

serie  
**techné**

**Stable  
computations  
by discrete  
mollification**

carlos daniel carlos enrique  
**acosta mejía**



UNIVERSIDAD  
**NACIONAL**  
DE COLOMBIA

VICERRECTORÍA DE INVESTIGACIÓN  
**EDITORIAL**



**Stable  
Computations  
by Discrete  
Mollification**



# Stable Computations by Discrete Mollification

carlos daniel  
**acosta**

carlos enrique  
**mejía**



UNIVERSIDAD **NACIONAL** DE COLOMBIA  
VICERRECTORÍA DE INVESTIGACIÓN  
**EDITORIAL**

Bogotá, D. C., mayo de 2014

© Universidad Nacional de Colombia  
© Editorial Universidad Nacional de Colombia  
© Carlos Daniel Acosta Medina  
© Carlos Enrique Mejía Salazar

First edition, 2014  
ISBN 978-958-761-753-5 (paperback)  
ISBN 978-958-761-755-9 (POD)  
ISBN 978-958-761-754-2 (e-book)

### **Serie Techné design**

Marco Aurelio Cárdenas

### **Edition**

Editorial Universidad Nacional de Colombia  
direditorial@unal.edu.co  
www.editorial.unal.edu.co

Bogotá, D. C., Colombia, 2014

All rights reserved. No part of this book may be reproduced in any form by any electronic or mechanical means (including photocopying, recording, or information storage and retrieval) without permission in writing from the copyright owner.

Made and printed in Bogotá, D. C., Colombia

---

### **Catalogación en la publicación Universidad Nacional de Colombia**

Acosta Medina, Carlos Daniel, 1975.

Stable computations by discrete mollification / Carlos Daniel Acosta, Carlos Enrique Mejía. – Bogotá : Universidad Nacional de Colombia. Vicerrectoría de Investigación, 2014.

110 páginas : ilustraciones. – (Serie Techné)

Incluye referencias bibliográficas

ISBN : 978-958-761-753-5 (rústico) – ISBN : 978-958-761-755-9 (impresión bajo demanda) – ISBN : 978-958-761-754-2 (e-book)

1. Ecuaciones diferenciales hiperbólicas - Soluciones numéricas 2. Diferencias finitas 3. Leyes de conservación (matemáticas) I. Mejía Salazar, Carlos Enrique, 1957- I. Título II. Serie

CDD-21 515.353 / 2014

# Contents

List of Tables	11
List of Figures	12
Nomenclature	13
Acknowledgements	15
Foreword	17
Abstract	19
Chapter 1	
OVERVIEW	21
1.1 Discrete Mollification	22
1.2 Mollification Weights	22
1.3 Consistency, Stability and Convergence	24
1.4 Thinking About Nonlinearity	26
1.5 Boundary Conditions	28
1.5.1 Zero Padding Boundary Condition	29
1.5.2 Scaled Boundary Condition	29
1.5.3 Periodic Boundary Condition	30
1.5.4 Boundary Condition by Reflection	30
1.6 Parameter Selection	32
1.7 Nonlinear Operator	32
1.8 Looking Back	33
1.9 Concluding Remarks	37
Chapter 2	
CONVECTION DOMINATED DIFFUSIVE PROBLEMS	39
2.1 The Basic Scheme	39
2.2 Two Mollified Explicit Schemes	40
2.2.1 Stability	41
2.2.2 Consistency	43
2.3 Numerical Experiments	43
2.4 Concluding Remarks	44

Chapter 3	
CONSERVATION LAWS	47
3.1 Mollified Lax-Friedrichs Schemes (MLxF)	47
3.1.1 Consistency of Mollified Schemes	48
3.1.2 Linear Stability Analysis	49
3.2 Mollified Nessyahu-Tadmor (MNT) Schemes	50
3.2.1 Methods $MNT1$ and $MNT2$	50
3.2.2 Method $MNT3$	52
3.3 Numerical Experiments	52
3.4 Concluding Remarks	55
Chapter 4	
NONLINEAR AND DEGENERATE DIFFUSIVE PROBLEMS	57
4.1 Mollified Operator Splitting	57
4.1.1 The Diffusion Step	58
4.1.2 The Operator Splitting Method	59
4.1.3 The Nonlinear Convection-Diffusion Equation	60
4.1.4 Implementation	63
4.1.5 Numerical Experiments	63
4.1.6 Concluding Remarks	66
4.2 Strongly Degenerate Parabolic Equations	67
4.2.1 The Schemes	68
4.2.2 Example: Sedimentation	70
4.3 Concluding Remarks	72
Chapter 5	
SYSTEM IDENTIFICATION	73
5.1 A Diffusion Coefficient	73
5.1.1 Direct Problem	74
5.1.2 Inverse Problem	74
5.2 Numerical Algorithms	75
5.2.1 Coefficient $K(x, u) = 1 + a(x)u^2$	76
5.2.2 Coefficient $K(x, u) = a(x) + u^2$	76
5.3 Numerical Experiments	77
5.4 Strongly Degenerate Parabolic Equations	83
5.5 Parameter Identification	85
5.6 Discretization of the Inverse Problem	87
5.7 Numerical Experiments	88
5.8 Concluding Remarks	90

Chapter 6	
LITERATURE REVIEW	93
6.1 System Identification	93
6.1.1 Source Terms in a 1-D IHCP	93
6.1.2 Source Terms in a 2-D IHCP	94
6.1.3 Parameters in a Transport Equation	95
6.1.4 Parameters in a Drying System	96
6.2 Fractional Calculus	96
6.2.1 Caputo Fractional Derivatives	96
6.2.2 Source Term Identification in a TFDE	97
6.3 Multiscale Analysis	98
6.4 Concluding Remarks	99
Bibliography	101
Index	107



## List of Tables

Table 1.1	Mollification weights	23
Table 2.1	First mollified FTCS scheme applied to a heat equation	44
Table 3.1	Mollified NT methods applied to the advection equation	53
Table 3.2	Mollified NT methods applied to Burgers' equation	53
Table 4.1	A heat equation solved by discrete mollification	64
Table 4.2	Order test for solution of Burgers' equation with viscosity by the mollified operator splitting algorithm	65
Table 4.3	Order test for the mollified operator splitting algorithm	65
Table 4.4	Order test for the solution of Burgers' equation by the mollified operator splitting method	66
Table 4.5	Comparison of basic and mollified schemes	71
Table 5.1	Example 29: incidence of random noise and discretization parameters in identification	83
Table 5.2	Example 30: Incidence of random noise and discretization parameters in identification	84
Table 5.3	Example 31: incidence of random noise and discretization parameters in identification	84
Table 5.4	Initial guesses in example 32	88
Table 5.5	Numerical identification without mollification in example 32	89
Table 5.6	Numerical identification for $\eta = 3$ in example 32	89
Table 5.7	Numerical identification for $\eta = 5$ in example 32	89
Table 5.8	Numerical identification for $\eta = 8$ in example 32	90
Table 5.9	Identification results for example 33	90
Table 5.10	Identification results for example 34	91

## List of Figures

Figure 2.1	CFL bounds for first mollified FTCS scheme	42
Figure 2.2	CFL bounds for second mollified FTCS scheme	42
Figure 2.3	Numerical solution by the first mollified FTCS scheme	44
Figure 2.4	Numerical solution by the second mollified FTCS scheme	45
Figure 3.1	Mollified Lax-Friedrichs methods	50
Figure 3.2	Mollified NT schemes MNT1 and MNT2	51
Figure 3.3	MNT3 applied to Burgers' equation at $T = 0.5$ .	54
Figure 3.4	MNT3 applied to Burgers' equation at $T = 1.0$ .	54
Figure 3.5	MNT3 applied to the Buckley-Leverett problem.	55
Figure 4.1	Minimum number of points for discretization of $u_0(x)$	63
Figure 4.2	Burgers' equation with a source solved by the mollified operator splitting algorithm	65
Figure 4.3	Solution of a linear convection-diffusion equation by the mollified operator splitting algorithm	66
Figure 4.4	Numerical solution of Burgers' equation by the mollified operator splitting method	67
Figure 5.1	No noisy data and no mollification	78
Figure 5.2	Noisy data and no mollification	79
Figure 5.3	Reliability test, first part, $h = k = 2^{-6}$	79
Figure 5.4	Reliability test, second part, $h = 2^{-6}$ , $k = 2^{-8}$	80
Figure 5.5	First trial of 100 runs with parameters $h = k = 2^{-6}$	80
Figure 5.6	Second trial of 100 runs with parameters $h = 2^{-6}$ and $k = 2^{-7}$	81
Figure 5.7	Reliability test, $h = k = 2^{-6}$	82
Figure 5.8	Second trial of 100 runs, parameters $h = 2^{-6}$ and $k = 2^{-7}$	82
Figure 5.9	Error term vs $h$	83
Figure 6.1	Decomposition tree	98

## Nomenclature

$A_p$	Normalization constant		
$h$	$\Delta x$ Space grid size	$\eta$	Mollification parameter
$k$	$\Delta t$ Time step	$\delta$	Mollification parameter
$w_i$	Mollification weights	$\lambda$	Ratio $k/h$
$J_{\delta\eta}$	$J, J_\delta, J_\eta$ Mollification operator	$\mu$	Ratio $k/h^2$
$G^\varepsilon$	Noisy version of $G$	$\varepsilon$	Viscosity
$C$	Arbitrary constant	$\kappa_{p\delta}$	Mollification kernel
$X$	Discrete domain	$\alpha_L$	Nonlinear mollification
$D_+$	Forward difference operator	$\alpha_R$	Nonlinear mollification
$D_-$	Backward difference operator	$\alpha_C$	Nonlinear mollification
$D_0$	Central difference operator	$\beta^L$	Nonlinear mollification
$K$	$[a, b]$	$\beta^R$	Nonlinear mollification
$\tilde{K}$	$[a - p\delta, b + p\delta]$	$\beta^C$	Nonlinear mollification
$t_i$	Integral limits for weights	$\omega_L$	Nonlinear mollification
$T$	Mollification central matrix	$\omega_R$	Nonlinear mollification
$T_l$	Mollification left matrix	$\omega_C$	Nonlinear mollification
$T_r$	Mollification right matrix	$\varphi_j^n$	Conservative form
$y_{\delta\eta}$	$J_{\delta\eta}y$	$\rho_l$	Conservative form
$H$	Nonlinear mollification operator	$\Delta_+$	Difference operator
$x_i$	Space coordinate	$\Delta_-$	Difference operator
$t_i$	Time coordinate	$\Delta_0$	Difference operator
$u(x, t)$	$u_t, u_x, u_{xx}$ State variables		
$v_m^n$	Approximation of $u(x_m, t_n)$		
$z_m^n$	$Jv_m^n$ space mollification		
$f(u)$	Flux function		
$F_m^n$	$J(v^n)$ at $x_m$		
$u_0(x)$	Initial data		
$L$	Lipschitz constant		
$F^{EO}$	Engquist-Osher flux		
$A(u)$	Diffusion coefficient		



## Acknowledgements

We would like to express our gratitude to the following institutions:

Universidad Nacional de Colombia, Vicerrectoría de Investigación and Direcciones de Investigación Medellín y Manizales. Centro de Investigación en Matemáticas CIMAT, Guanajuato, México. Universidad de Concepción, Concepción, Chile. COLCIENCIAS: Project number 1118-489-25120.



## Foreword

Carlos Daniel Acosta Medina and Carlos Enrique Mejía Salazar have written a very well crafted and interesting overview of new applications of the classical method of mollification: *Stable computations by discrete mollification*.

Their purpose is multi folded, beginning with a review and extension of the traditional methodology (chapter I), followed by the main content introducing new topics of stabilization of explicit marching numerical schemes in the context of diffusive problems, conservation laws and nonlinear and degenerate diffusive problems (chapters II-IV), and ending with novel applications of regularization techniques to system identification and fractional calculus (chapters V-VI).

All is done utilizing the development approach of suitable mollification variants supplementing well known techniques by adding new ideas specifically directed at the needs of the present and future researchers.

The authors can be certain that readers will gain a better and broader perspective of the mollification approaches involved in direct, inverse and ill-posed problems as a result of their efforts.

This is an undertaking of great potential value and is done "smoothly". It is my pleasure to provide this foreword and to greatly recommend this essay.

### **Diego A. Murio**

Professor Emeritus of Mathematical Sciences  
Department of Mathematical Sciences  
University of Cincinnati  
Cincinnati, January, 2012  
USA



## Abstract

Discrete mollification is a versatile convolution-based filtering procedure for the regularization of ill-posed problems and the stabilization of explicit schemes for the numerical solution of partial differential equations. The technique was initially introduced by Paolo Manselli and Keith Miller in 1980 but it was Diego A. Murio who continued with the development of the method.

The first overview on the subject was presented by Murio in a 1993 book whose title is *The mollification method and the numerical solution of ill-posed problems*. It included only the regularizing effect of mollification. However, further advances on mollification motivated Murio to prepare an update of his book, which he presented in 2002 as a book chapter entitled *Mollification and space marching*. In this opportunity, Murio considered the stabilization of explicit schemes by mollification, which he had developed earlier, around 1996.

The stabilization of numerical schemes by discrete mollification has experienced a lot of activity. Therefore, we believed that a second update to Murio's seminal work was in order and this monograph is the result. Our idea is to introduce the method in detail and present a well selected set of applications, most of which are related to stabilization of explicit schemes, giving importance to the more recent advances.

The topics of inverse problems and explicit schemes for partial differential equations are important in a variety of applications like heat conduction, reservoir simulation, flow in porous media, sedimentation of flocculated suspensions and other phenomena modeled with hyperbolic conservation laws or convection diffusion equations.

This work is original in the following sense: The theoretical and numerical findings of chapters 1 through 5 were obtained by groups of people to which at least one of the authors belongs. Furthermore, the review of chapter 6 was prepared by the authors for this monograph.



# Chapter 1

## OVERVIEW

Mollification operators and mollifiers were introduced by Friedrichs (1944) as a tool to study extensions of differential operators. Since then, mollifiers play a role in analysis when dealing with regularization and approximation by smooth functions (cf., e.g., Gilbarg and Trudinger, 2001; Majda and Bertozzi, 2002) and became a successful strategy to stabilize ill-posed problems (cf., e.g., Schuster and Quinto, 2005; Maass, Pidcock, and Sebu, 2008; Louis and Maass, 1990; Hao, 1994; Hao, 1996; Hao and Reinhardt, 1997; Böckmann, Biele, and Neuber, 1998; Hegland and Anderssen, 1998).

This monograph deals with a discrete mollification method defined as a finite dimensional operator, actually a discrete convolution. Therefore, the subject is numerical linear algebra, or more precisely, the interaction of numerical linear algebra and partial differential equations. Discrete mollification is a filtering procedure suitable for the regularization of ill-posed problems and for the stabilization of explicit schemes for the numerical solution of PDEs.

The method was introduced by Paolo Manselli and Keith Miller in 1980 (Manselli and Miller, 1980) but they did not elaborate further on their findings. The development of the method was continued by Diego A. Murio, who did his Ph. D. dissertation under the supervision of K. Miller.

Since the 80's, Murio and his collaborators have been presenting applications of mollification in international meetings and a series of papers and books (cf., e.g., Murio, 1993; Murio, 2002; Mejía and Murio, 1993; Mejía, Murio, and Zhan, 2001; Mejía, 2007; Acosta and Mejía, 2009; Acosta and Mejía, 2010). It has been used in a variety of situations, for instance, inverse heat conduction problems (cf., e.g., Murio 1981; Mejía and Murio, 1996; Zhan, Coles, and Murio, 2001), Abel's integral equation (cf., e.g., Murio, Hinestroza, and Mejía, 1992), automatic numerical differentiation (cf., e.g., Murio, Mejía, and Zhan, 1998), coefficient identification (cf., e.g., Murio, Hinestroza, 1989; Mejía and Murio, 1995; Coles and Murio, 2001; Yi and Murio, 2004a), stabilization and acceleration of explicit numerical schemes for partial differential equations (cf., e.g., Montoya, Mejía, and Toro, 2000; Acosta and Mejía, 2008; Acosta, Bürger, and Mejía, 2012) and problems related to partial differential equations with fractional derivatives (cf., e.g., Murio, 2007; Murio and Mejía, 2008).

In this chapter we introduce the concept of discrete mollification and its main properties. We follow closely Acosta and Mejía, (2008) and Acosta and Mejía, (2009). Chapter 2 deals with stabilization of forward-time central-space explicit schemes for convection-diffusion equations while chapter 3 is devoted to mollified central schemes for hyperbolic conservation laws. In chapter 4, we introduce two more stabilizations by mollification. The first is a mollified operator splitting

method that features a thorough convergence analysis and the second is a monotone difference scheme stabilized by discrete mollification for the numerical solution of strongly degenerate parabolic equations. Recent advances in the field of system identification by mollification are considered in chapter 5. Some of the material is new and has just been submitted for publication. Finally, a literature review is included in chapter 6.

## 1.1 DISCRETE MOLLIFICATION

Let  $y = \{y_j\}_{j \in \mathbb{Z}}$  be a discrete function, which can, for example, be evaluations or cell averages of a real function  $y = y(x)$  at equidistant grid points on

$$X = \{x_j : x_j = x_0 + jh, \quad j \in \mathbb{Z}\}.$$

Its mollified version  $Jy$ , where  $J$  is the so-called *mollification operator*, is defined by

$$(Jy)_j := \sum_{i=-\eta}^{\eta} w_i y_{j-i}, \quad (1.1)$$

where  $\eta$  is the integer support parameter and the weights  $w_i$  satisfy

$$w_i = w_{-i}, \quad 0 \leq w_i \leq w_{i-1}, \quad i = 1, \dots, \eta; \quad \sum_{i=-\eta}^{\eta} w_i = 1, \quad \sum_{i=-\eta}^{\eta} i w_i = 0. \quad (1.2)$$

## 1.2 MOLLIFICATION WEIGHTS

The first way to obtain the mollification weights  $w_i$  works as follows: Let  $\delta > 0$ ,  $h > 0$  and  $p > 0$ . We choose  $\eta$  as the unique non-negative integer such that

$$(\eta - 1/2)h < p\delta \leq (\eta + 1/2)h, \quad (1.3)$$

which turns out to be

$$\eta = \left\lceil \frac{p\delta}{h} - \frac{1}{2} \right\rceil,$$

where  $\lceil \cdot \rceil$  means the ceiling function ( $\lceil x \rceil =$  the smallest integer not less than  $x$ .) The weights are obtained by numerical integration of the truncated Gaussian kernel

$$\kappa_{p\delta}(t) = \begin{cases} A_p \delta^{-1} \exp(-t^2/\delta^2), & |t| \leq p\delta \\ 0, & |t| > p\delta. \end{cases} \quad (1.4)$$

where the normalization constant

$$A_p = \left( \int_{-p}^p \exp(-s^2) ds \right)^{-1}$$

is chosen in such a way that  $\int_{\mathbb{R}} \kappa_{p\delta} = 1$ . Other important properties of the kernel, besides  $\int_{\mathbb{R}} \kappa_{p\delta} = 1$ , are:  $\kappa_{p\delta} \geq 0$ ,  $\kappa_{p\delta} \in C^\infty(-p\delta, p\delta)$  and  $\kappa_{p\delta}$  is zero outside  $[-p\delta, p\delta]$ . Based on this kernel, the mollification weights are computed by

$$w_i = \int_{t_i}^{t_{i+1}} \kappa_{p\delta}(-s) ds, \tag{1.5}$$

where

$$t_j = (j - 1/2)h, \quad j \in \mathbb{Z}. \tag{1.6}$$

We usually take  $p = 3$  and refer to  $\delta$  as the mollification parameter, whose role is to determine the shape of the kernel Gaussian bell.

The second way to obtain the weights is the following: Choose  $p > 0$ ,  $h > 0$  and a positive integer  $\eta$ . Then, satisfy relation (1.3) by choosing

$$\delta = \frac{(\eta + 1/2)h}{p}. \tag{1.7}$$

In this case we refer to  $\eta$  as the mollification parameter, whose role is to define the width of the mollification stencil. The mollification weights are obtained as before by (1.5).

Chosen in this way, the mollification weights  $w_j$ 's are independent of  $h$ , namely,

$$\begin{aligned} w_i &= \int_{t_i}^{t_{i+1}} \kappa_{\delta p}(-s) ds = A_p \delta^{-1} \int_{t_i}^{t_{i+1}} \exp\left(-\frac{s^2}{\delta^2}\right) ds \\ &= A_p \int_{t_i/\delta}^{t_{i+1}/\delta} \exp(-s^2) ds. \end{aligned}$$

Recall that

$$\frac{t_i}{\delta} = p \frac{(i - 1/2)h}{(\eta + 1/2)h} = p \frac{(i - 1/2)}{(\eta + 1/2)},$$

then the  $w_i$ 's depend only on  $p$  and  $\eta$ . Some weights are shown in table 1.1.

**Table 1.1.** Mollification weights

$\eta$	$w_0$	$w_1$	$w_2$	$w_3$	$w_4$	$w_5$
1	8.4272e-1	7.8640e-2				
2	6.0387e-1	1.9262e-1	5.4438e-3			
3	4.5556e-1	2.3772e-1	3.3291e-2	1.2099e-3		
4	3.6266e-1	2.4003e-1	6.9440e-2	8.7275e-3	4.7268e-4	
5	3.0028e-1	2.2625e-1	9.6723e-2	2.3430e-2	3.2095e-3	2.4798e-4

### 1.3 CONSISTENCY, STABILITY AND CONVERGENCE

**Theorem 1** (Convergence). *Let  $g$  be a sufficiently smooth real function. Let  $G$  be its discrete version defined on  $X$ . If  $G^\varepsilon$  is another discrete function defined on  $X$  such that*

$$|G^\varepsilon(x_j) - G(x_j)| \leq \varepsilon, \text{ for } x_j \in X,$$

*then, for each compact set  $K = [a, b]$  there exists a constant  $C = C(K)$  such that if  $x_j \in K$*

$$\begin{aligned} |J_{\delta\eta} G^\varepsilon(x_j) - J_{\delta\eta} G(x_j)| &\leq \varepsilon, \\ |J_{\delta\eta} G(x_j) - g(x_j)| &\leq Ch^2. \end{aligned} \quad (1.8)$$

*Additionally,*

$$\begin{aligned} |D_+ J_{\delta\eta} G(x_j) - g'(x_j)| &\leq Ch, \\ |D_0 J_{\delta\eta} G(x_j) - g'(x_j)| &\leq Ch^2, \\ |D_- D_+ J_{\delta\eta} G(x_j) - g''(x_j)| &\leq Ch^2, \end{aligned} \quad (1.9)$$

*where  $D_+$ ,  $D_-$  and  $D_0$  are the forward, backward and central finite difference operators respectively.*

*Proof.* Stability. It is established by the following sequence of an equality and two inequalities.

$$\begin{aligned} |J_{\delta\eta} G^\varepsilon(x_j) - J_{\delta\eta} G(x_j)| &= \left| \sum_{i=-\eta}^{\eta} w_i (G^\varepsilon(x_{j+i}) - G(x_{j+i})) \right| \\ &\leq \sum_{i=-\eta}^{\eta} w_i |G^\varepsilon(x_{j+i}) - G(x_{j+i})| \leq \varepsilon. \end{aligned}$$

If we set  $G^\varepsilon(x_j) = 0$  for every  $x_j$  and  $\varepsilon = \|G\|_\infty$  we obtain the useful inequality

$$\|J_{\delta\eta} G\|_\infty \leq \|G\|_\infty \quad (1.10)$$

Consistency. Let  $x_j \in K = [a, b]$  then, from Taylor's theorem, for each  $i = -\eta, \dots, \eta$  there exists  $\xi_i \in \tilde{K} = [a - p\delta, b + p\delta]$  such that

$$g(x_{j+i}) = g(x_j) + (ih)g'(x_j) + \frac{(ih)^2}{2}g''(\xi_i).$$

Then

$$\begin{aligned} J_{\delta\eta} G(x_j) &= \sum_{i=-\eta}^{\eta} w_i g(x_{j+i}) \\ &= \sum_{i=-\eta}^{\eta} w_i \left( g(x_j) + (ih)g'(x_j) + \frac{(ih)^2}{2}g''(\xi_i) \right) \\ &= g(x_j) + \frac{h^2}{2} \sum_{i=-\eta}^{\eta} i^2 w_i g''(\xi_i), \end{aligned} \quad (1.11)$$

where we have used (1.2),  $\sum_{i=-\eta}^{\eta} iw_i = 0$ . Thus,

$$\begin{aligned} |J_{\delta\eta}G(x_j) - g(x_j)| &= \frac{h^2}{2} \sum_{i=-\eta}^{\eta} i^2 w_i |g''(\xi_i)| \\ &\leq \frac{h^2}{2} \sum_{i=-\eta}^{\eta} i^2 w_i \|g''\|_{\infty, \tilde{K}}. \end{aligned}$$

Numerical differentiation. As above, if  $x_{j-1}, x_{j+1} \in K$ , there exist  $\xi_i, \zeta_i \in \tilde{K}$  such that

$$\begin{aligned} J_{\delta\eta}G(x_{j+1}) &= g(x_{j+1}) + \frac{h^2}{2} g''(x_{j+1}) \sum_{i=-\eta}^{\eta} i^2 w_i + \frac{h^4}{24} \sum_{i=-\eta}^{\eta} i^2 w_i g^{(4)}(\xi_i), \\ J_{\delta\eta}G(x_{j-1}) &= g(x_{j-1}) + \frac{h^2}{2} g''(x_{j-1}) \sum_{i=-\eta}^{\eta} i^2 w_i + \frac{h^4}{24} \sum_{i=-\eta}^{\eta} i^2 w_i g^{(4)}(\zeta_i). \end{aligned}$$

Now, for some  $\vartheta_j$  and  $v_j$  between  $x_{j-1}$  and  $x_{j+1}$  we have

$$\begin{aligned} \left| \frac{g(x_{j+1}) - g(x_{j-1})}{2h} - g'(x_j) \right| &= \frac{h^2}{12} |g'''(\vartheta_j)| \leq \frac{h^2}{12} \|g^{(3)}\|_{\infty, \tilde{K}}, \\ \left| \frac{g''(x_{j+1}) - g''(x_{j-1})}{2h} \right| &= |g'''(v_j)| \leq \|g^{(3)}\|_{\infty, \tilde{K}}, \\ \left| \frac{g^{(4)}(\xi_i) - g^{(4)}(\zeta_i)}{2h} \right| &\leq \frac{\|g^{(4)}\|_{\infty, \tilde{K}}}{h}. \end{aligned}$$

Thus

$$\frac{J_{\delta\eta}G(x_{j+1}) - J_{\delta\eta}G(x_{j-1})}{2h} = g'(x_j) + O(h^2).$$

The other bounds in (1.9) are obtained in a similar way. Notice that the smoothness requirement is  $g^{(4)}$  continuous and bounded in  $\mathbb{R}$ .  $\square$

Since numerical differentiation is an ill-posed problem, it is no surprise the kind of estimates that are obtained in the next corollary.

**Corollary 2.** *With the hypothesis of theorem 1, further estimates of numerical differentiation are:*

$$\left| D_+ J_{\delta\eta} G^\varepsilon(x_j) - g'(x_j) \right| \leq 2 \left( \frac{\varepsilon}{h} \right) + Ch$$

and

$$\left| D_0 J_{\delta\eta} G^\varepsilon(x_j) - g'(x_j) \right| \leq \frac{\varepsilon}{h} + Ch^2.$$

*Proof.* We assume mollification and differentiation are taken with respect to the space variable. The first is easily proved by the following inequalities:

$$\begin{aligned} |D_+ J_{\delta\eta} G^\varepsilon(x_j) - g'(x_j)| &\leq |D_+ J_{\delta\eta} G^\varepsilon(x_j) - D_+ J_{\delta\eta} G(x_j)| \\ &\quad + |D_+ J_{\delta\eta} G(x_j) - g'(x_j)| \end{aligned} \quad (1.12)$$

$$\begin{aligned} &\leq \frac{1}{h} |J_{\delta\eta} G^\varepsilon(x_{j+1}) - J_{\delta\eta} G(x_{j+1})| \\ &\quad + \frac{1}{h} |J_{\delta\eta} G^\varepsilon(x_j) - J_{\delta\eta} G(x_j)| \end{aligned} \quad (1.13)$$

$$\begin{aligned} &+ |D_+ J_{\delta\eta} G(x_j) - g'(x_j)| \quad (1.14) \\ &\leq 2 \left( \frac{\varepsilon}{h} \right) + Ch. \end{aligned}$$

The second result is analogous.  $\square$

Here mollification and the derivative are taken both with respect to the space variable. For the difference  $\left| D_{+,t} J_{\delta\eta} G^\varepsilon(x_j) - \frac{d}{dt} g(x_j) \right|$ , where  $D_{+,t}$  means the forward in time difference operator, there is an estimate in chapter 5.

According to section 1.2,  $\delta = O(h)$ . Thus, when mollification and the derivative are both taken with respect to the space variable, the estimates of the last corollary may be presented as  $O\left(\frac{\varepsilon}{\delta} + \delta\right)$  or  $O\left(\frac{\varepsilon}{\delta} + \delta^2\right)$ . They say that the regularization parameter should be linked to the level of noise in the data, which is the drawback of any ill-posed problem.

When dealing with numerical schemes for the approximation of a partial differential equation with dependent variable  $u$ , we assume the following notation: Matrix  $V_j^n$  represents the discrete approximation of  $u(x_j, t^n)$  and matrix  $Z_j^n$  denotes the  $\delta\eta$ -discrete mollification of  $V^n$  at  $x_j$ , that is,  $J_{\delta\eta} V^n$  evaluated at  $x_j$ . Sometimes we write  $J_\delta$  or  $J_\eta$  or even  $J$  rather than  $J_{\delta\eta}$ . These abbreviations do not cause any confusion whenever it is clear what the parameters are.

## 1.4 THINKING ABOUT NONLINEARITY

Stability and convergence of explicit schemes for nonlinear equations require additional tools, since the Lax Equivalence Theorem is no longer valid (cf., e.g., chapter 15 of LeVeque, 1990). The following properties are helpful in this regard.

**Lemma 3.** *The  $\eta$ -discrete mollification satisfies the conservative equation*

$$Z_j = V_j + (\varphi_{j+1} - \varphi_j),$$

where

$$\varphi_j = \sum_{i=1}^{\eta} w_i \sum_{k=1}^i (V_{j+k-1} - V_{j-k}).$$

Furthermore, if  $V_{j+i}$  has constant value  $u$  for  $i = -\eta, \dots, \eta$ , then

$$\varphi_{j+1} = \varphi_j = 0.$$

*Proof.* By using the consistency of the mollification and the symmetry of the weights  $w_i$ , we get

$$\begin{aligned} Z_j - V_j &= \sum_{i=-\eta}^{\eta} w_i V_{j+i} - \sum_{i=-\eta}^{\eta} w_i V_j \\ &= \sum_{i=1}^{\eta} w_i (V_{j+i} - V_j) + \sum_{i=1}^{\eta} w_i (V_{j-i} - V_j) \\ &= \sum_{i=1}^{\eta} w_i \sum_{k=1}^i (V_{j+k} - V_{j+k-1}) + \sum_{i=1}^{\eta} w_i \sum_{k=1}^i (V_{j-k} - V_{j-(k-1)}) \\ &= \sum_{i=1}^{\eta} w_i \sum_{k=1}^i (V_{j+k} - V_{j-k+1}) - \sum_{i=1}^{\eta} w_i \sum_{k=1}^i (V_{j+k-1} - V_{j-k}) \\ &= \varphi_{j+1} - \varphi_j. \end{aligned}$$

The additional remark is immediate from the definition of  $\varphi_j$ . □

Recall that for a discrete function  $V$  its total variation is defined by

$$TV(V) = \sum_{j=-\infty}^{\infty} |V_{j+1} - V_j|.$$

Other important property of discrete mollification is given in the following lemma.

**Lemma 4.** *Discrete mollification is a Total Variation Diminishing (TVD) method, that is,*

$$TV(Jy) \leq TV(y)$$

for all grid functions  $y$ .

*Proof.* Since  $w_i \geq 0$ ,

$$\begin{aligned} \sum_{j=-\infty}^{\infty} |[Jy]_{j+1} - [Jy]_j| &= \sum_{j=-\infty}^{\infty} \left| \sum_{i=-\eta}^{\eta} w_i (y_{j-i+1} - y_{j-i}) \right| \\ &\leq \sum_{j=-\infty}^{\infty} \sum_{i=-\eta}^{\eta} w_i |y_{j-i+1} - y_{j-i}|. \end{aligned}$$

Furthermore, since  $\sum_{i=-\eta}^{\eta} w_i = 1$ ,

$$\begin{aligned} \sum_{j=-\infty}^{\infty} \sum_{i=-\eta}^{\eta} w_i |y_{j-i+1} - y_{j-i}| &= \sum_{i=-\eta}^{\eta} w_i \left( \sum_{j=-\infty}^{\infty} |y_{j-i+1} - y_{j-i}| \right) \\ &= \sum_{i=-\eta}^{\eta} w_i TV(y) = TV(y). \end{aligned}$$

□

## 1.5 BOUNDARY CONDITIONS

Due to the nature of convolutions on bounded domains, it is imperative to know some additional information to compute the  $\delta\eta$ -discrete mollification of a data vector  $y \in \mathbb{R}^m$ , denoted  $y_{\delta\eta}$  or simply  $Jy$ . More precisely, if  $y = [y_1, y_2, \dots, y_m]$ , the convolution  $[Jy]_j = \sum_{i=-\eta}^{\eta} w_i y_{j-i} = \sum_{i=-\eta}^{\eta} w_{-i} y_{j+i}$  requires values  $y_s$  that are out of the range  $1 : m$ . A solution to this situation was proposed in Murio (1993), where two constant extensions of the data are found according to a certain optimization procedure. But there are other ways to extend the data and here we present the ones based on the techniques for image reconstruction and digital signal processing described in Ng, Chan, and Tang (1999); Section 7 of Calvetti, Reichel, and Zhang (1999) and Chapter 24 of Smith (1999). Let us assume, following definition 1.1, that

$$y_j = G(x_j),$$

where  $a = x_1 < x_2 < \dots < x_m = b$  and  $x_j - x_{j-1} = h > 0$  for  $j = 2, \dots, m$ . By (1.5),

$$[y_{\delta\eta}]_j = J_{\delta\eta} G(x_j) = \sum_{i=-\eta}^{\eta} w_i y_{j+i}, \text{ where } w_i = \int_{t_i}^{t_{i+1}} \kappa_{\delta\rho}(-s) ds$$

and the  $t_i$ 's are given by (1.6); that is,

$$t_i = (i - 1/2)h, \quad i \in \mathbb{Z}.$$

This yields

$$y_{\delta\eta} = T_l y_l + Ty + T_r y_r, \quad (1.15)$$

where

$$T_l = \begin{bmatrix} w_{-\eta} & \cdots & w_{-1} \\ & \ddots & \vdots \\ & & w_{-\eta} \\ 0 & & \end{bmatrix}_{m \times \eta}, \quad y_l = \begin{bmatrix} y_{-\eta+1} \\ y_{-\eta+2} \\ \vdots \\ y_{-1} \\ y_0 \end{bmatrix}_{\eta \times 1},$$

$$T = \begin{bmatrix} w_0 & \cdots & w_\eta & & 0 \\ \vdots & \ddots & & \ddots & \\ w_{-\eta} & & & & w_\eta \\ & \ddots & & \ddots & \vdots \\ 0 & & w_{-\eta} & \cdots & w_0 \end{bmatrix}_{m \times m}, \quad y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{m-1} \\ y_m \end{bmatrix}_{m \times 1}, \quad (1.16)$$

$$T_r = \begin{bmatrix} & & & 0 \\ w_\eta & & & \\ \vdots & \ddots & & \\ w_1 & \cdots & w_\eta & \end{bmatrix}_{m \times \eta}, \quad y_r = \begin{bmatrix} y_{m+1} \\ y_{m+2} \\ \vdots \\ y_{m+\eta-1} \\ y_{m+\eta} \end{bmatrix}_{\eta \times 1}.$$

The selection of vectors  $y_l$  and  $y_r$  is what we call *Boundary Conditions* after Ng, Chan, and Tang (1999).

### 1.5.1 ZERO PADDING BOUNDARY CONDITION

In this case,  $y_l$  and  $y_r$  are null and  $y_{\delta\eta}$  is computed from  $y$  by the Toeplitz matrix  $T$ . More precisely,

$$y_{\delta\eta} = Ty, \quad (1.17)$$

where  $T$  and  $y$  are given by (1.16). The boundary conditions are problem dependent. This one is useful when it is known that  $y$  decays rapidly to zero outside the domain  $[a, b]$ . Some other times it is not appropriate as can be seen in Calvetti, Reichel and Zhang (1999) and Smith (1999) for digital image processing.

### 1.5.2 SCALED BOUNDARY CONDITION

In this case,  $y_l$  and  $y_r$  are null and  $y_{\delta\eta}$  is computed from  $y$  by a scaled version  $T_{\delta\eta}$  of matrix  $T$  given by

$$T_{\delta\eta} = D^{-1}T,$$

where  $D = \text{diag}[d_1, d_2, \dots, d_m]$  and  $d_k$  is the sum of the elements in the  $k$ th row of  $T$ . Thus,

$$y_{\delta\eta} = T_{\delta\eta}y. \quad (1.18)$$

This is a no extension of data condition and as such, it has plenty of limitations.

### 1.5.3 PERIODIC BOUNDARY CONDITION

In this case,

$$y_l = \begin{bmatrix} y_{-\eta+1} \\ y_{-\eta+2} \\ \vdots \\ y_{-1} \\ y_0 \end{bmatrix} = \begin{bmatrix} y_{m-\eta+1} \\ y_{m-\eta+2} \\ \vdots \\ y_{m-1} \\ y_m \end{bmatrix}, \quad y_r = \begin{bmatrix} y_{m+1} \\ y_{m+2} \\ \vdots \\ y_{m+\eta-1} \\ y_{m+\eta} \end{bmatrix} = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_{\eta-1} \\ y_\eta \end{bmatrix}$$

and it is possible to compute discrete mollification in a straightforward way. The associated linear operator is the circulant matrix

$$C_{\delta\eta} = [0_{m \times (m-\eta)} | T_l] + T + [T_r | 0_{m \times (m-\eta)}]$$

and

$$y_{\delta\eta} = C_{\delta\eta} y.$$

Since  $C_{\delta\eta}$  is circulant, it can be diagonalized by the discrete Fourier matrix. This is an important advantage of using periodic boundary conditions. Again, if periodicity of data is not known, this is not the right boundary condition.

### 1.5.4 BOUNDARY CONDITION BY REFLECTION

A successful way to extend the data that works pretty well in cases where there is no indication on the behavior of the data outside the domain is by reflection of the information in the domain. This reflection can be even or odd.

#### Even Reflection

In the even case, the mollification operator is a Toeplitz plus Hankel matrix. For this matter we follow Section 3 of Ng, Chan, and Tang, 1999. It is computed as follows:

$$y_l = \begin{bmatrix} y_{-\eta+1} \\ y_{-\eta+2} \\ \vdots \\ y_{-1} \\ y_0 \end{bmatrix} = \begin{bmatrix} y_\eta \\ y_{\eta-1} \\ \vdots \\ y_2 \\ y_1 \end{bmatrix}, \quad y_r = \begin{bmatrix} y_{m+1} \\ y_{m+2} \\ \vdots \\ y_{m+\eta-1} \\ y_{m+\eta} \end{bmatrix} = \begin{bmatrix} y_m \\ y_{m-1} \\ \vdots \\ y_{m-\eta+2} \\ y_{m-\eta+1} \end{bmatrix}$$

Thus

$$y_{\delta\eta} = ([0_{m \times (m-\eta)} | T_l] R + T + [T_r | 0_{m \times (m-\eta)}]) y,$$

where  $R$  is the  $m \times m$  reverse matrix

$$R = \begin{bmatrix} & & & & 1 \\ & & & 1 & \\ & & & & \\ & & & & \\ 1 & & & & \end{bmatrix}_{m \times m}.$$

Moreover, matrix

$$R_{\delta\eta} = [0_{m \times (m-\eta)} | T_l] R + T + [T_r | 0_{m \times (m-\eta)}] R$$

is so that

$$R_{\delta\eta} = T + H,$$

where

$$H = \begin{bmatrix} w_{-1} & w_{-2} & \cdots & w_{-\eta} & 0 \\ w_{-2} & \cdots & w_{-\eta} & & \\ \vdots & & & 0 & w_{\eta} \\ w_{-\eta} & & 0 & & \vdots \\ 0 & w_{\eta} & \cdots & w_2 & w_1 \end{bmatrix}.$$

Notice that  $H$  is a Hankel matrix and the discrete mollification operator is a Toeplitz+Hankel matrix. Due to the symmetry of the weights, i.e.  $w_{-j} = w_j$ , matrix  $R_{\delta\eta}$  can be diagonalized by the discrete cosine transform (DCT) as explained in Ng, Chan, and Tang (1999).

### Odd Reflection

This is the other way to reflect data.

$$y_l = \begin{bmatrix} y_{-\eta+1} \\ y_{-\eta+2} \\ \vdots \\ y_{-1} \\ y_0 \end{bmatrix} = \begin{bmatrix} 2y_1 - y_{\eta} \\ 2y_1 - y_{\eta-1} \\ \vdots \\ 2y_1 - y_2 \\ y_1 \end{bmatrix}, y_r = \begin{bmatrix} y_{m+1} \\ y_{m+2} \\ \vdots \\ y_{m+\eta-1} \\ y_{m+\eta} \end{bmatrix} = \begin{bmatrix} 2y_m - y_m \\ 2y_m - y_{m-1} \\ \vdots \\ 2y_m - y_{m-\eta+2} \\ 2y_m - y_{m-\eta+1} \end{bmatrix}.$$

In this case there is no diagonalizable operator and computations are performed with operator (1.15). More precisely,

$$y_{\delta\eta} = (T + H\vartheta)y,$$

where  $\vartheta$  is

$$\vartheta = \begin{bmatrix} A & 0 & 0_{\eta \times \eta} \\ 0 & I_{m-2\eta} & 0 \\ 0_{\eta \times \eta} & 0 & B \end{bmatrix}$$

and matrices  $A$  and  $B$  are

$$A = \begin{bmatrix} 1 & & & & 0 \\ 2 & -1 & & & \\ & & \ddots & & \\ \vdots & & & & \\ 2 & 0 & \cdots & 0 & -1 \end{bmatrix}_{\eta \times \eta} \quad \text{and} \quad B = \begin{bmatrix} -1 & 0 & \cdots & 0 & 2 \\ & \ddots & & & \\ & & & & \vdots \\ & & & -1 & 2 \\ 0 & & & & 1 \end{bmatrix}_{\eta \times \eta}.$$

## 1.6 PARAMETER SELECTION

When working with mollification as a regularization procedure, given  $p > 0$  and  $h > 0$ , the mollification parameter  $\delta$  is automatically selected by GCV, Generalized Cross Validation (Golub, Heath, and Wahba, 1979), and the integer  $\eta$  is obtained from  $\delta$  as

$$\eta = \left\lceil \frac{p\delta}{h} - \frac{1}{2} \right\rceil,$$

which is consistent with equation (1.3).

So far we have established that, even when boundary conditions are assumed, discrete mollification is defined by a linear operator. Therefore GCV can be applied in all cases. However, for stabilization, it is convenient to keep  $\eta$  as the main mollification parameter; that is, we know in advance the width of the mollification stencil. Given  $\eta$ , the value of  $\delta$  is found from (1.3) as

$$\delta = \frac{(\eta + \frac{1}{2})h}{p}.$$

Unlike the previous case, we do not have automatic selection of the mollification parameter  $\eta$  and we have mollification weights that are independent of the space discretization parameter  $h$  (see section 1.2.)

## 1.7 NONLINEAR OPERATOR

When explicit schemes are the way to go, high resolution methods are the goal. But a linear, monotonicity preserving method is at most first order accurate (Godunov's Theorem LeVeque, 1990). Thus, some kind of nonlinear component is mandatory in order to achieve higher orders and, at the same time, avoid spurious oscillations. In this section we define, via a WENO-like strategy and a particular limiter, a nonlinear  $\eta$ -discrete mollification. WENO stands for Weighted Essentially Nonoscillatory.

Basic references on limiters and WENO strategies are Shu, 1998; LeVeque, 2002 and the references therein. The new nonlinear mollification operator, denoted  $H$ , together with the MC limiter, provide the nonlinear component of scheme MNT3 in chapter 3.

We describe this operator by its action on a discrete function  $y$  defined on the grid  $X$ . For each  $j$ , we define three grids following Zhang and Shu, 2007:

$G_L(j)$  consists on the values defined by a mirror reflection to the left. That is,

$$s_{j+i} = \begin{cases} x_{j+i} & \text{if } i = 0, 1, \dots, \eta \\ x_{j-i} & \text{if } i = -\eta, -\eta + 1, \dots, -1 \end{cases}$$

$G_R(j)$  is based on a mirror reflection to the right. That is,

$$r_{j+i} = \begin{cases} x_{j+i} & \text{if } i = -\eta, -\eta + 1, \dots, -1, 0 \\ x_{j-i} & \text{if } i = 1, \dots, \eta \end{cases}$$

$$G_C(j) = \{x_{j+i} / i = -\eta, -\eta + 1, \dots, 0, 1, \dots, \eta\}.$$

We denote  $J_{LY}(j)$  or simply  $J_{LY}$ , the  $\eta$ -discrete mollification of the discrete function  $y$  restricted to  $G_L(j)$ . Similarly for  $J_{RY}$  and  $J_{CY}$ . Operator  $H$  is defined by the linear combination

$$[Hy]_j = \omega_L J_{LY} + \omega_C J_{CY} + \omega_R J_{RY},$$

where  $\omega_L + \omega_C + \omega_R = 1$ . These coefficients are positive nonlinear functions of  $y$ . They are

$$\omega_L = \frac{\alpha_L}{\alpha_L + \alpha_C + \alpha_R}, \quad \omega_C = \frac{\alpha_C}{\alpha_L + \alpha_C + \alpha_R}, \quad \omega_R = \frac{\alpha_R}{\alpha_L + \alpha_C + \alpha_R},$$

$$\alpha_L = \frac{0.2}{(\varepsilon + \beta^L)^2}, \quad \alpha_C = \frac{0.6}{(\varepsilon + \beta^C)^2}, \quad \alpha_R = \frac{0.2}{(\varepsilon + \beta^R)^2}.$$

Here  $\varepsilon > 0$  is introduced to avoid the denominator to become 0. The  $\beta$ 's are the so-called "smoothness indicators" and depend in a nonlinear way on the discrete function  $y$ . Following the ideas in Zhang and Shu (2007), the  $\beta$ 's are defined as

$$\begin{aligned} \beta^C &= (y_{j+1} - y_{j-1})^2, \\ \beta^L &= (-3y_j + 4y_{j+1} - y_{j+2})^2, \\ \beta^R &= (3y_j - 4y_{j-1} + y_{j-2})^2. \end{aligned}$$

## 1.8 LOOKING BACK

The following definitions, taken from Gilbarg and Trudinger (2001), set the route for the mollification development.

A *mollifier* is a non-negative function  $\rho$  in  $C^\infty(\mathbb{R}^n)$  vanishing outside the unit ball  $B_1(0)$  and satisfying  $\int \rho \, dx = 1$ . By a linear change of coordinates, this definition can be established with  $B_r(0)$  instead of  $B_1(0)$  for any positive real number  $r$ .

Let  $\Omega \subset \mathbb{R}^n$  be a domain. For  $u \in L^1_{loc}(\Omega)$  ( $u$  is a locally integrable function) and  $h > 0$ , the *regularization* of  $u$ , denoted by  $u_h$ , is defined by the convolution

$$u_h(x) = h^{-n} \int_{\Omega} \rho\left(\frac{x-y}{h}\right) u(y) \, dy,$$

provided  $h < \text{dist}(x, \partial\Omega)$ .

Recall that this monograph deals with a discrete mollification method defined as a discrete convolution and that the origin goes back to Manselli and Miller (1980) and Murio (1981). In search of a regularization by convolution with a mollifier, they try several kernels but no discrete mollification was properly defined. By the middle of the 80's, the Gaussian kernel

$$\rho_\delta(t) = \frac{1}{\delta\pi^{1/2}} \exp\left(-\frac{t^2}{\delta^2}\right) \quad (1.19)$$

was the preferred mollification kernel. It satisfies all but one of the conditions to be a mollifier, since it does not vanish outside any ball around 0 although it falls to nearly zero outside  $[-3\delta, 3\delta]$  without having compact support. For years the mollification of a function  $f$  defined on  $I = [a, b]$  was defined by the convolution

$$J_\delta f(t) = (\rho_\delta \star f)(t) = \int_{-\infty}^{\infty} \rho_\delta(t-s) f(s) \, ds,$$

where  $\delta$ , the radius of mollification, is a positive real number (cf., e.g., Mejía and Murio, 1993; Murio, Hinestroza, and Mejía, 1992; Mejía and Murio, 1995; Murio, 1993). In particular, since the beginning, mollification has been a linear operator.

Throughout the years, the main concerns were: The need to deal with boundary conditions, since most of the time, we are interested in mollifying functions defined on compact sets, the lack of compact support for the kernel and the lack of an automatic procedure to select mollification parameters.

With respect to the first concern, the boundary conditions, the problem is: The convolution  $\rho_\delta \star f$  requires either the extension of  $f$  to a slightly bigger set  $I' \supset I$ ; for instance  $I' = [a - 3\delta, b + 3\delta]$  or the consideration of  $f$  restricted to a suitable compact set  $K \subset I$ , namely  $K = [a + 3\delta, b - 3\delta]$ . Both sets  $I'$  and  $K$  depend on  $\delta$ . In Mejía and Murio (1993), the following approach was introduced: Truncate  $\rho_\delta$  in such a way that the new kernel, denoted  $\rho_\delta$  again, is 0 outside  $[-3\delta, 3\delta]$  and without loss of generality, let  $I = [0, b]$ . Define an extension  $\tilde{f}$  of  $f$  to the domain  $[-3\delta, b]$ , by minimizing

$$\|J_\delta(\tilde{f}) - f\|_{L^2[0,3\delta]}.$$

In order to go ahead with an optimization procedure, the extension  $\tilde{f}$  has a particular form, depending on parameters; for instance,

$$\tilde{f}(t) = \begin{cases} f(t), & t \in [0, b] \\ c_1 t + c_2, & t \in [-3\delta, 0). \end{cases} \quad (1.20)$$

In Mejía and Murio (1993) there is a proof for the following proposition:

**Proposition 5.** *Suppose  $f \in L^2([0, b])$ . There exist unique constants  $c_1$  and  $c_2$  such that, if  $\tilde{f}$  is given by (1.20), then  $\|J_\delta(\tilde{f}) - f\|_{L^2[0, 3\delta]}$  is minimum.*

A similar result holds at the other end of the interval and a variety of extensions might be implemented, including simple constant extensions or higher degree polynomials. The extension procedure works fine but at a very high price: The mollification operator is not linear anymore.

The first formal definition of discrete mollification appears in Mejía and Murio, 1996. It requires a slightly different kernel, namely the truncated version of (1.19) given by

$$\kappa_{p\delta}(t) = \begin{cases} A_p \delta^{-1} \exp(-t^2/\delta^2), & |t| \leq p\delta \\ 0, & |t| > p\delta. \end{cases}$$

with the normalization constant given by

$$A_p = \left( \int_{-p}^p \exp(-s^2) ds \right)^{-1}.$$

Here  $p > 0$  is a real number and the recommended one is  $p = 3$ . This truncated kernel fails to comply with a different condition of a mollifier, namely, it is not a continuous function in  $\mathbb{R}$ . Although, in principle, it is easy to build a smooth kernel that coincides with  $\rho_\delta$  in  $D = [-3\delta, 3\delta]$  and decays rapidly to zero outside  $D$ , it turns out that  $\kappa_{p\delta}$  is simpler and provides the necessary compact support. So the second concern is taken care of by this truncated kernel.

The third concern was addressed some time around 1997 and one of the first papers implementing the GCV procedure is that by Murio, Mejía, and Zhan (1998). Due to lack of linearity of the mollification operator, section 4.2 of this paper deals with a linear approximation to mollification in order to apply the automatic selection of parameters by GCV.

Approximately ten years later, a new approach to boundary conditions that does not affect linearity of the mollification operator, was presented in Acosta and Mejía, 2008. The main advantage of the new approach is that no approximation of the mollification operator is required for the calls to GCV. This is the discrete mollification presented in section 1.1 and used throughout this monograph. However, there are several recent papers (cf., e.g., Murio and Mejía, 2008; Murio, 2006) that define mollification as follows:

**Definition 6.** (Infinite dimensional mollification operator) For the mollification kernel

$$\kappa_{p\delta}(t) = \begin{cases} A_p \delta^{-1} \exp(-t^2/\delta^2), & |t| \leq p\delta \\ 0, & |t| > p\delta. \end{cases} \quad (1.21)$$

where the normalization constant

$$A_p = \left( \int_{-p}^p \exp(-s^2) ds \right)^{-1}$$

is chosen in such a way that  $\int_{\mathbb{R}} \kappa_{p\delta} = 1$  and for  $g \in L^1([a, b])$ , the  $\delta$ -mollification of  $g$  at  $x \in [a + p\delta, b - p\delta]$  is defined by the convolution

$$\begin{aligned} J_{\delta}g(x) &= (\kappa_{p\delta} * g)(x) = \int_a^b \kappa_{p\delta}(x-s)g(s)ds \\ &= \int_{x-p\delta}^{x+p\delta} \kappa_{p\delta}(x-s)g(s)ds. \end{aligned}$$

The following proposition establishes that for smooth functions these two definitions coincide. To state the proposition we need the appropriate setting. Let  $g$  be a sufficiently smooth real function defined on  $I = [0, 1]$  and let  $\{x_i : x_i = ih, i = 0 : n, x_n = b\}$  be a uniform partition of  $I$ . Let  $G$  be a discrete version of  $g$ , i.e.,  $G_j = g(x_j)$  for  $j = 0 : n$ . For  $x_j \in [3\delta, 1 - 3\delta]$ , there are two definitions of mollification:

1. Discrete mollification:

$$JG(x_j) = \sum_{i=-\eta}^{\eta} w_{-i} G_{j+i} \text{ where } \eta \text{ and the } w_i\text{'s are defined above in section 1.2.}$$

2. Infinite dimensional mollification:

$$J_{\delta}g(x_j) = \int_{-\infty}^{\infty} \kappa(-s)g(x_j + s)ds. \text{ (For simplicity of notation, we drop the subscript } p\delta\text{)}$$

**Proposition 7.** For a sufficiently smooth function  $g$  and  $x_j \in [3\delta, 1 - 3\delta]$ , the two definitions of mollification coincide up to a term of order  $O(h)$ .

*Proof.* First we expand both definitions:

$$\begin{aligned} JG(x_j) &= \sum_{i=-\eta}^{\eta} w_{-i} G_{j+i} \\ &= \sum_{i=-\eta}^{\eta} \int_{(i-1/2)h}^{(i+1/2)h} \kappa(-s)g(x_j + ih)ds \end{aligned}$$

and

$$\begin{aligned}
 J_\delta g(x_j) &= \int_{-\infty}^{\infty} \kappa(-s) g(x_j + s) ds \\
 &= \int_{-3\delta}^{3\delta} \kappa(-s) g(x_j + s) ds \\
 &= \sum_{i=-\eta}^{\eta} \int_{(i-1/2)h}^{(i+1/2)h} \kappa(-s) g(x_j + s) ds.
 \end{aligned}$$

The last step is possible thanks to (1.3).

$$\begin{aligned}
 JG(x_j) - J_\delta g(x_j) &= \sum_{i=-\eta}^{\eta} \int_{(i-1/2)h}^{(i+1/2)h} \kappa(-s) [g(x_j + ih) - g(x_j + s)] ds \\
 &= \sum_{i=-\eta}^{\eta} \int_{(i-1/2)h}^{(i+1/2)h} \kappa(-s) g'(\lambda_{ij}) [ih - s] ds,
 \end{aligned}$$

where  $\lambda_{ij}$  is a number between  $x_j + ih$  and  $x_j + s$ . But it turns out that

$$\begin{aligned}
 &\left| \sum_{i=-\eta}^{\eta} \int_{(i-1/2)h}^{(i+1/2)h} \kappa(-s) g'(\lambda_{ij}) [ih - s] ds \right| \\
 &\leq \frac{1}{2} h \|g'\|_\infty.
 \end{aligned}$$

□

## 1.9 CONCLUDING REMARKS

We have presented the linear mollification operator as a discrete convolution and do not approximate any infinite dimensional convolutions (except the convolution of the previous section.) Notice that, due to the lack of smoothness when considered as a function defined on all  $\mathbb{R}$ , our truncated kernel

$$\kappa_{p\delta}(t) = \begin{cases} A_p \delta^{-1} \exp(-t^2/\delta^2), & |t| \leq p\delta \\ 0, & |t| > p\delta \end{cases}$$

do not satisfy one of the conditions of a mollifier as defined by Gilbarg and Trudinger (2001). Nevertheless, the discrete mollification operator is appropriate as a smoothing or stabilizing tool as can be seen in the following chapters in which we consider several applications.

Like the linear mollification operator, the nonlinear mollification operator is a discrete operator. In chapter 3 we search for mollified central schemes for hyperbolic conservation laws and nonlinear mollification is one of the main ingredients.

Finally, it is clear that there is room for improvements, variants and applications of mollification and we plan to keep participating in the process. Most of the time, there are several options regarding the solution of a problem. We tend to prefer options based on mollification but we do not claim neither that this is the only way nor that this is the best way. It is just a matter of preference.

## Chapter 2

# CONVECTION DOMINATED DIFFUSIVE PROBLEMS

Explicit schemes for partial differential equations are subject to restrictions on the time step. By stabilization of an explicit scheme we mean a procedure that speeds up computations by allowing less stringent restrictions and, therefore, greater time steps. This is a topic of interest to many authors (cf., e.g., Wubs, 1986; Alexiades *et al.* 1996; Eriksson *et al.* 2003) since it corresponds to a desirable improvement in computations. We believe the first published work on stabilization by mollification is Montoya, Mejía, and Toro (2000) but the idea of that paper was suggested to C. E. Mejía by D. A. Murio, who, incidentally, published his perception of stabilization by mollification in 2002 (section 5 of Murio, 2002).

This chapter follows Acosta and Mejía (2008) closely. It establishes several alternatives for the stabilization by mollification of the forward-time central-space (FTCS) scheme for a particular convection-diffusion equation.

### 2.1 THE BASIC SCHEME

Consider the convection-diffusion equation

$$u_t + au_x = bu_{xx} \quad (2.1)$$

and the forward-time central-space finite difference scheme (FTCS)

$$\frac{v_m^{n+1} - v_m^n}{k} + a \frac{v_{m+1}^n - v_{m-1}^n}{2h} = b \frac{v_{m+1}^n - 2v_m^n + v_{m-1}^n}{h^2}. \quad (2.2)$$

Here,  $v_m^n$  is the discrete approximation to the value of  $u$  at the nodal point  $(x_m, t_n)$ ,  $h$  and  $k$  are the uniform mesh sizes for the space and the time variables  $x$  and  $t$  respectively. Notice that (2.2) is equivalent to

$$v_m^{n+1} = v_m^n + b\mu(1 + \alpha)v_{m-1}^n - 2b\mu v_m^n + b\mu(1 - \alpha)v_{m+1}^n, \quad (2.3)$$

where

$$\mu = \frac{k}{h^2}, \quad \lambda = \frac{k}{h} \quad \text{and} \quad \alpha = \frac{ha}{2b} = \frac{a\lambda}{2b\mu}.$$

The stability bounds for this scheme are

$$b\mu \leq 1/2, \quad \alpha \leq 1.$$

The plan is to, somehow, add mollification to (2.2) and obtain a scheme with favorable stability bounds. With this in mind, we begin with a stability theorem.

**Theorem 8.** *Let*

$$v_m^{n+1} = \sum_{j=-\eta-1}^{\eta+1} W_j v_{m+j}^n,$$

*be a time-stepping numerical scheme for solving the convection-diffusion equation*

$$u_t + au_x = bu_{xx},$$

*on an equally spaced discrete domain*

$$x_1 < x_2 < \cdots < x_N.$$

*If the entries' modulus of the Fourier transform of the  $\mathbb{R}^N$  vector*

$$\left[ 0 \quad W_1 \quad \cdots \quad W_{\eta+1} \quad 0 \quad \cdots \quad 0 \quad W_{-\eta-1} \quad \cdots \quad W_{-1} \right] \quad (2.4)$$

*are all less than or equal to one, then the scheme is stable under periodic boundary conditions.*

*Proof.* Under periodic boundary conditions, the numerical solution at  $t_{n+1}$  can be computed from the solution at  $t_n$ , by multiplying the latter by the circulant matrix with first row (2.4). But, the eigenvalues of such a matrix are those entries in the Fourier transform of (2.4). Then the spectral radius of the iteration matrix is the infinity norm of the Fourier transform of (2.4). Since the spectral radius of the iteration matrix is less than or equal to one, the scheme is stable.  $\square$

Now we are ready to tackle the problem of modifying the FTCS scheme by discrete mollification.

## 2.2 TWO MOLLIFIED EXPLICIT SCHEMES

Let  $z_m^n = J_{\delta\eta} v^n(x_m)$  be the  $\delta\eta$ -mollification (in space) of  $v^n$  evaluated at  $x_m$ . We consider two mollified versions for (2.3), the first one

$$v_m^{n+1} = v_m^n + b\mu(1 + \alpha)z_{m-1}^n - 2b\mu z_m^n + b\mu(1 - \alpha)z_{m+1}^n, \quad (2.5)$$

results from replacing the data coming from spatial discrete differentiation for their mollified versions. The second mollified scheme is obtained from (2.3) by replacing the right hand side terms by their mollified versions. This yields

$$v_m^{n+1} = z_m^n + b\mu(1 + \alpha)z_{m-1}^n - 2b\mu z_m^n + b\mu(1 - \alpha)z_{m+1}^n. \quad (2.6)$$

The convergence of (2.5) and (2.6) follows from writing them as convolutions. Let  $w_j = 0$  for  $|j| > \eta$ , so that

$$\begin{aligned} z_m^n &= \sum_{j=-\eta}^{\eta} w_j v_{m+j}^n = \sum_{j=-\eta-1}^{\eta+1} w_j v_{m+j}^n, \\ z_{m-1}^n &= \sum_{j=-\eta}^{\eta} w_j v_{m-1+j}^n = \sum_{j=-\eta-1}^{\eta+1} w_{j+1} v_{m+j}^n, \\ z_{m+1}^n &= \sum_{j=-\eta}^{\eta} w_j v_{m+1+j}^n = \sum_{j=-\eta-1}^{\eta+1} w_{j-1} v_{m+j}^n. \end{aligned}$$

For the first scheme, this yields

$$v_m^{n+1} = v_m^n + b\mu \sum_{j=-\eta-1}^{\eta+1} \{(1+\alpha)w_{j+1} - 2w_j + (1-\alpha)w_{j-1}\} v_{m+j}^n,$$

or equivalently

$$v_m^{n+1} = \sum_{j=-\eta-1}^{\eta+1} [\delta_{0,j} + b\mu \{(1+\alpha)w_{j+1} - 2w_j + (1-\alpha)w_{j-1}\}] v_{m+j}^n, \quad (2.7)$$

where  $\delta_{0,j}$  represents the Kronecker's delta.

Similarly, for the second mollified scheme, we obtain

$$v_m^{n+1} = \sum_{j=-\eta-1}^{\eta+1} [w_j + b\mu \{(1+\alpha)w_{j+1} - 2w_j + (1-\alpha)w_{j-1}\}] v_{m+j}^n. \quad (2.8)$$

### 2.2.1 STABILITY

Theorem 8 can be applied to schemes (2.7) and (2.8). Figure 2.1 shows the maximum  $b\mu$  allowed for the scheme (2.7).

The CFL condition for the mollified scheme (2.8) is shown in figure 2.2. The main feature here is stabilization for  $\alpha$  close to 1, which is missing from the previous scheme.

From figures 2.1 and 2.2 some remarks are in order:

1. For values of  $\alpha$  very close to 1, only the second scheme provides improvements in stability.
2. For values of  $\alpha$  close or equal to 0, the stability bound is greater than  $\frac{1}{2}$  whenever mollification is used ( $\eta > 0$ ). The case  $\alpha = 0$  corresponds to the heat equation and was considered in Murio, 2002.
3. It is possible to improve the stability bound of the FTCS explicit scheme by discrete mollification.

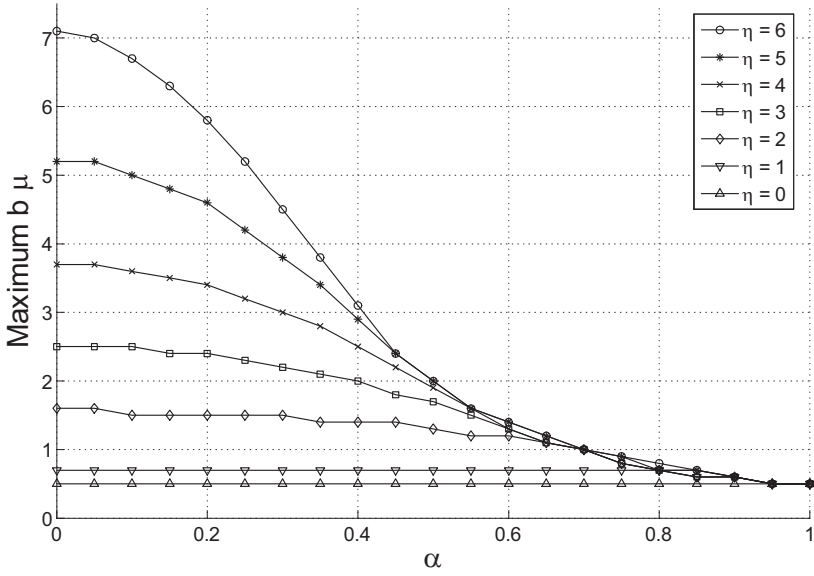


Figure 2.1. CFL bounds for first mollified FTCS scheme

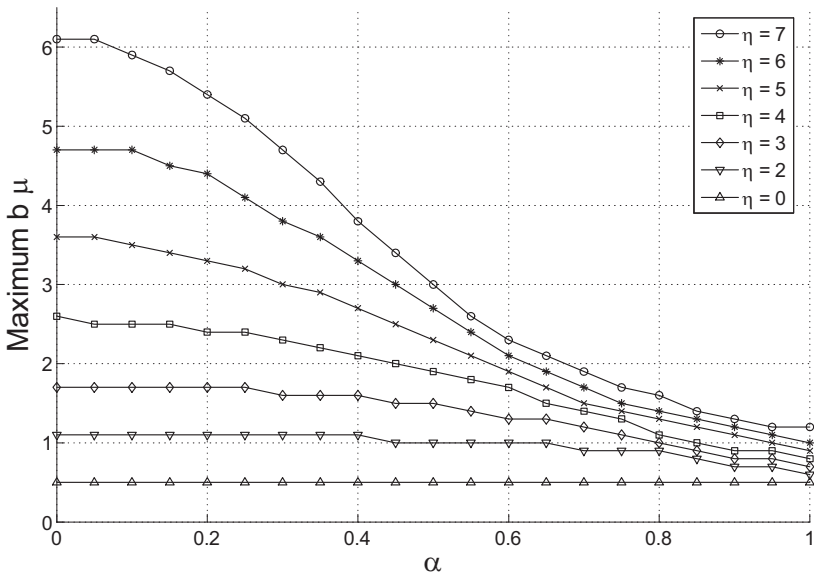


Figure 2.2. CFL bounds for second mollified FTCS scheme

### 2.2.2 CONSISTENCY

Schemes (2.5) and (2.6) are consistent with equation

$$u_t + au_x = bu_{xx}.$$

That this is so can be seen from

$$\frac{v_m^{n+1} - v_m^n}{k} + aD_0 z_m^n = bD_2 z_m^n, \quad (2.9)$$

$$\frac{v_m^{n+1} - z_m^n}{k} + aD_0 z_m^n = bD_2 z_m^n, \quad (2.10)$$

taking into consideration that

$$\begin{aligned} u_t - \frac{v_m^{n+1} - v_m^n}{k} &= \mathcal{O}(k) \\ u_t - \frac{v_m^{n+1} - z_m^n}{k} &= \mathcal{O}\left(k + \frac{h^2}{k}\right) \\ u_x(x_m, t_n) - D_0 z_m^n &= \mathcal{O}(h^2) \\ u_{xx}(x_m, t_n) - D_2 z_m^n &= \mathcal{O}(h^2). \end{aligned}$$

In both cases  $|v_m^n - u(x_m, t_n)| \rightarrow 0$  as  $k, h \rightarrow 0$  provided  $h^2/k \rightarrow 0$  as well.

## 2.3 NUMERICAL EXPERIMENTS

The first example illustrates the theory presented above and the other example is a preliminary encouraging experiment that shows the potential of discrete mollification as stabilizer of an explicit marching scheme for a nonlinear partial differential equation. We will expand on mollified schemes for nonlinear equations in the next chapters.

**Example 9.** *Mollified FTCS scheme (2.7) for the homogeneous heat equation*

$$\begin{aligned} u_t &= u_{xx}, \quad 0 < x < 1, \quad 0 < t \\ u(x, 0) &= \sin(\pi x), \quad 0 < x < 1, \\ u(0, t) &= u(1, t) = 0, \quad 0 < t \end{aligned}$$

The exact solution is known. For table 2.1 we use  $h = 1/64$ , the odd extension boundary condition and the maximum  $\mu$  allowed. The error norms are computed for all  $x$  and a set of  $t$ -values on the discrete grid.

**Example 10.** *Mollified FTCS scheme for the nonlinear viscous Burgers' equation*

$$u_t + \frac{1}{2}(u^2)_x = bu_{xx}$$

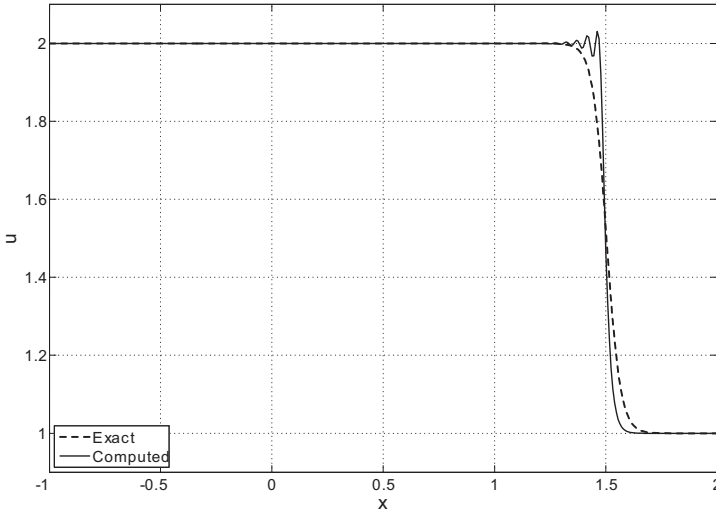
with

$$u(x, t) = a - c \tanh\left(\frac{c}{2b}(x - at)\right). \quad (2.11)$$

**Table 2.1.** First mollified FTCS scheme applied to a heat equation

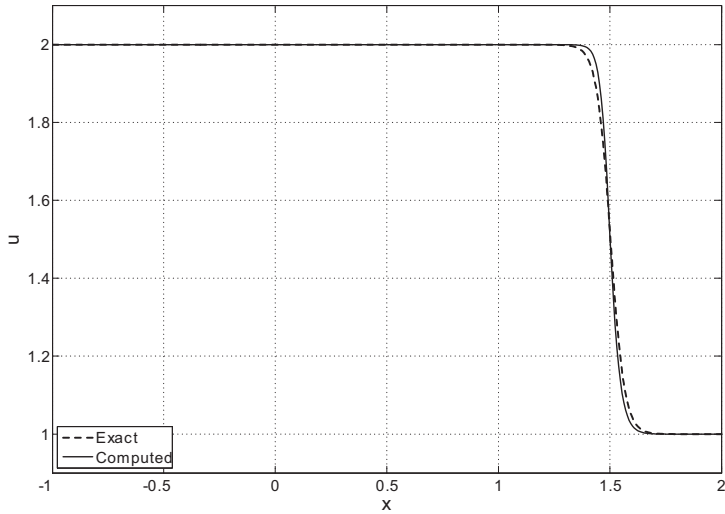
$\eta$	$b\mu$	<i>Inf Error</i>	$L_2$ Error	Time (sec)
0	0.5	$1.4563e-4$	$2.3612e-4$	1.532
1	0.6475	$2.5998e-3$	$7.3468e-4$	1.234
2	1.48	$6.4994e-3$	$1.8757e-3$	0.547
3	2.3125	$8.3427e-3$	$2.5916e-3$	0.344
4	3.4225	$8.6844e-3$	$3.1920e-3$	0.234
5	4.68	$8.7602e-3$	$3.8941e-3$	0.172
6	6.2125	$7.9251e-3$	$4.8629e-3$	0.125

Both schemes stabilize but the first one does not prevent spurious oscillations. Figures 2.3 and 2.4 illustrate this behavior at  $t = 1.0$  for  $a = 1.5, c = 0.5, h = 1/64, b = h(a+c), \eta = 4$  and  $b\mu = 1.9$ . This selection implies  $\alpha = 0.5$ . The reason for this behavior is that, unlike (2.7), scheme (2.8) is a TVD method.

**Figure 2.3.** Numerical solution by the first mollified FTCS scheme

## 2.4 CONCLUDING REMARKS

1. The first example illustrates how stabilization can reduce the computing time by relaxing the stability bound.



**Figure 2.4.** Numerical solution by the second mollified FTCS scheme

2. The second example shows that the stabilization procedure can be successfully implemented for nonlinear problems.
3. We visualize discrete mollification as an effective way to stabilize computations and, at the same time, control spurious oscillations.
4. A different approach provides a robust mollification strategy for the stabilization of explicit schemes and control of spurious oscillations for the solution of convection-diffusion equations. Such approach is presented in the next chapter.



## Chapter 3

# CONSERVATION LAWS

In this chapter, which follows Acosta and Mejía (2009) closely, we introduce the combination of discrete mollification with explicit central schemes for the approximation of one dimensional linear and nonlinear initial value problems based on hyperbolic conservation laws; that is,

$$\begin{aligned}\frac{\partial u}{\partial t} + \frac{\partial f(u)}{\partial x} &= 0, \\ u(x, 0) &= u_0(x).\end{aligned}\tag{3.1}$$

The mollified central methods presented here are based on the Lax-Friedrichs and the Nessyahu-Tadmor (NT) schemes (cf. LeVeque, 1990; Nessyahu and Tadmor, 1990). The new methods have less restrictive stability conditions and can be implemented in such a way that spurious oscillations are not present in numerical solutions.

Based on the Lax-Friedrichs (LxF) method, we introduce two mollified schemes:

1. MLxF1: linear, same monotonicity region of LxF, less restrictive CFL bound.
2. MLxF2: linear, less restrictive CFL bound and monotonicity region.

The new methods based on NT schemes satisfy the following conditions:

1. MNT1: linear and second order. Therefore, it does not control spurious oscillations.
2. MNT2: linear, consistent and monotone. Then it is at most first order accurate, nonoscillatory and convergent to the entropy solution.
3. MNT3: nonlinear and second order. Based on the encouraging numerical examples, it might be nonoscillatory.

### 3.1 MOLLIFIED LAX-FRIEDRICHS SCHEMES (MLxF)

The first order LxF method is

$$v_{j+1/2}^{n+1} = \frac{1}{2} (v_j^n + v_{j+1}^n) - \lambda (f(v_{j+1}^n) - f(v_j^n)),\tag{3.2}$$

where  $k$  is the time discretization parameter ( $t^n = nk$ ),  $h$  is the space discretization parameter,  $\lambda = k/h$  is constant and  $v_j^n$  is the approximation of the value of  $u$

at the point  $(x_j, t^n)$ . Method (3.2) is consistent, monotone (therefore TVD) and for  $\lambda$  constant, is convergent to the entropy solution as  $k \rightarrow 0$ . Moreover, it is conditionally stable with stability bound (CFL condition) given by

$$\lambda \max_u |f'(u)| \leq \frac{1}{2}.$$

The two mollified versions of (3.2) are

$$V_{j+1/2}^{n+1} = \frac{1}{2} (V_j^n + Z_{j+1}^n) - \lambda (F_{j+1}^n - F_j^n) \quad (3.3)$$

and

$$V_{j+1/2}^{n+1} = \frac{1}{2} (\tilde{Z}_j^n + Z_{j+1}^n) - \lambda (F_{j+1}^n - F_j^n), \quad (3.4)$$

where matrix  $V_j^n$  represents the discrete approximation of  $u(x_j, t^n)$ , matrix  $Z_j^n$  denotes the  $\eta$ -discrete mollification of  $V^n$  at  $x_j$ ,  $F_j^n$  is the  $\eta$ -discrete mollification of  $f(V^n)$  at  $x_j$  and  $\tilde{Z}_j^n$  indicates the  $(\eta + 1)$ -discrete mollification of  $V^n$  at  $x_j$ . We call them MLxF1 and MLxF2 respectively.

### 3.1.1 CONSISTENCY OF MOLLIFIED SCHEMES

The following theorem establishes consistency for several mollified central schemes.

**Theorem 11.** *Consider the conservation law (3.1) and a consistent staggered scheme*

$$v_{j+1/2}^{n+1} = \frac{1}{2} (v_j^n + v_{j+1}^n) - \lambda (g_{j+1}^n - g_j^n). \quad (3.5)$$

Moreover, consider the following mollified versions of (3.5),

$$\begin{aligned} V_{j+1/2}^{n+1} &= \frac{1}{2} (V_j^n + V_{j+1}^n) - \lambda (F_{j+1}^n - F_j^n), \\ V_{j+1/2}^{n+1} &= \frac{1}{2} (V_j^n + Z_{j+1}^n) - \lambda (F_{j+1}^n - F_j^n), \\ V_{j+1/2}^{n+1} &= \frac{1}{2} (\tilde{Z}_j^n + Z_{j+1}^n) - \lambda (F_{j+1}^n - F_j^n), \end{aligned}$$

where  $F_j$  denotes the  $\eta$ -discrete mollification of the flux  $g$  at  $V_j$ ,  $Z_j$  denotes the  $\eta$ -discrete mollification of  $V^n$  at  $x_j$  and  $\tilde{Z}_j^n$  denotes the  $(\eta + 1)$ -discrete mollification of  $V^n$  at  $x_j$ . Then, the three mollified schemes are consistent with (3.1).

*Proof.* The proof can be found in Acosta and Mejia (2009). It is based primarily on the conservative form of the mollification operator (lemma 3).  $\square$

**Corollary 12.** *The mollified methods MLxF1 (3.3) and MLxF2 (3.4) are consistent.*

### 3.1.2 LINEAR STABILITY ANALYSIS

We consider the linear flux  $f(u) = au$  with  $a$  a positive fixed constant. Then,  $F_j^n = aZ_j^n$  and scheme (3.3) becomes

$$\begin{aligned} V_{j+1/2}^{n+1} &= \frac{1}{2} (V_j^n + Z_{j+1}^n) - \lambda (F_{j+1}^n - F_j^n) \\ &= \frac{1}{2} V_j^n + a\lambda Z_j^n + \left( \frac{1}{2} - a\lambda \right) Z_{j+1}^n. \end{aligned}$$

Let  $w_i = 0$  for  $|i| > \eta$  and denote  $\delta_{0,i}$  the Kronecker's delta. Thus,

$$\begin{aligned} V_{j+1/2}^{n+1} &= \sum_{i=-\eta-1}^{\eta+1} \rho_{-i} V_{j+i}^n, \text{ where} \quad (3.6) \\ \rho_{-i} &= \frac{1}{2} \delta_{0,i} + a\lambda w_{-i} + \left( \frac{1}{2} - a\lambda \right) w_{-i+1}. \end{aligned}$$

Assume data vector  $V^n$  is periodic with period  $M$ . We only need to compute  $V_{j+1/2}^{n+1}$  for  $j = 1, 2, \dots, M$ . Additionally, thanks to (3.6), this can be done through the action of a circulant matrix  $C$  on the values  $V^n$  for the same range of  $j$ -values. Such a matrix  $C$  has as first row the  $\mathbb{R}^M$  vector

$$\left[ \rho_0 \quad \rho_{-1} \quad \dots \quad \rho_{-\eta-1} \quad 0_{1 \times M-2\eta-3} \quad \rho_1 \quad \dots \quad \rho_{\eta+1} \right]. \quad (3.7)$$

Thus, scheme (3.6) is stable if the spectral radius of matrix  $C$  is bounded by 1. But, given values of  $\lambda$  and  $a$  we can find this radius by using the Discrete Fourier Transform of (3.7). Furthermore, if every entry in (3.7) is non-negative, the method is monotone.

The CFL bounds for MLxF1 (3.6) are shown in figure 3.1, which suggests that the addition of mollification to the LxF scheme may accelerate computations because less restrictive CFL bounds are allowed. A monotonicity test was also performed. It turns out that method MLxF1 is monotone whenever  $a\lambda \leq 1/2$ . So, in this case the monotonicity region is the same as the one for the original scheme.

Scheme MLxF2 (3.4) for the linear flux  $f(u) = au$  can be written as

$$V_{j+1/2}^{n+1} = \sum_{i=-\eta-1}^{\eta+1} \tilde{\rho}_{-i} V_{j+i}^n, \text{ where} \quad (3.8)$$

$$\tilde{\rho}_{-i} = \left[ \frac{1}{2} \tilde{w}_{-i} + a\lambda w_{-i} + \left( \frac{1}{2} - a\lambda \right) w_{-i+1} \right], \quad (3.9)$$

and the  $\tilde{w}_i$  are the mollification weights for the  $(\eta + 1)$ -discrete mollification.

As above, we can draw a picture of the maximum  $a\lambda$  allowed while keeping iteration MLxF2 (3.8) stable under periodic boundary conditions. The CFL bounds and TVD region for the scheme are shown in figure 3.1. Method MLxF2 is stable and monotone up to values of  $a\lambda$  greater than  $\frac{1}{2}$ .

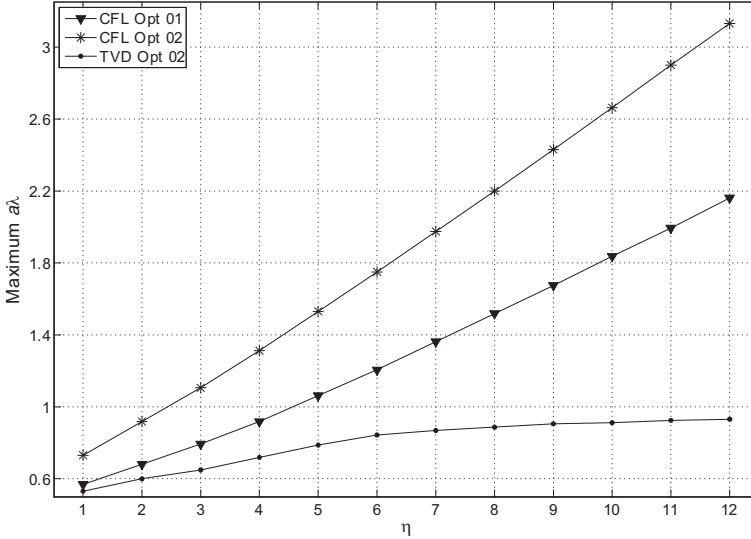


Figure 3.1. Mollified Lax-Friedrichs methods

### 3.2 MOLLIFIED NESSYAHU-TADMOR (MNT) SCHEMES

We consider the second order staggered NT algorithm developed in Nessyahu and Tadmor 1990 given by the following equations:

$$v_j^{n+1/2} = v_j^n - \frac{\lambda}{2} f'_j, \tag{3.10}$$

$$v_{j+1/2}^{n+1} = \frac{1}{2} (v_j^n + v_{j+1}^n) + \frac{1}{8} (v'_j - v'_{j+1}) - \lambda \left[ f(v_{j+1}^{n+1/2}) - f(v_j^{n+1/2}) \right]. \tag{3.11}$$

Here,  $f'_j$  and  $v'_j$  are discrete derivatives computed by a minmod-based limiter. The predictor-corrector scheme (3.10)-(3.11) is second order accurate with stability condition

$$\lambda \max_u |f'(u)| \leq 1/2.$$

#### 3.2.1 METHODS MNT1 AND MNT2

The first two mollified NT methods are:

1. MNT1, given by

$$V_j^{n+1/2} = V_j^n - \frac{\lambda}{2} f'_j, \tag{3.12}$$

$$V_{j+1/2}^{n+1} = \frac{1}{2} (V_j^n + V_{j+1}^n) + \frac{1}{8} (Z'_j - Z'_{j+1}) - \lambda \left[ F_{j+1}^{n+1/2} - F_j^{n+1/2} \right],$$

where  $Z_j^i$  denotes the undivided discrete slope of  $Z^n$  at the point  $x_j$  and the numerical flux  $F_j^{n+1/2}$  represents the space mollification of  $f(V^{n+1/2})$  at the point  $x_j$ .

2. MNT2, given by

$$\begin{aligned}
 V_j^{n+1/2} &= V_j^n - \frac{\lambda}{2} f'_j, \\
 V_{j+1/2}^{n+1} &= \frac{1}{2} (V_j^n + Z_{j+1}^n) + \frac{1}{8} (Z_j^i - Z_{j+1}^i) - \lambda [F_{j+1}^{n+1/2} - F_j^{n+1/2}],
 \end{aligned}
 \tag{3.13}$$

where variables are defined as before.

The numerical approximation of derivatives is not an easy task, since it is of paramount importance for the method to be nonoscillatory, as is clearly stated in Shu (1998) and Zhang and Shu (2007). Furthermore, the presence of nonlinear ingredients prevents us from writing (3.12) and (3.13) as discrete convolutions of the form (3.6), even for the case of linear flux.

In order to obtain some information of CFL and TVD bounds for (3.12) and (3.13), we proceed as follows. We consider the linear flux case  $f(u) = au$  and handle the discrete undivided slopes as computed by central differences. This puts the new methods in an entirely linear environment and the analysis of the previous subsection can be repeated. The results are on figure 3.2.

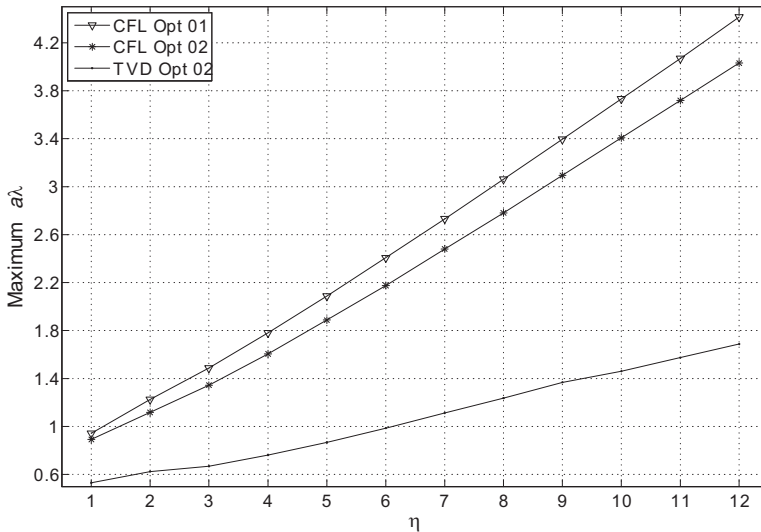


Figure 3.2. Mollified NT schemes MNT1 and MNT2

According to figure 3.2 both options of mollified NT schemes offer less restrictive CFL bounds. However, scheme MNT2 yields a gain in the TVD region (shown in

the figure) and scheme `MNT1` does not. Actually, the second option, with central differences as slope and a linear flux, has a region of monotonicity that contains the TVD region of the original NT method. That is, method `MNT2` is a convergent stabilization of NT but, since it is linear, it is at most first order accurate (Theorem 15.6 of LeVeque, 1990.)

### 3.2.2 METHOD MNT3

This is work in progress, since we still do not know whether it is possible to have a high resolution nonoscillatory method based on mollification. The compulsory nonlinear component is based on nonlinear  $\eta$ -discrete mollification and the MC limiter. If we set  $\eta = 0$  in `MNT3`, the scheme is the prototype of second order NT schemes developed in Nessyahu and Tadmor (1990) that we denote simply as NT.

The `MNT3` scheme is defined by

$$\begin{aligned} V_j^{n+1/2} &= V_j^n - \frac{\lambda}{2} F'_j \\ V_{j+1/2}^{n+1} &= \frac{1}{2} (V_j^n + V_{j+1}^n) + \frac{1}{8} (V'_j - V'_{j+1}) - \lambda \left[ G_{j+1}^{n+1/2} - G_j^{n+1/2} \right], \end{aligned} \quad (3.14)$$

where,  $F'_j$  and  $V'_j$  are discrete derivatives obtained by the MC limiter and  $G_j^{n+1/2}$  denotes the nonlinear  $\eta$ -mollification of  $f(V^{n+1/2})$  at  $x_j$ . As can be seen from the numerical experiments, method `MNT3` is second order accurate and has less stability restrictions than NT. We do not know whether it allows spurious oscillations or not, but the experiments are promising.

## 3.3 NUMERICAL EXPERIMENTS

In the tables, the errors for the nonlinear problems are measured against a reference solution obtained by using the original NT scheme with 4096 space subdivisions. Likewise, in the figures, the reference solutions are obtained by the NT scheme with  $h = 1/1024$  and  $\text{CFL} = 0.475$ . Additionally, the meaning of CFL is the number  $\lambda \max_u |f'(u)|$ , where such a maximum is computed on the relevant range of values of  $u$ . Finally, these CFL values are taken from figures 3.1 and 3.2 for the corresponding  $\eta$ .

**Example 13.** *Linear advection.* Equation  $u_t + u_x = 0$ , with initial data  $u(x, 0) = \sin(2\pi x)$ . The orders of accuracy are computed at  $T = 5.0$  and the results are in table 3.1.

**Example 14.** *Burgers' equation.*  $u_t + (\frac{1}{2}u^2)_x = 0$ , with initial data  $u(x, 0) = 1 + \frac{1}{2} \sin(\pi x)$ . The accuracy of the computations was checked at  $T = 0.33$ , that is,

**Table 3.1.** Mollified NT methods applied to the advection equation

Method	$N$	$L_1$ error	$L_1$ order	$L_\infty$ error	$L_\infty$ order
NT	32	1.1157e-2	–	2.2236e-2	–
	64	2.8793e-3	1.9541	7.6846e-3	1.5329
CFL = 0.475	128	7.1232e-4	2.0151	3.1640e-3	1.2802
	256	1.7172e-4	2.0524	1.2692e-3	1.3178
MNT1	32	4.9259e-1	–	4.9121e-1	–
$\eta = 3$	64	1.3039e-1	1.9176	1.3009e-1	1.9168
CFL = 1.4563	128	3.2827e-2	1.9898	3.2818e-2	1.9870
	256	8.2137e-3	1.9988	8.2136e-3	1.9984
MNT2	32	6.3531e-1	–	6.3531e-1	–
$\eta = 3$	64	3.7415e-1	0.76385	3.7415e-1	0.75713
CFL = 1.0063	128	2.0557e-1	0.86400	2.0557e-1	0.86401
	256	1.0833e-1	0.92416	1.0833e-1	0.92391
MNT3	32	8.7196e-2	–	1.2227e-1	–
$\eta = 3$	64	4.8267e-2	0.8532	6.6076e-2	0.8878
CFL = 0.5	128	1.2263e-2	1.9767	1.2366e-2	2.4177
	256	3.0815e-3	1.9926	3.0811e-3	2.0049
MNT3	256	2.4657e-3	–	2.4567e-3	–
$\eta = 3$	512	6.1666e-4	1.9995	6.1440e-4	1.9995
CFL = 0.66875	1024	1.5401e-4	2.0015	1.5352e-4	2.0007
	2048	3.8472e-5	2.0011	3.8380e-5	2.0000

**Table 3.2.** Mollified NT methods applied to Burgers' equation

Method	$N$	$L_1$ error	$L_1$ order	$L_\infty$ error	$L_\infty$ order
NT	32	1.5678e-3	–	4.8091e-3	–
	64	3.4827e-4	2.1704	1.8998e-3	1.3399
CFL = 0.475	128	7.5423e-5	2.2071	6.3009e-4	1.5922
	256	1.6863e-5	2.1612	2.0354e-4	1.6302
MNT1	32	6.9796e-3	–	2.6644e-2	–
$\eta = 2$	64	1.8380e-3	1.9250	8.4934e-3	1.6494
CFL = 1.1563	128	4.6429e-4	1.9850	2.3333e-3	1.8640
	256	1.1599e-4	2.0010	6.0026e-4	1.9587
MNT3	32	3.4145e-3	–	1.2722e-2	–
$\eta = 2$	64	7.7652e-4	2.1366	3.4765e-3	1.8716
CFL = 0.625	128	1.7900e-4	2.1170	8.7231e-4	1.9947
	256	4.1746e-5	2.1003	2.1341e-4	2.0312

before the shock develops and the results are presented in table 3.2. Figures 3.3 and 3.4 show the computed solutions by NT and MNT3 at  $T = 0.5$  and  $T =$

1 respectively. In both cases MNT3 is implemented with  $h = 1/32$ ,  $\eta = 2$  and  $CFL = 0.625$ . NT uses  $h = 1/1024$  and  $CFL = 0.475$ .

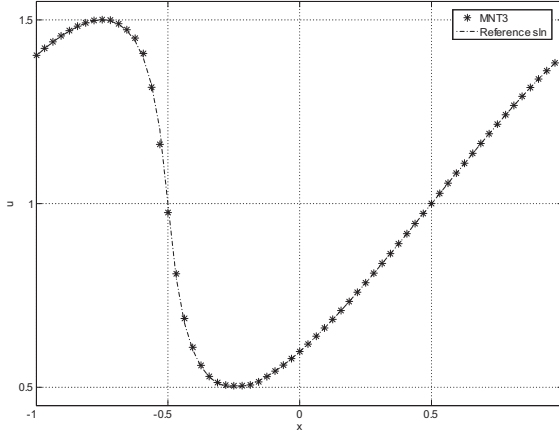


Figure 3.3. MNT3 applied to Burgers' equation at  $T = 0.5$ .

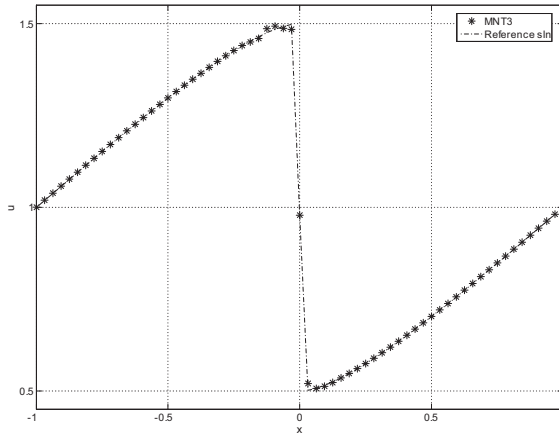


Figure 3.4. MNT3 applied to Burgers' equation at  $T = 1.0$ .

**Example 15.** *Buckley-Leverett problem. A second nonlinear equation to test the MNT3 scheme. The conservation law is*

$$u_t + f(u)_x = 0, \quad f(u) = \frac{u^2}{4u^2 + (1-u)^2},$$

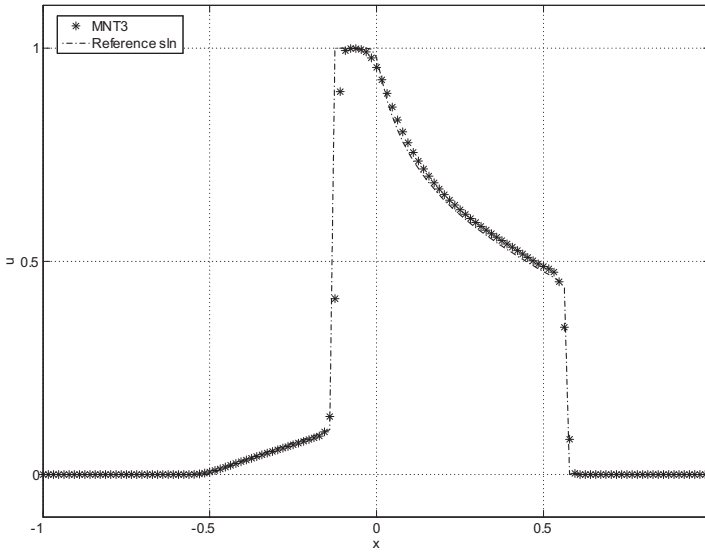


Figure 3.5. MNT3 applied to the Buckley-Leverett problem.

subject to the initial data

$$u_0(x) = \begin{cases} 1, & -0.5 \leq x \leq 0 \\ 0, & \text{otherwise} \end{cases}.$$

Computed solutions are shown at  $t = 1.4$ , with  $\text{CFL} = 0.475$  for the classic NT scheme and  $\eta = 2$  and  $\text{CFL} = 0.625$  for MNT3. The space domain is  $[-1, 1]$  and  $h = 1/64$  for both methods. Results are shown in figure 3.5.

### 3.4 CONCLUDING REMARKS

The addition of mollification to second order NT schemes may not decrease the order of accuracy, preserves consistency and allows larger time steps while preserving stability. It remains to be seen whether the mollified NT schemes can be simultaneously nonoscillatory and of second order of accuracy. So far, the only nonoscillatory method obtained from an NT scheme added with mollification is MNT2, which is at most first order accurate.



## Chapter 4

# NONLINEAR AND DEGENERATE DIFFUSIVE PROBLEMS

The first part of this chapter is based on Acosta and Mejía (2010) and is devoted to a mollified operator splitting strategy for nonlinear convection-diffusion equations. The second part, based on Acosta, Bürger, and Mejía (2012), consists of a brief discussion about a mollified method for the solution of strongly degenerate parabolic equations.

### 4.1 MOLLIFIED OPERATOR SPLITTING

The goals here are stabilization and convergence by means of a mollified explicit scheme for the numerical solution of the scalar nonlinear convection-diffusion equation

$$u_t + f(u)_x = \varepsilon u_{xx}, \quad u(x, 0) = u_0(x). \quad (4.1)$$

The mollified scheme has the form of an operator splitting, or fractional step, and it is a modification of an operator splitting method of Karlsen and Risebro (1997). The first step of the method is a convergent numerical scheme for the solution of the nonlinear conservation law

$$u_t + f(u)_x = 0. \quad (4.2)$$

The second step is a numerical scheme, based on discrete mollification, for the solution of the heat equation

$$u_t = \varepsilon u_{xx}, \quad u(x, 0) = u_0(x). \quad (4.3)$$

Many fractional step methods (LeVeque, 2002; Faragó, Gndt, and Havasi, 2008) are formally only first-order accurate and, according to our experiments, so is the new method presented here. Two examples of higher order operator splitting methods are the corrected operator splitting of Karlsen *et al.* (2001) and the additive and iterative splitting of Faragó, Gndt, and Havasi (2008). Nevertheless, the relevant matter is that mollification, for being explicit, unconditionally stable and easy to compute, is a suitable method for the solution of the diffusive step in operator splitting frameworks. For the conservation law  $u_t + f(u)_x = 0$ , any convergent method will do the job (see Holden and Risebro, 2002 and LeVeque, 2002 for well chosen selections).

Let us discuss first the mollification implementation for the solution of the heat equation.

### 4.1.1 THE DIFFUSION STEP

Consider a semi-discrete approximation of the Cauchy problem for the heat equation

$$\begin{cases} u_t = \varepsilon u_{xx}, \\ u(x, 0) = u_0(x) \end{cases} . \quad (4.4)$$

Let  $k > 0$  be the time discretization parameter and  $v^n(x)$  be the numerical approximation of  $u$  at instant  $t^n = nk$ . Then the convolution with the mollification kernel provides the semi-discrete scheme given by

$$v^{n+1}(x) = (\kappa_{3\delta} * v^n)(x) \quad (4.5)$$

that is consistent with equation (4.4). Assuming  $v^n(x)$  is sufficiently smooth as a function of  $x$ , we have

$$\begin{aligned} v^{n+1}(x) &= (\kappa_{3\delta} * v^n)(x) = \int_{-3\delta}^{3\delta} \kappa_{3\delta}(-s) v^n(x+s) ds \\ &= \int_{-3\delta}^{3\delta} \kappa_{3\delta}(-s) \left\{ v^n(x) + s v'(x) + \frac{s^2}{2} v''(x) + \frac{s^3}{6} v'''(x) + \frac{s^4}{24} v^{(4)}(\xi_s) \right\} ds \\ &= v^n(x) + v^{(2)}(x) \int_{-3\delta}^{3\delta} \frac{s^2}{2} \kappa_{3\delta}(-s) ds + \int_{-3\delta}^{3\delta} \frac{s^4}{24} \kappa_{3\delta}(-s) v^{(4)}(\xi_s) ds, \end{aligned} \quad (4.6)$$

where the spatial derivatives  $v^{(2)}(x)$  and  $v^{(4)}(\xi_s)$  are evaluated at  $t = t^n$ . Here  $\xi_s$  is a number between  $x$  and  $x+s$ . By integration by parts we obtain

$$v^{n+1}(x) = v^n(x) + \frac{\alpha \delta^2}{4} v^{(2)}(x) + O(\delta^4), \quad (4.7)$$

where  $\alpha = 1 - 6A_3 \exp(-9) > 0$ . Based on the differential equation, iteration (4.7) may be written

$$u_t + O(k) = \frac{\alpha \delta^2}{4k} u_{xx} + O(\delta^4/k),$$

thus, the semidiscrete approximation is consistent with the heat equation whenever

$$\varepsilon = \frac{\alpha \delta^2}{4k} \text{ and } \frac{\delta^4}{k} \rightarrow 0 \text{ as } k \rightarrow 0. \quad (4.8)$$

This indicates that, given  $k > 0$ , the mollification parameter  $\delta > 0$  should be selected according to the equality

$$k = \frac{\alpha \delta^2}{4\varepsilon}. \quad (4.9)$$

Now, let  $V^n$  be a piecewise constant approximation of  $v^n$  over the space grid  $X = \{x_j : x_j = x_0 + jh, \quad j \in \mathbb{Z}\}$ . Due to the smoothness of  $v^n$  as a function of  $x$ ,

$$|v^n(x_j) - V^n(x_j)| \leq Ch$$

for some constant  $C$ , thus, by theorem 1,

$$(\kappa_{3\delta} * v^n)(x_j) = (\kappa_{3\delta} * V^n)(x_j) + O(h)$$

so, consistency of the fully discrete method for the heat equation

$$V_j^{n+1} = (JV^n)_j \tag{4.10}$$

takes the form

$$u_t = \varepsilon u_{xx} + O(k) + O(h) + O(\delta^4/k). \tag{4.11}$$

Notice that the selection of  $h$  is independent of  $k$ .

Next, we discuss the stability of scheme (4.10). Given  $k > 0$  and  $h > 0$ , define  $\delta$  by (4.9) and select  $\eta$  from (1.3); that is,  $\eta = \left\lceil \frac{3\delta}{h} - \frac{1}{2} \right\rceil$ . From  $\delta$  and  $\eta$ , compute the weights  $w_i$  and iteration (4.10) can be written

$$V_j^{n+1} = \sum_{i=-\eta}^{\eta} w_{-i} V_{j+i}^n. \tag{4.12}$$

Thus,

$$\begin{aligned} |V_j^{n+1}| &= \left| \sum_{i=-\eta}^{\eta} w_{-i} V_{j+i}^n \right| \\ &\leq \sum_{i=-\eta}^{\eta} w_{-i} |V_{j+i}^n| \leq \sum_{i=-\eta}^{\eta} w_{-i} \|V^n\|_{\infty} = \|V^n\|_{\infty}. \end{aligned}$$

This result is stated next as a theorem.

**Theorem 16.** *Discrete mollification provides a consistent and unconditionally stable (therefore convergent) numerical scheme for the solution of (4.4).*

### 4.1.2 THE OPERATOR SPLITTING METHOD

The differential equation

$$u_t + f(u)_x = 0, \tag{4.13}$$

may be interpreted as the limit of the convection-diffusion equation

$$u_t + f(u)_x = \varepsilon u_{xx}$$

as the viscosity vanishes ( $\varepsilon \rightarrow 0$ .) It is known that the Cauchy problem (4.1) has a unique solution whenever the flux function  $f$  is Lipschitz continuous and the initial data  $u_0$  are in  $L_\infty \cap B.V$ . By  $B.V$ . we mean the space of functions with bounded total variation. We are interested in the solution of this Cauchy problem by an operator splitting technique based on discrete mollification. The idea of the splitting method as well as several theoretical results are inspired by Karlsen and Risebro (1997).

### 4.1.3 THE NONLINEAR CONVECTION-DIFFUSION EQUATION

For the nonlinear Cauchy problem (4.1) consider the operator splitting method

$$v_j^{n+1/2} = v_j^n - \lambda [F(v^n; j) - F(v^n; j-1)] \quad (4.14)$$

$$v_j^{n+1} = \left( Jv^{n+1/2} \right)_j \quad (4.15)$$

in which the first step is a convergent numerical method for the conservation law (3.1), parameter  $\lambda$  is defined as  $\lambda = \frac{k}{h}$  and the second step is mollification with  $\delta$  chosen as in subsection 4.1.1. The convergence of method (4.14) is understood as the convergence to the unique entropy weak solution of (3.1). In particular, we assume the numerical flux  $F(v^n; j)$  is Lipschitz continuous and of the form

$$F(v^n; j) = F(v_{j-p}^n, \dots, v_{j+q}^n).$$

In our analysis, we denote the action of the hyperbolic step (4.14) as  $\mathcal{F}(k)$  and the diffusive step (4.15) by  $\mathcal{H}(k)$ . So, (4.14)-(4.15) can be written as

$$v^{n+1} = \mathcal{H}(k)\mathcal{F}(k)v^n. \quad (4.16)$$

In Acosta and Mejía (2010) we prove that the fractional step method (4.14)-(4.15) is convergent to the unique classical solution of (4.1). More precisely, we check that the new splitting scheme satisfies the hypotheses of Lemma 3.1 and Theorem 3.2 of Karlsen and Risebro (1997) and conclude from there.

The idea of the proof comes from writing (4.14)-(4.15) in a conservative form (recall lemma 3)

$$v_j^{n+1} = v_j^n - \lambda [F_\delta(v^n; j) - F_\delta(v^n; j-1)], \text{ where} \quad (4.17)$$

$$F_\delta(v^n; j) = [JF(v^n; \cdot)]_j - \frac{1}{\lambda} \varphi_j^n,$$

$$\varphi_j^n = \sum_{l=1}^{\eta} \rho_l (v_{j+l} - v_{j+1-l}) \text{ and } \rho_l = \sum_{i=l}^{\eta} w_i.$$

The new flux  $F_\delta(v^n; j)$  is Lipschitz continuous as stated in the following lemma.

**Lemma 17.** *Suppose the numerical flux  $F(v^n; j)$  of (4.14) satisfies the Lipschitz continuity condition*

$$|F(u_{j-p}, \dots, u_{j+q}) - F(u_{j-p-1}, \dots, u_{j+q-1})| \leq L \sum_{i=-p}^q |u_{j+i} - u_{j+i-1}|,$$

for some constant  $L > 0$ . Then the numerical flux of the mollified scheme (4.17) satisfies

$$\begin{aligned} & |F_\delta(u_{j-p-\eta}, \dots, u_{j+q+\eta}) - F_\delta(u_{j-p-\eta-1}, \dots, u_{j+q+\eta-1})| \\ & \leq \left(L + \frac{\rho_1}{\lambda}\right) \sum_{i=-p-\eta}^{q+\eta} |u_{j+i} - u_{j+i-1}| \end{aligned}$$

where  $\rho_1 = \sum_{i=1}^{\eta} w_i$ .

*Proof.* We omit the proof, it can be found in Acosta and Mejia (2010).  $\square$

Stability and TV-stability properties of (4.14)-(4.15) are the subject of the next lemma. In the sequel,  $C_T$  denotes a positive constant that depends on the final time  $T > 0$ , but it is independent from the time step  $n$  and of the grid sizes  $h$  and  $k$ .

**Lemma 18.** *Suppose that under the CFL condition*

$$\lambda |f'(u)| \leq \mu, \quad (4.18)$$

the hyperbolic step (4.14) satisfies: for a prescribed final time  $T > 0$  there exists a constant  $C_T > 0$  such that for all  $n$  with  $nk \leq T$

$$\|\mathcal{F}(k)v^{n-1}\|_\infty \leq (1 + Ck) \|v^{n-1}\| \quad (4.19)$$

$$TV(\mathcal{F}(k)v^{n-1}) \leq (1 + Ck) TV(v^{n-1}) \quad (4.20)$$

Then, under the same CFL restriction (4.18) and for the same constant  $C_T$  the following inequalities hold:

$$\|v^n\|_\infty \leq (1 + C_T k) \|v^{n-1}\|, \quad (4.21)$$

$$TV(v^n) \leq (1 + C_T k) TV(v^{n-1}). \quad (4.22)$$

*Proof.* The stability bound (1.10) and the hypotheses yield

$$\begin{aligned} \|v^n\|_\infty &= \|\mathcal{H}(k)\mathcal{F}(k)v^{n-1}\|_\infty = \|J[\mathcal{F}(k)v^{n-1}]\|_\infty \\ &\leq \|\mathcal{F}(k)v^{n-1}\|_\infty \leq (1 + C_T k) \|v^{n-1}\|. \end{aligned}$$

Similarly, from the TVD property of mollification (lemma 4), we obtain

$$\begin{aligned} TV(v^n) &= TV(\mathcal{H}(k)\mathcal{F}(k)v^{n-1}) = TV(J[\mathcal{F}(k)v^{n-1}]) \\ &\leq TV(\mathcal{F}(k)v^{n-1}) \leq (1 + C_T k) TV(v^{n-1}). \end{aligned}$$

□

The following bound is also required for convergence.

**Lemma 19.** *Let  $v^n$  be defined by (4.14)-(4.15),  $T > 0$  be the prescribed final time and  $M$  be the maximum number of time steps allowed. Then, under the conditions of lemmas 17 and 18, for all non-negative  $m, n$  so that  $(n+m)k \leq Mk \leq T$ , it holds*

$$\|v^{n+m} - v^n\|_1 \leq C g_M(mk),$$

where  $g_M(mk)$  is a continuous function with  $g_M(0) = 0$  and  $C$  is a constant dependent on the final time  $T$ , but independent from  $n, m$  and  $k$ .

*Proof.* We omit the proof which is given in Acosta and Mejía (2010). □

Now we have all the ingredients for the convergence result, which is analogous to Theorem 3.2 of Karlsen and Risebro (1997).

**Theorem 20.** *The operator splitting method (4.14)-(4.15) produces a sequence of solutions convergent to the classical solution of*

$$u_t + f(u)_x = \varepsilon u_{xx}$$

as  $k \rightarrow 0$ , whenever the initial data  $u_0(x)$  is of class  $L_\infty \cap B.V.$

The gain in using a mollification based operator splitting technique for a convection-diffusion problem is that no new stability bounds are imposed, only the CFL restrictions of the advection step. In comparison with other operator splitting approaches, this is an improvement. For instance, if we replace discrete mollification for an explicit central finite difference scheme for the heat equation, then we will have to deal with a CFL restriction of the form

$$2\varepsilon \frac{\lambda}{h} \leq 1, \tag{4.23}$$

which means the method is more expensive than (4.18). If the alternative were an implicit finite difference scheme or some finite element approach, then we avoid (4.23), but a system of equations arise at each time step. We are confident discrete mollification is a serious alternative when solving inverse problems like flux identification (Berres *et al.* 2005) or for long-time behavior studies and steady-state computing (Bürger, Karlsen and Towers, 2005).

#### 4.1.4 IMPLEMENTATION

For the numerical implementation of (4.14)-(4.15) we take into account the following three conditions:

- To choose  $k = \lambda h$  in such a way that (4.14) is stable.
- To choose  $\delta$  in order to preserve the consistency of (4.15). This is achieved by selecting

$$\delta = 2\sqrt{\frac{\varepsilon}{\alpha}k}.$$

- For the correct application of the mollification kernel, the initial condition should be discretized in a sufficiently large number of points  $N$ . More precisely,

$$N \geq 2\eta + 1 \geq 6\delta/h = 12\sqrt{\frac{\varepsilon}{\alpha}\lambda}h^{-1/2}.$$

The particular case  $\lambda = 1$  and  $\varepsilon = 1$  is shown in figure 4.1.

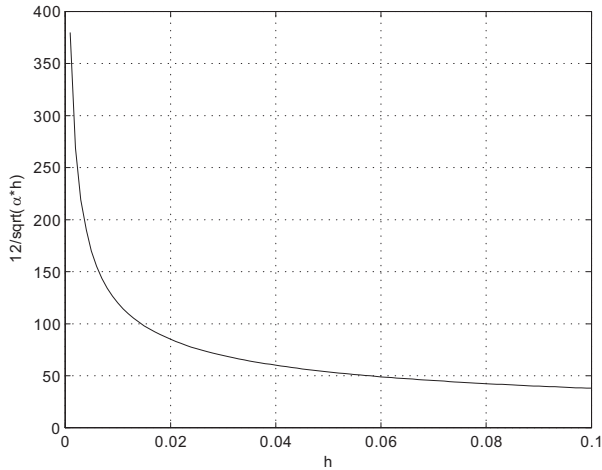


Figure 4.1. Minimum number of points for discretization of  $u_0(x)$

#### 4.1.5 NUMERICAL EXPERIMENTS

We include some illustrations of the operator splitting algorithm but the first example uses only the diffusive part. The second example is Burgers' equation with viscosity and a source term. Examples 23 (Cecchi and Pirozzi, 2005) and 24 consist on the application of the operator splitting method to the solution of a linear and a nonlinear one dimensional convection-diffusion problems respectively.

**Example 21** (Heat Equation). *The problem is*

$$\begin{aligned} u_t &= u_{xx}, & 0 < x < 2\pi, \quad t > 0 \\ u(x, 0) &= \sin(x), & 0 < x < 2\pi, \quad t > 0 \\ u(0, t) &= u(2\pi, t), & t > 0. \end{aligned}$$

The exact solution is  $u(x, t) = e^{-t} \sin(x)$  and mollification was implemented with periodic boundary conditions (section 1.5.) Any diffusive step should work well by itself (Cheng and Shu, 2009), not only as part of the operator splitting combination. Table 4.1 summarizes the results. Parameters are  $t = 1.0$ ,  $\lambda = 1$  and  $h = 2\pi/n$ .

**Table 4.1.** A heat equation solved by discrete mollification

$n$	$L_1 - er$	$L_1 - or$	$L_2 - er$	$L_2 - or$	$L_\infty - er$	$L_\infty - or$
64	4.1025e-3	–	4.1025e-3	–	4.1025e-3	–
128	2.0552e-3	0.99722	2.0552e-3	0.99722	2.0552e-3	0.99722
256	1.0282e-3	0.99916	1.0282e-3	0.99916	1.0282e-3	0.99916
512	5.2091e-4	0.98101	5.2091e-4	0.98101	5.2091e-4	0.98101

**Example 22.** *Viscous Burgers' equation with a source.*

$$u_t + \left(\frac{u^2}{2}\right)_x = \varepsilon u_{xx} - \pi \cos(\pi x)u.$$

The boundary conditions are  $u(0, t) = 1$  and  $u(1, t) = -0.1$ . The initial condition is

$$u(x, 0) = \begin{cases} 1, & \text{if } 0 \leq x < 0.3 \\ -0.1, & \text{if } 0.3 \leq x < 1.0 \end{cases}$$

Experiments were performed with a viscosity coefficient  $\varepsilon = 0.01$  and the reference solution is computed with the centered scheme

$$u_j^{n+1} = u_j^n - \frac{\lambda}{2} (f(u_{j+1}^n) - f(u_{j-1}^n)) + \varepsilon \frac{\lambda}{h} (u_{j+1}^n - 2u_j^n + u_{j-1}^n), \tag{4.24}$$

using  $h = 1/4096$  and  $\lambda = 1/128$ . For the mollified scheme (4.14)-(4.15), the parameters are  $\lambda = 1/1.1$  and  $h = 1/n$ . The errors are computed at  $t = 2.0$  and are reported in table 4.2. Figure 4.2 shows the solution obtained for  $h = 1/256$ .

**Example 23.** *A linear convection-diffusion equation with known exact solution.*

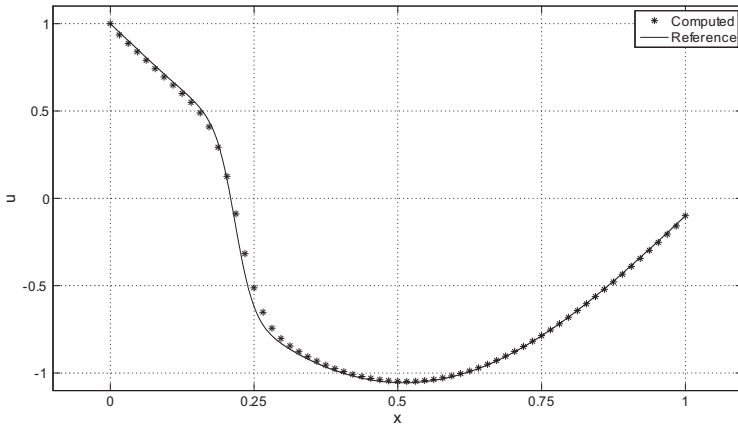
$$u_t + au_x = bu_{xx}, \quad -1 < x < 4, \quad 0 < t < 2,$$

where

$$u(x, t) = \frac{1}{\sqrt{1+t}} \exp\left(-\frac{(x-a(1+t))^2}{4b(1+t)}\right).$$

**Table 4.2.** Order test for solution of Burgers' equation with viscosity by the mollified operator splitting algorithm

$n$	$L_1 - er$	$L_1 - or$	$L_2 - er$	$L_2 - or$	$L_\infty - er$	$L_\infty - or$
64	1.1674e-1	–	1.8909e-1	–	5.0860e-1	–
128	5.1266e-2	1.1872	8.7231e-2	1.1162	2.6972e-1	0.91507
256	2.0618e-2	1.3141	3.4270e-2	1.3479	1.1281e-1	1.2576
512	1.2487e-2	0.72348	1.5102e-2	1.1822	3.5302e-2	1.6761



**Figure 4.2.** Burgers' equation with a source solved by the mollified operator splitting algorithm

**Table 4.3.** Order test for the mollified operator splitting algorithm

$n$	$L_1 - er$	$L_1 - or$	$L_2 - er$	$L_2 - or$	$L_\infty - er$	$L_\infty - or$
64	6.1776e-2	–	5.4565e-2	–	6.1456e-2	–
128	3.2175e-2	0.94111	2.8793e-2	0.92226	3.2967e-2	0.89853
256	1.6385e-2	0.97356	1.4698e-2	0.97010	1.6894e-2	0.96451
512	8.2636e-3	0.98753	7.4239e-3	0.98537	8.5556e-3	0.98157

For the advection step we use Lax-Friedrichs method with parameters  $a = 1, b = 1/64$  and  $\lambda = 1$ . Results for  $h = 5/n$  are shown in table 4.3 and figure 4.3. The figure represents the computed solution for  $h = 5/128$  at time  $t = 2$ .

**Example 24.** *Nonlinear convection-diffusion problem*

$$u_t + \left(\frac{u^2}{2}\right)_x = \varepsilon u_{xx}, \quad -1 < x < 4, \quad 0 < t < 1,$$

with known solution

$$u(x, t) = a - c \tanh\left(\frac{c}{2b}(x - at)\right).$$

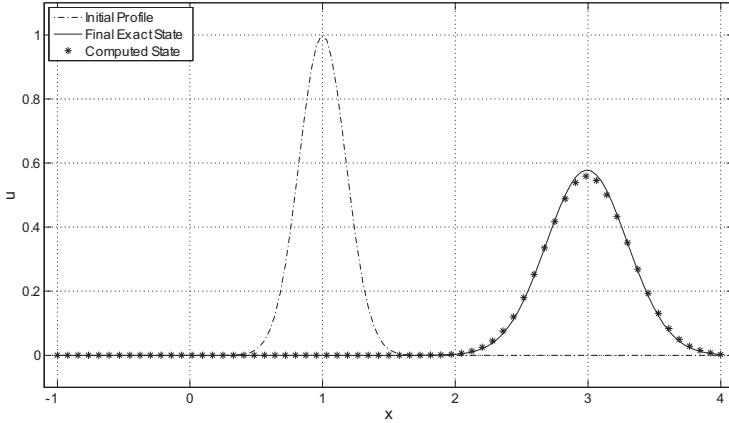


Figure 4.3. Solution of a linear convection-diffusion equation by the mollified operator splitting algorithm

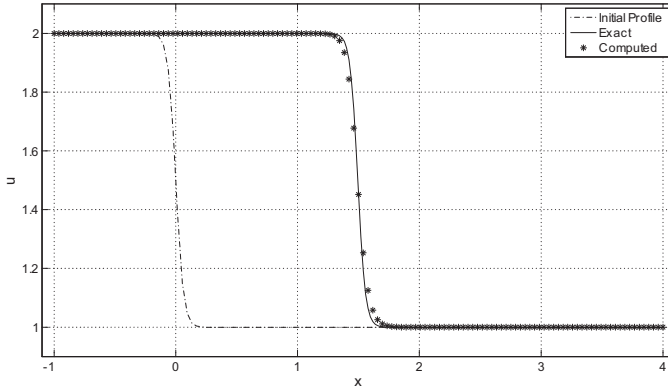
The nonlinear conservation law is solved by the second order NT method with parameters  $a = 1.5, c = 0.5, b = \varepsilon = 1/64$  and  $\lambda = 0.25$ . The results with  $h = 5/n$  are shown in table 4.4 and figure 4.4. For the mollification step  $h = k$ . For figure 4.4,  $t = 1$  and  $h = 5/128$ .

Table 4.4. Order test for the solution of Burgers' equation by the mollified operator splitting method

$n$	$L_1 - er$	$L_1 - or$	$L_2 - er$	$L_2 - or$	$L_\infty - er$	$L_\infty - or$
64	4.5115e-3	—	1.5155e-2	—	6.2002e-2	—
128	2.0529e-3	1.1359	7.8781e-3	0.94387	3.2932e-2	0.91282
256	9.3614e-4	1.1329	3.6678e-3	1.1029	1.7523e-2	0.91024
512	4.2851e-4	1.1274	1.7049e-3	1.1052	8.2849e-3	1.0807

### 4.1.6 CONCLUDING REMARKS

1. Operator splitting methods are efficient solvers for convection-diffusion problems.
2. Discrete mollification is a suitable method for solving diffusive steps of operator splitting methods for convection-diffusion equations. In particular, it is a natural resource for the solution of the linear homogeneous heat equation.



**Figure 4.4.** Numerical solution of Burgers' equation by the mollified operator splitting method

### 4.2 STRONGLY DEGENERATE PARABOLIC EQUATIONS

We consider some particular finite difference methods for the following initial value problem for a strongly degenerate parabolic equation:

$$u_t + f(u)_x = A(u)_{xx}, \quad (x, t) \in \Pi_T := \mathbb{R} \times (0, T), \quad T > 0, \quad (4.25)$$

$$u(x, 0) = u_0(x), \quad x \in \mathbb{R}, \quad (4.26)$$

where the integrated diffusion coefficient  $A$  is defined by

$$A(u) = \int_0^u a(s) ds, \quad a(u) \geq 0, \quad a \in L^\infty(\mathbb{R}) \cap L^1(\mathbb{R}). \quad (4.27)$$

The function  $a$  is allowed to vanish on  $u$ -intervals of positive length, on which (4.25) degenerates to a first-order scalar conservation law. Therefore, (4.25) is called *strongly degenerate*. Solutions of (4.25), (4.26) are, in general, discontinuous even if  $u_0$  is smooth, and need to be defined as weak solutions along with an entropy condition to select the physically relevant solution, the *entropy solution*. Applications of degenerate parabolic equations include two-phase flow in porous media, traffic flow and sedimentation-consolidation processes.

Evje and Karlsen (2000) introduced an explicit monotone difference scheme convergent to entropy solutions of (4.25), (4.26) based on the first-order accurate, monotone Engquist-Osher numerical flux (Engquist and Osher, 1981) for the convective part combined with a conservative discretization of the degenerate diffusion term. If  $h$  and  $k$  denote the spatial meshwidth and the time step, respectively, they proved convergence of the scheme to an entropy solution as  $h, k \downarrow 0$  provided that the following CFL stability condition is satisfied:

$$\lambda \|f'\|_\infty + 2\mu \|a\|_\infty \leq 1, \quad \lambda := k/h, \quad \mu := k/h^2. \quad (4.28)$$

In our case, the application of discrete mollification to the scheme introduced in Evje and Karlsen (2000) leads to a mollified scheme whose CFL stability condition is given by

$$\lambda \|f'\|_{\infty} + 2\mu\varepsilon_{\eta} \|a\|_{\infty} \leq 1, \quad \text{with } \varepsilon_{\eta} := C_{\eta}(1 - w_0) \in (0, 1) \quad (4.29)$$

and

$$C_{\eta} := \left( \sum_{i=-\eta}^{\eta} i^2 w_i \right)^{-1}.$$

Since  $\varepsilon_{\eta} \in (0, 1)$ , for given  $h$  condition (4.29) provides the way to use values of  $k$  that are up to several times larger than for the standard, un-mollified version of the scheme with the restriction (4.28). This accelerates the given scheme at the mild cost of adding some moderate error.

## 4.2.1 THE SCHEMES

### The Un-Mollified Scheme (Basic Scheme)

We select a mesh size  $h > 0$  and a time step  $k > 0$  such that there exists an integer  $N$  so that  $Nk = T$ . Let  $u_j^n$  denote the value of the difference approximation at  $(jh, nk)$  for  $j \in \mathbb{Z}$  and  $n = 1, \dots, N$ . We then discretize the initial data by

$$u_j^0 := \frac{1}{h} \int_{(j-1/2)h}^{(j+1/2)h} u_0(x) dx, \quad j \in \mathbb{Z},$$

and compute the solution values at time level  $t^{n+1}$  from those at time  $t^n$  by the explicit marching formula

$$u_j^{n+1} = u_j^n - \lambda \Delta_+ F^{\text{EO}}(u_{j-1}^n, u_j^n) + \mu \Delta^2 A(u_j^n). \quad (4.30)$$

Here  $j \in \mathbb{Z}$ ,  $n = 0, 1, 2, \dots, N-1$ , the implemented difference operators are

$$\Delta_+ V_{j-1} = \Delta_- V_j = \Delta V_{j-1/2} := V_j - V_{j-1}, \quad \Delta_0 V_j := (V_{j+1} - V_{j-1})/2,$$

$$\Delta^2 V_j := V_{j+1} - 2V_j + V_{j-1},$$

and use the Engquist-Osher numerical flux (Engquist and Osher, 1981) given by

$$F^{\text{EO}}(u_j^n, u_{j+1}^n) = f^+(u_j^n) + f^-(u_{j+1}^n), \quad \text{where}$$

$$f^+(u) := f(0) + \int_0^u \max\{f'(s), 0\} ds, \quad f^-(u) := \int_0^u \min\{f'(s), 0\} ds.$$

Under the CFL condition (4.28) the scheme (4.30) is monotone; therefore, first-order accurate, and converges to the unique entropy solution Evje and Karlsen, 2000.

The new scheme will be based on replacing the term  $\mu\Delta^2 A(u_j^n)$  in (4.30) by a different expression involving discrete mollification.

### The Mollified Scheme

In order to motivate the new scheme we introduce additional properties for discrete mollification that are not explicitly given in theorem 1. Let  $g$  be a sufficiently smooth real function, set  $y_j = g(x_j)$ , and employ a Taylor expansion

$$y_{j+i} = y_j + (ih)g'(x_j) + \frac{1}{2}(ih)^2g''(x_j) + \frac{1}{6}(ih)^3g'''(x_j) + \frac{1}{24}(ih)^4g^{(4)}(\xi_{j,i}),$$

where  $\xi_{j,i}$  is a real number between  $x_j$  and  $x_{j+i}$ . Then, defining

$$C_\eta := \left( \sum_{i=-\eta}^{\eta} i^2 w_{-i} \right)^{-1},$$

we can write

$$[J_\eta y]_j = \sum_{i=-\eta}^{\eta} w_{-i} y_{j+i} = y_j + \frac{h^2}{2C_\eta} g''(x_j) + \frac{h^4}{24} \sum_{i=-\eta}^{\eta} i^4 w_{-i} g^{(4)}(\xi_{j,i}) \quad (4.31)$$

and from here we obtain

$$\left| [J_\eta y]_j - g(x_j) - \frac{h^2}{2C_\eta} g''(x_j) \right| \leq Ch^4 \quad \text{for all } j \in \mathbb{Z}. \quad (4.32)$$

The new scheme is based on (4.32), which implies the approximation

$$g''(x_j) = \frac{2C_\eta}{h^2} \left( [J_\eta g]_j - g(x_j) \right) + \mathcal{O}(h^2) \quad \text{as } h \downarrow 0. \quad (4.33)$$

Assume now for the moment that  $A(u)$  is a smooth function of  $x$ . Then we have

$$\begin{aligned} \frac{1}{h^2} \Delta^2 A(u(x_j)) &= A(u)_{xx}|_{x=x_j} + \mathcal{O}(h^2), \\ \frac{2C_\eta}{h^2} \left( [J_\eta A(u)]_j - A(u(x_j)) \right) &= A(u)_{xx}|_{x=x_j} + \mathcal{O}(h^2), \end{aligned}$$

so, we obtain a consistent scheme after replacing the expression  $\mu\Delta^2 A(u_j^n)$  on the right-hand side of (4.30) by  $2\mu C_\eta ([J_\eta A(u)]_j - A(u_j^n))$ . The new scheme is

$$u_j^{n+1} = u_j^n - \lambda \Delta_+ F^{\text{EO}}(u_{j-1}^n, u_j^n) + 2\mu C_\eta \left( [J_\eta A(u^n)]_j - A(u_j^n) \right). \quad (4.34)$$

This scheme (4.34) has the more favorable CFL condition given by (4.29). Section 3.2 of Acosta, Bürger and Mejía (2012) presents a detailed convergence analysis for this scheme which we omit here. To illustrate the quality of numerical results, we include example 3 of Acosta, Bürger and Mejía (2012).

### 4.2.2 EXAMPLE: SEDIMENTATION

The sedimentation-consolidation process of solid-liquid suspensions, see e.g. Berres *et al.* (2003), important in mining activities, is modeled by a strongly degenerate parabolic equation in which the functions  $f(u)$  and  $A(u)$  model the effects of hindered settling and sediment compressibility, respectively, of a suspension of local solids volume fraction  $u$ . This example is presented by Bürger and Karlsen (2001, section 4.2.1) and Bürger, Kozakevicius and Sepúlveda (2007, example 1) where this problem was treated by an adaptive multiresolution technique. So, results may be compared. We consider (4.25) with the choice

$$f(u) = \begin{cases} v_\infty u(1-u)^C & \text{for } 0 < u < u_{\max}, \\ 0 & \text{otherwise,} \end{cases}$$

with parameters  $v_\infty < 0$ ,  $C > 1$  and a diffusion function  $A(u)$  defined by (4.27) with

$$a(u) = \frac{f(u)\sigma'_e(u)}{\Delta_\rho g u}, \quad \text{where} \quad \sigma_e(u) = \begin{cases} 0 & \text{for } 0 < u \leq u_c, \\ \sigma_0[(u/u_c)^\beta - 1] & \text{for } u > u_c, \end{cases}.$$

Here  $\sigma_0 > 0$  and  $\beta > 1$  are parameters,  $\Delta_\rho$  and  $g$  are constants and  $u_c$  is the so-called critical concentration. If  $\beta$  is an integer, then  $A(u)$  can be evaluated in closed form as follows:

$$A(u) = \begin{cases} 0 & \text{for } 0 < u \leq u_c, \\ \mathcal{A}(u) - \mathcal{A}(u_c) & \text{for } u > u_c, \end{cases}$$

where

$$\mathcal{A}(u) = \frac{v_\infty \sigma_0}{\Delta_\rho g u_c^\beta} \sum_{k=1}^{\beta} \left( \prod_{l=1}^k \frac{\beta + 1 - l}{C + l} \right) (1-u)^{C+k} u^{\beta-k}.$$

We consider  $x \in [0, L]$  and time  $t \in [0, T]$ . The initial and boundary conditions are of the form

$$\begin{aligned} u(x, 0) &= u_0(x) \quad \text{for } x \in [0, L], \\ f(u) - A(u)_x &= 0, \quad \text{for } x = 0 \text{ and } x = L. \end{aligned}$$

The values for the parameters are:  $v_\infty = -2.7 \times 10^{-4}$  m/s,  $u_c = 0.07$ ,  $u_{\max} = 0.5$ ,  $C = 21.5$ ,  $\sigma_0 = 1.2$  Pa,  $\beta = 5$ ,  $\Delta_\rho = 1660$  kg/m<sup>3</sup>,  $L = 0.16$  m,  $g = 9.81$  m/s<sup>2</sup> and the initial condition  $u_0(x) = 0.05$  for  $x \in [0, L]$ . The boundary conditions for  $u$  are the same of Bürger and Karlsen (2001, section 4.2.1) and Bürger, Kozakevicius and Sepúlveda (2007, example 1). For the discrete mollification of  $A(u^n)$ , we implement a linear extrapolation at the borders. For instance at  $x = 0$ , we use the knowledge of  $A(u_0^n)$  and  $A(u_0^n)_x = f(u_0^n)$  for getting an interpolating straight line with slope  $f(u_0^n)$ . Table 4.5 shows results of tests with different values of parameters  $h$  and  $\eta$  and different final times. The gain in CPU time is clearly observed and no accuracy is lost in the process.

**Table 4.5.** Comparison of basic and mollified schemes

(a) $T = 400s$							
$1/\Delta x$	Basic Scheme			Mollified Scheme with $\eta = 3$			
	$L^1$ -error	conv. rate	CPU time [s]	$L^1$ -error	conv. rate	CPU time [s]	
64	2.7063e-2	–	0.0199	2.3574e-2	–	0.0135	
128	1.1672e-2	1.2133	0.0388	1.1381e-2	1.0506	0.0257	
256	6.4428e-3	0.8572	0.1134	6.2139e-3	0.8730	0.0845	
512	3.2280e-3	0.9970	0.4365	3.0598e-3	1.0221	0.3255	
$1/\Delta x$	Mollified Scheme with $\eta = 5$			Mollified Scheme with $\eta = 8$			
	$L^1$ -error	conv. rate	CPU time [s]	$L^1$ -error	conv. rate	CPU time [s]	
64	2.1772e-2	–	0.0116	2.1746e-2	–	0.0116	
128	1.1781e-2	0.8860	0.0305	1.2684e-2	0.7777	0.0264	
256	6.3253e-3	0.8972	0.0779	6.6323e-3	0.9354	0.0695	
512	3.0530e-3	1.0509	0.2604	3.1926e-3	1.0548	0.2117	
(b) $T = 2400s$							
$1/\Delta x$	Basic Scheme			Mollified Scheme with $\eta = 3$			
	$L^1$ -error	conv. rate	CPU time [s]	$L^1$ -error	conv. rate	CPU time [s]	
64	4.0720e-2	–	0.1070	3.6909e-2	–	0.0763	
128	2.4783e-2	0.7164	0.2419	2.2321e-2	0.7256	0.1711	
256	1.5127e-2	0.7122	0.8116	1.3774e-2	0.6965	0.5954	
512	6.8969e-3	1.1331	3.5935	6.3001e-3	1.1285	2.5624	
$1/\Delta x$	Mollified Scheme with $\eta = 5$			Mollified Scheme with $\eta = 8$			
	$L^1$ -error	conv. rate	CPU time [s]	$L^1$ -error	conv. rate	CPU time [s]	
64	3.3140e-2	–	0.0751	3.0733e-2	–	0.0757	
128	2.0822e-2	0.6705	0.1676	1.9756e-2	0.6375	0.1598	
256	1.2763e-2	0.7061	0.5181	1.2194e-2	0.6961	0.4673	
512	5.7278e-3	1.1559	2.0387	5.1709e-3	1.2377	1.6909	
(c) $T = 4000s$							
$1/\Delta x$	Basic Scheme			Mollified Scheme with $\eta = 3$			
	$L^1$ -error	conv. rate	CPU time [s]	$L^1$ -error	conv. rate	CPU time [s]	
64	2.5560e-2	–	0.1670	2.2249e-2	–	0.1209	
128	1.5463e-2	0.7251	0.4342	1.3797e-2	0.6894	0.3133	
256	9.7331e-3	0.6679	1.6177	8.8938e-3	0.6335	1.1459	
512	3.7879e-3	1.3615	7.4223	3.3775e-3	1.3968	5.1181	
$1/\Delta x$	Mollified Scheme with $\eta = 5$			Mollified Scheme with $\eta = 8$			
	$L^1$ -error	conv. rate	CPU time [s]	$L^1$ -error	conv. rate	CPU time [s]	
64	1.9460e-2	–	0.1208	1.7743e-2	–	0.1211	
128	1.1946e-2	0.7039	0.3008	1.0622e-2	0.7402	0.2863	
256	7.8624e-3	0.6034	0.9887	7.1371e-3	0.5737	0.8896	
512	2.8738e-3	1.4520	4.0391	2.4282e-3	1.5555	3.3389	

### **4.3 CONCLUDING REMARKS**

1. The stabilizations presented in this chapter are specifically designed for non-linear equations and feature thorough convergence analysis.
2. Higher order mollified methods for hyperbolic conservation laws and convection-diffusion equations are a challenge and we do not know whether they exist or not. We are inclined to think they do exist and would like to develop at least one of them.

## Chapter 5

# SYSTEM IDENTIFICATION

Recent developments in mollification motivate us to consider brief descriptions of current work. The first part is devoted to the identification of distributed parameter systems in nonlinear parabolic equations. It is based on the recent paper Mejía, Acosta and Saleme (2011). The second part includes some encouraging results related to the simultaneous identification of system parameters for a strongly degenerate parabolic equation and follows partially the more recent paper Acosta, Bürger and Mejía (2014).

A variety of analytical and numerical methods for parameter estimation problems in evolution partial differential equations have been proposed in the literature. They are useful to model situations in geophysics, oil recovery and heat conduction to list only a few. Among the general references on the subject, it is worth notice the book edited by Colton, Ewing and Rundell (1990) or the more recent book Vogel (2006).

In the first part of this chapter the task is the identification by mollification of a space dependent ingredient in the thermal conductivity coefficient in a one dimensional nonlinear heat conduction model. Our strategy is to obtain a close formula for the approximation of the unknown ingredient by implementing a combination of discrete mollification and a space-marching finite difference numerical scheme. Furthermore, we introduce several types of coefficients, allow noise in the data and deduce convergence estimates for their respective numerical identifications. The second part deals with the estimation of system parameters in a sedimentation-consolidation model. It is based on work in progress of potential use in the mining industry.

The scope is the numerical identification problem and we do not address theoretical issues related to identifiability of the coefficients. This is the first time in this monograph that discrete mollification is implemented as a regularization tool but this is one of the early uses of the procedure (cf., e.g., Murio, 1993; Mejía and Murio, 1993; Mejía and Murio, 1995) together with the numerical solution of a variety of inverse heat conduction problems (cf., e.g., Murio, 1981; Mejía and Murio, 1996.)

### 5.1 A DIFFUSION COEFFICIENT

Inverse problems come paired with direct problems. In the case of a model identification problem, there are two main approaches to put this relationship to work: the first one is through overposed data in the form of measurements on the evolving process (direct problem). The second one is by an iterative procedure in charge of

updating the direct problem parameters; in each iteration the direct problem is numerically solved at least once. Our approach is the first one and as overdetermined data, we require temperature distribution measurements for two consecutive time steps, not both of them corresponding to steady state.

### 5.1.1 DIRECT PROBLEM

Find a temperature  $u$  that satisfies the nonlinear initial boundary value heat conduction problem

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial}{\partial x} \left( K(x, u) \frac{\partial u}{\partial x} \right) + f(x, t), & 0 < x < 1, \quad 0 < t \\ u(x, 0) &= g(x), & 0 \leq x \leq 1 \\ K(x, u) \frac{\partial u}{\partial x} \Big|_{x=0} &= q_0(t) > 0, & 0 < t \\ u(1, t) &= g_1(t), & 0 < t \end{aligned} \quad (5.1)$$

The diffusion coefficient  $K$  is positive and bounded, and as a function of  $u$ , is monotone. The temperature  $u$  is positive in the interior of the domain and bounded everywhere. The forcing term  $f(x, t)$  is known throughout the domain. Furthermore,  $u_x(x, t)$  is bounded and nonzero for  $0 < x \leq 1$ . From this assumption and the flux condition in (5.1), it is clear that  $u_x(x, t) > 0$  for all  $x$ .

### 5.1.2 INVERSE PROBLEM

Equations (5.1) are given for an unknown diffusion coefficient  $K$  of a particular form. In our case,  $K$  depends on an unknown function  $a(x)$  whose identification is the actual goal. The value of  $a(0)$  is obtained from the flux condition at  $x = 0$ . Examples of diffusion coefficients of interest are:  $K(x, u) = 1 + a(x)u^2$  and  $K(x, u) = a(x) + u^2$ . For both of them we obtain complete error estimates in the next section. As overposed condition for the correct definition of the inverse problem, we have the following: Two measurements  $u^\varepsilon(x, t_1)$  and  $u^\varepsilon(x, t_1 + k)$  are known and they satisfy

$$|u(x, t_1) - u^\varepsilon(x, t_1)| \leq \varepsilon$$

$$|u(x, t_1 + k) - u^\varepsilon(x, t_1 + k)| \leq \varepsilon$$

for  $0 \leq x \leq 1$ ,  $0 < t_1 < t_1 + k$ . At most one of the measurements corresponds to steady state. Here parameter  $k$  is the discretization time step, thus times  $t_1$  and  $t_1 + k$  are consecutive.

## 5.2 NUMERICAL ALGORITHMS

We implement a marching in space finite difference method defined on  $R = [0, 1] \times [0, t_f]$ , where  $t_f$  is the final time. The necessary regularization is provided by discrete mollification with automatic selection of parameters. In order to simulate measurements, we add random noise to the temperature distributions involved in the inverse problem solution process.

The numerical schemes depend on the following parameters:

$\varepsilon$ : Maximum level of noise in the data.  $0 < \varepsilon \ll 1$ .

$nx$ : Number of space divisions of the grid.  $M = nx + 1$ .

$h$ : Space step.  $0 < h \ll 1$ . The space nodes are  $x_j = jh$ , with  $j = 0, 1, \dots, nx + 1$ , so  $1 = hM$ .

$k$ : Time step.  $0 < k \ll 1$ .

$\delta$ : Maximum of the selected mollification parameters.  $0 < \delta \ll 1$ .

The unknown for the numerical method is the vector  $A$ . Its component  $A_j$  for  $j = 1 : 1 : nx + 1$  is the approximation to  $a(x_j)$ . The value  $A_0$  is obtained from the flux at  $x = 0$ . The other variables of the numerical method are the vectors  $U$ ,  $V$ ,  $W$  and  $F$  whose components are:

$$\begin{array}{ll}
 A_j \text{ the coefficient approximation at } x_j & j = 1 : 1 : nx + 1 \\
 U_j = J_{\delta\eta} u^\varepsilon(x_j, t_1) & j = 0 : 1 : nx + 1 \\
 V_0 = \frac{1}{h}(U_1 - U_0) & \\
 U_{j \pm \frac{1}{2}} = J_{\delta\eta} u^\varepsilon(x_j \pm \frac{h}{2}, t_1) & j = 1 : 1 : nx \\
 V_j = \frac{1}{h}(J_{\delta\eta} u^\varepsilon(x_j + \frac{h}{2}, t_1) - J_{\delta\eta} u^\varepsilon(x_j - \frac{h}{2}, t_1)) & j = 1 : 1 : nx \\
 W_j = \frac{1}{k}(J_{\delta\eta} u^\varepsilon(x_j, t_1 + k) - J_{\delta\eta} u^\varepsilon(x_j, t_1)) & j = 0 : 1 : nx \\
 F_j = f(x_j, t_1) & j = 0 : 1 : nx
 \end{array}$$

Likewise, we define the vectors  $a$ ,  $u$ ,  $v$  and  $w$  whose components  $a_j$ ,  $u_j$ ,  $v_j$  and  $w_j$  correspond to  $a(x_j)$ ,  $u(x_j, t_1)$ ,  $u_x(x_j, t_1)$  and  $u_t(x_j, t_1)$  respectively for  $j = 0 : 1 : nx + 1$ .

The following estimates are important in the sequel. The corresponding to derivatives include bounds that require some kind of linking among the parameters in order to be called error estimates. This is due to the ill-posedness of differentiation. A possible way to link parameters is  $h = O(k) = O(\varepsilon^{\frac{1}{2}})$ .

**Lemma 25.** *Assume hypothesis of theorem 1 holds. Then for each compact subset  $K = [a, b]$  there is a constant  $C = C(K)$  such that if  $x_j \in K$  then*

$$\begin{aligned}
 |U_j - u_j| &\leq C(\varepsilon + h + \delta) \\
 |V_j - v_j| &\leq C\left(\frac{\varepsilon}{h} + h + \delta\right) \\
 |W_j - w_j| &\leq C\left(k + \frac{\varepsilon}{k} + h + \delta\right)
 \end{aligned}$$

*Proof.* A complete proof can be found in Mejia, Acosta and Saleme (2011).  $\square$

According to section 1.2,  $h = O(\delta)$ , thus, the ratios  $\varepsilon/h$  and  $\varepsilon/k$ , taking into consideration the note before the lemma, may be considered  $\varepsilon/\delta$ . The situation is typical of ill-conditioned problems, the regularization parameter depends on the level of noise in the data.

### 5.2.1 COEFFICIENT $K(x, u) = 1 + a(x)u^2$

Forward finite difference discretization of the first equation of (5.1) yields

$$a_{j+1} = \frac{1}{u_{j+1}^2 v_{j+1}} [-v_{j+1} + (1 + a_j u_j^2) v_j + h(w_j - F_j)] + O(h) \quad (5.2)$$

Thus, the space-marching finite difference scheme is:

1. Let  $A_0 = \frac{(q_0(t_1) - V_0)}{U_0^2 V_0}$ .
2. For  $j = 0 : 1 : nx$ , compute

$$A_{j+1} = \frac{1}{U_{j+1}^2 V_{j+1}} [-V_{j+1} + (1 + A_j U_j^2) V_j + h(W_j - F_j)]. \quad (5.3)$$

The convergence result is the following:

**Theorem 26.** *If hypotheses on  $K$ ,  $u$  and  $u_x$  presented in section 5.1 are satisfied, then vector  $A$  defined above is so that  $|a_j - A_j| \rightarrow 0$  whenever  $h, k, \varepsilon \rightarrow 0$  and are linked in a certain way.*

*Proof.* We omit the proof. If a theoretical convergence result is desired, a link between  $h, k$  and  $\varepsilon$  is in order, for instance,  $h = O(\varepsilon^{\frac{1}{2}})$  and  $k = O(h)$ .  $\square$

### 5.2.2 COEFFICIENT $K(x, u) = a(x) + u^2$

The same strategy can be implemented for this inverse problem. From the first equation of (5.1) we obtain

$$a_{j+1} = \frac{1}{v_{j+1}} [-u_{j+1}^2 v_{j+1} + (a_j + u_j^2) v_j + h(w_j - F_j)] + O(h). \quad (5.4)$$

The marching in space finite difference scheme for the solution of the identification problem is:

1. Let  $A_0 = \frac{(q_0(t_1) - U_0^2 V_0)}{V_0}$ .

2. For  $j = 1 : 1 : nx$ , compute

$$A_{j+1} = \frac{1}{V_{j+1}} [-U_{j+1}^2 V_{j+1} + (A_j + U_j^2) V_j + h(W_j - F_j)]. \quad (5.5)$$

The convergence of the numerical scheme is summarized in the following theorem.

**Theorem 27.** *If hypotheses on  $K$ ,  $u$  and  $u_x$  of section 5.1 hold, then the vector  $A$  defined above is so that  $|a_j - A_j| \rightarrow 0$  whenever  $h, k, \varepsilon \rightarrow 0$  and are linked in a certain way.*

*Proof.* We omit the proof. The linking of parameters, in the same way as in the previous theorem, is also appropriate here.  $\square$

**Remark 28.** *Some comments are in order now.*

1. *Right-hand side expressions with  $h$  or  $k$  in denominators are clear indications of the ill-posedness of the identification problem, which is due in part to the numerical approximation of derivatives. They can be seen in the complete proofs presented in Mejía, Acosta and Saleme (2011).*
2. *The strategy described in this section is appropriate for the numerical identification of a variety of nonlinear coefficients depending on a space dependent function  $a(x)$ . In Saleme (2009) slightly more general coefficients are considered. They are*

$$K(x, t, u) = a(x)p(t) + b(x)u^2(x, t)$$

and

$$K(x, t, u) = b(x) + a(x)p(t)u^2(x, t)$$

where  $b$  and  $p$  are known positive functions of  $x$  and  $t$  respectively. For computations, noisy versions of both are available. Since the proofs follow the same ideas of the previous theorems, we omit them here.

3. *The exponent 2 of  $u$  can be replaced by any other positive integer.*

### 5.3 NUMERICAL EXPERIMENTS

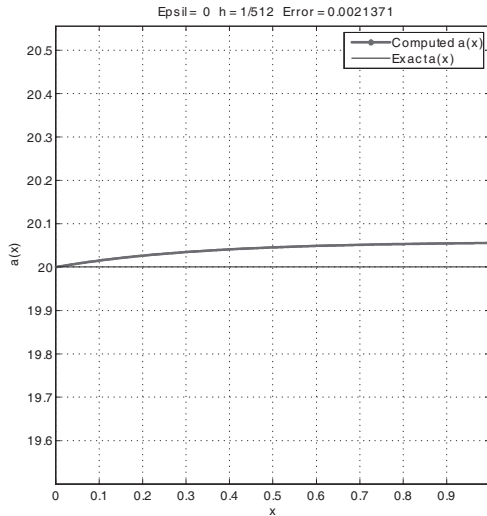
In order to obtain error estimates, exact distributions for  $a(x)$ ,  $u(x)$  and  $f(x, t)$  are known for each example. The  $l_2$  discrete norm is the default norm in the experiments. In all cases measurements are simulated by adding normally distributed random noise to temperature distributions. The maximum level of noise in the data is not taken into consideration for the selection of regularization parameters. It is useful only as a way to simulate measured temperatures. The mollification parameter selection is automatic and is based on Generalized Cross Validation (GCV).

**Example 29.** This experiment consists on the identification of  $a(x)$  in the nonlinear diffusion coefficient  $K(x, u) = 1 + a(x)u^2$ . The system of equations defining the inverse problem is

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial}{\partial x} \left( K(x, u) \frac{\partial u}{\partial x} \right) + f(x, t), & 0 < x < 1, \quad 0 < t \\ u(x, 0) &= \exp(x), & 0 \leq x \leq 1 \\ K(x, u) \frac{\partial u}{\partial x} \Big|_{x=0} &= (1 + 20 \exp(2t)) \exp(t), & 0 < t \\ u(1, t) &= \exp(1+t) & 0 < t \\ f(x, t) &= -60 \exp(3(x+t)) \end{aligned}$$

The exact data are  $a(x) = 20$  and  $u(x, t) = \exp(x+t)$ .

The first figure reports the consistency of the numerical method. No noise is added and no mollification is implemented and, as indicated by figure 5.1, the scheme by itself delivers a nice approximation.



**Figure 5.1.** No noisy data and no mollification

The second figure addresses a basic fact: If data is noisy, some regularization should be implemented. Figure 5.2 illustrates this fact. The poor performance of the same identification algorithm is due to lack of regularization.

Since random noise is involved in computations, it is appropriate to run the identification routine a sufficient number of times in order to detect a pattern of convergence. Two series of 100 routine runs are presented in figures 5.3 and 5.4. The level of noise is  $\varepsilon = 0.01$  in both experiments.

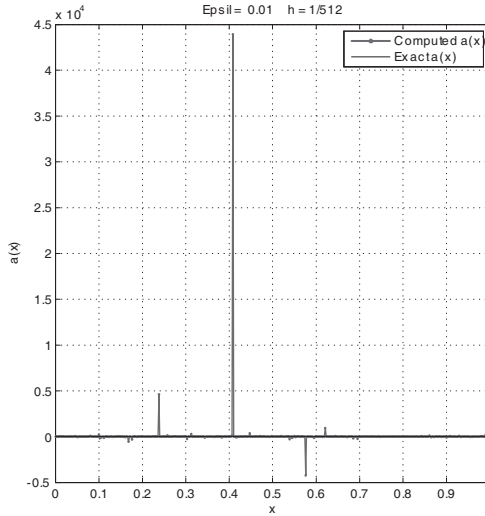


Figure 5.2. Noisy data and no mollification

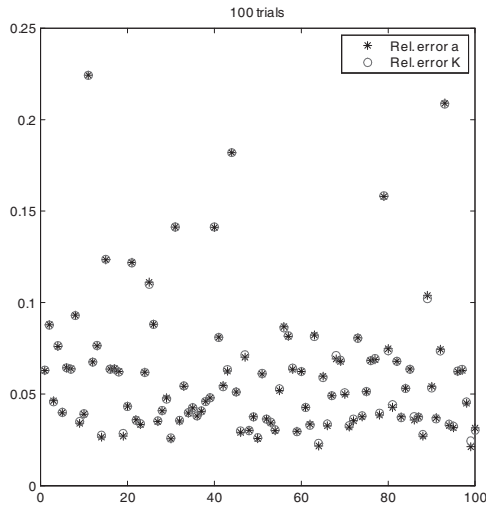


Figure 5.3. Reliability test, first part,  $h = k = 2^{-6}$

**Example 30.** We identify coefficient  $K(x, u) = a(x) + u^2$ , with  $a(x) = 1 + x^2$ . The diffusion equation is

$$\begin{aligned}
 \frac{\partial u}{\partial t} &= \frac{\partial}{\partial x} \left( K(x, u) \frac{\partial u}{\partial x} \right) + f(x, t), & 0 < x < 1, \quad 0 < t \\
 u(x, 0) &= \exp(x), & 0 \leq x \leq 1 \\
 K(x, u) \frac{\partial u}{\partial x} \Big|_{x=0} &= (1 + \exp(2t)) \exp(t), & 0 < t \\
 u(1, t) &= \exp(1 + t) & 0 < t \\
 f(x, t) &= -(x^2 + 2x) \exp(x + t) - 3 \exp(3(x + t))
 \end{aligned}$$

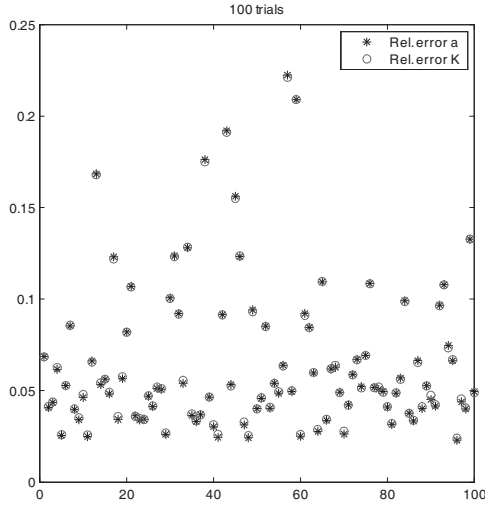


Figure 5.4. Reliability test, second part,  $h = 2^{-6}$ ,  $k = 2^{-8}$

The exact ingredient is  $a(x) = 1 + x^2$ . Once again, as a reliability test, we implement two 100 run trials with maximum level of noise in the data  $\varepsilon = 0.01$ . They appear in figures 5.5 and 5.6.

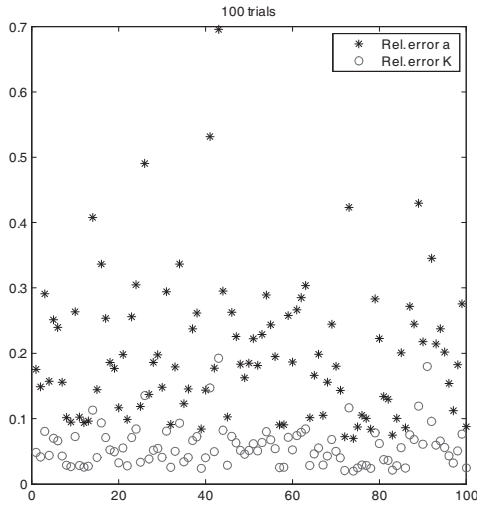


Figure 5.5. First trial of 100 runs with parameters  $h = k = 2^{-6}$

**Example 31.** The third example is a mixture of the previous two. It consists on the identification of  $a(x)$  in the nonlinear diffusion coefficient  $K(x, u) = 1 + a(x)u^2$ .

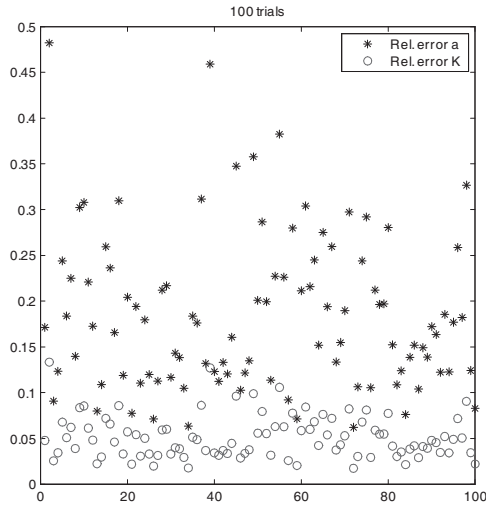


Figure 5.6. Second trial of 100 runs with parameters  $h = 2^{-6}$  and  $k = 2^{-7}$

The system of equations defining the identification problem is

$$\begin{aligned} \frac{\partial u}{\partial t} &= \frac{\partial}{\partial x} \left( K(x, u) \frac{\partial u}{\partial x} \right) + f(x, t), & 0 < x < 1, \quad 0 < t \\ u(x, 0) &= \exp(x), & 0 \leq x \leq 1 \\ K(x, u) \frac{\partial u}{\partial x} \Big|_{x=0} &= (1 + \exp(2t)) \exp(t), & 0 < t \\ u(1, t) &= \exp(1 + t) & 0 < t \\ f(x, t) &= -(3 + 2x + 3x^2) \exp(3(x + t)) \end{aligned}$$

The exact data are  $a(x) = 1 + x^2$  and  $u(x, t) = \exp(x + t)$ . A reliability test, consisting on two trials with 100 routine runs each, is presented in figures 5.7 and 5.8. The maximum level of noise in data is  $\varepsilon = 0.01$  and the average values of delta for each set of runs are 0.0658 and 0.0667 respectively. These numbers were obtained by collecting the delta values of each routine run and taking the average after the 100 runs were completed.

A summary of the experiments is presented in tables 5.1, 5.2 and 5.3, one for each example. They report computations for a variety of combinations of parameters. There is only one routine run for each combination of parameters and for each example. Some basic and important facts related with scientific computing are easy to see in the tables, namely:

1. In the absence of noise, results improve whenever a refinement in discretization parameters is made. Concurrently, such a refinement causes GCV to choose smaller values for delta. Of course this can not go forever, extremely small discretization parameters may contribute to deteriorate numerical solutions.

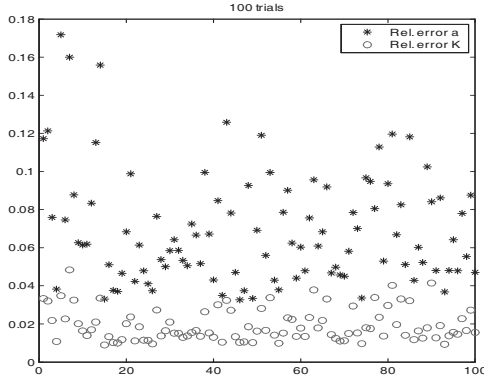


Figure 5.7. Reliability test,  $h = k = 2^{-6}$

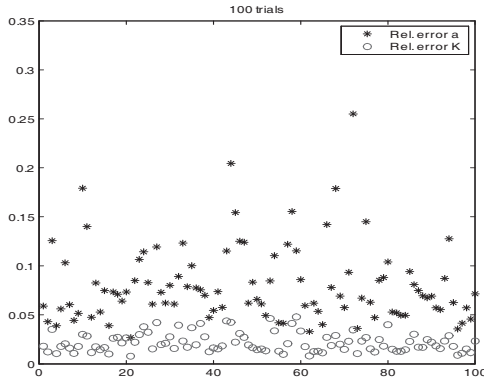


Figure 5.8. Second trial of 100 runs, parameters  $h = 2^{-6}$  and  $k = 2^{-7}$

- When noisy data is present, the ideal expected behavior of parameters and errors is:

$h, k$	Average $\delta$	Error
↘	The right amount	↘

However, many factors intervene and this is not always the case. Actually, a picture of  $f(h) = h + \frac{\epsilon}{h}$  is illustrative now (see figure 5.9.) Only some values of  $h$  are appropriate to reduce the error. Again, it is important to recall that we are dealing with random noise and with an automatic procedure for the selection of mollification parameters. Some times, GCV does not select optimal values. Fortunately, this happens only occasionally, as can be seen from the reliability tests presented above. Finally, the numerical algorithm

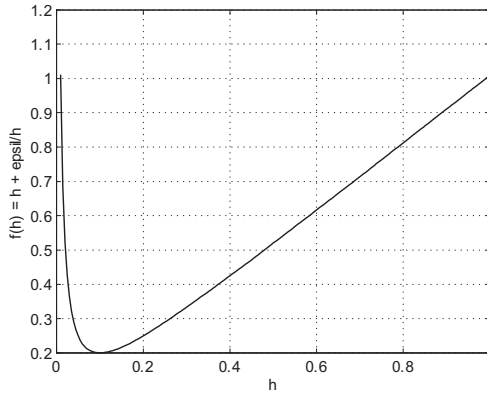


Figure 5.9. Error term vs  $h$

presented here is only first order accurate, so, moderate errors should be expected.

Table 5.1. Example 29: incidence of random noise and discretization parameters in identification

Relative errors in $a(x)$ and $K(x,u)$					
$\varepsilon$	$h$	$k$	Avg $\delta$	r. e. $a(x)$	r. e. $K(x,u)$
0.00	$2^{-5}$	$2^{-5}$	0.0210	0.0336	0.0335
	$2^{-6}$	$2^{-6}$	0.0118	0.0170	0.0169
	$2^{-8}$	$2^{-8}$	0.0030	0.0043	0.0043
	$2^{-9}$	$2^{-9}$	0.0014	0.0021	0.0021
0.01	$2^{-5}$	$2^{-6}$	0.0773	0.1407	0.1389
	$2^{-6}$	$2^{-8}$	0.0695	0.0325	0.0327
	$2^{-9}$	$2^{-9}$	0.0257	0.0618	0.0614
	$2^{-9}$	$2^{-10}$	0.0334	0.0441	0.0449
0.04	$2^{-6}$	$2^{-6}$	0.0710	0.1135	0.1143
	$2^{-6}$	$2^{-8}$	0.0798	0.0804	0.0811
	$2^{-7}$	$2^{-9}$	0.0778	0.1051	0.1046

## 5.4 STRONGLY DEGENERATE PARABOLIC EQUATIONS

We are interested in implementing the mollification method for the numerical identification of unknown parameters appearing on the flux and diffusion terms of initial boundary value problems (IBVP) for strongly degenerate parabolic equations in one space dimension.

**Table 5.2.** Example 30: Incidence of random noise and discretization parameters in identification

Relative errors in $a(x)$ and $K(x, u)$					
$\varepsilon$	$h$	$k$	Avg $\delta$	r. e. $a(x)$	r. e. $K(x, u)$
0.00	$2^{-5}$	$2^{-5}$	0.0259	0.1260	0.0356
	$2^{-7}$	$2^{-7}$	0.0077	0.0336	0.0092
	$2^{-8}$	$2^{-8}$	0.0041	0.0176	0.0048
0.01	$2^{-7}$	$2^{-7}$	0.0500	0.2672	0.0732
	$2^{-7}$	$2^{-8}$	0.0546	0.1293	0.0350
	$2^{-8}$	$2^{-8}$	0.0445	0.0602	0.0165
0.04	$2^{-6}$	$2^{-8}$	0.1067	0.2090	0.0570
	$2^{-7}$	$2^{-7}$	0.0834	0.1602	0.0442

**Table 5.3.** Example 31: incidence of random noise and discretization parameters in identification

Relative errors in $a(x)$ and $K(x, u)$					
$\varepsilon$	$h$	$k$	Avg $\delta$	r. e. $a(x)$	r. e. $K(x, u)$
0.00	$2^{-6}$	$2^{-6}$	0.0200	0.0330	0.0074
	$2^{-7}$	$2^{-7}$	0.0044	0.0132	0.0034
	$2^{-8}$	$2^{-8}$	0.0030	0.0066	0.0017
0.01	$2^{-6}$	$2^{-6}$	0.0761	0.0767	0.0159
	$2^{-6}$	$2^{-8}$	0.0767	0.0480	0.0108
	$2^{-8}$	$2^{-8}$	0.0414	0.0361	0.0087
0.04	$2^{-6}$	$2^{-8}$	0.1002	0.1271	0.0276
	$2^{-6}$	$2^{-6}$	0.0889	0.1132	0.0293

The strongly degenerate parabolic equations are useful as models for many applications, e.g. sedimentation-consolidation processes of solid-liquid suspensions and traffic flow. For the numerical solution of the equations (direct problem), the discrete mollification method has been successfully implemented in Acosta, Bürger and Mejía (2012), where an explicit finite difference scheme proposed by Evje and Karlsen (2000), is mollified in such a way that monotonicity and convergence are preserved and there are gains in CFL condition and CPU time. A two dimensional generalization of the method in Acosta, Bürger and Mejía (2012) has just been obtained (Acosta and Bürger, 2012) but it is not considered here.

Inverse problems for strongly degenerate parabolic equations are of current interest. In Berres, Bürger and Garcés (2010), the authors introduce a numerical scheme for the simultaneous identification of several parameters and include very well designed experiments to illustrate the quality of the recovery. Likewise reference Bürger, Coronel and Sepúlveda (2009) is focused on the numerical identification of unknown ingredients with the important feature of allowing noisy data in one of the examples.

References Berres *et al.* (2005) and Coronel, James and Sepúveda (2003) address the theoretical issue of identifiability and prove the existence of solutions for the identification problem. Furthermore, they include numerical schemes and a well selected set of numerically solved experiments, which include either noisy or simulated data.

Our scope is the numerical identification problem and we do not address theoretical issues related to identifiability of the coefficients. As it is the case in the references cited above, the identification problem is formulated through the minimization of a suitable cost function. For the numerical solution of the direct problem, we implement the schemes described above in subsection 4.2.1. The numerical experiments illustrate the quality of identifications even in the presence of noisy data.

## 5.5 PARAMETER IDENTIFICATION

One of the main applications of strongly degenerate parabolic equations is a model of sedimentation-consolidation processes of solid-liquid suspensions (see e.g. Berres *et al.* 2003) where the flux function  $f(u)$  and the diffusivity function  $A(u)$  model the effects of hindered settling and sediment compressibility, respectively, of a suspension of local solids volume fraction  $u$ .

### Direct Problem

A set of equations for the sedimentation-consolidation process is given (Bürger and Karlsen, 2001) by

$$\begin{cases} u_t + f(u)_x = A(u)_{xx}, & (x, t) \in \Pi_T := (0, L) \times (0, T), T > 0 \\ u(x, 0) = u_0(x), & x \in [0, L], \\ f(u) - A(u)_x = 0, & x \in \{0, L\}. \end{cases} \quad (5.6)$$

Here the hindered settling flux  $f(u)$  has the form

$$f(u) = \begin{cases} v_\infty u \left(1 - \frac{u}{u_{\max}}\right)^C, & \text{for } 0 < u < u_{\max}, \\ 0, & \text{otherwise,} \end{cases} \quad (5.7)$$

with parameters  $v_\infty < 0$  and  $C > 1$  and the sediment compressibility  $A(u)$  is defined by

$$A(u) = \int_0^u a(s) ds, \text{ where } a(u) = \frac{f(u)\sigma'_e(u)}{\Delta\rho g u} \quad (5.8)$$

with the function  $\sigma_e$  given by the power-law function

$$\sigma_e(u) = \begin{cases} 0, & \text{for } 0 < u \leq u_c, \\ \sigma_0 \left[ (u/u_c)^\beta - 1 \right], & \text{for } u > u_c, \end{cases} \quad \sigma_0 > 0, \beta > 1. \quad (5.9)$$

The values  $\Delta_\rho$  and  $g$  are constant, meanwhile  $\beta$ ,  $\sigma_0$  and  $u_c$  (the so-called critical concentration) are parameters. If  $\beta$  is an integer, then  $A(u)$  can be evaluated in closed form as follows:

$$A(u) = \begin{cases} 0, & \text{for } 0 < u \leq u_c, \\ \mathcal{A}(u) - \mathcal{A}(u_c), & \text{for } u > u_c, \end{cases} \quad (5.10)$$

where

$$\mathcal{A}(u) := \frac{v_\infty \sigma_0}{\Delta_\rho g u_c^\beta} \frac{1}{u_{\max}^C} \sum_{k=1}^{\beta} \left( \prod_{l=1}^k \frac{\beta+1-l}{C+l} \right) (u_{\max} - u)^{C+k} u^{\beta-k}. \quad (5.11)$$

For the numerical solution of (5.6) we consider the two convergent finite difference methods introduced in subsection 4.2.1 that we rewrite for completeness. The first is the *mollified* monotone difference scheme

$$u_j^{n+1} = u_j^n - \lambda \Delta_+ F^{EO}(u_{j-1}^n, u_j^n) + 2\mu C_\eta \left( [J_\eta A(u^n)]_j - A(u_j^n) \right) \quad (5.12)$$

whose CFL condition is

$$\lambda \|f'\|_\infty + 2\mu \varepsilon_\eta \|a\|_\infty \leq 1, \quad \text{where } \varepsilon_\eta := C_\eta(1 - w_0). \quad (5.13)$$

The second scheme is taken as a reference scheme for comparison purposes. We call it the *basic* scheme and has the form

$$u_j^{n+1} = u_j^n - \lambda \Delta_+ F^{EO}(u_{j-1}^n, u_j^n) + \mu \left( A(u_{j+1}^n) - 2A(u_j^n) + A(u_j^n) \right). \quad (5.14)$$

The basic scheme is monotone and convergent under the CFL condition

$$\lambda \|f'\|_\infty + 2\mu \|a\|_\infty \leq 1. \quad (5.15)$$

### Inverse Problem

Given an observation data  $u^{obs}(x)$ , at a time  $T > 0$ , and the function  $u_0$  explicitly, find the flux  $f$  and the diffusion function  $a$  of the form (5.7),(5.8),(5.9) such that the entropy solution  $u(x, T)$ , at time  $T$  of the problem (5.6), is as close as possible to  $u^{obs}(x)$  in some suitable norm.

Solutions of the initial boundary value problem 5.6 are, in general, discontinuous even if  $u_0$  is smooth, and need to be defined as weak solutions along with an entropy condition to select the physically relevant solution, the *entropy solution*. For the

definition, existence and uniqueness of entropy solutions of problem (5.6), see the considerations related to Problem B in Bürger, Evje and Karlsen (2000).

The inverse problem consists on the minimization of a suitable cost function  $\mathcal{Q}$ ; that is,

$$\min_{f,a} \mathcal{Q}(u(\cdot, T)), \quad (5.16)$$

where  $u$  is the entropy solution to (5.6), when working with flux  $f$  and diffusion function  $a$ . A particular choice for the cost function  $\mathcal{Q}$  is

$$\mathcal{Q}(u(\cdot, T)) = \frac{1}{2} \int_0^L |u(x, T) - u^{obs}(x)|^2 dx. \quad (5.17)$$

Since functions  $f$  and  $a$  are characterized by the parameters  $v_\infty, C, \beta, \sigma_0$  and  $u_c$ , the optimization process can be based on choosing suitable values for them. So, instead of the minimization problem (5.16), we state

$$\min_p \mathcal{Q}(u(\cdot, T)), \quad (5.18)$$

where  $p$  is the vector of unknown parameters and  $u$  is the entropy solution of (5.6) found with the functions  $f$  and  $a$  associated to the current values of  $p$ .

## 5.6 DISCRETIZATION OF THE INVERSE PROBLEM

The discrete setting is

$$\min_p \mathcal{Q}(p) \text{ with } \mathcal{Q}(p) = \sum_j \left( u_j^T - u^{obs}(x_j) \right)^2. \quad (5.19)$$

Here  $u_j^T$  is the numerical solution of (5.6) computed with one of the schemes (5.12) or (5.14) at instant  $T$  and at space grid point  $x_j$ . Functions  $f$  and  $a$  take parameter values from the current parameter vector  $p$ . So, each evaluation of the cost function  $\mathcal{Q}$  implies a numerical solution of the direct problem (5.6).

The optimization process of (5.19) is carried out by a restarted version of the Nelder-Mead simplex method (`fminsearch` in `MATLAB`). This is a derivative-free optimization method that is widely used by researchers in different fields, is very well documented and, nevertheless, has some serious drawbacks. For instance, reference reference Lagarias *et al.* (1998) shows that the method can fail to converge or converge to nonsolutions on certain classes of problems. Due to the lack of convergence in some cases, many modifications have been proposed (e.g. Price, Coope and Byatt, 2002; Kelley, 1999; Luersen and Riche, 2004 and Zhao *et al.*, 2009).

The resulting algorithm, based on Zhao *et al.* (2009), takes the form

1. Input  $p_0, \varepsilon$

2. for  $j = 1 : M$ 
  - (a)  $p_j = \text{fminsearch}(\mathcal{Q}, p_{j-1})$
  - (b) if  $\|(p_j - p_{j-1}) ./ p_{j-1}\|_\infty \leq \varepsilon$ , break, end
3. end

### 5.7 NUMERICAL EXPERIMENTS

In each case, the reference solution is generated with the corresponding numerical scheme (5.12) or (5.14) on a very fine grid at the instant  $T = 800s$ . The necessary constants are  $L = 0.16m, u_{\max} = 0.5, \Delta\rho = 1660kg/m^3$  and  $g = 9.81m/s^2$ . The parameters were set to  $v_\infty = -2.7 \times 10^{-4}m/s, C = 21.5, \beta = 5, u_c = 0.07$  and  $\sigma_0 = 1.2Pa$ . The restarting parameter for the optimization were  $M = 10$  and  $\varepsilon = 10^{-4}$ .

The goal is to obtain the numerical identification of the parameters  $u_c, \sigma_0$  and  $C$ . The experiments include clean and noisy observation data.

**Example 32.** Working with clean observation data and  $h = L/256$ . The results are summarized on tables 5.4 through 5.8. Notice that  $j$  stands for the number of calls to the *fminsearch* algorithm,  $P_j$  is the identified parameter values,  $J_v$  the required number of computed solutions of the direct problem and  $RE_\infty$  is the maximum relative error in the result for each parameter (usually due to  $\sigma_0$ ).

**Table 5.4.** Initial guesses in example 32

IG	Initial Guess	IG	Initial Guess
A:	[ 0.7 $u_c$ , 0.7 $\sigma_0$ , 0.7 $C$ ]	E:	[ 1.3 $u_c$ , 0.7 $\sigma_0$ , 0.7 $C$ ]
B:	[ 0.7 $u_c$ , 0.7 $\sigma_0$ , 1.3 $C$ ]	F:	[ 1.3 $u_c$ , 0.7 $\sigma_0$ , 1.3 $C$ ]
C:	[ 0.7 $u_c$ , 1.3 $\sigma_0$ , 0.7 $C$ ]	G:	[ 1.3 $u_c$ , 1.3 $\sigma_0$ , 0.7 $C$ ]
D:	[ 0.7 $u_c$ , 1.3 $\sigma_0$ , 1.3 $C$ ]	H:	[ 1.3 $u_c$ , 1.3 $\sigma_0$ , 1.3 $C$ ]

**Example 33.** This example checks the sensitivity of the procedure to the choice of the initial guess. For such a purpose we randomly generate 100 initial guesses and carry out the identification task. Each initial guess is generated in the form

$$\begin{aligned}
 u_c^0 &= (1 + 0.3\xi_1)u_c \\
 \sigma_0^0 &= (1 + 0.3\xi_2)\sigma_0 \\
 C^0 &= (1 + 0.3\xi_3)C
 \end{aligned}$$

where  $\xi$  is a random vectorial variable on  $\mathbb{R}^3$  uniformly distributed with each entry between  $-1$  and  $1$ . The results are indicated on table 5.9, which includes the average of  $RE_\infty$  and its standard deviation. Additionally, the Restart column stands

**Table 5.5.** Numerical identification without mollification in example 32

IG	$j$	$P_j$ (Basic)	$J_V$	$RE_\infty$	time[s]
A	2	[ 0.0697, 1.1219, 21.4706]	290	0.0651	79.863
B	4	[ 0.0696, 1.1111, 21.4700]	517	0.0741	90.310
C	5	[ 0.0697, 1.1324, 21.4700]	616	0.0564	163.74
D	3	[ 0.0696, 1.1252, 21.4699]	669	0.0623	127.96
E	3	[ 0.0696, 1.1179, 21.4700]	367	0.0684	59.662
F	6	[ 0.0696, 1.1117, 21.4700]	674	0.0736	108.77
G	3	[ 0.0696, 1.1114, 21.4700]	421	0.0738	71.391
H	2	[ 0.0696, 1.1180, 21.4700]	391	0.0684	64.747

**Table 5.6.** Numerical identification for  $\eta = 3$  in example 32

IG	$j$	$P_j$ ( $\eta = 3$ )	$J_V$	$RE_\infty$	time[s]
A	2	[ 0.0695, 1.1104, 21.5067]	356	0.0746	89.43
B	4	[ 0.0695, 1.1105, 21.5067]	490	0.0746	89.77
C	2	[ 0.0695, 1.1104, 21.5067]	296	0.0747	103.1
D	5	[ 0.0695, 1.1103, 21.5067]	797	0.0747	155.8
E	3	[ 0.0696, 1.1259, 21.5075]	332	0.0617	58.19
F	3	[ 0.0696, 1.1294, 21.5066]	358	0.0588	61.25
G	3	[ 0.0695, 1.1104, 21.5068]	402	0.0747	73.08
H	5	[ 0.0695, 1.1104, 21.5065]	676	0.0747	119.7

**Table 5.7.** Numerical identification for  $\eta = 5$  in example 32

IG	$j$	$P_j$ ( $\eta = 5$ )	$J_V$	$RE_\infty$	time[s]
A	5	[ 0.0500, 0.1486, 21.5456]	685	0.8762	152.8
B	5	[ 0.0697, 1.1651, 21.5465]	664	0.0291	105.8
C	3	[ 0.0695, 1.1124, 21.5466]	486	0.0730	103.6
D	3	[ 0.0696, 1.1301, 21.5465]	639	0.0582	107.1
E	3	[ 0.0696, 1.1301, 21.5466]	445	0.0583	69.26
F	6	[ 0.0695, 1.0945, 21.5466]	697	0.0879	107.7
G	3	[ 0.0696, 1.1299, 21.5466]	485	0.0584	78.23
H	4	[ 0.0696, 1.1472, 21.5465]	807	0.0440	128.4

for the number of calls to *fminsearch* and  $J_V$  for the number of solutions of the direct problem. The corresponding total time for the 100 identifications is also indicated.

**Example 34.** Now we study the effect of working with noisy observation data. We randomly generate 100 final profiles and associate them to the previously generated initial guesses. The corrupted profile is generated as follows

$$u_j^\varepsilon = (1 + \varepsilon\varphi_j) u^{obs}(x_j),$$

**Table 5.8.** Numerical identification for  $\eta = 8$  in example 32

IG	$j$	$P_j(\eta = 8)$	$Jv$	$RE_\infty$	$time[s]$
A	2	[ 0.0696, 1.1174, 21.5777]	293	0.0688	49.008
B	3	[ 0.0697, 1.1530, 21.5776]	461	0.0391	64.792
C	4	[ 0.0697, 1.1531, 21.5776]	649	0.0391	107.35
D	2	[ 0.0697, 1.1530, 21.5777]	418	0.0391	60.759
E	3	[ 0.0697, 1.1531, 21.5776]	535	0.0391	74.026
F	3	[ 0.0696, 1.1174, 21.5777]	543	0.0688	73.858
G	4	[ 0.0697, 1.1531, 21.5776]	576	0.0391	80.604
H	2	[ 0.0697, 1.1531, 21.5776]	640	0.0391	89.225

**Table 5.9.** Identification results for example 33

$h = L/128$					
<i>Scheme</i>	<i>Restart</i>	$Jv$	$\overline{RelErr}_\infty \pm std$	<i>time[min]</i>	
<i>Basic</i>	387	53693	0.1400±0.0329	49.17	
<i>Mollified, <math>\eta = 3</math></i>	340	49274	0.1295±0.0338	53.92	
<i>Mollified, <math>\eta = 5</math></i>	333	50659	0.0909±0.0315	50.91	
<i>Mollified, <math>\eta = 8</math></i>	276	41263	0.0534±3.7182e-05	39.36	
$h = L/256$					
<i>Basic</i>	388	48842	0.0696±0.0096	136.03	
<i>Mollified, <math>\eta = 3</math></i>	398	53513	0.0739±0.0081	160.74	
<i>Mollified, <math>\eta = 5</math></i>	355	48280	0.0587±0.0174	125.46	
<i>Mollified, <math>\eta = 8</math></i>	326	48392	0.0431±0.0190	114.35	
$h = L/512$					
<i>Basic</i>	334	38879	0.0312±0.0038	440.91	
<i>Mollified, <math>\eta = 3</math></i>	352	40416	0.0332±0.0049	439.55	
<i>Mollified, <math>\eta = 5</math></i>	346	41128	0.0278±0.0057	352.47	
<i>Mollified, <math>\eta = 8</math></i>	374	47250	0.0195±0.0072	335.24	

where  $\varepsilon = 0.01$  and  $\varphi_j$  is a uniformly distributed random variable between  $-1$  and  $1$ . The results are presented in table 5.10.

### 5.8 CONCLUDING REMARKS

1. The combination of discrete mollification and explicit space-marching finite difference schemes is a suitable technique for the identification of some kinds of nonlinear diffusion coefficients in initial boundary value problems (IBVP) of heat conduction. The procedure is easy to implement and generalizes known algorithms for linear models.

**Table 5.10.** Identification results for example 34

$h = L/128$				
<i>Scheme</i>	<i>Restart</i>	<i>J<sub>v</sub></i>	$\overline{RelErr}_\infty \pm std$	<i>time[<i>min</i>]</i>
<i>Basic</i>	359	50263	0.1360±0.0476	46.67
<i>Mollified, <math>\eta = 3</math></i>	348	51086	0.1269±0.0469	55.57
<i>Mollified, <math>\eta = 5</math></i>	327	47515	0.0936±0.0428	48.04
<i>Mollified, <math>\eta = 8</math></i>	253	38949	0.0745±0.0474	37.13
$h = L/256$				
<i>Basic</i>	385	50310	0.0661±0.0264	140.63
<i>Mollified, <math>\eta = 3</math></i>	367	50489	0.0674±0.0238	152.60
<i>Mollified, <math>\eta = 5</math></i>	368	52214	0.053±0.0256	135.85
<i>Mollified, <math>\eta = 8</math></i>	331	47153	0.0401±0.0260	110.56
$h = L/512$				
<i>Basic</i>	346	39464	0.0308±0.0103	452.16
<i>Mollified, <math>\eta = 3</math></i>	340	39954	0.0334±0.0166	436.76
<i>Mollified, <math>\eta = 5</math></i>	366	43215	0.0274±0.0150	367.66
<i>Mollified, <math>\eta = 8</math></i>	384	48706	0.0217±0.0141	346.08

2. The same strategy might not work in the case of several unknowns. Instead, a procedure based on the minimization of a cost functional seems to be the right methodology. In section 5.4 the feasibility of the procedure was shown, based on optimization for the simultaneous identification of parameters in a challenging, strongly degenerate parabolic equation.
3. For the simulation of experimental conditions, we add random noise to the data, but the maximum level of noise in the data, denoted  $\varepsilon$ , is not a parameter considered in computations. In particular, calls to GCV do not depend on the knowledge of this information.
4. What we consider in this chapter is a schematic view of our more recent research and does not include a further advance that is the subject of current work. The idea is to recover a nonlinear diffusion coefficient  $K(u)$  without making any use of a space dependent ingredient as in the first part of this chapter.



# Chapter 6

## LITERATURE REVIEW

In the previous chapters we presented with reasonable detail most of our original contributions to discrete mollification. In this chapter we introduce two more original contributions along with some others developed by Murio and his collaborators but not included in the first two overviews of mollification Murio (1993) and Murio (2002). The topics of this review are:

1. System identification:
  - (a) Source terms identification in 1-D (Yi and Murio, 2004b) and 2-D (Yi and Murio, 2004a) inverse heat conduction problems (IHCP).
  - (b) Simultaneous identification of population density and diffusion coefficient in a linear transport equation with advection (Coles, 2002).
  - (c) Simultaneous identification of parameters for a drying system in a porous medium (Coles and Murio, 2006)
2. Fractional calculus:
  - (a) Stable evaluation of Caputo fractional derivatives (Murio, 2006).
  - (b) Source term identification in a time fractional diffusion equation (TFDE) (Murio and Mejía, 2008).
3. Multiscale analysis for ECG noise reduction (Pulgarín *et al.* 2009).

### 6.1 SYSTEM IDENTIFICATION

#### 6.1.1 SOURCE TERMS IN A 1-D IHCP

In Yi and Murio, (2004b), the authors propose the simultaneous identification of general source terms, temperature distribution and temperature gradient distribution in parabolic equations by mollification techniques provided that suitable noisy data is available only at the active boundary. More precisely: The problem is to numerically determine  $u(x,t)$ ,  $u_x(x,t)$  and  $f(x,t)$  so that

$$\begin{aligned} u_t(x,t) &= (a(x,t)u_x(x,t))_x + f(x,t), & 0 \leq x < 1, 0 < t < 1, \\ u(0,t) &= \alpha^\varepsilon(t), & 0 \leq t \leq 1, \\ u_x(0,t) &= \beta^\varepsilon(t), & 0 \leq t \leq 1, \\ f(0,t) &= \gamma^\varepsilon(t), & 0 \leq t \leq 1, \end{aligned} \tag{6.1}$$

where  $a(x, t)$  is given and the available data functions  $\alpha^\varepsilon, \beta^\varepsilon$  and  $\gamma^\varepsilon$  are approximations to  $\alpha, \beta$  and  $\gamma$ , respectively, and satisfy  $\|\alpha^\varepsilon - \alpha\|_\infty \leq \varepsilon$ ,  $\|\beta^\varepsilon - \beta\|_\infty \leq \varepsilon$  and

$$\|\gamma^\varepsilon - \gamma\|_\infty \leq \varepsilon.$$

Here  $\varepsilon$  is a positive number called tolerance that is useful to state the problem and to prove error estimates. However, in actual computations, the selection of mollification parameters does not depend on the knowledge of tolerance  $\varepsilon$ .

The statement of the direct heat conduction problem based on the first equation in 6.1, requires an initial condition  $u(x, 0)$  known for all  $x \in [0, 1]$  and two boundary conditions,  $u(0, t)$  and  $u(1, t)$  known for all  $t > 0$ . Conversely, the statement of the inverse problem depends on three approximately known distributions of temperature  $\alpha^\varepsilon$ , temperature gradient  $\beta^\varepsilon$  and source term  $\gamma^\varepsilon$  at one of the boundaries, the active boundary. No information on the other boundary or initial condition are prescribed. Moreover, the presence of noisy data  $\alpha^\varepsilon, \beta^\varepsilon$  and  $\gamma^\varepsilon$  makes the problem a lot more difficult, taking into consideration that it is an ill-posed problem and therefore, small perturbations in the data may cause large errors in the computed solution.

When the direct problem associated with the inverse identification problem is linear, it is known (cf., e.g., Murio, 1993; Murio, 2002; Zhan and Murio, 1998) that mollified space marching explicit schemes are excellent strategies for the solution of the inverse problem. This is the case here. The authors state a regularized problem

$$\begin{aligned} v_t(x, t) &= (a(x, t) v_x(x, t))_x + f(x, t), & 0 \leq x < 1, 0 < t < 1, \\ v(0, t) &= J_{\delta_0} \alpha^\varepsilon(t), & 0 \leq t \leq 1, \\ v_x(0, t) &= J_{\delta_0^*} \beta^\varepsilon(t), & 0 \leq t \leq 1, \\ f(0, t) &= J_{\delta_0'} \gamma^\varepsilon(t), & 0 \leq t \leq 1, \end{aligned} \tag{6.2}$$

where all  $\delta$ -mollifications are taken with respect to  $t$  and the regularization parameters  $\delta_0, \delta_0^*$  and  $\delta_0'$  are automatically selected by GCV (see section 1.6 for details.) A finite difference approximation of 6.2 yields a mollified space marching algorithm that is stable and convergent (theorems 7 and 8 of Yi and Murio, 2004b).

### 6.1.2 SOURCE TERMS IN A 2-D IHCP

In Yi and Murio (2004a), the authors generalize the 1-D identification process of Yi and Murio (2004b) by studying the simultaneous identification of general source terms, temperature distribution and temperature gradient distribution in two-dimensional parabolic equations by mollification techniques provided that suitable noisy data is available only at the active boundary. They solve the following identification problem: Find  $u(x, y, t)$ ,  $\nabla u(x, y, t)$  and  $f(x, y, t)$  throughout the domain  $[0, x_{\max}] \times [0, 1] \times [0, 1]$  from measured approximations of  $\alpha(y, t), \beta(y, t)$  and

$\gamma(y,t)$  satisfying

$$\begin{aligned}
 u_t &= \nabla(a(x,y,t)\nabla u) + f(x,y,t), & 0 < x < x_{\max}, 0 < y < 1, \\
 & & 0 < t < 1 \\
 u(0,y,t) &= \alpha^\varepsilon(y,t), & 0 \leq y \leq 1, 0 \leq t \leq 1, \\
 u_x(0,y,t) &= \beta^\varepsilon(y,t), & 0 \leq y \leq 1, 0 \leq t \leq 1, \\
 f(0,y,t) &= \gamma^\varepsilon(y,t), & 0 \leq y \leq 1, 0 \leq t \leq 1,
 \end{aligned} \tag{6.3}$$

where  $a(x,y,t)$  is given and the available data functions  $\alpha^\varepsilon, \beta^\varepsilon$  and  $\gamma^\varepsilon$  approximate  $\alpha, \beta$  and  $\gamma$  respectively, and satisfy  $\|\alpha^\varepsilon - \alpha\|_\infty \leq \varepsilon$ ,  $\|\beta^\varepsilon - \beta\|_\infty \leq \varepsilon$  and  $\|\gamma^\varepsilon - \gamma\|_\infty \leq \varepsilon$ .

Following the pattern established in (6.2), the authors state a regularized problem. The main difference is the mollification filtering, which takes place in two directions,  $y$  and  $t$ .

A finite difference approximation of gradients and derivatives leads to a mollified marching in the  $x$ -direction algorithm that is stable and convergent (theorems 8 and 9 of Yi and Murio, 2004a).

### 6.1.3 PARAMETERS IN A TRANSPORT EQUATION

The subject is mathematical ecology and the parameters of interest are population density and dispersal rate of a species in its habitat. Following Coles, 2002, the problem is to identify the dispersal rate (diffusion coefficient)  $a(x)$  and the population density  $u(x,t)$  satisfying the transport equation with advection

$$\begin{aligned}
 u_t + (v(x,t)u)_x &= (a(x)u_x)_x, & 0 < x < 1, 0 < t < 1, \\
 u(0,t) &= \alpha^\varepsilon(t), & 0 \leq t \leq 1, \\
 u_x(0,t) &= \beta^\varepsilon(t), & 0 \leq t \leq 1, \\
 u(x,0) &= \gamma^\varepsilon(t), & 0 \leq t \leq 1, \\
 a(0) &= \eta^\varepsilon
 \end{aligned} \tag{6.4}$$

where  $v(x,t)$  is given for all  $(x,t) \in [0,1] \times [0,1]$  and the available data functions  $\alpha^\varepsilon, \beta^\varepsilon$  and  $\gamma^\varepsilon$  are approximations to  $\alpha, \beta$  and  $\gamma$ , respectively and satisfy  $\|\alpha^\varepsilon - \alpha\|_\infty \leq \varepsilon$ ,  $\|\beta^\varepsilon - \beta\|_\infty \leq \varepsilon$  and  $\|\gamma^\varepsilon - \gamma\|_\infty \leq \varepsilon$ . Here, as before,  $\varepsilon$  is positive tolerance.

From (6.4) a regularized problem is obtained by means of discrete mollification with automatic selection of regularization parameters. For functions  $\alpha^\varepsilon$  and  $\beta^\varepsilon$  it is mollification with respect to  $t$  while for function  $\gamma^\varepsilon$  it is space mollification. Again, through a finite difference approximation of derivatives, the regularized problem yields a mollified space marching algorithm that is stable and convergent (theorems 3.3 and 3.4 of Coles, 2002).

### 6.1.4 PARAMETERS IN A DRYING SYSTEM

The subject is thermal drying in a porous medium, therefore we may think of food industries or environmental engineering. There are heat conduction and mass transfer at the same time and the phenomenon is modeled by a parabolic system known as a Luikov system. The problem that appears in Coles and Murio (2006) consists on the simultaneous recovery of  $u(x, t)$ ,  $v(x, t)$ ,  $u_x(x, t)$  and  $v_x(x, t)$  throughout the domain  $[0, 1] \times [0, 1]$ , satisfying

$$\begin{pmatrix} u_t \\ v_t \end{pmatrix} = \begin{pmatrix} \alpha(x) & \beta(x) \\ \gamma(x) & \eta(x) \end{pmatrix} \begin{pmatrix} u_{xx} \\ v_{xx} \end{pmatrix}, \quad 0 < x < 1, t > 0,$$

with boundary conditions  $u(0, t) = g_1(t)$ ,  $v(0, t) = g_2(t)$ ,  $u_x(0, t) = g_3(t)$  and  $v_x(0, t) = g_4(t)$  not known exactly. The available data  $g_1^\epsilon$ ,  $g_2^\epsilon$ ,  $g_3^\epsilon$  and  $g_4^\epsilon$  are discrete noisy functions with maximum noise level  $\epsilon$ .

Assuming matrix  $A = \begin{pmatrix} \alpha(x) & \beta(x) \\ \gamma(x) & \eta(x) \end{pmatrix}$  has positive determinant bounded away from 0 for all  $x \in [0, 1]$ , it is possible to approximate derivatives by finite differences combined with discrete mollification and obtain a stable and convergent mollified space marching scheme for the identification process. The details appear in theorems 2.1 and 2.2 of Coles and Murio, 2006. As was stated above, the linearity of the direct problem expedites the design of the marching scheme.

## 6.2 FRACTIONAL CALCULUS

Differentiation and integration to arbitrary order appear in the early nineteenth century with Abel in 1823 and some other great mathematicians like Riemann and Liouville shortly after him. However, the first treatise devoted to the so-called fractional calculus appeared only in 1974, written by Oldham and Spanier and recently republished by Dover in Oldham and Spanier (2006). In recent decades this field has attracted the interest of researchers in several areas including physics, chemistry and many branches of engineering, with viscoelasticity the first in line. Two recommended monographs on the subject are Diethelm (2010) and Podlubny (1999).

### 6.2.1 CAPUTO FRACTIONAL DERIVATIVES

The Caputo fractional derivative of order  $\alpha > 0$ , of a differentiable function  $g$  defined on  $[0, T]$ , is given by the convolution integral

$$\begin{aligned} (D^{(\alpha)}g)(t) &= \frac{1}{\Gamma(n-\alpha)} \int_0^t \frac{g^{(n)}(s)}{(t-s)^{\alpha+1-n}} ds, \quad 0 \leq t \leq T, \quad n-1 < \alpha < n, \quad n \in \mathbb{N} \\ (D^{(\alpha)}g)(t) &= \frac{d^n g(t)}{dt^n}, \quad 0 \leq t \leq T, \quad \alpha = n. \end{aligned} \tag{6.5}$$

where  $\Gamma$  is the gamma function. Fractional differential operators are particular first-kind Volterra integral equations and it is known that first-kind integral equations are generally ill-posed, consequently, stable computation of fractional derivatives requires some kind of regularization whenever noisy data is present. In Murio (2006) he introduces discrete mollification as a means to overcome this difficulty for the case  $n = 1$ ,  $T = 1$  and noisy data  $g^\varepsilon$ ; that is,

$$(D^{(\alpha)}g^\varepsilon)(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{g^{\varepsilon(1)}(s)}{(t-s)^\alpha} ds, \quad 0 \leq t \leq 1, \quad 0 < \alpha < 1.$$

The approximate fractional derivative is

$$J_\delta \left( D^{(\alpha)}g^\varepsilon \right) (t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{(J_\delta g^\varepsilon)'(s)}{(t-s)^\alpha} ds, \quad (6.6)$$

with the following error estimate: Assuming  $g'$  and  $g^\varepsilon$  are uniformly Lipschitz on  $[0, 1]$  and  $\|g - g^\varepsilon\|_\infty \leq \varepsilon$ , there exists a constant  $C$ , independent of  $\delta$  such that

$$\left\| J_\delta \left( D^{(\alpha)}g^\varepsilon \right) - D^{(\alpha)}g \right\|_\infty \leq \frac{C}{(1-\alpha)\Gamma(1-\alpha)} \left( \delta + \frac{\varepsilon}{\delta} \right).$$

Recall that the numerical implementation is subtle as stated in Diethelm, 2008: *Traditional methods for the numerical approximation of fractional derivatives have a number of drawbacks due to the non-local nature of the fractional differential operators. The main problems are the arithmetic complexity and the potentially high memory requirements when they are implemented on a computer.* Nevertheless, Murio proposes an effective quadrature formula and enhances his paper with well chosen numerical examples.

## 6.2.2 SOURCE TERM IDENTIFICATION IN A TFDE

This time, the identification takes place in a time fractional diffusion equation of the form

$$D_t^{(\alpha)}u(x,t) = a(x,t) \frac{\partial^2 u(x,t)}{\partial x^2} + f(x,t), \quad 0 < x < 1, 0 < t < 1, \quad (6.7)$$

together with the corresponding boundary and initial conditions

$$\begin{aligned} u(0,t) &= u_0(t), & 0 \leq t \leq 1, \\ u(1,t) &= u_1(t), & 0 \leq t \leq 1, \\ u(x,0) &= u^0(x), & 0 \leq x \leq 1, \end{aligned}$$

where the symbol  $D_t^{(\alpha)}$  denotes the Caputo fractional derivative of order  $\alpha$  with respect to time,  $0 < \alpha \leq 1$ . In Murio and Mejia (2008), the authors prepare the identification algorithm taking into account the stable computation of Caputo fractional derivatives presented in section 6.2.1.

The available data is  $u^\varepsilon \in C^0(I)$  so that  $\|u - u^\varepsilon\|_{\infty, I} \leq \varepsilon$ , where  $I = I_x \times I_t = [0, 1] \times [0, 1]$ . If  $K_x$  and  $K_t$  denote uniform partitions of  $I_x$  and  $I_t$ , with step sizes  $\Delta x$  and  $\Delta t$  respectively, the discrete data temperature is measured at the grid points  $K_x \times K_t \subseteq I$ . The estimated source/sink term is

$$J_\delta f^\varepsilon(x_i, t_n) = (D_t^{(\alpha)} u^\varepsilon(x_i, t_n))_\delta - a(x_i, t_n) D_+ D_- (J_\delta u^\varepsilon(x_i, t_n)), \tag{6.8}$$

where the discrete computed fractional derivative, denoted  $(D_t^{(\alpha)} u^\varepsilon(x_i, t_n))_\delta$ , corresponds to (6.6).

This expression looks simple but computations are not because the problem is ill-posed and noisy data is the main ingredient of an approximate second derivative and a fractional derivative.

For the complete error analysis and numerical implementation, see Murio and Mejía (2008).

### 6.3 MULTISCALE ANALYSIS

In Pulgarín, Acosta and Castellanos (2009), the authors combine the Mallat algorithm with discrete mollification and apply the new procedure to electrocardiographic signals (ECG) contaminated with typical non-white noise. The new technique compares fairly well with wavelet transform procedures.

The starting point is the decomposition tree of figure 6.1. The  $A_i$ 's are filtered versions of  $S$  at different scales or resolutions and the  $D_i$ 's are complementary details.

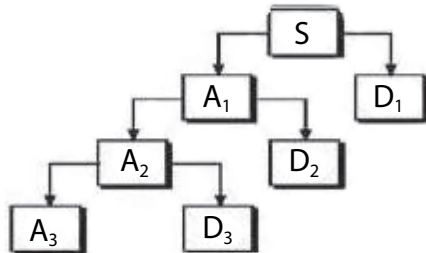


Figure 6.1. Decomposition tree

Thus, what Pulgarín, Acosta and Castellanos (2009) implement is a set of  $A_i$ 's with different resolutions by means of discrete mollification with different parameters  $\delta$ .

The following analysis is the support of the numerical implementation of the new technique. For  $\delta > 0$ , consider the Gaussian kernel

$$k^\delta(x) = \delta^{-1} \exp(-x^2/\delta^2), \quad x \in \mathbb{R}$$

Its Fourier transform is

$$\hat{k}^\delta(\omega) = \sqrt{\pi} \exp(-\delta^2 \omega^2/4), \quad \omega = 2\pi f$$

Then

$$\begin{aligned} \hat{k}^{2\delta}(\omega) &= \sqrt{\pi} \exp(-4\delta^2 \omega^2/4) \\ &= \sqrt{\pi} \exp(-\delta^2 (2\omega)^2/4) \\ &= \hat{k}^\delta(2\omega) \end{aligned} \tag{6.9}$$

So, a dyadic dilation in the kernel's parameter  $\delta$  generates a dyadic contraction in its spectrum. Furthermore, under periodic boundary conditions, by taking

$$h = \frac{2}{2\eta + 1} \quad \text{and} \quad \delta_1 = \frac{16}{\pi(2\eta + 1)}$$

the discrete mollification process cuts the upper half of the frequency components off. This analysis allows the authors to prepare dyadic spectral decompositions by mollification.

The technical details for the implementation of Mallat's algorithm appear in Pulgarín, Acosta and Castellanos (2009). The paper includes illustrative results obtained with ECG signals. The authors claim that the new procedure is capable of making an automatic filtering preserving not only low frequencies but also the main high frequency features of the signals.

## 6.4 CONCLUDING REMARKS

1. Approximating higher-order derivatives from noisy data is an inherently difficult problem to which there are no satisfactory algorithms. This is a liability in many of the identification processes described in this section.
2. Different phenomena may share the same mathematical model. Thus the identification problems presented here are appropriate in other contexts, e.g. the convection-diffusion equation of mathematical ecology presented above suits a variety of other situations outside ecology.
3. Differentiation and integration to arbitrary order are traditional subjects with more than one century of minimum activity. It might be because of the development of new fields in science and engineering or because of the appearance of computers that the subject is now very active.



## Bibliography

- Acosta, C. D. and Bürger, R. (2012). Difference schemes stabilized by discrete mollification for degenerate parabolic equations in two space dimensions. *IMA J. Numer. Anal.* 32, 1509-1540.
- Acosta, C. D., Bürger, R., and Mejía, C. E. (2012). Monotone difference schemes stabilized by discrete mollification for strongly degenerate parabolic equations. *Numerical methods for partial differential equations* 28, 38-62.
- Acosta, C.D., Bürger, R. and Mejía C.E. (2014). A stability and sensitivity analysis of parametric functions in a sedimentation model. *DYNA* 81, 22-30.
- Acosta, C. D. and Mejía, C. E. (2008). Stabilization of explicit methods for convection diffusion equations by discrete mollification. *Comput. Math. Appl.* 55, 368-380.
- Acosta, C. D. and Mejía, C. E. (2009). Approximate solution of hyperbolic conservation laws by discrete mollification. *Applied Numerical Mathematics* 59, 2256-2265.
- Acosta, C. D. and Mejía, C. E. (2010). A mollification based operator splitting method for convection diffusion equations. *Comput. Math. Appl.* 59, 1397-1408.
- Alexiades, V., Amienz, G., and Gremaud, P. (1996). Super-time-stepping acceleration of explicit schemes for parabolic problems. *Comm. Numer. Methods Engrg.* 12, 31-42.
- Berres, S., Bürger, R., and Garcés, R. (2010). Centrifugal settling of flocculated suspensions: a sensitivity analysis of parametric model functions. *Drying Technology* 28, 858-870.
- Berres, S., Bürger, R., Coronel, A., and Sepúlveda, M. (2005). Numerical identification of parameters for a strongly degenerate convection-diffusion problem modelling centrifugation of flocculated suspensions. *Applied Numerical Mathematics* 52, 311-337.
- Berres, S., Bürger, R., Karlsen, K. H., and Tory, E. M. (2003). Strongly degenerate parabolic-hyperbolic systems modeling polydisperse sedimentation with compression. *SIAM J. Appl. Math.* 64, 41-80.
- Böckmann, C., Biele, J., and Neuber, R. (1998). Analysis of multi-wavelength lidar data by inversion with mollifier method. *Pure Appl. Opt.* 7, 827.

- Bürger, R., Coronel, A., and Sepúlveda, M. (2009). Numerical solution of an inverse problem for a scalar conservation law modelling sedimentation. In J.-G. L. E. Tadmor and A. Tzavaras (Eds.), *Hyperbolic Problems: Theory, Numerics and Applications. Proceedings of Symposia in Applied Mathematics*, vol. 67, Part 2. American Mathematical Society.
- Bürger, R., Evje, S., and Karlsen, K. H. (2000). On strongly degenerate convection-diffusion problems modeling sedimentation-consolidation processes. *J. Math. Anal. Appl.* 247, 517-556.
- Bürger, R. and Karlsen, K. H. (2001). On some upwind schemes for the phenomenological sedimentation-consolidation model. *J Eng. Math.* 41, 145-166.
- Bürger, R., Karlsen, K. H., and Towers, J. D. (2005). A model of continuous sedimentation of flocculated suspensions in clarifier-thickener units. *SIAM J. Appl. Math.* 65, 882-940.
- Bürger, R., Kozakevicius, A., and Sepúlveda, M. (2007). Multiresolution schemes for strongly degenerate parabolic equations in one space dimension. *Numer Meth Partial Diff Eqns* 23, 706-730.
- Calvetti, D., Reichel, L., and Zhang, Q. (1999). Iterative solution methods for large linear discrete ill-posed problems. *Applied Comput. Control, Signals and Circuits* 1, 317-374.
- Cecchi, M. M. and Pirozzi, M. A. (2005). High order finite difference numerical methods for time dependent convection dominated problems. *Appl. Numer. Math.* 55, 334-356.
- Cheng, Y. and Shu, C.-W. (2009). Superconvergence of local discontinuous galerkin methods for one-dimensional convection-diffusion equations. *Computers and Structures* 87, 630-641.
- Coles, C. and Murio, D. A. (2001). Simultaneous space diffusivity and source term reconstruction in 2d ihcp. *Computers Math. Applic.* 42, 1549-1564.
- Colton, D., Ewing, R., and Rundell, W. (eds.) (1990). *Inverse Problems in Partial Differential Equations*. Philadelphia: SIAM.
- Coronel, A., James, F., and Sepúlveda, M. (2003). Numerical identification of parameters for a model of sedimentation processes. *Inverse Problems* 19 951-972.
- Engquist, B. and Osher, S. (1981). One-sided difference approximations for non-linear conservation laws. *Math. Comp.* 36, 321-351.
- Eriksson, K., Johnson, C., and Logg, A. (2003). Explicit time-stepping for stiff odes. *SIAM J. Sci. Comput* 25, 1142-1157.

- Evje, S. and Karlsen, K. H. (2000). Monotone difference approximations of by solutions to degenerate convection-diffusion equations. *SIAM J Numer. Anal.* 37, 1838-1860.
- Faragó, I., Gndt, B., and Havasi, A. (2008). Additive and iterative operator splitting methods and their numerical investigation. *Comput. Math. Appl.* 55, 2266-2279.
- Friedrichs, K. O. (1944). The identity of weak and strong extensions of differential operators. *Trans. AMS* 55, 132-151.
- Gilbarg, D. and Trudinger, N. S. (2001). *Elliptic partial differential equations of second order*. Springer.
- Golub, G., Heath, M., and Wahba, G. (1979, May). Generalized cross validation as a method for choosing a good ridge parameter. *Technometrics* 21(2), 215- 223.
- Hao, D. N. (1994). A mollification method for ill-posed problems. *Numer. Math.* 68, 469-506.
- Hao, D. N. (1996). A mollification method for a noncharacteristic cauchy problem for a parabolic equation. *J. Math. Anal. Appl.* 199, 873-909.
- Hao, D. N. and Reinhardt, H. J. (1997). On a sideways parabolic equation. *Inverse Problems* 13, 297-309.
- Hegland, M. and Anderssen, R. S. (1998). A mollified framework for improperly posed problems. *Numer. Math.* 78, 549-575.
- Holden, H. and Risebro, N. H. (2002). *Front Tracking for Hyperbolic Conservation Laws*. Springer.
- Karlsen, K. H., Lie, K. -A., Natvig, J. R., Nordhaug, H. F., and Dahle, H. K. (2001). Operator splitting methods for systems of convection-diffusion equations: Nonlinear error mechanisms and correction strategies. *Journal of Computational Physics* 173, 636-663.
- Karlsen, K. H. and Risebro, N. H. (1997). An operator splitting method for nonlinear convection-diffusion equations. *Numer. Math.* 77, 365-382.
- Kelley, C. (1999). Detection and remediation of stagnation in the nelder-mead algorithm using a sufficient decrease condition. *SIAM J. Optim.* 10, 43-55.
- Lagarias, J., Reeds, J. A., Wright, M. H., and Wright, P. E. (1998). Convergence properties of the nelder-mead simplex method in low dimensions. *SIAM Journal of Optimization* 9, 112-147.
- LeVeque, R. J. (1990). *Numerical Methods for Conservation Laws*. Birkhauser.
- LeVeque, R. J. (2002). *Finite Volume Methods for Hyperbolic Problems*. Cambridge University Press.
- Louis, A. K. and Maass, P. (1990). A mollifier method for linear operator equations of the first kind. *Inverse Problems* 6, 427-440.

- Luersen, M. A. and Riche, R. L. (2004). Globalized nelder-mead method for engineering optimization. *Computers and Structures* 82, 2251-2260.
- Maass, P. and Pidcock, M. K., and Sebu, C. (2008). A mollifier method for the inverse conductivity problem. *Journal of Physics: Conference Series* 135, 012068.
- Majda, A. and Bertozzi, A. (2002). *Vorticity and incompressible flow*. Cambridge: Cambridge University Press.
- Manselli, P. and Miller, K. (1980). Calculations of the surface temperature and heat flux on one side of a wall from measurements on the opposite side. *Ann. Mat. Pura Appl.* 123, 161-183.
- Mejía, C. E. (2007). *Sobre el método de molificación*. Universidad Nacional de Colombia: Trabajo presentado como requisito parcial para promoción a profesor titular.
- Mejía, C. E., Acosta, C. D., and Saleme, K. I. (2011). Numerical identification of a nonlinear diffusion coefficient by discrete mollification. *Comput. Math. Appl.* 62, 2187-2199.
- Mejía, C. E., Murio, D. A., and S. Zhan (2001). Some applications of the mollification method. In M. Lassonde (Ed.), *Approximation, Optimization and Mathematical Economics*, pp. 213-222. Physica-Verlag.
- Mejía, C. E. and Murio, D. A. (1993). Mollified hyperbolic method for coefficient identification problems. *Computers Math. Applic.* 26, 1-12.
- Mejía, C. E. and Murio, D. A. (1995). Numerical identification of diffusivity coefficient and initial condition by discrete mollification. *Comput. Math. Appl.* 30, 35-50.
- Mejía, C. E. and Murio, D. A. (1996). Numerical solution of generalized ihcp by discrete mollification. *Computers Math. Applic.* 32, 33-50.
- Montoya, L. J., Mejía, C. E., and Toro, F. M. (2000). Estabilización de esquemas por molificación discreta. *Avances en Recursos Hidráulicos* 7, 102-116.
- Murio, D. A. (1981). The mollification method and the numerical solution of an inverse heat conduction problem. *SIAM Journal on Scientific and Statistical Computing* 2(1), 17-34.
- Murio, D. A. (1993). *The Mollification Method and the Numerical Solution of Ill-Posed Problems*. John Wiley.
- Murio, D. A. (2002). Mollification and space marching. In K. Woodbury (Ed.), *Inverse Engineering Handbook*. CRC Press.
- Murio, D. A. (2006). On the stable numerical evaluation of caputo fractional derivatives. *Computers Math. Applic.* 51, 1539-1550.
- Murio, D. A. (2007). Stable numerical solution of a fractional-diffusion inverse heat conduction problem. *Computers Math. Applic.* 53, 1492-1501.

- Murio, D. A. and Mejía, C. E. (2008). Source terms identification for time fractional diffusion equation. *Revista Colombiana de Matemáticas* 42, 25-46.
- Murio, D. A. and Hinestroza, D. (1989). Numerical identification of forcing terms by discrete mollification. *Computers Math. Applic.* 17, 1441-1447.
- Murio, D. A., Hinestroza, D., and Mejía, C. E. (1992). New stable numerical inversion of abel-s integral equation. *Computers Math. Applic.* 23, 3-11.
- Murio, D. A., Mejía, C. E., and Zhan, S. (1998). Discrete mollification and automatic numerical differentiation. *Computers Math. Applic.* 35, 1-16.
- Nessyahu, H. and Tadmor, E. (1990). Non-oscillatory central differencing for hyperbolic conservation laws. *J. Comp. Phys.* 87(2), 408-463.
- Ng, M. K., Chan, R. H., and Tang, W. (1999). A fast algorithm for deblurring models with neumann boundary conditions. *SIAM J. Sci. Comput.* 21(3), 851- 866.
- Price, C., Coope, I. D., and Byatt, D. (2002). A convergent variant of the nelder-mead algorithm. *Journal of Optimization Theory and Applications* 113, 5-19.
- Pulgarín, J. D., Acosta, C. D. and Castellanos, G. (2009). Multiscale analysis by means of discrete mollification for ecg noise reduction. *DYNA* 76, 185-191.
- Saleme, K. (2009). Identificación de coeficientes y el método de molificación. Medellín, Colombia: Master's Thesis, Department of Mathematics, National University of Colombia.
- Schuster, T. and Quinto, E. T. (2005). On a regularization scheme for linear operators in distribution spaces with an application to the spherical radon transform. *SIAM J. Appl. Math.* 65, 1369-1387.
- Shu, C. W. (1998). Essentially non-oscillatory and weighted essentially nonoscillatory schemes for hyperbolic conservation laws. In A. Quarteroni (Ed.), *Advanced Numerical Approximation of Nonlinear Hyperbolic Equations, Lectures Notes in Mathematics*, Vol 1697, pp. 325-432. Springer, Berlin.
- Smith, S. W. (1999). *The Scientist and Engineer-s Guide to Digital Signal Processing*. San Diego, California, <http://www.dspguide.com>: California Technical Publishing.
- Vogel, C. R. (2006). *Computational methods for inverse problems*. SIAM.
- Wubs, F. W. (1986). Stabilization of explicit methods for hyperbolic partial differential equations. *Internat. J. Numer. Methods Fluids* 6, 641-657.
- Yi, Z. and Murio, D. A. (2004a). Identification of source terms in 2-d ihcp. *Computers Math. Applic.* 47, 1517-1533.
- Yi, Z. and Murio, D.A. (2004b). Source term identification in 1-D IHCP. *Computers Math. Applic.* 47, 1921-1933.
- Zhan, S., Coles, C., and Murio, D. A. (2001). Automatic numerical solution of generalized 2-d ihcp by discrete mollification. *Computers Math. Applic.* 41, 15-38.

- Zhang, S. and Shu, C.-W. (2007). A new smoothness indicator for weno schemes and its effect on the convergence to steady state solutions. *J. Sci. Comput.* 31, 273-305.
- Zhao, Q. H., Urosević, D., Mladenovi, N., and Hansen, P. (2009). A restarted and modified simplex search for unconstrained optimization. *Computers & Operations Research* 36, 3263-3271.

# Index

## A

algorithm, 48, 72, 85, 86, 92, 93, 95-97

## C

Cauchy problem, 56, 58  
central scheme, 35, 45, 48, 50  
CFL condition, 39, 46, 47, 49, 50, 59, 60, 65-67, 82  
conservation law, 45, 46, 52, 55, 58, 65, 70  
conservative equation, 25  
consistency, 22, 41, 46, 53, 57, 61  
consolidation, 65, 68, 82, 83  
convection, 20, 37, 38, 43, 55, 57, 60, 64, 70, 97  
convergence, 22, 24, 39, 50, 55, 60, 65, 67, 70, 71, 74-76, 82, 85  
convolution, 19, 26, 32, 35, 39, 49, 94  
cost function, 82, 84, 85, 89

## D

diffusion, 20, 37, 38, 43, 55-57, 60, 65, 68, 70-72, 76, 81, 84, 88, 93  
direct problem, 71, 72, 82, 83, 85, 86, 92, 94  
discrete operator, 19, 20, 22, 24, 28-30, 32, 35, 46, 66

## E

Engquist-Osher numerical flux, 65, 66  
entropy solution, 58, 65, 66, 84  
explicit scheme, 24, 37, 39, 41, 43, 45, 55, 60, 65-67, 82, 92-94

## F

flux function, 46, 47, 49, 50, 58-60, 65, 66, 72, 73, 81, 83, 84  
fminsearch, 85-87  
fractional derivative, 91, 94  
FTCS forward-time central-space, 37-39, 41

## G

GCV generalized cross validation, 30, 75, 92  
geophysics, 71

## H

heat equation, 39, 41, 55, 56, 60  
higher order method, 30, 55, 70

## I

identification, 19, 60, 71, 75, 76, 82, 86, 91-95  
ill-posed problem, 19, 23, 24, 75, 92, 95, 96  
inverse problem, 19, 60, 71-74, 76, 82, 84, 85, 92

**L**

Lax-Friedrichs scheme, 45, 63

**M**

mollification, 19-21, 24, 25, 30, 35, 37, 41, 43, 46, 47, 49, 53, 57, 58, 60, 61, 66, 68, 71, 81, 88, 91-96

boundary conditions, 26

even reflection, 28

odd reflection, 29, 41

periodic, 28, 38

scaled, 27

zero padding, 27

kernel, 20

nonlinear, 30

parameter delta, 21, 24, 30, 56, 73, 75, 92, 93, 96, 97

parameter eta, 20, 21, 30

parameter selection, 30

stencil, 21, 30

weights, 20, 21, 25, 29, 35, 47, 50, 57

mollified scheme, 20, 35, 38, 46, 49, 55, 59, 62, 66, 67, 84, 92-94

mollifier, 19, 35

monotone scheme, 20, 45-47, 65, 66, 84

multiscale analysis, 91, 96

**N**

Nelder-Mead simplex method, 85

noise, 24, 73, 75, 89, 91, 94-96

nonlinear, 30, 31, 41, 55, 58, 61, 64, 70-72, 88

normalization constant, 21, 34

NT Nessyahu-Tadmor methods, 45, 48, 50, 53

numerical differentiation, 19, 23, 24

**O**

oil recovery, 71

operator splitting, 20, 55, 57, 58, 60, 64

**P**

partial differential equation, 19, 24, 37, 41, 71

**R**

regularization, 19, 24, 30, 72, 75, 89, 92, 93, 95

**S**

sedimentation, 65, 68, 82, 83

smoothness indicator, 31

source term identification, 91, 92, 95

space marching scheme, 72, 74, 88, 92-94

spurious oscillation, 30, 43, 45, 50

stability, 22, 24, 37, 39, 42, 45-48, 50, 53, 57, 59, 60, 65

stabilization, 19, 30, 37, 39, 42, 55, 70

strongly degenerate parabolic equation, 20, 55, 65, 68, 71, 83, 89

**T**

Taylor expansion, 22, 67

traffic flow, 65, 82

TV-stability, 59

TVD total variation diminishing, 25, 42, 46, 47, 49, 50, 60

**V**

viscosity, 58, 62

**W**

WENO, 30

## Otros títulos de la Serie Techné

*Introducción a la computación numérica usando la herramienta Scilab* (2014)  
Javier Ignacio Carrero Mantilla

*La custodia de los hijos en las parejas separadas* (2014)  
Yolanda López Díaz

*Repelencia al agua en Andisoles de Antioquia* (2011)  
Daniel Francisco Jaramillo Jaramillo

*Gestión empresarial en Colombia: un aporte desde la administración* (2011)  
Gregorio Calderón  
Julia Naranjo  
Claudia Álvarez

*Desarrollo rural: superando el desarrollo agrícola* (2011)  
Fabio Alberto Pachón Ariza



carlos daniel  
**acosta**

carlos enrique  
**mejía**

*Carlos Daniel Acosta*, has been at Universidad Nacional de Colombia at Manizales since 2000 where he is an associate professor of Mathematics but he frequently teaches advanced engineering classes. His research is of an interdisciplinary nature but most of it is related to mollification.

*Carlos Enrique Mejía*, is a professor of Mathematics at Universidad Nacional de Colombia at Medellín since 1993. His graduate classes in applied mathematics are attended by students of several departments. His research work is almost entirely based on mollification.

The discrete mollification method is a data smoothing procedure, based on convolution, that is appropriate for the stabilization of explicit schemes for the numerical solution of partial differential equations and the regularization of ill-posed problems. This text introduces some of the main results and recent developments in discrete mollification and discusses several important topics of current research interest.

The book develops and applies numerical methods based on discrete mollification for a variety of situations arising in applied mathematics, including convection-diffusion equations, conservation laws, strongly degenerate parabolic equations and several system identification problems. For each topic there are theoretical considerations concerning stability and convergence and a generous amount of illustrative examples. The intended audience for this book includes mathematicians, physicists and engineers.

