift.

Relativistic effects can be accounted for by using the Dirac equation instead of the Schrödinger equation. For a more detailed explanation of relativistic quantum mechanics, see [vD11].

2.7 General Solution Properties

The Schrödinger equation consists of a symmetric Hamiltonian. Therefore the eigenvalues are real. However, the eigenvalues are not necessarily unique depending on the symmetry of the system. The eigenvalues of the hydrogen atom correspond to the energy levels and are interpreted as follows. When a hydrogen ion catches and binds an electron a photon is emitted. The electron is bound to one of the energy levels and the wavelength of the photon, e.g. the amount of released energy is determined by the energy level.

The squared eigenfunctions correspond to the probability P to find an electron at a given point inside

[2] The term error is used throughout this thesis in several contexts. Firstly there is the model error, which is introduced by approximating a complex physical system with a simplified mathematical model. Secondly there is is the discretisation error, also called truncation error, which is introduced by discretising the continuous function with a finite set of unknowns. Thirdly there is the error introduced by the iterative solver for linear and eigensystems. Finally there is the cancellation error which occurs due to the finite machine precision when numbers of similar magnitude are subtracted. The different errors reduce the accuracy of the approximate solution and should not be confused with *bugs*, where a programme is not working as intended.

the domain. Therefore the eigenfunction is normed such that the area beneath the probability function is one,

$$\int_{\Omega} \Psi^*(x)\Psi(x)dx = 1. \tag{2.33}$$

For illustration the probability function of the unconfined hydrogen atom is usually integrated over the sphere,

$$P(r) = 4\pi \left(rR(r)\right)^* \left(rR(r)\right). \tag{2.34}$$

2.8 Solution of the Unconfined Hydrogen Atom

The Schrödinger atom of the unconfined hydrogen atom has an analytic solution. As the Coulomb potential is spherically symmetric, a transformation to spherical coordinates allows a separation of variables. The Schrödinger equation in spherical coordinates is given by equation (2.35),

$$\frac{1}{2}\left(\frac{1}{r^2}\frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}\Psi\right) + \frac{1}{r^2\sin\phi}\frac{\partial}{\partial\phi}\left(\sin\phi\frac{\partial}{\partial\phi}\Psi\right) + \frac{1}{r^2\sin^2\phi}\left(\frac{\partial^2}{\partial\theta^2}\Psi\right)\right) + \frac{1}{r}\Psi = -\mathcal{E}\Psi. \tag{2.35}$$

The separation of variables leads to one radial and two angular equations. It is sufficient to consider only the radial equation (2.37) for determining the energy levels,

$$\Psi = R(r)\Phi(\phi)\Theta(\theta) \tag{2.36}$$

$$\frac{1}{2}\frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}R\right) + rR = -\mathcal{E}r^2R. \tag{2.37}$$

To simplify the radial equation further, one does a substitution and solves for the radial probability instead for the radial wave-function,

$$y = rR \tag{2.38}$$

$$\frac{1}{2}\frac{\partial}{\partial r}\left(r^2\frac{\partial}{\partial r}\left(\frac{1}{r}y\right)\right) + y = -\mathcal{E}ry \tag{2.39}$$

$$\frac{1}{2}\frac{\partial}{\partial r}\left(-y + ry'\right) + y = -\mathcal{E}ry \tag{2.40}$$

$$\frac{1}{2}ry'' + y = -\mathcal{E}ry \tag{2.41}$$

$$\frac{1}{2}y'' + \frac{1}{r}y = -\mathcal{E}y. \tag{2.42}$$

With the boundary conditions as stated below. The additional boundary condition at zero is deduced from the assumed square integrability of the solution,

$$y(0) = 0 \tag{2.43}$$

$$\lim_{r\to\infty} y(r) = 0. \tag{2.44}$$

Equation (2.42) determines the eigenvalues of the unconfined hydrogen atom. It is an ordinary differential equation, but has the same structure of the unseparated Schrödinger equation (2.29). Its analytic solutions come in the form of Laguerre polynomials and spherical harmonics. The radial equation can easily be solved by the finite difference method, as the singularity is on the boundary and the derivatives are defined everywhere inside the domain.

For the ground state, the radial wave function is given by

$$R_1(r) = \frac{1}{\sqrt{\pi a_0^3}}e^{-r/a_0}. \tag{2.45}$$

It is also depicted in figure 2.1. The wave function of higher energy levels reach further, but asymptotically all energy levels decay exponentially towards infinity. As in case of the ground state, wave functions of higher energy levels may not be differentiable at the nucleus, depending on the corresponding spherical harmonic.

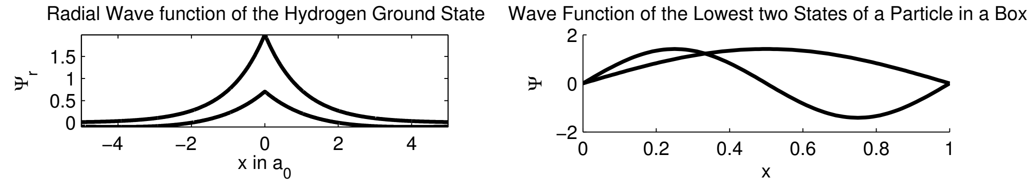


Figure 2.1: Wave functions of the unconfined hydrogen atom and of a particle in a box

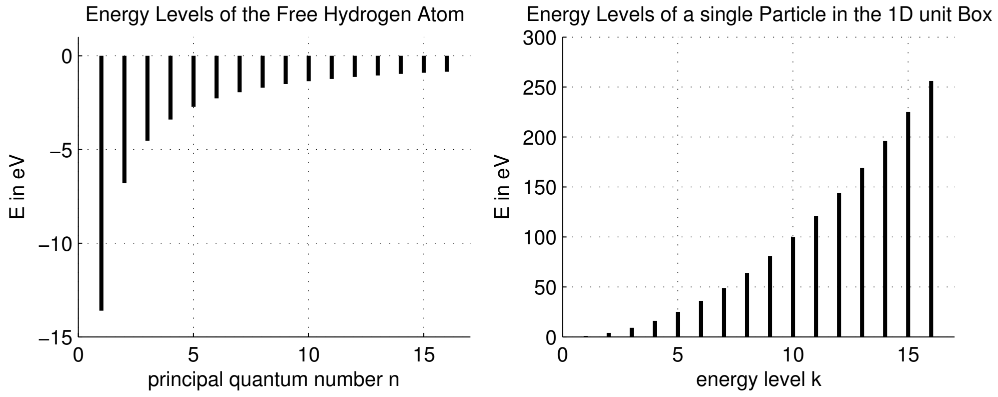


Figure 2.2: Energy levels of the unconfined hydrogen atom and of a particle in a box

2.9 Influence of the Confinement on the Solution

The cubically confined hydrogen atom is best understood if one considers it as the transition between two separate problems, namely that of the particle in the box and that of the unconfined hydrogen atom.

Unconfined Hydrogen Atom The unconfined hydrogen atom is the limit case of the confined hydrogen atom, if one considers the box of the confinement to be infinitely large. The energy levels of the free hydrogen atom follow a rational series given by equation (2.46) and are depicted in figure 2.2. Here Ry is the Rydberg constant and equals the energy of the hydrogen ground state,

$$E = -Ry\frac{1}{n^2}, n = 1, 2, 3, ... \tag{2.46}$$

$$Ry = hc\frac{m_e e^4}{8\epsilon_0^2 h^3 c} \approx 13.605eV.$$

The wave function takes the form of general Laguerre polynomials for the radial and spherical harmonics for the angular component. It decays exponentially with increasing distance from the nucleus (figure 2.1). A plot of the wave-function shows that the derivative at the location of the nucleus does not exist. This implies numerical complications discussed later.

Particle in a Box Here one considers only the electron confined in a box without a proton and the Schrödinger equation does not contain a potential. In this case the energy levels follow the series (2.47) which is also depicted in figure 2.2. The wave functions take the form of elementary sines,

$$E = \frac{n^2}{L^2}\frac{h^2}{8m_e}. \tag{2.47}$$

Implications for the Confined Hydrogen Atom The solution of the unconfined hydrogen atom and the particle in the box problem are obviously antagonistic. The energy levels of the unconfined hydrogen atom are negative and bound, as they converge to zero. The energy levels of the particle in the box are positive and approach infinity. The eigenvalue spectrum of a confined hydrogen atom is indefinite and resemble a combination of both spectra, with a limited number of negative eigenvalues being similar to those of the unconfined hydrogen atom and an infinite number of eigenvalues approaching asymptotically those of the particle in the box.

2.9.1 Position of the Nucleus Inside the Confining Box

The spherical confined hydrogen atom with centred nucleus can also be solved by separation of variables. However, the energy levels given as roots of the confluent hypergeometric function are not rational numbers anymore [SW38, p. 57], [SPM09, p. 127]. If the nucleus is shifted off-centre, its distance to the boundary varies depending on the direction. Thus the radial part cannot be separated from the angular parts. In this case it is simpler to consider the problem in Cartesian coordinates, which, however, do not allow a separation of variables.

2.9.2 Shape of the Confining Box

The behaviour of an confined hydrogen atom does not depend solely on the size of the box, but also on its form. For various reasons different shapes have been considered, spherically and parabolic shapes as the hydrogenic Schrödinger equation is separable in the respective coordinate systems, or cylindrical, triangular or rectangular shapes, as they best represent manufactured quantum structures.

In this thesis a rectangular domain is considered and further shapes are not investigated. Note that the qualitative results with different domains are expected to be similar.

2.9.3 Size of the Confining Box

The wave functions of the hydrogen atom decay exponentially. So does the radial wave function of the ground state R_1 (equation (2.45)) already falls below 10^{-15} parts of its maximum at a distance of about 35 Bohr radii. Therefore a box with a side length of only 70 Bohr radii becomes quasi infinitely large. The wave functions of the n-th energy level require moderately larger boxes, as they range in relation to the ground state n-times further.

Reversely one expects a completely different behaviour of the confined hydrogen atom compared to the unconfined atom once the box radius drops below one Bohr radius. For comparison: One Bohr radius (53pm) is the mean distance between the electron and the proton of the unconfined hydrogen atom and the the bond length of the hydrogen molecule is about two Bohr radii.

2.10 Confinement to Lower Dimensions

The spectrum of an confined hydrogen atom differs from that of an unconfined atom. So limits a confinement in all three dimensions the number of negative energy levels and makes the partition function converge. Of particular interest are the limit cases where the atom is confined some axes to an infinitesimal region whilst it remains unconfined in the remaining axes, viz. an atom constraint to a lower dimensional space. Such systems are described by hydrogen atoms constrained in their movement by strong magnetic or laser fields. Quasi lower dimensional solid state devices have also been manufactured. However, the behaviour of those devices is influenced by other adverse physical effects such as permeability of the material or penetrable boundaries. In these cases one usually does not consider atoms but electron-hole pairs, which are referred to as *excitons*.

2.10.1 Two Dimensions

If an atom resides in a quantum well, e.g. if its wave functions are constrained to two dimensions, its ground state energy is four times as large as that of the unconfined atom [Bry84]. Some authors replace the Coulomb potential by a logarithmic potential in the two dimensional Schrödinger equation. However, this is not considered here, as only the movement of the degrees of freedom of the atom, but not the forces are constrained to two dimensions.

2.10.2 One Dimension

The one dimensional hydrogen atom was extensively discussed by theorists with contradicting views. Loudon published the first comprehensive paper about this topic and claimed the existence of an infinite ground state and a double degeneracy of the remaining states, both conclusions are disputed until today [Lou59], [GC97]. In general the authors consent that the Balmer series of the three dimensional hydrogen atom is part of the spectrum of the one dimensional hydrogen atom as well.

One must keep in mind that truly one dimensional systems do not exist, but resemble limit cases where the atom is constrained to a long thin cylinder. Physical systems with this properties are quantum wires, nano-tubes or hydrogen atoms in strong laser field. Such systems indeed appear to have a ground state which is even much more amplified than that of the two dimensional system [Spa04].

No final conclusion about the one dimensional hydrogen atom shall be drawn here, as it only a side issue in this thesis and would require a much more extensive study of literature or even research to be resolved. However, the dissent about the findings of Loudon are of importance for this thesis as the findings are based on a weakened potential.

It should further be noted that the one proton nucleus itself is not infinitesimal small, but has a finite radius, called the *charge radius*. Albeit this radius is much smaller than the perturbation to achieve numerical stability it shows that any prediction about the wave function very close to the nucleus based on the Schrödinger equation are futile, as they are beyond the physical capability of the model, see also section 2.6.

2.10.3 Quasi Zero-Dimensions

A particle tightly constrained in all dimensions becomes a quantum dot. Here the ground state energy becomes also more and more amplified the smaller the confining box becomes. The high binding energy of an electron confined to a region close to the nucleus can be illustrated with an analogy to celestial mechanics, where by Kepler's third law, the velocity of planets orbiting close to the central star is larger compared to those orbiting further away.

Chapter 3

Discretisation of the Schrödinger Equation

As the the hydrogenic Schrödinger equation is not separable in Cartesian coordinates, only analytic approximations exist. A numerical approximation is therefore preferable. The numerical method should be chosen such that it is convergent and the result physically plausible.

The properties of the Hamiltonian resembling the hydrogenic Schrödinger equation are carried over to the discretised system. Hence the matrices will be real and symmetric, its eigenvalues and eigenvectors real and orthogonal. Finite difference and finite element discretisations furthermore yield sparse and usually banded discretisation methods. This gives rise to eigensolver for large symmetric systems as discussed in chapter 6.

3.1 Notation used throughout the Numerical Analysis Parts

If not otherwise declared the symbols used in the numerical analysis parts have the meaning explained in table 3.1

Symbol	Meaning
a capital letter, e.g. A	a matrix
a_i	the i-th column of a matrix A
a_{ij}	the j-th element of the i-th column of the matrix A
a minor Latin letter, e.g. x	a vector
x_i	the i-th element of the vector x
minor Greek letter α	a scalar
λ	exact eigenvalue
v	exact eigenvector
θ	eigenvalue of the projected system, approximated eigenvalue (Ritz value)
s	eigenvector of the projected system
z	approximated eigenvector (Ritz vector)
Q	an orthogonal matrix
T	a tridiagonal matrix
D	a diagonal matrix
$\|\alpha\| = \{\alpha \text{ if } \alpha \leq 0, -\alpha \text{ if } \alpha < 0\}$	absolute value
$\kappa(A) = \frac{max_{i=1..n}\|\lambda_i\|}{min_{j=1..n}\|\lambda_j\|}$	matrix condition number
$A^T : a_{ij} \leftarrow a_{ji}$	transpose, exchanges columns with rows
$A^* : a_{ij} \leftarrow a_{ji}^*$	conjugate transpose, changes columns with rows and negates imaginary parts

Table 3.1: Meaning of symbols frequently used in the numerical analysis parts

3.2 Finite Difference Discretisation

In this section the finite difference approximations and their respective discretisation errors are derived and applied to the Schrödinger equation (2.2). The finite difference method (FDM) is a numerical method to approximate the solution of differential equations. The approximation u_h is computed for a set of points x_i inside the subspace Ω_h of the domain Ω which are usually arranged in form of a structured grid. For homogeneous Dirichlet boundary conditions in one dimension there are m grid points inside the domain Ω and two on the boundary, where the solution is zero. There are therefore $m+1$ steps inside the domain Ω.

$$\Omega = [-L, L] \tag{3.1}$$

$$\Omega_h \subset \Omega \tag{3.2}$$

$$x_i \in \Omega_h, 0 \leq i \leq m+1 \tag{3.3}$$

At each grid point the derivatives are approximated by a linear equation of the point itself and its imminent neighbours. Here such an equation is called a difference kernel, as it is the same for each grid point. An approximation with m points inside the domain leads to m independent equations which are combined into a $m \times m$ matrix. The second derivative (Laplacian operator) with constant step size and homogeneous Dirichlet boundary conditions can be approximated in one dimension by the matrix

$$D_2^2 = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & & \\ 1 & -2 & ... & \\ & ... & ... & 1 \\ & & 1 & -2 \end{bmatrix}. \tag{3.4}$$

If the convergence requirements are met, the accuracy of the approximation increases with the number of grid points. An extensive listing of difference kernels is given in the appendix A.

The Schrödigner equation (2.2) discretisation consists of a matrix which is the sum of the Laplacian operator L and a diagonal matrix specifying the potential V,

$$A = -\frac{1}{2}D_2^2 + V. \tag{3.5}$$

The eigenvalues λ_k of this discretisation matrix are the approximated energy levels and the eigenvectors $\psi_k = u_k^h$ the approximated wave functions,

$$A\psi_k = \lambda_k \psi_k. \tag{3.6}$$

The matrix A is sparse, eq. it has only few non-zero entries. This allows to find even the eigenvalues of very large matrices with the algorithms introduced in chapter 6.

3.2.1 Definition of the Discretisation Error

The discretisation error is the difference between the exact solution u_* sampled at the grid points x_i and the approximate solution u_k at those points.

$$err_h(x_i) = u_*(x_i) - u_h(x_i) \tag{3.7}$$

An exact solution may be interpreted as an analytic solution or a converged approximated solution whos discretisation error is below machine precission. Goal of the discretisation is to find an accurate solution, e.g. to minimise the norm of the discretisation error.

3.2.2 Definition of the Order of Accuracy

An approximation based on a grid with constant step size is called n-th order accurate if the discretisation error decreases proportional to the n-th power of the step size $h = \frac{1}{m}$, if the convergence criteria are met. Similarly the the accuracy increases with the n-th power of the number of grid points m. This is abbreviated with the short hand notation $O(h^2)$.

$$err(x_i) = Ch^n u_*(x_i) = C\frac{1}{m^n}u_*(x_i) \tag{3.8}$$

So if a discretisation scheme is called second order accurate its error is quartered if the number of grid points is doubled.

3.2.3 Definition of the Derivative

In general the derivative $u'(x) = \frac{d}{dx}u(x)$ of a function $u(x)$ is given by equation (3.9) if the limits exists and are equal. The conditions for the derivative to exist are usually satisfied if the function is sufficiently smooth, e.g. if there are no peaks or high frequency oscillations,

$$u'(x)_+ = \lim_{h\to 0, h>0} \frac{u(x+h) - u(x)}{h} \tag{3.9}$$

$$u'(x)_- = \lim_{h\to 0, h>0} \frac{u(x) - u(x-h)}{h} \tag{3.10}$$

$$u'(x)_- = u'(x)_+. \tag{3.11}$$

The n-th derivative of the function can be determined by recursively forming the derivative,

$$\frac{d^n}{dx^n}u(x) = \frac{d^{n-1}}{dx^{n-1}}\left(\frac{d}{dx}u(x)\right). \tag{3.12}$$

3.2.4 Taylor's Theorem

Let the function $u(x)$ be at least k-times continuously differentiable in the neighbourhood of x_0, then $u(x)$ has a series expansion in the neighbourhood of x_0 in the form of equation (3.13),

$$u(x) = \lim_{x\to x_0} \sum_{i=0}^{k} \frac{1}{i!}(x - x_0)\left(\frac{d^k}{dx^k}u(x_0)\right). \tag{3.13}$$

3.2.5 Finite Difference Approximations on Uniform Grids

Basic finite difference schemes are set up on uniform grids. This means that the step size h, e.g. the distance between two adjacent points x_i is constant,

$$h = \frac{2L}{m+1} \tag{3.14}$$

$$x_i = i \cdot h, 0 \le i \le m+1. \tag{3.15}$$

Derivation of Finite Difference Schemes

Difference schemes can be derived with different approaches. In this subsection a derivation by Taylor expansion and by derivatives of polynomials is shown.

By Taylor expansion A finite difference approximation of the n-th derivative are derived by using the first $k \ge n$ terms of the Taylor series expansion (3.16). Note that according to the Taylor theorem this requires the first k derivatives of the function to exist,

$$u(x_0 + h) \approx \sum_{i=0}^{k} \frac{h^i}{i!}u^i(x_0). \tag{3.16}$$

The Taylor series expansion gives rise to a system of equations. For a second order accurate approximation the system becomes

$$\begin{bmatrix} -h & 1/2\,h^2 \\ h & 1/2\,h^2 \end{bmatrix} \begin{bmatrix} u_2'(x_i) \\ u_2''(x_i) \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} u_2(x_{i-1}) \\ u_2(x_i) \\ u_2(x_{i+1}) \end{bmatrix}}_{U}, \tag{3.17}$$

and the resulting convolution kernel of the second derivative is

$$u_2''(x) = \frac{1}{h^2}\begin{bmatrix} 1 & -2 & 1 \end{bmatrix} U + O(h^4). \tag{3.18}$$

By deriving polynomials Instead of using the Taylor expansion one can also derive approximations of the n-th derivative by constructing a polynomial from $k \geq n+1$ grid points and differentiating it n times,

$$\begin{bmatrix} 1 & x_{i-1} & x_{i-1}^2 \\ 1 & x_i & x_i^2 \\ 1 & x_{i+1} & x_{i+1}^2 \end{bmatrix} \begin{bmatrix} u_2(x_i) \\ u_2'(x_i) \\ u_2''(x_i) \end{bmatrix} = \begin{bmatrix} 1 & x_i & x_i^2 \\ 0 & 1 & 2x_i \\ 0 & 0 & 2 \end{bmatrix} \begin{bmatrix} u_2(x_{i-1}) \\ u_2(x_i) \\ u_2(x_{i+1}) \end{bmatrix}. \tag{3.19}$$

This leads to the same difference kernel as the Taylor expansion (3.18).

Discretisation Error of the Finite Difference Method

The finite difference approximation derived in the previous section is not exact, as the Taylor expansion is truncated after a finite number of steps. The resulting discretisation error can be approximated by expanding more Taylor terms,

$$\begin{bmatrix} -h & {}^1\!/_2\, h^2 \\ h & {}^1\!/_2\, h^2 \end{bmatrix} \begin{bmatrix} u_4'(x) \\ u_4''(x) \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & {}^1\!/_6\, h^3 & -{}^1\!/_{24}\, h^4 \\ 0 & -1 & 1 & -{}^1\!/_6\, h^3 & -{}^1\!/_{24}\, h^4 \end{bmatrix} \begin{bmatrix} u_4(x_{i-1}) \\ u_4(x_i) \\ u_4(x_{i+1}) \\ u_4'''(x_i) \\ u_4^{iv}(x_i) \end{bmatrix}. \tag{3.20}$$

The solution of this system yields

$$u_*''(x_i) = u_4''(x_i) + O(h^6) \tag{3.21}$$

$$= \frac{1}{h^2}[1, -2, 1]U - \frac{1}{12}h^2 u_4^{iv}(x_i) + O(h^6) \tag{3.22}$$

$$= u_2''(x_i) - \frac{1}{12}h^2 u_4^{iv}(x_i) + O(h^6). \tag{3.23}$$

So the approximation $u_2''(x)$ is second order accurate, as the discretisation error is proportional to h^2 and the fourth derivative of the solution. Note that this is just an estimation of the error and not an upper bound. The actual size of the error depends on how fast higher derivatives decay to zero.

In general the true solution u_* is not known. If the discretisation yields a linear system of equations, then the discretisation error can still be estimated with another linear system where the operator is the difference of two discretisation matrices of distinct order of accuracy.

$$err = u_2 - u_4 \tag{3.24}$$

$$= (A_2 - A_4)^{-1} f \tag{3.25}$$

Higher Order Finite Differences

As one desires a maximum of accuracy an extension to higher order accurate finite difference methods is logical. To derive an higher order accurated method one could simply derive a difference kernel with supported by more than three points.

By Taylor Expansion If more points are incorporated into the expansion and higher derivatives exist, then the order of accuracy increases. The system of a fourth order accurate approximation is given by

$$\begin{bmatrix} -2h & {}^4\!/_2\, h^2 & -{}^8\!/_6\, h^3 & {}^{16}\!/_{24}\, h^4 \\ -1h & {}^1\!/_2\, h^2 & -{}^1\!/_6\, h^3 & {}^1\!/_{24}\, h^4 \\ 1h & {}^1\!/_2\, h^2 & {}^1\!/_6\, h^3 & {}^1\!/_{24}\, h^4 \\ 2h & {}^4\!/_2\, h^2 & {}^8\!/_6\, h^3 & {}^{16}\!/_{24}\, h^4 \end{bmatrix} \begin{bmatrix} u_4'(x_i) \\ u_4''(x_i) \\ u_4'''(x_i) \\ u_4^{iv}(x_i) \end{bmatrix} = \begin{bmatrix} 1 & 0 & -1 & 0 & 0 \\ 0 & 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 & 0 \\ 0 & 0 & -1 & 0 & 1 \end{bmatrix} \underbrace{\begin{bmatrix} u_4(x_{i-2}) \\ u_4(x_{i-1}) \\ u_4(x_i) \\ u_4(x_{i+1}) \\ u_4(x_{i+2}) \end{bmatrix}}_{F}, \tag{3.26}$$

and the resulting convolution kernel of the second derivative is,

$$u_4''(x_i) = \frac{1}{12\,h^2} \begin{bmatrix} -1 & 16 & -30 & 16 & -1 \end{bmatrix} U + O(h^4) \tag{3.27}$$

$$u_4^{iv}(x_i) = \frac{1}{h^4} \begin{bmatrix} 1 & -4 & 6 & -4 & 1 \end{bmatrix} U + O(h^2). \tag{3.28}$$

A derivation by derivatives of polynomials yields again the same result.

The descretisation error of the finite difference solution to the analytic solution u_*^n can be estimated by comparing approximations consisting of differently many terms from the Taylor expansion,

$$\begin{aligned} u_2''(x_i) - u_*''(x_i) &= u_2''(x_i) - u_4''(x_i) + O(h^4) && (3.29)\\ &= \frac{1}{12h^2}\left[\begin{array}{ccccc} 1 & -4 & 6 & -4 & 1 \end{array}\right] U + O(h^4) && (3.30)\\ &= \frac{1}{12}h^2 u_4^{iv}(x_i) + O(h^6). && (3.31)\end{aligned}$$

Similarly one could make the second order accurate kernel fourth order accurate by subtracting the estimate of the fourth derivative from it. This leads to an setup scheme based on matrix powers.

By matrix powers Higher order difference methods can also be constructed by matrix powers, which is simpler as the construction based on the solution of linear systems as described above. section. The second order accurate finite difference approximations of the $2 \cdot n$-th derivative are given by the n-th power of the second order finite difference discretisation of the second derivative, the laplacian $D_2^2 = L$,

$$D_2^{2n} = \left(D_2^2\right)^n. \qquad (3.32)$$

A higher order accurate approximation to the second derivative is then obtained by combining the several second order accurate approximations of higher derivatives,

$$\begin{aligned} D_{2k}^2 &= \sum_{k=1}^{K} \frac{1}{c_{2k}} (-1h^2)^{k-1} D_2^{2k} && (3.33)\\ &= \sum_{k=1}^{K} \frac{1}{c_{2k}} (-1h^2)^{k-1} (D_2^2)^k && (3.34)\end{aligned}$$

Where the coefficients are given by the error terms of the finite difference method (cf. (3.31)),

$$\begin{aligned} c_{2k} &= \binom{2n+1}{n}(n+1)^2 && (3.35)\\ &= 1, 12, 90, 560, 3150, ... && (3.36)\end{aligned}$$

The kernels of higher order difference matrices can be found as convolutions of the second order kernel instead of computing the matrix powers explicitly, if one considers the kernels to be padded with zeros at the ends.

$$K_{2k+2} = (K_{2k} * K_2)[i] = \sum_{j=0}^{2k+1} K_2[i] * K_{2k}[j-i+1], i = 1, 2, ..., 2k+2, \qquad (3.37)$$

For a correct approximation of the homogeneous Dirichlet boundary the first terms of the sum have to be dropped. Note that the homogeneous Dirichlet boundary conditions are implicitly retained in the matrix powers. This means that matrices of higher oder derivatives need to be modified to satisfy the boundary conditions if higher derivatives are nonzero at the boundary.

Matrix powers can also be used to set up difference matrices of higher derivatives on non-uniform grids. However, the error of the finite difference approximation on a non-uniform grid is not anymore the multiple of higher derivatives and higher order accurate differences on non-uniform grids can therefore not be set up in this way.

Disadvatantages of Higher Order Accurate Finite Difference Approximations Higher order accurate finite difference approximations have three major disadvantages:

- The solution must be smooth enough, e.g higher order derivatives must exist, see section 3.2.6.
- A correct treatment of the boundary condition leads to asymmetric matrices for higher order derivatives, see section 3.2.5.
- The discretisation matrix contains more non-zero elements. However this in as much compensated as the accuracy increases exponentially compared to only a linear increase in non-zero elements.

Richardson Extrapolation

The special behaviour of the finite difference approximation to be proportional to higher derivatives allows to extrapolate solutions with a higher order of accuracy from two solutions on differently spaced grids [Ric11]. Assume that the error of the finite difference approximation is of the form

$$u_*(x_i) = u_h(x_i) + a(x_i)h^2 + b(x_i)h^4 + O(h^6). \tag{3.38}$$

Then, by doing another approximation with a different step-width of $c \cdot h$ again an equation system can be set up. If one eliminates the highest error term $a(x_i)$ this yields an approximation to the analytic solution $u_*(x_i)$ with an higher order of accuracy,

$$\begin{bmatrix} 1 & -h^2 \\ 1 & -c^2h^2 \end{bmatrix} \begin{bmatrix} u_*(x_i) \\ b(x_i) \end{bmatrix} = \begin{bmatrix} 1 & 0 & h^4 \\ 0 & 1 & c^4h^4 \end{bmatrix} \begin{bmatrix} u_h(x_i) \\ u_{c\cdot h}(x_i) \\ a(x_i) \end{bmatrix} \tag{3.39}$$

$$u_*(x_i) = \frac{u_{ch}(x_i) - c^2 u_h(x_i)}{1-c^2} - h^4 b(x_i)\frac{c^2-c^4}{1-c^2} \tag{3.40}$$

$$u_*(x_i) = \frac{4u_{ch}(x_i) - u_h(x_i)}{3} - \frac{1}{4}h^4 b(x_i), c = \frac{1}{2}. \tag{3.41}$$

This can be recursively continued to eliminate yet more higher order terms from the error. The coefficients of up to 10th order accuracy are given in table A.5.

Boundary Conditions

As the hydrogen atom is confined, homogenous Dirichlet boundary condition conditions are assumed where the wave function is zero at the boundary of the domain. Homogeneous Dirichlet boundary conditions can be imposed in two ways for the second order accurate finite difference approximation of the second derivative,

- either they are strongly imposed by stripping the rows and columns of the boundary points from the matrix,
- or they are weakly imposed by setting the diagonal element in the rows of the boundary points a very large value, e.g. 10^{12}.

Strongly imposed homogeneous Dirichlet boundary conditions are implicitely applied by the standard finite difference matrix setup.

For some problems higher order accurate difference matrices can simply be set up by assuming homogenous Dirichlet boundary conditions for higher derivatives. This is for example the case for the discrete Laplacian and the unconfined hydrogen atom. However, for the confined hydrogen atom higher order derivatives are in general non-zero at the boundary and special considerations have to be made.

A fourth order accurate finite difference matrix requires one "ghost" point oudside the domain. This point can be eliminated by an fourth order accurate extrapolation. A smarter approach is to use the polynomial approximation with an unequal number of points to the left and the right. The fourth order accurate difference kernel of the second derivative for a first point on the left side of the domain becomes

$$U = [u(0), u(h), u(2h), u(3h), u(4h)]^T \tag{3.42}$$

$$u_4''(h) = \frac{1}{12h^2}[-3, -10, 18, -6, 1]U + O(h^4) \tag{3.43}$$

$$u_4^{iv}(h) = \frac{1}{h^2}[1, -4, 6, -4, 1]U + O(h^2). \tag{3.44}$$

Due to the boundary conditions the discretisation matrix of higher order derivatives become asymmetric. As higher order accurate matrices are the sum of higher order derivative matrices, they are also asymmetric.

3.2.6 Influence of Smoothness on the Accuracy of Finite Difference Approximations

A function which is not sufficiently smooth cannot be accurately approximated by finite differences. This applies to highly oscillating functions and to functions with discontinuous derivatives where equation (3.11) does not hold. A quantitative criteria to determine whether a function is sufficiently smooth is found by investigating the basic oscillatory function

$$u(x) = e^{ikx}. \tag{3.45}$$

Where the first two analytic derivatives u'_* and u''_* are given by

$$u'_*(x) = ike^{ikx} \tag{3.46}$$

$$u''_*(x) = -k^2e^{ikx}. \tag{3.47}$$

The finite difference approximations u'_h and u''_h of the derivative differ from the analytic solution,

$$u'_h(x) = \frac{e^{ik(x+h)} - e^{ik(x-h)}}{2h} \tag{3.48}$$

$$= \frac{1}{h}\sinh(ikh)e^{ikx} \tag{3.49}$$

$$= \frac{i}{h}\sin(kh)e^{ikx} \tag{3.50}$$

$$u''_h(x) = \frac{e^{ik(x-h)} - 2e^{ikx} + e^{-ik(x+h)}}{h^2} \tag{3.51}$$

$$= \frac{2}{h^2}(1 - \cosh(ikh))e^{ikx} \tag{3.52}$$

$$= \frac{2}{h^2}(1 - \cos(kh))e^{ikx} \tag{3.53}$$

The relations (3.50) and (3.53) show that the approximations are more exact for small step sizes h and low frequency components k. The approximation is furthermore only valid as long as the argument of the trigonometric functions hk is smaller than π. This is also known as Nyquist-Shannon sampling theorem [Sha49].

Ergo, a second order finite difference approximation of the first derivative of a function with highest frequency component $k = 2\omega$ is only valid iff the step size h satisfies:

$$h < \frac{\pi}{k} = \frac{1}{2\omega}. \tag{3.54}$$

The second order finite difference approximation of the second derivative requires a step size of

$$h < \frac{\pi}{2k} = \frac{1}{\omega}. \tag{3.55}$$

The quality of approximation with respect to frequency and step size is depicted in figure 3.2.6.

This analysis can be repeated for higher order accurate approximations. Higher order accurate approximations require in general a smaller step size for the same frequency.

3.2.7 Smoothness of the Eigenfunctions of the Schrödinger Equation

The frequency components of the Schrödinger Equation are obtained transforming its solution into Fourier space. For the ground state (2.45) along the x-axis this yields with $\chi = a_0 x$

$$\hat{\psi}(k) = \int_{-\infty}^{\infty} \frac{1}{\sqrt{\pi a_0^3}} e^{-|\chi|} e^{-2\pi ikx} d\chi \tag{3.56}$$

$$= \frac{1}{\sqrt{\pi a_0^3}} \left(\int_{-\infty}^{0} e^{-\chi(2\pi ik-1)} d\chi + \int_{0}^{\infty} e^{-\chi(2\pi ik+1)} d\chi \right) \tag{3.57}$$

$$= \frac{1}{\sqrt{\pi a_0^3}} \left(\frac{-1}{2\pi ik - 1} + \frac{1}{2\pi ik + 1} \right) \tag{3.58}$$

$$= \frac{1}{\sqrt{\pi a_0^3}} \frac{2}{4\pi^2k^2 + 1} + 0\,i. \tag{3.59}$$

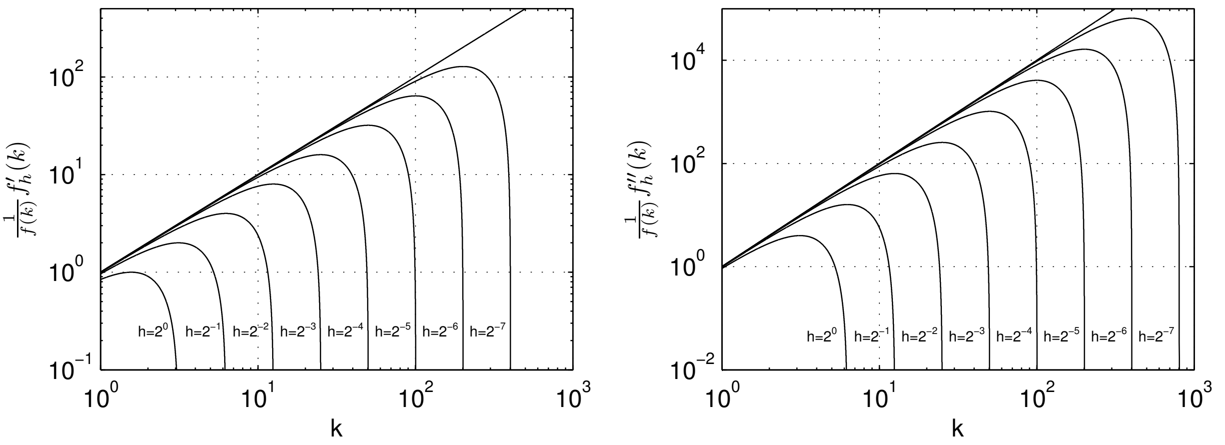


Figure 3.1: Influence of smoothness on the finite difference approximation demonstrated for function e^{ikx} for different step sizes h and frequencies k, left: first derivative, right: second derivative.

The Fourier coefficients (3.59) decay slowly compared to the coefficients of analytic functions, which decay exponentially. The solution contains therefore high frequency components and according to equation (3.53) this implies that a small step size is required before convergence of the finite difference method sets in.

Based on the Fourier transform of the ground state one can also reason that the solutions of the Schrödinger equation become smoother with increasing dimensions, as the Fourier coefficients decay faster due to the relation

$$e^{-\sqrt{x^2+y^2+z^2}} \leq e^{-\sqrt{x^2+y^2}} \leq e^{-\sqrt{x^2}}. \tag{3.60}$$

This implies that conversion sets in earlier in higher dimensions.

The solutions to the Schrödinger equation are not only none smooth, but the derivative of the wavefunction are undefined at the location of the nucleus. Therefore the grid has to be set up in a distinctive way to assure convergence. In two and higher dimensions the grid should be set up such that

- the nucleus is not located directly on a grid point,
- the grid spaces are fine close to the location of the nucleus.

In one dimension this approach breaks down as one would still differentiate over the location of the nucleus, even if it is not directly located on a grid point. An exception is the radial equation, where the discontinuity is located at the boundary and not inside the computational domain.

3.2.8 Finite Difference Approximation on Non-Uniform Grids

In the previous subsection it was shown that with a uniform grid there are many points required until the approximation of non-smooth functions converges. As a large number of grid points requires longer computation times and more memory, one wants to use as few grid points as possible. Furthermore it was shown that the eigenfunctions of the Schrödinger equation are not smooth close to the Coulomb singularity (cf. (2.45)). In such a case a higher accuracy with fewer grid points can be achieved by locally refining the grid where the function is not smooth. Such a grid is depicted in figure 3.2.

Although the grid points are not any more equidistant, they are located such that:

$$x_i < x_{i+1}, 0 < i < m+1 \tag{3.61}$$
$$x_0 = 0 \tag{3.62}$$
$$x_{m+1} = L \tag{3.63}$$
$$h_l = x_i - x_{i-1} \tag{3.64}$$
$$h_r = x_{i+1} - x_i. \tag{3.65}$$

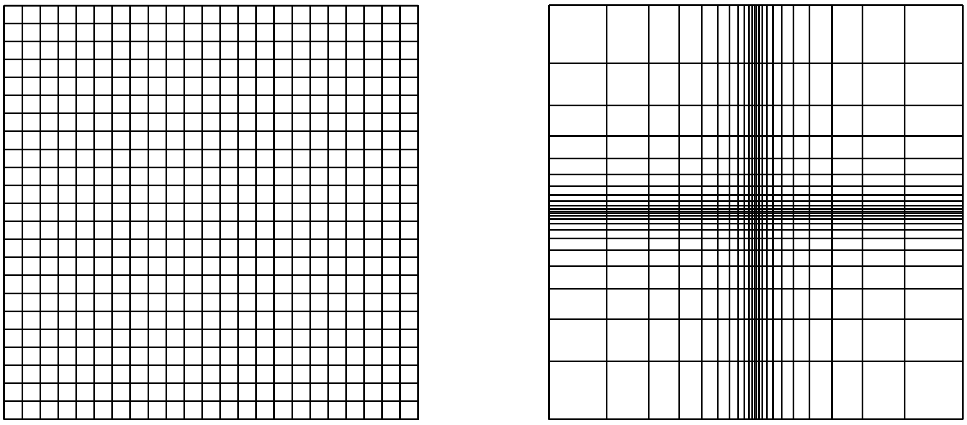


Figure 3.2: 2D finite difference grids, left: constant step, right: variable step

A priori Setup of a Non-Uniform Grid

A relocation scheme for the grid points can be set up by realising that, for a constant step size h the error varies proportional to the fourth derivative of the solution:

$$err(r_i) = c_1 h^2 \psi^{vi}(r_i) = c_1 h^2 e^{-r_i} \tag{3.66}$$

Which implies that the error remains constant if the step size h is chosen to be proportional to the inverse of the fourth derivative of the solution,

$$h(r_i) = c_2 e^{r_i/2}. \tag{3.67}$$

As the location of a grid point is equal to the sum of grid spaces between the origin and the grid point, this leads to a relocation scheme for the grid points of a constant grid

$$r_v(r) = \alpha_1 \left(e^{\alpha_2 r} - 1\right). \tag{3.68}$$

A varying grid in higher dimension can be set up by relocating the grid points according to equation (3.68) for each coordinate axis by choosing the distance to the nucleus as parameter r.

Adaptive Grid Setup

For a confined atom the solution differs from those of the unconfined atom and a predetermined grid based on the unconfined wave functions is therefore not anymore optimal. In this case it is preferable to refine the mesh locally according to algorithm 1.

Algorithm 1 Adaptive FDM Refinement

```
1. compute fourth derivative of the solution as $u^{iv}(x) = D_2^4 u(x)$
2. estimate local error as $err_{loc} = \frac{1}{12} h^2 u(x)$
3. interpolate error into element centres
4. find maximum absolute error $err_{max}$
5. mark elements with large error contribution $|err_{loc}| > \frac{1}{4}|err_{max}|$
6. add the midpoints of the marked elements to the grid
```

Difference Scheme

The matrix kernel for a variable grid can also be recovered by setting up a system of equations (3.2.8),

$$\begin{bmatrix} h_r & {}^1\!/_2\, h_r^2 \\ -h_l & {}^1\!/_2\, h_l^2 \end{bmatrix} \begin{bmatrix} u_2'(x_i) \\ u_2''(x_i) \end{bmatrix} = \begin{bmatrix} 0 & -1 & 1 \\ 1 & -1 & 0 \end{bmatrix} \begin{bmatrix} u_2(x_{i-1}) \\ u_2(x_i) \\ u_2(x_{i+1}) \end{bmatrix}, h_l = \qquad x_i - x_{i-1}$$

$$h_r = x_{i+1} - x_i.$$

Which has the solution

$$u_2''(x_i) = \left[\frac{2}{h_l(h_l+h_r)}, \frac{-2}{h_r h_l}, \frac{2}{h_r(h_l+h_r)}\right] U + O(h), \tag{3.69}$$

$$\tag{3.70}$$

As the coefficients of the difference matrices are not any more integers divided by a constant factor, it is especially important to avoid cancellation errors while setting up the difference matrices on a variable grid.

Preserving Symmetry

A discretisation matrix based on a grid with varying step-size is asymmetric. However, the asymmetric eigenvalue problem can be transformed into a symmetric generalised eigenvalue problem by a similarity transform (equation (3.71) and section 4.12),

$$DTD^{-1} = \tilde{T} \tag{3.71}$$

$$Tx = \lambda x \tag{3.72}$$

$$Dx = y \tag{3.73}$$

$$DTD^{-1}y = DD^{-1}y \tag{3.74}$$

$$\tilde{T}y = \lambda y. \tag{3.75}$$

Discretisation Error

The discretisation error is estimated by expanding more terms of the Taylor polynomial similarly as in the error estimation for constant grids,

$$\begin{bmatrix} h_r & {}^1\!/_2\, h_r^2 \\ -h_l & {}^1\!/_2\, h_l^2 \end{bmatrix} \begin{bmatrix} u_4'(x_i) \\ u_4''(x_i) \end{bmatrix} = \begin{bmatrix} 1 & -1 & 0 & {}^1\!/_6\, h_r^3 & -{}^1\!/_{24}\, h_r^4 \\ 0 & -1 & 1 & -{}^1\!/_6\, h_l^3 & {}^1\!/_{24}\, h_l^4 \end{bmatrix} \begin{bmatrix} u_4(x_{i-1}) \\ u_4(x_i) \\ u_4(x_{i+1}) \\ u_4'''(x_i) \\ u_4^{iv}(x_i) \end{bmatrix}. \tag{3.76}$$

Which leads to a third order accurate approximation u_4'' of the second derivative,

$$u_*''(x_i) = u_4''(x_i) + O(h^3) \tag{3.77}$$

$$= \left[\frac{2}{h_r(h_r+h_l)}, \frac{-2}{h_r h_l}, \frac{2}{h_l(h_r+h_l)}\right] U + \frac{1}{3}(h_r - h_l)u_4'''(x_i) + \frac{1}{12}(h_l^2 - h_l h_r + h_r^2)u_4^{iv}(x_i) + O(h^3) \tag{3.78}$$

$$= u_2''(x_i) + \frac{1}{3}(h_r - h_l)u_4'''(x_i) + \frac{1}{12}(h_l^2 - h_l h_r + h_r^2)u_4^{iv}(x_i) + O(h^3) \tag{3.79}$$

At those points where the two adjacent steps are equal this expression simplifies to the difference kernel derived for a constant step size (3.18) and the approximation becomes fourth order accurate. However, this does not hold in general and for a varying step the approximation (3.78) remains only first order accurate.

3.2.9 Finite Differences in Higher Dimensions

Total derivatives in higher dimensions are formed as the sum of partial derivatives of each dimension. So are the derivative matrices in higher dimension the sum of partial derivative matrices (equation (3.80)),

$$\Delta = \sum_{i=0}^{n} \frac{\partial^2}{\partial \rho_i^2}. \tag{3.80}$$

A partial derivative matrix is the Kronecker product of a one dimensional derivative matrix and further identity matrices. The Laplacian can be set up as described in algorithm 2. The discretisation kernels and respective structure of the matrices are shown in figure 3.3.

Algorithm 2 Matrix Combination in Higher dimensions

```
$A = A_x$
$I = I_x$
for $i = y, z$
    $A = kron(A, I_i) + kron(I, A_i)$
    $I = kron(I, I_i)$
```

The potential term does not contain any derivative and hence yields a diagonal matrix in real space.

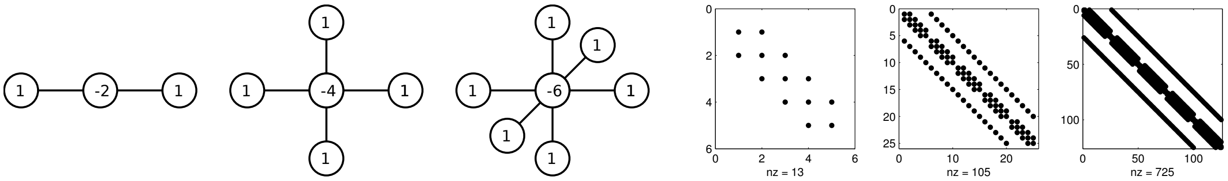


Figure 3.3: Finite difference kernels in 1, 2 and 3D (left) and structure of resulting discretisation matrices (right)

However, the potential contains a distance term which in Cartesian coordinates equals $\rho = ||x_c - x||$ for a nucleus located at x_c. The potential matrix is set up in three steps by first computing the distances in eanch dimension V_x^2, V_y^2, V_z^2 followed by the combining the matrices as in algorithm 2 into V^2 and completed by taking the square root of the elements.

If the nucleus is located such that the domain is symmetric then the energy levels degenerate. This means that some eigenvalues occur multiple times. For the hydrogen atom in free space the energy levels are n^2 degenerated. Non-unique eigenvalues may complicate the eigenvalue computation. One can avoid duplicate eigenvalues by slightly perturbing the location of the nucleus. This is justified, as long as the perturbation is smaller than the model error.

3.2.10 Finite Difference Solution of the Radial Schrödinger Equation

In the radial Schrödinger equation (2.37) the the Coulomb singularity is located on the boundary of the computational domain. Therefore a solution by the finite difference method is possible. Figure 3.2.10 shows the convergence of the finite difference approximation to the radial Schrödinger equation for the unconfined atom. As the solution is not smooth enough close to the Coulomb singularity, it cannot be sufficiently sampled by a uniform grid with few grid points. Therefore convergence for a uniform grid does not set in until the grid becomes very fine. However, if one uses a variable grid, such that the region close to the Coulomb singularity is sufficiently sampled even with few grid points, then convergence sets in immediately and the discretisation error is much lower.

Note that also finite difference discretisations on unstructured grids are possible [Dur02, ST03, GA06]. However, those suffer from similar discrepancies as the finite difference discretisations with variable step size on structured grid and are therefore not discussed here.

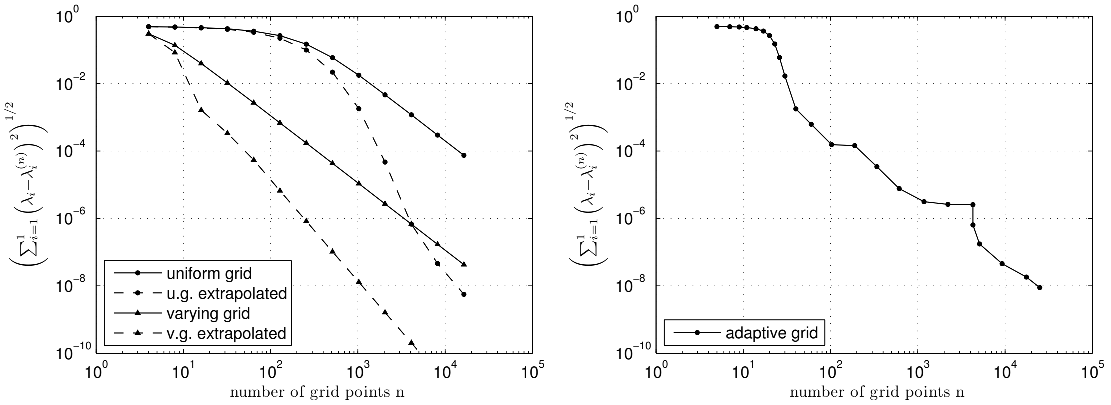


Figure 3.4: Finite difference approximation to the radial Schrödinger equation for the domain $\Omega = [0, 400]\, a_0$, the graphs show the discretisation error of the first eigenvalue compared to the analytic solution. Left: predetermined uniform and varying grid, right: adaptive grid.

3.3 Finite Element Discretisation

The finite element method (FEM) is another method to discretise differential equations and to numerically approximate their solutions. The finite element method has several advantages over the finite difference methods, such as:

- Error bounds can be derived for the approximation.
- Constraints on the existence of higher order derivatives are weaker.
- Local refinement with an unstructured meshes does not decrease the accuracy or increase the complexity of the assembly routine.
- Discretisations with arbitrarily shaped domains are feasible.
- The discretisation matrices remain symmetric for unstructured grids and higher order accurate approximations.

A practical introduction to the finite element method can be found in [Red05], a theoretical treatment is given in [Joh88], a chapter on eigenvalues is found in [SF88].

3.3.1 Foundation of the Finite Element Method

Compared to finite differences the approximation is not only found in a vector space consisting of a set of points Ω_h, but in in a function space defined everywhere in the domain Ω. The approximate solution f_h is the function in the vectore space V_h which is closest to the exact solution f_* defined in the vector space V. The finite element method can thus be formulated as a minimisation problem,

$$\begin{aligned} &\text{Find } u_h \in V_h, \text{ s.t.} \\ &|F(u_h)| < |F(v)| \; \forall v \in V_h \quad (3.81) \\ &V_h \subset V \quad (3.82) \\ &V : \{v : v \text{ is continuous}, \\ &\quad v' \text{ is piecewise continuous.} \quad (3.83) \end{aligned}$$

Where $F : V \rightarrow R$ is a functional and v is a function. V is the function space in which the exact solution is defined. As subspace V_h ususally piecewise polynomials are chosen. The solution u_h is the projection of the exact solution u_* to the subspace V_h. Note that a usual assumption is that F is positive definite. However, this is unimportant for eigensystems, as the spectrum can always be shifted to become definite or indefinite without changing the solution vectors.

The exact solution satisfies

$$< u_* - u_h, g >= 0, \forall v \in V_h. \quad (3.84)$$

And if the functional is positive definite than

$$||u_* - u_h|| \leq C||u_* - v||. \quad (3.85)$$

To find a finite element solution following steps are taken

- The weak formulation is derived.
- Compatible elements and quadrature rules are chosen.
- An a-priori error estimate is computed.
- The method is implemented in some programming language.
- The discretisaton matrices are assembled.
- The discretised system is solved.
- An a-posterior error estimate is computed. If necessary the mesh is refined and the computation repeated.

3.3.2 Some Definitions

Strong Convergence

A finite element solution u_h is strongly convergent to the exact solution u_* if

$$\lim_{h \to 0} ||u_* - u_h|| = 0. \tag{3.86}$$

Where h is a measure of the element size.

Degenerated Triangle

A triangle is degenerated if its corner points are colinear, that is there exist a constant λ_1, λ_2, that

$$p_3 = \lambda_1 p_1 + \lambda_2 p_2. \tag{3.87}$$

A degenerated triangle has an area and minimum corner angle of zero. Degenerate triangles are also called ill conditioned and triangles which are not close to being degenerate are called regular. Triangles far from being degenerate are almost equilateral, e.g. all sides are of similar length. An d-dimension an element does not degenerate as long as r^d/v is bounded, where r is a measure of the diameter and v a measure of the volume of the element [H.42].

Gradient

The gradient is the row vector containing the first order partial derivatives of a function. In two dimension it becomes

$$\nabla u(x,y) = \begin{bmatrix} \frac{\partial u}{\partial x} \\ \frac{\partial u}{\partial y} \end{bmatrix}. \tag{3.88}$$

Hessian

The Hessian matrix contains the second-order partial derivatives of a function. In two dimensions it becomes

$$H(u,x,y) = \begin{bmatrix} \frac{\partial^2 u}{\partial x^2} & \frac{\partial^2 u}{\partial x \partial y} \\ \frac{\partial^2 u}{\partial y \partial x} & \frac{\partial^2 u}{\partial y^2} \end{bmatrix}. \tag{3.89}$$

Similar matrices can be set up for higher order derivatives. Even orderer derivative matrices are always symmetric.

Definition of a Finite Element

A finite element consists of:

- A shape spanned by the set of points, basic shapes are lines (1D), triangles (2D) and tetrahedrons (3D).
- A set of unknowns (degrees of freedom). This is usually the function value or its derivative at a set of points (P). Basic schemes use function values at the corner points of the element. Although the solution is defined in the complete domain, explicit values are only computed for the degrees of freedom.
- A set of test functions (ϕ), usually polynomials, determined by the points. Basic test functions are linear.

Weak Formulation

A differential equation is solved by the finite element method in its weak form. The weak form is derived by multiplying the differential equation with a test function $v(x)$ and by integration of parts. Any solution of the strong form is also a solution of the weak form. However, the weak form itself is less stringent and admits further solutions. The test functions must satisfy the boundary conditions. The weak formulation of the radial Schrödinger (equation (3.90)) is given by equation (3.92),

$$\frac{1}{2}\ u''(x) + \frac{1}{x}u(x) = \lambda u(x) \tag{3.90}$$

$$\int_\Omega \left(\frac{1}{2}u''(x) + \frac{1}{x}u(x) - \lambda u(x)\right) v(x)dx = 0 \tag{3.91}$$

$$-\frac{1}{2}\int_\Omega u'(x)v'(x)dx + \frac{1}{2}\ u'(x)v(x)\Big|_\Gamma + \int_\Omega \frac{1}{x}u(x)v(x)dx = \lambda \int_\Omega u(x)v(x)dx. \tag{3.92}$$

Galerkin's Method

Galerkin's method is used to solve the weak formulation. This is done by choosing a set of n trial functions ϕ satisfying the weak formulation. An approximate solution u to the differential equation is given as a linear combination of the trial functions (equation (3.93)),

$$u(x) = \sum_{i=1}^{m} u_i\phi_i(x) \tag{3.93}$$

By plugging the the approximate solution U into the weak formulation one attains a system of equations consisting of the matrices A and B,

$$\begin{aligned}-\frac{1}{2}\int_\Omega u_i\phi_i(x)'\sum v_j(x)'dx + \frac{1}{2}\ u_i\phi_i(x)'\sum v_j(x)|_\Gamma \\ + \int_\Omega \frac{1}{x}u_i\phi_i(x)\sum v_j(x)dx = \lambda\int_\Omega u_i\phi_i(x)\sum v_j(x)dx \end{aligned} \tag{3.94}$$

$$Au = \lambda Bu. \tag{3.95}$$

This is a generalised eigenvalue problem where the matrix A is called stiffness matrix and the matrix B the mass matrix. The trial function are usually chosen to be identical to the test function $v(x)$. This makes the discretisation matrices symmetric. The test function are usually choosen such that they are only non-zero at a small subregion of the domain. This makes the discretisation matrix sparse, but it has usually slightly more nonezeros and no strictly diagonal pattern compared to finite difference matrices.

Variational methods are also based on the weak formulation, but the test functions span the complete domain. However, it is difficult to find suitable test functions for arbitrary equations and domains. The innovation of the finite element method is to use only local test functions.

3.3.3 Finite Element Approxiation in 1D

Test Functions If the approximate solution u consists of piecewise linear functions, then it is determined by the line segments

$$u(x) = \sum_{i=1}^{m} u_i\phi_i(x) \tag{3.96}$$

$$= \tilde{c}_{i,0} + \tilde{c}_{i,1}x. \tag{3.97}$$

Note that the derivative is constant inside the element,

$$\frac{\partial u(x)}{\partial x} = \sum_{i=1}^{3} \frac{\partial \phi(x,y)}{\partial x} = \tilde{c}_1. \tag{3.98}$$

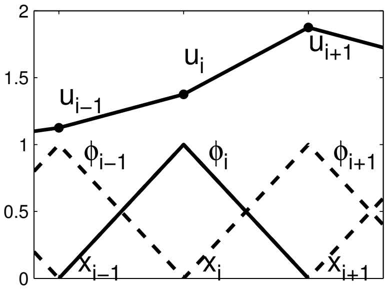
This can only be satisfied if the test function are likewise piecewise linear

$$\phi_i(x) = c_{i,0} + c_{i,1}x. \tag{3.99}$$

With the boundary conditions

$$\phi_i(x) = \begin{cases} 1 & x = x_i \\ 0 & x \le x_{i-1} \text{ or } x \ge x_{i+1} \end{cases} \tag{3.100}$$

Figure 3.5: Piecewise linear trial functions used in the finite element method



So only the test functions of adjacent elements form nonzero products. This can be expressed in matrix form as

$$U(x) = \begin{bmatrix} 1 & x \end{bmatrix} \begin{bmatrix} c_{i,0} & c_{i+1,0} \\ c_{i,1} & c_{i+1,1} \end{bmatrix} \begin{bmatrix} u_i \\ u_{i+1} \end{bmatrix}, \; x_i \leq x \leq x_{i+1}. \tag{3.101}$$

By probing this system of equations with the test functions it becomes

$$\begin{bmatrix} 1 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & x_i \\ 1 & x_{i+1} \end{bmatrix}}_{A} \begin{bmatrix} c_{i,0} & c_{i+1,0} \\ c_{i,1} & c_{i+1,1} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \end{bmatrix}. \tag{3.102}$$

The matrix A is known as Vandermonde matrix. It consists only of known quantities and is determined for each finite element by its end points. Its determinant equals the length of the element.

By inverting the matrix A one finds the coefficients of the test functions. The test functions take the form of pointed hats, see image 3.3.3 and equation (3.103).

$$\phi_i = \begin{cases} \dfrac{x - x_{i-1}}{x_i - x_{i-1}} & x_{i-1} \leq x \leq x_i \\ \dfrac{x_i - x}{x_{i+1} - x_i} & x_i \leq x \leq x_{i+1} \\ 0 & \text{otherwise} \end{cases} . \qquad \phi_i' = \begin{cases} \dfrac{1}{x_i - x_{i-1}} & x_{i-1} \leq x \leq x_i \\ \dfrac{-1}{x_{i+1} - x_i} & x_i \leq x \leq x_{i+1} \\ 0 & \text{otherwise} \end{cases} . \tag{3.103}$$

Convergence and Error Estimates

Error estimates for bounded and definite operators can be derived by Cea's lemma, which states that

$$||u_* - u_h|| \leq C||u_* - v||, \; \forall v \in V_h. \tag{3.104}$$

Note that the operator of the hydrogenic Schrödinger equation is indefinite. Which however does not matter, as one may consider the operator to be shifted such that it becomes definite.

Hence the error can be estimated by choosing v to be the projection of u_* onto the space V_h and considering its interpolation error $err(x)$,

$$|err| = |u - \pi|_\infty. \tag{3.105}$$

Where the linear interpolation $\pi(x)$ of the exact solution $u_*(x)$ is determined by the test functions and sampled function values

$$\pi(x) = \sum_{j=i}^{i+1} u(x_j)\phi(x_j) \tag{3.106}$$

$$= \frac{x_i - x}{x_{i+1} - x_i} u(x_i) + \frac{x - x_i}{x_{i+1} - x_i} u(x_{i+1}). \tag{3.107}$$

An approximation of the function value is also given by Taylor expansion

$$u(x_i) = (u(x) + ((x_i - x)u'(x) + \frac{1}{2}(x_i - x)^2 u''(x) + O((x_i - x)^3) \tag{3.108}$$

$$u(x_{i+1}) = (u(x) + ((x_{i+1} - x)u'(x) + \frac{1}{2}(x_{i+1} - x)^2 u''(x) + O((x_{i+1} - x)^3). \tag{3.109}$$

Susbstition of (3.108) and (3.109) into (3.107) and neglegting higher order terms yields

$$u(x) = u(x) + \frac{1}{2}\left((x_i - x)^2 \frac{x_{i+1} - x}{x_{i+1} - x_i} + (x_{i+1} - x)^2 \frac{x - x_i}{x_{i+1} - x_i}\right) u''(x) \tag{3.110}$$

$$= u(x) + \frac{1}{2}(x - x_i)(x_{i+1} - x)u''(x). \tag{3.111}$$

Which completes the interpolation error estimation by substitution into (3.105)

$$|u(x) - \pi(x)|_\infty = |{}^1\!/_2\, u''(x)(x - x_i)(x_{i+1} - x)| \tag{3.112}$$

$$\leq {}^1\!/_2 \max_{\tilde{x}} |u''(x)|(x - x_i)(x_{i+1} - x) \tag{3.113}$$

$$\leq {}^1\!/_8\, (x_i - x_{i+1})^2 \max_{\tilde{x}} |u''(\tilde{x})|,\ x_i \leq \tilde{x} \leq x_{i+1}. \tag{3.114}$$

Thus the finite element approximation with piecewise linear interpolation is second order acurate, if the second derivative exists. This is an a-priori error bound, as it can be determined before computing the numerical approximation.

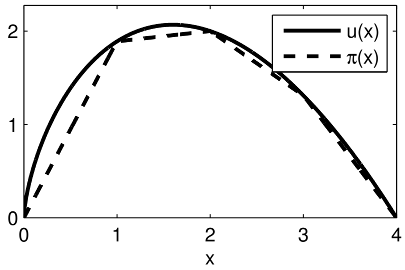


Figure 3.6: Interpolation error of the piecewise linear finite element method

Numerical Quadrature

If for the integrals no analytic solution exist or if the expressions become to costly to evaluate, then the integrals can be approximately solved by numerical quadrature. For numerical integration over a finite element the function u is sampled at a set of n points. Where the sample points do not necessarily correspond to mesh points. The sampled function values are then weighted by the coefficients w_k and the size A_i of the finite element,

$$\int_{x_i}^{x_{i+1}} f(x) \approx A_i \sum_{k=1}^{n} w_k f(x_i + p_k(x_{i+1} - x_i)),\ 0 < p_k < 1 \tag{3.115}$$

$$A_i = \begin{vmatrix} 1 & 1 \\ x_{i-1} & x_i \end{vmatrix} = x_i - x_{i-1} \tag{3.116}$$

$$\sum_{k=1}^{n} w_k = 1. \tag{3.117}$$

The accuracy of the quadarture depends on the number and the location of the sample points. For computing a finite element solution with nth-order polynomials for a mth-order partial differential equations the integration scheme must be at least kth-order accurate, with

$$k = 2(n + 1) - m. \tag{3.118}$$

Hence for solving the Schrödinger equation ($m = 2$) with piecewise linear elements ($n = 1$) the integration scheme should be at least second order accurate. Most oftenly integration schemes of the Gauss-Legendre, Newton-Cotes or Clenshaw-Curtis families are used. Some integration schemes are listed in table B.1. Tables with integration schemes can also be found in [Cow73, Dun85].

Newton-Cotes Quadrature

Newton-Cotes quadrature rules have two advantages compared to Gauss-quadrature rules. Firstly they yield diagonal mass matrices in combination with Lagrangian-dasis functions of the same order. Secondly they quadrature weights can be found as closed form expressions in arbitrarily many dimensions. However, contrary to Gauss quadrature rules which integrate polynomials with p-quadrature points up to degree $2p-1$ exactly, Newton-Cotes quadrature rules integrate polynomials only up to degree $p+1$ exactly. Examples for Newton-Cotes quadrature rules are the trapezoidal rule ($p=2$) and Simpson's rule ($p=3$). Below the the derivation is given for closed Newton-Cotes quadrature rules, which have the same quadrature points as the Lagrangian basis functions.

At first the Vandermonde matrix A of the quadrature points is set up. For the trapezoidal rule it is,

$$A = \begin{bmatrix} 1 & x_1 \\ 1 & x_2 \end{bmatrix}. \tag{3.119}$$

Then the coefficients $\phi_{i,j}$ of the polynomial basis functions are found by inverting the Vandermonde matrix,

$$\Phi = A^{-1}. \tag{3.120}$$

The basis functions are integrated once,

$$\psi_{ij} = \begin{cases} 0 & j = 1 \\ \frac{1}{i!}\phi_{i,j-1} & j = 2,..,p+1 \end{cases} \tag{3.121}$$

Where the approximate integral is given by,

$$I = \vec{w}\Psi\vec{x}. \tag{3.122}$$

Finally the quadrature weigths w are found by evaluating the integrals of the test functions in the interval $[0,1]$ and by multiplying with the length (determinand $|A|$) of the element.

$$w_i = |A|(\Psi\vec{1} - \Psi\vec{0}) \tag{3.123}$$

$$= |A| \sum_{i=2}^{p+1} \psi_{i,j} \tag{3.124}$$

Note that some weights for higher order Newton-Cotes rules are negative.

Diagonal Mass Matrices

Diagonal, or lumped mass matrices are of advantage, as they allow the reformulation of the generalised eigenvalue problem into an ordinary eigenvalue problem 4.12. As exact analytic integration usually does not yield a diagonal mass matrix, using an appropriate numerical quadrature rule can be more efficient. In addition to being diagonal the mass matrix should also be strictly positive definite. A second order accurate quadrature rule yielding a diagonal mass matrix is the trapezoidal rule. It samples at the end points of the elements and works also in higher dimensions. However, quadrature rules yielding diagonal mass matrices for finite elements with higher order of accuracy in higher dimensions are nontrivial and require additional degrees of freedom, see [CJJT01].

Higher Order Accurate Discretisations

A higher accuracy cannot only be achieved by using a finier grid, but also by using higher order interpolation polynomials. Higher interpolation polynomials require more degrees of freedom per element. These degrees of freedom could either be more function values at additional points, or values of the derivative.
A quadratic interpolation is

$$u(x) = \sum u_i \phi_i(x,y) \tag{3.125}$$

$$\phi(x) = c_0 + c_1 x + c_2 x^2. \tag{3.126}$$

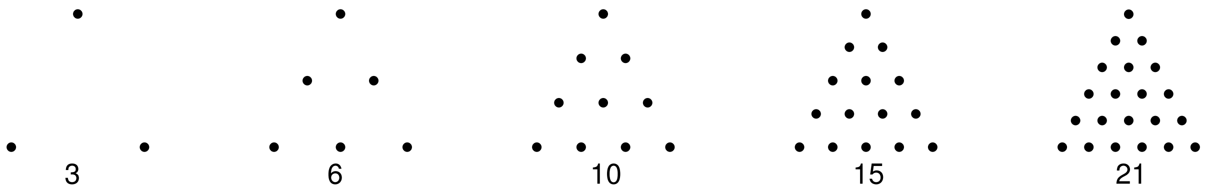


Figure 3.7: Degrees of freedom for the Lagrangian basis functions on the triangle for degree two to six. Note the similarity to Pascal's triangle.

If one uses one additional point in the centre of each element the coefficient equation (3.102) becomes

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & x_i & x_i^2 \\ 1 & x_{i+1} & x_{i+1}^2 \\ 1 & x_{i+2} & x_{i+2}^2 \end{bmatrix}}_{A} \begin{bmatrix} c_{i,0} & c_{i+1,0} & c_{i+2,0} \\ c_{i,1} & c_{i+1,1} & c_{i+2,1} \\ c_{i,2} & c_{i+1,1} & c_{i+2,2} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \tag{3.127}$$

An advantage of higher order accurate finite element methods over higher order accurate finite difference methods is that the discretisation matrices remain symmetric. Note that for irregular shaped domains boundary sides of elements locate at the domain boundary need to be curvilinear.

3.3.4 Finite Element Method in 2D

A basic finite element sheme in two dimensions uses triangular shaped elements with linear interpolation functions. The approximated solution $u(x, y)$ is given by

$$u(x, y) = \sum_{i=1}^{n} u_i \phi_i(x, y) \tag{3.128}$$

$$u(x, y) = \tilde{c}_{00} + \tilde{c}_{01} x + \tilde{c}_{10} y. \tag{3.129}$$

To ease the notation local indices for each element are furtheron used. The test functions $\phi_i(x, y)$ are given by

$$\phi_i(x, y) = c_{00} + c_{01} x + c_{10} y. \tag{3.130}$$

With the boundary condtions

$$\phi_1(x) = \begin{cases} 1 & x = x_1, y = y_1 \\ 0 & x = x_2, y = y_2 \\ 0 & x = x_3, y = y_3 \end{cases} \tag{3.131}$$

And hence with linear test functions the approximate solution inside a triangle is given by

$$u(x, y) = \begin{bmatrix} 1 & x & y \end{bmatrix} \begin{bmatrix} c_{1,0} & c_{2,0} & c_{3,0} \\ c_{1,1} & c_{2,1} & c_{3,1} \\ c_{3,2} & c_{3,2} & c_{3,2} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} \tag{3.132}$$

By probing this system of equations with the test functions it becomes

$$\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \underbrace{\begin{bmatrix} 1 & x_1 & y_1 \\ 1 & x_2 & y_2 \\ 1 & x_3 & y_3 \end{bmatrix}}_{A} \begin{bmatrix} c_{1,0} & c_{2,0} & c_{3,0} \\ c_{1,1} & c_{2,1} & c_{3,1} \\ c_{1,2} & c_{2,2} & c_{3,2} \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \tag{3.133}$$

Invering the Vandermonde matrix A yields again the test functions. The area of the triangle is equal to the determinant of A.

Error Estimation and Convergence

As in the one dimensional case the discretisation error is estimated by the interpolation error

$$||err|| \leq ||u - \pi||. \tag{3.134}$$

By neglecting higher order derivatives the two dimensional Taylor expansion for the function value u at the first corner point from any other point inside the triangle becomes

$$u(x_1, y_1) = u(x, y) + \nabla u(x, y)(p_1 - p) + \frac{1}{2}(p_1 - p)^T H(u, x, y)(p_1 - p). \tag{3.135}$$

The expansions for point two and three are similar. Substitution into the interpolation formula yields

$$\pi(x, y) = u(x, y) + \sum_{i=1}^{3} \phi_i(x, y)(p_i - p)^T H(u, x, y)(p_i - p) \tag{3.136}$$

$$|u - \pi| \leq \frac{1}{2}||H(u, x, y)||_\infty \sum_{i=1}^{3} \phi_i(x, y)(p_i - p)^T (p_i - p). \tag{3.137}$$

Where H is the Hessian matrix, c.f. (3.89). Solving this maximisation problem for p one finds

$$|u - \pi| = \frac{1}{2}||H(u, x, y)||_\infty \frac{a^2 b^2 c^2}{4|A|^2} \tag{3.138}$$

$$= \frac{1}{2}||H(u, x, y)||_\infty \frac{a^2 b^2 c^2}{4 a^2 b^2 \sin(\gamma)^2} \tag{3.139}$$

$$\leq \frac{1}{2}||H(u, x, y)||_\infty \frac{\max(a, b, c)^2}{4 \sin(\min(\alpha, \beta, \gamma))^2}. \tag{3.140}$$

Where a, b, c are the length of the triangle's side and α, β, γ are the inner angles of the triangle. Note the identity for the triangle area ${}^1\!/_2 ab \sin(\gamma) = {}^1\!/_2|A|$.

The finite element approximation with triangular shaped elements and piecewise linear interpolation polynomials is therefore second order accurate, if the second order partial derivatives exist and the triangles are not degenerated.

To estimate the error the second derivative of the approximate solution is required. However, as the approximate solution only consists of piecewise linear polynomials, the second derivative must be approximated. This can be done by first computing the first order partial derivatives at the element centres and then by approximating the second derivates between two neighbouring elements by finite differences.

$$\begin{aligned} H(u, x_1, y_1) &= H(u, x_2, y_2) \\ &= \begin{bmatrix} \frac{\frac{\partial u_1}{\partial x} - \frac{\partial u_2}{\partial x}}{\Delta x} & \frac{1}{2}\left(\frac{\frac{\partial u_1}{\partial x} - \frac{\partial u_2}{\partial x}}{\Delta y} + \frac{\frac{\partial u_1}{\partial y} - \frac{\partial u_2}{\partial y}}{\Delta x} \right) \\ \frac{1}{2}\left(\frac{\frac{\partial u_1}{\partial x} - \frac{\partial u_2}{\partial x}}{\Delta y} + \frac{\frac{\partial u_1}{\partial y} - \frac{\partial u_2}{\partial y}}{\Delta x} \right) & \frac{\frac{\partial u_1}{\partial y} - \frac{\partial u_2}{\partial y}}{\Delta y} \end{bmatrix} \end{aligned} \tag{3.141}$$

Where Δx and Δy are the midpoint distances. If Δx or Δy become are close to zero, then the respective contributions are set to zero, too. Here close to zero means

$$\Delta x^2 < 10^{-15}(\Delta x^2 + \Delta y^2) \tag{3.142}$$

After the error was computed for each element, those elements for refinement are selected where the local error satisfies

$$err_i > \frac{1^p}{2} \max_{0<j<n} err_j. \tag{3.143}$$

Where p is the order of the interpolation interpolinomial.

An alternative method to estimate the derivative is given in [EJ91].

One may choose alternative criteria to the minmimum angle such as the singular values of the edge-matrix [ABJ$^+$99]. Note that under certain cirumstances a solution may converge even if the minimum angle criterium is violated, [HKK12].

Triangulation of the Domain

The generation of good meshes for arbitrarily shaped domains is a nontrivial task. A good mesh, e.g. a mesh without degenerated triangles can be achieved by satisfying the delaunay condition, which is that no fourth point should be in the circumference of a triangle. For convex domains the tidieous task of triangulation can be avoided by simply starting with a mesh consisting of few triangles and then successively refining the mesh either uniformly or adaptively.

3.4 Increasing the Accuracy of the Finite Element Discritisation

There are three ways to make the numerical solution converge to the exact solution. Fistly to refine the mesh and to increase the number of grid points (h-adaptivity). Secondly to increase the order of the polynomial basis functions (p-adaptivity) and thirdly to relocate existing grid points inside the mesh (r-adaptivity).

3.4.1 Mesh Refinement

During mesh refinement the number of grid points is increased. There are three basic methods: uniform refinement, bisection refinement and red-green refinement. The latter two methods allow local refinement and with them meshes can be individually adapted for different problems. The number of grid points for the achieved accuracy will be close to optimal. Uniform refinement is in contrast only suitable for certain problems.

- By uniformly refining the mesh each triangle is split into four children. This method is trivial to implement.
- By refining a mesh with bisection one edge of each marked element is split into two, usually along its longest edge. This method can be generalised to higher dimensions without increasing complexity. However, it requires relatively many refinement steps to improve the accuracy of the solution. Bisection methods are covered in [Riv84]. Bisection method in arbitrarily many dimensions is discussed in [Tra97].
- By red-green refining the mesh each marked element is split into four children labeled red and triangles surrounding the refined region are split in two labeled green. This method requires that green triangles are merged before they are split again to avoid degeneration of triangles. The method is more complex to implement then the bisection method and its complexity is increasing with higher dimensions. However, the accuracy is improved with fewer refinements than with the bisection method. Red-green refinement was developed to set up hierarchical meshes for application in combination with Multigrid solvers [BDY88]. The red-green refinement in three dimensions is discussed in [Bey95, LJ96].
- Red-green-blue refinement is a special form of the red green refinement [Car04]. Which avoids the remerging of green triangles and hence does not require the storeage of colours. This is achieved by splitting the longest side of a green triangle as well (blue-refinement).

The red-green-blue refinement was implemented and used in the thesis.

3.4.2 Relocation of Meshoints

Meshpoint relocation is an integral part of anisotropic grid refinement. The mesh existing mesh points are moved to new location, but the number of meshpoints and structure of the are kept constant. The task to relocate meshpoints in a favourable way is a highly nonlinear optimisation problem. Furthermore is the the accuracy limited by the number of grid points in the initial mesh. Methods about grid-point relocation are covered in [JC96, Dol98, Mah02].

Mesh Refinement

Adaptive mesh refinement is more elaborate in two dimensions. The triangles selected for refinement are split into four triangles. However, if neighbouring triangles where not also selected to be splitted, there is a hanging node. To avoid the hanging node the neigbouring triangle is split in two, see figure 3.8. This is alone is sufficient to iteratively generate meshes for convergend solutions as triangles degenerate if they are

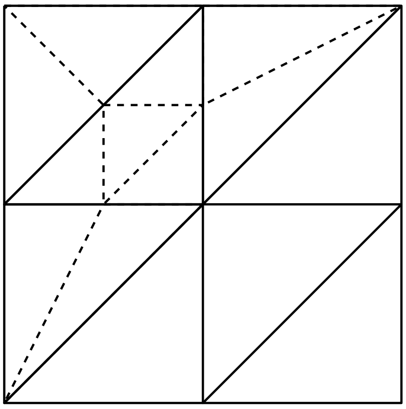


Figure 3.8: Mesh refinement in 2D, triangles marked for refinement and are split in four. Neighbouring trinangles have to be split as well to avoid hanging nodes.

split multiple times in four. There are two options to avoid degeneracy, either one remerges half triangles and split them into four or one always splits the longest side of a triangle to be halved as well.

Algorithm 3 Adaptive FEM Refinement

```
1. For each triangle marked for refinement
    (a) merge if triangle is a half triangle
    (b) split (merged) triangle into four
    (c) update hanging node list
2. While there are ordinary triangles with two hanging nodes or half triangles with one hanging node
    (a) merge if triangle is a half triangle
    (b) split (merged) triangle into four
    (c) update hanging node list
3. For each ordinary triangle with one hanging node
    (a) split into two and mark both halfes as half triangles
```

Note that in the uniformly refined region the number of points and triangles is quadrupled, while the side lengths are just halved. Finite element methods with nth-order polynomial test functions have therefore only an order of accuracy of $(n+1)/2$. Thus the convergence speed is halved compared to one dimensional problems.

Higher Order Accurate Discretisations in 2D

As in one dimension the accuracy in two dimensions can be increased by using higher order polynomials. The arangement of points for higher order elements is depicted in 3.7.

For quadratic interpolation polynomials the test functions become

$$u = \sum u_i \phi_i(x,y) \tag{3.144}$$

$$\phi(x,y) = c_{00} + c_{10}x + c_{01}y + c_{11}xy + c_{20}x^2 + c_{02}y^2. \tag{3.145}$$

And the Vandermonde matrix is

$$A = \begin{bmatrix} 1 & x_1 & y_1 & x_1^2 & x_1y_1 & y_1^2 \\ . & . & . & . & . & . \\ 1 & x_6 & y_6 & x_6^2 & x_6y_6 & y_6^2 \end{bmatrix}. \tag{3.146}$$

The discretisation matrix setup is similar to the case with linea polynomials.

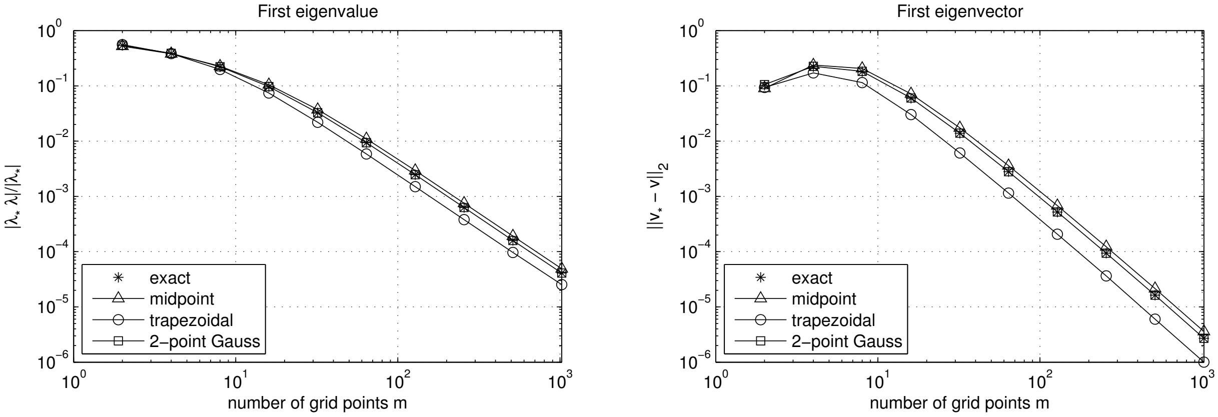


Figure 3.9: Convergence of the piecewise linear finite element method for the radial Schrödigner equation on a uniform grid. The difference between exact integration and numerical integration is negligable.

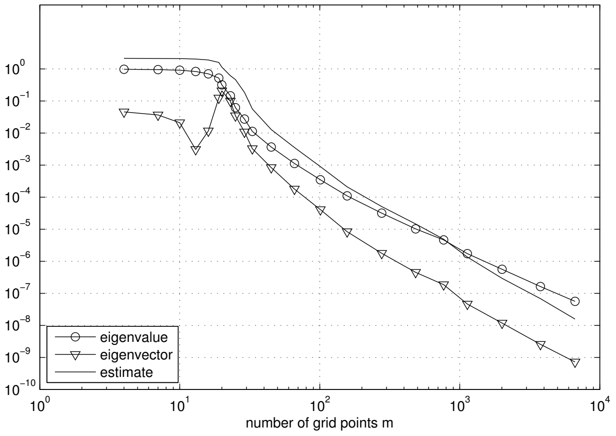


Figure 3.10: Convergence of the piecewise linear finite element method for the radial Schrödigner equation with adaptive refinement and exact integration in the domain $\Omega = [0, 400]$

3.4.3 Finite Element Discretisation of the Radial Schrödinger Equation

To practically compare the finite element and finite difference method the radial Schrödinger equation is discretised and its eigenvalues computed. The discretisation matrices are given in the appendix C.

Figure 3.4.3 shows the convergence of the finite element method for the radial Schrödinger equation on a uniform grid. One realises that numerical integration schemes do not deteriorate the convergence. The estimate computed with the trapezoidal rule is even more accurate.
An adaptively refined grid improves the accuracy drastically 3.4.3. Note that the convergence is monotoneous and that the accuracy of the approximation is higher as of that computed with the finite difference method (c.f. figure 3.2.10).

3.5 Convergence of the Finite Difference and Finite Element Method for the Hydrogenic Schrödigner Equation in Higher Dimensions

The first derivatives of the scaled hydrogen ground state (2.45) in two dimensions are

$$f = exp(-\sqrt{x^2+y^2}) \tag{3.147}$$

$$\frac{\partial f}{\partial x} = \left(\frac{-x}{(x^2+y^2)^{\frac{1}{2}}}\right) exp(-\sqrt{x^2+y^2}) \tag{3.148}$$

$$\frac{\partial^2 f}{\partial x^2} = \left(\frac{x^2}{(x^2+y^2)} + \frac{-1}{(x^2+y^2)^{\frac{1}{2}}} + \frac{x^2}{(x^2+y^2)^{\frac{3}{2}}}\right) exp(-\sqrt{x^2+y^2}) \tag{3.149}$$

$$\frac{\partial^3 f}{\partial x^3} = \left(\frac{3x-x^3}{(x^2+y^2)^{\frac{3}{2}}} + \frac{3x}{(x^2+y^2)} + \frac{-3x^3}{(x^2+y^2)^2} + \frac{-3x^3}{(x^2+y^2)^{\frac{5}{2}}}\right) exp(-\sqrt{x^2+y^2}) \tag{3.150}$$

$$\begin{aligned}\frac{\partial^4 f}{\partial x^4} = &\left(\frac{-6x^2+3}{(x^2+y^2)^{\frac{3}{2}}} + \frac{3}{(x^2+y^2)} + \frac{-18x^2+x^4}{(x^2+y^2)^2}\right. \\ &\left.+\frac{-18x^2+6x^4}{(x^2+y^2)^{\frac{5}{2}}} + \frac{15x^4}{(x^2+y^2)^3} + \frac{15x^4}{(x^2+y^2)^{\frac{7}{2}}}\right) exp(-\sqrt{x^2+y^2}).\end{aligned} \tag{3.151}$$

Assume that the error is dominated by contributions close to the Coulomb singularity. To investigate the convergence of the finite difference method one inserts the third (3.150) and fourth (3.151) derivative into the approximated error (3.78) and computes the limit towards zero.

$$\lim_{h\to 0}\frac{1}{3}h\left(\frac{3}{4h}+\frac{3}{4\sqrt{2}h^2}-\frac{1}{2\sqrt{2}}\right)+\frac{1}{12}h^2\left(\frac{1}{4}-\frac{9}{\sqrt{2}8h^3}-\frac{3}{2\sqrt{2}h}-\frac{9}{8h^2}\right)=\infty. \tag{3.152}$$

This term becomes arbitrarilly large and the finite difference method is therefore not convergend in the L_∞ norm. The finite difference method converges only if one considers the L_1 norm. In two dimensions the L_1 norm is approximated by weighting the local error with the element area h^2. This error norm vanishes close to the singularity,

$$\lim_{h\to 0} h^2\left(\frac{1}{3}h\left(\frac{3}{4h}+\frac{3}{4\sqrt{2}h^2}-\frac{1}{2\sqrt{2}}\right)+\frac{1}{12}h^2\left(\frac{1}{4}-\frac{9}{\sqrt{2}8h^3}-\frac{3}{2\sqrt{2}h}-\frac{9}{8h^2}\right)\right)=0. \tag{3.153}$$

The error of the finite element method can can be estimated similarly. The second derivative is (3.149) is inserted into the approximated error (3.140) and again the behaviour in the vicinity of zero is investigated,

$$err \leq \lim_{h\to 0} h^2 f''(h) = h^2\left(\frac{1}{2}-\frac{1}{\sqrt{2}2h}\right)=0. \tag{3.154}$$

This term vanishes at zero and one concludes that the finite element method converges for any L_p norm.

The behaviour in three dimensions is similar. Note that for the radial Schrödinger equation in one dimension one does the substitution $\tilde{f}(r) = rf(r)$ which yields a converging discretisation.

Chapter 4

Revision of Matrix Properties and Decompositions

Some fundamental matrix properties and decomposition are repeated first before some of the Hermitian eigenvalue methods are applied to the given physical problem. The interested reader may refer to a classic compendium such as [GvL96] for a more extensive coverage of this topic.

4.1 Symmetry

A matrix A is called Hermitian if it is equal to its conjugate transpose. If A is real, the conjugate transpose is simply the transpose and the matrix is called symmetric. Hermitian matrices expose certain favourable properties in regard to their decompositions,

$$A = A^*, A \in \mathbb{C} \tag{4.1}$$

$$A = A^T, A \in \mathbb{R}. \tag{4.2}$$

4.2 Orthogonality

A matrix U is called unitary, if its conjugate transpose is equal to its inverse. If real unitary matrix Q is more commonly referred to as orthogonal or orthonormal,

$$U^*U = I, U^* = U^{-1}, U \in \mathbb{C} \tag{4.3}$$

$$Q^TQ = I, Q^T = Q^{-1}, Q \in \mathbb{R}. \tag{4.4}$$

4.3 Similarity

The matrices A and B are similar, if there exists an invertible matrix of same rank P, such that

$$AP = PB. \tag{4.5}$$

Similar matrices have the same eigenvalues. If A and B are Hermitian, then P is unitary.

4.4 Projectors

Any projector P is an idempotent matrix:

$$P^2 = P. \tag{4.6}$$

The complementary projector is given by

$$P_c = I - P. \tag{4.7}$$

A projector which is also a symmetric matrix is an orthogonal projector, but it is usually not an orthogonal matrix,

$$P^T = P. \tag{4.8}$$

For each vector x there exists a projector P, such that

$$Px = x \tag{4.9}$$
$$P_c x = 0. \tag{4.10}$$

This projector is given by the outer product of x:

$$P = xx^T \tag{4.11}$$
$$P_c = I - xx^T. \tag{4.12}$$

4.5 Orthogonal Transformation

An orthogonal transformation is a transformation with an linear operator T, which preserves angles between vectors and lengths of vectors (4.13). The most common orthogonal transformations are rotations and mirroring,

$$v^T w = (Tv)^T (Tw). \tag{4.13}$$

4.6 Cholesky Decomposition

The Cholesky decomposition is a factorisation of a matrix A into the product of an upper tridiagonal L matrix and its conjugate transpose. It is one of the two possible interpretations of a matrix square root. The Cholesky factorisation only exists if the matrix A is positive definite,

$$L^T L = A. \tag{4.14}$$

4.7 Eigenvalue Decomposition

Each matrix A acts on a distinct set of vectors, its eigenvectors v, as if it were a scalar. The corresponding scalars are referred to as the eigenvalues λ and a pair of an eigenvector and its eigenvalue is referred to as an eigenpair. An eigenvector is not unique as it is only unanimously defined up to an arbitrary coefficient,

$$Av = \lambda v. \tag{4.15}$$

The eigenvalue decomposition of A is defined as

$$A = V \Lambda V^{-1}. \tag{4.16}$$

Where Λ is a diagonal matrix consisting of the eigenvalues λ of A and V is the matrix consisting of the eigenvectors corresponding to the eigenvalues in Λ.

If A is Hermitian, then V is orthogonal and Λ is real.

4.8 Generalised Eigenvalue Decomposition

A generalised eigenproblem is of the form

$$Ax = \lambda Bx. \tag{4.17}$$

Such problems arise in many physical applications such as mass spring systems which discretisation yields a mass and a stiffness matrix. The finite element discretisation of the Schrödinger equation also yields a generalised Hermitian eigenvalue problem.

Note that equation (4.17) can usually be transformed into a standard eigenvalue problem by inverting B,

$$B^{-1}Ax = \lambda x. \tag{4.18}$$

However if A and B are Hermitian, the symmetry is usually lost and $B^{-1}A$ is not Hermitian. If B is positive definite a Cholesky decomposition is preferable instead which keeps the eigenvalue problem Hermitian,

$$\begin{aligned} L^T L &= B & (4.19)\\ L^{-1}ALy &= \lambda y & (4.20)\\ x &= Ly. & (4.21)\end{aligned}$$

However, the Cholesky factor L is dense, as long as the matrix B is not close to being diagonal. Therefore specialised solvers for generalised eigenvalue problems exists which exploit symmetry as well as sparsity of the matrices A and B.

4.9 Determinant

The determinant is defined as the product of the eigenvalues λ of the matrix A,

$$det(A) = \prod_{i=1}^{n} \lambda_i. \tag{4.22}$$

4.10 Hessenberg Decomposition

The Hessenberg decomposition is defined as

$$AQ = QH. \tag{4.23}$$

Where the matrix H is upper triangular with one additional non-zero sub-diagonal and the matrix Q is orthogonal. Note that A and H are similar and hence have the same eigenvalues.

If A is Hermitian then H is tridiagonal symmetric and real,

$$AQ = QT, A = A^*. \tag{4.24}$$

4.11 QR-Decomposition

The QR-decomposition is defined as

$$A = QR. \tag{4.25}$$

Where the matrix Q is orthogonal and the matrix R upper triangular.

If A is tridiagonal so is R.

4.12 Similarity Transform into a Symmetric Matrix

Factors of asymmetric tridiagonal matrices are in general not tridiagonal anymore. This increases the computational effort and the round off error in some algorithms which build on repeated matrix factorisations. However, under certain circumstances it is possible to transform an asymmetric tridiagonal matrix T into a symmetric tridiagonal matrix $\tilde{T}$ with help of a diagonal matrix D.

For example, finite difference discretisations on a variable grid yield asymmetric matrices which can be transformed to become symmetric. In this case it is also possible to derive the matrices D and $\tilde{T}$ directly without actually performing the factorisation. See [TSCH90],

$$\begin{aligned} DTD^{-1} &= \tilde{T} \qquad (4.26) \\ d_1 &= 1 \qquad (4.27) \\ d_{i+1} &= d_i \sqrt{b_{i+1}/c_{i+1}} \qquad (4.28) \\ b_i/c_i &> 0 \qquad (4.29) \end{aligned}$$

$$\begin{bmatrix} d_1 & \\ & d_2 \end{bmatrix} \begin{bmatrix} a_1 & b_2 \\ c_2 & a_2 \end{bmatrix} \begin{bmatrix} d_1 & \\ & d_2 \end{bmatrix}^{-1} = \begin{bmatrix} a_1 & \tilde{b}_2 \\ \tilde{b}_2 & a_2 \end{bmatrix}. \tag{4.30}$$

4.13 Orthogonalising a Set of Vectors - The Modified Gram-Schmidt Process

The Gram-Schmidt algorithm orthogonalises a vector x with respect to an existing set of orthonormal basis vectors Q. Orthogonalisation can be achieved by an orthogonal projection (4.31),

$$x = (I - QQ^T)x. \tag{4.31}$$

An improved version which reduces round off errors is listed in algorithm 4.

Algorithm 4 Modified Gram-Schmidt Algorithm

```
$Q$ : orthonormal $m \times n$ matrix
$x$ : arbitrary $m \times 1$ vector
for $k = 1, 2, ..., n$
        $x = x - (q_k^T x) q_k$
end
$x = x / ||x||_2$
```

4.14 Least Squares Solution

If a matrix A is overdetermined, e.g. if it is not square such that it has more unique rows than colums, then there exists no solution to the linear system

$$Ax = b \tag{4.32}$$

However, such a system arises for example in the MINRES 6.2.8 method. Note that there still exists an x which minimises the residual r,

$$r = Ax - b. \tag{4.33}$$

The least squares solution to an overdetermined system is the one which minimises the residual in the L_2 norm

$$\min_x ||Ax - b||_{L_2}. \tag{4.34}$$

The minimum is found as the zero of the derivative of the residual norm

$$\begin{aligned} ||r||_{L_2} &= r^T r \qquad (4.35) \\ &= A^T A x - 2A^T b + b^T b \qquad (4.36) \\ 0 &= A^T A x - A^T b. \qquad (4.37) \end{aligned}$$

The normal equation (4.37) is quadratic and has therefore a unique solution. As the condition number is squared in the normal equation compared to the original system, one solves it with an QR factorisation to avoid a loss of accuracy,

$$\begin{aligned} A &= QR \qquad (4.38) \\ 0 &= (QR)^T QRx - (QR)^T b \qquad (4.39) \\ x_{min} &= R^{-1} Q^T b. \qquad (4.40) \end{aligned}$$

4.15 Kronecker Product

The Kronecker product forms the block matrix C of dimension $m_c \times n_c$ by scaling the matrix B for each element a_{ij} in A and writing it to the respective block in C.

$$C_{m_c \times n_c} = A_{m_a \times n_a} \otimes B_{m_b \times n_b}. \tag{4.41}$$

Where $m_c = m_a \cdot m_b$ and $n_c = n_a \cdot n_b$.

```
for $i = 1 : m_a$
    for $j = 1 : n_a$
        $C_{(i-1)\cdot m_b+1:i\cdot m_b,(j-1)\cdot n_b+1:j\cdot n_b} = a_{i,j} \cdot B$
    end
end
```

4.16 Norms

Throuhghout this literature study the $L-p$ norms are extensively used. The $L-p$ norms are defined as

$$||u(x)||_p = \sqrt{p}{\sum |f(x)|^p} \tag{4.42}$$

and in case of a discretised vector,

$$||u||_p = \sqrt{p}{\sum |u|^p}. \tag{4.43}$$

Of special interest is the case $p = 2$, the Euclidean norm, where the $L-p$ norm becomes the inner product

$$||u||_2^2 = u^T u. \tag{4.44}$$

For the finite element analysis one usually consideres the generalised inner product, yielding the energy norm,

$$||u||_B^2 = u^T B u. \tag{4.45}$$

Where the matrix B is positive definite. The matrix B is usually a differential operator and uBu^T contains the weighted sum of squares of the functions and its derivatives.

Norms satisfy certain properties, e.g. they are positive definite and the triangle inequality holds. If the matrix B is positive indefinit, one refers to its inner product as a seminorm.

Note that the operator A of the hydrogenic Schrödinger equation is indefinite with positive and negative eigenvalues and its inner product does not is not represent a norm.

Chapter 5

Basic Eigenvalue Algorithms

The eigenvalue problem can be formulated as finding the zeros of an n-degree polynomial, where n is the rank of the matrix A,

$$(A - \lambda I)x = 0. \tag{5.1}$$

The Abel-Ruffini theorem states that there exists no closed form solution in the general case if the polynomial is of order five or larger. Hence it is in general impossible to find the exact eigenvalues of a matrix in a finite number of steps and it is necessary to apply an iterative method to approximate the eigenvalues.

For the finite element method the generalised eigenvalue problem is of special interest, Algebraic Eigenvalue Problem

$$(A - \lambda B)x = 0. \tag{5.2}$$

Where B is a positive definite matrix.
The eigenvalue problem can be reformulated as a minimisation problem, which reads find the nonzero vector x which minimises the Rayleigh quotient λ_1,

$$\lambda_1 = min\frac{x_1^T A x_1}{x_1^T B x_1}, x \neq 0. \tag{5.3}$$

The other eigenvalues of the hermition eigenvalue problem are subsequently found by requiring the vectors x_k to be be orthogonal to the vectors already computed,

$$\lambda_k = \frac{x_k^T A x_k}{x_k^T B x_k}, x_k \perp x_1, ...x_{k-1} \tag{5.4}$$

5.1 Power Iteration

The power iterations is the most basic method for computing eigenvalues. It is slowly convergent and only yields the largest eigenvalue. It is stated here as most advanced eigenvalue algorithms can be retraced to a form of power iteration.

Assume that the matrix A a has a unique real eigenvalue whose absolute value is larger than the absolute values of the other eigenvalues,

$$|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq ... \geq |\lambda_n|. \tag{5.5}$$

For simplicity, but not necessity, consider the matrix A also to be symmetric. Then one realises that the powers of the matrix A converge of the powers of the eigenpair of λ_1,

$$A^k x_0 = V\Lambda^k V^T x_0 = \lambda_1^k v_1 v_1^T x_0 + \lambda_2^k v_2 v_2^T x_0 + ... + \lambda_n^k v_n v_n^T x_0. \tag{5.6}$$

The idea of the power method, given by algorithm 5 is to indirectly compute the matrix powers by iteratively forming a matrix vector product.

Algorithm 5 Power Iteration

```
$x_0 =$ random vector
for k=0,1,2,...
    $x_{k+1} = Ax_k$
    $\lambda_{k+1} = ||x_{k+1}||_2$
    $x_{k+1} = \frac{1}{\lambda_{k+1}} x_{k+1}$
    if $|\lambda_{k+1} - \lambda_k| < tol$
        $\lambda_{k+1} = x_k^T x_{k+1}$ (get correct sign of $\lambda$)
        break
```

5.2 Inverse Iteration

The inverse iteration is a variant of the power method, which allows to find eigenvalues and eigenvectors close to a value called shift μ. This method can be used to find eigenvectors for known eigenvalues and converges quicker than the power method at the cost of the solution of a linear system at each iteration, see algorithm 6. The shift μ can be updated during the iteration with the vector norm σ for faster convergence.

Algorithm 6 Inverse Iteration

```
$x_0$ = random vector
$\sigma_0$ = user defined shift
for k = 0,1,2,...
    $(A - \sigma_k I)x_{k+1} = x_k$
    $\lambda_{k+1} = \frac{1}{x_k^T x_{k+1}} + \sigma_k$
    $x_{k+1} = \frac{1}{x_{k+1}^T x_{k+1}} x_{k+1}$
    $\sigma_{k+1} = \lambda_{k+1}$ (optional shift update)
    if $|\lambda_{k+1} - \lambda_k| < tol$
        break
```

5.3 Simultaneous Iteration

Both the power iteration and the inverse iteration compute just a single eigenpair. It is possible to compute several eigenpairs simultaneously by using a block of initial vectors. However, the orthogonality of those vectors has to be ensured at every iteration, such that they converge to different eigenvectors. This can be achieved by QR-factorisation.

Algorithm 7 Simultaneous Iteration

```
    $Z_0 =$ random $n \times m$ matrix, $m < n$
    $Q_0 R_0 = Z_0$
    for k=1,2,...
        $Z_k = AQ_{k-1}$ (or $AZ_k = Q_{k-1}$ for inverse iteration)
        $Q_k R_k = Z_k$
```

5.4 QR-Algorithm

The QR-Algorithm is such an iterative method to compute the eigenvalue decomposition of the matrix A [Fra61]. The inner loop of the QR-algorithm with shift μ is given by algorithm 8.

If A is Hermitian then the off diagonal entries in A_k converge always to zero and A_k approaches the diagonal matrix of eigenvalues Λ [1]. $\tilde{Q}_k$ converges simultaneously to the matrix of eigenvectors V. However, it is

[1] For asymmetric matrices the matrix A_k converges to the Schur-factor of A, if A does not have multiple or complex eigenvalues.

Algorithm 8 QR-Algorithm

```
for i=n,n-1,...,2
    while $a_{i,i-1} > tol$
        $Q_k R_k = (A_k - \sigma_k I)$
        $A_{k+1} = R_k Q_k + \sigma_k I$
        $\sigma_{k+1} = a_{i,i}$
        $\tilde{Q}_k = \tilde{Q}_k Q_k$ (optional eigenvector computation)
    end
    deflate the matrix $A_k$ by removing the last row and column
end
```

not required to compute this matrix explicitly,

$$\begin{aligned} A_{k+1} &= Q_k^T A_k Q_k &(5.7)\\ &= Q_k^T ... Q_1^T A_1 Q_1 ... Q_k &(5.8)\\ &= \tilde{Q}_k^T A_1 \tilde{Q}_k &(5.9)\\ V \Lambda V^T &\approx \tilde{Q}^T A_1 \tilde{Q}. &(5.10)\end{aligned}$$

The lowest right element of A_k converges most rapidly. This permits to deflate the matrix by its last row and column each time an eigenvalue has converged. The iteration is then continued with the deflated matrix $\tilde{A}_k$. If the the lowest right element of the matrix A_k is chosen as shift, then this element converges cubically to an eigenvalue of A if A is Hermitian,

$$A_k = \begin{pmatrix} \tilde{A}_k & 0 \\ 0 & \lambda \end{pmatrix}. \tag{5.11}$$

A robust implementation of the QR-algorithm must also handle matrices with multiple and complex eigenvalues. This requires a different shift strategy, which is to choose the eigenvalue of the lower right 2x2 submatrix closest to to the lower right element. Complex arithmetic can be avoided by using a block-QR factorisation computing an even number of QR-factorisations simultaneously.
A QR-decomposition can be computed using the modified Gram-Schmidt algorithm in $O(n^3)$ steps. However in the QR-algorithm the explicit computation of Q can be avoided and if A is tridiagonal, then a single QR-iteration step can be computed in $O(n)$ steps.

The QR-Algorithm is numerically stable, in a sense that round off errors do not considerably accumulate during the iterations and compromise the computed eigenvalues.

5.4.1 Relation with the Power Iteration and Inverse Iteration

The QR-algorithm can be interpreted as a combination of the power iteration and inverse iteration [PP73]. If no shift is applied then the iteration (equation (5.9)) may be rewritten as follows:

$$\begin{aligned} \tilde{Q}_{k-1} A_k &= A_1 \tilde{Q}_{k-1} &(5.12)\\ \tilde{Q}_{k-1} Q_k R_k &= A_1 \tilde{Q}_{k-1} &(5.13)\\ \tilde{Q}_k R_k &= A_1 \tilde{Q}_{k-1}. &(5.14)\end{aligned}$$

By multiplying equation (5.14) with the first unit vector and considering that the R matrix is upper triangular one realises that the left part of (5.15) is just the power iteration,

$$\begin{aligned} \tilde{Q}_k &= A_1 \tilde{Q}_{k-1} R_k^{-1} &(5.15)\\ \tilde{Q}_k e_1 &= A_1 \tilde{Q}_{k-1} R_k^{-1} e_1 &(5.16)\\ \tilde{q}_1^{(k)} &= \frac{1}{r_{1,1}^{(k)}} A_1 \tilde{q_1}^{(k-1)}. &(5.17)\end{aligned}$$

If equation (5.14) is inverted and transposed before the multiplication with the first unit vector e_1, the inverse power method becomes apparent if the matrix A is symmetric,

$$R_k^{-1}\tilde{Q}_k^{-1} = \tilde{Q}_{k-1}^{-1}A_1^{-1} \tag{5.18}$$

$$\tilde{Q}_k R_k^{-T} = \tilde{A}_1^{-T}\tilde{Q}_{k-1} \tag{5.19}$$

$$A_1^T\tilde{Q}_k R_k^{-T} e_1 = \tilde{A}_1^{-T}\tilde{Q}_{k-1}e_1 \tag{5.20}$$

$$\frac{1}{r_{1,1}^{(k)}}A^T\tilde{q}_1^{(k)} = \tilde{q}_1^{(k-1)}. \tag{5.21}$$

Hence the topmost element a_{11} of A_k converges to the dominant eigenvalue λ_1 and the first column $\tilde{q}_1$ of $\tilde{(Q)}$ converges to the corresponding eigenvector v_1 of λ_1 if the convergence criteria of the power method are met,

$$\lim_{k\to\infty}\tilde{Q}_k^T A\tilde{Q}_k e_1 = \lim_{k\to\infty}\tilde{q}_1^{T(k)} r_{1,1}^{(k)} e_1\, \tilde{q}_1^{(k)} \tag{5.22}$$

$$= v_1^T\lambda_1 v_1 e_1 = \lambda_1 e_1. \tag{5.23}$$

During the QR iteration all the elements on the main diagonal of the R matrix converge simultaneously. It is de facto a combination of the simultaneous power iteration and the simultaneous inverse iteration. Note that above only the convergence of the first element of the R matrix was shown. However, in practise the last element of the R matrix converges first and the the deflation is carried out as shown in equation (5.11).

5.5 Eigenvalues by Bisection

If only few eigenvalues are required the bisection algorithm offers an alternative to the QR-algorithm. The bisection algorithm finds a single eigenvalue in an selected interval in $n\log(\epsilon)$ time. Recall that the determinant of a matrix is equal to the product of its eigenvalues (equation (5.24)) and that the eigenvalues of a matrix can be shifted by subtracting a multiple of the identity matrix (equation (5.25)),

$$det(A) = \prod\lambda \tag{5.24}$$

$$(A-\mu I)x = \lambda - \mu. \tag{5.25}$$

The upper principal minors of the matrix A form furthermore a Sturm sequence. This means that the number of sign changes of the determinants of the principal minors is equal to the number of negative eigenvalues. As determining the determinant of a tridiagonal matrix requires only $O(n)$ steps, one can find eigenvalues by shifting the spectrum in an interval which is successively halved in each step. A start interval can be determined by Gershgorin's circle theorem.

Algorithm 9 Bisection Algorithm

```
while ( abs($\lambda_l$ - $\lambda_u$) > tol*(abs($\lambda_l$) + abs($\lambda_u$)) && abs($\lambda_l$) + abs($\lambda_u$) > tol)
    % interval midpoint
    $\lambda_c$ = 1/2 ($\lambda_l$ + $\lambda_u$);
    n_s = 0;
    p_2 = 0;
    p_1 = 1;
    % count sign changes in upper determinant
    for idx=1:n
            $p = (A_{ii} - \lambda_c)p_1 - (A_{ii+1})^2 p_2$
            if ($p{\cdot}_1 \leq 0$)
                    n_s = n_s + 1
            end
            p_2 = p_1
            p_1 = p
    end
    % select half with sign change
    if (n_s > n-k)
        $\lambda_u$ = $\lambda_c$
    else
        $\lambda_l$ = $\lambda_c$
    end
end
```

Chapter 6

Eigenvalue Algorithms for Large Sparse Matrices

Even if a matrix where sparse the QR-algorithm becomes inapplicable once the matrix is to large. Exceptions are matrices with very small bandwidths. Two general methods are analysed in the following section which allow the computation of eigenvalues for large sparse matrices. The first approach uses the Lanczos algorithm to factorise the matrix into tridiagonal form before computing the eigenvalues. The second approach uses a correction method in form of the Jacobi-Davidson algorithm which directly finds the eigenpairs. Especially the well established Lanczos method is extensively covered in the literature [Saa11], [Par87], [CW02], [GvL96] and [BDJ$^+$00], [vdV].

6.1 Projection Methods

Projection methods approximate a matrix by its projection onto an orthogonal subspace. Orthogonal projection methods find wide application in the solution of linear systems and eigenvalue computation. The projection is chosen such that the original problem is much easier to solve with the projected matrix and that the solution approximated with the projected matrix is close to the proper solution of the original problem.

The eigenvalues of a matrix A are computed by constructing the projection matrix Q at first, then by finding the eigenvalues of the projected matrix Q^TAQ and finally by projecting the eigenvalues s of the projected system back to get approximate eigenvalues z,

$$Ax = \lambda x \tag{6.1}$$

$$Q^TAQs = \lambda s \tag{6.2}$$

$$z = Qs. \tag{6.3}$$

An intriguing property of the orthogonal projection methods is that the residual of the approximated solution r is orthogonal to the projection space,

$$r = AQs - \lambda Qs \tag{6.4}$$

$$Q^T r = 0. \tag{6.5}$$

The eigenvalues and eigenvectors of the projected system are referred to as Ritz values and Ritz vectors respectively.

6.2 Lanczos Iteration

The Lanczos iteration is a method to compute the Hessenberg decomposition of a Hermitian matrix [Lan50]. Such a decomposition precedes the QR-algorithm, as the iterations on the tridiagonal Hessian are much less costly, as shown in the previous chapter. Furthermore an incomplete decomposition can be computed such that the rank of T is much lower than the rank of A. A complete Lanczos factorisation takes $O(n^2)$

operations, if A is a banded sparse matrix with few nonzero diagonals. The computed Hessenberg matrix T computed by algorithm 10 takes the form given in equation (6.6),

$$T = \begin{bmatrix} a_1 & b_2 & & & \\ b_2 & a_2 & b_3 & & \\ & b_3 & ... & ... & \\ & & ... & a_{m-1} & b_m \\ & & & b_m & a_m \end{bmatrix}. \tag{6.6}$$

Algorithm 10 Lanczos Iteration

```
$q_0 = \vec{0}$
$q_1 =$ arbitrary nontrivial vector
$q_1 = \frac{1}{||q_1||} q_1$
for $i = 1..m$
        $q_{i+1} = Aq_i - b_i q_{i-1}$
        $a_i = q_{i+1}^T q_i$
        $q_{i+1} = q_{i+1} - a_i q_i$
        $b_{i+1} = ||q_{i+1}||$
        $q_{i+1} = \frac{1}{b_{i+1}} q_{i+1}$
end
% compute the eigenvalues of $T$ (approximate eigenvalues of $A$) by bisection
$T = S \Theta S$
% sort out spurious eigenvalues
...
% test for convergence
$r = \frac{|\beta s_{i,j}|}{||Q s_i||}$
% approximated eigenvectors by inverse iteration
...
```

6.2.1 The Lanczos-Iteration as Eigenvalue Method

The eigenvalues θ of T equal the eigenvalues λ of A upon complete factorisation with exact arithmetic as both matrices are similar. The same holds for the Ritz vectors z and eigenvectors v. If the factorisation is computed with finite precision or incomplete , i.e. only performed until step k, then the factorisation has an residual $r^{(k)}$ (equation (6.8)),

$$AQ_k - Q_k T_k = r^{(k)} e_k^T \tag{6.7}$$

$$= \beta_{k+1} q_{k+1} e_k^T \tag{6.8}$$

$$T_k = S_k \Theta_k S_k^T \tag{6.9}$$

$$Z_k = Q_k S_k. \tag{6.10}$$

Implementations of the Lanczos algorithm vary depending on when the eigenvalues are calculated. Parlett [Par87] recommends to calculate the eigenvalues after each iteration. However, this becomes expensive if many iterations are required to achieve convergence. Hence Cullum [CW02] recommends almost a complete factorisation before the eigenvalues are calculated.

6.2.2 Convergence Estimation

The residual $r^{(k)}$ of the factorisation also leads to an residual $r_i^{(k)}$ of the approximate eigenvectors $z_i^{(k)}$ (equation (6.14)),

$$AQ_k = Q_k T_k + r^{(k)} e_k^T \tag{6.11}$$

$$AQ_k = Q_k S_k \Theta_k S_k^T + r^{(k)} e_k^T \tag{6.12}$$

$$(A - \theta_i^{(k)} I) Q_k s_i^{(k)} = \beta_{k+1} q_{k+1} e_k^T s_i^{(k)} \tag{6.13}$$

$$(A - \theta_i^{(k)}) z_i^{(k)} = r_i^{(k)}. \tag{6.14}$$

The residual norm of the Ritz vectors $||r_i^{(k)}||$ is therefore determined only by the last coefficient β_{k+1} and the last element of the respective eigenvector $s_i^{(k)}$ of the matrix T_k (equation (6.15)). The error of the approximate eigenvalue can be estimated by the residual norm and the distance γ_i to the closest eigenvalue (equation (6.16)). Hence only the eigenvalues of the tridiagonal system, but not the Ritz-vectors themselves are required to estimate the convergence [PS79],

$$||r_i^{(k)}|| = \beta_{k+1} s_{i,k} \tag{6.15}$$

$$|\theta_i^{(k)} - \lambda_i| \approx (\beta_{k+1} s_{i,k})^2 / \gamma_i \tag{6.16}$$

$$\gamma_i = min_{i \neq j} |\theta_i^{(k)} - \theta_j^{(k)}|. \tag{6.17}$$

The Lanczos method usually requires many iterations m before the eigenvalues converge. Sometimes the number of iterations exceeds the matrix rank n, [CW81]. The eigenvalues in the required range are found by bisection, due to the large size of the projected system. The respective eigenvectors can be computed by inverse iteration afterwards.

6.2.3 Invariant Subspaces

If the matrix A contains at least one eigenvalues multiple times, i.e. if its spectrum is degenerated, then after some iterations the Lanczos has explored a complete invariant subspace. As a consequence the eigenvalues of the subspace are not only approximations but the exact eigenvalues of A as their residual βs_{ij} is zero up to rounding errors. However, the iteration cannot continue and some modifications are required to find the remaining eigenvalues. There are three possibilities:

- Restarting the Lanczos iteration with a start vector orthogonal to the converged eigenvectors. However, this requires the storage of the complete subspace.

- Using a block-version of the Lanczos algorithm starting with a set of orthogonal vectors. However, this produces a banded matrix with more than three diagonals and requires to deflate the block once a residual becomes zero.

- Discretising the PDE, such that it is not degenerated and hence does not contain multiple eigenvalues.

Even if the matrix does not contain invariant subspaces, a similar behaviour is observed once the eigenvalues of T start to converge.

6.2.4 Loss of Orthogonality and Spurious Eigenvalues

As the Lanczos algorithm is only based on a three term recurrence, orthogonality of the basis vectors Q is soon lost in finite precision arithmetic,

$$Err = Q^T Q - I. \tag{6.18}$$

As a consequence the matrices A and T are not anymore similar and have a different set of eigenvalues. However, the eigenvalues sets are not completely different, but

- The matrix T contains spurious eigenvalues. A spurious eigenvalue is an eigenvalue of the tridiagonal matrix T which is not an eigenvalue of the original matrix A. Spurious eigenvalues usually occur directly after one of the eigenvalues of T has converged to an eigenvalue of A. Thereafter the spurious eigenvalue converges to a copy of one of the already converged eigenvalues.

- The eigenvalues of A are still eigenvalues of the matrix T. To obtain all eigenvalues of A it is necessary to continue the iterations, such that the rank of T exceeds the rank of A.

Several methods have been developed to overcome the shortcomings of the Lanczos method and to compute the correct eigenvalues. There are basically two ideas, the first idea is to restore the orthogonality [PS79] and the second idea is to identify and remove spurious eigenvalues from the result, see [Pai80].

6.2.5 Partial Reorthogonalisation

A possible remedy is to keep track of the loss of orthogonality by a recurrence formula and to reorthogonalise the last two basis vectors against the complete basis [Sim84],

$$\max_{ij}(Q^TQ - I) > \sqrt{\epsilon}\kappa(A). \tag{6.19}$$

Although the partial reorthogonalisation is less expensive than a complete reorthogonalisation, it may become infeasible for large subspaces. It can be shown that the eigenvalues of the partially reorthogonalised matrix are as accurate as if they were computed with full orthognonalisation, as long as equation (6.19) is satisfied.

6.2.6 Filtering Spurious Eigenvalues

Paige [Pai80] showed that all the eigenvalues of A will eventually be contained in T, if the iteration is continued long enough. Note that this could yield a matrix T which is larger than A.

Spurious and truly multiple eigenvalues could be distinguished by testing the orthogonality of the approximate eigenvectors. The true multiplicity of the eigenvalue is equal to the number of approximate eigenvectors who's inner product is close to zero,

$$v_i^T v_j = (Qs_j)^T Qs_i, i \neq j. \tag{6.20}$$

However, this method can not be applied in practise as it requires to store the complete basis Q.

Cullum [CW81] described how one may sort out the spurious eigenvalues by comparing the eigenvalues of T to the eigenvalues of the matrix T without first row and column T_2.

- Keep a single copy of θ_i as converged eigenvalue, if is multiple eigenvalue of T.
- Keep θ_i as approximation if it is a single eigenvalue of T and not an eigenvalue of T_2.
- Discard θ_i as spurious if it is single eigenvalue of T and also eigenvalue of T_2.

6.2.7 Lanczos Iteration for Generalised Eigenvalue Problems

The Lanczos method is adapted for generalised eigenproblems by forming the basis Q such that it is orthogonal with respect to its generalised inner product with B,

$$Ax = \lambda Bx \tag{6.21}$$

$$AQ = BQT \tag{6.22}$$

$$Q^TBQ = I. \tag{6.23}$$

Algorithm 11 is the Lanczos iteration adapted for the generalised eigenvalue problem. The most obvious difference is the solution of a linear system at every iteration step. This can of course be formed with with an iterative solver such as the conjugate gradient or minimum residual method [PS75]. Note that in the actual implementation the matrix vector products of BQ are not explicitly formed but kept in an auxiliary matrix.

Algorithm 11 Generalised Lanczos Iteration

```
$q_0 = \vec{0}$
$q_1 =$ arbitrary nontrivial vector
$q_1 = \frac{1}{\sqrt{q_1^T B q_1}} q_1$
for $i = 1..m$
        $q_{i+1} \; = A q_i$
        $B q_{i+1} \; = q_{i+1}$
        $q_{i+1} \; = q_{i+1} - b_i q_{i-1}$
        $a_i \qquad = q_{i+1}^T B q_i$
        $q_{i+1} \; = q_{i+1} - a_i q_i$
        $b_{i+1} \; = \sqrt{q_{i+1}^T B q_{i+1}}$
        if $(b_{i+1} < \epsilon)$ break
        $q_{i+1} \; = \frac{1}{b_{i+1}} q_{i+1}$
end
% compute the eigenvalues of $T$ (approximate eigenvalues of $A$) by bisection
$T = S \Theta S$
% sort out spurious eigenvalues
...
% test for convergence
$r = \frac{|\beta s_{i,j}|}{||Q s_i||}$
% approximated eigenvectors by inverse iteration
...
```

6.2.8 Approximate the Solution of Linear Systems

Some of the eigenvalue methods for large sparse matrices introduced in this chapter require at each iteration the solution of a symmetric linear system of the form

$$Ax = b. \tag{6.24}$$

For large sparse systems it is reasonable to approximate the solution iteratively instead to compute it directly. The most elementary iterative solver for symmetric positive definite system is the conjugate gradient method (CG) [HS52]. However it is not applicable to indefinite systems [FM84] and hence not suited to compute the indefinit spectrum of the confined hydrogen atom. An iterative solver applicable to indefinite systems is the minimum residual method (MINRES), [PS75]. The key of this method is to factorise the matrix A into a tridiagonal matrix T and an orthogonal basis Q (equation (6.26)). The factorisation is computed usign the Lanczos method (algorithm 10).

$$\begin{aligned} AQ_k &= Q_k T_k + \beta_{k+1} q_{k+1} e_k^T && (6.25) \\ &= Q_{k+1} \tilde{T}_k && (6.26) \end{aligned}$$

The residual r at step k is given by the relation

$$\begin{aligned} r^{(k)} &= Ax^{(k)} - b && (6.27) \\ &= AQ_k y^{(k)} - b && (6.28) \\ &= Q_{k+1} \tilde{T}_k y^{(k)} - b. && (6.29) \end{aligned}$$

Where y is the projected solution

$$x^{(k)} = Q_k y^{(k)}. \tag{6.30}$$

If one chooses the initial vector as $q_1 = {}^1\!/_{\beta_1} b$ with $\beta_1 = ||b||_2$, then the residual and the residual norm simplify to

$$\begin{aligned} r^{(k)} &= V_{i+1}(\tilde{T}_k y^{(k)} - \beta_1 e_1) && (6.31) \\ ||r^{(k)}||_2 &= ||\tilde{T}_k y^{(k)} - \beta_1 e_1||_2 && (6.32) \end{aligned}$$

The projected solution y is found by minimizing the risidual, e.g. by solving the least squares problem (cf. 4.14)

$$\tilde{T}_k y^{(k)} = \beta_1 e_1. \tag{6.33}$$

The QR-factors of $\tilde{T}_i$ for solving the least squares problem can be computed iteratively by Given's rotations.

6.2.9 Numerical Experiment

The convergence of eigenvalues approximated by the Lanczos method depend strongly on the distribution of the eigenvalues, see figure 6.1. For the two dimensional Laplacian convergence sets in almost immediately and a new eigenvalue is found on average at every other iteration. However, in case of the one dimensional Laplacian none of the eigenvalues converges until the matrix is completely factorised, see figure 6.1.

The Lanczos routine is tested on a Laplacian operator for the one dimensional domain 1x250 and the two dimensional domain 10x25. The eigenvalues are compared to the analytic solutions and are accepted as converged if the error $|\lambda_i - \theta_i|$ is less than the square root of the machine precision ϵ. The start vector consists of random elements.

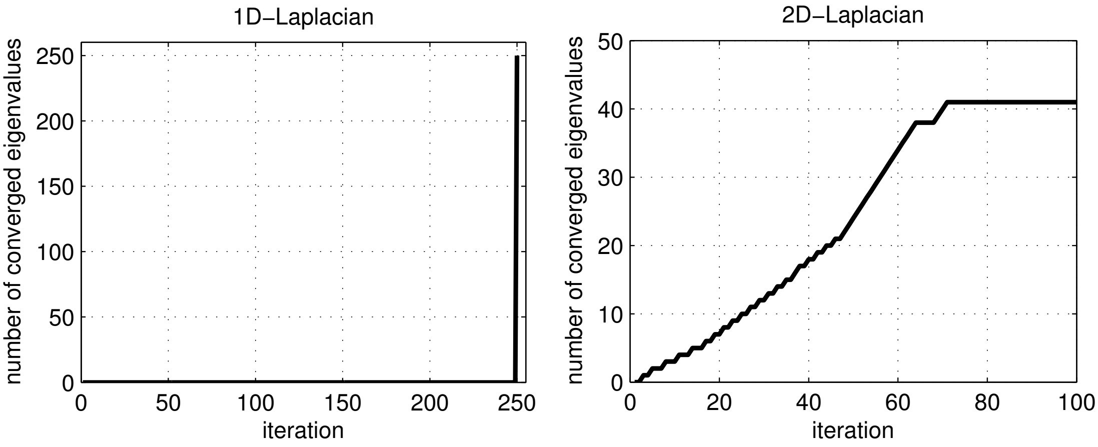


Figure 6.1: Convergence of eigenvalues approximated with the Lanczos method. Acceptance tolerance $\sqrt{\epsilon} \approx 10^{-7}$, see equation (6.16).

6.3 The Jacobi-Davidson Method

The Jacobi-Davidson method is best understood if one considers the Davidson method first.

6.3.1 The Davidson Method

Other than the Lanczos-Iteration, the Davidson method computes approximate eigenvectors v and eigenvalues θ directly without factorising the matrix A [Dav75]. It does so by constructing an orthogonal basis constituting of successive corrections to an initial vector (equation (6.39)). The correction step is a simplified inverse iteration step where the matrix A is approximated by its diagonal D. Therefore the computational costs per iteration are just of order $O(n)$. However, the algorithm converges only for matrices with dominant diagonal and usually requires a restart after some iteration to keep the basis at a tractable size. The core algorithm consists of three steps: calculating the residual (equation (6.38)), correcting the current approximation (equation (6.39)) and orthogonalising the new search vector against the existing search space (equation (6.40)). The complete algorithm is listed as algorithm 12,

$$M = V^T AV \tag{6.34}$$
$$S\Theta S^T = M \tag{6.35}$$
$$x = Vs_1 \tag{6.36}$$
$$r = Ax - \theta_1 x \tag{6.37}$$
$$r^{(k)} = (A - \theta_k I)x_k \tag{6.38}$$
$$(D - \theta_k I)v_{k+1} = -r^{(k)} \tag{6.39}$$
$$v_{k+1} = (I - VV^T)v_{k+1}. \tag{6.40}$$
$$v_{k+1} = \frac{1}{\sqrt{v_{k+1}^T v_{k+1}}} v_{k+1}. \tag{6.41}$$

Algorithm 12 Davidson Algorithm

```
% guess the initial search space (random vector or few Arnoldi steps)
V = ...
% orthogonalise the initial search space
$V = orth(V)$
$k = 1$;
while (1)
    % expand projected system
    $AV_{:,k} = AV_{:,k}$
    $H_{1:k,k} = V_{:,k}^T AV$
    $H_{k,1:k} = H_{1:k,k}^T$
    % BV and G are only updated in case of GHEP
    $BV_{:,k} = BV_{:,k}$
    $G_{1:k,k} = V_{:,k}BV$
    $G_{k,1:k} = G_{1:k,k}$
    % find approximate eigenvalue
    [y $\theta$] = eigs($H_{1:k,1:k}$, $G_{1:k,1:k}$, 1,'LA');
    % approximated eigenvector
    $x = Vy$;
    % residual ($r = Ax - \theta Bx$)
    $r = AVy - \theta BVy$
    % check for convergence
    if (||r|| < abstol)
        return;
    end
    % expand search space
    $V_{:,k+1}$ = minres($D - \theta I$, r);
    % orthogonalise by modified Gram–Schmidt
    $V_{:,k+1} = mgs(V_{:,1:k}$, $V_{:,k+1}$, k);
    k = k+1;
end % while 1
```

6.3.2 Jacobi-Davidson Algorithm

The Jacobi-Davidson [SvdV00] method combines the Davidson method with an idea from Jacobi, which is to search for corrections only in an space orthogonal to the current vector,

$$A(x + \delta x) = \lambda(x + \delta x) \tag{6.42}$$
$$x^T \delta x = 0. \tag{6.43}$$

This space is determined by the projector,

$$P = I - XX^T. \tag{6.44}$$

Thus the Jacobi-Davidson algorithm is identical to the Davidson method beside the correction step, which solves

$$\begin{aligned}
M &= V^T AV && (6.45)\\
S\Theta S^T &= M && (6.46)\\
x &= Vs_1 && (6.47)\\
Q &= [V, x] && (6.48)\\
r &= Ax - \theta_1 x && (6.49)\\
(I - QQ^T)(A - \theta_1 I)(I - QQ^T)v_{k+1} &= -(I - QQ^T)r && (6.50)\\
v_{k+1} &= (I - VV^T)v_{k+1} && (6.51)\\
v_{k+1} &= \frac{1}{\sqrt{v_{k+1}^T v_{k+1}}} v_{k+1}. && (6.52)
\end{aligned}$$

The correction equation is singular and has to be solved approximately. If the correction equation is solved exactly, the method converges cubically.

6.3.3 The Jacobi-Davidson Method for the Generalised Eigenvalue Problem

The extension of the Jacobi-Davidson method to generalised Hermitian eigenvalue problem requires three major changes. Firstly, the basis V is orthogonalised with respect to the generalised inner product with B,

$$V^T BV = I. \tag{6.53}$$

The matrix B is furthermore incorporated into the calculation of the residual. And thirdly the correction equation changes as follows:

$$\begin{aligned}
M &= V^T AV && (6.54)\\
S\Theta S^T &= M && (6.55)\\
x &= Vs_1 && (6.56)\\
Q &= [V, AVs_1] && (6.57)\\
Z &= [V, BVs_1] && (6.58)\\
r &= Ax - \theta_1 Bx && (6.59)\\
(I - Q^T Z)(A - \theta_1 B)(I - Q^T Z)v_{k+1} &= -(I - QZ^T)r && (6.60)\\
v_{k+1} &= (I - VV^T B)v_{k+1} && (6.61)\\
v_{k+1} &= \frac{1}{\sqrt{v_{k+1}^T Bv_{k+1}}} v_{k+1}. && (6.62)
\end{aligned}$$

Again the matrix-matrix products AV and BV are stored in auxiliary memory to avoid unnecessary recalculation.

Chapter 7

Available Software

As finding eigenvalues and eigenvectors is a standard task with many applications a variety of implementations exist.

7.1 LAPACK

The Linear Algebra PACKage, short LAPACK is a well established software suite. It provides basic eigenvalue routines such as the QR algorithm. However, it does not include specialist solvers for large sparse eigenproblems. LAPACK is released under the BSD license and is accessible in several ways

- original documentation and software from netlib [UoTUoCL], the latest stable release as of November 2011 is 3.4.0
- in most Linux distributions as precompiled package optimised for current processors, for example in Debian 6.0.4 (Squeeze) the package groups liblapack* and libatlas*
- in Matlab/Octave integrated and accessible via the command *eig*

There exist furthermore offshoots which offer support for SMP, GPU or quad precision computing.

7.2 ARPACK

The ARnoldi PACKage, short ARPACK provides solvers for large sparse eigensystems. It implements the restarted Arnoldi and Lanczos methods. It does not perform any matrix operations directly, but requires respective routines to be provided elsewise. ARPACK was originally released by the Rice University [LDC97], but is not maintained anymore. An up to date version named Arpack New Generation (arpack-ng) is provided by an open source initiative [Led11]. arpack is like LAPACK a standard package in most Linux distributions and is integrated in Matlab/Octave integrated as the command *eigs*.

7.3 Lanczos Algorithms for Large Symmetric Eigenvalue Computations

The book by Cullum covering extensively Eigenvalue computations with the Lanczos method [CW02] is accompanied by a second volume consisting of source code listings. The second volume and the source can also be downloaded at netlib [CW].

7.4 Jacobi-Davidson Implementation

Gerard L.G. Sleijpen provides a Matlab implementation of the Jacobi-Davidson method for the standard [Slea] and generalised eigenvalue problem [Sleb].

Chapter 8

Conclusion

8.1 Research Questions and Objectives

Following topics are planned to become part of the final thesis.

- Discretisation
  - Can the PDE be consistently discretised with the finite element method?
  - Comparing the results obtained with the consistent finite element method and the finite difference method with fixed singularity
- Numerical Analysis
  - Investigating and comparing the capability of both the Lanczos and Jacobi-Davidson method and selecting the more suitable algorithm
  - Determining the number of required grid points for computing the eigenvalues with sufficient accuracy
- Implementation
  - Implementing the final algorithm on an SMP or GPU system
  - Optional: Implementing on both systems and comparing the performance
- Physics
  - Using the application to compute the eigenvalues of the cubically confined hydrogen atom at least in two dimension, with respect to the
    - size of the cavity,
    - aspect ratio of the cavity, including the limit case towards one dimension,
    - location of the hydrogen atom inside the cavity.
  - Comparing of the results to the unconfined and spherically confined hydrogen atom
  - Optional: Computing the hydrogen atom in three dimensions and comparing to the energy levels in two dimension
  - How does value of the partition function change depending on the well size and the location of the atom inside the well?

8.2 Conclusion and Prospect

In this study numerical methods for the hydrogenic Schrödinger equation were analysed. It was shown that the separation of variables in spherical coordinates is possible for the equation of the unconfined atom. This allows the analytical or the numerical computation of the energy levels. However, if the atom is confined and shifted to an arbitrary position inside the cavity, a separation of variables becomes impossible. In this case a Cartesian coordinate system is preferred.

There exists no analytic solution to the unseparated equation and therefore an approximation is required. The solutions have furthermore a discontinuous derivative at the location of the nucleus due to the singular potential. If the variables are not separated then this point is located inside the domain. In two and higher dimensions a convergent finite difference scheme can be set up by using a variable grid. At the moment the variable grid is not optimal as it is set up heuristically. This could be improved by implementing adaptive grid refinement by using a posteriori information of computed solutions.

A finite element discretisation is proposed, as it offers a higher versatility for adaptive grids, and its convergence requirements respective smoothness of the solution are weaker than those of the finite difference method. Higher order approximations are not favoured, as the solutions are not smooth enough close to the nucleus.

Numerical methods for large sparse eigensystems are well developed. Both the Lanczos and Jacobi-Davidson method are suitable algorithms. Several established implementations of these algorithms are freely available. The algorithms and their existing implementations have to be closer analysed in the oncoming study to determine whether existing software should be used or yet another implementation on a recent GPU/SMP architecture is favourable.

Appendix A

Finite Difference Approximation Kernels

The following tables give the coefficients of the finite difference matrix convolution kernels. More kernels can be generated with the accompaniing MATLAB scripts. All kernels are given for the homogenous dirichlet boundary conditions, e.g. for $f(x_0) = f(x_n) = 0$. Boundary kernels are given for the left hand side only. The boundary kernels for the right hand side are flipped counterparts (even derivatives) or flipped negated counterparts (odd derivatives) respectively. Following naming convention is used:

$$x_l = x_{i-1} \tag{A.1}$$

$$x_r = x_{i+1} \tag{A.2}$$

$$h = x_{i+1} - x_i = \frac{L}{m+1}, \text{if constant} \tag{A.3}$$

$$n : \text{order of the derivative} \tag{A.4}$$

$$m : \text{number of grid points without boundary points = numer of unknowns} \tag{A.5}$$

A.1 Three Point Kernels Constant Step

Point	D^n	Den.	Numerator Coefficients		
$x_i,\ 0 < i < n$			$f(x-h)$	$f(x)$	$f(x+h)$
	1	$2h$	-1	0	1
	2	h^2	1	-2	1
x_0			$f(0)$	$f(h)$	$f(2h)$
	1	$2h$	1	-4	3
	2	h^2	1	-2	1

Table A.1: Three point finite difference kernels with constant step

A.2 Three Point Kernels Variable Step

Point	D^n	Coefficients		
$x_i,\ 0 < i < m$		$f(x_l)$	$f(x_i)$	$f(x_r)$
	1	$\frac{-(x_i-x_r)}{(x_i-x_l)(x_l-x_r)}$	$\frac{(2x_i-x_l-x_r)}{(x_i-x_l)(x_i-x_r)}$	$\frac{(x_i-x_l)}{(x_l-x_r)(x_i-x_r)}$
	2	$\frac{-2}{(x_i-x_l)(x_l-x_r)}$	$\frac{2}{(x_i-x_l)(x_i-x_r)}$	$\frac{2}{(x_l-x_r)(x_i-x_r)}$
x_0		$f(x_0)$	$f(x_1)$	$f(x_2)$
	1	$\frac{(x_1-x_2)}{(x_1-x_0)(x_0-x_2)}$	$\frac{-(x_0-x_2)}{(x_1-x_0)(x_1-x_2)}$	$\frac{-(x_1+x_0-2x_2)}{(x_0-x_2)(x_1-x_2)}$
	2	$\frac{-2}{(x_1-x_0)(x_0-x_2)}$	$\frac{2}{(x_1-x_0)(x_1-x_2)}$	$\frac{2}{(x_0-x_2)(x_1-x_2)}$

Table A.2: Three point finite difference kernels with variable step

A.3 Five Point Kernels Constant Step

Point	D^n	Den.	Numerator Coefficients				
$x_i,\ 1 < i < m-1$			f_k	f_l	f_c	f_r	f_s
	1	$12h$	1	-8	0	8	-1
	2	$12h^2$	-1	16	-30	16	-1
	3	$2h^3$	-1	2	0	-2	1
	4	$1h^4$	1	-4	6	-4	1
x_1			f_0	f_1	f_2	f_3	f_4
	1	$12h$	-3	-10	18	-6	1
	2	$12h^2$	11	-20	6	4	-1
	3	$2h^3$	-3	10	-12	6	-1
	4	$1h^4$	1	-4	6	-4	1
x_0			f_0	f_1	f_2	f_3	f_4
	1	$12h$	-25	48	-36	16	-3
	2	$12h^2$	35	-104	114	-56	11
	3	$2h^3$	-5	18	-24	14	-3
	4	$1h^4$	1	-4	6	-4	1

Table A.3: Five point finite difference kernels with constant step

A.4 Five Point Kernels Varying Step

Point	D^n	Coefficients	
x_i	1	1	$-(x_c - x_s)(x_c - x_r)(x_c - x_l)$
		2	$(x_c - x_s)(x_c - x_r)(x_c - x_k)$
		3	$4x_c^3 - 3x_c^2 x_l - 3x_c^2 x_r - 3x_c^2 x_s - 3x_c^2 x_k + 2x_c x_k x_l + 2x_c x_k x_r + 2x_c x_k x_s + 2x_c x_l x_r + 2x_c x_l x_s + 2x_c x_r x_s - x_k x_l x_r - x_k x_l x_s - x_k x_r x_s - x_l x_r x_s$
		4	$-(x_c - x_s)(x_c - x_l)(x_c - x_k)$
		5	$-(x_c - x_r)(x_c - x_l)(x_c - x_k)$
	2	1	$-2((x_r + x_s)(x_l - 2x_c) - 2x_c x_l + x_r x_s + 3x_c^2)$
		2	$2(x_k x_r - 2x_c x_r - 2x_c x_s - 2x_c x_k + x_k x_s + x_r x_s + 3x_c^2)$
		3	$2(x_k x_l - 3x_c x_l - 3x_c x_r - 3x_c x_s - 3x_c x_k + x_k x_r + x_k x_s + x_l x_r + x_l x_s + x_r x_s + 6x_c^2)$
		4	$-2(x_k x_l - 2x_c x_l - 2x_c x_s - 2x_c x_k + x_k x_s + x_l x_s + 3x_c^2)$
		5	$-2((x_l + x_r)(x_k - 2x_c) - 2x_c x_k + x_l x_r + 3x_c^2)$
	3	1	$-6(3x_c - x_l - x_r - x_s)$
		2	$6(3x_c - x_k - x_r - x_s)$
		3	$6(4x_c - x_k - x_l - x_r - x_s)$
		4	$-6(3x_c - x_k - x_l - x_s)$
		5	$-6(3x_c - x_k - x_l - x_r)$
	4	1	-24
		2	24
		3	24
		4	-24
		5	-24
den	all	1	$(x_c - x_k)(x_k - x_l)(x_k - x_r)(x_k - x_s)$
		2	$(x_l - x_r)(x_l - x_s)(x_k - x_l)(x_c - x_l)$
		3	$(x_c - x_k)(x_c - x_l)(x_c - x_r)(x_c - x_s)$
		4	$(x_r - x_s)(x_l - x_r)(x_k - x_r)(x_c - x_r)$
		5	$-(x_r - x_s)(x_l - x_s)(x_k - x_s)(x_c - x_s)$

Table A.4: Five point finite difference kernels with varying step size

A.5 Coefficients for Richardson Extrapolation

The table gives the weigths for combining solutions computed on a uniform grid after sucessive uniform refinements. Note that these weights do not apply to solutions computed on grids with varying step sizes, even if the final refinements where uniform, as the order of accuracy varies locally and hence local weights would be required. However, figure 3.4.3 shows that a one step Richardson extrapolation in such a case can nonetheless lead to a slight reduction of the error.

```
Denominator f_h f_h/2    f_h/4     f_h/8      f_h/16      f_h/32       Residual
          3, -1,    4,       0,        0,          0,          0,          -1/4 O(h^4)
         45,  1,  -20,      64,        0,          0,          0,          1/64 O(h^6)
       2835, -1,   84,   -1344,     4096,          0,          0,       -1/4096 O(h^8)
     722925,  1, -340,   22848,  -348160,    1048576,          0,     1/1048576 O(h^10)
  739552275, -1, 1364, -371008, 23744512, -357564416, 1073741824, -1/1073741824 O(h^12)
```

Table A.5: Coefficients of extrapolation and largest remaining residual term of the Richardson extrapolation

Appendix B

Numerical Integration Tables

B.1 Integration in 1D

Scheme	Point	x_1	x_2	weight	Accuracy
Midpoint	1	1/2	1/2	1	$O(h^2)$
Trapezoidal	1	1	0	1/2	$O(h^2)$
	2	0	1	1/2	

Table B.1: Rules for numerical integration in 1D

B.2 Integration in 2D

Scheme	Point	Point weights p			Value weights w	Accuracy
		x_1	x_2	x_3		
Face Midpoint	1	1/3	1/3	1/3	1	$O(h^2)$
Side Midpoint	1	1/2	1/2	0	1/3	$O(h^2)$
	2	1/2	0	1/2	1/3	$O(h^2)$
	3	0	1/2	1/2	1/3	$O(h^2)$
Trapezoidal	1	1	0	0	1/3	$O(h^2)$
	2	1	0	0	1/3	$O(h^2)$
	3	0	0	1	1/3	$O(h^2)$

Table B.2: Rules for numerical integration in 2D

Appendix C

Finite Element Discretisation of the Laplacian Operator and the Coulomb Potential

In this section the finite element discretisation matrix entries for one dimension are derived. Note that the finite element discretisation of a triangulation in two dimensions cannot be set up as a Kronecker product of one dimensional matrices, as it is possible with the finite difference method.

The stiffness matrix of the Laplacian operator is,

$$\int_{x_{i-1}}^{x_{i+1}} \phi_i'(x)^2 = \frac{1}{x_i - x_{i-1}} + \frac{1}{x_{i+1} - x_i} \tag{C.1}$$

$$\int_{x_{i-1}}^{x_i} \phi_i'(x)\phi_{i-1}'(x) = \frac{-1}{x_i - x_{i-1}}. \tag{C.2}$$

If one chooses a constant step size the stiffness matrix kernel becomes,

$$\frac{1}{h}[-1, 2, -1]. \tag{C.3}$$

The mass matrix entries of the Laplacian operator are,

$$\int_{x_{i-1}}^{x_{i+1}} \phi_i(x)\phi_i(x) = \int_{x_{i-1}}^{x_i} \frac{(x - x_{i-1})^2}{(x_i - x_{i-1})^2} dx + \int_{x_i}^{x_{i+1}} \frac{(x_{i+1} - x)^2}{(x_{i+1} - x_i)^2} dx \tag{C.4}$$

$$= \frac{1}{3}(x_i - x_{i-1}) + \frac{1}{3}(x_{i+1} - x_i) = \frac{1}{3}(x_{i+1} - x_{i-1}) \tag{C.5}$$

$$\int_{x_{i-1}}^{x_i} \phi_{i-1}(x)\phi_i(x) = \int_{x_{i-1}}^{x_i} \frac{(x - x_{i-1})(x_i - x)}{(x_i - x_{i-1})^2} dx \tag{C.6}$$

$$= -\frac{1}{6}\frac{(2x^3 - 3x^2(x_i + x_{i-1}) - 6xx_ix_{i-1}}{x_{i+1} - x_i)^2}\Big|_{x_{i-1}}^{x_i} = \frac{1}{6}(x_i - x_{i-1}). \tag{C.7}$$

If one chooses a constant step size the max matrix kernel becomes,

$$\frac{h}{6}[1, 4, 1]. \tag{C.8}$$

To set up the hydrogenic Schrödinger equation the stiffness matrix of the Laplacian opertator is weighted by $\frac{-1}{2}$ and the stiffness matrix of the potential is added,

$$\begin{aligned}
\int_{x_{i-1}}^{x_{i+1}} \frac{1}{x}\phi_i(x)^2 &= \int_{x_{i-1}}^{x_i} \frac{1}{x}\frac{(x-x_{i-1})^2}{(x_i-x_{i-1})^2}dx + \int_{x_i}^{x_{i+1}} \frac{1}{x}\frac{(x_{i+1}-x)^2}{(x_{i+1}-x_i)^2}dx \qquad &(\text{C.9})\\
&= \frac{x^2-4x_{i-1}x+2x_{i-1}^2\log(x))}{2(x_i-x_{i-1})^2}\Big|_{x_{i-1}}^{x_i} + \frac{x^2-4x_{i+1}x+2x_{i+1}^2\log(x))}{2(x_{i+1}-x_i)^2}\Big|_{x_i}^{x_{i+1}} \qquad &(\text{C.10})\\
&= -\frac{x_{i-1}}{x_i-x_{i-1}} + x_{i-1}^2\frac{\log(x_i/x_{i-1})}{(x_i-x_{i-1})^2} - \frac{x_{i+1}}{x_{i+1}-x_i} + x_{i+1}^2\frac{\log(x_{i+1}/x_i)}{(x_{i+1}-x_i)^2} \qquad &(\text{C.11})\\
\int_{x_{i-1}}^{x_i} \frac{1}{x}\phi_{i-1}(x)\phi_i(x) &= -\int_{x_{i-1}}^{x_i} \frac{1}{x}\frac{(x-x_{i-1})(x_i-x)}{(x_i-x_{i-1})^2}dx \qquad &(\text{C.12})\\
&= -\frac{1}{2}\frac{x^2-2x(x_i+x_{i-1})-2x_ix_{i-1}\log(x)}{(x_i-x_{i-1})^2}\Big|_{x_{i-1}}^{x_i} \qquad &(\text{C.13})\\
&= -\frac{1}{2}\frac{(x_i+x_{i-1})}{(x_i-x_{i-1})} + \frac{x_ix_{i-1}\log(x_{i-1}/x_i)}{(x_i-x_{i-1})^2}. \qquad &(\text{C.14})
\end{aligned}$$

This discretisation by exact integrals brings certain complications close to the origin, it is therefore meaningful to approximate the potential term by means of numerical quadrature, with the trapezoidal rule the integrals become,

$$\int_{x_{i-1}}^{x_{i+1}} \frac{1}{x}\phi_i(x)^2 = \frac{1}{2x_i(x_i-x_{i-1})} + \frac{1}{2x_i(x_{i+1}-x_i)} \int_{x_{i-1}}^{x_i} \frac{1}{x}\phi_{i-1}(x)\phi_i(x) = \qquad 0. \tag{C.15}$$

For a constant step size the discretisation kernel of the potential becomes,

$$\tfrac{1}{h}[0, \tfrac{1}{x_i}, 0]. \tag{C.16}$$

Appendix D

Eigenvalues of the Discrete Laplacian

D.1 Finite Difference Discretisation

The matrix kernel of the of the one dimensional Laplacian discretised with a 3-point finite diffrence scheme on a uniform grid is,

$$\lambda u(x_i) = \frac{1}{h^2}[1, -2, 1] \begin{bmatrix} u(x_i - h) \\ u(x_i) \\ u(x_i + h) \end{bmatrix}. \tag{D.1}$$

Assume eigenfunctions in form of trigonometric functions $u(x) = \sin(ax)$,

$$\lambda \sin(ax_i) = \frac{1}{h^2}(\sin(a(x_i - h)) - 2\sin(ax_i) + \sin(a(x_i + h))) \tag{D.2}$$

Close to the boundary for $x = h$ and $h = \frac{1}{n+1}$ with n being the number of grid points this becomes,

$$\lambda \sin(ah) = \frac{1}{h^2}(-2\sin(ah) + \sin(2ah)). \tag{D.3}$$

Application of some trigonometric identities yields,

$$\begin{aligned} \lambda \sin(ah) &= \frac{1}{h^2}(-2\sin(ah) + 2\sin(ah)\cos(ah)) & \text{(D.4)} \\ &= \frac{-2}{h^2}(1 - \cos(ah)). & \text{(D.5)} \end{aligned}$$

The Dirichlet boundary conditions require that,

$$\begin{aligned} u(0) &= \sin(a0) = 0 & \text{(D.6)} \\ u(1) &= \sin(a(n+1)h) = \sin(a) = 0. & \text{(D.7)} \end{aligned}$$

This is only satisfied if $a = \pi k$. Hence the eigenvalues are given by,

$$\begin{aligned} \lambda_k &= \frac{-2}{h^2}(1 - \cos(\pi k h)) & \text{(D.8)} \\ &= \frac{-4}{h^2}\sin^2(\tfrac{1}{2}\pi k h) & \text{(D.9)} \end{aligned}$$

For finite differences, this analysis can be continued into arbitrarily many dimension. So are the eigenvalues of the two dimensional discrete Laplacian found as,

$$\lambda u(h_x, h_y) = \frac{1}{h_x h_y}[-4, 1, 1] \begin{bmatrix} u(h_x, h_y) \\ u(2h_x, h_y) \\ u(h_x, 2 * h_y) \end{bmatrix}, \tag{D.10}$$

$$u(0, y) = 0, u(1, y) = 0, u(x, 0) = 0, u(x, 1) = 0. \tag{D.11}$$

Where the ste widths are defined as usual $h_x = \frac{1}{n_x+1}, h_y = \frac{1}{n_y+1}$. By assuming again solutions form of $u(x,y) = \sin(a_x x)\sin(a_y y)$ and using the identy $\sin(2x) = 2\sin(x)\cos(x)$ this becomes,

$$\lambda = \frac{1}{h_x h_y}(-4 + 2\cos(a_x h_x) + 2\cos(a_y h_y)). \tag{D.12}$$

The boundary conditions are satisfied for $a_x = \pi k_x h_x$ and $a_y = \pi k_y h_y$,

$$\lambda = \tfrac{-4}{h_x h_y}(sin^2(\tfrac{1}{2}\pi k_x h_x) + sin^2(\tfrac{1}{2}\pi k_y h_y)) \tag{D.13}$$

$$k_x = 1, 2, \ldots, n_x; k_y = 1, 2, \ldots, n_y \tag{D.14}$$

Note that there are multiple, e.g. degenerated eigenvalues if both axes are discretised with the same number of grid points. The degeneracy is completely reolved if one chooses $n_x \neq n_y$.

D.2 Finite Elemen Discretisation

The matrix kernels of the one dimensional Laplacian discretised with the finite element method using piecewise linear polynomials and a uniform grid are,

$$\lambda\frac{h}{6}[1,4,1]\begin{bmatrix} u(x_i - h) \\ u(x_i) \\ u(x_i + h) \end{bmatrix} = \frac{1}{h}[1,-2,1]\begin{bmatrix} u(x_i - h) \\ u(x_i) \\ u(x_i + h) \end{bmatrix}. \tag{D.15}$$

Assuming again a solution in form of trigonometric functions and evaluating the function at the boundary this becomes,

$$\lambda\frac{h}{6}(4\sin(ah) + \sin(2ah))) = \frac{1}{h}(-2\sin(ah) + \sin(2ah)) \tag{D.16}$$

$$\lambda\frac{h}{6}(4\sin(ah) + 2\sin(ah)\cos(ah))) = \frac{1}{h}(-2\sin(ah) + 2\sin(ah)\cos(ah)) \tag{D.17}$$

$$\lambda\frac{h}{6}(2 + \cos(ah))) = \frac{-1}{h}(1 - \cos(ah)) \tag{D.18}$$

$$\lambda = \frac{-6}{h^2}\frac{1 - \cos(ah)}{2 + \cos(ah)}. \tag{D.19}$$

Applying the boundary condition yields again the relation $a = \pi k$ and therefore

$$\lambda_k = \frac{-6}{h^2}\frac{1 - \cos(\pi kh)}{2 + \cos(\pi kh)}. \tag{D.20}$$

Contrary to the finite difference method, this analysis cannot simply be continued for the finite element method in higher dimensions. Even if one chooses a regular triangulation, the eigenvectors become perturbed by higher order effects in combination with Gaussian quadrature rules.

List of Algorithms

List of Figures

List of Tables

[ABJ+99] T.H. Apel, M. Berzins, P.K. Jimack, G. Kunert, A. Plaks, I. Tsukerman, and M. Wlakley. Mesh Shape and Anisotropic Elements - Theory and Practice. *The Mathematics of Finite Elements and Applications X: MAFELAP 1999*, page 367, 1999.

[Age12] European Space Agency. European Space Research and Technology Centre (ESA-ESTEC). `http://www.esa.int/esaMI/ESTEC/index.html`, 2012.

[And76] M. Andrews. Singular Potentials in one Dimension. *American Journal of Physics*, 44:1064–1066, 1976.

[BDJ+00] Z. Bai, J. Demmel, Dongarra J., A. Ruhe, and H.A. van der Vorst. *Templates for the Solution of Algebraic Eigenvalue Problems: A Practical Guide.* SIAM, Philadelphia, 2000.

[BDY88] R.E. Bank, T.F. Dupont, and H. Yserentant. The Hierarchical Basis Multigrid Method. *Numer. Math.*, 52:427–458, 1988.

[Bey95] J. Bey. Tetrahedral Grid Refinement. *Computing*, 55:355–378, 1995.

[BH86] D. Baye and P.-H. Heenen. Generalised Meshes for Quantum Mechanical Problems. *J. Phys. A: Math. Gen.*, 19:2041–2059, 1986.

[Bry84] G.W. Bryant. Hydrogenic Impurity States in Quantum-Well Wires. *Phys. Rev. B*, 29:6632–6639, 1984.

[BSR56] H.C. Bolton, H.I. Scoinsa, and G.S. Rushbrooke. Eigenvalues of Differential Equations by Finite-Difference Methods. *Mathematical Proceedings of the Cambridge Philosophical Society*, 52:215–229, 1956.

[Car04] C. Carstensen. Adaptive Mesh Refining Algorithm allowing for an H^1-Stable L^2-Projection onto Courant Finite Element Spaces. *Constructive Approximation*, 20(4):549–564, 2004.

[CCR09] S.A. Cruz and R. Colín-Rodríguez. Spheroidal Confinement of a Single Electron and of the Hydrogen Atom, the H_2^+ and HeH^{++} Molecular Ions With Arbitrary Nuclear Positions Along the Major Axis. *International Journal of Quantum Chemistry*, 109(13):3041–3054, 2009.

[CJJT01] G. Cohen, P. Joly, Roberts J.E., and N. Tordjman. Higher Order Triangular Finite Elements with Mass Lumping for the Wave Equation. *SIAM Journal on Numerical Analysis*, 38(6):2047–2078, 2001.

[Com12] European Commission. Erasmus Mundus. `http://ec.europa.eu/education/external-relation-programmes/doc72_en.htm`, 2012.

[Cow73] G.R. Cowper. Gaussian Quadrature Formulas for Triangles. *International Journal for Numerical Methods in Engineering*, 7(3):405–408, 1973.

[CW] J.K. Cullum and R.A. Willoughby. Lanczos algorithms for large symmetric eigenvalue computations, volume 2 - programs. `http://www.netlib.org/lanczos/`.

[CW81] J.K. Cullum and R.A. Willoughby. Computing Eigenvalues of Very Large Symmetric Matrices - An Implementation of a Lanczos Algorithm with no Reorthogonalization. *Journal of Computational Physics*, 44((2)):329–358, 1981.

[CW02] J.K. Cullum and R.A. Wiloughby. *Lanczos Algorithms for Large Symmetric Eigenvalue Computations Volume 1: Theory (Classics in Applied Mathematics).* SIAM: Society for Industrial and Applied Mathematics, 1st edition, 2002.

[Dav75] E.R. Davidson. The Iterative Calculation of a Few of the Lowest Eigenvalues and Corresponding Eigenvectors of Large Real-Symmetric Matrices. *Journal of Computational Physics*, 17(1):87–94, 1975.

[dGtSC46] S.R. de Groot and ten Seldam C.A. On the Energy Levels of a Model of the Compressed Hydrogen Atom. *Physica*, XII(9–10):669–682, 1946.

[Dol98] V. Dolejsi. Anisotropic Mesh Adaption for Finite Volume and Finite Element Methods on Triangular Meshes. *Computing and Visualization in Science*, 1:165–178, 1998.

[DSBE12] TU Delft, KTH Stockhoml, TU Berlin, and FAU Erlangen. COSSE - Computer Simulations for Science and Engineering. `http://www.kth.se/en/studies/programmes/master/em/cosse`, 2012.

[Dun85] D.A. Dunavant. High Degree Efficient Symmetrical Gaussian Quadrature Rules for the Triangle. *International Journal for Numerical Methods in Engineering*, 21:1129–1148, 1985.

[Dur02] Durbin, P.A. and Iaccarino, G. Approach to Local Refinement of Structured Grids. *Journal of Computational Physics*, 181:639–653, 2002.

[EJ91] K. Eriksson and C. Johnson. Adaptive Streamline Diffusion Finite Element Methods for Stationary Convection Diffusion Problems. *Mathematics of Computation*, 60(201):167–188, 1991.

[FM84] V. Faber and T. Manteuffel. Necessary and sufficient conditions for the existence of a conjugate gradient method. *SIAM J. Numer. Anal.*, 21:315–339, 1984.

[Fra61] J.G.F. Francis. The QR Transformation - A Unitary Analogue to the LR Transformation. *The Computer Journal*, 4(3):265–271, 1961.

[GA06] B. Grossman and Dadone A. Ghost-Cell Method with Far-Field Coarsening and Mesh Adaption for Cartesian Grids. *Computers and Fluids*, 35:676–687, 2006.

[Gar98] J. Garcke. Berechnung von Eigenwerten der Stationären Schrödingergleichung mit der Kombinationstechnik, 1998.

[GC97] A.N. Gordeyevy and S.C. Chhajlanyz. One-Dimensional Hydrogen Atom: A Singular Potential in Quantum Mechanics. *J. Phys. A: Math. Gen.*, 30:6893–6909, 1997.

[GvL96] G.H. Golub and C.F. van Loan. *Matrix Computations.* The John Hopkins University Press, Stanford, Cornell, 3rd edition, 1996.

[H.42] Freudenthal H. Simplizialzerlegung von Beschränkter Flachheit. *Annalen der Mathematik*, 43(3):580–582, 1942.

[HKK12] A. Hannukainen, S. Korotov, and M. Krizek. Maximum Angle Condition is not Necessary for the Convergence of the Finite Element Method. *Numerische Mathematik*, 120(1):79–88, 2012.

[HS52] M.R. Hestenes and E. Stiefel. Methods of Conjugate Gradients for Solving Linear Systems. *J. Res. Nat. Bur. Stand.*, 49:409–436, 1952.

[JC96] P.K. Jimack and P.J Capon. On the Adaptive Finite Element Solution of Partial Differentila Equations using h-r-Refinement, 1996.

[Joh88] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method.* Dover Publications, 2009 [1988].

[KAV09] M.K. Kretov, Scherbinin A.V., and Pupyshev V.I. States of a Hydrogen Atom in an Impenetrable Cubic Cavity. *Phys. Scr.*, 80, 2009.

[Kil77] J. Killingbeck. A Pocket Calculator Determination of Energy Eigenvalues. *J. Phys. A: Math. Gen.*, 10(6):L99–L103, 1977.

[KS34] G.E. Kimball and G.H. Shortley. The Numerical Solution of Schrödinger's Equation. *Phys. Rev.*, 45(11):815–820, Jun 1934.

[KSY98] D.S. Krähmer, W.P. Schleich, and V.P. Yakovlev. Confined Quantum Systems: The Parabolically Confined Hydrogen Atom. *J. Phys. A: Math. Gen.*, 31:4493–4520, 1998.

[Lan50] C. Lanczos. An Iteration Method for the Solution of the Eigenvalue Problem of Linear Differential and Integral Operators. *J. Res. Nat. Bureau Standards*, 45(4):255–282, 1950.

[LDC97] R.B. Lehoucq, Sorensen D.C., and Yang C. Arpack. `http://www.caam.rice.edu/software/ARPACK/`, 1997.

[Led11] S. Ledru. arpack-ng. `http://forge.scilab.org/index.php/p/arpack-ng/`, 2011.

[LJ96] A. Liu and B. Joe. Quality Local Refinement of Tetrahedral Meshes Based on 8-Subtetrahedron Subdivision. *Mathematics of Computation*, 65(215):1183–1200, 1996.

[LKR79] E. Ley-Koo and S. Rubinstein. The Hydrogen Atom within Spherical Boxes with Penetrable Walls. *J. Chem. Phys.*, 71:351–357, 1979.

[Lou59] R. Loudon. One-Dimensional Hydrogen Atom. *American Journal of Physics*, pages 649–655, 1959.

[LS85] F.S. Levin and J. Shertzer. Finite-element solution of the Schrodinger equation for the helium ground state. *Physical Review A*, 32(6), 1985.

[Mah02] R.S. Mahmood. Multilevel Mesh Adaptivity for Elliptic Boundary Value Problems in Two and Three Space Dimensions, 2002.

[MdBB37] A. Michels, J. de Boer, and A. Bijl. Remarks Concerning Molecular Interaction and their Influence on the Polarisation. *Phyisca*, VI(10):981–994, 1937.

[Pai80] C.C. Paige. Accuracy and Effectiveness of the Lanczos Algorithm for the Symmetric Eigenproblem. *Linear Algebra and its Applications*, 34:235–258, 1980.

[Par87] B.N. Parlett. *The Symmetric Eigenvalue Problem (Classics in Applied Mathematics).* Society for Industrial Mathematics, 1987.

[PP73] B.N. Parlett and W.G. Poole. A Geometric Theory for the QR, LU and Power Iterations. *Journal on Numerical Analysis*, 10(2):389–412, 1973.

[PS75] C.C. Paige and M.A. Saunders. Solution of Sparse Indefinite Systems of Linear Equations. *SIAM J. Numer. Anal.*, 12(4):617–629, 1975.

[PS79] B.N. Parlett and D.S. Scott. Lanczos Algorithm with Selective Orthogonalization. *Mathematics of Computation*, 33(145):217–238, 1979.

[Red05] J.N. Reddy. *An Introduction to the Finite Element Method, 3rd ed.* McGraw-Hill, 2005.

[Ric11] L.F. Richardson. The Approximate Arithmetical Solution by Finite Differences of Physical Problems Involving Differential Equations, with an Application to the Stresses in a Masonry Dam. *Transactions of the Royal Society of London. Series A*, 210:307–357, 1911.

[Riv84] M.C. Rivara. Algorithms for Refining Triangular Grids Suitable for Adaptive and Multigrid Techniques. *International Journal for Numerical Methods in Engineering*, 20:745–756, 1984.

[Saa11] Y. Saad. *Numerical Methods for Large Eigenvalue Problems (Algorithms and Architecture for Advanced Scientific Computing).* Halsted Pr, 2nd edition, 2011.

[Sch26] E. Schrödinger. Quantisierung als Eigenwertproblem (Erste Mitteilung). *Annalen der Physik*, 79:361–376, 1926.

[SF88] G. Strang and G. Fix. *An Analysis of the Finite Element Method, Second Edition.* Wellesley-Cambridge Press, 1973 [1988].

[Sha49] C.E. Shannon. Communication in the Presence of Noise. *Proceedings of the IRE*, 37(1):10 – 21, jan. 1949.

[Sim84] H.D. Simon. Analysis of the Symmetric Lanczos Algorithm with Reorthogonalization Methods. *Linear Algebra and Its Applications*, 61((C)):101–131, 1984.

[Slea] G.L.G. Sleijpen. jdqr. `http://www.staff.science.uu.nl/~sleij101/JD%5Fsoftware/JDQR.html`.

[Sleb] G.L.G. Sleijpen. jdqz. `http://www.staff.science.uu.nl/~sleij101/JD%5Fsoftware/JDQZ.html`.

[Spa04] et. al. Spataru. Excitonic Effects and Optical Spectra of Single-Walled Carbon Nanotubes. *Phys. Rev. Lett.*, 92(7), 2004.

[SPM09] K.D. Sen, V.I. Pupyshev, and H.E. Jr. Montgomery. Exact Relations for Confined One-Electron Systems. *Advances in Quantum Chemistry*, 57:25–77, 2009.

[ST03] M. Sun and K. Takayama. Error Localization in Solution-adaptive Grid Methods. *Journal of Computational Physics*, 190:346–350, 2003.

[SvdV00] G.L.G. Sleijpen and H.A. van der Vorst. A Jacobi-Davidson Iteration Method for Linear Eigenvalue Problems. *SIAM J. Matrix Anal. Appl*, 17:401–425, 2000.

[SW38] A. Sommerfeld and H. Welker. Künstliche Grenzbedingungen beim Keplerproblem. *Annalen der Physik*, 5. Folge 32:56–65, 1938.

[Tra97] C.T. Traxler. An Algorithm for Adaptive Mesh Refinement in n-Dimensions. *Computing*, 59:115–137, 1997.

[TSCH90] I.-H. Tan, G.L. Snider, L.D. Chang, and E.L. Hu. A Self-Consistent Solution of Schrödinger-Poisson Equations Using a nonuniform Mesh. *Journal of Applied Physics*, 68:4071–4076, 1990.

[UoTUoCL] Berkeley; Univ. of Colorado Denver; Univ. of Tennessee; Univ. of California and NAG Ltd. LAPACK (Linear Algebra PACKage). `http://www.netlib.org/lapack/`.

[vD11] L. van Dommelen. Quantum Mechanics for Engineers, 2011.

[vdV] H.A. van der Vorst. Computational Methods for Large Eigenvalue Problems.

[vN31] J. von Neumann. Die Eindeutigkeit der Schrödingerschen Operatoren. *Mathematische Annalen*, 104(1):570–578, 1931.

[YSP08] P.V. Yurenev, A.V. Scherbinin, and V.I. Pupyshev. Shifts of the Hydrogen Atom in a Cylindrical Cavity. *International Journal of Quantum Chemistry*, 108(14):2666–2667, 2008.