Adaptive finite element methods for eigenvalue problems with applications in quantum physics

Axel Målqvist

September 26, 2001

Abstract

In this paper we formulate the finite element method for a self-adjoint elliptic eigenvalue problem and present an a posteriori error bound of the eigenvalue error and the eigenvector error in the L^2-norm. In the eigenvector case we need to solve a dual problem. This is done numerically which means that we also need to do some perturbation theory to ensure that the numerical solution is a good approximation of the exact one. The error bounds from the a posteriori theory are expressed in terms of an error function defined as a constant on each element in the mesh and this function is used to refine the elements where the error is big. This gives us an adaptive algorithm. The algorithm is used to solve the time-independent Schrödinger equation for a hydrogen atom on a platinum surface. The system of a Hydrogen atom on a metal surface is of interest among other things for design of fuel cells and H_2-storage. We use this problem from physics to evaluate how well the adaptive algorithm works and we compare it to the non-adaptive solution. The result we get is that the adaptive algorithm gives what seams to be a sharp bound of the eigenvalue error and higher accuracy than the non-adaptive method with less than half of the number of nodes in the mesh.

Acknowledgements

I would like to thank my supervisor Kenneth Eriksson for his support throughout the writing of this paper and for all his valuable advice and comments. I also want to thank him for everything he has taught me in this field over the last three years. I want to thank Lars Langemyr and the others behind *FEMLAB* who have helped me with software issues and also given me inspiration for the topic of this paper and Göran Wahnström among others at the Department of Applied Physics at Chalmers University of Technology for finding the physical problem I have studied and for giving me several lectures in basic Solid State Physics. I also want to thank Endre Süli, who has been my supervisor in Oxford, for teaching me more about the finite element method, and finally Claes Johnson who collaborated with Endre to arrange my stay in Oxford this term. It has been very important to me in many ways.

Contents

1 Introduction

Eigenvalue problems occur in various branches of science. From the acoustic wave from a guitar string or a drum to electronic circuits, structural mechanics and quantum mechanics. The first step to take when solving these problems is to make a mathematical model. That is formulate the problems in mathematical terms, equations. But if every aspect of the problem has to be represented in the model we will almost always end up with a very complicated equation that will be impossible to solve by hand. From here there are mainly two ways to go. One is to simplify the problem until it is possible to solve by hand, that is to solve the approximate problem exactly. Another approach is to solve the original problem approximately. This paper will focus on the latter way of dealing with this problem.

We will describe how to solve these problems with the finite element method (FEM) both in theory and in practice on the computer. We will present an a posteriori error analysis which will give us an estimate of how close our approximate solution is the exact one which is very important, since an approximate solution does not really help us if we have no idea of how good it is. We will then focus on a specific eigenvalue problem that arises in quantum physics.

The wave equation in quantum physics is called Schrödingers equation named after the inventor, Erwin Schrödinger. Because of Heisenbergs uncertainty principle which becomes important in really small scales, the exact position and momentum of a particle cannot be found at the same time. Instead the Schrödinger equations gives the likelihood of finding the particle in a specific place as its eigenvectors or eigenfunctions and the energy of the state as its eigenvalues. We will sometimes refer to such a pair, of an eigenvalue with the corresponding eigenvector, as an eigenpair.

The theory and implementation will be done for one particle in a three dimensional potential. This could represent a light (weight) particle on a metal surface. There are many applications in this area and we will here calculate some results for the problem of a hydrogen atom on a platinum surface where the platinum surface is described as a potential. This problem has applications in the field of fuel cells.

So the aim of this paper is to present an error analysis that can be applied to the time-independent Schrödinger equation for one particle in a potential and to calculate eigenvalues of this problem that is of interest for physicists and give the eigenvalues with an error bound that is as sharp as possible.

2 Problem formulation

2.1 Preliminaries

Let us first settle some definitions and notations that will be frequently used in this paper. All the calculations are performed on a convex three dimensional domain Ω, see figure (1). The scalar product $(\cdot,\cdot)$ is the ordinary $L^2 = L^2(\Omega)$ product and $\|\cdot\|$ is the corresponding norm. The boundary of Ω will be refereed to as $\partial\Omega$ and it will be divided into two boundary parts Γ_1 and Γ_2 associated with Dirichlet and periodic boundary conditions respectively. For these parts we have $\Gamma_1 \cup \Gamma_2 = \partial\Omega$ and $\Gamma_1 \cap \Gamma_2 = \emptyset$. Γ_2 needs to have some more properties since its associated with periodic boundary conditions. Instead of describing this in a general case we just study the geometry and boundary conditions that are going to be used in the calculations. The boundary conditions on the two planes parallel to the xy-plane will always be homogeneous Dirichlet and the other four parts will either all have homogeneous boundary conditions or all have periodic boundary conditions.

We need to define a function space on this domain. Since we have two different choices of boundary conditions we need two spaces.

Definition 2.1 *The space H_0^1 is the set of all functions in $L^2(\Omega)$ such that ∇u also is in $L^2(\Omega)$ and that have the value zeros on $\partial\Omega$. The space $H^1_{0,per}$ is defined as the set of all functions in $L^2(\Omega)$ such that ∇u also is in $L^2(\Omega)$ and that have the value zeros on Γ_1 and are periodic on Γ_2.*

H_0^1 and $H^1_{0,per}$ are in fact Hilbert spaces [4] with associated norm [15]

$$\|u\|_{H^1(\Omega)} = \{\|u\|^2 + \sum_{j=1}^{3} \|\frac{\partial u}{\partial x_j}\|^2\}^{1/2}.$$

Since the results applies on both types of boundary conditions and for both these spaces we will reduce the number of equations by referring to both of them as H_0^1 from here on. For the discretised problem we also need a space containing piecewise polynomials. Let $V_h^p \subset H_0^1$ be the set of all piecewise polynomials of degree at most p on a partion $\mathcal{T}_h$ of Ω into tetrahedrons τ of size (diameter[1]) $h = h(\tau)$. The function $vol(\tau)$ denotes the volume of a tetrahedron $\tau \in \mathcal{T}_h$ and $\partial\tau$ will denote the boundary of tetrahedron τ in the interior of Ω. The norm $\|\cdot\|_\tau$ is the $L^2(\tau)$-norm.

We recall that a bilinear function ϕ is elliptic on H_0^1 if there exists a constant $K > 0$ such that $\phi(x,x) \geq K\|x\|^2_{H^1}(\geq K\|x\|^2)$ for all $x \in H_0^1$ and that an operator is self-adjoint if $(Lx,y) = (x,Ly)$ for all $x,y \in H_0^1$.

[1]Diameter is the length of the longest edge of the tetrahedron.

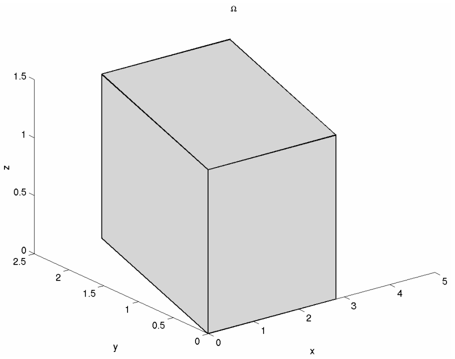

Figure 1: A plot of the three dimensional domain Ω.

2.2 The eigenvalue problem

Since the final aim here is to solve the time-independent Schrödinger equation we only consider a small class of eigenvalue problems namely the ones that can be written in the following way

$$\begin{aligned} -\triangle u + c\,u - \lambda u = 0 \quad & in\ \Omega, \\ u = 0 \quad & on\ \Gamma_1, \\ u\ periodic \quad & on\ \Gamma_2, \end{aligned} \tag{1}$$

where $c = c(x)$ is a function dependent of the three space variables. The operator $L = -\triangle + c$ is self-adjoint and the associated bilinear functional $a(v, w) = (\nabla v, \nabla w) + (c\,v, w)$ for $v, w \in H_0^1$ is elliptic. The reason why we do not get any contributions from the boundary in the bilinear functional is because $w = 0$ on Γ_1 and $\int_{\Gamma_2} n \cdot \nabla v\, w\, dx = 0$ due to periodicity.

To prove self-adjointness we just need to integrate by parts and use that $v, w \in H_0^1$ that is

$$(Lv, w) = (-\triangle v + c\,v, w) = (\nabla v, \nabla w) + (c\,v, w) = (v, Lw).$$

To ensure ellipticity we study how the bilinear form a can be estimated from below using a function $v \in H_0^1$

$$a(v, v) = (\nabla v, \nabla v) + (c\,v, v) \geq \min(1, c)\|v\|_{H^1}^2.$$

So we just need to assume that $c(x) \geq K > 0$ and ellipticity follows. For the error analysis we also assume that the eigenvalues of L are simple, which means that they have multiplicity one or a one dimensional eigenvalue space [4]. We will also assume that $c(x) \leq C < \infty\ \forall x \in \Omega$.

2.3 Weak formulation and properties of self-adjoint operators

We will start with deriving the weak formulation of the original problem (1). By multiplying (1) by a function $v \in H_0^1$, integrating over the domain Ω, and using integration by parts, we get the weak formulation which reads: find $u \in H_0^1$ and $\lambda \in \mathbf{R}$ such that

$$a(u,v) - \lambda(u,v) = 0, \quad \forall v \in H_0^1. \tag{2}$$

This equation is in fact equivalent with (1) since it follows from (2) that

$$(-\triangle u + c\,u - \lambda u, v) = 0 \quad \forall v \in H_0^1,$$

from which we may conclude that $-\triangle u + c\,u - \lambda u = 0$.

We will now derive some familiar properties of self-adjoint elliptic operators [4].

Theorem 2.2 *A self-adjoint elliptic operator A defined on a Hilbert space has the following properties (a) the eigenvalues are real (b) the eigenvalues are greater then 0 (c) the eigenvectors corresponding to different eigenvalues are orthogonal and (d) there exists an orthonormal basis $\{v_i\}_{i=1}^{\infty}$ of eigenvectors, with associated eigenvalues $\{\lambda_i\}_{i=1}^{\infty}$, of A in L_2.*

Proof. *Let $u_1 \neq 0$ and $u_2 \neq 0$ be eigenvectors corresponding to the distinct eigenvalues λ_1 and λ_2 of A.*
(a) self-adjointness is used

$$\lambda_1(u_1,u_1) = (\lambda_1 u_1, u_1) = (Au_1,u_1) = (u_1, Au_1) = (u_1, \lambda_1 u_1) = \overline{\lambda_1}(u_1,u_1)$$

Since $(u_1,u_1) > 0$ we conclude $\lambda_1 = \overline{\lambda_1}$.
(b) ellipticity is used

$$0 < K(u_1,u_1) \leq (Au_1,u_1) = \lambda_1(u_1,u_1)$$

Again since $(u_1,u_1) > 0$ we conclude $\lambda_1 > 0$.
(c) self-adjointness and distinct eigenvalues are used

$$\lambda_1(u_1,u_2) = (\lambda_1 u_1, u_2) = (Au_1,u_2) = (u_1, Au_2) = (u_1, \lambda_2 u_2) = \lambda_2(u_1,u_2)$$

Since λ_1 and λ_2 are distinct this means that $(u_1,u_2) = 0$.
(d) this requires a more complicated proof which can be found in [4]. We just notice that the inverse operator A^{-1} has the eigenvectors $\{v_i\}_{i=1}^{\infty}$ as well and that it is a compact operator so the Hilbert-Schmidt theorem can be used.
■

We let $\{\lambda_i\}_{i=1}^{\infty}$ be the eigenvalues of L and $\{v_i\}_{i=1}^{\infty}$ be the corresponding basis of orthonormal eigenvectors in L_2. $\lambda = \lambda_j$ will represent an element from the set $\{\lambda_i\}_{i=1}^{\infty}$. Since we have assumed simple eigenvalues we can also order the eigenvalues in the following way

$$0 < \lambda_1 < \lambda_2 < \lambda_3 < \dots .$$

2.4 The Finite element method

The equation that arise from Galerkins finite element method, based on replacing H_0^1 in (2) by the finite dimensional subspace V_h^p, is now: find $u_h \in V_h^p$ and $\lambda_h \in \mathbf{R}$ such that

$$a(u_h, v_h) - \lambda_h (u_h, v_h) = 0, \quad \forall v_h \in V_h^p. \tag{3}$$

From now on we also assume that $\|u_h\| = 1$. Our aim is to find this solution by choosing a suitable basis for V_h^p and solve the discrete eigenvalue problem that will arise. To do this we choose a basis $\{\varphi_i\}_1^n$, where n is the degree of freedom in the mesh, such that $span\{\varphi_i\} = V_h^p$[2]. This gives us the following version of the discretised weak formulation: find $u_h = \sum_{i=1}^n U_i \varphi_i$ and $\lambda_h \in \mathbf{R}$ such that

$$a(u_h, \varphi_j) - \lambda_h (u_h, \varphi_j) = 0, \quad \forall j \in \{1, \ldots, n\}. \tag{4}$$

This discrete problem has n eigenvalues and corresponding eigenvectors. We denote these eigenvalues by $\{\lambda_i^h\}_{i=1}^n$ and the eigenvectors by $\{u_i^h\}_{i=1}^n$. In the discrete case the eigenvectors can be identified with an orthogonal basis of the Hilbert space $\mathbf{R}^n$. In the next section we will see how well they approximate the exact eigenvalues and vectors.

[2]This makes (3) and (4) equivalent.

3 Error analysis

We are going to study two different measures of the error of a solution. The first one is the eigenvalue error $\lambda_h - \lambda$ and the other is the eigenvector error defined by $e = u_h - u$ where u is chosen such that $(e, u) = 0$ and $(u, u_h) > 0$.

3.1 A priori error estimate

We will present results from an a priori error analysis of the eigenvalue error [1], [7], [6] and the eigenvector error [7] in L^2-norm. We need to assume the following.

For any $v \in H_0^1$, $k \leq p$ there is a function $\pi v \in V_h^p$ such that

$$\begin{aligned} \|v - \pi v\| &\leq C_1 \|h^{k+1} D^{k+1} v\| \\ \|\nabla(v - \pi v)\| &\leq C_2 \|h^k D^{k+1} v\| \end{aligned} \tag{5}$$

with constants C_1 and C_2 independent of v and h.

We recall that the Rayleigh quotient of our operator can be expressed as

$$RQ(v) = \frac{a(v, v)}{(v, v)}.$$

This formula together with (2) gives that $RQ(u_j) = \lambda_j$. From the well known min-max principle [1] we have

$$\lambda_k = \min_{\substack{V_k \subset H_0^1 \\ dim(V_k)=k}} \max_{u \in V_k} RQ(u), \quad k = 1, 2, \ldots$$

for the exact eigenvalues and

$$\lambda_k^h = \min_{\substack{V_k \subset V_h^p \\ dim(V_k)=k}} \max_{u \in V_k} RQ(u), \quad k = 1, 2, \ldots, n$$

for the approximate eigenvalues and since $V_h^p \subset H_0^1$ we have that $\lambda_k^h \geq \lambda_k$ for $k = 1, \ldots, n$.

This gives us an estimate of the desired quantity $\lambda_h - \lambda$ form below. The next thing to do is to find an estimate from above. Here we just refer to [7] were this matter is discussed and we get the following result

$$0 \leq \lambda_h - \lambda \leq C \|h^p D^{p+1} u\|^2. \tag{6}$$

Higher eigenmodes tends to have eigenvectors which varies a lot on small distances which will make the derivatives of the solution bigger. This means that the approximate solution will be better for low eigenvalues then for high.

It is also possible to derive an a priori error estimate for the eigenvector error in terms of h and derivatives of u. Again this is done in [7] and the result is the following inequality

$$\|u - u_h\| \leq C(1 + \frac{2\lambda}{d})\|h^{p+1}D^{p+1}u\| \tag{7}$$

where $d > 0$ is the distance between the eigenvalue λ associated with u and its closest neighbour eigenvalue. p is as usual the maximum order of the piecewise polynomial in the approximation. Since we will consider eigenvalues of moderate size, and assume that there are well separated, have assumed simple eigenvalues the factor $C(1 + \frac{2\lambda}{d})$ will be of moderate size. These results are derived for homogeneous Dirichlet boundary conditions but should hold also for periodic since the bilinear form a is equivalent in the two cases.

3.2 A posteriori error estimate

In the a posteriori error analysis below we shall consider both the eigenvalue error $\lambda_h - \lambda$ for a chosen eigenvalue and the corresponding eigenvector error $e = u_h - u$. More precisely we shall estimate error functionals of the form (e, ψ) for a given function ψ. In particular we shall consider the case of $\psi = e/\|e\|$ to obtain an $L^2(\Omega)$-norm error estimate of e. To do this we set up a dual problem in the usual way by taking the adjoint of the u-dependent terms in the original problem in the left hand side and put ψ in the right hand side [15] that is $(L - \lambda I)^*\phi = \psi$, where I is the identity. Notice that λ now is a fixed chosen eigenvalue. Since L is a self-adjoint operator we just get

$$\begin{aligned} -\Delta\phi + c\,\phi - \lambda\,\phi = \psi \quad & in\ \Omega, \\ \phi = 0 \quad & on\ \Gamma_1, \\ \phi\ periodic \quad & on\ \Gamma_2. \end{aligned} \tag{8}$$

For the the eigenvector error $e = u_h - u$ we now obtain

$$\begin{aligned} &(e, \psi) \\ &= (u_h, \psi) - (u, \psi) \\ &= a(u_h, \phi) - \lambda(u_h, \phi) - (-\Delta u + cu - \lambda u, \phi) \\ &= a(u_h, \phi) - \lambda_h(u_h, \phi) + (\lambda_h - \lambda)(u_h, \phi), \end{aligned} \tag{9}$$

where we used integration by parts and the fact that $-\Delta u + c\,u - \lambda u = 0$. We recall that we have an orthonormal basis of eigenvectors $\{v_i\}_{i=1}^{\infty}$ of $-\Delta + c$. One of these is the eigenvector $u/\|u\| = v_j$ with corresponding eigenvalue $\lambda = \lambda_j$ and we now seek to express ϕ in terms of the basis $\{v_i\}_{i=1}^{\infty}$. We then notice that our left hand side operator in (8) is singular in the direction of

u. This component of ϕ will not contribute to the product (e,ψ) so to have uniqueness we may as well assume the solution of the dual problem to be orthogonal to u. Moreover, since the range of $-\triangle + c - \lambda$ does not contain u, (but all other eigenfunctions) we need to choose ψ to be orthogonal to u. The idea is now to get rid of the $(\lambda_h - \lambda)(u_h, \phi)$-term in (9), because λ is unknown. Using the equality

$$(\psi, v_i) = (-\triangle\phi + c\phi - \lambda_j\phi, v_i) = (\lambda_i - \lambda_j)(\phi, v_i)$$

we get

$$\phi = \sum_{i\neq j}(\phi, v_i)v_i = \sum_{i\neq j}\frac{(\psi, v_i)}{\lambda_i - \lambda_j}v_i,$$

which leads to

$$\begin{aligned} &|(u_h, \phi)| \\ =\ & |\sum_{i\neq j}\frac{(\psi, v_i)}{\lambda_i - \lambda_j}(u_h, v_i)| \\ \leq\ & \max_{i\neq j}\frac{1}{|\lambda_i - \lambda_j|}\sum_{i\neq j}|(\psi, v_i)(e, v_i)| \\ \leq\ & \max_{i\neq j}\frac{1}{|\lambda_i - \lambda_j|}\|e\|\|\psi\|. \end{aligned} \tag{10}$$

Since we just deal with distinct eigenvalues and we know from the a priori analysis (6) that the eigenvalue error is bounded from above by $C\|h^pD^{p+1}u\|^2$ so that for small enough h we can assume that $(\lambda_j^h - \lambda_j)\max_{i\neq j}|\frac{1}{\lambda_i - \lambda_j}| \leq \delta$, where δ is much smaller then one.

In the case when $\psi = e/\|e\|$ we now get the following bound of the eigenvector error in the L^2-norm

$$(1-\delta)\|e\| \leq |a(u_h, \phi) - \lambda_h(u_h, \phi)|. \tag{11}$$

To get a similar equation for the eigenvalue error we set $\psi = 0$. Then the dual problem (8) becomes equivalent to the original problem but with λ given which means $\phi = u$ is a solution. Here we have also assumed that u is normalised so that $(e, u) = 0$, that is, we consider the eigenfunctions u for which $(u, u) = (u_h, u)$, which is clearly possible provided u_h is not orthogonal to u. In particular, the dual problem is then well posed. From the a priori error estimate of the eigenvector error (7) we have for a small enough h that our solution has the property $(e, e) \leq \delta$ where δ is a number much less than one. Which means that we can write $(u_h, \phi) = (u_h, u) = (u_h, u_h) - (u + e, e) = 1 - (e, e) \geq 1 - \delta$. Inserted in (9) we get

$$(1-\delta)(\lambda_h - \lambda) \leq |a(u_h, \phi) - \lambda_h(u_h, \phi)|. \tag{12}$$

Since (3) holds we are allowed to replace ϕ with $\phi - \pi\phi$ in (11) and (12) where $\pi\phi \in V_h^p$. So we need to estimate the quantity $a(u_h, \phi - \pi\phi) - \lambda_h(u_h, \phi - \pi\phi)$ in order to get an upper error bound for the eigenvalue error and the eigenvector error as desired.

This section have so far followed [12] closely but with a slightly different operator. In [12] the dual problem is used to derive a stability constant which together with the residual gives an error bound on the eigenvalue and eigenvector error. We will here attend to actually solve the dual problem to get information of how the stability influences the error in different parts of the domain. Our final aim is to make an adaptive algorithm for solving these problems more efficiently by calculating the residual and stability and than get a local "error function" defined on each tetrahedron that can be used to refine the mesh on certain parts of the domain.

To achieve this we follow [6] closely by integrating $|a(u_h, \phi - \pi\phi) - \lambda_h(u_h, \phi - \pi\phi)|$ by parts on each tetrahedron in the following way

$$
\begin{aligned}
&|a(u_h, \phi - \pi\phi) - \lambda_h(u_h, \phi - \pi\phi)| \\
&\quad \leq \quad |\sum_{\tau \in \mathcal{T}_h} \int_\tau (-\triangle u_h + c u_h - \lambda_h u_h)(\phi - \pi\phi)\, dx| \\
&\quad + \quad |\sum_{\tau \in \mathcal{T}_h} \int_{\partial\tau} \frac{1}{2h}[n \cdot \nabla u_h](\phi - \pi\phi) h\, ds| \\
&\quad \leq \quad \sum_{\tau \in \mathcal{T}_h} \int_\tau (| - \triangle u_h + c u_h - \lambda_h u_h| \\
&\quad + \quad \frac{1}{2h} \max_{\partial\tau} |[n \cdot \nabla u_h]|)|\phi - \pi\phi|\, dx \\
&\quad = \quad \sum_{\tau \in \mathcal{T}_h} \int_\tau (R_i + R_e)|\phi - \pi\phi|\, dx
\end{aligned}
$$

where $R_i(u_h, \lambda_h)|_\tau = |\triangle u_h - c\, u_h + \lambda_h u_h|$ and $R_e(u_h)|_\tau = \frac{1}{2h} \max_{\partial\tau} |[n \cdot \nabla u_h]|$. $[\cdot]$ denotes the "jump" in the function over an interior boundary. The second term in the first inequality is not on the original form as it would appear after a partial integration. Each interior boundary will be counted twice so we divide the contributions from the normal derivative on these two tetrahedrons with a weight of one half [6].

We can now formulate the error function f defined on each tetrahedron, $\tau \in \mathcal{T}_h$ like this

$$
f(\tau) = f_\psi(\tau) = \frac{1}{vol(\tau)} \int_\tau (R_i + R_e)|\phi - \pi\phi|\, dx. \tag{13}
$$

So for $\psi = 0$ we get $\lambda_h - \lambda \leq C \sum_{\tau \in \mathcal{T}_h} f_0(\tau) \cdot vol(\tau) =: E(\lambda)$, while for the corresponding eigenfunction error $e = u_h - u$ we get $\|e\| \leq C \sum_{\tau \in \mathcal{T}_h} f_\psi(\tau) \cdot vol(\tau)$ where $\psi = e/\|e\|$ and $C = 1/(1-\delta)$. These formulas makes it possible to solve the problem adaptively and they also give an upper bound of the error.

As mentioned before we want to solve the dual problem and since it is equally hard to solve as the original problem we need to use the finite element method for solving this problem as well.

Unfortunately we can not solve the problem numerically as it is stated in (8). Both $\psi = e/\|e\|$ and λ are contained in the formulation of the dual problem and they are both unknown. This means that we need to make approximations of these quantities and we will now analyse how much perturbations affects the solution of the dual problem.

We start by discussing how to approximate $\psi = e/\|e\|$. The only thing we know about ψ for sure is that it has to be orthogonal to u so we have a consistent system of equations and that it has to have L^2-norm one. These properties can be guaranteed also when solving the dual problem numerically. The thing we can not guarantee is that our normed orthogonal function ψ is equal to the normed error. So we want to show that we can choose ψ quite far from e and still get an OK error bound on the eigenvector error in the L^2-norm.

If we combine (9) and (10) we get the following equation

$$|(e,\psi)| \leq |a(u_h, \phi - \pi\phi) - \lambda_h(u_h, \phi - \pi\phi)| + \delta\|e\|\|\psi\|$$

If we let $\psi = (e + \delta e)/\|e + \delta e\|$, where $\|\delta e\| < \|e\|$, we can estimate the left hand side from below in the following way

$$|(e, \frac{e + \delta e}{\|e + \delta e\|})| \geq \frac{\|e\|^2}{\|e\| + \|\delta e\|} - \frac{\|e\|\|\delta e\|}{\|e\| + \|\delta e\|} = \frac{\|e\|(\|e\| - \|\delta e\|)}{\|e\| + \|\delta e\|}$$

which gives us

$$\|e\|(\frac{\|e\| - \|\delta e\|}{\|e\| + \|\delta e\|} - \delta) \leq |a(u_h, \phi - \pi\phi) - \lambda_h(u_h, \phi - \pi\phi)|.$$

Again from the a priori error analysis we assume that $\delta \ll 1$. This means that the a posteriori analysis for the eigenvector (11) still holds but the $(1-\delta)$ in the left hand side is replaced by $(\frac{\|e\|-\|\delta e\|}{\|e\|+\|\delta e\|} - \delta)$. Since e is unknown we can not calculate this factor but we can still see that even if our guess of $e/\|e\|$ is not very accurate we still can get an OK error bound. For example if $\|\delta e\|/\|e\| = 1/2$ we get $(1/3 - \delta)$ which is not to bad.

The next issue to deal with is the fact that the dual problem contains the unknown λ. The only sensible thing to do must be to replace λ with λ_h which is the best approximation we have. To do this we need to ensure that small perturbations in the eigenvalue gives small changes in the solution of the dual problem. We will need to study a new space for the weak formulation of the dual problem to get a well posed problem. Lets call the space of all functions in H_0^1 that are orthogonal to a certain eigenvector u corresponding to the eigenvalue λ in the dual problem V.

We will now state the weak formulation of the dual problem: Find $\phi \in V$ such that

$$a(\phi, v) - \lambda(\phi, v) = (\psi, v), \quad \forall v \in V.$$

The aim in the following theorem is to give an upper bound of how much the solution changes for on small perturbations in λ.

Theorem 3.1 *Let (a) $a(\phi, v) - \lambda_j(\phi, v) = (\psi, v), \quad \forall v \in V$ and let (b) $a(\tilde{\phi}, v) - \lambda_j^h(\tilde{\phi}, v) = (\psi, v), \quad \forall v \in V$. Then we have*

$$\frac{\|\tilde{\phi} - \phi\|}{\|\phi\|} \leq \max_{i \neq j} \left|\frac{\lambda_j^h - \lambda_j}{\lambda_j^h - \lambda_i}\right|. \tag{14}$$

Proof. *Subtract (a) from (b) and we get*

$$a(\tilde{\phi} - \phi, v) - \lambda_j^h(\tilde{\phi} - \phi, v) = (\lambda_j^h - \lambda_j)(\phi, v)$$

but $\tilde{\phi} - \phi \in V \subset L_2$ which means that $\tilde{\phi} - \phi = \sum_{i \neq j}(\tilde{\phi} - \phi, v_i)v_i$. This gives us

$$\begin{aligned}
&(\lambda_j^h - \lambda_j)(\phi, v) \\
&= a\Big(\sum_{i \neq j}(\tilde{\phi} - \phi, v_i)v_i, v\Big) - \lambda_j^h\Big(\sum_{i \neq j}(\tilde{\phi} - \phi, v_i)v_i, v\Big) = \\
&= \sum_{i \neq j}(\tilde{\phi} - \phi, v_i)a(v_i, v) - \lambda_j^h \sum_{i \neq j}(\tilde{\phi} - \phi, v_i)(v_i, v) = \\
&= \sum_{i \neq j}(\tilde{\phi} - \phi, v_i)(\lambda_i - \lambda_j^h)(v_i, v)
\end{aligned}$$

since $a(v_i, v) = \lambda_i(v_i, v)$. With $v = v_k, \ k \neq j$ we get

$$(\tilde{\phi} - \phi, v_k) = \frac{\lambda_j^h - \lambda_j}{\lambda_k - \lambda_j^h}(\phi, v_k).$$

Using this equality we get

$$\|\tilde{\phi} - \phi\|^2 = (\tilde{\phi} - \phi, \tilde{\phi} - \phi) = (\tilde{\phi} - \phi, \sum_{i \neq j}(\tilde{\phi} - \phi, v_i)v_i) =$$

$$= \sum_{i \neq j}(\tilde{\phi} - \phi, v_i)\frac{\lambda_j^h - \lambda_j}{\lambda_i - \lambda_j^h}(\phi, v_i) \leq \max_{i \neq j}\left|\frac{\lambda_j^h - \lambda_j}{\lambda_j^h - \lambda_i}\right| \|\tilde{\phi} - \phi\| \|\phi\|$$

which proofs the theorem. ∎

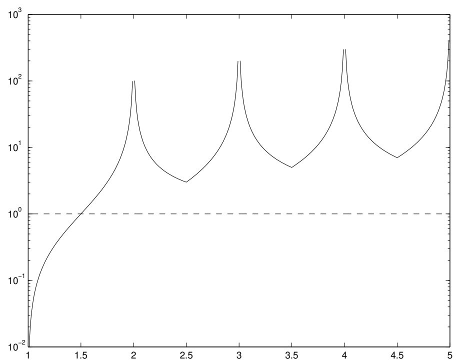

Figure 2: The upper bound of the error from (14) is plotted versus λ_h. The integers on the x-axis refers to eigenvalues λ_i starting with the one we want to approximate.

As long as the approximate eigenvalue is fairly close to the exact eigenvalue, that is much closer to the right one then to any other eigenvalue in the spectrum, will the solution to the modified problem be close to the solution of the exact problem. But if the approximate eigenvalue is so poor that it coincides with an other exact eigenvalue we will have no idea of how good the associated dual solution is from the estimate. This is not strange since we then get the dual solution associated with an other eigenvector which may vary a lot from the one we wanted. To get a better qualitative view of this we will plot (figure (2)) the right hand side in the theorem above versus the parameter λ_h in a special case. Notice the dashed line where the relative error is equal to one. When the right hand side is close to or bigger than that we can not say anything about the accuracy of the solution since the error is of the same order as the solution itself. Again we draw the conclusion that we need a good approximation of λ to get a good approximation of ϕ. We also notice the continuous behaviour of the right hand side close to the exact eigenvalue which gives convergence to ϕ when $\lambda_h \to \lambda$. It is important to notice that this theorem is also of interest when calculating the eigenvalue error. It means that we just have to worry about discretisation errors since we have control over the modification of the equation.

As a summary of the last part of this section we can say that we have noticed the singularity of the dual problem, understood that the solution in a certain direction is not interesting. Then reformulated the problem to the interesting part of the solution and got rid of the singularity and proved that the solution of the dual problem depends continuously on changes in the eigenvalue λ.

4 Implementation and numerical aspects

Lets go back to the discrete weak form of the original problem (4) with u_h replaced with $\sum_{i=1}^{n} U_i \varphi_i$. For all $j \in 1 \ldots n$

$$\begin{aligned} & 0 \\ = \; & a(\sum_{i=1}^{n} U_i \varphi_i, \varphi_j) - \lambda_h(\sum_{i=1}^{n} U_i \varphi_i, \varphi_j) \\ = \; & \sum_{i=1}^{n} U_i \, a(\varphi_i, \varphi_j) - \lambda_h \sum_{i=1}^{n} U_i \, (\varphi_i, \varphi_j) \end{aligned}$$

or on matrix form:

$$A_1 U = \lambda_h M_1 U \tag{15}$$

where $\{A_{1\ i,j}\} = a(\varphi_i, \varphi_j)$, $\{M_{1\ i,j}\} = (\varphi_i, \varphi_j)$ for all $i, j \in \{1, \ldots, n\}$ and U is a $(n \times 1)$ vector containing the U_i values.

Similarly we get the following equation for the dual problem

$$A_2 \Phi - \lambda M_2 \Phi = F \tag{16}$$

where F is a $(n \times 1)$ vector containing the nodal values of the scalar product (ψ, φ_i) for all $i \in \{1, \ldots, n\}$. The matrices A_1, A_2 and M_1, M_2 depends on the method and mesh which do not have to be the same for the original and dual problem.

There are some good reasons for solving the two problems with the same method and mesh thou. The first is of course that time can be saved since the matrices are already calculated. An even more important thing is that U will be known which means we know the null space of our matrix and we can use this to find the solution to the dual problem orthogonal to U. This reason is in fact good enough by itself as we will see later in this section. We can also see that it is going to be a big problem if our approximate eigenvalue λ_h is close to the corresponding eigenvalue of the alternative discretisation (which we hope it is!). This is because we can not take away the singular direction in the new discretisation since we have not solved the eigenvalue problem on this mesh which means that we will most likely end up with very ill-conditioned or singular matrices that we can not solve in an effective way.

A fourth reason comes from the discrete version of theorem (3.1) which reads

$$\frac{\|\tilde{\Phi} - \Phi\|}{\|\Phi\|} \leq \max_{i \neq j} |\frac{\lambda_j^h - \lambda_j}{\lambda_j^h - \lambda_i^h}| = C|\lambda_j^h - \lambda_j|$$

where we instead of the eigenvalues λ_i of the operator L now have the eigenvalues λ_i^h of the matrix $M^{-1}A$. This is derived by matrix pertubation theory. The constant C can now be calculated since the eigenvalues of $M^{-1}A$ are known and we get a good measure of the effect of replaceing λ with λ_h. We leave the this discussion and focus on how to construct the matrices A and M.

4.1 Basis functions and matrix assembling

We need to choose a proper set of basis functions for V_h^p. In this paper we will focus on two different choices.

Definition 4.1 (piecewise linear basis function) *Given a mesh we define ϕ_i, the i:th basis function, to be the piecewise linear function with value one in the i:th node of the mesh and zero in all other nodes.*

So we have n different basis functions all with local support. On each tetrahedron just four basis functions will be non-zero. Lets call these ϕ_1, ϕ_2, ϕ_3 and ϕ_4. They all have the value one in one node and the value zero in the other nodes on the tetrahedron. Since a tetrahedron has four nodes we get four different combinations. The resulting method with this choice of basis functions is often refereed to as cG1 or continuous Galerkin of order one. The other case is piecewise quadratic functions and can be defined like this:

Definition 4.2 (piecewise quadratic basis function) *Given a mesh we construct a new node in the middle of each edge on each tetrahedron. This means that each tetrahedron will consist of ten nodes. We define ψ_i, the i:th basis function, to be the piecewise quadratic function with value one in the i:th node of the new mesh (now consisting of more than n nodes) and zero in all other nodes.*

If we again look locally on one tetrahedron we will now have ten different basis functions with non-zero values. Lets call these functions $\{\psi_i\}_{i=1}^{10}$. We can construct them from our four linear functions on each tetrahedron in the following way:

Let $i = \{corner\ nodes\}$, then $\psi_i = 2\varphi_i^2 - \varphi_i$. Let $l = \{new\ node\ between\ j\ and\ k\}$, then $\psi_l = 4\varphi_j\varphi_k$. In analogy with the piecewise linear case this method is called cG2.

Now we're ready to discuss how to evaluate the A and M matrices. There are mainly two problems that arises here, the first problem is how to evaluate the scalar products for each entry in the matrices. Since the terms $(\nabla\varphi_i, \nabla\varphi_j)$ and $\lambda_h(\varphi_i, \varphi_j)$ has polynomials in the integrand they can be calculated exactly using the formula [3]

$$\int_\tau \phi_1^{m_1}\phi_2^{m_2}\phi_3^{m_3}\phi_4^{m_4}\,dx = \frac{3!\,m_1!\,m_2!\,m_3!\,m_4!}{(m_1+m_2+m_3+m_4+3)!}\,vol(\tau) \tag{17}$$

where $m_i \in \mathbf{N}$. In the cG2 case we just need to express the quadratic basis functions in terms of the linear as described above. This means that the integral is mainly independent of which tetrahedron it is calculated on. The only tetrahedron dependent factor is the volume . This can be used to speed up the matrix assembling algorithm a lot.

The remaining part to be evaluated now is $(c(x)\,\varphi_i, \varphi_j)$. Since $c(x)$ can be an arbitrary positive function it is often impossible to calculate the scalar product exactly. There are two ways to go from here, one is simply to approximate $c(x)$ with the value in the centre of each tetrahedron and move it outside the scalar product and proceed as above. The other way is to numerically calculate approximations to the whole product $(c(x)\,\varphi_i, \varphi_j)$ by calculate the integrand values in certain points called *Gauss points* and calculate a weighted sum of these values. For a discussion of how this is done in one dimension see [16].

Depending on the number of points it is possible to achieve exact integration up to a given polynomial degree and then hopefully a good approximation to the exact value. The problem in three dimensions is that the number of points increases rapidly for higher order of accuracy.

In the dual problem the products in the F-vector is calculated in the same way as the products in the matrices A and M. Whether *Gauss points* is used here or not depends on how hard the function ψ is to integrate.

The second problem is how to administrate all these different products in a computer code. To describe this we need to know how the mesh information is stored in the computer. The software used to generate mesh and plot solutions for this paper has been *FEMLAB* made by *COMSOL*. The mesh generator uses the *Delaunay algorithm* and stores the mesh information in three matrices p, t end e. The p-matrix contains three rows with the space coordinates for each node. The t-matrix contains one column for each tetrahedron. Each column mainly consists of the four nodes associated with that tetrahedron. If cG2 is used each column have to contain ten nodes. The e-matrix contains information of the boundary. Among other things each column contains the three node associated with each boundary triangle. The number of columns is the number of boundary triangles. With this in mind we can explain roughly how the matrix assembling is done in *MATLAB* using the cG1 method with the potential approximated by the centre value of each tetrahedron.

First allocate memory for the two matrices A and M. Then calculate the first part of (17) once for all combinations of the four basis functions, that gives a (4×4)-matrix, then loop through all tetrahedrons and calculate the gradients of the basis function, the volume of the current tetrahedron and the value of the function c. Finally add the contributions from this tetrahedron to the rows and columns in the matrices associated with the current tetrahedron in A and M. For a more extensive discussion of this matter see [15].

4.2 Boundary condition

The boundary conditions considered in this paper are homogeneous Dirichlet, that is $u = 0$, on $\partial\Omega \setminus \Gamma_2$ and periodic boundary conditions on Γ_2, where Γ_2 is either empty or the four sides of figure (1) perpendicular to the xy-plane. To get Dirichlet conditions we just need to take away the rows and columns associated with the boundary nodes from the matrices (and vector in the dual case). This works since we already know that the solution has the value zero in these nodes so associated rows and columns will have no influence on the solution in other nodes.

Periodic boundary conditions means that the solution repeats itself over the boundary to an imagined identical domain. They are a bit more tricky for several reasons. First of all the periodic condition must be applied on two identical parts of the boundary because there must exist two part with the same solution if the solution is periodic. These two parts must also have the same distribution of mesh points so each node on the first part has a "twin" node on the other part. Then it is important to give one of these parts the status of being the master and the other one to be the slave since we just want the actual equation system to contain interior nodes and master nodes. The slave nodes will just be given the same value as their twin master nodes. Unfortunately we can not just take away the rows and columns associated with the slave nodes, as in the case with Dirichlet conditions, because its "neighbours" (closest nodes) will now also be neighbours to the master nodes. So before erasing the rows and columns associated with the slave nodes the neighbour information must be passed on the rows and columns associated with master nodes. It is of course possible to mix Dirichlet and periodic condition as long as it is on different parts of the boundary and the "periodic part" fulfils the demands described above.

This ends the problem of calculating the matrices and vector needed to solve the discrete problems (15) and (16).

4.3 Numerical methods for solving eigenvalue problems

There are two main methods for solving eigenvalue problems, direct and iterative. Direct methods are used when all eigenvalues and optionally eigenvectors are wanted for a given matrix. The complexity of the direct methods are normally of the order $\mathcal{O}(n^3)$. In *MATLAB* the direct solver in is called *eig*. To read more about direct methods see [10]. Iterative methods are used when just some eigenvalues and eigenvectors in the spectra are desired. *MATLAB:s* iterative solver is called *eigs*.

Since the matrices that arises from a finite element matrix assembling are sparse and big the choice of method in this case is really simple. First of all would it be hard to store all eigenvectors that comes from the direct method because they do not need to have the same sparse structure as the original matrix, the complexity is to high, it would take to much time to solve the system and last in our application the only eigenvalues of interest are the lowest ones. So everything is then set to use iterative methods.

Let $Au = \lambda u$ be our example problem[3]. To get an idea of how an iterative

[3]Eigenvalue codes works for the equation $Au = \lambda Mu$ as well.

algorithm works we first need to define *Krylov subspace*.

Definition 4.3 (Krylov subspace) *The k-dimensional Krylov subspace* $\mathcal{K}_k(A, x)$ *is* $span(x, Ax, A^2x, \ldots, A^{k-1}x)$.

We can see that the power method [10] for finding the eigenvector corresponding to the highest eigenvalue given an initial vector x_1, spans the Krylov subspace $\mathcal{K}_k(A, x_1)$ with its k first approximations. In the power method the vector $x_k = A^{k-1}x_1$ is chosen as an approximation of the eigenvector corresponding to the highest eigenvalue, in more sophisticated methods the "best" approximation in the Krylov subspace is chosen instead of just x_k. In fact this subspace can be used to calculate the k first eigenpairs by using the Rayleigh-Ritz method. If other parts of the spectrum are desired we need to use inverse iteration with shift which means that we have to look for an optimiser in $\mathcal{K}_k((A - \sigma I)^{-1}, x_1)$ instead.

Let K denote the Krylov subspace $[x_1, Ax_1, \ldots, A^{k-1}x_1] = [x_1, x_2, \ldots, x_k]$. Then $A\,K = [x_2, \ldots, x_k, A^k x_1]$. Assume that K is non-singular, then let $c = -K^{-1}A^k x_1$ which gives $A\,K = K\,[e_2, e_3, \ldots, e_k, -c] \equiv K\,C$, where e_i is the i:th column in the identity matrix. C is upper Hessenberg [5] and $K^{-1}AK = C$. Let $K = QR$ be the QR-decomposition of K. Then we have

$$K^{-1}AK = (R^{-1}Q^T)A(QR) = C \;\Rightarrow\; Q^T AQ = RCR^{-1} = H. \qquad (18)$$

R and R^{-1} are both upper triangular (think of how you would calculate R^{-1} with the Jacobi method) and C is upper Hessenberg which means that H is upper Hessenberg. Since we are working with symmetric matrices this means that H is also lower Hessenberg hence tridiagonal since $H^T = (Q^T AQ)^T = Q^T A^T Q = Q^T AQ = H$. The columns in Q can be computed one at a time using the Arnoldi or Lanczos algorithm described in [5].

Let now $Q = [Q_k, Q_u]$ where Q_k is the first k columns of Q calculated with for example the Lanczos algorithm and Q_u are unknown columns of Q. Then the eigenvalues of A can be approximated by the Ritz values which are the eigenvalues of the tridiagonal matrix $T_k = Q_k^T AQ_k$. The corresponding eigenvectors are given by $Q_k V$ if $T_k = V\Lambda V^T$ is the eigendecomposition of T_k. It can be shown [5] that this approximation is the best in the current Krylov subspace in the sense that $\|A(Q_k V) - Q_k V\Lambda\|_2$ is minimised.

As mentioned before the built-in *MATLAB* function that uses iterative methods is *eigs*. There is an other code written by A. Ruhe at Chalmers University of Technology more suitable for solving big sparse eigenvalue problems called *fleig* which have been used in this paper. This program is included in *FEMLAB*.

4.4 Algorithms for solving singular systems of linear equations

The matrix equation that arises from the dual problem (16) may at first sight look like an ordinary system of linear equations. Well it is a system of linear equations but it is singular since $\Phi = \Phi_0 + \alpha v$, where $\alpha \in \mathbf{R}$ and v is the null space of $B = A - \lambda M$, solves the system as well. To get a consistent system we need as mentioned before to ensure that the right hand side has no component in the "singular direction". This means that it has to be orthogonal to U with respect to the scalar product $(v, w) = v^T M w$ which is the discrete variant of the L^2 scalar product. From the a posteriori analysis (3.2) we see that we just want to find a solution to (16) that is orthogonal to the null space of B. This can be done in different ways.

We have a matrix $B \in \mathbf{R}^{n\times n}$ with rank $n-1$ since if Φ is a solution to $B\Phi = F$ then so is $\Phi + v$ where $v \in null(B)$. We want to find the component Φ_0 of Φ that is orthogonal to v. This is the same as finding the solution Φ that has minimal 2-norm.

Let $[U, \Sigma, V]$ be the SVD of B with $U = [u_1, \ldots, u_n]$, $V = [v_1, \ldots, v_n]$ and $\Sigma = diag(\sigma_1, \ldots, \sigma_{n-1}, 0)$. Then it can be shown [9] that $\Phi = \sum_{i=1}^{n-1} \frac{u_i^T b}{\sigma_i} v_i$ minimises $\|B\Phi - F\|_2$ as desired. This can also be expressed in terms of the *Pseudo-Inverse*, B^+, of B. $B^+ = V\Sigma^+ U^T$ where $\Sigma^+ = diag(\frac{1}{\sigma_1}, \ldots, \frac{1}{\sigma_{n-1}}, 0)$. In this notation $\Phi = B^+ F$. The pseudo-inverse can be found in *MATLAB* and is called *pinv*.

This way of solving the problem may work in theory but it is not a good solution because it takes too much computation time. For a typical problem will the computation of the pseudo inverse could take ten times longer than all other computations together which is not acceptable. A better approach is to use the fact that we know the solution U of the original matrix problem (15) which means that we can construct the matrix $P = I - \frac{U(MU)^T}{U^T MU}$ which filters out the singular direction of the matrix. For example $U^T M(Pv) = U^T Mv - U^T M \frac{U(MU)^T}{U^T MU} v = U^T Mv - (MU)^T v = 0$ for any vector v since M is symmetric. By multiplying both sides of (16) with this matrix P we will remain in the orthogonal complement of U when we use iterative methods for solving the system of equations which means that we are solving the system $PB(P\Phi) = PF$ instead. This problem have been studied by G. Sleijpen and H. A. Van der Vorst among others and it occurs when deriving the correction in the Jacobi Davidson algorithm. To read more about this see [2].

The best thing to do from here, in fact this is almost always the best way to go when dealing with three dimensional problems, is to use iterative methods to solve the system of equations. Depending on the choice of eigenvalue in the dual problem we will get either a positive semi definite matrix or an indefinite matrix. These different cases require different iterative algorithms. In the first case, which only occurs for the lowest eigenvalue, conjugate gradients can be used. In the indefinite case we use the minimum residual method. It could of course also work for the lowest eigenvalue but conjugate gradient is a very fast algorithm and therefore better to use if possible. For an extensive explanation of these methods see [9]. These methods are so well implemented[4] that the time of solving the dual problem here is in the same order as the matrix assembling.

4.5 Adaption

We are now ready to discuss the adaptive algorithm which has been one of our final aims in this paper. The theory that will be used is all presented in the error analysis chapter. The equation of most interest is the error function (13) which as we will see gives all information about the refinement of the mesh we need.

The first thing to do when using an adaptive algorithm based on an a posteriori estimate is to solve the discrete version of the original problem (15) to get the approximate solutions to the eigenvectors and eigenvalues of the problem. The next step is to choose which eigenpair (u_h, λ_h) we want to analyse and improve our solution for. We will discuss the algorithm for eigenvector adaption since the eigenvalue analysis does not include the solving of a dual problem.

Given an approximate eigenpair we need to calculate the different components in the error function (13). The residual part is done by calculating the contribution for the interior and the boundary of the tetrahedrons separately. Since we are using piecewise linear functions the Laplacian will be zero in the interior. The values of $c(x)$ is just approximated by its value in the centre of the tetrahedron as when the original problem is solved. This value is taken minus the approximate eigenvalue and multiplied by the approximate solution in the centre of the tetrahedron. For the boundary part of the residual we use the fact that the gradients of linear basis function has already been calculated while deriving the A matrix. Since we are dealing with "jumps" here we also need to know which two tetrahedrons that lives on each side of every interior surface. This can be done by creating a structure with seven values for each surface. The first three are the nodes associated with the surface. Four and five are associated with one of the tetrahedrons that borders the surface. Five is the tetrahedron number and four is the local index of the corner of that tetrahedron which is not in the surface. Six and seven are the same for the tetrahedron on the "other side". This structure makes it possible to derive the "jumps" in the gradients of the approximate solutions. Finally the "jump" terms of the four (at most) surfaces associated with the current tetrahedron are compared and the max-

[4]Here have Fischers implementation been used from [8].

imum is taken. The total residual can now be calculated by adding these two contributions and take the absolute value.

For the other term in the error function we obviously need to solve the dual problem. Since we have the same method and mesh all the matrices will be the same. The only new thing to derive is the vector F in the right hand side. We then need to choose a function ψ. It is supposed to match the normed error $e/\|e\|$. We know that it needs to be orthogonal to u which in the discrete case means $F^T MU = 0$. This can be done by multiplying F with the matrix P mentioned earlier. It also needs to be normed which can be done by dividing it with $(\vec{\psi}^T M\vec{\psi})^{1/2}$, where $\vec{\psi}$ is a vector containing the nodal values of ψ. Then the vector F is calculated in the usual way by multiplying a test function and integrate over the domain. The dual problem can now be solved using methods described earlier. The dual solution is considered to be exact throughout the whole error estimate. This is important to have in mind when we approximate it with the numerical solution. We can not just subtract the nodal linear interpolant from it as written in (13) since we then will end up with the result zero. So we need to think about what this quantity is supposed to describe. It is a continuous function subtracted by a linear interpolant and from the estimates $\|\phi - \pi\phi\| \leq C\|h^2 D^2\phi\|$ we can see that it behaves like the second derivative. One way of getting the essence from this behaviour from a piecewise linear function is to subtract an other piecewise function from a more sparse mesh. This is done by creating a local interpolant on each tetrahedron by making a nodal interpolant from the four closest nodes outside the current tetrahedron, that is the four nodes not included in the current tetrahedron that belongs to the four surface neighbours to the tetrahedron. Both the dual solution and the interpolant are evaluated in the centre of each tetrahedron and the difference will after taking the absolute value be called the stability function.

We are now ready to multiply the residual and the stability function on each tetrahedron and perform the integration from (13). Since the functions are considered as constants on each tetrahedron the integration just means multiplying the integrand with the volume of the tetrahedron. To get a mean value of the error on each tetrahedron we finally divide the result with the volume. This gives us a piecewise constant function that defines an upper bound of the eigenvector error on each tetrahedron. From this function we determine which tetrahedrons that will be refined and how much. We know from the a posterior theory [12] that the eigenvector error depends on the square of the mesh parameter h. With this in mind we can find how many times we will divide each tetrahedron to get an equidistributed error on the whole mesh. So what we finally end up with is a vector telling us how many times we shall divide each tetrahedron. This is given to the mesh generator and we will get a refined mesh that we can use to calculate a new solution to the problem.

It should be mentioned that it can be quite hard to succeed with the

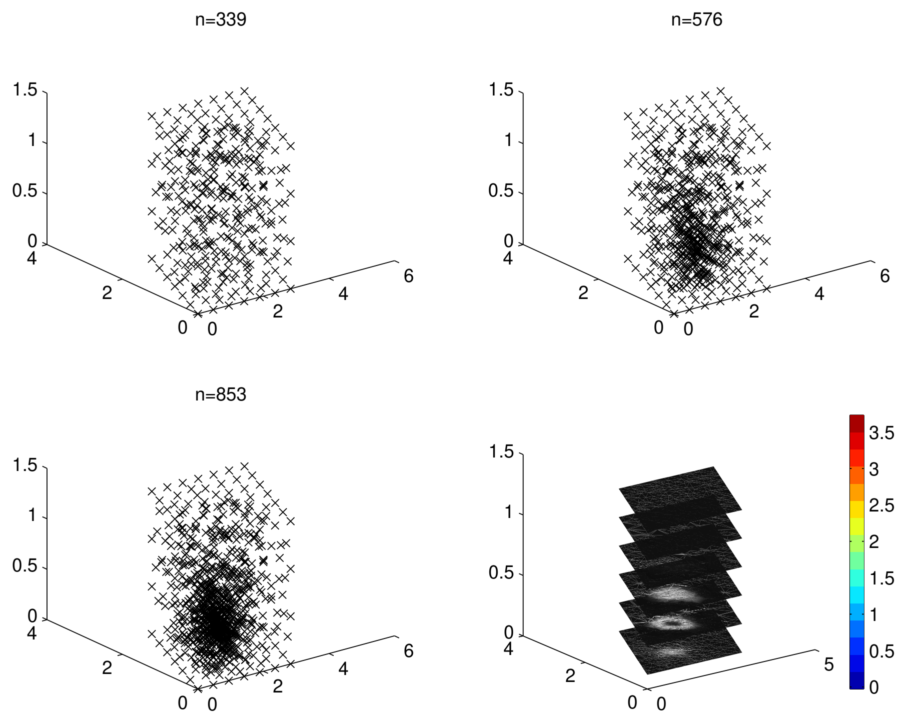

Figure 3: The three first plots shows the mesh after zero, one and two runs through the adaptive algorithm for the first eigenvalue. Each mesh point is marked by a $\times$. The fourth plot shows slices from the first eigenmode.

equidistribution of the error in one iteration and thats why we often uses the adaptive algorithm several times to get a small error. Unfortunately there is one problem that limits the number of adaptive refinements. Since the tetrahedrons in each refinement are divided they will have worse quality after each run through the adaptive algorithm. That is why we just uses the adaptive algorithm a few times and then start over again with a finer quasi-uniform mesh.

Figure (3) shows a sequence of refined meshes using the algorithm above on the hydrogen-platinum problem described in section (5.2) with homogeneous Dirichlet boundary conditions. The adaption is done for the eigenvalue error of the first eigenvalue. We can clearly see on the three first pictures how the new mesh points tends to focus in a small part of the domain. The fourth picture shows six slices of the solution.

As we can see the lowest eigenvector has the value zeros almost everywhere except in the lower left corner. This is probably the strongest reason for using adaptive FEM on this problem.

5 Application in Quantum physics

We will now leave the general discussion and focus on a certain eigenvalue problem from the field of quantum physics. To do this we need some background information about the Schrödinger equation.

5.1 The wave equation of Quantum Physics

Before we present a motivation of the formulation of Schrödinger equation lets underline that this is *not* a proof of the equation and there exist no such proof either. The only proof you will get is that the equation seems to give really reliable results compared to experiments.

After Maxwells work with the theory of electro-magnetics in the second half of the 19:th century and experiments with interference there where no doubt about that electro-magnetic radiation was a wave phenomenon rather than a particle one. This picture changes drastically in the early years of the 20:th century. Albert Einstein wrote a very famous paper about the photo electric effect in 1905 where light (electro-magnetic radiation) must be considered as particles or light quanta (photons). This meant that there where clear evidence for both interpretations at that time and this problem is often refereed to as the particle-wave duality of light.

In 1924 the French physicist de Broglie took a big step towards understanding the physics in really small scales. He claimed that the duality of light in fact also existed for other particles. This meant that all particles has an associated wave package. It can be described by a wave vector $\vec{k}$ pointing in the direction of the wave with magnitude $2\pi/\lambda$ where λ is the wavelength and an associated energy E. De Broglie came up with the following famous results in his doctoral theses

$$E = \hbar\omega,$$
$$\vec{p} = \hbar\vec{k},$$

where ω is the angular frequency and $\vec{p}$ is the momentum. The aim was now to find a mathematical expression for this wave package as a function of space and time that could describe the likelihood of finding the particle in position $\vec{r}$ at time t. It was also important that the solutions could be superpositioned to be able to describe interference phenomenon which means that the desired wave equation should be linear and with coefficients independent of quantities depending on the movement of the particle.

Starting with a plane wave, denoted $\Psi(\vec{r}, t)$, and de Broglies laws we get

$$\Psi(\vec{r}, t) = e^{i(\vec{k}\cdot\vec{r}-\omega t)} = e^{\frac{i}{\hbar}(\vec{p}\cdot\vec{r}-Et)}.$$

Than we construct an arbitrary wave function by superposition of plane ones with different wave vectors

$$\Psi(\vec{r}, t) = \int a(\vec{p}) e^{\frac{i}{\hbar}(\vec{p}\cdot\vec{r}-Et)} d^3\vec{p}.$$

Along with this information we know that for a particle in a potential we have $E = \frac{\vec{p}^2}{2m} + V(\vec{r}, t)$, where m is the mass of the particle and $V(\vec{r}, t)$ is the potential, that is the energy of a particle is the sum of the kinetic the potential energy. We can now formulate the following partial differential equation

$$i\hbar\frac{\partial\Psi}{\partial t} + \frac{\hbar^2}{2m}\triangle\Psi - V\Psi = \int a(\vec{p})[E - \frac{\vec{p}^2}{2m} - V]e^{\frac{i}{\hbar}(\vec{p}\cdot\vec{r}-Et)}d^3\vec{p} = 0.$$

This gives us finally the Schrödinger equation for a particle with mass m in a potential $V(\vec{r}, t)$

$$i\hbar\frac{\partial\Psi}{\partial t} = -\frac{\hbar^2}{2m}\triangle\Psi + V\Psi.$$

The quantity $|\Psi(\vec{r}, t)|^2$ describes the likelihood to find the particle at position $\vec{r}$ at time t.

If the potential term is independent of time we can separate the space and time dependence in the equation as follows

$$\Psi(\vec{r}, t) = u(\vec{r})v(t)$$

which gives us

$$i\hbar u\frac{\partial v}{\partial t} = -\frac{\hbar^2}{2m}v\triangle u + Vvu$$

after division by uv we get

$$\frac{i\hbar}{v}\frac{\partial v}{\partial t}(t) = \frac{1}{u}[-\frac{\hbar^2}{2m}\triangle u + Vu](x) = constant = E.$$

This gives the following two equations for u and v respectively

$$-\frac{\hbar^2}{2m}\triangle u + Vu = Eu \tag{19}$$
$$i\hbar\frac{\partial v}{\partial v} = Ev.$$

The first of these equations is often refereed to as the time-independent Schrödinger equation and is sometimes written as

$$\hat{H}u = Eu$$

where $\hat{H} = -\frac{\hbar^2}{2m}\triangle + V$ is called the *Hamiltonian operator*. The trivial solution to the second equation gives that $|\Psi| = |u|$ which means that the interesting quantity is time-independent.

So given a particle, with known mass, and a potential we can now solve the time-independent Schrödinger equation and get a picture of in which state the particle can be found. We will also get corresponding energies for these states as eigenvalues of our *Hamiltonian operator*.

5.2 The Hydrogen-Platinum problem

The interaction of hydrogen with metal surfaces has become a very interesting field lately due to the many applications that exists. Among other thing we can learn a lot about how to store H_2 which can help us to replace the fossil fuels used in cars today with fuel cells which is much better for the environment. There are mainly two reasons for this. Fuel cells use hydrogen which is a renewable resource, we will eventually run out of oil. The rest products from the fuel cells gives no pollution and does not harm the ozone layer which is not the case with oil. We can already see that some American cities tries to lower the amount of cars running on gasoline which means that there also exist commercial reasons to do research in the field of fuel cells. In this paper we will specially study a platinum (Pt) surface but it is important to point out that all the theory and results in this paper can directly be applied just by specifying an other material (potential). Platinum is often used in catalytic converter which is a vital part of the fuel cell.

The question we ask now is how does the eigenmodes of a hydrogen atom laying on a platinum surface look like and what energy levels (eigenvalues) do they have? The mass in the Schrödinger equation (19) will be the proton mass[5]. The units can of course be chosen in several ways but we use eV, fs and Å to make the matrix elements moderate in size[6]. This defines $\hbar$ and the unit of the potential in the equation. The crucial coefficient $c(x)$ is an approximation of the potential the hydrogen atom detects form the platinum surface and it is calculated numerically in certain points in the domain. This is by no means trivial since the potential is generated by a large number of Pt atoms but we will not go in to that problems here, see [11] for more information in this area. We just assume that a potential is given in these points. Then interpolation is done to the actual mesh points. A potential is defined only up to a constant since the zero level can be chosen arbitrarily. For simplicity we choose the zero level to be the lowest potential value in the domain. The coefficient $c(x)$ will be bounded by a constant C so we have fulfilled all the requirements of the operator we had in the theory.

To specify the domain in which the equation is solved we need to look at the structure of the Pt(111) surface [14]. In figure (4) the atoms are represented as big circles. From the structure of the atoms it is easy to see that just every second gap between three atoms can have an atom directly under it. This reduces the symmetry so the smallest cell becomes the rhombus cell shown in figure (1). The domain is of course three dimensional. So the ideal would be to take a prism that starts from the centre of the highest layer and goes to infinity since the likelihood of an hydrogen atom to penetrate through the first layer is very small. Unfortunately this is numerically

[5]The electron associated with the hydrogen atom is assumed to mix up with the valence electrons from the metal.

[6]We want to avoid ill-conditioned matrices.

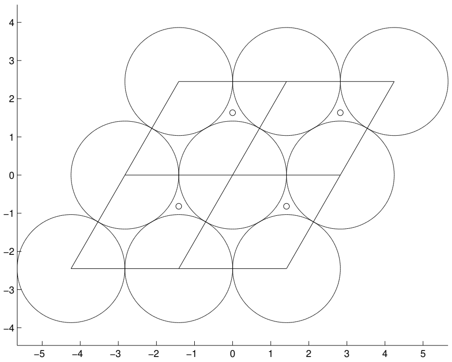

Figure 4: The big rings represents platinum atoms in the highest layer and the small rings indicates where the atoms are in the second highest layer. The surface is divided into identical cells which can cover the hole surface.

impossible so instead we just cut the prism when the potential from the surface gets to big which means that we approximate the solution most likely will be zero on the outer boundary. For these reasons we get homogeneous Dirichlet conditions in the xy-planes.

On the other boundaries periodic conditions would be the natural thing to have since the cell structure repeats itself. There are no reasons for the solution to look different on different cells. For physical reasons which we will not go into here it is of interest to consider homogeneous Dirichlet conditions on these boundaries as well. Both cases are studies in this paper. When we from here on talk about periodic boundary condition we mean the surfaces perpendicular to the xy-plane since we always have Dirichlet conditions on the two planes parallel to the xy-plane. Dirichlet boundary conditions will mean homogeneous Dirichlet conditions on *all* boundaries.

In figure (5) the minimum value of the potential in the z-direction is plotted in the xy-plane. We can see two wells, one down to the left and one up to the right. The first one is often refereed to as the fcc-site and the second one as the hcp-site.

From this plot we would expect that the lowest eigenmode will be localised in the fcc-site and the second lowest will be localised in the hcp-site. These two eigenmodes are two of the most interesting in the spectra and the typical measures the physicists are interested in is the difference in energy between the "fcc mode" and the "hcp mode". Another measurement of interest is how much the lowest eigenvalue changes when the periodic boundary conditions are replaced by homogeneous Dirichlet. This is called the bandwidth of the hydrogen ground state. For a more extensive discussion of these matters see [11]. We will also calculate the two lowest eigenmodes for a deuterium atom in the platinum potential. This is simply done by

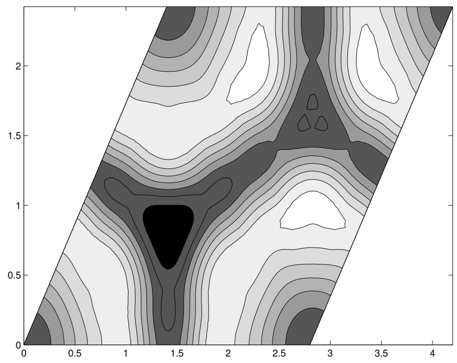

Figure 5: The minimum in the z-direction of the potential from the platinum surface plotted in the xy-plane.

multiplying the mass in the Schrödinger equation by a factor two.

There are more modes that are of physical interest but we will focus on these two in this paper. Mainly for two reasons. The first is that these modes together with a third one called the "top mode" has the most localised eigenvectors which means that the adaptive algorithm will work best on these. The reason why the top mode is not studied is that certain limitations in the *FEMLAB* mesh refinement code made it impossible to make an adaptive mesh for this eigenmode[7].

In the lower right picture in figure (3) we can see a typical solution in the "fcc mode". The solution is zeros almost in the whole domain which means that the hydrogen atom will most unlikely be found outside the "fcc site". The situation with the second eigenmode is very similar. We want to use the fact that the two lowest eigenmodes are very localised to small parts of the geometry to improve the solution by using more mesh points in these areas.

The adaptive FEM algorithm does exactly this in a systematic way since the aim with the algorithm is to minimise the error. It also gives an error bound of the approximate eigenvalue and eigenvector which of course is an important feature.

[7]Version 2.1 dose not support mesh refining for periodic boundary conditions. It is possible to go round this problem if the refinement is inside the domain but it is not so easy if the refinement needs to be done on the boundary.

6 Results

Before we present the results of the hydrogen-platinum problem we will study a test problem with known solution to ensure that the solver behaves properly.

6.1 Test problem

The test problem is created simply by replacing $c(x)$ in (1) with zero and letting Ω be the unit cube. That is

$$\begin{aligned} -\triangle u - \lambda u = 0 &\quad in\ \Omega, \\ u = 0 &\quad on\ \partial\Omega. \end{aligned}$$

To test the implementation of the matrix assembling we compare the first eigenvalue of the discrete problem with the exact first eigenvalue which is $3\pi^2$.

Figure (6) (left) shows how the error depends of the number of nodes in the mesh for the cG1 method. The dashed line is a approximated polynomial of the form Cn^k where n is the number of nodes in the mesh. C and k have been found using the least square method to be about 140 and -0.7 . We can get some understanding of why we got this dependents of n even if we do not have a totally uniform mesh. For a uniform quadrilateral mesh we have $h^3 n = vol$. To get a qualitative approximation the distances between nodes h we can use the same dependence between n and h for the almost uniform (quasi-uniform) mesh generated by *FEMLAB* if we approximate h by some constant in the whole domain. This can be motivated in the following way. The number of nodes is proportional to the number of tetrahedrons in the mesh. The volume of each tetrahedron can be written as Kh^3 where $0 < K_0 \leq K < 1$. So if we assume that the function h is fairly constant for the initial mesh we have that $h^3 t \sim vol \sim 1$, where t is the number of tetrahedrons, which means that $h \sim n^{-1/3}$.

From the a priori analysis we have that the error depends on h^2 which in terms of n would mean $n^{-2/3}$ which is close the value -0.7 we got from our guess of the form of the error. In the cG2 case we see the same behaviour now with $C \approx 1100$ and $k \approx -1.4$ which indicates an error dependence of the form h^4 which also agrees with the a priori estimate.

From this we draw the conclusion the matrix assembling seams to behave as we expected and that the discretisation error for linear basis functions is of the order h^2 which gives us information about how well we need to approximate the function $c(x)$ so the error will be of the same order as the discretisation error of the other matrices.

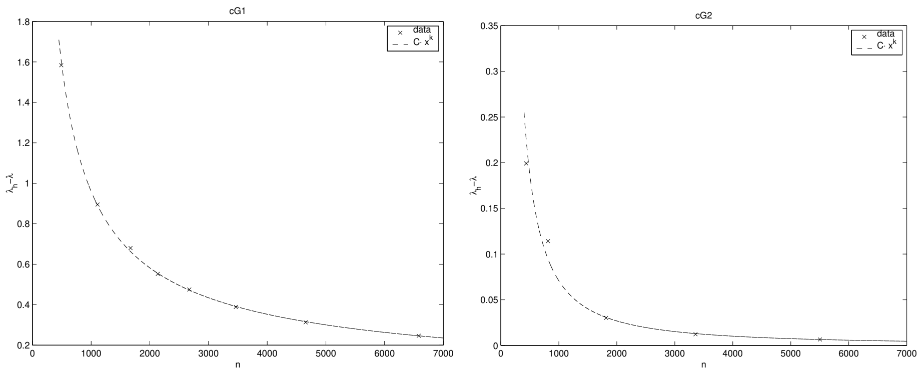

Figure 6: The eigenvalue error is plotted against the number of nodes in the mesh for the cG1 method (left) and cG2 (right).

6.2 The Hydrogen-Platinum problem

In this section we will only use the cG1 solver. This is due to the fact that the implementation of the cG2 code is to slow for this problem. Mainly because the potential needs to be calculated in so many points (gauss points) to keep the fourth order accuracy and also because the administrative part of the matrix assembly will be very slow since it is done in *MATLAB*. For each tetrahedron we have to insert a 10×10 matrix in the A and M matrices. This may look as a simple operation but in fact it takes more cpu time than calculating the entries of the 10×10 matrix. The best way of going around this problem is probably to use a lower level language such as *Fortran* or *C*.

We will start by presenting results where the adaptive algorithm has not been used. Again the mesh generated by *FEMLAB* will not be uniform in the sense that there will not exist a constant mesh parameter h to measure the distances between neighbour nodes. But we will still refer to this mesh as uniform or non-adaptive since it is close to uniform specially compared to the adaptive meshes we will study later, compare for example picture one and three in figure (3). These meshes are often refereed to as being "quasi-uniform".

Table (1) shows the results from a calculation of the two lowest eigenvalues using periodic boundary conditions for different uniform meshes. The quantity physicists are interested in is the difference between the two lowest eigenvalues. We notice quite slow convergence for the same reason as in the test example. The problem is again three dimensional so we have $h \sim (1/n)^{1/3}$. Convergence of the order h^2 then in terms of n means $n^{-2/3}$ which means we need to increase n 30 times to get a new correct digit. To confirm this we will again try to approximate the data by a function of the following form $C_1 + C_2 n^{-2/3}$. The coefficients for λ_1 are $C_1 \approx 0.123$ and $C_2 \approx 2.49$ and for λ_2, $C_1 \approx 0.151$ and $C_2 \approx 2.46$. The result is shown in

n	λ_1	λ_2	$\lambda_2 - \lambda_1$
1361	0.14523	0.17183	0.026603
2472	0.13677	0.16417	0.027397
4346	0.13247	0.16070	0.028227
5642	0.13058	0.15898	0.028394
6686	0.12995	0.15811	0.028161
9651	0.12858	0.15673	0.028151
11584	0.12783	0.15613	0.028305
14520	0.12714	0.15555	0.028409

Table 1: This table shows results from eigenvalue calculations on a non-adaptive mesh with periodic boundary conditions. n refers to the number of nodes used in the computation.

figure (7) (left). Again these numbers are just a quantitative measure of the error and are just presented here to show the rate of the convergence in a practical case. We will refer to these solutions (C_1) as an extrapolation of the approximate solutions.

We will use the values of table (1) as reference values rather than the coefficient C_1 when we evaluate the adaptive results. The same thing is now done with the homogeneous boundary conditions and the results are presented in table (2). Here we got the following values of the coefficients

n	λ_1	λ_2	$\lambda_2 - \lambda_1$
1355	0.14604	0.17279	0.026758
2482	0.13607	0.16484	0.028771
4272	0.13257	0.16085	0.028284
5609	0.13097	0.15926	0.028286
6833	0.13008	0.15819	0.028108
9718	0.12859	0.15686	0.028268
11885	0.12779	0.15619	0.028397
14641	0.12732	0.15563	0.028313

Table 2: This table shows results from eigenvalue calculations on a non-adaptive mesh with homogeneous Dirichlet boundary conditions. n refers to the number of nodes used in the computation.

$C_1 = 0.123$ and $C_2 = 2.49$ for the first eigenvalue and $C_1 = 0.151$ and $C_2 = 2.45$ for the second. The result is plotted in figure (7) to the right.

Next we will see how we can improve the convergence by using the algorithm for eigenvalue adaptivity. Table (3) shows the number of nodes, the first eigenvalue and the estimated eigenvalue error from the a posteriori analysis for the periodic problem.

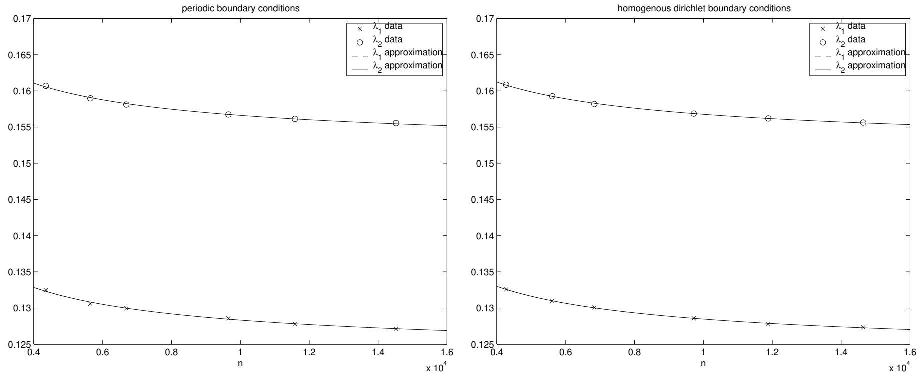

Figure 7: The two lowest eigenvalues are plotted against the number of nodes in the mesh. The lines are the $C_1+C_2n^{-2/3}$ approximations to the two eigenvalues. Periodic boundary condition to the left and Dirichlet conditions to the right.

n	λ_1^h	$E(\lambda_1)$
1361	0.14523	0.02616
1614	0.13630	0.01346
2054	0.13237	0.00877
2731	0.13031	0.00673
2472	0.13677	0.01725
2981	0.13115	0.01117
3593	0.12978	0.00608
4531	0.12859	0.00441
4346	0.13247	0.01222
5191	0.12875	0.00531
6515	0.12746	0.00362
8207	0.12687	0.00300
5642	0.13058	0.01009
6652	0.12805	0.00464
8297	0.12693	0.00318
10606	0.12636	0.00248

Table 3: We start with a uniform mesh of 1361 nodes and then use the adaptive algorithm three times to refine the mesh. Then we start over again from a 2472 node uniform mesh and so on. The third column contains the calculated upper bound of the eigenvalue error.

The final mesh with 2731 nodes gives for example a better approximation to the eigenvalue then we got from the uniform mesh with 5642 nodes, table (1). The second eigenvalue from this calculation was 0.16888 which is close to the result from a uniform mesh with 1361 points which means that the only the first eigenvalue has been improved. It is also interesting to see how the error bound seems to give a lower bound of the exact solution λ_1 to be about 0.123-0.124. Our best approximate eigenvalue from this calculation is $\lambda_1^h = 0.12636$ with $\lambda_1^h - \lambda_1 \leq 0.00248$ for 10606 nodes. The results from the same calculation but with adaptivity for the second eigenvalue with periodic boundary conditions is very similar so we just present the final sequence of the last adaption in table (4). In the homogeneous Dirichlet

n	λ_1^h	$E(\lambda_1)$
5642	0.15898	0.00927
6694	0.15668	0.00490
8339	0.15565	0.00337
10542	0.15511	0.00248

Table 4: We start with a uniform mesh of 5642 nodes and then use the adaptive algorithm three times to refine the mesh. The third column contains the calculated upper bound of the eigenvalue error.

case with adaptivity for the first eigenvalue we get similar results presented in table (5).

All over we can see that we roughly need half the number of nodes to get the same accuracy as we got with the uniform mesh. This together with the fact that we have an upper bound of the eigenvector error makes the adaptive method superior over the uniform and it shows that the a posteriori analysis works in a constructive way. The results of physical interest will be taken from the adaptive calculation starting with a quasi-uniform mesh with 5643 nodes and it will be presented in table (6).

To get a good estimation of the bandwidth of the ground state we would need to solve the problem with much more nodes since the upper bound of the errors are much greater than the difference of the eigenvalues. Another problem is that the meshes are not the same for periodic and Dirichlet conditions. The best approximation in this paper is probably the extrapolation which gives a bandwidth of the order of 0.1 meV. We see from the tables above that the errors of our calculations are at least 2 meV so this value is very uncertain. These values can be compared with the once found by using finite differences in [11] where $\lambda_2^h - \lambda_1^h$ was found to be 29 meV and the bandwidth 0.2 meV.

Results from computations in the exact same manner with deuterium instead of hydrogen is presented in table (7). The eigenvectors will play a more central role in this calculation. This is due to the fact that the "hcp

n	λ_1^h	$E(\lambda)$
1355	0.14604	0.02481
1689	0.13532	0.01179
2110	0.13241	0.00874
2837	0.13006	0.00629
2482	0.13607	0.01628
2974	0.13130	0.00781
3781	0.12900	0.00531
4946	0.12803	0.00409
4272	0.13257	0.01247
5020	0.12917	0.00612
6170	0.12762	0.00408
7715	0.12699	0.00306
5609	0.13097	0.00987
6658	0.12795	0.00445
8272	0.12690	0.00338
10591	0.12641	0.00247

Table 5: We start with a uniform mesh of 1355 nodes and then use the adaptive algorithm three times to refine the mesh. Then we start over again from a 2482 node uniform mesh and so on. The third column contains the calculated upper bound of the eigenvalue error.

	lower bound	upper bound
λ_1^h	0.12388	0.12636
λ_2^h	0.15263	0.15511
$\lambda_2^h - \lambda_1^h$	0.02627	0.03123

Table 6: These results are calculated on adaptive meshes with approximately 10600 nodes using periodic boundary conditions for the hydrogen-platinum problem.

	lower bound	upper bound
λ_1^h	0.08610	0.08920
λ_3^h	0.11723	0.12094
$\lambda_3^h - \lambda_1^h$	0.02803	0.03484

Table 7: These results are calculated on adaptive meshes with approximately 6300 nodes using periodic boundary conditions for the deuterium-platinum problem.

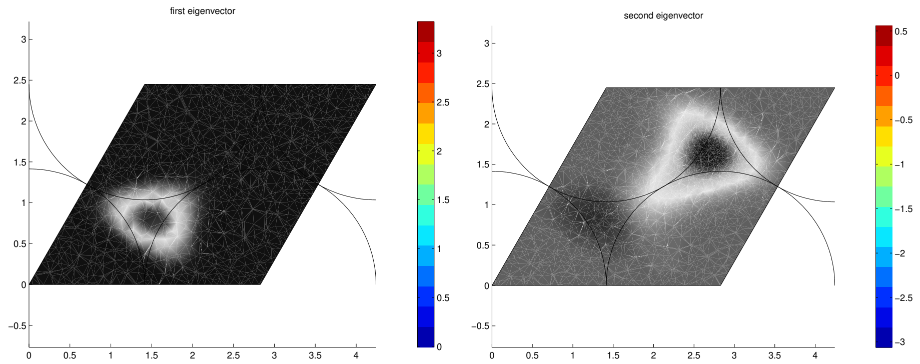

Figure 8: The eigenvector associated with the first eigenvalue to the left and the second to the right using Dirichlet boundary conditions. Slices are shown through the domain where the solution has its biggest value, z=0.35 for the first eigenvector and z=0.39 for the second.

mode" now is represented as the third eigenvalue. The diffusion term in the equation has become smaller which makes it possible for the first mode to go further down the "fcc well" so the former third eigenvalue will now decrease so much that it will be lower than the eigenvalue of the "hcp mode".

Finally we will solve the hydrogen-platinum problem using the eigenvector adaptivity algorithm instead. The physical properties of interest mentioned in this paper are mainly associated with the eigenvalues so in this last part we will focus on how the a posteriori analysis of the eigenvector differs from the analysis of the eigenvalue.

We will first plot the eigenvectors associated with the two lowest eigenvalues in figure (8). We can see how the first eigenvector is localised in the "fcc site" and the second one in the "hcp site". We will now focus on the first eigenvector and study how the eigenvector adaptivity improved the L^2-norm of the error. We will start from a mesh with 5609 nodes.

The residual looks a lot like the solution u_h since the main term is $|(c-\lambda_h)u_h|$. The stability factor behaves a bit different. In figure (9) is the solution to the dual problem plotted. We can see some big variations in the solution but fortunately they are far from the centre of the "fcc-site" where the residual will be small. The L^2-norm errors are presented in table (8). It is very hard to say anything about how sharp these bounds are. It is clear from experiments that the bounds in some cases becomes very bad. Probably because of the absolute value taken in (13). We can never use the fact that the dual solution changes sign when the residual does not which would mean cancellation.

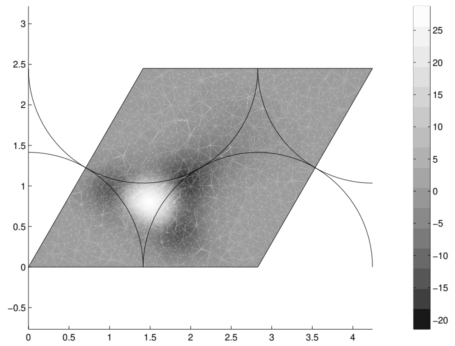

Figure 9: The solution of the dual problem using an adapted mesh of 6539 nodes.

nodes	upper bound of $\|e\|$
5906	0.12385
6593	0.06206

Table 8: These are results for the first eigenvector calculated with Dirichlet boundary conditions.

6.3 Future work

This paper is a starting point for further work in the area of using adaptive finite element methods for solving problems with light particles on a metal surface with high accuracy and error control.

The computational part of this paper has been done mainly to illustrating the theory and not to create a software that can compete with the best in solving the time-independent Schrödinger equation for a particle in a potential. To go one step further we would probably need to change language to a faster one such as *C* or *Fortran* and to think more about implementation aspects such at memory allocation. If this could be done higher order method such as cG2 would be very interesting to study. Since the boundary conditions are periodic and the geometry is fairly simple it would probably also be a good idea to try spectral methods on this problem specially for calculating the bandwidth in the deuterium case which we have not mentioned here. It suppose to be in the order of 10^{-7} eV [11] which of course is far from what we have been able to resolve in this paper. Spectral accuracy might do the trick here.

The software limitations with periodic boundary conditions mentioned before will hopefully disappear in the next versions of *FEMLAB* which will make it possible to study higher eigenmodes in the same manner as the two lowest ones. It would be nice to get a sharper upper bound for the eigenvector error in the L^2-norm when we can study these higher modes. Another thing that could improve the eigenvector error bound is to put more thought into how to choose ψ, in the dual problem and to bound the interpolation error for the potential term.

References

[1] I. Babuška and J.E Osborn, *Finite element-Galerkin approximation of eigenvalues and eigenvectors of self-adjoint problems,* Math. Comp. 52, 1989, 275-297.

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

[3] C. Cuvelier, A. Segal and A.A. van Steenhoven, *Finite element methods and Navier-Stokes equations,* Reidel publ., 1986, 165.

[4] L. Debnath and P. Mikusiński, *An Introduction to Hilbert Spaces with Applications (Second edition),* Academic Press, 1999.

[5] J. W. Demmel, *Applied Numerical Linear Algebra,* SIAM, 1997.

[6] K. Eriksson, D. Estep, P. Hansbo and C. Johnson, *Computational Differential Equations,* Studentlitteratur, 1996.

[7] K. Eriksson, C. Nystedt and L. B. Wahlbin, *Error estimates and adaptive finite element methods for a model problem,* Department of Mathematics, Chalmers University of Technology, Sweden, preprint.

[8] B. Fischer, *Polynomial Based Iteration Methods for Symmetric Linear Systems,* Wiley-Teubner Series: Advances in Numerical Mathematics, 1996.

[9] G. H. Golub and C. F. van Loan, *Matrix computations (third edition),* The Johns Hopkins University Press, 1996.

[10] M. T. Heath, *Scientific Computing: An introductory Survey.* McGraw-Hill, 1997.

[11] G. Källen and G. Wahnström, *Quantum behaviour of H on Pt(111),* Department of Applied Physics, Chalmers University of Technology, Sweden, 2001.

[12] M. G. Larson, *Aposteriori and a priori error analysis for finite element approximations of self-adjoint elliptic eigenvalue problems,* Department of Mathematics, Chalmers University of Technology, Sweden.

[13] B. Lundqvist, *Kvant Fysik del 1,* Department of Theoretical Physics and Mechanics, Chalmers University of Technology, Sweden, 1999.

[14] H. P. Myers, *Introductory Solid State Physics (second edition),* Taylor and Francis, 1997.

[15] E. Süli, *Finite element methods for partial differential equations*, lecture notes, 2000.

[16] L. N. Trefethen, *Spectral Methods in MATLAB*, SIAM, 2000.