A SOLUTION METHOD FOR INTEGRAL EQUATIONS OF THE FIRST KIND IN TWO AND THREE DIMENSIONS

by

HUMBERTO D. PEREZ GONZALEZ

Thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE
in
MATHEMATICS

UNIVERSITY OF PUERTO RICO
MAYAGÜEZ CAMPUS
2005

Approved by:

______________________________ ______________________________
Lev Steinberg, Ph.D. Date
President, Graduate Committee

______________________________ ______________________________
Haedeh Gooransarab, Ph.D. Date
Member, Graduate Committee

______________________________ ______________________________
Krzysztof Rozga, Ph.D. Date
Member, Graduate Committee

______________________________ ______________________________
Pedro Vásquez, D.Sc. Date
Chairperson of the Department

______________________________ ______________________________
Noel Artiles, Ph.D. Date
Representative of Graduate Studies

Abstract

The main objectives of this work are to describe the theoretical use of the regularization method to solve integral equations of the first kind and to develop a solution method for integral equations of the first kind in two and three dimensions. To solve such equations, the kernels are represented by two-dimensional matrices and singular value decomposition is used to get the solutions. The regularization parameter is computed by using the L-curve method and the Discrepancy Principle.

Resumen

Los objetivos principales de este trabajo son describir el uso teórico del método de regularización para resolver ecuaciones integrales de primera especie y desarrollar un método de solución para las ecuaciones integrales de primera especie en dos y tres dimensiones. Para resolver este tipo de ecuaciones los núcleos se representan por matrices bidimensionales y se usa descomposición en valores singulares para obtener las soluciones. El parámetro de regularización se calcula usando los métodos de la L-curva y el Principio de Discrepancia.

A mis padres Maria y Humberto a quienes les estaré
eternamente agradecido.

To my parents María and Humberto to whom
I will be grateful forever.

Acknowledgements

I want to thank the members of the graduate committee for their advice and guidance, specially Haedeh Gooransarab for her careful reading of the initial draft.

I express my gratitude to:

Antonio Estv́ez, Luis Cáceres and Pedro Vásquez who gave me the opportunity to study in UPR-RUM.

The Department of Mathematics for its financial support.

Ladys and José, my family in Puerto Rico.

All my classmates and friends of the Department of Mathematics.

To my family and friends in Colombia who kindly helped me to achieve my goals.

TABLE OF CONTENTS

List of Tables

List of Figures

INTRODUCTION

We have two main objectives in this work. The first objective is to describe theoretically the Tikhonov regularization method to solve inverse problems. The second objective is to solve integral equations of the first kind in two and three dimensions, especially in a compact parallelepiped in $\mathbb{R}^2$ or $\mathbb{R}^3$. The treatment of integral equations is self-contained. Because integral equations of the first kind are particular cases of inverse problems, some topics about functional analysis and theory of inverse problems are included.

Integral equations of the first kind for one dimensional case are well documented and a variety of programs to solve them can be found, for instance, in [21] and [25], but programs for other dimensions are not found frequently. Therefore, programs to solve integral equations of the first kind for two and three dimensions are desirable. This kind of equation arises frequently in applied sciences. Wing [28], Lamm [13] and Denisov [5] show some examples where integral equations are used. Steinberg [24] got a system of integral equations of the first kind in three dimensions while he was trying to find a linearization for an inverse problem for an elastic body. That was the source of this work.

For completeness, the functional analysis of integral equations is included in Chapter 1. Moreover, we include a theoretical treatment of inverse problems, especially the Tikhonov regularization method [5, 25] in Chapter 2. Chapter 3 contains a description of the tools used to solve integral equations of the first kind. There, we describe some numerical methods and some examples are presented.

Due to the ill-posedness of integral equations, it is not easy to deal with them. Considering these types of equations in dimensions other than one is a little bit harder because of the complexity of multiple integrals. This can be seen in the type of programs developed which demand a huge quantity of memory in order to achieve acceptable results.

Chapter 1

BASIC RESULTS OF FUNCTIONAL ANALYSIS

In this chapter we give some basic results of functional analysis which will help us to understand the mathematical theory of functional equations used to solve inverse problems. We present an overview about metric spaces, Banach spaces, Hilbert spaces and several concepts defined over these spaces such as convergence, compactness, completeness and weak convergence. Also how they interact with each other and with operators defined on them. All these topics aim at the notion of compact operators whose principal properties we discuss in Sections 1.2 and 1.4.

1.1 Linear Operators

Definition 1.1.1 (Metric or distance)**.** *A metric (or distance) d on a nonempty set X is a function $d : X \times X \to \mathbb{C}$ satisfying the following four properties. $\forall x, y, z \in X$:*

1. $d(x, y) \geq 0$
2. $d(x, y) = 0$ *if and only if* $x = y$

3. *$d(x, y) = d(y, x)$*

4. *$d(x, z) \leq d(x, y) + d(y, z)$.*

The pair (X, d) is called a ***metric space****.*

Definition 1.1.2 (Normed Space)**.** *Let X be a linear space over $\mathbb{K}$ where $\mathbb{K}$ is either $\mathbb{R}$ or $\mathbb{C}$.*

1. *$\|x\| \geq 0, \quad \forall x \in X$*

2. *$\|x\| = 0, \quad$ if and only if $x = 0$*

3. *$\|\lambda x\| = |\lambda|\|x\|, \quad \forall x \in X, \quad \lambda \in K$*

4. *$\|x + y\| \leq \|x\| + \|y\|, \quad \forall x, y \in X$*

The pair $(X, \|\cdot\|)$ is called normed vector space or simply normed space. It is either real or complex depending on the nature of $\mathbb{K}$.

Definition 1.1.3 (Convergent sequence)**.** *Let (X, d) be a metric space*

i) *The sequence $\{x_n\}_{n=1}^{\infty} \subseteq X$ is said to be convergent to $x \in X$ if $d(x_n, x) \to 0$ when $n \to \infty$.*

ii) *A sequence $\{x_n\}_{n=1}^{\infty}$ is said to be a Cauchy sequence if for each $\epsilon > 0$, there is $N = N(\epsilon)$ such that $d(x_m, x_n) < \epsilon$ for all $m, n \geq N$. When every Cauchy sequence converges in a metric space, The metric space is called complete metric space.*

Remark 1. *Every normed space becomes a metric space if the metric is defined as $d(x, y) = \|x - y\|$.*

From now on each time we mention convergence in a normed space it means convergence in the metric that the norm defines on it.

Definition 1.1.4 (Banach Space)**.** *A complete normed space $(X, \|\cdot\|)$ is called Banach Space.*

Definition 1.1.5 (Inner product)**.** *An inner product on a real-vector space X is a real valued function of two variables $(\cdot, \cdot) : X \times X \to \mathbb{R}$ such that:*

1. *$(\cdot, \cdot)$ is linear in the first variable, i.e, $(\alpha x + \beta y, z) = \alpha(x, z) + \beta(y, z)$ for all $x, y, z \in X$ and all real numbers α and β*

2. *$(\cdot,\cdot)$ is symmetric, i.e, $(x,y)=(y,x)$ for all $x,y\in X$ and*

3. *$(x,x)\geq 0$ for each $x\in X$ and $(x,x)=0$ if and only if $x=0$.*

Remark 2. - *An inner product can be complex-valued, in this case the second condition in the foregoing definition must be changed to $(x,y)=\overline{(y,x)}$.*

- *A real vector space (a complex vector space) equipped with a real inner product (a complex inner product) is called a real inner product space (complex inner product space).*

- *Every inner product induces a norm $\|u\|=(x,x)^{1/2}$ on the space and this norm satisfies*

$$\|u_1+u_2\|^2+\|u_1-u_2\|^2=2(\|u_1\|^2+\|u_2\|^2). \tag{1.1}$$

and the above relation defines an inner product.

- *The identity above gives us a criterion to decide when a norm is induced by an inner product.*

Definition 1.1.6 (Hilbert Space)**.** *A Hilbert space is an inner product space which is complete in the norm induced by its inner product.*

Theorem 1.1.7. *Let a set M of a metric space Z be one-to one mapped by an operator A onto a set N of a metric space U. If A is continuous in M, and M is a compact set, then the inverse operator A^{-1} is continuous in N.*

Proof. See [5, p. 16]

Definition 1.1.8 (Linear operator)**.** *Let $T:X\to Y$ a mapping between normed spaces.*

(i) T is called linear operator if $T(\alpha x+\beta y)=\alpha T(x)+\beta T(y)$

Let T a linear operator.

(ii) T is bounded if there exists a constant $C>0$ such that $\|Tx\|_Y\leq\|x\|_X$ for all x.

(iii) T *is continuous in* $x \in X$ *if for each sequence* (x_n) *converging to* x*, the sequence* $(T(x_n))$ *converges to* $T(x)$*.*

(iv) T *is continuous in* X*, if it is continuous for all* $x \in X$*.*

Theorem 1.1.9. *Let* $T : X \to Y$ *be a linear operator then the following are equivalents,*

(i) T *is bounded in* X

(ii) T *is continuous in* X

(iii) T *is continuous at 0*

If $(X, \| \cdot \|_X)$ and $(Y, \| \cdot \|_Y)$ are Banach spaces over $\mathbb{K}$, the set of all bounded linear operators from X to Y, represented by $B(X, Y)$, is a vector normed space. For $T_1, T_2 \in B(X, Y)$, $\lambda \in \mathbb{K}$ and $x \in X$ we define

$$(T_1 + T_2)x = T_1x + T_2x, \qquad (\lambda T)x = \lambda(Tx),$$

and if $T \in B(X, Y)$, $\|T\|_{B(X,Y)}$ denote the lowest $C > 0$ such that $\|T(x)\| \leq C\|x\|$ for all $x \in X$.

Theorem 1.1.10. *Let* $T \in B(X, Y)$ *then*

(i) $\| \cdot \|_{B(X,Y)}$ *is a norm on* $B(X, Y)$

(ii) For $T \in B(X, Y)$*,* $\|T\| = \sup\{\|Tx\| : x \in X \text{ with } \|x\| = 1\}$

(iii) $\|Tx\| \leq \|T\|\|x\|$

If T is a linear operator defined from X to Y we denote its domain by $D(T)$ and its range by $R(T)$ and we write $T : D(T) \subseteq X \to Y$.

Definition 1.1.11 (Dual of a normed space)**.** *Let* X *be a normed space over* $\mathbb{K}$*. The set of all continuous linear operators from* X *to* $\mathbb{K}$ *is called the dual of* X*. It is denoted by* X'*. Customarily the elements of* X' *are called linear functionals or simply functionals.*

When $f \in X'$, sometimes $f(x)$ is noted (f, x) or $\langle f, x \rangle$ because it resembles an inner product. The dual of the dual of X, i.e, $(X')'$ is called bidual. We use X'' for it. Each fixed element $f \in X'$ defines a linear operator from X to $\mathbb{K}$, say φ_f,

$$\begin{aligned} \varphi_f : X &\rightarrow \mathbb{K} \\ x &\mapsto f(x) \end{aligned}$$

in the same way, each $x \in X$ define a linear transformation from X' to $\mathbb{K}$, say φ_x,

$$\begin{aligned} \varphi_x : X' &\rightarrow K \\ f &\mapsto f(x). \end{aligned}$$

Henceforth a canonical injection between X and a subset of X'' can be defined. It is an isometry.

$$\begin{aligned} J : X &\rightarrow X^{''} \\ x &\mapsto J(x) = \varphi_x \end{aligned}$$

in this fashion X can be identified with a subset of its bidual.

Now we are going to describe some orthogonal relations in Banach Spaces

[3, p. 23]. If X is a Banach space we define

$$\begin{aligned} M^{\perp} &= \{f \in X'; \langle f, x\rangle = 0, \quad \forall x \in M\} \quad M \subset X \qquad (1.2) \\ N^{\perp} &= \{x \in X : \langle f, x\rangle = 0, \quad \forall f \in N\} \quad N \subset X'. \qquad (1.3) \end{aligned}$$

Some interesting relations are:

$$\left(M^{\perp}\right)^{\perp} = \overline{M}, \qquad \overline{N} \subset \left(N^{\perp}\right)^{\perp}. \qquad (1.4)$$

Definition 1.1.12 (Weak topology)**.** *Let X be a Banach space. The weak topology on X is the intersection of all topologies on X with respect to each of which, all $\{\varphi_f\}_{f\in X'}$ are continuous.*

To distinguish between the strong convergence and the weak convergence we are going to use → for the former and ⇀ for the latter.

Definition 1.1.13 (Weak* topology)**.** *Let X be a Banach space. The weak* topology on X' is the weakest topology on X' with respect to each of which, all $\{\varphi_x\}_{x\in X}$ are continuous.*

The topologies above described are Hausdorff. A complete description of this topologies can be found in Brezis [3], also in Yosida [30, Sect.IV.7, p.111], and Dunford and Schwartz [6, p.475].

On X' three topologies are defined:

1. The norm induces a topology in X'. It is called strong topology or uniform topology of operators.

2. The weak topology which is induced by X'' over X', $\sigma(X', X'')$

3. The weak* topology $\sigma(X', X)$.

Definition 1.1.14 (Weak convergence)**.** *The convergence in the weak topology is called weak convergence.*

The convergence in those metric and topological spaces are related as follow [3];

Theorem 1.1.15. *Let E be a Banach space and let (x_n) be a sequence in E. Then*

(i) $x_n \rightharpoonup x$ *for* $\sigma(E, E') \Longleftrightarrow \langle f, x_n \rangle \to \langle f, x \rangle \quad , \forall f \in E'$.

(ii) If $x_n \to x$ *in the strong topology then* $x_n \rightharpoonup x$ *in* $\sigma(E, E')$.

(iii) If $x_n \rightharpoonup x$ *for* $\sigma(E, E')$ *then* $\|x_n\|$ *is bounded and* $\|x\| \leq \liminf \|x_n\|$.

(iv) If $x_n \rightharpoonup x$ *for* $\sigma(E, E')$ *and* $x_n \to x$ *in the strong topology in* E' *then* $\langle f_n, x_n \rangle \to \langle f, x \rangle$.

(v) If $x_n \rightharpoonup x_0$ *and* $\|x_n\| \to \|x_0\|$, *then* $\|x_n - x_0\| \to 0$.

(vi) If x_n *is bounded in norm, then we can choose from it a subsequence weakly convergent to a certain element* x_0.

(vii) Let A *be a linear continuous operator if* $x_n \rightharpoonup x$ *then* $A(x_n) \rightharpoonup A(x)$. *See [17, p. 216].*

Definition 1.1.16 (Closed operator)**.** *Let* X *be a Banach Space and* $T : D(T) \subseteq X \to X$ *a linear operator.* T *is said to be a closed linear operator if* $x_n \in D(T);\ x_n \to x \in X;\ Tx_n \to y$ *implies* $x \in D(T)$ *and* $Tx = y$

An immediate consequence of this is that every bounded operator is closed.

Now we discuss the concept of adjoint operator which plays an important roll in functional analysis. We follow [3, p.27]. Let E, F be Banach spaces and $A : D(A) \subseteq E \to F$ a linear transformation with dense domain. We will define an operator $A^* : D(A^*) \subseteq F' \to E'$ as follow:

$$D(A^*) = \{v \in F' : \exists c \geq 0 \text{ such that } | \langle v, Au \rangle | \leq c\|u\|, \quad \forall u \in D(A)\}.$$

It is known that $D(A^*)$ is a subspace F'. Now we define A^*v for each $v \in D(A^*)$. Given $v \in D(A^*)$ consider the mapping $g : D(A) \to \mathbb{K}$ defined by $g(u) = \langle v, Au \rangle\,, \quad u \in D(A)$. It follows that $|g(u)| \leq c\|u\|$ for all $u \in D(A)$. The Hahn-Banach theorem allows to extent g to a liner operator $f_g : E \to \mathbb{K}$ such that $|f_g(u)| \leq c\|u\|$, for all $u \in E$. This extension is unique because f_g is continuous on $D(A)$ and $D(A)$ is dense. Thus the following definition make sense.

Definition 1.1.17 (A djoint Operator)**.** *Let $A : D(A) \subseteq E \to F$ a linear operator between the Banach spaces E y F with dense domain.We call adjoint of the operator A, noted A^*, to the operator $A^* : D(A^*) \subset F' \to E'$ defined as follow:*

$$D(A^*) = \{v \in F' : |\langle v, Au \rangle| \leq c\|u\|, \quad \forall u \in D(A)\}$$

$$A^*v = f_g,$$

where f_g is the extension in the foregoing discussion.

A^* is a linear operator and it is related to A as follow:

$$\langle v, Au \rangle_{F',F} = \langle A^*v, u \rangle_{E',E} \quad \forall u \in D(A), \forall v \in D(A^*) \tag{1.5}$$

1.2 Compact Operators

Unless otherwise is stated, the main results in this section were taken of [3]. In the following we define the notion of compact operator and we study some of its properties. This kind of operators are also called completely continuous operators [14], [3, Sect. VI.1 - VI.3]. Before giving the definition we remind that a set C is relatively compact in a metric space (M, d) if its closure, i.e $\overline{C}$, is a compact set in (M, d).

Definition 1.2.1 (Compact operator)**.** *Let B_E denote the closed unit ball in the normed space E. A linear operator $A : D(A) \subseteq E \to F$ is said to be a compact operator if $\overline{A(B_E)}$ is a compact set in the strong topology of F,i.e, if $A(B_E)$ is relatively compact in the strong topology of F.*

The set of all compact operators from E to F is noted by $\mathcal{K}(E, F)$. If $F = E$, it is noted by $\mathcal{K}(E)$. It can be shown that $A : D(A) \subseteq E \to F$ is a compact linear operator if and only if it maps every bounded subset of $D(A)$ into a relatively compact set in F and this is precisely the definition of completely continuous operator in [17]. Another equivalent definition is: $A \in L(E)$ is compact if any bounded (x_n) sequence has a subsequence (x_{n_k}) such that $\{Ax_{n_k}\}$ converges.

Theorem 1.2.2. *Let A be a compact operator and suppose that x_n weakly converge to x. Then $\{Ax_n\}$ strongly converges to Ax. See [17, p. 229]*

Definition 1.2.3 (Finite-range operator)**.** *A linear operator T is said to be of finite range if $dim R(T) < \infty$.*

It follows immediately that every continuous operator of finite range is compact [3].

Theorem 1.2.4 ([3])**.** *$\mathcal{K}(E, F)$ is a closed normed subspace of $B(E, F)$ for the norm $\| \cdot \|_{B(E,F)}$*

Corollary 1.2.5. *Let (T_n) be a sequence of continuous operators with finite range from E to F and $T \in B(E, F)$ such that $\|T_n - T\| \to 0$. Then $T \in \mathcal{K}(E, F)$.*

Theorem 1.2.6. *Let E,F and G be Banach spaces. If $T \in B(E, F)$ and $S \in \mathcal{K}(F, G)$ then $S \circ T \in \mathcal{K}(E, G)$.*

Now we present two very important examples of compact operators.

Example 1 ([1])**.** *Let $[a,b]$ be a finite closed interval and let $K : [a,b] \times [a,b] \to \mathbb{R}$ be a continuous function. Consider the vector space $C[a,b]$ with the sup norm ($\|\cdot\|_\infty$), and then define $T : C[a,b] \to C[a,b]$ by*

$$T(f)(x) = \int_a^b K(x,y)f(y)dy \tag{1.6}$$

for each $f \in C[a,b]$. This is a compact operator which is called **integral operator** *and the function K is the* **kernel** *of T. The linearity of the integral guarantees that this operator is a linear operator.*

Put

$$M = sup\{|K(x,y)| : (x,y) \in [a,b] \times [a,b]\} < \infty.$$

The calculations $|T(f)(x)| \le M(b-a)\|f\|_\infty$ yields $\|T(f)\|_\infty \le M(b-a)\|f\|_\infty$ and $\|T\| < \infty$, so T is a bounded operator.

Let $B = \{f \in C[a,b] : \|f\|_\infty \le 1\}$ the unit ball in $C[a,b]$. Observe that $\bar{B}$ is closed and bounded. Thus, by the Ascoli-Arzelà theorem we need only to show that $T(B)$ is a equicontinuous subset of $C[a,b]$. To prove this, fix $x_0 \in [a,b]$, and let $\epsilon > 0$. Since K is uniformly continuous, there is $\delta > 0$ such that $|K(x_1,y) - K(x_2,y)| \le \epsilon$ holds whenever $|x_1 - x_2| < \delta$. Henceforth, if $x \in [a,b]$ satisfies $|x - x_0| < \delta$ and $f \in B$, then

$$|T(f)(x) - T(f)(x_0)| = \left| \int_a^b [K(x,y) - K(x_0,y)]f(y)dy \right| \le (b-a)\epsilon$$

This shows that $T(B)$ is equicontinuous at x_0. Since x_0 is arbitrary, $T(B)$ is equicontinuous everywhere. Thus, $\overline{T(B)}$ is compact. This complete the proof of the compactness of T.

Example 2 ([27])**.** *Let X be a compact subset of a Euclidean n -space. Consider $L_p(X)$ and $L_p(X \times X)$, $1 < p < \infty$. For $K \in L_p(X \times X)$ define $\|K\| = \left\{\int \left[\int |K(x,y)|^q \, dy\right]^{p/q} dx\right\}^{1/p}$ where $\frac{1}{p} + \frac{1}{q} = 1$. Consider the integral operator*

$$T(f)x = \int_X K(x,y)f(y)dy \tag{1.7}$$

and suppose that Tf is in $L_p(X)$ whenever f is. Then T is a linear compact operator on $L_p(X)$ when $\|K\| < \infty$.

Now we are going to prove this fact.

$$\begin{aligned}
|T(f)(x)|^p &\leq \left\{\int_X |K(x,y)||f(y)|dy\right\}^p \\
\int_X |T(f)x|^p dx &\leq \int_X \left\{\left[\int_X |K(x,y)|^q dy\right]^{p/q} \|f\|_p^p\right\} dx \\
&= \|K\|_p^p \|f\|_p^p \\
\|Tf\|^p &\leq \|K\|_p^p \|f\|_p^p \\
\|T\|_p &\leq \|K\|_p
\end{aligned}$$

The last inequality shows that if $\|K_n - K\| \to 0$ then $\|T_n - T\| \to 0$. Since $X \times X$ is compact, continuous functions are dense in $L_p(X \times X)$, so K can be approximated by a sequence K_n of continuous functions. Using the Weierstrass theorem each K_n can be approximated uniformly by a sequence of polynomials K_{n_i}, i.e, $K_{n_i} \overset{u}{\to} K_n$ when $i \to \infty$. Every polynomial can be written as

$$K_{n_i}(x,y) = \sum_{j=1}^{n_i} \phi_{ij}(x)\psi_{ij}(y), \quad \phi_{ij} \in L_p(X) \ \psi_{ij} \in L_q(X), \quad 1/p + 1/q = 1.$$

Define T_{n_i} as the integral operator with kernel K_{n_i} and T_n the integral operator with kernel K_n. Each T_{n_i} has finite range, so it is compact. Since

$K_{n_i} \to K_n$ when $i \to \infty$, $T_{n_i} \to T_n$. Then T_n is compact because $\mathcal{K}(X)$ is closed. As $K_n \to K$, T_n converges to T which is compact because $\mathcal{K}(X)$ is closed.

Definition 1.2.7 (Resolvent of an operator)**.** *Let $A : D(A) \subseteq X \to X$ an operator over a complex normed space. The resolvent set of the operator A ,$\rho(A)$,is the set of $\lambda \in \mathbb{C}$ which $\lambda I - A$ is invertible.*

The resolvent set of A, $\rho(A)$, is an open set of $\mathbb{C}$

Definition 1.2.8. *(Spectrum of an operator) If A is an operator as in the above definition the complement of $\rho(A)$ in $\mathbb{C}$ is called spectrum of A. It is denoted $\sigma(A)$.*

Since $\rho(A)$ is an open set of $\mathbb{C}$, the spectrum is closed.

Theorem 1.2.9. *Let $A \in \mathcal{K}(E)$ with $\dim E = \infty$. Then*

a) $0 \in \sigma(A)$

b) $\sigma(A)\backslash\{0\}$ *equals the nonzero eigenvalues*

c) Only one of the following holds:

- $\sigma(A) = \{0\}$,
- $\sigma(A)\backslash\{0\}$ *is finite,*
- $\sigma(A)\backslash\{0\}$ *is a sequence converging to zero.*

Theorem 1.2.10 (Fredholm Alternative)**.** *Let $A \in \mathcal{K}(E)$, where E is a Banach space. Then*

a) $\ker(\lambda I - A)$ *has finite dimension*

b) $R(\lambda I - A)$ *is closed, besides,* $R(\lambda I - A) = \ker(\lambda I - A^*)^{\perp}$ *and* $R(\lambda I - A^*) = \ker(\lambda I - A)^{\perp}$

c) $\ker(\lambda I - A) = \{0\} \Leftrightarrow R(\lambda I - A) = E$

d) $\dim\ker(\lambda I - A) = \dim\ker(\lambda I - A^*)$.

The last theorem tell us something about the resolution of the equation $\lambda f - Af = g$. Either the equation $\lambda f - Af = g$ has a unique solution for all $f \in E$ or $\lambda f - Af = 0$ has n linearly independent solutions, this is, the non homogeneous equation $\lambda f - Af = g$ is solvable if g verifies a finite number of orthogonal conditions ($f \in \ker((\lambda I - A^*)^{\perp})$.

In words of the Definition 1.3, $\lambda f - Af = g$ has a unique solution f if and only if $\phi(g) = 0$ whenever $A^*\phi = \lambda\phi, \phi \in E'$. See [27].

1.3 Singular Vectors

This topic can be found in [18] and [28].

Each operator A has associated two sets of vectors v_i and u_i, which are eigenvectors of A^*A and AA^* respectively. They are related by

$$Av_i = \sigma_i u_i, \quad A^*u_i = \sigma_i v_i$$

where the σ_i are the positive square roots of the nonzero eigenvalues η_i of AA^* and A^*A, i.e, $A^*Av_i = \eta_i v_i$ and $\sigma_i = \sqrt{\eta_i}$. The functions v_i and u_i are called singular vectors of A. The σ_i are called singular values. The operators A^*A and AA^* have the same eigenvalues, so they have the same singular values.

1.4 Theory in Hilbert spaces

For this section we suppose that all Hilbert spaces are separable. The main references for this part are **??** and **??**.

The notion of orthogonality defined in (1.3) can be restated in terms of inner products using the following theorem:

Theorem 1.4.1 (F. Riesz)**.** *If $\mathcal{H}$ is a Hilbert space and $f : \mathcal{H} \to \mathbb{C}$ is a continuous linear functional then there exists a unique vector $y \in \mathcal{H}$ such that $\langle f, x\rangle = f(x) = (x, y)$ holds for all $x \in \mathcal{H}$. Moreover $\|f\| = \|y\|$.*

With the notation of inner products two vectors x and y are orthogonal if $(x, y) = 0$. It is noted $x \perp y$. The orthogonal complement of a set M is

$$M^{\perp} = \{x \in X; x \perp y, \quad \forall y \in M\} \quad M \subset X.$$

A set of pairwise orthogonal vectors is called an orthogonal set, if in addition each vector has norm one, it is called orthonormal set. A orthogonal set has the Pythagorean property, i.e, if $N = \{x_1, x_2, ..., x_n\}$ is an orthogonal set $\|\sum_{i=1}^{n} x_i\|^2 = \sum_{i=1}^{n} \|x_i\|^2$. If N is a closed subspace of $\mathcal{H}$ then span$(N^{\perp}) = \mathcal{H}$ and $\mathcal{H} = N \oplus N^{\perp}$

Theorem 1.4.2. *Let $\{x_i\}_{i \in I}$ be an orthonormal family of vectors in an inner product space, then*

$$\sum_{i \in S} |(x, x_i)|^2 \leq \|x\|^2$$

holds for each vector x. In particular, for each x all but at most countable number of (x, x_i) vanish.

Theorem 1.4.3. *A vector subspace N of a Hilbert space is dense if and only if the zero vector is the one and only vector orthogonal to N.*

Definition 1.4.4 (Complete orthogonal set). *Let $\mathcal{H}$ be an inner product space and $S \subset \mathcal{H}$. S is said to be complete if $x \perp s$ for all $s \in S$ implies $x = 0$.*

Beginning with a dense subset $\{x_1, x_2, ...\}$ of a Hilbert space, a complete orthogonal set can be constructed using Gram-Schmidt orthogonalization process. This kind of sets are called orthonormal basis and they will be noted as $\{e_i\}_{i\in I}$.

Theorem 1.4.5. *For an orthonormal family $\{e_i\}_{i\in I}$ of vectors in a Hilbert space the followings statements are equivalents*

1. *The family $\{e_i\}_{i\in I}$ is an orthonormal basis*
2. $x \perp e_i, \quad \forall i \in I$ *implies* $x = 0$
3. *For all $\in \mathcal{H}$, $\;(x, e_i) \neq 0$ for at most countably many indices and $x = \sum_{i\in I}(x, e_i)e_1$, where the series converge in norm.*
4. *For each pair of vectors x and y we have $(x, e_i) \neq 0$ and $\overline{(y, e_j)} \neq 0$ for at most countably many indices and $(x, y) = \sum_{i\in I}(x, e_i)\overline{(y, e_i)}$*
5. *(Parseval's identity) For each vector x we have $\|x\|^2 = \sum_{i\in I} |(x, e_i)|^2$.*

Theorem 1.4.6. *An infinite-dimensional Hilbert space $\mathcal{H}$ is separable if and only if it has a countable orthonormal basis. Moreover, in this case, every orthonormal basis is countable.*

By using the Riesz's theorem the relation (1.5) can be expressed as

$$(Tu, v) = (u, T^*v), \quad \forall u, v \in \mathcal{H}.$$

If the operator satisfies $T^* = T$ then

$$(Tu, v) = (u, Tv), \quad \forall u, v \in \mathcal{H}.$$

A operator satisfying $T^* = T$ is called self-adjoint or Hermitian and its eigenvalues are all real. Besides, its eigenvalues are non negative if $(Tx, x) \geq 0$ for all x.

Theorem 1.4.7. *Let $\mathcal{H}$ be a separable Hilbert space and let T be a self-adjoint compact operator. Then $\mathcal{H}$ admits a orthogonal basis which consists only of eigenvectors of the operator.*

The proof of theorems in this section can be found in [1, 3, 27].

Since every finite-range operator is compact and $\mathcal{K}(\mathcal{H})$ is closed then the last theorem gives us a way to approach T by finite-range operators. If T is compact self-adjoint operator in a Hilbert space, let $\{u_n\}_{n=1}^{\infty}$ its eigenvectors with eigenvalues $\{\gamma_n\}$. Put $E_0 = \ker(T)$ and $E_n = \ker(\gamma_n I - T)$. These sets are pairwise orthogonal [3, p. 97]. Then $Tu = \sum_{n=1}^{\infty} \gamma_n u_n,\ \gamma_n \in E_n$. Define $Tu = \sum_{n=1}^{k} \gamma_n u_n$. Henceforth T_k has finite range and

$$\|T_k - T\| \leq \sup_{n \geq k+1} |\gamma_n| \to 0$$

so $T_k \to T$.

Chapter 2

ELEMENTS OF INVERSE PROBLEMS

The principal objective of this chapter is to discuss some theoretical facts about inverse problems which will give us the tools to solve operator equations arising in the theory of inverse problems. We are mainly interested in integral operators involved in integral equations. For this part we follow [5]. Nevertheless, we can take a look at [13] and [12] to review some methods to solve this kind of problems.

2.1 Preliminaries

Consider the linear system $Az = u$ where

$$A = \begin{bmatrix} 0.6410 & 0.2420 \\ 0.6210 & 0.2300 \end{bmatrix} \quad u = \begin{bmatrix} 0.8830 \\ 0.8510 \end{bmatrix}.$$

The exact solution is $z = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$. Suppose that we do not know the exact right-hand side but an approximation $u_\delta = \begin{bmatrix} 0.8930 \\ 0.8210 \end{bmatrix}$ which correspond to a perturbation $\delta = \begin{bmatrix} 0.01 \\ -0.03 \end{bmatrix}$, i.e, $u = u + \delta$. We expect that for small changes δ in u we get solutions near to u. We solved the equation $Az = u_\delta$ with MATLAB using both the command $A\backslash u_\delta$ and $z = A^{-1}u_\delta$ to get $z_\delta = \begin{bmatrix} -2.3520 \\ 9.9201 \end{bmatrix}$. This solution is far from the exact solution.

Let us try another approximation. Take as approximated solution $z_\delta = \arg\min_z \|Az - u_\delta\|$ in this case we obtain an approximated solution $z_\delta = \begin{bmatrix} -2.3521 \\ 9.9201 \end{bmatrix}$. It is still far from the exact solution. We need to decrease the size of the obtained solution while we keep $\|Az - u_\delta\|$ as small as possible. It can be done by imposing a weight α over the size of the approximated solution. The original problem is replaced by the following. To approach the exact solution by

$$z_\alpha = \arg\min_z \|Az - u_\delta\| + \alpha\|z\|, \tag{2.1}$$

or

$$z_\alpha = \arg\min_z \|Az - u_\delta\| \text{ subject to } \|x\| \leq \alpha \tag{2.2}$$

for a suitable α. Using (2.1) we get the following z_α's for the indicated values of α.

α	App. Solution
10^{-5}	(-0.4827, 4.9227)
10^{-4}	(0.9005 1.2247)
10	(0.0992 0.0371)

Certain values of α can give approximated solutions which are close to the exact solution (third row in the above table), but others do not. Thus, the principal issue is how to choose α so that we can get "good" solutions.

Why is so troublesome to dealt with this simple example? The columns of A are very similar, A is near to singularity and it has a high condition number. To summarize, the matrix A is ill behaved. This situation is not the exception, it can be the rule. Consider the following example we take from [18, p. 176].

Suppose that $K(x, y)$ is an absolutely integrable function. Consider the integral operator $Tf = g$ with unknown f. Define

$$h_n(x) \equiv \int_a^b K(x, y) \cos(ny) dy \quad (c \leq x \leq d).$$

The Riemann-Lebesgue implies $h_n \to 0$ in the sup norm. Taking n large enough

$$\sup_{c \leq x \leq d} |h_n(x)| \leq \epsilon.$$

Put $Kf = g$ and $g_1 = g + Ch_n$ where f and f_1 are solution of integral equation with data g and g_1 respectively. Then

$$\int_a^b K(x, y)(f(y) - f_1(y))dy = g - g_1 = C \int_a^b K(x, y) \cos(ny)dy$$

The change in f corresponding to a change $\delta g = Ch_n(x)$ in g is $\delta f = C \cos(ny)$. So,

$$\frac{\|\delta f\|}{\|\delta g\|} \leq \frac{|C(b-a)|}{\|\delta g\|}.$$

Since $\|\delta g\|$ can be done as small as you want, $\|\delta f\|$ becomes as large as you like. The integral operator in this case is ill behaved too.

The equations (2.1) and (2.2) are particular examples of regularization methods. They are used to handle the kind of problems described above. The first one is called Tikhonov regularization method. We will deal with it in the next sections.

2.2 General Description of Inverse Problems

Denisov [5] said the development of the theory of inverse problems is mainly motivated by applied research, the inverse problems arise when it is necessary to process and to interpret observed data in different branches of science. Later in the text, he said that in these cases some characteristics of mathematical models are unknown and they need be determined using results of an experiment. A common characteristic of these problems is that the conclusions have to be derived of indirect manifestations of the object which can be measured experimentally.

These problems are related with:

- Chemical Kinetics [5]
- Inverse Heat Conduction Problems [2].
- Image Reconstructions [19]
- Parameter Estimation [13]
- Inverse Scattering Problems [15]

- Inverse Problems in Elasticity [22, 23, 24]

- Inverse Sturm-Liouville Problem [4, 16, 20, 26]

In an abstract setting, a inverse problem can be formulated as an operator equation

$$Az = u \tag{2.3}$$

where the operator A maps a linear normed space Z into a linear space U. Usually, Z is a complex Banach or Hilbert space and u is a data function. The operator A can be linear or nonlinear, depending on the application.

Lamm in [13] says that the purpose is to find an element $\bar{z} \in Z$ which solve (2.3). If the data u is not in the range of A there is no possibility to find a solution of 2.3, in this case, we can modify the problem and instead look for a solution $\bar{z}$ of the least squares problem

$$J(\bar{z}) = \min_{z} J(z) \tag{2.4}$$

where

$$J(u) := {\|A(z) - u\|^2}_U.$$

The operator equation (2.3) is well-posed if for each u a solution of the problem exists, it is unique and it continuously depends on u. A problem which is not well-posed is called ill-posed. According to this, the problem of solving (2.3) is well-posed if the inverse operator A^{-1} is well defined and continuous in U. We can take a look at Theorem 1.1.7 on Page 5, it gives some conditions under which the equation is well posed. When A is a linear completely continuous operator(see Section 1.2) is called an operator of the first kind.

Now we consider the operator equation (2.3) of first kind where A is a continuous operator mapping a metric space Z into a metric space (U, d). Usually this an ill-posed problem. In the sequel we are going to assume that for each $\bar{u}$ in a compact subset of Z, exist a unique $\bar{z}$ such that $A\bar{z} = \bar{u}$. Besides we are going to assume that $\bar{u}$ is approximated by u_δ, and $d(u_\delta, \bar{u}) \leq \delta$.

We hope that if δ becomes smaller and smaller, u_δ would get closer and closer to the real solution in some way. The following theorem states precisely the way in what sense we take the closeness.

Theorem 2.2.1 ([5])**.** *Let Z be a metric space and M a compact subset of Z. Consider*

$$Z_\delta^M = Z_\delta \cap M,$$

where $Z_\delta = \{z : d(Az, u_\delta) \leq \delta\} \neq \emptyset$. When $\delta \to 0$,

$$\sup_{z \in Z_\delta^M} d(z, \bar{z}) \to 0.$$

Proof. Suppose that the theorem is false, i.e, there exists a sequence $\delta_n \to 0$ and $z_{\delta_n} \in Z_{\delta_n}^M$ such that $d(z_{\delta_n}, \bar{z}) \geq \epsilon$, where ϵ is a positive constant. Since $z_{\delta_n} \subset M$, we can take limit in the inequality $d(z_{\delta_n}, \bar{z}) \geq \epsilon$, to obtain

$$d(z_0, \bar{z}) \geq \epsilon.$$

Since $z_{\delta_k} \in Z^M_{\delta_k} \leq \delta_k$, $d(Az_{\delta_k}, u_{\delta_k}) \leq \delta_k$. Taking limits in this inequality by $\delta_k \to 0$, we obtain that $d(Az_0, \bar{u}) = 0$. Therefore, $Az_0 = A\bar{z} = \bar{u}$ and by the uniqueness of the solution of 2.3 for $\bar{u}$, we have $z_0 = \bar{z}$. This equality contradicts $d(z_0, \bar{z}) \geq \epsilon$. This means the theorem is true.

2.3 Regularization Method

Now we are going to describe the so called Tikhonov regularization method [5]. It is presented in a general framework within a Hilbert space.

Suppose that, for a linear compact operator A, the Equation 2.3 has a unique solution $\bar{z}$ for each exact $\bar{u}$, and $\bar{u}$ is approximated by u_δ and we know the estimation $\|u_\delta - \bar{u}\| \leq \delta$. The main objective here is to construct an approximate solution z_δ of (2.3) which would converge to $\bar{z}$ for $\delta \to 0$. Since (2.3) can be an ill-posed problem, its solution can oscillate dramatically, see Section 2.1, we add an stabilizer term to the expression $\|Az - u\|$ in order to control this ill behavior.

Consider the functional

$$M^\alpha(z) = \|Az - u\|^2 + \alpha\|z\|^2,$$

where α is considered as a parameter. The roll of α is to coordinate the compromise between the accuracy of the model-fitting and the stability of the possible solutions [**?**].

Theorem 2.3.1 ([5]). *For every $u \in U$ and $\alpha > 0$ the functional $M^\alpha(z)$ attains its infimum in an element of U and it is unique.*

Proof. Since $M^\alpha(z)$ is nonnegative, then $M_0 = \inf\limits_{z\in Z} M^\alpha(z) \geq 0$. Now we consider the minimizing sequence z_n , $M^\alpha(z_n) \to M_0$ for $n \to \infty$. We can take the z_n ordered such that $M^\alpha(z_{n+1}) \leq M^\alpha(z_n)$; then

$$M^\alpha(z_n) = \|Az_n - u\|^2 + \alpha\|z_n\|^2 \leq M^\alpha(z_1) = M_1.$$

Therefore, $\|z_n\| \leq (M_1/\alpha)^{1/2}$. Since z_n is bounded in norm, by the Theorem 1.1.15(vi) on Page 9 we can choose from it a subsequence z_k weakly convergent to z_0. Now we shall prove that $M^\alpha(z_0) = M_0$. We have, by the theorem last mentioned, $\|z_0\| \leq \liminf \|z_k\|$ and by Theorem 1.2.2 $Az_k \to Az_0$, so $\|Az_k - u\| \to \|Az_0 - u\|$ and $\|Az_k - u\|^2 \to \|Az_0 - u\|^2$.

Now take $\epsilon > 0$, since $\lim\limits_{n\to\infty} M^\alpha(z_n) = M_0$ there is k_0 such that $|M^\alpha - M_0| < \epsilon$ for $n \geq k_0$. So, $M^\alpha(z_k) \leq M^\alpha(z_0)$. Besides $\alpha\|z_0\|^2 \leq \alpha\|z_k\|^2 + \epsilon$ because $\|z_0\| \leq \liminf \|z_k\|$. Also we have $\|Az_0 - u\|^2 \leq \|Az_k - u\|^2 + \epsilon$, $M^\alpha(z_k) \leq M_0$ because $\|Az_k - u\| \to \|Az_0 - u\|$. From the inequalities above we can conclude

$$\begin{aligned}
M^\alpha(z_0) &= \|Az_0 - u\|^2 + \alpha\|z_0\|^2 \\
&\leq \|Az_k - u\|^2 + \alpha\|z_k\|^2 + 2\epsilon \\
&= M^\alpha(z_k) + 2\epsilon \\
&\leq M_0 + 3\epsilon.
\end{aligned}$$

So, $M^\alpha(z_0) \leq M_0$ because ϵ is arbitrary. Since M_0 is infimum, $M_0 \leq M^\alpha(z_0)$. Henceforth $M^\alpha(z_0) = M_0$. Now we are going to prove the uniqueness of z_0. The functional $M^\alpha(z)$ is differentiable for each z, and its gradient is given by $2(A^*Az - A^*u + \alpha z)$. Extrema for $M^\alpha(z)$ must satisfy

$$2(A^*Az - A^*u + \alpha z) = 0 \tag{2.5}$$

which z_0 satisfies. This equation is linear and it has a unique solution if $\alpha z + A^*Az = 0$ has only the zero solution. Suppose that there is $z_1 \neq 0$ such that $\alpha z_+ A^*Az_1 = 0$. Then

$$(\alpha z_1 + A^*Az_1, \alpha z_1 + A^*Az_1) = 0$$

$$\alpha^2(z_1, z_1) + 2\alpha(Az_1, Az_2) + (A^*Az_1, A^*Az_1) = 0,$$

but this is not possible because the first term is a positive one and the other two are nonnegative, so the equation $\alpha z + A^*Az = 0$ has a unique solution. Hence z_0 is unique.

Suppose now that $z_{\alpha(\delta)}$ is the element which gives the minimum for $M^{\alpha(\delta)}(z) = \|Az - u_\delta\| + \alpha(\delta)\|z\|^2$ with both $\alpha(\delta)$ and δ greater than zero.

Theorem 2.3.2 ([5])**.** *If $\alpha(\delta) > 0$ for $\delta > 0$, $\alpha(\delta) \to 0$ and $\delta^2/\alpha(\delta) \to 0$ for $\delta \to 0$, then $\|z_{\alpha(\delta)} - \bar{z}\| \to 0$ for $\delta \to 0$.*

Proof. To reach a contradiction suppose that $\|z_{\alpha(\delta)} - \bar{z}\|$ does not converge to zero. Then, there is $\epsilon > 0$ and $\delta_k \to 0$ that $\|z_{\alpha(\delta_k)} - \bar{z}\| \geq \epsilon$. As the element $z_{\alpha(\delta_k)}$ gives the infimum for $M^{\alpha(\delta_k)}(z) = \|Az - u_{\delta_k}\|^2 + \alpha(\delta_k)\|z\|^2$ then

$$M^{\alpha(\delta_k)}(z_{\alpha(\delta_k)}) \leq M^{\alpha(\delta_k)}(\bar{z}).$$

Therefore,

$$\alpha(\delta_k)\|z_{\alpha(\delta_k)}\|^2 \leq \|A\bar{z} - u_{\delta_k}\|^2 + \alpha(\delta_k)\|\bar{z}\|^2.$$

Since

$$\|A\bar{z} - u_{\delta_k}\| = \|\bar{u} - u_{\delta_k}\| \leq \delta_k,$$

$$\|z_{\alpha(\delta_k)}\|^2 \leq \delta_k^2/\alpha(\delta_k) + \|\bar{z}\|^2.$$

By hypothesis, $\delta_k^2/\alpha(\delta_k)$ is bounded and so is $z_{\alpha(\delta_k)}$. Therefore we can choose a subsequence $z_{\alpha(\delta_m)}$ weakly convergent to a certain element z_0. Then

$$\|z_0\| \leq \liminf \|z_{\alpha(\delta_m)}\| \leq \limsup \|z_{\alpha(\delta_m)}\| \leq \|\bar{z}\|, \tag{2.6}$$

Since

$$\begin{aligned} M^{\alpha(\delta_m)}(z_{\alpha(\delta_m)}) &\leq M^{\alpha(\delta_m)}(\bar{z}) \\ \|Az_{\alpha(\delta_m)} - u_{\delta_m}\|^2 + \alpha(\delta_m)\|z_{\delta_m}\|^2 &\leq \|A\bar{z} - u_{\delta_m}\|^2 + \alpha(\delta_m)\|\bar{z}\|^2 \\ \|Az_{\alpha(\delta_m)} - u_{\delta_m}\| &\leq (\delta_m^2 + \alpha(\delta_m)\|\bar{z}\|^2)^{1/2}. \end{aligned}$$

We get the estimates for the norm

$$\begin{aligned} \|Az_{\alpha(\delta_m)} - \bar{u}\| &\leq \|Az_{\alpha(\delta_m)} - u_{\delta_m}\| + \|u_{\delta_m} - \bar{u}\| \\ &\leq (\delta_m^2 + \alpha(\delta_m)\|\bar{z}\|^2)^{1/2} + \delta_m. \end{aligned} \tag{2.7}$$

Taking into account that $z_{\alpha(\delta_m)}$ is weakly convergent to z_0, A is completely continuous, and taking limit when $\delta_m \to 0$, we get $\|Az_0 - \bar{u}\| = 0$. By the uniqueness of the solution of Equation 2.3, we obtain $z_0 = \bar{z}$. Inequality (2.6) yields $\|z_{\alpha(\delta_m)}\| \longrightarrow \|\bar{z}\|$; therefore, the sequence $z_{\alpha(\delta_m)}$ weakly converges to z_0 and $\|z_{\alpha(\delta_m)}\| \longrightarrow \|z_{\delta_0}\|$. Then

$$\|z_{\alpha(\delta_m)} - \bar{z}\| \to 0 \text{ for } \delta_m \to 0,$$

by Theorem 1.1.15(v). This contradicts the first inequality in the proof.

The foregoing theorem guarantees the convergence of the approximated solutions to the real solution when δ and $\delta^2/\alpha(\delta)$ tend to 0. In real problems we have specific values of δ, so it is recommendable to have a way to choose an α which will provide the convergence of the approximated solutions to the exact solution.

2.4 Choosing the parameter of regularization

In this section we are going to discuss how to choose the regularization parameter which could give the best regularized solution in a sense we illustrate below. Two methods treated here are the Discrepancy Principle and L-curve method.

2.4.1 Discrepancy Principle

Consider the equation (2.3) where A is a compact operator mapping a separable Hilbert space Z into another separable Hilbert space U such that $\overline{R(A)} = U$. Suppose that for a exact $\bar{u}$ the equation $Az = \bar{u}$ has a unique solution $\bar{z}$. Also suppose we do not know $\bar{u}$, but an approximation u_δ to which $\|u_\delta - \bar{u}\| \leq \delta$ and $0 < \delta < \|u_\delta\|$. Put $\hat{z}_\alpha$ the element which gives the infimum for the functional

$$\|Az - u_\delta\|^2 + \alpha\|z\|^2.$$

Lema 2.4.1. *Let u be a nonzero element of U, z_{α_1} and z_{α_2} the elements of Z which give the minimum of $M^{\alpha_1}(z)$ and $M^{\alpha_2}(z)$. For $\alpha_1 \neq \alpha_2$ then $z_{\alpha_1} \neq z_{\alpha_2}$*

Now define for $\alpha > 0$

$$\begin{aligned} m(\alpha) &= \|Az_\alpha - u\|^2 + \alpha\|z_\alpha\|^2 \\ \varphi(\alpha) &= \|Az_\alpha - u\|^2 \\ \Psi(\alpha) &= \alpha\|z_\alpha\|^2 \end{aligned}$$

Lema 2.4.2. 1. *For a nonzero u the functions m and φ are strictly increasing and Ψ is strictly decreasing.*

2. *The functions above defined are continuous for $\alpha > 0$*

3. *For $u \neq 0$,*

$$\lim_{\alpha \to 0^+} \varphi(\alpha) = 0, \qquad \lim_{\alpha \to \infty} \varphi(\alpha) = \|u\|^2$$

Under above conditions the following theorem holds:

Theorem 2.4.3 (Discrepancy Principle)**.** *For each $\delta \in (0, \|u_\delta\|)$ there is a unique solution $\alpha(\delta)$ of the equation $\hat{\varphi}(\alpha) = \|A\hat{z}_\alpha - u_\delta\|^2 = \delta^2$. Besides, $\|\hat{z}_{\alpha(\delta)} - \bar{z}\| \to 0$ for $\delta \to 0$.*

The nonlinear equation in the last theorem can be approximated by iterative methods such as Newton method.

2.4.2 L-Curve method

For this section we use as reference Section 2.9 of [11] and [10].

We saw in the preliminary section that when we minimized the Tikhonov functional, it was not necessarily true that for small values of $\|Az_\alpha - u_\delta\|$ the solutions were close to $\bar{z}$. Generally z_α can oscillate or to have a very high size. One can draw this two quantities as a function of the parameter α and to find α such that both quantities match the given problem. Intuitively, the ideal α is the α which gives the corner of the parametric curve. This corner is emphasized when the parametric curve is drawn in log-log scale.

The L-Curve is a log-log plot of the norm of a regularized solution ($\|z_\alpha\|$) versus the residual norm ($\|Az_\alpha - u\|$), see Figure 2.4.2. It shows the compromise between the regularized solutions and how well they fit the given data u when the parameter α varies. The L-curve is used both as an analysis tool and as a method for choosing the regularization parameter.

Basically, as it is stated in the foregoing references, the L-curve is implemented for discrete problem such as those derived from the discretization of the equation of the first kind by using either quadrature methods or Galerkin

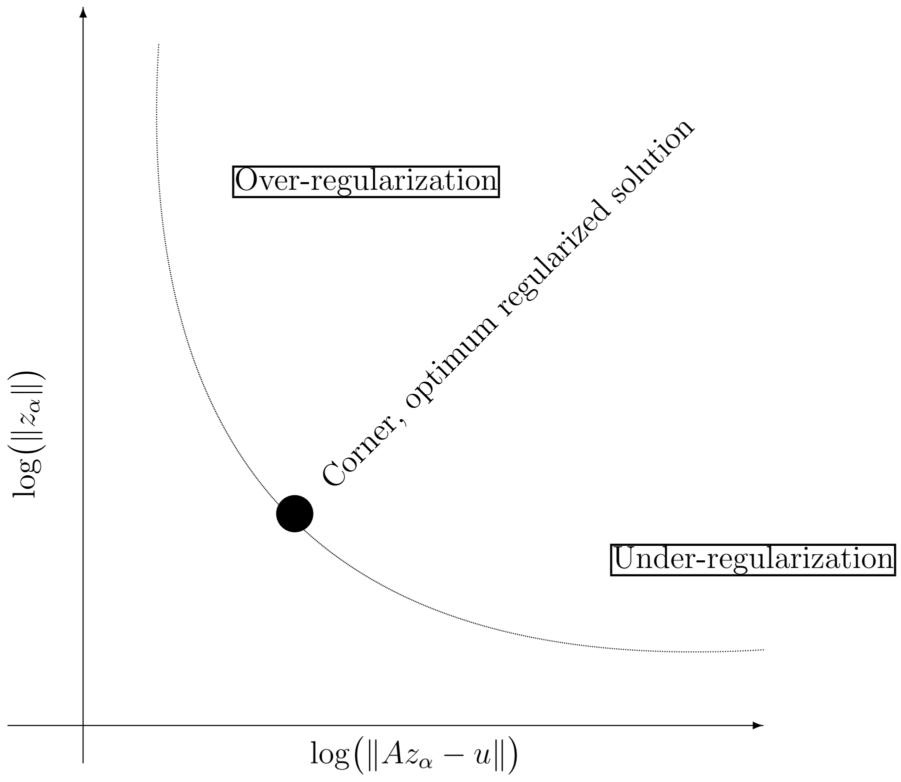

Over-regularization
Corner, optimum regularized solution
Under-regularization
$\log\big(\|z_\alpha\|\big)$
$\log\big(\|Az_\alpha - u\|\big)$

Figure 2.1: General form of the L-curve

method. Using the singular value decomposition, the residual norm and the norm of the regularized solution can be written as

$$\|z_\alpha\|^2 = \sum_{i=1}^{n} \left(f_i \frac{u_i^T u}{\sigma_i} \right)^2 \tag{2.8}$$

$$\|Az_\alpha - u\|^2 = \sum_{i=1}^{n} \left((1 - f_i) u_i^T u \right)^2 \tag{2.9}$$

where

$$f_i = \frac{\sigma_i^2}{\sigma_i^2 + \alpha^2}.$$

The vertical part of the L-curve corresponds to those regularized solutions to which $\|z_\alpha\|$ presents significant changes due to small variations in the parameter α because of the error δ prevails on z_α. The horizontal part of the L-curve corresponds to solutions where the residual norm is more sensitive to α because the regularization error prevails on z_α. Why is the L-curve important? Well, by locating the corner of the L-curve, an approximation to the optimal regularization parameter can be computed. Henceforth, α can be found such that z_α has a good trade-off between the error due to regularization and the error due to δ. The corner of the L-curve is defined in [10] as the point of maximum curvature. If we define

$$\eta = \|z_\alpha\|^2, \quad \rho = \|Az_\alpha - u\|^2$$

and

$$\hat{\eta} = \log(\|z_\alpha\|^2), \quad \hat{\rho} = \log(\|Az_\alpha - u\|^2)$$

the curvature of the L-curve is given by

$$\kappa = 2\frac{\hat{\rho}'\hat{\eta}'' - \hat{\rho}''\hat{\eta}'}{\left((\hat{\rho}')^2 + (\hat{\eta}')^2\right)^{3/2}}.$$

In that reference technical details can be found.

2.4.3 Similarities and differences between the discrepancy method and L-curve method

Now, some differences and similarities between the discrepancy method and L-curve method are established. They are related to the size of the regularization parameter and the way they are computed. Also some limitations of the methods are mentioned. The following summarizes the principal advantages and disadvantages mentioned by Hansen in [10].

The first important difference is that the discrepancy method is based on knowledge, or a good estimate, of $\|\delta\|$ while the L-curve method does not require δ, instead it uses the right-hand side u. The L-curve is not a rigorous method in the analytical sense. When a good estimate for δ is known, the discrepancy method gives a good regularization parameter which corresponds to a regularized solution just to the right of the L-curve corner[10]. Underestimated values of δ leads to under-regularized solutions, but overestimated values of δ leads to over-regularized solution with too large regularized error.

The L-curve has some limitations. Hanke [8] showed that for smooth exact solutions this criterion will fail in the estimation of α. The smoother the solution the worse the α computed by the L-curve criterion. Another limitation is related to the size of the discretization mesh. L-curve gives too large regularization parameters as the size of the mesh increases, i.e, the solutions are over-regularized.

Chapter 3

INTEGRAL EQUATIONS

3.1 Definition and classification

Let X be a measure space. Consider a measurable function $g : X \to \mathbb{R}$ defined in X. Let K a measurable function from $X \times X$ to $\mathbb{R}$ and dy a σ-finite measure on X and consider α as a parameter. In Section 1.2, Equation 1.7 we have defined the integral operator

$$T(f)(x) = \int_X K(x, y)f(y)dy.$$

It was shown it is a compact operator. Now consider

$$f = g + \alpha T f, \tag{3.1}$$

where f is an unknown function. This equation is called an integral equation. For a compact operator T, Equation 3.1 is called integral equation of the second kind and

$$g = T f \tag{3.2}$$

is called integral equation of the first kind.

Since T is a compact operator the Problem 3.2 is ill-posed. The solution can not be unique or a solution can not depend continuously on the data function. See example in Section 2.1.

We shall restrict ourselves to the case $X = L_p(X), \quad 1 \leq p \leq \infty$. When $p = 2$, $L_p(X)$ is a Hilbert space with the inner product defined by $(f, g) = \int_X f(y)g(y)dy$. We also assume that $T(f) \in L_p$ whenever f is. The adjoint of T is

$$T^*(f)y = \int_X K(x,y)f(x)dx$$

If we put $X = [a,b], \quad a, b \in \mathbb{R}$ and we take the Riemann integral. Integral equations of the form

$$\begin{aligned} f(x) &= g(x) + \int_a^b K(x,y)f(y)dy \qquad (3.3) \\ g(x) &= \int_a^b K(x,y)f(y)dy \qquad (3.4) \end{aligned}$$

are called linear Fredholm integral equations of the second kind and of the first kind respectively. For $X = \Omega \subseteq \mathbb{R}^n$ we write

$$f(\mathbf{x}) = g(\mathbf{x}) + \int_\Omega K(\mathbf{x},\mathbf{y})f(\mathbf{y})d\mathbf{y}, \quad \mathbf{x}, \mathbf{y} \in \Omega \subset \mathbb{R}^n.$$

3.2 Expansion in series of Singular Values and Singular Functions

In this section we will describe how to use the results of Chapter 1 to expand a function satisfying an integral equation of the first or second kind in a series by using singular values and singular functions . We deal with concrete cases of X which is going to be a compact interval in $\mathbb{R}$, a compact rectangle in $\mathbb{R}^2$ or a compact parallelepiped in $\mathbb{R}^3$.

In all cases we consider $L_2(X)$. So we consider the integral operator

$$\begin{aligned} T : L_2(X) &\rightarrow L_2(X) \\ f &\mapsto T(f)x = \int_X K(x,y)f(y)dy \end{aligned}$$

where $K \in L_2(X \times X)$. T is a compact operator over the Hilbert space $L_2(X)$.

We are mainly interested is solving integral equations of first kind. Nevertheless, we are going to describe how to solve integral equations of the second kind, specially of Fredholm type. These results will be used later.

The following can be found in [18] and [28]. Consider $g = Tf$, where T is a self-adjoint integral compact operator(this is the case when the kernel is symmetric). By Theorem 1.4.7 on page 18, $L_2(X)$ admits a orthonormal basis which consists only of eigenfunctions of the operator T. Let $\{u_n\}_{n=1}^{\infty}$ the eigenfunctions. Then g can be expanded in a Fourier series

$$g(x) = \sum_{n=1}^{\infty} a_n u_n(x) \text{ where } a_n = (g, u_n).$$

If the Kernel is expressible in a series of the form

$$K(x,y) = \sum_{m=1}^{\infty} c_m(y) u_m(x),$$

then multiplying $u_n(x)$ in both sides and taking integral we get

$$\int_X K(x,y)u_n(x)dx = \sum_{m=1}^{\infty} c_n(y) \int_X u_n(x)u_m(x)dx = c_n(y), \text{ so}$$

$$c_n(y) = \int_X K(x,y)u_n(x)dx = T(u_n)x = \lambda_n u_n(x).$$

Thus

$$K(x,y) = \sum_{n=1}^{\infty} \lambda_n u_n(x) u_n(y).$$

When K is not symmetric it is still possible to express g in a series of singular functions of a suitable symmetric operator. The equation $g = Tf$ is expressed as $T^*g = T^*Tf$. As a consequence of Fubini's theorem, T^*T and TT^* are symmetric compact operators, so $L_2(X)$ has a orthonormal basis of eigenfunctions of T^*T, say $\{v_n\}$, with eigenvalues $\{\lambda_n\}$ and a orthonormal basis of eigenfunctions of TT^*, say $\{u_n\}$. The singular values are $\{\sigma_n\}$. Thus

$$g(x) = \sum_{n=1}^{\infty}(g, u_n)u_n = \sum_{n=1}^{\infty}(Tf, u_n)u_n = \sum_{n=1}^{\infty}(f, T^*u_n)u_n$$

or simply

$$g(x) = \sum_{n=1}^{\infty} \sigma_n (f, v_n) u_n(x). \tag{3.5}$$

Here we used the relations in Section 1.3 and (1.5). In the same fashion $f(y)$ can be expanded as $f(y) = \sum_{n=1}^{\infty} (f, v_n) v_n(y)$. We put this expression in (3.4), taking g as in (3.5). Thus,

$$\sum_{n=1}^{\infty} (g, u_n) u_n = \sum_{n=1}^{\infty} (f, v_n) T v_n = \sum_{n=1}^{\infty} (f, v_n) \sigma_n u_n.$$

From here we get $(f, v_n) = \frac{(g, u_n)}{\sigma_n}$. Finally

$$f(x) = \sum_{n=1}^{\infty} \frac{(g, u_n)}{\sigma_n} v_n(x). \tag{3.6}$$

3.2.1 A Method for solving linear Fredholm integral equations of the second kind

Now we proceed to solve the Equation 3.3 in a formal way using expansion in Fourier series following [28]. So, consider the equation

$$f(x) = g(x) + \alpha \int_a^b K(x, y) f(y) dy, \quad K(x, y) = K(y, x), \alpha \neq 0.$$

This implies $(1/\alpha)(f - g)$ can be represented as Tf. Let $\{\phi_n\}$ be the eigenfunctions of the integral operator T. For a solution of Equation 3.3 we have

$$f - g = \sum_{n=1}^{\infty} a_n \phi_n(x), \quad a_n = (f - g, \phi_n) = (f, \phi_n) - (g, \phi_n).$$

Henceforth $(f, \phi_n) = (g, \phi_n)\left(\frac{1}{1-\alpha\lambda_n}\right)$. From this a_n can be expressed

$$a_n = (f - g, \phi_n) = (\alpha T f, \phi_n) = \alpha(f, T\phi_n) = \alpha\phi_n\lambda_n(f, \phi_n)$$

if T is a symmetric operator. Therefore,

$$a_n = (f - g, \phi_n) = \frac{\alpha\lambda_n}{1-\alpha\lambda_n}(g, \phi_n).$$

Thus we can get a representation of f in a series:

$$f(x) = g(x) + \sum_{n=1}^{\infty} \frac{\alpha\lambda_n}{1-\alpha\lambda_n}(g, \phi_n)\phi_n(x) \tag{3.7}$$

No solution f can exists if, for some j, $\alpha = \lambda_j^{-1}$ and $(g, \phi_j) \neq 0$. In a general way the Fredholm Alternative (Theorem 1.2.10) describes the situation. The Equation 3.3 has a unique solution for all $g \in L_2(X)$ or the homogeneous equation admits a finite number of linearly independent solutions, say k. In the latter case, the solution of (3.3) is not unique. For instance, if $(g, \phi_j) = 0$, $\alpha = \lambda_j$ and f is a solution, so is $f + C_j\phi_j$ for an arbitrary constant C_j. In fact, if λ_j has multiplicity k and $(g, \phi_{m+h}) = 0,\ h = 1, .., k$ then $f + \sum_{h=1}^{k} C_j\phi_{m+h}$ is also a solution.

3.2.2 A method to solve integral equations of the first kind

This section aims at solving the Equation 3.2 using the Regularization Method described in Section 2.3. We follow [28], [18] and [7]. In order to solve $g = Tf$ we take a initial guess, say f_0. Put $F = f - f_0$ and $G = g - Tf_0$. So

$$G = g - Tf_0 = Tf - Tf_0 = T(f - f_0) = TF,$$

this is, G satisfies an integral equation of the first kind. The associated functional is

$$\mathcal{F}(W, \alpha) = \|G - TW\|^2_{L_2(X)} + \alpha \|W\|^2_{L_2(X)}, \quad \alpha > 0.$$

The condition for the minimum of this functional, see Equation 2.5, leads to

$$W = \frac{T^*G}{\alpha} - \frac{1}{\alpha} T^*TW,$$

which is an integral equation of the second kind with unknown W. We can use the expression (3.7) with $-1/\alpha$ instead of α to get

$$W(x) = \frac{T^*G}{\alpha} - \sum_{n=1}^{\infty} \frac{\lambda_n^2}{\alpha + \lambda_n^2} \left(\frac{T^*G}{\alpha}, \phi_n \right) \phi_n(x).$$

Since $T^*G = T^*TF$, it is possible to write T^*G in a series of eigenfunctions $\{\phi_n\}$ of T^*T with eigenvalues λ_n^2, so

$$T^*G = \sum_{n=1}^{\infty} (T^*G, \phi_n) \phi_n(x)$$

thus

$$W_\alpha(x) = \sum_{n=1}^{\infty} \frac{(T^*G, \phi_n)}{\alpha + \lambda_n^2} \phi_n(x).$$

Using the properties of the adjoint operator and singular values and singular functions (Section 1.3), we set $(T^*G, \phi_n) = (G, T_n\phi), \quad \sigma_n = \lambda_n, \quad \phi_n = v_n, \quad , T\phi_n = Tv_n = \sigma_n u_n$.

A expression for W is

$$W_\alpha(x) = \sum_{n=1}^{\infty} \frac{\sigma_n (G, u_n)}{\alpha + \sigma_n^2} v_n(x).$$

Therefore, $\lim\limits_{\alpha\to 0^+} W_\alpha(x) = \sum_{n=1}^{\infty} \frac{(G,u_n)}{\sigma_n} v_n(x)$ which is precisely the expansion explained in (3.6). Thus

$$F(x) = \lim_{\alpha\to 0^+} W_\alpha(x).$$

Finally we obtain a expression for f, $f = F + f_0$

$$f(x) = \sum_{n=1}^{\infty} \frac{(G,u_n)}{\sigma_n} v_n(x) + f_0 \tag{3.8}$$

If G is in error we replace G by $G_c + \delta$, where G_c denotes the correct solution, then the solution obtained is

$$f(x) = \sum_{n=1}^{\infty} \sigma_n \frac{(G_c,u_n)}{\alpha + \sigma_n^2} v_n(x) + \sum_{n=1}^{\infty} \sigma_n \frac{(\delta,u_n)}{\alpha + \sigma_n^2} v_n(x),$$

and the error in the solution is

$$E(x) = \sum_{n=1}^{\infty} \frac{\lambda_n}{\alpha + \lambda_n^2} (\epsilon, u_n) v_n(x). \tag{3.9}$$

To reduce the error due to regularization, α must be taken small. Also α must be taken large enough in order to reduce the error $E(x)$ due to the error in the data g.

For a fixed α, the maximum for the function $\dfrac{x}{(x^2+\alpha)}$ is $\frac{1}{2\alpha^{1/2}}$, so $d \quad \frac{\lambda_n}{\alpha+\lambda_n^2} \leq \dfrac{1}{2\sqrt{\alpha}}$ and $\|E\|_{L_2} \leq \dfrac{1}{2\sqrt{\alpha}} \|\delta\|_{L_2}$. If we choose α such that

$$\frac{1}{2\sqrt{\alpha}} \|\delta\|_{L_2} < 1$$

a safe choice of α to keep E small would be $\alpha > \frac{\|\delta\|_{L_2}^2}{4}$. The Theorem 2.4.3 gives a criterion to choose an optimum α for a fixed δ.

3.3 Numerical Methods to solve Integral Equations

The purpose here is to solve integral equations of the first kind by using numerical approximations. We are going to solve Fredholm integral equations in two and three dimensions. We assume that kernels have no singularities.

$$\begin{aligned} g(x) &= \int_a^b K(x,y)f(y)dy & (3.10) \\ g(\mathbf{x}) &= \iint_\Omega K(\mathbf{x},\mathbf{y})f(\mathbf{y})d\mathbf{y},\ \Omega = [a,b]\times[c,d] & (3.11) \\ g(\mathbf{x}) &= \iiint_\Omega K(\mathbf{x},\mathbf{y})f(\mathbf{y})d\mathbf{y},\ \Omega = [a,b]\times[c,d]\times[e,f]. & (3.12) \end{aligned}$$

In order to solve the equations, we must compute the singular values and singular functions of the operator in the right-hand side of the above equations. It can be done by replacing integrals using a quadrature scheme such that the integral kernel is now represented by a matrix. The eigenvalues, singular vectors and singular values of this matrix give the approximated values for those items related to the integral operator. Thus if we use the regularization method with an initial data f_0 and $F = f - f_0$, the solution will be given by a sum like

$$f \approx f_0 + \sum_{i=1}^{n} \frac{\sigma_i(G,u_i)}{\alpha + \sigma_i^2} v_i$$

which is an approximation of (3.8).

3.3.1 Solution to one-dimensional Fredholm integral equation of the first kind

This topic is well documented. We can get many available programs written in different programming languages such as C, MATLAB and FORTRAN. For $x_1, x_2, \cdots, x_n$

$$T(f)x_i = \int_a^b K(x_i, y)dy \approx \sum_{j=1}^{n} K(x_i, y_j)f(y_j)w_j, \qquad \text{with weights } w_j$$

Put $k_{ij} = K(x_i, y_j)w_j$, $\mathbf{K} = (k_{ij})$, $f_j = f(y_j)$, $g_i = g(x_i)$, $\mathbf{f} = [f_1, \cdots, f_n]^t$, $\mathbf{g} = [g_1, \cdots, g_n]^t$. Thus $\mathbf{T} = \mathbf{Kf}$ is the matricial representation of the operator T.

Gutiérrez [7], following [28] solved the equation $g = Kf$. He developed a code in MATLAB to solve it by using the expansion in the series (3.7) which uses singular values and functions . The parameter α was chosen by trial and error, in spite of some algorithms to choose it.

Other interesting code to solve integral equation of the first kind, developed in FORTRAN, is found in [25]. There, the method of generalized discrepancy (2.4.3) is utilized to choose the best α. In contrast with Gutierrez, the solution was computed by solving directly the equation of the second kind (2.5) which gives the minimum for the Tikhonov functional.

For solving one-dimension integral equations of the first kind there are many available programs ready to use where different techniques are used.

3.3.2 Solution to two-dimension Fredholm integral equation of the first kind

The main objective here is to generalize the Gutierrez's method for integral equation of the first kind in one dimension to two-dimension integral equations of the first kind. It is not very common to find programs which solve this type of equations in two dimensions. Xiao [29] uses fast convergent algorithms so solve such equations.

The program that we are going to use is similar to the Gutierrez's program, i.e, the sum (3.7) is used. The first step is to represent the kernel of the integral equation by a bi-dimensional matrix, after that the singular value decomposition is done and finally we build the approximated solution by using (3.7).

Let be $K(\mathbf{x}, \mathbf{y})$ the kernel of the integral equation, where $\mathbf{x} = (u, v), \quad \mathbf{y} = (x, y)$, $u, v, x, y \in \mathbb{R}$. Let g be the data function and f the unknown function, the function which has to be found. Moreover, suppose that g is given in the grid in Figure 3.1 with an error and we know g_δ such that $\|g - g_\delta\| \leq \delta$. This norm is approximated by

$$\begin{aligned} \|g - g_\delta\|^2 &= \int_c^d \int_a^b \big(g(u, v) - g_\delta(u, v)\big)^2 du dv \\ &\approx \sum_{q=1}^{qmax} \sum_{p=1}^{pmax} \big(g(u_p, v_q) - g_\delta(u_p, v_q)\big)^2 w_p w_q^{'}. \end{aligned}$$

Here w_p and $w_q^{'}$ are weights of the quadrature used. The functions f and g are represented by the matrices $\mathbf{f}$ and $\mathbf{g}$ whose orders are $(imax)(jmax) \times 1$,

$(umax)(vmax) \times 1$ respectively.

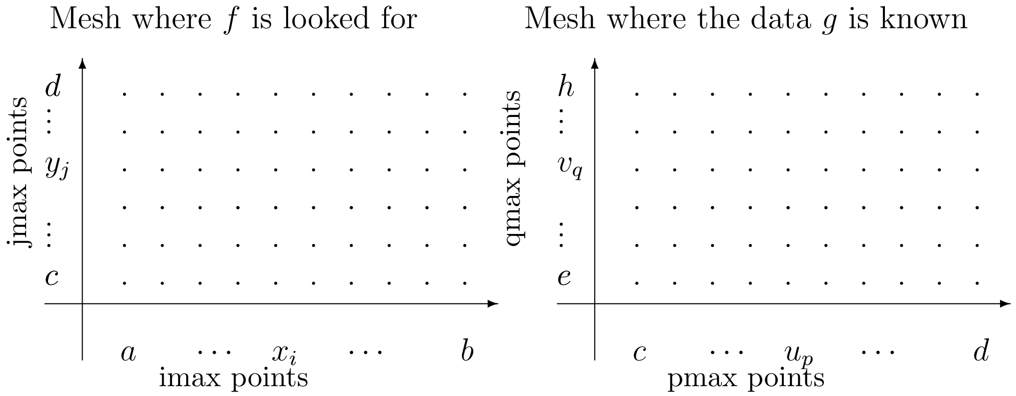


Figure 3.1: Grids used to approach functions f, K, and g. Case 2D

The operator T can be approximated by

$$\begin{aligned} T(f)(u_p, v_q) &= \int_c^d \int_a^b K(u_p, v_q, x, y) dx dy \\ &\approx \sum_{j=1}^{jmax} \sum_{i=1}^{imax} K(u_p, v_q, x_i, y_j) w_i w'_j \\ p = 1, \cdots, pmax & \qquad q = 1, \cdots, qmax. \end{aligned}$$

So, using the grids in Figure 3.1, we have the expression for $T(f)$

$$\begin{aligned} T(f)(u_p, v_q) &= [K(u_p, v_q, x_1, y_1) f(x_1, y_1) w_1 w'_1 + \cdots \\ &+ K(u_p, v_q, x_{imax}, y_1) f(x_{imax}, y_1) w_{imax} w'_1] + \cdots \\ &+ \big[K(u_p, v_q, x_1, y_{jmax}) f(x_1, y_{jmax}) w_1 w'_{jmax} + \cdots \\ &+ K(u_p, v_q, x_{imax}, y_{jmax}) f(x_{imax}, y_{jmax}) w_{imax} w'_{jmax}\big] \end{aligned}$$

The matrix $\mathbf{T}$ that approximates the kernel has order $(pmax)(qmax) \times (imax)(jmax)$, the element in the position $\big(pmax(q-1)+p, imax(j-1)+i\big)$ is given by $K(u_p, v_q, x_i, y_j)w_i w'_j$.

In Section A.1 we described the program that we use to do the computations for the two-dimensional case. As is usual in these cases, we are going to solve some test problems with known solutions.

Example 3.

We solved the Equation 3.11 with $g(x,y) = \frac{5000}{3}(x+y)$, kernel $K(u,v,x,y) = (u+v)(x+y)$, and $\Omega = [-5,5] \times [-5,5]$. The exact solution is $f(x,y) = x+y$ as we can verify by doing

$$\int_{-5}^{5}\int_{-5}^{5}(u+v)(x+y)(x+y)dxdy = \frac{5000}{3}(u+v).$$

Now we will compute the numerical solution. Put $a = c = -5, b = d = 5$ and used the same grid for f and g with $n = imax = jmax$. In Figure 3.2 we have the exact solution and the approximated solution computed with the method of the L-curve for $\delta = 10^{-3}$ and $n = 60$.

We solve the problem for different values of n. Table 3.1 show values of n, the regularization parameter, residual norm given by L-curve, norm of approximated solution, and the relative error. The solution gets better when n increases. The relative error of the best solution is 29.94%. Figures 3.4, 3.6, 3.8 show others solutions. Their respective L-curves are in Figures 3.3,

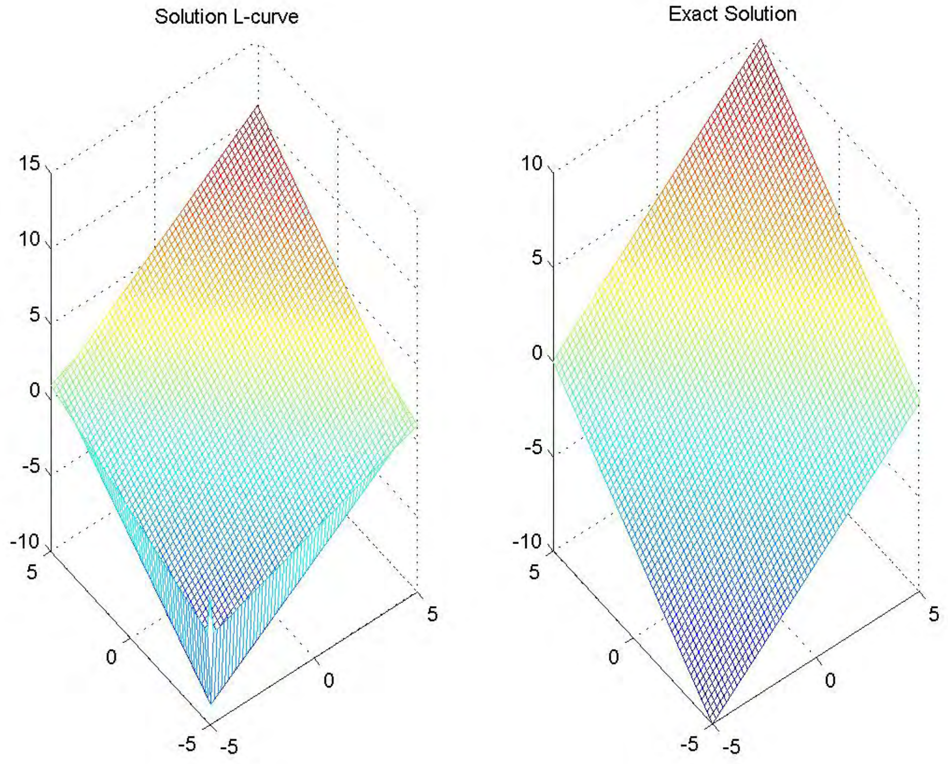


Figure 3.2: Solutions of Example 3, with $f_0(x, y) = 0$ and $n = 60$.

3.5 and 3.7. The function `l_curve` [11] was not able to draw an L-curve for n=40,50,60.

n	α	$\|Az_\alpha - g\|$	$\|z_\alpha\|$	$\frac{\|z_\alpha - \bar{z}\|}{\|z_\alpha\|}$
10	1667.6087	10000.0681	42.0036	0.43692
20	1585.2079	20000.1364	84.0353	0.37311
30	1559.0615	30000.2045	126.0443	0.34061
40	1546.1678	40000.2727	168.0527	0.32131
50	1538.4825	50000.3408	210.0614	0.30852
60	1533.3788	60000.409	252.0705	0.29942

Table 3.1: Some significant quantities for Example 3

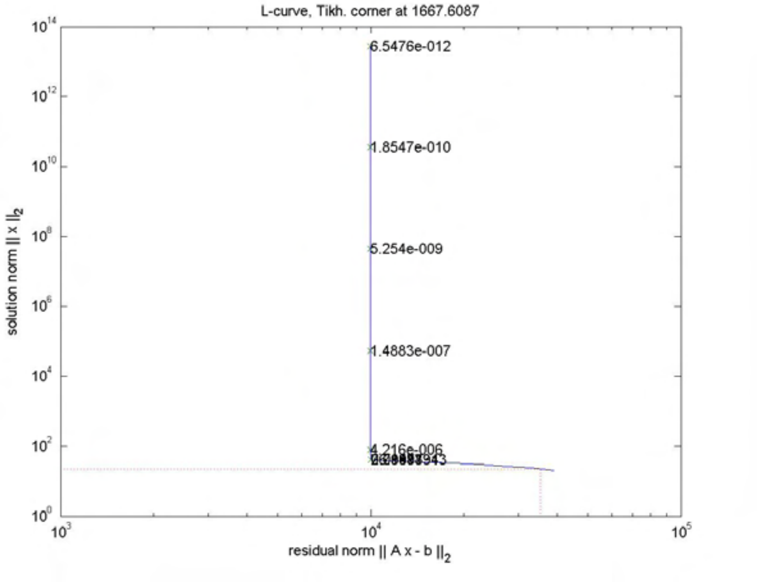


Figure 3.3: L-curve for n=10, Example 3.

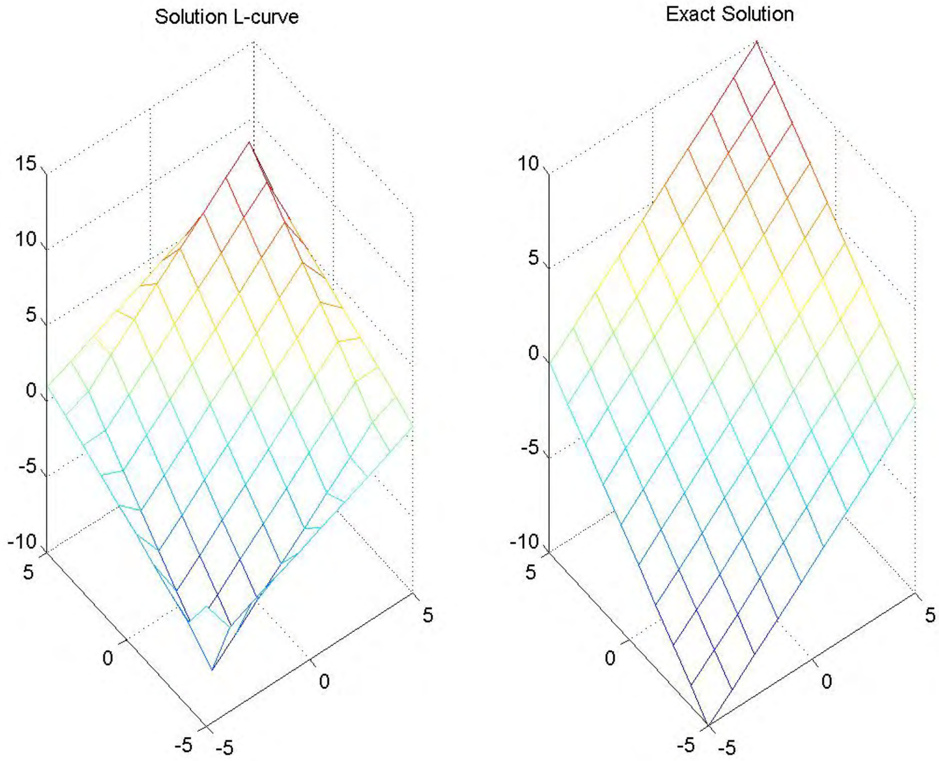


Figure 3.4: Approximated solution, $f_0(x, y) = 0$, n=10, Example 3.

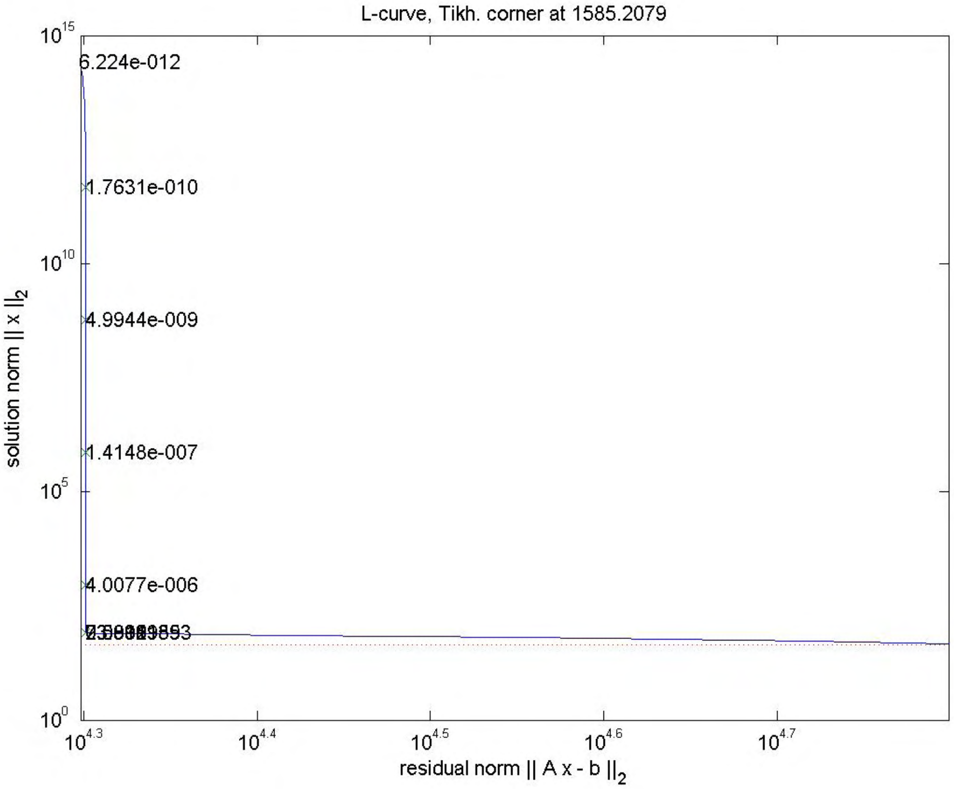


Figure 3.5: L-curve for n=20, Example 3.

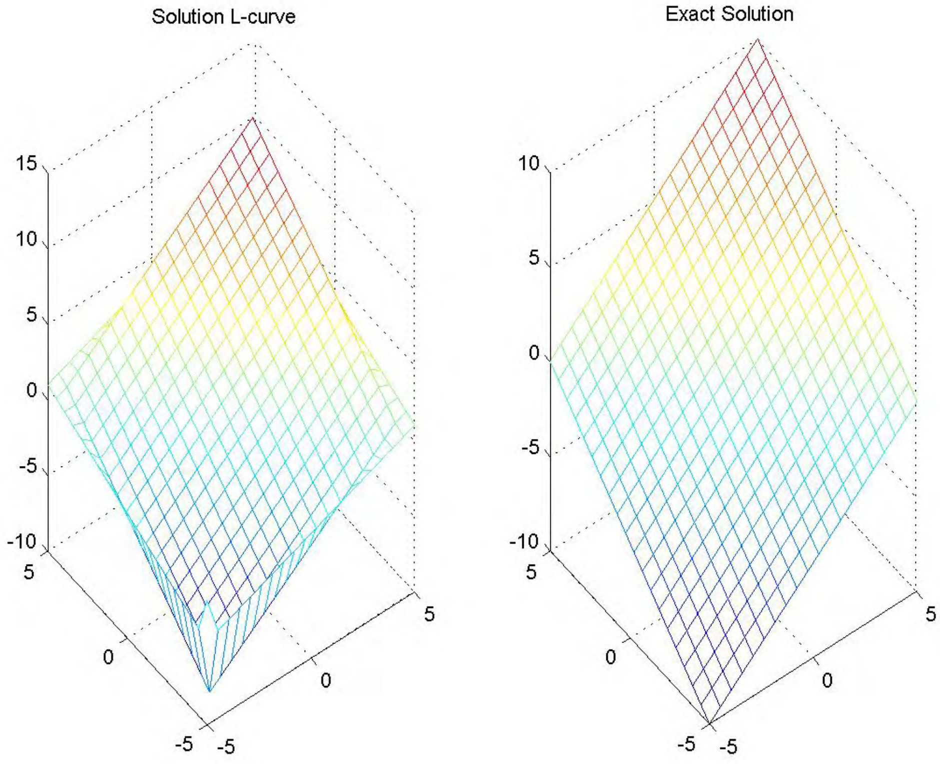


Figure 3.6: Approximated solution, $f_0(x, y) = 0$, n=20, Example 3.

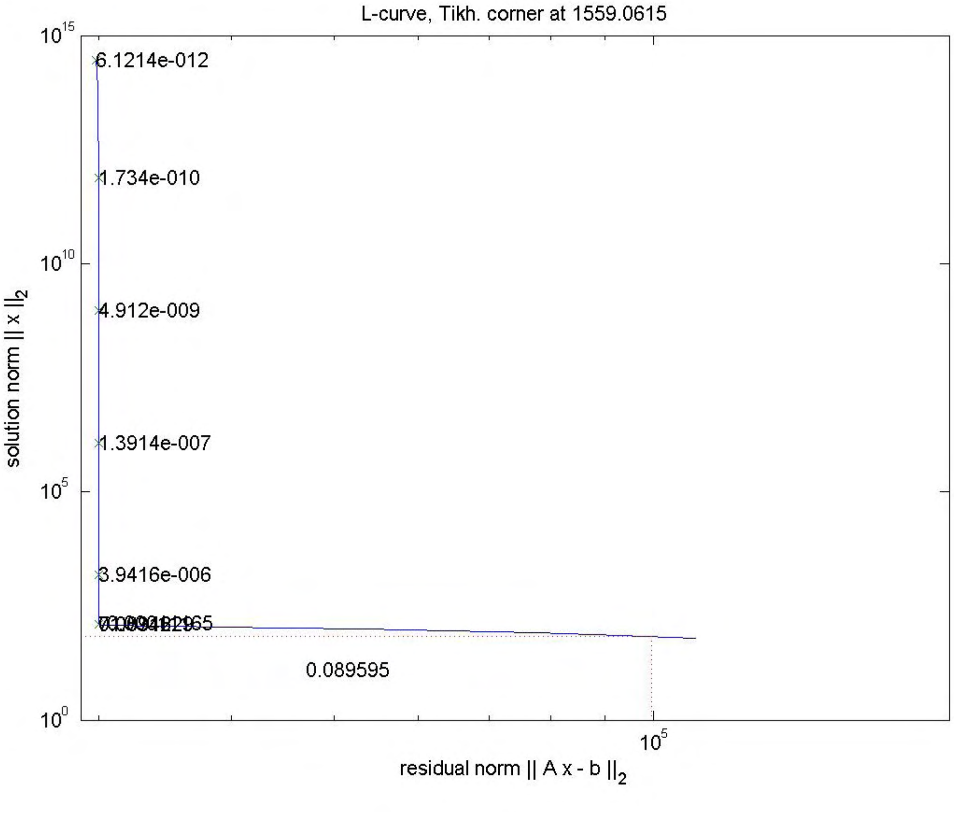


Figure 3.7: L-curve for n=30, Example 3.

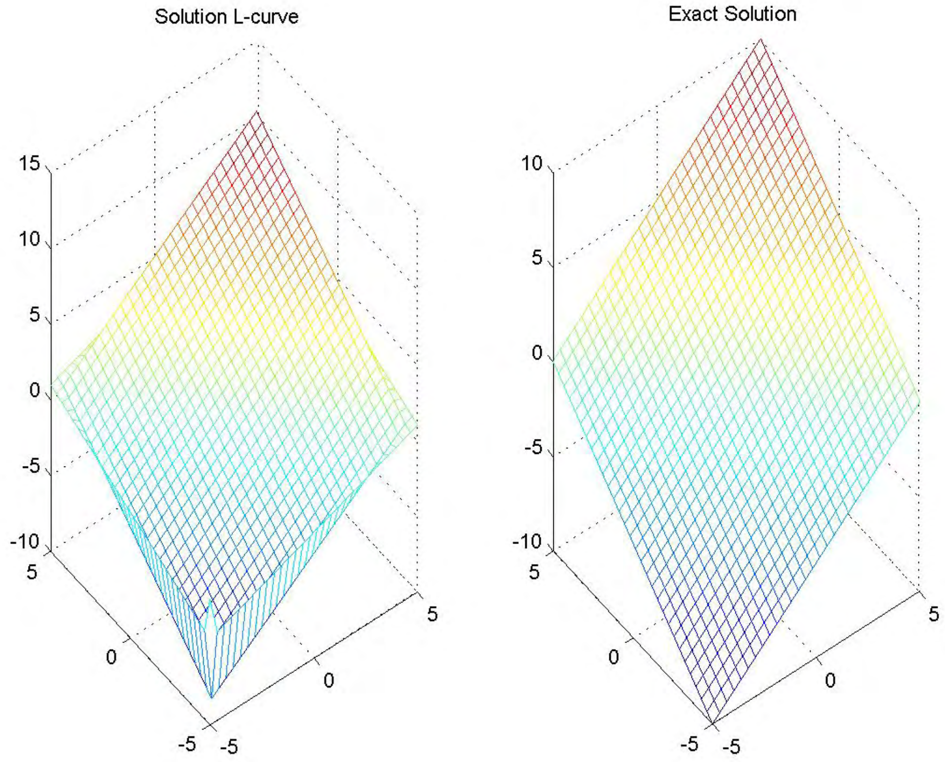


Figure 3.8: Approximated solution, $f_0(x, y) = 0$, n=30, Example 3.

Example 4.

In this example we keep the function $f(x,y) = x + y$, but we change the kernel. Now

$$K(u,v,x,y) = \frac{u}{1 + \sqrt{(x-u)^2 + (y-v)^2}},$$

and $\Omega = [-5,5] \times [-5,5]$. This time the data function g can not be calculated directly, instead we use the approximation $\mathbf{g} = \mathbf{Tf}$. We computed the numerical solutions and the results are summarized in Tables 3.2 and 3.3. The program gives better results when the parameter of regularization is computed with the L-curve.

n	α	$\lVert Az_\alpha - g \rVert$	$\lVert z_\alpha \rVert$	$\frac{\lVert z_\alpha - \bar{z} \rVert}{\lVert z_\alpha \rVert}$
10	0.16005	20399904.3604	44.4571	0.071712
20	0.010032	32107782.0407	85.0228	0.068475
30	0.0019668	44204436.8668	125.6645	0.066177
40	0.00061694	56410404.4078	166.3251	0.064659
50	0.00025081	68661540.5549	206.9986	0.063764
60	0.00012021	80935592.0947	247.6853	0.063274

Table 3.2: Some significant quantities for Example 4. L-curve Method. $f_0 = 1$. Case 2D

We got better results for this kernel. The relative error in Table 3.1 are over 29.94% and for same values of n, the relative errors in Table 3.2 are under 7.17%. This is due to the fact that the kernel on Example 3 is smoother than the Kernel in Example 4. In Figure 3.10 we can see some approximated solution of the Example 4. The residuals , $f^{exact} - f^{app}$, can

n	α	$\|Az_\alpha - g\|$	$\|z_\alpha\|$	$\frac{\|z_\alpha - \bar{z}\|}{\|z_\alpha\|}$
10	1.0111e-05	2.04e+07	89.533	0.98495
20	1.7101e-06	3.2108e+07	170.76	0.99041
30	5.9478e-07	4.4205e+07	247.04	0.96314
40	2.7392e-07	5.6411e+07	328.55	0.97231
50	1.4224e-07	6.8663e+07	410.15	0.97797
60	7.8849e-08	8.0937e+07	491.79	0.98178

Table 3.3: Some significant quantities for Example 4. Discrepancy Principle. $f_0 = 1$Case 2D

be seen in Figure 3.11 and more details can de found in Table B.1.

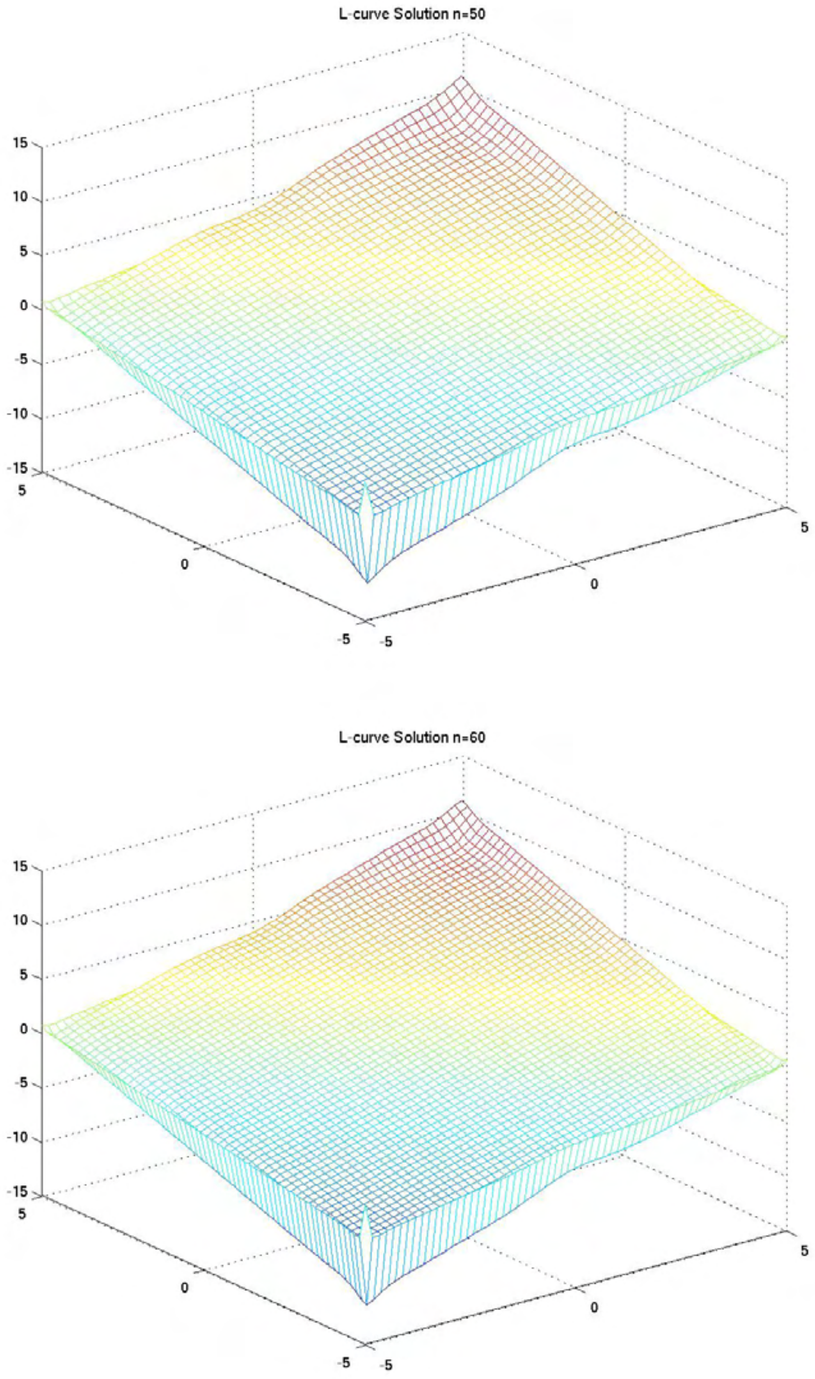


Figure 3.9: Approximated solutions for Example 4. L-curve method

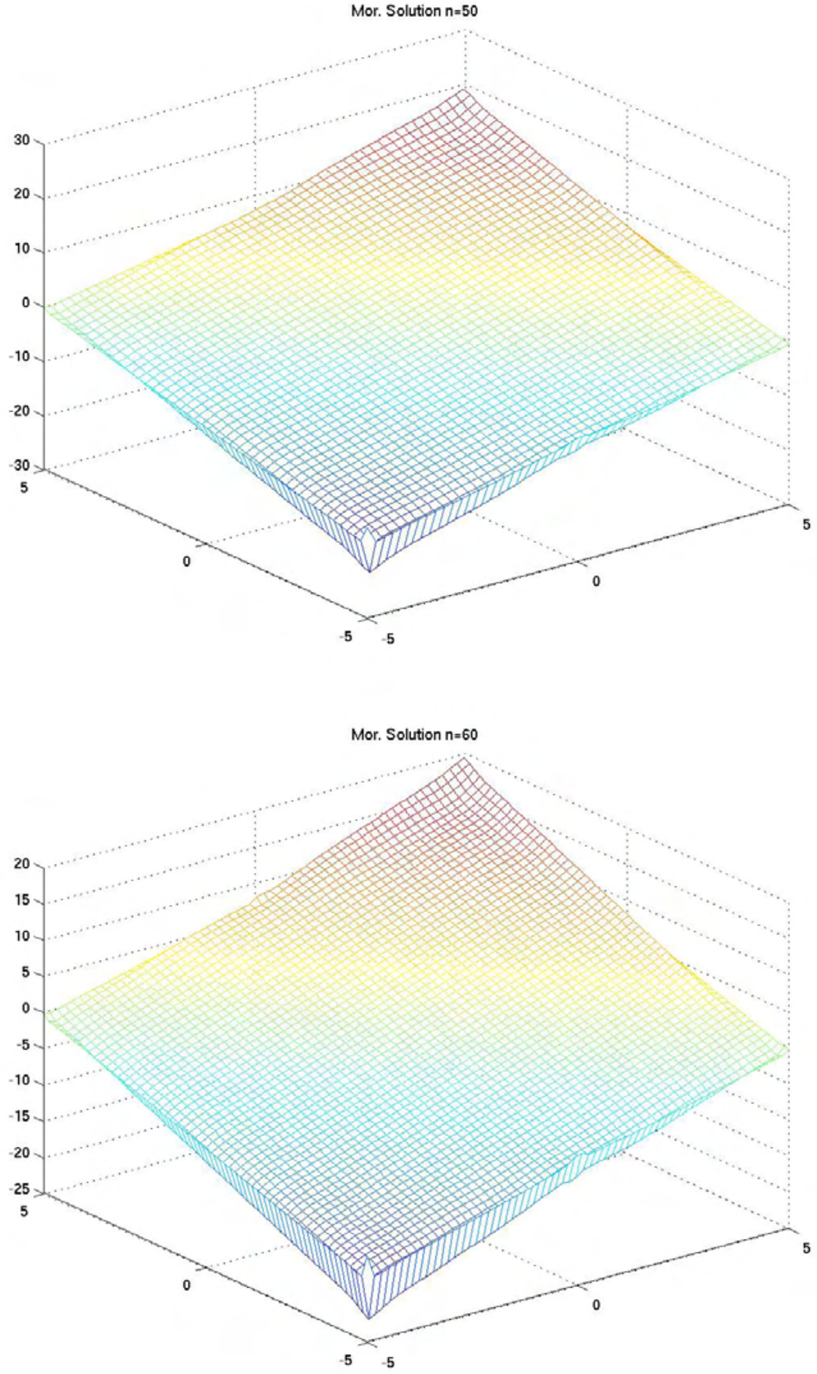


Figure 3.10: Approximated solutions for Example 4. Discrepancy Principle

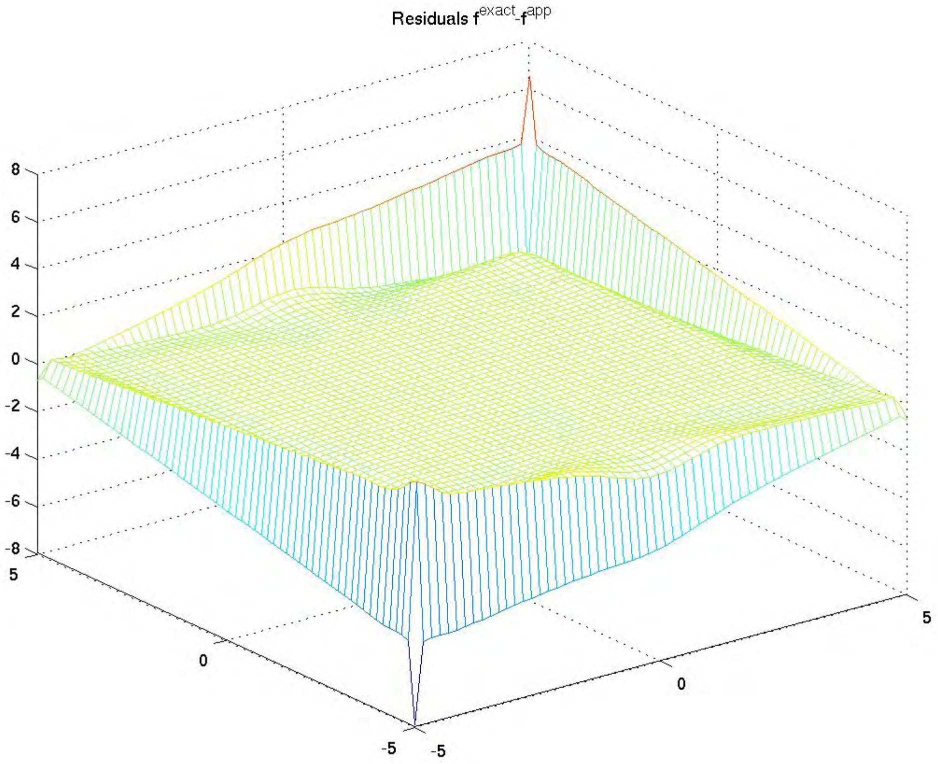


Figure 3.11: Residuals for Example 4. $f_0 = 1$, $n = 60$.

Example 5.

As a third example for testing the program we solve Equation 3.11 with kernel $K(u, v, x, y) = (x^2 + y)e^{u-v}$ and data function $g(x, y) = -\frac{50,000}{9}e^{x-y}$. The exact solution for this example is $f(x, y) = y^2 - x^2$ and $\Omega = [-5, 5] \times [-5, 5]$. We took the same grid for f and g with $n = imax = jmax$. We can see the approximated solution and exact solutions in Figures 3.12, 3.14, 3.16 and 3.18 with its respective L-curves in Figures 3.13, 3.15, 3.17, and 3.19. The graphics give a rough idea that the method is not giving good solutions for this problem. In Table 3.4 we can see that the values of the residual norms are very high they increase as n increases according to the explanations in Section 2.4.2. We encounter high relative error because both the kernel and the exact solution are smooth(see Section 2.4.3). We can see a complete list of numerical values in the Table B.2. The discrepancy principle was not able to compute the regularization parameter beyond $n = 30$ (See Table 3.5) because the values were zero to machine precision. The values computed by both methods are completely different. They are high in the first case and low in the second one. The residuals can be seen in Figure 3.20.

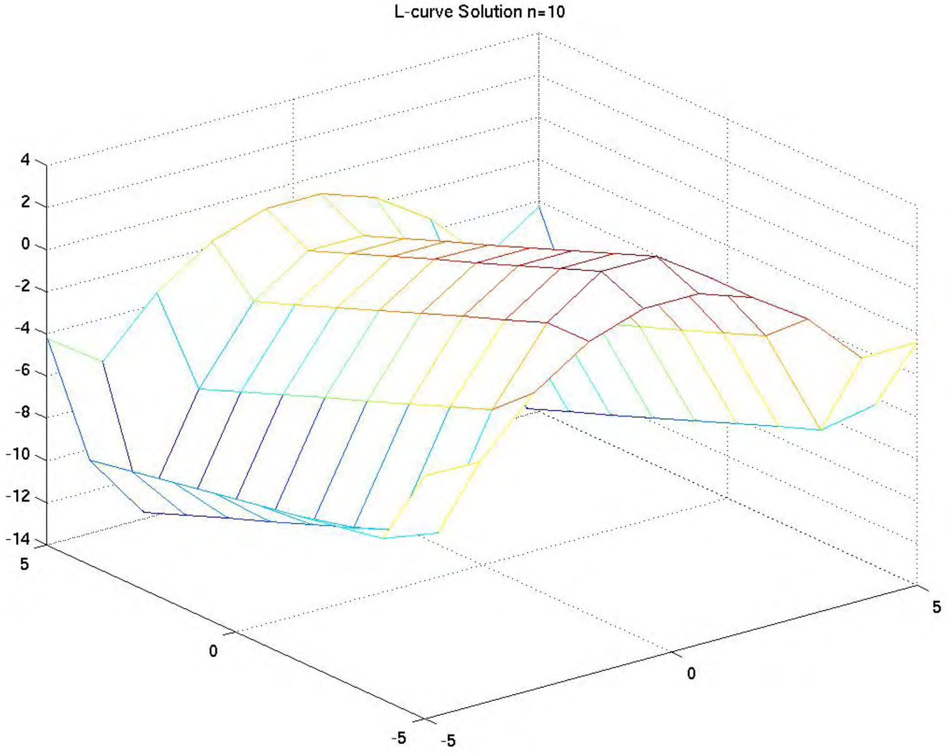


Figure 3.12: Approximated solution $f_0(x, y) = 1$, $n = 10$, Example 5.

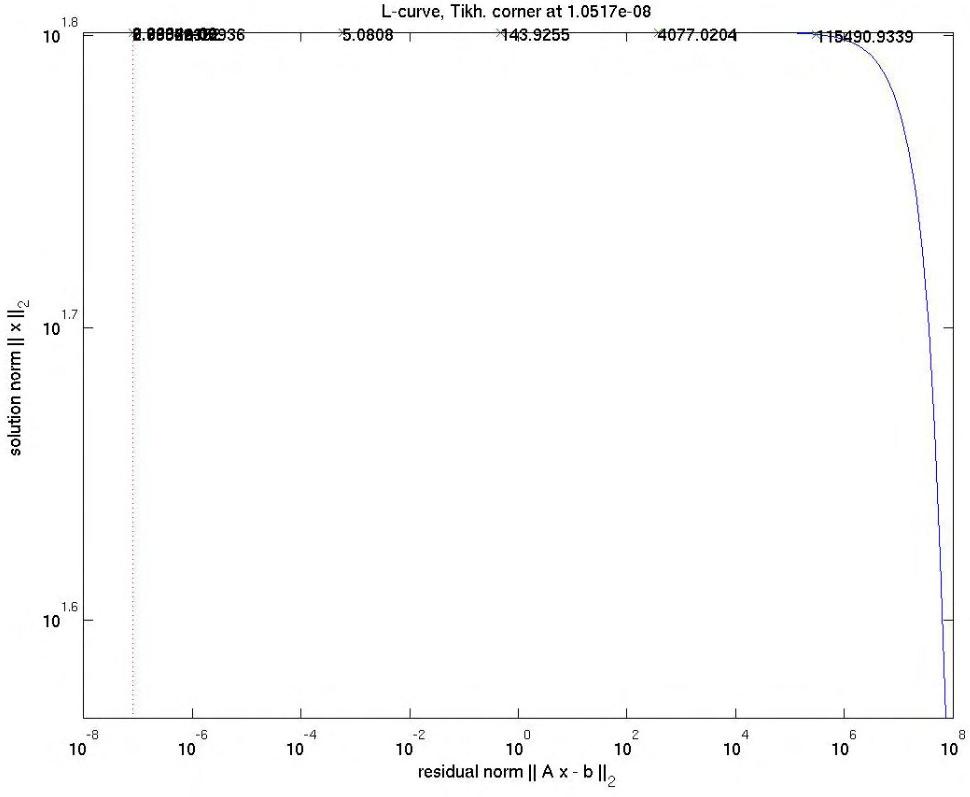


Figure 3.13: L-curve for $n = 10$, Example 5.

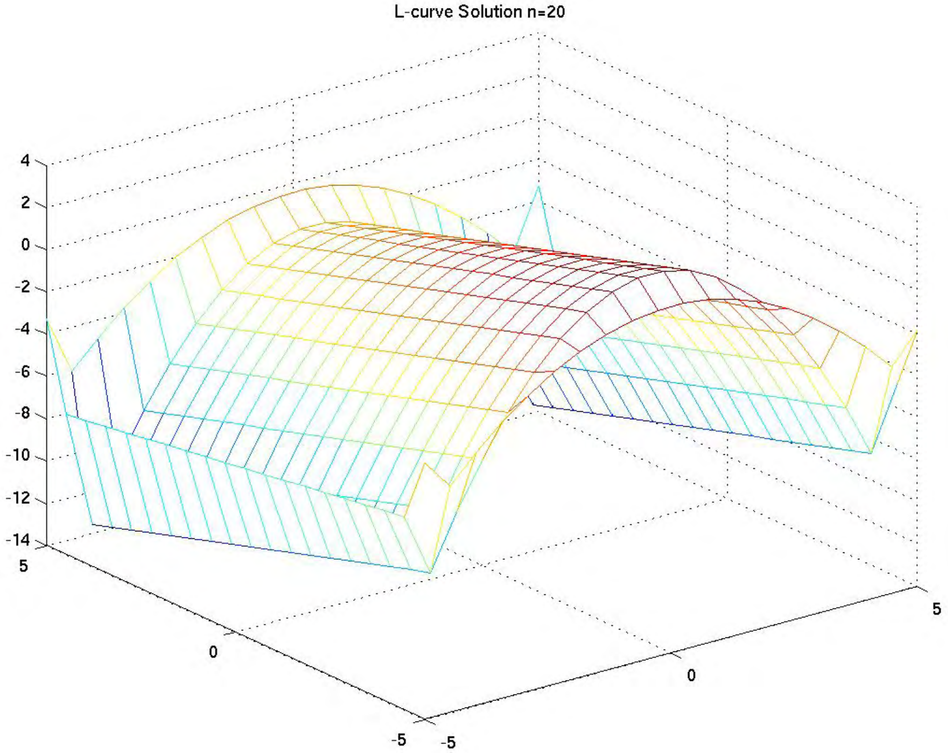


Figure 3.14: Approximated solution $f_0(x, y) = 1$, $n = 20$, Example 5.

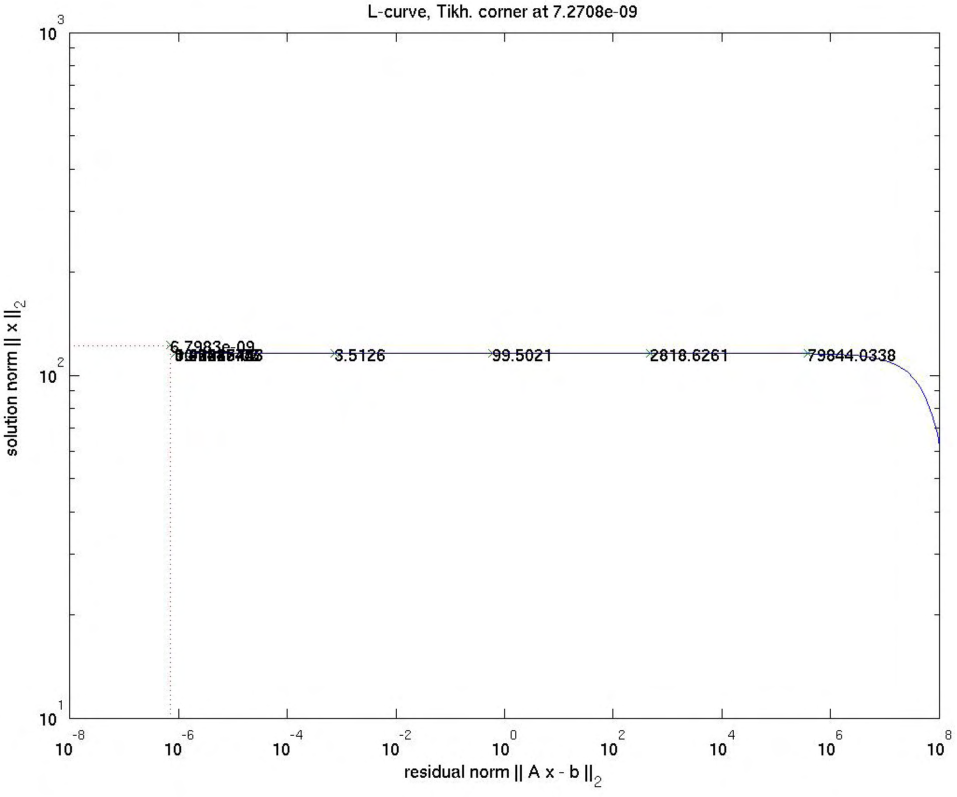


Figure 3.15: L-curve for $n = 20$, Example 5.

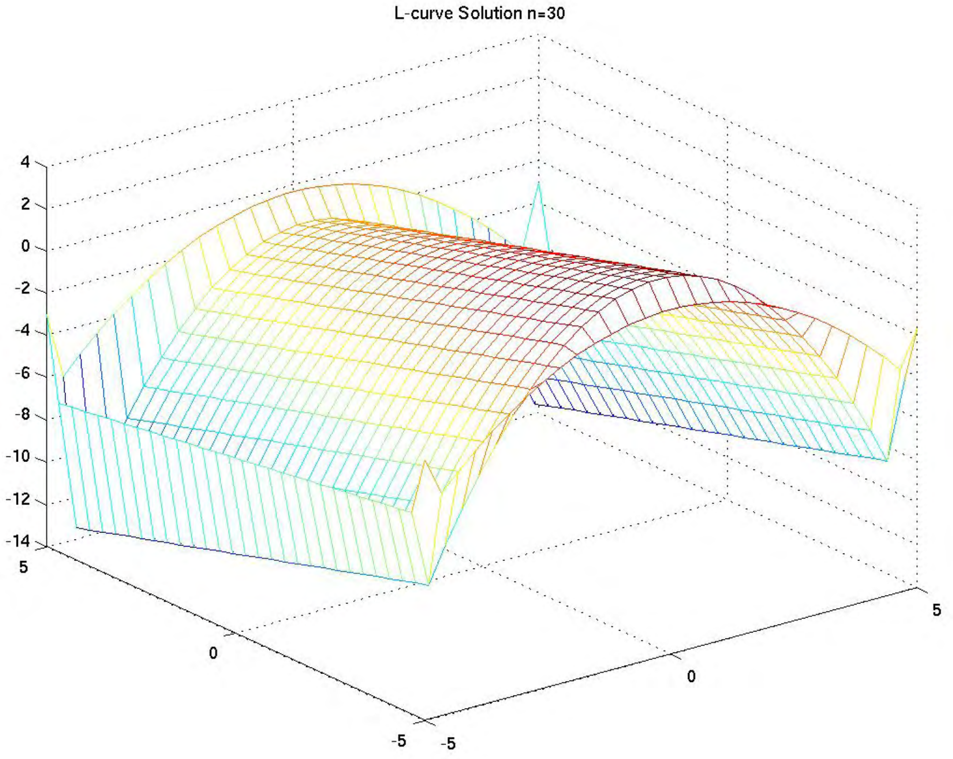


Figure 3.16: Approximated solution $f_0(x, y) = 1$, $n = 30$, Example 5.

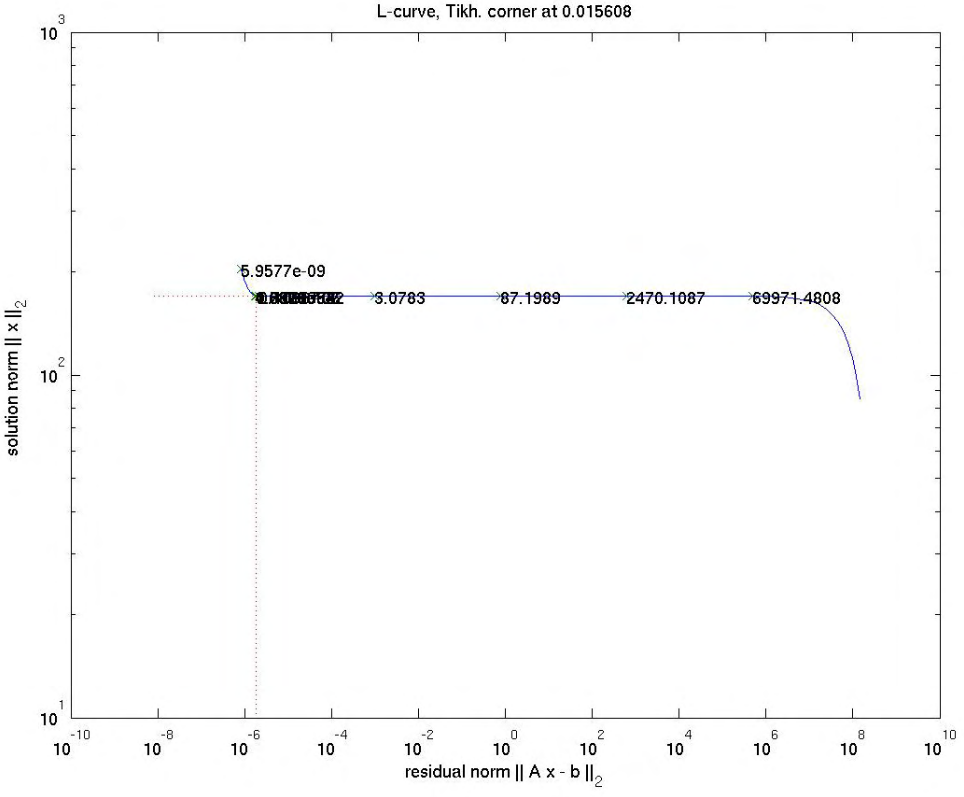


Figure 3.17: L-curve for $n = 30$, Example 5.

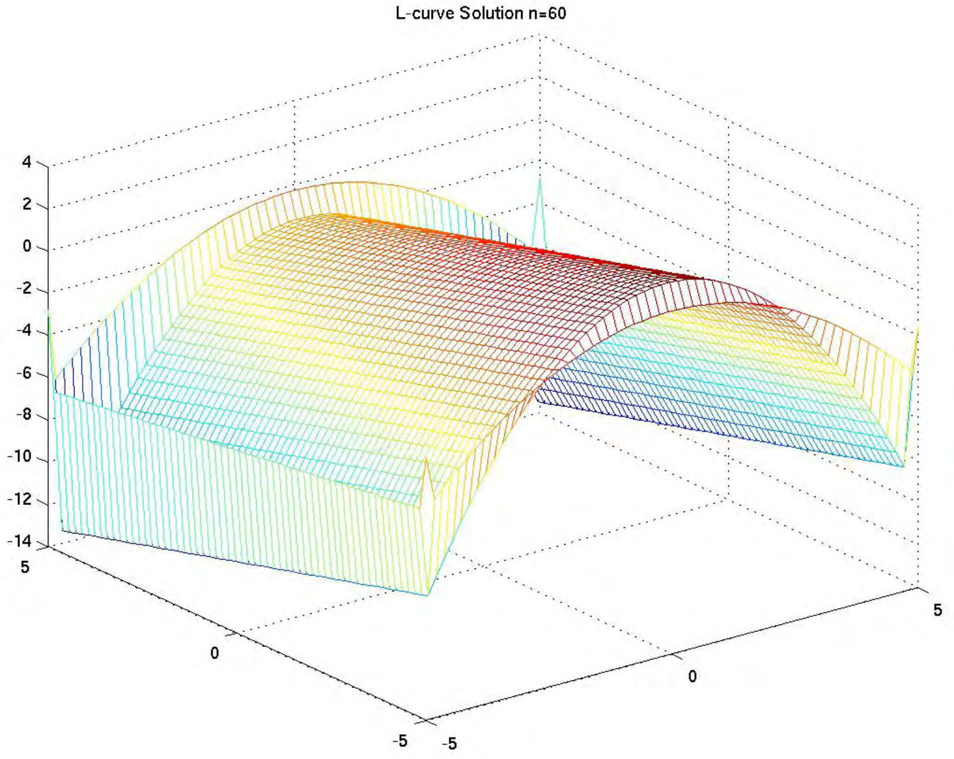


Figure 3.18: Approximated solution $f_0(x, y) = 1$, $n = 60$, Example 5.

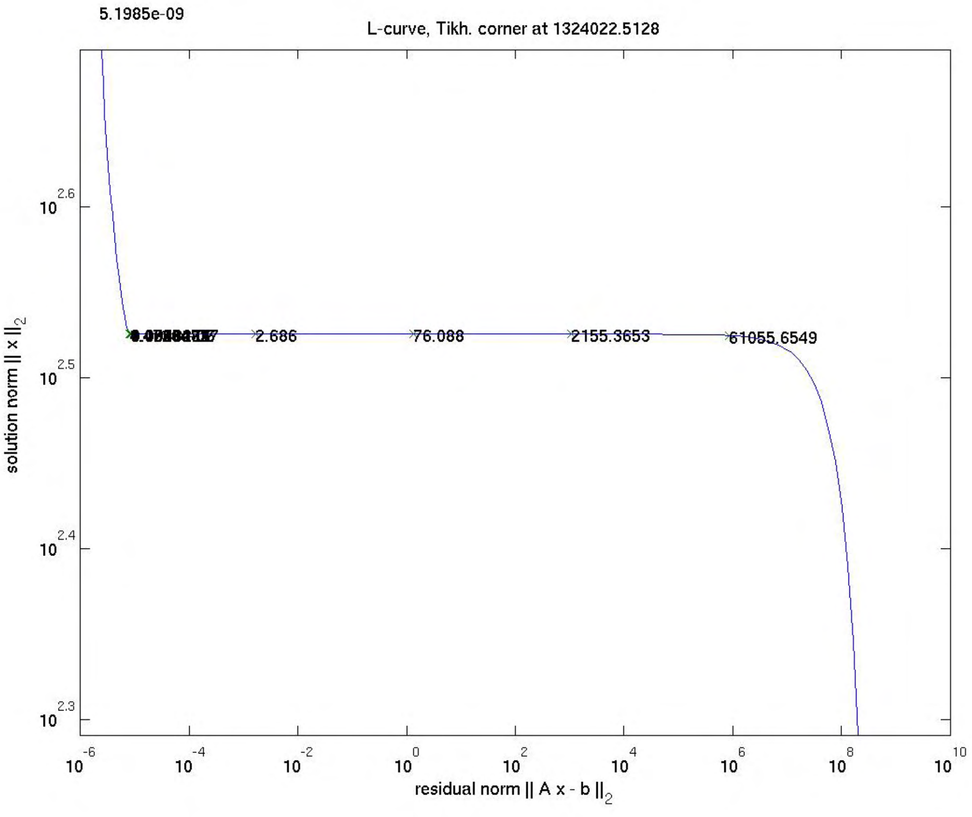


Figure 3.19: L-curve for $n = 60$, Example 5.

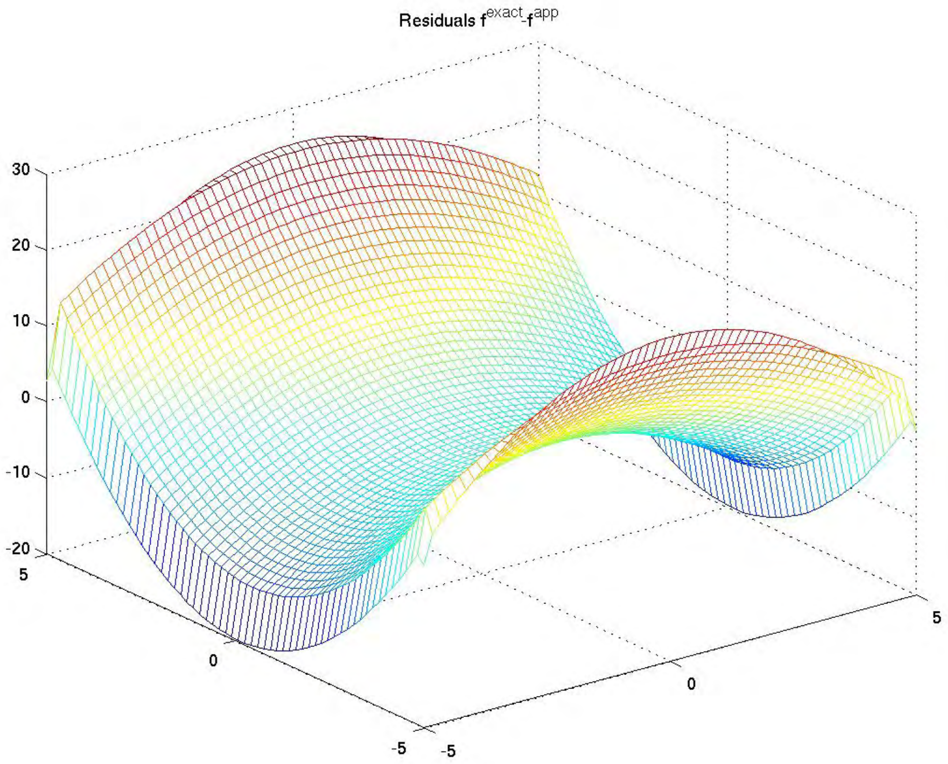


Figure 3.20: Residual for Example 5, $f_0 = 1$, $n = 60$.

n	α	$\|Az_\alpha - g\|$	$\|z_\alpha\|$	$\frac{\|z_\alpha - \bar{z}\|}{\|z_\alpha\|}$
10	496.5204	167397407.2	55.9982	0.87046
20	426.1497	214445602.8	101.8378	0.87105
30	1517366.672	277949675.8	149.5894	0.86804
40	1418143.192	344686997	197.7848	0.86587
50	1361064.37	412599052.3	246.1505	0.86435
60	1324022.513	481067923.3	294.5991	0.86323

Table 3.4: Some significant quantities for Example 5. L-curve method.

n	α	$\|Az_\alpha - g\|$	$\|z_\alpha\|$	$\frac{\|z_\alpha - \bar{z}\|}{\|z_\alpha\|}$
10	1.1075e-98	6.3017e+27	4.5739e+38	3.6053e+36
20	3.9927e-112	9.6515e+33	7.9234e+43	3.4133e+41
30	9.5882e-151	1.4186e+54	2.5954e+64	7.6906e+61

Table 3.5: Some significant quantities for Example 5. Discrepancy Principle

Example 6.

Using the same kernel as in Example 4,

$$K(u,v,x,y) = \frac{u}{1+\sqrt{(x-u)^2+(y-v)^2}},$$

and $\Omega = [-5,5] \times [-5,5]$ we try to solve Equation 3.11. The test function is $f(x,y) = y^2 - x^2$. Again g can not be computed exactly. So, an approximated data function $\mathbf{g} = \mathbf{Tf}$ works as input for the problem. Tables 3.6 and 3.5 summarize some important quantities like the size of the grid, the regularization parameter, residual norm, norm of the approximated solution and the relative error. In this case the relative errors are substantially smaller than those in Table 3.4. Values of the exact and approximated solutions are displayed in Table B.3. Residuals can be seen in Figure 3.24.

n	α	$\|Az_\alpha - g\|$	$\|z_\alpha\|$	$\frac{\|z_\alpha - \bar{z}\|}{\|z_\alpha\|}$
10	0.16005	20399627.4675	121.8057	0.14117
20	0.010032	32107265.7223	227.0518	0.11654
30	0.0019668	44203668.8915	331.9676	0.10684
40	0.00061694	56409382.9004	436.8627	0.10113
50	0.00025081	68660264.9581	541.7847	0.097212
60	0.00012021	80934062.1857	646.746	0.094292

Table 3.6: Some significant quantities for Example 6. Case 2D

In Figure 3.21 we can see some approximated solution of the Example 6.

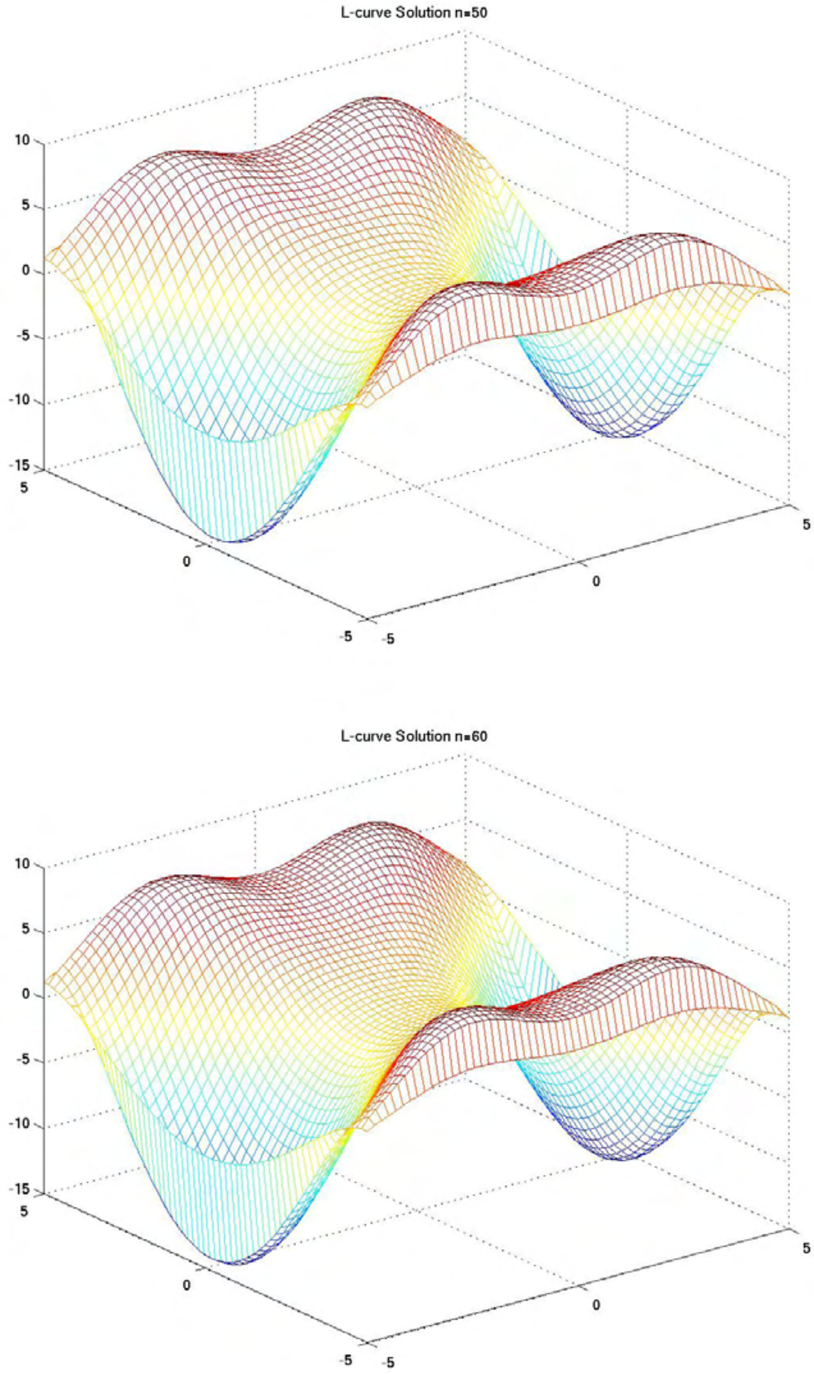
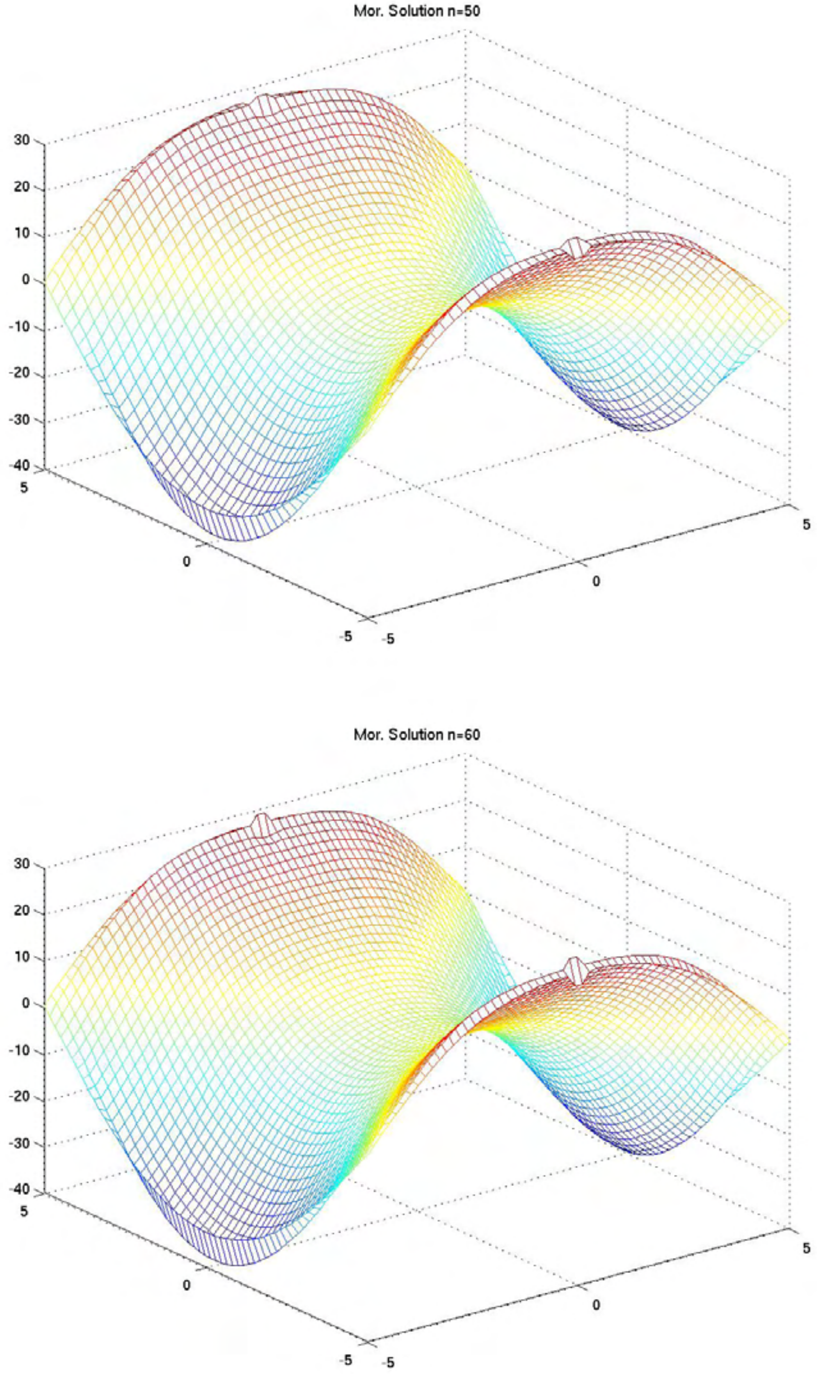


Figure 3.21: Approximated solutions for Example 6. L-curve method


Mor. Solution n=50

Mor. Solution n=60

Figure 3.22: Approximated solutions for Example 6. Discrepancy Principle

n	α	$\|Az_\alpha - g\|$	$\|z_\alpha\|$	$\frac{\|z_\alpha - \bar{z}\|}{\|z_\alpha\|}$
10	1.0111e-05	2.04e+07	185.57	0.51573
20	1.7101e-06	3.2108e+07	340.26	0.52229
30	5.9478e-07	4.4204e+07	500.4	0.53421
40	2.7392e-07	5.641e+07	661.39	0.5415
50	1.4224e-07	6.8662e+07	822.66	0.54626
60	7.8849e-08	8.0936e+07	984.06	0.54958

Table 3.7: Some significant quantities for Example 6. Discrepancy Principle. Case 2D

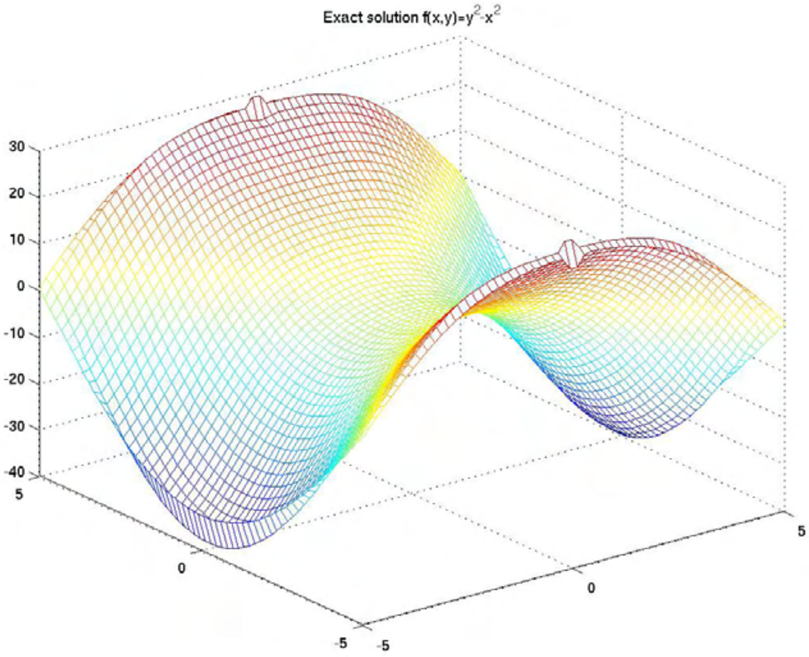


Figure 3.23: Exact Solution for Example 6, $n = 60$.

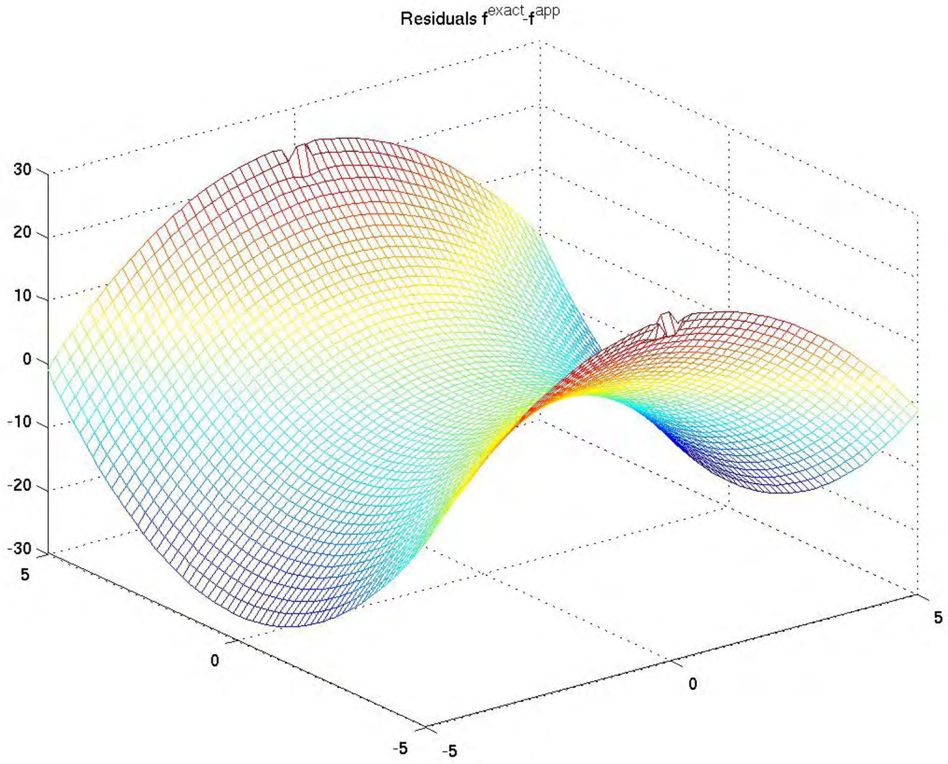


Figure 3.24: Residual Example 6, $f_0 = 1$, $n = 60$.

Example 7.

Lets check another example. Now

$$K(u,v,x,y) = \frac{\sin u}{1+\sqrt{(x-u)^2+(y-v)^2}}$$

and the test function is $f(x,y) = xe^{-x^2-y^2}$. The data function g is approximated as in the foregoing example. For this particular example the relative errors are high as we can see in Tables 3.8 and 3.9. If we take a closer look at Figure 3.25 it seems that the approximated solutions fit the exact solution in the inner of the region. In Tables B.4 and B.5 we have a sample of the values of the approximated solution and exact solution for $n = 60$ and different initial guesses. The Figures 3.28 and 3.29 show that certainly the numerical solution is closed to the exact solution in the inner but it does not fit well the exact solution in the boundary in the case of initial guess $f_0 = 1$. That is why we have high relatives errors for this case.

n	α	$\|Az_\alpha - g\|$	$\|z_\alpha\|$	$\frac{\|z_\alpha - \bar{z}\|}{\|z_\alpha\|}$
10	0.16005	20399627.4675	121.8057	0.14117
20	0.010032	32107265.7223	227.0518	0.11654
30	0.0019668	44203668.8915	331.9676	0.10684
40	0.00061694	56409382.9004	436.8627	0.10113
50	0.00025081	68660264.9581	541.7847	0.097212
60	0.00012021	80934062.1857	646.746	0.094292

Table 3.8: Some significant quantities for Example 7. Case 2D

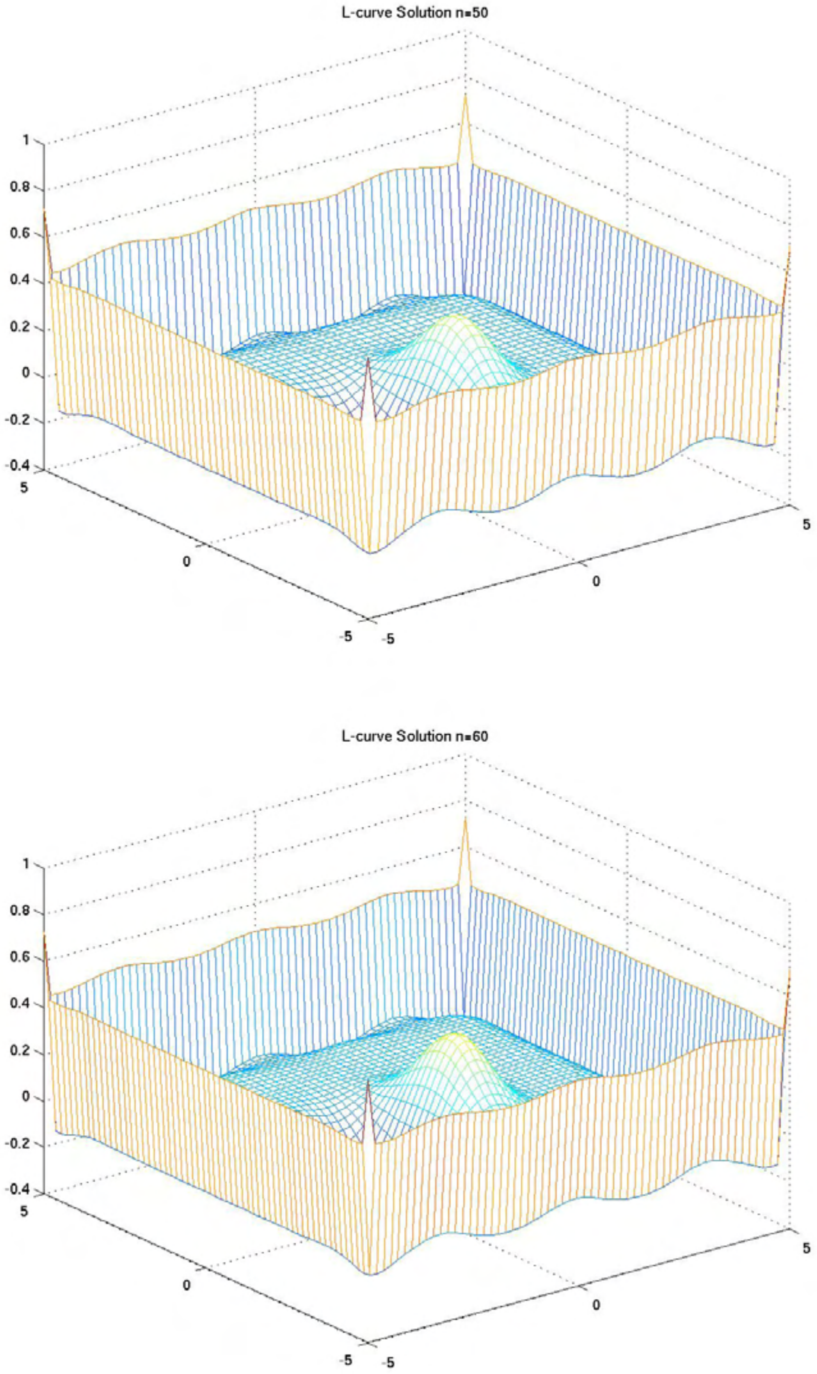


Figure 3.25: Approximated solutions for Example 7. Case 2D

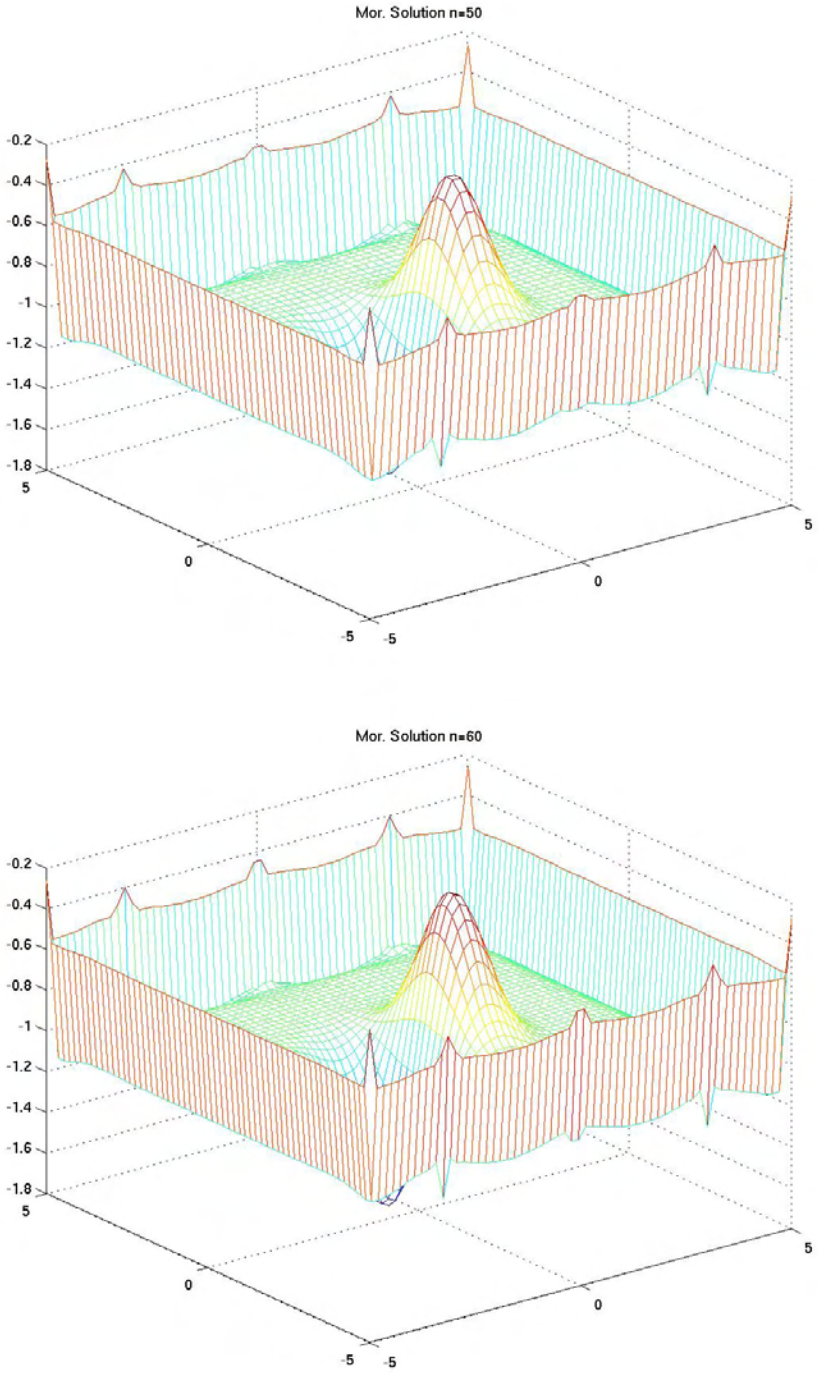


Figure 3.26: Approximated solutions for Example 7. Discrepancy Principle. Case 2D

n	α	$\|Az_\alpha - g\|$	$\|z_\alpha\|$	$\frac{\|z_\alpha - \bar{z}\|}{\|z_\alpha\|}$
0	1.2558e-06	2.04e+07	9.5255	15.299
20	1.0588e-07	3.2107e+07	19.525	16.343
30	4.5184e-08	4.4204e+07	29.61	16.228
40	4.1028e-08	5.641e+07	39.698	16.17
50	4.151e-08	6.866e+07	49.783	16.135
60	3.7903e-08	8.0934e+07	59.868	16.112

Table 3.9: Some significant quantities for Example 7. Discrepancy Principle. Case 2D

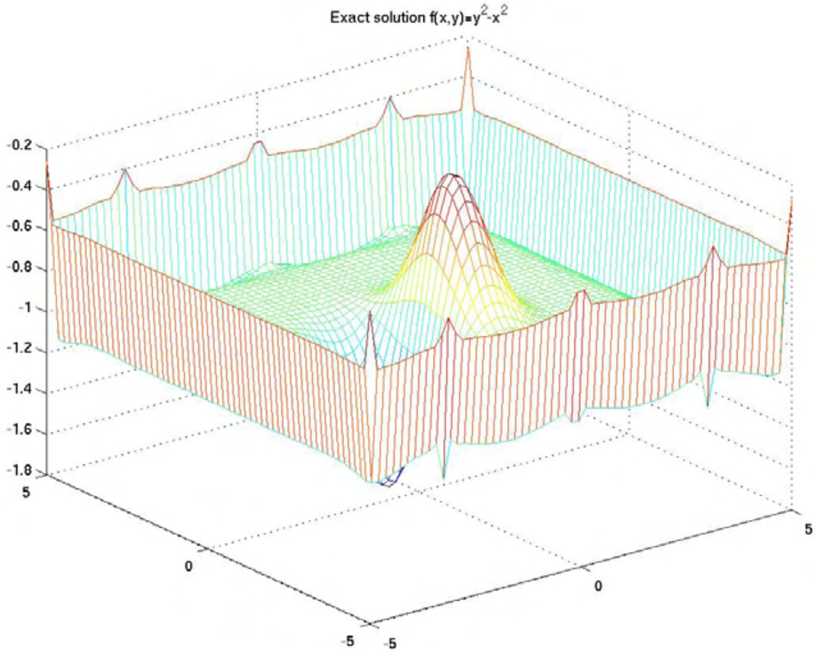


Figure 3.27: Exact solution for Example 7. Discrepancy Principle. Case 2D

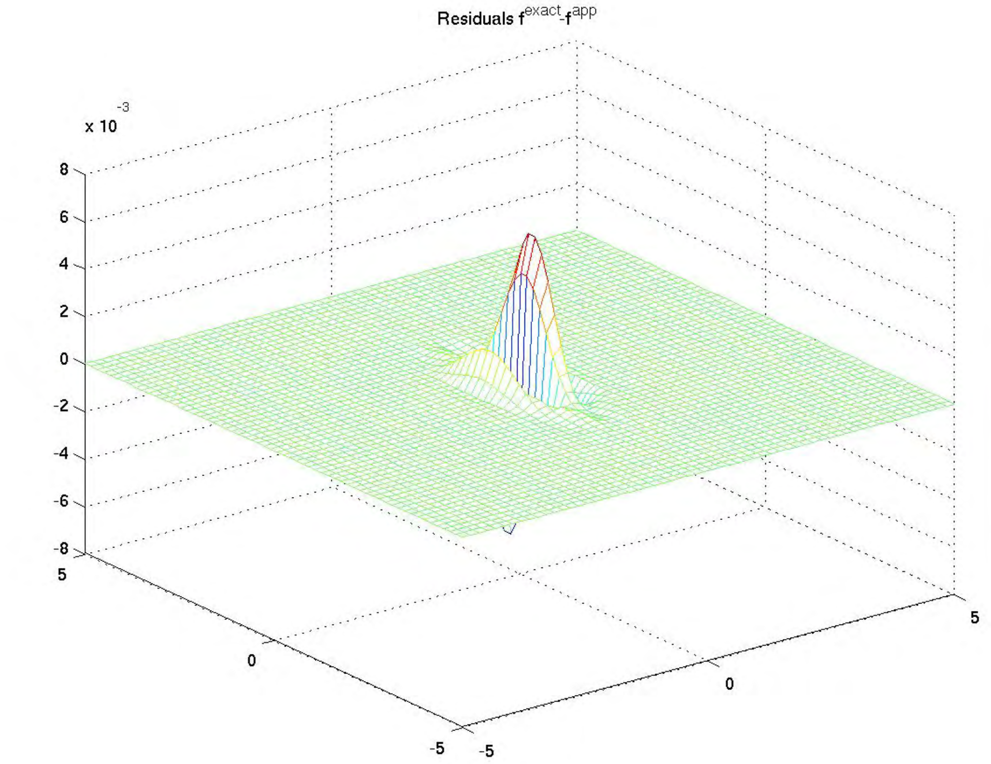


Figure 3.28: Residuals Example 7, $f_0 = 0$, $n = 60$.

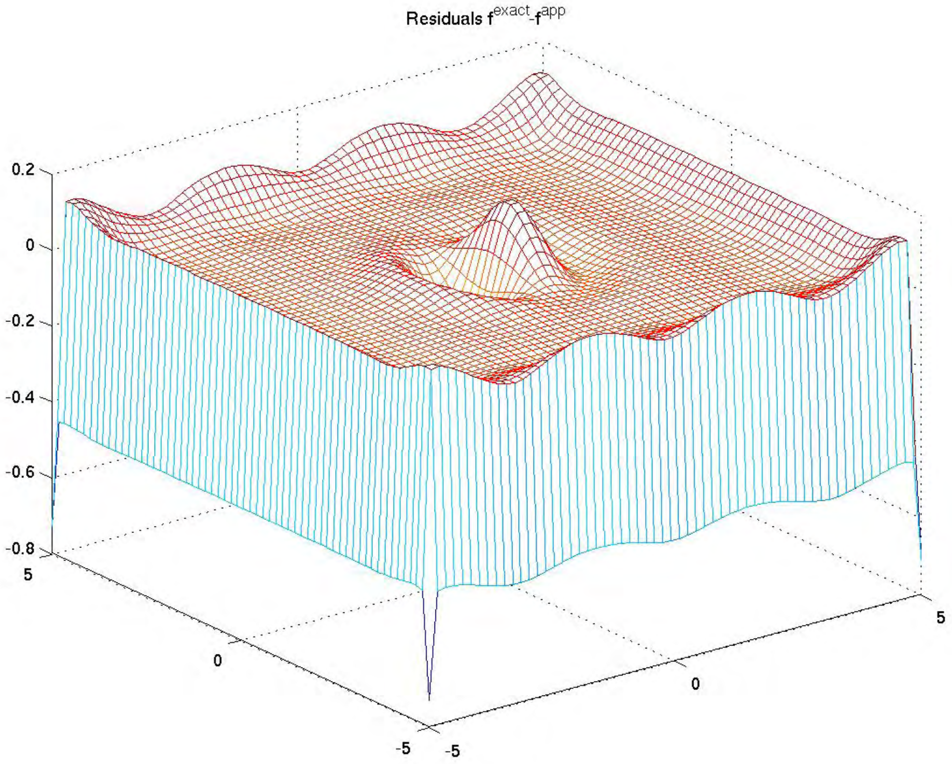


Figure 3.29: Residuals Example 7, $f_0 = 1$, $n = 60$.

3.3.3 Solution to three dimensional Fredholm integral equation of first kind

To solve Equation 3.12 let us take a straight parallelepiped $\Omega = [a, b] \times [, d] \times [e, h]$ which is sliced to obtain a grid in three dimension.

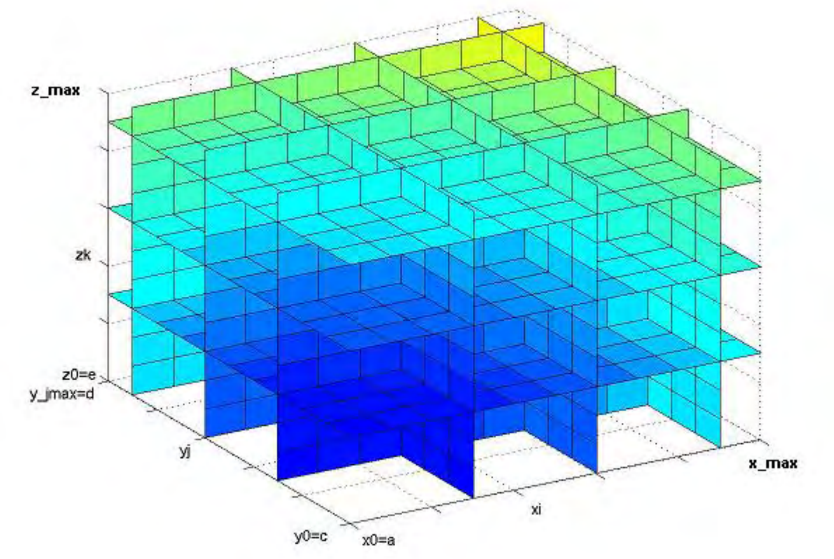


Figure 3.30: Grid where f is looked for.

Put $\mathbf{x} = (s, u, v)$, $\quad \mathbf{y} = (x, y, z)$ where s, u, v, x, y, z are real variables. Then for a sample of $imax \times jmax \times kmax$ the function f is approximated by $\mathbf{f}$ and for $pmax \times qmax \times rmax$ points the data function g can be approximated by $\mathbf{g}$. Besides

$$\begin{aligned} T(f)(s_p,u_q,v_r) &= \int_e^h\int_c^d\int_a^b K(s_p,u_q,v_r,x,y,z)f(x,y,z)dxdydz \\ &\approx \sum_{k=1}^{kmax}\sum_{j=1}^{jmax}\sum_{i=1}^{imax} K(s_p,u_q,v_r,x_i,y_j,z_k)f(x_i,y_j,z_k)w_iw'_jw''_k \\ & \quad p=1,\ldots pmax, q=1,\cdots,qmax, r=1,\cdots,rmax. \\ & \quad w_i, w'_j, w''_k \text{ are weights.} \end{aligned}$$

Thus the matrix $\mathbf{T}$ that approximates the kernel has order $(imax)(jmax)(kmax)\times(pmax)(qmax)(rmax)$, the element in the position $\big(qmax(pmax(r-1)+q-1)+p), jmax(qmax(k-1)+j-1)+i\big)$ is given by $K(s_p,u_q,v_r,x_i,y_j,z_k)w_iw'_jw''_k$. The problem is reduced to solve $\mathbf{g}=\mathbf{Tf}$.

In Section A.2 the program used to solve the following examples is shown.

Example 8.

We have solved the Equation 3.12 with $\Omega=[-5,4]\times[-5,5]\times[-5,5]$. The kernel is $K(s,u,v,x,y,z)=xye^{-s^2-y^2-v^2}$. The test function is $f(x,y,z)=x+y+z$ and the data function is $g(x,y,z)=-3750e^{-x^2-y^2-z^2}$ taken in the same grid as f. Here $n=imax=jmax=kmax$.

Table 3.10 shows values of the regularization parameter, residual norm, norm of the regularized solution and relative error for different values of n and $\delta=10^{-3}$. The relative errors are near 1. The L-curve method was not able to draw a curve, but still gives values of the regularization parameter.

n	α	$\|Az_\alpha - g\|$	$\|z_\alpha\|$	$\frac{\|z_\alpha - \bar{z}\|}{\|z_\alpha\|}$
5	426.8262	359.4216	6.9644	0.99448
10	235.7541	97.1448	18.0059	0.99433
15	258.3121	64.4325	32.4125	0.99418

Table 3.10: Relative errors and regularization parameters for Example 8

Example 9.

We have solved the Equation 3.12 with $\Omega = [-5,4] \times [-5,5] \times [-5,5]$. The kernel is

$$K(s,u,v,x,y,z) = \frac{1}{1+\sqrt{(x-s)^2+(y-u)^2+(z-v)^2}}.$$

The test function is $f(x,y,z) = x+y+z$ and the data function g is approximated by $\mathbf{g} = \mathbf{Tf}$ taken in the same grid as f. Here $n = imax = jmax = kmax$.

Table 3.11 summarizes the results for this example. Figures 3.31, 3.31 and 3.31 show L-curves for different values of n

n	α	$\|Az_\alpha - g\|$	$\|z_\alpha\|$	$\frac{\|z_\alpha - \bar{z}\|}{\|z_\alpha\|}$
5	68.9043	25435183925.5595	48.254	0.47825
10	63.0528	59327942223.8324	129.5619	0.52018
15	62.0819	102170668039.877	241.1695	0.46839

Table 3.11: Relative errors and regularization parameters for Example 9

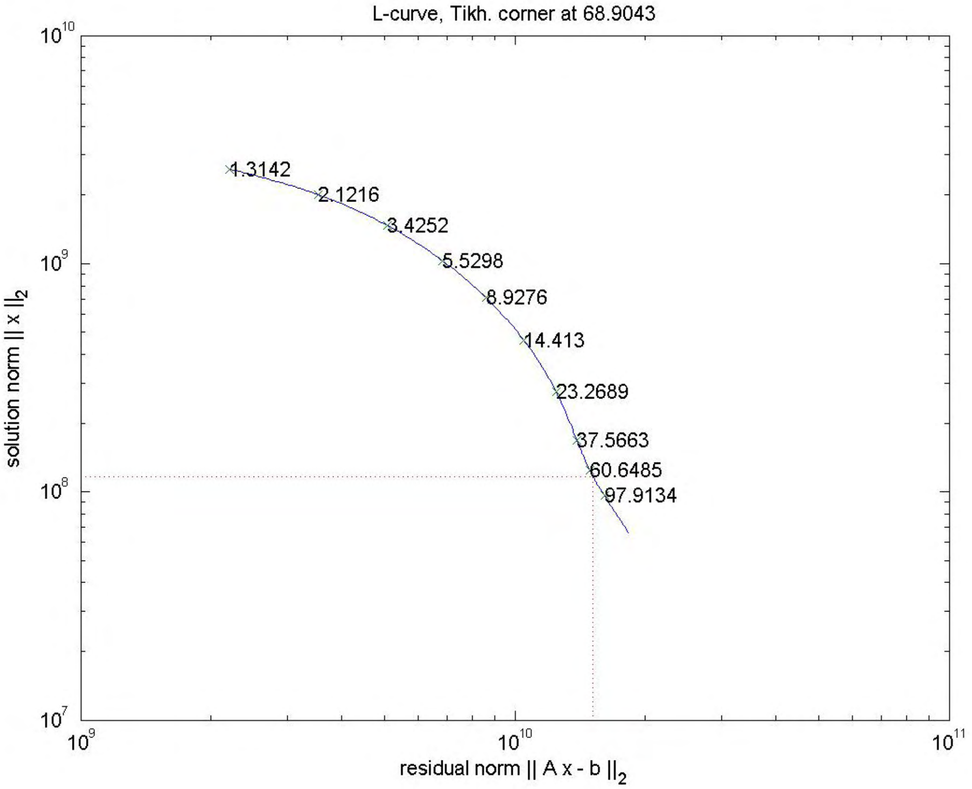


Figure 3.31: L-curve $n = 5$, Example 9. Case 3D

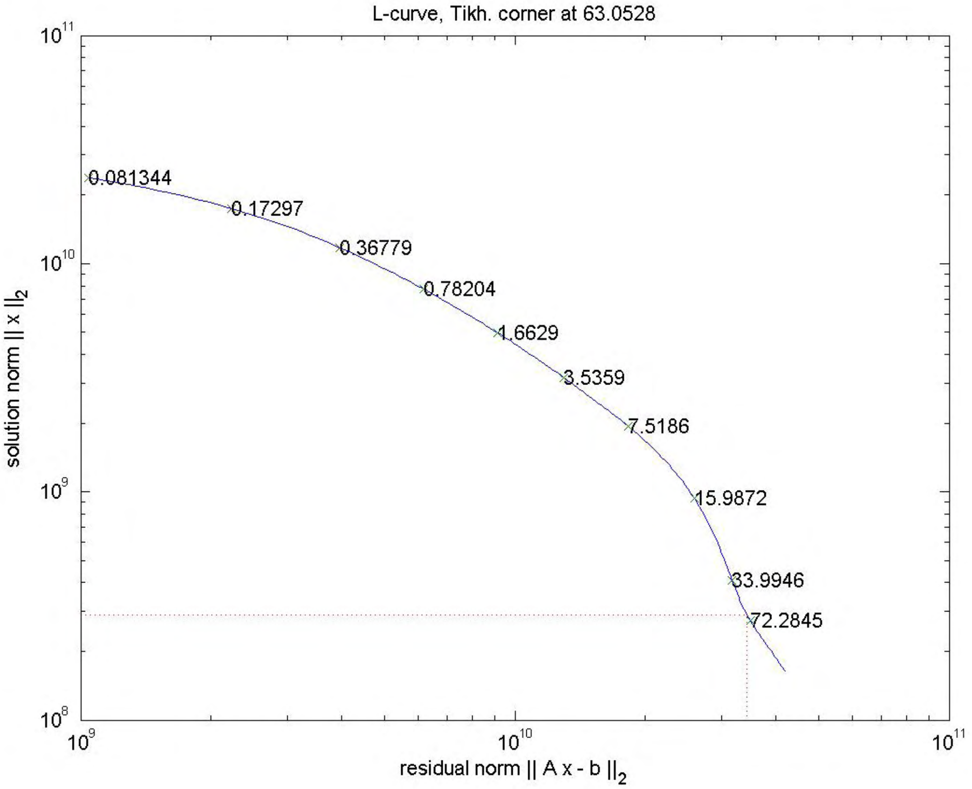


Figure 3.32: L-curve $n = 10$, Example 9. Case 3D

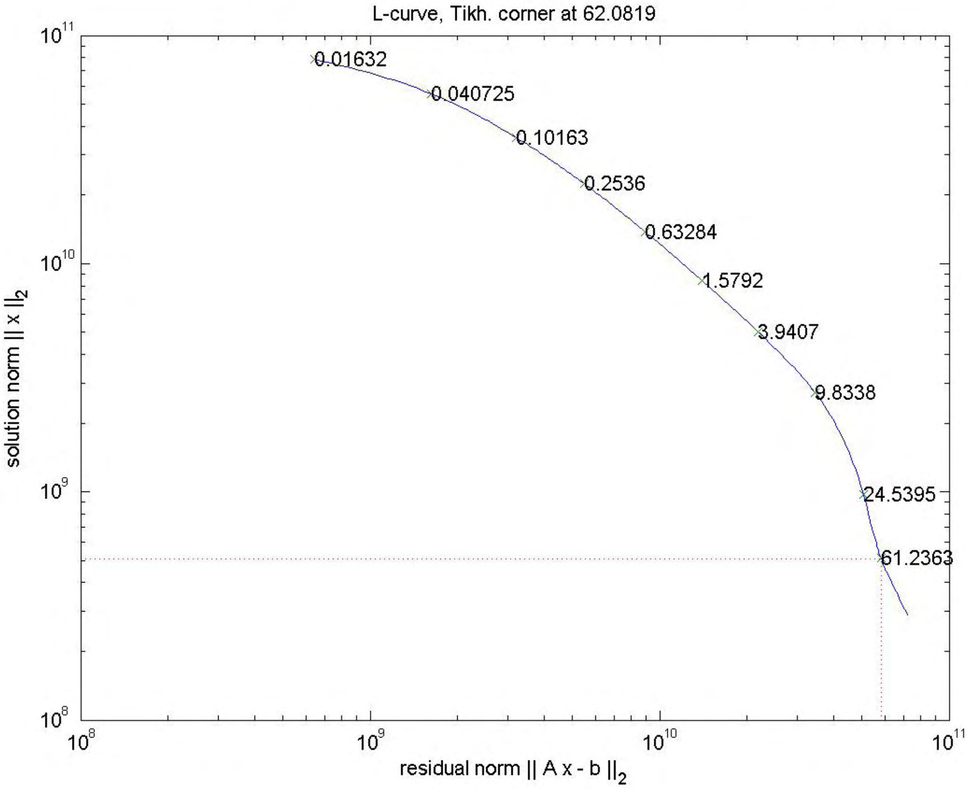


Figure 3.33: L-curve $n = 15$, Example 9. Case 3D

3.4 The results

The method has worked satisfactorily with the examples 4, 6 and 7. The approximated solutions were very closed to the exact solution. For examples 3 and 5 the approximated solutions were not good. The rate of change of the kernels for examples 3 and 5 is very slow and the corresponding exact solutions show the same behavior. Under these conditions the L-curve method failed in the estimation of the regularization parameter. This is similar to that Hanke showed in [9] and we mentioned in Section 2.4.3. The computed regularization parameters, which were obtained based on the discrepancy principle (2.4.1), were not able to give good approximated solutions.

For the case of three dimensions we could only use small values for n. Therefore, we obtained a bad approach to the original problem. The results that we get were not conclusive.

A more exhaustive numerical analysis is necessary for the better understanding of the results.

3.5 The programs

The programs were written in MATLAB because it has built-in subroutines which make the job easier and has a nice interface. We also wanted to take advantage of the work done by Gutierrez [7] and Hansen [11]. We have tried to write self-explained programs with a minimum of functions. Besides, they save relevant information in graphics and data files.

The hardest part of the job is the Singular Value Decomposition(SVD) and MATLAB does it very well. It uses LAPACK routines to compute the singular value decomposition. To see more technical details you can see MATLAB help, function ”svd”.

There are lots of comments in the code of the programs such that they can help you to understand how the programs work. The main functions are `inteq2D.m, inteq3D.m` which solve integral equation of the first kind in two and three dimensions respectively. The programs companion are `auxinteq2D.m` and `auxinteq3D.m` which helps to compute the approximated kernel and the approximated data function when analytical expressions for them are given . The general use of the programs is described in Appendix A.1.

3.6 Some Final Considerations

Inverse problems are not easy to handle with. Since they are ill-posed in general, one has to be very careful with approximations and computations. Integral equations of the first kind fall within the scope of inverse problems, they inherit the troubles that the latter have. In order to avoid those problems, a variety of techniques to solve inverse problems are considered. One of the most successful techniques is the regularization method which we have explained and applied in this work [5, 25]. The "solutions" obtained through this method must be taken carefully because they can not reflect the essence of the solution we look for. At the end, the knowledge of the particular case and "a priori" information about it can help to find the best solution. In real problems there is no way to compute the relative errors, so it is up to you to determine how well the numerical solution describes your problem [28].

The ill-posedness of the integral equations of the first kind make them prone to have very high fluctuations because of small errors in the data. In addition to this, when we use the computer to solve the problem we are adding not only an error due to discretization of the problem but also an error due to machine precision. To deal with this fact some basic assumptions are done. The number of points chosen to build the discrete version of the problem must be large enough to guarantee that it does not interfere significantly with the error in the data or error due to machine precision [25]. Sometimes the effects of this error in the solution can not be predicted.

One of the important things in solving an integral equation of first kind by regularization method is the choice of the regularization parameter α. We have mentioned the *L-curve method* and the *Discrepancy Principle*. They have their limitations. Therefore, it is not a easy matter to choose the best regularization parameter. We used here the MATLAB package described in [11]. It has several methods for choosing the parameter including the

L-curve Method (l_curve function) and the Discrepancy Method (discrep) function.

CONCLUSIONS

A solution method for integral equations of the first kind in two and three dimensions with no singularities in its kernel was implemented. It gave good results in the two-dimensional case. For three dimensions the method was not conclusive due to the small size of the discretization mesh that we had to use. In each case the quality of the solutions depended on the rate of change the kernels and the smottness of the exact solution. Also, the solutions were influenced by the error in the data and the size of the grid.

BIBLIOGRAPHY

[1] Aliprantis, C., and Burkinshaw, O. *Principles of Real Analysis*. Academic Press, 1998.

[2] Beck, J., Blackwell, B., and St. Clair, C. J. *Inverse Heat Conduction*. Wiley-Interscience, 1985.

[3] Brézis, H. *Analyse Fonctionelle*. Dunod, 1999.

[4] Brown, B., Sanko, V. S., J.W, K., and Marleta, M. Inverse spectral problem for the sturm-liouville equation. *Inverse Problems 19* (2003), 235–252.

[5] Denisov, A. *Elements of the theory of inverse problems*. VSP, Utrecht, 1999.

[6] Dunford, N., and Schwartz. *Linear Operators*, vol. 1. Interscience, New York, 1958.

[7] Gutierrez Ramirez, A. Numerical approximations of the solutions of integral equation of the first kind. Report of Project. In charge Dr. Lev Steinberg.

[8] Hanke, M. Limitation of the l-curve method in ill-posed problems. *BIT 36* (1996), 287–301.

[9] Hanke, M. Iterative regularization techniques. In *Surveys on Solutions methods for inverse problems*. Springer, Vienna, 2000, pp. 35–52.

[10] Hansen, P. The l-curve and its use in the numerical treatment of inverse problems. Tech. rep., Technical University of Denmark, Denmark, 1992.

[11] Hansen, P. Regularization tools: A matlab package for analysis and solution of discrete ill-posed problems. Tech. rep., Technical University of Denmark, 1992. Revised 2001.

[12] Klibanov, M. Carleman estimates and inverse problems in last two decades. In *Solutions Methods for inverse problems*. Springer Verlag, 2000.

[13] Lamm, P. Some recent developments and open problems in solutions methods for mathematical inverse problems. In *Proc. XXIV National Congress of Applied and Computational Mathematics* (Aug. 2001).

[14] Lebedev, V. *An introduction to Functional Analysis and Computational Mathematics.* Birkhauser, 1997.

[15] Leeb, H., Fiedeldey, H., Schnizer, W., Sofianos, S., and Lipperheide, R. Inverse scattering problem at fixed energy for a solvable class of potentials. *Inverse Problems 5* (1989), 817–830.

[16] Levitan, B. *Inverse Sturm-Liouville Problems.* VNU Science Press, Utrecht, 1987.

[17] Liusternik, L., and Sobolev, V. *Elements of Functional analysis.* Frederick Ungar, 1961.

[18] Miller, G. Fredholm equations of first kind. In *Numerical Solutions of Integral Equations*, L. Delves and J. Walsh, Eds. Clarendon Press, 1974, pp. 175–188.

[19] Muller, H., Kammarath, and Erne, S. Multimodal human brain mapping in a new analysis and visualization software environment. *Division for Technical Electromagnetism, Central Institute for Biomedical Engineering. Ulm University* (19??).

[20] Neher, M. Enclosing solutions of an inverse sturm-loiuville problem with finite data. *Computing 53* (1994), 379–395.

[21] Press, W., Flannery, B., Teukolsky, S., and Vetterling, W. *Numerical Recipes in C.* Cambridge University Press, 1992.

[22] Steinberg, L. The method of determination of dynamical moduli of elasticity. *Patent USSR # 1234753 MKI G 01n 3/32* (1984).

[23] Steinberg, L. Inverse problems for oscillations of cylinders and bars. *J. Elasticity 38* (1995), 133–151.

[24] Steinberg, L. The linearization of inverse spectral inverse problems for non homogeneous elastic bodies. *Inverse Problems 12* (1996), 483–497.

[25] Tikhonov, A., Goncharski, A., Stepanov, V., and A.G, Y. *Numerical Methods for the Solution of Ill-Posed Problems.* Kluwer Academic Press, 1995.

[26] Valdivia, N. Numerical solutions of the linearized inverse spectral sturm-liouville problem. Master's thesis, University of Puerto Rico, 1997.

[27] Widom, H. *Lectures on Integral Equations.* Van Nostrand, 1969.

[28] Wing, G. *A primer of Integral Equations of the First Kind.* SIAM, 1991.

[29] Xiao, Y. Fasts convergent algorithms for solving 2d integral equations of the first kind. *Lecture Notes in Physics*, 552 (2000), 153–161.

[30] Yosida, K. *Functional Analysis.* Springer-Verlag, 1974.

Appendix A

MATLAB CODE FOR THE EXAMPLES

This appendix shows the MATLAB code of the programs and explain how to use them. The common function used by the principal programs is `QuadWeights`. It finds the quadrature weights when we know the analytical expression for the kernel.

```
% function QuadWeights(a,b,npoints,typeq);
% this function returns weights of of a type of quadrature.
% a Initial point of interval
% b End point of the interval
% input npoints integer, how many points), type
%input typeq( integer, type of quadrature as follow:
%       1-trapezoidal rule
%       2-Simpson rule
%       n- Can be extended
function weight=QuadWeights(a,b,npoints,typeq);
  weight=ones(1,npoints);
  trapezoidal=1;
  simpson=2;
  cl_newton_cotes=3;
  switch typeq
    case trapezoidal
      weight(1)=1/2;
      weight(npoints)=1/2;
    case simpson
        if mod(npoints,2)==1
            weight(1)=1;
            weight(npoints)=1;
            for k=2:npoints-1
                if mod(k,2)==0
                  weight(k)=2;
                else
                  weight(k)=4;
                end
            end
            weight=weight*1/3;
        else
                   end
    case cl_newton_cotes
        switch npoints
            case 5
                weight=(2/45)*[7 32 12 32 7];
            case 6
                weight=(5/288)*[19 75 50 50 75 19];
            case 10
                weight=...
                (9/89600)*[2857 15471 1080 19344 ...
                5778 5778 19344 1080 15741 2857];
        end
  end

 weight=((b-a)/(npoints-1))*weight;
```

A.1 MATLAB code for two-dimensional case

A.1.1 Function inteq2D

It gives the solution of a integral equation of first case in two dimensions with no singularities in the kernel in a mesh $imax \times jmax$.

```
function [SOL2D,RegPar, SizeRegSol, SizeResidual]=...
    inteq2D(MatrixKernel2D,DataG,u,v,x,y,InitSol,ErrDelta)
% Function inteq2D.m.
% inteq2D(MatrixKernel2D,DataG,u,v,x,y,InitSol,ErrDelta,MeshSize)
% This Program compute the solution of a integral
% equation of first kind in two variables.
% INPUT
% MatrixKernel2D: Twodimensional matrix which approximates the kernel 2D.
% DataG: Is a column matrix with the known data.
% u,v  defines the grid where the data points are known
% x,y defines the grid we use  to compute the approximated solution
% Initsol: Real number Initial solution
% ErrDelta: is de estimated error in the right handside g
%*****************************************************************
% variable  *   counter *   N points   *   weights
%*****************************************************************
%   u       *           *       pmax   *
%   v       *           *       qmax   *
%   x       *           *       imax   *        w1
%   y       *           *       jmax   *        w2
%*****************************************************************
% OUTPUT
% SOL2D: Contain the solution over a mesh (pmax)(qmax)
% RegPar: Parameter of regularization given by l_curve
% SizeRegSol: norm of the regularized solution ||z||
% SizeResidual: Residual norm ||Az-g||

    format long
    hdp1=figure(1); %graphic handle to store temporarily the picture
    hdp2=figure(2); %graphic handle to store temporarily the picture
    pmax=length(u); qmax=length(v); imax=length(x) ;jmax=length(y);
    W=zeros(imax*jmax,1); % approximated solution
    isol=InitSol*ones(imax*jmax,1); %initial solution
    G=DataG-MatrixKernel2D*isol; %compute the new data
    %*************************************************************
    % HERE WE DO THE SINGULAR VALUE DECOMPOSITION
    %*************************************************************
    display('Doing SVD...')
    [U,S,V]=csvd(MatrixKernel2D);
    display('SVD is done...')
    figure(hdp1);
    % Compute the regularization parameter and draw the l-curve.
    % Save a copy of it in the current directory.
    [regc,rhoc,etac,regparc]= l_curve(U,S,G);
    saveas(hdp1,'LCURV2D','fig');
    RegPar=regc;
    %Now compute the solution
    for j=1:imax*jmax
        W=W+(S(j)/(regc+(S(j))^2))*((G’)*(U(:,j)))*V(:,j);
    end
    Sol{1}=W+isol;
    %Arrange the solution in a mesh in order to be plotted.
    for j=1:jmax
         SOL2D(j,:)=Sol{1}(imax*(j-1)+1:j*imax)’;
    end
    SizeResidual=norm(MatrixKernel2D*Sol{1}-DataG);
    SizeRegSol=norm(Sol{1});
    figure(2);
    hdp2=mesh(x,y,SOL2D);
    strtitle=strcat(strcat(strcat(’Approx. Solution, ||z||=’,...
        num2str(SizeRegSol)),’    ||Az-g||=’),num2str(SizeResidual));
    title(strtitle)
    saveas(hdp2,’SOL2D’,’fig’);
    display(’To save a copy of the approx. solution could take a while.’)
    if input(’Do you want save a copy of this solution,yes=1/no=0? ’)
        dlmwrite(’NUMSOL2D’,SOL2D,’\t’)
    end
```

A.1.2 Script auxinteq2D

This program helps to build the input for the function inteq2D when analytical expressions for the data function is given or, in the case of a test, when the exact solution is given.

```
format long clc clear all def={’0’}; num_lines=1;
InitSol=inputdlg(’Enter the initial solution’,...
    ’Initial Solution’,num_lines,def);
InitSol=str2num(InitSol{1}); def={’10^(-3)’}; num_lines=1;
ErrDelta=inputdlg(’Enter the error delta’,...
    ’Error in the right-hand side’,num_lines,def);
ErrDelta=str2num(ErrDelta{1}); prompt = {’Number points of
x’,’Number points of y’,’a’,’b’,’c’,’d’};
def={’10’,’10’,’-5’,’5’,’-5’,’5’}; num_lines=1; dlg_title = ’Input
for IFK 2D’; answer  = inputdlg(prompt,dlg_title,num_lines,def);
imax=str2num(answer{1}); jmax=str2num(answer{2});
a=str2num(answer{3}); b=str2num(answer{4}); c=str2num(answer{5});
```

```
d=str2num(answer{6}); prompt = {'Number points of u','Number
points of v','e','f','g','h'}; def={'10','10','-5','5','-5','5'};
num_lines=1; dlg_title = 'Input for IFK 2D, Data Points'; answer
= inputdlg(prompt,dlg_title,num_lines,def);
pmax=str2num(answer{1}); qmax=str2num(answer{2});
e=str2num(answer{3}); f=str2num(answer{4}); g=str2num(answer{5});
h=str2num(answer{6});prompt = {'Enter 2D kernel:',...
        'Enter data function:','Enter exact solution(test only)'};
def={'1./(1+sqrt((x-u).^2+(y-v).^2))','','x.*exp(-x.^2-y.^2)'};
dlg_title = 'Input for IFK 2D'; num_lines= 1; answer  =
inputdlg(prompt,dlg_title,num_lines,def);
Kernel2DA=inline(answer{1}); Kfunction2DA=inline(answer{2});
ExactSol2DA= inline(answer{3});
u=e:(b-a)/(pmax-1):f; % data
x=a:(b-a)/(imax-1):b; % quadrature points
v=g:(d-c)/(qmax-1):h; % data
y=c:(d-c)/(jmax-1):d; % quadrature points
[X,Y]=meshgrid(x,y); w1=QuadWeights(a,b,imax,1);
w2=QuadWeights(c,d,jmax,1);
matrix=zeros(imax*jmax); % to store the aprroximated kernel
kfun=zeros(pmax*qmax,1); %  matrix to store the  known function
rfun=zeros(imax*jmax,1); %  matrix to store the real solution
W=zeros(imax*jmax,1); % approximated solution
%********************************************
%COMPUTE THE APPROXIMATED KERNEL
%********************************************
for q=1:qmax; % counter for v
        for p=1:pmax; % counter for u
            for j=1:jmax; % counter for y
                for i=1:imax; % counter for x
                    matrix(pmax.*(q-1)+p,imax.*(j-1)+i)=Kernel2DA(...
                        u(p),v(q),x(i),y(j)).*w1(i).*w2(j);
                end
            end
        end
end

if ~strcmp(answer{2},'')
    %******************************************
    %COMPUTE THE DATA FUNCTION
    %******************************************
     for j=1:jmax
```

```
        for i=1:imax
               kfun(imax.*(j-1)+i)=Kfunction2DA(x(i),y(j));
        end
    end
    kfun=kfun+ErrDelta;
elseif ~strcmp(answer{3},'')
    %******************************************
    %COMPUTE THE EXACT SOLUTION
    %******************************************
        for j=1:jmax
        for i=1:imax
               rfun(imax.*(j-1)+i)=ExactSol2DA(x(i),y(j));
        end
    end
     kfun=matrix*rfun+ErrDelta;
else
   quit
end

[Sol,regpar,norreg,norres]=...
inteq2D(matrix,kfun,u,v,x,y,InitSol,ErrDelta);
```

A.2 MATLAB code for three-dimensional case

A.2.1 Function inteq3D

It gives the solution of a integral equation of first case in two dimensions with no singularities in the kernel in a mesh $imax \times jmax \times kmax$.

```
    function [SOL3D,RegPar, SizeRegSol, SizeResidual]=...
    inteq2D(MatrixKernel3D,DataG,s,u,v,x,y,z,InitSol,ErrDelta,MeshSize)
% Function inteq2D.m.
% inteq2D(MatrixKernel2D,DataG,u,v,x,y,InitSol,ErrDelta,MeshSize)
% This Program compute the solution of a integral
% equation of first kind in two variables.
% INPUT
% MatrixKernel3D: Twodimensional matrix which approximates the kernel 2D.
% DataG: Is a column matrix with the known data.
% s,u,v  defines the grid where the data points are known
% x,y,z defines the grid we use  to compute the approximated solution
% Initsol: Real number, Initial solution
% ErrDelta: is de estimated error in the right handside g
```

```
%*****************************************************************
% variable  *   counter *   N points *   weights
%*****************************************************************
%   s       *           *    pmax           *
%   u       *           *    qmax           *
%   v       *           *    rmax           *
%   x       *           *    imax           *      w1
%   y       *           *    jmax           *      w2
%   z       *           *    kmax           *      w3
%*****************************************************************
% OUTPUT
%[SOL3D,RegPar, SizeRegSol, SizeResidual]
% SOL3D: Contain the solution
% RegPar: Parameter of regularization given by l_curve
% SizeRegSol: norm of the regularized solution ||z||
% SizeResidual: Residual norm ||Az-g||


    format long
    imax=length(x); jmax=length(y); kmax=length(z);
    W=zeros(imax*jmax*kmax,1); %to store approximated solution
    isol=InitSol*ones(imax*jmax*kmax,1); %initial solution
    display('Doing SVD...')
    [U,S,V]=csvd(MatrixKernel3D);
    display('Done the singular values');
    G=DataG-MatrixKernel3D*isol;
    hdp1=figure(1);
    [regc,rhoc,etac,regparc]= l_curve(U,S,G);
    saveas(hdp1,'LCURV3D','fig');
    RegPar=regc;
    for j=1:imax*jmax*kmax
        W=W+(S(j)/(regc+(S(j))^2))*(G'*(U(:,j)))*V(:,j);
    end
    SOL3D=W+isol;
    SizeResidual=norm(MatrixKernel3D*SOL3D-DataG);
    SizeRegSol=norm(SOL3D);
    button = questdlg(...
    'Do you want to save the solution. It could take a while',...
       'Continue Operation','Yes','No','No');
   if strcmp(button,'Yes'); dlmwrite('NUMSOL3D',SOL3D,'\t'); end
```

A.2.2 Script auxinteq3D

This program helps to build the input for the function inteq2D when analytical expressions for the data function is given or, in the case of a test, when the exact solution is given.

```
 format long
clc clear
% Define the default options of the dialog box
% for entering the limits of intervals, kernel, known function.

def={'10^(-3)'}; num_lines=1; ErrDelta=inputdlg('Enter the error
delta',...
    'Error in the right-hand side',num_lines,def);
prompt = {'Number points of x','Number points of y',...
        'Number points of z','a','b','c','d','e','f'};
def={'5','5','5','-5','4','-5','5','-5','5'};
num_lines=1;
dlg_title
= 'Input for IFK 3D';
 answer  =
inputdlg(prompt,dlg_title,num_lines,def); imax=str2num(answer{1});
jmax=str2num(answer{2}); kmax=str2num(answer{3});
a=str2num(answer{4}); b=str2num(answer{5}); c=str2num(answer{6});
 d=str2num(answer{7}); e=str2num(answer{8}); f=str2num(answer{9});
prompt = {'Number points of s','Number points of u',...
        'Number points of v','a1','b1','c1','d1','e1','f1'};
def={'5','5','5','-5','4','-5','5','-5','5'}; num_lines=1; dlg_title
= 'Input for IFK 3D, Data Points'; answer  =
inputdlg(prompt,dlg_title,num_lines,def); pmax=str2num(answer{1});
qmax=str2num(answer{2}); rmax=str2num(answer{3});
s=str2num(answer{4}); u=str2num(answer{5}); v=str2num(answer{6});
a1=str2num(answer{4}); b1=str2num(answer{5}); c1=str2num(answer{6});
 d1=str2num(answer{7}); e1=str2num(answer{8}); f1=str2num(answer{9});
prompt = {'Enter 3D kernel:',...
        'Enter data function:','Enter exact solution(test only)'};
def={'x.*y.*exp(-s.^2-y.^2-v.^2)','','x+y+z'}; dlg_title = 'Input
for IFK 3D'; num_lines= 1; answer  =
inputdlg(prompt,dlg_title,num_lines,def);

%Build the function with the entered data.
Kernel3D=inline(answer{1}); Kunction3D=inline(answer{2});
ExactSol3D= inline(answer{3});
```

```
 s=a1:(b-a)/(pmax-1):b1; % data
 x=a:(b-a)/(imax-1):b; % quadrature point
 u=c1:(d-c)/(qmax-1):d1; % data
 y=c:(d-c)/(jmax-1):d; % quadrature point
 v=e1:(f-e)/(rmax-1):f1; % data
 z=e:(f-e)/(kmax-1):f; % quadrature point
 w1=QuadWeights(a,b,imax,1);
 w2=QuadWeights(c,d,jmax,1);
 w3=QuadWeights(e,f,kmax,1);
 matrix=zeros(imax*jmax*kmax);
    kfun=zeros(pmax*qmax*rmax,1); %column matrix that
                                  %store the known function
    W=zeros(imax*jmax*kmax,1); %approximated solution
    isol=0*ones(imax*jmax*kmax,1); %initial solution
    %********************************************
    %COMPUTE THE APPROXIMATED KERNEL
    %********************************************
      for r=1:rmax % counter for v
        for q=1:qmax %counter for u
         for p=1:pmax; % counter for s
          for k=1:kmax; % counter for v
           for j=1:jmax; % counter for q
            for i=1:imax; % counter for p
              matrix(qmax*(pmax*(r-1)+q-1)+p,...
                  kmax*(imax*(k-1)+j-1)+i)=...
                   Kernel3D(s(p),u(q),v(r),x(i),y(j),z(k))...
                      .*w1(i).*w2(j).*w3(k);
            end
           end
          end
         end
        end
     end
    %******************************************
    %COMPUTE THE KNOWN FUNCTION
    %******************************************
if ~strcmp(answer{2},'')
    %******************************************
    %COMPUTE THE DATA FUNCTION
    %******************************************
      for r=1:rmax
        for q=1:qmax
```

```
         for p=1:pmax
            kfun(qmax*(pmax*(r-1)+q-1)+p)=...
            Kfunction3D(s(p),u(q),v(r));
         end
        end
    end
    kfun=kfun+10^(-3);
elseif ~strcmp(answer{3},'')
    %******************************************
    %COMPUTE THE EXACT SOLUTION
    %******************************************
       for k=1:kmax
        for jmax=1:jmax
         for i=1:imax
            rfun(jmax*(imax*(k-1)+j-1)+i)=...
            ExactSol3D(x(i),y(j),z(k));
         end
        end
     end
     kfun=matrix*rfun'+10^(-3);
else
   quit
end
    kfun=matrix*rfun'+10^(-3); %add the error in the case of a test.
%compute the approximated solution.
  [Sol3D,RegPar,SizeRegSol,SizeRes]=...
   inteq3D(matrix,kfun,s,u,v,x,y,z,0,10^(-3))
```

Appendix B

TABLES OF NUMERICAL RESULTS

Exact Values											
-10	-9.1525	-8.3051	-7.4576	-6.6102	-5.7627	-4.9153	-4.0678	-3.2203	-2.3729	-1.5254	-0.67797
-9.1525	-8.3051	-7.4576	-6.6102	-5.7627	-4.9153	-4.0678	-3.2203	-2.3729	-1.5254	-0.67797	0.16949
-8.3051	-7.4576	-6.6102	-5.7627	-4.9153	-4.0678	-3.2203	-2.3729	-1.5254	-0.67797	0.16949	1.0169
-7.4576	-6.6102	-5.7627	-4.9153	-4.0678	-3.2203	-2.3729	-1.5254	-0.67797	0.16949	1.0169	1.8644
-6.6102	-5.7627	-4.9153	-4.0678	-3.2203	-2.3729	-1.5254	-0.67797	0.16949	1.0169	1.8644	2.7119
-5.7627	-4.9153	-4.0678	-3.2203	-2.3729	-1.5254	-0.67797	0.16949	1.0169	1.8644	2.7119	3.5593
-4.9153	-4.0678	-3.2203	-2.3729	-1.5254	-0.67797	0.16949	1.0169	1.8644	2.7119	3.5593	4.4068
-4.0678	-3.2203	-2.3729	-1.5254	-0.67797	0.16949	1.0169	1.8644	2.7119	3.5593	4.4068	5.2542
-3.2203	-2.3729	-1.5254	-0.67797	0.16949	1.0169	1.8644	2.7119	3.5593	4.4068	5.2542	6.1017
-2.3729	-1.5254	-0.67797	0.16949	1.0169	1.8644	2.7119	3.5593	4.4068	5.2542	6.1017	6.9492
-1.5254	-0.67797	0.16949	1.0169	1.8644	2.7119	3.5593	4.4068	5.2542	6.1017	6.9492	7.7966
-0.67797	0.16949	1.0169	1.8644	2.7119	3.5593	4.4068	5.2542	6.1017	6.9492	7.7966	8.6441
Approximated Values											
-2.1112	-4.7648	-4.188	-3.6028	-2.9926	-2.2789	-1.6955	-1.4799	-1.2339	-0.84356	-0.4141	0.033152
-4.8617	-8.0917	-7.4229	-6.6636	-5.9883	-5.0926	-4.0728	-3.4115	-2.5173	-1.5407	-0.66384	0.19991
-4.4024	-7.3508	-6.6232	-5.7305	-4.9018	-4.0974	-3.2292	-2.4097	-1.5116	-0.66165	0.1664	1.0228
-3.9205	-6.5034	-5.7781	-4.9209	-4.073	-3.2164	-2.354	-1.5305	-0.6808	0.16667	1.0144	1.8704
-3.4302	-5.6669	-4.9261	-4.0686	-3.2279	-2.3701	-1.5085	-0.68365	0.16445	1.0174	1.8635	2.7146
-2.9359	-4.8301	-4.0778	-3.2207	-2.3774	-1.5238	-0.66901	0.16455	1.0143	1.8653	2.7113	3.5591
-2.4397	-3.9935	-3.2288	-2.3732	-1.5267	-0.67598	0.17119	1.0143	1.8641	2.7129	3.5594	4.4036
-1.9425	-3.157	-2.3799	-1.5257	-0.67616	0.17165	1.0114	1.8638	2.7138	3.5606	4.4074	5.2483
-1.4448	-2.3206	-1.531	-0.67749	0.1734	1.0182	1.8519	2.7119	3.5634	4.4086	5.2547	6.0934
-0.94722	-1.4841	-0.68016	0.16968	1.0176	1.87	2.7013	3.5615	4.4066	5.2609	6.1073	6.9411
-0.45018	-0.64881	0.16124	0.99598	1.8786	2.7635	3.5724	4.4566	5.2387	6.0634	6.9332	7.7789
0.043567	0.1981	1.0435	1.9382	2.9032	3.6627	4.3374	5.4513	6.3709	7.111	7.8754	8.7291

Table B.1: Exact and Approximated solutions for Example 4, $n = 60$, $f_0 = 1$.

Exact values											
0	7.7564	14.076	18.96	22.407	24.418	24.993	24.131	21.833	18.098	12.927	6.32
-7.7564	0	6.32	11.204	14.651	16.662	17.236	16.375	14.076	10.342	5.1709	-1.4364
-14.076	-6.32	0	4.8837	8.3309	10.342	10.916	10.055	7.7564	4.0218	-1.1491	-7.7564
-18.96	-11.204	-4.8837	0	3.4473	5.4582	6.0327	5.1709	2.8727	-0.86182	-6.0327	-12.64
-22.407	-14.651	-8.3309	-3.4473	0	2.0109	2.5855	1.7236	-0.57455	-4.3091	-9.48	-16.087
-24.418	-16.662	-10.342	-5.4582	-2.0109	0	0.57455	-0.28727	-2.5855	-6.32	-11.491	-18.098
-24.993	-17.236	-10.916	-6.0327	-2.5855	-0.57455	0	-0.86182	-3.16	-6.8946	-12.065	-18.673
-24.131	-16.375	-10.055	-5.1709	-1.7236	0.28727	0.86182	0	-2.2982	-6.0327	-11.204	-17.811
-21.833	-14.076	-7.7564	-2.8727	0.57455	2.5855	3.16	2.2982	0	-3.7346	-8.9055	-15.513
-18.098	-10.342	-4.0218	0.86182	4.3091	6.32	6.8946	6.0327	3.7346	0	-5.1709	-11.778
-12.927	-5.1709	1.1491	6.0327	9.48	11.491	12.065	11.204	8.9055	5.1709	0	-6.6073
-6.32	1.4364	7.7564	12.64	16.087	18.098	18.673	17.811	15.513	11.778	6.6073	0
Approximated Values											
-1.5221	-2.088	-0.49399	0.73772	1.6072	2.1143	2.2592	2.0419	1.4623	0.52036	-0.7838	-2.4502
-4.2579	-5.6034	-2.4155	0.047962	1.7868	2.8012	3.091	2.6563	1.497	-0.38676	-2.9951	-6.3279
-4.4717	-6.0309	-2.8429	-0.37951	1.3594	2.3737	2.6635	2.2288	1.0696	-0.81423	-3.4226	-6.7554
-4.6854	-6.4584	-3.2704	-0.80699	0.93189	1.9462	2.2361	1.8013	0.64208	-1.2417	-3.85	-7.1829
-4.8992	-6.8858	-3.6979	-1.2345	0.50442	1.5188	1.8086	1.3739	0.21461	-1.6692	-4.2775	-7.6104
-5.1129	-7.3133	-4.1254	-1.6619	0.076944	1.0913	1.3811	0.94638	-0.21287	-2.0967	-4.705	-8.0378
-5.3266	-7.7408	-4.5528	-2.0894	-0.35053	0.66382	0.95363	0.51891	-0.64035	-2.5241	-5.1325	-8.4653
-5.5404	-8.1683	-4.9803	-2.5169	-0.77801	0.23634	0.52615	0.091434	-1.0678	-2.9516	-5.5599	-8.8928
-5.7541	-8.5957	-5.4078	-2.9444	-1.2055	-0.19113	0.09868	-0.33604	-1.4953	-3.3791	-5.9874	-9.3203
-5.9678	-9.0232	-5.8353	-3.3718	-1.633	-0.61861	-0.3288	-0.76352	-1.9228	-3.8066	-6.4149	-9.7477
-6.1816	-9.4507	-6.2627	-3.7993	-2.0604	-1.0461	-0.75627	-1.191	-2.3502	-4.234	-6.8424	-10.175
-6.3953	-9.8782	-6.6902	-4.2268	-2.4879	-1.4736	-1.1837	-1.6185	-2.7777	-4.6615	-7.2698	-10.603

Table B.2: Exact and Approximated solutions for Example 5, $n = 60$, $f_0 = 1$.

Exact Values											
0.19424	9.2351	16.532	21.837	25.11	26.725	24.485	26.51	24.604	20.945	15.222	7.5727
-8.3343	1.4137	10.779	17.287	20.531	21.537	21.671	21.444	20.108	16.259	9.0889	-0.64494
-16.483	-9.1687	0.96927	8.6076	12.766	14.295	14.568	14.129	12.189	7.3596	-0.93087	-11.252
-23.367	-18.937	-8.4002	0.22975	5.4571	7.7627	8.28	7.4783	4.6797	-1.2367	-10.448	-20.969
-28.468	-26.45	-15.764	-6.4626	-0.38382	2.6046	3.3468	2.213	-1.3292	-8.0886	-17.894	-28.416
-31.513	-31	-20.28	-10.621	-4.037	-0.61946	0.26757	-1.0796	-5.085	-12.337	-22.448	-32.917
-32.391	-32.316	-21.594	-11.84	-5.1123	-1.5698	-0.64006	-2.05	-6.1899	-13.58	-23.772	-34.219
-31.076	-30.344	-19.627	-10.017	-3.5045	-0.14915	0.71672	-0.5993	-4.5378	-11.72	-21.79	-32.269
-27.606	-25.17	-14.501	-5.307	0.62741	3.4966	4.1989	3.1238	-0.28905	-6.9068	-16.619	-27.148
-22.12	-17.13	-6.6481	1.8071	6.8305	8.9802	9.4473	8.7199	6.0923	0.38061	-8.6717	-19.175
-14.934	-7.0561	2.9597	10.372	14.318	15.707	15.939	15.562	13.779	9.1709	1.0974	-9.1433
-6.6148	3.4441	12.608	18.913	22.046	23.021	23.2	22.93	21.636	17.919	10.964	1.4067
Approximated Values											
1.1876	2.478	3.4554	3.8773	3.7046	3.3088	3.1444	3.3818	3.7726	3.8466	3.2942	2.2518
0.42118	2.4137	5.4585	7.0838	6.8804	5.8752	5.4312	6.0691	7.0313	6.9176	4.918	1.7914
-1.4082	-1.8487	1.9693	4.724	5.4352	4.9532	4.652	5.0744	5.4328	4.3378	1.2182	-2.4959
-3.4079	-6.7335	-2.5166	1.2297	3.0098	3.3045	3.2473	3.3073	2.8069	0.62512	-3.4149	-7.3288
-5.0619	-10.799	-6.4331	-2.0153	0.61618	1.5937	1.7613	1.4893	0.24536	-2.7795	-7.4145	-11.328
-6.0969	-13.338	-8.9382	-4.1629	-1.0261	0.38053	0.69303	0.20768	-1.4996	-5.0168	-9.9573	-13.819
-6.4001	-14.08	-9.6781	-4.807	-1.5269	0.0047652	0.35994	-0.18819	-2.0299	-5.6856	-10.706	-14.547
-5.9464	-12.969	-8.572	-3.8457	-0.78088	0.56357	0.85492	0.40069	-1.2396	-4.6871	-9.5862	-13.458
-4.7748	-10.094	-5.7445	-1.4343	1.0529	1.9111	2.039	1.8256	0.71095	-2.1723	-6.7136	-10.635
-3.023	-5.7881	-1.6263	1.9453	3.5214	3.6602	3.5528	3.6872	3.3578	1.3806	-2.5007	-6.3965
-1.0075	-0.88517	2.8106	5.3393	5.838	5.2163	4.8734	5.3579	5.8736	5	2.0973	-1.536
0.70438	3.0078	5.8515	7.2727	6.9585	5.9228	5.4769	6.119	7.1237	7.1409	5.357	2.4067

Table B.3: Exact and Approximated solutions for Example 5, $n = 60$, $f_0 = 1$.

Exact Values											
-9.6437e-22	-1.8713e-18	-8.275e-16	-8.1292e-14	-1.6732e-12	-5.9205e-12	1.1685e-12	5.4293e-12	1.0411e-12	3.6705e-14	2.757e-16	4.6314e-19
-2.2532e-18	-4.3723e-15	-1.9334e-12	-1.8993e-10	-3.9093e-09	-1.3833e-08	2.7302e-09	1.2685e-08	2.4324e-09	8.576e-11	6.4417e-13	1.0821e-15
-1.2519e-15	-2.4292e-12	-1.0742e-09	-1.0552e-07	-2.1719e-06	-7.6854e-06	1.5169e-06	7.0478e-06	1.3514e-06	4.7647e-08	3.5789e-10	6.0121e-13
-1.6539e-13	-3.2092e-10	-1.4191e-07	-1.3941e-05	-0.00028694	-0.0010153	0.0002004	0.00093111	0.00017854	6.2947e-06	4.7282e-08	7.9427e-11
-5.1956e-12	-1.0082e-08	-4.4582e-06	-0.00043796	-0.0090142	-0.031897	0.0062955	0.029251	0.0056088	0.00019775	1.4854e-06	2.4952e-09
-3.8812e-11	-7.5313e-08	-3.3303e-05	-0.0032716	-0.067338	-0.23827	0.047028	0.21851	0.041899	0.0014772	1.1096e-05	1.864e-08
-6.8943e-11	-1.3378e-07	-5.9158e-05	-0.0058115	-0.11961	-0.42325	0.083537	0.38814	0.074426	0.002624	1.971e-05	3.311e-08
-2.9121e-11	-5.6508e-08	-2.4988e-05	-0.0024547	-0.050524	-0.17878	0.035285	0.16395	0.031437	0.0011084	8.3253e-06	1.3985e-08
-2.9249e-12	-5.6757e-09	-2.5098e-06	-0.00024655	-0.0050746	-0.017957	0.0035441	0.016467	0.0031575	0.00011132	8.362e-07	1.4047e-09
-6.9858e-14	-1.3556e-10	-5.9943e-08	-5.8886e-06	-0.0001212	-0.00042887	8.4646e-05	0.00039329	7.5414e-05	2.6588e-06	1.9971e-08	3.3549e-11
-3.9674e-16	-7.6986e-13	-3.4043e-10	-3.3443e-08	-6.8834e-07	-2.4357e-06	4.8073e-07	2.2336e-06	4.283e-07	1.51e-08	1.1342e-10	1.9054e-13
-5.3579e-19	-1.0397e-15	-4.5975e-13	-4.5164e-11	-9.2958e-10	-3.2893e-09	6.4921e-10	3.0165e-09	5.784e-10	2.0393e-11	1.5318e-13	2.5732e-16
Approximated Values											
0.72909	0.4552	0.48451	0.4673	0.4472	0.4633	0.48191	0.45763	0.44837	0.47379	0.48118	0.44886
0.44862	-0.030287	-0.015322	-0.018454	-0.011376	-0.017417	-0.018913	-0.015779	-0.012233	-0.018411	-0.015269	-0.044669
0.45343	-0.0046866	0.0075776	0.0027778	0.0045837	0.0042099	0.0027364	0.003808	0.004371	0.0037432	0.0088339	-0.020182
0.45122	-0.0099802	0.0074268	-0.00047929	-0.00024273	-0.002162	0.0031566	-0.00088429	-0.0044512	0.0003674	0.0076787	-0.025252
0.45073	-0.010274	0.0072667	0.0014437	-0.01335	-0.0469	0.0092427	0.043253	0.0033643	-0.0025241	0.0061831	-0.025024
0.45056	-0.0097915	0.0079314	-0.0028461	-0.097014	-0.20044	0.02894	0.20708	0.060598	-0.00091364	0.0056214	-0.024702
0.45049	-0.009639	0.0079917	-0.008052	-0.15784	-0.29873	0.040668	0.31408	0.10353	0.0017518	0.005489	-0.024744
0.45059	-0.0098638	0.0078568	-0.0012301	-0.075346	-0.16384	0.024445	0.16755	0.045488	-0.0016924	0.005696	-0.024707
0.45077	-0.010382	0.0071275	0.0011412	-0.006529	-0.030264	0.0069004	0.026052	-0.0010571	-0.0021561	0.0063649	-0.025181
0.4515	-0.0093636	0.007762	-0.00033974	-0.00042577	-0.00024159	0.0031832	-0.0015368	-0.0035476	0.001163	0.0081961	-0.024779
0.45374	-0.0043388	0.005934	0.0029151	0.0068453	0.0044583	0.0012059	0.0050594	0.0066369	0.0031399	0.0073941	-0.019633
0.44552	-0.047381	-0.023554	-0.031847	-0.030707	-0.031975	-0.026839	-0.032165	-0.031315	-0.02974	-0.024958	-0.061927

Table B.4: Exact and Approximated solutions for Example 7, $n = 60$, $f_0 = 1$.

Exact Values											
-9.6437e-22	-1.8713e-18	-8.275e-16	-8.1292e-14	-1.6732e-12	-5.9205e-12	1.1685e-12	5.4293e-12	1.0411e-12	3.6705e-14	2.757e-16	4.6314e-19
-2.2532e-18	-4.3723e-15	-1.9334e-12	-1.8993e-10	-3.9093e-09	-1.3833e-08	2.7302e-09	1.2685e-08	2.4324e-09	8.576e-11	6.4417e-13	1.0821e-15
-1.2519e-15	-2.4292e-12	-1.0742e-09	-1.0552e-07	-2.1719e-06	-7.6854e-06	1.5169e-06	7.0478e-06	1.3514e-06	4.7647e-08	3.5789e-10	6.0121e-13
-1.6539e-13	-3.2092e-10	-1.4191e-07	-1.3941e-05	-0.00028694	-0.0010153	0.0002004	0.00093111	0.00017854	6.2947e-06	4.7282e-08	7.9427e-11
-5.1956e-12	-1.0082e-08	-4.4582e-06	-0.00043796	-0.0090142	-0.031897	0.0062955	0.029251	0.0056088	0.00019775	1.4854e-06	2.4952e-09
-3.8812e-11	-7.5313e-08	-3.3303e-05	-0.0032716	-0.067338	-0.23827	0.047028	0.21851	0.041899	0.0014772	1.1096e-05	1.864e-08
-6.8943e-11	-1.3378e-07	-5.9158e-05	-0.0058115	-0.11961	-0.42325	0.083537	0.38814	0.074426	0.002624	1.971e-05	3.311e-08
-2.9121e-11	-5.6508e-08	-2.4988e-05	-0.0024547	-0.050524	-0.17878	0.035285	0.16395	0.031437	0.0011084	8.3253e-06	1.3985e-08
-2.9249e-12	-5.6757e-09	-2.5098e-06	-0.00024655	-0.0050746	-0.017957	0.0035441	0.016467	0.0031575	0.00011132	8.362e-07	1.4047e-09
-6.9858e-14	-1.3556e-10	-5.9943e-08	-5.8886e-06	-0.0001212	-0.00042887	8.4646e-05	0.00039329	7.5414e-05	2.6588e-06	1.9971e-08	3.3549e-11
-3.9674e-16	-7.6986e-13	-3.4043e-10	-3.3443e-08	-6.8834e-07	-2.4357e-06	4.8073e-07	2.2336e-06	4.283e-07	1.51e-08	1.1342e-10	1.9054e-13
-5.3579e-19	-1.0397e-15	-4.5975e-13	-4.5164e-11	-9.2958e-10	-3.2893e-09	6.4921e-10	3.0165e-09	5.784e-10	2.0393e-11	1.5318e-13	2.5732e-16
Approximated Values											
-2.6883e-06	-5.8801e-07	-3.1015e-06	-3.6937e-07	-2.9005e-07	-2.966e-07	1.1942e-07	2.8723e-07	3.0412e-07	5.0387e-07	1.6047e-06	5.8127e-07
-1.1231e-06	1.4495e-08	-3.4866e-07	-5.1804e-08	-6.4215e-09	-1.0489e-08	4.0741e-08	1.3944e-08	5.0651e-09	-2.6832e-07	-5.0431e-07	-8.1015e-09
-1.3095e-06	-9.4719e-10	-5.3637e-07	-1.8801e-07	-2.21e-06	-7.8278e-06	1.3536e-06	7.1177e-06	1.3827e-06	-4.5188e-07	-9.0914e-07	-2.2785e-09
-1.547e-06	-8.5006e-09	-1.0612e-06	-1.4089e-05	-0.0002868	-0.0010129	0.00021138	0.00093075	0.0001785	6.0064e-06	-5.1283e-07	-3.8294e-10
-1.7754e-06	-1.698e-08	-4.9914e-06	-0.00043731	-0.0090093	-0.03187	0.0064258	0.029243	0.0056051	0.00019854	3.9565e-06	2.1767e-08
-1.9385e-06	-3.4434e-08	-3.025e-05	-0.0032689	-0.067342	-0.23855	0.045374	0.21852	0.041899	0.0014777	1.5057e-05	5.6663e-08
-1.9899e-06	-6.6269e-08	-5.3973e-05	-0.0058087	-0.11963	-0.42384	0.078986	0.38811	0.074434	0.0026242	2.183e-05	6.7256e-08
-1.9136e-06	-2.6861e-08	-2.2793e-05	-0.0024522	-0.050524	-0.17895	0.034367	0.16396	0.031434	0.001109	1.2541e-05	5.1015e-08
-1.7331e-06	-1.6196e-08	-3.3413e-06	-0.00024626	-0.005071	-0.01793	0.0036595	0.01646	0.0031549	0.00011193	2.5104e-06	1.5052e-08
-1.4985e-06	-6.3533e-09	-8.9326e-07	-6.0266e-06	-0.00012124	-0.00042835	8.8592e-05	0.00039337	7.5477e-05	2.2456e-06	-7.6299e-07	-1.6389e-09
-1.2657e-06	-4.2414e-10	-4.8144e-07	-1.0803e-07	-7.1125e-07	-2.5051e-06	4.0374e-07	2.2713e-06	4.4757e-07	-4.5614e-07	-8.5686e-07	-2.1399e-09
-1.1042e-06	7.023e-09	-1.9067e-07	-1.2564e-07	-3.4811e-09	-1.0358e-08	3.6657e-08	1.0557e-08	3.1662e-09	-1.5161e-07	-4.5944e-07	-7.0069e-09

Table B.5: Exact and Approximated solutions for Example 7, $n = 60$, $f_0 = 0$.