College of Graduate Studies
Department of Mathematics

Numerical Solution of the Dirac Equation Using the Petrov - Galerkin Finite Element Method

By:

Lana Ghandy Ahmad Abu-Rayyan

Supervised By:

Dr. Hasan Almanasreh

This thesis is submitted in partial fulfillment of the requirements for the degree of Master of Science in Mathematics, College of Graduate Studies and Academic Research, Hebron University, Palestine.

June, 2018

Numerical Solution of the Dirac Equation Using the Petrov - Galerkin Finite Element Method

By
Lana Ghandy Aburayyan

This thesis was defended successfully on 21/6/2018 and approved by:

Committee Members:		**Signature**
• Dr. Hasan Almanasreh	Supervisor	.....
• Dr. Tareq Amro	Internal Examiner	.....
• Dr. Ali Zein	External Examiner	.....

Dedications

To my parents Ghandi and Rania, my husband Mohammad, brothers,sister and my son Ahmad

Acknowledgements

First of all I am grateful to The Almighty Allah for helping me to complete this thesis, all the Praise and thanks to Allah.

I would like to express my sincere appreciation to my principal Supervisor, Dr. Hasan Almanasreh, for suggesting the problem and for his guidance and support and encouragement during the time working in this thesis, without who this work would not have been possible.

I'm also grateful to all lecturers in the department of mathematics, in particular Dr. Ali Tawayha, for their support towards the successful completion of my Thesis.

Without the financial support and the encourage of my husband, Mohammad, this work would not have been possible. My special thanks to my parents, sisters, brothers, the family of my husband and my friends at Hebron university.

الإقرار

أنا الموقع أدناه مقدم الرسالة التي تحمل العنوان:

الحل العددي لمعادلة ديراك باستخدام طريقة العناصر المحدودة لبتروف قالاركن

NUMARICAL
SOLUTION OF THE DIRAC EQUATION USING THE
PETROV - GALERKIN FINITE ELEMENT METHOD.

أقر بأن ما اشتملت عليه هذه الرسالة إنما هو نتاج جهدي الخاص، باستثناء ما تمت الإشارة إليه حيثما ورد، و أن هذه الرسالة ككل لم تقدم من قبل لنيل أية درجة علمية أو بحث علمي أو بحثي لدى أية مؤسسة تعليمية او بحثية أخرى.

Declaration

The work provided in this thesis, unless otherwise referenced, is the result of the researcher's work, and has not been submitted elsewhere for any other degree or qualification.

Lana Ghandy Abu rayyan

Signature:

Abstract

In this thesis, we study the eigenvalue problem of the Dirac operator focusing on the numerical computation of the eigenvalues using the finite element method (FEM) with linear basis functions. The problem in solving the Dirac eigenvalue problem is the occurrence of non-true values (called spurious eigenvalues) among the genuine ones. Here, we shed the light on the causality of the spurious eigenvalues from different point of views. Also a stable numerical scheme is discussed which provides a cleaned set of eigenvalues.

The Dirac equation is a convection-dominated problem, which means that the numerical methods applied to solve it encounter instability in the solution. The scheme of stability provided in this thesis is based on using the Streamline Upwind Petrov-Galerkin (SUPG) finite element method (FEM) instead of the usual FEM. This method creates balanced diffusion terms controlled by a stability parameter. The created diffusivity enhances the set of solution and guarantees a complete remedy of the spurious eigenvalues.

الملخص

في هذه الرسالة سوف ندرس معادلة ديراك والأخطاء في حلها، وذلك بالتركيز على الحل العددي لها باستخدام طريقة العناصر المحدودة مع اقتران خطي أساسي نعتمد عليه.

ان المشكلة في حل الأخطاء لمعادلة ديرك تكمن في القيم الكاذبة التي تظهر خلال الحل، كما وسيتم نقاش طرق مختلفة لحل هذه المشكلة، أيضا سوف يتم نقاش استقرار الأنظمة العددية التي من خلالها سوف نحصل على قيم نظيفة وليست كاذبه.

إن معادلة ديراك يسيطر عليها عدم الاستقرار في الحل، ولذلك سوف نستخدم طريقة العناصر المحدودة لبتروف قالاركن بدلا من طريقة العناصر المحدودة العادية، مؤديا لإنشاء طرف يتم التحكم به من خلال مقياس للاستقرار وبهذا يتم الوصول إلى الهدف

Contents

List of Figures

List of Tables

CHAPTER 1

Dirac equation

Physicist Jon Butterworth of University College London in the UK said about the Dirac equation: ”I love the Dirac equation because it combines elegant mathematics with huge physical consequences. Paul Dirac was determined to come up with a proper relativistic quantum equation for electrons. He did it, but the consequences were more far-reaching that anyone could have dreamed,”see [1].

Dirac Equation was found by Dual Adrien Maurice Dirac (1902-1982), British theoretical physicist and a founder of the filed of quantum mechanics, he worked as a teacher in math in Cambridge University (1932-1969) and a physics teacher in Florida university (1971-1984) .

In 1928 Dirac proposed the Dirac equation as a relativistic equation of the motion electron as a wave function.
Dirac used to find a partial differential equation with a positive probability density, which describes the behavior of electrons that have the velocity with closed speed of light.

So, we will have two solutions one returns to the electron and the other solution would be for a particle very similar to the electron except that it differs from the electric charge signal.
This discover that each particle has an anti-particle,see [1].

1.1 Relativistic free particle in one dimension

The free Dirac equation describes a relativistic electron or positron which moves freely, as if there were no external fields or other particles which are a relation between energy and momentum, that is:

$$E^2 = p^2c^2 + m^2c^4. \tag{1.1}$$

It is the classical relativistic energy -momentum equation, where, see [1],
E is the total electron energy,

p is the electron kinetic momentum,

c is the speed of light,

m is the electron rest mass.

The energy of a particle with momentum p can be written as

$$E = \alpha cp + \beta mc^2, \tag{1.2}$$

where α and β are square matrices with constant coefficients and $\alpha\ \beta \neq \beta\ \alpha$.
By squaring both sides, we have

$$E^2{=}\alpha^2c^2p^2 + \alpha\beta pmc^3 + \beta\alpha mc^3 + \beta^2m^2c^4. \tag{1.3}$$

Compare this result with the classical relation (1.1), we have the right hand sides are

equal if and only if

$$\alpha\beta + \beta\alpha = 0, \quad \text{and} \quad \alpha^2 = \beta^2 = I. \tag{1.4}$$

From the classical equation of motion (1.1) we obtain the wave equation of quantum mechanics when replacing the energy E and the momentum p by their quanta

$$E = i\hbar(\partial/\partial t), \qquad \text{and} \qquad p = -i\hbar\nabla, \tag{1.5}$$

where t denotes the time, $\nabla = (\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3})$, and $\hbar$ is the Planck; constant divided by 2π, where the resulting constant is called the reduced Planck constant or Dirac constant.

And is denoted by $\hbar$, i.e. $\hbar = \frac{h}{2\pi}$, where $h = 4.135667662(25)10^{-15}ev.s$ (electron volt.second, (it is the amount of energy gained (or lost) by the charge of a single electron moving across an electric potential difference of one volt). Thus $\hbar = 6.582119514(40)(10^{-16})ev.s)$.

By using (1.5) in (1.1), we will have the square-root Klein-Gordon equation

$$i\hbar\frac{\partial}{\partial t}\psi(t,x) = \sqrt{-c^2\hbar^2\triangle + m^2c^4}\psi(t,x), \tag{1.6}$$

where, $x \in \Re^3$, $\triangle = \frac{\partial^2}{\partial x_1^2} + \frac{\partial^2}{\partial x_2^2} + \frac{\partial^2}{\partial x_3^2}$ is the Laplace operator and ψ is the wave function. In this equation, Dirac found a problem include external electromagnetic fields in a relativistically invariant way, so he looked for another equation to solve the laplace operator under the square root, see [1].

The electron motion consistent with both the principles of quantum mechanics and the theory of special relativity.

So, the free Dirac space-time equation has the form

$$i\hbar\frac{\partial}{\partial t}\psi(t,x) = H_0\psi(t,x), \tag{1.7}$$

where $H_0 : H^1(\Re^3;\mathbb{C}^4) \to L^2(\Re^3;\mathbb{C}^4)$ is the free Dirac operator given by:

$$H_0 = c\alpha p + \beta mc^2. \tag{1.8}$$

By substituting $p \to -i\hbar\frac{\partial}{\partial x} \to -i\frac{\partial}{\partial x}$ ($\hbar = 1$), leads to the free particle Dirac operator

of quantum mechanics

$$H_0 = -ic\alpha.\nabla + \beta mc^2 = \begin{pmatrix} mc^2 I & -ic\sigma.\nabla \\ ic\sigma.\nabla & -mc^2 I \end{pmatrix}, \tag{1.9}$$

where

$$\alpha_j = \begin{pmatrix} 0 & \sigma_j \\ \sigma_j & 0 \end{pmatrix},$$

and

$$\beta = \begin{pmatrix} I & 0 \\ 0 & -I \end{pmatrix}.$$

Here 0, I denote the zero and unity matrices respectively, and σ_j, $j = 1, 2, 3$, given by a set of three 2×2 complex matrices:

$$\sigma_1 = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix},$$

$$\sigma_2 = \begin{pmatrix} 0 & -i \\ i & 0 \end{pmatrix},$$

and

$$\sigma_3 = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}.$$

H_0 is 4×4 matrix differential operator which acts on $\mathbb{C}^4$-valued functions of $x \in \Re^3$.

Define the Hilbert space
$D = L^2(\Re^3) \oplus L^2(\Re^3) \oplus L^2(\Re^3) \oplus L^2(\Re^3) = (L^2(\Re^3))^4 = L^2(\Re^3, \mathbb{C}^4)$, it consists of 4-component column vector $\psi = (\psi_1, \psi_2, \psi_3, \psi_4)^T$, where each component ψ_i is a complex valued function of the space variable x.

The scaler product defined on D is:

$$< \psi, \Phi > \equiv \int_{\Re^3} \sum_{i=1}^{4} \overline{\psi_i(x)} \Phi_i(x) d^3(x). \tag{1.10}$$

In this Hilbert space we want to define the free Dirac operator

$$H_0\psi \equiv -ic\alpha.\nabla\psi + \beta mc^2\psi,$$

$\forall\, \psi \in D(H_0)$.
When we separate the variables in (1.8), we have the free Dirac eigenvalue problem

$$H_0\psi(x) = E\psi(x). \tag{1.11}$$

Which can be written in the form

$$\begin{pmatrix} mc^2 I & -c.\nabla \\ c.\nabla & -mc^2 I \end{pmatrix} \begin{pmatrix} f(x) \\ g(x) \end{pmatrix} = E \begin{pmatrix} f(x) \\ g(x) \end{pmatrix}, \tag{1.12}$$

where f and g are the eigenfunctions and, E is the eigenvalue of the free particle Dirac operator, see [2].

1.2 Coulomb potential

When the Hamiltonian is of the particular form $H = H_0 + V(x)$, H_0 is the free particle Hamiltonian and $V(x)$ is a potential, then the physical system is spherically symmetric, whenever the potential V is spherically symmetric .

In our work we deal with Coulomb potential, which is about the electrostatic energy of an electron in the field of an atomic nucleus.

We will use a hydrogenic atom, which called a one-electron atom or ion (Hydrogen-like ions) such as H, He^{+},Li^{+2},...etc.
It consists of an atomic nucleus contain Z protons such that $Z \in \{1, 2, 3....137\}$ is the electric charge number, and an electron that interacts with the nucleus by the electrostatic Coulomb attraction.

Now the function V will be considered as the Coulomb potential which has the form $V(x) = \frac{-Z}{|x|} I$, here I is 4×4 identity matrix. The spectrum of the Dirac operator with Coulomb potential is $(-\infty, -mc^2) \cup {E^k}_k \cup (mc^2, +\infty)$, where ${E^k}_k \in \mathbb{N}$ is a discrete sequence of eigenvalues, see [4].

So, the magnitude of the force acting on an electron at a distance x from the nucleus is given by $F \propto \frac{Z}{x^2}$ where Z is the electric charge number. Moreover if we take the hydrogenic atom this leads to the nucleus which is a point mass, see [2].

If the nuclear charge Z is not too large, this assumption a cure with high accuracy, that by neglecting the dimension of the nucleus, so the Coulomb force on the electron is the negative derivative of a potential energy V, $F(x) = -\nabla V(x)$, then for the potential energy of a proton of electron is $V(x) = -\frac{Z}{|x|}$, so we will write the new Dirac operator as

$$H = -ic\alpha.\nabla + \beta mc^2 + V(x)I \tag{1.13}$$

which is called the Coulomb-Dirac operator, see [4].

1.3 The Radial Dirac Operator

The Dirac operator H is reduced by the subspaces of definite angular momentum, and definite parity on such a subspace it is unitary equivalent to 2×2 matrix operator H_k on $L^2(\Re_+, \mathbb{C}^2)$. The three dimensional Dirac operator with potential $V : \Re^3 \to \Re$, is given as

$$H = -i\alpha.\nabla + \beta + V \quad \text{on} \quad \bar{d} = L^2(\Re^3, \mathbb{C}^4), \tag{1.14}$$

where $\alpha.\nabla = \sum_{i=1}^{3} \alpha_i \frac{\partial}{\partial x_i}$ and in the standard representation

$$\alpha_i = \begin{pmatrix} 0 & \sigma_i \\ \sigma_i & 0 \end{pmatrix},$$

and

$$\beta = \begin{pmatrix} 1 & 0 \\ 0 & -1 \end{pmatrix}, \tag{1.15}$$

where the Pauli matrices σ_i are defined before.

In our units $\hbar = m = c = 1$, assume V is bounded and measurable, the Hamiltonian H is thus self -adjoint on $H^1(\Re^3, \mathbb{C}^4)$.

For simplicity we should use radial coulomb Dirac operator (eigenvalue problem with Coulomb potential) which is obtained by separation of variables of the radial and angular part, that to say, let

$$\psi(x) = \frac{1}{r} \begin{pmatrix} f(r) Z_{\kappa,m}(\Omega, \theta) \\ ig(r) Z_{-\kappa,m}(\Omega, \theta) \end{pmatrix}, \tag{1.16}$$

where r represents the radial variable, f and g are the Dirac radial functions referred to as the large and small components respectively, and $Z_{.,m}$ is the angular part of the wave function ψ, see [8].
So, the radial Dirac eigenvalue problem $H_\kappa \Phi(r) = E\Phi(r)$, where $\Phi(r) = (f(r), g(r))^T$ is the radial wave function, E is the electron relativistic energies (eigenvalues), and H_κ is the radial Dirac operator given by

$$H_\kappa = \begin{pmatrix} mc^2 + V(r) & c(\frac{-d}{dr} + \frac{\kappa}{r}) \\ c(\frac{d}{dr} + \frac{\kappa}{r}) & -mc^2 + V(r) \end{pmatrix}$$

and

$$\Phi(r) = \begin{pmatrix} f(r) \\ g(r) \end{pmatrix}.$$

The constant c is the speed of light, m is the electron mass, V is the coulomb potential, and κ is the spin-orbit coupling parameter defined as $\kappa = (-1)^{j+\ell+1/2}(j+1/2)$, where j and ℓ are the total and the orbital angular momentum quantum number respectively, see [8].

So, the radial Coulomb - Dirac eigenvalue problem is given as

$$\begin{pmatrix} mc^2 + V(r) & c(\frac{-d}{dr} + \frac{\kappa}{r}) \\ c(\frac{d}{dr} + \frac{\kappa}{r}) & -mc^2 + V(r) \end{pmatrix} \begin{pmatrix} f(r) \\ g(r) \end{pmatrix} = E \begin{pmatrix} f(r) \\ g(r) \end{pmatrix} \tag{1.17}$$

see, [2].

CHAPTER 2

Self-adjoint

2.1 The adjoint operator

Definition 2.1
If $L : V \to W$ is a closed bounded linear operator every where in a Hilbert space V, then the adjoint of L is defined to be the unique closed linear operator L^* acting on W defined as $< u, Lv >=< L^*u, v >$, $\forall u, v \in V$.

Definition 2.2
Inner Product "scaler product"
let $\mathbb{C}$ be the field of complex numbers, let F be a subfield of $\mathbb{C}$, and let V be a vector space over F, see [2]. An inner product is a mapping $< ., . >: V \times V \to F$ that satisfies the following properties:

1- Conjugate symmetry: $\forall x, y \in V, < x, y >= \overline{< y, x >}$.

2- Linearity $\forall x, y \in V, \forall a \in F, < ax + y, z >= a < x, z > + < y, z >$.

3- Non-Negative definiteness $\forall x \in V, < x, x > \geq 0$.

4- Positiveness $\forall x \in V, < x, x >= 0 \Rightarrow x = 0$.

Note that the adjoint of a matrix is its conjugate transpose.

Theorem 2.1.
The adjoint operator T^* of a bounded operator T is bounded, moreover we have $||T|| = ||T^*||$.

Proof.
By Cauchy Schwarz's inequality for any $u, v \in H$, we have
$| < T^*u, v > | = | < u, Tv > | \leq ||u|| \ ||Tv|| \leq ||T|| \ ||u|| \ ||v||$.
Hence, for $v = T^*u$ we obtain

$$||T^*u||^2 =< T^*u, T^*u > \leq ||T|| \ ||u|| \ ||T^*u||,$$

consequently, $||T^*|| \leq ||T||$,
this proves that the adjoint of a bounded operator is bounded. Therefor inequality can be used for T^* instead of T which gives $||T|| = ||(T^*)^*|| \leq ||T^*||$, thus $||T|| = ||T^*||$, see [2].

Adjoint of a differential operator
Let T be a linear operator in $L^2([a, b])$, an operator T^* is called the adjoint of T if $< Tu, v >=< u, T^*v >$, $\forall u \in D(T)$ and $\forall v \in D(T^*)$.

A Hermitian matrices are named after Charles Hermites demonstrated in 1855, these matrices share a property with real symmetric matrices of always having real eigenvalues.

A Hermitian matrix or (self - adjoint matrix) is a complex square matrix that is equal to its own conjugate transpose, that is the element in the $i - th$ row and $j - th$ column is equal to the complex conjugate of the element in the $j - th$ row and $i - th$ column for all i and j, that is $a_{ij} = \overline{a_{ij}}$ or $A = \overline{A^T}$.

So, Hermitian operators are represented by matrices that are equal to their own adjoint.

Definition 2.3.
An operator T is said to be self-adjoint if T* $= T$.

Definition 2.4.
An operator is called essentially self-adjoint, if it has a unique extension to a large domain, where it is self-adjoint.

Theorem 2.2.
The free Dirac operator is essentially self adjoint on the dense domain$C_0^\infty(\Re^3)^4$ and self adjoint on the Sobolave space $D(H_0) = H^1(\Re^3)^4$, its spectrum is purely absolutely continuous and given by $\sigma(H_0) = (-\infty, -mc^2) \bigcup (mc^2, \infty)$.
Proof: see [1].

2.2 H_0 with additional field or the Dirac operator in an external field

The free Dirac operator with external filed V is $H = H_0 + V$ such that $V(x) = V_{ij}(x)$, $i, j = 1, 2, 3, 4$ is a Hermitian 4×4 matrix valued function, acts as a multiplication operator in $C_0^\infty(\Re^3.\mathbb{C}^4)$. The following theorem provides a condition on $V(x)$ to guarantee the self-adjointness of H.

Theorem 2.3. (Kato-Rellich): If B is a bounded with A bound smaller than 1, then $A+B$ is self-adjoint on $D(A)$, and essentially self-adjoint on any core of A. Moreover, if A is bounded below, then so is $A + B$, see [1].

Theorem 2.4.
Let V be an external potential added to the free Dirac operator H_0, with a Hermiticity assumption such that each component V_{ij} is a function satisfying the following estimate $|V_{ij}(x)| \leq a\frac{c}{2|x|} + b$, $\forall x \in \Re^3/\{0\}$, $i, j = 1, 2, 3, 4$ where $a < 1$ and $b > 0$, then the operator $H = H_0 + V$ is essentially self-adjoint on $C_0^\infty(\Re^3/\{0\}, \mathbb{C}^4)$ and self-adjoint on the Sobolev space $D(H_0) = H^1(\Re^3.\mathbb{C}^4)$, see [1].

Proof.
From Hardy's inequality

$(r = |x|)$ $\int_{\Re^3} \frac{1}{4r^2}|\psi(x)|^2 d^3(x) \leq \int_{\Re^3} |\nabla\psi(x)|^2 d^3(x)$,

$\psi \in C_0^\infty(\Re^3)$.

We conclude the same inequality for $\psi \in C_0^\infty(\Re^3)^4$ by adding the results for the components. Next we note that

$$| - ic\alpha\nabla\psi(x)|^2 = c^2|\nabla\psi(x)|^2 \equiv c^2 \sum_{i=1}^{k=4} |\partial\psi_i(x)|dx_k,$$

and find that the multiplication operator $c/2r$ is bounded relative to $ic\alpha.\nabla$. From $|V_{ij}(x)| \leq \frac{ac}{2|x|} + b$,we obtain for all $\psi \in C_0^\infty(\Re^3)^4$

$$||V\psi|| \leq a|| - ic\alpha\nabla\psi|| + b||\psi|| \leq a||H_0\psi|| + (b + amc^2)||\psi||.$$

Since the possible singularity of V_{ik} at the origin is square integrable we have $D(V) \supset D(H_0) = H^1(\Re^3)^4$, The operator V is symmetric since the matrix $(V_{ik}(x))$ is Hermitian for all x. (Essential) self-adjointness of H finally follows from the (essential) self-adjointness of H_0 by the Kato-Rellich theorem, because $a < 1$.□

Theorem 2.5.
Assume that each element of the Hermitian potential matrixV is an infinitely differentiable function,$\mathrm{V}_{ik} \in C_0^\infty(\Re^3)$ for all $i, k = 1, 2, 3, 4$. Then $H = H_0 + V$ is essentially self-adjoint on $C_0^\infty(\Re^3)^4$.
Proof: See [1].

2.3 The Spectrum

We firstly talk about eigenvalues and eigenvectors. A real number λ is called an eigenvalue of H which is a self-adjoint operator on a Hilbert space V, if there is a non-zero vector u such that $Hu = \lambda u$.
Every nonzero vector u is satisfying $Hu = \lambda u$ is called an eigenvector of H corresponding to the eigenvalue λ, see [9].

Theorem 2.6.
All eigenvalues of a self-adjoint operator on a Hilbert space are real.

Proof:
Let λ be an eigenvalue of H, let $v \in H$ be an eigenvector for λ, i.e. $Hv = \lambda v$, then
$\lambda < v, v >=< \lambda v, v >$

$=< Hv, v >$

$=< v, Hv >$

$= \overline{< Hv, v >}$

$= \overline{< \lambda v, v >}$

$= \overline{\lambda} < v, v >,$

since v is being an eigenvector so, $v \neq 0$, this implies $< v, v > \neq 0$, then by dividing $< v, v >$, so we obtain $\lambda = \overline{\lambda}$, hence λ is real.□

Theorem 2.7.
Orthogonality
Eigenvectors corresponding to distinct eigenvalues of a self adjoint operator on a Hilbert space are orthogonal.

Proof:
Let H be a self adjoint operator, let u_1 and u_2 be eigenvectors corresponding to distinct eigenvalues λ_1 and λ_2, i.e. $Hu_1 = \lambda_1 u_1$ and $Hu_2 = \lambda_2 u_2$, $\lambda_1 \neq \lambda_2$.
By the last theorem λ_1 and λ_2 are real, then
$\lambda_1 < u_1, u_2 >=< \lambda_1 u_1, u_2 >=< Hu_1, u_2 >=< u_1, Hu_2 >=< u_1, \lambda_2 u_2 >=$
$\lambda_2 < u_1, u_2 >= \lambda_2 < u_1, u_2 >$
and hence $(\lambda_2 - \lambda_1) < u_1, u_2 >= 0$,
since $\lambda_1 \neq \lambda_2$,we have $< u_1, u_2 >= 0$, i.e. u_1 and u_2 are orthogonal, see [2].□

CHAPTER 3

The finite element method in one dimension

The finite element method is a function or tool used to solve partial differential equations (PDEs), static and dynamic problems - solid mechanics, fluid mechanics, electromagnetic, biomechanics, etc. It is based on a continuous piecewise linear approximation.

Definition 3.1.
The Galarkin method : is the method which used to rewrite the differential equation in a variational form, and then discretize the system.
So when we talk about Galarkin method which utilises piecewise polynomial as approximation functions, we mean the Finite Element Method, see [6].

The finite element method uses a special discretization to arrive at a system of matrix equation, and the result of the matrix equations leads by approximate solution to the original boundary value problem, see [15].

Let us take a mathematical example of a stationary reaction diffusion

$$\begin{cases} -\psi'' + \psi' + \psi = f, x \in [a, b] \\ \psi(a) = \psi(b) = 0 \end{cases} \tag{3.1}$$

Firstly we will derive the variational formulation of (3.1), so multiply the differential equation by a test functions $v(x)$, and integrate it over $[a, b]$

$-\int_a^b \psi'' v dx + \int_a^b \psi' v dx + \int_a^b \psi v dx = \int_a^b f v(x) dx.$

Now, take the first term to integrate it by parts

$$-[\int_a^b \psi'' v dx] = -[\psi'(b)v(b) - \psi'(a)v(a) - \int_a^b \psi' v' dx],$$

by using conditions,

$$\psi'(b)v(b) = \psi'(a)v(a) = 0,$$

so,

$$-\int_a^b \psi'' v dx = \int_a^b \psi' v' dx.$$

The final problem is :

$$\int_a^b \psi' v' dx + \int_a^b \psi' v dx + \int_a^b \psi v dx = \int_a^b f v dx.$$

Then make discretization by introducing V_h , which is a finite vector space of continuous linear function on the partition, $a = x_1 < x_2 < ... < x_n = b$ of $[a, b]$.
Now let $\Phi(x) \in V_h$ be an approximate of ψ, thus

$$\int_a^b \Phi' v' dx + \int_a^b \Phi' v dx + \int_a^b \Phi v dx = \int_a^b f v dx, \forall v \in V_h. \tag{3.2}$$

Secondly, the subspace V_h can be spanned by the following basis function $\varphi_i \subset V_h$, called hat functions.

$$\varphi_i(x) = \begin{cases} \frac{x - x_{i-1}}{h_i}, & x_{i-1} < x < x_i \\ \frac{x_{i+1} - x}{h_{i+1}}, & x_i < x < x_{i+1}, \\ 0, & \text{otherwise.} \end{cases} \tag{3.3}$$

Note that

$$\varphi_i(x_j) = \begin{cases} 1, & i = j, \\ 0, & \text{otherwise.} \end{cases}$$

Since $\varphi \in V_h$, then it can be written as $\Phi(x) = \sum_{j=1}^{n} \xi_j \varphi_j(x)$.
Note that $\xi_j = \psi(x_j)$, the nodal values of ψ at x_j.

Let $\Phi(x) = \sum_{j=1}^{n} \xi_j \varphi_j(x)$ in (3.2) and $v = \varphi_i$, $i = 1, 2, 3, ..., n$ to get
$\xi_j \int_a^b \varphi_j{}' \varphi_i{}' dx + \xi_j \int_a^b \varphi_j{}' \varphi_i dx + \xi_j \int_a^b \varphi_j \varphi_i dx = \xi_j \int_a^b f \varphi_i dx$.
The term $\xi_j \int_a^b \varphi_j{}' \varphi_i{}' dx$ global stiffness matrix which is for uniform mesh size h is given by:

$$\frac{1}{h} \begin{pmatrix} 2 & -1 & 0 & . & . & . & . & 0 & 0 & 0 \\ -1 & 2 & -1 & 0 & . & . & . & 0 & 0 & 0 \\ 0 & -1 & 2 & -1 & 0 & . & . & 0 & 0 & 0 \\ . & 0 & -1 & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & -1 & 0 & . & . \\ . & . & 0 & 0 & 0 & -1 & 2 & -1 & 0 & 0 \\ 0 & . & . & . & 0 & 0 & -1 & 2 & -1 & 0 \\ 0 & . & . & . & . & 0 & 0 & -1 & 2 & -1 \\ 0 & 0 & . & . & . & . & 0 & 0 & -1 & 2 \end{pmatrix}, \tag{3.4}$$

The term $\xi_j \int_a^b \varphi_j{}' \varphi_i dx$, is called the global convection matrix which is for uniform mesh size h is given by:

$$\frac{1}{2} \begin{pmatrix} 0 & 1 & 0 & . & . & . & . & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & . & . & . & 0 & 0 & 0 \\ 0 & -1 & 0 & 1 & . & . & . & 0 & 0 & 0 \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & 0 & -1 & 0 & 1 & 0 & 0 \\ 0 & . & . & . & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & . & . & . & 0 & 0 & -1 & 0 & 1 & 0 \\ 0 & 0 & . & . & . & . & 0 & 0 & -1 & 0 \end{pmatrix}, \tag{3.5}$$

Finally the term $\xi_j \int_a^b \varphi_j \varphi_i dx$ is called the global mass matrix and for uniform mesh size h is given by:

$$\frac{h}{6}\begin{pmatrix} 4 & 1 & 0 & . & . & . & . & 0 & 0 & 0 \\ 1 & 4 & 1 & 0 & . & . & . & 0 & 0 & 0 \\ 0 & 1 & 4 & 1 & . & . & . & 0 & 0 & 0 \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ . & . & . & . & . & . & . & . & . & . \\ . & . & 0 & . & . & . & . & 1 & 0 & 0 \\ 0 & . & . & . & . & 0 & 1 & 4 & 1 & 0 \\ 0 & . & . & . & . & 0 & 0 & 1 & 4 & 1 \\ 0 & 0 & . & . & . & . & 0 & 0 & 1 & 4 \end{pmatrix} \tag{3.6}$$

Note that the term $\int_a^b \varphi_j{}' \varphi_i{}' dx$ is called diffusion term, it is stable, because it is symmetric. While the term $\int \varphi_j{}' \varphi_i dx$ is called convection term and it is not stable, because it is not symmetric.

3.1 Convection dominated problem

Consider the problem:

$$\begin{aligned} -(k)\psi'' + u\psi' + s\psi &= f, \quad x \in [0,1] \\ \psi(0) &= \psi(1) = 0, \end{aligned} \tag{3.7}$$

where k, u and s are functions.
Firstly, we will talk about the following dimensionless numbers, which characterize the solution of the problem,

$$Pe_j = \frac{|u_j| hj}{2k},$$
$$Da_j = \frac{s_j h_j}{|u_j|},$$

where Pe_j is the element Peclet and Da_j is element Damköhler numbers.

$\diamondsuit$ h_j is the size of the element interval $I_j = [x_{j-1}, x_j]$.
$\diamondsuit$ u_j is the coefficient of the convection .
$\diamondsuit$ s_j is the reaction term.
$\diamondsuit$ k is the diffusion size .

So, when the convection coefficient is larger than the diffusion coefficient, i.e. when $Pe > 1$ or if

$$2Pe_j Da_j = 2\frac{|u_j h_j|}{2k}\frac{s_j h_j}{|u_j|} = s_j h_j^2/k > 1,$$

then the equation is convection dominated one, see [6, 20].
Numerical methods, including the finite element methods are not stable when they are applied to convection dominated problem. This un stability appears as spurious solutions or oscillations in the approximation solution.

In the following two sections we present two techniques of the finite element method to stabilize the numerical solution.

3.2 Two methods of stabilization

We now present two methods to stabilize the solutions:

1- The stream line upwind Petrov - Galerkin method (SUPG), which simply is assuming the test function as $v + \tau v'$ where τ is a stability parameter.

2- Adding artificial diffusion term $< (\tau)\psi', v' >$.
The **SUPG** method:-
This method will introduce artificial diffusion in streamline direction, and it is determined with the stabilization parameter τ, which depends on the position of some neighboring nodes inside the domain, and it is a small support stabilization parameter, see [14, 20].

Remark: for simplicity we will use the notation $< u, v >= \int_\Omega u(x)v(x)dx$.

Consider the finite element method to (3.7) by choosing the test function to be $v + \tau v'$ where $v(0) = v(1) = 0$
$< -k\psi'', v + \tau v' > + < u\psi', v + \tau v' > + < s\psi, v + \tau v' >$

$=< f, v + \tau v' >$

$-k < \psi'', v > -k < \psi'', \tau v' > +u < \psi', v > +u < \psi', \tau v' > +s < \psi, v >$

$+ s < \psi, \tau v' >=< f, v > + < f, \tau v' >.$

Note that

$-k < \psi'', v >= -k \int_0^1 \psi'' v dx$

$= -k(\psi(1)v'(1) - \psi(0)v'(0)) + k \int_0^1 \psi' v' dx = k \int_0^1 \psi' v' dx.$

Hence, we have the form

$k < \psi', v' > -k < \psi'', \tau v' > +u < \psi', v > +u < \psi', \tau v' > +s < \psi, v > +$

$s < \psi, \tau v' > =< f, v > + < f, \tau v' >$

$$k < \psi', v' > +u < \psi', v > +s < \psi, v > +\tau \big[-k < \psi'', v' > +u < \psi', v' >$$

$+ s < \psi, v' > \big] =< f, v > +\tau < f, v' > .$

Let

$$\psi = \sum_{j=1}^{n} \xi_j \varphi_j,$$

where:

$$\varphi_j = \begin{cases} \frac{x - x_{j-1}}{h_j}, & x_{j-1} < x < x_j, \\ \frac{x_{j+1} - x}{h_{j+1}}, & x_j < x < x_{j+1}, \\ 0, & \text{otherwise}, \end{cases}$$

where $h_j = x_j - x_{j-1}$, also let $v = \varphi_i$, $i = 1, 2, 3..., n$,

$k\xi_j < \varphi_j', \varphi_i' > +u\xi_j < \varphi_j', \varphi_i > +s\xi_j < \varphi_j, \varphi_i >$

$+ \tau[-k\xi_j < \varphi'', \varphi_i' > +u\xi_j < \varphi_j', \varphi_i' > +s\xi_j < \varphi_j, \varphi_i' >]$

$=< f, \varphi_i > +\tau < f, \varphi_i' > .$

Since a linear basis functions are assumed, then the term $< \varphi_j'', \varphi_i' >= 0.$

Adding artificial diffusion term.

In second method we will add artificial diffusivity to the convection dominated problems to stabilize the numerical solution, see [16].
After adding the artificial diffusion term $< \psi', \tau v' >$ to the variational formulation of (3.7) we have:

$< -k\psi'', v > + < u\psi', v > + < s\psi, v > + < \tau\psi'', v >=< f, v > .$

Integrating by parts yields

$$< -k\psi', v' > + < u\psi', v > + < s\psi, v > + < \psi', \tau v' > \ =< f, v > .$$

CHAPTER 4

Spurious eigenvalues in the computation of the Dirac operators

In this chapter we will discuss the computation of the electron energies of the system with one electron, known as Hydrogen -like atoms ions.

So, we will talk about eigenvalues which are neither related to mathematical interpretation, nor to physical explanation called spurious eigenvalues [17]. A real number λ is called a spurious eigenvalue that appears in the computation, if λ does not corresponding to any exact solution.

In general the computation of the eigenvalues of the radial Dirac operator are polluted by two different categories of spurious eigenvalues; the first one is the instilled spuriosity and the second is the nonphysical coincidence phenomenon. For example, computation of the electron eigenvalues in the Hydrogen atom, [18], using finite element method with linear basis functions (hat function), by the first category we mean the non - true values that appear between the energy levels within the true eigenvalues, which occur for all values of the quantum number κ.

The second type is the nonphysical assigning of the same entire set of eigenvalues for $\{2s_{1/2}(\kappa = -1), 2p_{1/2}(\kappa = 1)\}$, $\{ 3p_{3/2}(\kappa = -2), 3d_{3/2}(\kappa = 2)\}$, $\{4d_{5/2}(\kappa = -3), 4f_{5/2}(\kappa = 3)\}$ and so on, [19].

The exact eigenvalues can be obtained by using the relativistic formula for point nucleus.
$\lambda_{nr,\kappa} = \frac{mc^2}{\sqrt{1+\frac{z^2\alpha^2}{(n_r-1+\sqrt{\kappa^2-z^2\alpha^2})^2}}}$, see [3].

where α is the fine structure constant which has in atomic unit, the value $1/c$ and the orbital level number n_r takes the values $1, 2, 3...$, see [3],

Table 4.1: *The first computed eigenvalues of the electron in the Hydrogen-like Cesium ion using the FEM with linear basis functions for point nucleus.*

$\kappa = -1$		$\kappa = 1$	
Approximation	Exact	Approximation	Exact
-1578.87320060	-1578.87360265	-1578.87320060	SE
-398.9566872	-398.956306798	-398.956687265	-398.956306798
-175.3992611	-175.398291704	-175.399261141	-175.398291704
-98.3146870774	SE	-98.3146870774	SE
-97.8651048302	-97.8637523276	-97.8651048301	-97.863752327
-62.2787931909	-62.276000655	-62.2787931910	-62.276000655
-43.0720344726	-43.0679643349	-43.0720344725	-43.0679643349
-32.0173363865	SE	-32.0173363865	SE
-31.5483506089	-31.5427684300	-31.5483506089	-31.5427684300
-24.0984432587	-24.0910611453	-24.0984432587	-24.0910611453
-19.007241800	-18.9978018903	-19.0072418005	-18.9978018903
-15.8499059510	SE	-15.8499059510	SE

4.1 What about spuriosity in the Dirac eigenvalue problem

Definition 4.1.
The projection method: is an active device to compute the eigenvalue problems of a gap on the essential spectrum.

Now we discuss the nonphysical coincidence phenomenon, consider the radial Dirac eigenvalue problem

$$H_\kappa \varphi(r) = \lambda \varphi(r).$$

Shifting the radial Dirac operator by $-mc^2$ and assuming $m = 1$, it becomes :

$$H_k = \begin{pmatrix} V(r) & c(\frac{-d}{dr} + \frac{\kappa}{r}) \\ c(\frac{d}{dr} + \frac{\kappa}{r}) & -2c^2 + V(r) \end{pmatrix}, \tag{4.1}$$

then the eigenvalues are shifted, let

$$\ddot{U}_\kappa = -\frac{z_\kappa}{c|\kappa|(|\kappa| + \xi)}$$

with

$$\xi = \sqrt{\kappa^2 - \frac{z^2}{c^2}}.$$

Define the transformation

$$U_\kappa = \begin{pmatrix} 1 & \ddot{U}_\kappa \\ \ddot{U}_\kappa & 1 \end{pmatrix}, \tag{4.2}$$

and apply it to the radial function

$$\Phi(r) = \begin{pmatrix} f(r) \\ g(r) \end{pmatrix},$$

to get

$$\tilde{\varphi_\kappa}(r) \;=\; U_\kappa \begin{pmatrix} f(r) \\ g(r) \end{pmatrix} = \begin{pmatrix} f(r) + U_\kappa g(r) \\ g(r) + U_\kappa f(r) \end{pmatrix} = \begin{pmatrix} \tilde{f}_\kappa(r) \\ \tilde{g}_\kappa(r) \end{pmatrix}. \tag{4.3}$$

So, we obtain

$U_\kappa^{-1} H_\kappa U_\kappa^{-1} \tilde{\varphi_\kappa}(r) = \lambda_\kappa U_\kappa^{-2} \tilde{\varphi_\kappa}(r)$.

Next add the term $c^2(1 - \frac{|\kappa|}{\xi}) U_\kappa^{-2} \tilde{\varphi_\kappa}(r)$ to both sides to get

$$c^2(1 - \frac{|\kappa|}{\xi}) U_\kappa^{-2} \tilde{\varphi_\kappa}(r) + U_\kappa^{-1} H_\kappa U_\kappa^{-1} \tilde{\varphi_\kappa}(r) = \lambda_\kappa U_\kappa^{-2} \tilde{\varphi_\kappa}(r) + c^2(1 - \frac{|\kappa|}{\xi}) U_\kappa^{-2} \tilde{\varphi_\kappa}(r)$$

$$[c^2(1 - \frac{|\kappa|}{\xi}) U_\kappa^{-2} + U_\kappa^{-1} H_\kappa U_\kappa^{-1}] \tilde{\varphi_\kappa}(r) = [\lambda_\kappa + c^2(1 - \frac{|\kappa|}{\xi})] U_\kappa^{-2} \tilde{\varphi_\kappa}(r),$$

Then let $\Delta\mu_\kappa = c^2(1 - \frac{|\kappa|}{\xi})$, $\mu_\kappa = \lambda_\kappa + \Delta\mu_\kappa$,

$B_\kappa = \frac{d}{dr} + \xi\kappa/(|\kappa|r) - z/\kappa$ and $B_\kappa^+ = -B_{-\kappa}$.

Let

$$H_{\kappa,\mu} = U_\kappa^{-1} H_\kappa U_\kappa^{-1} + \Delta\mu_\kappa U_\kappa^{-2} = \begin{pmatrix} 0 & cB^+ \\ cB_\kappa & -2c^2 \end{pmatrix}, \tag{4.4}$$

the radial Dirac eigenvalue problem can be written in the form

$$H_{\kappa,\mu} \tilde{\varphi_\kappa}(r) = \mu_\kappa U_\kappa^{-2} \tilde{\varphi_\kappa}(r).$$

Then applied the Galarkin formulation of the radial Dirac eigenvalue problem in a finite dimensional subspace then the above transformation applied to the discretization of the Galarkin formulation of the radial Dirac eigenvalue problem given by

$(H_{\kappa,\mu})_{i,j}(\tilde{\varphi_\kappa})_{ij} = \mu_\kappa U_\kappa^{-2}(\tilde{\varphi_\kappa})_{ij,}$

notice that the vector

$$(\tilde{\varphi_\kappa})_{ij} = \begin{pmatrix} \tilde{f}_{\kappa ij} \\ \tilde{g}_{\kappa ij} \end{pmatrix},$$

is the unknowns and the matrix

$$(H_{\kappa,\mu})_{ij} = \begin{pmatrix} 0 & c(B_\kappa^+)_{ij} \\ c(B_\kappa)_{ij} & -2c^2 \end{pmatrix}, \tag{4.5}$$

when $(B_\kappa)_{ij}$ is the matrix obtained from the discretization of the Galarkin formulation on the finite basis set, so let
$A_\kappa = (B_\kappa)_{ij} - \mu_\kappa \kappa z/(|\kappa| c^2 \zeta)$ and $A_\kappa^+ = (B_\kappa^+)_{ij} - \mu_\kappa \kappa z/(|\kappa| c^2 \zeta)$, then suppose

$$Æ_\kappa = \begin{pmatrix} A_\kappa & 0 \\ 0 & -A_\kappa^+ \end{pmatrix},$$

then multiply

$$H_{\kappa,\mu}\tilde{\varphi_\kappa}(r) = \mu_\kappa U_\kappa^{-2}\tilde{\varphi_\kappa}(r)$$

by $Æ_\kappa$ from left, so we will have

$$(H_{-\kappa,\mu})_{ij} Æ_\kappa(\tilde{\varphi_\kappa})_{ij} = \mu_\kappa U_\kappa^{-2} Æ_\kappa(\tilde{\varphi_\kappa})_{ij},$$

by using

$$Æ_\kappa((H_\kappa)_{ij} - \mu_\kappa U_\kappa^{-2}) = ((H_{-\kappa,\mu})_{ij} - \mu_\kappa U_{-\kappa}^{-2})Æ_\kappa.$$

By Normalizing factor (N_κ) and for $\mu_\kappa \neq 0$, the eigenfunctions of $(H_{\kappa,\mu})_{ij}$ and $(H_{-\kappa,\mu})_{ij}$ are related by

$$(\tilde{\varphi_{-\kappa}})_{ij} = Æ_\kappa(\tilde{\varphi_\kappa})_{ij}/\sqrt{(N_\kappa)},$$

then by calculating we have

$$(H_{-\kappa,\mu})_{ij}(\tilde{\varphi_{-\kappa}})_{ij} = (\mu_\kappa U_\kappa^{-2})(\tilde{\varphi_{-\kappa}})_{ij}.$$

Note that $(H_{-\kappa,\mu})_{ij}$ and $(H_{\kappa,\mu})_{ij}$ are of the same size, then the number of their zero eigenvalues is the same.

The result is the eigenvalues of H_κ and $H_{-\kappa}$ would match in the projection method, onto the finite dimensional subspaces in the numerical computations, for more details, see [7].

4.2 Deeply in spuriosity in the Dirac eigenvalues problem

To study the spuriousity, consider the two equations of the radial Dirac equation

$(mc^2 + V(x))f(x) + c(-g'(x) + \frac{\kappa}{x}g(x)) = \lambda f(x),$
$c(f'(x) + \frac{\kappa}{x}f(x)) + (-mc^2 + V(x))g(x) = \lambda g(x).$

Using the notations $w^+ = mc^2 + V(x)$ and $w^- = -mc^2 + V(x)$,
the equations will be

$$w^+ f(x) + c(-g'(x) + \frac{\kappa}{x}g(x)) = \lambda f(x), \tag{4.6}$$

$$c(f'(x) + \frac{\kappa}{x}f(x)) + w^- g(x) = \lambda g(x). \tag{4.7}$$

From (4.6) we have $f(x) = \frac{1}{w^+ - \lambda}c(-g'(x) + \frac{\kappa}{x}g(x))$, and thus after differentiable it we get

$f'(x) = \frac{-V'}{(w^+ - \lambda)^2}(cg'(x) - c\frac{\kappa}{x}g(x)) + \frac{1}{w^+ - \lambda}(cg'' + \frac{c\kappa}{x^2}g(x) - \frac{c\kappa}{x}g').$

Substitute it in (4.7)

$c(\frac{-V'}{(w^+ - \lambda)^2}(cg'(x) - c\frac{\kappa}{x}g(x)) + \frac{1}{w^+ - \lambda}(cg'' + \frac{c\kappa}{x^2}g(x) - \frac{c\kappa}{x}g')) + \frac{c\kappa}{x}(\frac{1}{w^+ - \lambda}(cg'(x) - c\frac{\kappa}{x}g(x)) + w^- g(x) = \lambda g.$

Simplify to get,

$\frac{c^2}{w^+ - \lambda}g''(x) + [\frac{-c^2V'}{(w^+ - \lambda)^2} - \frac{c^2\kappa}{x(w^+ - \lambda)} + \frac{c^2\kappa}{x(w^+ - \lambda)}]g'(x) + [\frac{c^2V'(x)\kappa}{x(w^+ - \lambda)^2} + \frac{c^2\kappa}{x^2(w^+ - \lambda)} - \frac{c^2\kappa}{x^2(w^+ - \lambda)} + w^- - \lambda]g(x) = 0$

Consequently

$$g''(x) - \frac{-V'}{(w^+ - \lambda)}g'(x) + [\frac{V'(x)\kappa}{x(w^+ - \lambda)} - \frac{\kappa^2 - \kappa}{x^2} + \frac{(w^- - \lambda)(w^+ - \lambda}{c^2}]g(x) = 0 \tag{4.8}$$

use the notations

$\theta_1 = -\frac{-V'}{(w^+ - \lambda)},$

$\theta_2 = \frac{V'(x)\kappa}{x(w^+ - \lambda)} - \frac{\kappa^2 - \kappa}{x^2} + \frac{(w^- - \lambda)(w^+ - \lambda}{c^2}.$

to get

$g''(x) + \theta_1 g'(x) + \theta_2 g(x) = 0.$

About the other system, from (4.7) we have

$$g = \frac{c(f'(x)+\frac{\kappa}{x}f(x))}{(\lambda - w^-)},$$

then compute the derevative of g to get

$g'(x) = \frac{1}{(\lambda - w^-)}(cf''(x) + \frac{xc\kappa f'(x) - c\kappa f(x)}{x^2}) + (cf' + \frac{c\kappa f(x)}{x})(\frac{V'}{(\lambda - w^-)^2}).$

So, by substituting g and g' in (4.6) you will get

$w^+ f(x) + c(\frac{-cf''(x)}{w^- - \lambda} - \frac{xc\kappa f'}{(w^- - \lambda)x^2} + \frac{c\kappa f(x)}{(w^- - \lambda)x^2} - \frac{V'cf'}{(w^- - \lambda)^2} - \frac{c\kappa f(x)V'}{x(w^- - \lambda)^2} + \frac{c\kappa}{x}\frac{c(f'(x)+\frac{\kappa f(x)}{x}}{w^- - \lambda} = \lambda f(x).$

Equivalently

$-\frac{c^2 f''(x)}{w^- - \lambda} - \frac{xc^2\kappa f'(x)}{(\lambda - w^-)x^2} + \frac{V'c^2 f'(x)}{(\lambda - w^-)^2} + \frac{c^2\kappa f'(x)}{x(\lambda - w^-)} + \frac{c^2\kappa f(x)}{(\lambda - w^-)x^2} - \frac{c^2\kappa f(x)V'}{x(\lambda - w^-)^2} + \frac{c^2\kappa^2 f(x)}{x^2(\lambda - w^-)} - \lambda f(x) + w^+ f(x) = 0.$

Simplify to get

$f''(x) + \frac{\kappa}{x}f'(x) - \frac{V'f'}{(w^- - \lambda)} - \frac{\kappa f'(x)}{x} - \frac{\kappa}{x^2}f(x) + \frac{\kappa V'f(x)}{x(w^- - \lambda)} - \frac{\kappa^2}{x^2}f(x) + \frac{\lambda(w^- - \lambda)f(x)}{c^2} - \frac{(w^- - \lambda)w^+}{c^2}f = 0.$

Consequently

$$f'' - \frac{V\prime}{(w^- - \lambda)}f'(x) + \frac{(w^+ - \lambda)(w^- - \lambda)}{c^2} - \frac{\kappa^2 + \kappa}{x^2} - \frac{\kappa V'}{x(w^- - \lambda)}f(x) = 0.$$

Let

$\vartheta_1 = -\frac{V'}{(w^- - \lambda)},$

$\vartheta_2 = \frac{(w^+ - \lambda)(w^- - \lambda)}{c^2} - \frac{\kappa^2 + \kappa}{x^2} - \frac{\kappa V'}{x(w^- - \lambda)}.$

Then (4.6) has the form

$f'' + \vartheta_1 f'(x) + \vartheta_2 f(x) = 0,$

summing up, we have

$$f'' + \vartheta_1 f'(x) + \vartheta_2 f(x) = 0,$$

$$g''(x) + \theta_1 g'(x) + \theta_2 g(x) = 0,$$

where $\theta_1, \theta_2, \vartheta_1 \; and \; \vartheta_2$ are as defined above.
The two equations above are the formulae for the radial functions f and g separately.

Let us discuss the Pechlet and Damköhler numbers for these equations to determine whether they are convection dominated or diffusion dominated problems. Recall that

$$Pe_j = \frac{|u_j| hj}{2k},$$
$$Da_j = \frac{s_j h_j}{|\varphi_j|},$$

where h_j is the size of the element interval I_j, u_j and s_j are respectively the coefficients of the convection and the reaction terms corresponding to I_j, and k is the diffusivity size, see [3].
The related equations are convection dominated ones, when $Pe_j > 1$ or
$2Pe_jDa_j = (s_j h_j^2/k) > 1$, that to say the convection coefficient is larger than the diffusion coefficient.

Regarding the equations of f and g above, the quantity $2Pe_jDa_j$ for both equations admits very large values if small number of nodal points in the discretization is considered regardless the size of $|\lambda|$, Z and κ. Even with mesh refinement (increasing the number of nodal points) this quantities still admits very large values. Therefor, the two equations above and thus the Dirac equation are convection-dominated problem, this means that instability appears in the solution, i.e., the solution is perverted by spurious values.

CHAPTER 5

The Stability

5.1 The usual finite element method

By returning to the radial dirac eigenvalue problem, let $(\lambda, (f, g)) \in \Re \times H_0^1([0, \infty])^2$, such that $H_\kappa \varphi(x) = \lambda \varphi(x)$, where

$$H_\kappa = \begin{pmatrix} mc^2 + V(x) & c(-\frac{d}{dx} + \frac{\kappa}{x}) \\ c(\frac{d}{dx} + \frac{\kappa}{x}) & -mc^2 + V(x) \end{pmatrix}, \tag{5.1}$$

and

$$\varphi(x) = \begin{pmatrix} f(x) \\ g(x) \end{pmatrix}. \tag{5.2}$$

The two - equation system will be

$(mc^2 + V(x))f(x) + c(-g'(x) + \frac{\kappa}{x}g(x)) = \lambda f(x),$

$c(f'(x) + \frac{\kappa}{x}f(x)) + (-mc^2 + V(x))g(x) = \lambda g(x).$

The origin domain is $[0, \infty)$, but $x = 0$ is a singularity for the Coloumb potential $V(x) = \frac{-Z}{|x|}$, so usually it is assumed the Coloumb potential in $[R, \infty)$, where R is the radius of the nucleus, and another distributions, which are not singular at $x = 0$ of the charge in $[0, R]$ such as uniform distribution.

In our computation and since f and g are vanishing in the vicinity of x at infinity, and in order to compare our result to the exact values, we will assume a point nucleus is considered, i.e. a cut - off domain $\Omega = [a, b]$ is taken into account in all computations. Assume $\kappa_h : a = x_0 < x_1 < x_2 < ..... < x_{n+1} = b$, the result and partition of Ω with mesh size $h_i = x_i - x_{i-1}$, let V is a space of continuous functions, let V^L is subspace of continuous linear polynomials, and let V_h^L be the finite subspace consist of piecewise continuous linear polynomials on κ_h spanned by the basis functions φ_j, see [5].
Let V be the space of continuous functions and V^L be the subspace of continuous linear polynomials.

Now take k_h be a partition consisting of exponentially distributed points in $[a, b]$, and let $V_h^L \subset V^h$ be the finite subspace consisting of piecewise continuous linear polynomials spanned by the usual linear function φ_j on the partition k_h,

$$\varphi_j(x) = \begin{cases} \frac{x-x_{j-1}}{h_j}, & x \in I_J \\ \frac{x_{j+1}-x}{h_{j+1}}, & x \in I_{j+1} \\ 0, & \text{other wise.} \end{cases} , \tag{5.3}$$

where $j = 0, 1, 2, ..., n + 1$. If we have $f, g \in V_h^L$, then

$f(x) = \sum_{j=0}^{n} \zeta_j \varphi_j(x)$,

and

$g(x) = \sum_{j=0}^{n} \xi_j \varphi_j(x)$,

where ζ_j and ξ_j are the unknown values of the functions f and g at the nodal point x_j respectively, and φ_j is the basis function.

By Dirichlet conditions, the boundaries $\zeta_0 = \zeta_{n+1} = \xi_0 = \xi_{n+1}$=0, since $f = 0$ and $g = 0$ at $x = a, b$, so

$f(x) = \zeta_0\varphi_0(x) + \sum_{j=1}^{n} \zeta_j \varphi_j(x) + \zeta_{n+1}\varphi_{n+1}(x) = \sum_{j=1}^{n} \zeta_j \varphi_j(x)$,

and

$g(x) = \xi_0\varphi_0(x) + \sum_{j=1}^{n} \xi_j \varphi_j(x) + \xi_{n+1}\varphi_{n+1}(x) = \sum_{j=1}^{n} \xi_j \varphi_j(x)$.

5.2 Variation formulation

We now multiply by a test function $(v, 0)$ and $(0, v)$ and integrate over the domain, so it will be

$$\begin{pmatrix} v, 0 \end{pmatrix} \begin{pmatrix} (mc^2 + v)f + c(-g' + \frac{k}{x}g) \\ c(f' + \frac{k}{x}f) + (mc^2 + v)g \end{pmatrix} = \begin{pmatrix} v, 0 \end{pmatrix} \begin{pmatrix} \lambda f \\ \lambda g \end{pmatrix}, \tag{5.4}$$

equivalently

$$v(mc^2 + V)f + vc(-g + \frac{\kappa}{x}g) = v\lambda f. \tag{5.5}$$

Also,

$$\begin{pmatrix} 0, v \end{pmatrix} \begin{pmatrix} (mc^2 + V)f + c(-g' + \frac{k}{x}g) \\ c(f' + \frac{k}{x}f) + (mc^2 + V)g \end{pmatrix} = \begin{pmatrix} 0, v \end{pmatrix} \begin{pmatrix} \lambda f \\ \lambda g \end{pmatrix}, \tag{5.6}$$

which is equivalent to

$$vc(f' + \frac{\kappa}{x}f) + v(-mc^2 + V)g = v\lambda g. \tag{5.7}$$

Now integrate (5.5) and (5.7) over Ω to get:

$\int_\Omega (mc^2 + V)fvdx + \int_\Omega (-g' + \frac{k}{x}g)vcdx = \int_\Omega v\lambda f dx,$

and

$\int_\Omega (f' + \frac{\kappa}{x}f)v(x)cdx + \int_\Omega (-mc^2 + V)gv(x)dx = \int_\Omega v(x)\lambda g dx.$

Substitute the values of f and g to obtain :-

$\sum_{j=1}^n (\int_\Omega (mc^2 + V(x))\varphi_j(x)dx)\zeta_j + \sum_{j=1}^n (\int_\Omega (-c\varphi_j'(x) + \frac{c\kappa}{x}\varphi_j(x))v(x)dx)\xi_j$

$= \lambda \sum_{j=1}^n (\int_\Omega \varphi_j(x)v(x)dx)\zeta_j,$

and

$\sum_{j=1}^n (\int_\Omega (c\varphi_j'(x) + \frac{c\kappa}{x}\varphi_j(x))v(x)dx)\zeta_j + \sum_{j=1}^n (\int_\Omega -(mc^2 + V(x))\varphi_j(x)v(x)dx\xi_j$

$= \lambda \sum_{j=1}^n (\int_\Omega \varphi_j(x))v(x)dx)\xi_j,$

since the test function $v(x)$ from the same space V_L^h then $v(x) = \varphi_i(x)$, $i = 1, 2, 3....n$, thus above two equations become

first equation (1):

$\sum_{j=1}^{n}\big[mc^2\int_\Omega \varphi_j(x)\varphi_i(x)dx+\int_\Omega \varphi_j(x)\varphi_i(x)V(x)dx\big]\zeta_j+\sum_{j=1}^{n}\big[-c\int_\Omega \varphi_j'\varphi_i+$

$c\kappa\int_\Omega \varphi_j(x)\varphi_i(x)\frac{1}{x}dx\big]\xi_j=\lambda(\sum_{j=1}^{n}\int_\Omega \varphi_j\varphi_i dx)\zeta_j.$

And second equation (2):

$\sum_{j=1}^{n}\big[c\int_\Omega \varphi_j'\varphi_i+c\kappa\int_\Omega \varphi_j(x)\varphi_i(x)\frac{1}{x}dx\big]\zeta_j+\sum_{j=1}^{n}\big[-mc^2\int_\Omega \varphi_j(x)\varphi_i(x)dx+$

$\int_\Omega \varphi_j(x)\varphi_i(x)V(x)dx\big]\xi_j=\lambda(\sum_{j=1}^{n}\int_\Omega \varphi_j\varphi_i dx)\xi_j.$

Equations 1 for $i=1$ becomes

$(mc^2\int_\Omega \varphi_1(x)\varphi_1(x)dx+\int_\Omega \varphi_1(x)\varphi_1(x)V(x)dx)\zeta_1+(-c\int_\Omega \varphi_1'\varphi_1+$

$c\kappa\int_\Omega \varphi_1(x)\varphi_1(x)\frac{1}{x}dx)\xi_1+$

$(mc^2\int_\Omega \varphi_2(x)\varphi_1(x)dx+\int_\Omega \varphi_2(x)\varphi_1(x)V(x)dx)\zeta_2+(-c\int_\Omega \varphi_2'\varphi_1+$

$c\kappa\int_\Omega \varphi_2(x)\varphi_1(x)\frac{1}{x}dx)\xi_2+...+$

$(mc^2\int_\Omega \varphi_n(x)\varphi_1(x)dx+\int_\Omega \varphi_n(x)\varphi_1(x)V(x)dx)\zeta_n+(-c\int_\Omega \varphi_n'\varphi_1+$

$c\kappa\int_\Omega \varphi_n(x)\varphi_1(x)\frac{1}{x}dx)\xi_n=$

$\lambda(\int_\Omega \varphi_1\varphi_1dx)\zeta_1+\lambda(\int_\Omega \varphi_2\varphi_1dx)\zeta_2+...+\lambda(\int_\Omega \varphi_n\varphi_1dx)\zeta_n$

.
.
.

Also, and similarly, for $i=n$ for all j
we have

$(mc^2\int_\Omega \varphi_1(x)\varphi_n(x)dx+\int_\Omega \varphi_1(x)\varphi_n(x)V(x)dx)\zeta_1+(-c\int_\Omega' \varphi_1\varphi_n+$

$c\kappa\int_\Omega \varphi_1(x)\varphi_n(x)\frac{1}{x}dx)\xi_1+$

$(mc^2\int_\Omega \varphi_2(x)\varphi_n(x)dx+\int_\Omega \varphi_2(x)\varphi_n(x)V(x)dx)\zeta_2+(-c\int_\Omega \varphi_2'\varphi_n$

$+c\kappa\int_\Omega \varphi_2(x)\varphi_n(x)\frac{1}{x}dx)\xi_2+...+$

$(mc^2\int_\Omega \varphi_n(x)\varphi_n(x)dx+\int_\Omega \varphi_n(x)\varphi_n(x)V(x)dx)\zeta_n+(-c\int_\Omega \varphi\prime_n\varphi_n+$

$c\kappa\int_\Omega \varphi_n(x)\varphi_n(x)\frac{1}{x}dx)\xi_n=$

$\lambda(\int_\Omega \varphi_1\varphi_n dx)\zeta_1 + \lambda(\int_\Omega \varphi_2\varphi_n dx)\zeta_2 + ... + \lambda(\int_\Omega \varphi_n\varphi_n dx)\zeta_n$

.
.
.

Similarly for equation 2, when $i = 1$,for all $j = 1, 2, 3....n, wehave$

$(c\int_\Omega \varphi_1'\varphi_1 + c\kappa \int_\Omega \varphi_1\varphi_1\frac{1}{x}dx)\zeta_1 + (-mc^2 \int_\Omega \varphi_1\varphi_1 dx + \int_\Omega \varphi_1\varphi_1 V(x)dx)\xi_1 +$

$(c \int_\Omega \varphi_2'\varphi_1 + c\kappa \int_\Omega \varphi_2\varphi_1\frac{1}{x}dx)\zeta_2 + (-mc^2 \int_\Omega \varphi_2\varphi_1 dx + \int_\Omega \varphi_2\varphi_1 V(x)dx)\xi_2$

$+...+$

$(c \int_\Omega \varphi_n'\varphi_1 + c\kappa \int_\Omega \varphi_n\varphi_1\frac{1}{x}dx)\zeta_n + (-mc^2 \int_\Omega \varphi_n\varphi_1 dx + \int_\Omega \varphi_n\varphi_1 V(x)dx)\xi_n =$

$\lambda(\int_\Omega \varphi_1\varphi_1 dx)\xi_1 + \lambda(\int_\Omega \varphi_n\varphi_1 dx)\xi_2 + ... + \lambda(\int_\Omega \varphi_n\varphi_1 dx)\xi_n$.

.
.
.

For $i = n$ equation 2 becomes:-

$(c \int_\Omega \varphi_1'\varphi_n + c\kappa \int_\Omega \varphi_1\varphi_n\frac{1}{x}dx)\zeta_1 + (-mc^2 \int_\Omega \varphi_1\varphi_n dx + \int_\Omega \varphi_1\varphi_n V(x)dx)\xi_1 +$

$(c \int_\Omega \varphi_2'\varphi_n + c\kappa \int_\Omega \varphi_2\varphi_n\frac{1}{x}dx)\zeta_2 + (-mc^2 \int_\Omega \varphi_2\varphi_n dx + \int_\Omega \varphi_2\varphi_n V(x)dx)\xi_2$

$+...+$

$(c \int_\Omega \varphi_n'\varphi_n + c\kappa \int_\Omega \varphi_n\varphi_n\frac{1}{x}dx)\zeta_n + (-mc^2 \int_\Omega \varphi_n\varphi_n dx + \int_\Omega \varphi_n\varphi_n V(x)dx)\xi_n$

$= \lambda(\int_\Omega \varphi_1\varphi_n dx)\xi_1 + \lambda(\int_\Omega \varphi_2\varphi_n dx)\xi_2 + ... + \lambda(\int_\Omega \varphi_n\varphi_n dx)\xi_n.$

Using the notations

$A_0 = \int_\Omega \varphi_j(x)\varphi_i(x)dx,$

$A_I = \int_\Omega \varphi_j(x)\varphi_i(x)V(x)dx,$

$A_{II} = \int_\Omega \varphi_j(x)\varphi_i(x)\frac{1}{x}dx,$

$A_{III} = \int_\Omega \varphi_j'(x)\varphi_i(x)dx.$

Then we have the system $AX = \lambda BX_1$, where

$$A = \begin{pmatrix} mc^2[A_0] + [A_0] & -c[A_{III}] + c\kappa[A_{II}] \\ c[A_{III}] + c\kappa[A_{II}] & -mc^2[A_0] + [A_0] \end{pmatrix}, \tag{5.8}$$

$$B = \begin{pmatrix} [A_0] & 0 \\ 0 & [A_0] \end{pmatrix}, \tag{5.9}$$

and

$$X = \begin{pmatrix} \zeta \\ \xi \end{pmatrix}. \tag{5.10}$$

Note that A and B are both $2n \times 2n$ symmetric matrices, so we have a symmetric generalized eigenvalue problem.

Now, by using the streamline upwind Petrov-Galarkin (SUPG) FEM to stabilize the numerical computation we will introduce diffusion terms in our problem, instead of $(v, 0)$ and $(0, v)$ which is the usual Galerkin FEM, we will multiply the last system by $(v, \tau v')$ and $(\tau v', v)$ to introduce diffusion terms,

$$\begin{pmatrix} v & \tau v' \end{pmatrix} \begin{pmatrix} (mc^2 + V(x))f + c(-g' + \frac{\kappa}{x}g) \\ c(f' + \frac{\kappa}{x}f) + (-mc^2 + V(x))g \end{pmatrix} = \begin{pmatrix} v & \tau v' \end{pmatrix} \begin{pmatrix} \lambda f \\ \lambda g \end{pmatrix}.$$

$v(mc^2 + V(x))f + vc(-g' + \frac{\kappa}{x}g) + \tau v'(c(f' + \frac{\kappa}{x}f) + \tau v'(-mc^2 + v)g$

$= v\lambda f + \tau v'\lambda g$

$$\begin{pmatrix} \tau v' & v \end{pmatrix} \begin{pmatrix} (mc^2 + V(x))f + c(-g' + \frac{\kappa}{x}g) \\ c(f' + \frac{\kappa}{x}f) + (-mc^2 + V(x))g \end{pmatrix} = \begin{pmatrix} \tau v' & v \end{pmatrix} \begin{pmatrix} \lambda f \\ \lambda g \end{pmatrix},$$

$\tau v'(mc^2 + V(x))f + \tau v'c(-g' + \frac{\kappa}{x}g) + v(c(f' + \frac{\kappa}{x}f) + v(-mc^2 + v)g = \tau v'\lambda f + v\lambda g.$

So the system will be after integrate it on the domain,

$\int_\Omega (mc^2 + V(x))v(x)f(x)dx + \int_\Omega (-cg' + c\frac{\kappa}{x}g)v(x)dx + \tau \int_\Omega (cf' + c\frac{\kappa}{x}f)v'(x)$

$+ (-mc^2 + V(x))gv' + -v'\lambda g dx) = \lambda \int_\Omega f(x)v(x)dx.$

$\int_\Omega (c(f' + \frac{\kappa}{x} f) + (-mc^2 + V(x))g)v(x)dx + \tau(\int_\Omega (v'(mc^2 + V(x))f$

$+ v'c(-g' + \frac{\kappa}{x} g) - v'\lambda f) = \lambda \int_\Omega gv(x)dx.$

To discretize the equation let $f, g \in V_h^L$ then $f(x) = \sum_{j=1}^n \zeta_j \varphi_j(x)$ and

$g(x) = \sum_{j=1}^n \xi_j \varphi_j(x)$.

Also v$\in V_h^L so$,v=$\varphi_i, i = 1, 2, 3....n$,

$\int_\Omega (mc^2 + V(x))\varphi_i f(x)dx + \int_\Omega (-cg' + c\frac{\kappa}{x} g)\varphi_i f(x)dx + \tau \int_\Omega T_1(f(x), g(x))\varphi_i' dx$

$= \lambda \int_\Omega f(x)\varphi(x) - g(x)\varphi_i' dx.$

$\int_\Omega (cf\prime + c\frac{\kappa}{x} f)\varphi_i'(x)dx + \int_\Omega (-mc^2 + V(x))g(x)\varphi_i(x) + \tau \int_\Omega T_2(f(x), g(x))\varphi_i'(x)dx$

$= \lambda \int_\Omega g(x)\varphi_i(x) - f(x)\varphi_i'(x)dx$

Last term one:

$$T_1(f(x), g(x)) = (-mc^2 + V(x) - \lambda)g(x) + cf'(x) + c\frac{\kappa}{x} f(x).$$

Last term two:

$$T_2(f(x), g(x)) = (mc^2 + V(x) - \lambda)f(x) - cg'(x) + c\frac{\kappa}{x} g(x).$$

put $\mathring{A}_0 = \int_\Omega f(x)v(x)dx$,

$\mathring{A}_I = \int_\Omega f(x)V(x)v(x)dx$,

$\mathring{A}_{II} = \int_\Omega g'(x)v(x)dx$,

$\mathring{A}_{III} = \int_\Omega g(x)v(x)\frac{1}{x}dx$,

$\mathring{A}_{IV} = \int_\Omega f'(x)v(x)dx$,

$\mathring{A}_V = \int_\Omega f(x)v(x)\frac{1}{x}dx$,

$\mathring{A}_{VI} = \int_\Omega g(x)v(x)dx$,

$\mathring{A}_{VII} = \int_\Omega V(x)g(x)v(x)dx$.

And let

$\tau_0 = \int_\Omega \tau f'(x)v'(x)dx$,

$\tau_I = \int_\Omega \tau f(x) \frac{v'(x)}{x} dx,$

$\tau_{II} = \int_\Omega \tau g(x) v'(x) dx,$

$\tau_{III} = \int_\Omega \tau v'(x) f(x) dx,$

$\tau_{IV} = \int_\Omega \tau v'(x) g'(x) dx,$

$\tau_V = \int_\Omega \tau v'(x) g(x) \frac{1}{x} dx,$

$\tau_{VI} = \int_\Omega \tau f(x) v(x) dx,$

$\tau_{VII} = \int_\Omega \tau g(x) v(x).$

So, our system will be by noting $f = \sum_1^n \varphi_j \zeta_j$ and $g = \sum_1^n \varphi_j \xi_j$.

$mc^2 \mathring{A}_0 + \mathring{A}_I - c\mathring{A}_{II} + c\kappa \mathring{A}_{III} + \tau[c\tau_0 + c\kappa\tau_I - mc^2\tau_{II} + v\tau_{II}] = \lambda(\tau_{VI} - \tau_{II}),$

and the other equation will be

$c\mathring{A}_{IV} + c\kappa \mathring{A}_V - mc^2 \mathring{A}_{VI} + \mathring{A}_{II} + \tau[mc^2\tau_{III} + V(x)\tau_{III} -$
$c\tau_{IV} + \kappa c\tau_V = \lambda(\tau_{VII} - \tau_{III}).$

So, as in the usual FEM, we will have the generalized eigenvalue problem

$(mc^2 \mathring{A}_0 + \mathring{A}_I) + (-c\mathring{A}_{II} + c\kappa \mathring{A}_{III}) + \tau[(c\tau_0 + c\kappa\tau_I) +$
$(-mc^2\tau_{II} + v\tau_{II})] = \lambda(\tau_{VI} - \tau_{II}),$

$(c\mathring{A}_{IV} + c\kappa) + (-mc^2 \mathring{A}_{VI} + \mathring{A}_{II} + \tau[(mc^2\tau_{III} + V(x)\tau_{III}) +$
$(-c\tau_{IV} + \kappa c\tau_V)] = \lambda(\tau_{VII} - \tau_{III}).$

Let

$$\mathring{A} = \begin{pmatrix} mc^2 \mathring{A}_0 + \mathring{A}_I & -c\mathring{A}_{II} + c\kappa \mathring{A}_{III} \\ c\mathring{A}_{IV} + c\kappa & -mc^2 \mathring{A}_{VI} + \mathring{A}_{II} \end{pmatrix}. \tag{5.11}$$

$$Y = (\zeta_j, \xi_j)^T$$

$$\tau_* = \begin{pmatrix} c\tau_0 + c\kappa\tau_I & -mc^2\tau_{II} + v\tau_{II} \\ mc^2\tau_{III} & -c\tau_{IV} + \kappa c\tau_V \end{pmatrix}. \tag{5.12}$$

$$B = \begin{pmatrix} \tau_{VI} & 0 \\ 0 & \tau_{VII} \end{pmatrix}. \tag{5.13}$$

$$B_0 = \begin{pmatrix} -\tau_{II} & 0 \\ 0 & -\tau_{III} \end{pmatrix}. \tag{5.14}$$

Then the system will be $(\mathring{A} + \tau_*)Y = \lambda(B + B_0)Y$

5.3 Numerical example

We take an example to arrive to a system of matrix equations to approximate solution, so consider

$$-u'' + 200u' = x^2, x \in [0, 1]$$
$$u(0) = u(1) = 0.$$

In the first step we will multiply the differential equation by a test function $v(x)$ such that $v(0) = v(1) = 0$ and integrate it over the domain

$$\int_0^1 -u''vdx + \int_0^1 200u'vdx = \int_0^1 x^2vdx. \tag{5.15}$$

Integrate the first term by parts to get

$$-\int_0^1 u''vdx = -(u'(1)v(1) - u'(0)v(0)) - \int_0^1 u'v'dx.$$

Then by using $v(0) = v(1) = 0$, we get

$$-\int_0^1 u''vdx = \int_0^1 u'v'dx.$$

Then (5.15) becomes

$$\int_0^1 u'v'dx + 200\int_0^1 u'vdx = \int_0^1 x^2vdx.$$

After that when we make discretization on partition by introducing V_h as in Chapter 3, we will find $U(x) \in V_h$, such that

$$\int_0^1 U'v'dx + 200\int_0^1 U'vdx = \int_0^1 x^2vdx,$$

for all $v \in V_h$,

Let $U \in V_h^L$, where V_h^L is the subspace of linear functions defined on the partition k_h as before, then $U(x) = \sum_{j=1}^n \xi_j\varphi_j(x)$, where $\xi_j = U(x_j)$ is the nodal values of U at $x_j, j = 1, 2, ...n$.

Thus for $U = \varphi_i$, $i = 1, 2, 3, ...n$ we have

$$\xi_j \sum_{j=1}^{n} \left(\int_0^1 \varphi_j' \varphi_i' dx + 200 \int_0^1 \varphi_j' \varphi_i dx \right) = \int_0^1 x^2 \varphi_i dx, \qquad i = 1, 2, 3, ..., n.$$

which is equivalent to $(A + B)\xi = b$, where

$A = \int_0^1 \varphi_j' \varphi_i' dx$ is the stiffness matrix,

$B = \int_0^1 \varphi_j' \varphi_i dx$ is the convection matrix,

$b = \int_0^1 x^2 \varphi_i dx$ is the load vector, and

$\xi = (\xi_1, \xi_2, \xi_3, ..., \xi_n)^T$.

5.4 Method of stabilization

As discussed before, we have two methods of stabilization

1- SUPG.

2- Artificial diffusivity.

The first method for the diffusion convection problem:-
Firstly let

$$< -u'', v + \tau v' > + < 200u', v + \tau v' > = < x^2, v + \tau v' >$$

$$- < u'', v > - < u'', \tau v' > + 200 < u', v > + 200 < u', \tau v' > =$$
$$< x^2, v > + < x^2, \tau v' > .$$

Integrating by parts in the first term

$$- < u'', v > = - \int_0^1 u'' v dx = -(u(1)v'(1) - u(0)v'(0)) + \int_0^1 u'v' dx =$$

$$\int_0^1 u'v' dx.$$

So, the system will be

$$< u', v' > + \tau < u'', v' > + 200 < u', v > + 200\tau < u', v' > =$$
$$< x^2, v > + \tau < x^2, v' >$$

Note that $\sum_{j=1}^{n} \xi_j \varphi_j$

such that, see [12],

$$\varphi_j = \begin{cases} \frac{x-x_{j-1}}{h_j}, & x_{j-1} < x < x_j \\ \frac{x_{j+1}-x}{h_{j+1}}, & x_j < x < x_{j+1} \\ 0, & \text{otherwise.} \end{cases} \tag{5.16}$$

Since linear basis functions are assumed, then $< u'', v' >= 0$, so for $v = \varphi_i, i = 1, 2, ..., n$,

$\xi_j < \varphi_j', \varphi_i' > +200\xi_j < \varphi_j', \varphi_i' > +200\xi_j\tau < \varphi', \varphi_i' >$
$=< x^2, \varphi_i' > +\tau < x^2, \varphi_i' > .$

Then

$\xi_j \int_0^1 \varphi_j'\varphi_i' dx = 200\xi_j \int_0^1 \varphi_j'\varphi_i dx + 200\xi_j\tau \int_0^1 \varphi_j', \varphi_i' dx.$

Secondly adding an artificial diffusion term $\tau < u', v' >$

$- < u'', v > +200 < u', v > +\tau < u', v' >=< x^2, v > .$

Example:
Returning back to the problem

$$\begin{aligned} -u'' + 200u' &= x^2, \\ u(0) = u(1) &= 0. \end{aligned} \tag{5.17}$$

In the figure below dotted - standard curve is the usual finite element method approximation, and the continuous curve is the exact solution.
The solution is obtained using the Matlab software where the code is appended. Here we use 10 subintervals and for the figure after we use 50 subintervals.

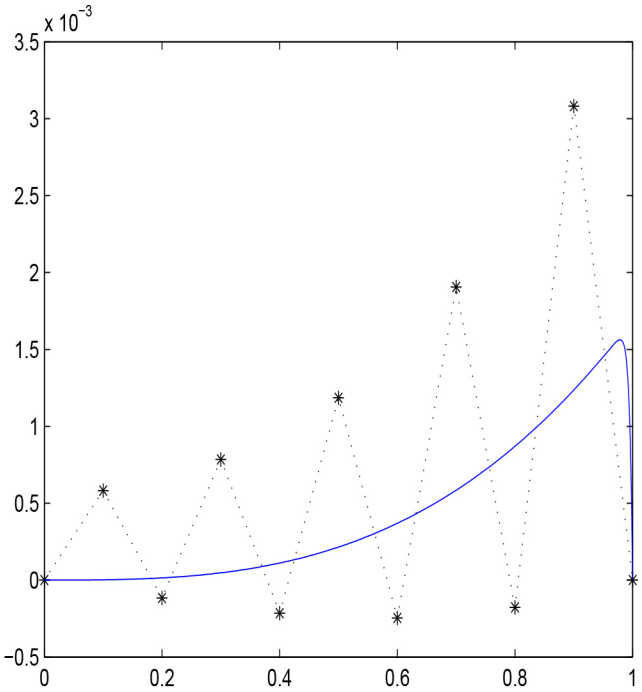

Figure 5.1: *Usual FEM with n = 10*

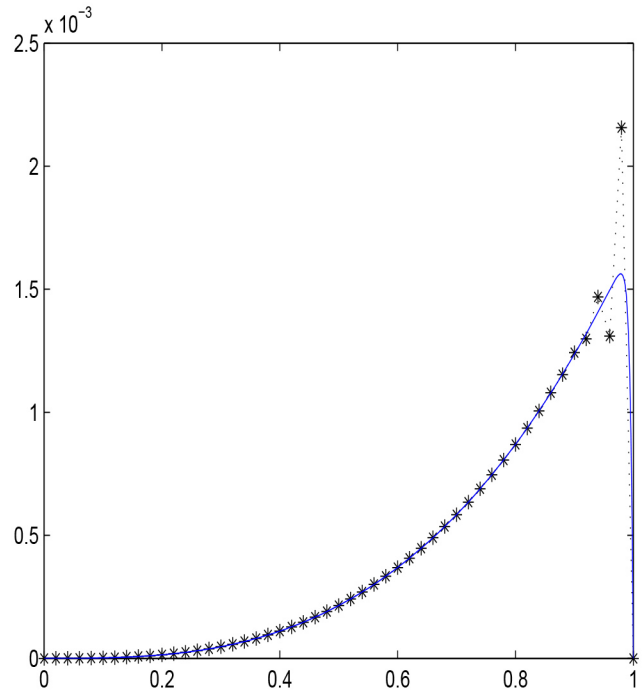

Figure 5.2: *Usual FEM with n = 50*

Its clear that increasing the number of subintervals (increasing the number of nodal points) provides more stable solution.

Next step we search another way to give us more stable solution, so we will use a stream line upwind Petrove Galarkin finite element method
Define

$$Pe = \frac{|p|h}{2|k|},$$

where
h is the step size,
p is the convection coefficient,
k is the diffusion coefficient.
Then a stability parameter that can be used for both the SUPG and the artificial diffusion methods is

$$\tau = \begin{cases} \frac{h^2}{12k}, & Pe < 0, \\ \frac{h}{2|P|}, & \text{otherwise.} \end{cases} \tag{5.18}$$

For the problem we consider $-u'' + 200u' = x^2$, $pe = \frac{200h}{2(1)} = 200h$.

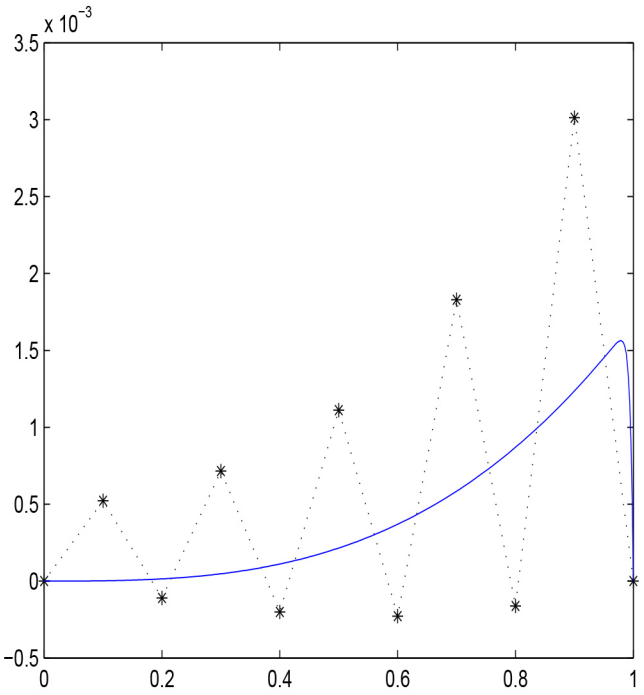

Figure 5.3: *SUPG FEM with* $n = 10$*, and given* τ

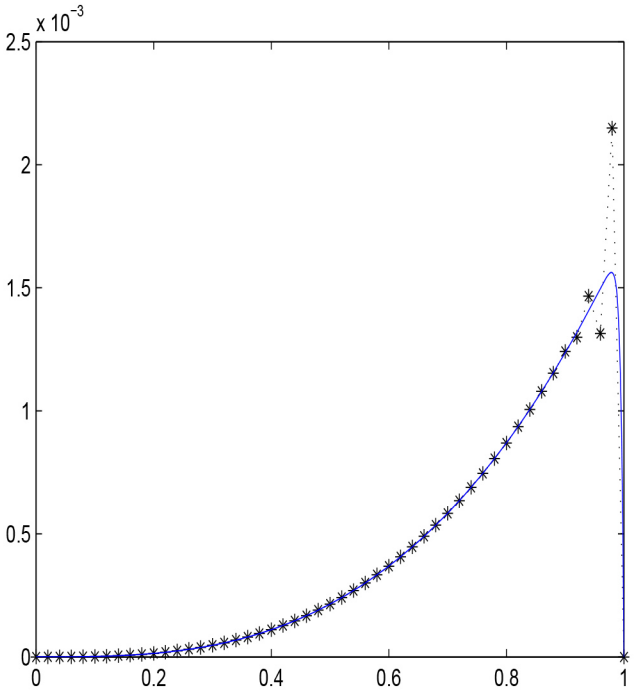

Figure 5.4: *SUPG FEM with n=50 and given τ*

Figures 5.3 and 5.4 is the SUPG finite element method with the stability parameter defined in equation (5.17).

Note that it seems the SUPG finite element method for the problem is not as defined. This way be because that convection term is very large.
We tried in figures 5.5 and 5.6 in next page to enlarge the size of the stability parameter, so we have consider $\tau^* = 100\tau$ where τ is as before.

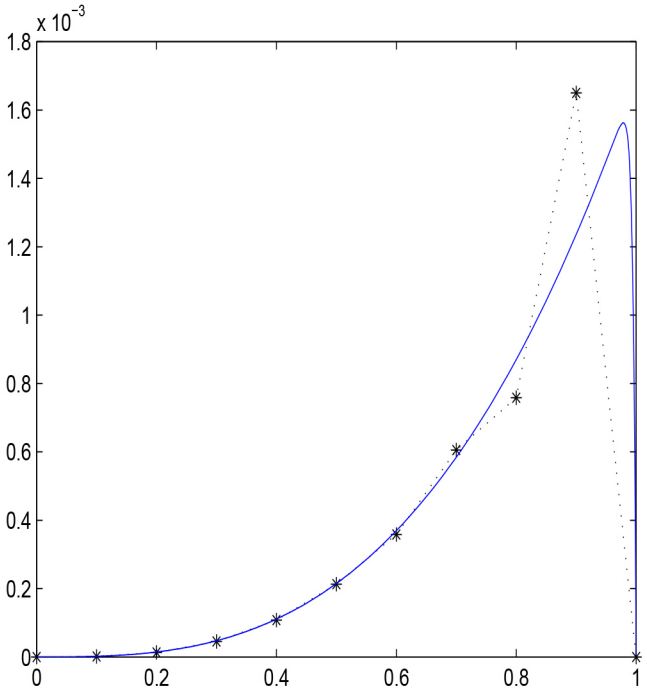

Figure 5.5: *SUPG FEM with n=10,* $\tau^* = 100\tau$

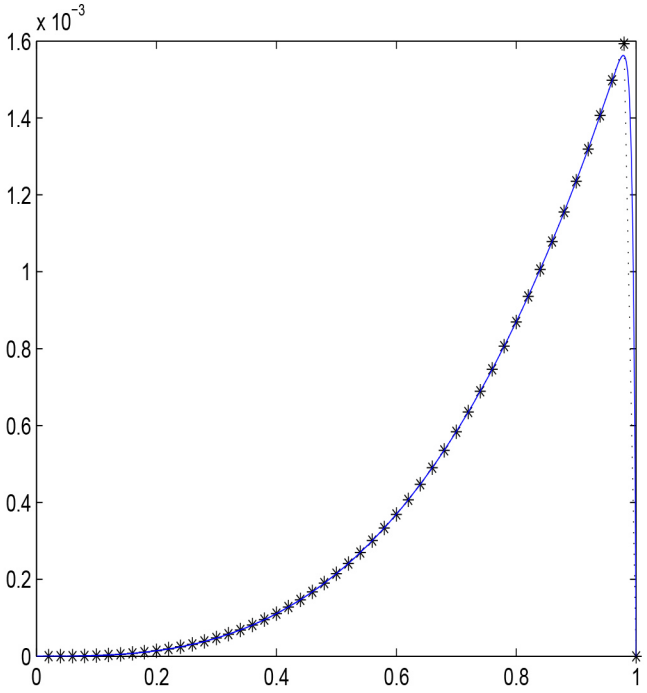

Figure 5.6: *SUPG FEM with n=50, and* $\tau^* = 100\tau$

we will see, by using Artificial Diffusivity FEM to go to our goal, note new figures:

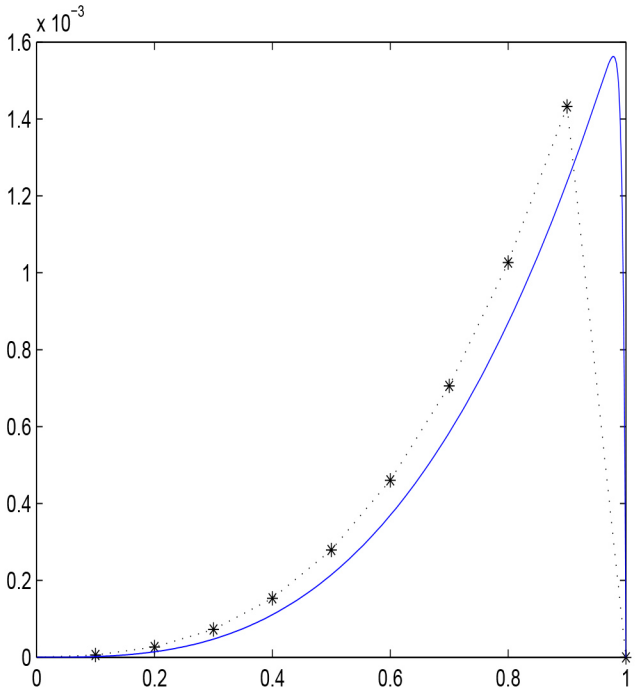

Figure 5.7: *Artificial diffusivity with n=10, and given τ*

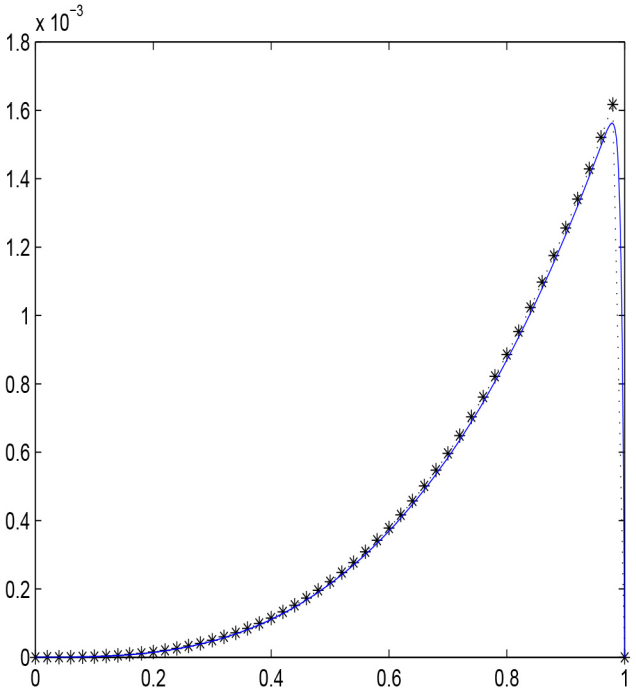

Figure 5.8: *Artificial diffusivity with n=50, and given τ*

In our figures the exact solution and the approximation solution are more near of each other, when we put $n = 10$ and $n = 50$, $Pe = \frac{|P|h}{2|k|} = \frac{200h}{2(1)}$ and use $\tau = (1-\frac{1}{P})\frac{h|P|}{2}$ by $h = \frac{b-a}{n}$, so n controls our problem to get a clearly and faster stability solution more than last way.

5.5 Numerical computation for the Dirac equation

Let the hydrogen like cesium ion an example to applied it on Dirac equation, firstly we will talk about the cesium, The German chemist Robert Bunsen and physicist Gustav Kirchhoff discovered caesium in 1860 by the newly developed method of flame spectroscopy, cesium is a chemical element with symbol Cs and atomic number 55. It is a soft, silvery-gold alkali metal with a melting point of $28.5C(83.3F)$, which makes it one of only five elemental metals that are liquid at or near room temperature.

Caesium has physical and chemical properties similar to those of rubidium and potassium,is used in medical applications, industrial gauges, and hydrology, see [21].
for the computation of the Dirac equation we will use the value of the following parameter

$$\tau = \frac{h_{j+1}-h_j}{3c(h_{j+1}+h_j)} - \frac{2}{3c}\frac{(h_{j+1}-h_j)^2}{(h_{j+1}+h_j)^2} + \sqrt{\frac{n^2}{9c^2}\left(\frac{(h_{j+1}-h_j)^2}{(h_{j+1}+h_j)^2} + \frac{4(h_{j+1}-h_j)^4}{(h_{j+1}+h_j)^4}\right) + \frac{4}{9c^2}\frac{(h_{j+1}-h_j)^3}{(h_{j+1}+h_j)^3}\left(\frac{h_{j+1}}{h_j} - 1\right) + \frac{h_{j+1}^2}{9}\frac{(h_{j+1}-h_j)^2}{(h_{j+1}+h_j)^2}}$$

This stability parameter is hopefully to be a promising value for a stable computation of the Dirac eigenvalue problem, and still under study.
In the first step we take $n = 200$, $\tau = 0$, $\kappa = -1$, $\kappa = 1$, note the result in our table:

Table 5.1: *The first computed eigenvalues of the electron in the Hydrogen-like Cesium ion using the FEM with linear basis functions for point nucleus.*

$\kappa = -1$		$\kappa = 1$	
Approximation	Exact	Approximation	Exact
-1578.87320060	-1578.87360265	-1578.87320060	SE
-398.9566872	-398.956306798	-398.956687265	-398.956306798
-175.3992611	-175.398291704	-175.399261141	-175.398291704
-98.3146870774	SE	-98.3146870774	SE
-97.8651048302	-97.8637523276	-97.8651048301	-97.863752327
-62.2787931909	-62.276000655	-62.2787931910	-62.276000655
-43.0720344726	-43.0679643349	-43.0720344725	-43.0679643349
-32.0173363865	SE	-32.0173363865	SE
-31.5483506089	-31.5427684300	-31.5483506089	-31.5427684300
-24.0984432587	-24.0910611453	-24.0984432587	-24.0910611453
-19.007241800	-18.9978018903	-19.0072418005	-18.9978018903
-15.8499059510	SE	-15.8499059510	SE

In this table $\tau = 0$ and take $n = 200$ by $Z = 55$, we note the result when put $\kappa = -1$, by using Matlab program .
Let consider another computational example by changing the value of κ, where we put $n = 400$ in hope of removing some of the spurious values.

Table 5.2: *The second computed eigenvalues of the electron in the Hydrogen-like Cesium ion using the FEM with linear basis functions for point nucleus.*

$\kappa = -2$		$\kappa = 2$	
Approximation	Exact	Approximation	Exact
-382.010539731	-382.01053973	-382.01053973	SE
-170.36161073	-170.361591916	-170.36161073	-170.3615919
-95.7467825975	-95.7467241032	-95.746782597	-95.74672410
-61.2791310287	SE	-61.279131028	SE
-61.1964033958	-61.1962864664	-61.196403396	-61.19628646
-42.4453689798	-42.4451768769	-42.445368979	-42.44517687
-31.1519085430	-31.1516254457	-31.151908543	-31.15162544
-23.9334440715	SE	-23.933444071	SE
-23.8299893952	-23.8295998649	-23.829989395	-23.82959986
-18.8150132340	-18.8145019570	-18.815013234	-18.81450195
-15.2309276862	-15.2302793351	-15.230927686	-15.23027933
-12.6917587425	SE	-12.691758742	SE

in another case:

Table 5.3: *The third computed eigenvalues of the electron in the Hydrogen-like Cesium ion using the FEM with linear basis functions for point nucleus.*

$\kappa = -3$		$\kappa = 3$	
Approximation	Exact	Approximation	Exact
-168.81434161	-168.81434161	-168.81434161	SE
-95.091778219	-95.091763120	-95.091778219	-95.09176312
-60.860865460	-60.86081163	-60.860865459	-60.8608116399
-42.313942179	SE	-42.31394217	SE
-42.251231877	-42.251116413	-42.25123187	-42.2511164130
-31.029692079	-31.029494954	-31.029692079	-31.0294949540
-23.748132283	-23.747835358	-23.748132283	-23.7478353584
-18.847807771	SE	-18.847807771	SE
-18.757525249	-18.75711145	-18.757525249	-18.7571114528
-15.189012251	-15.188465078	-15.189012251	-15.1884650785
-12.549625701	-12.548928919	-12.549625701	-12.5489289199
-10.644784653	SE	-10.644784653	SE

note the value in this table we have a gap between the value, so we have a spurious eigenvalue with big error, so we search an another type of solution to avoid the spurious value.
In this step we will take the value of τ, and $n = 400$:

Table 5.4: *The first computed eigenvalues of the electron in the Hydrogen-like Cesium ion using the FEM with linear basis functions for point nucleus.*

$\kappa = -1$		$\kappa = 1$	
Approximation	Exact	Approximation	Exact
-1579.00886636029	-1578.87360265569		
-398.974367746669	-398.956306798929	-398.955088749677	-398.956306798929
-175.402514465255	-175.398291704165	-175.397104187825	-175.398291704165
-97.8644458942581	-97.8637523276193	-97.862364150842	-97.8637523276193
-62.2752471932399	-62.276000655289	-62.274316244584	-62.276000655289
-43.0663366037479	-43.0679643349577	-43.0658991912351	-43.0679643349577
-31.5404325844393	-31.5427684300666	-31.5402390722593	-31.5427684300666
-24.0880446945594	-24.0910611453473	-24.0879844519659	-24.0910611453473
-18.994076615636	-18.9978018903676	-18.9940948737203	-18.9978018903676
-15.359214474789	-15.3637022562034	-15.3592817620265	-15.3637022562034
-12.6751255190175	-12.680442497367	-12.6752249109049	-12.680442497367
-10.6369420086448	-10.6431622655691	-10.6370632654107	-10.6431622655691

when we increase the nodal, the error will be decrease, and there is no spurious values, note in another example by changing κ:

Table 5.5: *The second computed eigenvalues of the electron in the Hydrogen-like Cesium ion using the FEM with linear basis functions for point nucleus.*

$\kappa = -2$		$\kappa = 2$	
Approximation	Exact	Approximation	Exact
-382.009928611933	-382.010539731622		
-170.360578731685	-170.361591916717	-170.361322160388	-170.361591916717
-95.7453719200967	-95.7467241032646	-95.746058972989	-95.7467241032646
-61.1945813879065	-61.1962864664019	-61.1951840908005	-61.1962864664019
-42.4430625224159	-42.4451768769468	-42.4436034051178	-42.4451768769468
-31.1490300490987	-31.1516254457274	-31.1495283981239	-31.1516254457274
-23.8264459048878	-23.8295998649264	-23.8269147463543	-23.8295998649264
-18.8107094054358	-18.8145019570329	-18.8111573362949	-18.8145019570329
-15.2257667250196	-15.2302793351628	-15.2261995149529	-15.2302793351628
-12.5750135313865	-12.5803287953531	-12.5754351529904	-12.5803287953531
-10.5599326339579	-10.566134227578	-10.5603458944388	-10.566134227578
-8.99234050780433	-8.99951322337802	-8.99274743832575	-8.99951322337802

Table 5.6: *The third computed eigenvalues of the electron in the Hydrogen-like Cesium ion using the FEM with linear basis functions for point nucleus.*

$\kappa = -3$		$\kappa = 3$	
Approximation	Exact	Approximation	Exact
-168.814051321289	-168.81434161295		
-95.0910516024815	-95.091763120734	-95.0915878203668	-95.091763120734
-60.8596420783615	-60.8608116399737	-60.8602714654626	-60.8608116399737
-42.2494612825431	-42.2511164130774	-42.2501010435481	-42.2511164130774
-31.0273060722539	-31.0294949540439	-31.0279375295904	-31.0294949540439
-23.745051024187	-23.7478353584775	-23.7456710955412	-23.7478353584775
-18.7536620875399	-18.7571114528109	-18.754271694972	-18.7571114528109
-15.1842762536035	-15.1884650785214	-15.1848771815203	-15.1884650785214
-12.5439229120166	-12.5489289199722	-12.5445169212326	-12.5489289199722
-10.5360558034554	-10.5419592044673	-10.5366444084175	-10.5419592044673
-8.97362350631374	-8.98050656266787	-8.97420794879145	-8.98050656266787
-7.73391642486604	-7.74186320553054	-7.73449770416846	-7.74186320553054

see above example, we put $n = 400$ (increases the nodal) and we changing κ, so the results are good, there is no spurious values, and the error will be decreasing, very small error between the exact and approximate solution, so the value of the stability parameter τ is best to avoid the spurious eigenvalues.

Conclusion

displayed Dirac equation and presented the Coulomb potential, the radial Dirac operator. Also, we reviewed the adjoint operator and some theorems of self adjoint. We discussed some basic theory of the finite element method. After that we discussed the spurious eigenvalues in the computation of the Dirac operator and tried to stabilize it.

The main goal of this thesis is to stabilize the numerical computation, we used some methods as usual finite element method by using Dirichlet conditions, the SUPG method by assuming the test function, and introduced diffusion terms in our problem. At the end, we gave numerical example and applied the methods on it, and compared the given result to appear the best method to access our goal.

Appendix

Matlab code for figures

Solver artificial diffusion

```
%The program is for the solution of -ku''+pu'+qu=f, with homogeneous
%Dirichlet boundary conditions using FEM and stable FEM by adding
% artificial viscosity (adding artificial diffusion term).
clear all
clc
format long
k=1;
p=200;
q=0;
Int=[0 1];
a=Int(1);
b=Int(2);

disp('"n" IS THE NUMBER OF SUBINTERVALS.');
n=input('n=');
h=(b-a)/n;
xx=a:h:b;
d1=1;
d2=0;

c4=[0.347854845137454 0.652145154862546 0.652145154862546 0.347854845137454];
x4=[-0.861136311594053 -0.339981043584856 0.339981043584856 0.861136311594053];
```

```
Pe=(abs(p)*h)/(2*k);
tau=(1-1/Pe)*(h*abs(p))/2;
fori=1:n
h(i+1)=xx(i+1)-xx(i);
end

%Computing M000
fori=d1:n-2+d2
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;
   m11m0(i-d1+1,i+1-
d1+1)=dd1*(c4(1)*f1(dd1*x4(1)+dd2,xx(i+2),h(i+2))*f2(dd1*x4(1)+dd2,xx(i+1),h(i+2)) ...
                        +c4(2)*f1(dd1*x4(2)+dd2,xx(i+2),h(i+2))*f2(dd1*x4(2)+dd2,xx(i+1),h(i+2)) ...
                        +c4(3)*f1(dd1*x4(3)+dd2,xx(i+2),h(i+2))*f2(dd1*x4(3)+dd2,xx(i+1),h(i+2)) ...
                        +c4(4)*f1(dd1*x4(4)+dd2,xx(i+2),h(i+2))*f2(dd1*x4(4)+dd2,xx(i+1),h(i+2)));
m11m0(i+1-d1+1,i-d1+1)=m11m0(i-d1+1,i+1-d1+1);
end
fori=d1:n-1+d2
   dd1=(xx(i+1)-xx(i))/2;
   dd2=(xx(i+1)+xx(i))/2;
   central11_m0_1(i)=dd1*(c4(1)*f1(dd1*x4(1)+dd2,xx(i+1),h(i+1))^2 ...
                        +c4(2)*f1(dd1*x4(2)+dd2,xx(i+1),h(i+1))^2 ...
                        +c4(3)*f1(dd1*x4(3)+dd2,xx(i+1),h(i+1))^2 ...
                        +c4(4)*f1(dd1*x4(4)+dd2,xx(i+1),h(i+1))^2);
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;
   central11_m0_2(i)=dd1*(c4(1)*f2(dd1*x4(1)+dd2,xx(i+1),h(i+2))^2 ...
                        +c4(2)*f2(dd1*x4(2)+dd2,xx(i+1),h(i+2))^2 ...
                        +c4(3)*f2(dd1*x4(3)+dd2,xx(i+1),h(i+2))^2 ...
                        +c4(4)*f2(dd1*x4(4)+dd2,xx(i+1),h(i+2))^2);
```

```
m11m0(i-d1+1,i-d1+1)=central11_m0_1(i)+central11_m0_2(i);
end

M000=m11m0;

%computing M010
fori=d1:n-2+d2
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;
c11c0(id1+1,i+1d1+1)=dd1*(c4(1)*difff1(dd1*x4(1)+dd2,xx(i+2),h(i+2))*f2(dd1*x4(1)+dd2,xx(i+1)
,h(i+2)) ...
                        +c4(2)*difff1(dd1*x4(2)+dd2,xx(i+2),h(i+2))*f2(dd1*x4(2)+dd2,xx(i+1),h(i+2))
...
                        +c4(3)*difff1(dd1*x4(3)+dd2,xx(i+2),h(i+2))*f2(dd1*x4(3)+dd2,xx(i+1),h(i+2))
...

+c4(4)*difff1(dd1*x4(4)+dd2,xx(i+2),h(i+2))*f2(dd1*x4(4)+dd2,xx(i+1),h(i+2)));
end
fori=d1:n-1+d2
   dd1=(xx(i+1)-xx(i))/2;
   dd2=(xx(i+1)+xx(i))/2;

central11_c0_1(i)=dd1*(c4(1)*difff1(dd1*x4(1)+dd2,xx(i+1),h(i+1))*f1(dd1*x4(1)+dd2,xx(i+1),h(i+
1)) ...
                        +c4(2)*difff1(dd1*x4(2)+dd2,xx(i+1),h(i+1))*f1(dd1*x4(2)+dd2,xx(i+1),h(i+1)) ...
                        +c4(3)*difff1(dd1*x4(3)+dd2,xx(i+1),h(i+1))*f1(dd1*x4(3)+dd2,xx(i+1),h(i+1)) ...
                        +c4(4)*difff1(dd1*x4(4)+dd2,xx(i+1),h(i+1))*f1(dd1*x4(4)+dd2,xx(i+1),h(i+1)));
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;

central11_c0_2(i)=dd1*(c4(1)*difff2(dd1*x4(1)+dd2,xx(i+1),h(i+2))*f2(dd1*x4(1)+dd2,xx(i+1),h(i+
2)) ...
                        +c4(2)*difff2(dd1*x4(2)+dd2,xx(i+1),h(i+2))*f2(dd1*x4(2)+dd2,xx(i+1),h(i+2)) ...
```

```
                +c4(3)*difff2(dd1*x4(3)+dd2,xx(i+1),h(i+2))*f2(dd1*x4(3)+dd2,xx(i+1),h(i+2)) ...
                +c4(4)*difff2(dd1*x4(4)+dd2,xx(i+1),h(i+2))*f2(dd1*x4(4)+dd2,xx(i+1),h(i+2)));
c11c0(i-d1+1,i-d1+1)=central11_c0_1(i)+central11_c0_2(i);
end
fori=d1:n-2+d2
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;
   c11c0(i+1-d1+1,i-
d1+1)=dd1*(c4(1)*difff2(dd1*x4(1)+dd2,xx(i+1),h(i+2))*f1(dd1*x4(1)+dd2,xx(i+2),h(i+2)) ...
                +c4(2)*difff2(dd1*x4(2)+dd2,xx(i+1),h(i+2))*f1(dd1*x4(2)+dd2,xx(i+2),h(i+2))
...
                +c4(3)*difff2(dd1*x4(3)+dd2,xx(i+1),h(i+2))*f1(dd1*x4(3)+dd2,xx(i+2),h(i+2))
...

+c4(4)*difff2(dd1*x4(4)+dd2,xx(i+1),h(i+2))*f1(dd1*x4(4)+dd2,xx(i+2),h(i+2)));
end


M010=c11c0;



%computing M110
fori=d1:n-2+d2
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;
   s11s(i-d1+1,i+1-
d1+1)=dd1*(c4(1)*difff1(dd1*x4(1)+dd2,xx(i+2),h(i+2))*difff2(dd1*x4(1)+dd2,xx(i+1),h(i+2)) ...

+c4(2)*difff1(dd1*x4(2)+dd2,xx(i+2),h(i+2))*difff2(dd1*x4(2)+dd2,xx(i+1),h(i+2)) ...

+c4(3)*difff1(dd1*x4(3)+dd2,xx(i+2),h(i+2))*difff2(dd1*x4(3)+dd2,xx(i+1),h(i+2)) ...

+c4(4)*difff1(dd1*x4(4)+dd2,xx(i+2),h(i+2))*difff2(dd1*x4(4)+dd2,xx(i+1),h(i+2)));
end
```

```
fori=d1:n-1+d2
    dd1=(xx(i+1)-xx(i))/2;
    dd2=(xx(i+1)+xx(i))/2;

central11_s0_1(i)=dd1*(c4(1)*difff1(dd1*x4(1)+dd2,xx(i+1),h(i+1))*difff1(dd1*x4(1)+dd2,xx(i+1),
h(i+1)) ...

+c4(2)*difff1(dd1*x4(2)+dd2,xx(i+1),h(i+1))*difff1(dd1*x4(2)+dd2,xx(i+1),h(i+1)) ...

+c4(3)*difff1(dd1*x4(3)+dd2,xx(i+1),h(i+1))*difff1(dd1*x4(3)+dd2,xx(i+1),h(i+1)) ...

+c4(4)*difff1(dd1*x4(4)+dd2,xx(i+1),h(i+1))*difff1(dd1*x4(4)+dd2,xx(i+1),h(i+1)));
    dd1=(xx(i+2)-xx(i+1))/2;
    dd2=(xx(i+2)+xx(i+1))/2;

central11_s0_2(i)=dd1*(c4(1)*difff2(dd1*x4(1)+dd2,xx(i+1),h(i+2))*difff2(dd1*x4(1)+dd2,xx(i+1),
h(i+2)) ...

+c4(2)*difff2(dd1*x4(2)+dd2,xx(i+1),h(i+2))*difff2(dd1*x4(2)+dd2,xx(i+1),h(i+2)) ...

+c4(3)*difff2(dd1*x4(3)+dd2,xx(i+1),h(i+2))*difff2(dd1*x4(3)+dd2,xx(i+1),h(i+2)) ...

+c4(4)*difff2(dd1*x4(4)+dd2,xx(i+1),h(i+2))*difff2(dd1*x4(4)+dd2,xx(i+1),h(i+2)));
s11s(i-d1+1,i-d1+1)=central11_s0_1(i)+central11_s0_2(i);
end
fori=d1:n-2+d2
    dd1=(xx(i+2)-xx(i+1))/2;
    dd2=(xx(i+2)+xx(i+1))/2;
    s11s(i+1-d1+1,i-
d1+1)=dd1*(c4(1)*difff2(dd1*x4(1)+dd2,xx(i+1),h(i+2))*difff1(dd1*x4(1)+dd2,xx(i+2),h(i+2)) ...

+c4(2)*difff2(dd1*x4(2)+dd2,xx(i+1),h(i+2))*difff1(dd1*x4(2)+dd2,xx(i+2),h(i+2)) ...

+c4(3)*difff2(dd1*x4(3)+dd2,xx(i+1),h(i+2))*difff1(dd1*x4(3)+dd2,xx(i+2),h(i+2)) ...
```

```
+c4(4)*difff2(dd1*x4(4)+dd2,xx(i+1),h(i+2))*difff1(dd1*x4(4)+dd2,xx(i+2),h(i+2)));
end

M110=s11s;

%Computing the loud vector bb00
fori=d1:n-1+d2
   dd1=(xx(i+1)-xx(i))/2;
   dd2=(xx(i+1)+xx(i))/2;
   bb00_1(i)=dd1*(c4(1)*f1(dd1*x4(1)+dd2,xx(i+1),h(i+1))*f(dd1*x4(1)+dd2) ...
                  +c4(2)*f1(dd1*x4(2)+dd2,xx(i+1),h(i+1))*f(dd1*x4(2)+dd2) ...
                  +c4(3)*f1(dd1*x4(3)+dd2,xx(i+1),h(i+1))*f(dd1*x4(3)+dd2) ...
                  +c4(4)*f1(dd1*x4(4)+dd2,xx(i+1),h(i+1))*f(dd1*x4(4)+dd2));
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;
   bb00_2(i)=dd1*(c4(1)*f2(dd1*x4(1)+dd2,xx(i+1),h(i+2))*f(dd1*x4(1)+dd2) ...
                  +c4(2)*f2(dd1*x4(2)+dd2,xx(i+1),h(i+2))*f(dd1*x4(2)+dd2) ...
                  +c4(3)*f2(dd1*x4(3)+dd2,xx(i+1),h(i+2))*f(dd1*x4(3)+dd2) ...
                  +c4(4)*f2(dd1*x4(4)+dd2,xx(i+1),h(i+2))*f(dd1*x4(4)+dd2));
bb00(i-d1+1)=bb00_1(i)+bb00_2(i);
end
bb00=bb00';
bb=bb00;
s=k*M110+p*M010+q*M000+tau*M110;
yy=double(s)\double(bb);
yy=yy';
z=[0 yy 0];
plot(xx,z,':*k');
hold on
```

```
%The following exact solution is when k=1, p=200, q=0.

fplot('(x^3)/600+(x^2)/40000+x/4000000+(1/600+1/40000+1/4000000)/(exp(200)-
1)+(1/600+1/40000+1/4000000)/(1-exp(200))*exp(200*x)',[a b]);
```

Solver SUPG

```
%The program is for the solution of -ku''+pu'+qu=f, with homogeneous
%Dirichlet boundary conditions using FEM and SUPG FEM.
clear all
clc
format long
k=1;
p=200;
q=0;
Int=[0 1];
a=Int(1);
b=Int(2);

disp('"n" IS THE NUMBER OF SUBINTERVALS.');
n=input('n=');
h=(b-a)/n;
xx=a:h:b;
d1=1;
d2=0;

c4=[0.347854845137454 0.652145154862546 0.652145154862546 0.347854845137454];
x4=[-0.861136311594053 -0.339981043584856 0.339981043584856 0.861136311594053];

Pe=(abs(p)*h)/(2*k);
ifPe<1
tau=h^2/(12*k)
else
tau=h/(2*abs(p))
end
```

```
fori=1:n
h(i+1)=xx(i+1)-xx(i);
end

%Computing M000
fori=d1:n-2+d2
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;
   m11m0(i-d1+1,i+1-
d1+1)=dd1*(c4(1)*f1(dd1*x4(1)+dd2,xx(i+2),h(i+2))*f2(dd1*x4(1)+dd2,xx(i+1),h(i+2)) ...
                        +c4(2)*f1(dd1*x4(2)+dd2,xx(i+2),h(i+2))*f2(dd1*x4(2)+dd2,xx(i+1),h(i+2)) ...
                        +c4(3)*f1(dd1*x4(3)+dd2,xx(i+2),h(i+2))*f2(dd1*x4(3)+dd2,xx(i+1),h(i+2)) ...
                        +c4(4)*f1(dd1*x4(4)+dd2,xx(i+2),h(i+2))*f2(dd1*x4(4)+dd2,xx(i+1),h(i+2)));
m11m0(i+1-d1+1,i-d1+1)=m11m0(i-d1+1,i+1-d1+1);
end
fori=d1:n-1+d2
   dd1=(xx(i+1)-xx(i))/2;
   dd2=(xx(i+1)+xx(i))/2;
   central11_m0_1(i)=dd1*(c4(1)*f1(dd1*x4(1)+dd2,xx(i+1),h(i+1))^2 ...
                       +c4(2)*f1(dd1*x4(2)+dd2,xx(i+1),h(i+1))^2 ...
                       +c4(3)*f1(dd1*x4(3)+dd2,xx(i+1),h(i+1))^2 ...
                       +c4(4)*f1(dd1*x4(4)+dd2,xx(i+1),h(i+1))^2);
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;
   central11_m0_2(i)=dd1*(c4(1)*f2(dd1*x4(1)+dd2,xx(i+1),h(i+2))^2 ...
                       +c4(2)*f2(dd1*x4(2)+dd2,xx(i+1),h(i+2))^2 ...
                       +c4(3)*f2(dd1*x4(3)+dd2,xx(i+1),h(i+2))^2 ...
                       +c4(4)*f2(dd1*x4(4)+dd2,xx(i+1),h(i+2))^2);
m11m0(i-d1+1,i-d1+1)=central11_m0_1(i)+central11_m0_2(i);
end
```

```
M000=m11m0;


%computing M010
fori=d1:n-2+d2
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;
   c11c0(i-d1+1,i+1-
d1+1)=dd1*(c4(1)*difff1(dd1*x4(1)+dd2,xx(i+2),h(i+2))*f2(dd1*x4(1)+dd2,xx(i+1),h(i+2)) ...
                        +c4(2)*difff1(dd1*x4(2)+dd2,xx(i+2),h(i+2))*f2(dd1*x4(2)+dd2,xx(i+1),h(i+2))
...
                        +c4(3)*difff1(dd1*x4(3)+dd2,xx(i+2),h(i+2))*f2(dd1*x4(3)+dd2,xx(i+1),h(i+2))
...

+c4(4)*difff1(dd1*x4(4)+dd2,xx(i+2),h(i+2))*f2(dd1*x4(4)+dd2,xx(i+1),h(i+2)));
end
fori=d1:n-1+d2
   dd1=(xx(i+1)-xx(i))/2;
   dd2=(xx(i+1)+xx(i))/2;

central11_c0_1(i)=dd1*(c4(1)*difff1(dd1*x4(1)+dd2,xx(i+1),h(i+1))*f1(dd1*x4(1)+dd2,xx(i+1),h(i+
1)) ...
                        +c4(2)*difff1(dd1*x4(2)+dd2,xx(i+1),h(i+1))*f1(dd1*x4(2)+dd2,xx(i+1),h(i+1)) ...
                        +c4(3)*difff1(dd1*x4(3)+dd2,xx(i+1),h(i+1))*f1(dd1*x4(3)+dd2,xx(i+1),h(i+1)) ...
                        +c4(4)*difff1(dd1*x4(4)+dd2,xx(i+1),h(i+1))*f1(dd1*x4(4)+dd2,xx(i+1),h(i+1)));
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;

central11_c0_2(i)=dd1*(c4(1)*difff2(dd1*x4(1)+dd2,xx(i+1),h(i+2))*f2(dd1*x4(1)+dd2,xx(i+1),h(i+
2)) ...
                        +c4(2)*difff2(dd1*x4(2)+dd2,xx(i+1),h(i+2))*f2(dd1*x4(2)+dd2,xx(i+1),h(i+2)) ...
                        +c4(3)*difff2(dd1*x4(3)+dd2,xx(i+1),h(i+2))*f2(dd1*x4(3)+dd2,xx(i+1),h(i+2)) ...
                        +c4(4)*difff2(dd1*x4(4)+dd2,xx(i+1),h(i+2))*f2(dd1*x4(4)+dd2,xx(i+1),h(i+2)));
```

```
c11c0(i-d1+1,i-d1+1)=central11_c0_1(i)+central11_c0_2(i);
end
fori=d1:n-2+d2
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;
   c11c0(i+1-d1+1,i-
d1+1)=dd1*(c4(1)*difff2(dd1*x4(1)+dd2,xx(i+1),h(i+2))*f1(dd1*x4(1)+dd2,xx(i+2),h(i+2)) ...
                        +c4(2)*difff2(dd1*x4(2)+dd2,xx(i+1),h(i+2))*f1(dd1*x4(2)+dd2,xx(i+2),h(i+2))
...
                        +c4(3)*difff2(dd1*x4(3)+dd2,xx(i+1),h(i+2))*f1(dd1*x4(3)+dd2,xx(i+2),h(i+2))
...

+c4(4)*difff2(dd1*x4(4)+dd2,xx(i+1),h(i+2))*f1(dd1*x4(4)+dd2,xx(i+2),h(i+2)));
end


M010=c11c0;


%computing M110
fori=d1:n-2+d2
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;
   s11s(i-d1+1,i+1-
d1+1)=dd1*(c4(1)*difff1(dd1*x4(1)+dd2,xx(i+2),h(i+2))*difff2(dd1*x4(1)+dd2,xx(i+1),h(i+2)) ...

+c4(2)*difff1(dd1*x4(2)+dd2,xx(i+2),h(i+2))*difff2(dd1*x4(2)+dd2,xx(i+1),h(i+2)) ...

+c4(3)*difff1(dd1*x4(3)+dd2,xx(i+2),h(i+2))*difff2(dd1*x4(3)+dd2,xx(i+1),h(i+2)) ...

+c4(4)*difff1(dd1*x4(4)+dd2,xx(i+2),h(i+2))*difff2(dd1*x4(4)+dd2,xx(i+1),h(i+2)));
end
fori=d1:n-1+d2
   dd1=(xx(i+1)-xx(i))/2;
   dd2=(xx(i+1)+xx(i))/2;
```

```
central11_s0_1(i)=dd1*(c4(1)*difff1(dd1*x4(1)+dd2,xx(i+1),h(i+1))*difff1(dd1*x4(1)+dd2,xx(i+1),
h(i+1)) ...

+c4(2)*difff1(dd1*x4(2)+dd2,xx(i+1),h(i+1))*difff1(dd1*x4(2)+dd2,xx(i+1),h(i+1)) ...

+c4(3)*difff1(dd1*x4(3)+dd2,xx(i+1),h(i+1))*difff1(dd1*x4(3)+dd2,xx(i+1),h(i+1)) ...

+c4(4)*difff1(dd1*x4(4)+dd2,xx(i+1),h(i+1))*difff1(dd1*x4(4)+dd2,xx(i+1),h(i+1)));
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;

central11_s0_2(i)=dd1*(c4(1)*difff2(dd1*x4(1)+dd2,xx(i+1),h(i+2))*difff2(dd1*x4(1)+dd2,xx(i+1),
h(i+2)) ...

+c4(2)*difff2(dd1*x4(2)+dd2,xx(i+1),h(i+2))*difff2(dd1*x4(2)+dd2,xx(i+1),h(i+2)) ...

+c4(3)*difff2(dd1*x4(3)+dd2,xx(i+1),h(i+2))*difff2(dd1*x4(3)+dd2,xx(i+1),h(i+2)) ...

+c4(4)*difff2(dd1*x4(4)+dd2,xx(i+1),h(i+2))*difff2(dd1*x4(4)+dd2,xx(i+1),h(i+2)));
s11s(i-d1+1,i-d1+1)=central11_s0_1(i)+central11_s0_2(i);
end
fori=d1:n-2+d2
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;
   s11s(i+1-d1+1,i-
d1+1)=dd1*(c4(1)*difff2(dd1*x4(1)+dd2,xx(i+1),h(i+2))*difff1(dd1*x4(1)+dd2,xx(i+2),h(i+2)) ...

+c4(2)*difff2(dd1*x4(2)+dd2,xx(i+1),h(i+2))*difff1(dd1*x4(2)+dd2,xx(i+2),h(i+2)) ...

+c4(3)*difff2(dd1*x4(3)+dd2,xx(i+1),h(i+2))*difff1(dd1*x4(3)+dd2,xx(i+2),h(i+2)) ...

+c4(4)*difff2(dd1*x4(4)+dd2,xx(i+1),h(i+2))*difff1(dd1*x4(4)+dd2,xx(i+2),h(i+2)));
end
```

```
M110=s11s;



%computing M100
fori=d1:n-2+d2
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;
   c11c0(i-d1+1,i+1-
d1+1)=dd1*(c4(1)*f1(dd1*x4(1)+dd2,xx(i+2),h(i+2))*difff2(dd1*x4(1)+dd2,xx(i+1),h(i+2)) ...

+c4(2)*f1(dd1*x4(2)+dd2,xx(i+2),h(i+2))*difff2(dd1*x4(2)+dd2,xx(i+1),h(i+2)) ...

+c4(3)*f1(dd1*x4(3)+dd2,xx(i+2),h(i+2))*difff2(dd1*x4(3)+dd2,xx(i+1),h(i+2)) ...

+c4(4)*f1(dd1*x4(4)+dd2,xx(i+2),h(i+2))*difff2(dd1*x4(4)+dd2,xx(i+1),h(i+2)));
end
fori=d1:n-1+d2
   dd1=(xx(i+1)-xx(i))/2;
   dd2=(xx(i+1)+xx(i))/2;

central11_c0_1(i)=dd1*(c4(1)*f1(dd1*x4(1)+dd2,xx(i+1),h(i+1))*difff1(dd1*x4(1)+dd2,xx(i+1),h(i+
1)) ...

+c4(2)*f1(dd1*x4(2)+dd2,xx(i+1),h(i+1))*difff1(dd1*x4(2)+dd2,xx(i+1),h(i+1)) ...

+c4(3)*f1(dd1*x4(3)+dd2,xx(i+1),h(i+1))*difff1(dd1*x4(3)+dd2,xx(i+1),h(i+1)) ...

+c4(4)*f1(dd1*x4(4)+dd2,xx(i+1),h(i+1))*difff1(dd1*x4(4)+dd2,xx(i+1),h(i+1)));
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;

central11_c0_2(i)=dd1*(c4(1)*f2(dd1*x4(1)+dd2,xx(i+1),h(i+2))*difff2(dd1*x4(1)+dd2,xx(i+1),h(i+
2)) ...
```

```
+c4(2)*f2(dd1*x4(2)+dd2,xx(i+1),h(i+2))*difff2(dd1*x4(2)+dd2,xx(i+1),h(i+2)) ...

+c4(3)*f2(dd1*x4(3)+dd2,xx(i+1),h(i+2))*difff2(dd1*x4(3)+dd2,xx(i+1),h(i+2)) ...

+c4(4)*f2(dd1*x4(4)+dd2,xx(i+1),h(i+2))*difff2(dd1*x4(4)+dd2,xx(i+1),h(i+2)));
c11c0(i-d1+1,i-d1+1)=central11_c0_1(i)+central11_c0_2(i);
end
fori=d1:n-2+d2
   dd1=(xx(i+2)-xx(i+1))/2;
   dd2=(xx(i+2)+xx(i+1))/2;
   c11c0(i+1-d1+1,i-
d1+1)=dd1*(c4(1)*f2(dd1*x4(1)+dd2,xx(i+1),h(i+2))*difff1(dd1*x4(1)+dd2,xx(i+2),h(i+2)) ...

+c4(2)*f2(dd1*x4(2)+dd2,xx(i+1),h(i+2))*difff1(dd1*x4(2)+dd2,xx(i+2),h(i+2)) ...

+c4(3)*f2(dd1*x4(3)+dd2,xx(i+1),h(i+2))*difff1(dd1*x4(3)+dd2,xx(i+2),h(i+2)) ...

+c4(4)*f2(dd1*x4(4)+dd2,xx(i+1),h(i+2))*difff1(dd1*x4(4)+dd2,xx(i+2),h(i+2)));
end


M100=c11c0;


%Computing the loud vector bb00
fori=d1:n-1+d2
   dd1=(xx(i+1)-xx(i))/2;
   dd2=(xx(i+1)+xx(i))/2;
   bb00_1(i)=dd1*(c4(1)*f1(dd1*x4(1)+dd2,xx(i+1),h(i+1))*f(dd1*x4(1)+dd2) ...
                  +c4(2)*f1(dd1*x4(2)+dd2,xx(i+1),h(i+1))*f(dd1*x4(2)+dd2) ...
                  +c4(3)*f1(dd1*x4(3)+dd2,xx(i+1),h(i+1))*f(dd1*x4(3)+dd2) ...
                  +c4(4)*f1(dd1*x4(4)+dd2,xx(i+1),h(i+1))*f(dd1*x4(4)+dd2));
   dd1=(xx(i+2)-xx(i+1))/2;
```

```
    dd2=(xx(i+2)+xx(i+1))/2;
    bb00_2(i)=dd1*(c4(1)*f2(dd1*x4(1)+dd2,xx(i+1),h(i+2))*f(dd1*x4(1)+dd2) ...
                        +c4(2)*f2(dd1*x4(2)+dd2,xx(i+1),h(i+2))*f(dd1*x4(2)+dd2) ...
                        +c4(3)*f2(dd1*x4(3)+dd2,xx(i+1),h(i+2))*f(dd1*x4(3)+dd2) ...
                        +c4(4)*f2(dd1*x4(4)+dd2,xx(i+1),h(i+2))*f(dd1*x4(4)+dd2));
bb00(i-d1+1)=bb00_1(i)+bb00_2(i);
end
bb00=bb00';
%Computing the loud vector bb10
fori=d1:n-1+d2
    dd1=(xx(i+1)-xx(i))/2;
    dd2=(xx(i+1)+xx(i))/2;
    bb10_1(i)=dd1*(c4(1)*difff1(dd1*x4(1)+dd2,xx(i+1),h(i+1))*f(dd1*x4(1)+dd2) ...
                        +c4(2)*difff1(dd1*x4(2)+dd2,xx(i+1),h(i+1))*f(dd1*x4(2)+dd2) ...
                        +c4(3)*difff1(dd1*x4(3)+dd2,xx(i+1),h(i+1))*f(dd1*x4(3)+dd2) ...
                        +c4(4)*difff1(dd1*x4(4)+dd2,xx(i+1),h(i+1))*f(dd1*x4(4)+dd2));
    dd1=(xx(i+2)-xx(i+1))/2;
    dd2=(xx(i+2)+xx(i+1))/2;
    bb10_2(i)=dd1*(c4(1)*difff2(dd1*x4(1)+dd2,xx(i+1),h(i+2))*f(dd1*x4(1)+dd2) ...
                        +c4(2)*difff2(dd1*x4(2)+dd2,xx(i+1),h(i+2))*f(dd1*x4(2)+dd2) ...
                        +c4(3)*difff2(dd1*x4(3)+dd2,xx(i+1),h(i+2))*f(dd1*x4(3)+dd2) ...
                        +c4(4)*difff2(dd1*x4(4)+dd2,xx(i+1),h(i+2))*f(dd1*x4(4)+dd2));
bb10(i-d1+1)=bb10_1(i)+bb10_2(i);
end
bb10=bb10';
bb=bb00+tau*bb10;
s=k*M110+p*M010+q*M000+tau*(p*M110+q*M100);
yy=double(s)\double(bb);
yy=yy';
z=[0 yy 0];
```

```
plot(xx,z,':*k');

hold on

%The following exact solution is when k=1, p=200, q=0.

fplot('(x^3)/600+(x^2)/40000+x/4000000+(1/600+1/40000+1/4000000)/(exp(200)-
1)+(1/600+1/40000+1/4000000)/(1-exp(200))*exp(200*x)',[a b]);
```

Bibliography

[1] Bernd Thaler, The Dirac Equation, Bernd Thaler, 1991, Springer-Verlag.

[2] Eng. Yong, Lay, Eigenvalue Problem for Dirac Equation, 2007, may.

[3] Hasan Almanasreh, hp-Cloud Approximation of the Dirac Eigenvalue Problem: The Way Of Stability, Hasan Almanasreh, Springer,2007.

[4] Hasan Almanasreh, The Dirac Equation: Numerical and Asymptotic Analysis, University of Gothenburg, 2012.

[5] Hasan Almanasreh, Stabilized Finite Element Method for The Radial Dirac Equation, in Journal of Computational Physics, November, 2011.

[6] Alexender Nelson Brooks, APetrov-Galarkin Finite Element Formulation for Convection Flows, California Institute of Technology Pasadena, 1981, may 15.

[7] I.I.Tupitsyn and V.M.Shabaev, Spurious states of the Dirac Equation in a Fimite Basis Set, 2008, Optika Spekroskopiya,203-209.

[8] V. M. Shabaev, I. I. Tupitsyn, V. A. Yerokhin, G. Plunien, and G. Soff, Dual kinetic balance approach to basis-set expansions for the Dirac equation, Phys. Rev. Lett., 93(2004).

[9] Mathieu Lewin, E'ricSe're'Spurious Modes in Dirac Calculations and How to avoid Them, Springer International Publishing Switzerland, 2014.

[10] Trends and Tricks in Spectral Theory, LyonellBoulton, Michael Lwvitin, 2007, Ediciones IVIC.

[11] Ruperto P. BonetChaple ,Numerical Stabilization of Convection-Diffusion-Reaction Problems, 2628 CD Delft, Netherlands, 2006, March .

[12] A. N. Brooks and T. J. R. Huges, Streamline Upwind/Petrove-Galerkin formulations for convection dominated flows with particular emphasis on the incompressible Navier – Stokes equation, compute. Methods Appl. Mech. Endng., 32(1982).

[13]P. T. Dawkins, S. R. Dunbar, and R. W. Douglass, The origin and nature of spurious eigenvalues in the spectral Tau method, J. Comp. Phys., 147(1998), pp. 441-462.

[14] P. A. B. De Sampaio, A Petrov-Galerkin/modified operator formulation for convection-diffusion problems, Int. J. Numer. methodsEngng., 30(1990).

[15] C. F. Fisher and O. Zatsarinny, A B-splines Galerkin method for the Dirac equation, Comp. Phys. Comm., 180(2009), pp. 879-886.

[16] S. Idelsohn, N. Nigro, M. Storti, and G. Buscaglia, A Petrov-Galerkin formulation for advectionreaction-diffusion problems, Comput. Methods Appl. Mech. Engng., 136(1996).

[17] G. Pestka, Spurious roots in the algebraic Dirac equation, Chem. Phys. Lett. 376(2003), pp. 659-661

[18] W. Schroeder and I. Wolf, The origin of spurious modes in numerical solutions of electromagnetic field eigenvalue problems, IEEE Tran. onMicr. Theory and Tech., 42(1994), pp. 644-653.

[19] S. Salomonson and P. Oster, ¨ Relativistic all-order pair functions from a discretized single particle Dirac Hamiltonian, Phys. Rev. A, 40(1989), pp. 5548-5558.

[20] Regina C. Almedia, Renato S. Silva , A stable petrov- Galerkin method for convection – dominated problems, Elsevier Science S. A, 1997.

[21] William A. Hart, O.F. Beumel, Jr. and Thomas P. Whaley, The Chemistry of Lithium, Sodium, Potassium, Rubidium, Cesium and Francium, 1973.