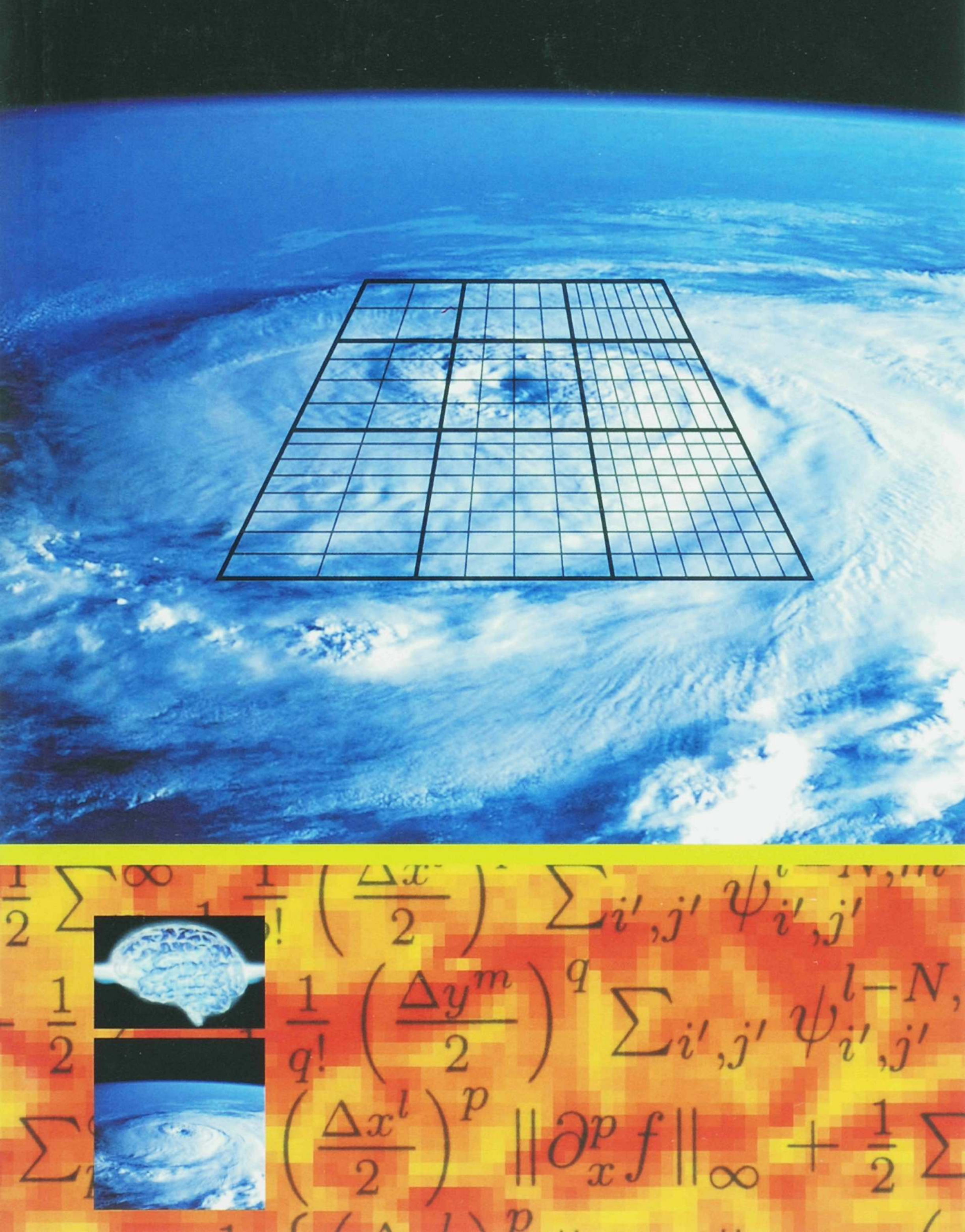
NUMERICAL TIME INTEGRATION ON SPARSE GRIDS

BORIS LASTDRAGER



NUMERICAL TIME INTEGRATION ON SPARSE GRIDS

BORIS LASTDRAGER

NUMERICAL TIME INTEGRATION ON SPARSE GRIDS

ACADEMISCH PROEFSCHRIFT

TER VERKRIJGING VAN DE GRAAD VAN DOCTOR
AAN DE UNIVERSITEIT VAN AMSTERDAM
OP GEZAG VAN DE RECTOR MAGNIFICUS
PROF. MR. P.F. VAN DER HEIJDEN TEN OVERSTAAN
VAN EEN DOOR HET COLLEGE VOOR PROMOTIES
INGESTELDE COMMISSIE, IN HET OPENBAAR TE
VERDEDIGEN IN DE AULA DER UNIVERSITEIT OP
WOENSDAG 18 SEPTEMBER 2002 TE 10:00 UUR.

DOOR

BORIS LASTDRAGER
GEBOREN TE WINSCHOTEN

promotiecommissie

promotor	prof. dr. J.G. Verwer
co-promotor	prof. dr. ir. B. Koren
overige leden	prof. dr. P.W. Hemker dr. W. Hoffmann prof. dr. P.J. van der Houwen prof. dr. ir. P. Wesseling prof. dr. M. van Veldhuizen prof. dr. M.N. Spijker

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

The work in this thesis was made possible by financial support from the Dutch Organization for Scientific Research NWO (Nederlandse Organisatie voor Wetenschappelijk Onderzoek) under project number 613-02-036.

Time Integration on Sparse Grids
Boris Lastdrager
PhD thesis University of Amsterdam

ISBN 90-6196-511-X

"PROGRAMMING TODAY IS A RACE BETWEEN SOFTWARE ENGINEERS STRIVING TO BUILD BIGGER AND BETTER IDIOT-PROOF PROGRAMS, AND THE UNIVERSE TRYING TO PRODUCE BIGGER AND BETTER IDIOTS. SO FAR, THE UNIVERSE IS WINNING."

RICH COOK

PREFACE

This PhD-thesis is based on four years of research that I have carried out at the Center for Mathematics and Computer science (CWI) in Amsterdam, the Netherlands, during the period 1998-2002. This research has resulted in four publications which are listed below, together with the corresponding chapters of this thesis. The chapters can be read independently.

Chapter 2 B. Lastdrager, B. Koren,
Error Analysis for Function Representation
by the Sparse-Grid Combination Technique,
Report MAS-R9823, CWI (1998)

Chapter 3 B. Lastdrager, B. Koren, J.G. Verwer,
The Sparse-Grid Combination Technique
Applied to Time-Dependent Advection Problems,
Applied Numerical Mathematics, Vol. 38, pp. 377-401 (2001)
Also appeared in reduced form in
Lecture Notes in Computational Science and Engineering,
Vol. 14, Multigrid Methods IV, Springer-Verlag (2000)

Chapter 4 B. Lastdrager, B. Koren, J.G. Verwer,
Solution of Time-Dependent Advection-Diffusion Problems
with the Sparse-Grid Combination Technique and a Rosenbrock Solver,
Journal of Computational Methods in Applied Mathematics,
Vol. 1, pp. 86-98 (2001)

Chapter 5 B. Lastdrager,
Numerical Solution of Mixed Gradient-Diffusion Equations
Modelling Axon Growth,
Report MAS-R0203, CWI (2002)

Prior to my PhD research I did research on theoretical physics for my Masters' thesis which led to the following publication

B. Lastdrager, A. Tip, J. Verhoeven, *Theory of Cherenkov and Transition Radiation from Layered Structures,* Physical Review E, Vol. 61, pp. 5767-5778 (2000)

and patent

J. Verhoeven, B. Lastdrager, A. Tip, D.K.G. de Boer, *New Source-Optics Combination for EUV Lithography,* Philips patent ID-number 600353, WK24.886 (1997)

CONTENTS

Preface i

1. Introduction 1
- 1.1 Sparse grid combination technique 2
- 1.2 Mixed gradient-diffusion equations 5
- 1.3 Thesis outline 5

2. Function Representation 9
- 2.1 Introduction 10
  - 2.1.1 Sparse-grid techniques 10
  - 2.1.2 The combination technique 11
- 2.2 Error accumulation 12
  - 2.2.1 Introduction 12
  - 2.2.2 The $[\frac{1}{2}, \frac{1}{2}, 0]$ combination scheme 13
  - 2.2.3 The $[1, 1, -1]$ combination scheme 15
  - 2.2.4 The $[0, 0, 1]$ combination scheme 17
  - 2.2.5 Discussion 17
- 2.3 Local errors 17
  - 2.3.1 Piecewise-constant interpolation 19
  - 2.3.2 Piecewise bi-linear interpolation 24
  - 2.3.3 A numerical test 27
  - 2.3.4 Discussion 27
- 2.4 Extension to three dimensions 30
  - 2.4.1 Piecewise-constant interpolation 31
  - 2.4.2 Piecewise tri-linear interpolation 32
  - 2.4.3 The semi-sparse grid 33
  - 2.4.4 A numerical test 35
  - 2.4.5 Discussion 36
- 2.5 Discontinuous functions 37
  - 2.5.1 The $[\frac{1}{2}, \frac{1}{2}, 0]$ piecewise-constant scheme. 41
  - 2.5.2 The $[1, 1, -1]$ piecewise-constant scheme. 42
  - 2.5.3 The $[0, 0, 1]$ piecewise-constant scheme 44
  - 2.5.4 A numerical test 45
  - 2.5.5 Discussion 45
- 2.6 Conclusions 46

3. Advection Problems 51
3.1 Introduction 52
3.2 Discretization error 53
3.2.1 Structure of the discretization error 54
3.2.2 Third-order upwind discretization 55
3.2.3 Time-dependent coefficients 55
3.3 Combination technique 56
3.4 Combined discretization error 57
3.4.1 Effect of the combination technique on a single error term . . 57
3.4.2 Leading-order results 60
3.4.3 Mapping of error terms 60
3.4.4 Additional error due to interpolation 61
3.4.5 Intermediate combinations 61
3.4.6 Qualitative behavior of the error 63
3.4.7 Time-dependent coefficients 63
3.5 Asymptotic efficiency 64
3.5.1 Computational work 65
3.5.2 Efficiency comparison 65
3.6 Numerical results 66
3.6.1 Numerical setup 66
3.6.2 Test cases 66
3.6.3 Results 67
3.6.4 Implementational issues 69
3.6.5 Richardson extrapolation 72
3.7 Conclusions 74

4. Advection-diffusion Problems 79
4.1 Introduction 80
4.2 The model problems 81
4.2.1 Model problem 1: The advection-diffusion equation 81
4.2.2 Model problem 2: Burgers' equations 81
4.3 The sparse grid combination technique 82
4.4 Spatial discretization errors 85
4.5 The Rosenbrock solver ROS3 86
4.5.1 Factorization 87
4.5.2 Time step size control 87
4.5.3 Numerical illustration of factorized ROS3 88
4.6 Results 89
4.6.1 Validation of the sparse grid error expression 90
4.6.2 Model problem 1: the advection diffusion equation 90
4.6.3 Model problem 2: Burgers' equations 93
4.7 Conclusions 94

5. Axon Growth 97
5.1 Introduction 98
5.2 Biological model 98

- 5.3 Spatial Discretization . . . 102
- 5.4 The Rosenbrock method . . . 103
  - 5.4.1 Stability . . . 105
  - 5.4.2 Spatial factorization . . . 107
- 5.5 Implementation of source terms and gradient . . . 107
  - 5.5.1 Highly localized source term . . . 109
  - 5.5.2 Resolved source term . . . 112
- 5.6 Numerical results . . . 114
  - 5.6.1 Convergence . . . 114
  - 5.6.2 Stability . . . 116
  - 5.6.3 Efficiency . . . 116
  - 5.6.4 Parameter sensitivity . . . 117
  - 5.6.5 Source term and gradient implementation . . . 118
- 5.7 Discussion . . . 118

Summary . . . 123

Samenvatting . . . 125

Dankwoord . . . 127

Curriculum Vitae . . . 129

1

INTRODUCTION

The field of applied mathematics consists of the part of mathematics that can be useful in solving real life-problems. Since surprisingly often even the most exotic mathematical techniques can be applied on real-life problems, applied mathematics is still a very broad term. In fact, most mathematical sub-fields can be classified both under applied mathematics and pure mathematics.

Numerical mathematics is a field that would rarely be seen as purely theoretical mathematics, since its applications are so obvious. The term numerical mathematics is not always clear to laymen; they sometimes wonder if not all mathematics is numerical. Numerical mathematics focuses on mathematical methods that can be implemented as computer programs, which then solve the problem under consideration.

Even before the advent of computers numerical mathematics existed, but then as a far more academic subject. It is only since the computer has become mainstream, that numerical mathematics has become a highly applied discipline. Due to the ingenuity of modern numerical algorithms and the computational power of modern computers, highly complex problems can be solved that could not be solved without numerical mathematics.

An important sub-field of numerical mathematics concerns the solution of ordinary differential equations (ODEs) and partial differential equations (PDEs). Many real world problems can be formulated in terms of systems of differential equations. These can often only be solved by means of numerical mathematics. Especially engineering and physics provide numerous problems formulated in terms of differential equations, but in other disciplines differential equations are frequently encountered as well.

In this thesis the focus lies on time-dependent PDEs. To solve these equations we apply the method of lines. This implies that first the spatial derivatives are approximated by finite differences, i.e., they are discretized, yielding ODEs in time. Then a time stepping method is applied to integrate the resulting semi-discrete problem in time.

We focus on problems with spatial variables, but the methods presented are equally well applicable to problems with other independent variables. For instance, in option pricing models one encounters the Black-Scholes equation [22]. This equation has the form of an advection-diffusion-reaction equation when one interprets the value of the underlying asset as a spatial variable.

The focus of this thesis lies mostly on systems of PDEs of the advection-diffusion type. These systems are frequently encountered in applications. They, for instance, play a prominent role in the mathematical modelling of pollution of atmospheric air, surface water and groundwater. Advanced models are three-dimensional in space. Their 3D nature and the necessity of modelling transport over long time spans requires very efficient algorithms and implementations of algorithms.

In the past, much research has been done on developing efficient solvers, notably advection schemes, tailored integrators for stiff systems of ordinary differential equations and other time stepping techniques. This has already led to significant progress. However, for advanced 3D modelling, computer capacity (computing time and memory) still is a severe limiting factor.

1.1 Sparse grid combination technique

A technique that might be capable to overcome the dimension obstacle is the sparse grid combination technique which is considered in Chapters 2, 3 and 4 of this thesis. This technique was proposed in 1992 by Griebel, Schneider and Zenger [9] and can be viewed as a specific implementation of the sparse grid approach as first introduced by Zenger in 1990 [23].

Proof of convergence of the combination technique for elliptic problems is given in [4] and [19]. Error analysis for the combination technique applied to elliptic differential equations can be found in [20]. In [16] we give a point-wise error analysis for the representation error that is inherent in the combination technique.

The idea of the sparse grid combination technique consists of solving the problem not on a single fine grid, but on a set of coarser grids with different mesh width aspect ratios. The resulting solutions are then combined to obtain a single, more accurate solution as if we are using a single fine grid. I.e., provided the solution is sufficiently smooth, the sparse grid combination technique can yield a solution comparable to a single fine grid solution at significantly lower computational costs.

For example, when we denote 2D spatial grids covering the unit square and having mesh widths $h_x = 2^{-l}$ and $h_y = 2^{-m}$ by $\Omega^{l,m}$, then the set of grids used in sparse grid combination consists of

$$\Omega^{N,0}, \Omega^{N-1,1}, \ldots, \Omega^{0,N} \tag{1.1}$$

and of

$$\Omega^{N-1,0}, \Omega^{N-2,1}, \ldots, \Omega^{0,N-1}. \tag{1.2}$$

The solutions $f^{l,m}$ found on $\Omega^{l,m}$ are then combined to get a single more accurate solution $\hat{f}^{N,N}$ on the grid $\Omega^{N,N}$ according to

$$\hat{f}^{N,N} = \sum_{l=0}^{N} P^{N,N} f^{l,N-l} - \sum_{l=0}^{N-1} P^{N,N} f^{l,N-1-l}, \tag{1.3}$$

where $P^{N,N}$ is a prolongation operator that maps a grid function from a coarser grid onto $\Omega^{N,N}$. In Figure 1.1 the grids with solid lines, except $\Omega^{N,N}$, are used in sparse grid combination.

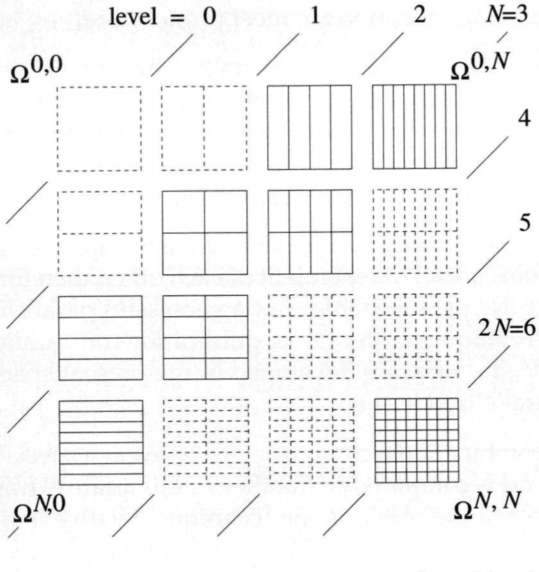


Figure 1.1: Set of grids used in sparse grid combination.

The total number of grid points used in the sparse grid combination technique can be found by a straightforward calculation that involves summation of grid points in the grids that are used, see e.g. Chapter 2 of this thesis. By doing this we find that, asymptotically for large numbers of nodes n per spatial direction, the number of scalar ODEs that must be solved for a spatially d-dimensional problem, is only proportional to

$$n(\log(n))^{d-1} \tag{1.4}$$

which, for large n, behaves as n. When we compare this with the n^d scalar ODEs that we obtain from discretization on a single grid, we see that we have essentially reduced a d-dimensional computation to a one-dimensional one.

This saving in complexity is only meaningful if the sparse grid combination technique yields a solution that is sufficiently accurate. By analyzing the discretization error we find that for a p-th order spatial discretization, the error due to the sparse grid combination technique is

$$O(h^p(\log(n))^{d-1}) \tag{1.5}$$

(see e.g. Chapters 2, 3 and 4 of this thesis), compared to $O(h^p)$ for a single grid of mesh width h.

Thus, neglecting the logarithmic factors in the asymptotic limit, the sparse grid approach is as accurate as a single grid approach while reducing a d-dimensional complexity to a one-dimensional complexity. These estimates make clear that the sparse grid is especially attractive for high dimensional problems, since then the savings from complexity reduction are most pronounced.

Above accuracy estimates are valid for smooth solutions, especially it is desirable that higher order cross derivatives take on only moderate values. When higher order cross derivatives are large, the sparse grid combination error is generally much larger than the above estimates indicate, rendering the sparse grid combination technique less effective than expected on the basis of the above estimates.

In the combination technique a number of problems are solved on conventional grids. These problems are all independent of each other, therefore the combination technique is inherently parallelizable. For a successful parallelization of a sparse grid combination technique see [8]. Other publications on parallelized sparse grids include [7] and [6]. The software developed in the present sparse grid research is currently being parallelized as well.

In [21] and [3] the combination technique is presented as a special case of multivariate extrapolation and is compared with other multivariate extrapolation methods. A distinct advantage of the combination technique, relative to the hierarchical basis implementation of the sparse-grid technique as introduced in [23], is that the former involves a straightforward discretization and solution of the PDEs on conventional grids while the latter leads to a linear algebra problem with nearly full matrix.

Generally the sparse grid method is studied in the context of finite element methods, however, in [13] and [14] a sparse grid finite volume method is presented. In [2] an implementation is worked out in detail for multidimensional Helmholtz equations.

There are similarities between sparse grid methods and semi-coarsened multi grid; in fact the methodologies can be combined (see for instance [18] and [11]).

Another important possibility of the sparse grid method is adaptivity. By introducing suitable accuracy measures, adaptive sparse grids can automatically capture important features of a solution. In [17] an adaptive sparse grid approach is successfully applied to a problem which has a solution with strongly localized, anisotropic features.

The majority of sparse grid publications is concerned with application to elliptic differential equations. Some exceptions are [10] which deals with time-dependent fluid dynamic problems, [12] which deals with hyperbolic conservation laws, and [1] which deals with parabolic problems.

1.2 Mixed gradient-diffusion equations

In Chapter 5 the focus of this thesis shifts towards the numerical solution of mixed gradient-diffusion equations. The gradient equations are ordinary differential equations while the diffusion equations are partial differential equations. Solving these different sets of equations simultaneously is a numerical challenge since it is not a priori obvious how to implement the coupling between the different sets of equations.

The motivating application behind the work on mixed gradient-diffusion equations is an axon growth model presented in [15], see also [5]. In this model the gradient equations describe the growth of the axons in linear combinations of gradient fields of chemical concentrations, so-called attractants and repellents. These attractants and repellents are solutions of the diffusion equations with source terms. The numerical challenge is to accurately compute the paths of the particles in the numerically computed concentration fields.

Numerical modelling of axon growth contributes to the understanding of the growth process of axons which in turn is indispensable for future medical applications. Treatment for neural disorders can be improved in a systematic way only with understanding of the underlying biological process. Axon growth is one of these underlying processes which is certainly not fully understood [15]. Besides modelling of axon growth, mixed gradient-diffusion equations have a much wider range of application, but in this thesis we focus on the axon growth application.

1.3 Thesis outline

Chapters 2, 3 and 4 deal with the sparse grid combination technique while Chapter 5 focusses on a method for solving mixed gradient-diffusion equations. Initially we did apply the sparse grid combination technique to the mixed gradient-diffusion problem, but we found that this yielded no significant gains relative to a single grid approach and abandoned it.

Chapter 2 focusses on the sparse grid combination technique as a means of function representation. We present expressions for the approximation error inherent in the sparse grid combination technique. We call this error the representation error. Some numerical examples illustrate that the sparse grid combination technique is an efficient means of function representation relative to representation on a single grid.

Chapter 3 considers the sparse grid combination technique applied to pure advection problems. Explicit error expressions are derived for the discretization error that comes forth from the spatial discretization of the PDE. Intermediate combinations are introduced as a means of communication between the different grids. A number of test cases is considered numerically. The sparse grid combination technique is shown to be efficient relative to a single grid, but it is also noted that an even greater gain in efficiency can be achieved by using the simpler Richardson extrapolation.

Chapter 4 deals with the sparse grid combination technique applied to mixed advection-diffusion problems. In this chapter more attention is paid to time integration aspects. In particular, a highly efficient third order Rosenbrock time marching scheme is used, with approximate matrix factorization. Again, expressions are derived for the discretization error. This chapter focusses especially on problems with grid aligned solution layers. It becomes clear that the sparse grid combination technique is especially suited for this type of problem. However, for a more general test case, the two dimensional Burgers' equations, the sparse grid combination technique is not more efficient than a single grid approach.

In Chapter 5 we study the axon growth model. This model takes the form of a mixed gradient-diffusion problem. The diffusion equations are spatially discretized yielding a large set of ODEs. Together with the gradient equations this set is solved using a second order Rosenbrock scheme with approximate Jacobian. A key question for axon growth is whether bundling and debundling of axons occurs. It is shown that for our model problem this can indeed occur, albeit for a narrow range of parameter values. Furthermore, it is shown that it is essential to properly match interpolation and computation of gradients and source terms. If these are not properly matched the axon path approximations can become highly irregular.

BIBLIOGRAPHY

[1] R. Balder, U. Rüde, S. Schneider and C. Zenger, Sparse grid and extrapolation methods for parabolic problems, in: Proc. *Int. Conf. on Computational Methods in Water Resources*, Heidelberg, 1994, A. Peters et al., eds., 1383–1392 (Kluwer Academic Publishers, Dordrecht, 1994). Pages: 5

[2] R. Balder and C. Zenger, The solution of multidimensional real Helmholtz equations on sparse grids, SIAM J. Sci. Comput., Vol. 17, No. 3, 631–646 (1996). Pages: 4

[3] H. J. Bungartz, Extrapolation, combination, and sparse grid techniques for elliptic boundary value problems, Comput. Methods Appl. Mech. Engrg., Vol. 116, 243–252 (1994). Pages: 4

[4] H. J. Bungartz, M. Griebel, D. Röschke and C. Zenger, Pointwise convergence of the combination technique for the Laplace equation, East-West J. Numer. Math., Vol. 2, No. 1, 21–45 (1994). Pages: 2

[5] G. J. Goodhill, Diffusion in axon guidance, Eur. J. Neurosci. Vol. 9, 1414-1421 (1997). Pages: 5

[6] M. Griebel, A parallelizable and vectorizable multi-level algorithm on sparse grids, in: *Parallel Algorithms for Partial Differential Equations*, W. Hackbusch, ed., Vol. 31 of *NNFM*, 94–100 (Vieweg, Braunschweig, 1991). Pages: 4

[7] M. Griebel, Parallel multigrid methods on sparse grids, in: *Multigrid Methods* **III**, W. Hackbusch and U. Trottenberg, eds., Vol. 98 of *ISNM*, 211–221 (Birkhäuser, Basel, 1991). Pages: 4

[8] M. Griebel, The combination technique for the sparse grid solution of pde's on multiprocessor machines, Parallel Processing Letters, Vol. 2, No. 1, 61–70 (1992). Pages: 4

[9] M. Griebel, M. Schneider and C. Zenger, A combination technique for the solution of sparse grid problems, in: R. Beauwens and P. de Groen, eds., *Iterative Methods in Linear Algebra*, 263–281 (North-Holland, Amsterdam, 1992). Pages: 2

[10] M. Griebel, V. Thurner, The efficient solution of fluid dynamics problems by the combination technique, SFB-Bericht Nr. 342/1/93 A (Technische Universität München, 1993). Pages: 4

[11] M. Griebel, C. Zenger and S. Zimmer, Multilevel Gauss-Seidel-algorithms for full and sparse grid problems, Computing, 50, 127–148 (1993). Pages: 4

[12] M. Griebel and G. Zumbusch, Adaptive sparse grids for hyperbolic conservation laws, *International Series of Numerical Mathematics*, Vol. 129 (Birkhäuser Verlag, Basel, 1999). Pages: 5

[13] P.W. Hemker, Sparse-grid finite-volume multigrid for 3D problems, Adv. Comput. Math., 4, 83–110 (1992). Pages: 4

[14] P.W. Hemker and P.M. de Zeeuw, BASIS3: A data structure for 3-dimensional sparse grids, in: *Euler and Navier-Stokes Solvers Using Multi Dimensional Upwind Schemes and Multigrid Acceleration*, H. Deconinck and B. Koren, eds., Vol. 56 of *NNFM*, 443–484 (Vieweg, Braunschweig, 1996). Pages: 4

[15] H. G. E. Hentschel, A. van Ooyen, *Models of axon guidance during development*, Proc. R. Soc. Lond. B 266, pp. 2231-2238 (1999). Pages: 5

[16] B. Lastdrager and B. Koren, *Error analysis for function representation by the sparse-grid combination technique*, Report MAS-R9823, CWI, Amsterdam, 1998. Pages: 2

[17] J. Noordmans and P.W. Hemker, Application of an adaptive sparse-grid technique to a model singular perturbation problem, Computing, 65, 357–378 (2000). Pages: 4

[18] J. Noordmans and P.W. Hemker, Convergence results for 3D sparse grid approaches, Numerical Linear Algebra with Applications, Vol. 1(1), 1–21 (1997). Pages: 4

[19] C. Pflaum, Convergence of the combination technique for second-order elliptic differential equations, SIAM J. Numer. Anal., Vol. 34, No. 6, 2431–2455, (1997). Pages: 2

[20] C. Pflaum and A. Zhou, Error analysis of the combination technique, Numerische Mathematik, 84, 327–350 (1999). Pages: 2

[21] U. Rüde, Multilevel, extrapolation and sparse grid methods, in: P.W. Hemker and P. Wesseling, eds., *Multigrid Methods*, **IV**, 281–294 (Birkhäuser, Basel, 1993). Pages: 4

[22] P. Wilmott, *Derivatives: The Theory and Practice of Financial Engineering*, John-Wiley, New York, 1998. Pages: 2

[23] C. Zenger, Sparse grids, in: W. Hackbusch, ed., *Notes on Numerical Fluid Mechanics*, **31**, 241–251 (Vieweg, Braunschweig, 1990). Pages: 2, 4

2

ERROR ANALYSIS FOR FUNCTION REPRESENTATION BY THE SPARSE-GRID COMBINATION TECHNIQUE

Abstract. Detailed error analyses are given for sparse-grid function representations through the combination technique. Two- and three-dimensional, and smooth and discontinuous functions are considered, as well as piecewise-constant and piecewise-linear interpolation techniques. Where appropriate, the results of the analyses are verified in numerical experiments. Instead of the common vertex-based function representation, cell-centered function representation is considered. Explicit, pointwise error expressions for the representation error are given, rather than order estimates. The paper contributes to the theory of sparse-grid techniques.

2.1 Introduction

2.1.1 Sparse-grid techniques

Sparse grids were introduced in 1990 by Zenger [6], in order to significantly reduce the number of degrees of freedom that describe the solution to a discretized partial differential equation (pde), while causing only a marginal increase in representation error relative to the standard discretization. Representing a solution as a piecewise-d-linear function on a conventional d-dimensional grid of mesh width h requires $\mathcal{O}(h^{-d})$ degrees of freedom, while the representation error is $\mathcal{O}(h^2)$. The piecewise-d-linear sparse-grid representation requires only $\mathcal{O}(h^{-1}(\log h^{-1})^{d-1})$ degrees of freedom. In fact, this is only a one-dimensional complexity, while the representation error is $\mathcal{O}(h^2(\log h^{-1})^{d-1})$, which is only slightly worse than for the conventional, full-grid representation. In 1992, Griebel, Schneider and Zenger [1] showed that, for two and three dimensions, the sparse-grid complexity and representation error can also be achieved by the so-called combination technique. This technique combines $\mathcal{O}((\log h^{-1})^{d-1})$ representations on conventional grids of different mesh widths in different directions, each containing $\mathcal{O}(h^{-1})$ points, into a representation on the conventional, full grid. One advantage of the combination technique relative to the sparse-grid technique, as introduced in [6], is that the former involves a straightforward discretization and solution of the pde's on the $\mathcal{O}((\log h^{-1})^{d-1})$ conventional grids while the latter requires discretization through a set of hierarchical basis functions, leading to a linear algebra problem with nearly full matrix. Since the problems to be solved on the $\mathcal{O}((\log h^{-1})^{d-1})$ conventional grids are all independent of each other, the combination technique is inherently parallelizable.

In the current work, combination techniques, for two-dimensional and three-dimensional functions, are analyzed in detail. In particular, expressions for the corresponding representation errors are derived. Within the current setup, only a single two-dimensional combination technique yields a representation error of order $\mathcal{O}(h^2 \log h^{-1})$. Likewise, only one three-dimensional combination technique yields a representation error of order $\mathcal{O}(h^2(\log h^{-1})^2)$. For these techniques, pointwise expressions for the representation errors are obtained. The expressions are power series that describe the errors without approximation, thus allowing a derivation of leading-order terms. Furthermore, a heuristic error analysis is given for the representation of two-dimensional discontinuous functions. It is shown that for a two-dimensional step function, the L_1-norm of the representation error is $\mathcal{O}(h^{1/2})$. Contrary to [1], the present derivations do not rely on the error results for sparse grids, as given in [6]. Instead, direct analyses are given of the steps that comprise the combination technique. An important advantage of the current approach is that for smooth functions, explicit expressions for the representation error are obtained, instead of just order estimates. Numerical results that confirm the analyses are presented.

The work is directed towards the numerical solution of large-scale transport problems, governed by systems of partial differential equations of the advection-diffusion-reaction type. These equations play a prominent role in the mathematical modeling of pollution of, e.g., atmospheric air, surface water and ground water. The three-dimensional nature of these models and the necessity of modeling transport and chemical reactions between different species over long time spans, requires very efficient algorithms. When using full-grid methods, computer capacity (computing time and memory) is and will probably remain to be a severe limiting factor. Sparse-grid methods hold out the promise of alleviating these limitations.

In order to successfully implement sparse-grid methods for complex time-dependent problems, a good understanding of the interaction between sparse-grid representation errors, discretization errors and time-integration errors is crucial. The current derivations yield expressions for the sparse-grid representation error that are sufficiently detailed to be used for the study of this interaction.

2.1.2 The combination technique

The two-dimensional combination technique is based on a grid of grids as shown in Figure 3.1.

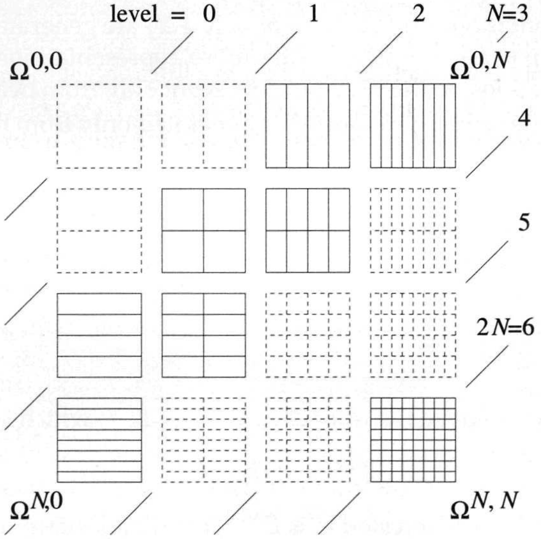


Figure 2.1: Grid of grids

The task at hand is to express a given function $f(x,y)$ on the grids $\Omega^{N,0}$, $\Omega^{N-1,1}$, …, $\Omega^{0,N}$ and on $\Omega^{N-1,0}$, $\Omega^{N-2,1}$, …, $\Omega^{0,N-1}$ and then to construct from these coarse representations a representation $\hat{f}^{N,N}$ on the grid $\Omega^{N,N}$. Throughout, upper indices label grids and lower indices label grid-point coordinates within a grid.

In sparse-grid literature, it is common to use vertex-centered grids. Yet, for our future application we intend to use cell-centered grids and therefore the current work deals solely with cell-centered grids, i.e., grid-function values are located in cell centers. Furthermore, grids extend over the unit square in two dimensions and over the unit cube in three dimensions. In two dimensions, the total number of degrees of freedom contained in the coarse representations, for two-dimensions, is given by $2^N(N-1)+1$, as can be seen by simply counting the total number of cells. The test procedure comprises the following steps:

1. The given function is restricted to the coarse grids $\Omega^{N,0},\ldots,\Omega^{0,N},\Omega^{N-1,0},\ldots,\Omega^{0,N-1}$.
2. The information on the coarse grids is used to construct a representation $\hat{f}^{N,N}$ on the finest grid.
3. The representation error is determined by comparing the representation $\hat{f}^{N,N}$ with $f^{N,N}$, i.e., with the function $f(x,y)$ directly restricted to the grid $\Omega^{N,N}$.

All restrictions are done by injection, i.e., to a cell $\Omega_{i,j}^{l,m}$, a function value

$$f_{i,j}^{l,m} \equiv f(x_i^l, y_j^m) \equiv f\left((i+\frac{1}{2})2^{-l}, (j+\frac{1}{2})2^{-m}\right)$$

is assigned. In step 2, the fine-grid representation is not found directly from the coarse-grid representations. Rather, given the representations on $\{\Omega^{l,m}, l+m = N, N-1\}$, representations on $\{\Omega^{l,m}, l+m = N+1\}$ are generated and this process is then repeated up to $l+m=2N$. Furthermore, representations are not generated from all representations on the previous levels but only from nearest neighbor representations, i.e., the representation $\hat{f}^{l,m}$ is generated only from the representations $\hat{f}^{l,m-1}$, $\hat{f}^{l-1,m}$ and $\hat{f}^{l-1,m-1}$.

2.2 Error accumulation

2.2.1 Introduction

In the following we analyze the *representation error* $E^{l,m}$, which we define as

$$E^{l,m} \equiv \hat{f}^{l,m} - f^{l,m}. \tag{2.1}$$

The quantity that we are interested in is $E^{N,N}$, the representation error on the finest grid. At this point, we introduce *prolongation operators* $P^{l,m}$ which are linear operators that map grid functions from a grid $\Omega^{l',m'}$ into grid functions on the finer grid $\Omega^{l,m}$ $(l \geq l', m \geq m')$. We consider representations that satisfy the following relation

$$\hat{f}^{l,m} = \begin{cases} f^{l,m}, & \text{for } l+m \leq N, \\ \alpha P^{l,m}\hat{f}^{l-1,m} + \beta P^{l,m}\hat{f}^{l,m-1} + \gamma P^{l,m}\hat{f}^{l-1,m-1}, & \text{for } l+m > N. \end{cases} \tag{2.2}$$

The coefficients α, β and γ, together with the choice of prolongation operator $P^{l,m}$, define the combination scheme. In Section 2.3, it will be shown that the choice $\alpha = \beta = 1, \gamma = -1$ causes a number of error terms to cancel, leading to a representation error of the desired order, $E^{N,N} = \mathcal{O}(h^2 \log h^{-1})$. We denote this choice by the $[1, 1, -1]$ scheme. Likewise, $[\frac{1}{2}, \frac{1}{2}, 0]$ and $[0, 0, -1]$ schemes are considered.

The *local error* $e^{l,m}$ is defined according to

$$e^{l,m} \equiv \alpha P^{l,m} f^{l-1,m} + \beta P^{l,m} f^{l,m-1} + \gamma P^{l,m} f^{l-1,m-1} - f^{l,m}, \tag{2.3}$$

in terms of which the following recursive relation for $E^{l,m}$ is obtained

$$E^{l,m} = e^{l,m} + \alpha P^{l,m} E^{l-1,m} + \beta P^{l,m} E^{l,m-1} + \gamma P^{l,m} E^{l-1,m-1}. \tag{2.4}$$

Equation (5.9) shows that to find the representation error $E^{N,N}$ we have to find an expression for the local error $e^{l,m}$ and solve $E^{N,N}$ from (5.9) such that $E^{N,N}$ is expressed solely in terms of local errors.

In the remainder of this section, we obtain expressions for the representation error $E^{N,N}$ in terms of local errors $e^{l,m}$ by solving the recursive relation (5.9) for the $[\frac{1}{2}, \frac{1}{2}, 0]$, the $[1, 1, -1]$ and the $[0, 0, -1]$ schemes. Furthermore, it is shown that these schemes can also be replaced by equivalent direct schemes that directly prolongate the coarse representations on $\Omega^{N,0}, \ldots, \Omega^{0,N}, \Omega^{N-1,0}, \ldots, \Omega^{0,N-1}$ onto the finest grid $\Omega^{N,N}$.

2.2.2 The $[\frac{1}{2}, \frac{1}{2}, 0]$ combination scheme

For the $[\frac{1}{2}, \frac{1}{2}, 0]$ *combination scheme*, the recursive relation (5.9) reduces to

$$E^{l,m} = e^{l,m} + \frac{1}{2} P^{l,m} E^{l-1,m} + \frac{1}{2} P^{l,m} E^{l,m-1}. \tag{2.5}$$

Using (2.5) and the fact that $E^{l,m} = 0$ for $l + m \leq N$, we prove the following theorem

Theorem 1 *For $\alpha = \beta = \frac{1}{2}, \gamma = 0$, the sparse-grid representation error on the finest grid is given by*

$$E^{N,N} = \sum_{l=1}^{N} \sum_{m=1}^{l} 2^{l+m-2N} \left(\begin{array}{c} 2N - l - m \\ N - l \end{array} \right) P^{N,N} e^{l,m}. \tag{2.6}$$

Proof:
Assume that

$$E^{N,N} = \sum_{n=0}^{m-1} 2^{-n} \sum_{i=0}^{n} \left(\begin{array}{c} n \\ i \end{array} \right) P^{N,N} e^{N-i,N-n+i} + 2^{-m} \sum_{i=0}^{m} \left(\begin{array}{c} m \\ i \end{array} \right) P^{N,N} E^{N-i,N-m+i} \tag{2.7}$$

holds for a certain m. (Note that it is true for $m = 1$ because then it simply reduces to (2.5).)

Then, by substituting (2.5) into (2.7), we obtain

$$
\begin{aligned}
E^{N,N} = \; & \textstyle\sum_{n=0}^{m-1} 2^{-n} \sum_{i=0}^{n} \begin{pmatrix} n \\ i \end{pmatrix} P^{N,N} e^{N-i,N-n+i} - 2^{-m} \sum_{i=0}^{m} \begin{pmatrix} m \\ i \end{pmatrix} \\
& P^{N,N} e^{N-i,N-m+i} \\
= \; & 2^{-(m+1)} \textstyle\sum_{i=0}^{m} \begin{pmatrix} m \\ i \end{pmatrix} \left(P^{N,N} E^{N-i-1,N-m+i} + P^{N,N} E^{N-i,N-m+i-1}\right) \\
= \; & 2^{-(m+1)} \textstyle\sum_{i=1}^{m+1} \begin{pmatrix} m \\ i-1 \end{pmatrix} P^{N,N} E^{N-i,N-m+i-1} \\
& +2^{-(m+1)} \textstyle\sum_{i=0}^{m} \begin{pmatrix} m \\ i \end{pmatrix} P^{N,N} E^{N-i,N-m+i-1} \\
= \; & 2^{-(m+1)} \textstyle\sum_{i=1}^{m} \left(\begin{pmatrix} m \\ i-1 \end{pmatrix} + \begin{pmatrix} m \\ i \end{pmatrix} \right) P^{N,N} E^{N-i,N-m+i-1} \\
& +2^{-(m+1)} \left(P^{N,N} E^{N-m-1,N-m} + P^{N,N} E^{N,N-m-1}\right) \\
= \; & 2^{-(m+1)} \textstyle\sum_{i=1}^{m} \begin{pmatrix} m+1 \\ i \end{pmatrix} P^{N,N} E^{N-i,N-m+i-1} \\
& +2^{-(m+1)} \left(\begin{pmatrix} m+1 \\ m+1 \end{pmatrix} P^{N,N} E^{N-m-1,N-m} + \begin{pmatrix} m+1 \\ 0 \end{pmatrix} \right. \\
& \left. P^{N,N} E^{N,N-m-1} \right) \\
= \; & 2^{-(m+1)} \textstyle\sum_{i=0}^{m+1} \begin{pmatrix} m+1 \\ i \end{pmatrix} P^{N,N} E^{N-i,N-(m+1)+i}
\end{aligned}
\tag{2.8}
$$

and thus

$$
\begin{aligned}
E^{N,N} = \; & \textstyle\sum_{n=0}^{m} 2^{-n} \sum_{i=0}^{n} \begin{pmatrix} n \\ i \end{pmatrix} P^{N,N} e^{N-i,N-n+i} + 2^{-(m+1)} \sum_{i=0}^{m+1} \begin{pmatrix} m+1 \\ i \end{pmatrix} \\
& P^{N,N} E^{N-i,N-(m+1)+i}.
\end{aligned}
\tag{2.9}
$$

Therefore, if (2.7) holds for m, then it holds for $m+1$ and since it is true for $m = 1$ it follows that (2.7) holds for all $m \geq 1$. Substituting $m = N$ into (2.7) and using the fact that $E^{l,m} = 0$ for $l + m \leq N$, yields

$$
E^{N,N} = \sum_{n=0}^{N-1} 2^{-n} \sum_{i=0}^{n} \begin{pmatrix} n \\ i \end{pmatrix} P^{N,N} e^{N-i,N-n+i},
\tag{2.10}
$$

which, after substituting $l = N - i$ and $m = N - n + i$, yields (2.6).

Theorem 2 *For $\alpha = \beta = \frac{1}{2}, \gamma = 0$, the representation on the finest grid is given by*

$$
\hat{f}^{N,N} = 2^{-N} \sum_{l=0}^{N} \begin{pmatrix} N \\ N-l \end{pmatrix} P^{N,N} f^{l,N-l}.
\tag{2.11}
$$

Proof: Assume that

$$
\hat{f}^{N,N} = 2^{-m} \sum_{i=0}^{m} \begin{pmatrix} m \\ i \end{pmatrix} P^{N,N} \hat{f}^{N-i,N-m+i}
\tag{2.12}
$$

holds for a certain m. (Note that it holds for $m = 1$ because then it reduces to (2.2).) Then, by substituting (2.2) into (2.12), we obtain,

$$\begin{aligned}
\hat{f}^{N,N} &= 2^{-m}\textstyle\sum_{i=0}^{m} \begin{pmatrix} m \\ i \end{pmatrix} \frac{1}{2}\left(p^{N,N}\hat{f}^{N-i-1,N-m+i} + p^{N,N}\hat{f}^{N-i,N-m+i-1}\right) \\
&= 2^{-(m+1)}\left(\textstyle\sum_{i=1}^{m+1} \begin{pmatrix} m \\ i-1 \end{pmatrix} p^{N,N}\hat{f}^{N-i,N-(m+1)+i}\right. \\
&\quad \left. +\textstyle\sum_{i=0}^{m} p^{N,N}\hat{f}^{N-i,N-(m+1)+i}\right) \\
&= 2^{-(m+1)}\textstyle\sum_{i=1}^{m} \begin{pmatrix} m \\ i-1 \end{pmatrix}\begin{pmatrix} m \\ i \end{pmatrix} p^{N,N}\hat{f}^{N-i,N-(m+1)+i} \\
&\quad +p^{N,N}\hat{f}^{N,N-(m+1)} + p^{N,N}\hat{f}^{N-(m+1),N} \\
&= 2^{-(m+1)}\textstyle\sum_{i=0}^{m+1} \begin{pmatrix} m+1 \\ i \end{pmatrix} p^{N,N}\hat{f}^{N-i,N-(m+1)+i}.
\end{aligned} \tag{2.13}$$

Therefore, if (2.12) holds for m, then it holds for $m + 1$ and since it is true for $m = 1$ it follows that (2.12) holds for all $m \geq 1$. Substituting $m = N$ into (2.7) and using the fact that $\hat{f}^{l,m} = f^{l,m}$ for $l + m \leq N$, yields

$$\hat{f}^{N,N} = 2^{-N}\sum_{i=0}^{N} \begin{pmatrix} N \\ i \end{pmatrix} p^{N,N} f^{N-i,i}, \tag{2.14}$$

which is equivalent to (2.11).

2.2.3 The $[1, 1, -1]$ combination scheme

For the $[1, 1, -1]$ *combination scheme*, the recursive relation (5.9) reads

$$E^{l,m} = e^{l,m} + p^{l,m}E^{l-1,m} + p^{l,m}E^{l,m-1} - p^{l,m}E^{l-1,m-1}. \tag{2.15}$$

Using (2.15), we proof the following theorem

Theorem 3 *For $\alpha = \beta = 1, \gamma = -1$, the representation error on the finest grid is given by*

$$E^{N,N} = \sum_{l=1}^{N}\sum_{m=1}^{l} p^{N,N}e^{l,m}. \tag{2.16}$$

Proof Assume that

$$E^{N,N} = \sum_{n=0}^{m-1}\sum_{i=0}^{n} p^{N,N}e^{N-i,N-n+i} + \sum_{i=0}^{m} p^{N,N}E^{N-i,N-m+i} - \sum_{i=1}^{m} p^{N,N}E^{N-i,N-m+i-1} \tag{2.17}$$

holds for a certain m. (Note that it is true for $m = 1$ because then it reduces to (2.15).) Then, by substituting (2.15) into (2.17), we obtain

$$\begin{aligned} E^{N,N} &- \textstyle\sum_{n=0}^{m}\sum_{i=0}^{n} P^{N,N}e^{N-i,N-n+i} \\ &= \textstyle\sum_{i=0}^{m}\big(P^{N,N}E^{N-i-1,N-m+i} + P^{N,N}E^{N-i,N-m+i-1} \\ &\quad -P^{N,N}E^{N-i-1,N-m+i-1}\big) - \textstyle\sum_{i=1}^{m} P^{N,N}E^{N-i,N-m+i-1} \\ &= \textstyle\sum_{i=0}^{m} P^{N,N}E^{N-i-1,N-m+i} - \sum_{i=0}^{m} P^{N,N}E^{N-i-1,N-m+i-1} \\ &\quad +P^{N,N}E^{N,N-m-1} \\ &= \textstyle\sum_{i=1}^{m+1} P^{N,N}E^{N-i,N-m+i-1} - \sum_{i=1}^{m+1} P^{N,N}E^{N-i,N-m+i-2} \\ &\quad +P^{N,N}E^{N,N-m-1}, \end{aligned} \tag{2.18}$$

hence,

$$\begin{aligned} E^{N,N} &= \textstyle\sum_{n=0}^{m}\sum_{i=0}^{n} P^{N,N}e^{N-i,N-n+i} + \sum_{i=0}^{m+1} P^{N,N}E^{N-i,N-(m+1)+i} \\ &\quad - \textstyle\sum_{i=1}^{m+1} P^{N,N}E^{N-i,N-(m+1)+i-1}. \end{aligned} \tag{2.19}$$

Thus, if (2.17) holds for m, then it holds for $m+1$ and since it holds for $m = 1$, it follows that (2.17) holds for all $m \geq 1$. Substituting $m = N$ into (2.17) and using the fact that $E^{l,m} = 0$ for $l + m \leq N$, yields

$$E^{N,N} = \sum_{n=0}^{N-1}\sum_{i=0}^{n} P^{N,N}e^{N-i,N-n+i}, \tag{2.20}$$

which is equivalent to (2.16).

Theorem 4 *For $\alpha = \beta = 1, \gamma = -1$, the representation on the finest grid is given by*

$$\hat{f}^{N,N} = \sum_{l=0}^{N} P^{N,N}f^{l,N-l} - \sum_{l=0}^{N-1} P^{N,N}f^{l,N-1-l}, \tag{2.21}$$

Proof: The proof is given by induction. Assume that

$$\hat{f}^{N,N} = \sum_{i=0}^{m} P^{N,N}f^{N-i,N-m+i} - \sum_{i=0}^{m-1} P^{N,N}f^{N-1-i,N-m+i} \tag{2.22}$$

holds for a certain m. (Note that it holds for $m = 1$ because then it reduces to (2.2).) Then, by substituting (2.2) into (2.22), we obtain

$$\begin{aligned} \hat{f}^{N,N} &= \textstyle\sum_{i=0}^{m}\big(P^{N,N}f^{N-i-1,N-m+i} + P^{N,N}f^{N-i,N-m+i-1} - \\ &\quad P^{N,N}f^{N-i-1,N-m+i-1}\big) - \textstyle\sum_{i=0}^{m-1} P^{N,N}f^{N-1-i,N-m+i} \\ &= \textstyle\sum_{i=0}^{m}\big(P^{N,N}f^{N-i,N-m+i-1} - P^{N,N}f^{N-i-1,N-m+i-1}\big) \\ &\quad +P^{N,N}f^{N-m-1,N} \\ &= \textstyle\sum_{i=0}^{m+1} P^{N,N}f^{N-i,N-(m+1)+i} - \sum_{i=0}^{m} P^{N,N}f^{N-i-1,N-(m+1)+i}. \end{aligned} \tag{2.23}$$

Therefore, if (2.22) holds for m, then it holds for $m+1$ and since it is true for $m = 1$ it follows that (2.22) holds for all $m \geq 1$. Substituting $m = N$ into (2.17) and using the fact that $\hat{f}^{l,m} = f^{l,m}$ for $l + m \leq N$, yields

$$\hat{f}^{N,N} = \sum_{i=0}^{N} P^{N,N} f^{N-i,i} - \sum_{i=0}^{N-1} P^{N,N} f^{N-1-i,i}, \tag{2.24}$$

which is equivalent to (2.21).

2.2.4 The $[0, 0, 1]$ combination scheme

For $\alpha = \beta = 0, \gamma = 1$, the recursive relation (5.9) reduces to

$$E^{l,m} = e^{l,m} + P^{l,m} E^{l-1,m-1}. \tag{2.25}$$

It is straightforward to show that (2.25) leads to

$$E^{N,N} = \sum_{l=\lceil N/2 \rceil+1}^{N} P^{N,N} e^{l,l}, \tag{2.26}$$

and to

$$\hat{f}^{N,N} = P^{N,N} f^{\lceil N/2 \rceil, \lceil N/2 \rceil}, \tag{2.27}$$

where $\lceil N/2 \rceil$ denotes the integer part of $N/2$.

2.2.5 Discussion

In the current section, the representation error $E^{N,N}$ was expressed in terms of the local errors $e^{l,m}$ for the $[\frac{1}{2}, \frac{1}{2}, 0]$, the $[1, 1, -1]$ and the $[0, 0, -1]$ schemes; see equations (2.6), (2.16) and (2.26), respectively. Furthermore, expressions (2.11), (2.21) and (2.27) were obtained. They express the representation $\hat{f}^{N,N}$ directly in terms of the coarse representations $f^{N,0}, f^{N-1,1}, \ldots, f^{0,N}$ and $f^{N-1,0}, f^{N-2,1}, \ldots, f^{0,N-1}$. Equation (2.21) corresponds to the combination technique as introduced in [1]. Inspection of (2.21) shows that the combination technique can be viewed as an extrapolation technique, see [5] and [4] for discussions of the combination technique from the extrapolation point of view. Note that for the $[1, 1, -1]$ scheme, the expression for the representation error (2.16) simply states that the representation error $E^{N,N}$ is equal to the sum of the local errors on the grids $\Omega^{l,m}$ satisfying $N > l + m \leq 2N$ (the lower-right half of the grid of grids depicted in Figure 3.1).

2.3 Local errors

We now turn to analyzing the local error $e^{l,m}$ for two-dimensional functions f, i.e., we will determine the error that we make when we approximate a grid function $f^{l,m}$ by the combination

$$\alpha P^{l,m} f^{l-1,m} + \beta P^{l,m} f^{l,m-1} + \gamma P^{l,m} f^{l-1,m-1}. \tag{2.28}$$

In Figure 2.2, corresponding sections from the grids $\Omega^{l-1,m}$, $\Omega^{l,m-1}$, $\Omega^{l-1,m-1}$ and $\Omega^{l,m}$ are shown. The squares mark locations for which function values are defined on $\Omega^{l-1,m-1}$. Likewise, the circles and the diamonds belong to $\Omega^{l-1,m}$ and $\Omega^{l,m-1}$, respectively. The cross ($\times$) represents the location of the cell center, on $\Omega^{l,m}$, at which the combination (2.28) will be generated. For the prolongations $P^{l,m}f^{l-1,m}$, $P^{l,m}f^{l,m-1}$ and $P^{l,m}f^{l-1,m-1}$, we take linear combinations of the function values on grids $\Omega^{l-1,m}$, $\Omega^{l,m-1}$ and $\Omega^{l-1,m-1}$, respectively, i.e.,

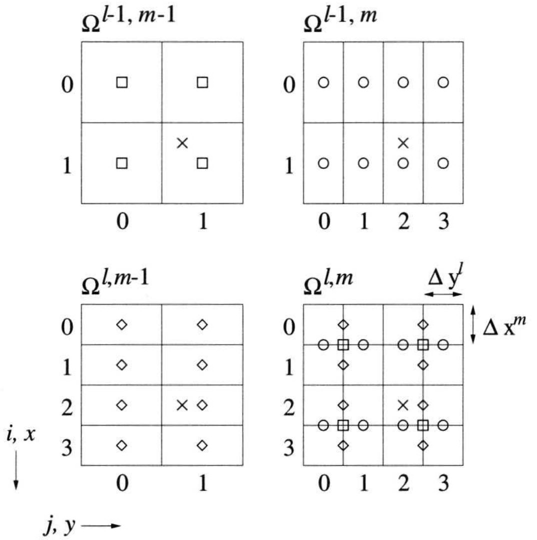


Figure 2.2: Sections of grids involved in combination

$$\left(P^{l,m} f^{l',m'}\right)_{i_\times,j_\times} = \sum_{i',j'} \psi^{l'-l,m'-m}_{i',j'} f^{l',m'}_{i',j'} . \tag{2.29}$$

Note that in Figure 2.2 both $i_\times$ and $j_\times$ are even; the $\psi^{l'-l,m'-m}_{i',j'}$ in (2.29) also correspond to this case, the dependence of the $\psi^{l'-l,m'-m}_{i',j'}$ on $i_\times$ and $j_\times$ is suppressed in the notation. The function values $f^{l',m'}_{i',j'}$ at positions $(x^{m'}_{i'}, y^{l'}_{j'})$, corresponding to the squares, circles and diamonds, are expressed as Taylor series taken at the location of the cross ($\times$), yielding

$$f^{l',m'}_{i',j'} = \sum_{p=0}^{\infty}\sum_{q=0}^{\infty} \left(X^{l'-l,m'-m}_{i',j'} \frac{\Delta x^{l'}}{2}\right)^{p} \left(Y^{l'-l,m'-m}_{i',j'} \frac{\Delta y^{m'}}{2}\right)^{q} \frac{\partial_x^p \partial_y^q f^{l',m'}_{i_\times,j_\times}}{p!q!}, \tag{2.30}$$

where

$$\begin{aligned}
\left[X^{-1,-1}\right] &= \left[Y^{-1,-1}\right]^T = \begin{pmatrix} -3 & -3 \\ 1 & 1 \end{pmatrix}, \\
\left[X^{-1,0}\right] &= \left[Y^{0,-1}\right]^T = \begin{pmatrix} -4 & -4 \\ -2 & -2 \\ 0 & 0 \\ 2 & 2 \end{pmatrix}, \\
\left[X^{0,-1}\right] &= \left[Y^{-1,0}\right]^T = \begin{pmatrix} -3 & -3 & -3 & -3 \\ 1 & 1 & 1 & 1 \end{pmatrix}.
\end{aligned} \tag{2.31}$$

Note that $X^{l'-l,m'-m}_{i',j'}$ and $Y^{l'-l,m'-m}_{i',j'}$ are scalars; they are elements of the matrices $\left[X^{l'-l,m'-m}\right]$ and $\left[Y^{l'-l,m'-m}\right]$, respectively. The indices on the matrix elements start at zero, i.e.,

$$[A] = \begin{pmatrix} A_{0,0} & A_{0,1} & \cdots \\ A_{1,0} & A_{1,1} & \cdots \\ \vdots & \vdots & \ddots \end{pmatrix}.$$

Again, the matrices $\left[X^{l'-l,m'-m}\right]$ and $\left[Y^{l'-l,m'-m}\right]$, as given by (2.31), are valid when $i_\times$ and $j_\times$ are both even, as in Figure 2.2. Combining equations (2.3), (2.29) and (2.30), the following expression for the local error is obtained,

$$e^{l,m}_{i_\times,j_\times} = -f^{l,m}_{i_\times,j_\times} + \sum_{p=0}^{\infty}\sum_{q=0}^{\infty} \phi_{p,q} \left(\frac{(-1)^{i_\times}\Delta x^l}{2}\right)^p \left(\frac{(-1)^{j_\times}\Delta y^m}{2}\right)^q \frac{\partial_x^p \partial_y^q f^{l,m}_{i_\times,j_\times}}{p!q!}, \tag{2.32}$$

$$\begin{aligned}
\phi_{p,q} \equiv{}& \alpha \textstyle\sum_{i=0}^{3}\sum_{j=0}^{1} \psi^{-1,0}_{i,j} \left(X^{-1,0}_{i,j}\right)^p \left(Y^{-1,0}_{i,j}\right)^q \\
+{}& \beta \textstyle\sum_{i=0}^{1}\sum_{j=0}^{3} \psi^{0,-1}_{i,j} \left(X^{0,-1}_{i,j}\right)^p \left(Y^{0,-1}_{i,j}\right)^q \\
+{}& \gamma \textstyle\sum_{i=0}^{1}\sum_{j=0}^{1} \psi^{-1,-1}_{i,j} \left(X^{-1,-1}_{i,j}\right)^p \left(Y^{-1,-1}_{i,j}\right)^q.
\end{aligned} \tag{2.33}$$

The factors $(-1)^{i_\times}$ and $(-1)^{j_\times}$ have been inserted to ensure that (2.32) is valid for arbitrary $i_\times$ and $j_\times$ while $\left[\psi^{l'-l,m'-m}\right]$, $\left[X^{l'-l,m'-m}\right]$ and $\left[Y^{l'-l,m'-m}\right]$ are taken to correspond to even $i_\times$ and $j_\times$. We refer to (2.32) as the *error expansion*. We will now work out the error coefficients $\phi_{p,q}$ for two specific prolongations, i.e., for specific choices of the interpolation weights $\psi^{l',m'}_{i',j'}$.

2.3.1 Piecewise-constant interpolation

For the prolongations, the simplest choice is piecewise-constant interpolation, which amounts to taking

$$\left[\psi^{-1,-1}\right] = \begin{pmatrix} 0 & 0 \\ 0 & 1 \end{pmatrix}, \quad \left[\psi^{0,-1}\right] = \left[\psi^{-1,0}\right]^T = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2.34}$$

From (2.33), we find that this leads to

$$[\phi] = \begin{pmatrix} \alpha+\beta+\gamma & \alpha+\gamma & \alpha+\gamma & \cdots \\ \beta+\gamma & \gamma & \gamma & \cdots \\ \beta+\gamma & \gamma & \gamma & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{2.35}$$

and therefore, according to the error expansion (2.32), to

$$\begin{aligned} e^{l,m}_{i_\times,j_\times} &= (\alpha+\beta+\gamma-1)\, f^{l,m}_{i_\times,j_\times} + (\beta+\gamma) \textstyle\sum_{p=1}^{\infty} \left((-1)^{i_\times} \frac{\Delta x^l}{2}\right)^p \frac{\partial_x^p f^{l,m}_{i_\times,j_\times}}{p!} \\ &+ (\alpha+\gamma) \textstyle\sum_{q=1}^{\infty} \left((-1)^{j_\times} \frac{\Delta y^m}{2}\right)^q \frac{\partial_y^q f^{l,m}_{i_\times,j_\times}}{q!} \\ &+ \gamma \textstyle\sum_{p=1}^{\infty} \sum_{q=1}^{\infty} \left((-1)^{i_\times} \frac{\Delta x^l}{2}\right)^p \\ &\quad \left((-1)^{j_\times} \frac{\Delta y^m}{2}\right)^q \frac{\partial_x^p \partial_y^q f^{l,m}_{i_\times,j_\times}}{p!q!}. \end{aligned} \tag{2.36}$$

From (2.36), it is apparent that, to obtain consistency, $\alpha+\beta+\gamma = 1$ must hold.

The $[\frac{1}{2}, \frac{1}{2}, 0]$ piecewise-constant scheme.

For a combination scheme that requires representation on only a single level of grids, either $\gamma = 0$ or $\alpha = \beta = 0$ must hold. In principle, the choice $\gamma = 0$ leaves us the freedom of choosing α and β, provided they satisfy $\alpha + \beta = 1$. However, we only consider the choice $\alpha = \beta = \frac{1}{2}$. This choice is not completely arbitrary; it provides a symmetric dependence of the local error $e^{l,m}$ on Δx^l and Δy^m. We thus obtain the $[\frac{1}{2}, \frac{1}{2}, 0]$ *piecewise-constant scheme*, to which corresponds the following local error

$$e^{l,m}_{i_\times,j_\times} = \frac{1}{2} \sum_{p=1}^{\infty} \left((-1)^{i_\times} \frac{\Delta x^l}{2}\right)^p \frac{\partial_x^p f^{l,m}_{i_\times,j_\times}}{p!} + \frac{1}{2} \sum_{q=1}^{\infty} \left((-1)^{j_\times} \frac{\Delta y^m}{2}\right)^q \frac{\partial_y^q f^{l,m}_{i_\times,j_\times}}{q!}. \tag{2.37}$$

Using (2.37) and (2.29), we obtain the following for $\left\| P^{N,N} e^{l,m} \right\|_\infty$

$$\begin{aligned} \left\| P^{N,N} e^{l,m} \right\|_\infty &= \Big\| \tfrac{1}{2} \textstyle\sum_{p=1}^{\infty} \frac{1}{p!} \left(\frac{\Delta x^l}{2}\right)^p \sum_{i',j'} \psi^{l-N,m-N}_{i',j'} (-1)^{i'p} \partial_x^p f^{l,m}_{i',j'} \\ &\quad + \tfrac{1}{2} \textstyle\sum_{q=1}^{\infty} \frac{1}{q!} \left(\frac{\Delta y^m}{2}\right)^q \sum_{i',j'} \psi^{l-N,m-N}_{i',j'} (-1)^{j'q} \partial_y^q f^{l,m}_{i',j'} \Big\|_\infty \\ &\leq \tfrac{1}{2} \textstyle\sum_{p=1}^{\infty} \frac{1}{p!} \left(\frac{\Delta x^l}{2}\right)^p \left\| \partial_x^p f \right\|_\infty + \frac{1}{2} \sum_{q=1}^{\infty} \frac{1}{q!} \left(\frac{\Delta y^m}{2}\right)^q \left\| \partial_y^q f \right\|_\infty \\ &= \tfrac{1}{2} \textstyle\sum_{p=1}^{\infty} \frac{1}{p!} \left\{ \left(\frac{\Delta x^l}{2}\right)^p \left\| \partial_x^p f \right\|_\infty + \left(\frac{\Delta y^m}{2}\right)^p \left\| \partial_y^p f \right\|_\infty \right\}. \end{aligned} \tag{2.38}$$

To obtain the desired expression for $E^{N,N}$, equation (2.38) is now substituted into (2.6), yielding

$$\begin{aligned} \left\| E^{N,N} \right\|_\infty \leq\ & \tfrac{1}{2} \textstyle\sum_{p=1}^{\infty} \frac{1}{p!} \sum_{l=1}^{N} \sum_{m=1}^{l} 2^{l+m-2N} \begin{pmatrix} 2N-l-m \\ N-l \end{pmatrix} \\ & \left\{ \left(\frac{\Delta x^l}{2}\right)^p \left\| \partial_x^p f \right\|_\infty + \left(\frac{\Delta y^m}{2}\right)^p \left\| \partial_y^p f \right\|_\infty \right\}. \end{aligned} \tag{2.39}$$

Since the grids $\Omega^{l,m}$ extend over the unit square, we can write $\Delta x^l = 2^{-l}$ and $\Delta y^m = 2^{-m}$. Substitution of these relations into (2.39) gives

$$\begin{aligned} \left\|E^{N,N}\right\|_\infty \;\leq\; & \tfrac{1}{2}\textstyle\sum_{p=1}^{\infty} \tfrac{2^{-p}}{p!} \sum_{l=1}^{N} \sum_{m=0}^{l} 2^{l+m-2N} \begin{pmatrix} 2N-l-m \\ N-l \end{pmatrix} \\ & \left\{ 2^{-lp} \left\|\partial_x^p f\right\|_\infty + 2^{-mp} \left\|\partial_y^p f\right\|_\infty \right\}. \end{aligned} \tag{2.40}$$

Performing the summations over l and m yields

$$\left\|E^{N,N}\right\|_\infty \leq \sum_{p=1}^{\infty} \frac{2^{-(N+1)p}}{p!} \frac{1-2^{-N}\left(2^p+1\right)^N}{1-2^p} \left\{ \left\|\partial_x^p f\right\|_\infty + \left\|\partial_y^p f\right\|_\infty \right\}. \tag{2.41}$$

Writing out the first few terms of this error expansion gives

$$\begin{aligned} \left\|E^{N,N}\right\|_\infty \;\leq\; & \tfrac{1}{2}\left(\left(\tfrac{3}{4}\right)^N - \left(\tfrac{1}{2}\right)^N\right) \left\{ \|\partial_x f\|_\infty + \|\partial_y f\|_\infty \right\} \\ & + \tfrac{1}{24}\left(\left(\tfrac{5}{8}\right)^N - \left(\tfrac{1}{4}\right)^N\right) \left\{ \|\partial_x^2 f\|_\infty + \left\|\partial_y^2 f\right\|_\infty \right\} + \cdots, \end{aligned} \tag{2.42}$$

so, to leading order,

$$\left\|E^{N,N}\right\|_\infty \leq \frac{1}{2}\left(\frac{3}{4}\right)^N \left\{ \|\partial_x f\|_\infty + \|\partial_y f\|_\infty \right\} + \mathcal{O}\left(\left(\frac{5}{8}\right)^N\right). \tag{2.43}$$

On the finest grid $\Omega^{N,N}$, the mesh widths in x- and y-directions are identical, $h = \Delta x^N = \Delta y^N = 2^{-N}$. Rewriting (2.43) in terms of this mesh width yields

$$\left\|E^{N,N}\right\|_\infty \leq \frac{1}{2} h^{(2-\log_2 3)} \left\{ \|\partial_x f\|_\infty + \|\partial_y f\|_\infty \right\} + \mathcal{O}\left(h^{(3-\log_2 5)}\right). \tag{2.44}$$

Equation (2.44) shows that the $[\frac{1}{2}, \frac{1}{2}, 0]$ piecewise-constant scheme has a representation error of order $2 - \log_2 3 \approx 0.42$.

As a test of the above derivations, we examine the simple case $f(x, y) = x + y$. This case is particularly attractive because it allows us to obtain an explicit expression for $\|E^{N,N}\|_\infty$ (in contrast to an upper bound). For $f(x, y) = x + y$, equation (2.37) reduces to

$$e_{i_\times, j_\times}^{l,m} = \frac{1}{4}\left((-1)^{i_\times} \Delta x^l + (-1)^{j_\times} \Delta y^m\right) \tag{2.45}$$

and thus

$$\begin{aligned} E_{i_\times, j_\times}^{N,N} \;=\; & \tfrac{1}{4} \textstyle\sum_{l=1}^{N} \sum_{m=1}^{l} 2^{2N-l-m} \begin{pmatrix} 2N-l-m \\ N-l \end{pmatrix} \sum_{i',j'} \psi_{i',j'}^{l-N,m-N} \\ & \left((-1)^{i'} 2^{-l} + (-1)^{j'} 2^{-m}\right). \end{aligned} \tag{2.46}$$

For piecewise-constant interpolation

$$\psi_{i',j'}^{l-N,m-N} = \delta_{\lceil i_\times 2^{l-N} \rceil - i', \lceil j_\times 2^{m-N} \rceil - j'}, \tag{2.47}$$

where δ is the Kronecker delta. Using (2.47), we transform (2.46) into

$$\begin{aligned} E^{N,N}_{i_\times,j_\times} &= \tfrac{1}{4} \textstyle\sum_{l=1}^{N} \sum_{m=0}^{l} 2^{l+m-2N} \begin{pmatrix} 2N-l-m \\ N-l \end{pmatrix} \\ &\quad \left((-1)^{\lceil i_\times 2^{l-N} \rceil} 2^{-l} + (-1)^{\lceil j_\times 2^{m-N} \rceil} 2^{-m} \right). \end{aligned} \tag{2.48}$$

This expression is maximal for $i_\times = j_\times = 0$, thus

$$\begin{aligned} \left\|E^{N,N}\right\|_\infty &= \tfrac{1}{4} \textstyle\sum_{l=1}^{N} \sum_{m=0}^{l} 2^{l+m-2N} \begin{pmatrix} 2N-l-m \\ N-l \end{pmatrix} \left(2^{-l} + 2^{-m}\right) \\ &= \left(\tfrac{3}{4}\right)^N - \left(\tfrac{1}{2}\right)^N. \end{aligned} \tag{2.49}$$

Numerical tests show that, for $f(x,y) = x + y$, the error $\left\|E^{N,N}\right\|_\infty$ is indeed exactly given by $\left(\frac{3}{4}\right)^N - \left(\frac{1}{2}\right)^N$.

$[1, 1, -1]$ **piecewise-constant scheme.**

Equation (2.36) reveals that when we take $\alpha + \gamma = \beta + \gamma = 0$, the error terms that depend only on Δx or only on Δy vanish. Combining these requirements with $\alpha + \beta + \gamma = 1$ gives

$$\alpha = \beta = 1, \quad \gamma = -1. \tag{2.50}$$

This choice of α, β and γ constitutes the $[1, 1, -1]$ *combination scheme*. For the present case, $[1, 1, -1]$ *combination with piecewise-constant interpolation*, equation (2.36) yields

$$e^{l,m}_{i_\times,j_\times} = -\sum_{p=1}^{\infty} \sum_{q=1}^{\infty} \left((-1)^{i_\times} \frac{\Delta x}{2} \right)^p \left((-1)^{j_\times} \frac{\Delta y}{2} \right)^q \frac{\partial_x^p \partial_y^q f^{l,m}_{i_\times,j_\times}}{p!q!} \tag{2.51}$$

and thus

$$\left\| P^{N,N} e^{l,m} \right\|_\infty \leq \sum_{p=1}^{\infty} \sum_{q=1}^{\infty} \frac{1}{p!q!} \left(\frac{\Delta x}{2} \right)^p \left(\frac{\Delta y}{2} \right)^q \left\| \partial_x^p \partial_y^q f \right\|_\infty. \tag{2.52}$$

Substitution of (2.52) into (2.16) yields

$$\left\| E^{N,N} \right\|_\infty \leq \sum_{p=1}^{\infty} \sum_{q=1}^{\infty} \frac{2^{-p-q}}{p!q!} \left\| \partial_x^p \partial_y^q f \right\|_\infty \sum_{l=1}^{N} \sum_{m=1}^{l} 2^{-lp-mq}. \tag{2.53}$$

Asymptotically, this yields

$$\left\| E^{N,N} \right\|_\infty \leq \frac{1}{4} \left(\frac{1}{2} \right)^N N \left\| \partial_x \partial_y f \right\|_\infty + \mathcal{O}\left(\left(\frac{1}{2} \right)^N \right), \tag{2.54}$$

in terms of the mesh width h, this becomes

$$\left\|E^{N,N}\right\|_{\infty} \leq \frac{1}{4} h \log_2 h^{-1} \left\|\partial_x \partial_y f\right\|_{\infty} + \mathcal{O}\left(h\right). \tag{2.55}$$

Thus, the $[1, 1, -1]$ piecewise-constant scheme has a representation error of order $h \log_2 h^{-1}$.

Again we examine a simple test case, viz. $f(x, y) = xy$, which yields

$$\left\|E^{N,N}\right\|_{\infty} = \frac{1}{4}\left(\left(\frac{1}{2}\right)^{N} N - \left(\frac{1}{2}\right)^{N} + \left(\frac{1}{4}\right)^{N}\right). \tag{2.56}$$

Numerical results confirm that representation of $f(x, y) = xy$ by the $[1, 1, -1]$ piecewise-constant scheme agrees with (2.56) within machine accuracy.

$[0, 0, 1]$ **piecewise-constant scheme.**

We now consider the choice $\alpha = \beta = 0$, $\gamma = 1$. This choice does not represent a real sparse-grid combination scheme because it constructs $\hat{f}^{N,N}$ from only a single coarse grid-function, e.g., from $f^{\lceil N/2 \rceil, \lceil N/2 \rceil}$. Yet, we do include the $[0, 0, 1]$ scheme for comparison, in particular with the $[\frac{1}{2}, \frac{1}{2}, 0]$ scheme. We make this comparison because Hemker [2] pointed out that direct prolongation of $f^{\lceil N/2 \rceil, \lceil N/2 \rceil}$ should be superior to the $[\frac{1}{2}, \frac{1}{2}, 0]$ scheme. It will turn out that this is indeed true. The $[0, 0, 1]$ piecewise-constant local error is given by

$$\begin{aligned} e^{l,m}_{i_\times, j_\times} \quad = \quad & \textstyle\sum_{p=1}^{\infty} \left((-1)^{i_\times} \frac{\Delta x^l}{2}\right)^p \frac{\partial_x^p f^{l,m}_{i_\times, j_\times}}{p!} + \sum_{q=1}^{\infty} \left((-1)^{j_\times} \frac{\Delta y^m}{2}\right)^q \frac{\partial_y^q f^{l,m}_{i_\times, j_\times}}{q!} \\ & \textstyle - \sum_{p=1}^{\infty} \sum_{q=1}^{\infty} \left((-1)^{i_\times} \frac{\Delta x^l}{2}\right)^p \left((-1)^{j_\times} \frac{\Delta y^m}{2}\right)^q \frac{\partial_x^p \partial_y^q f^{l,m}_{i_\times, j_\times}}{p! q!}, \end{aligned} \tag{2.57}$$

therefore,

$$\begin{aligned} \left\|P^{N,N} e^{l,m}\right\|_{\infty} \quad \leq \quad & \textstyle\sum_{p=1}^{\infty} \frac{1}{p!} \left(\frac{\Delta x^l}{2}\right)^p \left\|\partial_x^p f\right\|_{\infty} + \sum_{q=1}^{\infty} \frac{1}{q!} \left(\frac{\Delta y^m}{2}\right)^q \left\|\partial_y^q f\right\|_{\infty} \\ & \textstyle + \sum_{p=1}^{\infty} \sum_{q=1}^{\infty} \frac{1}{p! q!} \left(\frac{\Delta x^l}{2}\right)^p \left(\frac{\Delta y^m}{2}\right)^q \left\|\partial_x^p \partial_y^q f\right\|_{\infty}. \end{aligned} \tag{2.58}$$

Substitution of (2.58) into (2.26) yields

$$\begin{aligned} \left\|E^{N,N}\right\|_{\infty} \quad \leq \quad & \textstyle\sum_{p=1}^{\infty} \frac{2^{-Np}}{p!} \frac{1 - 2^{p(N/2+1)}}{1 - 2^p} \left\{\left\|\partial_x^p f\right\|_{\infty} + \left\|\partial_y^p f\right\|_{\infty}\right\} \\ & \textstyle + \sum_{p=1}^{\infty} \sum_{q=1}^{\infty} \frac{2^{-N(p+q)}}{p! q!} \frac{1 - 2^{(p+q)(N/2+1)}}{1 - 2^{p+q}} \left\|\partial_x^p \partial_y^q f\right\|_{\infty}, \end{aligned} \tag{2.59}$$

or, asymptotically,

$$\left\|E^{N,N}\right\|_{\infty} \leq 2 \left(2^{-1/2}\right)^{N} \left\{\left\|\partial_x f\right\|_{\infty} + \left\|\partial_y f\right\|_{\infty}\right\} + \mathcal{O}\left(\left(\frac{1}{2}\right)^{N}\right). \tag{2.60}$$

In terms of the mesh width h, this reads

$$\left\| E^{N,N} \right\|_\infty \leq 2h^{1/2} \left\{ \|\partial_x f\|_\infty + \|\partial_y f\|_\infty \right\} + \mathcal{O}(h) . \tag{2.61}$$

We see that, for piecewise-constant interpolation, the $[0,0,1]$ scheme has a representation error of order $\frac{1}{2}$, which is superior to the order $2 - \log_2 3 \approx 0.42$ for the $[\frac{1}{2}, \frac{1}{2}, 0]$ piecewise-constant scheme.

2.3.2 Piecewise bi-linear interpolation

Next, we consider bi-linear interpolation as a means of prolongation. The prolongations are therefore described by the following interpolation weights

$$\left[\psi^{-1,-1}\right] = \begin{pmatrix} \frac{1}{16} & \frac{3}{16} \\ \\ \frac{3}{16} & \frac{9}{16} \end{pmatrix}, \quad \left[\psi^{0,-1}\right] = \left[\psi^{-1,0}\right]^T = \begin{pmatrix} 0 & 0 & \frac{1}{4} & 0 \\ \\ 0 & 0 & \frac{3}{4} & 0 \end{pmatrix}, \tag{2.62}$$

leading to the following error coefficients

$$\begin{aligned} [\phi] &= \begin{pmatrix} \alpha+\beta+\gamma & 0 & (\beta+\gamma)\lambda_{0,2} & (\beta+\gamma)\lambda_{0,3} & \cdots \\ 0 & 0 & 0 & 0 & \cdots \\ (\alpha+\gamma)\lambda_{2,0} & 0 & \gamma\lambda_{2,2} & \gamma\lambda_{2,3} & \cdots \\ (\alpha+\gamma)\lambda_{3,0} & 0 & \gamma\lambda_{3,2} & \gamma\lambda_{3,3} & \cdots \\ \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix}, \\ \lambda_{p,q} &= \frac{(-3)^p+3}{4}\frac{(-3)^q+3}{4} \end{aligned} \tag{2.63}$$

and the following local error expansion

$$\begin{aligned} e^{l,m}_{i_\times,j_\times} &= (\alpha+\beta+\gamma-1)\, f^{l,m}_{i_\times,j_\times} + (\alpha+\gamma)\textstyle\sum_{p=2}^{\infty} \lambda_{p,0} \\ &\quad \left((-1)^{i_\times}\frac{\Delta x^l}{2}\right)^p \frac{\partial_x^p f^{l,m}_{i_\times,j_\times}}{p!} \\ &+ (\beta+\gamma)\textstyle\sum_{q=2}^{\infty} \lambda_{0,q} \left((-1)^{j_\times}\frac{\Delta y^m}{2}\right)^q \frac{\partial_y^q f^{l,m}_{i_\times,j_\times}}{q!} \\ &+ \gamma \textstyle\sum_{p=2}^{\infty}\sum_{q=2}^{\infty} \lambda_{p,q} \\ &\quad \left((-1)^{i_\times}\frac{\Delta x^l}{2}\right)^p \left((-1)^{j_\times}\frac{\Delta y^m}{2}\right)^q \frac{\partial_x^p \partial_y^q f^{l,m}_{i_\times,j_\times}}{p!q!}, \\ \lambda_{p,q} = \frac{(-3)^p+3}{4}\frac{(-3)^q+3}{4}. \end{aligned} \tag{2.64}$$

Again, for consistency, we must have $\alpha + \beta + \gamma = 1$.

$[\frac{1}{2}, \frac{1}{2}, 0]$ piecewise-bi-linear scheme.

For bi-linear interpolation, the choice $\alpha = \beta = \frac{1}{2}, \gamma = 0$ gives the following expansion for the local error

$$\begin{aligned} e^{l,m}_{i_\times,j_\times} &= \tfrac{1}{2}\textstyle\sum_{p=2}^{\infty} \frac{(-3)^p+3}{4} \left((-1)^{i_\times} \frac{\Delta x^l}{2}\right)^p \frac{\partial_x^p f^{l,m}_{i_\times,j_\times}}{p!} + \tfrac{1}{2}\textstyle\sum_{q=2}^{\infty} \frac{(-3)^q+3}{4} \\ &\quad \left((-1)^{j_\times} \frac{\Delta y^m}{2}\right)^q \frac{\partial_y^q f^{l,m}_{i_\times,j_\times}}{q!}. \end{aligned} \tag{2.65}$$

We write the first terms of the summations separately, yielding

$$\begin{aligned} e^{l,m}_{i_\times,j_\times} &= \tfrac{3}{16}\left\{\left(\Delta x^l\right)^2 \partial_x^2 f^{l,m}_{i_\times,j_\times} + (\Delta y^m)^2 \partial_y^2 f^{l,m}_{i_\times,j_\times}\right\} + \tfrac{1}{2}\textstyle\sum_{p=3}^{\infty} \frac{(-3)^p+3}{4p!} \\ &\quad \left\{\left((-1)^{i_\times} \frac{\Delta x^l}{2}\right)^p \partial_x^p f^{l,m}_{i_\times,j_\times} + \left((-1)^{j_\times} \frac{\Delta y^m}{2}\right)^p \partial_y^q f^{l,m}_{i_\times,j_\times}\right\}. \end{aligned} \tag{2.66}$$

For the prolongation of the local error we obtain

$$\begin{aligned} P^{N,N} e^{l,m} &= \tfrac{3}{16}\textstyle\sum_{i',j'} \psi^{l-N,m-N}_{i',j'} \left\{\left(\Delta x^l\right)^2 \partial_x^2 f^{l,m}_{i',j'} + (\Delta y^m)^2 \partial_y^2 f^{l,m}_{i',j'}\right\} \\ &\quad + \tfrac{1}{2}\textstyle\sum_{p=3}^{\infty} \frac{(-3)^p+3}{4p!} \textstyle\sum_{i',j'} \psi^{l-N,m-N}_{i',j'} \\ &\quad \left\{\left((-1)^{i'} \frac{\Delta x^l}{2}\right)^p \partial_x^p f^{l,m}_{i',j'} + \left((-1)^{j'} \frac{\Delta y^m}{2}\right)^p \partial_y^q f^{l,m}_{i',j'}\right\} \\ &= \tfrac{3}{16}\left\{\left(\Delta x^l\right)^2 \partial_x^2 f^{N,N} + (\Delta y^m)^2 \partial_y^2 f^{N,N}\right\} \\ &\quad + \mathcal{O}\left(\left(\Delta x^l\right)^3 + (\Delta y^m)^3\right). \end{aligned} \tag{2.67}$$

In obtaining (2.67), use has been made of the following property of bi-linear interpolation

$$\sum_{i',j'} \psi^{l-N,m-N}_{i',j'} f^{l,m} = P^{N,N} f^{l,m} = f^{N,N} + \mathcal{O}\left(\left(\Delta x^l\right)^2 + (\Delta y^m)^2\right). \tag{2.68}$$

Substitution of (2.67) into (2.6) yields

$$E^{N,N} = \frac{1}{8}\left(\frac{5}{8}\right)^N \left\{\partial_x^2 f^{N,N} + \partial_y^2 f^{N,N}\right\} + \mathcal{O}\left(\left(\frac{9}{16}\right)^N\right), \tag{2.69}$$

or, in terms of the mesh width h,

$$E^{N,N} = \frac{1}{8} h^{(3-\log_2 5)} \left\{\partial_x^2 f^{N,N} + \partial_y^2 f^{N,N}\right\} + \mathcal{O}\left(h^{(4-\log_2 9)}\right). \tag{2.70}$$

Thus, the $\frac{1}{2}, \frac{1}{2}$-bi-linear combination scheme has a representation error of order $3 - \log_2 5 \approx 0.68$.

$[1,1,-1]$ **piecewise-bi-linear scheme.**

Just as for the piecewise-constant case, taking $\alpha+\gamma=\beta+\gamma=0$ removes the error terms that depend only on Δx or only on Δy. So, again the choice $\alpha=\beta=1, \gamma=-1$ raises the order of the local error. For this choice we obtain

$$\begin{aligned} e^{l,m}_{i_\times,j_\times} &= -\sum_{p=2}^{\infty}\sum_{q=2}^{\infty} \frac{(-3)^p+3}{4}\,\frac{(-3)^q+3}{4}\left((-1)^{i_\times}\frac{\Delta x^l}{2}\right)^p \\ &\quad \left((-1)^{j_\times}\frac{\Delta y^m}{2}\right)^q \frac{\partial_x^p\partial_y^q f^{l,m}_{i_\times,j_\times}}{p!q!}. \end{aligned} \tag{2.71}$$

Substitution of (2.71) into (2.16) yields

$$\begin{aligned} E^{N,N} &= \sum_{p=2}^{\infty}\sum_{q=2}^{\infty}\frac{2^{-p-q}}{p!q!}\,\frac{(-3)^p+3}{4}\,\frac{(-3)^q+3}{4} \\ &\quad \sum_{l=1}^{N}\sum_{m=1}^{l} 2^{-lp-mq}\sum_{i_\times,j_\times}\psi^{l-N,m-N}_{i_\times,j_\times}(-1)^{i_\times p+j_\times q}\partial_x^p\partial_y^q f^{l,m}_{i_\times,j_\times}, \end{aligned} \tag{2.72}$$

which, in leading order, can be written as

$$E^{N,N} = -\frac{3}{64}\left(\frac{1}{4}\right)^N N\partial_x^2\partial_y^2 f^{N,N} + \mathcal{O}\left(\left(\frac{1}{4}\right)^N\right), \tag{2.73}$$

or, in terms of the mesh width h,

$$E^{N,N} = -\frac{3}{64}h^2\log_2 h^{-1}\partial_x^2\partial_y^2 f^{N,N} + \mathcal{O}\left(h^2\right). \tag{2.74}$$

So, the $[1,1,-1]$-bi-linear scheme has a representation error of order $h^2\log_2 h^{-1}$.

$[0,0,1]$ **piecewise-bi-linear scheme.**

For $\alpha=\beta=0$, $\gamma=1$ and prolongation by bi-linear interpolation we obtain

$$\begin{aligned} P^{N,N}e^{l,m} &= \tfrac{3}{8}\left\{\left(\Delta x^l\right)^2\partial_x^2 f^{N,N} + \left(\Delta y^m\right)^2\partial_y^2 f^{N,N}\right\} \\ &\quad +\mathcal{O}\left(\left(\Delta x^l\right)^3 + \left(\Delta y^m\right)^3 + \left(\Delta x^l\right)^2\left(\Delta y^m\right)^2\right). \end{aligned} \tag{2.75}$$

Substitution of (2.75) into (2.26) yields, asymptotically,

$$E^{N,N} = 2\left(\frac{1}{2}\right)^N\left\{\partial_x^2 f^{N,N} + \partial_y^2 f^{N,N}\right\} + \mathcal{O}\left(\left(\frac{1}{4}\right)^N\right). \tag{2.76}$$

In terms of the mesh width h, this reads

$$E^{N,N} = 2h\left\{\partial_x^2 f^{N,N} + \partial_y^2 f^{N,N}\right\} + \mathcal{O}\left(h^2\right). \tag{2.77}$$

We see that, for bi-linear interpolation, the $[0,0,1]$ scheme has a representation error of order 1, which is superior to the order $3-\log_2 5\approx 0.68$ for the $[\frac{1}{2},\frac{1}{2},0]$ bi-linear scheme.

2.3.3 A numerical test

We now turn to analyzing the representation error, corresponding to the $[1, 1, -1]$ piecewise-bi-linear scheme, for a specific example. We take

$$f(x, y) = \sin(\pi x)\sin(\pi y) \tag{2.78}$$

and compare the numerically observed error with the expression for the leading-order error term (2.73) and with the full error expansion (2.72). According to (2.73), the error corresponding to (2.78) is given by

$$E^{N,N} = -\frac{3\pi^4}{64}\left(\frac{1}{4}\right)^N N\sin(\pi x_i^N)\sin(\pi y_j^N) + \mathcal{O}\left(\left(\frac{1}{4}\right)^N\right). \tag{2.79}$$

In Figure 2.3, the solid line represents the analytical result (2.79) for the leading-order error term, the dotted line represents the numerically observed error. We consider the pointwise error measured at a grid point nearest to $x = y = \frac{1}{2}$ (four grid points qualify, but due to the symmetry of the function this is not a problem). From Figure 2.3, it appears that the experimental error is indeed converging to the analytical leading-order result as N increases. In Table 2.1, the ratio $(E^{N-1,N-1}_{\text{analytical}} - E^{N-1,N-1}_{\text{numerical}})/(E^{N,N}_{\text{analytical}} - E^{N,N}_{\text{numerical}})$ is listed for several values of N. Table 2.1 indicates that $E^{N,N}_{\text{analytical}} - E^{N,N}_{\text{numerical}} = \mathcal{O}\left((1/4)^N\right)$, as it should be according to (2.73). Figure 2.4 displays $E^{N,N}$ for $N = 4, 5, 6$. In this Figure, we do indeed recognize the product of sines prescribed by (2.79).

As a test of the validity of the error expansion (2.72), the numerically observed error is also compared with higher-order approximations of the error. The expansion (2.72) is evaluated for the test case (2.78) up to $p + q \leq 4, 5, 6, 7, 8$ and compared with the numerically observed error. The results are displayed in Table 2.2. Table 2.2 clearly suggests that the series (2.72) converges to the numerically observed error, as $\max(p + q)$ increases.

2.3.4 Discussion

In this section, the local errors $e^{l,m}$ were determined for the $[\frac{1}{2}, \frac{1}{2}, 0]$, $[1, 1, -1]$ and $[0, 0, 1]$ piecewise-constant and piecewise-bi-linear schemes. The local errors were inserted into the expressions for the representation error $E^{N,N}$, yielding error results for the six schemes. For the piecewise-constant schemes, upper bounds were given instead of pointwise expressions. The motivation for this is that for pointwise expressions for the piecewise-constant schemes, the summation over the grid of grids cannot be performed due to the factors $(-1)^{i\times p}$ and $(-1)^{j\times q}$ in the local error $e^{l,m}$. This complication does not appear for the bi-linear schemes since for these schemes the leading-order term corresponds to $p = q = 2$, which guarantees that $(-1)^{i\times p} = (-1)^{j\times q} = 1$.

The $[1, 1, -1]$ piecewise-bi-linear scheme is clearly the most interesting of the schemes considered since it has the smallest approximation error. In fact, the $[\frac{1}{2}, \frac{1}{2}, 0]$ and $[0, 0, 1]$ schemes were only included for comparison, they are not intended for actual use. When the leading-order error result is insufficient (on coarse grids or when higher derivatives are not small), it may be necessary to predict the error with the full error expansion. For the $[1, 1, -1]$ piecewise-bi-linear scheme, the full error expansion is given by (2.72).

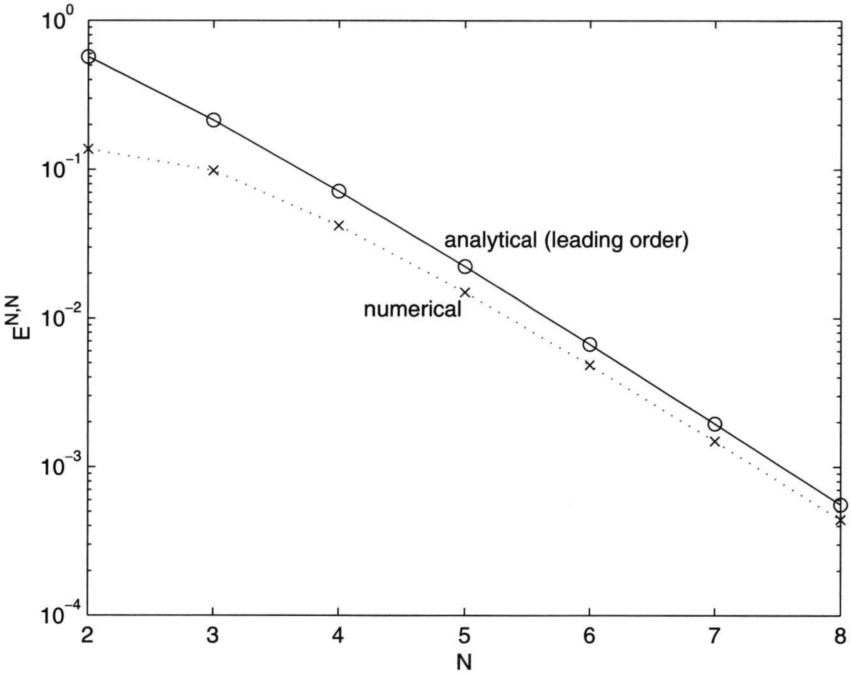


Figure 2.3: Numerically observed error converges to analytical leading-order result for $N \to \infty$

N	$\dfrac{E^{N-1,N-1}_{\text{analytical}} - E^{N-1,N-1}_{\text{numerical}}}{E^{N,N}_{\text{analytical}} - E^{N,N}_{\text{numerical}}}$
3	3.7553
4	3.9332
5	3.9817
6	3.9962
7	3.9984
8	3.9996

Table 2.1: Orders of convergence

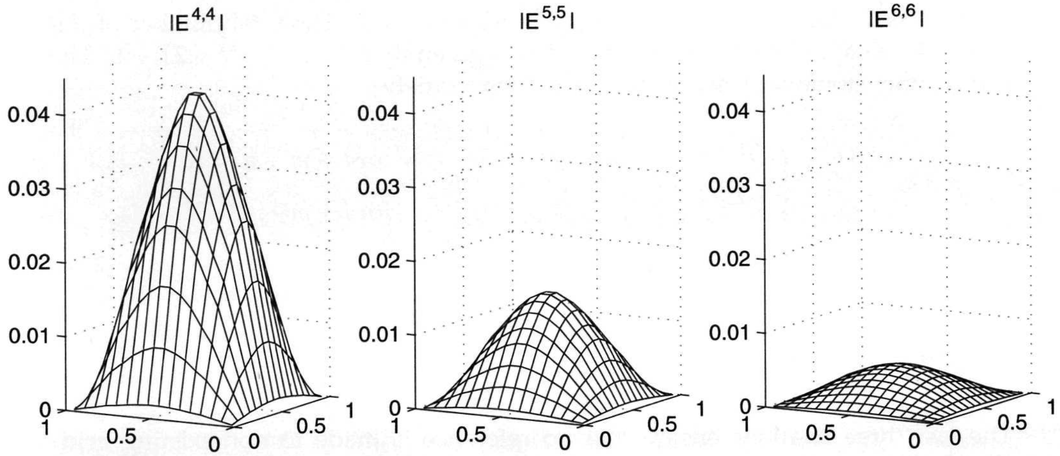


Figure 2.4: Spatial error distributions for $N = 4, 5, 6$

$\max(p+q)$	$\left\| E^{4,4}_{analytical} - E^{4,4}_{numerical} \right\|_{\infty}$
4	0.0131
5	0.0068
6	0.0036
7	0.0010
8	0.0005

Table 2.2: Higher-order error approximations

2.4 Extension to three dimensions

The current derivation for the sparse-grid representation error can easily be extended to three spatial dimensions. The given function $f(x,y,z)$ is then restricted to grids $\Omega^{l,m,n}$ satisfying $l+m+n = N-2, N-1, N$. The total number of degrees of freedom contained in these grids is given by $2^N\left(N^2-3N+2\right)-1$. The three-dimensional representations are taken to satisfy

$$\begin{array}{lcll}
\hat{f}^{l,m,n} & = & \left\{\begin{array}{l} f^{l,m,n}, \\ \sum_{l'=-1}^{0}\sum_{m'=-1}^{0}\sum_{n'=-1}^{0} \\ \alpha^{l',m',n'}p^{l,m,n}\hat{f}^{l+l',m+m',n+n'} \end{array}\right. & \begin{array}{l}\text{for } l+m+n \leq N, \\ \text{for } l+m+n > N,\end{array} \\
\alpha^{0,0,0} & \equiv & 0, & \\
\alpha^{-1,m',n'} & = & 0 \text{ if } l=0, & \\
\alpha^{l',-1,n'} & = & 0 \text{ if } m=0, & \\
\alpha^{l',m',-1} & = & 0 \text{ if } n=0. &
\end{array} \tag{2.80}$$

The last three relations ensure that no reference is made to non-existing grid-functions. Note that, due to the last three relations, the coefficients $\alpha^{l',m',n'}$ are now dependent on l, m and n. This dependence is suppressed in the notation. The local error is now given by

$$e^{l,m,n} = \sum_{l'=-1}^{0}\sum_{m'=-1}^{0}\sum_{n'=-1}^{0} \alpha^{l',m',n'}p^{l,m,n}\hat{f}^{l+l',m+m',n+n'} - f^{l,m,n}. \tag{2.81}$$

The recursive relation for $E^{l,m,n}$, for the three-dimensional case, reads

$$E^{l,m,n} = e^{l,m,n} + \sum_{l'=-1}^{0}\sum_{m'=-1}^{0}\sum_{n'=-1}^{0} \alpha^{l',m',n'}p^{l,m,n}E^{l+l',m+m',n+n'}. \tag{2.82}$$

For the two-dimensional case, the optimal combination scheme was found to be $[\alpha=\beta=1, \gamma=-1]$. For the three-dimensional case, the choice

$$\begin{array}{llllr}
\alpha^{-1,0,0} & =\alpha^{0,-1,0} & =\alpha^{0,0,-1} & =\alpha^{-1,-1,-1}= & 1 \\
\alpha^{-1,-1,0} & =\alpha^{-1,0,-1} & =\alpha^{0,-1,-1} & = & -1
\end{array} \tag{2.83}$$

represents the optimal combination scheme. Analogous to (2.16) for the $[\alpha=\beta=1, \gamma=-1]$ scheme, the combination scheme given by (2.83) leads to

$$E^{N,N,N} = \sum_{\substack{0\leq l,m,n\leq N \\ N<l+m+n}} P^{N,N,N}e^{l,m,n}. \tag{2.84}$$

To evaluate (2.84), we need the following equivalent form

$$\begin{aligned} E^{N,N,N} \quad = \quad & \textstyle\sum_{l=1}^{N} \sum_{m=1}^{l} P^{N,N,N} e^{l,m,0} \\ & + \textstyle\sum_{l=1}^{N} \sum_{n=1}^{l} P^{N,N,N} e^{l,0,n} \\ & + \textstyle\sum_{l=m}^{N} \sum_{n=m}^{l} P^{N,N,N} e^{0,m,n} \\ & + \left(\textstyle\sum_{l=1}^{N} \sum_{m=1}^{N} \sum_{n=1}^{N} - \sum_{l=1}^{N-2} \sum_{m=1}^{N-1-l} \sum_{n=1}^{N-l-m}\right) P^{N,N,N} e^{l,m,n}. \end{aligned} \tag{2.85}$$

Just as for the two-dimensional case, the combination scheme given by (2.80) and (2.83) can be expressed in a direct form that expresses $\hat{f}^{N,N,N}$ directly in terms of the coarse representations $\{\hat{f}^{l,m,n}, l+m+n = N-2, N-1, N\}$. The direct form reads

$$\hat{f}^{N,N,N} = \left(\sum_{\substack{0 \le l,m,n \le N \\ l+m+n=N}} -2 \sum_{\substack{0 \le l,m,n \le N \\ l+m+n=N-1}} + \sum_{\substack{0 \le l,m,n \le N \\ l+m+n=N-2}} \right) P^{N,N,N} \hat{f}^{l,m,n}. \tag{2.86}$$

The three-dimensional local error is given by

$$\begin{aligned} e^{l,m,n}_{i_\times,j_\times,k_\times} \quad = \quad & -f^{l,m,n}_{i_\times,j_\times,k_\times} + \textstyle\sum_{p=0}^{\infty} \sum_{q=0}^{\infty} \sum_{r=0}^{\infty} \phi_{p,q,r} \\ & \left(\frac{(-1)^{i_\times} \Delta x^l}{2}\right)^p \left(\frac{(-1)^{j_\times} \Delta y^m}{2}\right)^q \left(\frac{(-1)^{k_\times} \Delta z^n}{2}\right)^r \frac{\partial_x^p \partial_y^q \partial_z^r f^{l,m,n}_{i_\times,j_\times,k_\times}}{p!q!r!}, \end{aligned} \tag{2.87}$$

$$\begin{aligned} \phi_{p,q,r} \quad \equiv \quad & \textstyle\sum_{l'=-1}^{0} \sum_{m'=-1}^{0} \sum_{n'=-1}^{0} \sum_{i=0}^{1-2l'} \sum_{j=0}^{1-2m'} \sum_{k=0}^{1-2n'} \alpha^{l',m',n'} \\ & \psi^{l',m',n'}_{i,j,k} \left(X^{l',m',n'}_{i,j,k}\right)^p \left(Y^{l',m',n'}_{i,j,k}\right)^q \left(Z^{l',m',n'}_{i,j,k}\right)^r, \end{aligned} \tag{2.88}$$

where

$$\begin{aligned} X^{l',m',n'}_{i,j,k} &= -4 - l' + 2(1-l')i, \\ Y^{l',m',n'}_{i,j,k} &= -4 - m' + 2(1-m')j, \\ Z^{l',m',n'}_{i,j,k} &= -4 - n' + 2(1-n')k. \end{aligned} \tag{2.89}$$

2.4.1 Piecewise-constant interpolation

For piecewise-constant interpolation, the interpolation weights $\psi^{l',m',n'}_{i,j,k}$ are given by

$$\psi^{l',m',n'}_{i,j,k} = \delta_{i-2-l'} \delta_{j-2-m'} \delta_{k-2-n'}. \tag{2.90}$$

Substitution of (2.90) into (2.88) yields

$$\phi_{p,q,r} = \sum_{l'=-1}^{0} \sum_{m'=-1}^{0} \sum_{n'=-1}^{0} \left(\delta_{l'+1} + \delta_{l'} \delta_p\right) \left(\delta_{m'+1} + \delta_{m'} \delta_q\right) \left(\delta_{n'+1} + \delta_{n'} \delta_r\right) \alpha^{l',m',n'}. \tag{2.91}$$

Substitution of (2.91) into (2.87) and next of (2.87) into (2.81) yields

$$\left\|E^{N,N,N}\right\|_\infty \le \frac{1}{16}N^2\left(\frac{1}{2}\right)^N \left\|\partial_x\partial_y\partial_z f\right\|_\infty + \mathcal{O}\left(N\left(\frac{1}{2}\right)^N\right), \tag{2.92}$$

or, in terms of the mesh width,

$$\left\|E^{N,N,N}\right\|\infty \le \frac{1}{16}h\log_2^2 h^{-1}\left\|\partial_x\partial_y\partial_z f\right\|_\infty + \mathcal{O}\left(h\log_2 h^{-1}\right). \tag{2.93}$$

Thus, in three-dimensions, the piecewise-constant scheme has a representation error of order $h\log_2^2 h^{-1}$.

2.4.2 Piecewise tri-linear interpolation

For tri-linear interpolation, the $\psi_{i,j,k}^{l',m',n'}$ are given by

$$\begin{aligned}\psi_{i,j,k}^{l',m',n'} &= \textstyle\sum_{l'=-1}^{0}\sum_{m'=-1}^{0}\sum_{n'=-1}^{0}\chi_i^{l'}\chi_j^{m'}\chi_k^{n'},\\ \chi_i^{l'} &\equiv \delta_{l'}\delta_{i-2}+\delta_{l'+1}\left(\tfrac{1}{4}\delta_i+\tfrac{3}{4}\delta_{i-1}\right).\end{aligned} \tag{2.94}$$

Substitution of (2.94) into (2.88) yields

$$\begin{aligned}\phi_{p,q,r} &= \textstyle\sum_{l'=-1}^{0}\sum_{m'=-1}^{0}\sum_{n'=-1}^{0}\left(\delta_{l'+1}\frac{(-3)^p+3}{4}+\delta_{l'}\delta_p\right)\\ &\quad\left(\delta_{m'+1}\tfrac{(-3)^q+3}{4}+\delta_{m'}\delta_q\right)\left(\delta_{n'+1}\tfrac{(-3)^r+3}{4}+\delta_{n'}\delta_r\right)\alpha^{l',m',n'}.\end{aligned} \tag{2.95}$$

Substitution of (2.95) into (2.87) and next of (2.87) into (2.81) yields

$$\begin{aligned}E^{N,N,N} &= \textstyle\sum_{p=2}^{\infty}\sum_{q=2}^{\infty}\left(\eta_{xy}\right)^{p,q}+\sum_{p=2}^{\infty}\sum_{r=2}^{\infty}\left(\eta_{xz}\right)^{p,r}\\ &\quad+\textstyle\sum_{q=2}^{\infty}\sum_{r=2}^{\infty}\left(\eta_{yz}\right)^{q,r}+\sum_{p=2}^{\infty}\sum_{q=2}^{\infty}\sum_{r=2}^{\infty}\left(\eta_{xyz}\right)^{p,q,r},\end{aligned} \tag{2.96}$$

where

$$\begin{aligned}
(\eta_{xy})^{p,q} &\equiv \frac{(-3)^p+3}{4}\frac{(-3)^q+3}{4}\frac{2^{-p-q}}{p!q!}\sum_{l=1}^{N}\sum_{m=1}^{l}2^{-lp-mq} \\
&\quad \sum_{i_\times,j_\times,k_\times}\psi_{i_\times,j_\times,k_\times}^{l-N,m-N,-N}(-1)^{i_\times p+j_\times q}\partial_x^p\partial_y^q f_{i_\times,j_\times,k_\times}^{l,m,0}, \\
(\eta_{xz})^{p,r} &\equiv \frac{(-3)^p+3}{4}\frac{(-3)^r+3}{4}\frac{2^{-p-r}}{p!r!}\sum_{l=1}^{N}\sum_{n=1}^{l}2^{-lp-nr} \\
&\quad \sum_{i_\times,j_\times,k_\times}\psi_{i_\times,j_\times,k_\times}^{l-N,-N,n-N}(-1)^{i_\times p+k_\times r}\partial_x^p\partial_z^r f_{i_\times,j_\times,k_\times}^{l,0,n}, \\
(\eta_{yz})^{q,r} &\equiv \frac{(-3)^r+3}{4}\frac{(-3)^r+3}{4}\frac{2^{-q-r}}{q!r!}\sum_{m=1}^{N}\sum_{n=1}^{m}2^{-mq-nr} \\
&\quad \sum_{i_\times,j_\times,k_\times}\psi_{i_\times,j_\times,k_\times}^{-N,m-N,r-N}(-1)^{j_\times q+k_\times r}\partial_y^q\partial_z^r f_{i_\times,j_\times,k_\times}^{0,m,n}, \\
(\eta_{xyz})^{p,q,r} &\equiv \frac{(-3)^p+3}{4}\frac{(-3)^q+3}{4}\frac{(-3)^r+3}{4}\frac{2^{-p-q-r}}{p!q!r!} \\
&\quad \left(\sum_{l=1}^{N}\sum_{m=1}^{N}\sum_{n=1}^{N}-\sum_{l=1}^{N-2}\sum_{m=1}^{N-1-l}\sum_{n=1}^{N-l-m}\right) \\
&\quad 2^{-lp-mq-nr}\sum_{i_\times,j_\times,k_\times}\psi_{i_\times,j_\times,k_\times}^{l-N,m-N,n-N} \\
&\quad (-1)^{i_\times p+j_\times q+k_\times r}\partial_x^p\partial_y^q\partial_z^r f_{i_\times,j_\times,k_\times}^{l,m,n}.
\end{aligned}$$

The corresponding leading-order term is

$$E_{i,j,k}^{N,N,N}=\frac{9}{1024}N^2\left(\frac{1}{4}\right)^N\partial_x^2\partial_y^2\partial_z^2 f_{i,j,k}^{N,N,N}+\mathcal{O}\left(N\left(\frac{1}{4}\right)^N\right), \qquad (2.97)$$

or, in terms of the mesh width,

$$E_{i,j,k}^{N,N,N}=\frac{9}{1024}h^2\log_2^2 h^{-1}\partial_x^2\partial_y^2\partial_z^2 f_{i,j,k}^{N,N,N}+\mathcal{O}\left(h^2\log_2 h^{-1}\right). \qquad (2.98)$$

Thus, the three-dimensional piecewise-tri-linear scheme has a representation error of order $h^2\log_2^2 h^{-1}$.

2.4.3 The semi-sparse grid

The combination procedure in the current section started with restricting $f(x,y,z)$ to grids $\Omega^{l,m,n}$ satisfying $l+m+n = N-2, N-1, N$. As an alternative, we now consider the semi-sparse approach as introduced in [3], which amounts to restricting the function to the grids $\Omega^{l,m,n}$ satisfying $l+m+n = 2N-2, 2N-1, 2N$, causing the number of degrees of freedom to increase to $2^{2N-3}(7N^2 + 29N + 1)$. This is an asymptotically two-dimensional complexity, as opposed to the one-dimensional complexity of the sparse-grid approach. Of course, the semi-sparse approach is expected to have a smaller representation error.

For the semi-sparse, three-dimensional combination technique, the representations are taken to satisfy

$$\hat{f}^{l,m,n} = \begin{cases} f^{l,m,n}, & \text{for } l+m+n \leq 2N, \\ \sum_{l'=-1}^{0} \sum_{m'=-1}^{0} \sum_{n'=-1}^{0} \alpha^{l',m',n'} p^{l,m,n} \hat{f}^{l+l',m+m',n+n'} & \text{for } l+m+n > 2N. \end{cases} \tag{2.99}$$

The coefficients $\alpha^{l',m',n'}$ are again taken to be given by (2.88), yielding the following direct form of the semi-sparse combination technique

$$\begin{aligned} \hat{f}^{N,N,N} = & \left(\sum_{\substack{0 \leq l,m,n \leq N \\ l+m+n=2N}} - 2 \sum_{\substack{0 \leq l,m,n < N \\ l+m+n=2N-1}} + \sum_{\substack{0 \leq l,m,n < N \\ l+m+n=2N-2}} \right) p^{N,N,N} \hat{f}^{l,m,n} \\ & - \left(\sum_{\substack{l=N \\ 0 \leq m,n < N \\ l+m+n=2N-1}} + \sum_{\substack{m=N \\ 0 \leq l,n < N \\ l+m+n=2N-1}} + \sum_{\substack{n=N \\ 0 \leq l,m < N \\ l+m+n=2N-1}} \right) p^{N,N,N} \hat{f}^{l,m,n}. \end{aligned} \tag{2.100}$$

The error coefficients $\phi^{p,q,r}$ are the same as for the truely-sparse approach, e.g., for piecewise-constant prolongation they are given by (2.91) and for piecewise-trilinear prolongation they are given by (2.95). The representation error is now given by

$$E^{N,N,N} = \sum_{\substack{0 < l,m,n \leq N \\ 2N < l+m+n}} e^{l,m,n}. \tag{2.101}$$

For piecewise-constant prolongation, the error expansion reads

$$E^{N,N,N} = \sum_{p=1}^{\infty} \sum_{q=1}^{\infty} \sum_{r=1}^{\infty} \left(\eta_{xyz}^{\text{const}} \right)^{p,q,r},$$

where

$$\begin{aligned} \left(\eta_{xyz}^{\text{const}} \right)^{p,q,r} \equiv & \frac{2^{-p-q-r}}{p!q!r!} \sum_{l=1}^{N} \sum_{m=N+1-l}^{N} \sum_{n=2N+1-l-m}^{N} 2^{-lp-mq-nr} \\ & \sum_{i_\times, j_\times, k_\times} \psi_{i_\times, j_\times, k_\times}^{l-N,m-N,n-N} (-1)^{i_\times p + j_\times q + k_\times r} \partial_x^p \partial_y^q \partial_z^r f_{i_\times, j_\times, k_\times}^{l,m,n}. \end{aligned}$$

The corresponding leading-order result is

$$\left\|E^{N,N,N}_{i,j,k}\right\|_\infty \le \frac{1}{16}N^2\left(\frac{1}{4}\right)^N \partial_x\partial_y\partial_z f^{N,N,N}_{i,j,k} + \mathcal{O}\left(N\left(\frac{1}{4}\right)^N\right), \tag{2.102}$$

or, in terms of the mesh width,

$$\left\|E^{N,N,N}_{i,j,k}\right\|_\infty \le \frac{1}{16}h^2\log_2^2 h^{-1}\partial_x\partial_y\partial_z f^{N,N,N}_{i,j,k} + \mathcal{O}\left(h^2\log_2 h^{-1}\right). \tag{2.103}$$

For piecewise-tri-linear prolongation, the error expansion reads

$$E^{N,N,N} = \sum_{p=2}^{\infty}\sum_{q=2}^{\infty}\sum_{r=2}^{\infty}\left(\eta^{\mathrm{lin}}_{xyz}\right)^{p,q,r}, \tag{2.104}$$

where

$$\begin{aligned}\left(\eta^{\mathrm{lin}}_{xyz}\right)^{p,q,r} \equiv\ & \frac{(-3)^p+3}{4}\,\frac{(-3)^q+3}{4}\,\frac{(-3)^r+3}{4}\,\frac{2^{-p-q-r}}{p!q!r!} \\ & \sum_{l=1}^{N}\ \sum_{m=N+1-l}^{N}\ \sum_{n=2N+1-l-m}^{N} \\ & 2^{-lp-mq-nr}\sum_{i_\times,j_\times,k_\times}\psi^{l-N,m-N,n-N}_{i_\times,j_\times,k_\times} \\ & (-1)^{i_\times p+j_\times q+k_\times r}\partial_x^p\partial_y^q\partial_z^r f^{l,m,n}_{i_\times,j_\times,k_\times}.\end{aligned}$$

The corresponding leading-order term is

$$E^{N,N,N}_{i,j,k} = \frac{9}{1024}N^2\left(\frac{1}{16}\right)^N \partial_x^2\partial_y^2\partial_z^2 f^{N,N,N}_{i,j,k} + \mathcal{O}\left(N\left(\frac{1}{16}\right)^N\right), \tag{2.105}$$

or, in terms of the mesh width,

$$E^{N,N,N}_{i,j,k} = \frac{9}{1024}h^4\log_2^2 h^{-1}\partial_x^2\partial_y^2\partial_z^2 f^{N,N,N}_{i,j,k} + \mathcal{O}\left(h^4\log_2 h^{-1}\right). \tag{2.106}$$

2.4.4 A numerical test

As a test of the derivations in the current section, consider the following test case

$$f(x,y,z) = \sin(\pi x)\sin(\pi y)\sin(\pi z).$$

In Figures 2.5 and 2.6, the first-order error expressions (2.97) and (2.105) for the sparse and the semi-sparse schemes, respectively, are compared with corresponding numerical results. The errors are evaluated at a grid point nearest to $x = y = z = \frac{1}{2}$ (eight grid points qualify but due to the symmetry of the function this is not a problem). From Figures 2.5 and 2.6, it appears that the asymptotic expressions (2.97) and (2.105) indeed describe the numerical error of the sparse and the semi-sparse schemes, respectively, for $N \to \infty$. Convergence of the error expansions (2.96) and (2.104) for the sparse and the semi-sparse schemes to the corresponding numerical results as $\max(p+q+r) \to \infty$ is shown in Figures 2.7 and 2.8, respectively.

2.4.5 Discussion

In the current section, the error analysis introduced in Sections 2.2 and 2.3 was extended to three dimensions. Besides the sparse grid, also a so-called semi-sparse grid was considered. The semi-sparse grid was shown to have a representation error of $\mathcal{O}\left(h^4 \log_2^2 h^{-1}\right)$. If the (semi-) sparse-grid representation error would be the only error to deal with, then the semi-sparse-grid approach would be superior to the sparse-grid approach. This is illustrated in Figure 2.9, in which the numerically observed error at a grid point nearest to $x = y = z = \frac{1}{2}$ is plotted versus the number of degrees of freedom for the tri-linear sparse and semi-sparse schemes, for the test function $f(x, y, z) = \sin(\pi x)\sin(\pi y)\sin(\pi z)$. Figure 2.9 suggests that the semi-sparse-grid approach yields a smaller error for the same number of degrees of freedom than the sparse-grid approach. However, this suggestion is misleading since the sparse-grid representation error is not the only relevant error.

In the current setup, the sparse and semi-sparse representations $\hat{f}^{N,N,N}$ are piecewise-constant or piecewise-d-linear and hence contain an additional error of $\mathcal{O}(h)$ or $\mathcal{O}(h^2)$, respectively, when evaluated outside grid points. A sensible comparison of the sparse and semi-sparse approaches includes this error. In Figure 2.10, the sparse and semi-sparse approaches are again compared, now with inclusion of the $\mathcal{O}(h^2)$ tri-linear representation error. This error is included by comparing the average of grid-function-values nearest to $x = y = z = \frac{1}{2}$ with the exact value at $x = y = z = \frac{1}{2}$. From Figure 2.10, it is apparent that when the tri-linear representation error on the finest grid is included, the sparse-grid approach yields a smaller error than the semi-sparse-grid approach for the same number of degrees of freedom, as was expected. In Figure 2.10, we also plotted the conventional tri-linear representation error at $x = y = z = \frac{1}{2}$ versus the complexity of the conventional grid, 2^{3N}. Figure 2.10 clearly indicates that for the current test function, the sparse-grid representation is more efficient than the conventional representation and, for more than 10^5 degrees of freedom, the semi-sparse representation is also more efficient than conventional representation, but less efficient than a truely-sparse representation.

If we would only be interested in the solution at grid points of the finest grid $\Omega^{N,N,N}$, then we might argue that there is no reduction in representation error for the semi-sparse approach and hence that the semi-sparse approach is more efficient than the truely-sparse approach. However, so far, we have assumed that the function f is known exactly at the points contained in the coarse grids. Of course, when solving a differential equation this is not true. Then, the coarse-grid functions are subject to a discretization error. In general, a discretization error of order $\mathcal{O}\left(h^n_{\text{coarse}}\right)$ on the coarse grids leads to an error of order $\mathcal{O}\left(h^n\right)$ on the finest grid $\Omega^{N,N,N}$. Therefore, a very common discretization error of $\mathcal{O}\left(h^2_{\text{coarse}}\right)$ also reduces the representation error of the semi-sparse approach to $\mathcal{O}\left(h^2\right)$. To exploit the $\mathcal{O}\left(h^4 \log_2^2 h^{-1}\right)$ representation error of the semi-sparse approach, a discretization of $\mathcal{O}\left(h^4_{\text{coarse}}\right)$ would be required. However, if such a discretization were feasible, then it would be wiser to stick to the conventional full grid, since this would be more efficient then.

2.5 Discontinuous functions

In this section, we do not require f to be a smooth function. In particular, we examine the behavior of the error in the case that f is a two-dimensional step function of the type

$$f(x,y) = \begin{cases} -1, & (1+\lambda)x + (1-\lambda)y < 1. \\ 0, & (1+\lambda)x + (1-\lambda)y = 1. \\ +1, & (1+\lambda)x + (1-\lambda)y > 1. \end{cases} \tag{2.107}$$

We will obtain expressions for the local error $e^{l,m}$ directly from its defining equation (2.3) by substitution of values for α, β, γ and $\psi^{l,m}_{i,j}$. In general, we have

$$\begin{aligned} e^{l,m}_{i_\times,j_\times} = -f^{l,m}_{i_\times,j_\times} &+ \alpha \textstyle\sum_{i=0}^{3} \sum_{j=0}^{1} \psi^{-1,0}_{i,j} f\left(x^l_{i_\times} + X^{-1,0}_{i,j}(-1)^{i_\times}\frac{\Delta x^l}{2}, y^m_{j_\times}\right) \\ &+ \beta \textstyle\sum_{i=0}^{1} \sum_{j=0}^{3} \psi^{0,-1}_{i,j} f\left(x^l_{i_\times}, y^m_{j_\times} + Y^{0,-1}_{i,j}(-1)^{j_\times}\frac{\Delta y^m}{2}\right) \\ &+ \gamma \textstyle\sum_{i=0}^{1} \sum_{j=0}^{1} \psi^{-1,-1}_{i,j} \\ &\quad f\left(x^l_{i_\times} + X^{0,-1}_{i,j}(-1)^{i_\times}\frac{\Delta x^l}{2}, y^m_{j_\times} + Y^{-1,0}_{i,j}(-1)^{j_\times}\frac{\Delta y^m}{2}\right), \end{aligned} \tag{2.108}$$

where $X^{l',m'}_{i,j}$ and $Y^{l',m'}_{i,j}$ are given by (2.31) and where the coefficients $\psi^{l',m'}_{i,j}$ determine the prolongation. Since now $f(x,y)$ is a step function, we assume that prolongation by bi-linear interpolation will not be superior to piecewise-constant interpolation. Hence, we will only consider piecewise-constant interpolation. For piecewise-constant interpolation, $\psi^{l',m'}_{i,j}$ is given by (2.34). Substitution of (2.34) into (2.108) yields

$$\begin{aligned} e^{l,m}_{i_\times,j_\times} = -f^{l,m}_{i_\times,j_\times} &+ \alpha f\left(x^l_{i_\times} + (-1)^{i_\times}\frac{\Delta x^l}{2}, y^m_{j_\times}\right) + \beta f\left(x^l_{i_\times}, y^m_{j_\times} + (-1)^{j_\times}\frac{\Delta y^m}{2}\right) \\ &+ \gamma f\left(x^l_{i_\times} + (-1)^{i_\times}\frac{\Delta x^l}{2}, y^m_{j_\times} + (-1)^{j_\times}\frac{\Delta y^m}{2}\right). \end{aligned} \tag{2.109}$$

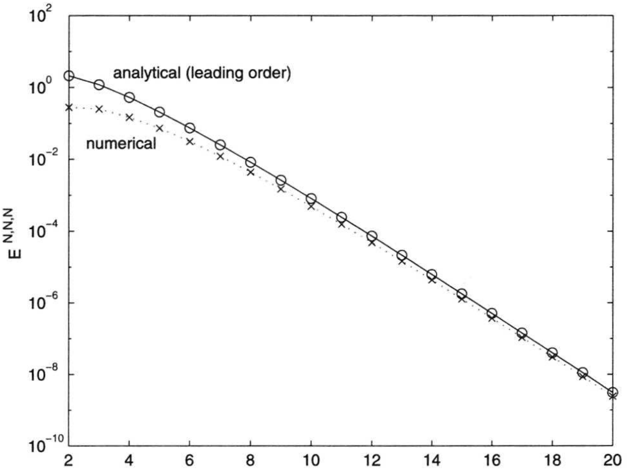


Figure 2.5: Convergence of the sparse-grid representation error to the analytical result

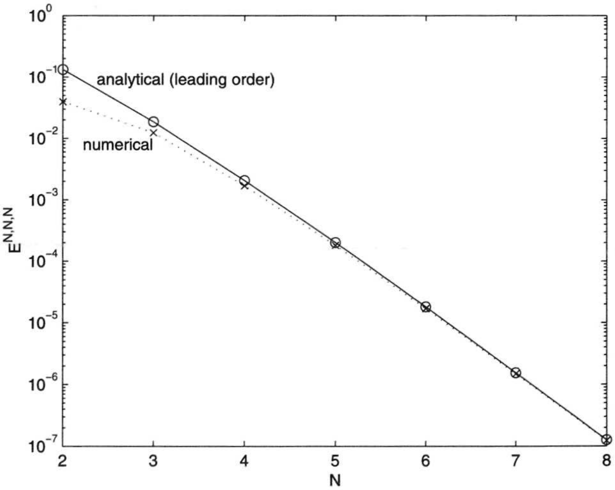


Figure 2.6: Convergence of the semi-sparse-grid representation error to the analytical result

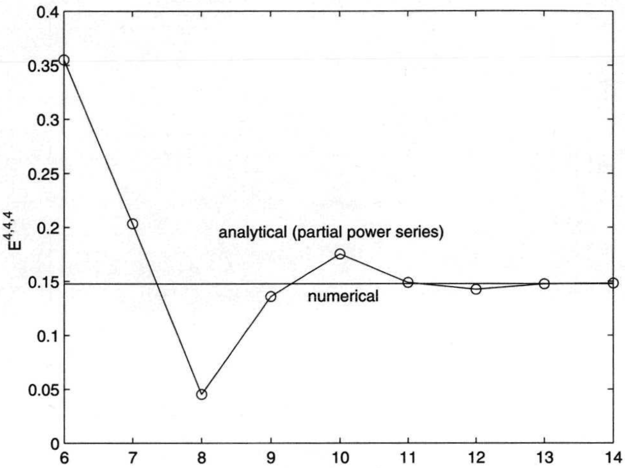


Figure 2.7: Convergence of the power series for the sparse-grid representation error

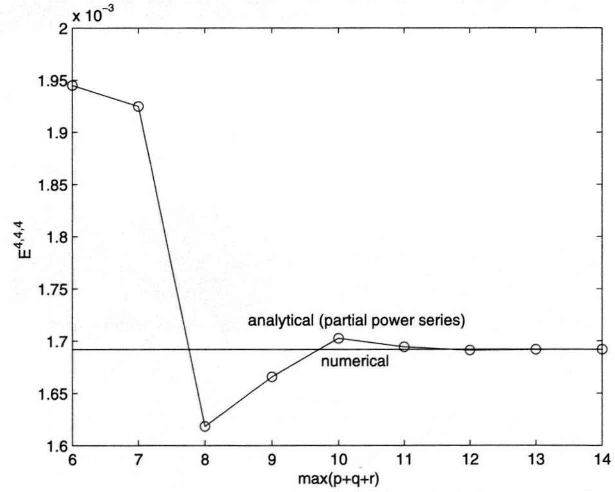


Figure 2.8: Convergence of the power series for the semi-sparse-grid representation error

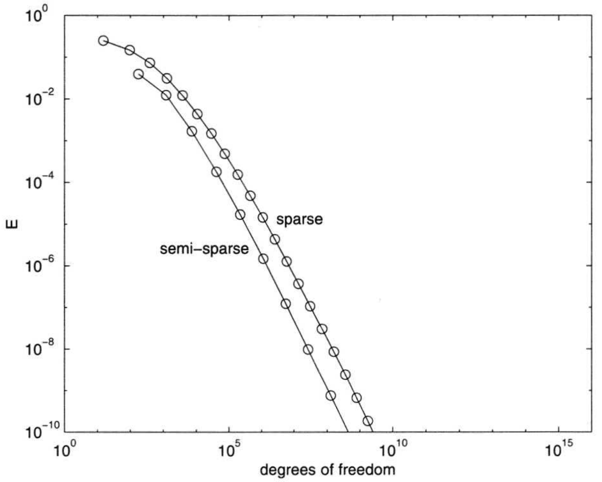


Figure 2.9: Sparse and semi-sparse representation errors (numerical), the conventional representation error is neglected

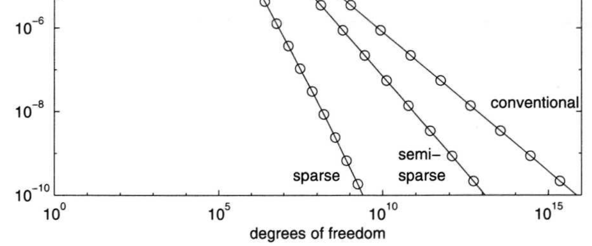


Figure 2.10: Conventional, sparse and semi-sparse representation errors (numerical), sparse and semi-sparse errors include the conventional representation error

2.5.1 The $[\frac{1}{2}, \frac{1}{2}, 0]$ piecewise-constant scheme.

For $\alpha = \beta = 1, \gamma = -1$, the local error $e^{l,m}$ takes the form

$$e^{l,m}_{i_\times,j_\times} = -f^{l,m}_{i_\times,j_\times} + \frac{1}{2} f\left(x^l_{i_\times} + (-1)^{i_\times}\frac{\Delta x^l}{2}, y^m_{j_\times}\right) + \frac{1}{2} f\left(x^l_{i_\times}, y^m_{j_\times} + (-1)^{j_\times}\frac{\Delta y^m}{2}\right). \tag{2.110}$$

This expression is only non-zero for a limited number of points, determined by the line $(1+\lambda)x + (1-\lambda)y = 1$. In Figure 2.11, black triangles have been drawn that correspond to equation (2.110). Triangles that are intersected by the line $(1+\lambda)x + (1-\lambda)y = 1$ correspond to points for which $e^{l,m}$ is of order 1 (proportional to the step). The number of triangles that are intersected is assumed to be proportional to $\max\left(2^l, \frac{1-\lambda}{1+\lambda}2^m\right)$ if $\lambda \geq 0$ and $\lambda \neq 1$, and to $\max\left(\frac{1+\lambda}{1-\lambda}2^l, 2^m\right)$ if $\lambda \leq 0$ and $\lambda \neq -1$. Thus, for $\lambda \geq 0$ and $\lambda \neq 1$, there is a $\kappa \in \mathbf{R}$ such that for all $l \geq 0$ and $m \geq 0$

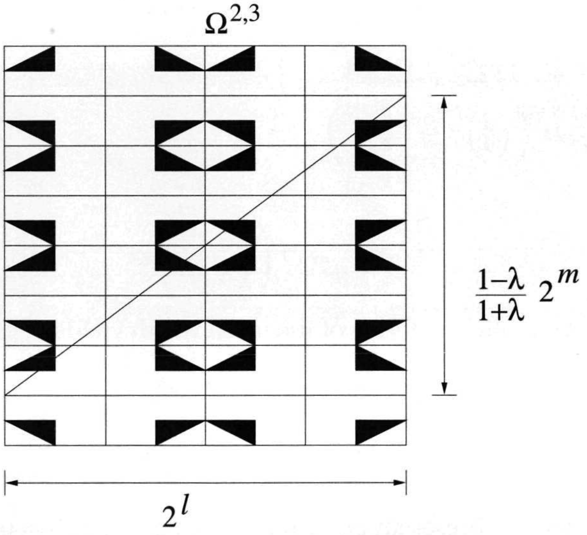


Figure 2.11: Counting errors, $\frac{1}{2}, \frac{1}{2}$ combination

$$\left\|e^{l,m}\right\|_1 \leq 2^{-l-m}\kappa \max\left(2^l, \frac{1-\lambda}{1+\lambda}2^m\right). \tag{2.111}$$

For the $[\frac{1}{2}, \frac{1}{2}, 0]$ scheme, the representation error $E^{N,N}$ is given by (2.6), which we use to obtain the following expression for $\left\|E^{N,N}\right\|_1$

$$\begin{aligned}\left\|E^{N,N}\right\|_1 &= \left\|\sum_{n=0}^{N-1} 2^{-n} \sum_{i=0}^{n} \begin{pmatrix} n \\ i \end{pmatrix} P^{N,N} e^{N-i,N-n+i}\right\|_1 \\ &\leq \sum_{n=0}^{N-1} 2^{-n} \sum_{i=0}^{n} \begin{pmatrix} n \\ i \end{pmatrix} \left\|P^{N,N} e^{N-i,N-n+i}\right\|_1 \\ &= \sum_{n=0}^{N-1} 2^{-n} \sum_{i=0}^{n} \begin{pmatrix} n \\ i \end{pmatrix} \left\|e^{N-i,N-n+i}\right\|_1 .\end{aligned} \tag{2.112}$$

Substitution of (2.111) into (2.112) gives

$$\begin{aligned}\left\|E^{N,N}\right\|_1 &\leq \kappa \sum_{n=0}^{N-1} 2^{-n} \sum_{i=0}^{n} \begin{pmatrix} n \\ i \end{pmatrix} 2^{n-2N} \max\left(\tfrac{1-\lambda}{1+\lambda} 2^{N-i}, 2^{N-n+i}\right) \\ &\leq \kappa \sum_{n=0}^{N-1} 2^{-n} \sum_{i=0}^{n} \begin{pmatrix} n \\ i \end{pmatrix} 2^{n-2N} \max\left(2^{N-i}, 2^{N-n+i}\right) \\ &= \kappa \sum_{n=0}^{N-1} 2^{-n} \sum_{i=0}^{n} \begin{pmatrix} n \\ i \end{pmatrix} 2^{-N} \max\left(2^{n-i}, 2^{i}\right) \\ &\leq 2^{1-N} \kappa \sum_{n=0}^{N-1} \sum_{i=0}^{n} \begin{pmatrix} n \\ i \end{pmatrix} 2^{-i} \\ &= 4\kappa \left(\left(\tfrac{3}{4}\right)^N - \left(\tfrac{1}{2}\right)^N\right) .\end{aligned} \tag{2.113}$$

Thus,

$$\left\|E^{N,N}\right\|_1 = \mathcal{O}\left(\left(\frac{3}{4}\right)^N\right) . \tag{2.114}$$

Rewriting the last equation in terms of the mesh width yields

$$\left\|E^{N,N}\right\|_1 = \mathcal{O}\left(h^{2-\log_2 3}\right) . \tag{2.115}$$

Note that we have taken $\lambda \geq 0$, $\lambda \neq 1$. It is obvious that $\lambda < 0$, $\lambda \neq -1$ gives the same result. Thus, for a step function described by (2.107), the $[\frac{1}{2}, \frac{1}{2}, 0]$ piecewise-constant scheme has a representation error of order $2 - \log_2 3 \approx 0.42$.

2.5.2 The $[1, 1, -1]$ piecewise-constant scheme.

For $\alpha = \beta = 1, \gamma = -1$, the local error $e^{l,m}$ takes the form

$$\begin{aligned}e^{l,m}_{i_\times,j_\times} = -f^{l,m}_{i_\times,j_\times} \quad &+ \quad f\left(x^l_{i_\times} + (-1)^{i_\times} \tfrac{\Delta x^l}{2}, y^m_{j_\times}\right) + f\left(x^l_{i_\times}, y^m_{j_\times} + (-1)^{j_\times} \tfrac{\Delta y^m}{2}\right) \\ &- \quad f\left(x^l_{i_\times} + (-1)^{i_\times} \tfrac{\Delta x^l}{2}, y^m_{j_\times} + (-1)^{j_\times} \tfrac{\Delta y^m}{2}\right) .\end{aligned} \tag{2.116}$$

This expression is also only non-zero for a limited number of points, determined by the line $(1 + \lambda)x + (1 - \lambda)y = 1$. In Figure 2.12, rectangles have been drawn that correspond to equation (2.110). Squares that are cut, through a horizontal

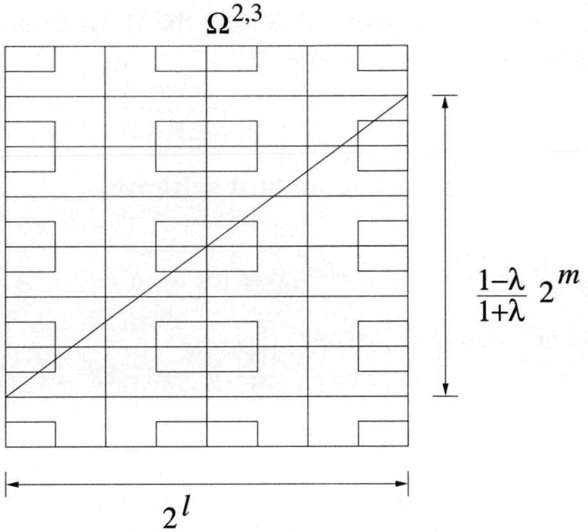


Figure 2.12: Counting errors, $1, 1, -1$ combination

and a vertical side, by the line $(1+\lambda)x + (1-\lambda)y = 1$ correspond to points for which $e^{l,m}$ is of order 1 (proportional to the step). The number of rectangles that are cut, through a horizontal and a vertical side, is assumed to be proportional to $\min\left(2^l, \frac{1-\lambda}{1+\lambda}2^m\right)$ if $\lambda \geq 0$ and $\lambda \neq 1$, and to $\min\left(\frac{1+\lambda}{1-\lambda}2^l, 2^m\right)$ if $\lambda \leq 0$ and $\lambda \neq -1$. Thus, for $\lambda \geq 0$ and $\lambda \neq 1$, there is a $\kappa \in \mathbf{R}$ such that for all $l \geq 0$ and $m \geq 0$

$$\left\|e^{l,m}\right\|_1 \leq 2^{-l-m}\kappa \min\left(2^l, \frac{1-\lambda}{1+\lambda}2^m\right). \tag{2.117}$$

For the $[1, 1, -1]$ scheme, the representation error $E^{N,N}$ is given by (2.16), from which we obtain the following relation for $\|E^{N,N}\|_1$

$$\left\|E^{N,N}\right\|_1 \leq \sum_{n=0}^{N-1}\sum_{i=0}^{n}\left\|e^{N-i,N-n+i}\right\|_1. \tag{2.118}$$

Substitution of (2.117) into (2.118) gives

$$\begin{aligned}
\|E^{N,N}\|_1 &\leq \kappa\textstyle\sum_{n=0}^{N-1}\sum_{i=0}^{n} 2^{n-2N}\min\left(\frac{1-\lambda}{1+\lambda}2^{N-i}, 2^{N-n+i}\right)\\
&\leq \kappa\textstyle\sum_{n=0}^{N-1}\sum_{i=0}^{n} 2^{n-2N}\min\left(2^{N-i}, 2^{N-n+i}\right)\\
&= 2^{-N}\kappa\textstyle\sum_{n=0}^{N-1}\sum_{i=0}^{n}\min\left(2^{n-i}, 2^{i}\right)\\
&= 2^{1-N}\kappa\textstyle\sum_{n=0}^{N-1}\sum_{i=0}^{n/2} 2^{i}\\
&= \frac{4}{\sqrt{2}-1}\kappa\left(\left(\frac{1}{\sqrt{2}}\right)^N - \left(\frac{1}{2}\right)^N\right) - 2\left(\frac{1}{2}\right)^N N.
\end{aligned} \tag{2.119}$$

Rewriting in terms of the mesh width yields

$$\left\|E^{N,N}\right\|_1 = \mathcal{O}\left(h^{1/2}\right). \tag{2.120}$$

Thus, for a step function described by (2.107), the $[1, 1, -1]$ piecewise-constant scheme has a representation error of order $\frac{1}{2}$.

2.5.3 The $[0, 0, 1]$ piecewise-constant scheme

For $\alpha = \beta = 0, \gamma = 1$, the local error $e^{l,m}$ takes the form

$$e^{l,m}_{i_\times,j_\times} = -f^{l,m}_{i_\times,j_\times} + f\left(x^l_{i_\times} + (-1)^{i_\times}\frac{\Delta x^l}{2}, y^m_{j_\times} + (-1)^{j_\times}\frac{\Delta y^m}{2}\right). \tag{2.121}$$

This expression is again only non-zero for a limited number of points, determined by the line $(1+\lambda)x + (1-\lambda)y = 1$. In Figure 2.13, diagonal lines have been drawn that correspond to equation (2.121). Diagonal lines that are cut by $(1+\lambda)x + (1-\lambda)y = 1$ correspond to points for which $e^{l,m}$ is of order 1 (proportional to the step). The number of diagonal lines that are cut is assumed to be proportional to $\max\left(2^l, \frac{1-\lambda}{1+\lambda}2^m\right)$ if $\lambda \geq 0$ and $\lambda \neq 1$, and to $\max\left(\frac{1+\lambda}{1-\lambda}2^l, 2^m\right)$ if $\lambda \leq 0$ and $\lambda \neq -1$. Thus, for $\lambda \geq 0$ and $\lambda \neq 1$, there is a $\kappa \in \mathbf{R}$ such that for all $l \geq 0$ and $m \geq 0$

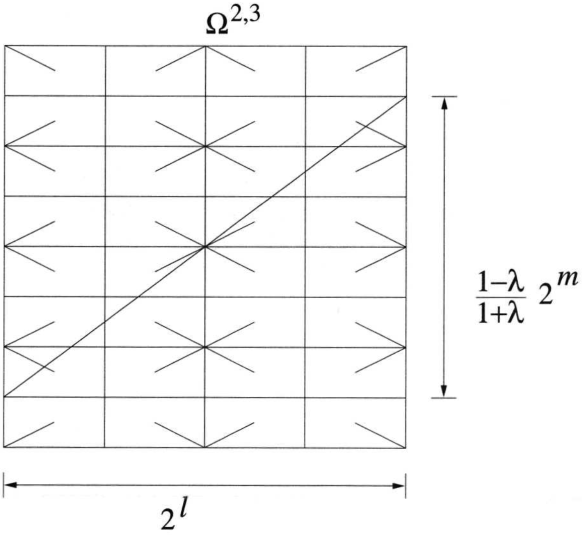


Figure 2.13: Counting errors, $\gamma = 1$ combination

$$\left\|e^{l,m}\right\|_1 \leq 2^{-l-m}\kappa \max\left(2^l, \frac{1-\lambda}{1+\lambda}2^m\right). \tag{2.122}$$

For the $\gamma = 1$ scheme, the representation error $E^{N,N}$ is given by (2.26), from which we obtain the following relation for $\|E^{N,N}\|_1$

$$\left\|E^{N,N}\right\|_1 \leq \sum_{l=N/2}^{N-1} \left\|e^{l,l}\right\|_1 . \tag{2.123}$$

Substitution of (2.122) into (2.123) gives

$$\begin{aligned} \|E^{N,N}\|_1 &\leq \kappa \textstyle\sum_{l=N/2}^{N-1} 2^{-l} \\ &= 2\kappa \left(\left(\frac{1}{\sqrt{2}}\right)^N - \left(\frac{1}{2}\right)^N \right) . \end{aligned} \tag{2.124}$$

Rewriting in terms of the mesh width yields

$$\left\|E^{N,N}\right\|_1 = \mathcal{O}\left(h^{1/2}\right) . \tag{2.125}$$

Thus, for a step function described by (2.107), the $[0,0,1]$ piecewise-constant scheme has a representation error of order $\frac{1}{2}$.

2.5.4 A numerical test

To test the validity of the conclusion that the $[1,1,-1]$ scheme has a representation error of order $\mathcal{O}\left(h^{1/2}\right)$ for the representation of a discontinuous function of the type (2.107), we now represent (2.107), with the $[1,1,-1]$ combination scheme, for $\lambda = 0$ (a diagonal line through the domain). In Table 2.3, the representation error in the L_1-norm is listed for $N = 2, 3, \ldots, 12$, together with convergence ratios. Table 2.3 shows that the L_1-norm of the representation error on sparse grids with N even is twice as small as on $N-1$, while going from even N to (odd) $N+1$ actually leads to a small rise in error. The explanation that the diagonal step function is better represented for N even than for N odd is that for N even there is a grid $\Omega^{N/2,N/2}$ within the set of coarse grids $\{\Omega^{l,m}, l+m = N-1, N\}$ on which the diagonal step function can be reasonably described. A more important observation is that the average convergence ratio (rightmost column) seems to tend to $\sqrt{2}$, as it should according to (2.120).

2.5.5 Discussion

In the current section, it was shown that the combination technique has a representation error of order $\mathcal{O}\left(h^{1/2}\right)$ when a step function is represented. This accuracy can also be obtained by interpolating solely from the grid $\Omega^{N/2,N/2}$, e.g., by conventional representation on the grid $\Omega^{N/2,N/2}$ which contains less degrees of freedom than the set of coarse grids $\{\Omega^{l,m}, l+m = N-1, N\}$ comprising the sparse grid. For the representation of genuinely discontinuous functions, the combination technique is not superior to conventional representation.

N	$\left\|E^{N,N}\right\|_1$	$\frac{\left\|E^{N-1,N-1}\right\|_1}{\left\|E^{N,N}\right\|_1}$	$\left(\frac{\left\|E^{N-2,N-2}\right\|_1}{\left\|E^{N,N}\right\|_1}\right)^{\frac{1}{2}}$
2	0.125000		
3	0.187500	0.666667	
4	0.093750	2.000000	1.154701
5	0.109375	0.857143	1.309307
6	0.054688	2.000000	1.309307
7	0.058594	0.933333	1.366260
8	0.029297	2.000000	1.366260
9	0.030273	0.967742	1.391217
10	0.015137	2.000000	1.391217
11	0.015381	0.984127	1.402945
12	0.007690	2.000000	1.402945

Table 2.3: Orders of convergence

2.6 Conclusions

The sparse-grid combination technique is an attractive alternative to the conventional representation of a function on a full grid. The reason for this is that, for the same number of degrees of freedom, the sparse-grid combination technique yields a significantly smaller representation error than conventional representation; see for instance Figure 2.10.

By analyzing the steps that make up the combination technique, explicit expressions for the representation error were obtained. The leading-order error terms contain cross derivatives of the function to be represented, instead of single-variable derivatives like the conventional representation error. The deficiency of the combination technique is that it will be less effective for functions that have large cross derivatives. This problem may be alleviated by adapting the grids to the geometry of the problem at hand.

For comparison, an alternative to the combination technique introduced in [1] was considered. This alternative technique, the $[\frac{1}{2}, \frac{1}{2}, 0]$ technique, appeared to perform less well than the technique in [1], the $[1, 1, -1]$ technique. In fact, the alternative technique even appeared to be inferior to conventional representation, such as the $[0, 0, 1]$ technique.

It was shown that for a step-function, which is not aligned with the grid, the combination technique performs less well than the standard representation. For such a non-aligned step-function, the order of the representation error was found to be $\mathcal{O}\left(h^{1/2}\right)$. (The explicit error expression derived may be useful for a combination technique that relies on grid refinement.)

The representation for the 3D semi-sparse combination technique, as proposed in [3], was analyzed. The representation error was found to be $\mathcal{O}\left(h^4(logh^{-1})^2\right)$. At first sight, this result implies that the 3D semi-sparse combination technique is to be preferred above the 3D truly-sparse combination technique. However, due to additional representation errors or discretization errors of $\mathcal{O}\left(h^2\right)$, the 3D semi-sparse representation error reduces to $\mathcal{O}\left(h^2\right)$, which makes it less attractive than the 3D truly-sparse combination technique.

BIBLIOGRAPHY

[1] M. Griebel, M. Schneider and C. Zenger, A combination technique for the solution of sparse grid problems, in: R. Beauwens and P. de Groen, eds., *Iterative Methods in Linear Algebra*, 263–281 (North-Holland, Amsterdam, 1992). Pages: 10, 17, 46

[2] P.W. Hemker, private communication (1998). Pages: 23

[3] B. Koren, P.W. Hemker and C.T.H. Everaars, Multiple semi-coarsened multigrid for 3D CFD, in: Proceedings of the *13th AIAA Computational Fluid Dynamics Conference*, 892–902 (AIAA, Reston, VA, 1997). Pages: 33, 46

[4] C.B. Liem, T. Lü and T.M. Shih, *The Splitting Extrapolation Method, A New Technique in Numerical Solution of Multidimensional Problems, Series on Applied Mathematics*, 7 (World Scientific, Singapore, 1995). Pages: 17

[5] U. Rüde, Multilevel, extrapolation and sparse grid methods, in: P.W. Hemker and P. Wesseling, eds., *Multigrid Methods*, **IV**, 281–294 (Birkhäuser, Basel, 1993). Pages: 17

[6] C. Zenger, Sparse grids, in: W. Hackbusch, ed., *Notes on Numerical Fluid Mechanics*, **31**, 241–251 (Vieweg, Braunschweig, 1990). Pages: 10

3

THE SPARSE-GRID COMBINATION TECHNIQUE APPLIED TO TIME-DEPENDENT ADVECTION PROBLEMS

Abstract. In the numerical technique considered in this paper, time-stepping is performed on a set of semi-coarsened space grids. At given time levels the solutions on the different space grids are combined to obtain the asymptotic convergence of a single, fine uniform grid. We present error estimates for the two-dimensional spatially constant-coefficient model problem and discuss numerical examples. A spatially variable-coefficient problem (Molenkamp-Crowley test) is used to assess the practical merits of the technique. The combination technique is shown to be more efficient than the single-grid approach, yet for the Molenkamp-Crowley test, standard Richardson extrapolation is still more efficient than the combination technique. However, parallelization is expected to significantly improve the combination technique's performance.

3.1 Introduction

The long-term aim of the present work is to make significant progress in the numerical solution of large-scale transport problems: systems of partial differential equations of the advection-diffusion-reaction type, used in the modeling of pollution of the atmosphere, surface water and ground water. The three-dimensional nature of these models and the necessity of modeling transport and chemical exchange between different components over long time spans, requires very efficient algorithms. For advanced three-dimensional modeling, computer capacity (computing time and memory) still is a severe limiting factor (e.g., see [8]). This limitation is felt in particular in the area of global air pollution modeling where the three-dimensional nature leads to huge numbers of grid points in each of which many calculations must be carried out. The application of sparse-grid techniques might offer a promising way-out.

Sparse-grid techniques were introduced by Zenger [10] in 1990 to reduce the number of degrees of freedom in finite-element calculations. The combination technique, as introduced in 1992 by Griebel, Schneider and Zenger [4], can be seen as a practical implementation of the sparse-grid technique. In the combination technique, the final solution is a linear combination of solutions on semi-coarsened grids, where the coefficients of the combination are chosen such that there is a canceling in leading-order error terms. As shown by Rüde in 1993 [7], the combination technique can be placed in a broader framework of multivariate extrapolation techniques.

We show that for our two-dimensional hyperbolic problems the combination technique requires $\sim h^{-2}$ operations to reach an accuracy of $O(h^p \log h^{-1})$ while the single grid requires $\sim h^{-3}$ operations to solve up to an accuracy of $O(h^p)$. Thus the combination technique is, asymptotically, more efficient than a single-grid solver. Another appealing property of the combination technique is that it is inherently parallel, i.e., it constructs the final solution from $\sim (\log h^{-1})^{d-1}$ independent solutions (d is the dimension of the problem) which can be computed in parallel. Parallel implementations of the combination technique were shown to be effective in [3] and [2].

Although we are ultimately interested in advection-diffusion-reaction equations, in the current work we restrict the attention to pure advection and leave the difussion and reaction processes to future research. In a number of articles the combination technique has already been analyzed both analytically and numerically, see for instance [1, 3, 4, 7]. However, in these references elliptic differential equations are considered, not hyperbolic equations like the time-dependent advection equation we are considering. In [5] the combination technique is shown to be promising for a constant coefficient advection equation. The current paper differs from [5] in that it focuses on error analysis while [5] focuses on numerical results. Furthermore, in [5] only constant coefficients are considered. Although we do not present error analysis for spatially variable coefficients, we do analyze this case numerically with the Molenkamp-Crowley test. The time-dependent coefficient case we analyze both numerically and analytically. When the combination technique is used to solve a differential equation, then a representation error and a combined discretization error are introduced. In [6] a detailed analysis is given of the representation error. In the current paper we focus on the combined discretization error.

The organization of the current paper is as follows. In Sections 3.2, 3.3 and 3.4 we derive leading order error expressions for the error that is introduced when we solve an advection equation, with spatially independent coefficients, with the combination technique. In the derivations we account for time-dependent coefficients and for intermediate combinations. In Section 3.5 we give some estimates for the asymptotic efficiency of the combination technique relative to the single-grid approach. In Section 3.6 four numerical test cases are analyzed, one of these is the Molenkamp-Crowley problem. The error estimates made in the earlier sections are verified and the combination technique is compared with the single-grid technique in terms of efficiency. The conclusions are summarized in Section 3.7. The main conclusion is that without parallelization - although marginally - the combination technique is already more efficient than the single-grid approach for a generic advection problem, such as the Molenkamp-Crowley test. Without parallelization, the combination technique still falls behind standard Richardson extrapolation, something which has also been concluded by Rüde [7] for elliptic problems.

3.2 Discretization error

In order to understand the combined discretization error we must first have a clear understanding of the discretization error itself. This section is devoted to the analysis of the error in the numerical solution that is due to spatial discretization. The temporal discretization errors are neglected. In the notation of functions only the relevant variables are printed, e.g., the function $f(x,y,t)$ can be referred to as $f(x,y,t), f(t), f(x,y)$ or simply as f, depending on context. The focus lies on the pure initial value problem for the spatially-constant coefficient, 2D advection equation

$$c_t + a\partial_x c + b\partial_y c = 0. \tag{3.1}$$

Equation (3.1) is integrated in time from $t = 0$ up to $t = 1$ with finite differences on the spatial domain $[-1,1] \times [-1,1]$. We denote the discretization of the advection operator $a\partial_x + b\partial_y$ by $aD_x + bD_y$. The corresponding spatially discretized equation reads

$$\frac{d}{dt}\omega + aD_x\omega + bD_y\omega = 0. \tag{3.2}$$

Here $\omega = \omega(t)$ denotes a continuous time grid function defined on a certain space grid. We define the (global) discretization error $d(t)$ according to

$$d(t) \equiv \omega(t) - c_h(t), \tag{3.3}$$

where $c_h(t)$ denotes the restriction of $c(t)$ to the space grid. We introduce the truncation error operator E according to

$$E \equiv aD_x + bD_y + \frac{d}{dt}. \tag{3.4}$$

The discretization error d can be seen to satisfy

$$\frac{d}{dt}d + Ec_h + aD_x d + bD_y d = 0,$$

with general solution

$$d(t) = e^{-\int_0^t \left(a(t')D_x + b(t')D_y\right)dt'} d(0) + \left(e^{-\int_0^t E(t')dt'} - I\right) c_h(t). \tag{3.5}$$

When a and b are independent of time then (3.5) reduces to

$$d(t) = e^{-t\left(aD_x + bD_y\right)} d(0) + \left(e^{-tE} - I\right) c_h(t),$$

which we expand as

$$d(t) = \sum_{i=0}^{\infty} \frac{(-tE)^i}{i!} e^{-t\left(a\partial_x + b\partial_y\right)} d(0) + \sum_{i=1}^{\infty} \frac{(-tE)^i}{i!} c_h(t). \tag{3.6}$$

3.2.1 Structure of the discretization error

In general, when the initial profile is error free a dimensionally split discretization of order p gives rise to a discretization error given by

$$d(t) = \sum_{i=1}^{\infty} \frac{t^i}{i!} \left(\sum_{j=p}^{\infty} \left(\alpha_j a h_x^j \partial_x^{j+1} + \beta_j b h_y^j \partial_y^{j+1} \right) \right)^i c_h(t), \tag{3.7}$$

where the constants α_j and β_j are the error constants in the truncation error. Equation (3.7) can be rewritten in the generic form

$$d(t) = \sum_{i=p}^{\infty} \left(h_x^i A_i(t) + h_y^i B_i(t) \right) + \sum_{j=p}^{\infty} \sum_{k=p}^{\infty} h_x^j h_y^k \gamma_{j,k}(t), \tag{3.8}$$

showing that the discretization error consists of terms proportional to $h_x^p, h_x^{p+1}, \cdots$ and $h_y^p, h_y^{p+1}, \cdots$ and $h_x^p h_y^p, h_x^{p+1} h_y^p, h_x^p h_y^{p+1}, h_x^{p+1} h_y^{p+1}, \cdots$.

3.2.2 Third-order upwind discretization

To introduce spatial discretizations we make use of the shift operators

$$\begin{aligned} S_{h_x} f(x,y) &\equiv f(x+h_x,y) = \sum_{i=0}^{\infty} \frac{(h_x \partial_x)^i}{i!} f(x,y), \\ S_{h_y} f(x,y) &\equiv f(x,y+h_y) = \sum_{i=0}^{\infty} \frac{(h_y \partial_y)^i}{i!} f(x,y), \end{aligned}$$

where we have supposed f to be a C^∞ function. We focus on the third-order upwind biased scheme which is given by

$$D_x = \begin{cases} \frac{\frac{1}{6}S_{-2h_x} - S_{-h_x} + \frac{1}{2} + \frac{1}{3}S_{h_x}}{h_x}, & a > 0, \\ -\frac{\frac{1}{6}S_{2h_x} - S_{h_x} + \frac{1}{2} + \frac{1}{3}S_{-h_x}}{h_x}, & a < 0, \end{cases} \quad D_y = \begin{cases} \frac{\frac{1}{6}S_{-2h_y} - S_{-h_y} + \frac{1}{2} + \frac{1}{3}S_{h_y}}{h_y}, & b > 0. \\ -\frac{\frac{1}{6}S_{2h_y} - S_{h_y} + \frac{1}{2} + \frac{1}{3}S_{-h_y}}{h_y}, & b < 0. \end{cases}$$

This yields the discretization error

$$d(t) = \sum_{i=1}^{\infty} \frac{t^i}{i!} \left(\sum_{j=3}^{\infty} \frac{(-2)^j - 3(-1)^j - 1}{3(j+1)!} \left(\frac{a^{j+1}}{|a|^j} h_x^j \partial_x^{j+1} + \frac{b^{j+1}}{|b|^j} h_y^j \partial_y^{j+1} \right) \right)^i c(t), \quad (3.9)$$

provided $d(0) = 0$. Neglecting $\mathcal{O}(h_x^4)$ and $\mathcal{O}(h_y^4)$ but including $\mathcal{O}(h_x^3 h_y^3)$ for later reference, equation (3.9) leads to the following leading order expression

$$d(t) = -\frac{t}{12} \left(|a|\, h_x^3 \partial_x^4 + |b|\, h_y^3 \partial_y^4 \right) c(t) + \frac{t^2}{144} |ab|\, h_x^3 h_y^3 \partial_x^4 \partial_y^4 c(t) + \mathcal{O}(h_x^4) + \mathcal{O}(h_y^4). \quad (3.10)$$

This leading-order result makes sense only when t, a, b and the derivatives of $c(t)$ are moderate.

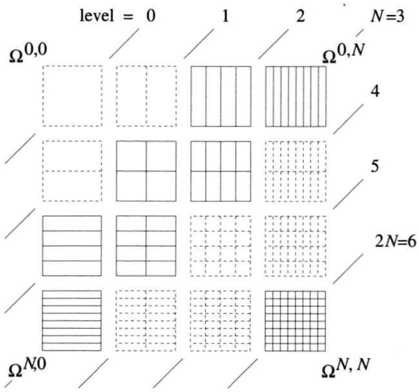
3.2.3 Time-dependent coefficients

To handle time-dependent coefficients we expand (3.5) as

$$d(t) = \sum_{i=0}^{\infty} \frac{(-\int_0^t E(t')dt')^i}{i!} e^{-\int_0^t (a(t')\partial_x + b(t')\partial_y)dt'} d(0) + \sum_{i=1}^{\infty} \frac{(-\int_0^t E(t')dt')^i}{i!} c(t).$$

For $d(0) = 0$, the time-dependent equivalent to (3.10) then reads

$$\begin{aligned} d(t) \quad = \quad & -\tfrac{1}{12} \left(\int_0^t |a(t')|\, dt'\, h_x^3 \partial_x^4 + \int_0^t |b(t')|\, dt'\, h_y^3 \partial_y^4 \right) c(t) \\ & + \tfrac{1}{144} \left(\int_0^t |a(t')|\, dt' \right) \left(\int_0^t |b(t')|\, dt' \right) h_x^3 h_y^3 \partial_x^4 \partial_y^4 c(t) + \mathcal{O}(h_x^4) + \mathcal{O}(h_y^4). \end{aligned} \quad (3.11)$$



Notation	Description
$\Omega^{N,N}$	finest grid of mesh width $h=2^{-N}H$
$\Omega^{l,m}$	semi-coarsened grid, of mesh widths $h_x=2^{-l}H$ and $h_y=2^{-m}H$
u	continuous, exact solution
$R^{l,m}$	restriction operator that maps onto $\Omega^{l,m}$
$P^{N,N}$	prolongation operator that maps onto $\Omega^{N,N}$
$U^{l,m}$	semi-discrete approximate solution on $\Omega^{l,m}$

Figure 3.1: Grid of grids.

3.3 Combination technique

The two-dimensional combination technique is based on a grid of grids as shown in Figure 3.1. Grids within the grid of grids are denoted by $\Omega^{l,m}$ where upper indices label the level of refinement relative to the *root grid* $\Omega^{0,0}$. The mesh widths in x-and y-direction of $\Omega^{l,m}$ are $h_x = 2^{-l}H$ and $h_y = 2^{-m}H$, where H is the mesh width of the uniform root grid $\Omega^{0,0}$. We denote the mesh width of the finest grid $\Omega^{N,N}$ by h. Note that h_x and h_y are dependent on the position (l,m) in the grid of grids while h is not.

In the time-dependent combination technique a given initial profile $c(x,y,0)$ is restricted, by injection, onto the grids $\Omega^{N,0}$, $\Omega^{N-1,1}$, $\cdots$, $\Omega^{0,N}$ and onto $\Omega^{N-1,0}$, $\Omega^{N-2,1}$, $\cdots$, $\Omega^{0,N-1}$, see Figure 3.1. The resulting coarse representations are then all evolved in time (exact time integration is assumed in the current paper). Then, at a chosen point in time, the coarse approximations are prolongated with q-th order interpolation onto the finest grid $\Omega^{N,N}$, where they are combined according to (4.3) to obtain a more accurate solution. The notation is summarized in Figure 3.1.

Considering the exact solution c, the combination technique, as introduced in [4], constructs a grid function $\widehat{c}^{N,N}$ on the finest grid $\Omega^{N,N}$ in the following manner,

$$\widehat{c}^{N,N} \equiv \sum_{l+m=N} P^{N,N}R^{l,m}c - \sum_{l+m=N-1} P^{N,N}R^{l,m}c.$$

The corresponding so-called *representation error* $r^{N,N}$ is

$$r^{N,N} \equiv \widehat{c}^{N,N} - R^{N,N}c. \tag{3.12}$$

Likewise, considering the semi-discrete solutions $\omega^{l,m}$, the combination technique constructs an approximate solution $\widehat{\omega}^{N,N}$ on the finest grid $\Omega^{N,N}$ from the coarse-grid approximate solutions according to

$$\widehat{\omega}^{N,N} = \sum_{l+m=N} P^{N,N}\omega^{l,m} - \sum_{l+m=N-1} P^{N,N}\omega^{l,m}. \tag{3.13}$$

Let $d^{l,m}$ denote the discretization error on grid $\Omega^{l,m}$, i.e.,

$$d^{l,m} \equiv \omega^{l,m} - R^{l,m}c. \tag{3.14}$$

The total error $e^{N,N} = \widehat{\omega}^{N,N} - R^{N,N}c$ present in $\widehat{\omega}^{N,N}$ is written as

$$e^{N,N} = r^{N,N} + \widehat{d}^{N,N},$$

where the *combined discretization* error $\widehat{d}^{N,N} = \widehat{\omega}^{N,N} - \widehat{c}^{N,N}$ is given by

$$\widehat{d}^{N,N} = \sum_{l+m=N} P^{N,N}d^{l,m} - \sum_{l+m=N-1} P^{N,N}d^{l,m}. \tag{3.15}$$

In [6] a detailed analysis is given of the representation error $r^{N,N}$. In the current paper we focus on the combined discretization error $\widehat{d}^{N,N}$.

3.4 Combined discretization error

3.4.1 Effect of the combination technique on a single error term

Inspection of (3.7) shows that the discretization error $d^{l,m}$ can be expanded as

$$d^{l,m}(t) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} h_x^i h_y^j R^{l,m}\theta_{i,j}(t)c(x,y,t), \tag{3.16}$$

where the powers of t and the spatial differential operators are hidden in $\theta_{i,j}(t)$, equation (3.16) allows us to concentrate on powers of h_x and h_y. Since $h_x = 2^{-l}H$ and $h_y = 2^{-m}H$ we can rewrite (3.16) as

$$d^{l,m}(t) = \sum_{i=0}^{\infty}\sum_{j=0}^{\infty} H^{i+j}\epsilon_{i,j}^{l,m}(t), \tag{3.17}$$

where

$$\epsilon_{i,j}^{l,m}(t) \equiv 2^{-il-jm}R^{l,m}\theta_{i,j}(t)c(x,y,t). \tag{3.18}$$

Insertion of (3.17) into the expression for the combined discretization error (4.5) yields

$$\widehat{d}^{N,N} = \sum_{i,j} H^{i+j}\widehat{\epsilon}_{i,j}^{N,N},$$

where

$$\widehat{\epsilon}_{i,j}^{N,N} \equiv \sum_{l+m=N} P^{N,N}\epsilon_{i,j}^{l,m} - \sum_{l+m=N-1} P^{N,N}\epsilon_{i,j}^{l,m}.$$

We now focus on the contribution that a single error term $\epsilon_{i,j}^{l,m}$ makes to the combined discretization error, i.e., we analyze $\widehat{\epsilon}_{i,j}^{N,N}$. The error terms $\epsilon_{i,j}^{l,m}$ are prolongated onto the finest grid $\Omega^{N,N}$ with interpolation of order q, yielding interpolation errors $\zeta_{i,j}^{N,N}$ and grid functions $\xi_{i,j}^{N,N}$ that are free of interpolation errors, i.e.,

$$P^{N,N}\epsilon_{i,j}^{l,m} = \xi_{i,j}^{N,N} + \zeta_{i,j}^{N,N}.$$

For $\widehat{\epsilon}_{i,j}^{N,N}$ this leads to the splitting

$$\widehat{\epsilon}_{i,j}^{N,N} = \widehat{\xi}_{i,j}^{N,N} + \widehat{\zeta}_{i,j}^{N,N}.$$

Error without interpolation effects.

According to (3.18) we have

$$\xi_{i,j}^{N,N} \equiv 2^{-il-jm} R^{N,N}\theta_{i,j}c,$$

hence

$$\widehat{\xi}_{i,j}^{N,N} = \left(\sum_{l+m=N} - \sum_{l+m=N-1}\right) 2^{-il-jm} R^{N,N}\theta_{i,j}c,$$

which is equivalent to

$$\begin{aligned} \widehat{\xi}_{i,j}^{N,N} &= \left(\sum_{l=0}^{N} 2^{-il-j(N-l)} - \sum_{l=0}^{N-1} 2^{-il-j(N-1-l)}\right) R^{N,N}\theta_{i,j}c \\ &= \left(2^{-iN} + 2^{-jN}\left[1-2^{j}\right]\sum_{l=0}^{N-1} 2^{l(j-i)}\right) R^{N,N}\theta_{i,j}c. \end{aligned} \tag{3.19}$$

For $i = j$ this yields

$$\widehat{\xi}_{i,i}^{N,N} = \left(2^{-iN} + 2^{-iN}\left[1-2^{i}\right]N\right) R^{N,N}\theta_{i,j}c, \tag{3.20}$$

while for $i \neq j$

$$\widehat{\xi}_{i,j}^{N,N} = \left(\frac{1}{2^{j}-2^{i}}\left[2^{-jN}\left(2^{i+j}-2^{i}\right) + 2^{-iN}\left(2^{j}-2^{i+j}\right)\right]\right) R^{N,N}\theta_{i,j}c. \tag{3.21}$$

Equations (3.20) and (3.21) lead to the following order estimates

$$\widehat{\xi}_{i,j}^{N,N} = \begin{cases} \mathcal{O}\left(2^{-jN}\right) & \text{if } i = 0, j \neq 0. \\ \mathcal{O}\left(2^{-iN}\right) & \text{if } j = 0, i \neq 0. \\ \mathcal{O}\left(N2^{-iN}\right) & \text{if } i = j \neq 0. \\ \mathcal{O}\left(2^{-\min(i,j)N}\right) & \text{if } i \neq j, i \neq 0, j \neq 0. \end{cases} \tag{3.22}$$

Additional error due to interpolation.

In leading order the interpolation error is given by

$$\zeta_{i,j}^{N,N} = \left(\lambda_l h_x^q \partial_x^q + \lambda_m h_y^q \partial_y^q\right) \xi_{i,j}^{N,N},$$

or equivalently,

$$\zeta_{i,j}^{N,N} = H^q R^{N,N} \left(2^{-(q+i)l-jm} \lambda_l \partial_x^q + 2^{-(q+j)m-il} \lambda_m \partial_y^q\right) \theta_{i,j} c,$$

where the λ_l and λ_m are coefficients dependent on l and m respectively and on the choice of interpolation. For the combined interpolation error $\hat{\zeta}_{i,j}^{N,N}$ we have

$$\begin{aligned}\hat{\zeta}_{i,j}^{N,N} &= H^q R^{N,N} \left(\sum_{l+m=N} - \sum_{l+m=N-1}\right) 2^{-(q+i)l-jm} \lambda_l \partial_x^q \theta_{i,j} c \\ &\quad + H^q R^{N,N} \left(\sum_{l+m=N} - \sum_{l+m=N-1}\right) 2^{-(q+j)m-il} \lambda_m \partial_y^q \theta_{i,j} c.\end{aligned}$$

For the first term,

$$\left(\sum_{l+m=N} - \sum_{l+m=N-1}\right) 2^{-(q+i)l-jm} \lambda_l \partial_x^q \theta_{i,j} c,$$

we obtain

$$\left(2^{-(q+i)N} \lambda_N + \sum_{l=0}^{N-1} \left(2^{-(q+i)l-j(N-l)} - 2^{-(q+i)l-j(N-1-l)}\right) \lambda_l\right) \partial_x^q \theta_{i,j} c,$$

which, in absolute value, is bounded from above by

$$|\lambda|_{\max} \left| \left(2^{-(q+i)N} + \sum_{l=0}^{N-1} \left(2^{-(q+i)l-j(N-l)} - 2^{-(q+i)l-j(N-1-l)}\right)\right) \partial_x^q \theta_{i,j} c \right|.$$

Likewise, the second term,

$$\left(\sum_{l+m=N} - \sum_{l+m=N-1}\right) 2^{-(q+j)m-il} \lambda_m \partial_x^q \theta_{i,j} c,$$

is in absolute value bounded from above by

$$|\lambda|_{\max} \left| \left(2^{-(q+j)N} + \sum_{m=0}^{N-1} \left(2^{-(q+j)m-i(N-m)} - 2^{-(q+j)m-i(N-1-m)}\right)\right) \partial_y^q \theta_{i,j} c \right|.$$

Together these bounds lead to the following order estimates, in the same way as the estimates in the previous section were obtained

$$\widehat{\zeta}_{i,j}^{N,N} = \begin{cases} \mathcal{O}\left(H^q 2^{-qN}\right) & \text{if } i = 0 \text{ or } j = 0. \\ \mathcal{O}\left(H^q N 2^{-jN}\right) & \text{if } q + i = j. \\ \mathcal{O}\left(H^q N 2^{-iN}\right) & \text{if } q + j = i. \\ \mathcal{O}\left(H^q 2^{-\min(i,j)N}\right) & \text{if } 0 \neq j \neq q + i \text{ and } 0 \neq i \neq q + j. \end{cases} \tag{3.23}$$

3.4.2 Leading-order results

By combining the order estimates for a single error term (3.22) and equations (3.20) and (3.21) with the structure of a dimensionally split discretization error (3.8), we see that in the discretization error the following terms are of particular interest

$$\begin{aligned} d &= t(\alpha_p a h_x^p \partial_x^{p+1} + \beta_p b h_y^p \partial_y^{p+1})c \\ &\quad + t^2 \alpha_p \beta_p a b h_x^p h_y^p \partial_x^{p+1} \partial_y^{p+1} c + \mathcal{O}\left(h_x^{p+1}\right) + \mathcal{O}\left(h_y^{p+1}\right). \end{aligned} \tag{3.24}$$

We have omitted the upper indices N, N. Equation (3.24) leads to the following leading-order expression for the combined discretization error

$$\begin{aligned} \widehat{d} &= t(\alpha_p a h^p \partial_x^{p+1} + \beta_p b h^p \partial_y^{p+1})c \\ &\quad + t^2 \alpha_p \beta_p a b H^p h^p (1 + (1 - 2^p) \log_2 \tfrac{H}{h}) \partial_x^{p+1} \partial_y^{p+1} c + \mathcal{O}\left(h^{p+1} \log_2 \tfrac{1}{h}\right). \end{aligned} \tag{3.25}$$

More specifically, for the third-order upwind scheme,

$$\widehat{d} = -\frac{th^3}{12}(|a|\,\partial_x^4 + |b|\,\partial_y^4)c + \frac{t^2}{144}\,|ab|\,H^3 h^3 (1 - 7\log_2)\partial_x^4 \partial_y^4 c + \mathcal{O}\left(h^4 \log_2 \frac{1}{h}\right). \tag{3.26}$$

3.4.3 Mapping of error terms

We illustrate the effect of a single term of the discretization error on the error that is observed on the finest grid after applying the combination technique. We view the combination technique as a mapping that maps terms from the discretization error onto a leading-order error term on the finest grid. We assume that the order of the prolongation q is greater than the order of the discretization p. The order estimate (3.22) shows that, for $i \neq j$, $i \neq 0$, $j \neq 0$, we have a mapping according to Table 3.1. While the discretization error's leading-order terms, proportional to h_x^p and h_y^p yield error terms of $\mathcal{O}(h^p)$, the cross-derivative term proportional to $h_x^p h_y^p$ surpasses these and yields the new formal leading-order error term proportional to $h^p \log h^{-1}$.

Table 3.1: Mapping of error terms from the semi-coarsened grids to the finest grid.

Error term on $\{\Omega^{l,m}\}$	Effect on $\Omega^{N,N}$
h_x^i or h_y^i	$\mathcal{O}(h^i)$
$h_x^i h_y^j$	$\mathcal{O}(h^{\min(i,j)})$
$h_x^i h_y^i$	$\mathcal{O}(h^i \log h^{-1})$

3.4.4 Additional error due to interpolation

From the order estimates (3.23) we find that:

- if $q \neq p$ then the contribution of the interpolation error is
$$\mathcal{O}\left(H^p h^q\right), \tag{3.27}$$
- if $q = p$ then the contribution of the interpolation error is
$$\mathcal{O}\left(H^p h^p \log \frac{H}{h}\right). \tag{3.28}$$

According to (3.27) the interpolation leaves the leading-order result (3.25) unaffected, provided the order of interpolation q is greater than the order of discretization p. When $q = p$, according to (3.28), the effect of the interpolation is of the same order as the second term in the leading-order result (3.25). For $q < p$ the interpolation error is in fact larger than the leading-order result (3.25) itself. Thus choosing $q < p$ is not sensible since it leads to an order reduction in the error. Choosing $q = p$ is acceptable when the parameters of the combination technique are such that the second term in (3.25) is dominated by the first term. When this is not the case, q must be chosen larger than p.

3.4.5 Intermediate combinations

When the combination technique is used in conjunction with a time-stepping technique, like we do, then we can choose to make intermediate combinations. At an intermediate combination the solutions on the semi-coarsened grids are combined onto the finest grid and then the fine-grid function is projected back onto the semi-coarsened grids. We will now analyze the influence of intermediate combinations on the error, specifically we consider $M-1$ intermediate combinations made at times $\frac{t}{M}, \frac{2t}{M}, \cdots, \frac{(M-1)t}{M}$. For a single semi-coarsened grid $\Omega^{l,m}$ onto which an intermediate solution was restricted at $\frac{t}{M}$, we have, according to (3.6),

$$d^{l,m}\left(\frac{2t}{M}\right) = \sum_{j=0}^{\infty} \frac{\left(-\frac{t}{M}E\right)^j}{j!} e^{-\frac{t}{M}(a\partial_x + b\partial_y)} R^{l,m} \widehat{d}^{N,N}\left(\frac{t}{M}\right) + \sum_{i=1}^{\infty} \frac{\left(-\frac{t}{M}E\right)^i}{i!} R^{l,m} c\left(\frac{2t}{M}\right). \tag{3.29}$$

Due to the leading order result (3.25) we have

$$\begin{aligned} e^{-\frac{t}{M}(a\partial_x + b\partial_y)} R^{l,m} \widehat{d}^{N,N}(\frac{t}{M}) &= \frac{t}{M}(\alpha_p a h^p \partial_x^{p+1} + \beta_p b h^p \partial_y^{p+1}) R^{l,m} c(\frac{2t}{M}) \\ &+ \frac{t^2}{M^2} \alpha_p \beta_p a b H^p h^p (1 + (1-2^p)\log_2 \frac{H}{h}) \\ &\partial_x^{p+1} \partial_y^{p+1} R^{l,m} c(\frac{2t}{M}) + \mathcal{O}\left(h^{p+1} \log_2 \frac{1}{h}\right). \end{aligned}$$

Here we have used $e^{-\frac{t}{M}(a\partial_x + b\partial_y)} c(\frac{t}{M}) = c(\frac{2t}{M})$. In the first summation in (3.29), terms with $j > 0$ will only contribute in higher order because E is a power expansion in mesh widths h_x and h_y. Hence we can neglect the $j > 0$ terms in (3.29) for a leading-order result, yielding

$$\begin{aligned} d^{l,m}(\tfrac{2t}{M}) &= \tfrac{t}{M}(\alpha_p a h^p \partial_x^{p+1} + \beta_p b h^p \partial_y^{p+1}) R^{l,m} c(\tfrac{2t}{M}) \\ &+ \tfrac{t^2}{M^2} \alpha_p \beta_p a b H^p h^p (1 + (1-2^p)\log_2 \tfrac{H}{h}) \partial_x^{p+1} \partial_y^{p+1} R^{l,m} c(\tfrac{2t}{M}) \\ &+ \textstyle\sum_{i=1}^{\infty} \frac{(-\frac{t}{M}E)^i}{i!} R^{l,m} c(\tfrac{2t}{M}) \\ &+ \mathcal{O}\left(h^{p+1} \log_2 \tfrac{1}{h}\right) + \mathcal{O}\left(\left(h_x^p + h_y^p + h_x^p h_y^p\right)\left(h^p + h^p \log_2 \tfrac{1}{h}\right)\right). \end{aligned} \tag{3.30}$$

The above expression immediately leads to the leading-order result for the combined discretization error $\widehat{d}^{N,N}(\frac{2t}{M})$ taking into account an intermediate combination at $\frac{t}{M}$. The first two terms and the $\mathcal{O}\left(h^{p+1} \log_2 \frac{1}{h}\right)$ term carry over into $\widehat{d}^{N,N}(\frac{2t}{M})$ without alterations since we neglect representation errors. The summation yields the two terms in (3.25) as was argued in Sections 3.4.1 and 3.4.2. The last $\mathcal{O}$-term translates according to the rules stated in Section 3.4.1. Thus, (3.30) yields the following for the combined discretization error $\widehat{d}^{N,N}(\frac{2t}{M})$ taking into account an intermediate combination at $\frac{t}{M}$:

$$\begin{aligned} \widehat{d}^{N,N}(\tfrac{2t}{M}) &= 2\Big[\tfrac{t}{M}(\alpha_p a h^p \partial_x^{p+1} + \beta_p b h^p \partial_y^{p+1}) R^{N,N} c(\tfrac{2t}{M}) \\ &+ \tfrac{t^2}{M^2} \alpha_p \beta_p a b H^p h^p (1 + (1-2^p)\log_2 \tfrac{H}{h}) \partial_x^{p+1} \partial_y^{p+1} R^{N,N} c(\tfrac{2t}{M}) \\ &+ \mathcal{O}\left(h^{p+1} \log_2 \tfrac{1}{h}\right)\Big] \\ &+ \mathcal{O}\left(\left(h^p + h^p + h^p \log_2 \tfrac{1}{h}\right)\left(h^p + h^p \log_2 \tfrac{1}{h}\right)\right). \end{aligned}$$

By induction this leads to the following result for the combined discretization error at t, taking into account intermediate combinations at $\frac{t}{M}, \frac{2t}{M}, \cdots, \frac{(M-1)t}{M}$,

$$\begin{aligned} \widehat{d}^{N,N}(t) &= t(\alpha_p a h^p \partial_x^{p+1} + \beta_p b h^p \partial_y^{p+1}) R^{N,N} c(t) \\ &+ \tfrac{1}{M} t^2 \alpha_p \beta_p a b H^p h^p (1 + (1-2^p)\log_2 \tfrac{H}{h}) \partial_x^{p+1} \partial_y^{p+1} R^{N,N} c(t) \\ &+ \mathcal{O}\left(h^{p+1} \log_2 \tfrac{1}{h}\right), \end{aligned} \tag{3.31}$$

i.e., the term proportional to $h^p \log h^{-1}$ is attenuated by a factor $\frac{1}{M}$. For the third-order upwind discretization equation (3.31) yields

$$\begin{aligned}\hat{d} &= -\tfrac{th^3}{12}(|a|\,\partial_x^4 + |b|\,\partial_y^4)c + \tfrac{t^2}{144M}\,|ab|\,H^3h^3(1-7\log_2\tfrac{H}{h})\partial_x^4\partial_y^4 c \\ &\quad +\mathcal{O}\left(h^4\log_2\tfrac{1}{h}\right).\end{aligned} \qquad (3.32)$$

3.4.6 Qualitative behavior of the error

Provided the effects of interpolation can be neglected the error in the combined solution is given by (3.31). The competition between the two terms in (3.31) is determined by the time up to which we integrate, the number of combinations M, the coefficients a and b, the root mesh width H, the number of grids (through $\log_2\frac{H}{h}$), the order of discretization p (through α_p, β_p and 2^p) and by the derivatives of the exact solution. Given this multitude of dependencies it seems likely that in general both terms can be important in describing the error.

When $a \approx b$ (i.e. advection diagonal to the grid) or when the exact solution has a large cross derivative $\partial_x^{p+1}\partial_y^{p+1}c$ compared to the derivatives $\partial_x^{p+1}c$ and $\partial_y^{p+1}c$, then the second term in (3.31) gains importance. Since this term represents the additional error due to using the combination technique, rather than a single grid, we see that the combination technique is less well suited to problems with $a \approx b$ or with large cross derivatives. Both are features of a problem that is not grid-aligned, i.e., the combination technique works better for grid-aligned problems.

We mention two mechanisms that will attenuate the second term in (3.31). First, the semi-coarsened grids used in the combination technique need to be sufficiently fine to describe the solution. This requires H to be small and thus attenuates the second term in (3.31), which has H^p as a prefactor . Second, it is a practical observation that a number of intermediate combinations $(M-1)$ is needed to successfully apply the combination technique, causing a further reduction of the second term by a factor $1/M$.

3.4.7 Time-dependent coefficients

Up to now the results in the current section are valid for coefficients that are independent of time. We now state the leading-order results for time-dependent coefficients. The statements about the interpolation error still hold. The leading-order expression for the combined discretization error becomes

$$\begin{aligned}\hat{d} &= \left(\int_0^t \alpha_p(t')a(t')dt'\right)h^p\partial_x^{p+1}c + \left(\int_0^t \beta_p(t')b(t')dt'\right)h^p\partial_y^{p+1}c \\ &\quad + \left(\int_0^t \alpha_p(t')a(t')dt'\right)\left(\int_0^t \beta_p(t')b(t')dt'\right)H^ph^p(1+(1-2^p)\log_2\tfrac{H}{h}) \\ &\quad \partial_x^{p+1}\partial_y^{p+1}c + \mathcal{O}\left(h^{p+1}\log_2\tfrac{1}{h}\right).\end{aligned}$$

For third-order upwind discretization this yields

$$\begin{aligned}
\widehat{d} &= -\tfrac{h^3}{12}\left(\int_0^t |a(t')|\,dt'\,\partial_x^4 + \int_0^t |b(t')|\,dt'\,\partial_y^4\right) c \\
&\quad + \tfrac{H^p h^p}{144}(1+(1-2^p)\log_2 \tfrac{H}{h})\left(\int_0^t |a(t')|\,dt'\right)\left(\int_0^t |b(t')|\,dt'\right)\partial_x^4\partial_y^4 c \\
&\quad + \mathcal{O}\left(h^4 \log_2 \tfrac{1}{h}\right).
\end{aligned} \tag{3.33}$$

When $M-1$ intermediate combinations are made the combined discretization error is given by

$$\begin{aligned}
\widehat{d} &= \left(\int_0^t \alpha_p(t')a(t')dt'\right) h^p \partial_x^{p+1} c + \left(\int_0^t \beta_p(t')b(t')dt'\right) h^p \partial_y^{p+1} c \\
&\quad + \left(\sum_{n=0}^{M-1}\left(\int_{\frac{n}{M}t}^{\frac{n+1}{M}t} \alpha_p(t')a(t')dt'\right)\left(\int_{\frac{n}{M}t}^{\frac{n+1}{M}t} \beta_p(t')b(t')dt'\right)\right) \\
&\quad H^p h^p (1+(1-2^p)\log_2 \tfrac{H}{h})\partial_x^{p+1}\partial_y^{p+1} c + \mathcal{O}\left(h^{p+1}\log_2 \tfrac{1}{h}\right).
\end{aligned}$$

For third-order upwind discretization this yields

$$\begin{aligned}
\widehat{d} &= -\ \tfrac{h^3}{12}\left(\int_0^t |a(t')|\,dt'\,\partial_x^4 + \int_0^t |b(t')|\,dt'\,\partial_y^4\right) c \\
&\quad + \tfrac{H^p h^p}{144}(1+(1-2^p)\log_2 \tfrac{H}{h})\sum_{n=0}^{M-1}\left(\int_{\frac{n}{M}t}^{\frac{n+1}{M}t} |a(t')|\,dt'\right) \\
&\quad \left(\int_{\frac{n}{M}t}^{\frac{n+1}{M}t} |b(t')|\,dt'\right)\partial_x^4\partial_y^4 c + \mathcal{O}\left(h^{p+1}\log_2 \tfrac{1}{h}\right).
\end{aligned} \tag{3.34}$$

3.5 Asymptotic efficiency

When making efficiency comparisons the number of cell updates C is used as a measure of required computational work. On a single grid this is simply defined as the product of the number of cells and the number of time steps required. Within the combination technique it is the sum of products of cells and time steps required on all grids within the grid of grids.

The cost estimates presented in this section are based on $\Delta t = 0.1\min(h_x, h_y)$, as are the numerical results in Section 3.6. Note that the time steps on the different grids within the combination technique are not equal, i.e., larger steps are taken on coarser grids. We identify a combination technique with a root mesh width $H = 2\cdot 2^{-L_R}$, where L_R is the *root level*, and a finest mesh width $h = 2\cdot 2^{-L_R-N}$, where N is the *sparseness level*. The number of grids within a combination technique is given by $2N+1 = 2\log_2(H/h)+1$.

3.5.1 Computational work

For a single grid with $h = 2 \cdot 2^{-L}$ the number of cell updates required is given by

$$C_1 = 5 \cdot 2^{3L}.$$

For the combination technique the number of cell updates is given by

$$C_{CT} = \begin{cases} 5 \cdot 2^{3L_R}\left(5 \cdot 2^{2N} - 4 \cdot 2^{3N/2}\right), & \text{for } N \text{ even.} \\ 5 \cdot 2^{3L_R}\left(5 \cdot 2^{2N} - \frac{11}{4} \cdot 2^{(3N+1)/2}\right), & \text{for } N \text{ odd.} \end{cases}$$

For fixed L_R the combination technique has asymptotic complexity

$$C_{CT} \sim 2^{2N} \sim h^{-2} \tag{3.35}$$

while the single grid has asymptotic complexity

$$C_1 \sim 2^{3L} \sim h^{-3}. \tag{3.36}$$

3.5.2 Efficiency comparison

For fixed L_R the combination technique has, according to (3.25), the following asymptotic error

$$\hat{d} \sim h^p \log_2(h^{-1}) \sim 2^{-pN}N$$

while a single grid of mesh width $h = 2 \cdot 2^{-L}$ has the following asymptotic error

$$d \sim h^p \sim 2^{-pL}.$$

If we require a single grid to yield the same error as the combination technique for a given N, i.e., we put

$$N2^{-pN} \sim 2^{-pL}$$

then we obtain

$$L = N - \frac{\log_2 N}{p}.$$

According to (3.36) this yields, for the complexity of the single grid,

$$C_1 \sim 2^{3N}\left(\frac{1}{N}\right)^{3/p} \sim h_{CT}^{-3}\left(\log_2(h_{CT}^{-1})\right)^{-3/p},$$

while according to (3.35) , the complexity of the combination technique is given by

$$C_{CT} \sim 2^{2N} \sim h_{CT}^{-2}$$

showing that, asymptotically, the combination technique reduces the three-dimensional single-grid complexity to a two-dimensional complexity, while obtaining the same level of accuracy.

3.6 Numerical results

3.6.1 Numerical setup

All the numerical results presented in this paper were obtained with fourth-order explicit Runge-Kutta time integration with time step $\Delta t = 0.1 \min(h_x, h_y)$ which satisfies the CFL condition for all considered test cases. Furthermore, the time-discretization error is always negligible compared to the spatial discretization error. For spatial discretization we have used third-order upwind discretization as described in Section 3.2.2, the prolongations are done with fourth-order interpolation. All analytical error predictions for the combination technique refer solely to the combined discretization error. The interpolation and representation errors due to the combination technique are neglected.

3.6.2 Test cases

We consider the following four test cases :

1. Horizontal advection, characterized by $a = 1/2, b = 0$.
2. Diagonal advection with $a = b = 1/2$.
3. Time-dependent advection with

$$(a,b) = \begin{cases} (0,2), & 0 \leq t < 1/4. \\ (2,0), & 1/4 \leq t < 1/2. \\ (0,-2), & 1/2 \leq t < 3/4. \\ (-2,0), & 3/4 \leq t < 1. \end{cases}$$

4. The Molenkamp-Crowley test case with $a = 2\pi y, b = -2\pi x$.

Test cases 1-3 have as initial profile

$$c(x,y,0) = 0.01^{4\left((x+0.25)^2+(y+0.25)^2\right)}, \tag{3.37}$$

which is depicted in Figure 3.2(a), while test case 4 has as initial profile

$$c(x,y,0) = 0.01^{4\left((x+0.5)^2+y^2\right)}, \tag{3.38}$$

which is depicted in Figure 3.2(d). All test cases are integrated up to $t = 1$ and have $-1 \leq x, y \leq 1$. In [9] solutions for the Molenkamp-Crowley test case obtained with various numerical methods are presented, given the initial condition (3.38).

Besides initial profiles, Figure 3.2 displays a number of typical error profiles observed in the numerical solutions of the test cases. The single-grid technique's (SG) results in Figure 3.2 were obtained on a 513×513 grid corresponding to $L = 9$ and the combination technique (CT) used a grid of 9 grids given by $L_r = 5$ and $N = 4$, i.e, the combination technique also produced its solutions on a 513 × 513 grid. The results for the combination technique with intermediate combinations (ICT) were obtained by making 8 combinations.

Figure 3.3 illustrates the performance of the single-grid and the combination technique on the test cases. The number of cell updates is plotted along the horizontal axis, which is a direct measure of the required CPU time, see Section 3.5.1. Any additional CPU time required to make the 7 intermediate combinations to obtain the ICT results was neglected, which is fully justified for the limited number of combinations considered here. The error is shown in the L_∞ norm, the results for the L_1 norm are similar. In obtaining Figure 3.3 the combination technique had $L_r = 5$ fixed and $N = 2, 3, 4, 5$. The single-grid results were obtained using $L = 7, 8, 9$.

In Figure 3.4 the effect of the number of combinations is shown on the L_∞ error due to a combination technique characterized by $L_r = 5$ and $N = 4$. In Figure 3.4 only test cases 2,3 and 4 are considered because for test case 1 the error is independent of the number of combinations.

Except for numerically observed results Figures 3.3 and 3.4 also contain analytical predictions. For test cases 1 and 2 these were obtained from (3.10) for the single grid, from (3.26) for the combination technique and from (3.32) for the combination technique with intermediate combinations. For test case 3 the error predictions were obtained from (3.11) for the single grid, from (3.33) for the combination technique and from (3.34) for the combination technique with intermediate combinations. Note that test case 4 is not time-dependent but spatially dependent. The error predictions that we have derived are not valid for spatially dependent coefficients.

3.6.3 Results

Horizontal test case.

We do not show any error profiles for the horizontal test case. For this test case the single-grid error and the errors due to the combination technique with and without intermediate combinations are all practically equal and are almost perfectly described by the analytical prediction (3.10). The combination technique does not introduce any additional error relative to the single grid because the second term in (3.26) vanishes due to $b = 0$. The combination technique works very well for this fully grid-aligned test case, as can be seen in Figure 3.3(a). Figure 3.3(a) also shows that intermediate combinations do not improve the efficiency for the horizontal test case. In fact, the ICT results coincide with the CT results.

Diagonal test case.

For the diagonal test case, error profiles are shown for the combination technique and the single grid in Figures 3.2(b) and 3.2(c) respectively. We see that for this test case the error due to the combination technique is somewhat larger than the single grid error and has a different shape. This figure also shows that the combination technique can be made more efficient by applying 8 combinations. Figure 3.4(a) shows how the error due to the combination technique decreases as the number of combinations is increased. The ICT error converges to the single-grid error as the number of combinations is increased. The first couple of combinations strongly decrease the error, a further increase in the number of combinations does not decrease the error much further.

Time-dependent test case.

For the time-dependent test case the error profiles for the CT and the ICT are plotted in Figures 3.2(e) and 3.2(f), respectively. We see that making intermediate combinations influences both the shape and size of the error. Note that Figures 3.3(b) and 3.3(c) are similar, i.e., just like the diagonal test case the time-dependent test case is solved more efficiently with intermediate combinations (ICT) than without (CT). However, the reason for the efficiency of the ICT is somewhat more complex for the time-dependent test case than for the diagonal test case. As we can see from Figure 3.4(b) the ICT error does not decrease monotonically with the number of combinations and this is correctly predicted by our theory. We can see that when a multiple of four combinations is made the ICT error becomes equal to the single grid error. This follows from (3.34) due to the fact that the product of integrals in the summation in the second term is always zero when a multiple of four combinations is made. When a multiple of four combinations is made the time-dependent test case is effectively split into two horizontal and two vertical advection problems and these are solved very well by the combination technique, as we know from the first test case.

For the time-dependent test case the agreement between predicted and observed error is very good for the single grid and the ICT. For the combination technique without intermediate combinations the agreement is a little weaker. This can be understood as follows. The combination technique tends to amplify cross-derivative terms in the single-grid error and of these amplified terms only one is included in our analytical predictions, viz. the second term in (3.26). The discrepancy between the predicted and observed CT errors is to be ascribed to the amplified cross-derivative terms that are not included in our analytical predictions. These terms are proportional to a second or higher power of t and are therefore, according to Section 3.4.5, inversely proportional to a first or higher power of M if M combinations are made . Hence, the terms that cause the discrepancy are significantly smaller for the ICT than for the CT, especially for higher numbers of combinations.

Molenkamp-Crowley test case.

Error profiles for the Molenkamp-Crowley test case are shown in Figures 3.2(g), 3.2(h) and 3.2(i) for the SG, CT and ICT, respectively. We see that the CT error is larger than the SG error, but intermediate combinations help considerably, i.e., the ICT error lies much closer to the SG error than to the CT error. Figure 3.3(d) shows that the Molenkamp-Crowley test case is a tough case to solve efficiently with the combination technique. Figure 3.3(d) shows that CT is less efficient than the single-grid technique, whereas ICT is more efficient in solving the Molenkamp-Crowley test case. For completeness, Figure 3.4(c) shows how the ICT error decreases with increasing number of combinations.

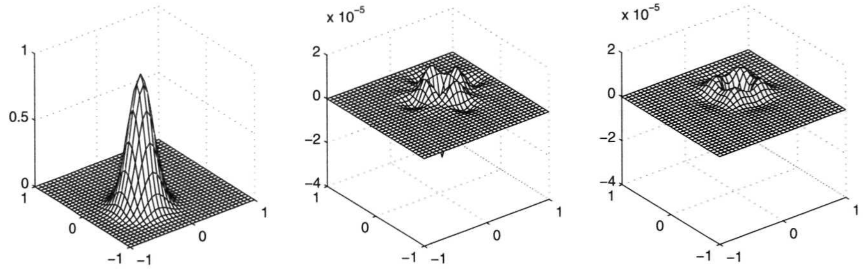
3.6.4 Implementational issues

Boundary complications.

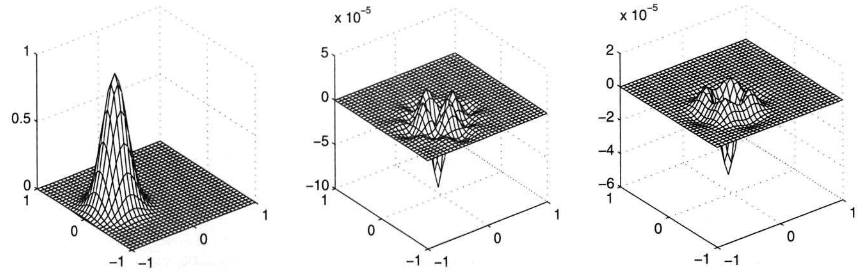
The L_∞ errors for the Molenkamp-Crowley test case were determined after the solutions were restricted to the 33×33 root grid. We were forced to do this because at high accuracies the fourth-order interpolation produced wiggles near the boundaries that dominate the combined discretization error. These wiggles do not appear in the nodes of the root grid, because for those nodes no interpolation is necessary. However, at very high resolution wiggles near the boundaries appear in the nodes of the root grid as well. In particular for $L_R \geq 6$ the wiggles are of equal or greater magnitude than the combined discretization error itself. The cause for these wiggles lies in the fact that the discretization near the boundaries is of lower order which obstructs the cancellation of errors required by the combination technique to function properly. An illustration of wiggles near the boundary is shown in Figure 3.5(b). Above difficulties were not observed for the other test cases because there the solutions stayed away from the boundaries. We also ran the Molenkamp-Crowley test case for the initial profile (3.37) shown in Figure 3.2(a) which stays away from the boundaries. This removed the problems near the boundaries but introduced a similar wiggle in the origin. We believe that this wiggle is also due to an order reduction caused by the switching of the upwind discretization stencil in horizontal and vertical directions due to the sign change of the coefficients in the origin.

Choosing an optimal root mesh-width.

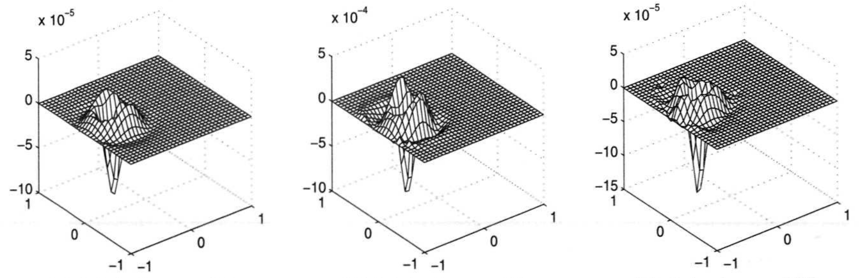
All numerical results for the combination technique were obtained with a root mesh width $H = 1/16$ corresponding to a root level $L_R = 5$. This choice was made to optimize the performance of the combination technique when applied to the Molenkamp-Crowley test case. This is illustrated in Figure 3.5(a). In this figure the performance of the combination technique with 8 combinations which has $L_R + N = 10$ fixed (ICT) is compared with the single-grid performance (SG). We



(A) Initial profile 1,2,3 (B) diagonal, CT (C) diagonal, SG



(D) initial profile 4 (E) time-dependent, CT (F) time-dependent, ICT



(G) Molenkamp, SG (H) Molenkamp, CT (I) Molenkamp, ICT

Figure 3.2: Initial profiles and numerically observed errors for the single-grid technique (SG), the combination technique (CT) and the combination technique with intermediate combinations (ICT), applied to the diagonal, time-dependent and Molenkamp-Crowley test cases.

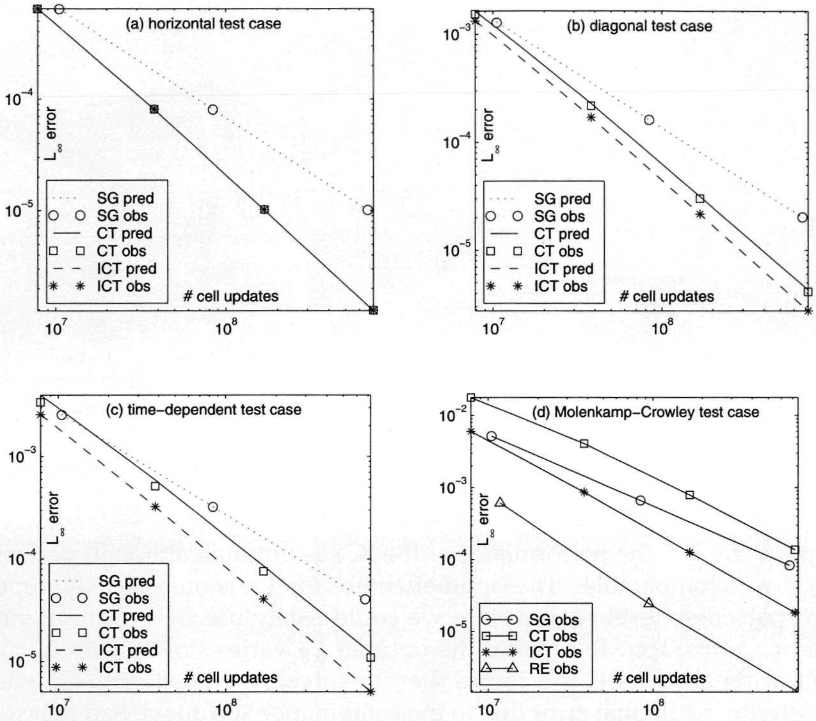


Figure 3.3: Numerically observed (obs) and analytically predicted (pred) performance of the single-grid technique (SG), combination technique (CT) and combination technique with intermediate combinations (ICT) applied to the test cases.

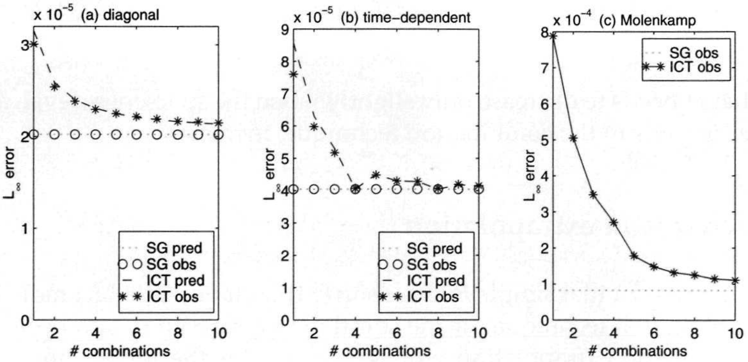
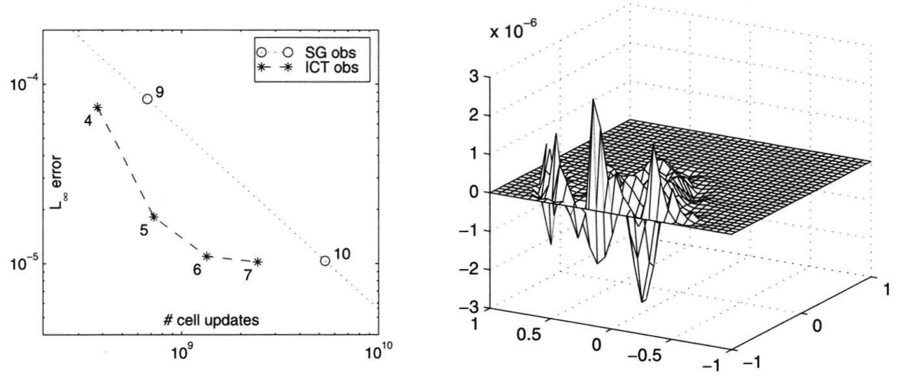


Figure 3.4: L_∞ error versus number of combinations for three test cases.



(A) Performance of the combination technique with 8 combinations (ICT) for root levels 4,5,6 and 7.

(B) Error profile due to a combination technique with root level 5, sparseness level 6 and 8 combinations.

Figure 3.5: Implementational issues; Molenkamp-Crowley test case.

see that for $L_R = 5$ the performance of the ICT is optimal, although performance for $L_R = 6$ is comparable. The optimal choice for L_R is only weakly dependent on the sparseness level N, therefore we could safely use $L_R = 5$ throughout for optimal performance. To see that the optimal L_R varies slowly with N consider the following argument. We found that, to solve the Molenkamp-Crowley test efficiently, the additional error due to the combination technique had to be of comparable magnitude as the single-grid error. According to our error analysis for constant coefficients (3.26) this implies

$$h^3 \sim H^3 h^3 \log_2 \frac{H}{h}$$

which leads to

$$H \sim \left(\frac{1}{N}\right)^{1/3},$$

showing that H needs to decrease only slightly when the sparseness level, and thus the number of grids in the combination technique, increases.

3.6.5 Richardson extrapolation

In [7] Rüde points out that simple Richardson extrapolation is in fact more efficient than the combination technique for the solution of a smooth Poisson problem. To see how Richardson extrapolation would perform for the Molenkamp-Crowley test case, we considered the following Richardson extrapolant

$$\omega_R^{N,N} \equiv \frac{8}{7}\omega^{N,N} - \frac{1}{7}P^{N,N}\omega^{N-1,N-1},$$

it cancels so the leading third-order term in the error expansion (3.9). The new leading-order terms are proportional to $h^4\partial_x^5 c$ and $h^4\partial_y^5 c$ and are thus of a dispersive nature which is shown in the $N = 9$ error profile for Richardson extrapolation in Figure 3.6. The Richardson extrapolant has an asymptotic error

$$d_{RE} \sim h_{RE}^4$$

while it has the same asymptotic complexity as a single grid,

$$C_{RE} \sim h_{RE}^{-3}.$$

If we consider a combination technique and a Richardson extrapolation of equal complexity, i.e., we put

$$C_{RE} \sim C_{CT}$$

then we obtain

$$h_{RE} \sim h_{CT}^{2/3}$$

which leads to

$$d_{RE} \sim h_{CT}^{8/3}. \tag{3.39}$$

According to (3.26) the combination technique has

$$\hat{d} \sim h_{CT}^3 \log h_{CT}^{-1}. \tag{3.40}$$

Comparison of (3.39) with (3.40) shows that in the limit $h \to 0$ the combination technique is more efficient than Richardson extrapolation.

In Figure 3.3(d)the numerically observed performance of Richardson extrapolation (RE) is compared with that of the single grid (SG) and the combination technique with intermediate combinations (ICT) when applied to the Molenkamp-Crowley test case. Figure 3.3(d) clearly shows that Richardson extrapolation is very efficient for the Molenkamp-Crowley test case, much more so than the combination technique, even though we expect the combination technique to be superior to Richardson extrapolation in the asymptotic limit $h \to 0$. For the Molenkamp-Crowley test case, without parallelization and on grids of practically relevant mesh width, the combination technique can not compete with Richardson extrapolation. Note that Richardson extrapolation and the combination technique strive for higher efficiency in different ways. Richardson extrapolation generates a higher-order solution for a marginally larger complexity, while the combination technique requires lower complexity for a marginally larger error.

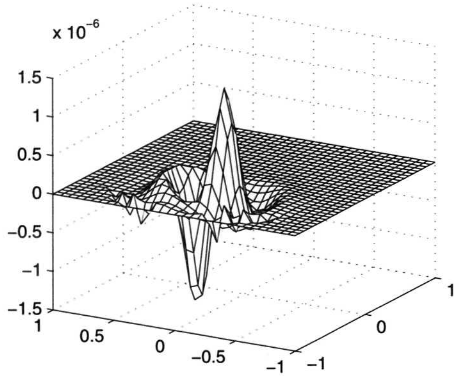


Figure 3.6: Error profile present in an $N = 9$ Richardson extrapolant.

3.7 Conclusions

We have derived leading-order expressions for the error that is introduced when a spatially constant coefficient advection equation is solved with the combination technique. In our derivations we have accounted for time-dependent coefficients and for intermediate combinations. When a constant coefficient advection equation

$$c_t + ac_x + bc_y = 0 \tag{3.41}$$

is solved on a grid of mesh width h, this will introduce an error d into the numerical solution which is in leading order given by

$$d = t\phi h^p(|a|\,\partial_x^{p+1} + |b|\,\partial_y^{p+1})c + \mathcal{O}\left(h^{p+1}\right), \tag{3.42}$$

where c is the exact solution, p is the order of discretization and ϕ is an error constant. We have shown that when we solve (3.41) with the combination technique, we obtain an error $\widehat{d}$ which is in leading order given by

$$\begin{aligned}\widehat{d} &= t\phi h^p(|a|\,\partial_x^{p+1} + |b|\,\partial_y^{p+1})c \\ &\quad + \tfrac{1}{M}t^2\phi^2\,|ab|\,H^p h^p(1 + (1-2^p)\log_2 \tfrac{H}{h})\partial_x^{p+1}\partial_y^{p+1}c + \mathcal{O}\left(h^{p+1}\log_2 \tfrac{1}{h}\right),\end{aligned} \tag{3.43}$$

where H is the mesh width of the coarsest grid in the combination technique and M is the number of combinations. We see that the leading-order term from the single grid error (3.42) reappears in the combination technique error (3.43) and is accompanied by a new term which is formally of order $h^p \log h^{-1}$. Focusing only on the order in terms of h, this new term has to be identified as the leading-order term in (3.43). The numerical experiments suggest, however, that the term proportional to h^p in (3.43), which is also present in the single-grid error, is of equal importance as the new term proportional to $h^p \log h^{-1}$. The additional error due to the combination technique, corresponding to the second term in (3.43), is proportional to $1/M$. This suggests that the error due to the combination technique can be strongly reduced by making a couple of intermediate combinations. The numerical results confirm this. For our test case that has time-dependent coefficients it turns out that the number of combinations has to be chosen such that the problem is split up in problems which have a constant direction of advection. This agrees with our error analysis. Finally, the combination technique proved more efficient for grid-aligned problems than for non-grid-aligned problems, which follows from numerical observations and from analysis.

For the Molenkamp-Crowley test simple Richardson extrapolation proved more efficient than the combination technique, even though the combination technique is expected to be more efficient in the asymptotic limit $h \rightarrow 0$. Rüde made the same observation for a smooth Poisson problem in [7].

When going to three spatial dimensions (or even higher dimensional problems), the combination technique will perform significantly better. Furthermore, very significant gains in performance can be obtained when the combination technique is parallelized.

BIBLIOGRAPHY

[1] H.J. Bungartz, M. Griebel, D. Roschke and C. Zenger, *Pointwise convergence of the combination technique for the Laplace equation*, East-West J. Numer. Math., Vol. 2, 21-45, 1994. Pages: 52

[2] C.T.H. Everaars and B. Koren, *Using coordination to parallelize sparse-grid methods for 3D CFD problems*, Parallel Computing, Vol. 24, 1081-1106, 1998. Pages: 52

[3] M. Griebel, *The combination technique for the sparse grid solution of pde's on multiprocessor machines*, Parallel Processing Letters, Vol. 2, 61-70, 1992. Pages: 52

[4] M. Griebel, M. Schneider and C. Zenger, A combination technique for the solution of sparse grid problems, in: R. Beauwens and P. de Groen, eds., *Iterative Methods in Linear Algebra*, 263-281 (North-Holland, Amsterdam, 1992). Pages: 52, 56

[5] M. Griebel and G. Zumbusch, Adaptive sparse grids for hyperbolic conservation laws, in: M. Fey and R. Jeltsch, eds., *Hyperbolic Problems: Theory, Numerics, Applications, International Series of Numerical Mathematics*, Vol. 129, 411-422 (Birkhäuser, Basel, 1999). Pages: 52

[6] B. Lastdrager and B. Koren, *Error analysis for function representation by the sparse-grid combination technique*, Report MAS-R9823, CWI, Amsterdam 1998. Pages: 53, 57

[7] U. Rüde, Multilevel, extrapolation and sparse grid methods, in: P.W. Hemker and P. Wesseling, eds., *Multigrid Methods IV, International Series of Numerical Mathematics*, Vol. 116, 281-294 (Birkhäuser, Basel, 1993). Pages: 52, 53, 72, 75

[8] J.G. Verwer, W.H. Hundsdorfer and J.G. Blom, *Numerical time integration for air pollution models*, Report MAS-R9825, CWI, Amsterdam, 1998. Pages: 52

[9] C.B. Vreugdenhil and B. Koren, eds., *Numerical Methods for Advection-Diffusion Problems, Notes on Numerical Fluid Mechanics*, Vol. 45 (Vieweg, Braunschweig, 1993). Pages: 66

[10] C. Zenger, Sparse grids, in: W. Hackbusch, ed., *Notes on Numerical Fluid Mechanics*, Vol. 31, 241-251 (Vieweg, Braunschweig, 1991). Pages: 52

4

SOLUTION OF TIME-DEPENDENT ADVECTION-DIFFUSION PROBLEMS WITH THE SPARSE-GRID COMBINATION TECHNIQUE AND A ROSENBROCK SOLVER

Abstract. In the current paper the efficiency of the sparse-grid combination technique applied to time-dependent advection-diffusion problems is investigated. For the time integration we employ a third-order Rosenbrock scheme implemented with adaptive step-size control and approximate matrix factorization. Two model problems are considered, a scalar 2D linear, constant-coefficient problem and a system of 2D nonlinear Burgers' equations. In short, the combination technique proved more efficient than a single grid approach for the simpler linear problem. For the Burgers' equations this gain in efficiency was only observed when one of the two solution components was set to zero, making the problem more grid-aligned.

4.1 Introduction

In modern CFD codes accurate resolution of thin solution layers is still very time consuming. Especially for high Reynolds numbers many grid points are needed to resolve the very thin layers. The common remedy is to use adapted grids that have small cells near the layers and large cells elsewhere. In this paper we investigate another approach to resolve the thin layers, namely the sparse grid combination technique (CT) as introduced by Griebel, Schneider and Zenger [4].

The CT is attractive because, asymptotically, it can yield a smaller spatial error for a given complexity than a single grid approach (SG) can [14], [1]. Consider a problem of spatial dimension d that is solved on a single grid with spatial discretization of order p, i.e., on a single grid of mesh width h the spatial error is $O(h^p)$. On a single grid this problem would have a complexity $\sim h^{-d}$. With the CT a spatial error of order $O(h^p(\log h)^{d-1})$ can be obtained with a complexity $\sim h^{-1}(\log h)^{d-1}$, i.e., an asymptotically first-order complexity is obtained with only a slightly larger error than for the SG. Furthermore, the CT can be easily and efficiently implemented on a parallel computer, see [3].

In [9] we investigated the efficiency of the CT when applied to a pure advection equation and concluded that for a non-grid-aligned solution the CT does not perform very well (see [9] for a more complete report). In [11] this was also found for some elliptic PDEs. Note that in [5] the CT is also applied to a pure advection equation, but here the efficiency of the CT is not considered.

In practice, advection-diffusion problems are usually solved on boundary-fitted grids. The corresponding solutions are usually grid-aligned. In this paper we study model advection-diffusion problems having this type of solution.

An essential ingredient for a CT solver for time-dependent problems is an efficient time accurate integrator. We use a three-stage, third-order Rosenbrock method implemented with built-in step-size control and approximate matrix factorization. Without step-size control the method can be implemented as a two-stage scheme. It uses approximate matrix factorization to greatly speed up the solution process, hence we call it factorized ROS3. In [7] the same factorized ROS3 has been used, independently from the current paper and without the CT.

As model problems we consider a scalar two-dimensional, constant-coefficient advection-diffusion equation and a system of two-dimensional Burgers' equations. To evaluate the efficiency of the CT we compare it with a straightforward SG approach.

4.2 The model problems

4.2.1 Model problem 1: The advection-diffusion equation

We consider the constant-coefficient advection-diffusion equation

$$u_t + u_x - \varepsilon\left(u_{xx} + u_{yy}\right) = 0 \tag{4.1}$$

on the spatial domain $[-1,1] \times [-1,1]$ and take $u(x,y,0) = 0$ as initial solution. As boundary conditions we impose

$$u(-1,y,t) = \begin{cases} 0, & y < 0 \\ \frac{1}{2}, & y = 0 \\ 1, & y > 0 \end{cases}, \quad u_y(x,\pm 1,t) = 0, \quad u(1,y,t) = 0.$$

For $\varepsilon = 10^{-2}$ the solution at $t = 1$ is shown in Fig. 1. It possesses a horizontal and a vertical grid-aligned solution layer. The thickness of both layers is proportional to $\sqrt{\varepsilon}$ as $\varepsilon \to 0$. For the steady state solution we have derived an exact expression in terms of a Fourier sum,

$$\begin{aligned} u(x,y) &= \frac{3/2}{\left(1 - e^{\frac{1}{\varepsilon}}\right)} e^{\frac{x}{\varepsilon}} \left(1 - e^{\frac{(1-x)}{\varepsilon}}\right) + \sum_{n=1}^{\infty} B_n(x) \cos\left(n\pi y\right), \\ B_n(x) &= \frac{2\sin\left(\frac{n\pi}{2}\right)/n\pi}{e^{2\sqrt{\frac{1}{4\varepsilon^2}+n^2\pi^2}} - 1} e^{\frac{x}{2\varepsilon}} \left(e^{x\sqrt{\frac{1}{4\varepsilon^2}+n^2\pi^2}} - e^{(2-x)\sqrt{\frac{1}{4\varepsilon^2}+n^2\pi^2}}\right), \end{aligned}$$

and have used this expression to confirm that our numerical method converges to the correct solution in the limit $t \to \infty$.

4.2.2 Model problem 2: Burgers' equations

The two-dimensional Burgers' equations

$$\begin{aligned} u_t &= -uu_x - vu_y + \varepsilon\left(u_{xx} + u_{yy}\right), \\ v_t &= -uv_x - vv_y + \varepsilon\left(v_{xx} + v_{yy}\right), \end{aligned}$$

are considered on the spatial domain $[-1,1] \times [-1,1]$. The boundary conditions we impose are

$$u(-1,y,t) = \begin{cases} 1 - 4(y - \frac{1}{2})^2, & y \geq 0 \\ 1 - 4(y + \frac{1}{2})^2, & y < 0 \end{cases}, \quad u(x,\pm 1,t) = 0, \quad u_x(1,y,t) = 0,$$

and

$$v(-1,y,t) = -0.35 \sin\left(\frac{1}{2}\pi y\right), \quad v_y(x,\pm 1,t) = 0, \quad v_x(1,y,t) = 0.$$

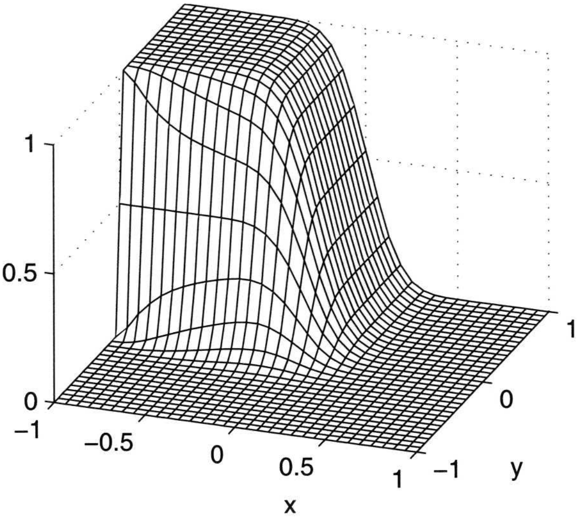


Figure 4.1: Solution of model problem 1 at $t = 1$ for $\epsilon = 0.01$

As initial solutions we take

$$\begin{aligned} u(x,y,0) &= \begin{cases} 1-4(y-\frac{1}{2})^2, & y \geq 0 \\ 1-4(y+\frac{1}{2})^2, & y < 0 \end{cases}, \\ v(x,y,0) &= -0.35 \sin\left(\frac{1}{2}\pi y\right). \end{aligned}$$

In Figures 2 and 3 the u and v components of the solution at $t = 3$ are shown for $\varepsilon = 10^{-2}$. The v component shows a sharpening from the sinusoidal inlet condition at $x = -1$ to a much steeper slope at the outflow boundary at $x = 1$. This is a grid-aligned phenomenon since near the outflow boundary the solution varies much stronger in y direction than in x direction. The u component shows a mixing of two jets. This phenomenon is not especially grid-aligned.

4.3 The sparse grid combination technique

In the CT several solutions on different grids are combined to get a solution which has the accuracy of a much finer grid. The two-dimensional CT is based on a grid of grids as shown in Fig. 4. Grids within the grid of grids are denoted by $\Omega^{l,m}$ where upper indices label the level of refinement relative to the *root grid* $\Omega^{0,0}$. The mesh-widths in x and y direction of $\Omega^{l,m}$ are $h_x = 2^{-l}H$ and $h_y = 2^{-m}H$, where H is the mesh width of the uniform root grid $\Omega^{0,0}$. We denote the mesh width of the finest grid $\Omega^{N,N}$ by h. Note that h_x and h_y are dependent on the position (l, m) in the grid of grids while h is not.

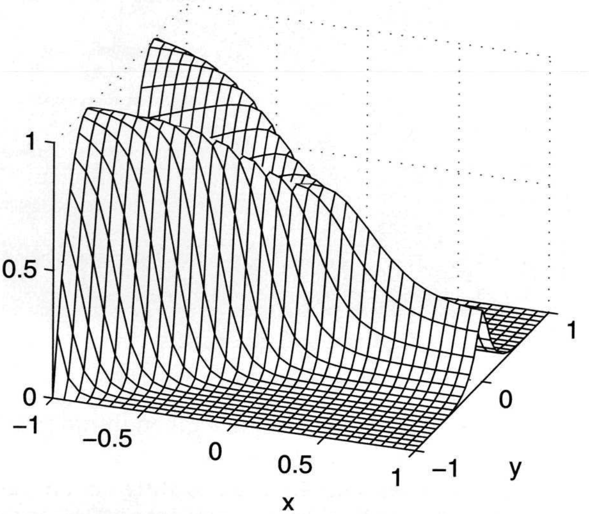


Figure 4.2: u-component of the solution of model problem 2 at $t = 3$ for $\epsilon = 0.01$

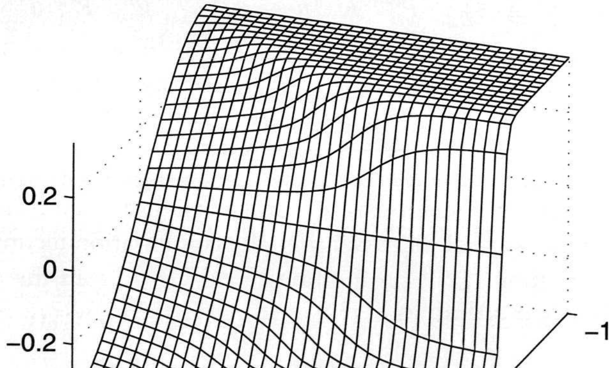
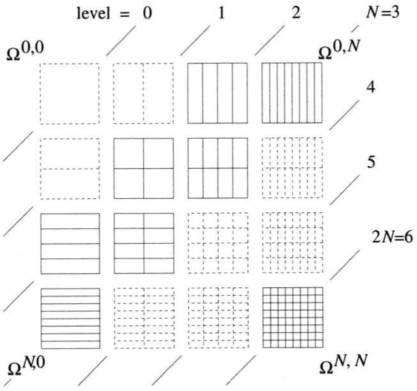


Figure 4.3: v-component of the solution of model problem 2 at $t = 3$ for $\epsilon = 0.01$



Notation	Description
$\Omega^{N,N}$	finest grid of mesh width $h = 2^{-N} H$
$\Omega^{l,m}$	semi-coarsened grid, of mesh widths $h_x = 2^{-l} H$ and $h_y = 2^{-m} H$
u	continuous, exact solution
$R^{l,m}$	restriction operator that maps onto $\Omega^{l,m}$
$P^{N,N}$	prolongation operator that maps onto $\Omega^{N,N}$
$U^{l,m}$	semi-discrete approximate solution on $\Omega^{l,m}$

Figure 4.4: Grid of grids

In the time-dependent combination technique a given initial profile $u(x,y,0)$ is restricted, by injection, onto the grids $\Omega^{N,0}, \Omega^{N-1,1}, \ldots, \Omega^{0,N}$ and onto $\Omega^{N-1,0}, \Omega^{N-2,1}, \ldots, \Omega^{0,N-1}$, see Fig. 4. The resulting coarse representations are then all evolved in time with our ROS3 time integrator. Then, at a chosen point in time, the coarse approximations are prolongated with q-th order interpolation onto the finest grid $\Omega^{N,N}$, where they are combined to obtain a more accurate solution. The notation is summarized in Fig. 4.

Considering the exact solution u, the combination technique, as introduced in [4], constructs a grid function $\widehat{u}^{N,N}$ on the finest grid $\Omega^{N,N}$ in the following manner,

$$\widehat{u}^{N,N} \equiv \sum_{l+m=N} P^{N,N} R^{l,m} u \; - \sum_{l+m=N-1} P^{N,N} R^{l,m} u.$$

The corresponding so-called *representation error* $r^{N,N}$ is

$$r^{N,N} \equiv \widehat{u}^{N,N} - R^{N,N} u. \tag{4.2}$$

Likewise, assuming exact time integration and considering semi-discrete solutions $U^{l,m}$, resulting from a spatial discretization, the combination technique constructs an approximate solution $\widehat{U}^{N,N}$ on the finest grid $\Omega^{N,N}$ from the coarse-grid approximate solutions according to

$$\widehat{U}^{N,N} = \sum_{l+m=N} P^{N,N} U^{l,m} - \sum_{l+m=N-1} P^{N,N} U^{l,m}. \tag{4.3}$$

Let $d^{l,m}$ denote the discretization error on grid $\Omega^{l,m}$, i.e.,

$$d^{l,m} \equiv U^{l,m} - R^{l,m} u. \tag{4.4}$$

The total error $e^{N,N} = \widehat{U}^{N,N} - R^{N,N} u$ present in $\widehat{U}^{N,N}$ is written as

$$e^{N,N} = r^{N,N} + \widehat{d}^{N,N},$$

where the *combined discretization* error $\widehat{d}^{N,N} = \widehat{U}^{N,N} - \widehat{u}^{N,N}$ is given by

$$\widehat{d}^{N,N} = \sum_{l+m=N} P^{N,N} d^{l,m} - \sum_{l+m=N-1} P^{N,N} d^{l,m}. \tag{4.5}$$

In [8] the representation error $r^{N,N}$ is analysed and in [10] an analysis is given of the combined discretization error $\widehat{d}^{N,N}$ for pure advection problems. In the next section we give similar results for the combined discretization error for our model problem 1, the linear, constant-coefficient advection-diffusion equation.

4.4 Spatial discretization errors

For the first test problem, the linear constant-coefficient advection-diffusion problem, we can derive an expansion in mesh widths for the spatial discretization error, as we did for the pure advection problem in [10]. Since essentially the same approach is used as in [10] we state only the results. We consider the error in the spatially discrete solution due to spatial discretization only, i.e., we assume here time integration to be exact. In (4.1) the diffusion terms are discretized with second-order central differences and the advection term is discretized with the third-order upwind biased discretization [6]. We only consider the error away from the boundaries, i.e., we neglect the influence of boundary conditions. When solved on a single grid with mesh widths h_x and h_y in x- and y-direction, the resulting spatial discretization error can then formally be expanded as

$$\begin{aligned} d(x,y,t) &= \sum_{i=1}^{\infty} \frac{(-tE_{adv} - tE_{diff})^i}{i!} u(x,y,t), \\ E_{adv} &= \sum_{j=3}^{\infty} \frac{-(-2)^j + 3(-1)^j + 1}{3(j+1)!} h_x^j \partial_x^{j+1}, \\ E_{diff} &= \varepsilon \sum_{j=2}^{\infty} \frac{(-1)^j + 1}{(j+2)!} \left(h_x^j \partial_x^{j+2} + h_y^j \partial_y^{j+2} \right), \end{aligned}$$

assuming that $u(x,y,t)$ is a C^∞ function. Neglecting $O(h_x^4)$ and $O(h_y^4)$ but including $O(h_x^2 h_y^2)$ for later comparison yields the following leading order expression

$$\begin{aligned} d(x,y,t) &= -\frac{t\varepsilon}{12} \left(h_x^2 \partial_x^4 + h_y^2 \partial_y^4 \right) u(x,y,t) - \frac{t}{12} h_x^3 \partial_x^4 u(x,y,t) \\ &\quad + \frac{t^2 \varepsilon^2}{144} h_x^2 h_y^2 \partial_x^4 \partial_y^4 u(x,y,t) + \mathcal{O}(h_x^4) + \mathcal{O}(h_y^4). \end{aligned}$$

Just as in [10] we use this result to determine the resulting spatial discretization error in the combined solution. It is given by

$$\begin{aligned}\widehat{d}(t) &= -\frac{t\varepsilon h^2}{12}\left(\partial_x^4+\partial_y^4\right)u(t)-\frac{th^3}{12}\partial_x^4 u(t) \\ &\quad +\frac{t^2\varepsilon^2}{144}H^2h^2(1-3\log_2\frac{H}{h})\partial_x^4\partial_y^4 u(t)+\mathcal{O}(h^3\log_2\frac{1}{h}).\end{aligned} \tag{4.6}$$

The first error term is the usual leading error term on $\Omega^{N,N}$ coming from the diffusion operator. Similarly, the second term comes from the advection operator. The third term comes forth from the mixing of diffusion in x- and y-direction in the combination process. Since there is only advection in the x-direction, advection does not produce any additional error in the combined solution. In order for the CT to be effective the third term should be small compared to the first two terms. Asymptotically (as h and H tend to zero) this is clearly the case. In practice the asymptotics are not always strong enough for the third term, and higher mixed terms, to be negligible.

4.5 The Rosenbrock solver ROS3

We consider autonomous ODE systems of the form

$$\frac{dU}{dt}=f(U),$$

which are supposed to result from spatial discretization on one of our grids and seek a numerical approximation $U_n \approx U(t)$ at $t=t_n$. To obtain this approximation we apply a third-order consistent two-stage Rosenbrock method, ROS3 (also being used in [7]), which can be written as

$$\begin{aligned}U_{n+1} &= U_n+\frac{5}{4}k_1+\frac{3}{4}k_2, \\ (I-\gamma\tau A)k_1 &= \tau F(U_n), \\ (I-\gamma\tau A)k_2 &= \tau F(U_n+\frac{2}{3}k_1)-\frac{4}{3}k_1,\end{aligned}$$

where $\tau=t_{n+1}-t_n$ is the step size and A is the Jacobian matrix $f'(U_n)$ or an $O(\tau)$ approximation thereof. This scheme is a variation to the scheme ROS2 as presented in [13] and belongs to a family of schemes discussed on p. 233 of [2]. Its stability function is

$$R(z)=\frac{1+(1-2\gamma)z+(\frac{1}{2}-2\gamma+\gamma^2)z^2}{(1-\gamma z)^2},$$

which shows that the scheme is A-stable if and only if $\gamma \geq 1/4$. The scheme is third-order accurate provided A is an $O(\tau)$ approximation of the Jacobian matrix and $\gamma=1/2+\sqrt{3}/6$. Note that this specific γ yields A-stability. Because our spatially discrete problems are stiff due to the diffusion term, A-stability is a desirable property.

4.5.1 Factorization

Since the ROS3 scheme remains of third-order for any $O(\tau)$ perturbation to $A = f'(U_n)$, we can split A as $A = A_1 + A_2$ and use

$$\begin{aligned} U_{n+1} &= U_n + \frac{5}{4}k_1 + \frac{3}{4}k_2, \\ (I - \gamma\tau A_1)(I - \gamma\tau A_2)k_1 &= \tau F(U_n), \\ (I - \gamma\tau A_1)(I - \gamma\tau A_2)k_2 &= \tau F(U_n + \frac{2}{3}k_1) - \frac{4}{3}k_1. \end{aligned}$$

The latter, factorized ROS3 scheme, is still of third-order since

$$(I - \gamma\tau A_1)(I - \gamma\tau A_2) = I - \gamma\tau(A - \gamma\tau A_1 A_2).$$

In the current work we use directional factorization to separate the horizontal and vertical coupling such that A_1 only couples unknowns in the horizontal direction and A_2 only couples unknowns in the vertical direction. This leads to enormous savings in required computational work since it reduces the two-dimensional linear algebra to one-dimensional linear algebra.

Without factorization, spatial discretization leads to pq coupled linear algebraic equations for the Rosenbrock vectors k_1 and k_2 where p is the number of unknowns in horizontal direction and q the number in vertical direction. With factorization, we have p sets of q coupled equations and q sets of p coupled equations for k_1 and likewise for k_2. This is a clear advantage of factorization since p sets of q coupled equations are solved much faster than one set of pq coupled equations. Another benefit of directional factorization is that the resulting sets of equations have band diagonal matrices and can therefore be solved very efficiently by means of LU decomposition.

In [7] it has been proven that a similar property as A-stability holds for the factorized ROS3 scheme. For our model problems this means that we have unconditional stability in the sense of Fourier-Von Neumann. Finally it should be noted that the above approximate matrix factorization is well known in the numeric PDE literature, see [7] for some references.

4.5.2 Time step size control

In our implementation of ROS3 we compute another auxiliary vector, k_3, to obtain an estimate for the local time error. The corresponding extra auxiliary equation is

$$(I - \gamma\tau A_1)(I - \gamma\tau A_2)k_3 = \tau F(U_{n+1}) + \frac{24\gamma^2 - 9\gamma - 1}{6\gamma(1 - 2\gamma)}k_1 + \frac{3\gamma - 1}{2\gamma(1 - 2\gamma)}k_2.$$

Our error estimate is

$$\begin{aligned} E_{est} &= -\frac{6\gamma^2-1}{6\gamma(1-2\gamma)}k_1 + \frac{6\gamma^2-6\gamma+1}{2\gamma(1-2\gamma)}k_2 - k_3 \\ &= \frac{1}{6}\tau^3\frac{d^3c}{ct^3} + O(\tau^4), \end{aligned}$$

which is the last term in the Taylor expansion of the updated solution that our scheme still handles correctly. We strive for an equidistribution of errors, i.e., we attempt to keep E_{est}, measured in the L_1 norm, fixed at some tolerance Tol during the integration. To achieve this we adjust the step size τ according to

$$\tau_{new} = 0.8\tau_{old}\left(\frac{Tol}{\|E_{est}\|_1}\right)^{1/3}.$$

Solution updates are only performed when $\|E_{est}\|_1 \leq Tol$ at the new time level, otherwise the update is computed again with a smaller step size. The factor 0.8 is a safety factor and serves to avoid excessive numbers of rejected updates. In our implementation the ratio τ_{new}/τ_{old} was kept bounded between 0.1 and 10.

Now consider the global time error e_n at time level t_n, i.e., the difference between the computed solution at time level t_n and the exact solution at the same time level. This error is in fact proportional to the tolerance Tol that we imposed, i.e.,

$$e_n \sim Tol.$$

This property of tolerance proportionality follows from [12], p. 350, when we identify our scheme as an XEPS scheme, i.e., an error per step control with local extrapolation. The proportionality between the imposed tolerance and the global time error is a nice property since it allows the user to control the global error in a very direct manner.

4.5.3 Numerical illustration of factorized ROS3

Figure 5 displays the integration history for the Burgers' equations solved up to $t = 3$ on a single 33×33 spatial grid with $Tol = 10^{-3}$. The step size τ is shown in the left graph and the error estimate $\|E_{est}\|_1$ in the right graph. We start with an initial step size $\tau = 10^{-2}$ which turns out to be somewhat too small for the imposed tolerance value. As the integration progresses larger step sizes are permissible. In the intermediate stage of the integration the step size remains almost constant. Finally, as the solution approaches steady state the size of the allowed step size quickly grows. During the integration the step size control keeps the error estimate $\|E_{est}\|_1$ at a nearly constant level, as can be seen from Fig. 5.

In Table 1 the ratio of maximal global time errors E_{Tol} is shown for solutions with tolerance Tol and tolerance $Tol/2$ as a function of the tolerance. The time errors were estimated by subtracting a reference solution obtained with $Tol = 10^{-8}$. As

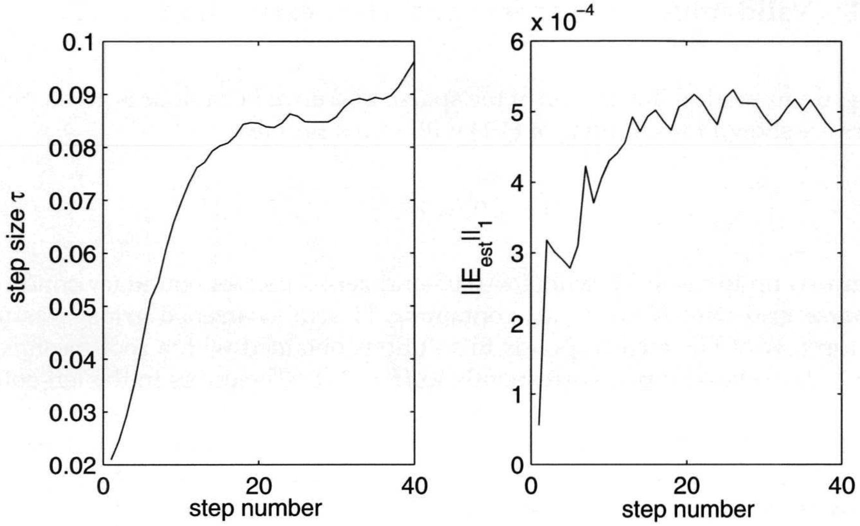


Figure 4.5: Integration history of model problem 2

the tolerance, and hence the step size, gets smaller we see that the ratio approaches 2, which confirms that the global time error is proportional to the imposed tolerance.

Tol	$L_\infty(E_{Tol})/L_\infty(E_{Tol/2})$
10^{-3}	1.748
10^{-4}	1.597
10^{-5}	1.878
10^{-6}	1.973

Table 4.1: Ratio of global time errors for model problem 2

4.6 Results

In this section the CT is compared with the standard SG approach. Both are implemented with the same spatial discretization, i.e., second-order central discretization for the diffusion operator and third-order upwind-biased discretization for the advection part. The Neumann condition on the outflow boundary in model problem 1 is only imposed on the diffusion operator to avoid spurious reflections at that boundary.

4.6.1 Validation of the sparse grid error expression

In Fig. 6 a numerical illustration of the sparse grid error behaviour is given. Spatial errors are shown for solutions of (4.1) with initial profile

$$u(x,y,0) = e^{-16(x^2+y^2)},$$

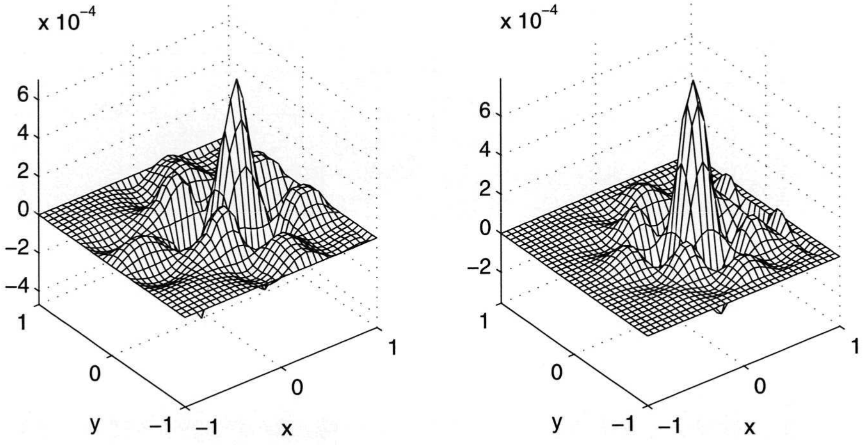
integrated up to $t = 0.25$, with $\varepsilon = 0.05$ and zero Dirichlet boundary conditions. A sparse grid with $N = 5$, i.e., containing 11 semi-coarsened grids, was used. The top row of Fig. 6 corresponds to solutions obtained with a root mesh width $H = 1/2$, the bottom row corresponds to $H = 1/8$. The errors in the left column were obtained numerically, i.e., by subtracting a reference solution obtained on a finer grid ($N = 5$, $H = 1/32$). The errors in the right column are predictions according to (4.6) where the derivatives of the solution were replaced by numerical differences of the reference solution.

The errors in the top row show oscillatory behaviour that is due to the third term in (4.6), i.e., the term due to combination. This behaviour is absent in the lower row. Here the third term, which is proportional to H^2, is negligible due to the smaller $H = 1/8$. The error prediction (4.6) illustrated in the right column clearly matches this transition in error behaviour.

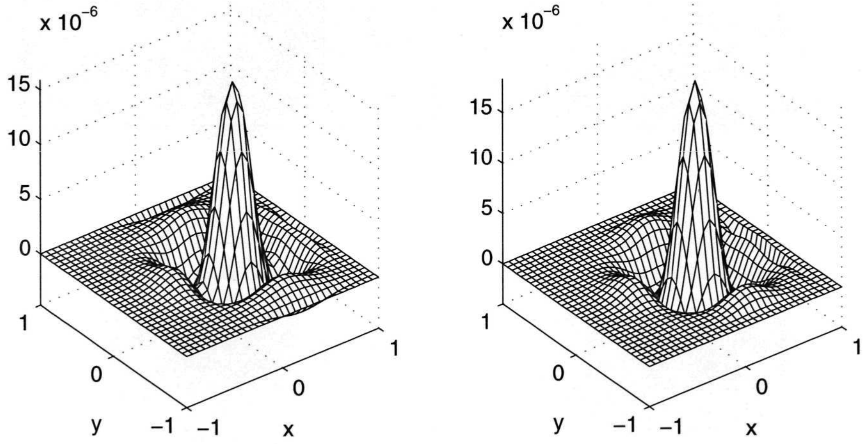
4.6.2 Model problem 1: the advection diffusion equation

In Fig. 7 the efficiency of the CT is compared with the SG when applied to the linear constant-coefficient advection-diffusion equation. Along the vertical axes the error is plotted, measured in the L_1 norm for the left column of graphs and in the L_∞ norm for the right column. Along the horizontal axes the computational work is plotted in terms of number of required cell updates. The graphs in the top, middle and bottom row correspond to $\varepsilon = 10^{-2}, 10^{-3}$ and 10^{-5}, respectively.

We see that for all these ε the CT is more efficient than the SG when we consider the errors in the L_1 norm. Also, the gain in efficiency becomes larger as ε is decreased. This is expected since for small ε the grid-aligned advection becomes more dominant rendering the test case more grid-aligned and hence better suited to the CT. For $\varepsilon = 10^{-3}$ and 10^{-5} the same holds for the L_∞ norm. For $\varepsilon = 10^{-2}$ the CT does not perform well when measured in the L_∞ norm. Examination of the corresponding spatial error distribution shows that the maximum error occurs near the discontinuity in the inlet condition. The mixed derivative u_{xxyy} is large near this discontinuity leading, for large ε, to a large term $\varepsilon^2 u_{xxyy}$ present in the spatial error due to the CT. Hence it is to be expected that for relatively large ε the CT performs poorly locally near the discontinuity.



Observed error, H=1/2 Predicted error, H=1/2



Observed error, H=1/8 Predicted error, H=1/8

Figure 4.6: Spatial errors

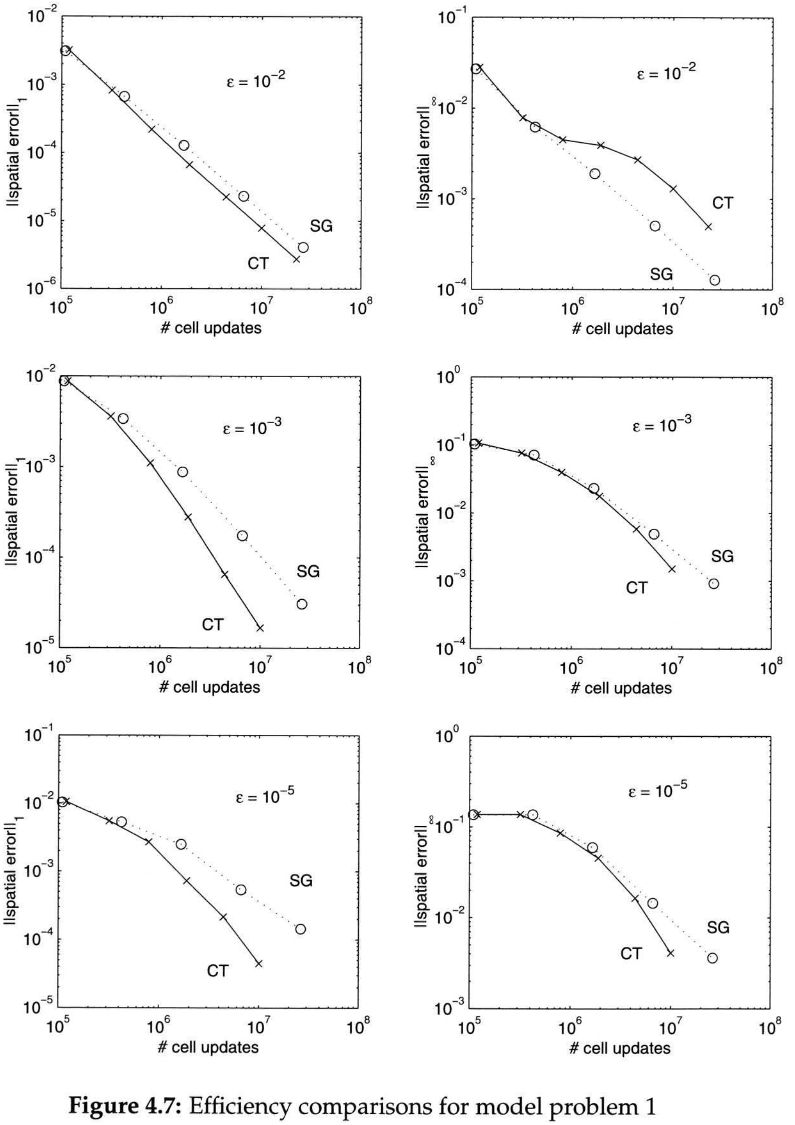


Figure 4.7: Efficiency comparisons for model problem 1

4.6.3 Model problem 2: Burgers' equations

In Fig. 8 again the CT and SG are compared in terms of efficiency, this time for the 2D Burgers' test case. In Fig. 8 the diffusion parameter is kept fixed at $\varepsilon = 10^{-2}$ since varying the diffusion parameter does not change the qualitative conclusions that can be drawn from this figure. The top row corresponds to the Burgers' test case as described in Section 4.2.2. For this test case it is clear that the CT does not perform very well relative to the SG, either when measured in L_1 norm or in L_∞ norm. It was expected that the Burgers' test case would be less well suited to the CT than the linear test case since the former is not as clearly grid-aligned.

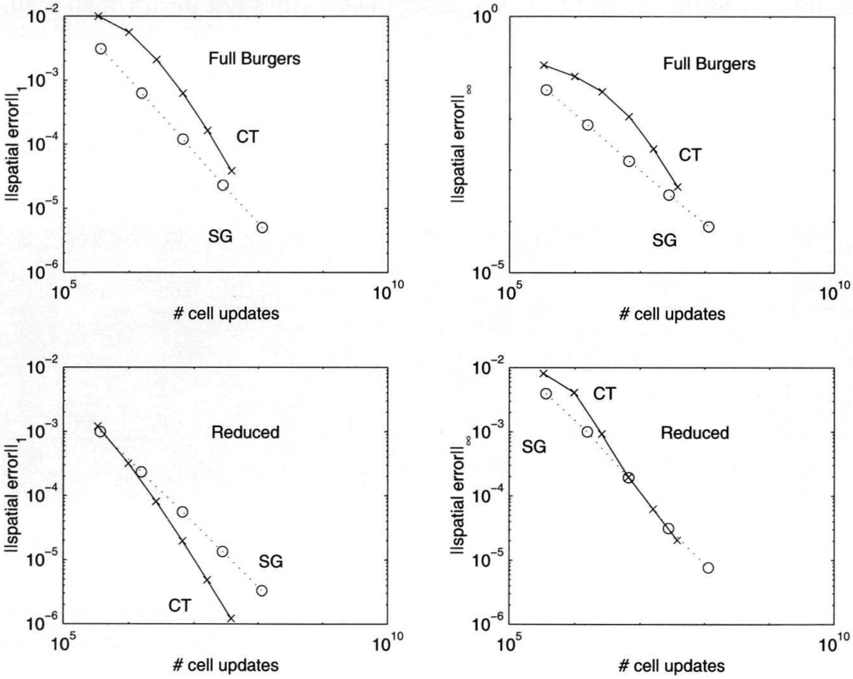


Figure 4.8: Efficiency comparisons for model problem 2

To see how the CT performs on the Burgers' test case when this is made more grid-aligned, we now take as initial condition $v = 0$ which guarantees that v remains zero. Furthermore we replace the parabolic inlet condition by

$$u(-1,y,t) = \begin{cases} \cos^2(y-\frac{1}{2}),\ y \geq 0, \\ \cos^2(y+\frac{1}{2})^2,\ y < 0. \end{cases}$$

This removes a strong peak in the error at $(x,y) = (-1,0)$ which would otherwise dominate the error. The results for this reduced Burgers' test case are shown in the lower row of Fig. 8. Measured in the L_1 norm the CT outperforms a SG when applied to this reduced test case. Measured in the L_∞ norm this is still not the case, but at least the CT is comparable.

4.7 Conclusions

When applied to the simple grid-aligned, linear constant-coefficient test case the CT is clearly superior to the SG approach in terms of efficiency. Especially when the diffusion parameter ε is small, the linear test case is strongly grid-aligned and very well suited to the CT.

When applied to the 2D Burgers' test case, the CT does not perform so well. The CT does perform reasonably well for a reduced version of the Burgers' test case with advection in only one direction.

Based on these observations, our expectation that the CT is well suited to advection-diffusion problems that are strongly grid-aligned has been confirmed. But it seems that the CT is less suited to more general problems.

BIBLIOGRAPHY

[1] H. J. Bungartz, M. Griebel, D. Roschke and C. Zenger, *Pointwise convergence of the combination technique for the Laplace equation,* East-West J. Numer. Math., Vol. 2, No. 1, pp. 21-45, 1994. Pages: 80

[2] K. Dekker and J. G. Verwer, *Stability of Runga-Kutta Methods for Stiff Nonlinear Differential Equations,* Elsevier North-Holland, Amsterdam, 1984. Pages: 86

[3] M. Griebel, *The combination technique for the sparse grid solution of pde's on multiprocessor machines,* Parallel Processing Letters Vol. 2 No. 1 61–70, 1992. Pages: 80

[4] M. Griebel, M. Schneider and C. Zenger, A combination technique for the solution of sparse grid problems, in: R. Beauwens and P. de Groen, eds., *Iterative Methods in Linear Algebra,* 263–281 (North-Holland, Amsterdam, 1992). Pages: 80, 84

[5] M. Griebel and G. Zumbusch, Adaptive sparse grids for hyperbolic conservation laws, in: W. Hackbusch and G. Wittum, eds., *Notes on Numerical Fluid Mechanics,* (Vieweg, Braunschweig, 1999). Pages: 80

[6] B. Koren, 'A robust upwind discretization method for advection, diffusion and source terms', in: *Numerical Methods for Advection-Diffusion Problems* (C. B. Vreugdenhil and B. Koren, eds.), *Notes on Numerical Fluid Mechanics,* **45,** 117–138, Vieweg, Braunschweig (1993). Pages: 85

[7] D. Lanser, J. G. Blom, J. G. Verwer, *Time integration of the shallow water equations in spherical geometry,* J. Comput. Phys., **171,** pp. 373–393 (2001). Pages: 80, 86, 87

[8] B. Lastdrager and B. Koren, *Error analysis for function representation by the sparse-grid combination technique,* Report MAS-R9823, CWI, Amsterdam 1998. Pages: 85
http://www.cwi.nl/static/publications/reports/MAS-1999.html

[9] B. Lastdrager, B. Koren and J. G. Verwer, The sparse-grid combination technique applied to time-dependent advection problems, in: E. Dick, K. Riemslagh and J. Vierendeels, eds., Proceedings of the *Sixth European Multigrid Conference,* Gent, 1999, *Lecture Notes in Computational Science and Engineering,* **14,** 143–149, (Springer, Berlin, 2000). Pages: 80

[10] B. Lastdrager, B. Koren and J. G. Verwer, *The sparse-grid combination technique applied to time-dependent advection problems*, Report MAS-R9930, CWI, Amsterdam, 1999. Pages: 85
http://www.cwi.nl/static/publications/reports/MAS-1999.html

[11] U. Rüde, Multilevel, extrapolation and sparse grid methods, in: P. W. Hemker and P. Wesseling, eds., *Multigrid Methods*, **IV**, 281-294 (Birkhäuser, Basel, 1993). Pages: 80

[12] L. F. Shampine, *Numerical Solution of Ordinary Differential Equations*, Chapman & Hall, New York, 1994. Pages: 88

[13] J. G. Verwer, E. J. Spee, J. G. Blom and W. Hundsdorfer, *A second-order Rosenbrock method applied to photochemical dispersion problems*, SIAM J. Sci. Comput. Vol. 20, No. 4, pp. 1456-1480, 1999. Pages: 86

[14] C. Zenger, Sparse grids, in: W. Hackbusch, ed., *Notes on Numerical Fluid Mechanics*, **31**, 241-251 (Vieweg, Braunschweig, 1991). Pages: 80

5

NUMERICAL SOLUTION OF MIXED GRADIENT-DIFFUSION EQUATIONS MODELLING AXON GROWTH

Abstract. In the current paper a numerical approach is presented for solving a system of coupled gradient-diffusion equations which acts as a first model for the growth of axons in brain tissue. The presented approach can be applied to a much wider range of problems, but we focus on the axon growth problem. In our approach time stepping is performed with a Rosenbrock solver with approximate matrix factorization. For the Jacobian an approximation is used that simplifies the solution of the coupled parabolic and gradient equations. A possible complication in the implementation of source terms is noted and a criterion that helps to avoid it is presented.

5.1 Introduction

In biological experiments it is often observed that in the initial growth phase axons approach each other to form a bundle. Then, in the intermediate phase the axons grow jointly towards a remotely located concentration of so-called targets. Once the axons have sufficiently approached the targets they debundle and attach to different individual targets. The ultimate goal of our research is to develop a numerical modelling tool that can establish which physical processes are necessary and which are not to predict above described behavior of bundling and debundling.

It is known that one of the mechanisms by which axons are guided to their targets relies on the diffusion of chemo-attractant molecules from the target through the tissue. The concentration of target derived chemo-attractant is largest near the targets and decays away from the targets. The growth cones at the tip of the axons can sense and follow the gradient in concentration to reach the targets [3]. Furthermore, it is also known that axons are repelled by diffusible molecules secreted by tissues the axons need to grow away from.

Several mechanisms have been suggested to explain that axons approach each other in the initial growth phase. Random movement might bring the axons together, repulsive signals from surrounding cells could do the same or diffusible molecules that the axons secrete may guide them towards each other. Following [5], we focus on the latter mechanism; we assume that the axons growth cones emit a diffusible chemo-attractant to which the other axons growth cones are sensitive.

Furthermore we assume that the growth cones can secrete diffusible substances that act as chemorepellant to the other axons. In particular we allow the rate of secretion of chemorepellant to be dependent on the concentration of target derived chemo-attractant. This enables the axons to repel one another progressively stronger as they approach the target area, ultimately leading to debundling.

In the current paper we restrict the interaction between axons to growth-cone to growth-cone interaction. I.e., axons only secrete chemicals from their growth cones and can only sense with their growth cones.

5.2 Biological model

In this paper we consider essentially the same model as in [5]. We consider a fixed spatial domain $\Omega = [-0.5, 0.5] \times [-0.5, 0.5]$, measured in millimeters. The model assumes that both the targets and the growth cones secrete diffusible chemical compounds to which the growth cones are sensitive. In particular, the targets release an attractant and the growth cones release both an attractant and a repellant. The rate of release of repellant is dependent on the concentration of target derived attractant. This allows the axons to repel each other once they have reached the target area, where there is a high concentration of target derived attractant.

The time evolution of the concentration fields is governed by diffusion equations of the form

$$\frac{\partial u}{\partial t} = d\Delta u - \kappa u + s,$$

where u is the concentration of a diffusible chemical compound. The diffusion constant d measures how quickly the compound diffuses through the medium, the loss constant κ measures the rate of absorption of the medium and the source term s contains the release of mass by the axons and targets. The growth of the axons is governed by gradient equations of the form

$$\frac{d\mathbf{r}}{dt} = \lambda \nabla u(\mathbf{r}(t), t), \tag{5.1}$$

where $\mathbf{r}$ is the position of an axon. The parameter λ measures the sensitivity of the axon to gradients in the concentration field u.

A property of (5.1) is that in a steady solution field u, limit points for $\mathbf{r}$ coincide with a maximum, minimum or saddle point in u. Depending on the sign of λ, maxima or minima can either be stable or unstable limit points. For $\lambda > 0$, $\mathbf{r}$ will move in the direction of the gradient of u, i.e. it will move towards a maximum of u. Once at a maximum, a small displacement in $\mathbf{r}$ will cause $\mathbf{r}$ to move back towards the maximum. Hence, for $\lambda > 0$ maxima in u are stable limit points for $\mathbf{r}$. Likewise, for $\lambda < 0$ minima in u are stable limit points for $\mathbf{r}$. For $\lambda < 0$ maxima are unstable limit points for $\mathbf{r}$ since a small displacement from a maximum will cause $\mathbf{r}$ to move away from the maximum. Likewise, for $\lambda > 0$ minima in u are unstable limit points for $\mathbf{r}$. Saddle points are always unstable since a small displacement from the saddle point can place $\mathbf{r}$ at a new position where the gradient in u can either point towards or away from the saddle point.

For the current model of cone derived attractant and repellant and target derived repellant the growth of the axons is governed by the following set of coupled gradient-diffusion equations:

$$\begin{aligned}
\frac{\partial u(\mathbf{x},t)}{\partial t} &= d_u \Delta u(\mathbf{x},t) - \kappa_u u(\mathbf{x},t) + \sigma_u \sum_\beta s(\mathbf{x} - \mathbf{T}_\beta), && (5.2)\\
\frac{\partial v(\mathbf{x},t)}{\partial t} &= d_v \Delta v(\mathbf{x},t) - \kappa_v v(\mathbf{x},t) + \sigma_v \sum_\alpha s(\mathbf{x} - \mathbf{r}_\alpha(t)), && (5.3)\\
\frac{\partial w(\mathbf{x},t)}{\partial t} &= d_w \Delta w(\mathbf{x},t) - \kappa_w w(\mathbf{x},t) + \sum_\alpha \sigma_w \left(u(\mathbf{x},t)\right) s(\mathbf{x} - \mathbf{r}_\alpha(t)), && (5.4)\\
\frac{d\mathbf{r}_\alpha(t)}{dt} &= \lambda_u \nabla u(\mathbf{r}_\alpha(t),t) + \lambda_v \nabla v(\mathbf{r}_\alpha(t),t) - \lambda_w \nabla w(\mathbf{r}_\alpha(t),t), && (5.5)
\end{aligned}$$

together with homogenous Dirichlet boundary conditions on u, v, and w.

Here t denotes time, $\mathbf{x} = (x, y)$ denotes a position in two dimensional space, u, v and w denote the concentrations of respectively, target derived attractant, cone derived attractant and cone derived repellant. Furthermore, d_u, d_v and d_w are the corresponding diffusion coefficients and κ_u, κ_v and κ_w are loss coefficients due to absorption in the tissue. The positions of the targets and growth cones are denoted by $\mathbf{T}_\beta$ and $\mathbf{r}_\alpha$, respectively, where β ranges from 1 to the number of targets and α ranges from 1 to the number of cones. The coefficients σ_u, σ_v and σ_w denote the rate of release of the different chemical compounds. The term $s(\mathbf{x})$ denotes a localised symmetric source term function with maximum at $\mathbf{x} = \mathbf{0}$. Finally, the positive coefficients λ_u, λ_v and λ_w measure the sensitivity of the cones to the gradients in the corresponding chemical concentration fields.

Note that in the gradient equation λ_u and λ_v both enter with a positive sign while λ_w enters with a negative sign. This represents the fact that both u and v act as attractants to the cones while w acts as a repellant. A positive λ causes $\mathbf{r}_\alpha$ to grow towards a maximum in the corresponding concentration field while a negative λ causes $\mathbf{r}_\alpha$ to grow away from a maximum.

Example
In Figure 1 the stationary solution for u is shown for a configuration with 5 targets located at $y = 0.25$ and $x = -0.25, -0.125, 0, 0.125, 0.25$. In Figures 2a-f the cones growth trajectories are displayed corresponding to the stationary target derived attractant field shown in Figure 1 with initial cone positions given by $y = -0.25$ and $x = -0.25, -0.125, 0, 0.125, 0.25$. The trajectories are shown at different time levels between $t = 0$ and $t = 2000$, measured in seconds, together with contour lines for the net gradient field $\lambda_u u + \lambda_v v - \lambda_w w$.

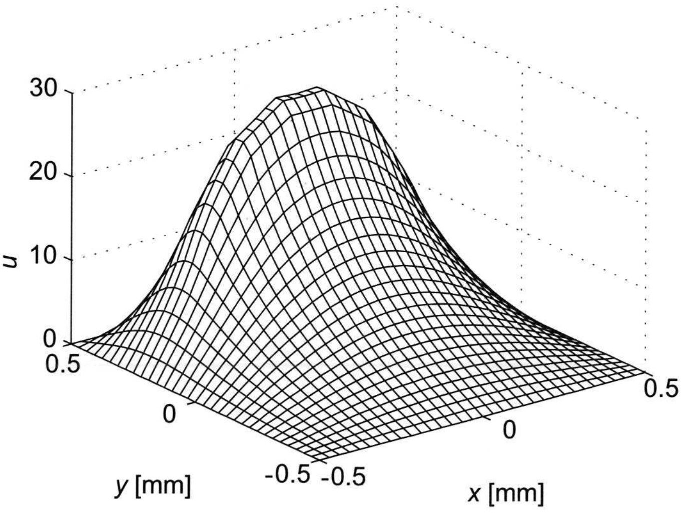
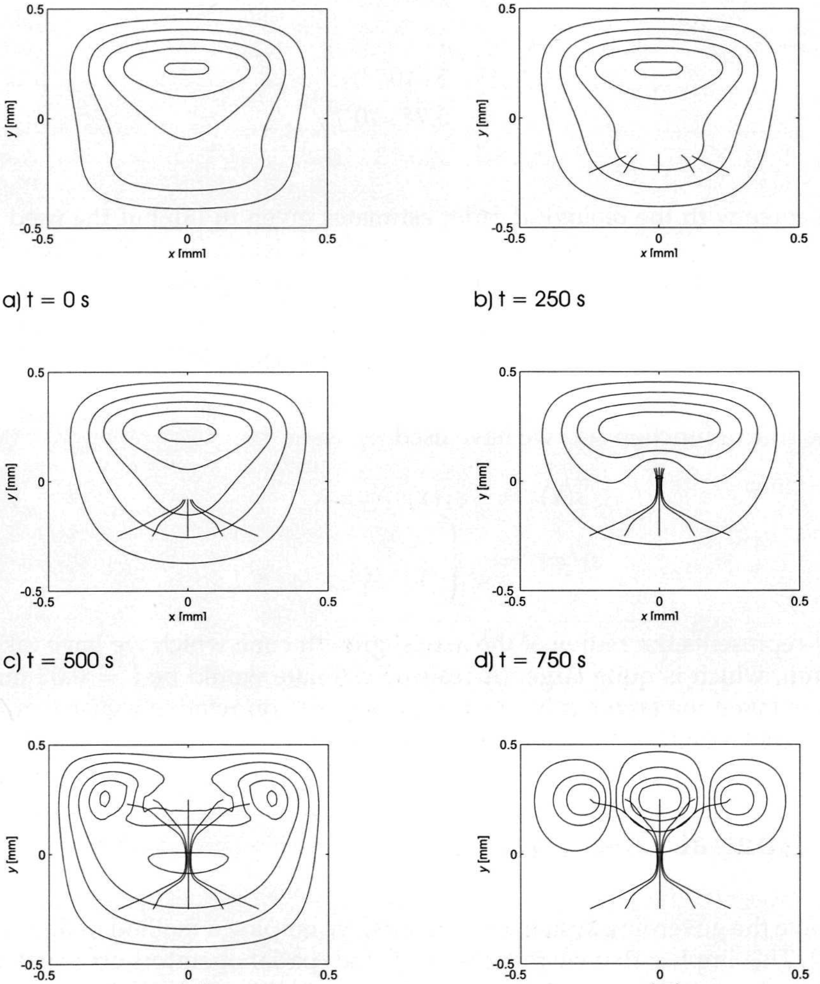


Figure 5.1: Stationary solution for u

Since the axons grow in the direction of the gradient in $\lambda_u u + \lambda_v v - \lambda_w w$, the direction of growth of a cone is always perpendicular to the contour lines. This does not imply that the growth paths should be perpendicular to the contour lines. The paths represent the growth of the axons at earlier times and need not be perpendicular to the contour lines at the current time.



a) t = 0 s

b) t = 250 s

c) t = 500 s

d) t = 750 s

e) t = 1000 s

f) t = 2000 s

Figure 5.2: Axon growth trajectories together with contour lines for the net gradient field $\lambda_u u + \lambda_v v - \lambda_w w$

Figures 1 and 2 are based on the following parameters:

$$\begin{aligned}
d_u &= d_v = d_w = 10^{-4}, \\
\kappa_u &= \kappa_v = \kappa_w = 10^{-4}, \\
\lambda_u &= 10^{-5}, \\
\lambda_v &= 5 \cdot 10^{-6}, \\
\lambda_w &= 3.75 \cdot 10^{-5}, \\
\sigma_u &= \sigma_v = 3 \cdot 10^{-3}.
\end{aligned}$$

These agree with the biological order estimates given in [8], but the used values were tuned to get a clear pattern of bundling and debundling. Following [5], the dependence of σ_w on $u(\mathbf{x}, t)$ is modelled by the following relationship,

$$\sigma_w = \frac{3 \cdot 10^{-3} u^2}{u^2 + \rho^2}, \quad \rho = 25.$$

For the source function $s(\mathbf{x})$ we have used

$$\begin{aligned}
s(\mathbf{x}) &= s_1(x) s_1(y), \\
s_1(\varphi) &= \begin{cases} 0, & |\varphi| > l. \\ 1 - \frac{|\varphi|}{l}, & |\varphi| \leq l. \end{cases}
\end{aligned}$$

Here l represents the radius of the axons' growth cone which we have taken $l = 0.15$ mm, which is quite large. A realistic estimate would be $l = 0.02$ mm, but we have taken the larger value to get convergence on relatively coarse grids; the model is not significantly altered by this choice.

5.3 Spatial Discretization

To evolve the governing system (5.3) in time, we employ a method of lines scheme (MOL). This implies that we first discretize the spatial operators on a spatial grid. This transforms the set of PDEs for the field quantities u, v, w into a large set of ODEs in time, i.e., we get an ODE for every spatial point. In the current paper the resulting set of ODEs together with the ODEs for the growth cone positions $\mathbf{r}_\alpha$ are solved with a Rosenbrock time stepping technique, as is explained in the next section.

The set of equations (5.3) presents a numerical challenge due to the presence of the equations for the growth cone positions $\mathbf{r}_\alpha$. Without these, the set could be solved in a straightforward manner with existing numerical techniques for PDEs. In its present form the model requires the solution of field equations coupled to particle equations. In principle it might be feasible to reformulate the field equations into a particle form or to rewrite the particle equations into field equations to obtain either a pure particle or a pure field problem, we however choose to apply a mixed field/particle setup. This approach has as an advantage in that both the field equations and the particle equations are tackled in a natural, efficient manner.

Consider a uniform spatial grid Ω_h, with mesh width $h = 1/(N+1)$, consisting of grid cells

$$\Omega_{i,j} = \{(x,y)|x_{i-1} \leq x \leq x_i, y_{j-1} \leq y \leq y_j\},$$

where $x_i = ih - 0.5$, $y_j = jh - 0.5$ and $i = 1, 2, \cdots, N$ and $j = 1, 2, \cdots, N$. We approximate the spatially dependent function $u(\mathbf{x}, t)$ with a grid function $u_h(t)$ such that at node (x_i, y_j) the grid function u_h has the value $u_{i,j}(t) \approx u(x_i, y_j, t)$. The spatial differential operators are approximated with second order central finite difference operators denoted by Δ_h and ∇_h. After discretization the ODEs for the axon positions read

$$\frac{d\mathbf{r}_\alpha(t)}{dt} = \lambda_u P_h(\mathbf{r}_\alpha(t))\nabla_h u_h(t) + \lambda_v P_h(\mathbf{r}_\alpha(t))\nabla_h v_h(t) - \lambda_w P_h(\mathbf{r}_\alpha(t))\nabla_h w_h(t),$$

where $\mathbf{r}_\alpha(t)$ now represents the solution to the spatially discretized equation. To avoid excessive indices we do not replace $\mathbf{r}_\alpha(t)$ by $\mathbf{r}_{\alpha,h}(t)$. Note the appearance of $P_h(\mathbf{r}_\alpha(t))$. This is an interpolation operator which satisfies

$$P_h(\mathbf{r}_\alpha(t))\nabla_h u_h(t) \approx \nabla u(\mathbf{r}_\alpha(t), t),$$

provided $\nabla_h u_h(t)|_{i,j} \approx \nabla u(x_i, y_j, t)$. The interpolation that we use is simple bilinear interpolation, combined with second order central differences for ∇_h, hence we have

$$P_h(\mathbf{r}_\alpha(t))\nabla_h u_h(t) = \nabla u(\mathbf{r}_\alpha(t), t) + O(h^2).$$

Bilinear interpolation of the differences is sufficient to guarantee that $d\mathbf{r}_\alpha(t)/dt$ is continuous across cell interfaces and hence yields smooth trajectories for $\mathbf{r}_\alpha$. Note that with bilinear interpolation $d^2\mathbf{r}_\alpha(t)/dt^2$ does not exist across cell interfaces. This implies that bilinear interpolation is not to be used in conjunction with a higher order time integration method.

5.4 The Rosenbrock method

The full system of ODEs that results from spatial discretization, here denoted by $dc/dt = F(c)$, is stiff and therefore integrated with a second order Rosenbrock solver using an approximate Jacobian matrix. In [9] this solver has been successfully applied to an advection-diffusion-reaction system from air pollution modelling. The system is autonomous since the right hand side $F(c)$ contains no explicit time dependence; all time dependence in $F(c)$ enters through components of the solution being time dependent. The solution is advanced over a time step with

$$c^{n+1} = c^n + \frac{3}{2}\tau\kappa_1 + \frac{1}{2}\tau\kappa_2,$$

where τ is the step size $t_{n+1} - t_n$, c^n the approximation for $c(t_n)$, and

$$\begin{aligned} (I - \gamma\tau J)\,\kappa_1 &= F(c^n), \\ (I - \gamma\tau J)\,\kappa_2 &= F(c^n + \tau\kappa_1) - 2\kappa_1. \end{aligned}$$

Here J is an approximation of the Jacobian $\partial F/\partial c$ at $c = c^n$ and γ is a free parameter. With the exact Jacobian for J, the stability function reads

$$R(z) = \frac{1 + (1 - 2\gamma)z + (\frac{1}{2} - 2\gamma + \gamma^2)z^2}{(1 - \gamma z)^2}, \tag{5.6}$$

from which it follows that the method is A-stable if and only if $\gamma \geq 1/4$ [4]. Furthermore the method is L-stable if $\gamma = 1 \pm \frac{1}{2}\sqrt{2}$ and the exact Jacobian matrix is used for J. The scheme is of second order in τ regardless of the choice for J.

In [4] one finds ample evidence that Rosenbrock methods are well suited to solve stiff ODEs in the low to moderate accuracy range. However, with the exact Jacobian $\partial F/\partial c$ the method cannot be efficiently applied since the linear system solutions are much too expensive. We therefore apply it with an approximation for $\partial F/\partial c$.

In the current work $c = (\rho_h, r_h)$, where $\rho_h = (u_h, v_h, w_h)$ and r_h is the vector of the r_α. To simplify the notation, in the following we suppress the index h. Furthermore, in an obvious notation, we write $F(c) = (F_\rho(c), F_r(c))$, then

$$\frac{\partial F}{\partial c} = \begin{pmatrix} \frac{\partial F_\rho}{\partial \rho} & \frac{\partial F_\rho}{\partial r} \\ \frac{\partial F_r}{\partial \rho} & \frac{\partial F_r}{\partial r} \end{pmatrix}.$$

We now exploit the fact that the Rosenbrock solver remains of second order for any choice of approximate matrix J and put

$$J = \begin{pmatrix} \frac{\partial F_\rho(c_n)}{\partial \rho} & 0 \\ 0 & 0 \end{pmatrix}.$$

That means we treat the $\partial F_\rho/\partial \rho$ part of the system linearly implicitly and the remainder explicitly. In other words, without the gradient equation we apply the A-stable Rosenbrock solver to the semi-discrete field equations with an exact Jacobian matrix, and without the field equations we integrate the gradient equations explicitly. The explicit method is obtained by substituting for J the zero matrix in the Rosenbrock method. This gives the explicit trapezoidal rule

$$c^{n+1} = c^n + \frac{1}{2}\tau F(c^n) + \frac{1}{2}\tau F(c^n + \tau F(c^n)). \tag{5.7}$$

We have also experimented with another implementation that does treat the whole system implicitly. In principle this allowed us to use the third order Rosenbrock method from [7], but we did not pursue this further because complications arose in the spatial discretization. The third order Rosenbrock method requires a smoother spatial discretization of higher order. So far we did not succeed in implementing such a discretization that worked better than the existing one.

5.4.1 Stability

While the order of the Rosenbrock method remains 2 with an approximate Jacobian, the stability properties can drastically change. Due to our choice for J, intuition says that the step size restriction will come from the explicit trapezoidal rule applied to the gradient equations and that the field equations do not give a restriction since these are solved linearly implicitly.

To get some insight into the stability of our method we borrow a model problem from [8] which the authors propose as a test model to investigate stability. The test model is a 2 × 2 linear ODE of the form

$$\frac{dc}{dt} = \begin{pmatrix} d_0 & 1 \\ 0 & d \end{pmatrix} c, \tag{5.8}$$

which in a number of steps with simplifying assumptions is obtained from (5.3) through 'linearization, freezing of coefficients, and Fourier-Von Neumann analysis'. Specifically, we have $-\infty < d_0 < 0$ with d_0 representing eigenvalues of the discrete Laplacian Δ_h. Hence d_0 depends on h^{-2} and can become large negative. Further, d represents an eigenvalue for the gradient equation and is determined by second order spatial derivatives of u, v, w. To see this, consider the 1D gradient equation

$$\begin{aligned} \frac{dr(t)}{dt} &= f(r(t)), \\ f(r(t)) &= \lambda u_x(r(t), t), \end{aligned}$$

then

$$\frac{\partial f(r(t))}{\partial r} = \lambda u_{xx}(r(t), t).$$

So, for d we can think of finite values. However, d can also take on positive values.

With the stability test model we have

$$J = \begin{pmatrix} d_0 & 0 \\ 0 & 0 \end{pmatrix}.$$

The Rosenbrock method then gives the recursion

$$c^{n+1} = \begin{pmatrix} R_{11} & R_{12} \\ 0 & R_{22} \end{pmatrix} c^n, \tag{5.9}$$

where

$$\begin{aligned} R_{11} &= 1 + \tau\left(2(1-\gamma\tau d_0)^{-1} - (1-\gamma\tau d_0)^{-2}\right) d_0 + \frac{1}{2}\tau^2(1-\gamma\tau d_0)^{-2} d_0^2, \\ R_{12} &= \tau\left(2(1-\gamma\tau d_0)^{-1} - (1-\gamma\tau d_0)^{-2}\right) \\ &\quad + \frac{1}{2}\tau^2\left((1-\gamma\tau d_0)^{-1} d + (1-\gamma\tau d_0)^{-2} d_0\right), \\ R_{22} &= 1 + \tau d + \frac{1}{2}\tau^2 d^2. \end{aligned}$$

Note that $R_{11} = R(\tau d_0)$ with R the stability function (5.6). Likewise $R_{22}(z) = 1 + z + z^2/2$ is the stability function of (5.7). Iterating the recursion (5.9) gives

$$\begin{aligned} c^n &= \begin{pmatrix} R_{11}^n & Q \\ 0 & R_{22}^n \end{pmatrix} c^0, \\ Q &= R_{12} \sum_{i=0}^{n-1} R_{22}^i R_{11}^{n-i-1}. \end{aligned}$$

For a stable scheme we must have power boundedness, i.e., $\|c^n\| \leq C \left\|c^0\right\|$, with C independent of n. We can guarantee power boundedness if we have

$$|R_{11}|^n \leq C_{11}, \ |R_{22}|^n \leq C_{22}, \ |Q| \leq C_{12}, \text{ with } C_{11}, C_{22}, C_{12} \text{ independent of } n.$$

In our implementation we limit ourselves to $\gamma \geq 1/4$, hence we have $|R_{11}| \leq 1$. Likewise, we consider only $\tau \leq 2/|d|$ and $d < 0$ ensuring $|R_{22}| \leq 1$. Now let $R_* = \max\left(|R_{11}|, |R_{22}|\right)$, then

$$\begin{aligned} |Q| &= |R_{12}| \left| \sum_{i=0}^{n-1} R_{22}^i R_{11}^{n-i-1} \right| \\ &\leq |R_{12}| \sum_{i=0}^{n-1} |R_{22}|^i |R_{11}|^{n-i-1} \\ &\leq |R_{12}| \sum_{i=0}^{n-1} R_*^{n-1} \\ &= |R_{12}| (n-1) R_*^{n-1} \\ &\leq |R_{12}| e^{-1} \frac{1}{\ln\left(R_*^{-1}\right)}. \end{aligned}$$

Hence $|Q|$ is bounded independent of n if $|R_{12}|$ and $1/\ln\left(R_*^{-1}\right)$ are bounded. For $1/\ln\left(R_*^{-1}\right)$ to be bounded it is necessary to have $|R_{11}|$ and $|R_{22}|$ strictly smaller than one, instead of $|R_{11}| \leq 1$ and $|R_{22}| \leq 1$. The entry R_{12} is even $O(\tau)$ and thus bounded. Hence we see that for a stable method we must have some damping for the test model (5.8), provided by the Rosenbrock method and the explicit trapezoidal rule, in addition to the step size restriction $\tau \leq 2/|d|$. If d is positive, some error growth is natural. Note that these conclusions are valid uniformly for $-\infty < d_0 < 0$.

5.4.2 Spatial factorization

Since we neglect the terms $\partial F_\rho/\partial r$, $\partial F_r/\partial \rho$ and $\partial F_r/\partial r$ in the true Jacobian, the systems that need to be solved are

$$\begin{aligned}
\left(I-\gamma\tau\frac{\partial F_\rho}{\partial \rho}\right)\kappa_{1\rho} &= F_\rho(c^n), && (5.10)\\
\kappa_{1r} &= F_r(c^n), && (5.11)\\
\left(I-\gamma\tau\frac{\partial F_\rho}{\partial \rho}\right)\kappa_{2\rho} &= F_\rho(c^n+\tau\kappa_1)-2\kappa_{1\rho}, && (5.12)\\
\kappa_{2r} &= F_r(c^n+\tau\kappa_1)-2\kappa_{1r}, && (5.13)
\end{aligned}$$

with the Jacobian $\partial F_\rho/\partial \rho$ evaluated at $c = c^n$. To further speed up the linear system solution, we approximate the matrix with the following spatial factorization,

$$\left(I-\gamma\tau\frac{\partial F_{\rho x}}{\partial \rho}\right)\left(I-\gamma\tau\frac{\partial F_{\rho y}}{\partial \rho}\right), \qquad (5.14)$$

where $F_{\rho x}$ and $F_{\rho y}$ contain only difference operators in either the x- or the y-direction and $F_{\rho x} + F_{\rho y} = F_\rho$. Further, in our implementation we have distributed the loss terms and source terms equally (with factor 0.5) over $F_{\rho x}$ and $F_{\rho y}$. Solving the systems with factorized matrices requires the successive solution of systems of a much smaller dimension, that are only coupled in the x- or y-direction. For each grid line in Ω_h we encounter such a smaller sized system. These systems can be solved far more quickly than the original ones. This is an application of approximate matrix factorization. Examples of approximate matrix factorization can be found in [2], [1], [9], [6] and [7].

The use of (5.14) does not cause a loss of order in accuracy of the numerical solution since the order remains two for any approximation to $\partial F_\rho/\partial \rho$. Note that (5.14) implies that we approximate $\partial F_\rho/\partial \rho$ by

$$\partial F_{\rho x}/\partial \rho + \partial F_{\rho y}/\partial \rho + \gamma\tau \partial F_{\rho x}/\partial \rho \; \partial F_{\rho y}/\partial \rho.$$

The factorization does somewhat weaken the stability properties, e.g., L-stability is lost but otherwise the stability behaviour remains quite satisfactory.

5.5 Implementation of source terms and gradient

In our axon growth model it is assumed that all growth cones secrete identical chemical compounds through which they attract and repel each other. It therefore is possible that a certain growth cone responds to molecules secreted by that same growth cone. For instance, a growth cone can be slowed down in its growth due to the trail of attractant that builds up behind the growth cone. This self-interaction can be very troublesome in a numerical implementation. In the model the growth cones are described as small particles that secrete chemicals in a small spatial region and sense the gradients in chemical concentrations in the same region. To avoid self-interaction we consider only symmetrical source terms $s(x)$ which do not yield a direct gradient at the location of the emitting axon. Later on we will present a condition that the numerical implementation must satisfy to mimic this property.

To better understand how we can have self-interaction, and how this relates to the symmetry of $s(x)$, consider the following simplified version of the model which contains only one axon at position $r(t)$ and no targets:

$$\frac{\partial u(x,t)}{\partial t} = \frac{\partial^2 u(x,t)}{\partial x^2} - ku(x,t) + s(x - r(t)), \tag{5.15}$$

$$\frac{dr(t)}{dt} = \frac{\partial u(r(t),t)}{\partial x}. \tag{5.16}$$

By differentiating (5.16) with respect to t and substituting (5.15) we get

$$\begin{aligned}\frac{d^2r(t)}{dt^2} &= \frac{\partial^3 u(r(t),t)}{\partial x^3} - k\frac{\partial u(r(t),t)}{\partial x} \\ &\quad + \frac{\partial^2 u(r(t),t)}{\partial x^2}\frac{\partial u(r(t),t)}{\partial x} + \left.\frac{\partial s(x-r(t))}{\partial x}\right|_{x=r(t)}.\end{aligned}$$

Hence when $\partial s(0)/\partial x \neq 0$, then the evolution of $r(t)$ will be influenced by the presence of $s(x)$. This is a self-interaction since $s(x - r(t))$ represents an emission by the axon located at $r(t)$. Higher temporal derivatives of $r(t)$ also only contain odd spatial derivatives of $s(x)$, i.e., $d^n r(t)/dt^n$ contains

$$\frac{\partial^{2n-1}s(0)}{\partial^{2n-1}x}, \frac{\partial^{2n-3}s(0)}{\partial^{2n-3}x}, \ldots, \frac{\partial s(0)}{\partial x}.$$

To avoid self-interaction completely we must have that all odd derivatives of $s(x)$ vanish at $x = 0$, which holds if $s(x)$ is symmetric.

To see what kind of numerical complications we can have due to self-interaction we now consider some numerical implementations of (5.15) and (5.16). After spatial discretization at grid points x_i we get

$$\begin{aligned}\frac{du_i(t)}{dt} &= D_{xx}u_i(t) - ku_i(t) + s(x_i - r(t)), \\ \frac{dr(t)}{dt} &= P_h(r(t))D_x u_h(t).\end{aligned}$$

Note that we use D_x and D_{xx} to denote spatial discretizations of $\partial/\partial x$ and $\partial^2/\partial x^2$, instead of ∇_h and Δ_h, to stress that we now model a one-dimensional case. Further, $P_h(r(t))$ interpolates the grid function $D_x u_h(t)$ at location $r(t)$. Note that as in Section 3 we keep the same notation for r after spatial discretization. When we now consider $d^2r(t)/dt^2$ we see the source term contribution appearing as

$$P_h(r(t))D_x s(x_i - r(t)). \tag{5.17}$$

In the remainder of this section we will examine this term for different spatial discretizations. It will turn out that depending on the discretization chosen, the contribution of this interpolated source term either vanishes or not. In our implementation a non-vanishing contribution is noticed as a growth cone that is blinded by its own attractant and repellant and randomly wanders about the domain. It will turn out that the choice for $P_h(r)D_x$ and $s(x - r(t))$ must be matched in some sense to avoid an excessive contribution of the interpolated source term.

Depending on the size of the mesh width, we treat the source terms in one of two ways. If the grid is coarse relative to the size of the sources, we treat the sources as point sources. When the grid is sufficiently fine we take into account the finite spatial extent of the sources.

5.5.1 Highly localized source term

First we consider the case of a relatively coarse grid. When the spatial extent of the source term is smaller than twice the mesh width, we distribute the source term over the nearest grid points. Note that in this section we attempt to model a point source and must therefore choose an appropriate distribution over the nearest grid points. In the next section we consider a source term with a spatial extent and the source term is in principle fixed by the biological model.

It is insightful to examine a few source term implementations together with choices for the gradient operator on a grid of points $x_i = ih$.

Example

As a first implementation for the discrete gradient operator consider for a grid funtion f_h

$$P_h(r)D_x f_h = \frac{f_{i+1} - f_i}{h}, \quad x_i \leq r < x_{i+1}. \tag{5.18}$$

As implementation of the source term consider

$$s(x_i - r) = s(x_{i+1} - r) = \frac{1}{2h}, \quad x_i \leq r < x_{i+1}, \quad s_i(x_j - r) = 0, \quad j \neq i, i+1,$$

i.e., the source flux is divided evenly over the two nearest grid points. Taking these implementations together yields for the term (5.17)

$$P_h(r)D_x s(x_i - r) = \frac{s(x_{i+1} - r) - s(x_i - r)}{h} = \frac{1/2 - 1/2}{h^2} = 0, \quad x_i \leq r < x_{i+1},$$

hence we have a vanishing contribution, which is satisfactory. Now consider the same implementation for the gradient but replace the source term interpolation with

$$s_i(x_i - r) = 1/h, \quad s(x_j - r) = 0, \quad r - h/2 \leq x_i < r + h/2, \quad j \neq i,$$

i.e., all source flux is attributed to the nearest grid point. The term (5.17) then becomes

$$P_h(r)D_x s(x_i - r) = \frac{s(x_{i+1} - r) - s(x_i - r)}{h} = -\frac{1}{h^2}, \quad x_i \leq r \leq x_i + h/2.$$

Hence, we end up with a non-vanishing contribution that is inversely proportional to h^2. In the case that we would attempt to model a delta function type source term, this contribution would lead to a scheme that does not converge when we consider the limit $h \to 0$. In the case of a source term of finite spatial extent, the problem is less severe since, eventually, decreasing the mesh width h will resolve the source term on the spatial grid. Resolved source terms are discussed in the next section.

Condition on implementation of source term and gradient

We have seen that for an acceptable implementation of source term and gradient we must demand that (5.17) vanishes. Now we introduce some notation to work out the consequences of this restriction. For the discretized approximation of the source term we write

$$s(x_i - r) = \frac{1}{h}\theta_i,$$

where θ_i are weight functions that smear out the point source behaviour over a limited number of points near r. To conserve mass the θ_i must satisfy

$$\sum_i \theta_i = 1. \tag{5.19}$$

For the discretized gradient operator, acting on a grid function f_h evaluated in point r, we write

$$P_h(r)D_x f_h = \sum_i \phi_i f_i,$$

where ϕ_i are weight functions that, under summation, construct an approximation to the gradient of f_h in point r out of surrounding values of f_h.

Using the above notation we have

$$P_h(r)D_x s(x_i - r) = \sum_j \frac{\phi_j \theta_j}{h}.$$

For a suitable combination of source term and gradient discretization

$$P_h(x)D_x s(x_i - r) \tag{5.20}$$

must vanish, hence we must have

$$\sum_i \phi_i \theta_i = 0. \tag{5.21}$$

Using this condition we can choose matching discretizations for source term and gradient. Again this is clarified by looking at some examples.

Example 1

As before, consider

$$P_h(r)D_x f_h = \frac{f_{i+1} - f_i}{h}, \quad x_i \leq r < x_{i+1},$$

as an implementation for the gradient. This corresponds to

$$\phi_{i+1} = \frac{1}{h}, \quad \phi_i = -\frac{1}{h}, \quad \phi_j = 0, \quad j \neq i, i+1.$$

Imposing restriction (5.21) then yields

$$\frac{1}{h}\theta_i - \frac{1}{h}\theta_{i+1} = 0,$$

while we also impose (5.19), which yields

$$\theta_i + \theta_{i+1} = 1.$$

These restrictions together yield

$$\theta_i = \theta_{i+1} = 1/2, \quad x_i \leq r < x_{i+1}, \quad \theta_j = 0, \quad j \neq i, i+1.$$

Note that this corresponds to the source term implementation that led to $P_h(r)D_x s(x_i - r) = 0$ in the previous example.

Example 2

Now consider

$$P_h(r)D_x f_h = \frac{x_{i+1} - r}{h}\frac{f_{i+1} - f_{i-1}}{2h} + \frac{r - x_i}{h}\frac{f_{i+2} - f_i}{2h}, \quad x_i \leq r < x_{i+1}, \tag{5.22}$$

which corresponds to

$$\begin{aligned}
\phi_{i-1} &= \frac{r - x_{i+1}}{2h^2}, && (5.23)\\
\phi_i &= \frac{x_i - r}{2h^2}, && (5.24)\\
\phi_{i+1} &= \frac{x_{i+1} - r}{2h^2}, && (5.25)\\
\phi_{i+2} &= \frac{r - x_i}{2h^2}, && (5.26)\\
\phi_j &= 0, \quad j \neq i-1, i, i+1, i+2. && (5.27)
\end{aligned}$$

This implementation for the gradient corresponds to second order central differences in x_i and x_{i+1} combined with linear interpolation. In choosing an implementation for the source term we now have four degrees of freedom, i.e., θ_{i-1}, θ_i,

$\theta_{i+1}, \theta_{i+2}$. To fix these degrees of freedom we have only two restrictions, (5.19) and (5.21). To close the system we simply choose $\theta_{i-1} = \theta_{i+2} = 0$, which makes the discretized source term more local which is natural since the spatial extent of the source is smaller than twice the mesh width. We are left with

$$\frac{x_i - r}{2h^2}\theta_i + \frac{x_{i+1} - r}{2h^2}\theta_{i+1} = 0,$$

and

$$\theta_i + \theta_{i+1} = 1,$$

which leads to

$$\theta_i = 1 - \frac{r - x_i}{h}, \quad \theta_{i+1} = 1 + \frac{r - x_{i+1}}{h}, \tag{5.28}$$

i.e., the source flux is linearly distributed over the nearest grid points.

Implementations used for the current work

Implementations (5.22) and (5.28) for the source term and gradient prove very useful in numerical practice. Hence these implementations were used to obtain the numerical results presented further on. Since we consider a spatially two-dimensional problem, the implementations were extended to two dimensions with a straightforward tensor product approach.

Higher order discretizations

It seems straightforward to extend the above reasoning to obtain higher order discretizations for (highly localized) source terms and gradients. In particular, we have examined a source term implementation compatible with fourth order Hermite interpolation. However, this higher order implementation did not compete with (5.23) and (5.28) in numerical experiments.

5.5.2 Resolved source term

When the mesh width is smaller than the spatial extent of the source term we say that the source term is resolved. The source term function $s(x - r)$, which is still localized around $x = r$, then spans several mesh widths. Depending on the choice for the implementation of the gradient operator and interpolation, there exist leading order expressions (in the mesh width h) for the contribution of $P_h(r)D_x s(x_i - r)$. For the implementation (5.18) we obtain

$$P_h(r)D_x s(x_i - r) = \frac{1}{2}(1 - 2\alpha)hs^{(2)}(0) + R_1,$$

assuming smoothness, where $\alpha \in [0, 1]$ measures the position r relative to the nearest grid points and R_1 is a remainder term of $O(h^2)$ that vanishes for $h \to 0$. For what follows R_1 is not negligible, but we focus on the formal leading order term anyway since that suffices to make our point. Likewise, for the implementation (5.22) we obtain

$$P_h(r)D_x s(x_i - r) = \frac{1}{6}(1-2\alpha)(\alpha-1)\alpha h^3 s^{(4)}(0) + R_2.$$

Note the absence of a term $\sim h^2$. This is due to $s^{(3)}(0) = 0$ since $s(x)$ is supposed to be symmetric around $x = 0$. Since these expressions vanish as $h \to 0$ they seem quite satisfactory. In practice h is finite of course and the contribution depends on the size of $s^{(2)}(0)$ and $s^{(4)}(0)$, respectively.

Smooth Gaussian type source term

Like in [5] we now consider a source term of the type

$$s(x) = e^{-\gamma x^2},$$

which does not truly vanish away from $x = 0$ but can be made very small away from $x = 0$ by taking γ sufficiently large. In a practical implementation we typically choose the mesh width h such that the source term is resolved on several grid cells, i.e., we can put

$$s(h) = e^{-\beta},$$

where β is of order 1. For such a mesh width we thus have

$$\gamma = \beta / h^2.$$

For $s^{(2)}(0)$ and $s^{(4)}(0)$ this yields

$$s^{(2)}(0) = -\frac{2\beta}{h^2}, \quad s^{(4)}(0) = \frac{12\beta^2}{h^4}.$$

Hence for the contributions of the term $P_h(r)D_x s$ we get

$$P_h(r)D_x s(x_i - r) = -\beta(1-2\alpha)\frac{1}{h} + R_1,$$

for the implementation (5.18) and

$$P_h(r)D_x s(x_i - r) = 2\beta^2(1-2\alpha)(\alpha-1)\alpha\frac{1}{h} + R_2,$$

for the implementation (5.22). For both implementations we find

$$P_h(r)D_x s(x_i - r) \sim \frac{1}{h} + R.$$

In fact, for other gradient implementations we will still usually have

$$P_h(r)D_x s(x_i - r) = h^n s^{(n+1)}(0) + R \sim \frac{1}{h} + R.$$

Note that we are not stating this as an asymptotic result; when we consider the limit $h \to 0$ and keep γ fixed, the contribution $P_h(r)D_x s(x_i - r)$ will vanish. However, for a fixed sensible choice of the mesh width h relative to the spatial extent of the source term, the term $P_h(r)D_x s(x_i - r)$ yields a term proportional to $1/h$.

Implemented resolved source term

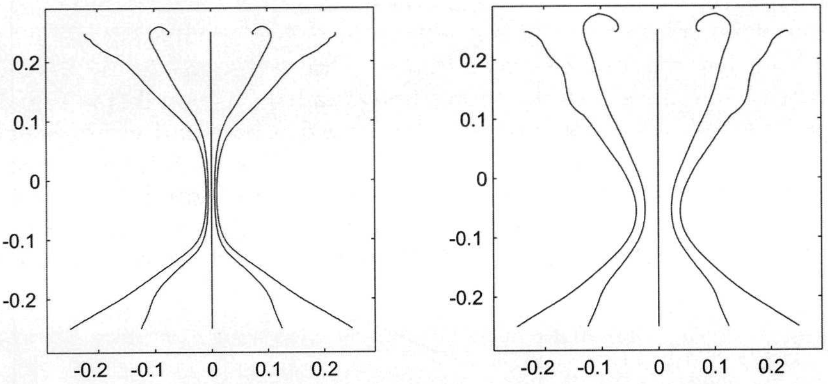
We now consider a very simple type of source term for which $P_h(r)D_x s(x_i - r)$ vanishes in a natural manner. Denote the source term's spatial extent by l. When $h > l/2$ we treat the source term as a highly localized source term discussed earlier. When $h < l/2$ we treat the source term as a hat function that is zero outside its base of length l, has its maximum halfway that base and varies linearly between the maximum and the end points of the base. A straightforward calculation shows that for this source term the gradient implementation (5.22) always yields $P_h(r)D_x s(x_i - r) = 0$. The implementation (5.18) still gives $P_h(r)D_x s(x_i - r) \sim 1/h + R$ and should therefore not be used in conjunction with this source term.

5.6 Numerical results

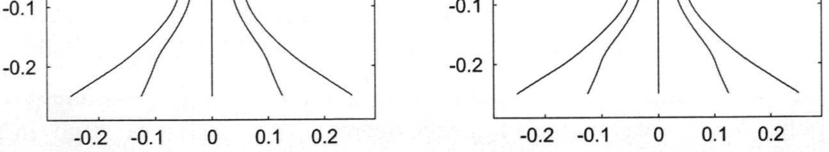
5.6.1 Convergence

In Figure 3a-d axon growth paths are shown for 5 axons growing towards 5 targets. See section 2 for the values of the problem parameters that were used. The sub-figures are computed on grids of increasing resolution ranging from 65 × 65 grid cells to 513 × 513 grid cells. Before starting the actual integration the field equations were marched to steady state while keeping the axons fixed at their initial positions. During the integration the target concentration field was kept fixed at the steady state, which is natural since it is independent of the axons positions. Putting the field equations in steady state beforehand is justified if the time scale of the gradient equation is an order of magnitude larger then the time scale of the field equations. The source term was taken to be of the hat function type, the differential operators were discretized by central differences and the interpolation was done with bi-linear interpolation, as in (5.22) for the 1D case. The time stepsizes used for time integration were taken inversely proportional to the mesh width and are given in the figures' caption.

In Figure 4 the path of a single axon is plotted for different grid resolutions. From this plot we see that the observed path converges, however non-monotonically. Hence, provided the mesh width and step size are small enough, taking a smaller mesh width and step size does not yield a different path for the axons. Mesh width and step-size are reduced simultaneously to avoid stability problems. When only the mesh width is reduced, the method eventually becomes unstable. This is not in accordance with the stability analysis for the simple test model from Section 4. Convergence is only observed for a very fine mesh width due to the sensitivity



a) 65 x 65 grid, step size = 4.0 b) 129 x 129 grid, step size = 2.0



c) 257 x 257 grid, step size = 1.0 d) 513 x 513 grid, step size = 0.5

Figure 5.3: Axon growth paths for different grid resolutions

of the problem. Small changes in mesh width can have a strong impact on the paths of the axons. This is not unexpected since the effects of mesh width related discretization errors on the axon path accumulate over the entire time integration interval. Therefore it seems attractive to use an effective higher order spatial discretization.

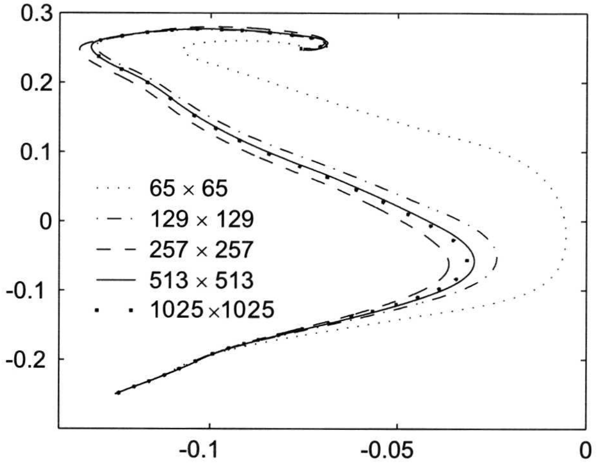


Figure 5.4: Single axon growth path for different grid resolutions

5.6.2 Stability

Since we do not have a textbook condition for the maximum allowable time step-size for our method applied to the axon growth problem, we have done numerical experiments to determine a maximum step size. We ran our program with different step-sizes and looked for signs of instability, e.g., amplified wiggles in the axon paths. When we saw these signs we deemed the step size too large. In Table 1 we have listed the thus determined maximum step sizes for several different mesh widths. From the table we see that a smaller step size is required on finer spatial grids. Halving the mesh widths requires halving the time step-size, approximately. It thus seems worthwhile to examine more stable methods.

5.6.3 Efficiency

An important merit of our method is that it solves the problem at hand efficiently compared to a number of other methods that one could consider. This merit comes forth from the explicit gradient equation treatment and the spatial factorization which greatly reduces the complexity of the linear algebra problem that needs to

Grid	maximum step size
65 × 65	12.5
129 × 129	6.1
257 × 257	3.0
513 × 513	1.4

Table 5.1: Experimentally observed maximum step sizes

be solved. Due to the factorization, small sized one-dimensional systems need to be solved instead of one large two-dimensional system.

To get some quantitative idea of the efficiency of our method, we compared it with another one, i.e., the Runge-Kutta-Chebyshev (RKC) method described in [8]. This method is fully explicit and is stabilized at the expense of additional function evaluations. We applied the RKC method to the same problem and obtained the same solutions with it. In Table 2 the wall-clock times for RKC and our Rosenbrock method (ROS2) are listed. These times refer to runs on a single processor on identical hardware with the same time step-sizes. The RKC program was written in Fortran while our ROS2 method was implemented in C, but this should not have a strong influence on the run times. As we can see from Table 2, ROS2 is faster than RKC, however the difference is only about 30%.

Grid	ROS2 runtime	RKC runtime
65 × 65	14	19
129 × 129	110	149
257 × 257	815	1207
513 × 513	7124	11431

Table 5.2: Runtimes for our ROS2 method versus the RKC method

5.6.4 Parameter sensitivity

For the numerical results to be in qualitative agreement with biological experiments, we ought to observe a bundling and debundling of the axons. For certain sets of problem parameters, this indeed does occur, see Figure 3. However, it is interesting to know whether this behaviour persists when the parameters are slightly perturbed. We have found that around a set of parameters for which bundling and debundling occurs, an interval of parameter values exists for which this still holds. Outside this interval, either bundling, debundling or both no longer occur. In Table 3 this parameter sensitivity is portrayed. The results in Table 3 refer to the same experiment as before, performed on a 257 × 257 grid. All parameters are kept fixed, except for λ_v, the parameter for growth cone sensitivity to cone derived attractant. For $\lambda_v \in [0, 3.88 \cdot 10^{-6})$ the axons do bundle but do not debundle, i.e., they grow jointly to a single target. For $\lambda_v \in [3.88 \cdot 10^{-6}, 6.04 \cdot 10^{-6})$ the axons bundle and partially debundle; they ultimately grow towards three targets. For

$\lambda_v \in [6.04 \cdot 10^{-6}, 7.81 \cdot 10^{-6})$ the axons bundle and debundle completely in that they each grow to a different target. For $\lambda_v \geq 7.81 \cdot 10^{-6}$ the axons no longer bundle and hence do not debundle either. Instead they grow away from each other and eventually leave the computational domain. We see that around $\lambda_v = 5 \cdot 10^{-6}$ full bundling and debundling occurs but also that a change of λ_v by about 20% prohibits either bundling or debundling to occur. In other words, the system is rather sensitive to changes in parameters since a 20% change in a single parameter can change the qualitative behavior of the solution completely.

End points	λ_v
1	$[0, 3.88 \cdot 10^{-6})$
3	$[3.88 \cdot 10^{-6}, 6.04 \cdot 10^{-6})$
5	$[6.04 \cdot 10^{-6}, 7.81 \cdot 10^{-6})$
-	$[7.81 \cdot 10^{-6}, \infty)$

Table 5.3: Degree of debundling for different parameter ranges

5.6.5 Source term and gradient implementation

According to Section 5 it is imperative to have a source term and gradient implementation such that the discrete gradient of the discrete source vanishes at the origin of the source. To illustrate this claim numerically, we considered a single parameter set and ran tests for different implementations of source term and gradient. In Figure 5a the final axon paths are shown for the bilinear source term and the bilinear interpolation and gradient implementation. In Figure 5b the axon paths are shown for a piecewise constant source term and piecewise constant gradient implementation. We see that both figures show qualitatively the same set of growth paths. The paths shown in Figure 5c were obtained with piecewise constant source term and bilinear interpolation while the paths in Figure 5d were obtained with bilinear source term and piecewise constant interpolation. If we now consider condition (5.21) then we see that the implementations that yield Figures 5a and 5b satisfy this condition while the implementations that yield Figures 5c and 5d do not. This is clearly reflected in the axon paths; the paths seen in Figures 5c and 5d are qualitatively different from each other and from the paths in Figures 5a and 5b. Figures 5c and 5d clearly illustrate what can go wrong if the implementation of the source term and gradient are not chosen carefully.

5.7 Discussion

Rosenbrock time stepping with an appropriate Jacobian matrix proved well suited to the mixed parabolic-gradient problem that was considered. The freedom in choosing an approximation for the Jacobian allows the parabolic and gradient equations to be treated almost independently from each other. An extension to

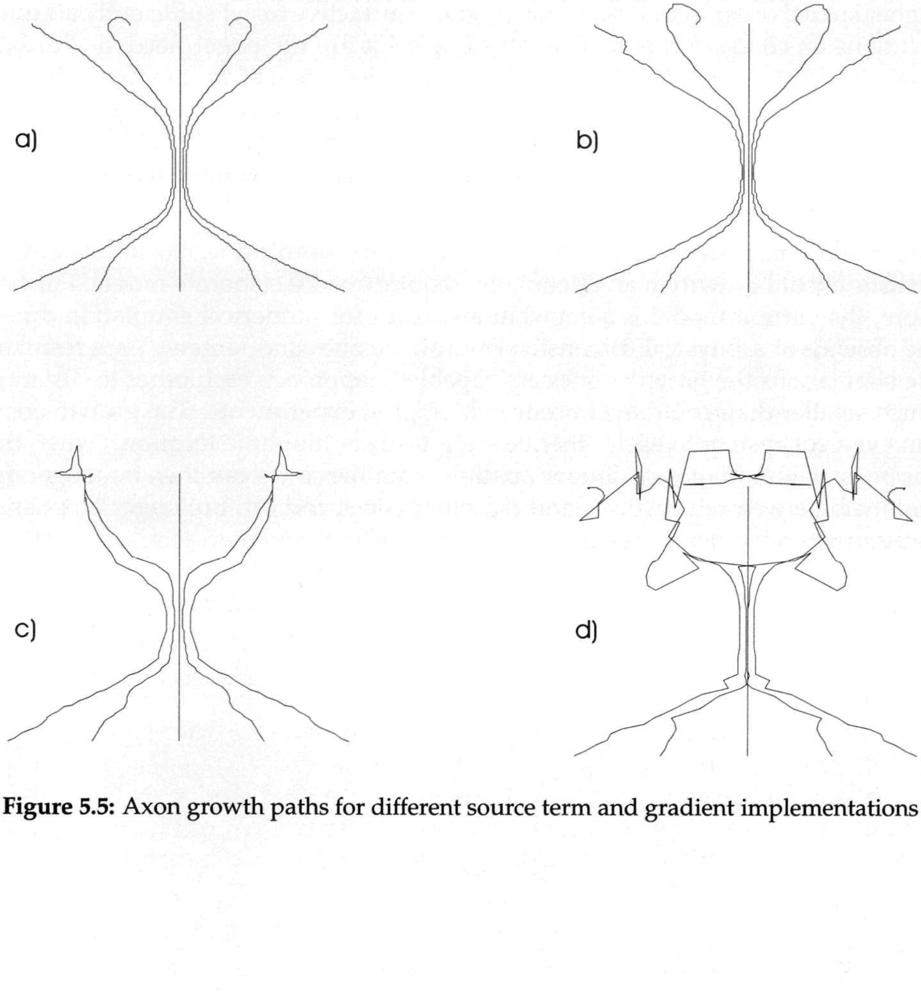


Figure 5.5: Axon growth paths for different source term and gradient implementations

spatially three-dimensional problems seems certainly feasible. Spatial factorization would then be done for all three spatial dimensions leading to even larger gains in efficiency relative to a non-factorized approach.

It was shown that the implementation of the source term and gradient detection is somewhat delicate and should not be constructed independently of each other. In particular, a condition was derived that the combined implementation of source term and gradient should fulfil, see condition (5.21).

Since the axon paths are quite sensitive to refinements in the mesh width very fine meshes are needed to get spatially converged solutions. In a future method higher spatial order of discretization might be attractive to get sufficiently accurate solutions on coarser grids so that very fine grids are no longer needed. Furthermore, it might be interesting to see wether the integration of the gradient equation can be stabilized since in the current Rosenbrock-Approximate Jacobian approach it seems to dictate the overall stability. The gradient equation could be handled implicitly or some form of RKC method could be used where the number of integration stages is increased to enhance stability, as in [8].

The model considered for axonal growth is a very simple one. To simulate more realistic axonal growth in any detail will require a more elaborate model. Furthermore, the current model is somewhat awkward for numerical simulation due to the absence of a physical dimension of growth cones and targets. As a result, in the simulations the growth cones are capable to approach each other to distances much smaller than could ever occur in biological experiments. The growth cones can even collapse onto each other, leaving them at the same location. When this happens, debundling is no longer possible since the cones can then no longer differentiate between themselves and the other cones and can no longer be pushed away from each other.

BIBLIOGRAPHY

[1] R. M. Beam, R. F. Warming, *An implicit finite-difference algorithm for hyperbolic systems in conservation-law form*, J. Comput. Phys. 22, pp. 87-110, 1976. Pages: 107

[2] E. G. D'yakonov, *Difference systems of second order accuracy with a divided operator for parabolic equations without mixed derivatives*, USSR Comput. Math. Math. Phys., 4(5), pp. 206-216, 1964. Pages: 107

[3] G. J. Goodhill, *Diffusion in axon guidance*, Eur. J. Neurosci. 9, pp. 1414-1421, 1997. Pages: 98

[4] E. Hairer, G. Wanner, *Solving ordinary differential equations II. Stiff and differential algebraic problems*, 2nd ed., Springer-Verlag, Berlin, 1996. Pages: 104

[5] H. G. E. Hentschel, A. van Ooyen, *Models of axon guidance during development*, Proc. R. Soc. Lond. B 266, pp. 2231-2238, 1999. Pages: 98, 102, 113

[6] P. J. van der Houwen, B. P. Sommeijer, *Approximate factorization for time-dependent partial differential equations*, J. Comput. Appl. Math. 128, pp. 447-466, 2000. Pages: 107

[7] B. Lastdrager, B. Koren, J. G. Verwer, *Solution of time-dependent advection-diffusion problems with the sparse-grid combination technique and a Rosenbrock solver*, Journal of Computational Methods in Applied Mathematics, Vol. 1, number 1, pp. 86-98, 2001. Pages: 104, 107

[8] J. G. Verwer, B. P. Sommeijer, *A numerical study of mixed parabolic-gradient systems*, J. Comput. Appl. Math. 132, pp. 191-210, 2001. Pages: 102, 105, 117, 120

[9] J. G. Verwer, E. J. Spee, J. G. Blom, W. Hundsdorfer, *A second-order Rosenbrock method applied to photochemical dispersion problems*, Siam J. Sci. Comput. 20, pp. 1456-1480, 1999. Pages: 103, 107

SUMMARY

Since many real-life processes from engineering, physics, economics and a range of other disciplines can be described with differential equations, there is a need for practical methods for solving differential equations. Only in rare cases can differential equations be solved analytically. For the majority of differential equations, at best approximate solutions can be computed with the help of computers. With the power of modern computers and the sophistication of current algorithms this can often be done in a straightforward manner. However, there still exist numerous problems for which the numerical solution of the underlying differential equations is not straightforward.

A well-known example of a numerically difficult problem is the solution of the full Navier-Stokes equations, especially under turbulent conditions. Another example is that of global atmospheric transport models used for modeling pollution or forecasting the weather. In these models the number of unknowns required to accurately capture the spatial variations of the solutions can be excessively large.

The main focus of this thesis lies on a method that holds the promise of alleviating the restriction of excessively large numbers of unknowns. This is the sparse grid combination technique, which aims to solve a set of differential equations using significantly fewer unknowns. In fact, in the limit of high accuracy, the number of unknowns required by the sparse grid combination technique is independent of the spatial dimensionality of the problem. E.g., asymptotically a spatially 3D problem requires the same order of unknowns as a spatially 1D problem.

The sparse grid combination technique can be understood as a multivariate extrapolation technique. Instead of solving a set of differential equations on a single grid, solutions are obtained on a number of semi-coarsened grids. After solving these semi-coarsened problems, the solutions are combined to obtain a single, more accurate solution. In Chapters 2, 3 and 4 of this thesis error expressions are derived that measure the approximation error due to the sparse grid combination technique. Furthermore, test cases are considered numerically to validate these expressions and to test the applicability of the technique for a number of problems.

It becomes apparent that the sparse grid combination technique can be highly efficient for some problems. Especially problems that contain locally lower dimensionality are well suited for the sparse grid combination technique. E.g., in Chap-

ter 4 the sparse grid combination technique is shown to be effective for a 2D-flow problem containing locally 1D solution layers. However, it also becomes apparent that the sparse grid combination technique is less well suited for other problems. E.g., in Chapter 3 it is shown that for a model problem without locally lower dimensionality the sparse grid combination technique is less efficient than Richardson extrapolation.

In Chapter 5 of this thesis a mixed gradient-diffusion problem is considered. The motivating application for this problem is that of axon growth studied in neurobiology. In biological experiments it is observed that axons bundle and debundle during their growth. An initial model is considered which assumes that the axons secrete chemical substances through which they communicate with each other. It is shown that this initial model can already predict the bundling and debundling behavior, albeit for a small range of parameters. Furthermore, an important complication inherent in the model is pointed out. I.e., in a numerical implementation there exists a danger that the axons blind themselves with their own chemical emissions. A condition is presented that the numerical scheme must satisfy in order to avoid this self-blinding of the axons.

SAMENVATTING

Aangezien veel alledaagse verschijnselen uit de techniek, natuurkunde, economie en een aantal andere disciplines beschreven worden door differentiaalvergelijkingen is er een behoefte aan praktische oplosmethoden voor differentiaalvergelijkingen. Alleen in uitzonderlijke gevallen kunnen differentiaalvergelijkingen analytisch worden opgelost. Voor het merendeel van de differentiaalvergelijkingen kunnen hoogstens benaderende oplossingen uitgerekend worden met behulp van computers. Met de rekenkracht van moderne computers en algoritmes zijn deze benaderende oplossingen vaak relatief eenvoudig uit te rekenen. Er bestaan echter ook ettelijke problemen waarvoor het numeriek oplossen van de onderliggende differentiaalvergelijkingen niet eenvoudig is.

Een bekend voorbeeld van een numeriek moeilijk probleem is het oplossen van de volledige Navier-Stokes vergelijkingen, met name in het geval van turbulente stroming. Een ander uitdagend probleem ligt in globale atmosferische transportmodellen voor het modelleren van luchtvervuiling en het doen van weersvoorspellingen. In deze modellen is het aantal vrijheidsgraden dat nodig is om de ruimtelijke variaties te beschrijven erg groot.

De nadruk van dit proefschrift ligt op een methode welke in potentie het probleem van te veel vrijheidsgraden kan oplossen. Het gaat om de 'sparse-grid combinatietechniek', bedoeld om een stelsel differentiaalvergelijkingen op te lossen met significant minder vrijheidsgraden. In de limiet van hoge nauwkeurigheid is het aantal vrijheidsgraden zelfs onafhankelijk van de ruimtelijke dimensionaliteit van het probleem. Een ruimtelijk 3D probleem vraagt bijvoorbeeld asymptotisch dezelfde ordegrootte van onbekenden als een ruimtelijk 1D probleem.

De sparse-grid combinatietechniek kan worden opgevat als een multivariate extrapolatietechniek. In plaats van een stelsel differentiaalvergelijkingen op te lossen op een enkel rooster worden oplossingen uitgerekend op meerdere grovere roosters. Na het oplossen van deze grovere problemen worden de oplossingen gecombineerd om zo een enkele, meer nauwkeurige oplossing te verkrijgen. In de Hoofdstukken 2, 3 en 4 wordt het duidelijk dat de sparse-grid combinatietechniek zeer efficiënt kan zijn voor bepaalde problemen. Met name problemen met lokale lagere dimensionaliteit zijn zeer geschikt voor de sparse-grid combinatietechniek. In Hoofdstuk 4 blijkt de sparse-grid combinatietechniek bijvoorbeeld effectief voor een 2D stromingsprobleem met lokaal ééndimensionaal oplossingsgedrag. Het blijkt echter ook dat de sparse-grid combinatietechniek minder geschikt is voor andere problemen. Zo wordt in Hoofdstuk 3 aangetoond dat voor een model probleem zonder lokale lagere dimensionaliteit de sparse-grid combinatietechniek minder efficiënt is dan Richardson extrapolatie.

In Hoofdstuk 5 wordt een gemengd gradiënt-diffusie probleem beschouwd. De motiverende applicatie voor dit probleem is een axonen groeimodel uit de neurobiologie. In biologische experimenten blijkt dat axonen tijdens hun groei eerst een bundel vormen en later deze bundel weer verlaten. In een initieel model wordt verondersteld dat de axonen chemische substanties uitscheiden waarmee zij met elkaar communiceren. Het wordt aangetoond dat dit initiële model voor een klein parameterbereik inderdaad voorspelt dat bundelvorming en -verbreking optreedt. Verder wordt een belangrijke complicatie aangetoond, welke inherent is aan het model. Er bestaat namelijk het gevaar dat de axonen zichzelf verblinden. Een voorwaarde wordt gepresenteerd waaraan het numerieke schema moet voldoen om te garanderen dat deze zelfverblinding niet optreedt.

DANKWOORD

Op deze plek wil ik graag een aantal mensen noemen die direct of indirect hebben bijgedragen aan mijn proefschrift. In de eerste plaats ben ik dank verschuldigd aan mijn co-promotor Barry Koren en mijn promotor Jan Verwer voor de goede begeleiding die zij mij gegeven hebben gedurende mijn promotieonderzoek. Barry's enthousiaste begeleiding heb ik altijd als zeer motiverend ervaren. Jan is voor mij een zeer gedegen begeleider geweest. Met name zijn expertise op het gebied van tijdsintegratie was voor mij zeer leerzaam.

Het Centrum voor Wiskunde en Informatica heb ik als een zeer prettige werkomgeving ervaren, met name door de mensen die er werken. Mijn kamergenote Debby Lanser was voor mij bepalend voor de sfeer. Ik heb haar leren kennen als een grappige mix van vrolijkheid en een enorme werkijver. De aanwezigheid van andere mede oio's en postdocs, met name Sander, Patrick, Harald, Mervyn, Lubor, Jason, Johannes, Bob en Menno heb ik ook altijd als erg gezellig ervaren. Van de vaste staf wil ik Ben, Willem, Piet, Mark en Joke bedanken omdat ze altijd bereid waren mee te denken. Joke wil ik ook bedanken voor het organiseren van de inline-skate tochtjes en het geven van noodzakelijke skate tips.

I would like to thank Ulrich Rüde for his interest in my work. I especially appreciate the opportunity he gave me to do some joint work with him in Erlangen. His view of the sparse grid combination technique as a special case of a multivariate extrapolation method was very clarifying to me.

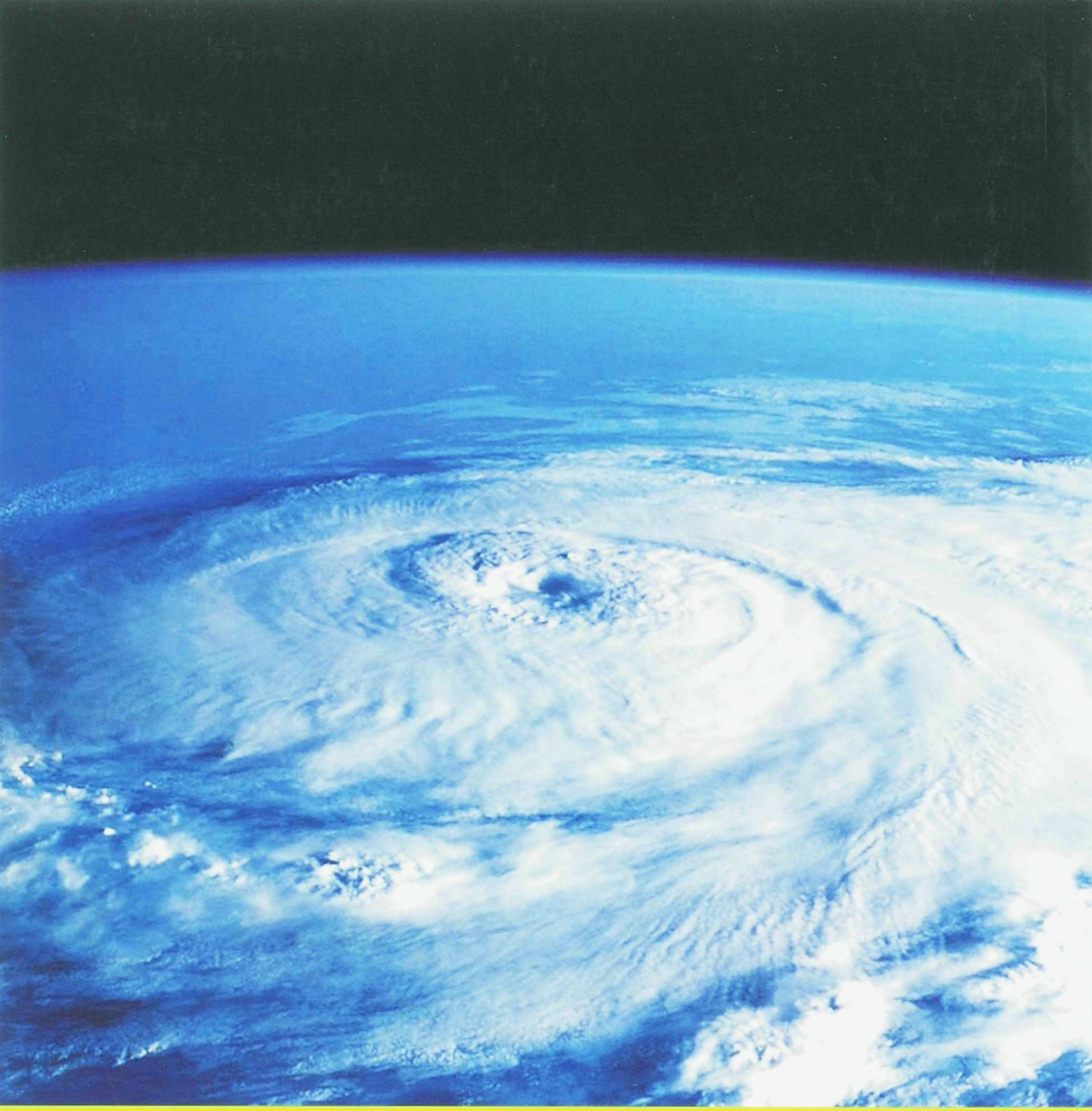
Op het persoonlijk vlak ben ik mijn moeder erg dankbaar voor al haar steun en motivatie. Mijn keuze voor een exacte richting en aansluitende promotie zijn voor een belangrijk deel haar verdienste. Tenslotte wil ik Ancella bedanken voor al haar liefde en warmte. Ancella, zonder jou was er niets aan geweest. Ik bewonder je karakter en ik ben ontzettend blij met je.

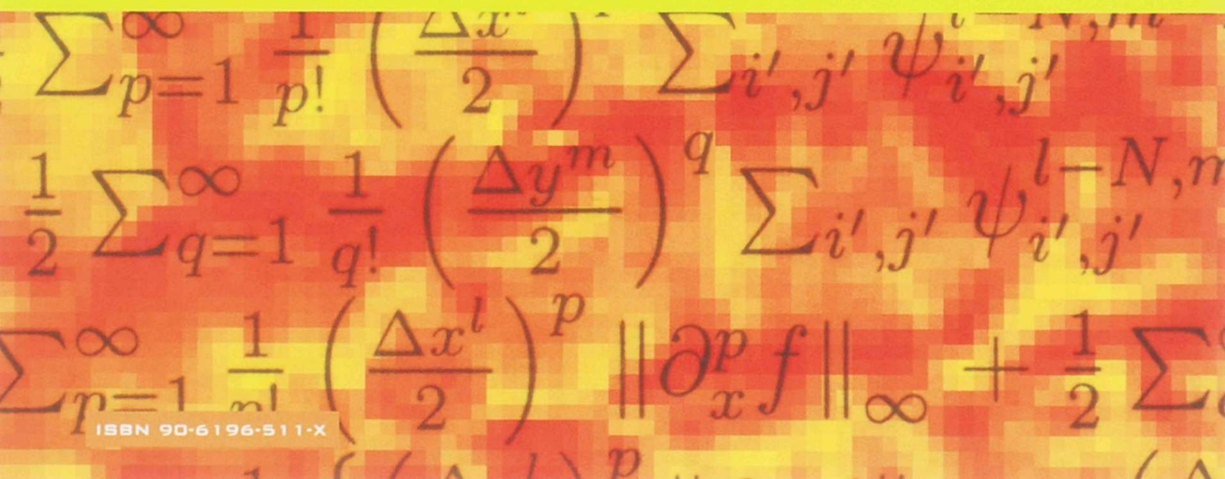
CURRICULUM VITAE

I studied theoretical physics at the University of Amsterdam during the period 1993-1997 and obtained my Master of Science degree in August of 1997 with honors. I did the research for my Masters' thesis at the institute for Atomic and Molecular Physics, AMOLF, in Amsterdam under supervision of dr. Adriaan Tip and dr. Jan Verhoeven. This research led to a patent on a new type of x-ray source and a publication in Physical Review. Philips funded the patent after the Philips Patent Office classified it to have "very high potential commercial importance". AMOLF, Philips NatLab and Eindhoven University of Technology are currently developing the source.

During the period 1998-2002, I worked as a math Ph.D. student at the Center for Mathematics and Computer Science, CWI, in Amsterdam under supervision of prof.dr. Jan Verwer and dr.ir. Barry Koren. This work led to several presentations at international conferences, publications in international journals and this thesis.

Starting April 2002, I will be employed by BHP Billiton International Metals as a deal structurer/analyst in the energy coal division at the marketing head office in The Hague.




$\sum_{p=1}^{\infty} \frac{1}{p!} \left(\frac{\Delta x^l}{2}\right)^p \sum_{i',j'} \psi_{i',j'}$
$\frac{1}{2} \sum_{q=1}^{\infty} \frac{1}{q!} \left(\frac{\Delta y^m}{2}\right)^q \sum_{i',j'} \psi_{i',j'}^{l-N,m}$
$\sum_{p=1}^{\infty} \frac{1}{p!} \left(\frac{\Delta x^l}{2}\right)^p \|\partial_x^p f\|_\infty + \frac{1}{2} \sum$

ISBN 90-6196-511-X