Discrete total variation in multiple spatial dimensions and its applications

by

Alexey Smirnov

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Applied Mathematics

Waterloo, Ontario, Canada, 2024

Examining Committee Membership

The following served on the Examining Committee for this thesis. The decision of the Examining Committee is by majority vote.

External Examiner:	Christina Christara Professor, Dept. of Computer Science, University of Toronto
Supervisor:	Lilia Krivodonova Professor, Dept. of Applied Mathematics, University of Waterloo
Internal Member:	Hans De Sterck Professor, Dept. of Applied Mathematics, University of Waterloo
Internal-External Member:	Justin Wan Professor, David R. Cheriton School of Computer Science, University of Waterloo
Other Member(s):	Stephen Vavasis Professor, Dept. of Applied Mathematics, University of Waterloo

Author's Declaration

This thesis consists of material all of which I authored or co-authored: see Statement of Contributions included in the thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Statement of Contributions

A. Smirnov was the sole author of Chapters 1, 2, 5 and 6, which were written under the supervision of Dr. L. Krivodonova. Chapters 1, 2, and 6 were not written for publication. The research presented in Chapters 3 and 4 was conducted at the University of Waterloo by A. Smirnov under the supervision of Dr. L. Krivodonova and been submitted for publication in peer-reviewed journal, the preprint is available online (https://arxiv.org/abs/2110.00067). The contents of the Chapter 5 are prepared for publication and have not been published.

A. Smirnov contributions to Chapters 3 and 4 are clarified below.

Chapter 3: The research presented in this chapter is a product of collaboration between A. Smirnov and Dr. L. Krivodonova and is partially included in the preprint above. A. Smirnov and developed the theoretical formalism, worked out all of the technical details and proofs, performed the numerical simulations, and wrote the first draft of the manuscript. Editing and proof verification were provided by Dr. L. Krivodonova.

Chapter 4: The research presented in this chapter is a product of collaboration between A. Smirnov and Dr. L. Krivodonova and is partially included in the preprint above. A. Smirnov and developed the theoretical formalism, worked out all of the technical details and proofs, performed the numerical simulations, and wrote the first draft of the manuscript. Editing and proof verification were provided by Dr. L. Krivodonova.

Abstract

Total variation plays an important role in the analysis of stability and convergence of numerical solutions for one-dimensional scalar conservation laws. However, extending this approach to two or more spatial dimensions presents a formidable challenge. Existing literature indicates that total variation diminishing solutions for two-dimensional hyperbolic equations are limited to at most first-order accuracy.

The presented research contributes to overcoming the challenges associated with extending total variation to higher dimensions, particularly in the context of hyperbolic conservation laws. By addressing the limitations of conventional discrete total variation definitions, we seek answers to critical questions associated with the total variation diminishing property of solutions of scalar conservation laws in multiple spatial dimensions. We adopt a more accurate dual discrete definition of total variation, recently proposed in [40], for measuring the total variation of grid-based functions. Dual total variation can be computed as a solution to a constrained optimization problem. We propose a set of conditions on the coefficients of a general five-point scheme so that the numerical solution is total variation diminishing in the dual discrete sense and validate that through numerical experiments.

Apart from the contributions to the analysis of numerical methods for two-dimensional scalar conservation laws, we develop an algorithm to efficiently compute the dual discrete total variation and develop an imaging method, based on this algorithm. We study its performance in computed tomography image reconstruction and compare it with the state-of-the-art total variation minimization-based imaging methods.

Acknowledgements

I would like to thank my supervisor, Lilia, for her continued guidance and support throughout my time in at University of Waterloo. Next, I would like to thank Dr. Vavasis and Dr. De Sterck for the many fruitful discussions we had. Finally, I would like to thank my wife Irina for her love, patience, and support, which helped me get through the challenging times.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Total variation (TV) is a convex functional that became a common choice for regularization in imaging methods. There is much evidence for its effectiveness in addressing such problems as denoising, image recovery from noisy measurements, and image analysis. Total variation of a characteristic function of a set is related to the length of the boundary. This has been successfully employed for image segmentation problems. In particular, it can be used to express minimal surface problems. In the domain of computer vision, for instance, TV became a fundamental tool for object recognition, and its recent applications in machine learning algorithms showcase its contribution to feature selection and dimensionality reduction.

In the field of scientific computing, total variation plays a crucial role as a tool to ensure the nonlinear stability of high-order numerical schemes for solving hyperbolic partial differential equations (PDEs). These equations often involve sharp discontinuities that require careful handling to maintain accuracy and stability. TV is used in a special class of schemes to address this challenge by ensuring that the total variation of the numerical solution, a measure of its fluctuations, does not increase over time. This effectively prevents the growth of oscillations near discontinuities. We will introduce two main uses of TV that found the most success. First, we will introduce the TV as a property of the solutions of scalar conservation laws.

Physical laws dictate that key quantities, including mass, momentum and others, are globally conserved. Over time, these quantities evolve by mathematical expressions known as conservation laws, which appear in various practical applications.

Let $u = u(\boldsymbol{x}, t) \in \mathbb{R}^N \times [0, \infty)$ represent the density of a physical quantity, for instance, mass density. Let t represent time, and $\boldsymbol{x} \in \mathbb{R}^N$ be a spatial variable. Then a conservation law can be written in an integral form as

$$\frac{d}{dt} \int_\Omega u \, d\boldsymbol{x} = - \int_\Sigma f \cdot \boldsymbol{n} \, dS \tag{1.1}$$

Here, Σ is the boundary of Ω, $f = f(u, \boldsymbol{x}, t)$ is a smooth nonlinear mapping, which denotes the flux of the conserved quantity, and $\boldsymbol{n}$ is the unit outer normal to the boundary of Σ. The expression indicates that, in the absence of flux, u is preserved in the domain, and any changes in u over time result from the inflow or outflow across the boundary of Σ.

Assume that $f(u, \boldsymbol{x}, t)$ is sufficiently smooth. Then we can rewrite (1.1) as

$$\int_\Omega \left(\frac{du}{dt} + \nabla_{\boldsymbol{x}} f \right) d\boldsymbol{x} = 0,$$

which yields the following differential equation

$$u_t + \nabla_{\boldsymbol{x}} f = 0, \tag{1.2}$$

that must hold for all $\boldsymbol{x} \in \mathbb{R}^N, t > 0$. This is also known as the differential form of the conservation law.

If we consider $\Omega = \mathbb{R}^N$ and complement the PDE (1.2) with an appropriate initial condition, we will arrive at Cauchy problem

$$u_t + \nabla_{\boldsymbol{x}} f d\boldsymbol{x} = 0, \ \boldsymbol{x} \in \mathbb{R}^N, \ t > 0, \tag{1.3}$$

$$u(\boldsymbol{x}, 0) = u_0, \ \boldsymbol{x} \in \mathbb{R}^N. \tag{1.4}$$

It is well known that this Cauchy problem may not have smooth solutions due to the nonlinear structure of the eigenvalues. Moreover, the solution may be non-unique, i.e. it is possible for many solutions to share the same initial data. This is due to the inaccuracy of the differential equation expressing the physical law.

While solving hyperbolic equations can be used as a powerful tool, we must ensure the problem has a unique and physically accurate weak solution by imposing additional constraints. Naturally, we would want the solution to match the vanishing viscosity limit of a proper viscous equation. However, directly applying this in the hyperbolic context proves challenging. Instead, a variety of other conditions admissibility conditions, have been developed. These can be directly applied to weak solutions of hyperbolic equations to verify their physical validity.

In this work, we will restrict our attention to hyperbolic scalar conservation laws with strictly convex entropy. It was established that such problems have unique weak solutions, satisfying the entropy criterion. For this class of PDEs, it was established that TV of the weak solutions is a non-increasing function of time [60, 39]. This fact was used to a great extent to develop high-order accurate numerical methods for scalar conservation laws in one spatial dimension.

The second major use of TV is as a regularization tool for image reconstruction or recovery from noisy data. A classic example of the TV minimization-based imaging problem is the Rudin-Osher-Fatemi (ROF) model [114].

Let us consider a bounded domain $\Omega \in \mathbb{R}^2$. Given $v : \Omega \to \mathbb{R}$ find $u : \Omega \to \mathbb{R}$ such that

$$\int_\Omega |\nabla u| \, d\boldsymbol{x} + \lambda \int_\Omega (u - v)^2 \, d\boldsymbol{x} \tag{1.5}$$

is minimized. In (1.5), v is an approximation of the desired solution u, obtained in an experiment or by any other means. The term $\int_\Omega |\nabla u| \, d\boldsymbol{x}$ is the total variation for a differentiable function u and $\lambda > 0$ is the regularization parameter that balances the trade-off between data fidelity and the smoothness of the reconstructed image.

In this context, the TV functional measures the variation in intensity within a grayscale image u and minimal TV corresponds to the smallest possible oscillations in the image, i.e. lowest levels of random noise. The goal of (1.5) is to remove noise from the image u by minimizing the total variation under the fidelity constraints. The solution to (1.5) finds a piecewise constant approximation to u, preserving its important features, such as sharp edges. The ROF problem has had a profound impact on image processing and related fields.

In most practical applications (1.5) is solved for discrete images and therefore, the solution requires discretization of the TV functional. Despite its simplicity, the ROF model remains widely employed across various applications and serves as a benchmark for evaluating the efficiency of contemporary learning-based image reconstruction approaches. More details can be found in [114], [22]. A noteworthy extension of this model is TV-based denoising and processing for color images, known as Colour TV [12].

The accuracy of TV estimation depends heavily on the chosen discrete TV definition or discretization scheme. This thesis focuses on studying the properties of TV discretizations when applied to solutions of two-dimensional scalar conservation laws. The majority of existing literature relies on TV discretizations that use finite difference approximations of the gradient. Notably, only a handful of contributions explore alternative approaches [5, 70, 133, 40]. The work of [5] proposes a discrete approximation of total variation using discretization of the Raviart-Thomas dual field which are conforming finite element approximations of divergence on the square grid [110]. Recently, [26] proposed a different discrete total variation, based on the approximation with Crouzeix–Raviart finite element space. This approach has the advantage of having the error depending only on the local curvature of the mesh and not on grid orientation as in conventional discretizations.

This idea was used to formulate several definitions of discrete TV, e.g. [70, 40]. In [70] the authors considered a staggered grid approximation of the divergence operator and enforce the constraints on the dual field at two points per pixel. Then a constrained optimization is solved to find TV. This approach based on the optimization problem for discrete grid function is referred to as the discrete dual TV. Since TV is a convex functional, the resulting optimization problem has a global maximum and can efficiently be solved by standard iterative algorithms. The convergence of discrete dual TV under mesh refinement has been shown in [28, 27].

We particularly focus here on the discrete TV definition proposed in [40], as it is superior to other discrete dual TV definitions, as shown in several studies, e.g. [27, 28]. This definition has been developed and studied in application to image analysis but has not yet been applied to stability analysis of numerical schemes.

By exploring different approaches to TV discretization we aim to obtain a better definition for discrete TV as a tool for studying stability of numerical schemes. We aim to improve existing techniques, with the overarching goal of contributing to various fields, including the stability of numerical methods for hyperbolic conservation laws in multiple dimensions, imaging algorithms, and broader computational contexts.

1.1 Weak solutions of hyperbolic conservation laws

Nonlinear hyperbolic equations are known to develop shocks over time. To resolve discontinuous solutions, a weak solution to (1.6)-(1.7) is usually considered.

In one-dimensional space, let us assume that $f = f(u)$ and set $u(\cdot, 0) = u_0$ to obtain the following Cauchy problem

$$u_t + f(u)_x = 0, \ x \in \mathbb{R}, \ t > 0 \tag{1.6}$$
$$u(x, 0) = u_0, \ x \in \mathbb{R}. \tag{1.7}$$

We multiply the differential equation (1.6) by a test function $\varphi \in C_c^1(\mathbb{R}; [0, \infty))$ and inte-

grate over space and time to obtain

$$\int_0^\infty \int_{-\infty}^\infty \varphi u_t + \varphi f(u)_x dx dt = 0. \tag{1.8}$$

Next, apply integration by parts to (1.8) to get

$$\int_0^\infty \int_{-\infty}^\infty \varphi_t u + \varphi f'(u) u_x dx dt = -\int_{-\infty}^\infty \varphi(x,0) u(x,0) dx. \tag{1.9}$$

We call u a weak solution of the hyperbolic equation (1.6) if (1.9) holds for all $\varphi \in C_c^1(\mathbb{R}; [0,\infty))$. Notice that a function u does not need to be differentiable or even continuous for it to satisfy (1.9).

The notion of a weak solution is not strict enough to determine a unique solution to a Cauchy problem for the scalar conservation law [60]. Let us recall that a function $\eta : \Omega \to [0,\infty)$ defined on an open domain $\Omega \subset \mathbb{R}^N$ is an entropy for (1.6) with entropy flux $q : \Omega \to \mathbb{R}$ if all smooth solutions with range in Ω satisfy

$$\eta(u)_t + q(u)_x = 0. \tag{1.10}$$

Most of the conservative laws in continuum mechanics are endowed with a globally defined strictly convex entropy [87].

For the class of equation of the form (1.6), endowed with a strictly convex entropy η, the following entropy criterion is postulated. A solution u with a range in Ω is admissible if

$$\eta(u)_t + q(u)_x < 0. \tag{1.11}$$

We require weak solutions to satisfy the entropy inequality (1.11) for all entropy functions η of (1.6) to establish uniqueness. The existence of the entropy solution for scalar conservation laws is obtained by the vanishing viscosity method while uniqueness was established in [79], which also shows that the solution has a finite domain of dependence.

As for the numerical methods to solve (1.6)-(1.7), modern theory for conservation laws has three major classes of methods that found the most use in practice. They are finite-difference methods (FDMs), finite-volume methods (FVMs), and finite-element methods (FEMs). Additionally, several semidiscrete methods like the method of lines and conservative front-tracking methods can be employed.

Finite-difference methods are commonly used for solving hyperbolic conservation laws like equations. They approximate the partial derivatives in (1.6) using function values on a discrete grid of points. This transforms the partial differential equation into a system of algebraic equations that can be solved efficiently. The key advantage of FDMs is their simplicity and computational efficiency, making them particularly well-suited for large-scale simulations. However, applying FDMs effectively requires careful consideration, especially when dealing with the concept of conservative form. The difficulty lies in ensuring that the discretized terms, built from function values at grid points, still represent the net flux of the conserved quantity across the boundaries of each computational cell. Non-conservative schemes can introduce artificial sources or sinks for the conserved quantity within the computational domain, leading to inaccurate solutions that violate the conservation law. Several strategies exist to construct conservative finite-difference schemes. These often involve careful manipulation of the discretized terms to ensure they cancel out within

each cell, which aims to implement the idea of a net flux. Popular approaches include Lax-Wendroff schemes and Godunov schemes. However, these methods can become more complex to implement, particularly in higher dimensions or when dealing with complex boundary conditions.

First-order FVMs employ piecewise constant approximations to flux, typically ensuring stability, but at the cost of significant numerical diffusion, which mitigate sharp discontinuities. Higher-order FV methods use polynomials of higher degree, to reduce smearing, i.e. smoothing of discontinuous profiles. However, this may produce spurious oscillations in the solution regions that contain discontinuities. A common strategy that is used to capture the solution behaviour near shocks and retain the stability of the numerical scheme is to use a high-order scheme on regions where the solution is smooth and a lower order approximation around discontinuities. There are many known methods of this type, such as Godunov methods and the monotonic upstream-centered schemes [131], wave propagation methods [89, 90], the central difference schemes [103], and the essentially non-oscillatory and weighted essentially non-oscillatory schemes [116, 118].

Conservative front-tracking methods combine the FDM/FVM with the standard front-tracking [59]. These methods are use a high-order FDM/FVM scheme and in addition they track the location of the discontinuities, essentially treating them as moving boundaries. The complexity of the problem increases with the number of shocks to be tracked. The computational complexity grows quickly as these shock interact which makes these methods too complex in practice. Moreover, predicting the shock formation and its location is a very tedious task.

Discontinuous Galerkin (DG) methods, prevalent in Finite Element Method (FEM) settings, use finite element spatial discretization with piecewise polynomial approximations. This approach allows for discontinuities at cell boundaries. DG method found a lot of success in problems involving nonuniform domains and complex boundaries. Some numerical methods can be directly extended to the multidimensional case. Evaluating the performance of numerical algorithms typically involves solving benchmark problems, for which we have limited theoretical understanding beyond scalar conservation laws. Additionally, constructing efficient high-order numerical methods for systems of conservation laws in both one and multiple spatial dimensions remains a formidable challenge.

We will now formulate the general framework for FVMs for scalar conservation laws.

1.2 Finite volume methods

Finite volume methods in one spatial dimension are based on the division of the domain Ω into a number of subdomains, called finite volumes, or cells.

Let the computational domain $\Omega \in \mathbb{R}$ be divided uniformly into elements Ω_i with left, $x_{i-1/2}$, and right, $x_{i+1/2}$, end points, where $\Delta x = x_{i+1/2} - x_{i-1/2}$ is the grid step size. The elements Ω_i are commonly called finite volumes. The FV method, akin to FD, ensures the conservation of $u(x,t)$ within these finite volumes. Assume that the numerical solution is given by a grid function $\{U^n\}_{i=1}^N$, that approximates cell averages of $u(x,t^n)$ at cell centers $\{x_0, x_1, \ldots, x_{N+1}\}$ at $t = t^n$.

We integrate the conservation law (1.6) over the cell Ω_i

$$\frac{d}{dt}\int_{\Omega_i} u(x,t)\,dx = f(u_{i-1/2}) - f(u_{i+1/2}), \tag{1.12}$$

where $u_{i-1/2} = u(x_{i-1/2}, t)$. Let

$$U_i^n = \frac{1}{\Delta x} \int_{\Omega_i} u(x, t_n)\, dx.$$

We can use this expression to develop an explicit scheme. For first order accuracy in time scheme we can find U_{n+1}^i, the average value of u over Ω_i at $t = t_{n+1}$ from the given cell averages U_n^i at time $t = t_n$. Integrate (1.12) in time from t_n to t_{n+1} to get

$$\int_{\Omega_i} u(x, t_{n+1})\, dx - \int_{\Omega_i} u(x, t_n)\, dx = \int_{t_n}^{t_{n+1}} f(u_{i-1/2}, t) - f(u_{i+1/2}, t) \tag{1.13}$$

The equation (1.13) gives us the formula to update the cell average of u after a single time step. However, we cannot evaluate the time integral on the right-hand side of (1.13) exactly, since $u(x_{i\pm1/2}, t)$ varies with time along each edge of the cell. However, this suggests that we should look for the numerical schemes of the form

$$U_i^{n+1} = U_i^n - \frac{\Delta t}{\Delta x}(F_{i+1/2}^n - F_{i-1/2}^n), \tag{1.14}$$

where $\Delta t = t_{n+1} - t_n$ and $F_{i-1/2}^n$ is an approximation to the average flux $F(U^n, t_n)$ along $x = x_{i-1/2}$ boundary of the finite volume

$$F_{i-1/2}^n \approx \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} f(u_{i-1/2}, t)\, dt. \tag{1.15}$$

For hyperbolic problems, we can approximate this average flux based on the values of U to obtain a fully discrete method. Let us assume that for some $k \geq 1$

$$F_{i-1/2}^n = F\left(U_{i-k+1}^n, \ldots, U_{i+k}^n\right).$$

We call a scheme conservative if the flux entering a given volume is identical to that leaving the adjacent volume. Finite volume schemes are conservative by construction. It follows from (1.14) that $\sum U_i^{n+1} = \sum U_i^n$ up to boundary fluxes.

We call a scheme flux-consistent, if its numerical flux satisfies

$$F(u, \ldots, u) = f(u), \quad \forall u \in \mathbb{R}. \tag{1.16}$$

The importance of this formalization of the conservative condition is expressed by the following fundamental theorem.

Theorem 1.2.1 ([87]). *If the solution U of the flux-consistent and conservative scheme (1.14) is bounded and converges almost everywhere to some function $u(x,t)$ as $\Delta x, \Delta t \to 0$, then $u(x,t)$ is a weak solution of (1.6)-(1.7).*

This theorem guarantees that when the numerical solution converges, it will converge to a solution of the conservation law, that satisfies the Rankine-Hugoniot relations in the presence of discontinuities.

In two dimensions $\Omega \subset \mathbb{R}^2$, which is subdivided into N^2 non-intersecting square subdomains $\Omega_{i,j}$. These subdomains are often referred to as finite volumes, also called cells.

Assuming each cell contains a point of the grid we write the typical 2D FV method as

$$U_{ij} - \frac{1}{\Delta x \Delta y} \int_{x_{i-1/2}}^{x_{i+1/2}} \int_{y_{j-1/2}}^{y_{j+1/2}} u(x,y)\, dx\, dy, \tag{1.17}$$

where $i = 1, \ldots, N_x - 1, \quad j = 1, \ldots, N_y - 1$ and the $U_{i,j}$ denotes the i,j-th cell average of u. The corresponding semi-discrete scheme for two-dimensional scalar conservation law

$$u_t + f(u)_x + g(u)_y = 0, \;\; x, y \in \Omega \tag{1.18}$$

can be written as

$$U_{i,j}^{n+1} = U_{i,j}^n - \frac{\Delta t}{\Delta x}(F_{i+1/2,j}^n - F_{i-1/2,j}^n) - \frac{\Delta t}{\Delta y}(G_{i,j+1/2}^n - G_{i,j-1/2}^n), \tag{1.19}$$

where $F_{i-1/2}^n$ is an approximation to the average flux along $x = x_{i-1/2}$

$$F_{i-1/2,j}^n \approx \frac{1}{\Delta t \Delta y} \int_{t_n}^{t_{n+1}} \int_{y_{j-1/2}}^{y_{j+1/2}} f(u_{i-1/2})\, dy dt. \tag{1.20}$$

where $u_{i-1/2} = u(x_{i-1/2}, y, t)$ and $G_{j-1/2}^n$ is an approximate flux along $y = y_{j-1/2}$

$$G_{i,j-1/2}^n \approx \frac{1}{\Delta t \Delta x} \int_{t_n}^{t_{n+1}} \int_{x_{i-1/2}}^{x_{i+1/2}} g(u_{j-1/2})\, dx dt \tag{1.21}$$

where $u_{j-1/2} = u(x, y_{j-1/2}, t)$.

The generalization of the Theorem 1.2.1 to multiple dimensions is straightforward: the above conditions must hold separately for all components of the flux.

In the subsequent sections, we investigate the consistency and convergence of FV methods and consider second-order FVMs.

1.3 Stability of one-dimensional numerical schemes

As it was mentioned in the previous section, weak solution is not strict enough to determine a unique solution to a scalar conservation law. Weak solutions must satisfy additional restrictions. We say that a weak solution u is admissible if and only if any curve of discontinuity for u is a shock curve.

Let u be a weak solution of (1.6)-(1.7), and let u have a discontinuity at $x = \gamma(t)$, but u is smooth on the rest of the real line.

Assume that u^- is the limit of u at $\gamma(t)$, approaching from the left, i.e. $u^-(t) = \lim_{\varepsilon \to 0} u(\gamma(t) - \varepsilon, t)$ and let u^+ be the limit of u at $\gamma(t)$ approaching from the right, i.e $u^+(t) = \lim_{\varepsilon \to 0}(\gamma(t) + \varepsilon, t)$. Let $f(u)$ in (1.6) be strictly convex, then it follows from (1.9) that

$$\int_0^T \int_{-\infty}^{\infty} \varphi_t u + \varphi f'(u) u_x dx dt + \int_{-\infty}^{\infty} \varphi(x,0) u(x,0) dx = 0. \tag{1.22}$$

We denote

$$\Omega^- = \{(x,t) \ : \ 0 < t < T, \ -\infty < x < \gamma(t)\},$$
$$\Omega^+ = \{(x,t) \ : \ 0 < t < T, \ \gamma(t) < x < \infty\},$$

and rewrite (1.22) as

$$\int\int_{\Omega^-} \varphi_t u + \varphi f'(u) u_x dx dt + \int\int_{\Omega^-} \varphi_t u + \varphi f'(u) u_x dx dt + \int_{-\infty}^{\infty} \varphi(x,0) u(x,0) dx = 0.$$

Choosing $\varphi(x,0) = 0$, we apply the divergence theorem to get

$$\int\int_{\Omega^-} \varphi_t u + \varphi f'(u) u_x dx dt = -\int\int_{\Omega^-} (u_t + f(u)_x)\varphi dx dt + \int_{x=\gamma(t)} u^- \varphi \nu_2 + f(u^-)\varphi \nu_1 ds = 0,$$

and

$$\int\int_{\Omega^+} \varphi_t u + \varphi f'(u) u_x dx dt = -\int\int_{\Omega^+} (u_t + f(u)_x)\varphi dx dt - \int_{x=\gamma(t)} u^+ \varphi \nu_2 + f(u^+)\varphi \nu_1 ds = 0,$$

where (ν_1, ν_2) is the outward normal to Ω^-.

By assumption, u is a weak solution and since u is smooth on any interval containing $x = \gamma(t)$, then u

$$\int\int_{\Omega^-} \varphi_t u + \varphi f'(u) u_x dx dt = \int\int_{\Omega^+} \varphi_t u + \varphi f'(u) u_x dx dt = 0.$$

Combining this with the expressions above yields

$$\int_{x=\gamma(t)} u^- \varphi \nu_2 + f(u^-)\varphi \nu_1 ds = \int_{x=\gamma(t)} u^+ \varphi \nu_2 + f(u^+)\varphi \nu_1 ds = 0.$$

It follows that

$$u^- \nu_2 + f(u^-)\nu_1 = u^+ \nu_2 + f(u^+)\nu_1,$$

which implies

$$\frac{f(u^-) - f(u^+)}{u^- - u^+} = -\frac{\nu_2}{\nu_1}.$$

The slope at the discontinuity is given by

$$\frac{dt}{dx} = \frac{1}{\gamma'(t)} = -\frac{\nu_1}{\nu_2}.$$

Hence $\gamma(t)$ must satisfy

$$\frac{1}{\gamma'(t)} = \frac{f(u^-) - f(u^+)}{u^- - u^+},$$

which is called the Rankine-Hugoniot jump condition.

Now, we use the uniform convexity of the flux. In particular, this means if $f'(u)$ is strictly increasing, then u will satisfy the entropy condition (1.11) if and only if $u^- > u^+$ at any discontinuity. Therefore, for uniformly convex flux, u is an admissible weak solution to (1.6)-(1.7) if and only if u satisfies the Rankine-Hugoniot conditions and $u^- > u^+$ along any curves of discontinuity.

Finally, the following entropy conditions can be derived

$$f'(u^+) < \gamma'(t) < f'(u^-), \tag{1.23}$$

where $s = \gamma'(t)$ is the speed of propagation of the discontinuity. The inequality (1.23) is known as Lax entropy condition and it is necessary for the stability of the solution in the linear case.

For the numerical method for (1.6) to be convergent, the numerical solution should converge to exact solution of the differential equation as the grid is refined. For linear methods to be convergent we require the method to be consistent with the differential equation and stable.

We call a FV scheme given by an operator $F : \mathbb{R}^N \to \mathbb{R}^N$ stable on $(0, T]$ if there exists a function $\tau : \mathbb{R}^+ \to \mathbb{R}^+$, which we refer to as the maximal time step for a given mesh size Δx, such that:

$$\forall \Delta x > 0,\ \Delta t \in (0, \tau(\Delta x)],\ \forall n,\ 0 \leq n\Delta t \leq T.$$

The following inequality holds:

$$\|(Id + \Delta t F)^n\| < K, \tag{1.24}$$

where Id is an identity operator and K is a certain constant that depends on the norm of the operator F only. The inequality

$$\Delta t < \tau(\Delta x) \tag{1.25}$$

is referred to as the Courant-Friedrich-Lewy (CFL) stability condition.

The numerical solutions of high-order numerical schemes may produce spurious oscillations near discontinuities. This leads to nonlinear instabilities and unbounded (numerical) solutions. Physically, this may produce nonphysical solutions such as negative pressures or temperatures. This can lead to physical instabilities and hence non-feasible solutions, such as the well-known Gibbs phenomenon [74].

For nonlinear equations stability and consistency of the numerical method do not guarantee convergence. Instead, a number of different approaches have been developed to ensure convergence. One convergent set of schemes is the schemes that maintain a monotonic solution or monotone scheme. We say that a scheme is monotone if $F(U)$ in (3.2) is a monotone nondecreasing function of each of its arguments. These schemes are typically limited to first-order accuracy, requiring fine meshes for accurate computations over extended periods of time [65]. The characteristic property of such schemes is that they represent the solution with an excessive amount of dissipation. The goal of the construction of a high-order method is to produce a numerical solution with neither excessive diffusion nor non-physical oscillations. Several techniques or combinations of several techniques are known to achieve that.

The standard approach is to suppress oscillations by adding artificial viscosity or diffusion terms into the equation (1.6). These terms are added to dampen the growth of spurious oscillations and mimic the effects of viscosity. Introducing artificial viscosity into (1.6) results in

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

where $\sigma > 0$ is a small number that represents the artificial viscosity coefficient.

We conjecture that the entropy-admissible solutions of (1.6), can be obtained as a limit

of solutions of the parabolic equation (1.3). This approach aids in stabilizing the numerical solution at the cost of severely affecting the rates of convergence and requires careful tuning of the viscosity parameter to balance stability and accuracy [90, 137]. A common problem with the the application of artificial viscosity to the solution of nonlinear problems is that they are either not robust or not accurate enough, or both.

A more popular slope-limiting technique is a nonlinear procedure, that constructs a numerical scheme that may exhibit both high-order accuracy and monotonic behavior simultaneously. Most common limiting methods (limiters) compare slopes or curvatures of the solution on each element of the grid (pointwise or cell-wise) with the neighboring elements [130, 68, 122, 37]. To avoid the occurrence of spurious solution behavior the limiter would check a certain relation between the function values or derivatives. Once the measured relation for a particular element exceeds a specified threshold, the limiter would modify the value of the solution locally to get the relation below the threshold value. In other words, limiters identify oscillations and suppress them by changing the solution at a point and/or neighboring values. For example, the limiters for FV methods reconstruct the slope without changing the average value in the given element.

There is a common problem for most limiters in use. Different limiters may often incorrectly select an element or a whole region near a smooth extrema of the numerical solution as an element or a region that requires limiting [69]. This leads to the reduction of the solution accuracy, ultimately reducing the convergence rate of the method. Since most limiting techniques commonly control the solution gradient within an element, it is a logical choice for second-order numerical methods, particularly in the context of FV methods. The resulting schemes exhibit second-order accuracy, as assessed through truncation error analysis, and genuinely demonstrate nonlinearity, see [68] and numerical experiments mentioned therein.

A special class of limiters, that is particularly efficient is based on the total variation diminishing (TVD) property of the solutions of (1.6). In the framework of numerical schemes, a discrete definition for TV has been successfully used to develop high-order methods in one-dimension [122, 66, 67, 123]. The TVD property, when imposed on the numerical solution, prevents the creation of spurious oscillations. The limiters are designed to ensure the solution lies within the Sweby region, a specific range of solution slopes that guarantees non-increasing total variation. More specifically, this region is characterized as an intersection of the TVD region and the high-order scheme accuracy region.

The set of TVD schemes contains monotone schemes, as detailed in [66]. However, in contrast to monotone schemes, a TVD scheme does not automatically have consistency with the entropy inequality (1.11). An extension of the idea of TVD limiting is total variation bounded (TVB) methods. These methods allow for a controlled increase in the TV of the numerical solution. This approach retains the concept of total variation diminishing (TVD) schemes by introducing a weaker criterion for limiting to ensure the reduction of numerical oscillations [115]. Other notable contributions to this area include [38, 69, 104]. These studies have demonstrated the effectiveness of TVB methods in maintaining stability. TVB methods into numerical schemes represent a significant advancement in the pursuit of accurate and stable high-order methods, providing a valuable tool for various applications in computational mathematics and fluid dynamics.

1.4 Stability in multiple spatial dimensions

In one dimension, TVD schemes, frequently used as slope limiters to ensure a non-increasing total variation of the solution over time, have been used to a great extent. While in two dimensions, enforcing a TVD property can lead to schemes with at most first-order accuracy [61]. In absence of a solid theoretical basis for the TVD property in high dimensions, most practical limiters rely on geometrical arguments or directly extend one-dimensional concepts, such as enforcing that solution values at specific points don't surpass the average within a chosen neighborhood [6, 83].

The lack of high-order limiters may stem from limitations in analytical tools. While Harten's TVD theory led to powerful second-order limiters in one dimension, it reduces to first-order accuracy near smooth extrema [104, 105]. Alternative constructions exist for piecewise parabolic solutions in one dimension that are TVD in a different sense, considering the total variation of the entire function including jumps between cells [68, 97].

Alternatively, enforcing a local maximum principle (LMP) on the numerical solution offers a weaker yet more productive approach to the development of limiters for second-order methods [54, 56]. LMP-based limiting is particularly beneficial for scalar problems, where the LMP allows to ensure bounded solution averages throughout the computation, which enhances the solution's stability. However, we know of no result on the sufficiency of LMP for scheme stability in two dimensions for a general scalar conservation law.

The main principle here is to ensure that the maximum value within an element of the mesh remains bounded by the maximum values in neighboring elements, preventing overshoots and maintaining solution stability. A popular LMP-based limiter involves reconstructing the solution as the sum of the cell mean and slope, followed by scaling the slope to limit the solution within a predetermined interval [98]. This approach, while effective, can be computationally expensive for complex meshes.

Unlike one dimension, higher dimensions lack unique directions for limiting gradients and introduce additional complexity due to mixed derivatives. As a result, limiting in multiple dimensions is significantly more challenging. For this purpose, several approaches were developed, for example, directional derivative limiters and moment limiters [54, 56]. These techniques aim to limit partial derivatives along specific directions to ensure the solution falls within a locally defined interval. While effective on structured meshes, these approaches may not be easily adaptable to unstructured meshes.

FV limiters have been successfully adapted for the construction of stable and high-order Discontinuous Galerkin (DG) methods. Additionally, DG-specific limiters have been developed [11, 83, 2]. Examples include WENO reconstruction [139, 118], hierarchical limiting [136, 84], and limiting along medians [19]. However, existing limiters for the DG method often suffer from limitations like computational cost, restrictiveness, or lack of robustness, requiring problem-dependent tuning parameters. Modern areas of study focus on tailoring limiters to specific problem characteristics or leveraging problem structure could potentially improve efficiency and robustness.

In summary, the literature showcases various strategies employed to stabilize numerical solutions to hyperbolic PDEs in multiple dimensions, ranging from artificial viscosity and limiting methods to shock-capturing schemes. The choice of approach often depends on the specific characteristics of the problem at hand, reflecting the ongoing pursuit of robust and efficient numerical methodologies in the face of high-dimensional challenges.

1.5 Outline

In this work, we consider the following problem for two-dimensional hyperbolic scalar conservation laws

$$u_t + f(u)_x + g(u)_y = 0, \ (x, y) \in \Omega, \ t > 0, \tag{1.26}$$
$$u(x, 0) = u_0, \tag{1.27}$$

and appropriate boundary conditions. We denote by $f(u)$ and $g(u)$ the flux components in the x- and y-directions, respectively. Assuming $f(u)$ and $g(u)$ depend on the function u only, we study TV of the numerical solutions to (1.26) under several discrete TV definitions. The main goal of the present work is to challenge the long-standing negative result of J. Goodman and R. LeVeque [61] and to demonstrate that the limitation on the order of accuracy of the TVD schemes in multiple dimensions can be overcome by changing the discrete TV definition to a more accurate one, more suitable for measuring TV of $u \in L^1(\Omega)$ solutions of (1.26)-(1.27).

For this purpose, we adopt a relatively recent alternative discrete TV definition, proposed in [40] in the context of image processing, more precisely for image denoising, and restoration. We study its properties in Chapter 2 and discuss dual discrete TV stability for a general two-dimensional scheme in Chapter 3. We suggest a set of limiting conditions on a five-point finite volume scheme coefficients, to guarantee non-increasing TV of the scheme in the sense of the new dual discrete TV. We provide numerical evidence to support the claim, including a consistent second-order scheme for scalar conservation laws and randomly generated schemes in two dimensions.

The applications of the dual definition are not limited to the study of the stability properties of the numerical schemes for scalar conservation laws. Its use is complicated by the fact that dual TV discretization requires solving the associated optimization problem. We give details on how it can efficiently be done in Chapter 4. Then, we extend its use to formulate a modified version of the projection onto convex sets (POCS) imaging algorithm for computed tomography (CT) image reconstruction in Chapter 5. This allows us to enhance and refine state-of-the-art POCS algorithms for sparse-view, low-dose, and limited-angle CT applications. We show that the use of the dual discrete TV allows us to surpass the limitations associated with conventional TV discretizations, suppress artifacts, and improve the quality of reconstruction. The proposed DTV-ASD-POCS imaging algorithm has the potential to contribute significantly to the advancement of CT scan reconstruction, offering a new promising approach to enhancing the overall effectiveness of medical imaging techniques. We provide conclusions in Chapter 6.

The organization of this manuscript is as follows Chapter 2: Total variation and its discretizations; Chapter 3: Total variation stability of numerical methods for scalar conservation laws; Chapter 4: A primal-dual algorithm for computing dual discrete total variation; Chapter 5: Applications to image reconstruction; Chapter 6: Conclusions.

Chapter 2

Total variation and its discretizations

2.1 Total variation in one spatial dimension

We begin by introducing a definition of the total variation of a function $u = u(x)$.

Definition 2.1.1. *Let u be a real-valued function defined on an open interval $I \subset \mathbb{R}$. Its total variation $TV(u)$ is defined as*

$$TV(u) = \sup \sum_i |u(x_i) - u(x_{i-1})|, \tag{2.1}$$

where the supremum is taken over all partitions $\{x_1 < x_2 < \cdots < x_N\}$ of I. We define the space $BV(I)$ as a space of functions on I with finite total variation. If $TV(u) < \infty$, we say that u is of bounded variation on I and write $u \in BV(I)$.

If $u \in BV(I)$, then for any $\varepsilon > 0$ we have

$$\frac{1}{\varepsilon} \int_I |u(x+\varepsilon) - u(x)| dx \leq TV(u).$$

Then, it can be shown that

$$TV(u) = \limsup_{\varepsilon \to 0} \left\{ \frac{1}{\varepsilon} \int_I |u(x+\varepsilon) - u(x)| dx \right\}. \tag{2.2}$$

While (2.1) and (2.2) hold for any function u on I, for a differentiable u we have a simpler expression

$$TV(u) = \int_I |u'(x)| \, dx. \tag{2.3}$$

For a non-differentiable u, we define Du, a distributional (or weak) derivative. We call Du a weak derivative of $u \in L^1(I)$ if the following holds for all $\varphi \in C_c^1(I)$

$$\int_I u\varphi' \, dx = -\int_I Du \, \varphi \, dx.$$

If $u \in BV(I)$, then $TV(u)$ can be written as

$$TV(u) = \sup_{\varphi \in C_c^1(I),\ |\varphi| \leq 1} \left\{ -\int_I u \, \varphi' dx \right\} = \int_I |Du| dx. \tag{2.4}$$

Figure 2.1 shows three functions: u,v, and w. The first two are not differentiable at the middle points. It is clear from (2.1) that these three functions have the same TV, i.e. $TV(u) = TV(v) = TV(w) = 2$. For one-dimensional BV functions, the geometric representation of their TV is the sum of distances between consecutive minimum and maximum values.

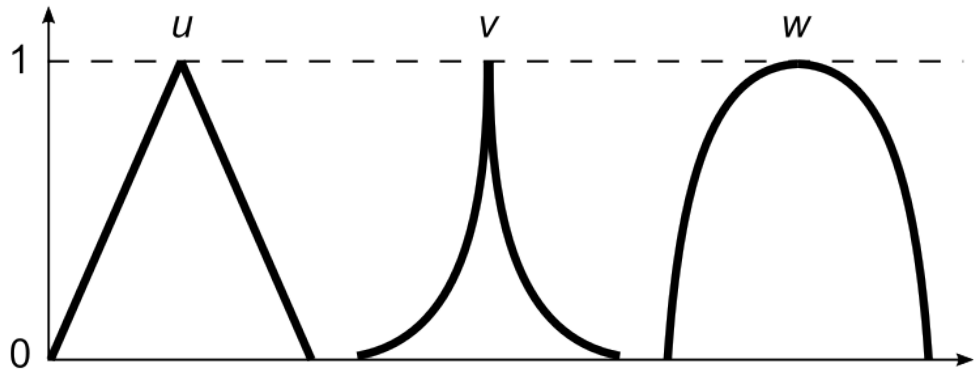


Figure 2.1: Total variation of functions u, v, w are $TV(u) = TV(v) = TV(w) = 2$.

Total variation is widely used in analysis of solutions of one-dimensional conservation laws. Below we list some relevant results and provide references. In this and next chapters we use citation after the number of the theorem to distinguish between the known results and the new results proven here.

Theorem 2.1.1 ([18] Helly's theorem)**.** *Let $\{u_n\}$ be a sequence of functions $u_n : \mathbb{R} \to \mathbb{R}$ such that*

$$TV(u_n) \ \leq C \ \text{and} \ \ |u_n(x)| \leq M, \quad \forall x \in \mathbb{R}, \ n = 1, 2, \ldots,$$

with some constants C and M. Then there exists a function u and a subsequence $\{u_{n_k}\}$ such that

$$\lim_{n_k \to \infty} u_{n_k}(x) = u(x), \quad \text{for each fixed } x \in \mathbb{R},$$

$$TV(u) \leq C, \ |u(x)| \leq M.$$

Theorem 2.1.1 implies that BV is a compact space. Unfortunately, the convergence here is pointwise and not uniform.

Theorem 2.1.2 ([18])**.** *Let $\{u_n\}$ be a sequence of functions $u_n : \mathbb{R} \times [0, \infty) \to \mathbb{R}$ such that*

$$TV(u_n) < C \ \text{and} \ \ |u_n(x,t)| \leq M$$

$$\int_{-\infty}^{\infty} |u_n(x,t) - u_n(x,s)| dx \leq L|t-s|, \quad \forall t, s \geq 0,$$

with some constants C, M, L. Then, there exists a subsequence $\{u_{n_k}\}$ that converges in L^1 to a locally L^1-integrable function u, such that

$$\int_{-\infty}^{\infty} |u(x,t) - u(x,s)| dx \leq L|t-s|, \quad \forall t, s \geq 0,$$

$$TV(u) \leq C, \ |u(x,t)| \leq M.$$

This theorem establishes that we can construct an approximation to $u(x,t)$ at each moment in time $t \geq 0$ using bounded functions from $BV(\mathbb{R})$, and it will hold at all but countably many points x. Finally, we state the existence of BV solutions of scalar conservation laws.

Theorem 2.1.3 ([60]). *Let $f(u)$ be locally Lipschitz continuous and let $u_0 \in BV(\mathbb{R})$. he following theorem establishes the BV property of the unique entropy solution. Then, the Cauchy problem (1.6)-(1.7) has a unique weak entropy solution $u \in L^\infty(\mathbb{R}^N \times (0,T))$, which satisfies for almost all $t \in [0,T]$*

$$\|u(\cdot,t)\|_{L^\infty} \leq \|u_0\|_{L^\infty} \text{ a.e. for } x \in \mathbb{R}^N. \tag{2.5}$$

Moreover, if u and v are the entropy solution of (1.6) associated with initial conditions u_0 and v_0, respectively, such that $u_0 \geq v_0$, then we have

$$u(\cdot,t) \geq v(\cdot,t) \quad a.e. \tag{2.6}$$

Finally, if u_0 belongs to $L^\infty(\mathbb{R}^N) \cap BV(\mathbb{R}^N)$, then $u(\cdot,t)$ belongs to $BV(\mathbb{R}^N)$ with:

$$TV(u(\cdot,t)) \leq TV(u_0). \tag{2.7}$$

Theorem 2.1.3 states that the total variation of weak solutions of (1.6)-(1.7) is a bounded function.

2.2 Total variation in multiple spatial dimensions

TV definition (2.2) can be naturally extended to multiple spatial dimensions.

Definition 2.2.1. *Let $u : \Omega \to \mathbb{R}$, where $\Omega \subset \mathbb{R}^2$ is an open set. Then its TV can be defined as*

$$TV(u) = \limsup_{\varepsilon \to 0} \left\{ \frac{1}{\varepsilon} \int_\Omega |u(x+\varepsilon, y) - u(x,y)| + |u(x, y+\varepsilon) - u(x,y)| \, dxdy \right\}. \tag{2.8}$$

For a differentiable u, (2.8) can be rewritten as

$$TV(u) = \int_\Omega |u_x| + |u_y| dxdy. \tag{2.9}$$

While the definition (2.8)-(2.9) is straightforward, it is not rotation invariant. For functions in $L^1(\Omega)$, we can introduce a different definition. Similarly to the one-dimensional case (2.4), we introduce the weak derivative Du as

$$\int_\Omega u \nabla \cdot \boldsymbol{\varphi} \, dxdy = -\int_\Omega Du \, \boldsymbol{\varphi} \, dxdy, \quad \boldsymbol{\varphi} \in C_c^1(\Omega). \tag{2.10}$$

where $\boldsymbol{\varphi} = (\varphi, \psi)$ is a differentiable test function with compact support. Then we can define TV as

Definition 2.2.2.

$$TV(u) = \sup_{\boldsymbol{\varphi} \in C_c^1(\Omega)} \left\{ -\int_\Omega u \nabla \cdot \boldsymbol{\varphi} \; dxdy : \|\boldsymbol{\varphi}\|_\infty \leq 1 \right\} = \int_\Omega |Du| \; dxdy, \tag{2.11}$$

where, the norm $\|\cdot\|_\infty$ is defined by

$$\|\boldsymbol{\varphi}\|_\infty = \sup_{(x,y)\in\Omega} \|\boldsymbol{\varphi}(x,y)\|_2 = \sup_{(x,y)\in\Omega} \sqrt{\varphi(x,y)^2 + \psi(x,y)^2}. \tag{2.12}$$

This is known as TV definition in weak sense. If we again assume that u is differentiable, (2.11) becomes

$$TV(u) = \int_\Omega \sqrt{u_x^2 + u_y^2} \; dxdy. \tag{2.13}$$

TV (2.13) is commonly referred to as isotropic TV. We indicate this by the subscript "is". The name is due to the fact that the Euclidean norm used in (2.13) is invariant under rotation of the coordinate system. For $TV_a(u)$ we can easily see that it is not rotation invariant. For instance, a vector $(u_x, u_y) = (0, 1)$ will have $|u_x| + |u_y| = 1$, where $\|\cdot\|_1$ stands for the L^1-norm. Then, if we rotate it by $\pi/4$ we will get $|u_x| + |u_y| = \sqrt{2}$. In the following sections we will see how this can cause imaging artifacts and significant errors in the value of TV.

In two and more spatial dimensions, analogues of Theorems 2.1.1 and 2.1.2 hold for (2.11). We will now state some properties of total variation and the space of functions of bounded variation that are important for the discussion in Chapter 3.

Theorem 2.2.1 (Convexity of TV [29])**.** *Let $u_1, u_2 \in L^1(\Omega)$ and $\alpha \in [0, 1]$, then*

$$TV(\alpha u_1 + (1-\alpha)u_2) \leq \alpha TV(u_1) + (1-\alpha)TV(u_2).$$

The proof follows from (2.11) and properties of supremum.

Theorem 2.2.2 (L^1-lower semicontinuity [29])**.** *Suppose $u_n \to u$ in L^1. Then*

$$TV(u) = \int_\Omega |Du| \, dx \, dy \leq \liminf_{n\to\infty} \int_\Omega |Du_n| \, dx \, dy.$$

In particular, if $\{u_n\}$ is a sequence of bounded functions such that $u_n \in BV(\Omega)$, then $u \in BV(\Omega)$, $\forall n$.

For any $\boldsymbol{\varphi} \in C_c^1(\Omega)$ that satisfies $\|\boldsymbol{\varphi}\|_\infty \leq 1$ we have by the assumption the theorem and (2.11)

$$\int_\Omega u \nabla \cdot \boldsymbol{\varphi} \, dx \, dy = \lim_{n\to\infty} \int_\Omega u_n \nabla \cdot \boldsymbol{\varphi} \, dx \, dy \leq \liminf_{n\to\infty} \int_\Omega |Du_n| \, dx \, dy. \tag{2.14}$$

Taking supremum over $\boldsymbol{\varphi}$ of (2.14), yield

$$\begin{aligned} \sup_{\boldsymbol{\varphi}\in C_c^1(\Omega),\ \|\boldsymbol{\varphi}\|_\infty \leq 1} \left\{ \int_\Omega u \nabla \cdot \boldsymbol{\varphi} \, dx \, dy \right\} &= \sup_{\boldsymbol{\varphi}\in C_c^1(\Omega),\ \|\boldsymbol{\varphi}\|_\infty \leq 1} \left\{ -\int_\Omega u \nabla \cdot \boldsymbol{\varphi} \, dx \, dy \right\} \\ &\leq \liminf_{n\to\infty} \int_\Omega |Du_n| \, dx \, dy. \end{aligned}$$

Noticing that the middle expression is equal to $TV(u)$ concludes the proof.

Next we show that $BV(\Omega)$ equipped with a norm

$$\|u\|_{BV} = \int_\Omega |u(x,y)|\,dx\,dy + \int_\Omega |Du|\,dx\,dy, \tag{2.15}$$

is a Banach space. It follows from (2.15) that $BV(\Omega)$ is a subset of $L^1(\Omega)$. To establish completeness, we consider a bounded sequence $\{u_n\}$ that is Cauchy under $\|\cdot\|_{BV}$. Then $\{u_n\}$ is also Cauchy in L^1. Let $u \in L^1$ be its limit. Then, by the semicontinuity property, $u \in BV(\Omega)$.

Next, we apply the semicontinuity argument again to $\{u_m - u_n\}$, where $\{u_m\}$ is a subsequence of $\{u_n\}$. For each u_m we can write

$$\int_\Omega |Du_m - Du|\,dx\,dy \le \liminf_{n\to\infty} \int_\Omega |Du_m - Du_n|\,dx\,dy.$$

Since $\{u_n\}$ is a Cauchy sequence, we have that

$$\liminf_{n\to\infty} \int_\Omega |Du_m - Du_n|\,dx\,dy = 0.$$

Thus $u_m \to u$ in $BV(\Omega)$.

Another key property of the BV space is its closeness to $W^{1,1}(\Omega)$.

Theorem 2.2.3 (Mollification of BV [29])**.** *For any $u \in BV(\Omega)$, we can find a sequence of approximations $\{u_n\}$ such that:*

(a) $u_n \in C^\infty(\Omega)$ for $n = 1, 2, \ldots$

(b) $u_n \to u$ in $L^1(\Omega)$ as $n \to \infty$,

(c) $\int_\Omega |Du_n|\,dx\,dy \to \int_\Omega |Du|\,dx\,dy$.

Theorem 2.2.4 (Weak Compactness of BV [29])**.** *Let $\{u_n\} \in BV(\Omega)$, where Ω is a Lipschitz domain, be a bounded sequence. Then, there exists a subsequence that converges in $L^1(\Omega)$.*

It follows from Theorem 2.2.3, that for each u_n there exists $w_n \in W^{1,1}(\Omega)$ approximating it such that

$$\int_\Omega |u_n - w_n|\,dx\,dy \le \frac{1}{n} \quad \text{and} \quad \int_\Omega |\nabla w_n|\,dx\,dy \le \int_\Omega |Du_n| + 1\; dxdy.$$

Therefore the sequence $\{w_n\}$ must be bounded in $W^{1,1}(\Omega)$. Due to the weak compactness of $W^{1,1}(\Omega)$ (or the Rellich theorem [112, 45]), $\{w_n\}$ will contain a subsequence $\{w_{n_k}\}$, that will converge in $L^1(\Omega)$. Then the subsequence of $\{u_n\}$ with the same indices n_k, i.e. $\{u_{n_k}\}$ must also converge in $L^1(\Omega)$.

Finally, we state an important result for application of TV to solutions of hyperbolic conservation laws. Let us consider a scalar conservation law in N-dimensional space

$$u_t + \nabla \cdot f(u, \boldsymbol{x}, t) = q(u, \boldsymbol{x}, t), \quad (\boldsymbol{x}, t) \in \mathbb{R}^N \times [0, \infty), \tag{2.16}$$

$$u(\boldsymbol{x}, 0) = u_0(\boldsymbol{x}), \tag{2.17}$$

where f and q are Lipshitz continuous flux and source term, respectively. This Cauchy problem has a unique weak entropy solution [79].

Let $\nabla \cdot f = \partial_u f \cdot \nabla u + \nabla_{\boldsymbol{x}} \cdot f$, where $\nabla_{\boldsymbol{x}} f$ denotes the gradient of f with respect to the spatial variables only. Assume that $\partial_t f_u, \partial_t(\nabla_{\boldsymbol{x}} \cdot f), f_u, \nabla_{\boldsymbol{x}} \cdot f_u, \partial_t q, q - \nabla_{\boldsymbol{x}} \cdot f$, $\partial_u(q - \nabla_{\boldsymbol{x}} \cdot f), (q - \nabla_{\boldsymbol{x}} \cdot f) \in L^\infty(\mathbb{R}^N)$, $f \in C^2(\mathbb{R}^N)$, $q \in C^1(\mathbb{R}^N)$ and

$$\int_0^T \int_{\mathbb{R}^N} \|\nabla_{\boldsymbol{x}}(q - \nabla_{\boldsymbol{x}} \cdot f)\|_\infty d\boldsymbol{x} dt < \infty, \quad \forall T > 0.$$

Then the following theorem holds.

Theorem 2.2.5 (Theorem 2.5 of [39]). *Let $u_0 \in BV(\mathbb{R}^N)$ be bounded. Then, the weak entropy solution u of (2.16)-(2.17) is of bounded variation, i.e. $u \in BV(\mathbb{R}^N)$ for all $t > 0$.*

Moreover, if

$$\kappa_0 = N \cdot W_N \left((2N+1)\|\nabla_{\boldsymbol{x}} \partial_u f\|_\infty + \|q_u\|_\infty\right)$$

with W_N given by

$$W_N = \int_0^{\pi/2} (\cos(\theta))^N \, d\theta,$$

then for all $T > 0$,

$$TV(u(\boldsymbol{x}, T)) \leq TV(u_0) e^{\kappa_0 T} + N \cdot W_N \int_0^T e^{\kappa_0 (T-t)} \int_{\mathbb{R}^N} \|\nabla_{\boldsymbol{x}}(q - \nabla_{\boldsymbol{x}} \cdot f)\|_\infty d\boldsymbol{x} \, dt. \quad (2.18)$$

In this work we consider problems with $f = f(u)$, i.e. with the flux that does not depend explicitly on $\boldsymbol{x}, t$, and the source term $q = 0$. In this case $\kappa_0 = 0$, and the integral in right hand side of (2.18) is zero. Then we get the following bound for the total variation of the solution: $TV(u(\boldsymbol{x}, T)) \leq TV(u_0)$.

Geometric interpretation of TV.

While total variation has an easy geometric interpretation for functions of one variable, it is more challenging to visualize it in higher dimensions. In particular, (2.11) does not allow an easy interpretation. Below we will try to get some insight into this issue.

First, we will analyze a simple case of a smooth function $u(x, y)$ and then state a general result.

Let $u(x, y)$, $(x, y) \in \Omega$, describe a smooth surface in three-dimensional space and let $u(x, y) = \lambda$ be a level curve for some $\lambda \in \mathbb{R}$. Assume that some point (x_0, y_0) is lying on the curve and $\nabla u(x_0, y_0) \neq \mathbf{0}$. Then the level curve $(x, y(x, \lambda))$ can be parameterized as

$$x = x(s, \lambda) \quad \text{and} \quad y = y(s, \lambda),$$

where s is the arc length. In a local neighborhood of (x_0, y_0), we can relate

$$u(x, y) = \lambda \iff y = y(x, \lambda).$$

This change of variables is well-defined at least locally, and the following relations hold

$$\lambda = u(x(s, \lambda), y(s, \lambda)) \quad \text{and} \quad x_s^2 + y_s^2 = 1. \quad (2.19)$$

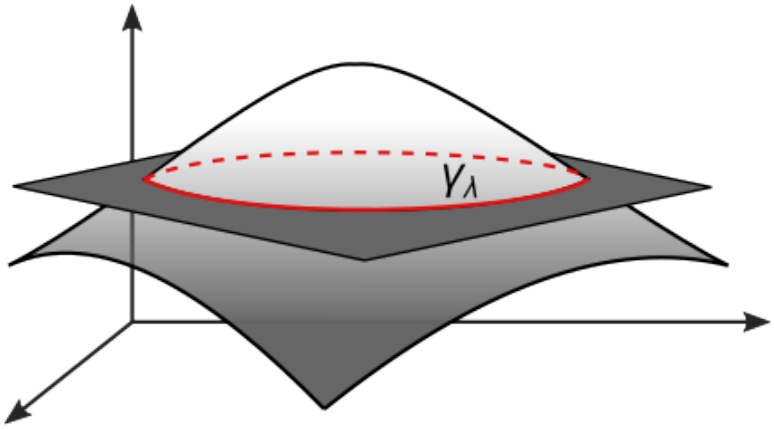


Figure 2.2: Level set γ_λ (in red) for a smooth function of two variables.

Computing partial derivatives of the first expression with respect to s and λ yields

$$0 = (x_s, y_s) \cdot \nabla u \quad \text{and} \quad 1 = (x_\lambda, y_\lambda) \cdot \nabla u. \tag{2.20}$$

The first expression in (2.20) results in $\nabla u = \pm|\nabla u|(-y_s, x_s)$. Then we can rewrite the second expression in (2.20) as

$$1 = \pm|\nabla u|(x_s y_\lambda - y_s x_\lambda) = \pm|\nabla u|\frac{\partial(x,y)}{\partial(s,\lambda)}. \tag{2.21}$$

Then

$$\int_\Omega |\nabla u|\, dx\, dy = \int_{\Omega_{s,\lambda}} |\nabla u| \left|\frac{\partial(x,y)}{\partial(s,\lambda)}\right| ds\, d\lambda = \int_{\Omega_{s,\lambda}} ds\, d\lambda.$$

Above, we changed variables in the first integral. In the second integral the integrand is equal to one by (2.21).

Let γ_λ denote the λ-level curve (Figure 2.2), then

$$\int_{\gamma_\lambda} ds = \text{length}(\gamma_\lambda).$$

Assuming that $\text{length}(\gamma_\lambda) = 0$ if the level curve is an empty set, we obtain

$$\int_\Omega |\nabla u|\, dx\, dy = \int_{-\infty}^{\infty} \text{length}(\gamma_\lambda)\, d\lambda. \tag{2.22}$$

Therefore, we can interpret total variation of a smooth shape as the sum of the lengths of all level curves.

For a general $u \in BV$ the level curve γ_λ might not be regular. So, for such nonsmooth functions formula (2.22) has to be adjusted. Let us define the level domain $E_\lambda = \{(x,y) \in \Omega \ : \ u < \lambda\}$ and its characteristic function

$$\iota_{E_\lambda} = \begin{cases} 1 & \text{if } \ (x,y) \in E_\lambda, \\ 0 & \text{otherwise.} \end{cases}$$

We further can define the perimeter of the level domain as

$$Per(E_\lambda) = \int_\Omega |D\iota_{E_\lambda}| \, dxdy,$$

where $D\iota_{E_\lambda}$ denotes the weak derivative of ι_{E_λ}. Then TV of a function can be computed by integrating the perimeter of all lower level domains.

Theorem 2.2.6 ([57]). *Suppose $u \in BV(\Omega)$, then*

$$TV(u) = \int_{-\infty}^{\infty} Per(E_\lambda) d\lambda. \tag{2.23}$$

The expression in (2.2.6) is known as the co-area formula. It was first proposed in [50] and then proved in [57].

Remark 2.2.6. For a smooth function, the expressions for the surface area

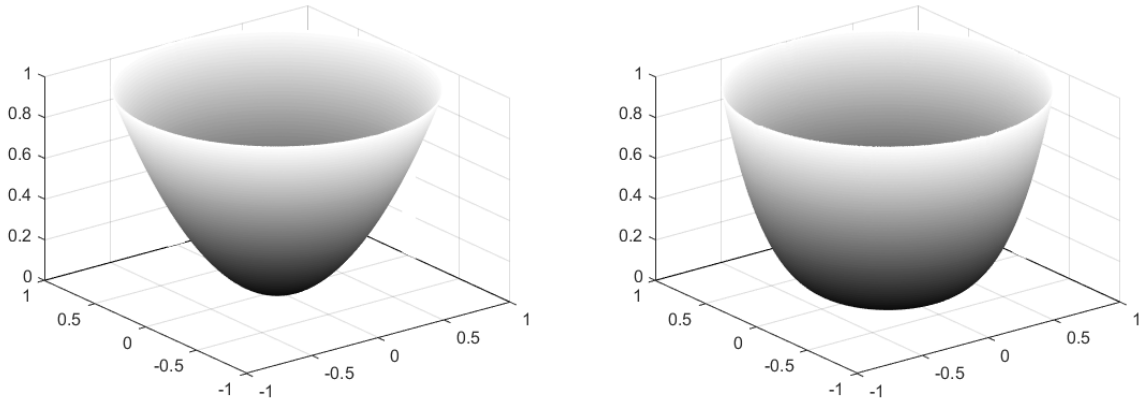
$$S(u) = \int_\Omega \sqrt{1 + u_x^2 + u_y^2} \, dxdy \quad \text{and} \quad TV(u) = \int_\Omega \sqrt{u_x^2 + u_y^2} \, dxdy.$$

are similar. In particular, for problems where we need to minimize surface area, any solution that minimizes $S(u)$ would give the minimal value of $TV(u)$. Also any oscillations in u would increase both the surface area and the value of total variation.

While in one spatial dimension TV depends on the values of the extrema and the shape of u is not relevant (Figure 2.1), this is not the case in two dimensions. Consider two smooth convex functions u and v defined on $\Omega = \{(x, y) \in \mathbb{R}^2 : x^2 + y^2 \leq 1\}$

$$u = x^2 + y^2, \quad v = (x^2 + y^2)^2,$$

see Figure 2.3. First we compute TV defined by (2.13). They are



(a) $u = x^2 + y^2$ (b) $v = (x^2 + y^2)^2$

Figure 2.3: Convex functions $u(x, y)$ and $v(x, y)$ defined inside the circle $x^2 + y^2 \leq 1$.

$$TV_{is}(u) = \int_\Omega \sqrt{4x^2 + 4y^2} \, dxdy = \int_0^1 \int_0^{2\pi} 2r^2 \, d\theta dr = \frac{4\pi r^3}{3}\Big|_0^1 = \frac{4\pi}{3}.$$

and

$$TV_{is}(v) = \int_\Omega \sqrt{(4x(x^2+y^2))^2 + (4y(x^2+y^2))^2}\; dxdy$$
$$= \int_0^1 \int_0^{2\pi} 4r^4 \sqrt{(\cos\theta)^2 + (\sin\theta)^2}\; d\theta dr = \frac{8\pi r^5}{5}\Big|_0^1 = \frac{8\pi}{5}.$$

We observe that $TV(u)$ and $TV(v)$ have different values. Similarly, for TV defined by (2.9) we have

$$TV_a(u) = \int_\Omega |u_x| + |u_y|\; dxdy = \int_{\Omega'} |2x| + |2y|\; dxdy$$
$$= \int_0^1 \int_0^{2\pi} 2r^2\,(|\cos(\theta)| + |\sin(\theta)|)\;\; d\theta dr = \frac{16}{3},$$

and

$$TV_a(v) = \int_\Omega |4x(x^2+y^2)| + |4y(x^2+y^2)|\; dxdy$$
$$= \int_0^1 \int_0^{2\pi} 4r^4\,(|\cos(\theta)| + |\sin(\theta)|)\;\; d\theta dr = \frac{32}{5},$$

which also have different values. We conclude here that TV in two dimensions and inherently its discretizations will be sensitive to the curvature of a function. For two smooth functions that have the same maximal and minimal values, TV can be different without introducing new extrema. This distinguishes the one-dimensional TV and TV in multiple dimensions and its discretizations.

Rotational and translational invariance of total variation.

Let u have a finite support in Ω. Consider geometric translation: $x'' = x + x', \quad y'' = y + y'$, with some constant x', y' such that $u(x'', y'')$ is fully contained in Ω after translation. Then

$$TV(u(x+x', y+y')) = \sup_{\boldsymbol{\varphi}\in C_c^1(\Omega)} \left\{ -\int_\Omega u(x+x', y+y')\nabla_{x,y}\cdot\boldsymbol{\varphi}\; dxdy : \|\boldsymbol{\varphi}\|_\infty \le 1 \right\}$$
$$= \sup_{\boldsymbol{\varphi}\in C_c^1(\Omega)} \left\{ -\int_\Omega u(x'', y'')\nabla_{x'',y''}\cdot\boldsymbol{\varphi}\; dx''dy'' : \|\boldsymbol{\varphi}\|_\infty \le 1 \right\} = TV(u),$$

since the divergence operator is invariant to translation of coordinates, i.e. $\nabla_{x,y}\cdot\boldsymbol{\varphi} = \nabla_{x'',y''}\cdot\boldsymbol{\varphi}$.

The rotational invariance of total variation follows from the co-area formula (Theorem 2.2.4) and the fact that perimeter $Per(E_\lambda)$ of any domain in two dimensions does not change under rotation.

2.3 Discrete total variation in multiple spatial dimensions

In this section we will introduce several conventional definitions of TV of discrete functions in one and two dimensions. Then we will investigate their properties and point out the

differences between them. Finally, we will describe an alternative approach to defining discrete TV and study its properties.

For a one-dimensional grid function U, which is a vector of size N, discrete TV is defined as

$$TV(U) = \sum_{i=1}^{N} |U_{i+1} - U_i|. \tag{2.24}$$

In two dimensions, we consider a grid function U defined on a $N \times N$ grid of cells $\Omega_{i,j} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$ with constant values $U_{i,j}$ in each cell. We associate $U_{i,j}$ with cell centers (x_i, y_j). First order forward difference discretization of (2.9) results in the following discrete TV

$$TV_a(U) = \sum_{i,j} \Delta x_i |U_{i+1,j} - U_{i,j}| + \Delta y_j |U_{i,j+1} - U_{i,j}|, \tag{2.25}$$

where the sum is taken over all cells in the grid. For uniform square grids we have $\Delta x_i = \Delta y_j = \Delta x$, and (2.25) can be written as

$$TV_a(U) = \Delta x \sum_{i,j} |U_{i+1,j} - U_{i,j}| + |U_{i,j+1} - U_{i,j}|. \tag{2.26}$$

The TV defined by (2.26) is referred to in the literature as "anisotropic TV", hence the subscript "a".

Discretization of (2.13) results in the following discrete TV

$$TV_{is}(U) = \sum_{i,j} \Delta x_i \Delta y_j \sqrt{\frac{1}{\Delta x_i^2}(U_{i+1,j} - U_{i,j})^2 + \frac{1}{\Delta y_j^2}(U_{i,j+1} - U_{i,j})^2}.$$

On a uniform square grid, $TV_{is}(U)$ simplifies to

$$TV_{is}(U) = \Delta x \sum_{i,j} \sqrt{(U_{i+1,j} - U_{i,j})^2 + (U_{i,j+1} - U_{i,j})^2}. \tag{2.27}$$

Several other discrete TV definitions have been proposed in the literature. For example, Vitali-Lebesgue-Fréchet-de la Vallée Poussin TV [33]

$$TV_V(U) = \sum_{i,j} |\Delta_{11} U_{i,j}|, \tag{2.28}$$

and Fréchet TV [33]

$$TV_F(U) = \max_{\epsilon_i, \tilde{\epsilon}_j = \pm 1} \sum_{i,j} \epsilon_i \tilde{\epsilon}_j \Delta_{11} U_{i,j}, \tag{2.29}$$

where the maximum is taken over all vectors $\epsilon, \tilde{\epsilon} \in \mathbb{R}^{N-1}$ with components equal to ± 1 and $\Delta_{11} U_{i,j} = U_{i+1,j+1} - U_{i+1,j} + U_{i,j} - U_{i,j+1}$. These definitions are based on a finite difference approximation of the divergence of u.

Other definitions start with a given $u(x, y)$. We use the oscillation of $u(x, y)$ inside each cell $\Omega_{i,j}$, given by $\omega_{i,j} = \sup_{(x,y)\in\Omega_{i,j}} (u(x, y)) - \inf_{(x,y)\in\Omega_{i,j}} (u(x, y))$ as in [33]. Then the

Pierpont TV is defined by

$$TV_P(U) = D \sum_{i,j} \omega_{i,j}, \tag{2.30}$$

where D is some constant depending on the size of the grid only. Hahn's version of the definition given by

$$TV_H(U) = \sum_{i,j} \frac{\omega_{i,j}}{N^2}. \tag{2.31}$$

A comprehensive discussion of these definitions, together with a proof of their equivalence, can be found in [33]. Higher-order versions of TV discretizations (2.26), (2.27) are sometimes used. They are obtained by using second and higher order finite difference approximations to the gradient.

Discrete TV under rotation and translation.

Consider $u(x, y)$ defined on a square domain $\Omega = [-1, 1] \times [-1, 1]$. Let U be its projection onto a $N \times N$ grid with $U_{i,j}$ being the average of u in $\Omega_{i,j}$. While we do not expect any discrete TV to be rotation invariant. It is reasonable to expect that $TV_{is}(U)$ and $TV_d(U)$ to be both rotation and translation invariant under mesh refinement, i.e. as $\Delta x \to 0$, as they approximate a rotation-invariant TV (2.11). It appears that this is not the case for $TV_{is}(U)$ as can be seen from the numerical experiments in Section 2.4 of this chapter. One possible explanation is that rotation of a continuous function and rotation of the grid function are two different transformations, that result in different functions. In other words, rotating a projection of a function on the grid is not the same as rotating the function itself and then taking its projection onto a grid. Hence, there is no one-to-one mapping between a function and its projection.

Consider a square pulse given by

$$u(x, y) = \begin{cases} 1, & \text{if } x \in [-1/2, 1/2], \quad y \in [-1/2, 1/2], \\ 0, & \text{otherwise.} \end{cases}$$

We project $u(x, y)$ onto a square, 8×8 mesh as shown on Figure 2.4 (left). Then, we rotate u counterclockwise about the origin by an angle $\pi/6$ and call the result a new function v. Next, we project v onto the grid to get a discrete function V (Figure 2.4 (right)). Computing TV of U and V

$$TV_a(U) = 16\Delta x = 4, \quad TV_a(V) \approx 4.9558,$$

we observe that the value of TV is not preserved. This is expected as $TV_a(u) \neq TV_a(v)$.

Next, we consider a smooth function

$$u = e^{-10(x^2+y^2)}. \tag{2.32}$$

and find its projection U onto the grid (Figure 2.5 (left)). Translating u by $\Delta x/2$ in the horizontal direction and Δx in the vertical direction we obtain a new function v. Its projection on the grid is given by

$$V_{i,j} = \frac{1}{\Delta x^2} \int_{\Omega_{i,j}} e^{-10((x-\frac{\Delta x}{2})^2+(y-\Delta x)^2)} dxdy.$$

We plot V in Figure 2.5 (right).

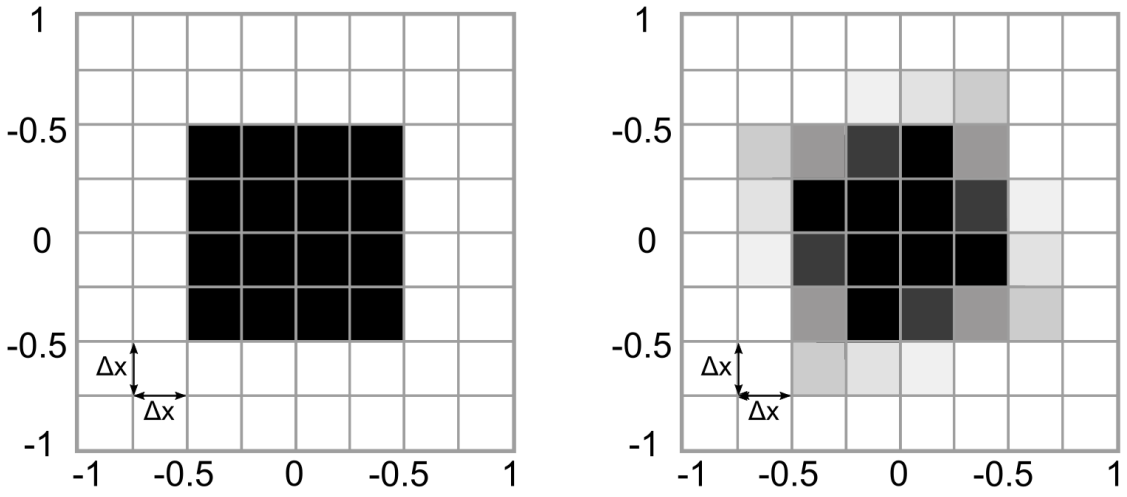


Figure 2.4: Projections U (left) and V (right) of square pulse functions u and v onto a 8×8 mesh. Black and white correspond to 0 and 1, respectively. Dark gray and light gray correspond to intermediate values between 0 an 1.

In this example, we use TV_{is} to find the TV of U and V

$$TV_{is}(U) \approx 1.6262, \quad TV_{is}(V) \approx 1.5996.$$

Therefore, both TV_a and TV_{is} depend on the orientation of the function u with respect to grid and may change under translation and rotation. Moreover, discrete TV depends on grid size. Similar results can be expected for other discrete TV definitions.

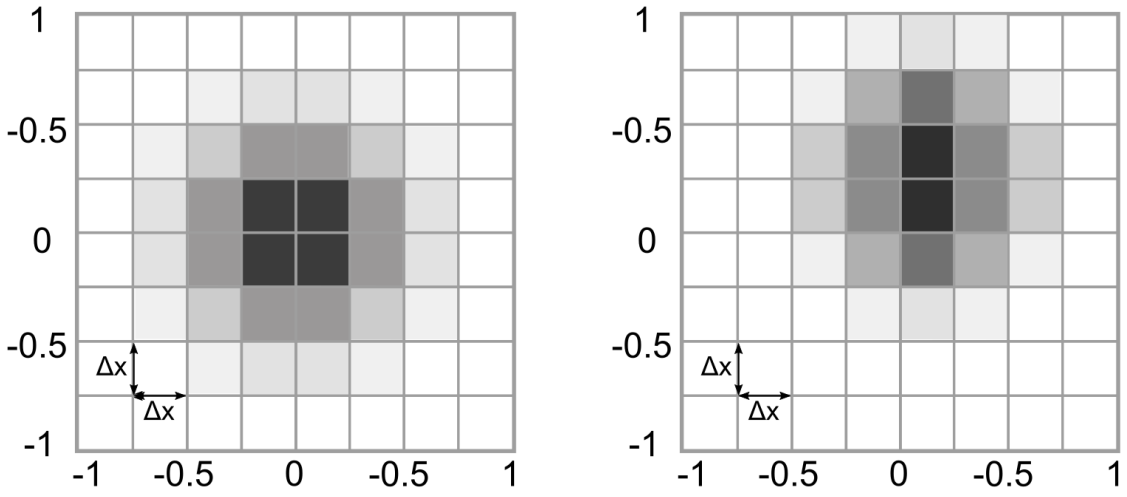


Figure 2.5: Projections U (left) and V (right) of the bell shape functions u and v onto a 8×8 square mesh. Black and white correspond to 0 and 1, respectively. Dark and light gray correspond to intermediate values between 0 an 1.

We will now introduce an alternative approach, proposed in [70], to construct a definition of dual discrete TV, that is based directly on the optimization problem (2.11).

Dual discrete total variation.

Let $u \in BV(\Omega)$, then it has a distributional derivative Du. Let a test function $\boldsymbol{\varphi} = (\varphi, \psi)$ be a differentiable vector field. We will further assume that $\|\boldsymbol{\varphi}(x,y)\| \le 1,\ \forall (x,y) \in \Omega$, and $(\boldsymbol{\varphi} \cdot \vec{\boldsymbol{n}}) = 0$ on $\partial\Omega$, where $\vec{\boldsymbol{n}}$ is the unit normal vector along the boundary $\partial\Omega$. We start with TV defined by (2.11), that we rewrite as

$$TV(u) = \sup_{\boldsymbol{\varphi} \in C_c^1(\Omega)} \left\{ \int_\Omega \boldsymbol{\varphi} Du \; dxdy : \|\boldsymbol{\varphi}(x,y)\| \le 1,\ \forall (x,y) \in \Omega \right\}.$$

Assuming for simplicity that Ω is square, we discretize it into a uniform $N \times N$ grid of square elements $\Omega_{i,j}$, of size Δx^2. Using (2.10) we write

$$\int_\Omega u \nabla \cdot \boldsymbol{\varphi} \; dxdy \approx \sum_{i,j} U_{i,j} \left(\frac{\varphi_{i+1/2,j} - \varphi_{i-1/2,j}}{\Delta x} + \frac{\psi_{i,j+1/2} - \psi_{i,j-1/2}}{\Delta x} \right) \Delta x^2, \tag{2.33}$$

where $U_{i,j}$ is the average of u in $\Omega_{i,j}$, $\varphi_{i+1/2,j} = \varphi(x_{i+1/2}, y_j)$, $\psi_{i,j+1/2} = \psi(x_i, y_{j+1/2})$, $\varphi_{i-1/2,j} = \varphi(x_{i-1/2}, y_j)$, $\psi_{i,j-1/2} = \psi(x_i, y_{j-1/2})$.

Next, we apply summation by parts to the right hand side of (2.33) to obtain a discrete analogue of (2.10)

$$\begin{aligned} &\Delta x \sum_{i,j} U_{i,j} \left((\varphi_{i+1/2,j} - \varphi_{i-1/2,j}) + (\psi_{i,j+1/2} - \psi_{i,j-1/2}) \right) \\ &= -\Delta x \sum_{i,j} \varphi_{i+1/2,j}(U_{i+1,j} - U_{i,j}) + \psi_{i,j+1/2}(U_{i,j+1} - U_{i,j}) = -\Delta x \sum_{i,j} \langle \boldsymbol{D} U_{i,j}, \boldsymbol{\varphi}_{i,j} \rangle, \end{aligned} \tag{2.34}$$

where $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product of vectors in $\mathbb{R}^2$, and $\boldsymbol{D} U_{i,j} = (D^1 U_{i,j}, D^2 U_{i,j})$, with

$$D^1 U_{i,j} = U_{i+1,j} - U_{i,j}, \quad D^2 U_{i,j} = U_{i,j+1} - U_{i,j}.$$

$\boldsymbol{D} U_{i,j}$ can be viewed as a forward difference approximation of the gradient of u at the centroid of $\Omega_{i,j}$, up to division by Δx. Alternatively, $D^1 U_{i+1/2,j}$ can be viewed as a centered approximation of the partial derivative of u with respect to x at $(x_{i+1/2}, y_j)$, the midpoint of the right edge of $\Omega_{i,j}$, and $D^2 U_{i,j+1/2}$ as the partial derivative with respect to y at $(x_i, y_{j+1/2})$, the upper edge's midpoint, up to division by Δx. Similarly, $\varphi_{i+1/2,j}$ and $\psi_{i,j+1/2}$ are combined into vector $\boldsymbol{\varphi}_{i,j} = (\varphi_{i+1/2,j}, \psi_{i,j+1/2})$. Note that although the values of φ and ψ are computed at edge midpoints, we associate $\boldsymbol{\varphi}_{i,j}$ with $\Omega_{i,j}$ and summation over i and j in (2.34).

Thus, for a discrete function U we can write a semi-discrete version of (2.11)

$$TV(U) = \max_{\boldsymbol{\varphi} \in C_c^1(\Omega)} \left\{ \Delta x \sum_{i,j} \langle \boldsymbol{D} U_{i,j}, \boldsymbol{\varphi}_{i,j} \rangle : \|\boldsymbol{\varphi}(x,y)\| \le 1,\ \forall (x,y) \in \Omega \right\}. \tag{2.35}$$

Since computation of the inner products in (2.35) requires values of the continuous function $\boldsymbol{\varphi}$ at edge midpoints only, $\boldsymbol{\varphi}$ can be replaced with a discrete function $\widetilde{\boldsymbol{\varphi}}$, with $\widetilde{\boldsymbol{\varphi}}_{i,j} = (\varphi_{i+1/2,j}, \psi_{i,j+1/2})$ defined on a $(N+1) \times (N+1)$ grid of edge midpoints $(x_{i+1/2}, y_j)$, $(x_i, y_{j+1/2})$.

To obtain a fully discrete expression for $TV(U)$, the constraint $\|\boldsymbol{\varphi}(x,y)\| \le 1$ on $\boldsymbol{\varphi}$ should be replaced with an equivalent constraint on entries of $\widetilde{\boldsymbol{\varphi}}$. In [28, 40, 70], this idea

has been extensively studied and used to construct fully discrete dual TV definitions for TV-regularization based optimization with application to imaging problems.

There are many ways to impose the bound on the norm of the discrete test function $\widetilde{\boldsymbol{\varphi}}$. An obvious constraint results from the bounds on the entries of $\boldsymbol{\varphi}$ at edge midpoints, i.e. $\sqrt{\varphi_{i+1/2,j}^2 + \psi_{i+1/2,j}^2} \leq 1$ and $\sqrt{\varphi_{i,j+1/2}^2 + \psi_{i,j+1/2}^2} \leq 1$. However, by the derivation above $\psi_{i+1/2,j}$ and $\varphi_{i,j+1/2}$ are not included in $\widetilde{\boldsymbol{\varphi}}$. Since these values are not available, they need to be defined outside of the definition (2.35). We follow the work of [70] and define them by linear interpolation. For example, the value of $\psi_{i+1/2,j}$ on a uniform grid can be approximated using by an average of $\psi_{i,j+1/2}, \psi_{i+1,j+1/2}, \psi_{i,j-1/2}$, and $\psi_{i+1,j-1}$

$$\psi_{i+1/2,j} = \frac{\psi_{i,j+1/2} + \psi_{i,j-1/2} + \psi_{i+1,j+1/2} + \psi_{i+1,j-1/2}}{4},$$

see Figure 2.6 and Figure 2.7 (right). We can write this in operator notation by setting $(\varphi_{i+1/2,j}, \psi_{i+1/2,j}) \equiv (\boldsymbol{P}^1\widetilde{\boldsymbol{\varphi}})_{i,j}$ where

$$\begin{aligned}(\boldsymbol{P}^1\widetilde{\boldsymbol{\varphi}})_{i,j} &= \left((\boldsymbol{P}^1\widetilde{\boldsymbol{\varphi}})_{i,j}^x, (\boldsymbol{P}^1\widetilde{\boldsymbol{\varphi}})_{i,j}^y\right) \\ &= \left(\varphi_{i+1/2,j}, \quad \frac{\psi_{i,j+1/2} + \psi_{i,j-1/2} + \psi_{i+1,j+1/2} + \psi_{i+1,j-1/2}}{4}\right). \end{aligned} \tag{2.36}$$

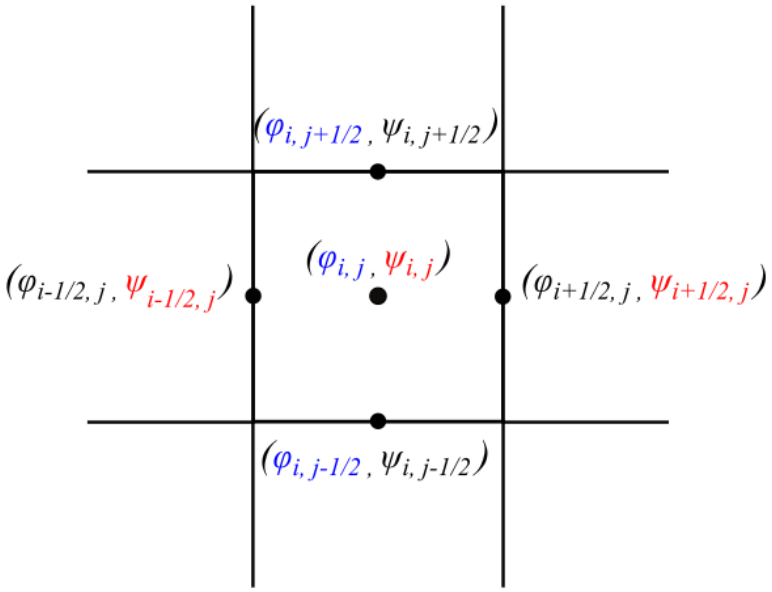


Figure 2.6: The stencil of the discrete test function in the dual definition (2.39) on $\Omega_{i,j} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$. Components of $\widetilde{\boldsymbol{\varphi}}$ are shown in black, interpolated values are shown in red and blue.

In $\boldsymbol{P}^1(\widetilde{\boldsymbol{\varphi}})$, the first component is the identity operator. The second component averages the entries of ψ on the four horizontal edges around the point $(x_{i+1/2}, y_j)$ and assigns this value to $\psi_{i+1/2,j}$, see Figure 2.7 (right). Similarly, we define $(\varphi_{i,j+1/2}, \psi_{i,j+1/2}) \equiv (\boldsymbol{P}^2\widetilde{\boldsymbol{\varphi}})_{i,j}$, where the first component is the average of the entries of φ on the four vertical edges

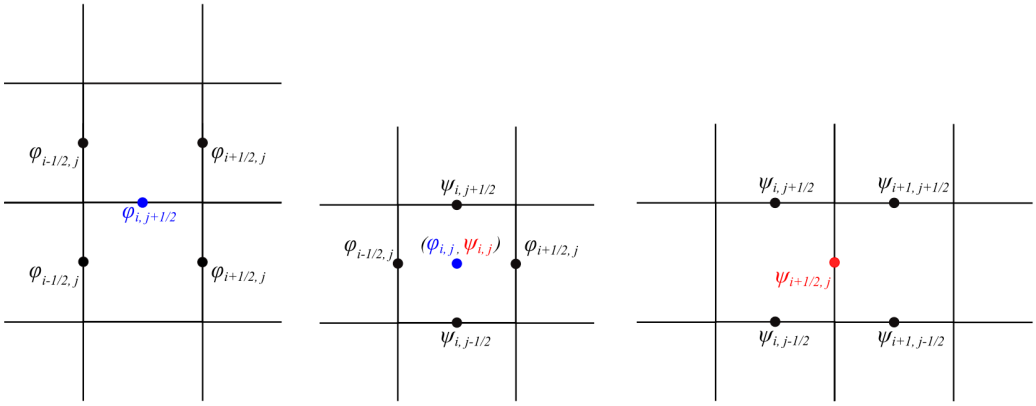


Figure 2.7: Interpolation stencils for $\varphi_{i,j+1/2}$ (left), $\varphi_{i,j}$, and $\psi_{i,j}$ (center), $\psi_{i+1/2,j}$ (right). Components of $\widetilde{\boldsymbol{\varphi}}$ are shown in black, interpolated values are shown in red and blue.

around the point $(x_i, y_{j+1/2})$ and the second component is the identity operator, see Figure 2.7 (left). Finally, we define the centroid value $(\varphi_{i,j}, \psi_{i,j}) \equiv (\boldsymbol{P}^3\widetilde{\boldsymbol{\varphi}})_{i,j}$, as an average of edge values in the horizontal and vertical directions, see Figure 2.7 (center). The operators $\boldsymbol{P}^2$ and $\boldsymbol{P}^3$ are defined as

$$\begin{aligned}(\boldsymbol{P}^2\widetilde{\boldsymbol{\varphi}})_{i,j} &= \big((\boldsymbol{P}^2\widetilde{\boldsymbol{\varphi}})^x_{i,j}, (\boldsymbol{P}^2\widetilde{\boldsymbol{\varphi}})^y_{i,j}\big) \\ &= \left(\frac{\varphi_{i+1/2,j} + \varphi_{i+1/2,j+1} + \varphi_{i-1/2,j} + \varphi_{i-1/2,j+1}}{4},\ \psi_{i,j+1/2}\right), \end{aligned} \tag{2.37}$$

$$(\boldsymbol{P}^3\widetilde{\boldsymbol{\varphi}})_{i,j} \ = \ \big((\boldsymbol{P}^3\widetilde{\boldsymbol{\varphi}})^x_{i,j}, (\boldsymbol{P}^3\widetilde{\boldsymbol{\varphi}})^y_{i,j}\big) \ = \ \left(\frac{\varphi_{i+1/2,j} + \varphi_{i-1/2,j}}{2},\ \frac{\psi_{i,j+1/2} + \psi_{i,j-1/2}}{2}\right). \tag{2.38}$$

We note here, that the operators (2.36)-(2.38) linearly interpolate the grid function $\widetilde{\boldsymbol{\varphi}}$, given on the grid of edge midpoints $(x_{i+1/2}, y_j), (x_i, y_{j+1/2})$, on a twice finer grid (x_i, y_j),$(x_i, y_{j+1/2})$, $(x_{i+1/2}, y_j)$.

Since $\boldsymbol{\varphi}$ was assumed to satisfy $(\boldsymbol{\varphi}\cdot\vec{\boldsymbol{n}}) = 0$ on $\partial\Omega$, we require $(\widetilde{\boldsymbol{\varphi}}\cdot\vec{\boldsymbol{n}}) = 0$, which means that the boundary entries of $\widetilde{\boldsymbol{\varphi}}$ are equal zero, i.e. $\varphi_{1/2,j} = \varphi_{N+1/2,j} = \psi_{i,1/2} = \psi_{i,N+1/2} = 0$, $0 \le i \le N,\ 0 \le j \le N$. Then, using the constraints and notations developed above, we arrive at a fully discrete expression for the dual total variation [40]

$$\begin{aligned}TV_d(U) = \max_{\widetilde{\boldsymbol{\varphi}}:(\widetilde{\boldsymbol{\varphi}}\cdot\vec{\boldsymbol{n}})=0} \Bigg\{ \Delta x \sum_{i,j} \langle\, \boldsymbol{D}U_{i,j}, \widetilde{\boldsymbol{\varphi}}_{i,j}\rangle :\ & \sqrt{\varphi^2_{i+1/2,j} + \psi^2_{i+1/2,j}} \le 1, \\ & \sqrt{\varphi^2_{i,j+1/2} + \psi^2_{i,j+1/2}} \le 1,\ \sqrt{\varphi^2_{i,j} + \psi^2_{i,j}} \le 1,\ \forall i,j \Bigg\}, \end{aligned} \tag{2.39}$$

where the subscript d stands for "dual". Let us define a vector space $\mathcal{P}$

$$\begin{aligned}\mathcal{P} = \Big\{ \boldsymbol{\varphi} = (\varphi, \psi),\ \varphi, \psi \in \mathbb{R}^{(N+1)\times(N+1)}\ : \\ (\boldsymbol{\varphi}\cdot\vec{\boldsymbol{n}}) = 0,\ \|(\boldsymbol{P}^k\boldsymbol{\varphi})_{i,j}\|_2 \le 1,\ k = 1,2,3,\ \forall i,j \Big\}, \end{aligned} \tag{2.40}$$

where $\|\cdot\|_2$ denotes the Euclidean norm of vectors $(\boldsymbol{P}^1\widetilde{\boldsymbol{\varphi}})_{i,j}, (\boldsymbol{P}^1\widetilde{\boldsymbol{\varphi}})_{i,j}, (\boldsymbol{P}^3\widetilde{\boldsymbol{\varphi}})_{i,j}$ in $\mathbb{R}^2$. Then

the dual discrete TV (2.39) can be rewritten as

$$TV_d(U) = \max_{\widetilde{\boldsymbol{\varphi}} \in \mathcal{P}} \Delta x \sum_{i,j} \langle \boldsymbol{D}U_{i,j}, \widetilde{\boldsymbol{\varphi}}_{i,j} \rangle. \tag{2.41}$$

Notice that computation of (2.41) cannot be done via an explicit formula and requires numerical computation of the maximizer $\boldsymbol{\varphi}$. We discuss the algorithm for computing $\boldsymbol{\varphi}$ in Chapter 4. In one spatial dimension (2.41) reduces to (2.24).

Establishing the consistency of this approach to discretizing the TV with the TV functional (2.11) is not as straightforward as it is for more conventional discretizations. A different approach to assessing the consistency should be used, the Γ-convergence, that ensures that minimizers of the discrete total variation (together with other lower-order terms) will converge to a minimizer of the exact TV as the mesh size Δx approaches zero. While this does not provide specific convergence rates or error bounds, it does guarantee the overall consistency of the dual discretizations. The main motivation behind this approach to TV discretization is the fact that the discrete dual definition considered here has been shown to exhibit better accuracy of approximation of the value of TV given by (2.11) on the grid and proper asymptotic behaviour in practical applications when compared to conventional definitions for discrete TV, see [40, 28, 76] and numerical examples therein.

The new discretization (2.41) can be extended to a three-dimensional cube Ω by using forward differences

$$D^1 U_{i,j} = U_{i+1,j,l} - U_{i,j,l}, \quad D^2 U_{i,j,l} = U_{i,j+1,l} - U_{i,j,l}, \quad D^3 U_{i,j,l} = U_{i,j,l+1} - U_{i,j,l} \tag{2.42}$$

to get $\boldsymbol{D}U_{i,j,l} = (D^1 U_{i,j,l}, D^2 U_{i,j,l}, D^3 U_{i,j,l})$, and

$$\mathcal{P}' = \Big\{ \widetilde{\boldsymbol{\varphi}} = (\varphi, \psi, \chi),\ \varphi, \psi, \chi \in \mathbb{R}^{(N+1)\times(N+1)\times(N+1)} \ : \\ (\boldsymbol{\varphi} \cdot \vec{\boldsymbol{n}}) = 0,\ \|(\boldsymbol{P}^k \boldsymbol{\varphi})_{i,j,l}\|_2 \le 1,\ k = 1, \dots, 4,\ \forall i, j, l \Big\}. \tag{2.43}$$

Then

$$TV_d(U) = \max_{\widetilde{\boldsymbol{\varphi}} \in \mathcal{P}'} \Delta x \sum_{i,j} \langle \boldsymbol{D}U_{i,j,l}, \widetilde{\boldsymbol{\varphi}}_{i,j,l} \rangle, \tag{2.44}$$

where $\widetilde{\boldsymbol{\varphi}}_{i,j,l} = (\varphi_{i+1/2,j,l}, \psi_{i,j+1/2,l}, \chi_{i,j,l+1/2})$. Additional constraints $\|\boldsymbol{P}^k \widetilde{\boldsymbol{\varphi}}\|_2 \le 1$ should be imposed and with projection operators $\boldsymbol{P}^k,\ k = 1, \dots, 4$ defined as

$$(\boldsymbol{P}^1 \widetilde{\boldsymbol{\varphi}})_{i,j,l} = \Bigg(\varphi_{i+1/2,j,l}, \quad \frac{\psi_{i,j+1/2,l} + \psi_{i,j-1/2,l} + \psi_{i+1,j+1/2,l} + \psi_{i+1,j-1/2,l}}{4}, \\ \frac{\chi_{i,j,l+1/2} + \chi_{i,j,l-1/2} + \chi_{i+1,j,l+1/2} + \chi_{i+1,j,l-1/2}}{4} \Bigg),$$

$$(\boldsymbol{P}^2 \widetilde{\boldsymbol{\varphi}})_{i,j,l} = \Bigg(\frac{\varphi_{i+1/2,j,l} + \varphi_{i+1/2,j+1,l} + \varphi_{i-1/2,j,l} + \varphi_{i-1/2,j+1,l}}{4}, \quad \psi_{i,j+1/2,l}, \\ \frac{\chi_{i,j,l+1/2} + \chi_{i,j+1,l+1/2} + \chi_{i,j,l-1/2} + \chi_{i,j+1,l-1/2}}{4} \Bigg),$$

$$(\boldsymbol{P}^3\widetilde{\boldsymbol{\varphi}})_{i,j,l} = \left(\frac{\varphi_{i+1/2,j,l} + \varphi_{i+1/2,j,l+1} + \varphi_{i-1/2,j,l} + \varphi_{i-1/2,j,l+1}}{4}, \right.$$
$$\left. \frac{\psi_{i,j+1/2,l} + \psi_{i,j+1/2,l+1} + \psi_{i,j-1/2,l} + \psi_{i,j-1/2,l+1}}{4}, \quad \chi_{i,j,l+1/2} \right),$$

$$(\boldsymbol{P}^4\widetilde{\boldsymbol{\varphi}})_{i,j,l} \quad = \quad \left(\frac{\varphi_{i+1/2,j,l} + \varphi_{i-1/2,j,l}}{2}, \quad \frac{\psi_{i,j+1/2,l} + \psi_{i,j-1/2,l}}{2}, \quad \frac{\chi_{i,j,l+1/2} + \chi_{i,j,l-1/2}}{2} \right).$$

We will limit our analysis to the two-dimensional case.

2.4 Numerical experiments

Here we provide several simple numerical tests designed to point out the features that distinguish conventional and dual TV discretizations. In this section we use a square domain $\Omega = [-2, 2] \times [-2, 2]$. In this section and in all further numerical experiments presented in this thesis, except for the numerical examples of Chapter 5, we use a laptop with a 2.3 GHz CPU and 8 GB RAM running MATLAB ver. R2022b under the University of Waterloo license.

TV of a bell shape.

We begin the numerical study of the discrete TV definitions from a continuous bell shape function discretized of the sequence of grids of decreasing grid element size Δx. The main purpose of the following example is to test the accuracy of TV_d computation and to tune the hyperparameters, including the stopping criterion. Let

$$u = e^{-10(x^2+y^2)}, \quad \forall (x, y) \in \Omega. \tag{2.45}$$

Since $u \in W^{1,1}(\Omega)$ we can use (4.10) to find its TV

$$TV(u) = \int_\Omega \sqrt{u_x^2 + u_y^2}\, dxdy \approx 1.76086 \tag{2.46}$$

We now project the function on the square $N \times N$ grid by averaging over each cell $\Omega_{i,j}$ and the grid size $\Delta x = \Delta y = \frac{4}{N}$ to get

$$U_{i,j} = \int_{\Omega_{i,j}} u(x, y)dxdy \tag{2.47}$$

forming a grid function given by $U \in \mathbb{R}^{N\times N}$. Then we compute its' TV according to three discrete TV definitions: $TV_a(U)$, $TV_{is}(U)$, and $TV_d(U)$. TV_d is computed via Algorithm 1, we found $\varepsilon = (\Delta x)^{-2}$ to be sufficient level of tolerance so that the Algorithm error is always lower than the discretization error for the TV_d.

We note here that even though the function u in this case is different from zero on the boundary of the domain Ω, numerically its' value is an order of magnitude smaller than 10^{-16}, i.e. the machine error, which makes it zero in the numerical experiment. We report the obtained values in Table 2.1.

N	$TV_a(U)$	$TV_{is}(U)$	$TV_d(U)$	δTV_a	δTV_{is}	δTV_d
10	1.3498	1.2428	1.3012	0.4111	0.5181	0.4597
20	2.0068	1.6922	1.7023	0.2460	0.0686	0.0586
40	2.1838	1.7498	1.7547	0.4229	0.0110	0.0061
80	2.2277	1.7588	1.7593	0.4668	0.0020	0.0015
160	2.2385	1.7604	1.7601	0.4776	0.0004	0.0007
320	2.2411	1.7608	1.7605	0.4803	0.0001	0.0003

Table 2.1: TV values for the bell shape function, $\delta TV = |TV(U) - TV(u)|$, where the value of $TV(u)$ is given by (2.46).

We observe that TV_a and TV_{is}, TV_d converge to a different limiting values, which are $TV(u)$ given by (2.9) and $TV(u)$ given by (2.13) respectively. We gave the approximate value for the latter above and observe that both TV_{is}, TV_d converge to that limit, which is expected for a smooth function u. While we don't give an approximate value for the limit of $TV_a(U)$ it is clear that this value will always be greater than the one given by given by (2.13), i.e. $TV_{is}(U) \leq TV_a(U)$.

Remark 2.4.1. We argue that $TV_{is}(U) \leq TV_a(U)$ and $TV_d(U) \leq TV_a(U)$, for any grid function U. These statements are obvious if we recall what constraints on the test functions $\boldsymbol{\varphi}$ correspond to each of the definitions. Any test function that satisfies the constraint for the dual discrete TV or isotropic Tv will immediately satisfy the constraints for anisotropic TV, i.e. $|\boldsymbol{\varphi}| \leq 1$. We conclude here that the space of the test function for dual discrete TV or isotropic TV is the subset of the one for anisotropic TV and therefore $TV_a(U)$ is an upper bound on the value of $TV_d(U)$.

Rotation of a square step.

Next we consider a discontinuous function, and investigate how the discrete TV of its projection onto a Cartesian grid varies under rotation. Consider a square pulse defined on Ω by

$$u(x,y) = \begin{cases} 1, & \text{if } x \in [-1/\sqrt{2}, 1/\sqrt{2}], \quad y \in [-1/\sqrt{2}, 1/\sqrt{2}], \\ 0, & \text{otherwise.} \end{cases} \tag{2.48}$$

After projecting u onto the square grid we obtain $U \in \mathbb{R}^{N\times N}$ (Figure 2.8 (A)). For simplicity we choose N to be a multiple of 4. Then $U_{i,j}$ takes three distinct values: $U_{i,j} = 1$ in the interior of the square, $U_{i,j} = \{N/(4\sqrt{2})\}$ on the elements containing the edges of the pulse and $U_{i,j} = \{N/(4\sqrt{2})\}^2$ on the elements containing the corners. Here, $\{\cdot\}$ denotes the fractional part of a real number. By (2.26), the TV_a is equal to

$$TV_a(U) = 4(2\lfloor N/(4\sqrt{2})\rfloor + 2\{N/(4\sqrt{2})\})\Delta x,$$

where $\lfloor\cdot\rfloor$ is the floor function. Under mesh refinement, i.e. as $N \to \infty$, $TV_a(U)$ tends to $4\sqrt{2}$. Using (2.27), it is straightforward to show that $TV_{is}(U)$ also converges to $4\sqrt{2}$ as $N \to \infty$. The values of $TV_a(U)$, $TV_{is}(U)$, and $TV_d(U)$ are reported in Table 2.2. We observe that all of them converge to the same value.

Next, we consider function $v(x,y) = u(x\cos\theta + y\sin\theta, -x\sin\theta + y\cos\theta)$, which is a counterclockwise rotation of u by the angle θ. Since rotation does not change $TV(u)$ given by (2.11), we have $TV(v) = TV(u)$. Choosing $\theta = \pi/4$, we obtain a square pulse whose diagonals are aligned with the coordinate axes. Using the same meshes as above,

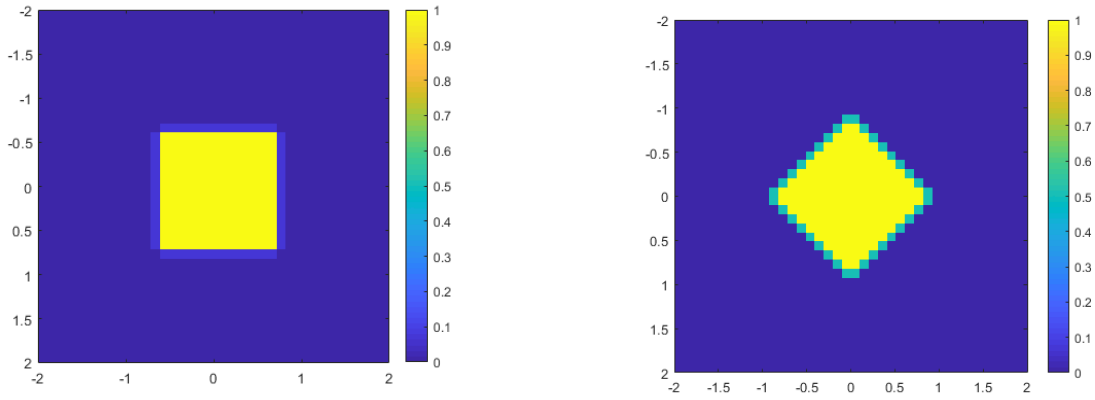
we construct discrete functions V from v (Figure 2.8 (B)). $V_{i,j}$ takes three values: 0 in the exterior of the square, 1 in the interior, and 0.5 on the elements containing edges of the pulse. We calculated the value of TV_a to be

$$TV_a(V) = (2N - 4)\Delta x = 8 - 16/N,$$

which tends to 8 as $N \to \infty$. Then, at the limit $TV_a(V)$ is greater than the limit of $TV_a(U)$ by a factor of $\sqrt{2}$. Similarly,

$$TV_{is}(V) = ((3\sqrt{2} + 2)N/4 - 5\sqrt{2}/2 + 2)\Delta x,$$

converges to $3\sqrt{2} + 2 \approx 6.24$, as $N \to \infty$, which is greater than the limit of $TV_{is}(U)$ by a factor of $(3 + \sqrt{2})/4 \approx 1.1$.



(a) Function U, square pulse. (b) Function V, rotated square pulse.

Figure 2.8: Projection of the square pulse onto 40-by-40 mesh.

N	$TV_a(U)$	$TV_{is}(U)$	$TV_d(U)$	$TV_a(V)$	$TV_{is}(V)$	$TV_d(V)$	δTV_d
20	5.1125	5.0280	5.0758	7.2000	6.0991	5.8569	0.7811
40	5.3916	5.3402	5.3628	7.6000	6.1727	5.7543	0.3915
80	5.5259	5.4919	5.5018	7.8000	6.2081	5.7050	0.2031
160	5.5918	5.5709	5.5763	7.9000	6.2255	5.6808	0.1045

Table 2.2: TV values for the square pulse U and the rotated square pulse V, $\delta TV_d = |TV_d(U) - TV_d(V)|$.

2.5 Properties of the dual discrete total variation

We study the dual definition of TV (2.41) introduced in Section 2.3 and provide results on the convexity of the test function space $\mathcal{P}$ and other properties. An essential tool for this section is Jensen's inequality in the finite form.

Jensen's inequality.

For a real convex function $g(x) : \mathbb{R} \to \mathbb{R}$, a set of real numbers $x_1, x_2, \ldots, x_m$ in its domain, and a set of positive weights $a_1, a_2, \ldots, a_m$, Jensen's inequality states that

$$g\left(\frac{\sum_{i=1}^m a_i x_i}{\sum_{i=1}^m a_i}\right) \leq \frac{\sum_{i=1}^m a_i g(x_i)}{\sum_{i=1}^m a_i}.$$

We apply the above inequality to $g(x) = x^2$ to obtain

$$\left(\sum_{i=1}^m a_i x_i\right)^2 \leq \left(\sum_{i=1}^m a_i\right)\left(\sum_{i=1}^m a_i x_i^2\right). \tag{2.49}$$

Let us consider the space $\mathcal{P}$ defined in (2.40), with projection operators $\boldsymbol{P}^k$ defined in (2.36)-(2.38). The following lemma establishes convexity of $\mathcal{P}$.

Lemma 2.5.1. *Let $\boldsymbol{\varphi}^1 = (\varphi^1, \psi^1)$ and $\boldsymbol{\varphi}^2 = (\varphi^2, \psi^2)$ belong to $\mathcal{P}$. Then for any $\alpha \in [0,1]$ we have $\alpha\boldsymbol{\varphi}^1 + (1-\alpha)\boldsymbol{\varphi}^2 \in \mathcal{P}$, i.e. $\mathcal{P}$ is convex.*

Proof. For any $\alpha \in [0,1]$, consider $\boldsymbol{\varphi} = \alpha\boldsymbol{\varphi}^1 + (1-\alpha)\boldsymbol{\varphi}^2$. To establish that $\boldsymbol{\varphi}$ belongs to $\mathcal{P}$ we need to show that $\|(\boldsymbol{P}^k\boldsymbol{\varphi})_{i,j}\|_2 \leq 1,\ \forall i,j,\ k = 1,2,3$. The first operator norm can be written as

$$\begin{aligned}
\|(\boldsymbol{P}^1\boldsymbol{\varphi})_{i,j}\|_2^2 &= (\varphi_{i+1/2,j})^2 + \left(\frac{\psi_{i,j+1/2} + \psi_{i,j-1/2} + \psi_{i+1,j+1/2} + \psi_{i+1,j-1/2}}{4}\right)^2 \\
&= (\alpha\varphi^1 + (1-\alpha)\varphi^2)_{i+1/2,j}^2 + \left(\frac{(\alpha\psi^1 + (1-\alpha)\psi^2)_{i,j+1/2} + (\alpha\psi^1 + (1-\alpha)\psi^2)_{i,j-1/2}}{4}\right. \\
&\left. + \frac{(\alpha\psi^1 + (1-\alpha)\psi^2)_{i+1,j+1/2} + (\alpha\psi^1 + (1-\alpha)\psi^2)_{i+1,j-1/2}}{4}\right)^2,
\end{aligned}$$

We apply Jensen's inequality (2.49) to both squared terms in the right hand side to get

$$\begin{aligned}
\|(\boldsymbol{P}^1\boldsymbol{\varphi})_{i,j}\|_2^2 &\leq \alpha(\varphi_{i+1/2,j}^1)^2 + (1-\alpha)(\varphi_{i+1/2,j}^2)^2 + \alpha\left(\frac{\psi_{i,j+1/2}^1 + \psi_{i,j-1/2}^1 + \psi_{i+1,j+1/2}^1 + \psi_{i+1,j-1/2}^1}{4}\right)^2 \\
&+ (1-\alpha)\left(\frac{\psi_{i,j+1/2}^2 + \psi_{i,j-1/2}^2 + \psi_{i+1,j+1/2}^2 + \psi_{i+1,j-1/2}^2}{4}\right)^2.
\end{aligned}$$

Combining terms with α and $(1-\alpha)$ yields

$$\begin{aligned}
\|(\boldsymbol{P}^1\boldsymbol{\varphi})_{i,j}\|_2^2 &\leq \alpha\left((\varphi_{i+1/2,j}^1)^2 + \left(\frac{\psi_{i,j+1/2}^1 + \psi_{i,j-1/2}^1 + \psi_{i+1,j+1/2}^1 + \psi_{i+1,j-1/2}^1}{4}\right)^2\right) \\
&+ (1-\alpha)\left((\varphi_{i+1/2,j}^2)^2 + \left(\frac{\psi_{i,j+1/2}^2 + \psi_{i,j-1/2}^2 + \psi_{i+1,j+1/2}^2 + \psi_{i+1,j-1/2}^2}{4}\right)^2\right).
\end{aligned}$$

Noticing that the expressions in brackets are $\|(\boldsymbol{P}^1\boldsymbol{\varphi}^1)_{i,j}\|_2^2$ and $\|(\boldsymbol{P}^1\boldsymbol{\varphi}^2)_{i,j}\|_2^2$, we use $\|(\boldsymbol{P}^1\boldsymbol{\varphi}^1)_{i,j}\|_2 \leq 1$ and $\|(\boldsymbol{P}^1\boldsymbol{\varphi}^2)_{i,j}\|_2 \leq 1,\ \forall i,j$ to conclude that

$$\|(\boldsymbol{P}^1\boldsymbol{\varphi})_{i,j}\|_2^2 \leq \alpha\|(\boldsymbol{P}^1\boldsymbol{\varphi}^1)_{i,j}\|_2^2 + (1-\alpha)\|(\boldsymbol{P}^1\boldsymbol{\varphi}^2)_{i,j}\|_2^2 = 1.$$

Similarly, for the second and third norm we have

$$\|(\boldsymbol{P}^2\boldsymbol{\varphi})_{i,j}\|_2^2 = (\psi_{i,j+1/2})^2 + \left(\frac{\varphi_{i+1/2,j} + \varphi_{i+1/2,j+1} + \varphi_{i-1/2,j} + \varphi_{i-1/2,j+1}}{4}\right)^2$$
$$= (\alpha\psi^1 + (1-\alpha)\psi^2)_{i,j+1/2}^2 + \left(\frac{(\alpha\varphi + (1-\alpha)\varphi^2)_{i+1/2,j} + (\alpha\varphi + (1-\alpha)\varphi^2)_{i-1/2,j+1}}{4}\right.$$
$$\left. + \frac{((\alpha\varphi + (1-\alpha)\varphi^2)_{i-1/2,j} + (\alpha\varphi + (1-\alpha)\varphi^2)_{i+1/2,j+1}}{4}\right)^2$$

and

$$\|(\boldsymbol{P}^3\boldsymbol{\varphi})_{i,j}\|_2^2 = \left(\frac{\varphi_{i+1/2,j} + \varphi_{i-1/2,j}}{2}\right)^2 + \left(\frac{\psi_{i,j+1/2} + \psi_{i,j-1/2}}{2}\right)^2$$
$$= \left(\frac{(\alpha\varphi^1 + (1-\alpha)\varphi^2)_{i+1/2,j} + (\alpha\varphi^1 + (1-\alpha)\varphi^2))_{i-1/2,j}}{2}\right)^2$$
$$+ \left(\frac{(\alpha\psi^1 + (1-\alpha)\psi^2)_{i,j+1/2}}{2} + \frac{(\alpha\psi^1 + (1-\alpha)\psi^2))_{i,j-1/2}}{2}\right)^2.$$

We apply Jensen's inequality to get

$$\|(\boldsymbol{P}^2\boldsymbol{\varphi})_{i,j}\|_2^2 \le \alpha(\psi^1_{i,j+1/2})^2 + (1-\alpha)(\psi^2_{i,j+1/2})^2 + \alpha\left(\frac{\varphi^1_{i+1/2,j} + \varphi^1_{i+1/2,j+1} + \varphi^1_{i-1/2,j} + \varphi^1_{i-1/2,j+1}}{4}\right)^2$$
$$+ (1-\alpha)\left(\frac{\varphi^2_{i+1/2,j} + \varphi^2_{i+1/2,j+1} + \varphi^2_{i-1/2,j} + \varphi^2_{i-1/2,j+1}}{4}\right)^2$$

and

$$\|(\boldsymbol{P}^3\boldsymbol{\varphi})_{i,j}\|_2^2 \le \alpha\left(\frac{\varphi^1_{i+1/2,j} + \varphi^1_{i-1/2,j}}{2}\right)^2 + \alpha\left(\frac{\psi^1_{i,j+1/2} + \psi^1_{i,j-1/2}}{2}\right)^2$$
$$+ (1-\alpha)\left(\frac{\varphi^2_{i+1/2,j} + \varphi^2_{i-1/2,j}}{2}\right)^2 + (1-\alpha)\left(\frac{\psi^2_{i,j+1/2} + \psi^2_{i,j-1/2}}{2}\right)^2.$$

We use the properties of $\boldsymbol{\varphi}^1$ and $\boldsymbol{\varphi}^2$ to conclude that

$$\|(\boldsymbol{P}^2\boldsymbol{\varphi})_{i,j}\|_2^2 \le \alpha\left((\psi^1_{i,j+1/2})^2 + \left(\frac{\varphi^1_{i+1/2,j} + \varphi^1_{i+1/2,j+1} + \varphi^1_{i-1/2,j} + \varphi^1_{i-1/2,j+1}}{4}\right)^2\right)$$
$$+ (1-\alpha)\left((\psi^2_{i,j+1/2})^2 + \left(\frac{\varphi^2_{i+1/2,j} + \varphi^2_{i+1/2,j+1} + \varphi^2_{i-1/2,j} + \varphi^2_{i-1/2,j+1}}{4}\right)^2\right)$$
$$\le \alpha + (1-\alpha) = 1$$

and

$$\begin{aligned}\|(\boldsymbol{P}^3\boldsymbol{\varphi})_{i,j}\|_2^2 \leq \alpha &\left(\left(\frac{\varphi^1_{i+1/2,j}+\varphi^1_{i-1/2,j}}{2}\right)^2+\left(\frac{\psi^1_{i,j+1/2}+\psi^1_{i,j-1/2}}{2}\right)^2\right)\\ &+(1-\alpha)\left(\left(\frac{\varphi^2_{i+1/2,j}+\varphi^2_{i-1/2,j}}{2}\right)^2+(1-\alpha)\left(\frac{\psi^2_{i,j+1/2}+\psi^2_{i,j-1/2}}{2}\right)^2\right)\\ &\leq \alpha+(1-\alpha)=1.\end{aligned}$$

Therefore, we have shown that $\boldsymbol{\varphi}\in P$. □

Lemma 2.5.1 can be extended to an arbitrary number of functions.

Lemma 2.5.2. *Let $\boldsymbol{\varphi}^1,\boldsymbol{\varphi}^2,\ldots,\boldsymbol{\varphi}^m\in\mathcal{P}$, then their convex combination is also in $\mathcal{P}$.*

Proof. Similarly to Lemma 2.5.1, the statement follows from constraints (2.36)-(2.38) by direct application of Jensen's inequality.

Lemma 2.5.3. *Consider $\boldsymbol{\varphi}^1=(\varphi^1,\mathbf{0})$, $\boldsymbol{\varphi}^2=(\varphi^2,\mathbf{0})$, $\ldots$, $\boldsymbol{\varphi}^m=(\varphi^m,\mathbf{0})\in\mathcal{P}$. Let $\boldsymbol{\alpha}^1,\boldsymbol{\alpha}^2,\ldots,\boldsymbol{\alpha}^m\in[0,1]^{N\times N}$ be such that $\sum_{k=1}^m\alpha^k_{i+1/2,j}=1$, $\forall i,j\in[1,N]$. Let $\boldsymbol{\varphi}$ be defined as*

$$\boldsymbol{\varphi}_{i,j}=(\varphi_{i+1/2,j},0)=(\alpha^1_{i+1/2,j}\varphi^1_{i+1/2,j}+\alpha^2_{i+1/2,j}\varphi^2_{i+1/2,j}+\cdots+\alpha^m_{i+1/2,j}\varphi^m_{i+1/2,j},0).$$

Then $\boldsymbol{\varphi}$ belongs to $\mathcal{P}$.

Proof. Similarly to the proof of Lemma 2.5.1, we check the constraints (2.36)-(2.38). Consider $\|(\boldsymbol{P}^1\boldsymbol{\varphi})_{i,j}\|_2^2$

$$\|(\boldsymbol{P}^1\boldsymbol{\varphi})_{i,j}\|_2^2=(\varphi_{i+1/2,j})^2+\left(\frac{\psi_{i,j+1/2}+\psi_{i,j-1/2}+\psi_{i+1,j+1/2}+\psi_{i+1,j-1/2}}{4}\right)^2.$$

Since all components in the second term are identically zero, we have

$$\|(\boldsymbol{P}^1\boldsymbol{\varphi})_{i,j}\|_2^2=\left(\sum_k\alpha^k_{i+1/2,j}\varphi^k_{i+1/2,j}\right)^2.$$

Applying Jensen's inequality, yields

$$\|(\boldsymbol{P}^1\boldsymbol{\varphi})_{i,j}\|_2^2\leq\sum_k\alpha^k_{i+1/2,j}(\varphi^k_{i+1/2,j})^2\leq\sum_k\alpha^k_{i+1/2,j}=1,$$

because $(\varphi^k_{i+1/2,j})^2\leq 1$. Next, we show that $\|(\boldsymbol{P}^2\boldsymbol{\varphi})_{i,j}\|_2\leq 1$ and $\|(\boldsymbol{P}^3\boldsymbol{\varphi})_{i,j}\|_2\leq 1$.

$$
\begin{aligned}
\|(\boldsymbol{P}^2\boldsymbol{\varphi})_{i,j}\|_2^2 &= \psi_{i,j+1/2}^2 + \left(\frac{\varphi_{i+1/2,j} + \varphi_{i+1/2,j+1} + \varphi_{i-1/2,j} + \varphi_{i-1/2,j-1}}{4}\right)^2 \\
&= \frac{1}{4}\Bigg(\sum_k \alpha_{i+1/2,j}^k \varphi_{i+1/2,j}^k + \alpha_{i+1/2,j+1}^k \varphi_{i+1/2,j-1}^k \\
&+ \alpha_{i-1/2,j}^k \varphi_{i-1/2,j}^k + \alpha_{i-1/2,j+1}^k \varphi_{i-1/2,j+1}^k \Bigg)^2 \\
&\leq \frac{1}{16}\Bigg(\sum_k \alpha_{i+1/2,j}^k (\varphi_{i+1/2,j}^k)^2 + \sum_k \alpha_{i+1/2,j+1}^k (\varphi_{i+1/2,j+1}^k)^2 \\
&+ \sum_k \alpha_{i-1/2,j}^k (\varphi_{i-1/2,j}^k)^2 + \sum_k \alpha_{i-1/2,j+1}^k (\varphi_{i-1/2,j+1}^k)^2 \Bigg) \\
&\leq \frac{1}{4}\left(\sum_k \alpha_{i+1/2,j}^k\right) + \frac{1}{4}\left(\sum_k \alpha_{i+1/2,j+1}^k\right) + \frac{1}{4}\left(\sum_k \alpha_{i-1/2,j}^k\right) + \frac{1}{4}\left(\sum_k \alpha_{i-1/2,j+1}^k\right) \\
&\leq 1
\end{aligned}
$$

and

$$
\begin{aligned}
\|(\boldsymbol{P}^3\boldsymbol{\varphi})_{i,j}\|_2^2 &= \left(\frac{\varphi_{i+1/2,j} + \varphi_{i-1/2,j}}{2}\right)^2 + \left(\frac{\psi_{i,j+1/2} + \psi_{i,j-1/2}}{2}\right)^2 \\
&= \left(\frac{\sum_k \alpha_{i+1/2,j}^k \varphi_{i+1/2,j}^k + \sum_k \alpha_{i+1/2,j}^k \varphi_{i-1/2,j}^k}{2}\right)^2 \\
&\leq \frac{1}{2}\left(\sum_k \alpha_{i+1/2,j}^k (\varphi_{i+1/2,j}^k)^2 + \sum_k \alpha_{i+1/2,j}^k (\varphi_{i-1/2,j}^k)^2\right) \leq 1.
\end{aligned}
$$

We conclude that $\boldsymbol{\varphi} \in \mathcal{P}$. □

Remark 2.5.3. Lemma 2.5.3 is a more general statement than Lemma 2.5.2, but it can be applied only to $\boldsymbol{\varphi}$ of specific form,i.e. only to $\boldsymbol{\varphi}$ with one component equal zero. The Lemma can also be written for $\boldsymbol{\varphi} = (\mathbf{0}, \psi)$ with a similar proof. This lemma will be useful in Chapter 3.

Remark 2.5.4. We observe that

$$
TV_a(U) \;=\; \max_{\boldsymbol{\varphi}\in\mathbb{R}^2\,:\,(\boldsymbol{\varphi}\cdot\vec{\boldsymbol{n}})=0} \left\{ \Delta x \sum_{i,j} \langle\, \boldsymbol{D}U_{i,j}, \boldsymbol{\varphi}_{i,j}\rangle \;:\; |\varphi_{i+1/2,j}| \,\leq\, 1,\; |\psi_{i,j+1/2}| \,\leq\, 1 \;\forall i,j \right\},
$$

and

$$
TV_{is}(U) \;=\; \max_{\boldsymbol{\varphi}\in\mathbb{R}^2\,:\,(\boldsymbol{\varphi}\cdot\vec{\boldsymbol{n}})=0} \left\{ \Delta x \sum_{i,j} \langle\, \boldsymbol{D}U_{i,j}, \boldsymbol{\varphi}_{i,j}\rangle \;:\; \sqrt{\varphi_{i+1/2,j}^2 + \psi_{i+1/2,j}^2} \;\leq\; 1 \;\forall i,j \right\},
$$

and we have an explicit formula for the maximizer

$$
(\boldsymbol{\varphi}^a)_{i,j} = \left(sgn(D^1U_{i,j}), sgn(D^2U_{i,j})\right),
$$

for anisotropic TV, where $sgn(\cdot)$ denotes the function that returns 1 for positive arguments and -1 for negative. Similarly

$$(\boldsymbol{\varphi}^{is})_{i,j} = \left(\frac{D^1 U_{i,j}}{\sqrt{(D^1 U_{i,j})^2 + (D^2 U_{i,j})^2}}, \frac{D^2 U_{i,j}}{\sqrt{(D^1 U_{i,j})^2 + (D^2 U_{i,j})^2}} \right),$$

for isotropic TV, while for the dual discrete TV (2.41) we have no explicit formula.

2.6 Results

The first example compares the accuracy of various discrete Total Variation (TV) discretizations. For smooth functions, the error between the discretized $TV_{is}(U)$, $TV_d(U)$ and $TV(u)$ is due to the averaging of the function only. Additionally, TV_d computation involves an error term not present in TV_{is}. We observe that the discretization error for TV_d scales as $O(\Delta x)$. While TV_a converges to a different value (the integral of the L^1-norm as defined in (2.9)), both TV_{is} and TV_d converge to the exact value given by (1.6), as expected.

The second example focuses on a discontinuous square shape. We find that $TV(U) \neq TV(V)$ for all considered TV definitions, which is unsurprising since U and V represent different projections. However, we're interested in how these values converge asymptotically. Notably, out of the three TVs studied, only $TV_d(V)$ exhibits diminishing differences between the original and rotated functions (represented by $\delta TV = |TV_d(U) - TV_d(V)|$) as the mesh is refined (as shown in the last column of Table 2.2).

Both TV_a and TV_{is} change with rotation of the discontinuous pulse, and this dependence remains even with mesh refinement. In contrast, while the TV_d also changes under rotation, this difference diminishes under mesh refinement and therefore the error due to rotation can be reduced by using a finer grid. This sensitivity to rotation suggests that TV_a and TV_{is} might not be suitable for accurately capturing the total variation of grid functions.

Finally, we establish several important properties of the space of dual test functions of the dual discrete TV.

2.7 Summary

In this chapter, we have introduced the definition of TV of grid functions in one spatial dimension. We then explored several definitions for discrete TV in two dimensions, highlighting the key differences between each approach and the continuous definition.

Several illustrative examples were presented to differentiate between the continuous definitions (2.11) and (2.8), along with their corresponding discretizations. We also observed and discussed the lack of translational and rotational isotropy in conventional discrete TV definitions.

Next, an alternative approach, based on an optimization problem ((2.11)), to TV discretization was introduced. This approach was then used to derive a definition for the discrete dual TV given in [40]. Numerical experiments were conducted to evaluate the accuracy of the dual TV computation compared to conventional discrete definitions. The results demonstrate that, unlike conventional anisotropic and isotropic TV definitions, the dual TV achieves rotational invariance under mesh refinement.

Finally, we investigate the properties of the dual discrete TV, focusing on the convexity of the test function space and the collinearity of the maximizer vector with the forward difference vector. Proofs for these results are provided in Section 2.5.

Chapter 3

Total variation stability of numerical methods for scalar conservation laws

In this chapter, we discuss stability properties of numerical methods used for solution of scalar conservation laws.

Non-increasing total variation is a key property of the exact solution, which holds in one and multiple space dimensions. Theorem 2.1.3 in Chapter 2 proves this in one spatial dimension and Theorem 2.2.5 extends the result to multiple dimensions. This property of the exact solution is highly desirable to have when designing higher-order numerical methods. As was discussed in the introduction, numerical solutions might have spurious oscillations near discontinuities that increase the total variation. One way to suppress oscillations is by enforcing the TVD property on the numerical solution at each time step.

3.1 Total variation diminishing schemes in one spatial dimension

In one spatial dimension, we consider scalar conservation laws of the form

$$u_t + f(u)_x = 0, \ x \in I \tag{3.1}$$

with appropriate initial and boundary conditions.

Let $\bar{I}$ be a uniform grid of N elements of I. A one-step discretization scheme for solution of (3.1) can be written as

$$U^{n+1} = F(U^n), \tag{3.2}$$

where U^n is the numerical solution at $t = t^n$.

Since the TV of the exact solution does not increase with time we require the TV of the numerical solution to not increase at each time step. Let Δt^n denote the time step size at step n, then $t^{n+1} = t^n + \Delta t^n$.

Definition 3.1.1. *A consistent method is called total variation diminishing (TVD) if for any U^n approximating the solution $u(x,t)$ on $\bar{I}$ at the time level $t = t^n$, the values U^{n+1} at the next time level $t = t^{n+1}$ satisfy*

$$TV(U^{n+1}) \leq TV(U^n). \tag{3.3}$$

Note that the condition (3.3) requires TV to be non-increasing. Initially, it was called "total variation non-increasing" (TVNI) by Harten [67]. Now it is referred to as TVD.

We call a numerical scheme total-variation stable if $TV(U^n)$ is uniformly bounded for all n. If the scheme is TV-stable, then all numerical solutions lie in some compact set and we have the following convergence result.

Theorem 3.1.1 ([66]). *Let us assume that the scheme $U^{n+1} = F(U^n)$ is consistent with the conservation law (3.1) and its entropy inequality (1.11). If the resulting numerical approximation is TV-stable, then the scheme is convergent in the L^1-norm, and its limit is the unique weak solution of (3.1) that satisfies the entropy inequality (1.11).*

Definition 3.1.2. *A method is called monotonicity-preserving (MP) if $U_i^n \geq U_{i+1}^n$ for all i implies that $U_i^{n+1} \geq U_{i+1}^{n+1}$ for all i.*

This definition implies that if a numerical solution is monotone at $t = t^n$, it will remain monotone for all future time steps.

Theorem 3.1.2. *Any TVD method for the solution of (3.1) is monotonicity preserving.*

Proof. Let us assume $U_i^n \geq U_{i+1}^n$ for all i and $TV(U^n) < \infty$. Then we must have

$$TV(U^n) = |U_R - U_L|,$$

where U_L is the minimal value of U^n on the grid and U_R is the maximum value.

Suppose U^{n+1} is not monotone. Then it has at least one local minimum U' and one local maximum U''. Then

$$TV(U^{n+1}) \geq |U'' - U'| + |U_R - U_L| \geq TV(U^n),$$

which contradicts the assumption that the scheme is TVD.

□

The case of $U_i^{n+1} \leq U_{i+1}^{n+1}$ has a similar proof. It is unclear how to extend the monotone-preserving property to two and more spatial dimensions, therefore, there is no known version of Theorem 3.1.2 in two or more dimensions. For more details, see example in Chapter 2 (page 20).

Any oscillation appearing in a monotone solution would necessarily increase its total variation, hence a TVD method must be MP and the set of monotone schemes contains TVD schemes in one spatial dimension [67].

3.2 Harten's Lemma

In [67] Harten considered a class of non–linear one-dimensional schemes of the form

$$U_i^{n+1} = U_i^n + C_i^n DU_i^n - D_{i-1}^n DU_{i-1}^n, \tag{3.4}$$

where $DU_i^n = U_{i+1}^n - U_i^n$ and the coefficients C_i^n, D_{i-1}^n are assumed to be non-constant and dependent on U^n. Harten proved the following important result.

Theorem 3.2.1 (Harten's Lemma)**.** *For any scheme of the form* (3.4) *for solving* (3.1) *with periodic boundary conditions, the following restrictions on coefficients are sufficient for the scheme to be TVD*

$$C_i \geq 0, \quad D_i \geq 0, \quad 0 \leq C_i + D_i \leq 1. \tag{3.5}$$

Proof. We write (3.4) for U_i^{n+1} and U_{i+1}^{n+1}

$$U_i^{n+1} = U_i^n + C_i^n DU_i^n - D_{i-1}^n DU_{i-1}^n,$$

and

$$U_{i+1}^{n+1} = U_{i+1}^n + C_{i+1}^n DU_{i+1}^n - D_i^n DU_i^n.$$

Subtract U_i^{n+1} from U_{i+1}^{n+1}, to get

$$U_{i+1}^{n+1} - U_i^{n+1} = (1 - C_i^n - D_i^n)DU_i^n + C_{i+1}^n DU_{i+1}^n + D_{i-1}^n DU_{i-1}^n.$$

Taking absolute values of both sides and applying the triangle inequality yields

$$|U_{i+1}^{n+1} - U_i^{n+1}| \leq |1 - C_i^n - D_i^n||DU_i^n| + |C_{i+1}^n||DU_{i+1}^n| + |D_{i-1}^n||DU_{i-1}^n|.$$

Using assumption (3.5), we obtain

$$|U_{i+1}^{n+1} - U_i^{n+1}| \leq (1 - C_i^n - D_i^n)|DU_i^n| + C_{i+1}^n|DU_{i+1}^n| + D_{i-1}^n|DU_{i-1}^n|.$$

Summing the inequalities over grid points i, we get

$$\begin{aligned}\sum_i |U_{i+1}^{n+1} - U_i^{n+1}| \leq \sum_i |U_{i+1}^n - U_i^n| - \sum_i C_i^n|U_{i+1}^n - U_i^n| - \sum_i D_i^n|U_{i+1}^n - U_i^n| \\ + \sum_i C_{i+1}^n|U_{i+2}^n - U_{i+1}^n| + \sum_i D_{i-1}^n|U_i^n - U_{i-1}^n|.\end{aligned}$$

After changing the index of summation in the second and fourth sums on the right–hand side, we see that they sum up to zero. And so do the third and fifth sums, leading to

$$TV(U^{n+1}) = \sum_i |U_{i+1}^{n+1} - U_i^{n+1}| \leq \sum_i |U_{i+1}^n - U_i^n| = TV(U^n).$$

Theorem 3.2.2 ([128] Godunov's theorem)**.** *Linear monotone numerical schemes for solution of* (3.1) *are at most first-order accurate.*

It can be shown that any linear TVD scheme is monotone and any monotone linear scheme is TVD [67]. Hence, a linear TVD scheme can be at most first-order accurate. Harten's lemma can be used for construction of nonlinear, high-order, TV-stable numerical schemes, overcoming the restriction of Godunov's theorem.

Theorem 3.2.3 ([66])**.** *Assume that the scheme* (3.4)-(3.5) *is consistent with the conservation law* (3.1) *and its entropy inequality* (1.11)*. If the resulting numerical approximation is TV stable, then the scheme is convergent and its limit is the unique weak solution of* (3.1) *that satisfies* (1.11)*.*

By construction, TVD schemes are only first-order accurate at solution extrema [104, 105] and can be second-order accurate in the rest of the domain. In a later study [107], a

trade-off between second-order accuracy and the TVD requirement was demonstrated for schemes equipped with non-linear TVD limiters.

3.3 Total variation stability for higher order methods

In Chapter 1 we have discussed a simple FV method that was based on piecewise constant approximation. Most schemes of this form are at most first-order accurate. To achieve higher-order spatial accuracy, piecewise linear or higher-order reconstruction is needed.

Monotone upstream-centered schemes for conservation laws (MUSCL) [131, 86] are one of the most popular methods for the solution of (3.1). We give a short derivation of the second-order MUSCL scheme with a limiter for solution of the linear advection equation. We use a scheme of this type for numerical experiments in Section 3.5.

Consider the following Cauchy problem

$$u_t + a u_x = 0, \;\; x \in \mathbb{R}, \tag{3.6}$$
$$u(x,0) = u_0(x). \tag{3.7}$$

We discretize the equation using a finite volume formulation as in (1.14). A MUSCL-type numerical scheme uses piecewise linear reconstruction to achieve second-order accuracy. Let us consider the case $a > 0$, the case of $a < 0$ is similar. Let U_i^n be an approximation of the average of $u(x,t^n)$ on the interval $[x_{i-1/2}, x_{i+1/2}]$ at $t = t^n$. We write a linear reconstruction formula in the i-th cell as

$$\widetilde{U}^n(x) = U_i^n + \sigma_i^n (x - x_i), \qquad \text{for } x \in [x_{i-1/2}, x_{i+1/2}], \tag{3.8}$$

where σ_i^n is the reconstructed slope. The cell average of $\widetilde{U}^n(x)$ in (3.8) over $[x_{i-1/2}, x_{i+1/2}]$ is equal to U_i^n for any σ_i^n.

The exact solution to (3.6)-(3.7) at $t = t^n$ is given by $u(x,t^n) = u_0(x - at^n)$. We also have that $u(x,t^{n+1}) = u(x - a\Delta t, t^n)$. Using (3.8),(1.14) and (1.15), the piecewise linear FV approximation to the solution at $t = t^{n+1}$ can be computed as

$$\widetilde{U}^{n+1}(x) = \widetilde{U}^n(x - a\Delta t).$$

Then, after cell-averaging we obtain

$$U_i^{n+1} = U_i^n - \nu(U_i^n - U_{i-1}^n) - \frac{\nu(1-\nu)}{2}\Delta x(\sigma_i^n - \sigma_{i-1}^n), \tag{3.9}$$

where $\nu = a\frac{\Delta t}{\Delta x}$. It is possible to extend (3.8) to higher-order reconstructions, e.g. parabolic interpolation in [81] results in a third-order accurate scheme.

The slope in (3.8) can be reconstructed in multiple ways, for example we can approximate the slope with $\sigma_i^n = \frac{U_i^n - U_{i-1}^n}{\Delta x}$ on a uniform grid. However, this may result in oscillations or an unstable solution.

We look for slopes in the following form

$$\sigma_i^n = \frac{U_{i+1}^n - U_i^n}{\Delta x}\phi(\theta_i^n),$$

where $\phi(\theta_i)$ is the limiter function and $\theta_i^n = \frac{U_i^n - U_{i-1}^n}{U_{i+1}^n - U_i^n}$. The limiter function $\phi(\theta)$ aims to restrict the slope to ensure that the solution is TVD. With this choice, we write (3.9) as

$$U_i^{n+1} = U_i^n - \nu(U_i^n - U_{i-1}^n) - \frac{\nu(1-\nu)}{2}(\phi(\theta_i^n)(U_{i+1}^n - U_i^n) - \phi(\theta_{i-1}^n)(U_i^n - U_{i-1}^n)). \quad (3.10)$$

In (3.10), the flux limiter $\phi(\theta)$ can be reinterpreted as a slope limiter.

The scheme will satisfy the TVD condition provided that $0 \leq \nu \leq 1$ (the CFL condition) and the following inequality holds [90]

$$\left| \frac{\phi(\theta_1)}{\theta_1} - \phi(\theta_2) \right| \leq 2, \quad \forall \theta_1, \theta_2 \in \mathbb{R}. \quad (3.11)$$

In the case $\theta \leq 0$, we have an extremum. In order to retain the TVD property, we must set $\phi(\theta) = 0$ for θ_1 and θ_2. When $\theta > 0$, we need to have $\phi(\theta) > 0$, since we would like to preserve the sign of the slope when applying the limiter. Then we must require

$$0 \leq \theta \cdot \phi(\theta) \leq 2 \quad \text{and} \quad 0 \leq \phi(\theta) \leq 2, \quad \forall \theta \geq 0.$$

These constraints define the TVD region in the θ-ϕ plane (Figure 3.1 (left)). For second-order accuracy, we additionally require $\phi(1) = 1$, i.e. that a linear solution is not limited. The admissible limiter region for second-order TVD schemes is known as the Sweby region (Figure 3.1 (right)).

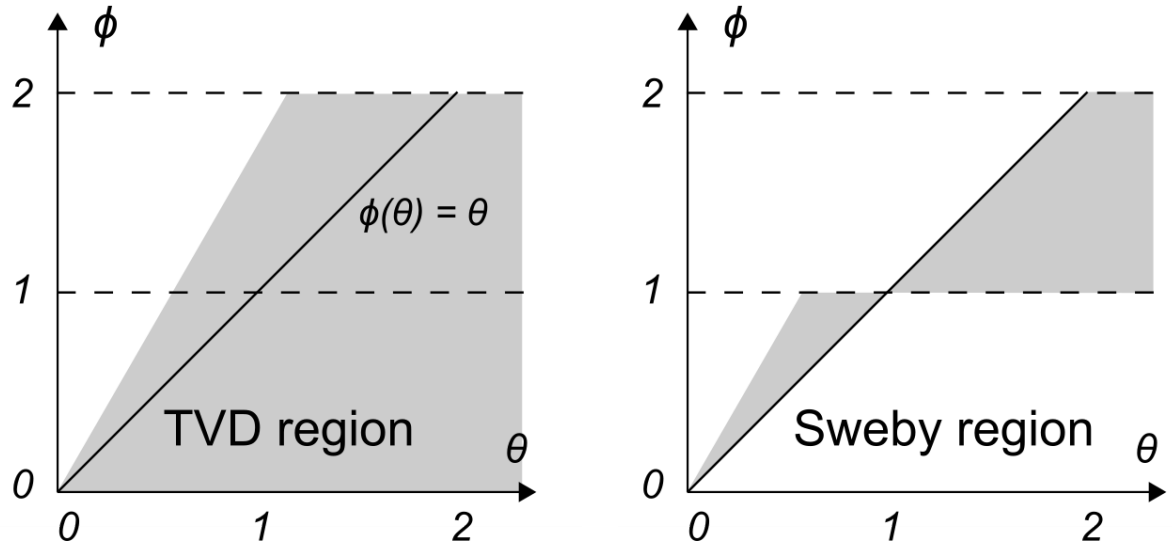


Figure 3.1: TVD and Sweby (second order TVD) regions in the ϕ-θ plane.

There are many choices for the limiting function $\phi(\theta)$. One of the first limiters, monotonized central-difference (MC), was proposed in [131] and is given by

$$\phi(\theta) = \max\left(\min\left(2\theta, \frac{1+\theta}{2}, 0\right), 0\right),$$

or in the slope-limiter form

$$\sigma_i^n = minmod\left(2\frac{U_{i+1}^n - U_i^n}{\Delta x}, \frac{U_{i+1}^n - U_{i-1}^n}{2\Delta x}, 2\frac{U_i^n - U_{i-1}^n}{\Delta x}\right),$$

where $minmod(\cdot,\cdot,\cdot)$ is defined by

$$minmod(a,b,c) = \frac{1}{2}(sgn(c) + sgn(minmod(a,b)))\min\left(|c|, |minmod(a,b)|\right),$$
$$minmod(a,b) = \frac{1}{2}(sgn(a) + sgn(b))\min\left(|a|, |b|\right). \tag{3.12}$$

A good example of a second-order MUSCL scheme is the Kurganov-Tadmor (KT) scheme that we use for numerical experiments in Section 3.5. This scheme employs the following variation of the MC limiter

$$\sigma_i^n = \ minmod\left(r\frac{U_{i+1}^n - U_i^n}{\Delta x}, \frac{U_{i+1}^n - U_{i-1}^n}{2\Delta x}, r\frac{U_i^n - U_{i-1}^n}{\Delta x}\right),$$

where $1 \le r \le 2$ is a parameter.

KT scheme is TVD in one spatial dimension and satisfies the local maximum principle in two spatial dimensions [82]. We provide more details on KT scheme in two spatial dimensions in Section 3.5. The full derivation of the algorithm can be found in the original paper [82]. The approach can also be extended to third-order schemes [81]. We will demonstrate on a number of numerical examples in Section 3.5 that KT method is TVD in the dual TV sense.

3.4 Total variation stability in two spatial dimensions

We consider a scalar nonlinear conservation law

$$u_t + f(u)_x + g(u)_y = 0 \tag{3.13}$$

with an appropriate initial condition, and periodic boundary conditions in x and y directions. We discretize it with a five-point stencil scheme on a uniform $N \times N$ mesh with discretization step $\Delta x > 0$, using

$$\begin{aligned} U_{i,j}^{n+1} = U_{i,j}^n + A_{i,j}^n(U_{i+1,j}^n - U_{i,j}^n) + B_{i-1,j}^n(U_{i-1,j}^n - U_{i,j}^n) \\ + C_{i,j}^n(U_{i,j+1}^n - U_{i,j}^n) + D_{i,j-1}^n(U_{i,j-1}^n - U_{i,j}^n), \end{aligned} \tag{3.14}$$

where

$$A_{i,j}^n = A_{i,j}^n(\dots U_{i,j-1}^n, \dots, U_{i-1,j}^n, U_{i,j}^n, U_{i+1,j}^n, \dots, U_{i,j+1}^n, \dots), \ \forall i,j \in [1,N],$$

and, similarly, $B_{i-1,j}^n, C_{i,j}^n$, and $D_{i,j-1}^n$ depend on U^n. The coefficients are related to the partial derivatives of flux functions f and g. In the simpler case of first-order scheme, the relation between the coefficients $A_{i,j}, B_{i-1,j}^n, C_{i,j}^n, D_{i,j-1}^n$ and fluxes f and g can be found in [67]. For a general nonlinear scheme the relation cannot be written explicitly, because of the presence of a limiter function ϕ, for example (3.10).

We can rewrite (3.14) in the following compact form

$$U_{i,j}^{n+1} = U_{i,j}^n + A_{i,j}^n D^1 U_{i,j}^n - B_{i-1,j}^n D^1 U_{i-1,j}^n + C_{i,j}^n D^2 U_{i,j}^n - D_{i,j-1}^n D^2 U_{i,j-1}^n, \tag{3.15}$$

using the forward difference notation introduced in Section 2.3.

In two spatial dimensions imposing the limiting conditions of Harten's Lemma in each space dimension is insufficient to guarantee non-increasing total variation. Goodman and LeVeque [61] showed that except in certain trivial cases any method of the form (3.15) that is TVD is at most first-order accurate [61], if the anisotropic definition of TV ((2.26)) is used. To prove this, an associated one-dimensional scheme with the same order of accuracy as the two-dimensional scheme and special initial data were considered. If the two-dimensional scheme is TVD in definition (2.26), then the one-dimensional scheme should be monotone at least on certain initial data and, therefore, should be at most first-order accurate. Thus, the original two-dimensional scheme is also at most first-order accurate.

For steady-state solutions of (3.13), the local maximum principle (LMP) property

$$\min(U^n_{i-1,j}, U^n_{i+1,j}, U^n_{i,j-1}, U^n_{i,j+1}) \le U^{n+1}_{i,j} \le \max(U^n_{i-1,j}, U^n_{i+1,j}, U^n_{i,j-1}, U^n_{i,j+1})$$

has been proven [121] under the following set of conditions on scheme coefficients

$$A^n_{i,j} \ge 0, \quad B^n_{i,j} \ge 0, \quad C^n_{i,j} \ge 0, \quad D^n_{i,j} \ge 0, \tag{3.16}$$

$$A^n_{i,j} + B^n_{i-1,j} + C^n_{i,j} + D^n_{i,j-1} \le 1. \tag{3.17}$$

LMP is a weaker property than monotonicity or TVD, but it is easier to prove. It has been shown that many conventional methods are LMP, e.g. FVM [7] and DG [54].

Several first-order schemes can be written in the form (3.15), e.g. upwind, Godunov, and Lax-Friedrichs methods. Additionally, a variety of second-order schemes have been developed for solution of scalar conservation laws in two dimensions, see e.g. Nessyahu-Tadmor [103, 109], Osher-Tadmor [105], Bouchut [14], Kurganov-Tadmor [82], discontinuous Galerkin (DG) [116, 75].

We rewrite (3.15) in a matrix-vector form as

$$U^{n+1} = LU^n, \quad L \in \mathbb{R}^{S\times S}, \tag{3.18}$$

where we write the numerical solution as a vector U of size $S = N^2$ by traversing the mesh columns

$$U = \left(U_{1,1}, U_{2,1}, \ldots, U_{N,1}, U_{1,2}, U_{2,2}, \ldots, U_{N,2}, \ldots, U_{1,N}, U_{2,N}, \ldots, U_{N,N}\right)^T.$$

Generally, the matrix L is not constant as its elements will change from one time step to another, but for simplicity we omit the superscript n for it.

Then the rows of $L = (l_1, l_2, \ldots, l_S)^T$ are given by

$$l_s = \left(0, \ldots, D^n_{i,j-1}, \ldots, B^n_{i-1,j}, 1 - A^n_{i,j} - B^n_{i-1,j} - C^n_{i,j} - D^n_{i,j-1}, A^n_{i,j}, \ldots, C^n_{i,j}, \ldots, 0\right),$$

for $s = i + (j-1)N, 1 < i < N, 1 < j < N$, with non-zero entries $D^n_{i,j-1}$ located at $i + (j-2)N$, $B^n_{i-1,j}$ at $i - 1 + (j-1)N$, $1 - A^n_{i,j} - B^n_{i-1,j} - C^n_{i,j} - D^n_{i,j-1}$ at $i + (j-1)N$, $A^n_{i,j}$ at $i + 1 + (j-1)N$, and $C^n_{i,j}$ at $i + jN$. The entries of L that correspond to the cells lying on the boundary of the domain must be handled separately. We make adjustments to such entries, i.e. $i = 1, i = N$ or $j = 1, j = N$, to account for periodic boundary conditions, see Appendix A.

Using (3.15), we compute $D^1U^{n+1}_{i,j} = U^{n+1}_{i+1,j} - U^{n+1}_{i,j}$ as follows

$$\begin{aligned} D^1U^{n+1}_{i,j} &= (U^n_{i+1,j} - U^n_{i,j}) + A^n_{i+1,j}(U^n_{i+2,j} - U^n_{i+1,j}) - (A^n_{i,j} + B^n_{i,j})(U^n_{i+1,j} - U^n_{i,j}) \\ &+ B^n_{i-1,j}(U^n_{i,j} - U^n_{i-1,j}) + C^n_{i+1,j}(U^n_{i+1,j+1} - U^n_{i+1,j}) - C^n_{i,j}(U^n_{i,j+1} - U^n_{i,j}) \\ &- D^n_{i+1,j-1}(U^n_{i+1,j} - U^n_{i+1,j-1}) + D^n_{i,j-1}(U^n_{i,j} - U^n_{i,j-1}). \end{aligned} \tag{3.19}$$

This can be written as

$$\begin{aligned} D^1U^{n+1}_{i,j} &= (1 - A^n_{i,j} - B^n_{i,j})D^1U^n_{i,j} + A^n_{i+1,j}D^1U^n_{i+1,j} + B^n_{i-1,j}D^1U^n_{i-1,j} \\ &+ C^n_{i+1,j}D^2U^n_{i+1,j} - C^n_{i,j}D^2U^n_{i,j} - D^n_{i+1,j-1}D^2U^n_{i+1,j-1} + D^n_{i,j-1}D^2U^n_{i,j-1}. \end{aligned} \tag{3.20}$$

Similarly, for $D^2U^{n+1}_{i,j} = U^{n+1}_{i,j+1} - U^{n+1}_{i,j}$ we have

$$\begin{aligned} D^2U^{n+1}_{i,j} &= (U^n_{i,j+1} - U^n_{i,j}) + C^n_{i,j+1}(U^n_{i,j+2} - U^n_{i,j+1}) - (C^n_{i,j} + D^n_{i,j})(U^n_{i,j+1} - U^n_{i,j}) \\ &+ D^n_{i,j-1}(U^n_{i,j} - U^n_{i,j-1}) + A^n_{i,j+1}(U^n_{i+1,j+1} - U^n_{i,j+1}) - A^n_{i,j}(U^n_{i+1,j} - U^n_{i,j}) \\ &- B^n_{i-1,j+1}(U^n_{i,j+1} - U^n_{i-1,j+1})) + B^n_{i-1,j}(U^n_{i,j} - U^n_{i-1,j}) \end{aligned} \tag{3.21}$$

or

$$\begin{aligned} D^2U^{n+1}_{i,j} &= (1 - C^n_{i,j} - D^n_{i,j})D^2U^n_{i,j} + C^n_{i,j+1}D^2U^n_{i,j+1} + D^n_{i,j-1}D^2U^n_{i,j-1} \\ &+ A^n_{i,j+1}D^1U^n_{i,j+1} - A^n_{i,j}D^1U^n_{i,j} - B^n_{i-1,j+1}D^1U^n_{i-1,j+1} + B^n_{i-1,j}D^1U^n_{i-1,j}. \end{aligned} \tag{3.22}$$

We concatenate D^1U^{n+1} and D^2U^{n+1} into a single vector of length $2S$ as follows

$$\begin{aligned} \boldsymbol{D}U^{n+1} = \Big(& D^1U^{n+1}_{1,1}, D^1U^{n+1}_{2,1}, \ldots, D^1U^{n+1}_{N,1}, D^1U^{n+1}_{1,2}, \ldots, D^1U^{n+1}_{N,2}, \ldots D^1U^{n+1}_{N,N}, \\ & D^2U^{n+1}_{1,1}, D^2U^{n+1}_{2,1}, \ldots, D^2U^{n+1}_{N,1}, D^2U^{n+1}_{1,2}, \ldots, D^2U^{n+1}_{N,2}, \ldots D^1U^{n+1}_{N,N} \Big)^T. \end{aligned} \tag{3.23}$$

Then we can rewrite (3.23) in matrix form

$$\boldsymbol{D}U^{n+1} = M\boldsymbol{D}U^n,$$

where M is a $2S$-by-$2S$ multi-diagonal matrix. The rows of matrix $M = (m_1, m_2, \ldots, m_{2S})^T$ in the upper half of the matrix, i.e for $1 \le s \le S$, are of the following form

$$\begin{aligned} m_s = \Big(& \ldots, B^n_{i-1,j}, 1 - A^n_{i,j} - B^n_{i,j}, A^n_{i+1,j}, \ldots, \\ & D^n_{i,j-1}, -D^n_{i+1,j-1}, \ldots, -C^n_{i,j}, C^n_{i+1,j}, \ldots \Big), \end{aligned} \tag{3.24}$$

where s is related to indices i, j as $s = i + (j-1)N, 1 < i < N, 1 < j < N$. The non-zero entries $B^n_{i-1,j}$ are located at $i - 1 + (j-1)N$, $1 - A^n_{i,j} - B^n_{i,j}$ at $i + (j-1)N$, $A^n_{i+1,j}$ at $i - 1 + jN$, $D^n_{i,j-1}$ at $i + jN$, $-D^n_{i+1,j-1}$ at $S + i + (j-2)N$, $-C^n_{i,j}$ at $S + i + (j-1)N$, and $C^n_{i+1,j}$ at $S + i + jN$. In the lower half of the matrix M, i.e for $s = S + i + (j-1)N$, we

have

$$m_s = \Big(\dots, B^n_{i-1,j}, -A^n_{i,j}, \dots, -B^n_{i-1,j+1}, A^n_{i,j+1}, \dots, \\ D^n_{i,j-1}, \dots, 1 - C^n_{i,j} - D^n_{i,j}, \dots, C^n_{i,j+1}, \dots \Big), \quad (3.25)$$

where non-zero entries $B^n_{i-1,j}$ are located at $i-1+(j-1)N$, $-A^n_{i,j}$ at $i+(j-1)N$, $-B^n_{i-1,j+1}$ at $i-1+jN$, $A^n_{i,j+1}$ at $i+jN$, $D^n_{i,j-1}$ at $S+i+(j-2)N$, $1-C^n_{i,j}-D^n_{i,j}$ at $S+i+(j-1)N$, and $C^n_{i,j+1}$ at $S+i+jN$. Matrix entries that correspond to the cells lying on the boundary of the domain must be handled separately. We make adjustments to such entries, i.e. $i=1, i=N$ or $j=1, j=N$, to account for periodic boundary conditions, see Appendix A.

We begin the discussion of stability of two-dimensional schemes (3.15) with an analogue of Harten's Lemma. We show that schemes (3.15) are TVD in the dual TV sense on essentially one-dimensional data.

Lemma 3.4.1. *Let $U^{n+1} = LU^n$ be a scheme of the form (3.15) and let*

$$D^2 U^n_{i,j} = 0, \ \forall i, j. \quad (3.26)$$

Assume that the coefficients satisfy the following conditions

$$A^n_{i,j} \geq 0, \quad B^n_{i,j} \geq 0, \quad C^n_{i,j} \geq 0, \quad D^n_{i,j} \geq 0, \quad (3.27)$$

$$A^n_{i,j} + B^n_{i-1,j} \leq 1, \ \text{and } C^n_{i,j} + D^n_{i,j-1} \leq 1, \quad \forall i, j \in [1, N], \quad \forall n. \quad (3.28)$$

Then the total variation of the solution does not increase with time, i.e.

$$TV_d(U^{n+1}) \leq TV_d(U^n).$$

Proof. We consider

$$\begin{aligned} TV_d(U^n) - TV_d(U^{n+1}) &= \Delta x \sum_{i,j} \langle \boldsymbol{D} U^n_{i,j}, \boldsymbol{\varphi}^n_{i,j} \rangle - \Delta x \sum_{i,j} \langle \boldsymbol{D} U^{n+1}_{i,j}, \boldsymbol{\varphi}^{n+1}_{i,j} \rangle \\ &= \Delta x \sum_{i,j} \langle \boldsymbol{D} U^n_{i,j}, \boldsymbol{\varphi}^n_{i,j} \rangle - \Delta x \sum_{i,j} \langle M \boldsymbol{D} U^n_{i,j}, \boldsymbol{\varphi}^{n+1}_{i,j} \rangle \\ &= \Delta x \sum_{i,j} \langle \boldsymbol{D} U^n_{i,j}, \boldsymbol{\varphi}^n_{i,j} \rangle - \Delta x \sum_{i,j} \langle \boldsymbol{D} U^n_{i,j}, M^T \boldsymbol{\varphi}^{n+1}_{i,j} \rangle \\ &= \Delta x \sum_{i,j} \langle \boldsymbol{D} U^n_{i,j}, \boldsymbol{\varphi}^n_{i,j} - M^T \boldsymbol{\varphi}^{n+1}_{i,j} \rangle, \end{aligned}$$

where $\boldsymbol{\varphi}^n$ and $\boldsymbol{\varphi}^{n+1}$ are maximizers for U^n and U^{n+1} concatenated into one-dimensional vectors in the same way as $\boldsymbol{D}U$. We proceed by examining the components of $M^T \boldsymbol{\varphi}^{n+1}$. We will omit the superscript n in scheme coefficients for brevity. Using (3.24) and (3.25) we write

$$\begin{aligned} (M^T \boldsymbol{\varphi}^{n+1})_s = A_{i,j} \varphi^{n+1}_{i-1/2,j} + (1 - A_{i,j} - B_{i,j}) \varphi^{n+1}_{i+1/2,j} + B_{i,j} \varphi^{n+1}_{i+3/2,j} \\ + A_{i,j} \psi^{n+1}_{i,j-1/2} - B_{i,j} \psi^{n+1}_{i+1,j-1/2} - A_{i,j} \psi^{n+1}_{i,j+1/2} + B_{i,j} \psi^{n+1}_{i+1,j+1/2}, \quad (3.29) \end{aligned}$$

for $s = i + (j-1)N$ and

$$(M^T\boldsymbol{\varphi}^{n+1})_s = C_{i,j}\psi^{n+1}_{i,j-1/2} + (1 - C_{i,j} - D_{i,j})\psi^{n+1}_{i,j+1/2} + D_{i,j}\psi^{n+1}_{i,j+3/2} + C_{i,j}\varphi^{n+1}_{i-1/2,j} - D_{i,j}\varphi^{n+1}_{i-1/2,j+1} - C_{i,j}\varphi^{n+1}_{i+1/2,j} + D_{i,j}\varphi^{n+1}_{i+1/2,j+1}, \quad (3.30)$$

for $s = S + i + (j-1)N$. We split $M^T\boldsymbol{\varphi}^{n+1}$ into a sum of two vectors $\boldsymbol{p}^{n+1}$ and $\boldsymbol{q}^{n+1}$ as $M^T\boldsymbol{\varphi}^{n+1} = \boldsymbol{p}^{n+1} + \boldsymbol{q}^{n+1}$, where we define

$$p^{n+1}_s = A_{i,j}\varphi^{n+1}_{i-1/2,j} + (1 - A_{i,j} - B_{i,j})\varphi^{n+1}_{i+1/2,j} + B_{i,j}\varphi^{n+1}_{i+3/2,j},$$

for $s = i + (j-1)N$, and

$$p^{n+1}_s = C_{i,j}\psi^{n+1}_{i,j-1/2} + (1 - C_{i,j} - D_{i,j})\psi^{n+1}_{i,j+1/2} + D_{i,j}\psi^{n+1}_{i,j+3/2}, \quad (3.31)$$

for $s = S + i + (j-1)N$. We define

$$q^{n+1}_s = A_{i,j}\psi^{n+1}_{i,j-1/2} - B_{i,j}\psi^{n+1}_{i+1,j-1/2} - A_{i,j}\psi^{n+1}_{i,j+1/2} + B_{i,j}\psi^{n+1}_{i+1,j+1/2}, \quad (3.32)$$

for $s = i + (j-1)N$, and

$$q^{n+1}_s = C_{i,j}\varphi^{n+1}_{i-1/2,j} - D_{i,j}\varphi^{n+1}_{i-1/2,j+1} - C_{i,j}\varphi^{n+1}_{i+1/2,j} + D_{i,j}\varphi^{n+1}_{i+1/2,j+1}, \quad (3.33)$$

for $s = S + i + (j-1)N$.

Using (3.22) we establish that for one-dimensional data (3.26) we have $D^2U^{n+1}_{i,j} = 0$ at $t = t^{n+1}$ and all subsequent steps of the scheme. Without loss of generality, we can assume that for $D^2U^{n+1}_{i,j} = 0$, the corresponding values of the test function $\psi^{n+1}_{i,j+1/2} = 0$, as this does not contradict the fact that $\boldsymbol{\varphi}^{n+1}$ is the maximizer for U^{n+1} in (2.41). Since the assumption (3.26) holds at every point, we conclude that $\psi^{n+1}_{i,j+1/2} = 0$, $\forall i, j$. This yields $p^{n+1}_s = 0$ for $s = i + (j-1)N + S$ in (3.31) and $q^{n+1}_s = 0$ for $s = i + (j-1)N$ in (3.32). Then we have

$$\Delta x \sum_{i,j} \langle \boldsymbol{D}U^n_{i,j}, \boldsymbol{q}^{n+1}_{i,j} \rangle = \Delta x \sum_{i,j} \langle D^1U^n_{i,j}, 0 \rangle + \Delta x \sum_{i,j} \langle 0, (C_{i,j}\varphi^{n+1}_{i-1/2,j} - D_{i,j}\varphi^{n+1}_{i-1/2,j+1} - C_{i,j}\varphi^{n+1}_{i+1/2,j} + D_{i,j}\varphi^{n+1}_{i+1/2,j+1}) \rangle = 0.$$

Moreover, since $p^{n+1}_s = 0$ for $s = S + i + (j-1)N$, then $\boldsymbol{p}^{n+1} \in \mathcal{P}$ under assumptions (3.27)-(3.28) by Lemma 2.5.3. This yields

$$\begin{aligned} TV_d(U^n) - TV_d(U^{n+1}) &= \Delta x \sum_{i,j} \langle \boldsymbol{D}U^n_{i,j}, \boldsymbol{\varphi}^n_{i,j} - M^T\boldsymbol{\varphi}^{n+1}_{i,j} \rangle \\ &= \Delta x \sum_{i,j} \langle \boldsymbol{D}U^n_{i,j}, \boldsymbol{\varphi}^n_{i,j} \rangle - \Delta x \sum_{i,j} \langle \boldsymbol{D}U^n_{i,j}, \boldsymbol{p}^{n+1}_{i,j} + \boldsymbol{q}^{n+1}_{i,j} \rangle \\ &= \Delta x \sum_{i,j} \langle \boldsymbol{D}U^n_{i,j}, \boldsymbol{\varphi}^n_{i,j} \rangle - \Delta x \sum_{i,j} \langle \boldsymbol{D}U^n_{i,j}, \boldsymbol{p}^{n+1}_{i,j} \rangle - \Delta x \sum_{i,j} \langle \boldsymbol{D}U^n_{i,j}, \boldsymbol{q}^{n+1}_{i,j} \rangle \\ &= \Delta x \sum_{i,j} \langle \boldsymbol{D}U^n_{i,j}, \boldsymbol{\varphi}^n_{i,j} \rangle - \Delta x \sum_{i,j} \langle \boldsymbol{D}U^n_{i,j}, \boldsymbol{p}^{n+1}_{i,j} \rangle. \end{aligned}$$

Since $\boldsymbol{\varphi}^n$ is the maximizer in (2.41), we have

$$\Delta x \sum_{i,j} \langle \boldsymbol{D}U^n_{i,j}, \boldsymbol{\varphi}^n_{i,j} \rangle - \Delta x \sum_{i,j} \langle \boldsymbol{D}U^n_{i,j}, \boldsymbol{p}^{n+1}_{i,j} \rangle \geq 0,$$

and

$$TV_d(U^n) - TV_d(U^{n+1}) \geq 0 \tag{3.34}$$

□

The case $D^1 U^n_{i,j} = 0,\ \forall i, j$ is similar.

Notice that Lemma 3.4.1 is valid for $TV_a(U)$ and $TV_{is}(U)$ with one dimensional data (3.26), since the conditions $\|\boldsymbol{P}^k \boldsymbol{\varphi}\|_2 \leq 1,\ k = 1, \ldots, 3,\ \forall i, j$ imposed on the dual functions $\boldsymbol{\varphi}^n, \boldsymbol{\varphi}^{n+1}$ reduce to $|\varphi^n_{i+1/2,j}| \leq 1,\ |\varphi^{n+1}_{i+1/2,j}| \leq 1 \quad \forall i, j$. In this case, there is no difference between the anisotropic, isotropic and dual discrete TVs.

Next, let us consider the linear advection equation

$$u_t + a u_x + b u_y = 0, \quad a, b > 0, \quad (x, y) \in \Omega,$$

with a suitable initial condition and periodic boundary conditions, discretized with the upwind scheme on a uniform grid

$$U^{n+1}_{i,j} = U^n_{i,j} - \frac{a\Delta t}{\Delta x}(U^n_{i,j} - U^n_{i-1,j}) - \frac{b\Delta t}{\Delta y}(U^n_{i,j} - U^n_{i,j-1}), \tag{3.35}$$

which is (3.15) with $A^n_{i,j} = 0,\ B^n_{i-1,j} = \frac{a\Delta t}{\Delta x},\ C^n_{i,j} = 0,\ D^n_{i,j-1} = \frac{b\Delta t}{\Delta y}$.

Lemma 3.4.2. *Assume that the coefficients satisfy* $B^n_{i-1,j} + D^n_{i,j-1} \leq 1,\ B^n_{i-1,j} \geq 0,\ D^n_{i,j-1} \geq 0,\ , \quad \forall i, j \in [1, N], \quad \forall n,$ *i.e.*

$$\frac{a\Delta t}{\Delta x} + \frac{b\Delta t}{\Delta y} \leq 1,\ \frac{a\Delta t}{\Delta x} \geq 0,\ \frac{b\Delta t}{\Delta y} \geq 0. \tag{3.36}$$

Then the (3.35) *is TVD in the dual sense*

$$TV_d(U^{n+1}) \leq TV_d(U^n).$$

Proof. Since $B_{i-1,j}$ and $D_{i,j-1}$ are constant we write (3.35) as

$$\boldsymbol{D}U^{n+1}_{i,j} = (1 - B - D)\boldsymbol{D}U^n_{i,j} + B\boldsymbol{D}U^n_{i-1,j} + D\boldsymbol{D}U^n_{i,j-1}.$$

Therefore

$$\begin{aligned}
TV_d(U^n) - TV_d(U^{n+1}) &= \Delta x \sum_{i,j} \langle \boldsymbol{D}U^n_{i,j}, \boldsymbol{\varphi}^n_{i,j} \rangle - \Delta x \sum_{i,j} \langle \boldsymbol{D}U^{n+1}_{i,j}, \boldsymbol{\varphi}^{n+1}_{i,j} \rangle \\
&= \Delta x \sum_{i,j} \langle \boldsymbol{D}U^n_{i,j}, \boldsymbol{\varphi}^n_{i,j} \rangle - \Delta x \sum_{i,j} \langle (1 - B - D)\boldsymbol{D}U^n_{i,j} + B\boldsymbol{D}U^n_{i-1,j} + D\boldsymbol{D}U^n_{i,j-1}, \boldsymbol{\varphi}^{n+1}_{i,j} \rangle \\
&= \Delta x \sum_{i,j} \langle \boldsymbol{D}U^n_{i,j}, \boldsymbol{\varphi}^n_{i,j} \rangle - (1 - B - D)\Delta x \sum_{i,j} \langle \boldsymbol{D}U^n_{i,j}, \boldsymbol{\varphi}^{n+1}_{i,j} \rangle - B\Delta x \sum_{i,j} \langle \boldsymbol{D}U^n_{i-1,j}, \boldsymbol{\varphi}^{n+1}_{i,j} \rangle \\
&\quad - D\Delta x \sum_{i,j} \langle \boldsymbol{D}U^n_{i,j-1}, \boldsymbol{\varphi}^{n+1}_{i,j} \rangle.
\end{aligned}$$

Since by assumption, $(1-B-D)$, B and D are nonegative numbers and $\boldsymbol{\varphi}^n$ is the maximizer in (2.41), we have the following

$$-B\Delta x\sum_{i,j}\langle\, \boldsymbol{D}U^n_{i-1,j},\boldsymbol{\varphi}^{n+1}_{i,j}\rangle \geq -B\Delta x\sum_{i,j}\langle\, \boldsymbol{D}U^n_{i-1,j},\boldsymbol{\varphi}^{n}_{i-1,j}\rangle,$$

and similar inequalities for the other two terms. Then

$$\begin{aligned}TV_d(U^n)-TV_d(U^{n+1}) \geq \Delta x\sum_{i,j}\langle\, \boldsymbol{D}U^n_{i,j},\boldsymbol{\varphi}^{n}_{i,j}\rangle - (1-B-D)\Delta x\sum_{i,j}\langle\, \boldsymbol{D}U^n_{i,j},\boldsymbol{\varphi}^{n}_{i,j}\rangle\\ - B\Delta x\sum_{i,j}\langle\, \boldsymbol{D}U^n_{i-1,j},\boldsymbol{\varphi}^{n}_{i-1,j}\rangle - D\Delta x\sum_{i,j}\langle \boldsymbol{D}U^n_{i,j-1},\boldsymbol{\varphi}^{n}_{i,j-1}\rangle \geq 0,\end{aligned}$$

since

$$TV_d(U^n)=\Delta x\sum_{i,j}\langle \boldsymbol{D}U^n_{i-1,j},\boldsymbol{\varphi}^{n}_{i-1,j}\rangle=\Delta x\sum_{i,j}\langle \boldsymbol{D}U^n_{i,j-1},\boldsymbol{\varphi}^{n}_{i,j-1}\rangle.$$

□

The upwind scheme (3.35) is only first-order accurate. It is stable under the CFL condition $|\frac{a\Delta t}{\Delta x}|+|\frac{b\Delta t}{\Delta y}|\leq 1$, which is satisfied under assumption of Lemma 3.4.2. Then the scheme is stable and TVD under Harten's-like conditions.

The conditions on the coefficients used in Lemma 3.4.2 are not sufficient to guarantee the TVD property of the schemes of the form (3.15) in the dual discrete sense. For nonlinear schemes, the conditions need to be more restrictive. The following example is of particular interest as it presents a set of necessary conditions on the coefficients of the scheme of the form (3.15) to be TVD in the dual discrete sense.

Necessary TVD conditions.

Let $U^{n+1}=LU^n$ be a scheme of the form (3.15) and assume that its coefficients in the first time step satisfy

$$A^0_{i,j}+B^0_{i-1,j}\leq 1/4,\quad C^0_{i,j}+D^0_{i,j-1}\leq 1/4, \tag{3.37}$$

$$A^0_{i,j}\geq 0,\ B^0_{i-1,j}\geq 0,\ C^0_{i,j}\geq 0,\ D^0_{i,j-1}\geq 0,\ ,\quad \forall i,j\in[1,N]. \tag{3.38}$$

Let the initial condition be given by

$$U^0_{i.j}=\begin{cases}1, & \text{if } i=p,\ j=q,\\ 0, & \text{otherwise,}\end{cases} \tag{3.39}$$

for some p,q, such that (x_p,y_q) is not lying on the boundary of the domain. Then we have

$$TV_d(U^1)\leq TV_d(U^0).$$

To show that $TV_d(U^1)\leq TV_d(U^0)$, we compute DU^0

$$D^1U^0_{p,q}=-1,\quad D^1U^0_{p-1,q}=1,\quad D^2U^0_{p,q}=-1,\quad D^2U^0_{p,q-1}=1.$$

Using (2.41) we find that $TV_d(U^0)=4\Delta x$. Then, using (3.19)-(3.22), after one step of the

scheme we obtain

$$
\begin{aligned}
&D^1U^1_{p,q} = -1 + A_{p,q} + B_{p,q} + B_{p-1,q} + C_{p,q} + D_{p,q-1},\\
&D^1U^1_{p-2,q} = A_{p-1,q},\\
&D^1U^1_{p+1,q} = -B_{p,q},\\
&D^1U^1_{p-1,q} = 1 - A_{p-1,q} - B_{p-1,q} - A_{p,q} - C_{p,q} - D_{p,q-1},\\
&D^1U^1_{p-1,q+1} = D_{p,q}, \quad D^1U^1_{p,q+1} = -D_{p,q}, \quad D^1U^1_{p-1,q-1} = C_{p,q-1} \quad D^1U^1_{p,q-1} = -C_{p,q-1},
\end{aligned}
$$

$$
\begin{aligned}
&D^2U^1_{p,q} = -1 + C_{p,q} + D_{p,q} + D_{p,q-1} + A_{p,q} + B_{p-1,q},\\
&D^2U^1_{p,q-2} = C_{p,q-1},\\
&D^2U^1_{p,q+1} = -D_{p,q},\\
&D^2U^1_{p,q-1} = 1 - C_{p,q-1} - D_{p,q-1} - C_{p,q} - A_{p,q} - B_{p-1,q},\\
&D^2U^1_{p+1,q-1} = B_{p,q}, \quad D^2U^1_{p+1,q} = -B_{p,q}, \quad D^2U^1_{p-1,q-1} = A_{p-1,q}, \quad D^2U^1_{p-1,q} = -A_{p-1,q},
\end{aligned}
$$

where we omitted the superscript index 0 in the coefficients for brevity. Under conditions (3.37)-(3.38) we can determine the signs of forward differences

$$
\begin{aligned}
&D^1U^1_{p,q} \le 0, && D^1U^1_{p+1,q} \le 0, && D^1U^1_{p-2,q} \ge 0, && D^1U^1_{p-1,q} \ge 0,\\
&D^1U^1_{p-1,q+1} \ge 0, && D^1U^1_{p,q+1} \le 0, && D^1U^1_{p-1,q-1} \ge 0, && D^1U^1_{p,q-1} \le 0,\\
&D^2U^1_{p,q} \le 0, && D^2U^1_{p,q+1} \le 0, && D^2U^1_{p,q-2} \ge 0, && D^2U^1_{p,q-1} \ge 0,\\
&D^2U^1_{p+1,q-1} \ge 0, && D^2U^1_{p+1,q} \le 0, && D^2U^1_{p-1,q-1} \ge 0, && D^2U^1_{p-1,q} \le 0.
\end{aligned} \tag{3.40}
$$

Omitting the superscript 1 for the test function and substituting the expressions for the forward differences into the formula for TV yields

$$
\begin{aligned}
TV_d(U^1) &= \Delta x \sum_{i,j} \langle \boldsymbol{D}U^1_{i,j}, \boldsymbol{\varphi}^1_{i,j} \rangle\\
&= -(1 - A_{p,q} - B_{p,q} - B_{p-1,q} - C_{p,q} - D_{p,q-1})\varphi_{p+1/2,q}\\
&+ (1 - A_{p-1,q} - B_{p-1,q} - A_{p,q} - C_{p,q} - D_{p,q-1})\varphi_{p-1/2,q}\\
&+ (1 - C_{p,q-1} - D_{p,q-1} - C_{p,q} - A_{p,q} - B_{p-1,q})\psi_{p,q-1/2}\\
&- (1 - C_{p,q} - D_{p,q} - D_{p,q-1} - A_{p,q} - B_{p-1,q})\psi_{p,q+1/2}\\
&- A_{p-1,q}\varphi_{p-3/2,q} + B_{p,q}\varphi_{p+3/2,q} - D_{p,q}\varphi_{p-1/2,q+1} + D_{p,q}\varphi_{p+1/2,q+1}\\
&+ C_{p,q-1}\varphi_{p+1/2,q-1} - C_{p,q-1}\psi_{p,q-3/2} + D_{p,q}\psi_{p,q+3/2} - B_{p,q}\psi_{p+1,q-1/2}\\
&- A_{p-1,q}\psi_{p-1,q-1/2} + A_{p-1,q}\psi_{p-1,q+1/2} + B_{p,q}\psi_{p+1,q+1/2} - C_{p,q-1}\varphi_{p-1/2,q-1}.
\end{aligned} \tag{3.41}
$$

We look for the maximum value in 3.41 when the scheme coefficients and test functions satisfy (3.37)-(3.38), (2.36)-(2.38). Differentiating with respect to $\varphi_{p+1/2,q}$, we find that the maximum cannot be achieved inside the domain given by the above restrictions. Thus, the maximum must lie on the boundary of the domain. By checking the boundaries, we we find that the maximum in (3.41) is obtained when all of the scheme coefficients are equal to zero. Hence, we obtain

$$
TV_d(U^1) \le (-\varphi_{p+1/2,q} + \varphi_{p-1/2,q} - \psi_{p,q-1/2} + \psi_{p,q+1/2})\Delta x \le 4,
$$

where the second inequality follows from the constraints on $\boldsymbol{\varphi}$.

Since we have already established that $TV_d(U^0) = 4\Delta x$, we conclude that

$$TV_d(U^1) \leq TV_d(U^0)$$

. Conditions (3.37) can be relaxed to the following

$$A^0_{i,j} + B^0_{i-1,j} \leq 1/3, \quad C^0_{i,j} + D^0_{i,j-1} \leq 1/3.$$

Similarly, we can show that under the same set of constraints on the $U^{n+1} = LU^n$ scheme coefficients and a special initial condition

$$U^0_{i.j} = \begin{cases} 1, & \text{if } p_1 \leq i \leq p_2,\ q_1 \leq j \leq q_2, \\ 0, & \text{otherwise,} \end{cases} \tag{3.42}$$

for some p_1, p_2, q_1, q_2, such that $(x_{p_1}, y_{q_1}), (x_{p_1}, y_{q_2}), (x_{p_2}, y_{q_1}), (x_{p_2}, y_{q_1})$ are not lying on the boundary of the domain, we have

$$TV_d(U^1) \leq TV_d(U^0).$$

Remark 3.4.1. Note that for $TV_a(U)$ we cannot guarantee $TV_a(U^0) \leq TV_a(U^1)$ under (3.37)-(3.38). Similarly to $TV_d(U^0)$, we can compute $TV_a(U^0) = 4\Delta x$. Then, after one step of the scheme, we get

$$\begin{aligned}
TV_a(U^1) &= \Delta x \sum_{i,j} |D^1 U^1_{i,j}| + |D^2 U^1_{i,j}| \\
&= \Delta x \Big[| -1 + A_{p,q} + B_{p,q} + B_{p-1,q} + C_{p,q} + D_{p,q-1}| + |A_{p-1,q}| + | - B_{p,q}| \\
&+ |1 - A_{p-1,q} - B_{p-1,q} - A_{p,q} - C_{p,q} - D_{p,q-1}| + |D_{p,q}| + | - D_{p,q}| \\
&+ |C_{p,q-1}| + | - C_{p,q-1}| + | - 1 + C_{p,q} + D_{p,q} + D_{p,q-1} + A_{p,q} + B_{p-1,q}| \\
&+ |C_{p,q-1}| + | - D_{p,q}| + |1 - C_{p,q-1} - D_{p,q-1} - C_{p,q} - A_{p,q} - B_{p-1,q}| \\
&+ |B_{p,q}| + | - B_{p,q}| + |A_{p-1,q}| + | - A_{p-1,q}| \Big] \\
&= \Delta x [4 - 4A_{p,q} - 4B_{p-1,q} - 4C_{p,q} - 4D_{p,q-1} + 2A_{p-1,q} + 2B_{p,q} + 2D_{p,q} + 2C_{p,q-1}].
\end{aligned}$$

It is clear that $TV_a(U^1)$ can be greater than $TV_a(U^0) = 4\Delta x$, for some combination of coefficients from the given range, which means that $TV_a(U)$ may increase. Similarly, it can be shown that $TV_{is}(U)$ may increase on these initial data.

One common approach to stability of the numerical schemes is examining the spectrum of the discretization matrices L and M. The stability of the scheme is closely tied to the properties of the eigenvalues of these matrices.

We proceed with the following lemma on the spectrum of matrices L and M, which connects the TVD property with stability of numerical schemes (3.18).

Lemma 3.4.3. *Assume the following conditions on the coefficients* (3.15) *hold*

$$A^n_{i,j} + B^n_{i-1,j} < 1/4, \quad C^n_{i,j} + D^n_{i,j-1} < 1/4, \tag{3.43}$$

$$A^n_{i,j} \geq 0,\ B^n_{i-1,j} \geq 0,\ C^n_{i,j} \geq 0,\ D^n_{i,j-1} \geq 0,\ , \quad \forall i, j \in [1, N]. \tag{3.44}$$

Then the matrices L and M for this scheme are strictly row diagonally dominant, their

eigenvalues $\mu_k(L)$ and $\mu_s(M)$ have positive real parts and the following estimates hold

$$0 < |\mu_k(L)| \le 1, \quad 0 < |\mu_s(M)| \le 2, \quad 1 \le k \le S,\ 1 \le s \le 2S. \tag{3.45}$$

We write conditions on coefficients (3.43)-(3.44) for the interior points of the domain and omit conditions for the boundary points, since they are given by similar expressions.

Proof. The diagonal elements of matrix L are given by

$$|L_{ss}| = |1 - A^n_{i,j} - B^n_{i-1,j} - C^n_{i,j} - D^n_{i,j-1}|,$$

where $s = i + (j-1)N, \quad 1 < i, j < N$. By assumption (3.43), L_{ss} is positive. We observe that with (3.44) matrix L is strictly row diagonally dominant

$$\begin{aligned}|L_{ss}| - \sum_{k \ne s} |L_{sk}| &= |1 - A^n_{i,j} - B^n_{i-1,j} - C^n_{i,j} - D^n_{i,j-1}| - |A^n_{i,j}| - |B^n_{i-1,j}| - |C^n_{i,j}| - |D^n_{i,j-1}| \\ &= 1 - 2A^n_{i,j} - 2B^n_{i-1,j} - 2C^n_{i,j} - 2D^n_{i,j-1} > 1 - 2 \cdot \frac{1}{4} - 2 \cdot \frac{1}{4} > 0,\end{aligned}$$

Therefore and all eigenvalues of L have positive real parts.

We look for estimates of the smallest $\mu_{min}(L)$ and largest $\mu_{max}(L)$ in magnitude eigenvalues of L. We use Gelfand's formula

$$\rho(L) \le \|L\|_\infty$$

for the spectral radius $\rho(L) = \max_k |\mu_k(L)|$. Then we get

$$\begin{aligned}\|L\|_\infty &= \max_{1 \le k \le S} \sum_{p=1}^{S} |L_{kp}| \\ &= \max_{i,j} \{1 - A^n_{i,j} - B^n_{i-1,j} - C^n_{i,j} - D^n_{i,j-1} + A^n_{i,j} + B^n_{i-1,j} + C^n_{i,j} + D^n_{i,j-1}\} = 1.\end{aligned}$$

Then $\rho(L) \le 1$ and we conclude that $0 < |\mu_k(L)| \le 1, \ \ k \in [1, S]$.

Similarly for M we have

$$|M_{ss}| = |1 - A^n_{i,j} - B^n_{i,j}| = 1 - A^n_{i,j} - B^n_{i,j},$$

for $s = i + (j-1)N$ and

$$|M_{ss}| = |1 - C^n_{i,j} - D^n_{i,j}| = 1 - C^n_{i,j} - D^n_{i,j},$$

for $s = S + i + (j-1)N$. Then by assumptions (3.43)-(3.44) we have

$$\begin{aligned}|M_{ss}| - \sum_{l \ne s} |M_{sl}| &= (1 - A^n_{i,j} - B^n_{i,j}) - B^n_{i-1,j} - A^n_{i+1,j} - D^n_{i,j-1} - D^n_{i+1,j-1} - C^n_{i,j} - C^n_{i+1,j} \\ &= 1 - (A^n_{i,j} + B^n_{i-1,j}) - (A^n_{i+1,j} + B^n_{i,j}) - (C^n_{i,j} + D^n_{i,j-1}) - (C^n_{i+1,j} + D^n_{i+1,j-1}) > 0,\end{aligned}$$

for $s = i + (j-1)N$ and

$$|M_{ss}| - \sum_{l \neq s} |M_{sl}| = (1 - C^n_{i,j} - D^n_{i,j}) - B^n_{i-1,j} - A^n_{i,j} - B^n_{i-1,j+1} - A^n_{i,j+1} - D^n_{i,j-1} - C^n_{i,j+1}$$
$$= 1 - (C^n_{i,j} + D^n_{i,j-1}) - (A^n_{i,j} + B^n_{i-1,j}) - (A^n_{i,j+1} + B^n_{i-1,j+1}) - (C^n_{i,j+1} + D^n_{i,j}) > 0,$$

for $s = S + i + (j-1)N$. Thus, M is strictly row dominant, and all of its eigenvalues have positive real parts.

We now look for the magnitude of the smallest $\mu_{min}(M)$ and largest $\mu_{max}(M)$ eigenvalues of M. We use another variant of Gelfand's formula

$$\rho(M) \leq \|M\|_1$$

to obtain the following estimate

$$\|M\|_1 = \max_{1 \leq l \leq 2S} \sum_{s=1}^{2S} |M_{sl}| = \max_{i,j} \{1 - A^n_{i,j} - B^n_{i,j} + A^n_{i,j} + B^n_{i,j} + A^n_{i,j} + B^n_{i,j} + A^n_{i,j} + B^n_{i,j}\}$$
$$= \max_{i,j} \{1 + 2A^n_{i,j} + 2B^n_{i,j}\} \leq 2,$$

for $1 \leq l \leq S$ and

$$\|M\|_1 = \max_{1 \leq l \leq 2S} \sum_{s=1}^{2S} |M_{sl}| = \max_{i,j} \{1 - C^n_{i,j} - D^n_{i,j} + C^n_{i,j} + D^n_{i,j} + C^n_{i,j} + D^n_{i,j} + C^n_{i,j} + D^n_{i,j}\}$$
$$= \max_{i,j} \{1 + 2C^n_{i,j} + 2D^n_{i,j}\} \leq 2,$$

for $l > S$.

Therefore $\rho(M) \leq 2$. We conclude that $0 < |\mu_s(M)| \leq 2,\ s \in [1, 2S]$. □

Remark 3.4.2. Under the conditions of Lemma 3.4.3 matrices L and M are invertible.

Lemma 3.4.4. *Assume the following conditions on the coefficients in (3.15) hold*

$$A^n_{i,j} + B^n_{i,j} < 1/4, \quad C^n_{i,j} + D^n_{i,j} < 1/4, \tag{3.46}$$
$$A^n_{i,j} \geq 0,\ B^n_{i-1,j} \geq 0,\ C^n_{i,j} \geq 0,\ D^n_{i,j-1} \geq 0. \tag{3.47}$$

Then the matrices L^T and M^T are strictly row diagonally dominant, their eigenvalues $\mu_k(L^T)$ and $\mu_s(M^T)$ have positive real parts and the following estimates hold

$$0 < |\mu_k(L^T)| \leq 1, \quad 0 < |\mu_s(M^T)| \leq \frac{3}{2}, \quad 1 \leq k \leq S,\ 1 \leq s \leq 2S. \tag{3.48}$$

The proof is similar to that of Lemma 3.4.4. The conditions (3.46)-(3.47) resemble those of Harten's lemma (3.5).

Remark 3.4.3. The necessary TVD conditions we use in Lemma 3.4.3 and 3.4.4 are motivated by the relations on the coefficients of the KT scheme [81], a second order MUSCL scheme, for which it can be shown that $A^n_{i,j} + B^n_{i-1,j} \leq 15/64, \quad C^n_{i,j} + D^n_{i,j-1} \leq 15/64, \quad \forall n.$

3.5 Numerical experiments

In this section, we present numerical experiments to verify the TVD property of numerical solutions of scalar conservation laws using the KT scheme. We compute TV using TV_a, TV_{is} and TV_d discrete definitions. TV_d was computed using Algorithm 1, that is described in Chapter 4, with tolerance $\varepsilon = 10^{-6}$.

All problems are solved on a square domain $\Omega = [-2, 2] \times [-2, 2]$ discretized into square $N \times N$ meshes with elements $\Omega_{i,j}$. The initial condition U^0 is computed by averaging of $u_0(x, y)$ in each cell.

A five-point scheme with random coefficients.

First, consider schemes of the form (3.15) with randomly generated and limited coefficients.

We use

$$u_0(x, y) = \begin{cases} 1, & \text{if } x \in [-1, 1], \quad y \in [-1, 1], \\ 0, & \text{otherwise}, \end{cases} \tag{3.49}$$

as an initial condition. We look for restrictions on the scheme coefficients $A^n_{i,j}, B^n_{i-1,j}, C^n_{i,j}$, and $D^n_{i,j-1}$ that would not result in increase of $TV_d(U)$ after one time step. We seek constraints α, β on the coefficients such that

$$A^0_{i,j} + B^0_{i-1,j} \le \alpha, \quad C^0_{i,j} + D^0_{i,j-1} \le \alpha, \tag{3.50}$$

or

$$A^0_{i,j} + B^0_{i-1,j} + C^0_{i,j} + D^0_{i,j-1} \le \beta, \tag{3.51}$$

We also require the coefficients to be non-negative. The possible bounds are given by the time step restrictions of the DG and KT methods. For KT scheme, it follows from the CFL condition (3.57) that (3.50) is satisfied with $\alpha = 15/64$. This and the results of Section 3.4 (Lemma 3.4.3, 3.4.4) motivate using $\alpha = 1/4$. To check this hypothesis, we drew $S = 10^6$, $N \times N$ matrices A, B, C, D from the uniform multivariate distribution using the rand$(\cdot)$ function in MATLAB. Then, we scaled the entries of the matrices so that the conditions (3.50) are satisfied. i.e.

$$A^0_{i,j} + B^0_{i-1,j} = \alpha, \quad C^0_{i,j} + D^0_{i,j-1} = \alpha, \ \forall i, j.$$

Numerical experiments with different values of N were performed. We report here result of computations on a coarse grid, as large cell size results in more spreading and therefore larger changes in TV. In the presented examples we used $N = 20$. The initial condition in (3.49) was chosen to be a discontinuous function because TV of the discontinuous shape after one step of a scheme would change more than that of a smooth initial condition. Random coefficients represent many possible combinations of numerical fluxes $f(u)$ and $g(u)$ in (3.1).

We compute U^1 using (3.15) for each sample set of scaled matrices A, B, C, D. We then run Algorithm 1 to compute $TV_d(U^0)$ and $TV_d(U^1)$ and compare these two values (Figure 3.2 (left)). We report that the largest value of α for which the condition $TV_d(U^1) \le TV_d(U^0)$ is satisfied for all 10^6 random sets of scheme coefficients is $\alpha \le 1/4$. We note that both TV_a and TV_{is} fail this test. We also note that higher values of α result in growth of all three TVs.

Then we repeat the tests on the same set of matrices A, B, C, D, which we scale to satisfy restrictions (3.51) with $\beta = 1/3$, i.e.

$$A^0_{i,j} + B^0_{i-1,j} + C^0_{i,j} + D^0_{i,j-1} = \beta, \ \forall i, j.$$

We conduct the numerical experiments and report the value of $\delta TV_d^{1,0} = TV_d(U^1) - TV_d(U^0)$ for all 10^6 sets of scaled scheme coefficients in Figure 3.2 (right).

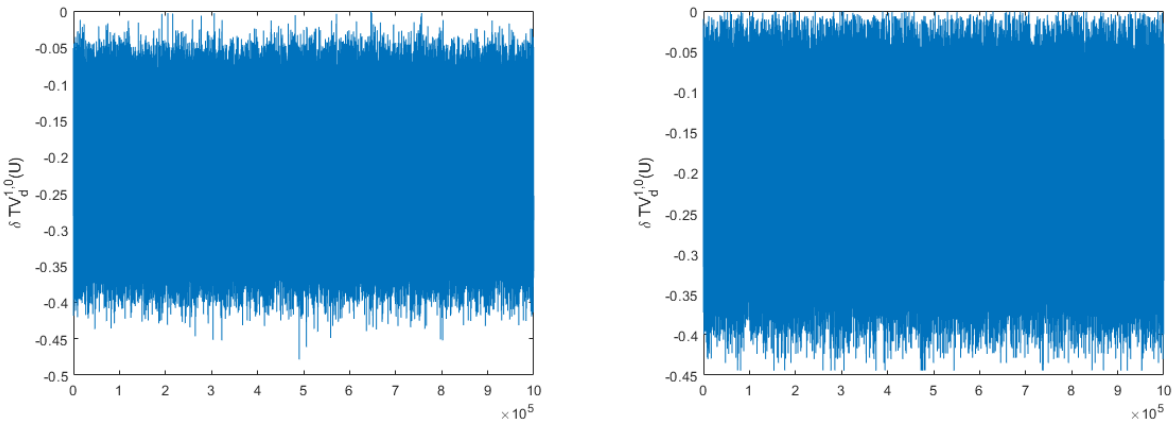


Figure 3.2: $\delta TV_d^{1,0}(U) = TV_d(U^1) - TV_d(U^0)$ for $S = 10^6$ cases under conditions (3.50), $\alpha = 1/4$ (left) and (3.51), $\beta = 1/3$ (right).

Thus, we hypothesize that (3.50) and (3.51) with $\alpha = 1/4$ and $\beta = 1/3$, respectively, are good candidates for possible sufficient conditions for a scheme to be TVD in the TV_d sense.

Next, we solve (3.1) with KT scheme and compute TV of the obtained solution using three definitions.

TV of numerical solutions for Kurganov-Tadmor scheme.

We apply a second-order central finite-difference scheme of Kurganov and Tadmor with MUSCL flux [82] to (1.6)

$$U^{n+1}_{i,j} = U^n_{i,j} - \frac{\Delta t^n}{\Delta x}\left(H^x_{i+1/2,j}(t^n) - H^x_{i-1/2,j}(t^n)\right) - \frac{\Delta t^n}{\Delta y}\left(H^y_{i,j+1/2}(t^n) - H^y_{i,j-1/2}(t^n)\right), \quad (3.52)$$

where $H^x(t), H^y(t)$ are numerical fluxes given by

$$H^x_{i+1/2,j}(t) = \frac{f(U^+_{i+1/2,j}(t)) + f(U^-_{i+1/2,j}(t))}{2} - \frac{a^x_{i+1/2,j}(t)}{2}\left(U^+_{i+1/2,j}(t) - U^-_{i+1/2,j}(t)\right), \quad (3.53)$$

$$H^y_{i,j+1/2}(t) = \frac{g(U^+_{i,j+1/2}(t)) + g(U^-_{i,j+1/2}(t))}{2} - \frac{a^y_{i,j+1/2}(t)}{2}\left(U^+_{i,j+1/2}(t) - U^-_{i,j+1/2}(t)\right). \quad (3.54)$$

Here, $a^x_{i+1/2,j}(t), a^y_{i,j+1/2}(t)$ are lthe ocal wave speeds given by (4.19) in [82], $U^\pm_{i+1/2,j}(t), U^\pm_{i,j+1/2}(t)$ are linearly reconstructed values at cell edge midpoints

$$U^\pm_{i+1/2,j}(t) = U_{i+1/2\pm1/2,j} \mp \frac{\Delta x}{2}(U_x)^n_{i+1/2\pm1/2,j}(t),$$
$$U^\pm_{i,j+1/2}(t) = U_{i,j+1/2\pm1/2} \mp \frac{\Delta y}{2}(U_y)^n_{i,j+1/2\pm1/2}(t).$$

The slopes of the numerical solution are computed using the MC function (3.12) as

$$(U_x)^n_{i,j}(t) = minmod\left(r\frac{U^n_{i+1,j}-U^n_{i,j}}{\Delta x}, \frac{U^n_{i+1,j}-U^n_{i-1,j}}{2\Delta x}, r\frac{U^n_{i,j}-U^n_{i-1,j}}{\Delta x}\right), \tag{3.55}$$

$$(U_y)^n_{i,j}(t) = minmod\left(r\frac{U^n_{i,j+1}-U^n_{i,j}}{\Delta y}, \frac{U^n_{i,j+1}-U^n_{i,j-1}}{2\Delta y}, r\frac{U^n_{i,j}-U^n_{i,j-1}}{\Delta y}\right). \tag{3.56}$$

The two-dimensional scheme considered here satisfies the local maximum principle for $1 \le r \le 2$, under the following CFL condition

$$\max\left(\frac{\Delta t^n}{\Delta x}\max_u|f'(u)|, \frac{\Delta t^n}{\Delta y}\max_u|g'(u)|\right) \le \frac{1}{8}. \tag{3.57}$$

We set $r = 1.5$ in the numerical examples below. We use the classical second-order Runge-Kutta scheme for time integration with (3.52). A detailed derivation of the algorithm and its analysis can be found in [82].

Example 1 (Rotation of a square pulse).
We consider the rotation of a square pulse around the origin described by

$$u_t + 2\pi y u_x - 2\pi x u_y = 0, \tag{3.58}$$

and

$$u_0(x,y) = \begin{cases} 1, & \text{if } x \in [-0.5, 0.5], \quad y \in [-0.5, 0.5], \\ 0, & \text{otherwise.} \end{cases} \tag{3.59}$$

with outflow boundary conditions.

We solve the problem for $t \in [0,1]$ on $N = 40, 80, 160$ meshes. We plot the values of TV_a, TV_{is} and TV_d as a function of time in Figure 3.4. TV at selected times are also reported in Table 3.1.

TV of the exact solution does not change in time. However, we observe that the TV_a do not decrease with time. In fact, there are intervals with marked increases in TV. The plot has a scallop-like shape with peaks at $t = 0.25, 0.5, \ldots$ corresponding to rotation by $\pi/4$, $3\pi/4$, etc. and troughs corresponding to rotation by $\pi/2$, π, etc., see Figure 3.5 (bottom right). This demonstrates the behavior of TV_a discussed in Chapter 2. That is, the computed TV_a value at the first peak is greater than that at the initial moment by a factor approaching $\sqrt{2}$, as we observe an increase in peak values with mesh refinement. The decrease in peak values with time can be attributed to the numerical diffusion of the scheme and the spreading of the solution.

The plot of TV_{is} shows similar behavior. The value of TV_{is} grows on the interval $[0, 0.06]$ and diminishes after $t = 0.06$ for $N = 40, 80$. Thus, the scheme is not TVD in the TV_a and TV_{is} sense. Finally, the computed dual TV forms a monotonically decreasing sequence for all meshes. We observe that the limited solution of (3.58) is oscillation-free (Figure 3.3) and is TVD in the TV_d sense (Figure 3.4). Finally, we compute the L^1,L^∞ errors at $T = 1$ using the exact solution, i.e a rotated square pulse. and report the convergence rate of the error in Table 3.3.

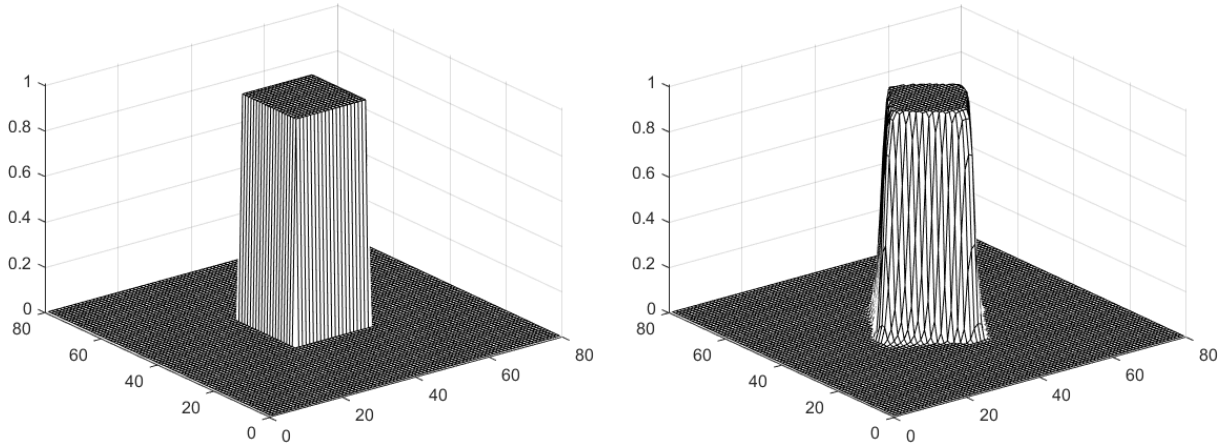


Figure 3.3: Initial condition (left) and numerical solution (right) of (3.58)-(3.59) on $N = 80$ mesh, at $t = 0.125$.

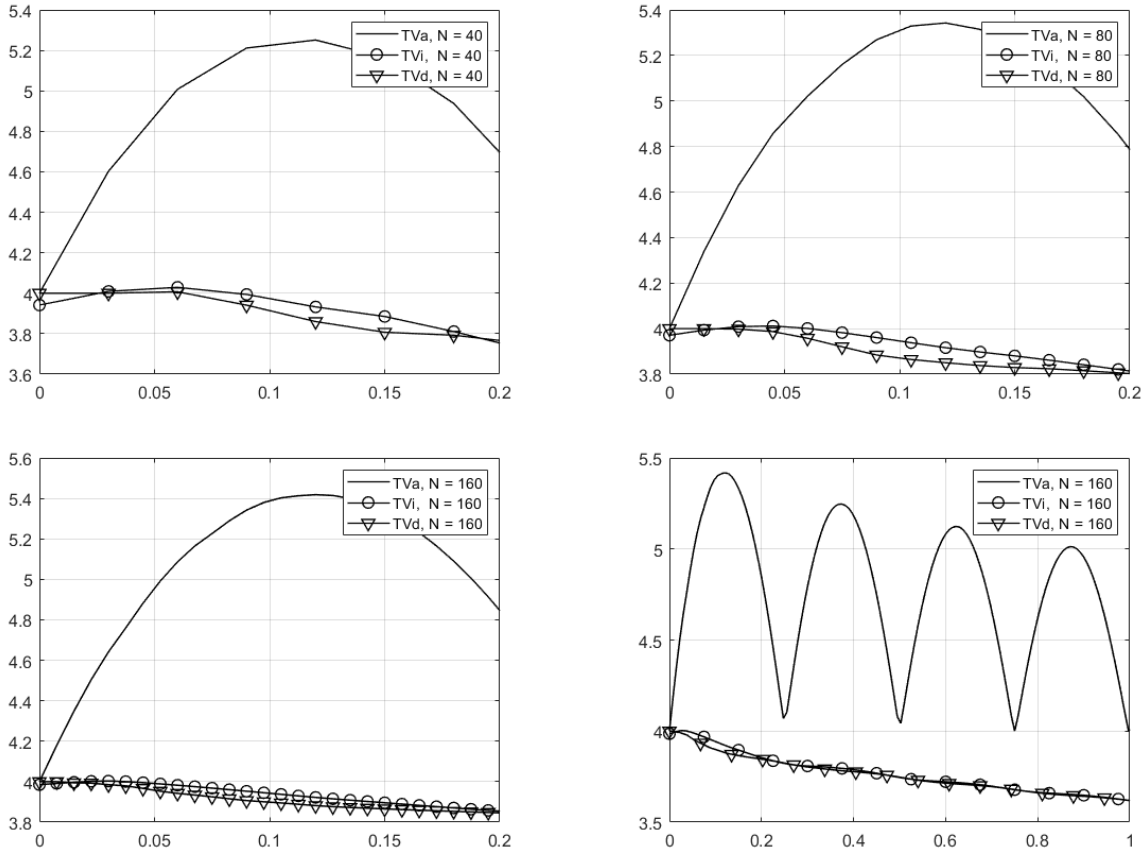


Figure 3.4: TV of the solutions of (3.58),(3.59) on $N = 40, 80, 160$ meshes, for $t \in [0, 0.2]$. The lower right figure shows all three TVs computed on $N = 160$ mesh for $t \in [0, 1]$. TVi stands for isotropic TV.

	$N = 40$				$N = 80$				$N = 160$		
t	TV_a	TV_{is}	TV_d	t	TV_a	TV_{is}	TV_d	t	TV_a	TV_{is}	TV_d
0	4.0000	3.9414	4.0000	0	4.0000	3.9707	4.0000	0	4.0000	3.9854	4.0000
0.0600	5.0077	4.0291	3.9999	0.0600	5.0209	4.0010	3.9577	0.0600	5.0863	3.9825	3.9424
0.1200	5.2513	3.9323	3.8598	0.1200	5.3427	3.9161	3.8501	0.1200	5.4183	3.9223	3.8818
0.1800	4.9376	3.8106	3.7924	0.1800	5.0187	3.8414	3.8150	0.1800	5.0899	3.8702	3.8534
0.2400	4.0994	3.6567	3.7050	0.2400	4.1814	3.7620	3.7741	0.2400	4.1988	3.8279	3.8295
0.3000	4.5139	3.5989	3.6090	0.3000	4.6848	3.7245	3.7220	0.3000	4.7931	3.8082	3.8031
0.3600	4.8146	3.5745	3.5522	0.3600	5.0573	3.7091	3.6889	0.3600	5.2351	3.7987	3.7859
0.4201	4.6175	3.5258	3.5219	0.4200	4.8880	3.6810	3.6699	0.4200	5.0566	3.7811	3.7728
0.4803	4.0972	3.4628	3.4801	0.4800	4.2326	3.6367	3.6430	0.4800	4.3229	3.7546	3.7553
0.5406	4.2459	3.4219	3.4360	0.5400	4.4412	3.6029	3.6052	0.5400	4.5799	3.7318	3.7303
0.6010	4.5127	3.3977	3.3963	0.6000	4.8472	3.5870	3.5777	0.6000	5.0815	3.7216	3.7143
0.6618	4.4074	3.3645	3.3657	0.6600	4.7630	3.5676	3.5598	0.6600	5.0118	3.7099	3.7036
0.7227	4.0439	3.3224	3.3370	0.7200	4.2643	3.5374	3.5402	0.7200	4.4099	3.6898	3.6893
0.7841	4.0628	3.2889	3.3026	0.7800	4.2465	3.5096	3.5125	0.7800	4.3881	3.6683	3.6678
0.8458	4.2818	3.2643	3.2713	0.8400	4.6545	3.4936	3.4892	0.8400	4.9319	3.6571	3.6520
0.9081	4.2266	3.2355	3.2420	0.9001	4.6570	3.4779	3.4733	0.9000	4.9555	3.6476	3.6421
0.9708	3.9436	3.2012	3.2150	0.9902	4.0144	3.4429	3.4459	0.9900	4.0927	3.6219	3.6218

Table 3.1: Total variation values for Example 1 on $N = 40, 80, 160$ meshes.

Example 2 (Burgers' equation).
We consider the inviscid Burgers' equation

$$u_t + uu_x + uu_y = 0 \tag{3.60}$$

with the initial condition

$$u_0(x,y) = \begin{cases} 1, & \text{if } x \in [-1,0], \quad y \in [-1,0], \\ 0, & \text{otherwise.} \end{cases} \tag{3.61}$$

We solve the problem for $t \in [0, 0.5]$ on $N = 40, 80, 160$ meshes. We plot TV_a, TV_{is} and TV_d in Figure 3.5. TV values at selected times are presented in Table 3.2. We compute the L^1,L^∞ errors at $T = 0.5$ using the numerical solution on the $N = 1280$ mesh as ground truth and report the convergence rate of the error in Table 3.3.

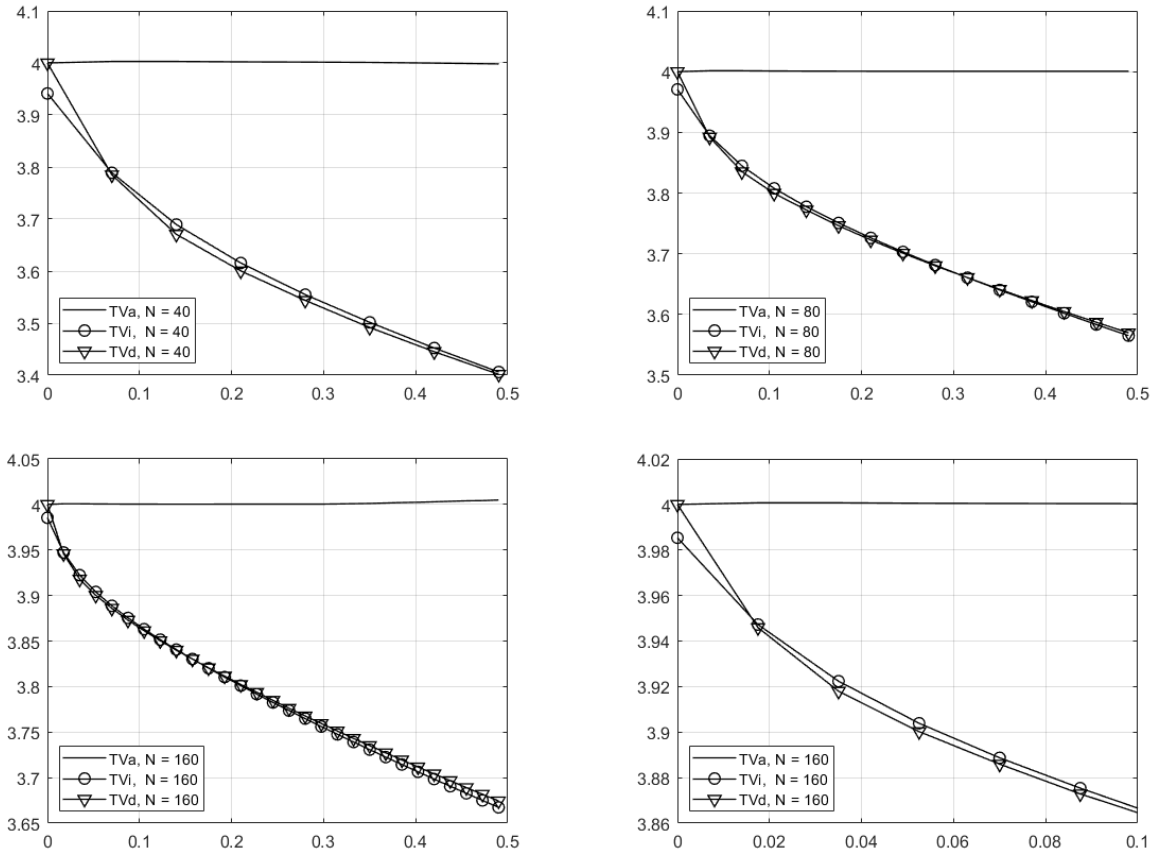


Figure 3.5: TV of the solutions of (3.60),(3.61) on $N = 40, 80, 160$ meshes, for $t \in [0, 0.5]$ The lower right figure shows the three TVs computed on $N = 160$ mesh for $t \in [0, 0.1]$.

	$N = 40$				$N = 80$				$N = 160$		
t	TV_a	TV_{is}	TV_d	t	TV_a	TV_{is}	TV_d	t	TV_a	TV_{is}	TV_d
0	4.0000	3.9414	4.0000	0	4.0000	3.9707	4.0000	0	4.0000	3.9853	4.0000
0.0700	4.0029	3.7889	3.7848	0.07	4.0014	3.8447	3.8351	0.07	4.0004	3.8887	3.8859
0.1400	4.0029	3.6894	3.6707	0.14	4.0009	3.7774	3.7720	0.14	4.0003	3.8406	3.8399
0.2100	4.0021	3.6155	3.6003	0.21	4.0007	3.7261	3.7226	0.21	4.0003	3.8010	3.8023
0.2800	4.0017	3.5547	3.5437	0.28	4.0006	3.6813	3.6801	0.28	4.0004	3.7648	3.7676
0.3500	4.0010	3.5015	3.4916	0.35	4.0006	3.6403	3.6414	0.35	4.0011	3.7307	3.7352
0.4201	3.9997	3.4521	3.4451	0.42	4.0007	3.6020	3.6048	0.42	4.0029	3.6984	3.7041
0.4903	3.9981	3.4060	3.4015	0.49	4.0007	3.5650	3.5696	0.49	4.0049	3.6673	3.6744

Table 3.2: Total variation for the limited solutions for Example 2 on $N = 40, 80, 160$ meshes.

	Example 1				Example 2			
N	L^1-err	Rate	L^∞-err	Rate	L^1-err	Rate	L^∞-err	Rate
40	1.293e-02	–	2.514e-02	–	7.501e-02	–	6.3422e-02	–
80	3.079e-03	2.07	7.955e-03	1.66	1.632e-02	2.20	2.196e-02	1.53
160	8.081e-04	1.93	2.642e-03	1.59	3.703e-03	2.14	7.244e-03	1.60
320	1.978e-04	2.03	8.595e-04	1.62	8.228e-04	2.17	2.597e-03	1.48
640	4.582e-05	2.11	2.955e-04	1.54	1.987e-04	2.05	8.746e-04	1.57

Table 3.3: Convergence rates of the KT solutions at $T = 1$ for Examples 1 and $T = 0.5$ for Example 2.

Similarly to Example 1, we observe (Table 3.2) that TV_d monotonically decreases with time while TV_a does not. The growth of TV_a is mainly due to bad approximation of solution gradients on the edges of the pulse. In contrast to the previous example, the TV_{is} values do not grow with time and mesh refinement. Additionally, the refinement process does not reduce the initial increase in the TV_a value.

3.6 Results

We study a five-point scheme for two-dimensional scalar conservation laws and propose a set of conditions on scheme coefficients that guarantee several desirable properties and are a good candidate for imposing TV-stability in the dual discrete TV sense.

We use the proposed set of conditions both analytically and numerically. We test conditions (3.50) as well as an independent set of conditions (3.51) to limit the solution of the five-point scheme with random coefficients. We demonstrate numerically that these conditions deliver a solution with non-increasing dual discrete total variation after one step of the scheme on a statistically large set of data. Notably, these constraints are closely related to the CFL conditions of KT and DG schemes. Consequently, we can conclude that less dissipative second-order schemes, and potentially higher-order schemes, which are TVD in the dual discrete sense, should exist.

Finally, we have presented numerical evidence that the KT scheme (a second-order order fully-discrete scheme) exhibits the TVD property in the dual discrete TV sense. These results hold when measuring TV using definition (2.39). Hence, we provide an example of a scheme that is both TVD and second-order accurate.

3.7 Summary

In this chapter, we have reviewed the approach taken by Harten to ensure the stability of high-order nonlinear schemes for hyperbolic conservation laws in one spatial dimension and discussed the MUSCL limiting technique to construct such schemes. We have described the main features of the limiters commonly used to enforce the TVD property in one spatial dimension.

We adopt a new definition for discrete TV to be used as a criterion to be checked for the TVD property of a numerical scheme in two dimensions. We formulate the necessary conditions on the scheme coefficients to guarantee the TVD property in the dual discrete sense for a five-point scheme. We first show the TVD property in the new sense for one-dimensional data and a two-dimensional scheme and then for a special initial data.

In the numerical experiments section, we study the TVD property in different definitions under the proposed conditions on the coefficients. We consider several scalar conservation laws and use a KT scheme that is of second order and is not TVD in the anisotropic TV sense. We demonstrate in these experiments that the second order scheme can be of second order and TVD in the sense of the dual discrete TV, while the anisotropic and isotropic TVs increase. This chapter is a crucial part of the thesis, containing the most interesting results, challenging the long-standing assertion of Goodman and LeVeque, which can have an extensive impact on future research endeavors.

Chapter 4

A primal-dual algorithm for computing dual discrete total variation

While gradient-based TV discretizations, such as (2.26) and (2.13), are given by explicit formulas that are easy and straightforward to implement, the dual TV discretization requires solving an associated optimization problem.

We begin by introducing the terms in which the optimization problem for dual discrete TV (2.41) can be expressed.

Fenchel's duality theorem.

For functions $f : \mathbb{R}^N \to \mathbb{R}$, the Legendre–Fenchel transform is defined by

$$f^*(m) = \sup_{x \in \mathbb{R}^N} \{\langle m, x\rangle - f(x)\}, \tag{4.1}$$

where $m \in \mathbb{R}^N$ and $\langle m, x\rangle$ denotes the dot product of m and x in $\mathbb{R}^N$. The domain of f^* is

$$X^* = \left\{ x^* \in \mathbb{R}^N : \sup_{x \in \mathbb{R}^N} (\langle x^*, x\rangle - f(x)) < \infty \right\}. \tag{4.2}$$

We state the Fenchel's duality theorem for functions and then use it to derive a numerical procedure to find the dual total variation numerically.

Theorem 4.0.1 (Fenchel's duality theorem [94])**.** *Let X and Y be Banach spaces, $f : X \to \mathbb{R}$ and $g : Y \to \mathbb{R}$ be convex functions and $A : X \to Y$ be a bounded linear operator. Then the Fenchel's primal*

$$p^* = \inf_{x \in X} \{f(x) + g(Ax)\}$$

and dual

$$d^* = \sup_{y^* \in Y^*} \{-f^*(A^* y^*) - g^*(-y^*)\}$$

problems satisfy weak duality, i.e., $p^* \geq d^*$. Above Y^* is the domain of g^* and A^* is the adjoint operator for A, i.e $\langle Ax, y\rangle = \langle x, A^* y\rangle$.

If f, g, and A are such that they satisfy one of the conditions

(a) *f and g are lower semi-continuous, and 0 is an element of the algebraic interioir of $(\operatorname{dom} g - A \operatorname{dom} f)$, where $\operatorname{dom} f = \{x : f(x) < \infty\}$, or*

(b) *$A \operatorname{dom} f \cap \operatorname{cont} g \neq \emptyset$, where $\operatorname{cont} g$ denotes the points where the function g is continuous.*

Then the strong duality holds, i.e., $p^* = d^*$.

If additionally d^* is finite, then the supremum is attained. In finite-dimensional spaces, we can replace supremum and infimum with maximum and minimum.

The Fenchel's duality theorem guarantees that the optimal value of the dual problem provides a lower bound on the optimal value of the primal problem (the original minimization problem). Similarly, the primal objective function evaluated at a feasible point provides an upper bound on the optimal value. Iterative algorithms can be designed that solve both the primal and dual problems simultaneously. As the solutions progress, the lower bound and upper bound converge towards the true optimal value.

4.1 Primal-dual formulation

Computing the maximizer $\boldsymbol{\varphi}$ of the constrained optimization problem (2.41) with conventional optimization techniques is a difficult task. For this reason we derive an equivalent saddle-point formulation [113]. We will use Fenchel-Rockafellar duality to state the primal-dual problem.

We will first introduce $\boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3) \in \mathbb{R}^{3\times 2\times N\times N}$, with $\boldsymbol{v}_k = (v_k^x, v_k^y) \in \mathbb{R}^{2\times N\times N}$, $k = 1, 2, 3$, that approximates ∇u on grid of points $(x_{i+1/2}, y_j)$, $(x_i, y_{j+1/2})$ and (x_i, y_j), i.e. on the grid with twice as many points as the original discretization of Ω (Figure 4.1). Let $\boldsymbol{v} \in \boldsymbol{V}$, where

$$\boldsymbol{V} = \{\boldsymbol{v} : \|\boldsymbol{v}\|_{1,1,2} < +\infty\}, \tag{4.3}$$

$$\|\boldsymbol{v}\|_{1,1,2} = \sum_k \|\boldsymbol{v}_k\|_{1,2}, \;\; \|\boldsymbol{v}_k\|_{1,2} = \sum_{i,j} \|(\boldsymbol{v}_k)_{i,j}\|_2 = \sum_{i,j} \sqrt{(v_k^x)_{i,j}^2 + (v_k^y)_{i,j}^2}. \tag{4.4}$$

We set $\boldsymbol{v} = \boldsymbol{0}$ at the boundary of the domain.

We use the notation of [40] and define the coarsening operator $\boldsymbol{F} : \boldsymbol{V} \to \mathbb{R}^{2\times N\times N}$: $\boldsymbol{F}\boldsymbol{v}_{i,j} = ((F^1\boldsymbol{v})_{i,j}, (F^2\boldsymbol{v})_{i,j})^T$, with components

$$(F^1\boldsymbol{v})_{i,j} = \left((v_1^x)_{i+1/2,j} + \frac{(v_2^x)_{i,j+1/2} + (v_2^x)_{i,j-1/2} + (v_2^x)_{i+1,j+1/2} + (v_2^x)_{i+1,j-1/2}}{4} + \frac{(v_3^x)_{i,j} + (v_3^x)_{i+1,j}}{2}\right), \tag{4.5}$$

$$(F^2\boldsymbol{v})_{i,j} = \left((v_2^y)_{i+1/2,j} + \frac{(v_1^y)_{i+1/2,j} + (v_1^y)_{i+1/2,j+1} + (v_1^y)_{i-1/2,j} + (v_1^y)_{i-1/2,j+1}}{4} + \frac{(v_3^y)_{i,j} + (v_3^y)_{i,j+1}}{2}\right). \tag{4.6}$$

The first component of $\boldsymbol{F}\boldsymbol{v}_{i,j}$, i.e. $(F^1\boldsymbol{v})_{i,j}$ is assigned to midpoints $(x_{i+1/2}, y_j)$ of vertical edges and the second component $(F^2\boldsymbol{v})_{i,j}$ to midpoints $(x_i, y_{j+1/2})$ of horizontal edges.

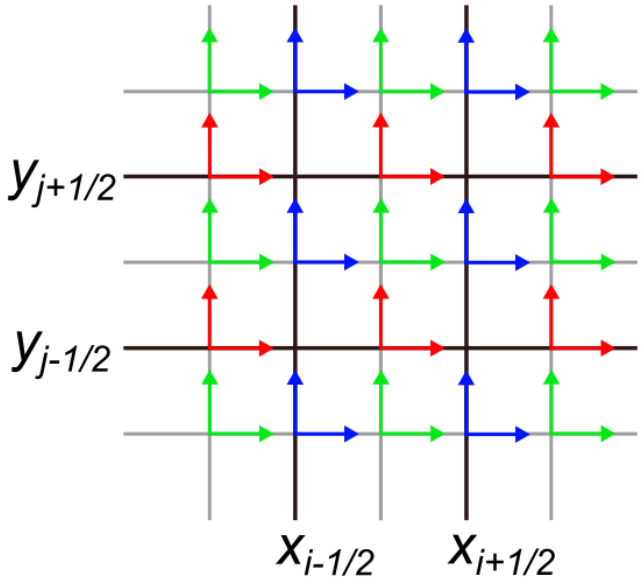


Figure 4.1: The grid of cells $\Omega_{i,j} = [x_{i-1/2}, x_{i+1/2}] \times [y_{j-1/2}, y_{j+1/2}]$ (black) and a twice finer grid for components of $\boldsymbol{v}$. $\boldsymbol{v}_1$ are shown in blue, $\boldsymbol{v}_2$ in blue and $\boldsymbol{v}_3$ in green.

The operator $\boldsymbol{F}$ is a projection of $\boldsymbol{v}$ onto a coarser grid of edge midpoints (Figure 2.6). As with $\boldsymbol{D}U$, we will construct a single vector $\boldsymbol{F}\boldsymbol{v}$ via concatenation

$$\begin{aligned}\boldsymbol{F}\boldsymbol{v} = \Big(&(F^1\boldsymbol{v})_{1,1}, (F^1\boldsymbol{v})_{2,1}, \dots, (F^1\boldsymbol{v})_{N-1,1}, (F^1\boldsymbol{v})_{1,2}, (F^1\boldsymbol{v})_{2,2}, \dots, (F^1\boldsymbol{v})_{N-1,2}, \dots, (F^1\boldsymbol{v})_{1,N-1},\\ &(F^1\boldsymbol{v})_{2,N-1}, \dots, (F^1\boldsymbol{v})_{N-1,N-1}, (F^2\boldsymbol{v})_{1,1}, (F^2\boldsymbol{v})_{2,1}, \dots, (F^2\boldsymbol{v})_{N-1,1}, (F^2\boldsymbol{v})_{1,2},\\ &(F^2\boldsymbol{v})_{2,2}, \dots, (F^2\boldsymbol{v})_{N-1,2}, \dots, (F^2\boldsymbol{v})_{1,N-1}, (F^2\boldsymbol{v})_{2,N-1}, \dots, (F^2\boldsymbol{v})_{N-1,N-1}, \Big)^T.\end{aligned}$$

We begin with the TV definition (2.36)-(2.41) and rewrite it using an objective $\mathcal{L}(\boldsymbol{v}, \boldsymbol{\varphi})$

$$\begin{aligned}TV_d(U) &= \max_{\boldsymbol{\varphi}\in\mathcal{P}} \Delta x \sum_{i,j} \langle\, \boldsymbol{D}U_{i,j}, \boldsymbol{\varphi}_{i,j}\rangle\\ &= \max_{\boldsymbol{\varphi}\in\mathcal{P}} \min_{\boldsymbol{v}\in\boldsymbol{V}} \mathcal{L}(\boldsymbol{v}, \boldsymbol{\varphi}) = \max_{\boldsymbol{\varphi}\in\mathcal{P}} \min_{\boldsymbol{v}\in\boldsymbol{V}} \left\{ \Delta x \sum_{i,j} \langle\, \boldsymbol{D}U_{i,j}, \boldsymbol{\varphi}_{i,j}\rangle - \langle \boldsymbol{v}, \boldsymbol{P}\boldsymbol{\varphi}\rangle_{\boldsymbol{V}} + \|\boldsymbol{v}\|_{1,1,2} \right\}\\ &\le \min_{\boldsymbol{v}\in\boldsymbol{V}} \max_{\boldsymbol{\varphi}\in\mathcal{P}} \left\{ \|\boldsymbol{v}\|_{1,1,2} + \sum_{i,j} \langle \boldsymbol{F}\boldsymbol{v}_{i,j} - \boldsymbol{D}U_{i,j}, \boldsymbol{\varphi}_{i,j}\rangle \right\}, \qquad (4.7)\end{aligned}$$

where $\langle\cdot,\cdot\rangle_{\boldsymbol{V}}$ denotes the inner product in the space $\boldsymbol{V}$ and $\boldsymbol{P}\boldsymbol{\varphi}$ is the concatenation of vectors $\boldsymbol{P}^1\boldsymbol{\varphi}, \boldsymbol{P}^2\boldsymbol{\varphi}, \boldsymbol{P}^3\boldsymbol{\varphi}$.

Observe that $\|\boldsymbol{v}\|_{1,1,2}$ in (4.7) is lower-semicontinuous and its convex conjugate is the characteristic of $\mathcal{P}$, provided the last minimum is infinity if the feasible set $\{\boldsymbol{v} \,:\, \boldsymbol{F}\boldsymbol{v} = \boldsymbol{D}U\}$ is empty. Then we get

$$\min_{\boldsymbol{v}\in\boldsymbol{V}} \max_{\boldsymbol{\varphi}\in\mathcal{P}} \left\{ \|\boldsymbol{v}\|_{1,1,2} + \sum_{i,j} \langle \boldsymbol{F}\boldsymbol{v}_{i,j} - \boldsymbol{D}U_{i,j}, \boldsymbol{\varphi}_{i,j}\rangle \right\} = \min_{\boldsymbol{v} \,:\, \boldsymbol{F}\boldsymbol{v} = \boldsymbol{D}U} \|\boldsymbol{v}\|_{1,1,2}. \qquad (4.8)$$

Hence, the discrete dual TV can be written as a minimization problem

$$TV_d(U) = \min_{\boldsymbol{v}\in\boldsymbol{V}} \Big\{ \Delta x \|\boldsymbol{v}\|_{1,1,2} \ : \ \boldsymbol{F}\boldsymbol{v} = \boldsymbol{D}U \Big\}. \tag{4.9}$$

Note that $\boldsymbol{v}$ is then an approximation of the gradient of u on the twice finer grid, see Chapter 2 for the meaning of u. By convention, the statement of the optimization problem (4.9) is called primal, and (2.41) is its dual problem.

There many ways to define projection operators $\boldsymbol{P}^k$ and $\boldsymbol{F}$. A particular choice of $\boldsymbol{P}^k$ and $\boldsymbol{F}$ made here, i.e. (2.36)-(2.38), and (4.5)-(4.6), allows us to establish consistency in the sense of Γ-convergence with the exact TV (2.11), see Theorem 2.4 of [28].

The strong duality between (2.41) and (4.9) holds, see Proposition 1 of [40] for a proof based on the Fenchel's duality theorem. That is, if $\boldsymbol{\varphi}^\dagger$ is a maximizer of the primal problem and $\boldsymbol{v}^\dagger$ is a minimizer of the dual problem, then we have

$$TV_d(U) = \Delta x \sum_{i,j} \langle\, \boldsymbol{D}U_{i,j}, \boldsymbol{\varphi}^\dagger_{i,j} \rangle = \Delta x \|\boldsymbol{v}^\dagger\|_{1,1,2}. \tag{4.10}$$

A number of methods for solving (2.41),(4.9) have been proposed in the literature. In most practical applications the first-order methods are viewed as most reliable and efficient. These methods fall into four principal classes: forward-backward [53, 129], double-backward [106], Peaceman-Rachford, and Douglas-Rachford [92]. We will focus on the "Douglas-Rachford" class, which has been shown to have most general convergence properties.

In this case, a solution to the primal-dual problem (2.41), (4.9) can be computed by pointwise shrinkage operations, such as the alternating direction method of multipliers (ADMM) [43]. An augmented Lagrangian for this problem is given by

$$\mathcal{L}_\mu(\boldsymbol{v}, \boldsymbol{\varphi}) = \Delta x \|\boldsymbol{v}\|_{1,1,2} - \langle \boldsymbol{\varphi}, \boldsymbol{F}\boldsymbol{v} - \boldsymbol{D}U \rangle - \frac{1}{2\mu} \|\boldsymbol{F}\boldsymbol{v} - \boldsymbol{D}U\|_2^2,$$

where $\mu > 0$ is a penalty parameter and the ADMM iterations can be written as

$$\boldsymbol{v}^{n+1} = \arg\min_{\boldsymbol{v}} \left(\Delta x \|\boldsymbol{v}^n\|_{1,1,2} + \frac{1}{2\mu} \|\boldsymbol{F}\boldsymbol{v}^n - \boldsymbol{D}U - \mu\boldsymbol{\varphi}^n\|_2^2 \right), \tag{4.11}$$

$$\boldsymbol{\varphi}^{n+1} = \boldsymbol{\varphi}^n - (\boldsymbol{F}\boldsymbol{v}^{n+1} - \boldsymbol{D}U)/\mu. \tag{4.12}$$

with suitably chosen initial values for $\boldsymbol{v}^0$, $\boldsymbol{\varphi}^0$. For convex objective functions, it has been shown to converge globally [44, 52], estimates on the convergence rate have been derived in [13]. The use of ADMM requires one to solve a number of subproblems in (4.11) to find the next iterate $\boldsymbol{v}^{n+1}$, see [43]. This is computationally expensive. Instead, we use the approach taken in [95] and use an approximation to the general ADMM algorithm, called an alternating proximal gradient method (APGM). This method replaces the optimization problem (4.11) with proximal mapping

$$pr_{\gamma\mu}(\boldsymbol{w}_{i,j}) = \boldsymbol{w}_{i,j} - \frac{\boldsymbol{w}_{i,j}}{\max\left(\|\boldsymbol{w}_{i,j}\|_2/\gamma\mu,\ 1\right)}. \tag{4.13}$$

A particular form of the proximal mapping operator (4.13) for this problem was derived

in [40]. Then, the APGM algorithm is given by

$$\boldsymbol{v}_k^{n+1} = pr_{\gamma\mu}\left(\boldsymbol{v}_k^n - \gamma \boldsymbol{P}^k\left(\boldsymbol{D}U - \boldsymbol{F}\boldsymbol{v}^{\boldsymbol{n}} + \mu\boldsymbol{\varphi}^n\right)\right), \tag{4.14}$$

$$\boldsymbol{\varphi}^{n+1} = \boldsymbol{\varphi}_{i,j}^n - \left(\boldsymbol{F}\boldsymbol{v}^{n+1} - \boldsymbol{D}U\right)/\mu, \tag{4.15}$$

where $k = 1, 2, 3$ denotes the component of $\boldsymbol{v}^{n+1}$ and γ, μ are hyperparameters. This algorithm is shown to be convergent in [95].

We use two stopping criteria for a given tolerance ε. The distance between consecutive iterations of $\boldsymbol{v}$

$$\|\boldsymbol{v}^{n+1} - \boldsymbol{v}^n\|_{1,1,2} \leq \varepsilon,$$

and the distance between the approximate optimal values of (2.41) and (4.9)

$$\left|\|\boldsymbol{v}^{n+1}\|_{1,1,2} - \sum_{i,j}\langle\, \boldsymbol{D}U_{i,j}, \boldsymbol{\varphi}_{i,j}^n\rangle\right| \leq \varepsilon.$$

The algorithm is described below as Algorithm 1.

Algorithm 1 Modified APGM.

1: $\boldsymbol{v}^0 := ((D^1U, \boldsymbol{0}), (\boldsymbol{0}, D^2U), (\boldsymbol{0}, \boldsymbol{0}))$, $\boldsymbol{\varphi}^0 := \boldsymbol{0}$ $\mu^0 := \mu$, $n := 0$
2: **while** $\|\boldsymbol{v}^{n+1} - \boldsymbol{v}^n\|_{1,1,2} > \varepsilon$ and $\left|\|\boldsymbol{v}^{n+1}\|_{1,1,2} - \sum_{i,j}\langle\, \boldsymbol{D}U_{i,j}, \boldsymbol{\varphi}_{i,j}^n\rangle\right| > \varepsilon$ **do**
3: **for** $k := 1 \ldots 3$ **do**
4: $\boldsymbol{v}_k^{n+1} := (\boldsymbol{v}_k^n + \gamma \boldsymbol{P}^k(\boldsymbol{D}U - \boldsymbol{F}\boldsymbol{v}^n + \mu\boldsymbol{\varphi}^n))(1 - 1/[\max(|\boldsymbol{v}_k^n|/\gamma\mu, 1)])$
5: **end for**
6: $\boldsymbol{\varphi}^{n+1} := \boldsymbol{\varphi}^n + (\boldsymbol{D}U - \boldsymbol{F}\boldsymbol{v}^{n+1})/\mu^n$
7: $\mu^{n+1} := \theta\mu^n$
8: **if** $rem(n, 100) = 0$ **then**
9: $\mu^{n+1} := \mu$
10: **end if**
11: $n := n + 1$
12: **end while**

Algorithm 1 is a modification of the original APGM algorithm that uses a progressive step size μ. To speed up convergence and avoid tuning of μ, we use $\mu^{n+1} = \theta\mu^n$ with $\mu^0 = 1$, and $\theta = 0.96$. The function $rem(x, y)$ on line 8 denotes the remainder of the division of x by y. We use $\gamma = 1/3$, as suggested in [40]. The Algorithm 1 converges globally when $\gamma\|\boldsymbol{F}\|^2 < 1$ and $\mu > 0$ [28], where $\|\boldsymbol{F}\|$ is the operator norm of $\boldsymbol{F}$ and $\|\boldsymbol{F}\|^2 \leq 3$, see Lemma 3.1 of [95]. The only hyperparameter of Algorithm 1 left to choose is ε. The accuracy of discrete TV as an estimate of the value of the exact TV depends not only on the definition, but also on the mesh resolution. Choosing $\varepsilon = (\Delta x)^2$ yields sufficient accuracy for our purposes. As such a choice provides an error in computed value of the dual TV at least by an order of magnitude smaller than the error due to mesh resolution. The MATLAB implementation of the algorithm can be can be found at MATLAB Codes.

Remark 4.1.1. The idea of progressive stepping is not new, similar approach have been used to improve convergence of the Lagrange multipliers method. This modification allows us to reduce the number of iterations by at least a factor of 2-2.5, see Appendix B. However this does not eliminate the dependence of the convergence rate on the mesh size, which should be accounted for. We investigate the convergence of the modified algorithm in the next section.

4.2 Numerical experiments

The global linear convergence of ADMM was established in [95] for strongly convex objective with a Lipschitz gradient. The result can be extended to various generalizations of ADMM, including the APGM. Iteration of APGM with a constant step size lead to diminishing updates of the numerical solution. Because of that and despite the proven convergence estimates, without tuning of parameters, the algorithm may stall and fail to converge.

The proposed adaptive step size in the modified version reduces the risk of stalling or at least postpones it. We show that this enhancement is essential to achieve an accurate result and to reliably compute dual discrete TV.

We test convergence on the function we used in Section 2.4

$$u = e^{-10(x^2+y^2)}, \quad \forall (x,y) \in \Omega = [-2,2] \times [-2,2]. \tag{4.16}$$

We use cell-averaged values of u to compute its projection $U \in \mathbb{R}^{N\times N}$ onto a grid. Then we run the APGM with $\mu = 0.01, 0.1, 0.5, 1, 10$ and compare it to Algorithm 1 for $K = 1000$ iterations. We report $TV_d(U)$ at $k = 100, 500, 600, 700, 800, 900, 1000$ iterations in Figure 4.2 and Table 4.1.

We compute the error of the k-th iterate using

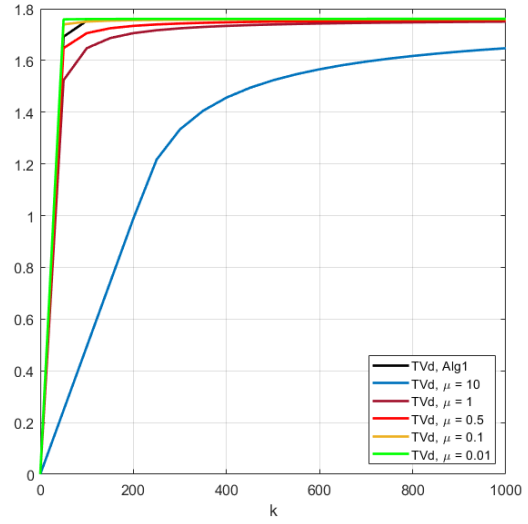
$$\Delta TV_k = \Delta x \Big| \sum_{i,j} \langle \boldsymbol{D}U_{i,j}, \boldsymbol{\varphi}_{i,j}^k \rangle - \|\boldsymbol{v}^k\|_{1,1,2} \Big|. \tag{4.17}$$

Then we plot the loss function ΔTV_k in Figure 4.3. The final error ΔTV_K is reported in the last row of Table 4.1.

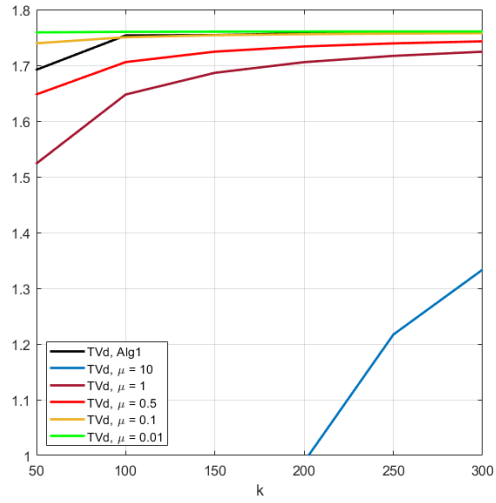
We observe that $\mu = 0.01$ yields the best overall result for APGM on the grid with $N = 128$. It has the smallest error and arrives at the computed value with fewer iterations. The final error of Algorithm 1 is smaller than that of APGM with constant step for all but one value of constant step size $\mu = 0.01$. For other choices of μ, Algorithm 1 outperforms the APGM. The run time is about 4s for $K = 1000$ iteration both APGM and Algorithm 1. It does not depend on the constant step size. If we would perform computation up to the given tolerance, Algorithm 1 would outperform APGM because it would converge faster to the desired accuracy.

We note that the choice of μ influences the performance of the APGM method. The convergence for all algorithms slows down as iterations progress. Notice that the values of TV monotonically increase with iterations as previously discussed. An arbitrary chosen $\mu > 0$ may result in $TV_d(U)$ far from the exact value even on fine grids. The optimal step size depends on a number of variables, e.g. mesh size and properties of function u. Thus a suitable μ can be only determined experimentally. While $\mu = 0.01$ works the best in this example, it will not necessarily be optimal for other problems. To show that the modified version of the algorithm efficiently eliminates the need for careful tuning of the step size we repeat the computation on the refined grid with $N = 256$ and report the results in Table 4.2. We observe that, in this case, Algorithm 1 yields the best solution and we also note that in this case the best choice for the step size is $\mu = 0.1$, is different from the case $N = 128$. Since the error for the best choice of μ is greater than that for the case with $N = 128$, we conclude that we are far from the optimal value of μ.

The use of the modified APGM results in better accuracy than most constant rate

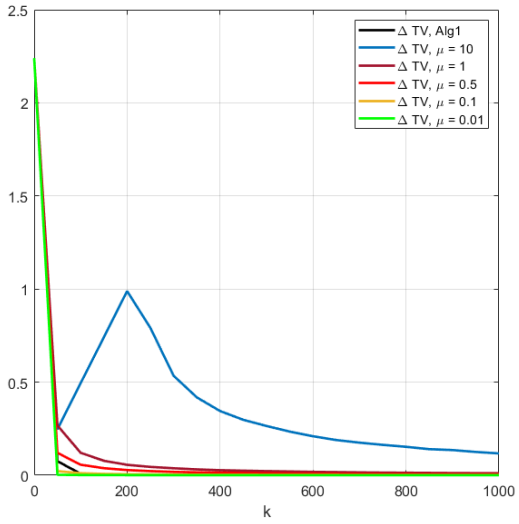


(a) $TV_d(U)$ over 1000 iterations.

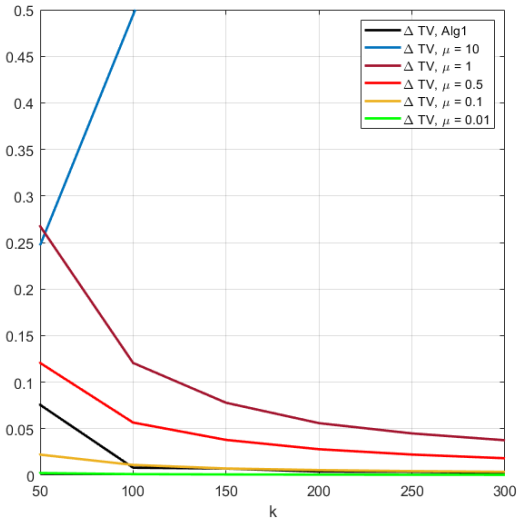


(b) $TV_d(U)$ iterations 50 to 300.

Figure 4.2: $TV_k(U)$ and $\Delta TV_k(U)$ for APGM with different μ and for modified APGM (Alg1).



(a) ΔTV_k, over 1000 iterations.



(b) ΔTV_k, iterations 50 to 300.

Figure 4.3: $TV_k(U)$ and $\Delta TV_k(U)$ for APGM with different μ and for modified APGM (Alg1).

results and saves time by eliminating the need for careful tuning of the step size of APGM. The best possible reconstruction is achieved when using values of θ closer to one and larger ranges of μ^n.

4.3 Summary

In this chapter we have discussed the primal-dual approach for computation of the dual TV given by (2.41). We have consider the APGM [95] that is a fast and efficient version of alternating directions method of multipliers. We have proposed a modification to the

k	$Alg1$	$\mu = 10$	$\mu = 1$	$\mu = 0.5$	$\mu = 0.1$	$\mu = 0.01$
100	1.7533	0.4944	1.6482	1.7062	1.7504	1.7602
500	1.7599	1.5234	1.7392	1.7512	1.7592	1.7605
600	1.7600	1.5665	1.7432	1.7522	1.7592	1.7606
700	1.7600	1.5964	1.7451	1.7533	1.7595	1.7607
800	1.7601	1.6173	1.7471	1.7541	1.7597	1.7607
900	1.7601	1.6340	1.7490	1.7555	1.7599	1.7607
1000	1.7601	1.6471	1.7502	1.7561	1.7599	1.7607
ΔTV_K	0.0008	0.1173	0.0112	0.0053	0.0011	0.0002

Table 4.1: Computed $TV_d^k(U)$ after k iterations of Algorithm 1 (column 2) and the APGM algorithm with various values of μ (columns 3–7) for $N = 128$. The bottom row shows the error ΔTV_K defined in (4.17) after $K = 1000$ iterations for both algorithms.

k	$Alg1$	$\mu = 10$	$\mu = 1$	$\mu = 0.5$	$\mu = 0.1$	$\mu = 0.01$
100	1.7446	1.2518	1.7167	1.7390	1.7567	1.7556
500	1.7578	1.3520	1.7244	1.7425	1.7573	1.7564
600	1.7583	1.4162	1.7296	1.7455	1.7579	1.7571
700	1.7589	1.4633	1.7333	1.7475	1.7582	1.7576
800	1.7595	1.4985	1.7367	1.7488	1.7586	1.7580
900	1.7598	1.5266	1.7389	1.7501	1.7588	1.7582
1000	1.7603	1.6717	1.7522	1.7565	1.7601	1.7598
ΔTV_K	0.0006	0.0892	0.0087	0.0043	0.0007	0.0010

Table 4.2: Computed $TV_d^k(U)$ after k iterations of Algorithm 1 (column 2) and the APGM algorithm with various values of μ (columns 3–7) for $N = 256$. The bottom row shows the error ΔTV_K defined in (4.17) after $K = 1000$ iterations for both algorithms.

APGM that uses a progressively diminishing step size. We conduct numerical experiments demonstrating the performance of the proposed algorithm and compare it to APGM. Finally, we show that the proposed algorithm outperforms the original version in terms of accuracy with a fixed number of iterations if the step-size of APGM is not fine-tuned. We observe similar accuracy of the proposed algorithm to that of the APGM with optimal step. The modified version allows us to significantly reduce the number of steps required to achieve a given accuracy of TV computation and hence the computational time. We implement both algorithms in MATLAB, see Appendix B. The modified algorithm eliminates the need of tuning and allows to construct a fast and accurate imaging algorithm for application in computed tomography image reconstruction that we consider in the next section.

Chapter 5

Applications to image reconstruction

In many imaging applications, the aim is to find a discrete image using given measurements. The relation between the image $U \in \mathbb{R}^{N^2}$ and a vector of measured projection data $f \in \mathbb{R}^M$, can be represented by a linear system

$$f = AU, \tag{5.1}$$

where A is a linear projection operator matrix of size $M \times N^2$. The entries of A depend on the configuration of the imaging apparatus and the mathematical model that describes the imaging method. The relationship between the measurement and the image for parallel beam computed tomography (CT) can be described by the Radon transform with A being the Radon transform matrix. In other settings A can have a different form, e.g. for two-dimensional fan beam CT it models a ray transform [16].

In imaging applications, matrix A is large, usually ill-posed, and often underdetermined. The ill-posedness of (5.1) can originate from multiple sources, most common of these are insufficient coverage, projection data truncation, and under-sampling, i.e. $M \ll N^2$. Additionally, measurements f might be polluted by noise. For example, for the parallel beam CT that we will consider here, all of these issues are present. It makes it difficult to solve (5.1) directly using linear algebra tools.

Instead of directly solving (5.1), we can look for an approximation of U by minimizing

$$\min_U \|AU - f\|_2^2. \tag{5.2}$$

The norm above is called the least-squares objective function. The problem has a closed-form solution $U = (A^T A)^{-1} A^T f$ for $M \geq N^2$, but due to poor conditioning of A it is severely polluted by noise in the directions corresponding to small singular values of A. In the case of an underdetermined system, i.e. $M < N^2$, solutions of (5.2) are not unique. For this reason, a regularization approach is more commonly used. It consists of solving a modified minimization problem

$$\min_U \ \|AU - f\|_2^2 + \lambda TV(U), \quad \text{s.t.} \quad U \geq 0, \tag{5.3}$$

where $\|AU - f\|_2^2$ is the least-squares error or data fidelity, $\lambda > 0$ is a regularization parameter, and $TV(U)$ is the penalty term aiming to reduce the impact of noise on the approximate solution and as a result to obtain a satisfactory CT reconstructed image. We require U in (5.3) to be non-negative because U in this model represents the attenuation coefficient, which is always positive. The model is based on the Lambert-Beer law of

radiation attenuation. The reconstructed image can be viewed as a representation of the spatial distribution of the attenuation coefficient in an object. Since different materials in the object have different values of the coefficient, we can distinguish between them. When the matrix A has full column rank, the norm in (5.3) is a strictly convex function and $TV(U)$ is a convex function. Therefore, their sum is strictly convex for $\lambda > 0$. Hence, (5.3) always has a unique minimizer.

Total variation as a penalty term in image reconstruction was first considered by L. Rudin, S. Osher, and E. Fatemi in [114]. Even though isotropic TV used in [114] is not accurate in approximating the exact TV of non-smooth functions, as shown in Chapter 2, it can still offer improved accuracy in the solution of (5.1). This approach has been shown to accurately restore edges in images while preserving boundaries and eliminating noise and other artifacts. In later years, TV has been extensively used as a regularizer for inverse and ill-posed problems.

In the next sections, we show how the TV regularization approach can be used for parallel beam computed tomography applications and how more accurate approximations of TV result in better image reconstruction quality.

5.1 Computed tomography by total variation minimization

We are interested in reconstructing images for two-dimensional parallel beam CT with sparse and noisy measurements that can be described by (5.1). The right-hand side of (5.1), f, is given by experimental measurements. U is a square image that we want to reconstruct. U can be naturally described as a square $N \times N$ matrix with elements $U_{i,j}$, where each value $U_{i,j}$ corresponds to the center of (i, j)-th pixel. We also use a concatenated version of U, a vector of length N^2 with elements U_s, $1 \leq s \leq N^2$, which is constructed by traversing the matrix row by row. We will be using U in both senses. It will be clear from the subscript when we view U as a vector and when as a matrix.

A two-dimensional parallel beam CT can be described as follows. The scanning apparatus shoots X-ray beams from M directions. Each of the directions corresponds to a location of a source emitting X-rays and a detector capturing the attenuated radiation levels (Figure 5.1). Then the measurements are stored as components of f_m, $1 \leq m \leq M$.

Assuming that within each beam all rays follow the same line (Figure 5.1), A_{mn} is an attenuation length of the n-th pixel, $1 \leq n \leq N^2$, for the m-th line, $1 \leq m \leq M$. In other words, an element A_{mn} corresponds to the contribution of the n-th pixel to the m-th measurement. That is, A_{mn} is the depth of penetration into the material at which the intensity of X-rays falls by a factor $1/e$ from its value at the surface.

As we have mentioned in the introduction, the result of the product AU is the Radon transform of U. For a function $g(x, y)$ defined on $\Omega = [-1, 1] \times [-1, 1]$ and a given line, the Radon transform of g is the value of its integral along the line. This models the total attenuation of X-rays along this line.

Next, we describe the discrete Radon transform of U. In our model, lines correspond to the directions of X-ray beams. For each beam, we replace the integration along the line with a summation of the entries of U along it. An arbitrary line in the plane can be described by an equation in the form $y = sx + t$ or $x = \bar{s}y + \bar{t}$. Between the two, we pick the expression with the slope less than one (Figure 5.2 (left)). We start with the case

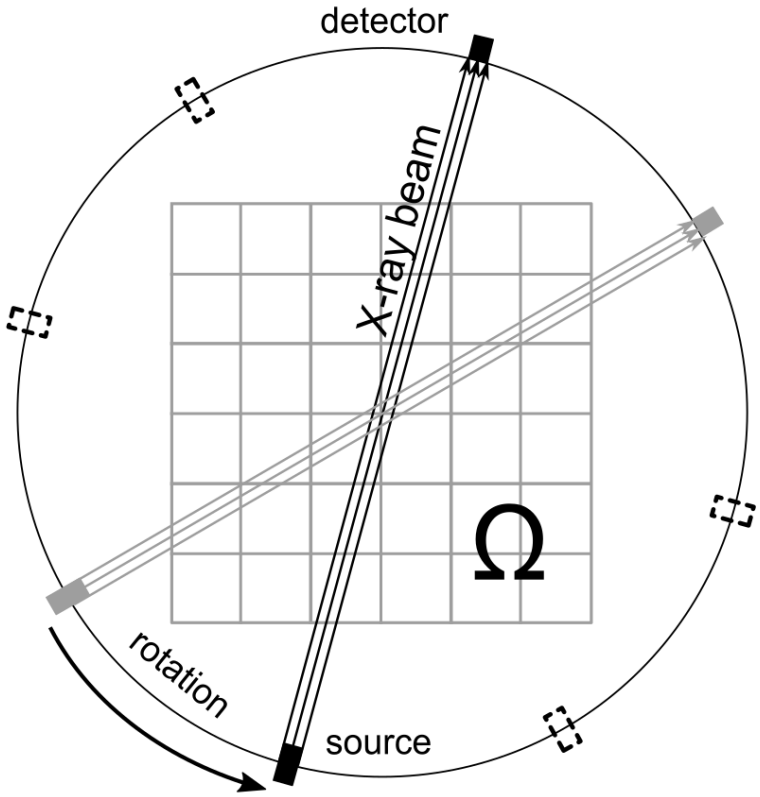


Figure 5.1: X-ray computed tomography configuration. The dashed rectangles show the positions where measurements are taken.

$y = sx + t,\ |s| \leq 1$ (Figure 5.2 (right)). For each cell $\Omega_{i,j}$, we take $x = x_i$, i.e. the centroid of the cell, and find the corresponding point on the line, $(x_i, y(x_i))$. We approximate U at this point by $\widetilde{U}^1(x_i, sx_i + t)$ which is a trigonometric polynomial interpolation based on the entries of U in the corresponding column

$$\widetilde{U}^1(x_i, y) = \sum_{j=1}^{N} U_{i,j} D_N\left(y - y_j\right), \tag{5.4}$$

where D_N is the Dirichlet kernel given by

$$D_N(y) = \frac{\sin(\pi y)}{(2N+1)\sin\left(\pi y/(2N+1)\right)}. \tag{5.5}$$

For the points $(x_i, sx_i + t)$ that lie outside of Ω we set $\tilde{U}^1(x_i, y) = 0$. Additionally, trigonometric interpolation requires the periodicity of data. Normally U in CT applications is zero on the boundary of the domain. If this is not the case we can use periodic padding.

We have M directions that can be parametrized with slopes s_m and intercepts t_m. Then we compute the discrete Radon transform of U along the line $y = s_m x + t_m$ as

$$(AU)_m = \sum_{i=1}^{N} \widetilde{U}^1(x_i, s_m x_i + t_m). \tag{5.6}$$

Similarly, for lines of the form $x = \bar{s}y + \bar{t},\ |\bar{s}| \leq 1$ we interpolate U at points $(x(y_j), y_j)$

using the data from row j

$$\widetilde{U}^2(x, y_j) = \sum_{i=1}^{N} U_{i,j} D_N (x - x_i) , \tag{5.7}$$

where the kernel D_N is given by (5.5). Then

$$(AU)_m = \sum_{j=1}^{N} \widetilde{U}^2(\bar{s}_m y_j + \bar{t}_m, y_j). \tag{5.8}$$

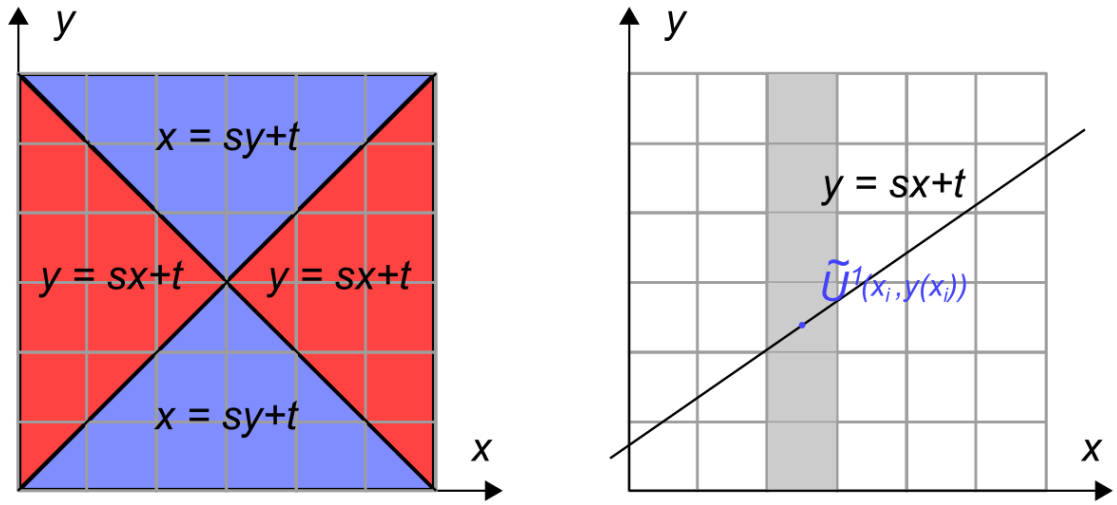


Figure 5.2: Line equations over the image domain (left) and interpolation at a point on the line $y = sx + t$ (right), the column values used for interpolation are shown in gray, no padding applied.

It can be shown that using $M = 2N$ equiangular projections, which correspond to the Nyquist frequency, ensures that the discrete Radon transform is invertible. However, in practice, the number of projections is severely limited and much smaller than the number of pixels in the image, i.e. $M \ll 2N$.

In the following sections, we propose to employ the dual discrete TV (2.41) in sparse-view CT, low-dose CT, and limited-angle CT applications. We implement a new version of the adaptive steepest descent projection onto convex sets (ASD-POCS) algorithm with the dual discrete TV. We demonstrate that the improvement in approximating the gradient of TV increases the accuracy of projection onto convex sets (POCS) algorithms. This allows us to enhance image quality in medical diagnostics and reduce artifacts that arise due to the flaws of conventional TV discretizations.

5.2 Projection onto convex sets algorithms

The projection onto convex sets method for constrained TV minimization uses a different approach to solve (5.1). It is based on the idea that for noisy data f the data fidelity $\|AU - f\|_2^2$ will never be zero. Instead, we consider the TV as the functional to be minimized, while the data fidelity $\|AU - f\|_2^2 \leq \varepsilon$ is used as a constraint. That is a solution to

the constrained optimization problem that has the smallest TV and will agree with the available data up to an admissible error ε. Thus, we look for U, an approximate solution of (5.1) in the form

$$\min_{U\geq 0} TV(U) \quad \text{s.t.} \quad \|AU - f\|_2^2 \leq \varepsilon. \tag{5.9}$$

A number of algorithms for numerical solution of (5.9) have been proposed. Among the most common are the steepest gradient descent (SGD), alternating direction method of multipliers (ADMM), and Chambolle-Pock algorithms.

We use a version of the steepest gradient descent algorithm that was proposed in [120], which consists of two independent steps. In the first step, the simultaneous algebraic reconstruction technique (SART), which is an iterative algorithm for solving $AU = f$, is applied. SART is an improved version of the algebraic reconstruction technique (ART). Which in turn is a version of Kaczmarz's algorithm,

At the $(k+1)$-st iteration of ART we look for U^{k+1}, a minimizer of

$$\min_U \frac{1}{2}\|U - U^k\|_2^2, \text{ s.t. } A_m^T U = f_m, \tag{5.10}$$

where U^k is the previous iterate and A_m is the m-th row of A. The Lagrangian for (5.10) is given by

$$\mathcal{L}_\beta(U,\beta) = \frac{1}{2}\|U - U^k\|_2^2 + \beta(A_m^T U - f_m),\ \beta > 0. \tag{5.11}$$

The next iterate U^{k+1} is the stationary point of the Lagrangian. To find it, we compute the partials of $\mathcal{L}(U,\beta)$ and set them to zero

$$\nabla_U \mathcal{L}(U,\beta) = U^{k+1} - U^k + \beta A_m = 0,$$
$$\partial_\beta \mathcal{L}(U,\beta) = A_m^T U^{k+1} - f_m = 0.$$

It follows from the first equation that

$$U^{k+1} = U^k - \beta A_m. \tag{5.12}$$

Substituting (5.12) into the second equation, we get

$$A_m^T(U^k - \beta A_m) - f_m = 0. \tag{5.13}$$

Then

$$\beta = \frac{A_m^T U^k - f_m}{A_m^T A_m} = \frac{1}{\sum_{s=1}^{N^2} A_{ms}^2}\left(\sum_{s=1}^{N^2} A_{ms}U_s^k - f_m\right). \tag{5.14}$$

Therefore, the formula to update U^k can be written as

$$U^{k+1} = U^k + \frac{\sum_{s=1}^{N^2} A_{ms}}{\sum_{s=1}^{N^2} A_{ms}^2}\left(\sum_{s=1}^{N^2} A_{ms}U_s^k - f_m\right). \tag{5.15}$$

The difference between the measured and projected data $f_m - \sum_{s=1}^{N^2} A_{ms}U_s^k$ is used here as the correction term. SART differs from ART in that we compute the average of the individual corrections first and then update the reconstructed image [3]. Then SART is

written as

$$\hat{U}_s^{k+1} = U_s^k - \frac{\lambda_k}{\sum_{m=1}^{M} A_{ms}} \sum_{m=1}^{M} \frac{A_{ms}}{\sum_{l=1}^{N^2} A_{ml}} \left(f_m - \sum_{s=1}^{N^2} A_{ms} U_s^k \right), \ s = 1, \ldots, N^2. \tag{5.16}$$

where elements A_{ms} are defined in (5.4),(5.7). The term $\sum_{s=1}^{N^2} A_{ms}^2$ in the denominator of (5.15) was replaced in (5.16) with $\sum_{l=1}^{N^2} A_{ml}$ for uniformity of reconstruction. We also introduced $\lambda_k \in (0, 1)$, an optional relaxation parameter. Relaxation parameters of POCS depend on the M/N ratio. It increases computation time but can improve the signal-to-noise ratio in the reconstructed image. The original SART was stated with $\lambda_k = 1$.

SART iterations aim to find U^{k+1} such that $\|AU - f\|_2^2 \leq \varepsilon$ [63]. The convergence of the algorithm was shown in [72] for $\lambda_k \in (0, 1)$ under the following conditions

$$A_{ms} \geq 0, \ \forall m, s, \quad \sum_{m=1}^{M} A_{ms} > 0, \ \forall s, \quad \sum_{s=1}^{N^2} A_{ms} > 0, \ \forall m.$$

The first condition is satisfied because in our model A_{ms} is an attenuation length in this configuration and therefore it is nonnegative. The next, $\sum_{m=1}^{M} A_{ms} > 0$ means that there is at least one nonzero element in each column of A. In other words, there is a contribution to the measurement from each pixel of U is measured at each pixel. Finally, the last condition requires at least one nonzero element in each row of A, i.e. every measurement f_m carries a certain amount of information about U. We use a shorthand notation $SART(U^k)$ for (5.16) in the algorithm description below.

SART, when applied alone is known to converge to the solution of $AU = f$ [72]. However, this solution will not have the smallest TV. Additionally, it is not guaranteed that all entries of $\hat{U}^{k+1}$ are positive. We correct this by setting negative entries of $\hat{U}^{k+1}$ equal to zero after each SART update.

The second step of a POCS algorithm is the gradient descent. For (5.9) it is given by

$$U^{k+1} = \hat{U}^{k+1} - \mu_{TV} \frac{\partial_U TV(\hat{U}^{k+1})}{|\partial_U TV(\hat{U}^{k+1})|}, \tag{5.17}$$

where $\mu_{TV} > 0$ is the step size or learning rate, and we use the notation $\partial_U TV(\hat{U}^{k+1})$ for the subgradient of the discrete TV with respect to image pixel values. This process minimizes the TV of the image. Since TV minimization does not create new extrema, the image remains non-negative. Thus the POCS algorithm consists of alternating (5.16)-(5.17). In practical applications, the second step is often repeated several times before performing step (5.16). We note here that the parameters of the algorithms need to be determined empirically and may generally depend on several features, such as image size, noise level, and others.

In [120], anisotropic discrete TV was used to compute the gradient in (5.17). A number of modifications to the original POCS algorithm have been proposed in the literature [30, 126, 93, 100]. In [30] a prior image-constrained compressed sensing model was used. In [126] an edge-preserving TV was proposed for low-dose CT applications. Then a more accurate adaptively-weighted isotropic TV image reconstruction was proposed in [93]. More recently, a non-local total generalized variation method for sparse-view X-ray CT was proposed in [100] and achieved better reconstruction quality relative to the previous TV definitions.

These improved TV definitions have gained a lot of attention and led to improved reconstruction quality for sparse-view, low-dose, and limited-angle CT applications. However, most of these methods are still prone to producing blocky artifacts, line artifacts, and other spurious effects. An artifact is any systematic discrepancy between the reconstructed image and the true attenuation of the object. Artifacts could be any features appearing in the reconstructed image that are not present in the real object. Since the image is reconstructed using an optimization model, these artifacts are specific to each penalty term (TV definition) used.

The versions of the POCS algorithm differ by the definition of discrete TV they use in the second step. Below we derive the the expressions for $\partial_U TV(U)$ for a few of them, compare the algorithms on several cases, and report the results in the numerical experiments section (Section 5.4).

First, we consider the original TV penalty term that is used in [120], i.e. $TV_a(U)$. The components of $\partial_U TV_a$ are given by

$$\begin{aligned}\frac{\partial TV_a(U)}{\partial U_{i,j}} &= \Delta x \left(-sgn\left(U_{i+1,j} - U_{i,j}\right) + sgn\left(U_{i,j} - U_{i-1,j}\right)\right) \\ &\quad + \Delta x \left(-sgn\left(U_{i,j+1} - U_{i,j}\right) + sgn\left(U_{i,j} - U_{i,j-1}\right)\right). \end{aligned} \tag{5.18}$$

Next, consider the adaptively weighted TV, a modified version of $TV_{is}(U)$.

$$TV_{aw}(U) = \Delta x \sum_{ij} \sqrt{w^1_{i,j}(D^1 U_{i,j})^2 + w^2_{i,j}(D^2 U_{i,j})^2}, \tag{5.19}$$

where $w^1, w^2 \in \mathbb{R}^{N\times N}$ are weight matrices with components given by

$$w^1_{i,j} = \exp\left(-\left(\frac{D^1 U_{i,j}}{\sigma}\right)^2\right), \quad w^2_{i,j} = \exp\left(-\left(\frac{D^2 U_{i,j}}{\sigma}\right)^2\right).$$

The parameter $\sigma > 0$ controls the amount of smoothing of sharp edges in the image. This TV is used in adaptively weighted ASD-POCS algorithm (Aw-ASD-POCS) [93]. Its gradient is given by

$$\begin{aligned}\frac{\partial TV_{aw}(U)}{\partial U_{i,j}} &= \Delta x \left(\frac{2w^1_{i-1,j} D^1 U_{i-1,j} + 2w^2_{i,j-1} D^2 U_{i,j-1}}{\sqrt{w^1_{i-1,j}(D^1 U_{i-1,j})^2 + w^2_{i,j-1}(D^2 U_{i,j-1})^2}}\right) \\ &\quad - \Delta x \left(\frac{2w^1_{i,j} D^1 U_{i,j}}{\sqrt{w^1_{i,j}(D^1 U_{i,j})^2 + w^2_{i+1,j-1}(D^2 U_{i+1,j-1})^2}}\right) \\ &\quad - \Delta x \left(\frac{2w^2_{i,j} D^2 U_{i,j}}{\sqrt{w^1_{i-1,j+1}(D^1 U_{i-1,j+1})^2 + w^2_{i,j}(D^2 U_{i,j})^2}}\right). \end{aligned} \tag{5.20}$$

The parameter σ should be chosen carefully, as large values of σ will result in gradient values very close to that of $TV_{is}(U)$, while small values tend to give low weights to almost every pixel, which makes Aw-ASD-POCS inefficient in removing noise and artifacts. The partial derivatives of (5.20) may not be properly defined in the regions where U is close to zero. To avoid division by zero, a small positive constant δ is introduced in the denominator

of (5.20)

$$\begin{aligned}\frac{\partial TV_{aw}(U)}{\partial U_{i,j}} \approx & \Delta x \left(\frac{2w^1_{i-1,j} D^1 U_{i-1,j} + 2w^2_{i,j-1} D^2 U_{i,j-1}}{\sqrt{\delta + w^1_{i-1,j}(D^1 U_{i-1,j})^2 + w^2_{i,j-1}(D^2 U_{i,j-1})^2}} \right) \\ & - \Delta x \left(\frac{2w^1_{i,j} D^1 U_{i,j}}{\sqrt{\delta + w^1_{i,j}(D^1 U_{i,j})^2 + w^2_{i+1,j-1}(D^2 U_{i+1,j-1})^2}} \right) \\ & - \Delta x \left(\frac{2w^2_{i,j} D^2 U_{i,j}}{\sqrt{\delta + w^1_{i-1,j+1}(D^1 U_{i-1,j+1})^2 + w^2_{i,j}(D^2 U_{i,j})^2}} \right).\end{aligned}$$

Above, we use an approximate sign due to the added δ. Unfortunately, this approximation of the gradient also introduces over-smoothing in the regions near sharp edges.

We propose a version of the ASD-POCS algorithm that uses discrete dual TV as the objective function. Let $\boldsymbol{\varphi} = (\varphi, \psi)$ be the computed maximizer for $TV_d(U)$ in (2.41). Then the partial subderivatives of TV with respect to $U_{i,j}$ can be computed as

$$\begin{aligned}\frac{\partial TV_d(U)}{\partial U_{i,j}} &= \frac{\partial}{\partial U_{i,j}} \left(\Delta x \sum_{i,j} \langle \boldsymbol{D} U_{i,j}, \boldsymbol{\varphi}_{i,j} \rangle \right) \\ &= -\Delta x (\varphi_{i+1/2,j} - \varphi_{i-1/2,j} + \psi_{i,j+1/2} - \psi_{i,j-1/2}). \end{aligned} \tag{5.21}$$

The expression (5.21) is the result of formal differentiation of the product $\langle \boldsymbol{D} U_{i,j}, \boldsymbol{\varphi}_{i,j} \rangle$, which does not account for the fact that $\boldsymbol{\varphi}$ depends on U.

Computing the dual TV is expensive especially when it needs to be repeated for many iterations and may result in an unreasonably long computational time. We fix this by using the maximize from the previous iteration $\boldsymbol{\varphi}$ as an initial guess for the next iteration. This results in an algorithm that is a magnitude faster.

Assume that we have computed a saddle-point $\boldsymbol{\varphi}, \boldsymbol{v}$ for the primal-dual problem (2.41)-(4.9) for a given image U. Then on the first step of the new iteration, we set $\boldsymbol{\varphi}^{k+1} = \boldsymbol{\varphi}^k$. This way we take into account that the image U is static, and the new maximizer $\boldsymbol{\varphi}^{k+1} \approx \boldsymbol{\varphi}^k$. Therefore, the number of steps required for the algorithm to converge is significantly reduced as opposed to recomputing the maximizer from zero initial guesses.

5.3 Parallel implementation of the DTV-ASD-POCS

While CT can achieve great results in terms of image quality and resolution, reconstruction of CT images remains a computationally intensive task, demanding a lot of time. In recent years, computing on Graphics Processing Units (GPUs) has become a popular source for accelerating imaging methods, including CT image reconstruction via TV minimization.

In [9, 10] parallel versions of ASD-POCS and AwASD-POCS algorithms were implemented using a multi-GPU approach for parallel beam and fan-beam three-dimensional CT. This acceleration allowed for real-time CT reconstruction. Here we propose a parallel version for the proposed DTV-ASD-POCS algorithm and compare its performance with ASD-POCS and AwASD-POCS on real experimental data. Our goal is to show the impact the change of discrete TV definition makes on the image reconstruction quality. We also

show that this result is achieved in approximately the same computational time as other POCS-type algorithms.

Pseudo-code for the parallel version of DTV-ASD-POCS algorithm.

We employ the modified APGM, i.e. Algorithm 1 described in Section 4.1, and parallelized it (Algorithm 3). Algorithm 3 is used in the second step of DTV-ASD-POCS. The pseudo-code for the parallel version of the DTV-ASD-POCS algorithm is given below as Algorithm 2.

Algorithm 2 DTV-ASD-POCS

1: $\alpha := 0.2$, $\alpha_{\text{red}} := 0.95$, $\theta := 0.96$, $\gamma := 1/3$, $\lambda := 1.0$, $\lambda_{\text{red}} := 0.995$
2: $r_{max} := 0.95$, N_{iter} , N_{TV}, ε
3: $U := \mathbf{0}$, $U^0 := \mathbf{0}$, A, f, $\boldsymbol{v}^0 := ((D^1U, \mathbf{0}), (\mathbf{0}, D^2U), (\mathbf{0}, \mathbf{0}))$, $\quad \boldsymbol{\varphi}^0 := \mathbf{0}$, $\quad \mu^0 = \mu$
4: **for** $k = 1, \ldots, N_{iter}$ **do**
5: $\quad U := \text{SART}(U)$ (perform SART)
6: $\quad$ **for** $i = 1, \ldots, N^2$ **do**
7: $\quad\quad$ **if** $U_i < 0$ **then**
8: $\quad\quad\quad U_i := 0$ (enforce positivity)
9: $\quad\quad$ **end if**
10: $\quad$ **end for**
11: $\quad r := |AU - f|$
12: $\quad$ **if** iter $== 1$ **then**
13: $\quad\quad \mu_{TV} := \alpha \cdot |U - U^1|$
14: $\quad$ **end if**
15: $\quad r_u = |U - U^k|$
16: $\quad U^k := U$
17: $\quad$ **for** $i = 1, \ldots, N_{TV}$ **do**
18: $\quad\quad \boldsymbol{\varphi}^{k+1}, \boldsymbol{v}^{k+1} := APGM(U, \boldsymbol{\varphi}^k, \theta, \gamma, \mu^0)$
19: $\quad\quad \partial\hat{U} := \partial TV_d(U)$
20: $\quad\quad \partial\hat{U} := \partial\hat{U} \ / \ |\partial\hat{U}|$
21: $\quad\quad U := U - \mu_{TV} \cdot \partial\hat{U}$
22: $\quad\quad \boldsymbol{\varphi}^k := \boldsymbol{\varphi}^{k+1}$
23: $\quad\quad \boldsymbol{v}^k := \boldsymbol{v}^{k+1}$
24: $\quad$ **end for**
25: $\quad \delta U := |U - U^k|$
26: $\quad$ **if** $\delta U > r_{max} \cdot r_U$ and $r > \varepsilon$ **then**
27: $\quad\quad \mu_{TV} := \mu_{TV} \cdot \alpha_{\text{red}}$
28: $\quad$ **end if**
29: $\quad \lambda := \lambda \cdot \lambda_{\text{red}}$
30: **end for**
31: return U

In Algorithm 2 $APGM(U, \boldsymbol{\varphi}^k)$ stands for Algorithm 1, initialized with U and $\boldsymbol{\varphi}^k$. At each iteration, we use the maximizer that was computed for U^k as an initial guess for $\boldsymbol{\varphi}^{k+1}$, line 18. Algorithm 1 (lines 4 – 15) is run on GPU with one thread per pixel. We improve the computational efficiency by performing N_{th} steps of the algorithm on each thread before updating the vectors $\boldsymbol{v}^{k+1}, \boldsymbol{\varphi}^{k+1}$. The parallelized version of Algorithm 1 is shown below.

We do not specify values of parameters N_{iter}, N_{TV}, ε as they are problem-dependent. We will provide them in the numerical experiments section. Other parameters $\alpha, \alpha_{\text{red}}, \lambda, \lambda_{\text{red}}, r_{max}, \varepsilon$

Algorithm 3 Modified APGM (parallelized version).

1: $\boldsymbol{v}^0 := ((D^1U, \mathbf{0}), (\mathbf{0}, D^2U), (\mathbf{0}, \mathbf{0})), \quad \boldsymbol{\varphi}^0 := \mathbf{0} \quad \mu^0 := \mu, \quad n := 0$
2: **while** $\|\boldsymbol{v}^{n+1} - \boldsymbol{v}^n\|_{1,1,2} > \varepsilon$ and $\left|\|\boldsymbol{v}^{n+1}\|_{1,1,2} - \sum_{i,j}\langle \boldsymbol{D}U_{i,j}, \boldsymbol{\varphi}^n_{i,j}\rangle\right| > \varepsilon$ **do**
3: load a copy of $\boldsymbol{v}^n, \boldsymbol{\varphi}^n$ to each thread of GPU
4: **for** it $= 1 \dots N_{th}$ **do**
5: **for** $k := 1 \dots 3$ **do**
6: $(v_k^{n+1})_{i,j} := pr_{\gamma\mu}\left(\boldsymbol{v}^n - \gamma \boldsymbol{P}\left(\boldsymbol{D}U - \boldsymbol{F}\boldsymbol{v} + \mu\boldsymbol{\varphi}^{n+1}\right)\right)$
7: $(v_k^n)_{i,j} := (v_k^{n+1})_{i,j}$
8: **end for**
9: $(\boldsymbol{\varphi}^{n+1})_{i,j} := (\boldsymbol{\varphi}^n)_{i,j} + (\boldsymbol{D}U - \boldsymbol{F}\boldsymbol{v}^{n+1})/\mu^n$
10: $(\boldsymbol{\varphi}^n)_{i,j} = (\boldsymbol{\varphi}^{n+1})_{i,j}$
11: $\mu^{n+1} := \theta\mu^n$
12: **if** $rem(n, 100) = 0$ **then**
13: $\mu^{n+1} := \mu$
14: **end if**
15: **end for** } in parallel
16: collect all entries of $\boldsymbol{v}^{n+1}, \boldsymbol{\varphi}^{n+1}$
17: $n := n + 1$
18: **end while**

control the accuracy of the algorithm and have to be tuned for particular applications [120]. The parameters μ, θ, γ are as in Algorithm 1.

The reconstructed image is initialized with zeros in line 3. Algorithm 2 is run for each pixel of the image U on a separate processor multiple times. In each run of the algorithm, the pixel values are updated, processors are synchronized and the updated image is stored in memory before it is used in the next run of the algorithm. The main loop performs $N_{iter} > 1$ iterations. We choose the number of iterations for the main loop based on the required accuracy of the reconstruction or available computational time. Inside the main loop, we perform the two main steps of the algorithm. First, we make SART, which adjusts the image to satisfy the data fidelity. The relaxation parameter λ is initialized with 1 and slowly decreases to 0 over iterations. Then in lines 6–10, the positivity of the pixel values of the reconstructed image is enforced.

The data residuals are recomputed in line 11. In lines 12–14 the step-size for TV gradient descent is initialized. The change in the image due to POCS is computed in line 15. Lines 17–24 implement the TV-steepest descent towards the direction with minimal dual discrete TV value. The reconstructed image after each step of the main loop is stored in U. Additionally, $\boldsymbol{\varphi}^k, \boldsymbol{v}^k$ are stored for the next gradient descent steps. In practice, we find that it is sufficient to make a few steps of the TV update for each POCS step, and then the image and the dual function $\boldsymbol{\varphi}$ need to be updated. Then, we compute the residual after a step of the steepest descent with the change in the image after SART and compare the two by computing their ratio. If the ratio is greater than r_{max}, we reduce the step size of the gradient descent by α_{red}. Every time the ratio is large enough, the step size will be decreased. Finally, in line 29, the SART relaxation parameter is reduced. Finally, if λ is too small, the iteration is stopped.

CUDA code implementation for the DTV-ASD-POCS can be found at CUDA Codes (requires TIGRE framework).

5.4 Numerical experiments

In this section, we provide three numerical examples of CT image reconstruction from sparse-view, noisy, and limited-angle projections. We reconstruct images using the following TV minimization-based imaging methods: ASD-POCS, OS-ASD-POCS, Aw-ASD-POCS, SART-TV, and DTV-ASD-POCS. We have discussed ASD-POCS, Aw-ASD-POCS, and DTV-ASD-POCS in Section 5.2. Ordered subset ASD-POCS (OS-ASD-POCS) and simultaneous algebraic reconstruction with TV minimization (SART-TV) can be found in [134] and [32] respectively. We refer to [9, 10] for details on the implementation. OS-ASD-POCS a faster version of ART, named OS-SART, and SART-TV is not a POCS algorithm and is considered here only for a broader comparison [138].

We use parallel implementations of ASD-POCS, OS-ASD-POCS, Aw-ASD-POCS, and SART-TV algorithms from the TIGRE package, a GPU-based CT reconstruction software repository that contains a wide variety of iterative algorithms and image analysis methods released under the BSD License [9, 10]. We implemented the parallel version of the DTV-ASD-POCS algorithm in Compute Unified Device Architecture (CUDA). Then we merged the algorithm into the TIGRE framework package to perform computations and assess the image quality of the images reconstructed by different algorithms. The codes were run on a Windows GPU server provided by the University of Waterloo with Intel Xeon Gold 6254, a 3.10GHz processor, and 3 NVIDIA Tesla T4 16GB GPUs.

In all examples in this section, we set the stopping criterion of Algorithm 1 to 10^{-5} and use $N_{TV} = 5$. We set $\sigma = 10^{-7}$ for Aw-ASD-POCS, i.e in (5.20). We choose $N_{iter} = 50$ for ASD-POCS, while for the rest of the algorithms, we choose N_{iter} to match the computation time to that of ASD-POCS. To assess the quality of reconstructed images we will measure signal-to-noise ratio (SNR) and root-mean-square error (RMSE)

$$SNR = 10 \log_{10} \left(\frac{\sum_{i,j} U_{i,j}^2}{\sum_{i,j} (U_{i,j} - G_{i,j})^2} \right), \quad RMSE = \sqrt{\sum_{i,j} \frac{(U_{i,j} - G_{i,j})^2}{N}}, \tag{5.22}$$

where $U_{i,j}$ are pixel values of the image found by an optimization algorithm and $G_{i,j}$ are pixel values of the ground truth image. When comparing two methods, a better reconstruction is usually determined by the higher SNR value and lower $RMSE$. We test the quality and accuracy of obtained reconstructions of POCS-type algorithms against the proposed DTV-ASD-POCS algorithm.

Shepp-Logan phantom.

Shepp-Logan phantom is a conventional benchmark problem with known parameters. It represents a human head and is commonly used for CT tuning and calibration purposes. The model with all specifications can be found in the MATLAB package. We use `phantom('Shepp-Logan',128)` command to generate a 128-by-128 pixel image of the phantom, which we use as a ground truth G image for error evaluation and image quality assessment.

We use G to generate the projection data f via discrete Radon transform (5.6)-(5.8). Then the data is used to reconstruct the image of the phantom. We then add white Gaussian noise to f using `awgn(x,10,'measured')` command, which corresponds to the signal-to-noise ratio of 10dB or 10% of noise in the data.

We compare the performance of ASD-POCS, OS-ASD-POCS, Aw-ASD-POCS, SART-TV, and DTV-ASD-POCS on sparse (120 projections, full 360 degree view angle) simulated

data without noise and with added noise. We first run the algorithms until they converge to $\varepsilon = 0.005$, which is the maximum normalized L^2-error $\frac{1}{N^2}\|AU - f\|_2$ to accept an image as valid. We demonstrate the reconstructed images in Figures 5.3-5.4 and report the total computational time T, SNR and $RMSE$ in Table 5.1.

Then, we restrict the number of iteration of ASD-POCS to 50, check the total running time and compute SNR and $RMSE$ of the reconstructed image. Then we choose the number of iterations and other parameters for the rest of the algorithms to match the total time of computation to that of ASD-POCS. This approach allows us to compare the quality of reconstructions obtained with different algorithms in approximately equal reasonable time. We demonstrate the reconstructed images in Figures 5.5-5.9 and report the total computational time T, SNR and $RMSE$ in Table 5.2.

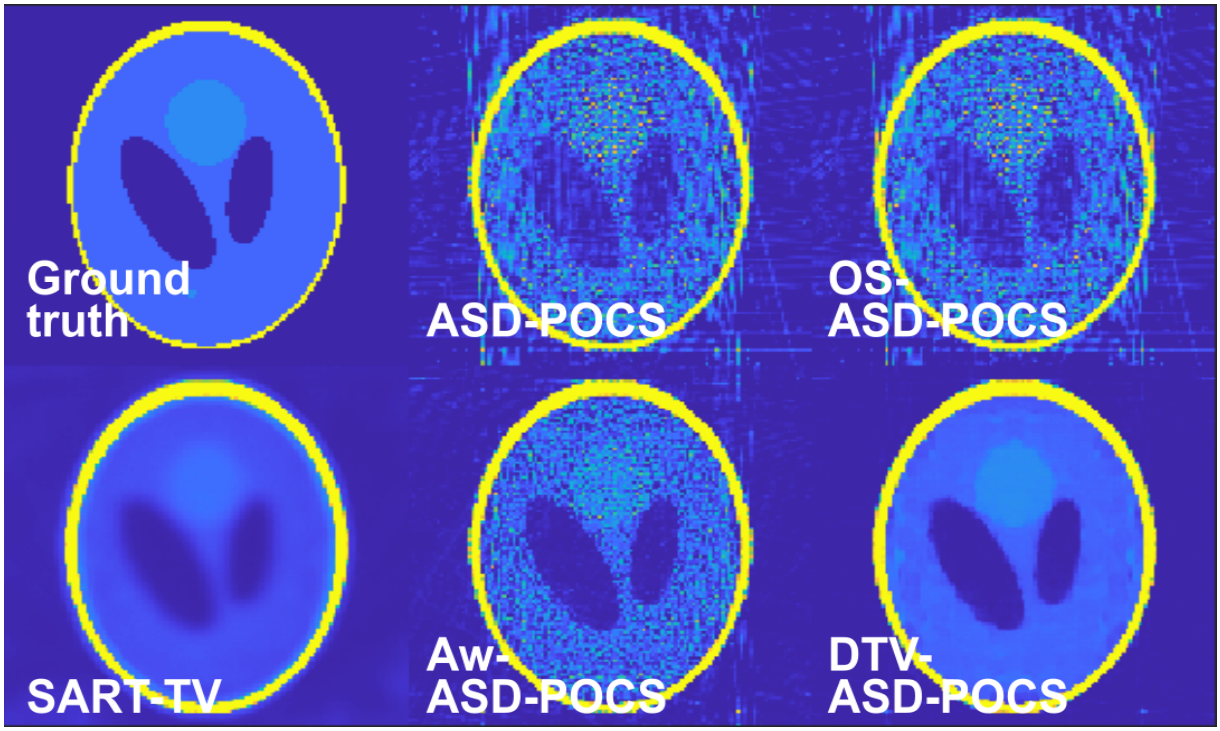


Figure 5.3: Reconstructed CT scan images for Shepp-Logan phantom, no noise, full-view angle, $M = 120$ projections, scaled to $[0, 0.1]$.

	SNR, dB	$RMSE$	T, min	N_{iter}	$\|AU^K - f\|_2/N^2$
ASD-POCS	15.0746	0.0255	41.8	97	0.0048
OS-ASD-POCS	14.3309	0.0278	32.6	48	0.0048
Aw-ASD-POCS	16.8855	0.0169	44.1	79	0.0047
DTV-ASD-POCS	16.9420	0.0166	46.4	62	0.0047
SART-TV	14.2851	0.0264	41.1	119	0.0049

Table 5.1: Shepp-Logan phantom CT reconstruction by TV minimization based methods, convergence test with $\varepsilon = 5 \cdot 10^{-3}$, $M = 120$, no noise. The U^K denotes the reconstucted image obtained after N_{iter} iteration and $\|AU^K - f\|_2$ is its data fidelity.

The computational time of the ASD-POCS in the first test is 41.8 min. The rest of the algorithms ran for 32-47 min. Among the considered algorithms the OS-ASD-POCS and

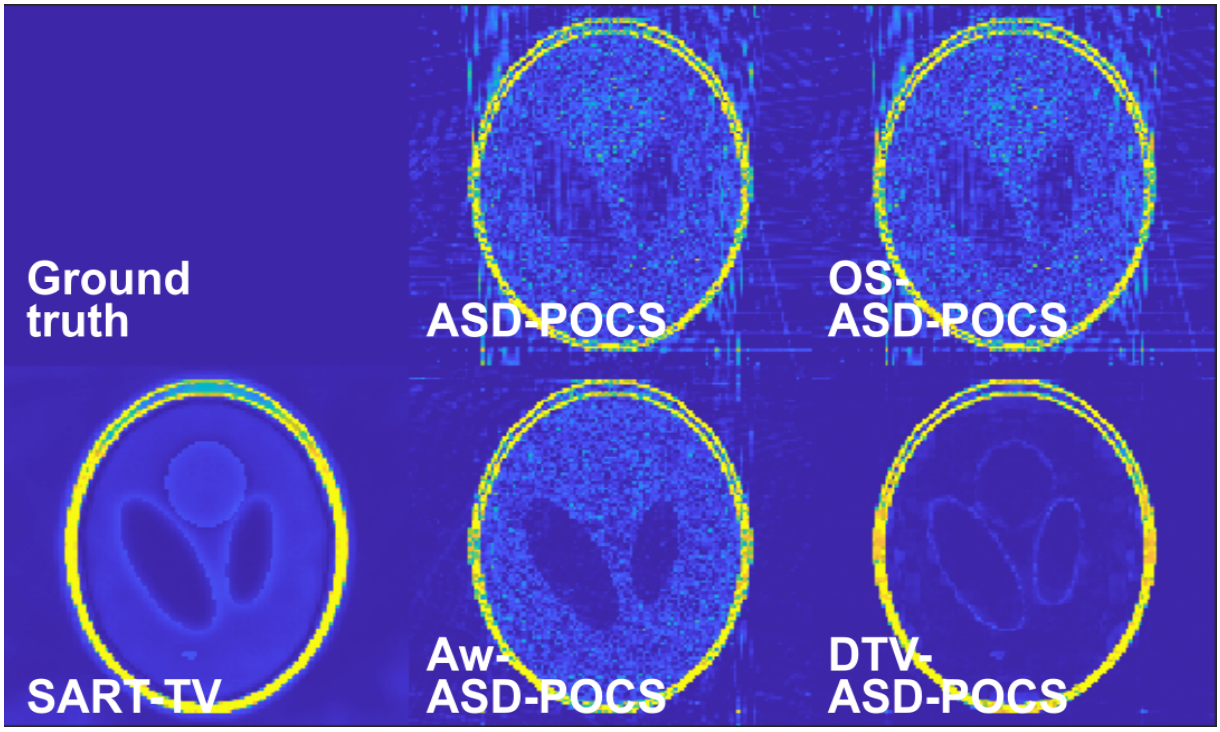


Figure 5.4: Absolute error of reconstructed CT scan images for Shepp-Logan phantom, no noise, full-view angle, $M = 120$ projections, scaled to $[0, 0.1]$.

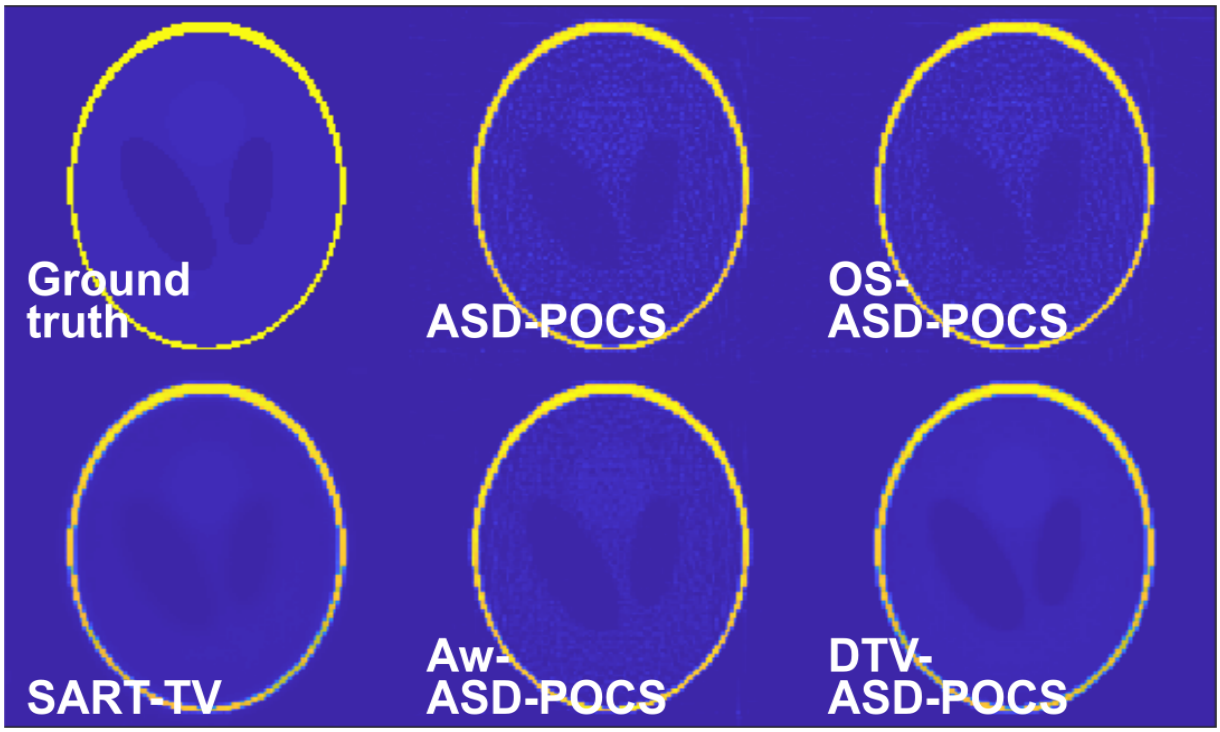


Figure 5.5: Reconstructed CT scan images for Shepp-Logan phantom, no noise, full-view angle, $M = 120$ projections, no scaling applied.

SART-TV are the fastest and least accurate, while DTV-ASD-POCS is the most accurate and the slowest one to converge. Note that it took DTV-ASD-POCS the least number of iterations to converge (Table 5.1), almost twice as little the number of iterations of ASD-POCS. However each iteration of DTV-ASD-POCS requires more time to compute than

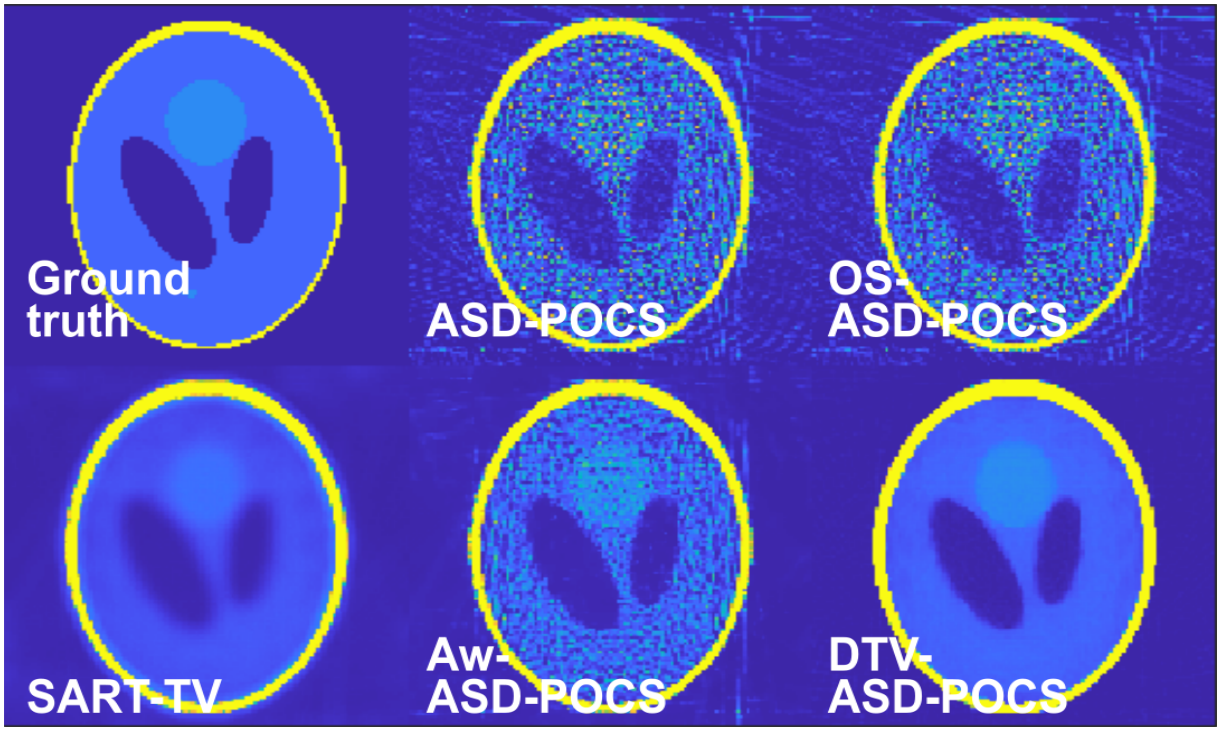


Figure 5.6: Reconstructed CT scan images for Shepp-Logan phantom, no noise, full-view angle, $M = 120$ projections, scaled to $[0, 0.1]$.

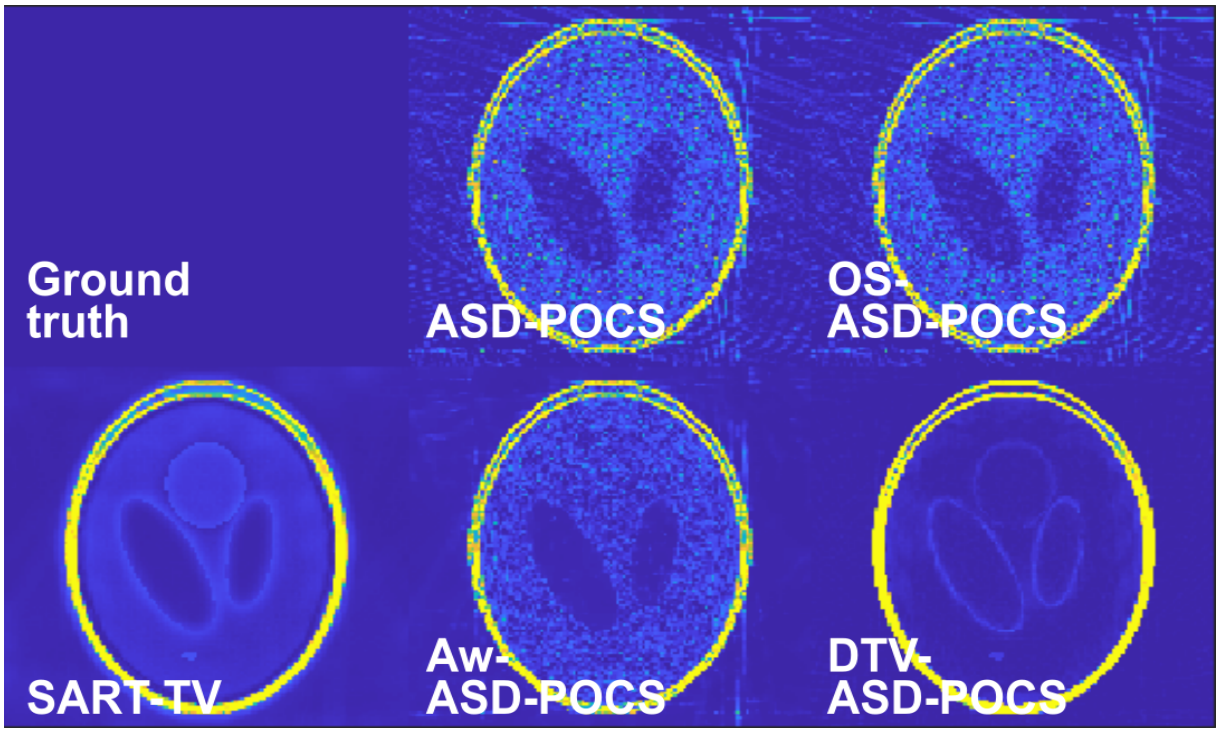


Figure 5.7: Absolute error of reconstructed CT scan images for Shepp-Logan phantom, no noise, full-view angle, $M = 120$ projections, scaled to $[0, 0.1]$.

for other algorithms.

The computational time of the ASD-POCS algorithm in the following tests is approximately 14 min for both experiments, i.e. with and without noise. The rest of the algorithms ran for 10-16 min. Among the considered algorithms the OS-ASD-POCS is the fastest and

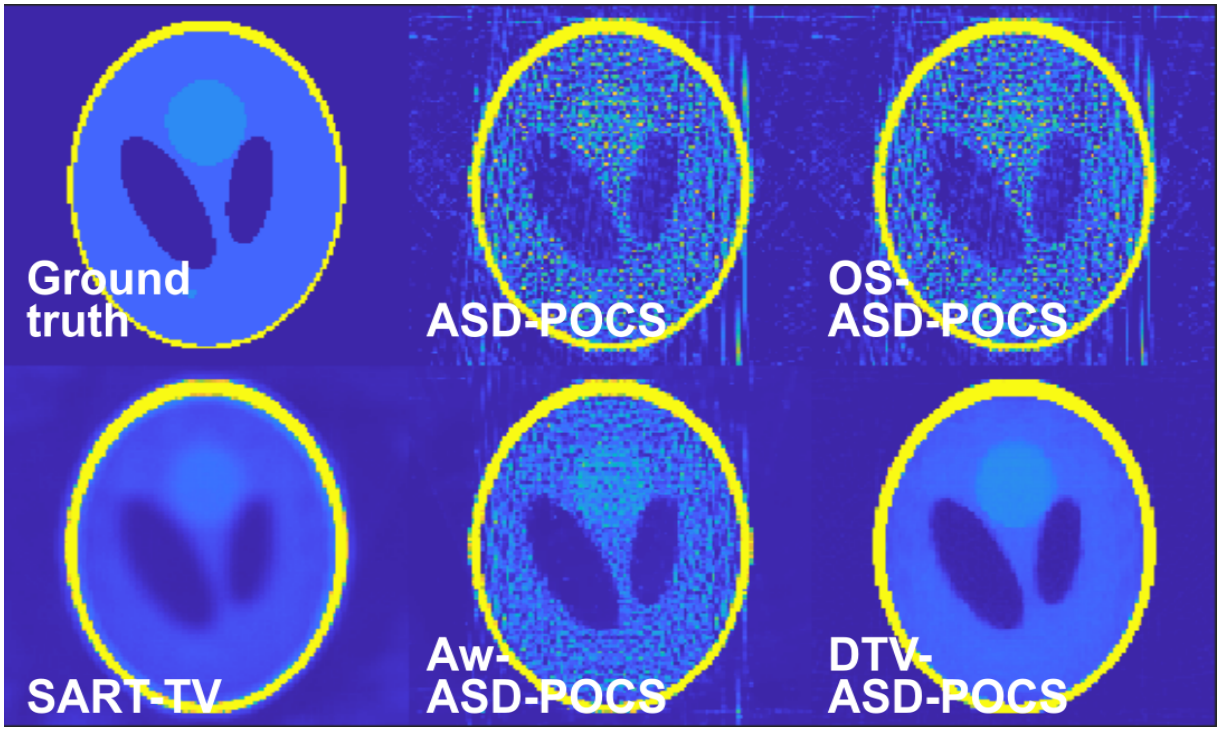


Figure 5.8: Reconstructed CT scan images for Shepp-Logan phantom, 10% noise, full-view angle, $M = 120$ projections, scaled to $[0, 0.1]$.

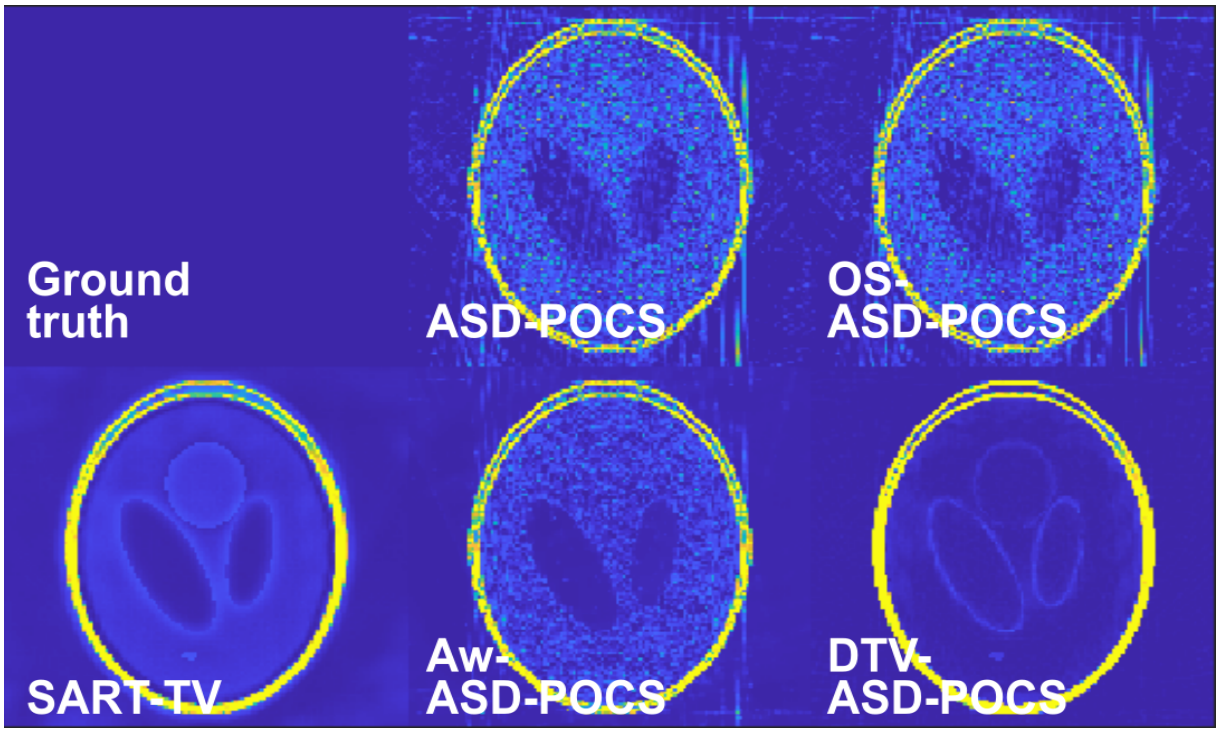


Figure 5.9: Absolute error of reconstructed CT scan images for Shepp-Logan phantom, 10% noise, full-view angle, $M = 120$ projections, scaled to $[0, 0.1]$.

least accurate, while DTV-ASD-POCS is the most accurate. Since the Shepp-Logan phantom image has low contrast it is hard to visually inspect the difference in the reconstructed images. We scale the image to the interval $[0, 0.1]$, i.e. we find U_max the maximal value of U, and map all other pixels to $[0, 0.1U_{max}]$ interval. We plot the normalized version in

	$M = 120$, no noise			$M = 120$, 10% noise		
	SNR, dB	$RMSE$	T, min	SNR, dB	$RMSE$	T, min
ASD-POCS	12.6025	0.0483	14.3	12.5675	0.0508	14.2
OS-ASD-POCS	11.2790	0.0504	10.8	11.1991	0.0510	11.1
Aw-ASD-POCS	14.2532	0.0373	13.1	14.2268	0.0374	13.5
DTV-ASD-POCS	14.3832	0.0370	14.0	14.3631	0.0371	14.4
SART-TV	12.0001	0.0461	13.8	11.9830	0.0462	14.0

Table 5.2: Shepp-Logan phantom CT reconstruction by TV minimization based methods.

Figure 5.6. We use the shorthand notation "scaled to $[0, 0.1]$" to state that the scaling was applied as described above. Then we plot the absolute error, i.e. $|U - \hat{U}|$ scaled to $[0, 0.1]$ in Figure 5.7. The absolute error for noisy data is shown in Figure 5.9.

We observe there is not only a difference in accuracy but a qualitative difference between the images obtained with considered algorithms and the proposed DTV-ASD-POCS algorithm. More specifically, the proposed method delivers a reconstructed image that does not have the line artifacts, that are clearly visible in Figures 5.6 and 5.8 for ASD-POCS, OS-ASD-POCS, and less severe line artifacts with Aw-ASD-POCS. We note here that these artifacts are not due to the noise, as they are present on both noisy and noiseless data. We also note that line artifacts appear only for POCS algorithms and are not present on the image reconstructed with SART-TV. While SART-TV images are artifact-free, they have significantly lower contrasts and no sharp edges when compared to POCS algorithms. The line artifacts are due to the sparsity of measured data, they only appear for conventional POCS algorithms and not for the proposed method.

Finally, we see that DTV-ASD-POCS and SART-TV images are much less noisy for the chosen noisy data, though SART-TV results in a loss of contrast. The difference in image quality and the presence of artifacts can be observed through SNR and $RMSE$ values, see Table 5.1. While the SNR and $RMSE$ for Aw-ASD-POCS and DTV-ASD-POCS are quite close, we will see in the following experiments that this is not always the case, as the performance of the Aw-ASD-POCS depends on the tuning of the parameter σ and can quickly degrade for other images.

Head phantom.

In this example we use a RANDO head phantom dataset, which is a high-quality copy of a real human head. The dataset was obtained in Christie Hospital in Manchester and is available with the TIGRE software package. It consists of 360 projections over a full rotation. The dataset contains a set of noisy measurements together with scan parameters set up for head low-dose CT, i.e. low-intensity X-rays (exact parameters of which are unknown). In the absence of the exact solution, we use the full dataset to generate a 256-by-256 pixel image via 50 iterations of ASD-POCS, which we use as a ground truth. Then we use a limited number of projections that fall into a 120 degree angle. We limit the angle of measurements to compare the performance of algorithms on incomplete data. There is no need to add noise to the data, as it was measured in the experiment and already contains noise.

We compare the performance of five imaging methods using limited view angle (120 projections over 120 degree view angle) data and sparse (90 projections over 120 degree

view angle) data. We show the reconstructed images in Figures 5.10-5.13 and report the total computational time T, SNR, and $RMSE$ in Table 5.3.

	$M = 90$			$M = 120$		
	SNR, dB	$RMSE$	T, min	SNR, dB	$RMSE$	T, min
ASD-POCS	12.0018	0.0935	9.9	12.2102	0.0806	13.7
OS-ASD-POCS	11.4943	0.0965	8.3	12.0323	0.0869	12.5
Aw-ASD-POCS	12.8912	0.0737	10.1	13.7062	0.0682	13.8
DTV-ASD-POCS	13.1603	0.0699	10.2	14.0942	0.0679	14.9
SART-TV	12.6924	0.0712	10.3	13.2655	0.0684	14.2

Table 5.3: RANDO head CT reconstruction by TV minimization based methods.

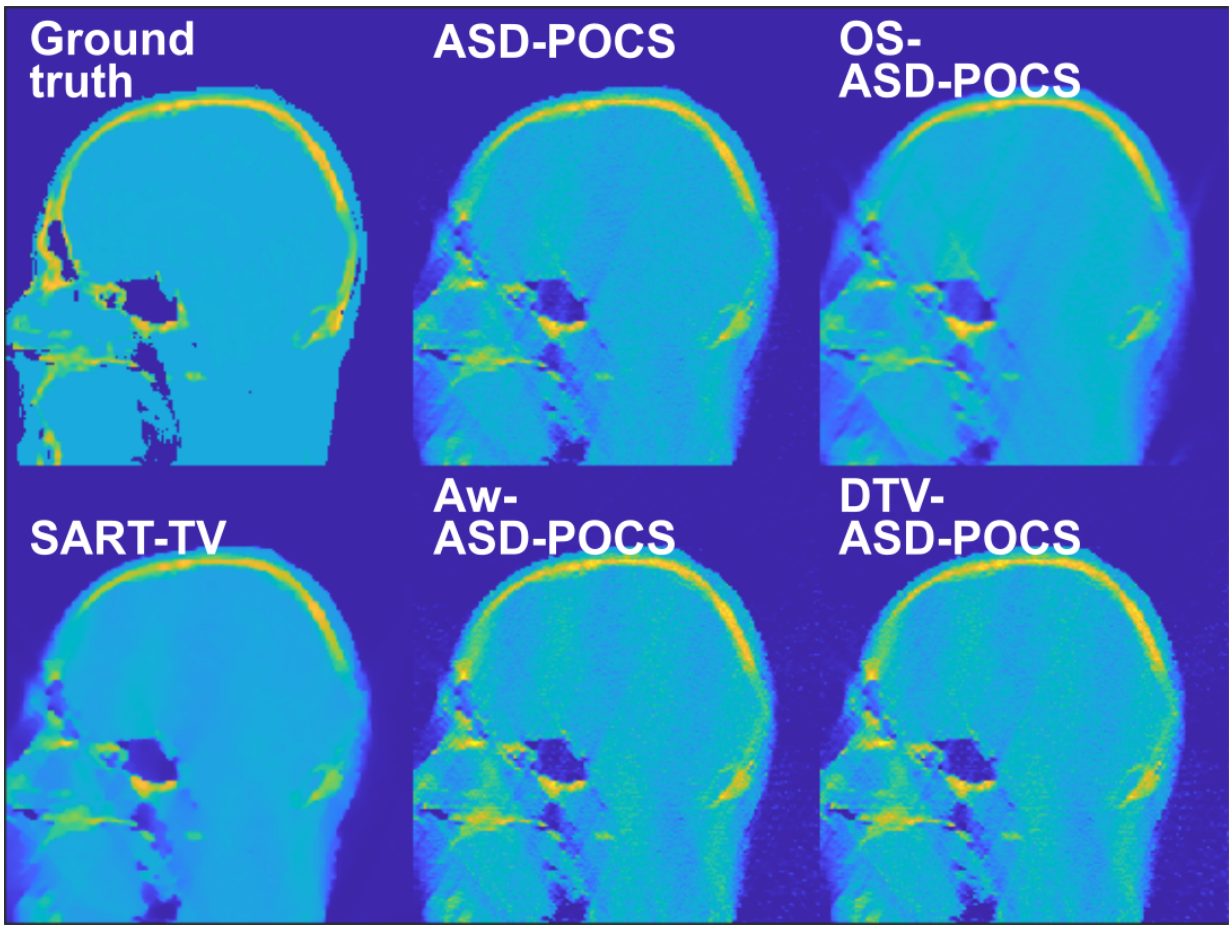


Figure 5.10: RANDO head CT scan images using $M = 120$ projections, over a 120 degree view-angle.

The computational time for the SART-TV is the greatest, while DTV-ASD-POCS provides the best reconstruction in terms of SNR and $RMSE$, with approximately $30\% - 40\%$ smaller errors. However, the quality of reconstruction is just slightly better than that of other POCS algorithms. The reconstructed images for RANDO head CT scan images using $M = 120$ projections, over 120 degree view-angle are depicted in Figure 5.10, and

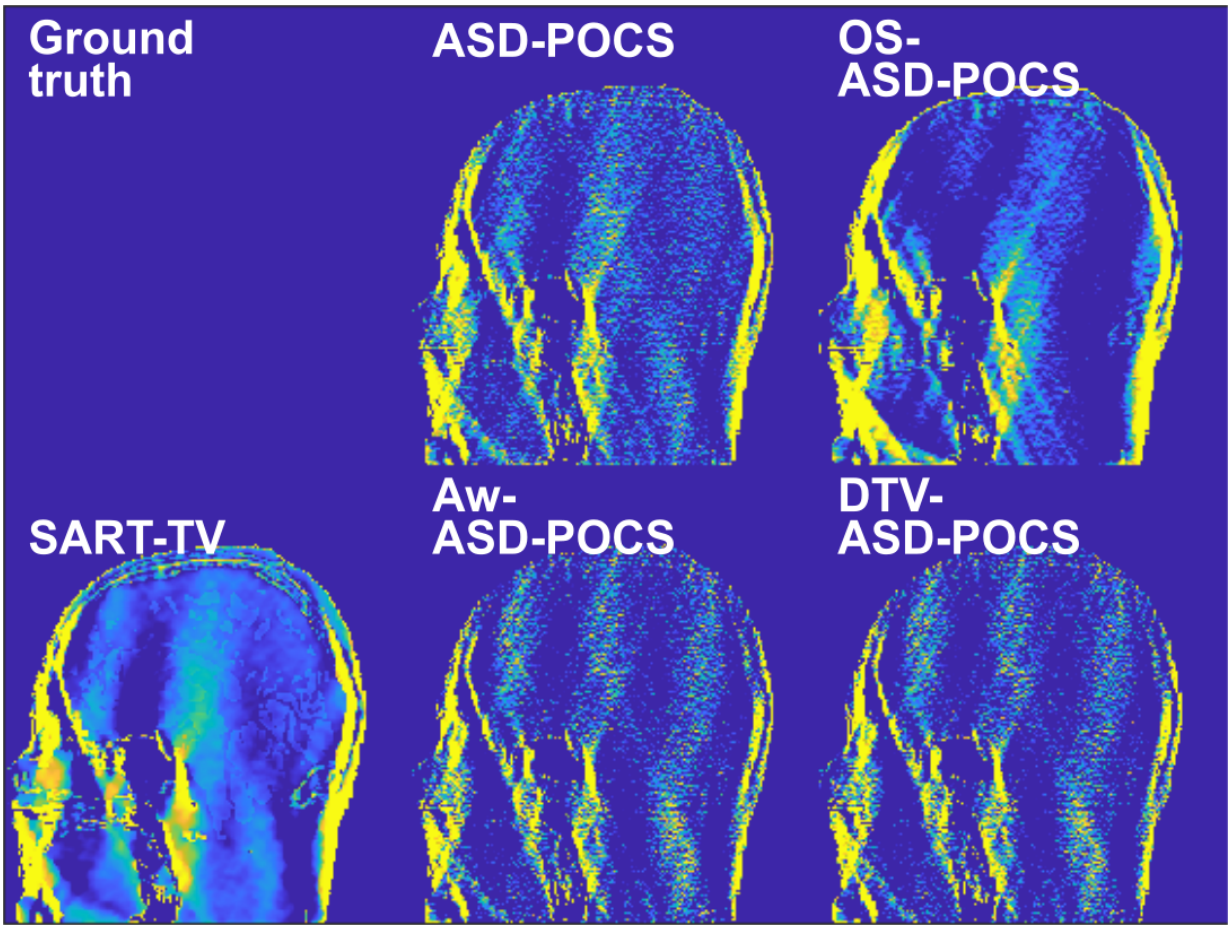


Figure 5.11: Absolute error of reconstructed RANDO head CT scan, using $M = 120$ projections, over a 120 degree view-angle, scaled to $[0, 0.1]$.

we plot the absolute error in Figure 5.11. The reconstructed images for RANDO head CT scan images using $M = 90$ projections, over 120 degree view-angle are depicted in Figure 5.12 and the absolute error in Figure 5.13.

We observe that it is impossible to eliminate the artifacts in this case due to the limited view angle. However, there are fewer blurry artifacts in the lower part of the image with the proposed DTV-ASD-POCS algorithm, than for other algorithms. We have less smearing of hollow regions with Aw-ASD-POCS and DTV-ASD-POCS than with other algorithms inside the head. It can also be seen in Figure 5.11 that DTV-ASD-POCS reduces error in the edges of the skull when compared to other algorithms. Aw-ASD-POCS delivers a similar solution in this case. It is important to note that artifacts protruding beyong the skull, which are due to the limited angle data are also significantly reduced when using Aw-ASD-POCS and DTV-ASD-POCS algorithms.

We compute T, SNR, and $RMSE$ for sparse data, i.e. RANDO head CT scan images obtained using $M = 90$ projections, over 120 degree view-angle and show the reconstructed images in Figure 5.12, we plot the absolute error, i.e. $|U - \hat{U}|$ in Figure 5.13. We note that in the case of limited-angle tomography, none of the algorithms used here can eliminate the artifacts in the images.

SophiaBeads dataset.

SophiaBeads dataset was collected specifically for testing and comparing reconstruction methods for X-ray computed tomography. The dataset was acquired using the Nikon

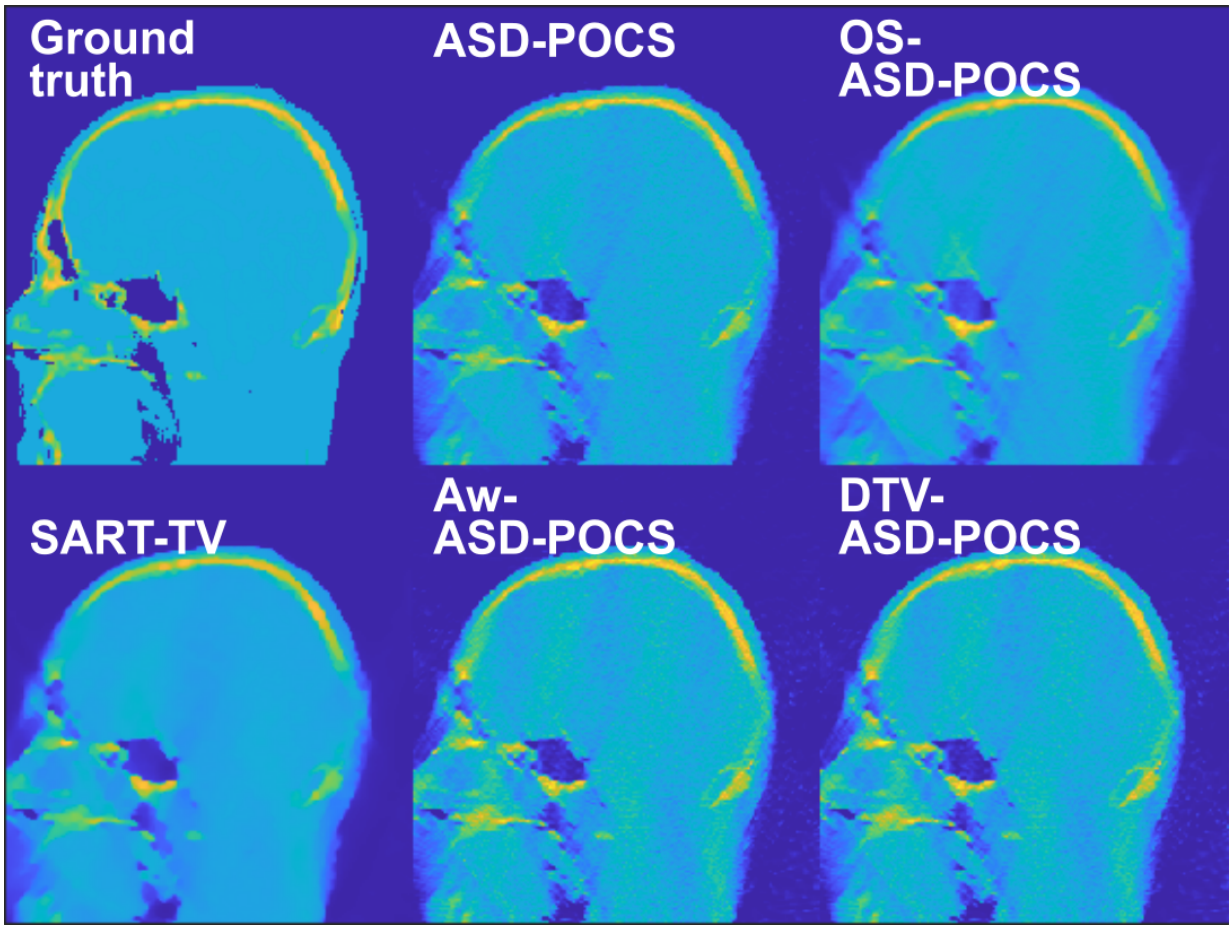


Figure 5.12: RANDO head CT scan images using $M = 90$ projections, over a 120 degree view-angle.

Custom Bay machine, located in the Manchester X-ray Imaging Facility. The dataset and the codes for processing the raw data are publicly available [35, 34].

The object is a plastic tube with a diameter 25 mm, filled with uniform Soda-Lime glass (SiO2-Na2O) beads with diameters of 2.5 mm. The dataset contains measurements of X-ray intensity. We use a higher resolution, i.e. 512 projections, and perform 15 iterations of conjugate gradient least squares (CGLS) algorithm to compute the ground truth image. We use it to compare performances of TV minimization-based methods with 256 and 128 projections. This allows us to demonstrate differences in errors for these methods in the absence of the exact solution and study their quality on sparse data.

First, we use 256 projections and full-view angle and then we use 128 projections over full-view angle. There is no need to add additional noise to the data, as it was measured and already contains noise. We use the ground truth image resized to 256×256 pixel images and then to 128×128 to match the size of the reconstructed images and to compute SNR and $RMSE$. We demonstrate the resized CGLS reconstruction together with the images reconstructed with TV minimization-based algorithms in Figure 5.14-5.17 and report the total computational time T, SNR and $RMSE$ in Table 5.4.

In this experiment, SART-TV provides the worst reconstruction both visually and according to the values of SNR, $RMSE$. The image quality for Aw-ASD-POCS significantly deteriorated. This can be explained by the fact that the same value of σ for Aw-ASD-POCS in all imaging experiments, and is not optimal. We can observe over-smoothing of

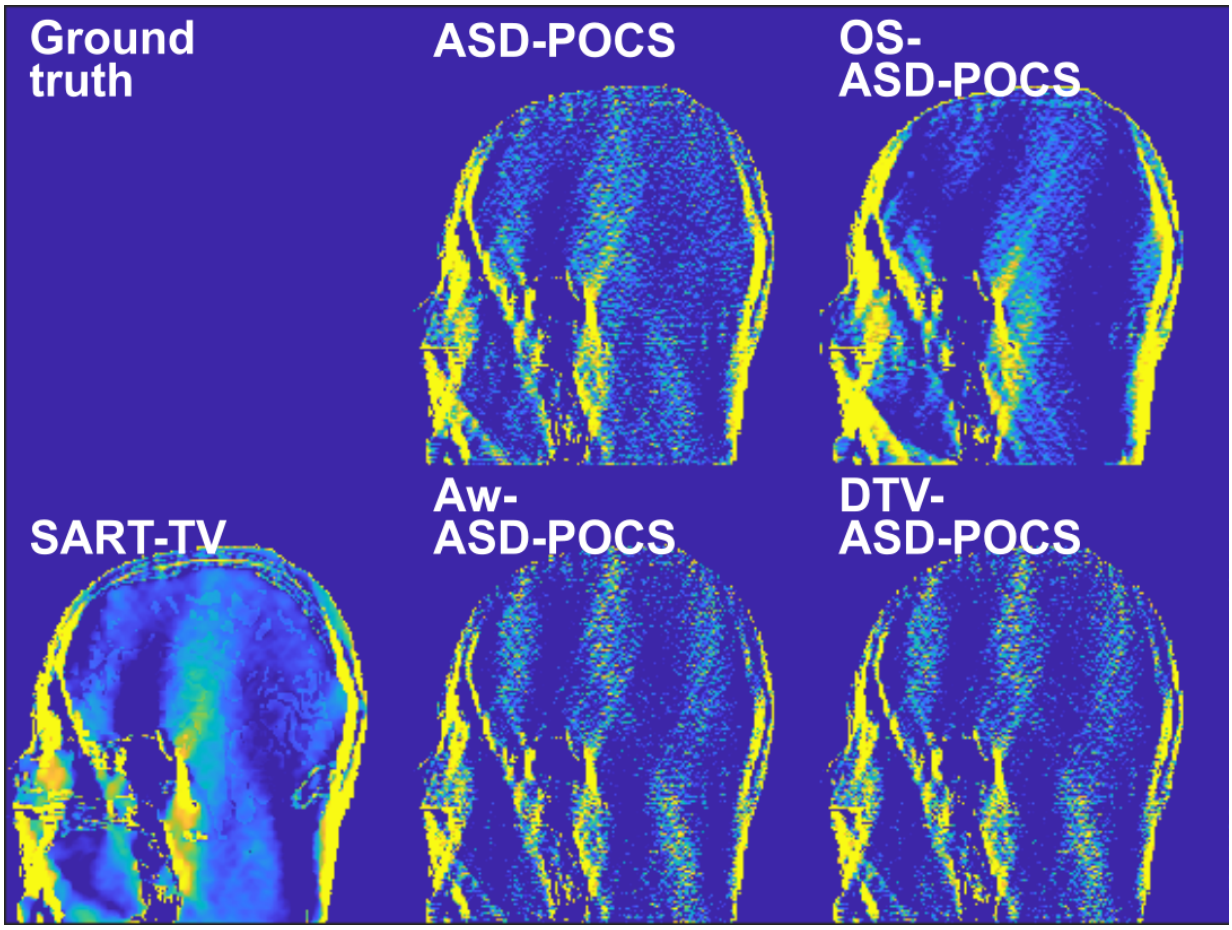


Figure 5.13: Absolute error of reconstructed RANDO head CT scan, using $M = 90$ projections, over a 120 degree view-angle, scaled to $[0, 0.1]$.

	$M = 128$			$M = 256$		
	SNR	$RMSE$	T, min	SNR	$RMSE$	T, min
ASD-POCS	4.8927	0.1134	15.7	5.4567	0.1134	31.1
OS-ASD-POCS	4.9230	0.1178	14.6	5.5051	0.1178	30.2
Aw-ASD-POCS	3.7672	0.0732	15.1	4.0697	0.0579	32.6
DTV-ASD-POCS	5.5581	0.0923	15.5	6.9682	0.0754	36.6
SART-TV	1.7308	0.2021	16.6	2.1568	0.1445	30.5

Table 5.4: Glass beads image reconstruction by TV minimization based methods.

bead edges here due to this fact and reduced SNR value. While in the previous experiments, the quality of reconstruction using Aw-ASD-POCS was comparable to that of the proposed method, in this experiment we observe a significant difference in quality between them.

The $RMSE$ value for Aw-ASD-POCS in this example did not change significantly, which can be explained by the fact that $RMSE$ is not very sensitive to the smoothing of the edges. If we compare the SNR value for example, we see that Aw-ASD-POCS reconstruction provides lower SNR than DTV-ASD-POCS, because SNR is more suitable to measure the accuracy reconstructions of object boundaries. DTV-ASD-POCS again

outperforms other methods and delivers at least 10% bigger SNR values. However, it is clear from Figures 5.14-5.17 that the bad choice of the smoothing parameter σ for Aw-ASD-POCS method here leads to a totally smeared reconstruction without visible edges.

Finally, the DTV-ASD-POCS method among other POCS methods provides a reconstruction that suppresses the line artifacts the most, as can be seen in Figures 5.14, 5.16.

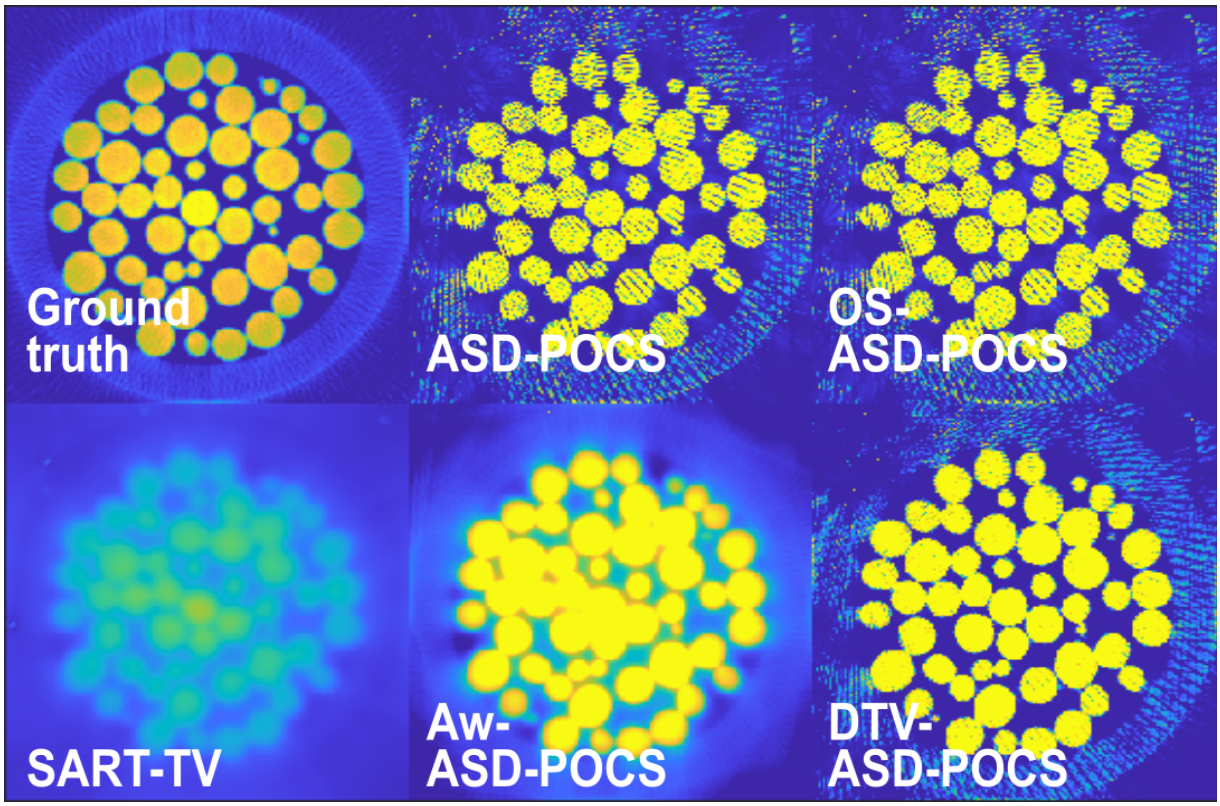


Figure 5.14: Glass beads reconstructed CT scan images using $M = 256$ projections, full-view angle, scaled to $[0, 0.25]$.

5.5 Results

In the numerical tests of Section 5.4, we have considered three different cases for CT image reconstruction problems. Our main goal was to demonstrate the efficiency of the proposed algorithm and qualitative improvement in the reconstructions obtained with DTV-ASD-POCS when compared to known POCS-type methods.

In the first experiment, we aim to reconstruct the Shepp-Logan phantom image given sparse and noisy data. We use the ground truth image to compute the measurements f, then we use this data with and without added noise. We observe in both scenarios that the DTV-ASD-POCS algorithm eliminates line artifacts and improves the overall quality of reconstructions. Moreover, among POCS-type algorithms, DTV-ASD-POCS is the only one that achieves reconstruction without line artifacts and sharp edges. We note that even though all algorithms we consider here converge for the given data tolerance ε, they converge to different approximations of U and we should compare $RMSE$ and SNR as true measures of reconstructed image quality.

In the second example, we use the RANDO head phantom and test the reconstruction quality for limited-angle CT. We observe comparable performance of the DTV-ASD-POCS

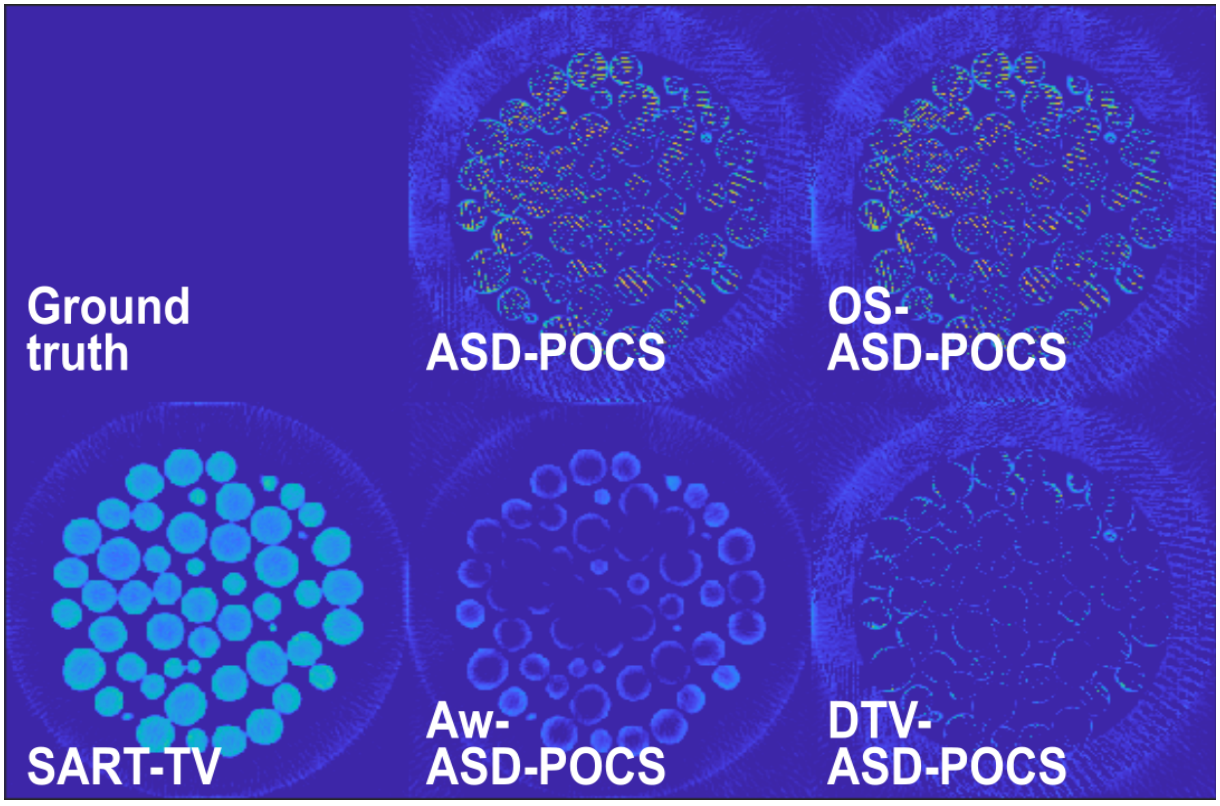


Figure 5.15: Absolute error of reconstructed glass beads CT scan, using $M = 256$ projections, full-view angle, scaled to $[0, 0.25]$.

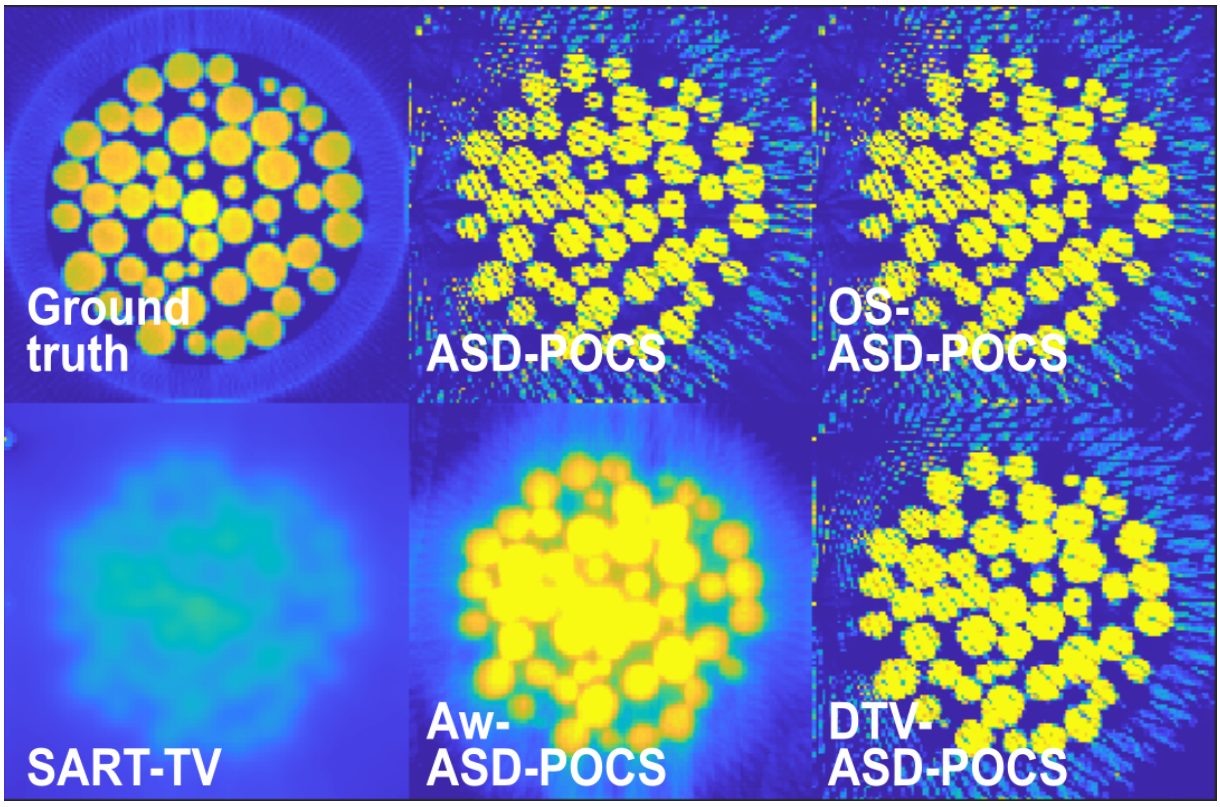


Figure 5.16: Glass beads reconstructed CT scan images using $M = 128$ projections, full-view angle, scaled to $[0, 0.25]$.

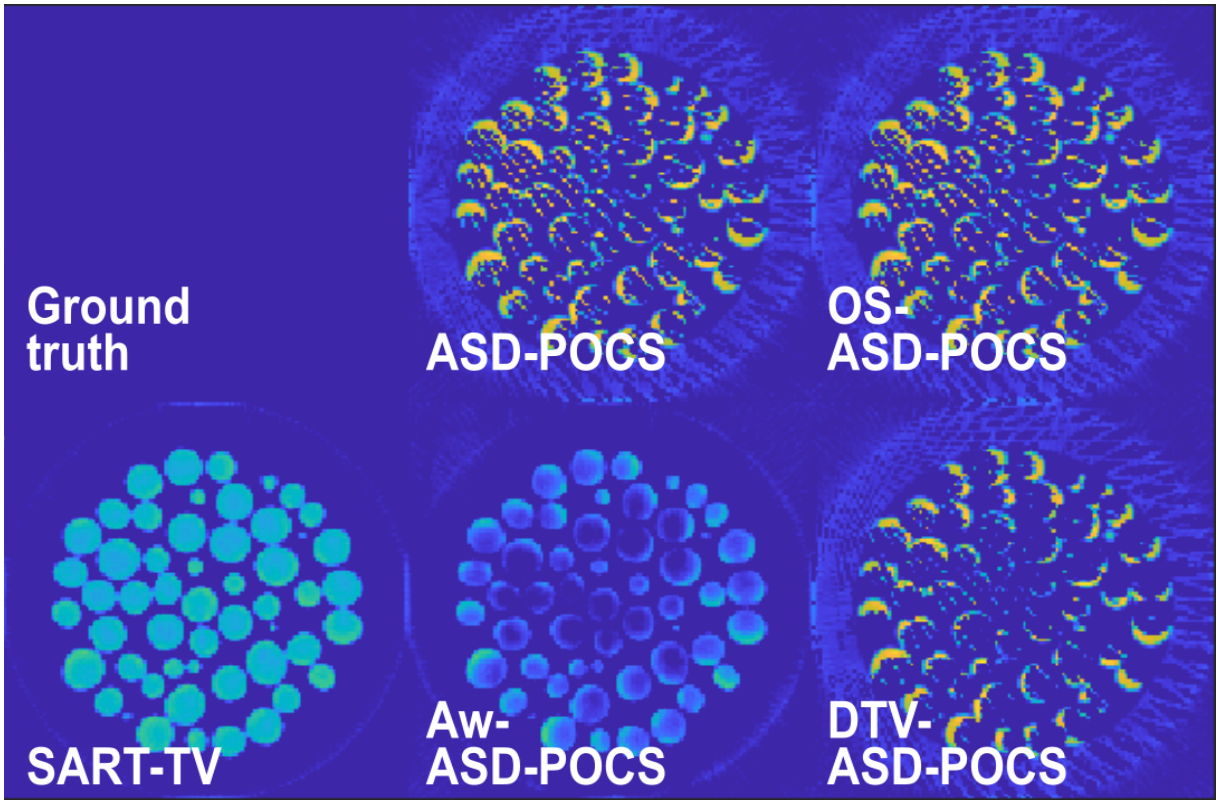


Figure 5.17: Absolute error of reconstructed glass beads CT scan, using $M = 128$ projections, full-view angle, scaled to $[0, 0.25]$.

method to that of ASD-POCS, OS-ASD-POCS, and Aw-ASD-POCS, with slightly better SNR and $RMSE$ values for the proposed algorithm, however at the cost of increased computational time when compared to other algorithms. Note, that in the limited angle tomography, the artifacts remain in the image no matter what definition of the gradient of the TV norm is used. We conclude that the change of TV definition alone in this scenario is not sufficient to achieve significantly better reconstruction results.

In the third example, the glass beads image was reconstructed from real experimental data. In this experiment, we use very sparse measurements to test the worst-case performance of the considered algorithms. We demonstrate that DTV-ASD-POCS achieves superior quality of reconstructions and similarly to the Shepp-Logan phantom significantly reduces artifacts and preserves sharp edges. In this case, we also see no artifacts for Aw-ASD-POCS reconstructions, however, the quality of the reconstructed images with this method is very low. This is due to the fact that we did not tune the parameter for this algorithm and therefore observe the over-smoothing effects.

We observe in all experiments that the change in the discrete TV definition leads to better reconstruction quality and more importantly it allows to suppress of artifacts in the reconstructed images, which was not possible with other definitions. We note here that the computational time can be further reduced using a multi-GPU approach and possibly a more efficient implementation of the algorithm.

5.6 Summary

In this chapter, we considered the application of the dual discrete TV to computed tomography image reconstruction for noisy and sparse data. We consider a number of projections onto convex sets imaging algorithms for TV minimization-based CT scan reconstruction and propose a new POCS-type algorithm that minimizes dual discrete TV of the image. Then we developed a GPU-accelerated version for fast computation of the dual discrete TV (Algorithm 2). Finally, we compare DTV-ASD-POCS to the other POCS algorithms on several examples, including both simulated and real experimentally measured data. We show numerically the proposed algorithm's effectiveness and discuss its potential advantages over existing methods.

Chapter 6

Conclusion

In this thesis, we have presented discrete total variation as a tool for ensuring the nonlinear stability of high-order numerical schemes for scalar conservation laws. We have reviewed the theory behind the total variation diminishing numerical methods in one space dimension and made considerable extensions for the two-dimensional schemes both numerically and theoretically.

In Chapter 2 we provided background theory for total variation and its discretization and introduced the recent developments in this field, including the dual discrete total variation. Several examples of the behavior of the TV functional discretizations on finite grids have been demonstrated. We also made a clear distinction between the properties of one-dimensional discrete total variation and that of the two-dimensional ones. We establish several new theoretical results for the dual discrete total variation.

Chapter 3 contains the main numerical and theoretical evidence for the existence of second-order TVD numerical schemes in two space dimensions. We provide substantial numerical evidence that demonstrates that the KT scheme as well as other schemes of a certain class, equipped with a limiter retain the theoretical convergence rate for smooth solutions and eliminate spurious oscillations in the presence of discontinuities. We tested three different discrete TV definitions for the solutions of the KT scheme on a number of numerical examples and we demonstrated that KT and some special numerical schemes do not increase the dual discrete total variation. It is important to note here that we used different settings and several scalar laws to establish that, as well as the fact that conventional discrete total variations do increase for these schemes. This led us to formulate a hypothesis, that the dual discrete TV presented in Chapter 2 allows us to avoid the conjecture of J. Goodman and R. LeVeque on the accuracy of TVD schemes in two dimensions. We analyze the scheme to establish sufficient conditions for the numerical method to have the TVD property in two spatial dimensions and prove them under certain simplifying assumptions.

The numerical results of Chapter 3 rely on the stable and accurate estimation of the conventional discrete TV and dual discrete TV value of the numerical solution. More specifically the dual TV can only be computed via an iterative process as a solution to an optimization problem. In Chapter 4, we presented the general framework for the ADMM algorithm to find the value of dual discrete TV and we also introduce the simpler and more efficient APGM algorithm and its improved version for dual TV computation. We then test its accuracy and convergence. The performance of the modified APGM algorithm proposed here is compared to that of the original APGM algorithm. In this process, we also arrive at a good heuristic rule for the choice of algorithm hyperparameters. As a result,

the total computation time of the modified APGM is drastically reduced when compared to the original one. Finally, a parallel version of the proposed algorithm is developed and its implementation is used to formulate a new imaging algorithm in Chapter 5 of the manuscript.

In Chapter 5, we have discussed a different area of interest as an application of discrete total variation – image reconstruction from noisy and incomplete measurements by TV minimization. The problem of computed tomography imaging was considered and a new algorithm based on the new discrete TV definition was proposed. We use the parallel version of the modified APGM of Chapter 4 and compare it with other state-of-the-art TV minimization-based projection-onto-convex-sets algorithms. We conduct numerical tests to demonstrate the superiority of the new imaging method in low radiation and limited angle imaging configurations.

Future work involves the development of theory for high-order TVD schemes in multiple space dimensions using the new discrete TV definition. Including, but not limited to possible extensions for general limited schemes, an analogue of Harten's lemma, and possible extensions to three spatial dimensions. It would be interesting to study extensions of the algorithm to compute dual discrete TV in three dimensions or find an alternative approach to computing TV and establish its convergence. Finally, designing more efficient imaging methods based on the dual discrete TV is of great interest.

References

[1] Feriel Abboud, Emilie Chouzenoux, Jean-Christophe Pesquet, Jean-Hugues Chenot, and Louis Laborelli. An alternating proximal approach for blind video deconvolution. *Signal Processing: Image Communication*, 70:21–36, 2019.

[2] Vadym Aizinger, Adam Kosík, Dmitri Kuzmin, and Balthasar Reuter. Anisotropic slope limiting for Discontinuous Galerkin methods. *International Journal for Numerical Methods in Fluids*, 84(9):543–565, 2017.

[3] Anders Andersen and Avinash Kak. Simultaneous algebraic reconstruction technique (SART): a superior implementation of the ART algorithm. *Ultrasonic imaging*, 6(1):81–94, 1984.

[4] Paul Arminjon and Claude Beauchamp. Numerical solution of Burgers' equations in two space dimensions. *Computer Methods in Applied Mechanics and Engineering*, 19(3):351–365, 1979.

[5] Sören Bartels. Total variation minimization with finite elements: convergence and iterative solution. *SIAM Journal on Numerical Analysis*, 50(3):1162–1180, 2012.

[6] Timothy Barth and Dennis Jespersen. The design and application of upwind schemes on unstructured meshes. In *27th Aerospace sciences meeting*, page 366, 1989.

[7] Timothy Barth and Mario Ohlberger. *Finite volume methods: foundation and analysis*. Encyclopedia of Computational Mechanics Second Edition, 2003.

[8] Amir Beck and Marc Teboulle. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM Journal on Imaging Sciences*, 2(1):183–202, 2009.

[9] Ander Biguri, Manjit Dosanjh, Steven Hancock, and Manuchehr Soleimani. TIGRE: a matlab-gpu toolbox for cbct image reconstruction. *Biomedical Physics & Engineering Express*, 2(5):055010, 2016.

[10] Ander Biguri, Reuben Lindroos, Robert Bryll, Hossein Towsyfyan, Hans Deyhle, Ibrahim El khalil Harrane, Richard Boardman, Mark Mavrogordato, Manjit Dosanjh, Steven Hancock, et al. Arbitrarily large tomography with iterative algorithms on multiple gpus using the TIGRE toolbox. *Journal of Parallel and Distributed Computing*, 146:52–63, 2020.

[11] Rupak Biswas, Karen Devine, and Joseph Flaherty. Parallel, adaptive finite element methods for conservation laws. *Applied Numerical Mathematics*, 14(1-3):255–283, 1994.

[12] Peter Blomgren and Tony Chan. Color TV: total variation methods for restoration of vector-valued images. *IEEE transactions on image processing*, 7(3):304–309, 1998.

[13] Daniel Boley. Local linear convergence of the alternating direction method of multipliers on quadratic or linear programs. *SIAM Journal on optimization*, 23(4):2183–2207, 2013.

[14] François Bouchut, Christian Bourdarias, and Benoit Perthame. A MUSCL method satisfying all the numerical entropy inequalities. *Mathematics of Computation*, 65(216):1439–1461, 1996.

[15] François Bouchut and Benoıt Perthame. Kruzkov's estimates for scalar conservation laws revisited. *Transactions of the American Mathematical Society*, 350(7):2847–2870, 1998.

[16] Ronald Bracewell. *Two-dimensional imaging*. Prentice-Hall, Inc., 1995.

[17] James Bramble and Xuejun Zhang. The analysis of multigrid methods. *Handbook of numerical analysis*, 7:173–415, 2000.

[18] Alberto Bressan. *Hyperbolic systems of conservation laws: the one-dimensional Cauchy problem*, volume 20. Oxford Lecture Mathematics and, 2000.

[19] Anne Burbeau, Pierre Sagaut, and Ch-H Bruneau. A problem-independent limiter for high-order runge–Kutta Discontinuous Galerkin methods. *Journal of Computational Physics*, 169(1):111–150, 2001.

[20] Corentin Caillaud and Antonin Chambolle. *Error estimates for finite differences approximations of the total variation*, volume 43(2). IMA, 2020.

[21] Neal L Carothers. *Real analysis*. Cambridge University Press, 2000.

[22] Vicent Caselles, Antonin Chambolle, and Matteo Novaga. The discontinuity set of solutions of the tv denoising problem and some extensions. *Multiscale modeling & simulation*, 6(3):879–894, 2007.

[23] Noel Chalmers and Lilia Krivodonova. A robust CFL condition for the Discontinuous Galerkin method on triangular meshes. *Journal of Computational Physics*, 403:109095, 2020.

[24] Antonin Chambolle, Stacey Levine, and Bradley Lucier. An upwind finite-difference method for total variation–based image smoothing. *SIAM Journal on Imaging Sciences*, 4(1):277–299, 2011.

[25] Antonin Chambolle and Thomas Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1):120–145, 2011.

[26] Antonin Chambolle and Thomas Pock. Crouzeix–Raviart approximation of the total variation on simplicial meshes. *Journal of Mathematical Imaging and Vision*, 62(6):872–899, 2020.

[27] Antonin Chambolle and Thomas Pock. Chapter 6 - approximating the total variation with finite differences or finite elements. In Andrea Bonito and Ricardo H. Nochetto, editors, *Geometric Partial Differential Equations - Part II*, volume 22 of *Handbook of Numerical Analysis*, pages 383–417. Elsevier, 2021.

[28] Antonin Chambolle and Thomas Pock. Learning consistent discretizations of the total variation. *SIAM Journal on Imaging Sciences*, 14(2):778–813, 2021.

[29] Tony Chan and Jianhong Shen. *Image processing and analysis: variational, PDE, wavelet, and stochastic methods*. SIAM, 2005.

[30] Guang-Hong Chen, Jie Tang, Brian Nett, Zhihua Qi, Shuai Leng, and Timothy Szczykutowicz. Prior image constrained compressed sensing (PICCS) and applications in x-ray computed tomography. *Current Medical Imaging*, 6(2):119–134, 2010.

[31] Tianheng Chen and Chi-Wang Shu. Entropy stable high order discontinuous /galerkin methods with suitable quadrature rules for hyperbolic conservation laws. *Journal of Computational Physics*, 345:427–461, 2017.

[32] Zhiqiang Chen, Xin Jin, Liang Li, and Ge Wang. A limited-angle ct reconstruction method based on anisotropic tv minimization. *Physics in Medicine & Biology*, 58(7):2119, 2013.

[33] James A Clarkson and C Raymond Adams. On definitions of bounded variation for functions of two variables. *Transactions of the American Mathematical Society*, 35(4):824–854, 1933.

[34] Sophia Coban. Sophiabeads datasets project codes, 2015.

[35] Sophia Coban. Sophiabeads datasets project documentation and tutorials. *MIMS Eprints*, 2015.

[36] Bernardo Cockburn, Suchung Hou, and Chi-Wang Shu. The Runge-Kutta local projection discontinuous Galerkin finite element method for conservation laws. iv. The multidimensional case. *Mathematics of Computation*, 54(190):545–581, 1990.

[37] Bernardo Cockburn and Chi-Wang Shu. TVB Runge-Kutta local projection Discontinuous Galerkin finite element method for conservation laws. ii. general framework. *Mathematics of computation*, 52(186):411–435, 1989.

[38] Bernardo Cockburn and Chi-Wang Shu. TVB Runge-Kutta local projection Discontinuous Galerkin finite element method for conservation laws. ii. general framework. *Mathematics of Computation*, 52(186):411–435, 1989.

[39] Rinaldo M Colombo, Magali Mercier, and Massimiliano D Rosini. Stability and total variation estimates on general scalar balance laws. *Commun. Math. Sci.*, 7(1):37–65, 2009.

[40] Laurent Condat. Discrete total variation: New definition and minimization. *SIAM Journal on Imaging Sciences*, 10(3):1258–1290, 2017.

[41] Michael Crandall and Andrew Majda. Monotone difference approximations for scalar conservation laws. *Mathematics of Computation*, 34(149):1–21, 1980.

[42] Edisson Savio de Goes Maciel and Cláudia Regina de Andrade. Comparison among unstructured TVD, ENO and UNO schemes in two-and three-dimensions. *Applied Mathematics and Computation*, 321:130–175, 2018.

[43] Wei Deng and Wotao Yin. On the global and linear convergence of the generalized alternating direction method of multipliers. *Journal of Scientific Computing*, 66(3):889–916, 2016.

[44] Jonathan Eckstein and Dimitri P Bertsekas. On the douglas—rachford splitting method and the proximal point algorithm for maximal monotone operators. *Mathematical programming*, 55:293–318, 1992.

[45] Lawrence C Evans. *Partial differential equations*, volume 19. American Mathematical Society, 2022.

[46] Lawrence C Evans and Ronald F Gariepy. *Measure theory and fine properties of functions*, volume 5. CRC press Boca Raton, 1992.

[47] Ulrik Fjordholm, Roger Käppeli, Siddhartha Mishra, and Eitan Tadmor. Construction of approximate entropy measure-valued solutions for hyperbolic systems of conservation laws. *Foundations of Computational Mathematics*, 17(3):763–827, 2017.

[48] Ulrik Fjordholm, Siddhartha Mishra, and Eitan Tadmor. Arbitrarily high-order accurate entropy stable essentially nonoscillatory schemes for systems of conservation laws. *SIAM Journal on Numerical Analysis*, 50(2):544–573, 2012.

[49] Ulrik Fjordholm, Siddhartha Mishra, and Eitan Tadmor. On the computation of measure-valued solutions. *Acta numerica*, 25:567–679, 2016.

[50] Wendell Fleming and Raymond Rishel. An integral formula for total gradient variation. *Archiv der Mathematik*, 11:218–222, 1960.

[51] Gerald Folland. *Real analysis: modern techniques and their applications*, volume 40. John Wiley & Sons, 1999.

[52] Jerome Friedman, Trevor Hastie, and Robert Tibshirani. Sparse inverse covariance estimation with the graphical lasso. *Biostatistics*, 9(3):432–441, JUL 2008.

[53] D. Gabay. Chapter IX Applications of the method of multipliers to variational inequalities. In *Studies in mathematics and its applications*, volume 15, pages 299–331. Elsevier, 1983.

[54] Andrew Giuliani and Lilia Krivodonova. Analysis of slope limiters on unstructured triangular meshes. *Journal of Computational Physics*, 374:1–26, 2018.

[55] Andrew Giuliani and Lilia Krivodonova. A moment limiter for the discontinuous Galerkin method on unstructured triangular meshes. *SIAM Journal on Scientific Computing*, 41(1):A508–A537, 2019.

[56] Andrew Giuliani and Lilia Krivodonova. A moment limiter for the Discontinuous Galerkin method on unstructured tetrahedral meshes. *Journal of Computational Physics*, 404:109106, 2020.

[57] Enrico Giusti and Graham Williams. *Minimal surfaces and functions of bounded variation*, volume 80. Springer, 1984.

[58] James Glimm and Peter Lax. *Decay of solutions of systems of nonlinear hyperbolic conservation laws*, volume 101. American Mathematical Soc., 1970.

[59] James Glimm, Xiao Lin Li, Yingjie Liu, and Ning Zhao. Conservative front tracking and level set algorithms. *Proceedings of the National Academy of Sciences*, 98(25):14198–14201, 2001.

[60] Edwige Godlewski and Pierre-Arnaud Raviart. *Numerical approximation of hyperbolic systems of conservation laws*, volume 118. Springer Science & Business Media, 2013.

[61] Jonathan Goodman and Randall LeVeque. On the accuracy of stable schemes for 2d scalar conservation laws. *Mathematics of computation*, 45(171):15–21, 1985.

[62] Jonathan Goodman and Randall LeVeque. A geometric approach to high resolution TVD schemes. *SIAM journal on numerical analysis*, 25(2):268–284, 1988.

[63] Richard Gordon, Robert Bender, and Gabor T Herman. Algebraic reconstruction techniques (art) for three-dimensional electron microscopy and x-ray photography. *Journal of theoretical Biology*, 29(3):471–481, 1970.

[64] Sigal Gottlieb, Chi-Wang Shu, and Eitan Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM review*, 43(1):89–112, 2001.

[65] Sergei Gudonov. A finite difference method for the numerical computation of discontinuous solutions of the equations of fluid dynamics. *Mat. Sb*, 1:271–306, 1959.

[66] Ami Harten. On a class of high resolution total-variation-stable finite-difference schemes. *SIAM Journal on Numerical Analysis*, 21(1):1–23, 1984.

[67] Ami Harten. High resolution schemes for hyperbolic conservation laws. *Journal of computational physics*, 135(2):260–278, 1997.

[68] Ami Harten and Stanley Osher. Uniformly high-order accurate nonoscillatory schemes. i. *SIAM Journal on Numerical Analysis*, 24(2):279–309, 1987.

[69] Ami Harten, Stanley Osher, Björn Engquist, and Sukumar R Chakravarthy. Some results on uniformly high-order accurate essentially nonoscillatory schemes. *Applied Numerical Mathematics*, 2(3-5):347–377, 1986.

[70] Michael Hintermüller, Carlos Rautenberg, and Jooyoung Hahn. Functional-analytic and numerical issues in splitting methods for total variation-based image reconstruction. *Inverse Problems*, 30(5):055014, 2014.

[71] Roger Horn and Charles Johnson. *Matrix analysis*. Cambridge university press, 2012.

[72] Ming Jiang and Ge Wang. Convergence of the simultaneous algebraic reconstruction technique (SART). *IEEE Transactions on Image Processing*, 12(8):957–961, 2003.

[73] Avinash Kak and Malcolm Slaney. *Principles of computerized tomographic imaging*. SIAM, 2001.

[74] Heinz-Otto Kreiss. Numerical methods for hyperbolic partial differential equations. In *Numerical methods for partial differential equations*, pages 213–254. Elsevier, 1979.

[75] Lilia Krivodonova. Limiters for high-order discontinuous Galerkin methods. *Journal of Computational Physics*, 226(1):879–896, 2007.

[76] Lilia Krivodonova and Alexey Smirnov. On the TVD property of second order methods for 2D scalar conservation laws. *preprint*, 2021. arXiv:2110.00067.

[77] Lilia Krivodonova and Alexey Smirnov. Discrete total variation in multiple spatial dimensions. *preprint*, 2024.

[78] Lilia Krivodonova and Alexey Smirnov. A novel total variation model for sparse low-dose CT reconstruction on GPUs. *preprint*, 2024.

[79] Stanislav Kružkov. First order quasilinear equations with several space variables. *Math. USSR. Sb*, 10:217–243, 1970.

[80] Alexander Kurganov and Doron Levy. A third-order semidiscrete central scheme for conservation laws and convection-diffusion equations. *SIAM Journal on Scientific Computing*, 22(4):1461–1488, 2000.

[81] Alexander Kurganov and Eitan Tadmor. New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations. *Journal of computational physics*, 160(1):241–282, 2000.

[82] Alexander Kurganov and Eitan Tadmor. New high-resolution central schemes for nonlinear conservation laws and convection–diffusion equations. *Journal of computational physics*, 160(1):241–282, 2000.

[83] Dmitri Kuzmin. A vertex-based hierarchical slope limiter for p-adaptive discontinuous Galerkin methods. *Journal of computational and applied mathematics*, 233(12):3077–3085, 2010.

[84] Dmitri Kuzmin. Slope limiting for Discontinuous Galerkin approximations with a possibly non-orthogonal taylor basis. *International Journal for Numerical Methods in Fluids*, 71(9):1178–1190, 2013.

[85] Dmitri Kuzmin. A new perspective on flux and slope limiting in Discontinuous Galerkin methods for hyperbolic conservation laws. *Computer Methods in Applied Mechanics and Engineering*, 373:113569, 2021.

[86] Culbert Laney. *Computational gasdynamics*. Cambridge university press, 1998.

[87] Peter D Lax. Hyperbolic systems of conservation laws II. In *Selected Papers Volume I*, pages 233–262. Springer, 2005.

[88] Triet Le, Rick Chartrand, and Thomas J Asaki. A variational approach to reconstructing images corrupted by poisson noise. *Journal of mathematical imaging and vision*, 27(3):257–263, 2007.

[89] Randall LeVeque. *Numerical methods for conservation laws*, volume 132. Springer, 1992.

[90] Randall J LeVeque. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002.

[91] Wanai Li, Yu-Xin Ren, Guodong Lei, and Hong Luo. The multi-dimensional limiters for solving hyperbolic conservation laws on unstructured grids. *Journal of Computational Physics*, 230(21):7775–7795, 2011.

[92] Pierre-Louis Lions and Bertrand Mercier. Splitting algorithms for the sum of two nonlinear operators. *SIAM Journal on Numerical Analysis*, 16(6):964–979, 1979.

[93] Yan Liu, Jianhua Ma, Yi Fan, and Zhengrong Liang. Adaptive-weighted total variation minimization for sparse data toward low-dose x-ray computed tomography image reconstruction. *Physics in Medicine & Biology*, 57(23):7923, 2012.

[94] David Luenberger. *Optimization by vector space methods*. John Wiley & Sons, 1997.

[95] Shiqian Ma. Alternating proximal gradient method for convex minimization. *Journal of Scientific Computing*, 68(2):546–572, 2016.

[96] Shev MacNamara and Gilbert Strang. Operator splitting. *Splitting methods in communication, imaging, science, and engineering*, pages 95–114, 2016.

[97] Antonio Marquina. Local piecewise hyperbolic reconstruction of numerical fluxes for nonlinear scalar conservation laws. *SIAM Journal on Scientific Computing*, 15(4):892–915, 1994.

[98] Orhan Mehmetoglu and Bojan Popov. Maximum principle and convergence of central schemes based on slope limiters. *Mathematics of Computation*, 81(277):219–231, 2012.

[99] Marijo Milicevic. *Finite element discretization and iterative solution of total variation regularized minimization problems and application to the simulation of rate-independent damage evolutions*. PhD thesis, Albert-Ludwigs-Universität Freiburg, 2018.

[100] Jiang Min, Hongwei Tao, Xinglong Liu, and Kai Cheng. A non-local total generalized variation regularization reconstruction method for sparse-view x-ray ct. *Measurement Science and Technology*, 35(4):045404, 2024.

[101] Charles Bradfield Morrey Jr. *Multiple integrals in the calculus of variations*. Springer Science & Business Media, 2009.

[102] Mirko Myllykoski, Roland Glowinski, T Karkkainen, and Tuomo Rossi. A new augmented lagrangian approach for L^1-mean curvature image denoising. *SIAM Journal on Imaging Sciences*, 8(1):95–125, 2015.

[103] Haim Nessyahu and Eitan Tadmor. Non-oscillatory central differencing for hyperbolic conservation laws. *Journal of computational physics*, 87(2):408–463, 1990.

[104] Stanley Osher and Sukumar Chakravarthy. High resolution schemes and the entropy condition. *SIAM Journal on Numerical Analysis*, 21(5):955–984, 1984.

[105] Stanley Osher and Eitan Tadmor. On the convergence of difference approximations to scalar conservation laws. *Mathematics of Computation*, 50(181):19–51, 1988.

[106] G. Passty. Ergodic convergence to a zero of the sum of monotone operators in Hilbert space. *Journal of Mathematical Analysis and Applications*, 72(2):383–390, 1979.

[107] Serge Piperno and Sophie Depeyre. Criteria for the design of limiters yielding efficient high resolution TVD schemes. *Computers & fluids*, 27(2):183–197, 1998.

[108] Kévin Polisano, Laurent Condat, Marianne Clausel, and Valérie Perrier. A convex approach to superresolution and regularization of lines in images. *SIAM Journal on Imaging Sciences*, 12(1):211–258, 2019.

[109] Bojan Popov and Ognian Trifonov. Order of convergence of second-order schemes based on the minmod limiter. *Mathematics of Computation*, 75(256):1735–1753, 2006.

[110] Pierre-Arnaud Raviart and Jean-Marie Thomas. A mixed finite element method for 2-nd order elliptic problems. In *Mathematical Aspects of Finite Element Methods: Proceedings of the Conference Held in Rome, December 10–12, 1975*, pages 292–315. Springer, 2006.

[111] Deep Ray, Praveen Chandrashekar, Ulrik Fjordholm, and Siddhartha Mishra. Entropy stable scheme on two-dimensional unstructured grids for euler equations. *Communications in Computational Physics*, 19(5):1111–1140, 2016.

[112] Franz Rellich. Ein satz über mittlere konvergenz. *Nachrichten von der Gesellschaft der Wissenschaften zu Göttingen, Mathematisch-Physikalische Klasse*, 1930:30–35, 1930.

[113] Tyrell Rockafellar. *Convex analysis*. Princeton University Press, 2015.

[114] Leonid Rudin, Stanley Osher, and Emad Fatemi. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4):259–268, 1992.

[115] Chi-Wang Shu. TVB uniformly high-order schemes for conservation laws. *Mathematics of Computation*, 49(179):105–121, 1987.

[116] Chi-Wang Shu. Discontinuous Galerkin methods: general approach and stability. *Numerical solutions of partial differential equations*, 201, 2009.

[117] Chi-Wang Shu. High order weighted essentially nonoscillatory schemes for convection dominated problems. *SIAM review*, 51(1):82–126, 2009.

[118] Chi-Wang Shu. Essentially non-oscillatory and weighted essentially non-oscillatory schemes. *Acta Numerica*, 29:701–762, 2020.

[119] Chi-Wang Shu and Stanley Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, ii. *Journal of Computational Physics*, 83(1):32–78, 1989.

[120] Emil Sidky and Xiaochuan Pan. Image reconstruction in circular cone-beam computed tomography by constrained, total-variation minimization. *Physics in Medicine & Biology*, 53(17):4777, 2008.

[121] Stefan Spekreijse. Multigrid solution of monotone second-order discretizations of hyperbolic conservation laws. *Mathematics of Computation*, 49(179):135–155, 1987.

[122] Peter K Sweby. High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM journal on numerical analysis*, 21(5):995–1011, 1984.

[123] Eitan Tadmor. Convenient total variation diminishing conditions for nonlinear difference schemes. *SIAM journal on numerical analysis*, 25(5):1002–1014, 1988.

[124] Eitan Tadmor. Chapter 18 - entropy stable schemes. In Rémi Abgrall and Chi-Wang Shu, editors, *Handbook of Numerical Methods for Hyperbolic Problems*, volume 17 of *Handbook of Numerical Analysis*, pages 467–493. Elsevier, 2016.

[125] Kunio Tanabe. Projection method for solving a singular system of linear equations and its applications. *Numerische Mathematik*, 17:203–214, 1971.

[126] Zhen Tian, Xun Jia, Kehong Yuan, Tinsu Pan, and Steve B Jiang. Low-dose ct reconstruction via edge-preserving total variation regularization. *Physics in Medicine & Biology*, 56(18):5949, 2011.

[127] Vladimir Titarev and Eleuterio Toro. WENO schemes based on upwind and centred TVD fluxes. *Computers & Fluids*, 34(6):705–720, 2005.

[128] Eleuterio Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer Science & Business Media, 2013.

[129] P. Tseng. Applications of a splitting algorithm to decomposition in convex programming and variational inequalities. *SIAM Journal on Control and Optimization*, 29(1):119–138, 1991.

[130] Bram Van Leer. Towards the ultimate conservative difference scheme. iv. a new approach to numerical convection. *Journal of computational physics*, 23(3):276–299, 1977.

[131] Bram Van Leer. Towards the ultimate conservative difference scheme. v. A second-order sequel to Godunov's method. *Journal of computational Physics*, 32(1):101–136, 1979.

[132] Aizik Volpert. The spaces BV and quasilinear equations, USSR Math. *Sb*, 2:225–267, 1967.

[133] Gerd Wachsmuth, Marc Herrmann, Roland Herzog, Stephan Schmidt, and José Vidal-Núñez. Discrete total variation with finite elements and applications to imaging. *Journal of Mathematical Imaging and Vision*, 61(4):411–413, 2019.

[134] Ge Wang and Ming Jiang. Ordered-subset simultaneous algebraic reconstruction techniques (os-sart). *Journal of X-ray Science and Technology*, 12(3):169–177, 2004.

[135] Jingyue Wang and Bradley Lucier. Error bounds for finite-difference methods for Rudin—Osher—Fatemi image smoothing. *SIAM Journal on Numerical Analysis*, 49(1/2):845–868, 2011.

[136] Michael Yang and Zhi-Jian Wang. A parameter-free generalized moment limiter for high-order methods on unstructured grids. In *47th AIAA Aerospace Sciences Meeting Including The New Horizons Forum and Aerospace Exposition*, page 605, 2009.

[137] Jian Yu and Jan S Hesthaven. A study of several artificial viscosity models within the Discontinuous Galerkin framework. *Communications in Computational Physics*, 27(5):1309–1343, 2020.

[138] Zhengshan Yu, Xingya Wen, and Yan Yang. Reconstruction of sparse-view x-ray computed tomography based on adaptive total variation minimization. *Micromachines*, 14(12):2245, 2023.

[139] Jun Zhu, Xinghui Zhong, Chi-Wang Shu, and Jianxian Qiu. Runge–Kutta Discontinuous Galerkin method using a new type of WENO limiters on unstructured meshes. *Journal of Computational Physics*, 248:200–220, 2013.

Appendix

Appendix A

An example of matrices $L,\ M$.

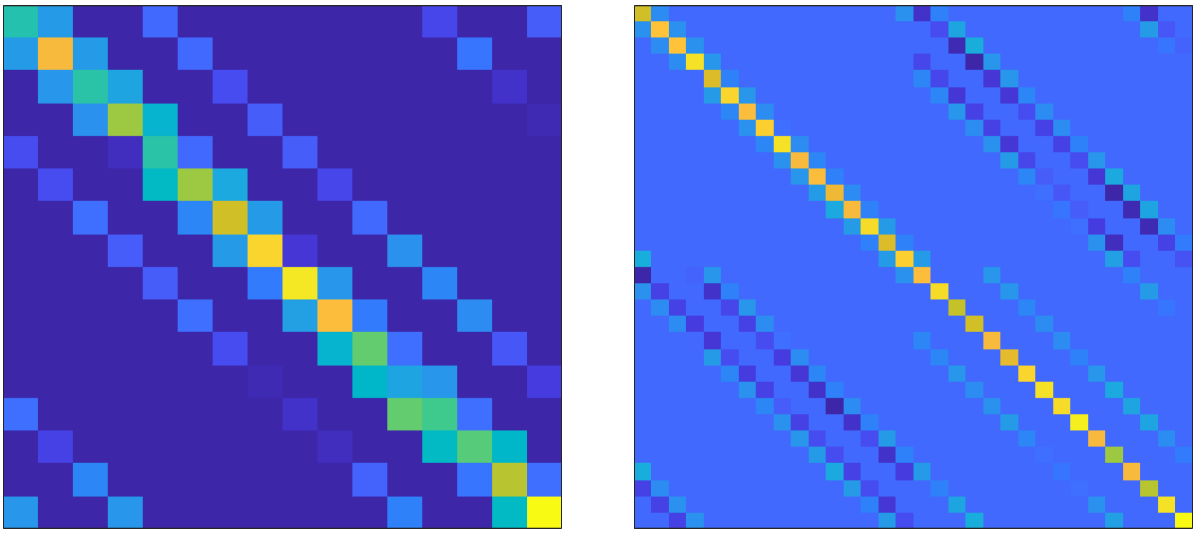


Figure A.1: Sparse structure of the matrices L (left) and M (right) with randomly generated coefficients satisfying conditions of Lemma 3.4.3., on $N = 4$ mesh. Matrix L has 5 main diagonals, 2 diagonals that account for periodic boundary conditions and 2 more entries in the first and in the last row. Matrix M is divided into 4 matrices, each of them has has 3-4 main diagonals, and 2 diagonals that account for periodic boundary conditions. The structure of each of the matrices is given below.

$$L = \begin{bmatrix} 1-R_1^n & A_{1,1}^n & \dots & & C_{1,1}^n & \dots & D_{1,N}^n & \dots & B_{N,N} \\ B_{1,1} & 1-R_2^n & A_{2,1}^n & & \dots & C_{2,1}^n & \dots & D_{2,N}^n & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \dots & D_{N-1,N}^n & \dots & C_{N-1,N}^n & \dots & B_{N-2,N}^n & 1-R_{S-1}^n & A_{N-1,N}^n \\ A_{N,N} & \dots & \dots & D_{N,N}^n & \dots & C_{N,N}^n & \dots & B_{N-1,N}^n & 1-R_S^n \end{bmatrix}$$

$$\begin{aligned} R_1^n &= A_{1,1}^n + B_{N,1}^n + C_{i,j}^n + D_{1,N}^n, \\ R_2^n &= A_{2,1}^n + B_{1,1}^n + C_{2,1}^n + D_{2,N}^n, \dots \\ &\vdots \\ R_{S-1}^n &= A_{N-1,N}^n + B_{N-1,N}^n + C_{N-1,N}^n + D_{N-1,N-1}^n, \\ R_S^n &= A_{N,N}^n + B_{N-1,N}^n + C_{N,N}^n + D_{N,N-1}^n. \end{aligned}$$

$$M = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix}$$

$$M_{11} = \begin{bmatrix} 1 - A^n_{1,1} - B^n_{1,1} & A^n_{2,1} & \dots & \dots & B_{N,N} \\ B^n_{1,1} & 1 - A^n_{2,1} - B^n_{2,1} & A^n_{3,1} & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \vdots \\ A_{1,1} & \dots & & \dots B^n_{N-1,N} & 1 - A^n_{N,N} - B^n_{N,N} \end{bmatrix}$$

$$M_{12} = \begin{bmatrix} -C^n_{1,1} & C^n_{2,1} & \dots & D^n_{1,N} & -D^n_{2,N} & \dots & \dots \\ \vdots & -C^n_{2,1} & C^n_{3,1} & \dots & D^n_{2,N} & -D^n_{3,N} & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \dots & D^n_{N,N} & -D^n_{1,N} & \dots & -C^n_{N-1,N} & C_{N-1,N} \\ C^n_{1,N} & \dots & \dots & D^n_{N,N} & -D^n_{1,N} & \dots & -C^n_{N,N} \end{bmatrix}$$

$$M_{21} = \begin{bmatrix} -A^n_{1,1} & \dots & -B^n_{N,2} & A^n_{1,2} & \dots & \dots & \dots & B^n_{1,N} \\ B^n_{1,1} & -A^n_{2,1} & \dots & -B^n_{1,2} & A^n_{2,2} & \dots & \dots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & -B_{N-1,N} & \dots & A^n_{N,N} & \dots & B^n_{N-2,N} & -A^n_{N-1,N} & \vdots \\ \dots & \dots & -B_{N,N} & \dots & A^n_{N,N} & \dots & B^n_{N-1,N} & -A^n_{N,N} \end{bmatrix}$$

$$M_{22} = \begin{bmatrix} 1 - C^n_{1,1} - D^n_{1,1} & \cdots & C^n_{1,2} & \cdots & D^n_{1,N} & \cdots & \cdots \\ \vdots & 1 - C^n_{2,1} - D^n_{2,1} & \cdots & C^n_{2,2} & \cdots & D^n_{2,N} & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & C^n_{N-1,N} & \cdots & D^n_{N-1,N-1} & \cdots & 1 - C^n_{N-1,N} - D^n_{N-1,N} & \vdots \\ \cdots & \cdots & C^n_{N,N} & \cdots & D^n_{N,N-1} & \cdots & 1 - C^n_{N,N} - D^n_{N,N} \end{bmatrix}$$

Appendix B

B.1 APGM algorithm (L. Condat's version)

The following MATLAB listing is L. Condat's code to compute TV of a 32×32 image using 2000 steps of APGM algorithm with a constant step size $\mu = 0.5$. The original code is available in the supplementary material to [40] at M107524 01.zip.

```
function main

Nbiter= 2000;
sigma = 0.99/3;
mu = 0.5;

S=32;
[n1,n2]=meshgrid(1:S,1:S);

Pattern = 3;
switch Pattern
        case 1, x=double((n1-S/2)>=1);
        case 2, x=double((n1-S/2)+(n2-S/2)>=1);
        case 3, x=double((n1-S/2)+(n2-S/2)>=1);
                idx=find((n1-S/2)+(n2-S/2)==1);
                x(idx)= 0.5;
 end

        opDx = cat(3,[diff(x,1,1);zeros(1,size(x,2))],...
             [diff(x,1,2) zeros(size(x,1),1)]);
        opDadj = @(u) -[u(1,:,1);diff(u(:,:,1),1,1)]-...
             [u(:,1,2) diff(u(:,:,2),1,2)];
        prox_mu_sigma_g = @(t) t-bsxfun(@rdivide, t,...
            max(sqrt(sum(t.^2,3))/(mu*sigma),1));

        u = zeros([size(x) 2]);
        v = zeros([size(x) 2 3]);
        tmp = opDx;
        v(:,:,1,1) = tmp(:,:,1);
        v(:,:,2,2) = tmp(:,:,2);
        fprintf('0 %f\n',sum(sum(sum(sqrt(sum(v.^2,3))))));
        for iter = 1:Nbiter
```

```
                v = ...
                prox_mu_sigma_g(v-sigma*opL(-opDx+opLadj(v)-mu*u));
                u = u-(-opDx+opLadj(v))/mu;
                if mod(iter,400)==0
                  %we display the primal and dual cost functions,
                        %which reach equal values at convergence
                          fprintf('%d %f %f\n',iter,...
                              sum(sum(sum(sqrt(sum(v.^2,3))))),...
                                  sum(u(:).*opDx(:)));
                end
        end
end

function t = opL(u)
        [height,width,d]=size(u);
        t=zeros(height,width,2,3);
        t(:,:,1,1)=u(:,:,1);
        t(1:end-1,2:end,2,1)=(u(2:end,1:end-1,2)+...
        u(1:end-1,1:end-1,2)+...
        u(2:end,2:end,2)+u(1:end-1,2:end,2))/4;
        t(1:end-1,1,2,1)=(u(1:end-1,1,2)+u(2:end,1,2))/4;
        t(:,:,2,2)=u(:,:,2);
        t(2:end,1:end-1,1,2)=...
    (u(2:end,1:end-1,1)+u(1:end-1,1:end-1,1)+...
        u(2:end,2:end,1)+u(1:end-1,2:end,1))/4;
        t(1,1:end-1,1,2)=(u(1,1:end-1,1)+u(1,2:end,1))/4;
        t(2:end,:,1,3) = (u(2:end,:,1)+...
        u(1:end-1,:,1))/2;
        t(1,:,1,3) = u(1,:,1)/2;
        t(:,2:end,2,3) = (u(:,2:end,2)+u(:,1:end-1,2))/2;
        t(:,1,2,3) = u(:,1,2)/2;
end

function u = opLadj(t)
        [height,width,d,c]=size(t);
        u=zeros(height,width,2);
        u(1:end-1,2:end,1)=t(1:end-1,2:end,1,1)+...
        (t(1:end-1,2:end,1,2)+...
        t(1:end-1,1:end-1,1,2)+t(2:end,2:end,1,2)+...
        t(2:end,1:end-1,1,2))/4+(t(1:end-1,2:end,1,3)+...
        t(2:end,2:end,1,3))/2;
        u(1:end-1,1,1)=t(1:end-1,1,1,1)+(t(1:end-1,1,1,2)+...
        t(2:end,1,1,2))/4+(t(1:end-1,1,1,3)+...
        t(2:end,1,1,3))/2;
        u(2:end,1:end-1,2)=t(2:end,1:end-1,2,2)+...
        (t(2:end,1:end-1,2,1)+...
        t(1:end-1,1:end-1,2,1)+t(2:end,2:end,2,1)+...
        t(1:end-1,2:end,2,1))/4+(t(2:end,1:end-1,2,3)+...
        t(2:end,2:end,2,3))/2;
        u(1,1:end-1,2)=t(1,1:end-1,2,2)+(t(1,1:end-1,2,1)+...
```

```
        t(1,2:end,2,1))/4+(t(1,1:end-1,2,3)+...
        t(1,2:end,2,3))/2;
end
```

B.2 Modified APGM (Algorithm 1)

Here we provide the MATLAB listing for Algorithm 1, to compute TV of 32×32 images used in the Appendix B.1. For ease of comparison with the original code of L. Condat we provide here a simplified implementation of Algorithm 1, without the stopping criterion. We also omit the repetition of `opL` and `opLadj` functions declaration here for brevity. The number of steps `Nbiter = 800` in this case, because the proposed algorithm achieves the same accuracy of the solution with 2.5 times less iterations. The MATLAB codes used for other numerical experiments presented in this work can be found at MATLAB Codes.

```
function main

Nbiter= 800;
sigma = 0.99/3;
mu0 = 1;
mu = mu0;
theta = 0.96;

S=32;
[n1,n2]=meshgrid(1:S,1:S);

Pattern = 3;
switch Pattern
        case 1, x=double((n1-S/2)>=1);
        case 2, x=double((n1-S/2)+(n2-S/2)>=1);
        case 3, x=double((n1-S/2)+(n2-S/2)>=1);
                idx=find((n1-S/2)+(n2-S/2)==1);
                x(idx)= 0.5;
 end

        opDx = cat(3,[diff(x,1,1);zeros(1,size(x,2))],...
             [diff(x,1,2) zeros(size(x,1),1)]);
        opDadj = @(u) -[u(1,:,1);diff(u(:,:,1),1,1)]-...
             [u(:,1,2) diff(u(:,:,2),1,2)];
        prox_mu_sigma_g = @(t,mu) t-bsxfun(@rdivide, t,...
            max(sqrt(sum(t.^2,3))/(mu*sigma),1));

        u = zeros([size(x) 2]);
        v = zeros([size(x) 2 3]);
        tmp = opDx;
        v(:,:,1,1) = tmp(:,:,1);
        v(:,:,2,2) = tmp(:,:,2);
        fprintf('0 %f\n',sum(sum(sum(sqrt(sum(v.^2,3))))));
        for iter = 1:Nbiter
```

```
                    v = ...
                    prox_mu_sigma_g(v-sigma*opL(-opDx+opLadj(v)-mu*u),mu);
                    u = u-(-opDx+opLadj(v))/mu;
                    mu = theta*mu;
                    if mod(iter,100)==0
                      mu = mu0;
                      %we display the primal and dual cost functions,
                            %which reach equal values at convergence
                              fprintf('%d %f %f\n',iter,...
                                    sum(sum(sum(sqrt(sum(v.^2,3))))),...
                                          sum(u(:).*opDx(:)));
                    end
            end
end
```