2006

Applications of regularization techniques to data fitting with PDE models

Alina Alt
San Jose State University

Follow this and additional works at: https://scholarworks.sjsu.edu/etd_theses

Recommended Citation

Alt, Alina, "Applications of regularization techniques to data fitting with PDE models" (2006). *Master's Theses*. 2867.

DOI: <https://doi.org/10.31979/etd.z9n9-ejet>
https://scholarworks.sjsu.edu/etd_theses/2867

This Thesis is brought to you for free and open access by the Master's Theses and Graduate Research at SJSU ScholarWorks. It has been accepted for inclusion in Master's Theses by an authorized administrator of SJSU ScholarWorks. For more information, please contact scholarworks@sjsu.edu.

**APPLICATIONS OF REGULARIZATION TECHNIQUES TO DATA
FITTING WITH PDE MODELS**

A Thesis Presented to
The Faculty of the Department of Mathematics
San Jose State University

In Partial Fulfillment
of the Requirements for the Degree
Master of Science

by Alina Alt

May 2006

UMI Number: 1436892

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI Microform 1436892

Copyright 2006 by ProQuest Information and Learning Company.

All rights reserved. This microform edition is protected against unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

©2006

Alina Alt

ALL RIGHTS RESERVED

APPROVED FOR THE DEPARTMENT OF MATHEMATICS

Igor Malyshev
Professor Igor Malyshev

Roger Dodd
Professor Roger Dodd

Maria Cayco
Professor Maria Cayco

APPROVED FOR THE UNIVERSITY

Rhea I. Williamson

ABSTRACT

APPLICATIONS OF REGULARIZATION TECHNIQUES TO DATA FITTING WITH PDE MODELS

by Alina Alt

The notion of a *well-posed* problem goes back to a famous paper by Jacques Hadamard published in 1902. He stated that many problems that adhere to a physical phenomena should always have a unique solution, which depends continuously on the data.

Problems that are not well-posed in the sense of Hadamard are termed *ill-posed*. *Inverse problems* are concerned with determining causes for a desired or an observed effect, and are often ill-posed.

If a problem is well-posed, then it stands a good chance of solution on a computer using a stable algorithm. If it is not well-posed, it needs to be re-formulated for numerical treatment. Typically this involves including additional assumptions, such as smoothness of solution. This process is known as *regularization*.

This work explores regularization, ill posedness of inverse problems, and ill posedness in general in the context of data fitting with a PDE model.

I dedicate this manuscript to my father who is my teacher, my inspiration, and my
role model.

I would like to thank my thesis advisor, Dr. Malyshev, for all his guidance and help,
and the committee members, Dr. Dodd and Dr. Cayco, for being so pleasant to
work with.

A special thanks to Marv White for his valuable input.

Contents

List of Figures	viii
1 Diffusion Equation and Data Fitting	1
1.1 PDE model	1
1.2 Data fitting problem outline	2
1.3 Ill-posed problems	2
1.4 Inverse problems and generalized inverses	4
1.5 Completely continuous operators and ill posed inverse problems	7
1.6 Data fitting approach outline	10
1.6.1 Derivation of the matrix approach	10
1.6.2 Ill Conditioned matrix equations	12
1.6.3 Almost singular matrices	14
1.6.4 Overcoming ill-posedness using regularization	15
2 Other Partial Differential Models and Data Fitting	19
2.1 Data fitting approach outline	19
2.2 Fundamental solutions and Green's function	20
2.3 PDEs from the Fokker-Planck equation family	24
3 Regularized Compact Linear Operators	32
3.1 Singular value decomposition for a generalized inverse	32
3.2 Regularization methods	36
3.3 Construction of regularization methods	39
4 Simulations	43
4.1 Green's function simulation	43

4.2	Fake data generator	44
4.3	PDE solution time slice reconstruction	46
4.3.1	Singular value perturbation (Tikhonov's regularization)	46
4.3.2	Singular value truncation	50
5	Conclusions	53
	References	54
	Appendix: MATLAB programs	56
A	DiracDelta.m	56
B	InitialCondition.m	57
C	fp.m	58
D	fpGreensFunction.m	60
E	fpGreensFunctionGenerator.m	60
F	DataSetGenerator.m	61
G	fpSolutionReconstructor.m (Tikhonov's Regularization Version)	63
H	fpSolutionReconstructor.m (Singular Value Truncation Version)	64
I	MainSimulator.m	65

List of Figures

Figure 1. Fake Data Plot.	45
Figure 2. Singular Value Perturbation L-Curve.	47
Figure 3. Singular Value Perturbation Data Approximation.	48
Figure 4. Effect of Regularization Parameter ε on Solution x_ε	49
Figure 5. Singular Value Truncation L-Curve.	51
Figure 6. Singular Value Truncation Data Approximation.	52

1 Diffusion Equation and Data Fitting

Diffusion equation will provide an easy introduction to the problem area and applied theory.

1.1 PDE model

The diffusion equation is an important partial differential equation which describes such physical situations as heat conduction in a one-dimensional solid body, spread of a dye in a stationary fluid, population dispersion, and other similar processes. The following is a homogeneous diffusion equation that will be used for introduction to the problem domain.

$$\begin{aligned}\frac{\partial u}{\partial t} &= D \frac{\partial^2 u}{\partial x^2} \\ u(x, 0) &= g(x) \\ u(x, t) &\rightarrow 0, \text{ when } x \rightarrow \pm\infty, (t \geq 0)\end{aligned}\tag{1.1}$$

For these boundary conditions to be consistent, assume that $g(x) \rightarrow 0$ for $x \rightarrow \pm\infty$. In the example of the heat conduction, solution to this equation describes the temperature of an infinite, homogeneous rod at position x and time t . Solutions of the heat equation are characterized by a gradual smoothing of the initial temperature distribution by the flow of heat from warmer to colder areas of an object. The solution of the diffusion equation has the following form:

$$u(x, t) = \int_{-\infty}^{\infty} K(x - y, t) g(y) dy, \quad t > 0,\tag{1.2}$$

where $K(x, t) = \frac{\theta(t)}{\sqrt{4\pi Dt}} \exp(-\frac{x^2}{4Dt})$ is the Gaussian kernel. K is also the Green's

function of the equation ($\theta(t)$ is the Heaviside function).

The integral representation of the diffusion model is dealt with from here on.

1.2 Data fitting problem outline

A set of measurements in some moment in time is given by:

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\},$$

Consider heat diffusion model in one-dimensional body: x_i describes the position on the body and y_i would describe a temperature measurement. Since this model fits the data, the data-set is considered to be a result of evolution in time of some unknown initial condition $g(x)$. This evolution is described with 1.2. Since the exact moment in time these measurements were obtained is not known, an arbitrary $t_* > 0$ is chosen and is used to fit the data set with $u(x, t_*)$ described by 1.2. Finding $u(x, t_*)$ involves some reconstruction of the function $g(x)$ and this reconstruction procedure will turn out to be *ill-posed*.

1.3 Ill-posed problems

The rapidly increasing use of computational technology requires the development of computational algorithms for solving broad class of problems.

One important property of mathematical problems is the *stability of their solutions to small changes in the initial data*. Problems that fail to satisfy this stability condition are called, following Hadamard, *ill posed*. For a long time mathematicians felt that ill-posed problems cannot describe real phenomena and objects. However, as it appears to be, many classical mathematical problems are ill posed, and, most significantly, many of these problems have important applications.

According to Hadamard, the conditions for a **well posed problem** are:

1. **Existence:** The solution must exist.
2. **Uniqueness:** The solution must be uniquely determined.
3. **Stability:** The solution must depend continuously on the initial data and/or boundary data (i.e., slight variation of the data causes the solution to vary only slightly). The data is usually a subject to numerical approximations, and so it is required for the solution to be stable under small variations in initial and/or boundary values.

These conditions represent reasonable requirements for a problem arising in a physical context. In particular, an ill posed problem that doesn't possess the stability property makes it difficult to give a correct physical interpretation of the measurement results. This paper will mostly concentrate on the instability of a problem and the ways to overcome that difficulty.

A vast majority of *inverse* problems concerned with finding a cause for an observed effect, are ill posed, and are reduced to solving an integral equation of the first kind.

Discrete versions of ill posed integral equations or inverse problems might satisfy the conditions for well-posedness described above. Therefore, it makes sense to use a different measure for ill-posedness for discrete problems : a condition number.

In order for the developed algorithms or theories to be applicable in real life situations, when faced with an inverse problem, or similarly, an integral equation of the first kind, whether it is in a discrete form or not, the problem must be well posed. If it's not, the problem settings must be modified accordingly, or in other words, consider a *regularized* version of the problem.

1.4 Inverse problems and generalized inverses

Consider integral equation 1.2 for a fixed moment in time t_* . The time can be dropped from the notation because it is fixed, and denote $K(x - y, t_*)$ by $K(x, y)$ for simplicity, $u(x, t_*)$ by $u(x)$, and $u(x, 0)$ by $u_0(x)$. Also, since $u(x) \rightarrow 0$ for $x \rightarrow \pm\infty$, it can be assumed that $u(x) = 0$ and $u_0(x) = 0$ outside some interval $[a, b]$.

Thus the following relation between u and u_0 :

$$u(x) = \int_a^b K(x, y) u_0(y) dy \quad (1.3)$$

$A : L_2[a, b] \rightarrow L_2[a, b]$ will denote the operator

$$\int_a^b K(x, s) \bullet ds \quad (1.4)$$

Then

$$u(x) = Au_0(x). \quad (1.5)$$

Now assume there is a continuum of measurement points and that for all i , $x_i \in [a, b]$. Without loss of generality let's assume that $y(x) \in L_2[a, b]$. The task at hand is to find $u(x)$ described by 1.5 that fits this data-set. Equivalently, $f \in L_2[a, b]$ must be obtained for which Af would approximate y , or in other words: a solution to the equation $Af = y$.

This is an integral equation of the first kind, and as it was mentioned earlier, many inverse, ill conditioned problems are reduced to an equation of this type. So in order for this equation to represent a well-posed problem, an inverse operator of A should exist and it should be continuous. An exact inverse might not exist, but a

generalized inverse might be a good substitute.

Definition 1 Consider an equation

$$Ax = y \tag{1.6}$$

where $A : X \rightarrow Y$, a bounded linear operator acting between Hilbert spaces X and Y . An element $x \in X$ is called

- *least-squares* solution of 1.6 if

$$\|Ax - y\|_Y = \inf\{\|Az - y\|_Y \mid z \in X\} \tag{1.7}$$

- *best-approximate* solution or *minimal-norm* solution of 1.6 if

$$\|x\| = \inf\{\|z\| \mid z \text{ is least squares solution to 1.6}\} \tag{1.8}$$

Theorem 2 The least squares solution to equation 1.6 will exist if and only if there exists a projection of y to the range of the operator A , R_A .

Proof. Straight-forward from the definition of projection:

Let x denote an element and Y , a subset of space X . An element y of the set Y is called a projection of element x onto the set Y (denoted as $y = Px$) if $\|x - y\|_X = \inf_{z \in Y} \|x - z\|$. ■

Let X be a Hilbert space, and $Y \subset X$. Note that a projection function is well defined for all $x \in X$ only when Y is a closed subspace of X .

Also note that for each minimizing x , Ax is a projection of y onto the range of operator A . So in order for such x to exist, the projection should be well defined onto the range of operator A . For arbitrary operator A the existence of projection cannot be guaranteed, since it is not provided that the operator is a closed space. In addition such x might not be unique if the null space N_A of the operator is nonzero. But if a least-squares solution exists, then the minimal-norm solution is unique, because it is the minimizer of a strictly convex (quadratic) functional on a linear subspace.

Bounded operators have closed ranges. As it will be seen in the next section, the integral operator with a Hilbert-Schmidt kernel is bounded, and therefore it has a closed range. The theory for finding a minimal-norm solution will be derived for operators with closed ranges, since in this case a least-squares solution will exist for any $y \in Y$.

Let $\tilde{A} : N_A^\perp \rightarrow R_A$ denote the restriction of A . ($X = N_A \oplus N_A^\perp$ since N_A is a closed subspace. Therefore $AX = AN_A^\perp = \tilde{A}N_A^\perp$.)

Definition 3 *The Moore-Penrose generalized inverse(also referred to as pseudo-inverse) $A^\dagger$ is defined as the unique linear extension of $\tilde{A}^{-1}$ to $D(A^\dagger) := R_A \oplus R_A^\perp$ (Since R_A is assumed to be closed space, $Y = R_A \oplus R_A^\perp = D(A^\dagger)$) with $N_{A^\dagger} = R_A^\perp$.*

Note that the Moore-Penrose inverse $A^\dagger$ is well-defined: First of all, due to the restriction to $N_A^\perp$ and R_A , the operator $\tilde{A}$ is injective and surjective, and hence, $\tilde{A}^{-1}$ exists. As a consequence, $A^\dagger$ is well-defined on R_A . Now for arbitrary $y \in D(A^\dagger)$ there exists a unique $y_1 \in R_A$ and $y_2 \in R_A^\perp$ such that $y = y_1 + y_2$. Thus $A^\dagger y = A^\dagger y_1 + A^\dagger y_2 = \tilde{A}^{-1}y_1$ obtained using linearity and the fact that $N_{A^\dagger} = R_A^\perp$.

Theorem 4 *For each $y \in D(A^\dagger)$, the equation 1.6 has a unique minimal-norm*

solution given by $x^\dagger := A^\dagger y$. The set of all least-squares solutions is given by $\{x^\dagger\} + N_A$.

Theorem 5 For given $y \in D(A^\dagger)$, $x \in X$ is a least-squares solution of 1.6 if and only if x satisfies the Gaussian normal equation

$$A^*Ax = A^*y \quad (1.9)$$

Proof. An element $x \in X$ is a least-squares solution if and only if Ax is a projection of y onto R_A , which is equivalent to $Ax - y \in R_A^\perp$. In general, $u \in R_A^\perp$ is equivalent to $\langle Av, u \rangle = 0$, $\forall v \in X$. This is equivalent to $\langle v, A^*u \rangle = 0$, $\forall v \in X$, which means that $u \in N_{A^*}$, therefore $N_{A^*} = R_A^\perp$. And so $A^*(Ax - y) = 0$ which is equivalent to 1.9. ■

Note that all the results above are applicable in the finite-dimensional case as well.

1.5 Completely continuous operators and ill posed inverse problems

As it has been observed in the previous section, if A is an operator with a closed range, then it has a generalized inverse that will provide first two properties of well-posedness for equation 1.6: namely, an existence and uniqueness of solution. But what about the third property: stability?

For 1.6 to satisfy the third property of well-posedness, the generalized inverse $A^\dagger$ should be a continuous operator. It turns out that in many important cases, for example, for a so called completely continuous operator A , the general inverse operator $A^\dagger$ is not continuous.

Definition 6 A linear operator $A : X \rightarrow Y$, where both spaces are linear and normed, is **completely continuous (or compact)**, if it maps any bounded set of the functions in the space X into a precompact set of the space Y . In other words: each bounded infinite sequence in E_1

$$f_1, f_2, \dots, f_i, \dots \quad \|f_i\|_X \leq c.$$

is mapped in E_2 into a sequence:

$$Af_1, Af_2, \dots, Af_i, \dots$$

and from that sequence a Cauchy subsequence $Af_{i_1}, Af_{i_2}, \dots, Af_{i_k}, \dots$ can be extracted.

Theorem 7 Let $A : X \rightarrow Y$ be a compact linear operator between the infinite-dimensional Hilbert spaces X and Y , such that the dimension of R_A is infinite. Then $A^\dagger$ is discontinuous.

Proof. Since X and R_A are infinite-dimensional, there exists a sequence (x_n) with $x_n \in N_A^\perp$,

$$\|x_n\| = 1, \text{ and } \langle x_i, x_j \rangle = 0 \text{ for } i \neq j.$$

Since A is a compact operator, the sequence $(y_n) := (Ax_n)$ is compact and hence for each $\delta > 0$, there exist k and l , such that $\|y_k - y_l\| < \delta$, but

$$\|A^\dagger y_k - A^\dagger y_l\|^2 = \|x_k - x_l\|^2 = \|x_k\|^2 + \|x_l\|^2 - 2\langle x_k, x_l \rangle = 2.$$

Hence $A^\dagger$ is not continuous. ■

Theorem 8 The operator $A : L_2[a, b] \rightarrow L_2[a, b]$, defined by 1.4, is completely continuous:

Proof. The proof will apply for any integral operator with a Hilbert-Schmidt Kernel (Kernel that belongs to $L_2([a, b] \times [a, b])$).

First, it will be shown that

$$\|A\|_{op} \leq \|K\|_{L_2([a,b] \times [a,b])} \quad (1.10)$$

$$\|A\varphi\|^2 = \int_a^b \left| \int_a^b K(s, t) \varphi(t) dt \right|^2 ds.$$

$$\text{But } \left| \int_a^b K(s, t) \varphi(t) dt \right|^2 \leq \int_a^b |K(s, t)|^2 dt \int_a^b |\varphi(t)|^2 dt = \|\varphi\|^2 \int_a^b |K(s, t)|^2 dt, \text{ so}$$

$$\|A\varphi\|^2 \leq \|\varphi\|^2 \int_a^b \int_a^b |K(s, t)|^2 ds dt = \|K\|_{L_2([a,b] \times [a,b])}^2.$$

Therefore operator A is bounded.

Now let $\{\psi_n\}$ be a complete orthonormal system in $L_2[a, b]$. Therefore the set of products : $\psi_m(s)\psi_n(t)$ make a complete system in $L_2([a, b] \times [a, b])$ and therefore

$$K(s, t) = \sum_{m,n=1}^{\infty} a_{mn} \psi_m(s) \psi_n(t).$$

Lets introduce a series of Kernels that are the partial sums of the above series: $K_N(s, t) = \sum_{m,n=1}^N a_{mn} \psi_m(s) \psi_n(t)$, and let A_N be an integral operator defined by $K_N(s, t)$. That operator is compact, because it transforms $L_2[a, b]$ into a finite dimensional space induced by vectors $\psi_1, \dots, \psi_N$.

$$A_N \varphi = \int_a^b K_N(s, t) \varphi(t) dt = \sum_{m,n=1}^N a_{mn} \psi_m(s) \int_a^b \varphi(t) \psi_n(t) dt = \sum_{m=1}^N \psi_m(s) \sum_{n=1}^N a_{mn} b_n,$$

$$\text{where } b_n := \int_a^b \varphi(t) \psi_n(t) dt.$$

Since K_N is a partial sum of the series for K the following takes place:

$$\int_a^b \int_a^b (K(s, t) - K_N(s, t))^2 ds dt \rightarrow 0, \text{ when } N \rightarrow \infty.$$

Now the property 1.10 can be applied to operator $A - A_N$. Making $\|A - A_N\|_{op} \rightarrow$

0 as $N \rightarrow \infty$. The fact that the limit of a sequence of compact operators is a compact operator concludes the proof. ■

It was just shown that A is completely continuous, making $A^\dagger$ discontinuous using the preceding theorem. Thus the problem of solving 1.6 is ill-posed.

1.6 Data fitting approach outline

1.6.1 Derivation of the matrix approach

In reality it is impossible to obtain a continuum of measurements, so a discrete problem must be solved. In this case a function $u(x, t_*)$, or an approximation to that function for some moment in time t_* that fits a finite data-set

$$\{(x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)\}$$

must be found.

As it was done in the previous section, t_* can be dropped from the notation and representation 1.3 can be used for $u(x, t_*)$.

So $u(x) = \int_a^b K(x, y)u_0(y)dy$. The data-set defines a partition using which one

can approximate the integral with a Reiman sum: $u(x) = \sum_{j=1}^n K(x, x_j)u_0(x_j)\Delta x_j$.

Denote $u_0(x_j)\Delta x_j$ by c_j .

So

$$u(x) = \sum_{j=1}^n c_j K(x, x_j)$$

and

$$u(x_i) = \sum_{j=1}^n c_j K(x_i, x_j) \tag{1.11}$$

It is natural to seek $\mathbf{u} = [u(x_i)]_{\mathbb{R}^n}$ that would minimize the error of between a vector of type 1.11 and initial data $\mathbf{y} = [y_i]_{\mathbb{R}^n}$. Thus $\mathbf{u}$ should be the minimum of

$$\{\|\mathbf{y} - \mathbf{u}\|_{\mathbb{R}^n} / \mathbf{u} = \mathbf{K}\mathbf{c}, \mathbf{c} \in \mathbb{R}^n\} \quad (1.12)$$

where $\mathbf{K} = (K(x_i, x_j))_{i,j}$, $\mathbf{c} = (c_j)_j$ is the unknown vector of coefficients of $\mathbf{u}$. In other words, to find such $\mathbf{u}$, one needs to find a least squares solution to the *Kernel matrix equation*:

$$\mathbf{K}\mathbf{c} = \mathbf{y} \quad (1.13)$$

Proving invertability of matrix $\mathbf{K}$ is not a trivial task and is in fact unnecessary since it is always possible to find a generalized inverse. But it is worthwhile noting an important result that will warranty $\mathbf{K}$'s invertability and will therefore maybe ease the computation of the solution. To do that, some auxiliary theory will be necessary.

Definition 9 Let $K : X \times X \rightarrow \mathbb{R}$ be continuous and symmetric function. K is called **Positive Definite** if for any finite set $\{x_1, \dots, x_n\} \in X$, the $k \times k$ matrix $\mathbf{K}$ whose i, j entry is $K(x_i, x_j)$ is positive definite (i.e. $\langle \mathbf{K}\mathbf{x}, \mathbf{x} \rangle \geq 0 \forall \mathbf{x} \in \mathbb{R}^k$). Such function is also referred to as a **Mercer Kernel**.

Definition 10 A function $f : [0, \infty] \rightarrow \mathbb{R}$ is called **Completely Monotonic** if it is infinitely differentiable, and for all $r > 0$ and $k \geq 0$, $(-1)^k f^{(k)}(x) > 0$ (here $f^{(k)}(x)$ denotes the k^{th} derivative of f).

Theorem 11 Let $X \subset \mathbb{R}$, $f : [0, \infty] \rightarrow \mathbb{R}$ and $K : X \times X \rightarrow \mathbb{R}$ be defined by $K(x, y) = f(\|x - y\|^2)$. If f is completely monotonic, then K is positive definite.

Proof. The proof of this theorem is beyond the scope of this paper and can be found in Schoenberg's paper [13]. ■

Corollary 12 *The Gaussian kernel is a positive definite-function.*

Proof. Indeed, this can be seen that by taking $f(r) = e^{-r/c}$, where c is some positive constant. ■

This result will yield a positive-definite matrix $\mathbf{K}$. Note that positive-definite matrices are invertible. Therefore, since matrix $\mathbf{K}$ is symmetric and positive definite it is nonsingular. But it is important to note that since 1.13 is just a discrete version of an ill-posed integral equation, $\mathbf{K}$ will be almost singular or ill-conditioned.

1.6.2 Ill Conditioned matrix equations

Consider a linear matrix equation $\mathbf{Ax} = \mathbf{y}$, where $\mathbf{A} \in \mathbb{R}^{n \times n}$, is a symmetric, positive definite matrix (therefore nonsingular so the solution always exists). From the spectral theory of symmetric matrices, there exist eigenvalues $\lambda_i \in \mathbb{R}$ and corresponding eigenvectors $u_i \in \mathbb{R}^n$ ($\|u_i\| = 1$), such that $\mathbf{A}$ has a representation of the form

$$\mathbf{Ax} = \sum_{i=1}^n \lambda_i \mathbf{u}_i \langle \mathbf{u}_i, \mathbf{x} \rangle \quad (1.14)$$

For positive semi-definite matrices these eigenvalues would satisfy: $0 \leq \lambda_1 \leq \dots \leq \lambda_n$ and for positive definite $\lambda_1 > 0$. Also these eigenvectors represent an orthonormal basis of $\mathbb{R}^n$.

It is well-known that the condition number ($\lambda_1 \neq 0$ since the matrix is positive definite) of $\mathbf{A}$ is given by the quotient of largest and smallest eigenvalue, i.e., $\kappa = \frac{\lambda_n}{\lambda_1}$. For simplicity it can be assumed that the scaling is such that $\lambda_n = 1$, then the condition number corresponds to $\kappa = \lambda_1^{-1}$.

The condition number is a measure for the stable solvability of the problem. Assume that there is noisy data $\mathbf{y}^\delta$ instead of $\mathbf{y}$, which satisfies $\|\mathbf{y}^\delta - \mathbf{y}\| \leq \delta$ in the Euclidean norm on $\mathbb{R}^n$. Let $\mathbf{x}^\delta$ denote the solution with right-hand side $\mathbf{y}^\delta$.

Then, from the spectral representation one can obtain

$$\mathbf{x}^\delta - \mathbf{x} = \sum_{i=1}^n \lambda_i^{-1} \mathbf{u}_i \langle \mathbf{u}_i, (\mathbf{y}^\delta - \mathbf{y}) \rangle.$$

Hence, it is possible to estimate (using orthogonality of eigenvectors)

$$\|\mathbf{x}^\delta - \mathbf{x}\|^2 = \sum_{i=1}^n \lambda_i^{-2} |\langle \mathbf{u}_i, (\mathbf{y}^\delta - \mathbf{y}) \rangle|^2 \leq \lambda_1^{-2} \|\mathbf{y}^\delta - \mathbf{y}\|^2$$

or in other words

$$\|\mathbf{x}^\delta - \mathbf{x}\| \leq \kappa \|\mathbf{y}^\delta - \mathbf{y}\| \leq \kappa \delta. \quad (1.15)$$

The sharpness of this estimate can be seen immediately for $\mathbf{y}^\delta - \mathbf{y} = \delta \mathbf{u}_1$.

One observes that with increasing condition number of the matrix $\mathbf{A}$, the noise amplification increases in the worst-case. For large κ one therefore speaks of an *ill-conditioned problem*. Note that a finite-dimensional linear problem is never ill-posed in the sense that the third condition in Hadamards definition is violated, but for κ large one certainly approximates this case.

In case of $\mathbf{A}$ being a positive semi-definite matrix the vector space $\mathbb{R}^n$ can be spanned with the eigenvector system of $\mathbf{A}$. Or in other words: $\mathbf{Ax} = \sum_{i=m}^n \lambda_i \mathbf{u}_i \langle \mathbf{u}_i, \mathbf{x} \rangle$,

where λ_m denotes the minimal nonzero eigenvalue.

Also, $\mathbb{R}^n = N(\mathbf{A}) + R(\mathbf{A})$, so the range is spanned by the eigenvectors $\mathbf{u}_i$, $i \geq m$.

Since $\mathbf{x}$ is a least squares solution, $\mathbf{Ax}$ is the projection of $\mathbf{y}$ onto the range.

Therefore

$$\sum_{i=m}^n \lambda_i \mathbf{u}_i \langle \mathbf{u}_i, \mathbf{x} \rangle = \sum_{i=m}^n \mathbf{u}_i \langle \mathbf{u}_i, \mathbf{y} \rangle.$$

So

$$\lambda_i \langle \mathbf{u}_i, \mathbf{x} \rangle = \langle \mathbf{u}_i, \mathbf{y} \rangle \iff \langle \mathbf{u}_i, \mathbf{x} \rangle = \lambda_i^{-1} \langle \mathbf{u}_i, \mathbf{y} \rangle,$$

and thus a minimal norm solution would be spanned by eigenvectors $\mathbf{u}_i$, $i \geq m$ and would satisfy

$$\mathbf{x} = \sum_{i=m}^n \mathbf{u}_i \langle \mathbf{u}_i, \mathbf{x} \rangle = \sum_{i=m}^n \lambda_i^{-1} \langle \mathbf{u}_i, \mathbf{y} \rangle.$$

Therefore:

$$\mathbf{x}^\delta - \mathbf{x} = \sum_{i=m}^n \lambda_i^{-1} \mathbf{u}_i \langle \mathbf{u}_i, (\mathbf{y}^\delta - \mathbf{y}) \rangle.$$

Similarly to the above estimate, it can be seen that

$$\|\mathbf{x}^\delta - \mathbf{x}\| \leq \lambda_m^{-1} \|\mathbf{y}^\delta - \mathbf{y}\| \leq \lambda_m^{-1} \delta.$$

Thus the noise amplification is actually determined by minimal nonzero eigenvalue.

For arbitrary matrices similar arguments can be applied with the aid of singular value decomposition to prove that the noise amplification is determined by the minimal singular value.

1.6.3 Almost singular matrices

A matrix is called *almost singular* if minor changes to its elements could cause it to become a singular matrix. This property is of a major importance in computational algorithms today, since computers have a limited floating point precision which may become a road block in trying to compute an inverse of a matrix that might be well-behaved on paper.

As it turns out, the determinant of a matrix does not indicate whether the matrix is almost singular. As an example, consider a diagonal matrix 11×11 whose diagonal elements are equal to 10 except for last one that is equal to 10^{-10} . The determinant of such matrix is equal to 10^{-10} but with a slight change of the last element it would turn to a singular matrix. Another example is a matrix $A = \frac{1}{2}I$. One needs to go to great length to turn it to a singular matrix by slightly change the matrix elements, but if I is of a high enough order, $\det A$ could get extremely close to 0.

Consider a positive definite matrix A with eigenvalues $0 < \lambda_1 \leq \dots \leq 1$. There exist orthogonal matrices S, P , such that $A' = SAP$ and A' is a diagonal matrix with λ_i on the diagonal. Matrix A' can be turned to a singular matrix by adding to it a matrix F' , such that $\|F'\| = \lambda_1$ (for example, a matrix that is all 0, except for $-\lambda_1$ on the diagonal in the same place where A' has it. Suppose $\det(A' + F') = 0$. Thus $\det(A' + F') = \det S \det P \det(A - S^{-1}F'P^{-1}) = 0$. Therefore A can be turned into a singular matrix by adding a matrix $F = S^{-1}F'P^{-1}$. Note, that $\|F\| = \|F'\| = \lambda_1$, or, using the terminology above, $\|F\| = \frac{1}{\kappa}$, where κ is the condition number of A . Therefore, the original matrix A can be turned into a singular matrix by perturbing it with a matrix whose norm is κ^{-1} , and the bigger the condition number, the more sensitive the matrix is for small changes.

Note that for an arbitrary matrix A it can be shown, with the aid of singular value decomposition, that sensitivity of matrix A to small changes is measured by its minimum singular value.

1.6.4 Overcoming ill-posedness using regularization

Both difficulties that arise with solving a matrix equation have the same origin: small singular values, or small eigenvalues if the matrix is symmetric and positive semi-definite. Regularization techniques manipulate the singular values in attempt of making the problem better conditioned:

Singular value perturbation

It seems natural to approximate $\mathbf{A}$ by a family of matrices, whose smallest singular values are shifted away from zero.

For a symmetric positive definite matrix $\mathbf{A}$ a simple candidate is matrix $\mathbf{A}_\varepsilon := \varepsilon \mathbf{I} + \mathbf{A}$, $\varepsilon > 0$, a perturbed version of the matrix $\mathbf{A}$.

It is easy to see that eigenvalues of $\mathbf{A}_\varepsilon$ are given by $\lambda_i + \varepsilon$ and eigenvectors are the same as for $\mathbf{A}$.

Thus

$$\mathbf{x} - \mathbf{x}_\varepsilon = \sum_{i=1}^n (\lambda_i^{-1} - (\lambda_i + \varepsilon)^{-1}) \mathbf{u}_i \langle \mathbf{u}_i, \mathbf{y} \rangle = - \sum_{i=1}^n \frac{\varepsilon}{\lambda_i(\lambda_i + \varepsilon)} \mathbf{u}_i \langle \mathbf{u}_i, \mathbf{y} \rangle$$

, for $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$ and $\mathbf{x}_\varepsilon = \mathbf{A}_\varepsilon^{-1}\mathbf{y}$

Approximation error $E_a(\varepsilon)$ of this perturbation can be estimated by

$$E_a(\varepsilon) := \|\mathbf{x} - \mathbf{x}_\varepsilon\| \leq \frac{\varepsilon}{\lambda_1(\lambda_1 + \varepsilon)} \|\mathbf{y}\|$$

and in particular $E(\varepsilon)$ decays to zero for $\varepsilon \rightarrow 0$.

Besides an approximation error for exact data $\mathbf{y}$, one can derive an error estimate for perturbed solution with noisy data from the spectral representation (i.e., the error for the difference $\mathbf{x}_\varepsilon$ and $\mathbf{x}_\varepsilon^\delta = \mathbf{A}_\varepsilon^{-1}\mathbf{y}^\delta$):

$$\mathbf{x}_\varepsilon^\delta - \mathbf{x}_\varepsilon = \sum_{i=1}^n (\lambda_i + \varepsilon)^{-1} \mathbf{u}_i \langle \mathbf{u}_i, (\mathbf{y}^\delta - \mathbf{y}) \rangle.$$

The error $E_r(\varepsilon, \delta)$ can be estimated as

$$E_r(\varepsilon, \delta) := \|\mathbf{x}_\varepsilon^\delta - \mathbf{x}_\varepsilon\| \leq \frac{1}{\lambda_1 + \varepsilon} \|\mathbf{y}^\delta - \mathbf{y}\| \leq \frac{\delta}{\lambda_1 + \varepsilon}$$

In this case, the closer ε is to 0, the closer the error is to $\frac{\delta}{\lambda_1}$. Therefore, unlike the case with $E_a(\varepsilon)$, $E_r(\varepsilon, \delta)$ gets better with increasing the value of ε .

Singular value truncation

For each particular problem, taking the precision error of the initial data and the rounding errors of singular value calculations into consideration, some ε is set as a threshold: if a singular value is less than ε , it is considered to be 0. New matrix $\mathbf{A}_\varepsilon$ is better conditioned but some information from matrix $\mathbf{A}$ might be lost.

Again, consider the error estimate for $\mathbf{x} = \mathbf{A}^{-1}\mathbf{y}$ and $\mathbf{x}_\varepsilon = \mathbf{A}_\varepsilon^{-1}\mathbf{y}$:

$\mathbf{x} - \mathbf{x}_\varepsilon = \sum_{i=1}^m \lambda_i^{-1} \mathbf{u}_i \langle \mathbf{u}_i, \mathbf{y} \rangle$, where m is the last $\lambda_i < \varepsilon$. As ε gets bigger, the number of summands increases, which in turn increases the error $\|\mathbf{x} - \mathbf{x}_\varepsilon\|$.

On the other hand, one can obtain error estimate for the perturbed solution with noisy data from the spectral representation (i.e., the error for the difference $\mathbf{x}_\varepsilon$ and $\mathbf{x}_\varepsilon^\delta = \mathbf{A}_\varepsilon^{-1} \mathbf{y}^\delta$):

$$\mathbf{x}_\varepsilon^\delta - \mathbf{x}_\varepsilon = \sum_{i=m}^n \lambda_i^{-1} \mathbf{u}_i \langle \mathbf{u}_i, (\mathbf{y}^\delta - \mathbf{y}) \rangle$$

Therefore

$$\|\mathbf{x}_\varepsilon^\delta - \mathbf{x}_\varepsilon\| \leq \frac{1}{\varepsilon} \|\mathbf{y}^\delta - \mathbf{y}\| \leq \frac{\delta}{\varepsilon}.$$

In other words, the bigger ε is, the better the error is.

Analysis

In general, both techniques offer ways to overcome ill-posedness but one can't get carried away, since as the problem becomes better posed (bigger ε), the regularized solution gets further away from the exact solution and vice versa.

In Simulations section one heuristic for finding such ε would be presented that would try to get the best of both worlds.

Equation 1.13 is ill-conditioned, since it approximates an ill-posed continuum problem. Therefore, a regularized versions of this equation should be considered. $\mathbf{K}$ is a symmetric positive definite matrix, so a perturbed matrix equation can be solved instead:

$$(\varepsilon \mathbf{I} + \mathbf{K}) \mathbf{c} = \mathbf{y}. \quad (1.16)$$

For the second regularization method all singular values smaller ε will be truncated.

Denote the solution vector to this equation as $\mathbf{c}_\varepsilon$ while the solution vector to **1.13** remains to be $\mathbf{c}$, therefore $\mathbf{c}_\varepsilon \xrightarrow{\varepsilon \rightarrow 0} \mathbf{c}$.

Also $\mathbf{u}_\varepsilon = \mathbf{K}\mathbf{c}_\varepsilon \xrightarrow{\varepsilon \rightarrow 0} \mathbf{u} = \mathbf{K}\mathbf{c}$, since $\|\mathbf{u}_\varepsilon - \mathbf{u}\| = \|\mathbf{K}\mathbf{c}_\varepsilon - \mathbf{K}\mathbf{c}\| \leq \|\mathbf{K}\| \|\mathbf{c}_\varepsilon - \mathbf{c}\|$, with the upper bound approaching 0 as $\varepsilon \rightarrow 0$.

Thus $\mathbf{u}_\varepsilon$ approximates the minimizer of 1.12.

A general theory of Regularization and Singular Value Decomposition will be unfolded in chapter 4.

2 Other Partial Differential Models and Data Fitting

What good is a solution approach if it cannot have more than one simple application? The data fitting approach presented in the previous section for the diffusion PDE does not let down.

2.1 Data fitting approach outline

The above techniques and theory are applicable not only for a simple diffusion PDE, but also for a wide class of PDEs with linear differential operators for which it's possible to find the so called *Green's function*.

Remember that the key to the approach lies in the fact that the solution to the PDE can be represented as a convolution of the Green's function with the initial condition function. Suppose that in the integral representation there is an arbitrary function G instead of a Gaussian kernel function K . Then, by taking all the steps described in the previous section the discrete data fitting problem can get reduced to solving a matrix equation:

$$\mathbf{G}\mathbf{c} = \mathbf{y} \tag{2.1}$$

Matrix $\mathbf{G}$ is not necessarily symmetric and positive definite, therefore the existence of an exact solution cannot be guaranteed. This means that a least squares solution should be sought. According to the previous section, a least squares solution has to satisfy the normal Gaussian equation :

$$\mathbf{G}^T\mathbf{G}\mathbf{c} = \mathbf{G}^T\mathbf{y} \tag{2.2}$$

Equation 2.1 is ill-conditioned, since it approximates an ill-posed continuum problem. Thus regularization methods should be applied to modify the equation.

Equation 2.2 is analogous to equation 1.13. Therefore, when singular value perturbation method is used, it makes sense to modify equation 2.2 instead of equation 2.1 to obtain the following:

$$(\varepsilon \mathbf{I} + \mathbf{G}^T \mathbf{G}) \mathbf{c} = \mathbf{G}^T \mathbf{y}. \quad (2.3)$$

Singular truncation method will be applied to the equation 2.1 as is.

To continue, some background about Green's function is required.

2.2 Fundamental solutions and Green's function

Definition 1 A *distribution or generalized function* is a continuous linear functional over a space of infinitely differentiable functions of compact support (zero outside of a compact set), with a particular topology.

The most commonly encountered generalized function is the *delta function*.

Definition 2 *Delta function* δ is a generalized function such that, when it acts on a function from the domain space f , it gives the value at 0 of f . Namely $\langle \delta, f \rangle = f(0)$. Also $f * \delta = f$.

Let $L(D)$ be a differential operator with constant coefficients.

Definition 3 Then a *fundamental solution* for $L(D)$ is a distribution G , satisfying the equation:

$$L(D)G = \delta .$$

The significance of the fundamental solution lies in the fact that:

$$L(D)(G * f) = (L(D)G) * f = \delta * f = f$$

Therefore $G * f$ is a solution of the inhomogeneous equation $L(D)u = f$. Such fundamental solutions allow solution of $L(D)u = f$ on all of space. They are also referred in the literature as "free space" fundamental solutions. In practical applications, however, one does not usually want to solve problems posed on all of space; rather one wants to solve PDEs on some domain, subject to certain initial/boundary conditions. *Green's functions* are the analogue of fundamental solutions for this situation. They are also referred to as "fundamental solutions of an initial value problem."

Suppose there is a Cauchy problem at hand where $L(D)$ is a linear differential operator of $u(x, t)$. The differential operator is a sum of different order differentials by x and t , with coefficients that depend on x and t .

$$L(D)u = 0 \quad t > 0 \tag{2.4}$$

$$u(x, 0) = f(x) \tag{2.5}$$

First consider the "distributional" problem:

$$L(D)G(x, t; x', 0) = 0, \quad t > 0 \tag{2.6}$$

$$G(x, 0; x', 0) = \delta(x - x') \tag{2.7}$$

Definition 4 A G satisfying 2.6, 2.7 is called a **Green's function** for the initial value problem. $G(x, t; x', t')$ depends on the position x' of the delta function (point source of unit of intensity) at time t' , and must predict disturbance at any other point x at any subsequent time t .

Provided with G , the solution to 2.4, 2.5 is given by the following superposition integral:

$$u(x, t) = \int_{-\infty}^{\infty} G(x, t; y, 0) f(y) dy, \quad (2.8)$$

Really,

$$L(D)u = \int_{-\infty}^{\infty} L(D)G(x, t; y, 0) f(y) dy = 0,$$

and

$$u(x, 0) = \int_{-\infty}^{\infty} G(x, 0; y, 0) f(y) dy = \int_{-\infty}^{\infty} \delta(x - y) f(y) dy = f(x)$$

In the inhomogeneous version of 2.4 i.e.: $L(D)u = g(x, t)$, with the same initial conditions, the solution would be composed of the solution to the homogeneous problem with the initial condition, and the solution to the inhomogeneous problem with 0 initial condition.

It is possible to express the solution to an inhomogeneous initial-value problems through Green's function for a special kind of linear differential operators: $\partial_t - L$, where L is a constant coefficient differential operator in x :

Consider the Cauchy problem:

$$(\partial_t - L)u(x, t) = g(x, t), \quad x \in \mathbb{R}, \quad t > 0 \quad (2.9)$$

$$u(x, 0) = f(x), \quad x \in \mathbb{R} \quad (2.10)$$

As it was mentioned above,

$$u(x, t) = u_h(x, t) + u_r(x, t) \quad (2.11)$$

where, $u_h(x, t)$ is the solution to the homogeneous problem with the initial condition, and $u_r(x, t)$ is the solution to the inhomogeneous problem with 0 as the initial condition.

Using the theory above, the solution to the homogeneous problem, $u_h(x, t)$, can be expressed through Green's function:

$$u_h(x, t) = \int_{-\infty}^{\infty} G(x, t; y, 0) f(y) dy. \quad (2.12)$$

Duhamel's Principle states that $u_r(x, t) = \int_0^t w(x, t; s) dy$, where

$$(\partial_t - L)w(x, t; s) = 0, \quad t > s$$

$$w(x, s; s) = g(x, s)$$

The solution to the PDE is expressed again through the Green's function:

$$w(x, t; s) = \int_{-\infty}^{\infty} G(x, t; y, s) g(y, s) dy.$$

and therefore:

$$u_r(x, t) = \int_0^t \int_{-\infty}^{\infty} G(x, t; y, s) g(y, s) dy ds. \quad (2.13)$$

Thus, using 2.11, 2.12, and 2.13 it is possible to obtain:

$$u(x, t) = \int_{-\infty}^{\infty} G(x, t; y, 0) f(y) dy + \int_0^t \int_{-\infty}^{\infty} G(x, t; y, s) g(y, s) dy ds. \quad (2.14)$$

So, as it can be seen, the only difficulty in data fitting with a general PDE model is coming up with the Green's function which in some cases might be theoretically possible to derive, but in practice extremely difficult and sometimes even impossible to compute. Below, several examples of PDEs where the above approach can be used will be presented. In each example, an attempt is made to find the Green's function.

2.3 PDEs from the Fokker-Planck equation family

This equation (also known as the Kolmogorov Forward equation) describes the time evolution of the probability density function of position and velocity of a particle. The first use of the Fokker-Planck equation was the statistical description of Brownian motion of a particle in a fluid. Qualitatively, the dynamics of Brownian motion can most easily be understood by thinking of a small particle immersed in a fluid. Molecules of the fluid kick the particle around in an unpredictable way so that the position of the particle fluctuates. Much like Schroedinger's Equation in Quantum Mechanics, because of the fluctuations, the exact position cannot be determined and therefore the probability of finding the particle in a given region is needed. Solutions to the Fokker-Planck equation determine these probabilistic averages. Denote the probability density function of the particle velocity in the interval $(v, v + dv)$ by $W(v)$. $W(v)$ times the length of the interval dv is the probability of finding the particle in the interval $(v, v + dv)$.

The general form of the equation of motion for $W(v)$ in a one dimensional Brown-

ian Motion case is given by:

$$\frac{\partial W}{\partial t} = \left[-\frac{\partial}{\partial v} D^{[1]}(v) + \frac{\partial^2}{\partial v^2} D^{[2]}(v) \right] W, \quad (2.15)$$

where $D^{[1]}(v)$ is known as the Drift Coefficient, and $D^{[2]}(v)$ is the Diffusion Coefficient, the latter of which results from the presence of the stochastic force.

The following example is a particular case of the general equation, and even though it doesn't have constant coefficients it can be reduced to one.

Example 1:

$$\frac{\partial w}{\partial t} = D \frac{\partial^2 w}{\partial x^2} + \gamma x \frac{\partial w}{\partial x} + \gamma w, \quad w(x, 0) = w_0(x) \quad (2.16)$$

This is a particular case of an equation:

$$\frac{\partial w}{\partial t} = a \frac{\partial^2 w}{\partial x^2} + bx \frac{\partial w}{\partial x} + (cx + d)w \text{ with an obvious choice of coefficients.}$$

The following substitution of variables

$$w(x, t) = e^{\gamma t} u(z(x, t), \tau(x, t)), \quad (2.17)$$

where

$$\begin{aligned} \tau(x, t) &= \frac{D}{2\gamma} e^{2\gamma t}, \quad \tau \in [\tau_0, \infty), \quad \tau_0 = \frac{D}{2\gamma} \\ z(x, t) &= x e^{\gamma t} \end{aligned} \quad (2.18)$$

(Note: consider $\tau(x, t) = \frac{D}{2\gamma} e^{2\gamma t} - \frac{D}{2\gamma}$, and then τ_0 would be equal to 0)

reduces this equation to the diffusion equation with constant coefficients:

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial z^2}, \quad (2.19)$$

Proof.

$$\begin{aligned} \text{Then } \frac{\partial w}{\partial t} &= \frac{\partial u}{\partial t} e^{\gamma t} + \gamma u e^{\gamma t}, \\ \frac{\partial w}{\partial x} &= \frac{\partial u}{\partial x} e^{\gamma t} + u \frac{\partial (e^{\gamma t})}{\partial x} = \frac{\partial u}{\partial x} e^{\gamma t}, \\ \frac{\partial^2 w}{\partial x^2} &= \frac{\partial}{\partial x} \left(\frac{\partial w}{\partial x} \right) = \frac{\partial}{\partial x} \left(\frac{\partial u}{\partial x} e^{\gamma t} \right) = \frac{\partial^2 u}{\partial x^2} e^{\gamma t}. \end{aligned}$$

$$\begin{aligned} \text{So } \frac{\partial w}{\partial t} &= D \cdot \frac{\partial^2 w}{\partial x^2} + \gamma x \frac{\partial w}{\partial x} + \gamma w \iff \\ \iff \frac{\partial u}{\partial t} e^{\gamma t} + \gamma u e^{\gamma t} &= D \cdot \frac{\partial^2 u}{\partial x^2} e^{\gamma t} + \gamma x \frac{\partial u}{\partial x} e^{\gamma t} + \gamma u e^{\gamma t} \iff \\ \iff \frac{\partial u}{\partial t} e^{\gamma t} &= D \cdot \frac{\partial^2 u}{\partial x^2} e^{\gamma t} + \gamma x \frac{\partial u}{\partial x} e^{\gamma t} \iff \\ \iff \frac{\partial u}{\partial t} &= D \cdot \frac{\partial^2 u}{\partial x^2} + \gamma x \frac{\partial u}{\partial x}. \end{aligned}$$

$$\begin{aligned} \text{Now since } \frac{\partial z}{\partial t} &= x \gamma e^{\gamma t}, \quad \frac{\partial z}{\partial x} = e^{\gamma t}, \\ \frac{\partial \tau}{\partial t} &= D e^{2\gamma t}, \quad \frac{\partial \tau}{\partial x} = 0, \end{aligned}$$

then

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial u}{\partial z} \cdot \frac{\partial z}{\partial t} + \frac{\partial u}{\partial \tau} \cdot \frac{\partial \tau}{\partial t} = x \gamma e^{\gamma t} \frac{\partial u}{\partial z} + D e^{2\gamma t} \frac{\partial u}{\partial \tau}, \\ \frac{\partial u}{\partial x} &= \frac{\partial u}{\partial z} \cdot \frac{\partial z}{\partial x} + \frac{\partial u}{\partial \tau} \cdot \frac{\partial \tau}{\partial x} = e^{\gamma t} \frac{\partial u}{\partial z}, \\ \frac{\partial^2 u}{\partial x^2} &= \frac{\partial}{\partial x} \left(\frac{\partial u}{\partial x} \right) = \frac{\partial}{\partial x} \left(e^{\gamma t} \frac{\partial u}{\partial z} \right) = \frac{\partial}{\partial z} \left(e^{\gamma t} \frac{\partial u}{\partial z} \right) \cdot \frac{\partial z}{\partial x} + \frac{\partial}{\partial \tau} \left(e^{\gamma t} \frac{\partial u}{\partial z} \right) \cdot \frac{\partial \tau}{\partial x} = \\ &= e^{\gamma t} \frac{\partial}{\partial z} \left(e^{\gamma t} \frac{\partial u}{\partial z} \right) = e^{\gamma t} \left(\frac{\partial^2 u}{\partial z^2} e^{\gamma t} + \frac{\partial u}{\partial z} \cdot \frac{\partial (e^{\gamma t})}{\partial z} \right) = e^{2\gamma t} \frac{\partial^2 u}{\partial z^2} e^{\gamma t}. \end{aligned}$$

$$\begin{aligned} \text{Therefore, } \frac{\partial u}{\partial t} &= D \cdot \frac{\partial^2 u}{\partial x^2} + \gamma x \frac{\partial u}{\partial x} \iff \\ \iff x \gamma e^{\gamma t} \frac{\partial u}{\partial z} + D e^{2\gamma t} \frac{\partial u}{\partial \tau} &= D e^{2\gamma t} \frac{\partial^2 u}{\partial z^2} e^{\gamma t} + \gamma x e^{\gamma t} \frac{\partial u}{\partial z} \iff \end{aligned}$$

$$\Longleftrightarrow \frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial z^2}. \quad \square$$

In an effort to find the Green's function G for 2.16, consider a delta function as an initial condition:

$$G(x, 0; x', 0) = \delta(x - x').$$

Now, by evaluating 2.17 for $t = 0$,

$$\delta(x - x') = G(x, 0; x', 0) = u(z(x, 0), \tau(x, 0)) = u(z, \tau_0).$$

Expressing x through z, τ using 2.18:

$$x = z \sqrt{\frac{\tau}{\tau_0}},$$

thus

$$u(z, \tau_0) = \delta(z - x').$$

Therefore an initial-value problem for u can be obtained:

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial z^2}, \tau \geq \tau_0, \quad (2.20)$$

$$u(z, \tau_0) = \delta(z - x'). \quad (2.21)$$

Green's function for the diffusion equation described by 2.20, 2.21 is :

$$K(z, \tau; z', \tau') = \frac{\theta(\tau - \tau')}{\sqrt{4\pi D(\tau - \tau')}} \exp\left(-\frac{(z - z')^2}{4D(\tau - \tau')}\right), \quad \text{and } \theta \text{ is the Heaviside function}$$

The solution to 2.20, 2.21 can be described by:

$$u(z, \tau) = \int_{-\infty}^{\infty} K(z, \tau; y, \tau_0) \delta(z - x') dy. \quad (2.22)$$

Therefore:

$$u(z, \tau) = \int_{-\infty}^{\infty} K(z, \tau; y, \tau_0) \delta(z - x') dy = K(z, \tau; x', \tau_0), \quad \tau \geq \tau_0. \quad (2.23)$$

Going back to the original problem and its solution, recall 2.17 and 2.18 to obtain:

$$\tau - \tau_0 = \frac{De^{2\gamma t}(1 - e^{-2\gamma t})}{2\gamma}.$$

Note also that

$$\theta(\tau - \tau_0) = \theta(t).$$

Thus

$$\begin{aligned} G(x, t; x', 0) &= w(x, t) = e^{\gamma t} u(z(x, t), \tau(x, t)) = \\ &= e^{\gamma t} K(z(x, t), \tau(z, t); x', \tau_0) = \\ &= e^{\gamma t} \frac{\theta(\tau - \tau_0)}{\sqrt{4\pi D(\tau - \tau_0)}} \exp\left(-\frac{(z - x')^2}{4D(\tau - \tau_0)}\right) = \\ &= \frac{e^{\gamma t} \theta(t)}{\sqrt{4\pi \frac{De^{2\gamma t}(1 - e^{-2\gamma t})}{2\gamma}}} \exp\left(-\frac{(xe^{\gamma t} - x')^2}{4 \frac{De^{2\gamma t}(1 - e^{-2\gamma t})}{2\gamma}}\right) \end{aligned}$$

Or simplified

$$G(x, t; x', 0) = \frac{\sqrt{\gamma} \theta(t)}{\sqrt{2\pi D(1 - e^{-2\gamma t})}} \exp\left(-\frac{\gamma(x - e^{-\gamma t} x')^2}{2D(1 - e^{-2\gamma t})}\right) \quad (2.24)$$

Remark 5

It suffices to replace t with $t - t'$ in the formula 2.24 to reflect the shift of the initial condition ($\delta(x - x')$ in our case) from moment 0 to moment t' . A straight forward substitution of the time variable in 2.16 does not affect the equation and the derivation of 2.24. Therefore the final formula for Green's function takes on the form:

$$G(x, t; x', t') = \frac{\sqrt{\gamma} \theta(t - t')}{\sqrt{2\pi D(1 - e^{-2\gamma(t-t')})}} \exp\left(-\frac{\gamma(x - e^{-\gamma(t-t')} x')^2}{2D(1 - e^{-2\gamma(t-t')})}\right) \quad \square$$

Example 2:

The modified Fokker Plank equation has the form:

$$\frac{\partial w}{\partial t} = D \frac{\partial^2 w}{\partial x^2} - \frac{\gamma^2}{4D} x^2 w + \frac{\gamma}{2} w, \quad w(x, 0) = w_0(x) \quad (2.25)$$

It corresponds to another particular case of

$$\frac{\partial w}{\partial t} = a \frac{\partial^2 w}{\partial x^2} - bx^2 w - cw$$

with the obvious choice of coefficients.

A substitution of variables similar to the one in example 1

$$w(x, t) = e^{\gamma t + \frac{\gamma}{4D} x^2} u(z(x, t), \tau(x, t)), \quad (2.26)$$

where

$$\begin{aligned} \tau(x, t) &= \frac{D}{2\gamma} e^{2\gamma t}, \quad \tau \in [\tau_0, \infty), \quad \tau_0 = \frac{D}{2\gamma}, \\ z(x, t) &= x e^{\gamma t} \end{aligned} \quad (2.27)$$

Following the same line of reasoning as in example 1, one can arrive to the diffusion equation:

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial z^2}, \quad \tau \geq \tau_0, \quad (2.28)$$

$$u(z, \tau_0) = e^{\frac{\gamma}{4D} z^2} \delta(z - x'). \quad (2.29)$$

Thus, similar to example 1, the solution to 2.28, 2.29 is:

$$u(z, \tau) = e^{\frac{\gamma}{4D} z^2} K(z, \tau; x', \tau_0)$$

Following the reverse substitution:

$$\begin{aligned}
G(x, t; x', 0) &= (e^{\frac{\gamma}{4D}z(x,t)^2} K(z(x, t), \tau(x, t); x', \tau_0)) e^{\gamma t + \frac{\gamma}{4D}x^2} = \\
&= e^{\frac{\gamma}{4D}(x^2 - x'^2)} \frac{\sqrt{\gamma}\theta(t)}{\sqrt{2\pi D(1 - e^{-2\gamma t})}} \exp\left(-\frac{\gamma(x - e^{-\gamma t}x')^2}{2D(1 - e^{-2\gamma t})}\right)
\end{aligned} \tag{2.30}$$

Transition to the case of initial condition at time t' is identical to Example 1.

Thus:

$$G(x, t; x', t') = e^{\frac{\gamma}{4D}(x^2 - x'^2)} \frac{\sqrt{\gamma}\theta(t - t')}{\sqrt{2\pi D(1 - e^{-2\gamma(t-t')})}} \exp\left(-\frac{\gamma(x - e^{-\gamma(t-t')}x')^2}{2D(1 - e^{-2\gamma(t-t')})}\right) \quad \square$$

Example 3:

$$\frac{\partial w}{\partial t} = (\gamma x^2 + b) \frac{\partial^2 w}{\partial x^2} - \gamma x \frac{\partial w}{\partial x} + cw, \quad b > 0, \quad w(x, 0) = w_0(x) \tag{2.31}$$

which can be classified as a special case of a general Fokker Plank equation in one space variable.

The substitution of $x = x(z)$ defined by the one to one function

$$z(x) = \int_0^x \frac{d\xi}{\sqrt{\gamma\xi^2 + b}} \tag{2.32}$$

and

$$u(z, t) = w(x(z), t) \tag{2.33}$$

changes equation 2.31 into an equation with constant coefficients:

$$\frac{\partial u}{\partial \tau} = \frac{\partial^2 u}{\partial z^2} + cu \tag{2.34}$$

whose Green's function is

$$K_c(z, t; z', t') = e^{ct} K(z, t; z', t'), \quad (2.35)$$

where K is Green's function of the simple diffusion equation.

The initial condition for 2.31 in the form $w(x, 0) = \delta(x - x')$ implies the corresponding initial condition for 2.34 as follows:

$$u(z, 0) = w(x(z), 0) = \delta(x(z) - x') \quad (2.36)$$

Solution to initial value problem 2.34, 2.36 can be expressed via Green's function as follows:

$$u(z, t) = \int_{-\infty}^{\infty} \delta(x(\eta) - x') K_c(z, t; \eta, 0) d\eta \quad (2.37)$$

Using substitution

$$\eta = z(\xi), \quad d\eta = z' d\xi = \frac{d\xi}{\sqrt{\gamma\xi^2 + b}}, \quad x(\eta) = x(z(\xi)) = \xi$$

one can obtain:

$$u(z, t) = \int_{-\infty}^{\infty} \delta(\xi - x') K_c(z, t; z(\xi), 0) \frac{d\xi}{\sqrt{\gamma\xi^2 + b}} = \frac{1}{\sqrt{\gamma x'^2 + b}} K_c(z, t; z(x'), 0)$$

By performing reverse substitution $z = z(x)$, fundamental solution to 2.31 can be obtained in the form:

$$G(x, t; x', 0) = \frac{1}{\sqrt{\gamma x'^2 + b}} K_c(z(x), t; z(x'), 0) \quad (2.38)$$

Transition to the case of initial condition at time t' is identical to Example 1. Thus:

$$G(x, t; x', t') = \frac{1}{\sqrt{\gamma x'^2 + b}} K_c(z(x), t; z(x'), t') \quad \square$$

3 Regularized Compact Linear Operators

This section is dedicated to a general discussion of regularization theory.

3.1 Singular value decomposition for a generalized inverse

As for matrices, one can obtain a spectral representation theorem for a compact linear self adjoint operators in Hilbert spaces.

Theorem 1 (*First Hilbert Schmidt theorem*)

For every compact self adjoint operator $A : X \rightarrow Y$, $A \neq 0$, there exists a set of eigenvalues $\{\lambda_n\}_{n=1}^{\infty}$ such that

$$|\lambda_1| \geq \dots \geq |\lambda_i| \geq |\lambda_{i+1}| \geq \dots$$

(converging to 0 if the sequence is infinite)

and an orthonormal system $\{e_n\}_{n=1}^{\infty}$ of eigenvectors : $Ae_i = \lambda_i e_i$, such that

$$(i) \quad \forall x \in X, x = y + \sum_{n=1}^{\infty} \langle x, e_n \rangle e_n, \quad \text{where } y \in N(A)$$

$$(ii) \quad Ax = \sum_{n=1}^{\infty} \langle Ax, e_n \rangle e_n \quad \text{and } \forall z \in \overline{R(A)} \quad z = \sum_{n=1}^{\infty} \langle z, e_n \rangle e_n.$$

or in other words

$$Ax = \sum_{n=1}^{\infty} \lambda_n x_n \langle x, x_n \rangle \quad \forall x \in X \quad (3.1)$$

In the case when A is not self-adjoint (this applies to the matrices as well), consider operators

$$B = A^*A$$

and

$$C = AA^*.$$

Both B and C are compact, self-adjoint and even positive semi-definite, so that both admit a spectral representation with positive real eigenvalues of the form

$$Bx = \sum_{n=1}^{\infty} \sigma_n^2 u_n \langle x, u_n \rangle, \quad \forall x \in X \quad (3.2)$$

and

$$Cy = \sum_{n=1}^{\infty} \tilde{\sigma}_n^2 v_n \langle y, v_n \rangle, \quad \forall y \in Y \quad (3.3)$$

Note that $\tilde{\sigma}_n^2$, being an eigenvalue of C , and the corresponding eigenvector v_n of C satisfy the following series of equalities:

$$\tilde{\sigma}_n^2 A^* v_n = A^* C v_n = A^* A A^* v_n = B A^* v_n.$$

Hence $\tilde{\sigma}_n^2$ is an eigenvalue of operator B (with eigenvector $A^* v_n$). Vice versa, it follows that an eigenvalue σ_n^2 of B is also an eigenvalue of C (with eigenvector $A u_n$). Thus, there is a one-to-one relation between eigenvalues and eigenvectors of B and C , and one can assume without restriction of generality that

$$\sigma_n = \tilde{\sigma}_n, \quad v_n = \frac{A u_n}{\|A u_n\|}.$$

Above construction of eigensystems for B and C is the basis of singular value decomposition of compact linear operators. System (σ_n, u_n, v_n) is referred to as singular. Using this system one can obtain

$$Ax = \sum_{n=1}^{\infty} \sigma_n \langle x, u_n \rangle v_n, \quad \forall x \in X \quad (3.4)$$

which is called singular value decomposition of the operator A . For the adjoint operator one can derive an analogous relation of the form

$$A^* y = \sum_{n=1}^{\infty} \sigma_n \langle y, v_n \rangle u_n, \quad \forall y \in Y \quad (3.5)$$

Note that the sums on the right-hand side of 3.4 and 3.5 converge due to the square integrability of coefficients $\langle x, u_n \rangle$ (respectively $\langle x, v_n \rangle$), orthogonality of singular vectors, and boundedness of the singular values.

E.g. for 3.4 the finite sums

$$\left\| \sum_{n=1}^N \sigma_n \langle x, u_n \rangle v_n \right\|^2 \leq \sum_{n=1}^N \sigma_n^2 \langle x, u_n \rangle^2 \leq \sigma_1^2 \sum_{n=1}^N \langle x, u_n \rangle^2 \leq \sigma_1^2 \|x\|^2$$

are uniformly bounded with respect to N , which allows to pass to the limit $N \rightarrow \infty$.

Now a representation of the generalized inverse $A^\dagger$ in terms of the singular system can be derived. Recall that a least squares solution x satisfies the equation

$$A^*Ax = A^*y$$

and hence

$$\sum_{n=1}^{\infty} \sigma_n^2 \langle x, u_n \rangle u_n = A^*Ax = A^*y = \sum_{n=1}^{\infty} \sigma_n \langle y, v_n \rangle u_n,$$

and therefore, by comparison of the respective linearly independent components it is not difficult to see that

$$\langle x, u_n \rangle = \frac{1}{\sigma_n} \langle y, v_n \rangle. \quad (3.6)$$

As a direct consequence, the singular value decomposition of the generalized inverse $A^\dagger$ is obtained via

$$x^\dagger = A^\dagger y = \sum_{n=1}^{\infty} \frac{1}{\sigma_n} \langle y, v_n \rangle u_n \quad (3.7)$$

Again, the sum on the right-hand side of 3.7 must be checked for convergence. Unlike the case of A and A^* this is not always true for $A^\dagger$ and clearly reflects the unboundedness of the generalized inverse. The convergence criteria for the sum, namely

$$\|A^\dagger y\|^2 = \sum_{n=1}^{\infty} \frac{1}{\sigma_n^2} \langle y, v_n \rangle^2 < \infty \quad (3.8)$$

is called *Picard criterion*. This criterion can be interpreted as a smoothness condition on data y , since its coefficients with respect to the singular vectors v_n must decay faster than the singular values. The unboundedness of the generalized inverse for compact operators can again be seen from 3.8, since for normalized sequence v_n the norm is calculated by

$$\|A^\dagger v_n\| = \frac{1}{\sigma_n}.$$

By Hilbert Schmidt theorem, since σ_n^2 is an eigenvalue of B , $\sigma_n^2 \rightarrow 0$ as $n \rightarrow \infty$, and therefore

$$\|A^\dagger v_n\| \rightarrow \infty \text{ as } n \rightarrow \infty.$$

One can observe that errors in high frequencies, i.e., in the coefficients $\langle y, v_n \rangle$ corresponding to singular vectors with large n (and small σ_n) are amplified much stronger than those for low frequencies (large σ_n).

The way the high frequency errors are amplified depends on the operator A , more precisely on the decay speed of its singular values. The faster this decay happens, the more severe the Picard criterion 3.8 becomes.

3.2 Regularization methods

Let $A : X \rightarrow Y$.

A linear regularization method can be described by a family of continuous linear operators $R_\alpha : Y \rightarrow X$, for $\alpha \in I \subset (0, \alpha_0)$, where the index set I includes at least one sequence $\alpha_n \rightarrow 0$. The regularization operator should converge to the generalized inverse in some sense as $\alpha \rightarrow 0$. This means that, as $\alpha \rightarrow 0$, convergence $R_\alpha y \rightarrow A^\dagger y$ for $y \in D(A^\dagger)$ is necessary. If $y \in Y \setminus D(A^\dagger)$, it should be expected that $\|R_\alpha y\| \rightarrow \infty$ due to unboundedness of the generalized inverse.

Suppose now

$$\|y - y^\delta\| \leq \delta.$$

One has to choose the regularization parameter α in dependence of the noise level δ . Clearly, for $y \in D(A^\dagger)$, it is desired to obtain $R_\alpha y^\delta \rightarrow A^\dagger y$ as $\delta \rightarrow 0$. This desired convergence property leads to the following definition:

A family $\{R_\alpha\}_{\alpha \in I}$ of continuous linear operators is called regularization (or regularization operator) for $A^\dagger$, if for all $y \in D(A^\dagger)$ there exists a parameter choice rule $\alpha : \mathbb{R}^+ \rightarrow I$ such that

$$\limsup_{\delta \rightarrow 0} \{\|R_{\alpha(\delta)} y^\delta - A^\dagger y\| \mid y^\delta \in X, \|y - y^\delta\| \leq \delta\} = 0 \quad (3.9)$$

and

$$\limsup_{\delta \rightarrow 0} \{\alpha(\delta) \mid y^\delta \in X, \|y - y^\delta\| \leq \delta\} = 0 \quad (3.10)$$

Definition 2 For a specific $y \in D(A^\dagger)$, the pair (R_α, α) is called (*convergent*) *regularization method* of the equation $Ax = y$.

Theorem 3 *Let $A : X \rightarrow Y$ be a continuous linear operator and $R_\alpha : Y \rightarrow X$ be a family of continuous operators, $\alpha \in \mathbb{R}^+$. Then, the family $\{R_\alpha\}_{\alpha \in I}$ is a regularization of $A^\dagger$ if $R_\alpha y \rightarrow A^\dagger y$ pointwise on $D(A^\dagger)$ as $\alpha \rightarrow 0$.*

Proof.

Let $y \in D(A^\dagger)$ be arbitrary, but fixed. Due to pointwise convergence, there exists a monotone function $\sigma : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ such that for every $\varepsilon > 0$

$$\|R_{\sigma(\varepsilon)}y - A^\dagger y\| \leq \frac{\varepsilon}{2}$$

The operator $R_{\sigma(\varepsilon)}$ is continuous for fixed ε and hence, there exists $\rho(\varepsilon) \in I$ such that

$$\|R_{\sigma(\varepsilon)}z - R_{\sigma(\varepsilon)}y\| \leq \frac{\varepsilon}{2}, \quad \text{if } \|z - y\| \leq \rho(\varepsilon).$$

Without restriction of generality, one can assume that ρ is monotone increasing, continuous, and $\lim_{\varepsilon \rightarrow 0} \rho(\varepsilon) = 0$. Hence, there exists a well-defined inverse function ρ^{-1} on the range of ρ with the same properties. Now ρ^{-1} can be extended to a continuous, strictly monotone function on $\mathbb{R}^+$ and define the parameter choice rule as

$$\alpha : \mathbb{R}^+ \rightarrow \mathbb{R}^+, \quad \delta \longmapsto \sigma(\rho^{-1}(\delta))$$

By construction, for $\delta := \rho(\varepsilon)$ the following inequality is satisfied

$$\|R_{\sigma(\varepsilon)}y^\delta - A^\dagger y\| \leq \|R_{\sigma(\varepsilon)}y - A^\dagger y\| + \|R_{\sigma(\varepsilon)}z - R_{\sigma(\varepsilon)}y\| \leq \varepsilon$$

if $\|y - y^\delta\| \leq \delta$, because $\alpha(\delta) = \sigma(\varepsilon)$.

Hence, (R_α, α) is a convergent regularization method for $Ax = y$. □

This proves that any family of continuous operators that converges pointwise to the generalized inverse defines a regularization method. Vice versa, it can be concluded from 3.10 that

$$\lim_{\delta \rightarrow 0} R_{\alpha(\delta)}y = A^\dagger y \quad \forall y \in D(A^\dagger)$$

and thus, if α is continuous in δ , this implies

$$\lim_{\sigma \rightarrow 0} R_\sigma y = A^\dagger y$$

i.e., a convergent regularization method with continuous parameter choice rule implies pointwise convergence of the regularization operators.

Theorem 4 *Let $A : X \rightarrow Y$ be a continuous linear operator and $R_\alpha : Y \rightarrow X$ be a family of continuous operators with parameter choice rule $\alpha = \alpha(\delta)$. Then, (R_α, α) is a convergent regularization method if and only if*

$$\lim_{\delta \rightarrow 0} \alpha(\delta) = 0, \quad \lim_{\delta \rightarrow 0} \delta \|R_{\alpha(\delta)}\| = 0 \quad (3.11)$$

holds.

Proof.

($\Leftarrow$)

Suppose 3.11 holds. Then for all $y^\delta \in Y$ with $\|y - y^\delta\| \leq \delta$,

$$\begin{aligned} \|R_{\alpha(\delta)} y^\delta - A^\dagger y\| &\leq \|R_{\alpha(\delta)} y^\delta - R_{\alpha(\delta)} y\| + \|R_{\alpha(\delta)} y - A^\dagger y\| \leq \\ &\leq \|R_{\alpha(\delta)}\| \|y - y^\delta\| + \|R_{\alpha(\delta)} y - A^\dagger y\| \leq \delta \|R_{\alpha(\delta)}\| + \|R_{\alpha(\delta)} y - A^\dagger y\| \xrightarrow{\delta \rightarrow 0} 0 \end{aligned}$$

Therefore (R_α, α) is a convergent regularization method.

($\Rightarrow$)

Suppose (R_α, α) is a convergent regularization method. Assume that there exists a sequence $\delta_n \rightarrow 0$ such that $\delta_n \|R_{\alpha(\delta_n)}\| \geq C > 0$ for some constant C . Then there exists a sequence z_n , $\|z_n\| = 1$ such that $\delta_n \|R_{\alpha(\delta_n)} z_n\| \geq \frac{C}{2}$.

Then, for any $y \in D(A^\dagger)$ and $y_n = y + \delta_n z_n$ inequality $\|y_n - y\| \leq \delta_n$ is satisfied, but

$$R_{\alpha(\delta_n)} y_n - A^\dagger y = R_{\alpha(\delta_n)} y - A^\dagger y + \delta_n R_{\alpha(\delta_n)} z_n$$

does not converge, since the second term is unbounded. This contradicts the initial assumption that (R_α, α) is a convergent regularization method.

3.3 Construction of regularization methods

As it has been observed, instability of a problem arises if the spectrum of the operator $A : X \rightarrow Y$ is not bounded away from zero. Thus, it seems natural to construct regularizing approximations via modifying the smallest singular values. With singular value decomposition of the generalized inverse for compact linear operator it is easy to realize such modification of small singular values and construct regularization operators of the form

$$R_\alpha y = \sum_{n=1}^{\infty} g_\alpha(\sigma_n) \langle y, v_n \rangle u_n, \quad y \in Y \quad (3.12)$$

with some function $g_\alpha : \mathbb{R}^+ \rightarrow \mathbb{R}^+$ such that $g_\alpha(\sigma) \rightarrow \frac{1}{\sigma}$ for $\sigma > 0$ as $\alpha \rightarrow 0$. Such operator R_α is a regularization operator if

$$g_\alpha(\sigma) \leq C_\alpha < \infty, \quad \forall \sigma \in \mathbb{R}^+ \quad (3.13)$$

If 3.13 is satisfied, one can estimate

$$\|R_\alpha y\|^2 = \sum_{n=1}^{\infty} (g_\alpha(\sigma_n))^2 |\langle y, v_n \rangle|^2 \leq C_\alpha^2 \sum_{n=1}^{\infty} |\langle y, v_n \rangle|^2 \leq C_\alpha^2 \|y\|^2$$

Therefore

$$\|R_\alpha\| \leq C_\alpha. \quad (3.14)$$

From the pointwise convergence of g_α , the pointwise convergence of R_α to $A^\dagger$ can be immediately concluded. Moreover, condition 3.11 on the choice of the regulariza-

tion parameter can be replaced by

$$\lim_{\delta \rightarrow 0} \delta C_{\alpha(\delta)} = 0 \quad (3.15)$$

due to 3.14.

Hence, if 3.15 is satisfied, then (R_α, α) with R_α defined by 3.12 is a convergent regularization method. In order to make this construction more concrete, consider some particular examples:

Example 1 (Truncated Singular Value Decomposition).

The main idea of truncated singular value decomposition is to ignore all singular values below a certain threshold value, which can be identified with the regularization parameter α .

Thus, the function g_α is given by

$$g_\alpha(\sigma) = \begin{cases} \frac{1}{\sigma} & \text{if } \sigma \geq \alpha \\ 0 & \text{if } \sigma < \alpha \end{cases} \quad (3.16)$$

Obviously

$$C_\alpha = \frac{1}{\alpha}$$

and hence, truncated singular value decomposition is a convergent regularization method if

$$\frac{\delta}{\alpha} \rightarrow 0.$$

The representation of the regularized solution is given by

$$x_\alpha = R_\alpha y = \sum_{\sigma_n \geq \alpha} \frac{1}{\sigma_n} \langle y, v_n \rangle u_n, \quad y \in Y \quad (3.17)$$

which explains the name truncated singular value decomposition, since all terms in the sum corresponding to small singular values are truncated. Note that since 0 is the only accumulation point of the singular values of a compact operator, the sum in 3.17 is always finite for $\alpha > 0$. In particular, only a finite number of singular values and singular vectors has to be computed in order to realize this method. On the other hand, for α being small, the number of singular values that need to be computed can increase strongly.

Example 2 (Lavrentiev Regularization).

The main idea of Lavrentiev Regularization is to shift all singular values by α :

$$g_\alpha(\sigma) = \frac{1}{\sigma + \alpha} \quad (3.18)$$

so

$$x_\alpha = R_\alpha y = \sum_{n=1}^{\infty} \frac{1}{\sigma_n + \alpha} \langle y, v_n \rangle u_n, \quad y \in Y \quad (3.19)$$

In this case, the sum is really infinite and the full singular system is needed in order to compute the solution. However, if A is a positive semi-definite operator (and thus $\lambda_n = \sigma_n$, $u_n = v_n$), one obtains

$$(A + \alpha I)x_\alpha = \sum_{n=1}^{\infty} \frac{1}{\sigma_n + \alpha} \langle x_\alpha, u_n \rangle u_n = \sum_{n=1}^{\infty} \langle y, u_n \rangle u_n = y$$

Hence, in this case, regularized solution can also be obtained as the solution of

the linear equation without any knowledge of the singular system:

$$(A + \alpha I)x = y$$

Again, since $\frac{1}{\sigma + \alpha} \leq \frac{1}{\alpha}$,

$$C_\alpha = \frac{1}{\alpha},$$

and the condition to obtain a convergent regularization method is again

$$\frac{\delta}{\alpha} \rightarrow 0.$$

Example 3 (Tikhonov Regularization).

For Tikhonov Regularization, the function g_α is given by

$$g_\alpha(\sigma) = \frac{\sigma}{\sigma^2 + \alpha} \quad (3.20)$$

and the regularized solution is

$$x_\alpha = R_\alpha y = \sum_{n=1}^{\infty} \frac{\sigma_n}{\sigma_n^2 + \alpha} \langle y, v_n \rangle u_n \quad y \in Y \quad (3.21)$$

One can estimate $\sigma^2 + \alpha \geq 2\sigma\sqrt{\alpha}$ and hence,

$$g_\alpha(\sigma) \leq C_\alpha = \frac{1}{2\sqrt{\alpha}}$$

Thus, the condition for a convergent regularization method in this case becomes

$$\frac{\delta}{\sqrt{\alpha}} \rightarrow 0.$$

As in the case of Lavrentiev regularization, one can compute x_α defined by 3.21 without any knowledge of the singular system, but now for arbitrary linear operators A .

It is easy to see that

$$(A^*A + \alpha I)x_\alpha = A^*y$$

and hence, a well-posed linear system can be solved to obtain x_α . From this

representation one also observes that Tikhonov regularization is just Lavrentiev regularization applied to the Gaussian normal equation.

4 Simulations

Equation 2.16 was used for simulations of the presented techniques and theory.

4.1 Green's function simulation

Without entering the detail of generalized functions/distributions and their spaces, delta sequence will be defined as a "cap-shaped function" with property:

$$\lim_{\varepsilon \rightarrow 0^+} \int \omega_\varepsilon(x) \varphi(x) dx = \varphi(0) \quad (4.1)$$

which means that $\omega_\varepsilon(x) \rightarrow \delta(x)$ in the distributional sense. Different examples of $\omega_\varepsilon(x)$ are available in the literature. The following definition is used:

$$\omega_\varepsilon(x) = \begin{cases} C_\varepsilon \exp(-\frac{\varepsilon^2}{\varepsilon^2 - |x|^2}) & , \quad |x| \leq \varepsilon \\ 0 & , \quad |x| > \varepsilon \end{cases} \quad (4.2)$$

where constant C_ε is chosen so that

$$\int \omega_\varepsilon(x) dx = 1, \quad x \in \mathbb{R}.$$

Green's function for the equation model is found as solutions to the equation with $\delta(x - x')$ as initial condition, thus representing a response of the system at point (x, t) to instantaneous unit delta source(disturbance) applied at the point $(x', 0)$.

For simulation purposes, instead the delta function itself, delta-sequence $\omega_\varepsilon(x)$ shall be used as the initial condition for the pdepe MATLAB solver.

Note that the computation of the Green's function for $t \rightarrow 0$ is unstable. Really, at $t = 0$, $u_0(x) = u(x, 0) = \int_{\mathbb{R}} u_0(x') G(x, x'; 0) dx'$. Thus $G(x, x'; 0) = \delta(x - x')$. The Dirac's δ -function is a theoretical construct, and attempting to represent it as a matrix of finite numbers is impossible because it involves infinities. As a result, implementing any formulation of a Green's function is problematic for small values

of t .

The MATLAB files for this portion are DiracDelta.m, fp.m, fpGreensFunction.m and fpGreensFunctionGenerator.m. Green's Function matrix is precalculated and saved into a file before prior to the experiments.

4.2 Fake data generator

The following PDF function is used as an initial condition for the Fokker Planck equation:

$$u_0(x) = \frac{\frac{5}{8}e^{-\frac{1}{4}(x+7)^2} + \frac{2}{8}e^{-\frac{1}{4}(1-x)^2} + \frac{1}{8}e^{-\frac{1}{4}(6-x)^2}}{2\pi^{\frac{1}{2}}}.$$

A solution function $u(x, t)$ is being generated for 51 time slices for t values between 0 and 5 and pick a slice with time index 5 to play a role of collected data, y .

To simulate data collected from the real world situation, the original data gets "damaged" by introduction of random variations at each point as follows:

$$y_{damaged} = y + (0.1 * y) \cdot \text{rand}$$

In other words the original data vector y is perturbed by 10% modulated, normally distributed random (at each point) noise.

Figure 1 contains the plots of y and $y_{damaged}$ for one of the runs.

Figure 1. Fake Data Plot.

MATLAB programs used for this portion are DataGenerator.m, fp.m and Initial-Condition.m.

4.3 PDE solution time slice reconstruction

As it has been observed in previous sections, it is necessary to apply regularization techniques on the matrix so the solution-finding process would be more stable, but by stabilizing the process, one cannot distant ourselves from the "real" solution. In reality, there is no "real" solution to compare to; there is only the data vector $y=y_{damaged}$. Denote x_ε as a solution obtained via the new regularized problem, and x is the "real" solution, such that $Ax = y$. It is easy to verify that $\|x_\varepsilon\| \leq \|x\|$, and that $\|x_\varepsilon\|$ is decreasing as ε gets bigger.

Also note that the bigger the regularization parameter ε is, the more stable the solution-finding process is, but also $\|x_\varepsilon - x\|$ gets bigger. There is a direct correspondence between $\|x_\varepsilon - x\|$ and $\|Ax_\varepsilon - y\|$, and between stability of the problem and $\|x_\varepsilon\|$.

For each of the regularization techniques simulations the process is to vary ε , compute x_ε , and plot $\|x_\varepsilon\|$ versus $\|Ax_\varepsilon - y\|$. The curve that's obtained in most cases is referred to in the literature as the *L-Curve* because it is L-shaped [14]. Parameter ε is chosen such that $(\|Ax_\varepsilon - y\|, \|x_\varepsilon\|)$ corresponds to a point on the "vertex" of the curve.

4.3.1 Singular value perturbation (Tikhonov's regularization)

To approximate measured data vector $\mathbf{y}$ with a time slice of the solution to the PDE equation, equation $(\varepsilon\mathbf{I} + \mathbf{G}^T\mathbf{G})\mathbf{c} = \mathbf{G}^T\mathbf{y}$ is used, where ε is the regularization parameter, $\mathbf{G}$ is the matrix of Green's function values for some fixed moment in time, and $\mathbf{c}$ is an unknown vector.

Once vector $\mathbf{c}$ is found, vector $\mathbf{f} = \mathbf{Gc}$ gets computed and then used as an approximation function. Note, that it is unnecessary to use Singular Value decomposition

when inverting the matrix, because it is no longer ill-conditioned.

Figure 2 illustrates the effect of ε on $\|x_\varepsilon\|$ and $\|Ax_\varepsilon - y\|$. Each point represents different ε , which varies between 10^{-1} to 10^{-9} .

Figure 2. Singular Value Perturbation L-Curve

Notice that for $\varepsilon = 10^{-5}$ the point is on the "vertex" of the curve. This ε is chosen to calculate the data approximation. Figure 3 compares collected data with the obtained approximation.

Figure 3. Singular Value Perturbation Data Approximation

Figure 4 depicts the way regularization parameter ε affects x_ε . For each subfigure, a through f, parameter ε varies between 10^{-1} to 10^{-6} .

Notice how smooth x_ε gets for large ε : the more ε increases, the more x_ε takes the form of the initial condition for the PDE. Also, the behavior of x_ε gets more unpredictable as ε gets closer to 0. This shows the ill-posedness of the initial-condition reconstruction problem :

Figure 4. Effect of Regularization Parameter ϵ on Solution x_ϵ : a ($\epsilon = 10^{-1}$) through f ($\epsilon = 10^{-6}$).

4.3.2 Singular value truncation

To approximate measured data vector $\mathbf{y}$ with a time slice of the solution to the PDE equation, equation $\mathbf{G}\mathbf{c} = \mathbf{y}$ is used, where $\mathbf{G}$ is the matrix of Green's function values for some fixed moment in time, and $\mathbf{c}$ is an unknown vector. The process to find vector $\mathbf{c}$ is first to find a singular value decomposition for matrix $\mathbf{G}$, then eliminate singular values that are less than regularization parameter ε , and then find a pseudo-inverse for $\mathbf{G}$.

Once vector $\mathbf{c}$ is found, vector $\mathbf{f} = \mathbf{G}\mathbf{c}$ gets computed and then used as an approximation function.

Figure 5 illustrates the effect of ε on $\|x_\varepsilon\|$ and $\|Ax_\varepsilon - y\|$. Each point represents different ε , which varies between 10^{-1} to 10^{-9} . Because there is only a finite amount of singular values and for this particular matrix there are no singular values less than 10^{-5} , the last 5 points coincide.

Figure 5. Singular Value Truncation L-Curve

Notice that for $\epsilon = 10^{-2}$ the point is on the "vertex" of the curve. This ϵ is chosen to calculate the data approximation. Figure 6 compares collected data with the obtained approximation:

Figure 6. Singular Value Truncation Data Approximation

The MATLAB files for this portion are `fpSolutionReconstructor.m` (2 versions) and `MainSimulator.m`, which is a program that also generates a Green's function based pdf surface and compares it with the original surface produced by the MATLAB `pdepe` solver using mean squared error computation.

5 Conclusions

Through the problem of data fitting with a PDE model with unknown initial condition it has been observed that reconstruction of the initial condition for PDE model is an ill-posed (ill-conditioned in the discrete version), linear inverse problem. Several regularization techniques were applied to make reconstruction better posed (better conditioned in the discrete version). The only problem encountered in these applications was that regularization parameter was not easy to find. In the simulations section the L-Curve heuristic was used to find a good regularization parameter and an intuitive justification for this method was provided, but a formal justification for this heuristic is still sought after.

Possible extensions of this work might be an investigation of other parameter-finding techniques, regularization of non-linear problems, or other applications for regularization methods. Examples for such applications include inverse problems such as:

- Image processing: reconstruction of blurred and unfocused images
- Computerized tomography: reconstruction of spatially varying absorption coefficients within the human body from measurements of intensity decays of X-rays sent through the body
- Parameter identification: here, spatial and/or temporal parameters appearing in e.g., partial differential equations have to be determined from measurements of the solution, either in the whole domain, or on the boundary only.

References

- [1] Malyshev, I. and Pryvarnikova, A. Translating Differential Equation Models Into Kernel Methods for Data Analysis - Phase I, Department of Mathematics and Computer Science, San Jose State University, 2003.
- [2] Malyshev, I. Translating Differential Equation Models Into Kernel Methods for Data Analysis - Phase IV, Department of Mathematics and Computer Science, San Jose State University, 2005.
- [3] Vapnik, V. Statistical Learning Theory, Wiley, 1998.
- [4] Cucker, F. and Smale, S. On the Mathematical Foundations of Learning, Bulletin of the AMS, Vol. 39, Number 1, pp 1-49, 2001.
- [5] Renardy, M. An introduction to partial differential equations, Springer, 2004.
- [6] Burger, M. Inverse Problems, Industrial Mathematics Institute, Johannes Kepler University, 2005.
- [7] Bakushinsky, A. and Goncharsky, A. Ill-Posed Problems: Theory and Applications, Kluwer, 1989.
- [8] Beklemishev, D. Additional Chapters to Linear Algebra, Nauka, 1983.
- [9] Kolmogorov, A. and Fomin, S. Elements of Functions and Functional Analysis, Nauka, 1976.
- [10] Trenogin, V. Functional Analysis, Nauka, 1980.
- [11] Gelfand, I. and Shilov, G. Generalized functions, Vol. 1, Academic Press, 1964.
- [12] Risken, H. The Fokker-Planck Equation, Springer, 1984.

- [13] Schoenberg, I. Metric Spaces and positive definite functions, Transactions of the American Mathematical Society, Vol. 44, pp 522-536, 1938.
- [14] Hansen, P. The L-Curve and its use in the numerical treatment of inverse problems, Dept. of Math. Modeling, Technical University of Denmark, 2001.

Appendix: MATLAB programs

A DiracDelta.m

```
function delta = DiracDelta(x,x0)

%
%Gives an approximation function to the Dirac delta function
%
p = -5;
e = 10^p;
if abs(x-x0)<e
omega_e = exp(-(e^2/(e^2-(x-x0)^2)))/0.14715;
else
omega_e = 0;
end
delta = omega_e;
%
```

B InitialCondition.m

```
function u0 = InitialCondition(x)

%
% PDF that is used as an initial condition for Fokker Planck equation to generate
data.
u0 = ((5/8) * exp(-(-7-x)^2/4) + (2/8) * exp(-(1-x)^2/4) + (1/8) * exp(-(6-x)^2/4))
/ ((4*pi)^(1/2));
%
```

C **fp.m**

```
function [u] = fp(ic,x,t)
%
%outputs a solution function to Fokker Planck equation described in 2.16
%for an initial condition function passed in as the argument ic
%
m = 0;
global g
g = .5;
sol = pdepe(m,@fppde,ic,@fpbc,x,t,[]);
u = sol(:,1);

% subfunctions

function [c,f,s] = fppde(x,t,u,DuDx)
%
global g
c = 1;
f = DuDx;
s = g*x*DuDx+g*u;
%

function [pa,qa,pb,qb] = fpbc(xa,ua,xb,ub,t)
% bc - boundary cond.
pa = ua;
qa = 0;
pb = ub;
```

```
qb = 0;
```

```
%
```

D **fpGreensFunction.m**

```
function [W] = fpGreensFunction(x,t)
%
%Produces a matrix that represents the Green's function for equation 2.16
for J = 1:length(x)
    v = fp(@(y)DiracDelta(y,x(J)),x,t); %swfp_delta must be re-calibrated if
number of points is changed
    W(:,J,:) = v';
end
%
```

E **fpGreensFunctionGenerator.m**

```
%
%This has to get executed before you call MainSimultator.m
%
a = -40;
b = 40;
x = linspace(a,b,101);
t0 = 0;
tf = 5;
t = linspace(t0,tf,51);
%
W = fpGreensFunction(x,t);
save fpGreensFunction.mat W
%
```


F DataSetGenerator.m

```
function [x,yn,u] = DataSetGenerator(x,t,t_gen)

%
%This function generates a solution to 2.16 with initial condition function
%described in InitialCondition.m and introduces noise to each dataset point.
%
u = fp(@InitialCondition,x,t)';
% generation slice
y = u(:,t_gen);
% introduction of noise
yn = y + (0.1*y).*randn(101,1);
%yn = y;
a1 = -40;
b1 = 40;
a2 = -16; % NOTE: These values will be reassigned shortly to exactly
b2 = 16; % match values in x = [a1,b1]
a2index = 1 + (a2-a1)/(b1-a1)*(length(x) - 1);
b2index = 1 + (b2-a1)/(b1-a1)*(length(x) - 1);
a2 = x(a2index);
b2 = x(b2index);
display(['a1 = ', num2str(a1), ', b1 = ', num2str(b1),...
',a2 = ', num2str(a2), ', b2 = ', num2str(b2)]);
figure;
plot(x(a2index:b2index),y(a2index:b2index),'-r',x(a2index:b2index),yn(a2index:b2index),'-
.b');
```

```
axis manual
```

```
axis tight
```

```
h = legend('fp generated data','"damaged" data',2,'Location','BestOutside');
```

```
input__title = sprintf('input data set: for time %d',t_gen);
```

```
%
```

G `fpSolutionReconstructor.m` (Tikhonov's Regularization Version)

```
function [f,c] = fpSolutionReconstructor(x,y,epsilon,t_sim)
%
%This function takes in the data set generated by DataSetGenerator and
%produces an approximation using the Thikhonov's regularization technique.
%
I = speye(length(x));
load fpGreensFunction.mat
P = W(:,:,t_sim);
M = (epsilon*I + P'*P);
c = inv(M)*(P'*y); %regular matrix inverse
f = (P*c)';
for i = 1:length(x)
    if f(i)< 0
        ff(i) = 0;
    else
        ff(i) = f(i);
    end
end
f = ff;
int = trapz(x,f);
f = f/int;
%
```

H **fpSolutionReconstructor.m (Singular Value Truncation Version)**

```
function [f,c] = fpSolutionReconstructor(x,y,epsilon,t_sim)

%This function takes in the data set generated by DataSetGenerator and
%produces an approximation using the Singular Value Truncation technique

I = speye(length(x));

load fpGreensFunction.mat

P = W(:,t_sim);

c = pinv(P,epsilon)*y; %pseudo inverse with tolerance

f = (P*c)';

for i = 1:length(x)
    if f(i)< 0
        ff(i) = 0;
    else
        ff(i) = f(i);
    end
end

f = ff;

int = trapz(x,f);

f = f/int;

%
```

I MainSimulator.m

```
%  
  
%Before you call execute this function, you need to run fpGreensFunctionGener-  
ator  
  
%to precalculate Greens function for equation 2.16  
  
% Produces various outputs to showcase the techniques introduced in the thesis  
  
%  
  
% Pick a1 and b1 so as to specify the domain, x, over which the pde will be  
% solved. x1 is determined by  $x1 = [a1, b1]$ .  
  
a1 = -40;  
b1 = 40;  
x1 = linspace(a1,b1,101);  
  
t0 = 0;  
tf = 5;  
t = linspace(t0,tf,51);  
  
a2 = -16; % NOTE: These values will be reassigned shortly to exactly  
b2 = 16; % match values in  $x = [a1, b1]$   
a2index = 1 + (a2-a1)/(b1-a1)*(length(x1) - 1);  
b2index = 1 + (b2-a1)/(b1-a1)*(length(x1) - 1);  
a2 = x1(a2index);  
b2 = x1(b2index);  
display(['a1 = ', num2str(a1), ', b1 = ', num2str(b1), ...  
, 'a2 = ', num2str(a2), ', b2 = ', num2str(b2)]);  
  
p = 10;  
epsilon = 10^-p;
```

```

t_sim = 15;
t_gen = 5;
%Generate data
[x,y,u]=DataSetGenerator(x1,t,t_gen);
[g,c] = fpSolutionReconstructor(x,y,epsilon,t_sim);
load fpGreensFunction.mat;
for t_index = t_sim:length(t)
P = W(:,t_index);
f(:,t_index) = (P*c)';
% making a pdf out f; elimination of very small and negative (non-typical) values
for i = 1:length(x1)
if f(i,t_index)< 0
ff(i) = 0;
else
ff(i) = f(i,t_index);
end
end
sim_pdf(:,t_index) = ff;
%normalization
int = trapz(x1,sim_pdf(:,t_index));
sim_pdf(:,t_index) = sim_pdf(:,t_index)/int;
%end of normalization
% end of pdf
%MSE calculation
z1 = u(:,t_gen+t_index-t_sim);
sum1 = 0;

```

```

for i = 1:length(z1)
    sum1 = sum1 + (z1(i) - sim_pdf(i,t_index))*(z1(i) - sim_pdf(i,t_index));
end
mse1(t_index) = sum1 / length(z1);
disp(['t_index = ', num2str(t_index), ', mse1 = ', num2str(mse1(t_index))]);
end;
figure;
t_test = 30;
plot(x(a2index:b2index),sim_pdf((a2index:b2index),t_test),'-b',...
x(a2index:b2index),u((a2index:b2index),t_test + (t_gen-t_sim)),'-go');
axis manual
axis tight
h = legend('approximated data','exact data',3,'Location','BestOutside' );
test_title = sprintf('test : Comparison of solution data at time index %d and
approximation data at time index %d',t_test + (t_gen-t_sim),t_test);
title(test_title);
figure;
mesh(t,x,sim_pdf);
xlabel('x','FontSize',12);
ylabel('t','FontSize',12);
zlabel('sim_pdf','FontSize',12,'Rotation',0);
sim_title = sprintf('simulated solution surface: starts at time index %d',t_sim);
title(sim_title);
figure;
zzz = zeros(length(x),1);
for i = 1:t_gen-1

```

```

u(:,i) = zzz;
end;
mesh(t,x,u);
exact_title = sprintf('exact solution surface: starts at time index %d',t_gen);
title(exact_title);
P = W(:,t_sim);
pstart = 1;
pend = 9;
figure;
Lx = zeros(pend,1);
Ly = zeros(pend,1);
for p = pstart:pend
    epsilon = 10^-p;
    [f,c] = fpSolutionReconstructor(x,y,epsilon,t_sim);
    for i = 1:length(y)
        Lx(p) = Lx(p) + (y(i) - f(i))*(y(i) - f(i));
    end
    for i = 1:length(y)
        Ly(p) = Ly(p) + (c(i))*(c(i));
    end
end
Lx = sqrt(Lx)
Ly = sqrt(Ly)
plot(Lx,Ly,'-o'); %plot of the L-Curve
%
```