

https://dspace.org/

Inverse eigenvalue problem

Boamah, Edward Kwasi

2016-08-16
http://hdl.handle.net/10539/20865

Downloaded from a DSpace repository. https://dspace.org/

UNIVERSITY OF THE WITWATERSRAND

INVERSE EIGENVALUE PROBLEM

By
Edward Kwasi Boamah

A RESEARCH REPORT
submitted to the Faculty of Science of the
University of the Witwatersrand in **partial fulfilment** of the degree of
MASTER OF SCIENCE

Johannesburg, Republic of South Africa
December 3, 1998.

UNIVERSITY OF THE WITWATERSRAND

A RESEARCH REPORT
submitted in partial fulfilment of the degree of
MASTER OF SCIENCE

INVERSE EIGENVALUE PROBLEM

E K Boamah

Approved:

Prof. C J Wright (Supervisor)	Head, Department of Computatinal & Applied Mathematics

DECLARATION

I declare that this research report is my own, unaided work. It is being submitted for th degree of **Master of Science** in the University of the Witwatersrand, Johannesburg. It has not being submitted before for any degree or examination in any other university except for references which have been acknowledged.

(Signature of candidate)

3RD day of DECEMBER 1998

Abstract

This work is concerned with the Inverse Eigenvalue Problem for ordinary differential equations of the Stürm-Liouville type in the general form

$$-\frac{d}{dx}\left(r(x)\frac{du(\lambda,x)}{dx}\right)+\{q(x)-\lambda p(x)\}\,u(\lambda,x)=0,$$

$$a\leq x\leq b.$$

The central problem considered in this research is the approximate reconstruction of the unknown coefficient function $q(x)$ in the Stürm-Liouville equation from a given finite spectral data set $\hat{\lambda}_i(\hat{q})$, for $i=1:n$. A solution is sought using a finite element discretization method. The method works by solving the non-linear system arising out of the difference between the eigenvalues $\lambda_i(q)$ of the Stürm-Liouville differential equation and the given spectral data $\hat{\lambda}_i(\hat{q})$. Numerical results are presented to illustrate the effectiveness of the discretization method in question.

DEDICATION

To My Dear Wife

Nana Ama Boamah

With All My Heart.

ACKNOWLEDGEMENTS

I honour the almighty God for this research report would not exist in its present form without His grace. My deepest gratitude is to my supervisor, Professor C J Wright who suggested the topic for this research report. His critical appraisal of all aspect of the contents of this research report, at all stages of its development has tremendously contributed to the realization of the contents of this research report.

My heartfelt gratitude to my mother Miss Gladys Owusu Darkwaa and my parents-in-law, Mr and Mrs Badu Adumatta for their love, support and encouragement.

I agree to the saying that, "behind every successful man is a woman ". On this note, I render my special thanks to my dear wife Georgina Boamah for her never failing patience, moral support and encouragement- and not to mention the hours spent in the computer laboratory with me whilst working on this research report. Nana, you will always remain precious to me.

I am also thankful to Dr. Alex Appiah Jameson who financially made it possible for me to come to South Africa for further studies, Lecturers in the

Department, our computer technician S'bu, our secretaries Alice, Emily, Wendy and friends who in diverse ways assisted me to successfully complete this research report. God bless you all.

Last but not the least, I acknowledge with gratitude the financial support provided by the Postgraduate Merit Award scheme administered by the University of the Witwatersrand and of Anglo American and De Beers Bursary.

E K Boamah.

Contents

Dedication ii

Acknowledgements iii

1 Background and Introduction **1**

2 Numerical formulation of the Inverse Problem. **7**

2.1 Finite Element Discretization Method. 8

2.2 Derivation of Element Equations 9

3 Numerical methods for solving the finite IEP **16**

3.1 Numerical Examples, Results and Comments 20

3.1.1 Sensitivity of data to random noise 49

4 Concluding Remarks **50**

5 Appendix **52**

Chapter 1

Background and Introduction

Inverse Eigenvalue Problems (IEP) occur in many real life situations. Several important motivating applications arise in the physical and social sciences such as nuclear and molecular spectroscopy, structural analysis, vibrating strings, graph partitioning, the educational testing problem, the problem of communality which arises in factor analysis, control design, system identification, seismic tomography, principal component analysis, exploration and remote sensing, antenna array processing, geophysics, particle physics, circuit theory, mechanical system simulation and several others lead to closely related variations of the model of IEP.

A significant common phenomena in all these applications is that the

physical parameters of a certain system are to be reconstructed from the knowledge of its dynamical behaviour, in particular its natural frequencies and (or) normal modes.

The classical Stürm-Liouville Problem (SLP) consists of a linear second-order ordinary differential equation written in the formally self-adjoint form

$$-\frac{d}{dx}\left(r(x)\frac{du(\lambda,x)}{dx}\right)+\{q(x)-\lambda p(x)\}\,u(\lambda,x)=0, \qquad (1.1)$$

defined over an interval $a \leq x \leq b$ with appropriate boundary conditions (BCs) at a and b. Since about 1800, the study of the eigenvalues and eigenfunctions of SLPs arising in mathematical physics has given rise to many deep results of classical and modern analysis. An eigenvalue is a value of λ for which (1.1) has a nontrivial solution subject to the BCs, and the solution, unique up to scalar multiples, is the associated eigenfunction. The regular theory which dates back to Stürm (1809-1882) and Liouville(1803-1835) assumes that the coefficient functions are well-behaved. Consider $r(x)$, $q(x)$, $p(x)$ to be piecewise continuous with p strictly >0 on a bounded closed interval [a,b] and that regular boundary conditions are imposed, namely

$$l_1u'(\lambda,a)+l_2u'(\lambda,b)+l_3u(\lambda,a)+l_4u(\lambda,b)=0, \qquad (1.2)$$

$$h_1 u'(\lambda, a) + h_4 u'(\lambda, b) + h_2 u(\lambda, a) + h_3 u(\lambda, b) = 0. \qquad (1.3)$$

The inverse nature of the problem requires that one (or more) of the coefficient functions in the differential equation be computed from a knowledge of its spectral data. Classical theory insists that two full sets of eigenvalues, each derived from a different set of boundary conditions, are necessary in order to reconstruct an exact analytic coefficient function. Frequently, in real-life situations, this amount of data is simply not available. A second real difficulty is that the finite data available is also noisy.

The goal of this Research Report is to attempt to approximately reconstruct the coefficient function $q(x)$ in the regular Stürm-Liouville problem (1.1) with the boundary conditions (1.2) and (1.3) and given a finite number of exact and noisy eigenvalues.

An early important result in this direction which gave vital impetus for the further development of inverse problem theory, was obtained in 1929 by Ambartsumyan ([1]). He proved that if we denote by $\lambda_0 < \lambda_1 < \lambda_2 < ....$ the eigenvalues of the Stürm-Liouville problem

$$\begin{aligned} -u'' + q(x)u &= \lambda u, \quad (0 \leq x \leq \pi) \qquad (1.4) \\ u(0) &= u(\pi) \end{aligned}$$

where $q(x)$ is a real continuous function, and if $\lambda_n = n^2$ $(n = 0, 1, 2, ...)$ then $q(x) \equiv 0$.

The Swedish mathematician Börg ([4]) was the first to indicate the importance of Ambartsumyan's result. He was the first to carry out a systematic investigation of an important inverse problem, namely, the inverse problem for the classical Stürm-Liouville equation of the form (1.4) from given spectra. Börg showed that, in general, one spectrum does not determine a Stürm-Liouville equation, so that the result of Ambartsumyan is an exception to the general rule. In the same paper, Börg showed that two full spectra of a Stürm-Liouville operator for different boundary conditions determine it uniquely. In this paper Börg proved the following:

Theorem 1 *Let*

$$\{\lambda_i\}_{i=0}^{\infty} \text{ and } \{\mu_j\}_{j=0}^{\infty}, \tag{1.5}$$

where $\lambda_0 < \lambda_1 < \lambda_2 < \cdots$ *and* $\mu_0 < \mu_1 < \mu_2 < \cdots$ *be two sets of eigenvalues of* (1.4) *with boundary conditions* (1.2) *where* $\{l_m\}_{m=1}^4$ *and* $\{h_n\}_{n=1}^4$ *are real numbers; then, the infinite sequences* (1.5) *determine the function* $q(x)$ *and numbers* $\{l_m\}_{m=1}^4$ *and* $\{h_n\}_{n=1}^4$ *uniquely.*

Immediately after Börg's paper had been published, important research on inverse problem theory was carried out by Levinson ([14]). His 1949 paper

contains proofs of some of the results obtained by Börg. In the same paper, he proved that in the absence of negative eigenvalues (bound states), a scattering phase given for any positive energy and any fixed angular momentum determines the potential uniquely.

Since Levinson's work, a number of important research papers on the Inverse Eigenvalue Problem have been published. These include Yitshak ([24]) who worked on the Inverse Eigenvalue Problem for a vibrating system. In this paper, he denoted the first n eigenvalues of a simply connected mass-spring system by $\{\lambda_i\}_{i=1}^{n}$. He then supposed that a simple oscillator of mass m and stiffness k is attached to the system and denoted the eigenvalues of this system by $\{\mu_i\}_{i=1}^{n}$. The central problem of his work was the construction of the physical elements of the system from $\{\lambda_i\}_{i=1}^{n}$, $\{\mu_i\}_{i=1}^{n}$, m and k. Also, classical approaches to determining the solvability of IEPs involve techniques developed from algebraic theory and geometry. A list of some of the related papers can be found in [[2], [7], [9], [10], [11], [12], [15], [17], [23]].

Research on development and implementation of numerical algorithms for IEPs has also received some attention. Published papers on numerical algorithms using finite-difference schemes can be found in [[8], [20], [18], [19]]. In the paper by Richard H Fabiano, Roger Knobel and Bruce Lowe, ([20]) a cen-

tred finite-difference scheme was used to reduce the inverse Stürm-Liouville problem with Dirichlet eigenvalues to a matrix inverse eigenvalue problem. In 1983 a paper was published by Wang and Garbow, ([22]), on a method using Newton iteration and the least squares techniques to solve inverse real symmetric eigenvalue problems. What has been said so far is a summary of some previous work done in the area in question.

What follows in subsequent chapters is a description of the research work we have pursued. It is a Computational M.Sc and therefore, we will not concentrate on stating and proving theorems but rather on developing an algorithm and computer programme for numerical and graphical representation of results.

The next chapter describes the Stürm-Liouville system and the mathematical and physical description of the problem. We will describe in chapter 3 the numerical formulation , analysis of the inverse problem and test our ideas by considering numerical examples. In the concluding chapter, we will discuss our results and give hints of future possible research work.

Chapter 2

Numerical formulation of the Inverse Problem.

We discuss in this chapter, the formulation of the inverse problem. There are at least two inverse problems which could be considered for (1.1). The first, we call the infinite inverse problem which consists of determining the function $q(x)$ given enough spectral data together with an asymptotic formula for the computation of an infinite number of eigenvalues. The second is the finite inverse problem.

In this research we shall assume that only a finite amount of spectral data has been collected. To solve such a finite inverse problem, we hope to produce

a function $q(x)$ which has approximately the correct spectral behaviour.

2.1 Finite Element Discretization Method.

A solution to the Stürm-Liouville equation is sought using the Galerkin finite-element method and a trial solution based on linear elements. The method works by discretizing the problem. Here we consider a finite dimensional subspace consisting of continuous piecewise linear functions. For this purpose, we let $a = x_1 < x_2 < \cdots < x_n = b$ be a partition of the interval $[a, b]$ into subintervals $I_i = [x_i, x_{i+1}]$ of length $h_i = x_{i+1} - x_i, \quad i = 1 : n-1$. The basis of the finite element method is to select a set of linearly independent functions $N_i(x), \quad i = 1 : n$, and then to look for an approximate solution to the differential equation of the form

$$u(x) \simeq \hat{u}(x) = \sum_{j=1}^{n} u_j N_j(x). \tag{2.1}$$

The problem now is to decide which basis functions to use and then how to determine the coefficients $u_j, \quad j = 1 : n$, in such a way as to obtain $\hat{u}(x)$ as the best approximation available.

2.2 Derivation of Element Equations

The Galerkin weighted residual method forms algebraic equations by evaluating the inner product $(R,\ N_i) = 0,\quad i = 1 : n$, that is

$$\int_a^b RN_i dx = 0, \tag{2.2}$$

where

$$R(x;u) = -\frac{d}{dx}\left(r(x)\frac{d\hat{u}}{dx}(\lambda,x)\right) + \{q(x) - \lambda p(x)\}\,\hat{u}(\lambda,x),$$

is a non-zero function called the residual of the equation. By substitution, (2.2) will now become,

$$\int_a^b \left[-\frac{d}{dx}\left(r(x)\frac{d\hat{u}}{dx}(\lambda,x)\right) + q(x)\hat{u}(\lambda,x) - \lambda p(x)\hat{u}(\lambda,x)\right] N_i dx = 0. \tag{2.3}$$

We will now consider the weak form of the governing equation (2.3) by integrating the first term by parts once and rewriting as

$$\int_a^b \frac{dN_i}{dx} r(x)\frac{d\hat{u}}{dx}dx + \int_a^b N_i q(x)\hat{u}dx - \lambda\int_a^b N_i p(x)\hat{u}dx =$$
$$-\left[\left(r(x)\frac{d\hat{u}}{dx}\right)N_i\right]_{x_1}^{x_n},\quad i = 1 : n. \tag{2.4}$$

If we let

$$\left[-\left(r(x)\frac{d\hat{u}}{dx}\right)N_i\right]_{x_1}^{x_n} = [\hat{\tau}N_i]_{x_1}^{x_n}\,, \tag{2.5}$$

then the boundary term contains the flux $\hat{\tau}$, but for the eigenproblem this term must vanish from the system of equations. The boundary term occurs in two different ways in the system of equations, that is $\hat{\tau}N_i$ at each node on the boundary of the domain, and as the difference of two such expressions at each node on inter-element boundaries. In the first case, the boundary conditions require that the terms vanish at the domain boundary nodes. Thus, imposing $\hat{\tau}(x) = 0$ and $\hat{u}(x) = 0$ clearly eliminates the term and the entire equation in which the term appears respectively. In the second case, a nonzero difference in flux at an inter-element boundary represents an applied concentrated load, however the eigenproblem does not permit applied loads.

Now, since the boundary term must vanish from the system of equations, we will ignore it by eliminating it from the weak form of the element equations. By summing (2.2) over the number of elements we will obtain

$$\sum_{e=1}^{n-1}\int^{(e)} RN_i dx = 0, \quad i = 1:n, \text{ which implies}$$

$$\sum_{e=1}^{n-1}\int^{(e)}\frac{dN_i}{dx}r(x)\frac{d\hat{u}}{dx}dx+\sum_{e=1}^{n-1}\int^{(e)}N_iq(x)\hat{u}dx-\lambda\sum_{e=1}^{n-1}\int^{(e)}N_ip(x)\hat{u}dx=0, \tag{2.6}$$

where e denotes an element. Substituting the general form of the element trial solution into the weighted residual equation in (2.6), we obtain

$$\sum_{e=1}^{n-1}\sum_{j=1}^{n-1}\left[\int^{(e)}\frac{dN_i}{dx}r(x)\frac{dN_j}{dx}dx\right]u_j+\sum_{e=1}^{n-1}\sum_{j=1}^{n-1}\left[\int^{(e)}N_iq(x)N_jdx\right]u_j-$$

$$\lambda\sum_{e=1}^{n-1}\sum_{j=1}^{n-1}\left[\int^{(e)}N_ip(x)N_jdx\right]u_j=0. \tag{2.7}$$

Equation (2.7) is written in a conventional matrix eigenvalue problem as

$$[A]\,u=\lambda[B]\,u, \tag{2.8}$$

where (λ_j,u_j) is an eigenpair and $[A]=[Kr]+[Kq]$. For a typical element $[x_i,x_{i+1}]$, we will denote

$$\begin{aligned} Kr_{ij} &= \int_{x_i}^{x_{i+1}}\frac{dN_i}{dx}r(x)\frac{dN_j}{dx}dx\,, \\ Kq_{ij} &= \int_{x_i}^{x_{i+1}}N_iq(x)N_jdx, \\ B_{ij} &= \int_{x_i}^{x_{i+1}}N_ip(x)N_jdx\,,\quad i=1:n-1,\ j=1:n. \end{aligned} \tag{2.9}$$

The matrices Kr, Kq and B formed are all tridiagonal since $Kr_{ij}=0$, $Kq_{ij}=0$, $B_{ij}=0$ for all $\mid i-j\mid>1$. We choose the linear basis

functions $\{N_i\}_{i=1}^{n-1}$ as

$$N_i(x) = \begin{cases} \frac{x-x_{i-1}}{h_{i-1}}, & x_{i-1} \le x \le x_i \\ \frac{x_{i+1}-x}{h_i}, & x_i \le x \le x_{i+1} \\ 0, & \text{otherwise} \end{cases} \tag{2.10}$$

where $h = x_{i+1} - x_i$.

The only nonzero global trial functions on element e are N_i and N_{i+1}, and so $N_k = 0$ on element e if node k does not belong to that element. If for simplicity we assume that $r(x)$ and $p(x)$ are constants then we can generally write for the element e, $A_{kl}^{(e)} = 0, \quad l, k \neq i+1$,

$$Kr_{ii}^{(e)} = r\int_{x_i}^{x_{i+1}} (\frac{dN_i}{dx})^2 dx = \frac{r}{h},$$

$$Kr_{ij}^{(e)} = Kr_{ji}^{(e)} = r\int_{x_i}^{x_{i+1}} \frac{dN_i(x)}{dx}\frac{dN_j(x)}{dx}dx, \quad j = 1:n,$$

$$= -\frac{r}{h}.$$

With the components of the element matrix $Kr^{(e)}$ completely defined in this manner and summing over all $n-1$ elements, the entries of the matrix Kr are given by

$$Kr = \frac{1}{h}\begin{pmatrix} 1 & -1 & & & & \\ -1 & 2 & -1 & & & \\ & -1 & 2 & -1 & & \\ & & -1 & . & . & \\ & & & . & . & . \\ & & & & . & 1 \end{pmatrix}.$$

Similarly, the matrix B is given by

$$B = \frac{h}{6}\begin{pmatrix} 2 & 1 & & & & \\ 1 & 4 & 1 & & & \\ & 1 & 4 & . & & \\ & & . & . & . & \\ & & & . & . & 1 \\ & & & & 1 & 2 \end{pmatrix}.$$

Now assume $q(x)$ in the form $\hat{q}(x) = \sum_{l=1}^{n} q_l N_l$, where N_l and q_l are piecewise linear basis functions and constants to be determined respectively. The entries of the matrix Kq are then

$$Kq_{ii} = \int_{x_i}^{x_{i+1}} N_i^2 \left(q_i N_i + q_{i+1} N_{i+1}\right) dx\,, \tag{2.11}$$

$$Kq_{i,i+1} = Kq_{i+1,i} = \int_{x_i}^{x_{i+1}} N_i N_{i+1} \left(q_i N_i + q_{i+1} N_{i+1}\right) dx\,, \tag{2.12}$$

$$Kq_{i+1,i+1} = \int_{x_i}^{x_{i+1}} N_{i+1}^2 \left(q_i N_i + q_{i+1} N_{i+1}\right) dx\,, \quad i = 1 : n-1. \tag{2.13}$$

Evaluate the integrals in equations (2.11), (2.12) and (2.13)to obtain

$$Kq_{i,i+1} = Kq_{i+1,i} = \frac{h}{12}(q_i + q_{i+1}),$$

$$Kq_{ii} = \frac{h}{12}(3q_i + q_{i+1}),$$

$$Kq_{i+1,i+1} = \frac{h}{12}(q_i + 3q_{i+1}), \quad i = 1 : n-1.$$

In an expanded matrix form, we obtain the entries of Kq as

$$Kq = \frac{h}{12}\begin{pmatrix} 3q_1+q_2 & q_1+q_2 & & & & \\ q_1+q_2 & 4(q_2+q_3) & q_2+q_3 & & & \\ & q_2+q_3 & 4(q_3+q_4) & . & & \\ & & . & ..... & . & \\ & & & . & 4(q_{n-2}+q_{n-1}) & q_{n-2}+q_{n-1} \\ & & & & q_{n-2}+q_{n-1} & q_{n-1}+3q_n \end{pmatrix}.$$

We can now write

$$(Kr + Kq)u = \lambda B u \tag{2.14}$$

where $u = [u_1, u_2, ..., u_n]^T$ and $\lambda = [\hat{\lambda}_1, \hat{\lambda}_2, ..., \hat{\lambda}_n]$.

The problem may now be stated as follows: Given Kr and the spectrum $\{\hat{\lambda}_i\}_{i=1}^n$, we must approximately reconstruct the potential $q(x)$ such that $Kr+Kq$ has the given spectrum. This is known as the additive inverse eigenvalue problem (AIEP) which may be written in the form

$$A(q) = Kr + Kq. \tag{2.15}$$

As indicated AIEP means that a given matrix Kr is perturbed by the addition of a specially structured matrix Kq in order to match the eigenvalues.

In conclusion, we have discussed numerical formulation of the IEP by the use of a finite element discretization method and have now obtained the discretized Stürm-Liouville differential equation in n by n matrix forms.

Chapter 3

Numerical methods for solving the finite IEP

We now discuss in this chapter, a numerical method for solving the finite inverse problem. Given distinct eigenvalues $\left\{\hat{\lambda}_i\right\}_{i=1}^{n}$, the method works by solving approximately the nonlinear system arising out of the difference between the computed eigenvalues $\{\lambda_i(q)\}_{i=1}^{n}$ and the given spectral data $\left\{\hat{\lambda}_i\right\}_{i=1}^{n}$. The general system of equations to be solved is

$$f(q) = \begin{bmatrix} \lambda_1(q) - \hat{\lambda}_1 \\ . \quad . \quad . \\ \lambda_n(q) - \hat{\lambda}_n \end{bmatrix} = 0. \tag{3.1}$$

The overall algorithm is outlined as follows.

(i) Choose an initial $q(x)$.

(ii) Use a well known documented routine to calculate n exact eigenvalues, $\{\hat{\lambda}_i\}_{i=1}^n$.

(iii) (a) Use these exact eigenvalues to reconstruct $q(x)$.

OR (b) Use perturbed $\{\hat{\lambda}_i\}_{i=1}^n$ to reconstruct $q(x)$.

METHOD

We now solve the non-linear system in (3.1) by applying Newton's method. This method gives a very efficient means of converging to a root, if a sufficiently good initial guess is used. Newton's method is given by the following. For some initial vector q, if $f(q) = 0$ where f denotes the column vector of n components $(f_1, f_2, ..., f_n)^T$ and q is a column vector of n components $(q_1, q_2, ..., q_n)^T$, if $f(q)$ is continuously differentiable then,

$$\frac{\partial f_i}{\partial q_j} = J(q), \quad i = 1:n, \quad j = 2:n-1$$

where J is the Jacobian matrix and the range in j is due to boundary conditions imposed on q_j. Newton's method is

$$J(q^k)\Delta q = -f(q^k), \tag{3.2}$$

$$\Delta q = q^{k+1} - q^k, \quad k = 1:n.$$

In each step of Newton's method we have to solve a system of linear equations to obtain Δq^k. Now, from equation (2.14)

$$\lambda = \frac{u(q)^T(Kr + Kq)u(q)}{u(q)^T B u(q)}. \tag{3.3}$$

Differentiating (3.3) with respect to q_j, we obtain

$$\frac{\partial \lambda_i(q)}{\partial q_j} = \frac{u_i(q)^T K q_j u_i(q)}{u_i(q)^T B u_i(q)}, \quad i = 1:n, \quad j = 2:n-1,$$

where Kq_j is the partial derivative with respect to q_j of the system Kq. Thus, the components of the Jacobian of f is

$$J_{ij}(q) = \frac{u_i(q)^T K q_j u_i(q)}{u_i(q)^T B u_i(q)}, \tag{3.4}$$

and one step of Newton's method is defined by (3.2).

The system in (3.2) to be solved is over-determined and this is due to the fact that the first and last elements on the leading diagonal of the system Kq were evaluated and kept fixed and as a result, the number of columns of the system were reduced by 2 whilst that of rows were not changed. Solving this over-determined system does not pose problems to MATLAB, a programming language used in this research report because it uses the least-square method to solve the over-determined system. MATLAB solves the system in question

directly by simply using the operator "\" as

$$\Delta q = -J(q^k)\backslash \begin{pmatrix} \lambda_1(q) & - & \hat{\lambda}_1 \\ . & . & . \\ \lambda_n(q) & - & \hat{\lambda}_n \end{pmatrix}.$$

Now, the method for solving (3.1) is as follows:

ALGORITHM

Choose a starting value q^0.

Steps

for $k = 1$:until convergence

1. Construct $Kr + Kq$.

2. Find the eigenvalues and eigenvectors of $Kr + Kq$.

3. Stop if

$$\|\lambda_i(q^k) - \hat{\lambda}_i\|, \quad i = 1 : n \quad \text{is sufficiently small,}$$

4. Form $J(q^k)$ by

$$J_{ij} = \frac{u_i(q)^T K q_j u_i(q)}{u_i(q)^T B u_i(q)}.$$

5. Compute q^{k+1} by solving the equation $J(q^k)(q^{k+1} - q^k) = -f(q^k)$.

6. Form $Kr + Kq$.

7. Find the eigenvalues $\left\{\lambda(q^{k+1})\right\}$ and the corresponding eigenvectors

$$\left\{u(q^{k+1})\right\} \ of \ Kr + Kq.$$

8. Repeat the process starting from (1) until (3) is satisfied.

End of algorithm.

3.1 Numerical Examples, Results and Comments

In this section, we test our ideas by presenting numerical solutions to a few examples of the Inverse Stürm-Liouville problem. The spectral data in each test case for a given $q(x)$ was computed using the well known documented routine SLEIGN2 (SLN2) which is described in ([6]). SLN2 is an interactive and easy to use package. Throughout tests conducted, SLN2 performed in a generally accurate and reliable manner ([16]). Its ability to solve problems with exceptional robustness and very little user input makes it a first-class code. Its robustness is far in excess of that of any other code which has been used to test Stürm Liouville computations ([16]). The SLN2 code can be down-loaded from www.math.niu.edu/~zettl/sl2/sleign2.f on the World Wide Web.

In choosing our examples for numerical experiments, we took into consideration the fact that we have used linear basis functions in the discretization process. The examples include constant, linear, quadratic, transcendental, function discontinuity, derivative discontinuity and oscillatory functions as choices for $q(x)$. The constant, linear and quadratic examples show the behaviour of the method on low-order polynomials whilst the $2sin(2x)$ example indicates its ability to handle simple trigonometric functions. In order to simulate real-life situations, we also considered examples having function discontinuity, derivative discontinuity and rapid oscillatory functions. The latter three are simple models of shocks, waves of different types, earthquakes, impulses in mechanical systems and so on.

Our choice of these examples was motivated by the desire to test our computational ideas on a problem in which a polynomial $q(x)$ has an order higher than the basis functions as well as on problems which approximate real-life geophysical occurences.

In all the examples below, we consider the simple SLP

$$-u'' + q(x)u = \lambda u, \qquad x \in [a, b],$$

subject to $u'(a) = u'(b) = 0$. We concentrate on derivative boundary conditions in all our examples because they automatically help in our simpli-

fications as explained in section 2.2. $\hat{q}(x)$ is the coefficient function to be reconstructed. With these numerical examples, we shall be able to point out the strengths and weaknesses of the method used. We will discuss results of solving test problems, investigate the accuracy of the method used and determine how the choice of the number of eigenvalues and initial guess $q(x)$ affect the accuracy of the coefficient function $\hat{q}(x)$.

The table of values in each of the examples shows the 2−norm of the difference between SLN2 computed ("exact") eigenvalues $\lambda_i(q)$ and the eigenvalues $\hat{\lambda}_i(\hat{q})$, $i = 1 : n$, obtained by solving the inverse problem. We denote this by $\|f\|_2 = \|\lambda(q) - \hat{\lambda}(q)\|_2$ after the corresponding number of iterations.

In all our numerical examples, we will use $q(x).*(randn(n,1)*K+1)$, where $0 < K < 1$ as the initial guess. The variable K determines the closeness of this initial guess to $q(x)$. The function $randn(n,1)$ is a built-in MATLAB function which defines n random numbers. For best results, the value of the tolerance (the difference between SLN2 computed eigenvalues and eigenvalues obtained from inverse problem) should be kept small. The tolerances used in all examples were in the range $(10^{-6}, 10^{-3})$.

Examples 1-7 below are test examples to show the reconstruction of coefficient functions $q(x)$. The number of nodal points which correspond to the

number of Neumann eigenvalues used for a good approximation are in the range of $40 - 100$. A good approximation here means that the coefficient function $q(x)$ was reconstructed with a very small relative error (see figures 1.2, 2.2, 3.2, 4.2, 5.2, 6.2). We also noticed that as we increased the number of discretized points which corresponds to the number of eigenvalues, a better approximation of the coefficient function was reconstructed until the accuracy of $q(x)$ remained the same (see figures 1.1a, 2.1a, 3.1a, 4.1a, 5.1a, 6.1a). For examples 1 through 6 the the accuracy of $q(x)$ remained the same for 60 and above SLN2 computed eigenvalues .

Example 7 was used as a test example to verify the effect of the method on the number of turning points in the coefficient function to be reconstructed. The accuracy remained the same for more than 100 SLN2 computed eigenvalues used for a coefficient function with two turning points. For more than two turning points in the coefficient functions, poor approximations were obtained and in some cases convergence was not possible for even more than 100 SLN2 computed eigenvalues used (see fig 7.1). In all our graphs "o" represents exact solution and "$-$" represents approximate solution.

Example 1

$q(x) = 10$, subject to $u'(0) = u'(1) = 0$, $x \in [0,1]$.

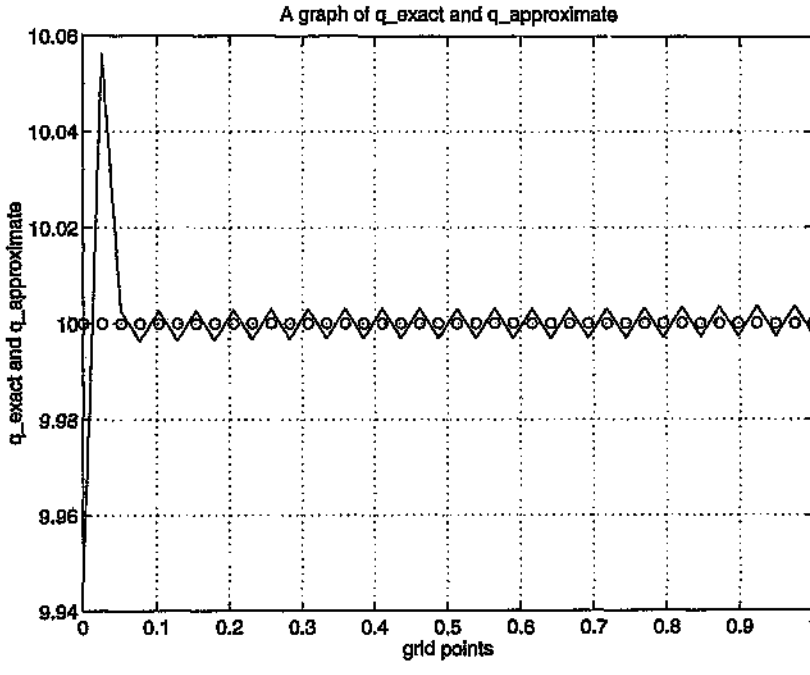

Fig 1.1a

Fig 1.1a shows the reconstruction of $q(x) = 10$ from 40 SLN2 computed eigenvalues. An increase in the number of SLN2 computed eigenvalues from

40 to 60 gives a better approximation of $q(x)$. (See figs 1.1 and 1.2 below).

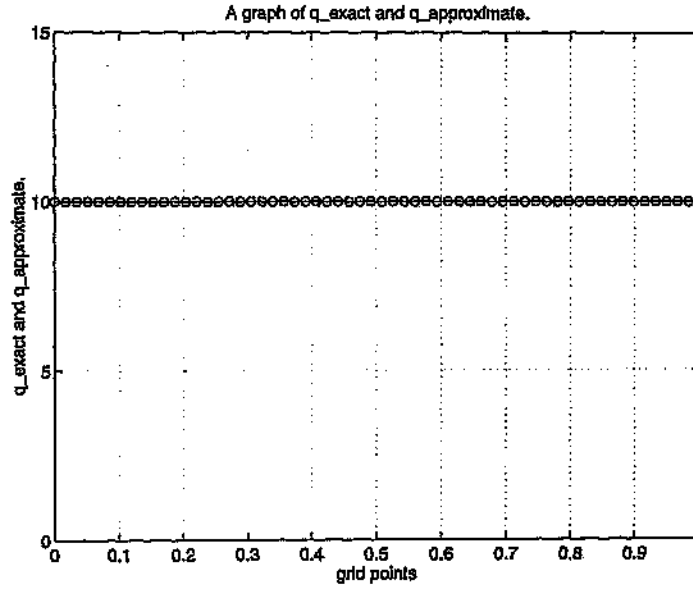

Fig 1.1b

Fig 1.1b shows the construction of $\hat{q}(x) = 10$ for $n = 60$ SLN2 computed eigenvalues as exact input data. $q(x)$ was randomly perturbed by $randn(n,1) * 0.025 + 1$ and used as initial guess. A graph of the relative error, fig 1.2 and table 1.2 showing iteration number and convergence are

respectively shown below.

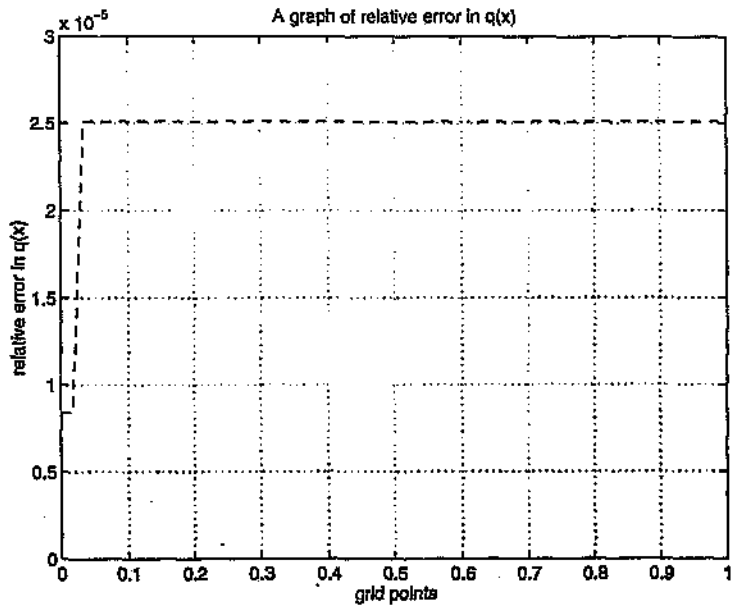

Fig 1.2

Table 1.2

Iteration No.	$\|\|f\|\|_2$	Iteration No.	$\|\|f\|\|_2$
1	1.4e+001	7	2.3e-004
2	4.9e-001	8	1.0e-004
3	1.6e-001	9	7.2e-005
4	1.9e-002	10	6.4e-005
5	7.6e-003	11	5.2e-005
6	2.2e-003	12	1.0e-005

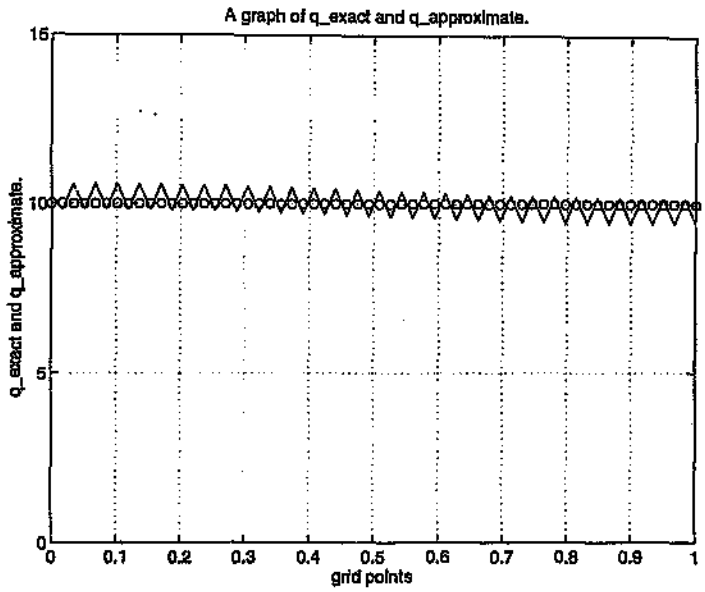

Fig 1.3

Fig 1.3 shows the graph of $\hat{q}(x)$ when $q(x)$ is perturbed by $randn(n,1) * 0.025 + 1$ and used as initial guess. A graph of the relative error, fig 1.4 and table 1.3 showing iteration number and convergence are respectively shown below:

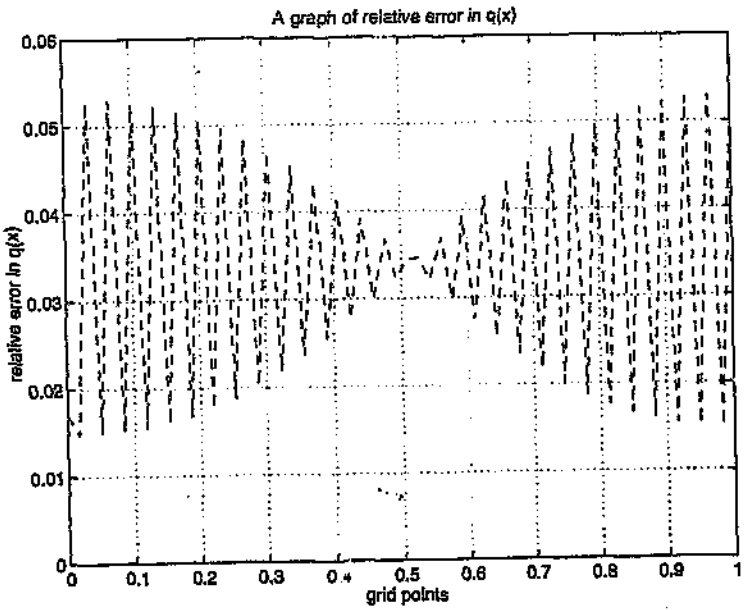

Fig 1.4

Table 1.3

Iteration No.	$\|\|f\|\|_2$	Iteration No.	$\|\|f\|\|_2$
1	1.9e+001	6	9.3e-004
2	3.8e-001	7	1.0e-004
3	7.6e-002	8	3.4e-004
4	2.4e-002	9	8.4e-005
5	8.4e-003	10	4.9e-005

Example 2

$q(x) = x$, subject to $u'(0) = u'(1) = 0$, $x \in [0, 1]$.

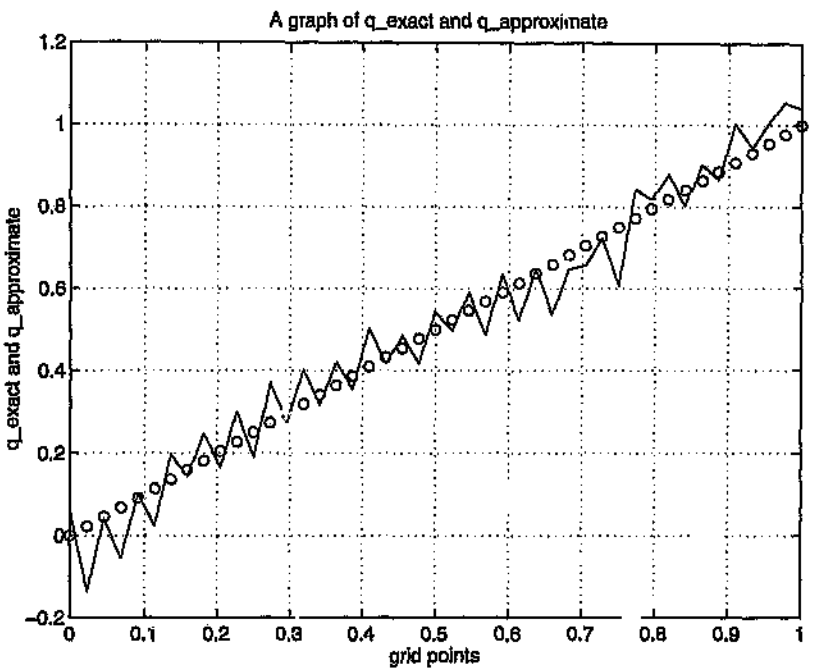

Fig 2.1a

Fig 2.1a shows the reconstruction of $q(x) = x$, using 45 SLN2 computed eigenvalues. An increase in the number of SLN2 computed eigenvalues gives better approximation of $q(x)$. (See figs 2.1b and 2.2).

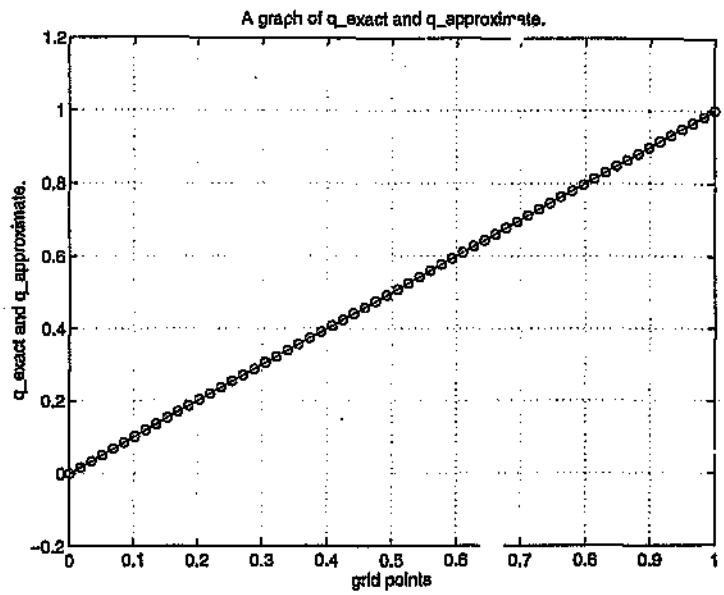

Fig 2.1b

Fig 2.1b shows the construction of $\hat{q}(x)$ for $n = 60$ SLN2 computed eigenvalues as exact input data. $q(x)$ was randomly perturbed by $randn(n,1)*0.05+1$ and used as initial guess. A graph of the relative error, fig 2.2 and table 2.1

fig 2.2 and table 2.1 showing convergence are respectively shown below:

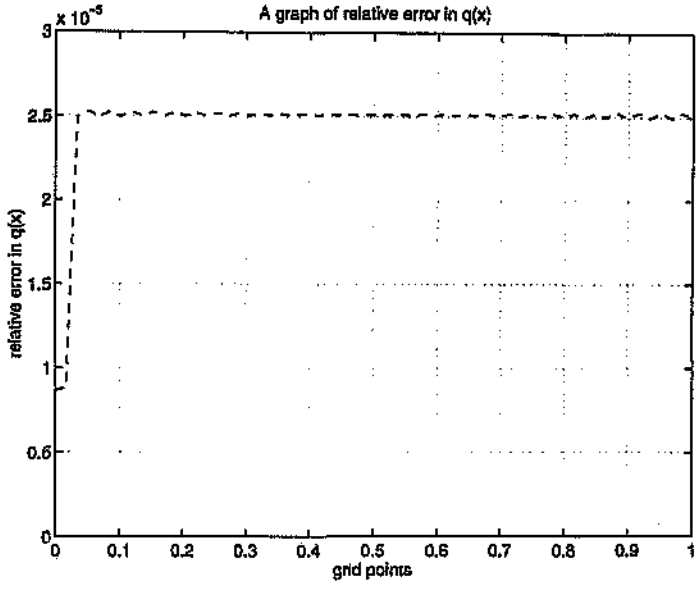

Fig 2.2

Table 2.1

Iteration No.	$\|\|f\|\|_2$	Iteration No.	$\|\|f\|\|_2$
1	1.9e+002	7	5.0e-004
2	3.9e+000	8	1.7e-004
3	7.5e-001	9	1.1e-004
4	9.8e-002	10	6.0e-005
5	1.2e-002	11	5.1e-005
6	4.2e-003	12	3.0e-005

Example 3

$q(x) = x^2$, subject to $u'(0) = u'(1) = 0$, $x \in [0, 1]$

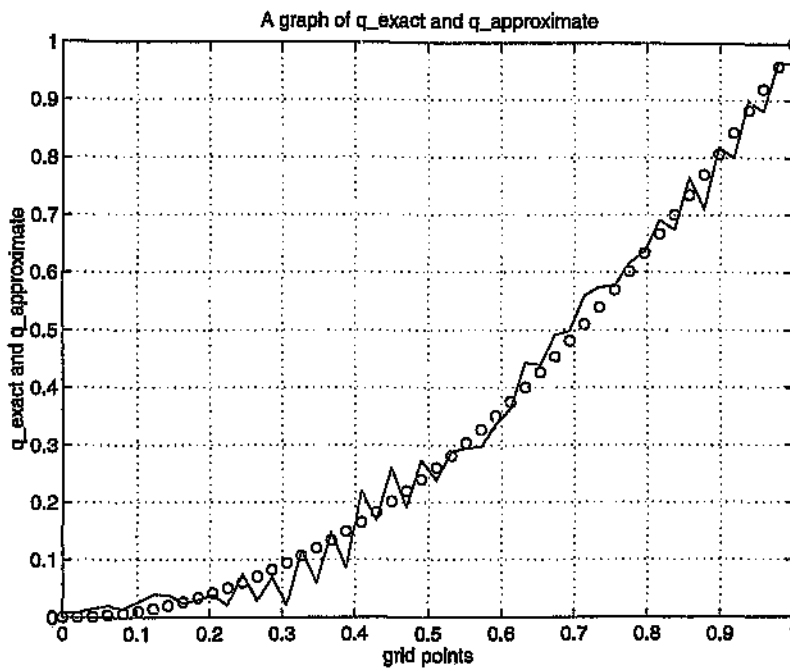

Fig 3.1a

Fig 3.1a shows the reconstruction of $q(x) = x^2$ for 50 SLN2 computed eigenvalues. An increase in the number from 50 to 60 gives a better approximation. (See figs 3.1b and 3.2).

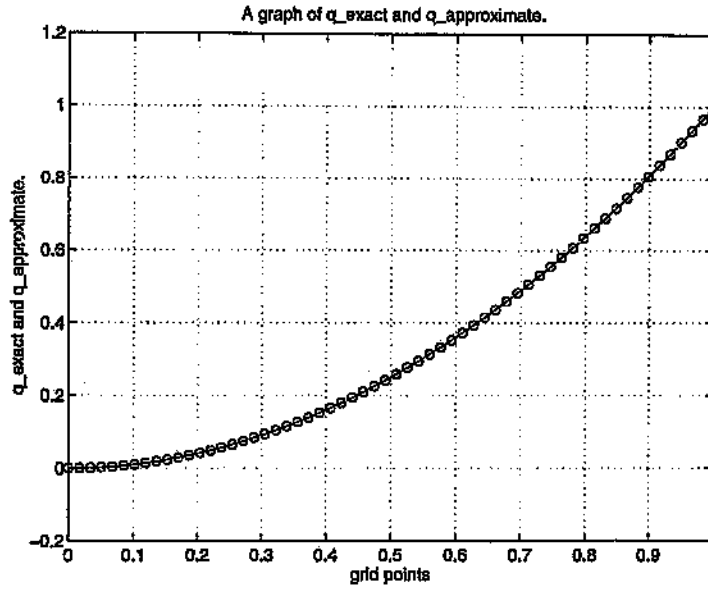

Fig 3.1b

Fig 3.1b shows the construction of $\hat{q}(x)$ from $n = 60$ SLN2 computed eigenvalues as input data . q(x) was perturbed randomly by $randn(n,1)*0.025+1$ and used as initial guess. Since $q(x)$ is symmetric, we only show the construction in the interval [0,1]. A graph of relative error, fig 3.2 and table 3.1

iteration number and convergence are respectively seen below:

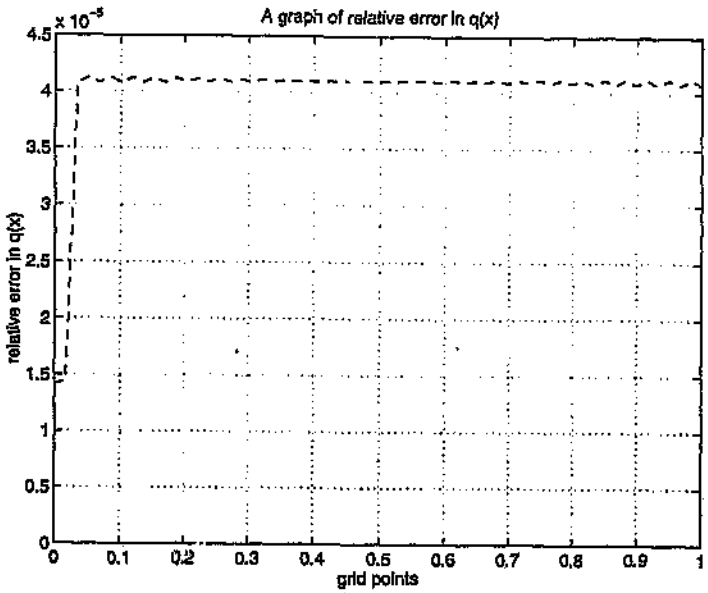

Fig 3.2

Table 3.1

Iteration No.	$\|\|f\|\|_2$	Iteration No.	$\|\|f\|\|_2$
1	6.5e+000	6	2.1e-004
2	6.8e-001	7	1.0e-004
3	2.2e-002	8	6.7e-005
4	8.5e-003	9	5.4e-005
5	2.7e-003	10	4.2e-005

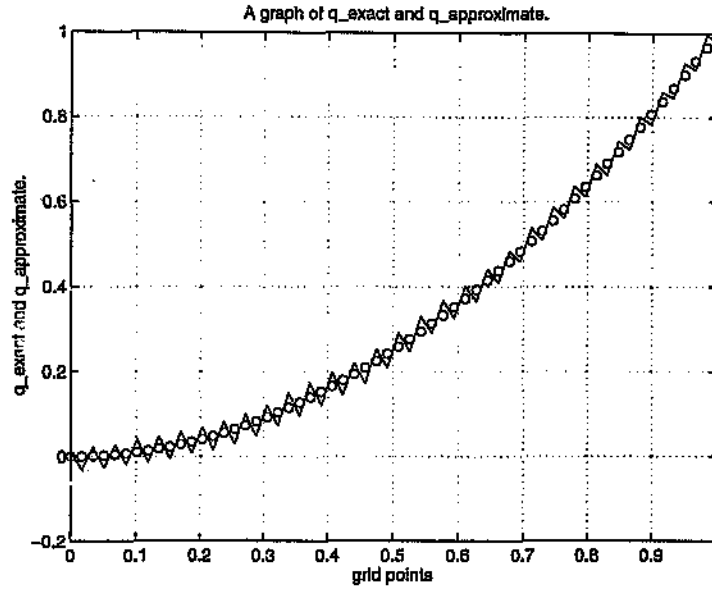

Fig 3.3

Fig 3.3 shows the reconstruction of $q(x) = x^2$ with a random noise of $(randn(n,1) * 0.05 + 1)$ in the SLN2 computed eigenvalues.

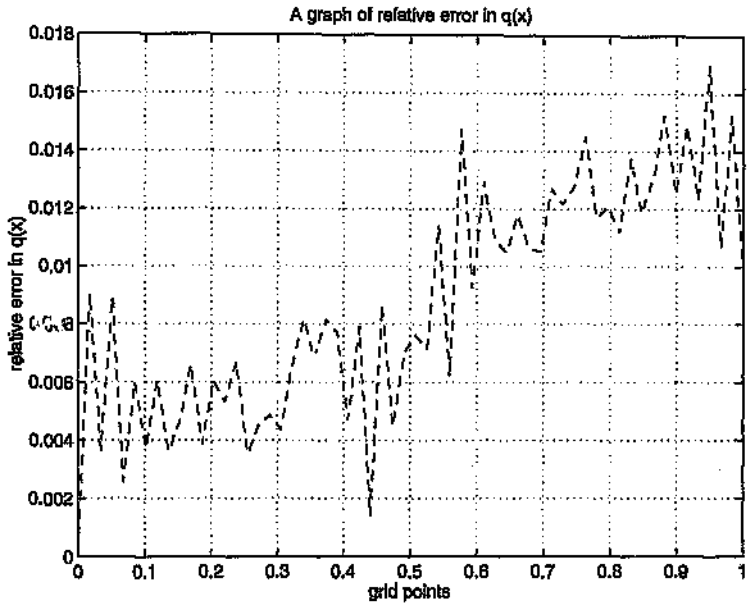
A graph of relative error in q(x)
relative error in q(x)
grid points
0.018
0.016
0.014
0.012
0,01
0.008
0.006
0.004
0.002
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Fig 3.4

Example 4

$q(x) = 2\sin(2x)$, subject to $u'(0) = u'(\frac{\pi}{2}) = 0$, $x \in [0, 1]$.

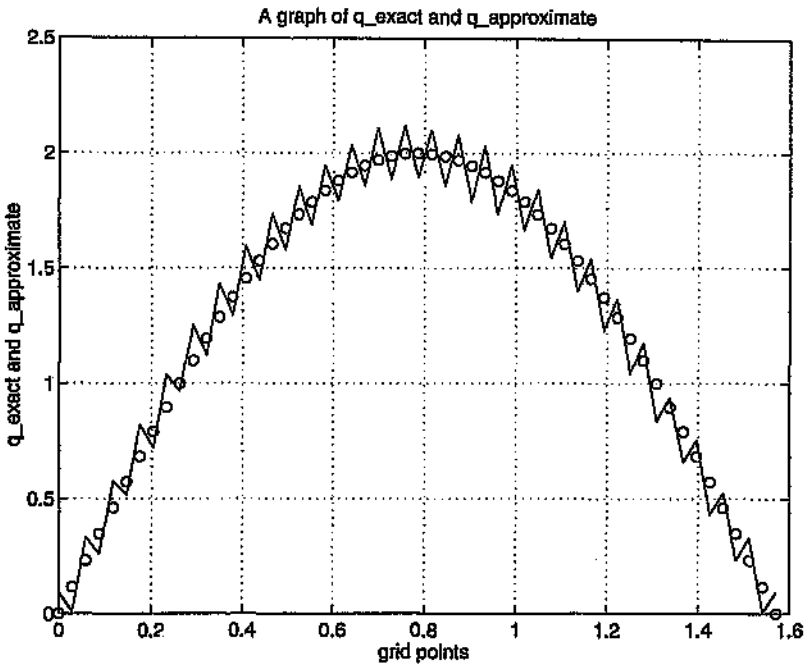

Fig 4.1a

Fig 4.1a shows the reconstruction of $q(x) = 2\sin 2x$ for 55 SLN2 computed eigenvalues. An increase from 55 to 60 computed eigenvalues gives a better approximation of $q(x)$. (See figs 4.1b and 4.2).

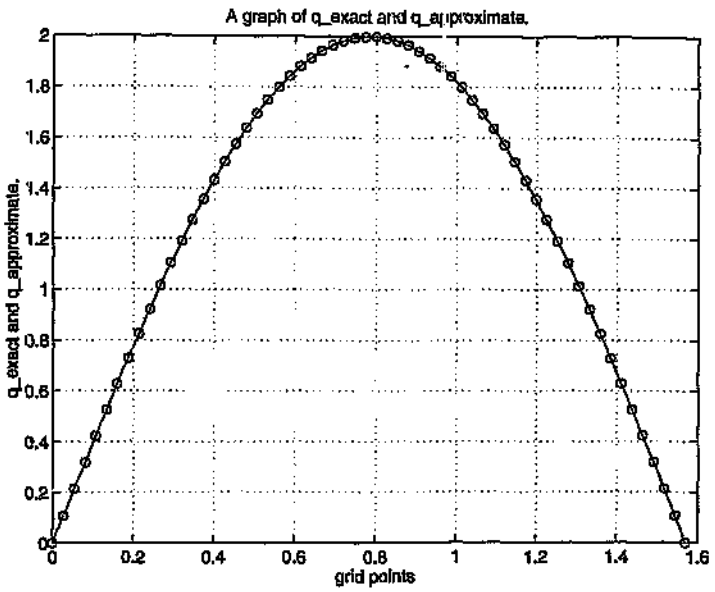

Fig 4.1b

Fig 4.1b shows the construction of $\hat{q}(x)$ for $n = 60$ SLN2 computed eigenvalues as input data. $q(x)$ was randomly perturbed by $randn(n,1)*0.05+1$ and used as initial guess . A graph of relative error, fig 4.2 and table 4.1 showing

showing iteration number and convergence are respectively shown below:

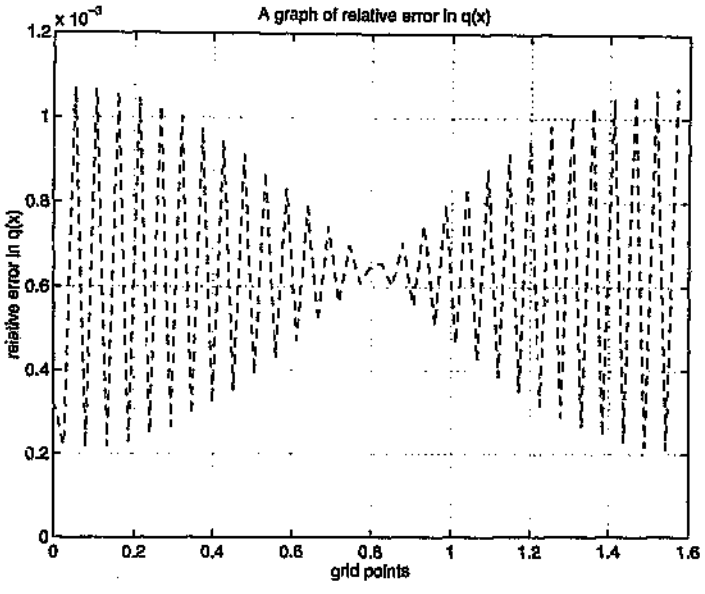

Table 4.1 **Fig4.2**

Iteration No.	$\|\|f\|\|_2$	Iteration No.	$\|\|f\|\|_2$
1	8.5e+001	7	8.0e-003
2	2.0e+000	8	2.9e-003
3	4.1e-001	9	1.1e-003
4	1.4e-001	10	4.2e-004
5	5.5e-002	11	1.5e-004
6	2.1e-002	12	4.8e-005

In fig 4.3 , we show the sensitivity of the method to noise in the computed eigenvalues. A random noise level of $randn(n,1)*0.05+1$ is applied to SLN2 computed eigenvalues. Initial guess of $q(x).*randn(n,1)*0.05+1$ was used.

Fig 4.4 shows a graph of the relative error.

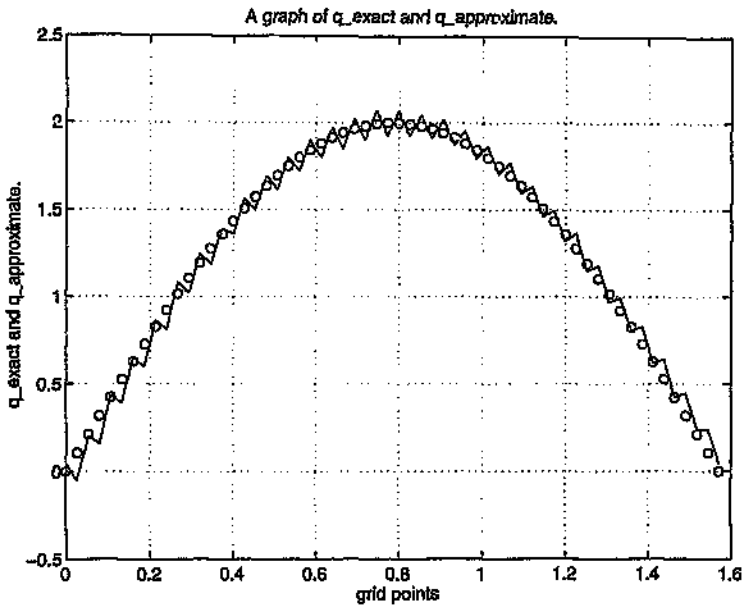

Fig 4.3

Fig 4.3 shows the reconstruction of $q(x) = 2\sin 2x$ with a random noise of $(randn(n,1) * 0.05 + 1)$ in the SLN2 computed eigenvalues.

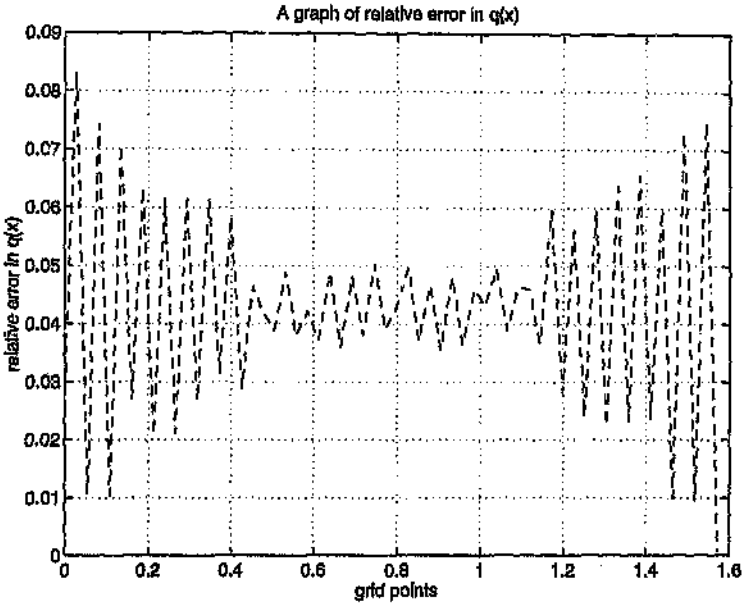

Fig 4.4

Example 5

$$q(x) = \begin{cases} 2\ , & 0 \leq x \leq 0.5 \\ 3, & 0.5 < x \leq 1 \end{cases}$$

and subject to the boundary conditions $u'(0) = u'(1) = 0$.

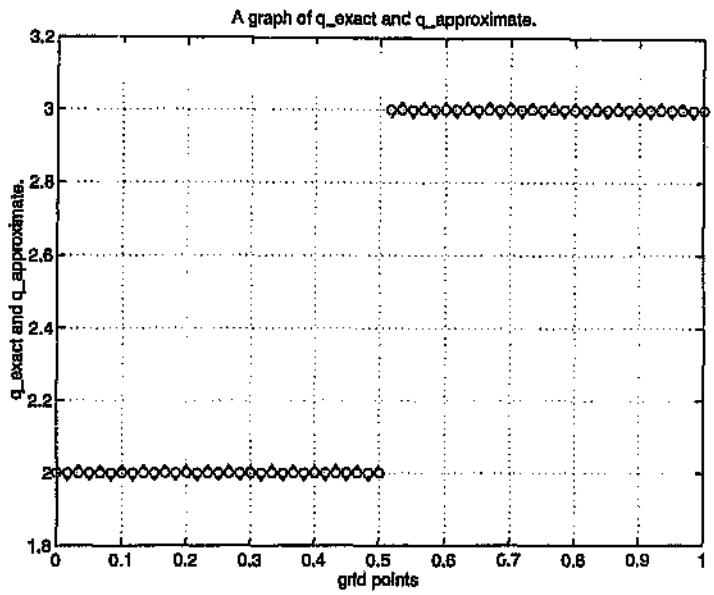

Fig 5.1.

Fig 5.1 shows the construction of $\hat{q}(x)$ for $n = 60$ SLN2 computed eigenvalues as input data. $q(x)$ was randomly perturbed by $randn(n,1) * 0.025 + 1$ and used as initial guess . A graph of relative error, fig 5.2 and table 5.1 showing

5.1 showing iteration number and convergence are respectively shown below:

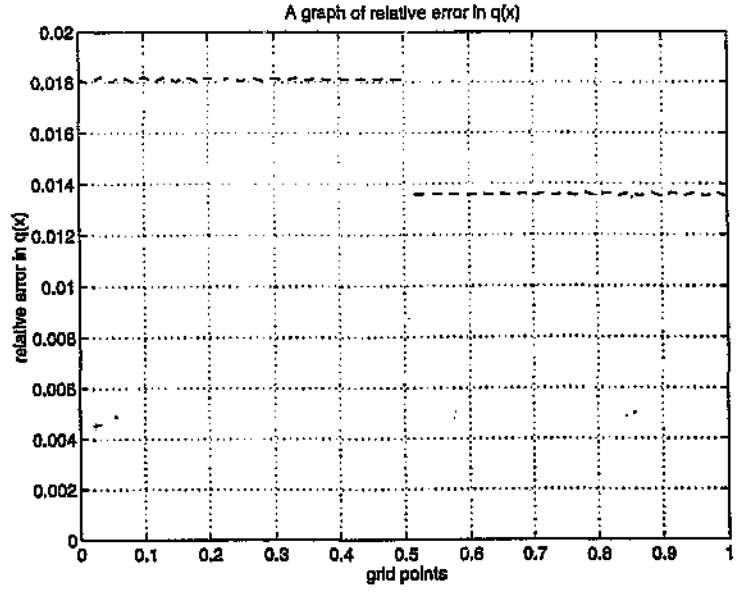

Fig 5.2

Table 5.1

Iteration No.	$\|\|f\|\|_2$	Iteration No.	$\|\|f\|\|_2$
1	1.9e+000	5	6.5e-005
2	3.7e-003	6	6.0e-005
3	1.2e-004	7	5.2e-005
4	7.5e-005	8	3.1e-005

Example 6

$$q(x) = \begin{cases} 4x^2 , & 0 \leq x \leq 0.5 \\ 4(x-1)^2, & 0.5 < x \leq 1 \end{cases}$$

and subject to the boundary conditions $u'(0) = u'(1) = 0$.

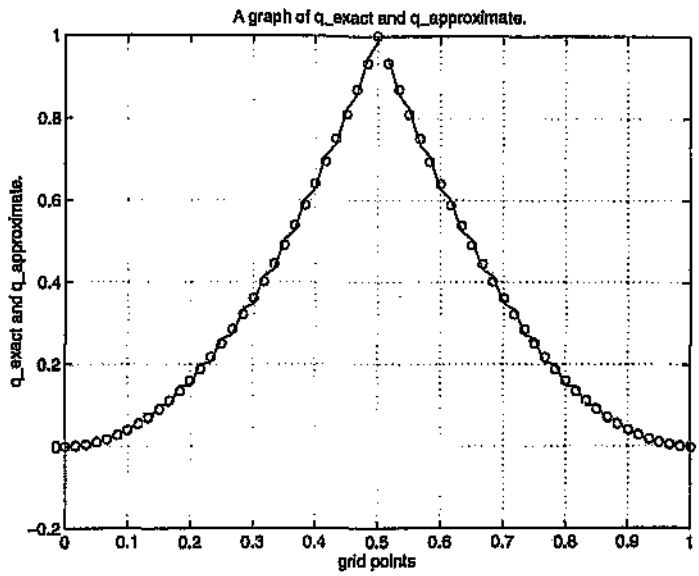

Fig 6.1

Fig 6.1 shows the construction of $\hat{q}(x)$ for $n = 60$ SLN2 computed eigenvalues as input data. $q(x)$ was randomly perturbed by $randn(n, 1) * 0.05 + 1$ and used as initial guess . A graph of relative error, fig 6.2 and table 6.1 showing

6.1 showing iteration number and convergence are respectively shown below:

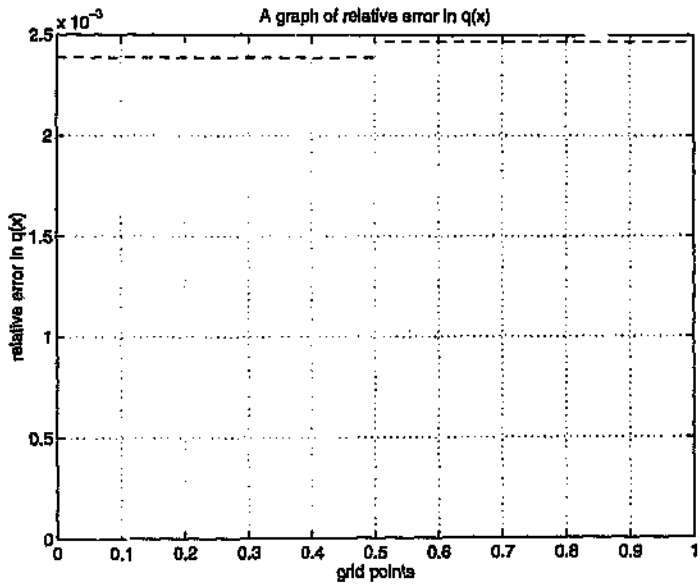

Fig 6.2

Table 6.1

Iteration No.	$\|f\|_2$	Iteration No.	$\|f\|_2$
1	6.6e-001	4	5.4e-005
2	6.2e-004	5	1.1e-005
3	1.5e-004		

The graphs in Fig 6.3 and Fig 6.4 below show the construction of $\hat{q}(x)$ and the relative error respectively when $q(x)$ was perturbed randomly by $randn(n,1) * 0.5 + 1$ and used as initial guess.

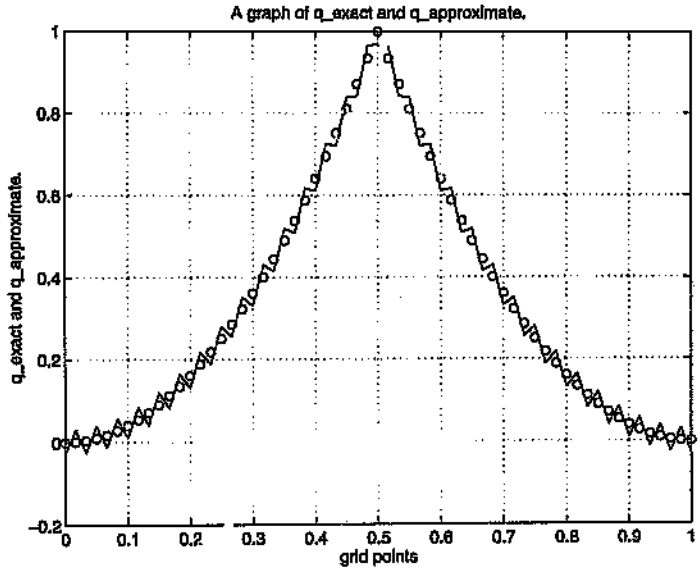
A graph of q_exact and q_approximate.
q_exact and q_approximate.
grid points
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Fig 6.3

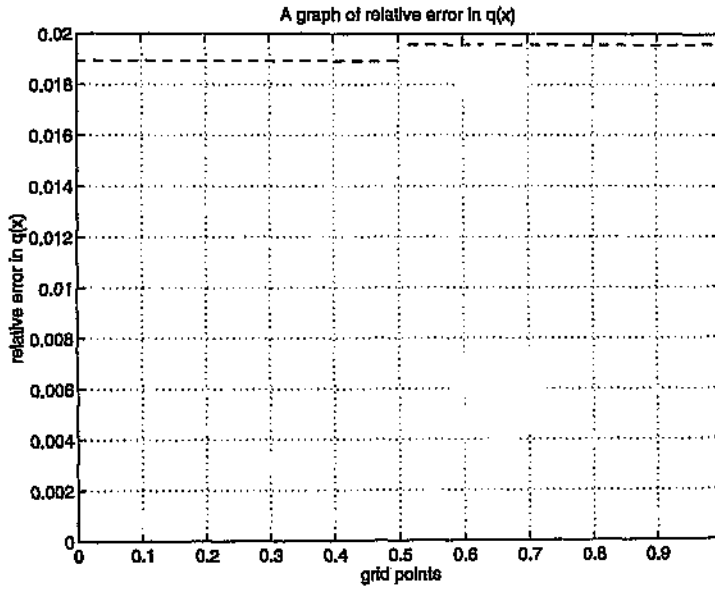
A graph of relative error in q(x)
relative error in q(x)
grid points
0.02
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
0.1
0.2
0.3
0.4
0.5
0.6
0.7
0.8
0.9
1

Fig 6.4

Example 7

$q(x) = \sin(2x)$, and subject to $u'(0) = u'(\pi) = 0$.

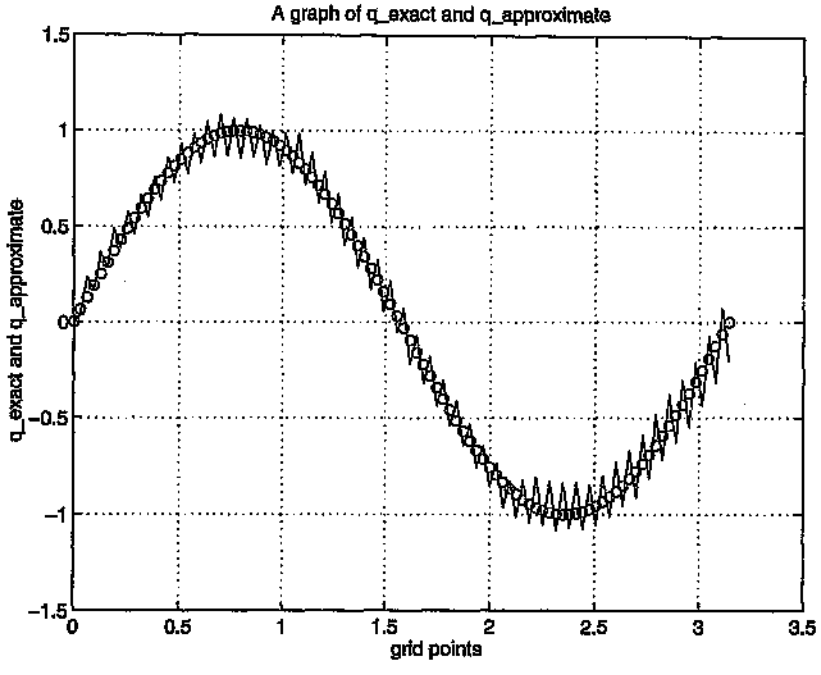

Fig 7.1

Fig 7.1 shows the construction of $\hat{q}(x)$ for $n = 100$ SLN2 computed eigenvalues as input data. $q(x)$ was randomly perturbed by $randn(n, 1) * 0.0025 + 1$ and used as initial guess . In examples with more than two turning points not reported, we attempted to reconstruct $q(x)$ without much success. (See

example 7).

3.1.1 Sensitivity of data to random noise

The sensitivity of the method to noise is investigated. In the examples shown, we calculated our spectral data using SLN2. In real-life situations, data obtained by other means (e.g experiment) may contain noise. To test our ideas for such noisy data, we investigated the sensitivity of the SLN2 computed eigenvalues to random noise of $(randn(n,1) * K + 1$, where $0 < K < 1$. The variable K in this case allowed us to determine the noise level. Curves (figs 3.3 and 4.3) obtained as a result of noise in data were not smooth but the least-square method can be used to smoothing them for better approximations.

Chapter 4

Concluding Remarks

The goal of this research report has been to use a finite element discretization method to investigate the approximate reconstruction of the coefficient function $q(x)$ in the Stürm-Liouville differential equation given a finite set of Neumann eigenvalues. We have so far dealt with coefficient functions that range from low order polynomials (constant, linear and quadratic), higher order to slowly and more rapidly oscillating functions, a discontinuous function and a simple approximation to shock.

Examining our work, we find that the use of the finite element method in question appears to be an effective way of solving the IEP. Some computational evidence was used to show how these can be obtained in prac-

tice. Promising results were discussed although basis functions used were linear.

The following are noted. Although linear basis functions are easy to program it may not be very effective for the reconstruction of coefficient functions that are of very high order. The numerical scheme developed uses equispaced discretized points which might not be efficient for certain coefficient functions. Too little or no work has been done on Stürm-Liouville inverse eigenvalue problems with periodic and Dirichlet-Neumann boundary conditions.

It may therefore be necessary to investigate the inverse Stürm-Liouville problem with the application of more sophisticated schemes such as the approximate reconstruction of at least two coefficient functions, the application of basis functions of higher orders, variable discretized points subject to any of the aforementioned boundary conditions.

It is therefore hoped that this will serve to stimulate our research further in future, concentrating mostly on problems and research areas identified above.

Chapter 5

Appendix

Programmes

Matlab is the programming language used in this research report. It is well suited for this study because it allows interactive experimentatio· · and graphical insights into the behaviour of methods and functions. It is also designed for easy computation of various matrix based scientific and engineering problems. A remarkable feature of MATLAB is its graphic capabilities. Below is the programme.

```
function [q,it]=newt13(a,b,n,tol)
% ************************************************************
% solving the additive inverse eigenvalue problem by the      *
% finite element discritization method, that is the programme *
% approximately reconstructs the potential q from             *
% -u''+(q(x)-lambda)u=0 where lambda is a given set of spectral *
% data (the spectral data involved consist of partial information*
% of the eigenvalues). a,b are initial and final values of x, n *
% is the length of the reconstruction vector, h is the increment *
% in x and tol is the tolerance.                              *
% ************************************************************
format compact
format long e
h=(b-a)/(n-1);
x=[a:h:b];
load eigval.dat          %load given spectrum
lambda=eigval(:,1);
q=input initial q        %initial guess of q
```

```
[B,Kr0]=constmat(n,h); %function assembling constant matrices
it=0;
Kq1=zeros(size(B));Kq2=zeros(size(B));Kq3=zeros(size(B));
Jb=zeros(size(B));
z1=[1,1,0;1,4,1;0,1,0];z2=[4,1,0;1,4,1;0,1,0];
z3=[0,0,0;0,4,1;0,1,1]; %elements of the matrix Kqj
Kq1(1:3,1:3)=z1;Kq3(n-2:n,n-2:n)=z3;
ndf=1000;
while ndf>tol
  Kq=recmat(a,q,n,b,h);  %function assembling reconstruction
                         %matrix
  sum=Kr0+Kq;        %summation of constant and reconstruction
                     %matrices Kr0 and Kq respectively

  [v,lam]=eig(sum,B); %computation of eigenvalues 'lam' and the
                      %corresponding eigenvectors 'v' of the
  [lam,v]=eigsot(lam,v);      %function updating sorting
  df=lam-lambda;              %update solution of nonlinear
                              %system
```

```
  ndf=norm(df);              %monitors rate of convergence
                              %at each iteration
  Jb=Jacmat(n,v,B,Kq1,Kq2,Kq3,z1,z2,z3,h); %function computing
                                          %Jacobian
  newq=q-(Jb\df);            %over-determined system
  q=newq;
it=it+1;
end
q_exact=exact values of the coeff. function q(x) for grph work;
disp('please enter to see graph')
pause
plot(x,q_exact,'o',x,q);
xlabel('grid points');ylabel('q_exact and q_approx.'),grid on;
gtext(' "o" reps. exact and "-" reps. approx.')
title('A graph of q_exact and q_approx.')

function [B,Kr0]=constmat(n,h)
% This function computes the entries of the n-by-n
% symmetric tridiagonal matrices B and Kr0.
B=zeros(n,n);
```

```
Kr0=zeros(n,n);
B(1,1)=2;
Kr0(1,1)=1;
for i=2:n-1
    B(i-1,i)=1;
    B(i,i-1)=B(i-1,i);
    B(i,i)=4;
    Kr0(i-1,i)=-1;
    Kr0(i,i-1)=Kr0(i-1,i);
    Kr0(i,i)=2;
end
B(n-1,n)=1;B(n,n-1)=B(n-1,n);B(n,n)=2;
Kr0(n-1,n)=-1;Kr0(n,n-1)=Kr0(n-1,n);Kr0(n,n)=1;
B=(h/6)*B;
Kr0=(1/h)*Kr0;

function Kq=recmat(a,q,n,b,h)
% This function computes the entries of the n-by-n
% symmetric tridiagonal matrix Kq which contains the
% coefficient function q(x).
```

```
Kq=zeros(n,n);

Kq(1,1)=30+q(2);

Kq(1,2)=q(1)+q(2);

for i=2:n-1

    Kq(i,i-1)=q(i-1)+q(i);

    Kq(i,i)=4*(q(i)+q(i+1));

    Kq(i,i+1)=q(i)+q(i+1);

end

Kq(n,n-1)=q(n-1)+q(n);

Kq(n,n)=q(n)+30;

Kq=(h/12)*Kq;
```

```
function [lam,v]=eigsot(lam,v)

% This function sorts the eigenvalues "lam" and the

% corresponding eigenvectors "v".

   lam=diag(lam);

   [lam,eip]=sort(abs(lam));

   v=v(:,eip);
```

```
function Jb=Jacmat(n,v,B,Kq1,Kq2,Kq3,z1,z2,z3,h)

% This function computes the entries of the Jacobian matrix Jb.Kq1 is the
```

```
% matrix obtained as a result of the partial derivatives of Kq w.r.t q2,
% Kq2 is w.r.t qi, i=3:n-1 and Kq3 is w.r.t qn. z1,z2,z3 are submatrices
% of Kq1,Kq2 and Kq3 respectively.
  for k1=1:n
      Jb(k1,1)=(1/(v(:,k1).'*B*v(:,k1)))*v(:,k1).'*Kq1*v(:,k1));
  end
  k2=2;
  while k2<=n-1
    Kq2(k2-1:k2+1,k2-1:k2+1)=[4,1,0;1,4,1;0,1,0];
    for i=1:n
       Jb(i,k2)=(1/(v(:,i).'*B*v(:,i)))*(v(:,i).'*Kq2*v(:,i));
    end
    Kq2(k2-1:k2+1,k2-1:k2+1)=zeros(3,3);k2=k2+1;
  end
  for k3=1:n
    Jb(k3,n)=(1/(v(:,k3).'*B*v(:,k3)))*(v(:,k3).'*Kq3*v(:,k3));
  end
  Jb=(h/12)*Jb;
```

Bibliography

[1] Ambartsumyan V A, *Über Eine Frage der Eigenwerttheorie*, Physics, 53 (1929) pp 690-695.

[2] Barnes D C, *The inverse eigenvalue problem with finite data*, SIAM Journal on Mathematical Analysis, 22 (1991) pp 732-753.

[3] Boley D and Golub G H, *Inverse Eigenvalue Problem for band matrices*, Lecture notes in Mathematics 630, Springer, Berlin, (1977) pp 23-31.

[4] Börg G, *Eine Umkehrung der Stürm-Liouville Eigenwertaufgab*, Acta Mathematics, 76 (1946) pp 1-96.

[5] Chen X and Wright J W, *The education testing problem revisited*, IMA Journal on Numerical Analysis, 15 (1995) pp 141-160.

[6] Dwyer H I and Zettl A, *Computing Eigenvalues of Regular Stürm Liouville Problems,* Electronic Journal of Differential Equations, 6 (1994) pp 1-10.

[7] Friedland S, *Inverse eigenvalue problems,* Linear Algebra and its Applications, 17 (1977) pp 15-51.

[8] Friedland S, Nocedal J & Overton M L, *The formulation and analysis of numerical methods for inverse eigenvalue problems,* SIAM Journal on Numerical Analysis, 24 (1987) pp 634-667.

[9] Hald O, *The inverse Sturm-Liouville problem with symmetric potentials,* Acta Mathematica, 141 (1978) pp 263-291.

[10] Hochstadt H, *On the Construction of a Jacobi Matrix from Spectral Data,* Linear Algebra and its Applications, 8 (1974) pp 435-446.

[11] Landau H J, *The Inverse Eigenvalue Problem for real symmetric Toeplitz matrices,* Journal of the American Mathematical Society, 7 (1994) pp 749-767.

[12] Lowe B D, Pilant W & Rundell W, *The recovery of potentials from finite spectral data*, SIAM Journal on Mathematical Analysis, 23 (1992) pp 482-504.

[13] Li R C, *Algorithm for Inverse Eigenvalue Problems*, Journal of Computational Mathematics 10 (1992) pp 97-111.

[14] Livinson N, *The Inverse Stürm-Liouville problem*, Mathematics, Tidsskr B25 (1949a) pp 25-30.

[15] Livinson N, *On the uniqueness of the potential in a Schrödinger equation for a given asymptotic phase*, Danske Vid. Selsk, Mat.-Fys. Medd., 25 (1949b) pp 25.

[16] Marletta M, *Numerical tests of the SLEIGN Software for Stürm-Liouville problems*, ACM Transactions on Mathematical Software, 17 (1991) pp 481-490.

[17] McLaughlin J R and Rundell W, *A uniqueness theorem for an inverse Stürm-Liouville problem*, Journal of Mathematical Physics, 28 (1987) pp 1471-1472.

[18] Nocedal J and Overton M L, *Numerical methods for solving Inverse Eigenvalue Problems* in Springer Lecture notes in Mathematics 1005, Numerical Methods Proceedings Caracas, (1982) pp 212-226.

[19] Paine J, *A Numerical Method for the Inverse Stürm-Liouville problem*, SIAM Journal on Scientific and Statistical Computing, 5 (1984) pp 149-156.

[20] Richard H Fabiano, Roger Knobel and Bruce D Lowe, *A finite-difference algorithm for an inverse Stürm-Liouville problem*, IMA Journal of Numerical Analysis, 15 (1995) pp 75-88.

[21] Trench W F, *Numerical solution of the Inverse Eigenvalue Problem for real symmetric Toeplitz matrices*, SIAM Journal on Scientific Computing, 18 (1997) pp 1722-1736.

[22] Wang J Y and Garbow B S, *A Numerical method for solving Inverse Eigenvalue Problem*, SIAM Journal on Scientific and Statistical Computing, 4 (1983) pp 45-51.

[23] Wang S J, *An algebraic approach to the Inverse Eigenvalue Problem for a quantum system with a dynamical group*, Journal of Physics A, Mathematical and General, 27 (1994) pp 5655-5671.

[24] Yitshak M R, *Inverse Eigenvalue Problem for a modified vibrating system*, SIAM Journal on Applied Mathematics, 53 (1993) pp 1762-1775.

Author: Boamah, Edward Kwasi.
Name of thesis: Inverse eigenvalue problem.

PUBLISHER:
University of the Witwatersrand, Johannesburg
©2015

LEGALNOTICES:

Copyright Notice: All materials on the University of the Witwatersrand, Johannesburg Library website are protected by South African copyright law and may not be distributed, transmitted, displayed or otherwise published in any format, without the prior written permission of the copyright owner.

Disclaimer and Terms of Use: Provided that you maintain all copyright and other notices contained therein, you may download material (one machine readable copy and one print copy per page)for your personal and/or educational non-commercial use only.

The University of the Witwatersrand, Johannesburg, is not responsible for any errors or omissions and excludes any and all liability for any errors in or omissions from the information on the Library website.