# A PRIMAL-DUAL ACTIVE-SET METHOD FOR NON-NEGATIVITY CONSTRAINED TOTAL VARIATION DEBLURRING PROBLEMS

**DILIP KRISHNAN**

*B. A. Sc. (Comp. Engg.)*

A THESIS SUBMITTED FOR THE DEGREE OF

MASTER OF SCIENCE

DEPARTMENT OF MATHEMATICS

NATIONAL UNIVERSITY OF SINGAPORE

2007

# Acknowledgements

The research work contained in this thesis would not be possible without the rigorous, methodical and enthusiastic guidance of my supervisor, Dr. Andy Yip. Andy has opened my eyes to a new level of sophistication in mathematical thinking and relentless questioning. I am grateful to him for this. I would like to thank my co-supervisor, Dr. Lin Ping, for his support and guidance over the last two years. Thanks are also due to Dr. Sun Defeng for useful discussions on semi-smooth Newton's methods. Last but not least, I would like to thank my wife, Meghana, my parents, and brother for their love and support through the research and thesis phases.

# Contents

# Summary

This thesis studies image deblurring problems using a total variation based model, with a non-negativity constraint. The addition of the non-negativity constraint improves the quality of the solutions but makes the process of solution a difficult one. The contribution of our work is a fast and robust numerical algorithm to solve the non-negatively constrained problem. To overcome the non-differentiability of the total variation norm, we formulate the constrained deblurring problem as a primal-dual program which is a variant of the formulation proposed by Chan, Golub and Mulet [1] (CGM) for unconstrained problems. Here, dual refers to a combination of the Lagrangian and Fenchel duals. To solve the constrained primal-dual program, we use a semi-smooth Newton's method. We exploit the relationship, established in [2], between the semi-smooth Newton's method and the Primal-Dual Active Set (PDAS) method to achieve considerable simplification of the computations. The main advantages of our proposed scheme are: no parameters need significant adjustment, a standard inverse preconditioner works very well, quadratic rate of local convergence (theoretical and numerical), numerical evidence of global convergence, and high accuracy of solving the KKT system. The scheme shows robustness of performance over a wide range of parameters. A comprehensive set of numerical comparisons are provided against other methods to solve the same problem which show the speed and accuracy advantages of our scheme. The Matlab and C (Mex) code for all the experiments conducted in this thesis may be downloaded from `http://www.math.nus.edu.sg/~mhyip/nncgm/`.

# Chapter 1

# Introduction

## 1.1 Image Deblurring and Denoising

In this thesis, we study models to solve image deblurring and denoising problems. During the image acquisition process, images can suffer from various types of degradation. Two of the most common problems are that of noise and blurring. Noise is introduced because of the behaviour of the camera capture circuitry and exposure conditions. Blurring may be introduced due to a combination of physical phenomena and camera processes. For example, during the acquisition of satellite (atmospheric) images, the effect of the atmosphere acts as a blurring filter to blur the captured images of heavenly bodies.

Image blur is usually modelled as a convolution of the image data with a blurring kernel. The kernel may vary depending on the type of blur. Two common blurs are Gaussian blur and out-of-focus blur. Image noise is usually modelled as additive Gaussian distributed noise or uniformly distributed noise. Denoising can be considered to be a special case of deblurring with identity blur.

Digital images are represented as two-dimensional arrays $f(x, y)$. where each integer coordinate $(x, y)$ represents a single *pixel*. At each pixel, an integer value which varies between 0 and 255, for images of bit depth 8, represents the intensity or *gray level* of the pixel. 0 represents the color black, 255 represents the color white. All gray levels lie in between these two exteme values. Image deblurring may be represented as $f = k * u + n$, where $f$ is the observed degraded image, $u$ is the original image, $k$ is the blurring function and $n$ is the additive noise.

Deblurring and denoising of images is important for scientific and aesthetic reasons. For example, police agencies require deblurring of images captured from security cameras; denoising of images helps significantly in their compression; deblurring of atmospheric images is useful for the accurate identification of the heavenly bodies. See [3], [4] for more examples and overview.

A number of different models have been proposed to solve deblurring and denoising problems. It is not our purpose here to study all the different models. We restrict our attention to a model based on the Total Variation, which has proven to be succesful in solving a number of different image processing problems, including deblurring and denoising.

## 1.2 Total Variation Minimization Problems

Total Variation (TV) minimization problems were first introduced into the context of image denoising in the seminal paper [5] by Rudin, Osher and Fatemi. They have proven to be successful in dealing with image denoising and deblurring problems [1,6–8], image inpainting problems [9], and image decomposition [10]. Recently, they have also been applied in various areas such as CT imaging [11,12] and confocal microscopy [13]. The main advantage of the TV formulation is the ability to preserve edges in the image. This is due to the piecewise smooth regularization property of the TV norm.

A discrete version of the unconstrained TV deblurring problem proposed by Rudin *et al.* in [5] is given by

$$\min_u \frac{1}{2}\|Ku - f\|^2 + \beta\|u\|_{TV}, \tag{1.1}$$

where $\|\cdot\|$ is the $l^2$ norm, $f$ is the observed (blurred and noisy) data, $K$ is the blurring operator corresponding to a *point spread function* (PSF), $u$ is the unknown data to be recovered, and $\|\cdot\|_{TV}$ is the discrete TV regularization term. We assume that the $m \times n$ images $u = (u_{i,j})$ and $f = (f_{i,j})$ have been rearranged into a vector form using the lexicographical ordering. Thus, $K$ is an $mn \times mn$ matrix. The discrete TV norm is defined as

$$\|u\|_{TV} := \sum_{i=1}^{m-1}\sum_{j=1}^{n-1} |(\nabla u)_{ij}|, \quad \text{where} \quad (\nabla u)_{ij} = \begin{bmatrix} u_{i+1,j} - u_{i,j} \\ u_{i,j+1} - u_{i,j} \end{bmatrix}. \tag{1.2}$$

Here, $i$ and $j$ refer to the pixel indices in the image and $|\cdot|$ is the Euclidean norm for $\mathbb{R}^2$. Regularization is necessary due to the presence of noise, see [14]. Without regularization, noise amplification would be so severe that the resulting output data is useless, especially when $K$ is very ill-conditioned. Even when $K$ is well-conditioned, regularization is still needed to remove noise. The regularization parameter $\beta$ needs to be selected as a tradeoff between oversmoothing and noise amplification. When $K = I$, the deblurring problem becomes a pure denoising problem. When $K$ is derived from a non-trivial PSF (i.e. apart from the Dirac Delta function), the problem is harder to solve since the pixels of $f$ are coupled together to a greater degree. When $K$ is unknown, the problem becomes a blind deblurring problem [6, 15]. In this paper, we assume that the PSF, and hence $K$, is known.

One of the major reasons for ongoing research into TV deblurring problems is that the non-differentiability of the TV norm makes it a difficult task to find a fast numerical method. The (formal) first order derivative of the TV norm involves the term $\frac{\nabla u}{|\nabla u|}$ which is degenerate when $|\nabla u| = 0$. This could happen in flat areas of the image. Methods that can effectively deal with such singularities are still actively sought.

A number of numerical methods have been proposed for unconstrained TV denoising and/or deblurring models. These include partial differential equation based methods such as explicit [5], semi-implicit [16] or operator splitting schemes [17] and fixed point iterations [7]. Optimization oriented techniques include Newton-like methods [1], [18], [8], multilevel [19], second order cone programming [20] and interior-point methods [21]. Recently, graph based approaches have also been studied [22]. It is also possible to apply Additive Operator Splitting (AOS) based schemes such as those proposed originally in [23] to solve in a fast manner, the Euler-Lagrange equation corresponding to the primal problem.

Carter [24] presents a dual formulation of the TV denoising problem and studies some primal-dual interior-point and primal-dual relaxation methods. Chambolle [25] presents a semi-implicit scheme and Ng *et al.* [26] present a semi-smooth Newton's method for solving the same dual problem. These algorithms have the advantage of not requiring an extra regularization of the TV norm. Being faithful to the original TV norm without applying any regularization, these methods often require many iterations to converge to a moderate level of accuracy for the underlying optimization problem is not strictly convex.

Hintermüller and Kunisch [27] have derived a dual version of an

anisotropic TV deblurring problem. In the anisotropic formulation, the TV norm $\|u\|_{TV}$ in Eq. (1.2) is replaced with $\sum\limits_{i,j} |(\nabla u)_{ij}|_1$ where $|\cdot|_1$ is the $l^1$ norm for $\mathbb{R}^2$. This makes the dual problem a quadratic one with linear bilateral constraints. In contrast, the isotropic formulation is based on the $l^2$ norm and has the advantage of being rotation invariant. However, the dual problem corresponding to the isotropic TV norm has quadratic constraints which are harder to deal with. Hintermüller and Kunisch have solved the anisotropic formulation using a primal-dual active-set method, but the algorithm requires several additional regularization terms.

Chan, Golub and Mulet present a primal-dual numerical method [1]. This algorithm (which we henceforth call the CGM algorithm) simultaneously solves both the primal and (Fenchel) dual problems. In this work, we propose a variant of their algorithm to handle the non-negativity constraint.

It should be noted that many of the aforementioned numerical methods are specific to denoising and cannot be readily extended to a general deblurring problem. Fewer papers focus on TV deblurring problems. Still fewer focus on constrained TV deblurring problems. But our method works for the more difficult non-negativity constrained isotropic TV deblurring problem and is faster than other existing methods for solving the same problem.

Image values which represent physical quantities such as photon count or energies are often non-negative. For example, in applications such as gamma ray spectral analysis [28], astronomical imaging and spectroscopy [29], the physical characteristics of the problem require the recovered data to be non-negative. An intuitive approach to ensuring non-negativity is to solve the unconstrained problem first, followed by setting the negative components of the resulting output to zero. However, this approach may result in the presence of spurious ripples in the reconstructed image. Chopping off the negative values may also introduce patches of black color which could be visually unpleasant. In biomedical imaging, the authors in [12, 13] also stressed the importance of non-negativity in their TV based models. But they obtain non-negative results by tuning a regularization parameter similar to the $\beta$ in Eq. (1.1). This may cause the results to be under- or over-regularized. Moreover, there is no guarantee that such a choice of the parameter exists. Therefore, a non-negativity constraint on the deblurring problem is a natural requirement.

The non-negatively constrained TV deblurring problem is given by

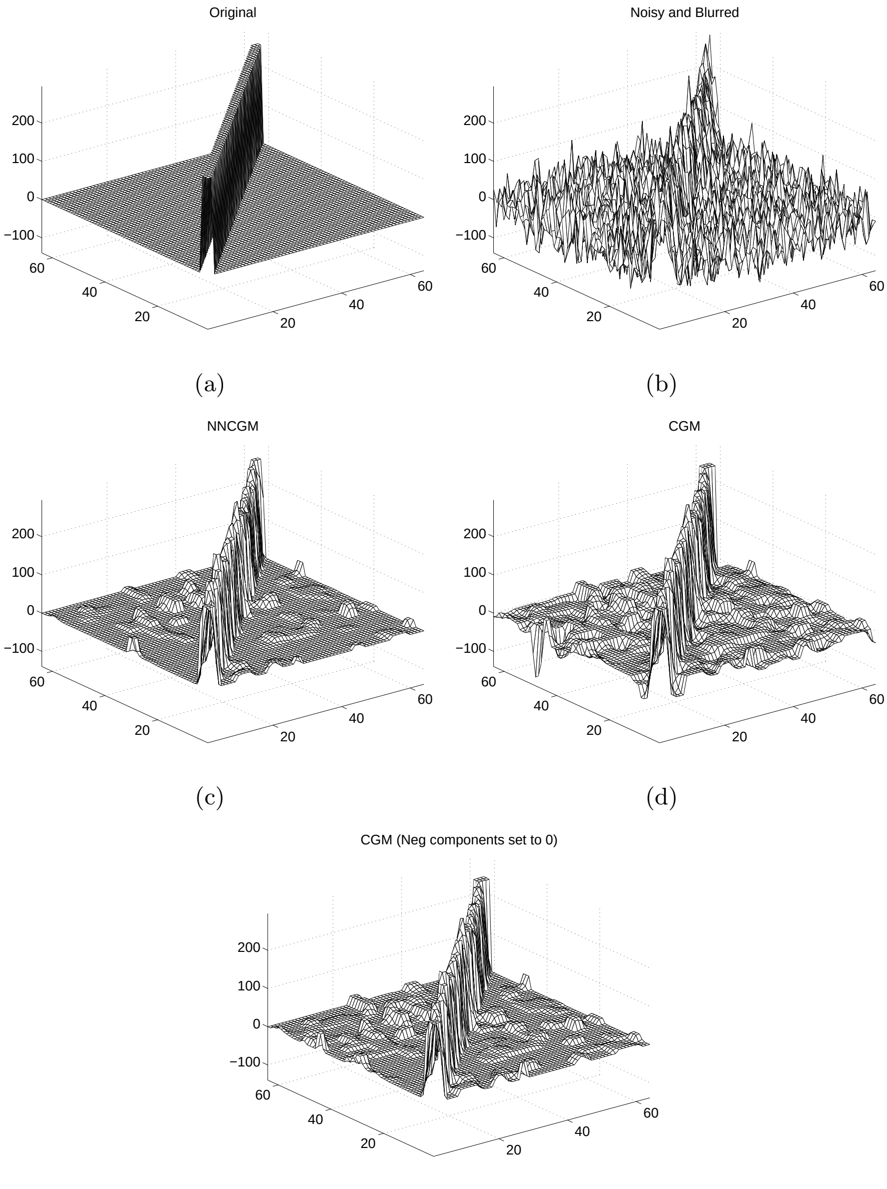
$$\min_{u, u \geq 0} \frac{1}{2} \|Ku - f\|^2 + \beta \|u\|_{TV}. \tag{1.3}$$

This problem is convex for all $K$ and is strictly convex when $K$ is of full rank. We shall assume that $K$ is of full rank so that the problem has a unique solution. For denoising problems, imposing the non-negativity constraint becomes unnecessary for it is equivalent to solving the unconstrained problem followed by setting the negative components to zero. For deblurring problems, even if the observed data are all positive, the deblurred result may contain negative values if non-negativity is not enforced.

Schafer *et al.* [28], have studied the non-negativity constraint on gamma-ray spectral data and synthetic data. They have demonstrated that such a constraint helps not only in the interpretability of the results, but also helps in the reconstruction of high-frequency information beyond the Nyquist frequency (in case of bandlimited signals). Reconstruction of the high-frequency information is an important requirement for image processing since the details of the image are usually the edges.

Fig. 1.1 gives another example. The reconstructions 1.1(d) and 1.1(e) based on the unconstrained primal-dual method presented in [1] show a larger number of spurious spikes. It is also clear that the intuitive method of solving the unconstrained problem and setting the negative components to zero still causes a number of spurious ripples. In contrast, the constrained solution 1.1(c) has much fewer spurious ripples in the recovered background. The unconstrained results have larger $l^2$ reconstruction error compared to the constrained reconstruction.

Some examples showing the increased reconstruction quality of imposing non-negativity can be found in [30]. Studies on other non-negativity constrained deblurring problems such as Poisson noise model, linear regularization and entropy-type penalty can be found in [31, 32].



(a)

(b)

(c)

(d)

(e)

Figure 1.1: Comparison of constrained and unconstrained deblurring. (a) Original synthetic data. (b) Blurred and noisy data with negative components. (c) Non-negatively constrained NNCGM result; $l_2$ error = 361.43. (d) Unconstrained CGM result; $l_2$ error = 462.74. (e) Same as (d) but with negative components set to 0; $l_2$ error = 429.32.

Very few numerical approaches have been studied for non-negatively constrained total variation deblurring problems. In [30, 33], a projected Newton's method based on the algorithm of [34] is presented to solve the non-negatively constrained problem. We study the performance of this algorithm in this work. Fu *et al.* [21] present an algorithm based on interior-point methods, along with very effective preconditioners. The total number of outer iterations is small. However, the inner iterations, corresponding to Newton steps in the interior-point method, take long to converge. Moreover, Fu *et al.* study the anisotropic TV formulation, which can be reduced to a linear programming problem, whereas the isotropic formulation is more difficult to solve. We have studied the interior-point method for the isotropic TV norm and observed significant slow down in the inner iterations as the outer iterations proceed. This is because of the increased ill-conditioning of the linear systems that are to be solved in the inner iterations. In contrast, the primal-dual method presented in this work does not suffer from this drawback – the number of inner *conjugate gradient* (CG) iterations [35] shows no significant increase as the system approaches convergence.

The rest of the thesis is organized as follows: Chapter 2 presents our proposed primal-dual method (which we call NNCGM) for non-negatively constrained TV deblurring, along with two other algorithms to which we compare the performance of the NNCGM algorithm. These two algorithms are a dual-only Alternating Minimization method and a primal-only Projected Newton's method. Chapter 3 provides numerical results to compare NNCGM with these two methods and also shows the robustness of NNCGM. Chapter 4 gives conclusions. Appendix A gives the technical details of the derivation of the primal-dual formulation and the NNCGM algorithm. Appendix B gives the default parameters that were used for all the numerical results given in this thesis.

A paper [36] based on the results presented in this thesis has recently (August 2007) been accepted for publication in the IEEE Transactions on Image Processing.

# Chapter 2

# The Non-Negatively Constrained Primal-Dual Program

## 2.1 Dual and Primal-Dual Approaches

Solving the primal TV deblurring problem, whether unconstrained or constrained, poses numerical difficulties due to the non-differentiability of the TV norm. This difficulty is usually overcome by the addition of a perturbation $\epsilon$. That is, to replace $|\nabla u|$ with $|\nabla u|_\epsilon = \sqrt{|\nabla u|^2 + \epsilon}$ which is a differentiable function. The trade-off in choosing this smoothing parameter $\epsilon$ is the reconstruction error versus the speed of convergence. The smaller the perturbation term, the more accurate is the final reconstruction. However, convergence takes longer since the corresponding objective function to be optimized becomes increasingly closer to the original non-differentiable objective function. See [37] for more details on convergence in relation to the value of $\epsilon$.

Owing to the above numerical difficulties, some researchers have studied a dual approach to the TV deblurring problem. Carter [24] and Chambolle [25] present a dual problem based on the Fenchel dual formulation for the TV denoising problem. See [38] for details on the Fenchel dual. Chambolle's scheme is based on the minimization problem

$$\min_{p, |p_{i,j}| \leq 1} \|f + \beta \mathrm{div} p\|^2. \tag{2.1}$$

Here,

$$p_{i,j} = \begin{bmatrix} p^x_{i,j} \\ p^y_{i,j} \end{bmatrix} \tag{2.2}$$

is the dual variable at each pixel location with homogeneous Dirichlet boundary conditions $p^x_{0,j} = p^x_{m,j} = 0$ for all $j$ and $p^y_{i,0} = p^y_{i,n} = 0$ for all $i$. The vector $p$ is a concatenation of all $p_{i,j}$. The discrete divergence operator div is defined such that the vector div$p$ is given by

$$(\mathrm{div} p)_{i,j} = p^x_{i,j} - p^x_{i-1,j} + p^y_{i,j} - p^y_{i,j-1}. \tag{2.3}$$

It can be shown (see [25]) that $(-\mathrm{div})^T = \nabla$ defined in (1.2). The constraints $|p_{i,j}| \leq 1$ are quadratic after squaring both sides. The update is given by

$$p^{n+1}_{i,j} = \frac{p^n_{i,j} + \tau\beta(\nabla(\beta \mathrm{div} p + g))_{i,j}}{1 + \tau\beta|(\nabla(\beta \mathrm{div} p + g))_{i,j}|}, \qquad \forall\, i, j,$$

where $\tau$ is the step size (which, as shown in [25], needs to be less than 1/8). Once the optimal solution, denoted by $p^*$, is obtained, the denoised image $u^*$ can be reconstructed by $u^* = \beta \mathrm{div} p^* + f$. An interesting aspect of the algorithm is that even without the $\epsilon$-perturbation of the TV norm, the objective function becomes a quadratic function which is infinitely differentiable. But the dual variable $p$ becomes constrained. Unfortunately, being based on a steepest descent technique, the algorithm slows down towards convergence and requires a large number of iterations for even a moderate accuracy.

Hintermüller and Kunisch [27] have also used the Fenchel dual approach to formulate a constrained quadratic dual problem and to derive a very effective method. They consider the case of anisotropic TV norm so that the dual variable is bilaterally constrained, i.e. $-1 \leq p_{i,j} \leq 1$, whereas the constraints in Eq. (2.1) are quadratic. The smooth (quadratic) nature of the dual problem makes it much more amenable to solution by a Newton-like method. To deal with the bilateral constraints on $p$, the authors propose to use the Primal-Dual Active-Set (PDAS) algorithm. Consider the general quadratic problem

$$\min_{y \leq \psi} \frac{1}{2}\langle y, Ay \rangle - \langle f, y \rangle$$

whose KKT system [39] is given by

$$\begin{aligned} Ay + \lambda &= f, \\ C(y, \lambda) &= 0, \end{aligned}$$

where $C(y, \lambda) = \lambda - \max(0, \lambda + c(y - \psi))$ for an arbitrary positive constant $c$, and $\lambda$ is the Lagrange multiplier. The max operation is understood to be component-wise. Then the PDAS algorithm is given by

1. Initialize $y^0$, $\lambda^0$. Set $k = 0$.
2. Set $\mathcal{I}^k = \{i : \lambda_i^k + c(y^k - \psi)_i \leq 0\}$ and $\mathcal{A}^k = \{i : \lambda_i^k + c(y^k - \psi)_i > 0\}$.
3. Solve

$$\begin{aligned} Ay^{k+1} + \lambda^{k+1} &= f, \\ y^{k+1} &= \psi \qquad \text{on } \mathcal{A}^k, \\ \lambda^{k+1} &= 0 \qquad \text{on } \mathcal{I}^k. \end{aligned}$$

4. Stop, or set $k = k + 1$ and return to Step 2).

In their work in [2], the authors show that the PDAS algorithm is equivalent to a semi-smooth Newton's method for a class of optimization problems that includes the dual anisotropic TV deblurring problem. Local superlinear convergence results are derived. Conditional global convergence results based on the properties of the matrix $K$ are also derived. However, their formulation only works for the anisotropic TV norm and the dual problem requires several extra regularization terms to achieve a numerical solution.

Chan *et al.* [1] present a primal-dual numerical method which has a much better global convergence behaviour than a primal-only method for the unconstrained problem. As the name suggests, this algorithm simultaneously solves both the primal and dual problems. The algorithm is derived as a Newton step for the following equations

$$\begin{aligned} p|\nabla u|_\epsilon - \nabla u &= 0, \\ -\beta \mathrm{div} p - K^T f + Au &= 0, \end{aligned}$$

where $A := K^T K + \alpha I$. At each Newton step, both the primal variable $u$ and the dual variable $p$ are updated. The dual variable can be thought of as helping to overcome the singularity in term $\mathrm{div}\left(\frac{\nabla u}{|\nabla u|}\right)$. An advantage of this method is that a line search is required only for the dual variable $p$ to maintain the feasibility $|p_{i,j}| \leq 1$ whereas a line search for $u$

is unnecessary. Furthermore, while still requiring an $\epsilon$-regularization as above, it converges fast even when the perturbation $\epsilon$ is small. Our algorithm is inspired by the CGM method. But the CGM method does not handle the non-negativity constraint on $u$.

Note that dual-only methods for TV deblurring need an extra $l^2$ regularization which is a disadvantage for these methods. This is because the matrix $(K^TK)^{-1}$ is involved and one needs to replace it by $(K^TK+\alpha I)^{-1}$ to make it well-conditioned. In denoising problems, we have $K = I$ so that the ill-conditioning problem of $(K^TK)^{-1}$ in dual methods is absent. But in deblurring problems, some extra care needs to be taken. The modified TV deblurring problem is then given by

$$\min_{u,u\geq 0} \frac{1}{2}\|Ku-f\|^2 + \beta\|u\|_{TV} + \frac{\alpha}{2}\|u\|^2. \tag{2.4}$$

Primal-dual methods such as CGM and our NNCGM do not require the extra $l^2$ regularization term, i.e. $\alpha = 0$. This is because $K^TK$ instead of $(K^TK)^{-1}$ is involved in the primal-dual formulation. See Appendix A.

## 2.2 NNCGM Algorithm

As stated above, our algorithm is inspired by the CGM method. Hence we call it the NNCGM algorithm (Non-Negatively constrained Chan-Golub-Mulet). We derived the dual of the constrained problem (2.4) as follows:

$$\min_{p,|p_{i,j}|\leq 1} \ \min_{\lambda,\lambda\geq 0} \left\{ \frac{1}{2}\|B^{1/2}(K^Tf+\beta \text{div} p+\lambda)\|^2 \right\}, \tag{2.5}$$

where $B := (K^TK+\alpha I)^{-1}$. The dual variable $p$ has constraints on it arising from the Fenchel transform of the TV norm in the primal objective function Eq. (2.4). The variable $\lambda$ has a non-negativity constraint since it arises as a Lagrange multiplier for the non-negativity constraint on $u$. See Appendix A for the detailed derivation. We remark that the parameter $\alpha$ can be set to 0 in our method, see Chapter 2.1.

The primal-dual program associated with the problem (2.5) is given by:

$$\begin{aligned} p|\nabla u|_\epsilon - \nabla u &= 0, &(2.6)\\ -\beta \text{div} p - K^Tf - \lambda + Au &= 0, &(2.7)\\ \lambda - \max\{0, \lambda - cu\} &= 0, &(2.8)\end{aligned}$$

where $c$ is an arbitrary positive constant. We have identified the Lagrange multiplier for $\lambda \geq 0$ with the primal variable $u$. This leads to the presence of $u$ in the system. Note that we have transformed all inequality constraints and complementarity conditions on $u$ and $\lambda$ into the single equality constraint in Eq. (2.8).

The NNCGM algorithm is essentially a semi-smooth Newton's method for the system in Eq. (2.6)-(2.8). It has been shown by Hintermüller *et al.* [27] that the semi-smooth Newton's method is equivalent to the PDAS algorithm for a certain class of optimization problems. Although the equivalency does not hold in our problem, the two methods are still highly related. We exploit this relationship, and use some ideas of the PDAS algorithm to significantly simplify the computations involved in solving Eq. (2.6)-(2.8), see Appendix A. The full NNCGM algorithm is as follows:

1. Select parameters based on Table B.1.

2. Initialize $p^0, u^0, \lambda^0$. Set $k = 0$.

3. Set $\mathcal{I}^k = \{i : \lambda_i^k - cu_i^k \leq 0\}$ and $\mathcal{A}^k = \{i : \lambda_i^k - cu_i^k > 0\}$. In the rest of the algorithm below, these two quantities are represented as $\mathcal{I}$ and $\mathcal{A}$ respectively.

4. Compute $\delta u_{\mathcal{I}}^k$ by solving the linear system using PCG (cf. Eq. (A.21)):
$$\begin{aligned} D_{\mathcal{I}} \left[ -\beta \mathrm{div} \frac{1}{|\nabla u^k|_\epsilon} \left( I - \frac{p^k (\nabla u^k)^T + (\nabla u^k)(p^k)^T}{2|\nabla u^k|_\epsilon} \right) \nabla + A \right] D_{\mathcal{I}}^T \delta u_{\mathcal{I}}^k & \\ = g(p^k, u^k, \lambda^k). & \qquad (2.9) \end{aligned}$$
   The details on the preconditioner used are given in Section 2.3.

5. Compute $\delta p^k$ by (cf. Eq. (A.19)):
$$\begin{aligned} \delta p^k \quad = \quad & \frac{1}{|\nabla u^k|_\epsilon} \times \\ & \left[ \left( I - \frac{p^k (\nabla u^k)^T}{|\nabla u^k|_\epsilon} \right) \nabla (D_{\mathcal{I}}^T \delta u_{\mathcal{I}}^k - D_{\mathcal{A}}^T u_{\mathcal{A}}^k) - F_1(p^k, u^k, \lambda^k) \right]. \end{aligned}$$

6. Compute $\delta \lambda_{\mathcal{A}}^k$ by (cf. Eq. (A.18)):
$$\delta \lambda_{\mathcal{A}}^k = -\beta D_{\mathcal{A}} \mathrm{div} \delta p^k + A_{\mathcal{A}\mathcal{I}} \delta u_{\mathcal{I}}^k + D_{\mathcal{A}} F_2(p^k, u^k, \lambda^k) - A_{\mathcal{A}} u_{\mathcal{A}}^k.$$

7. Compute the step size $s$ by $s = \rho \sup_\gamma \{|p_{i,j}^k + \gamma \delta p_{i,j}^k| \leq 1 \;\; \forall \; i, j\}$.

8. Update

$$\begin{aligned} p^{k+1} &= p^k + s\delta p^k, \\ u_{\mathcal{I}}^{k+1} &= u_{\mathcal{I}}^k + \delta u_{\mathcal{I}}^k, \\ u_{\mathcal{A}}^{k+1} &= 0, \\ \lambda_{\mathcal{I}}^{k+1} &= 0, \\ \lambda_{\mathcal{A}}^{k+1} &= \lambda_{\mathcal{A}}^k + \delta\lambda_{\mathcal{A}}^k. \end{aligned}$$

9. Either stop if the desired KKT residual accuracy is reached, or set $k = k+1$ and go back to Step 3). The KKT residual is given by $\left(\|F_1\|^2 + \|F_2\|^2 + \|F_3\|^2\right)^{1/2}$ where $F_1, F_2, F_3$ are defined by the left hand side of Eq. (2.6)-(2.8).

At every iteration, the current iterates for $\lambda$ and $u$ are used to predict the active ($\mathcal{A}^k$) and inactive ($\mathcal{I}^k$) sets for the next iteration. This is the fundamental mechanism of the PDAS method as presented in [2]. A line search is only required in Step 7), for $p$. We found numerically that there was no need to have a line search in the $u$ and $\lambda$ variables. Occasional infeasibility (i.e. violation of non-negativity) of these variables during the iterations did not prevent convergence. The algorithm requires the specification of the following parameters:

1. $c$: $c$ is a positive value used to determine the active and inactive sets at every iteration, see Step 3) above. In our tests we found that the performance of the algorithm is independent of the value of $c$, as long as $c$ is a positive value. This is consistent with the results obtained in [2]. Hence using a fixed value of $c$ was sufficient for all our numerical tests.

2. $\rho$: Setting $\rho$ to 0.99 worked for all our numerical tests. This parameter is used only to make the step size a little conservative.

3. $\epsilon$: The perturbation constant is to be selected at a reasonably small value to achieve a trade-off between reconstruction error and time for convergence. We found that setting this to $10^{-2}$ worked for all cases. Reducing it further did not significantly reduce the reconstruction error. See Chapter 3 for results.

The regularization parameter $\beta$ decides the trade-off between the reconstruction error and noise amplification. It is a part of the deblurring model, rather than our algorithm. The value of $\beta$ must be selected carefully for any TV deblurring algorithm.

Our NNCGM algorithm was inspired by using the CGM algorithm [1] to handle the TV deblurring, and the PDAS algorithm [2] to handle the non-negativity constraint. The CGM algorithm was shown to be very fast in solving the unconstrained TV deblurring problem, and involved a minimal number of parameters. It also handles the inequality constraint on the dual variable $p$ by a simple line search. Furthermore, the numerical results in [1] show a locally quadratic rate of convergence. The PDAS algorithm handles unilateral constraints effectively. While Hintermüller and Kunisch [27] apply PDAS to handle the constraints $-1 \leq p \leq 1$, we apply it to handle the non-negativity constraints $u, \lambda \geq 0$. Under our formulation, the quadratic constraints $|p_{i,j}| \leq 1$ for all $i, j$ are implied Eq. (2.6). But we found it more convenient to maintain the feasibility of these quadratic constraints by a line search as in the CGM method. This can make sure that the linear system Eq. (2.9) to solve in each Newton step is positive definite. Since the NNCGM method is basically a semi-smooth Newton's method and the system of equations to solve in our formulation is strongly semi-smooth, it can therefore be expected that the NNCGM algorithm should exhibit a quadratic rate of local convergence. The numerical results of Chapter 3 show a locally quadratic rate of convergence.

## 2.3 Preconditioners

The most computationally intensive step of the NNCGM algorithm is Step 4) which involves solving the linear system in Eq. (2.9). Though significantly smaller than the original linear system (A.17) obtained by linearizing Eq. (2.6)-(2.8), it is still a large system. We therefore explored the use of preconditioners, and discovered that the standard ILU preconditioner [35] and the Factorized Banded Inverse Preconditioner (FBIP) [40] worked well to speed up the solution of the linear system. The FBIP preconditioner, in particular, worked extremely well. Using the FBIP preconditioner to solve the linear system requires essentially $O(N \log N)$ operations, where $N$ is the total number of pixels in the image. This is including the use of FFT's for computations involving the matrix $K$.

The original system Eq. (A.17) has different characteristics in each of its blocks. It is therefore harder to construct an effective preconditioner. In contrast, the reduced system Eq. (2.9) has a simpler structure so that standard preconditioners work well.

## 2.4 Comparative Algorithms

We compare the performance of the NNCGM with two other algorithms: a primal-only Projected Newton's (PN) algorithm, and a dual-only Alternating Minimization (AM) algorithm. To the best of our knowledge, the PN algorithm is the only algorithm proposed for solving non-negativity constrained isotropic TV deblurring problems that is designed for speed. The AM algorithm, derived by us, is a straightforward and natural way to reduce the problem into subproblems that are solvable by existing solvers. A common way used in application-oriented literature is to cast the TV minimization problem as a maximum *a posteriori* estimation problem and then apply the expectation maximization (EM) algorithm with multiplicative updates to ensure non-negativity [41]. The algorithm is usually straightforward to implement. However, it is well-known that the convergence of EM-type algorithms is slow. We experimented with a number of other algorithms as well, but the performance of those algorithms was quite poor. For example, we tried using a barrier method to maintain feasibility, but the method was very slow. The method given in Fu *et al.*'s paper, [21] uses linear programming. Since they use the anisotropic model for the TV, it is not a problem to maintain feasibility in their approach. We tried to adopt this approach for our isotropic formulation, but it is difficult to maintain the feasibility of the problem. Other interior-point methods require a feasible initial point, which is very difficult to obtain for this problem owing to the non-linearity.

The PN algorithm is based on that presented in [30, 33]. At each outer iteration, active and inactive sets are identified based on the primal variable $u$. Then a Newton step is taken for the inactive variables whereas a projected steepest descent is taken for the active ones. A line search ensures that the step size taken in the inactive variables is such that they do not violate the non-negativity constraint. A few parameters have to be modified to tune the line search. The method is quite slow, for only a few inactive variables are updated at each step. Active variables which are already at the boundary of the feasible set, cannot be updated. Theoretically, once all the active variables are identified, the convergence is quadratic. However, it takes many iterations to find all the active variables. In all of our experiments, a quadratic convergence has not been observed within the limit of 300 iterations. More importantly, the Newton iterations diverge for many initial data since the non-differentiability of the TV norm has not been dealt with properly [37].

A natural way to solve the dual problem (2.5) is by alternating minimization. The AM algorithm is based on the convexity of the dual problem. The problem is solved by the alternating solution of two subproblems: the $\lambda$ subproblem for fixed $p$ and the $p$ subproblem for a fixed $\lambda$. The $\lambda$ subproblem is given by

$$\min_{\lambda,\lambda\geq 0} \frac{1}{2}\|B^{1/2}(b+\lambda)\|^2, \tag{2.10}$$

where

$$b = K^T f + \beta \text{div} p^c,$$

and $p^c$ is the latest value of $p$ from the previous iteration. The $p$ subproblem is given by

$$\min_{p,|p_{i,j}|\leq 1} \frac{1}{2}\|B^{1/2}(g+\beta \text{div} p)\|^2, \tag{2.11}$$

where

$$g = K^T f + \lambda^c,$$

and $\lambda^c$ is the latest value of $\lambda$ from the previous iteration. The solution of the $\lambda$ subproblem (2.10) uses the PDAS algorithm presented in [2]. The solution of the $p$ subproblem (2.11) is based on an extension of Chambolle's steepest descent technique presented in [25], modified for deblurring problems. The Euler-Lagrange equation corresponding to the problem is

$$-\beta \left[\nabla(B(\beta \text{div} p + g))\right]_{i,j} + \mu_{i,j} p_{i,j} = 0, \qquad \forall\ i,j$$

where, as before, $i,j$ refer to individual pixels in the image. Note that, as discussed earlier, $\alpha > 0$ in case of AM since this is a dual-only method. The above equation is used to derive the following steepest-descent algorithm to solve the $p$ subproblem:

$$p_{i,j}^{n+1} = \frac{p_{i,j}^n + \tau\beta \left[\nabla(B(\beta \text{div} p + g))\right]_{i,j}}{1 + \tau\beta |\left[\nabla(B(\beta \text{div} p + g))\right]_{i,j}|}, \qquad \forall\ i,j.$$

Here, the step size $\tau$ is inversely proportional to the square root of the condition number of $K^T K + \alpha I$ which is very small for most reasonable choice of $\alpha$ (which is set to 0.008 in our experiments). Thus, a large number of steps is expected. Once the dual problem is solved, the solution $u^*$ to the original problem is recovered as

$$u^* = B(K^T f + \beta \mathrm{div} p^* + \lambda^*),$$

where $p^*$ and $\lambda^*$ are the optimal solution to the dual problem. Duality arguments can be used to show that the recovered optimal $u^*$ satisfies the non-negativity constraint, cf. Eq. (A.9) and (A.12).

# Chapter 3

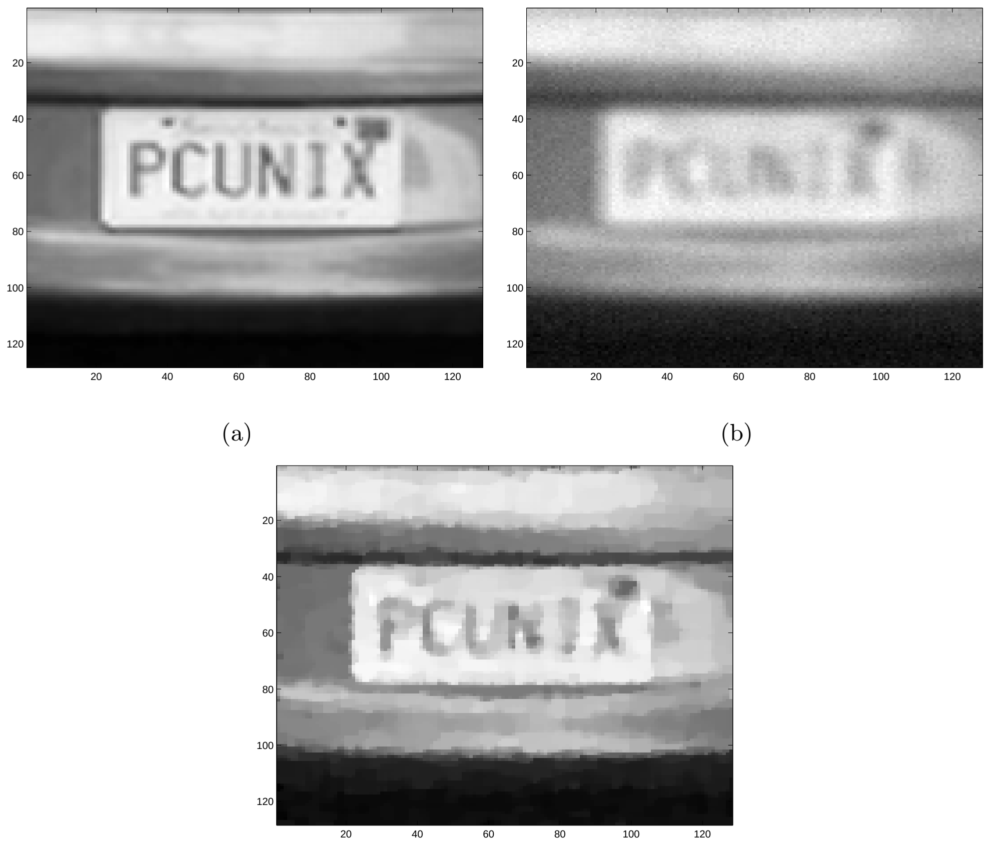
# Numerical Results

## 3.1 Introduction

In this section, we present extensive numerical results to demonstrate the performance of the NNCGM algorithm. We consider various conditions: different *signal-to-noise ratios* (SNR), different types and sizes of *point spread functions* (PSF) and different values of the smoothing parameter $\epsilon$. We also show the robustness of the NNCGM algorithm with respect to various parameters, and the performance of the FBIP preconditioner. The two images that will be used for the comparison purposes are the License Plate and Satellite images. The original images and typical results of TV deblurring for NNCGM are shown in Fig. 3.1 and 3.2.

## 3.2 Numerical Comparison with the PN and AM Algorithms

In the tests below, we vary one condition at a time, leaving the others to their moderately chosen values. In each test, we run each algorithm for a few different values of $\beta$ and choose the optimal $\beta$ that minimizes the $l^2$ reconstruction error. Unless otherwise mentioned, all results for NNCGM are with the use of the FBIP preconditioner, which significantly speeds up the processing. For PN, we tested both the ILU and FBIP preconditioners, and they caused the processing to be slower. Therefore, the results reported are without the use of any preconditioner. Our primary interest in Fig. 3.3 to Fig. 3.10 is the outer iterations

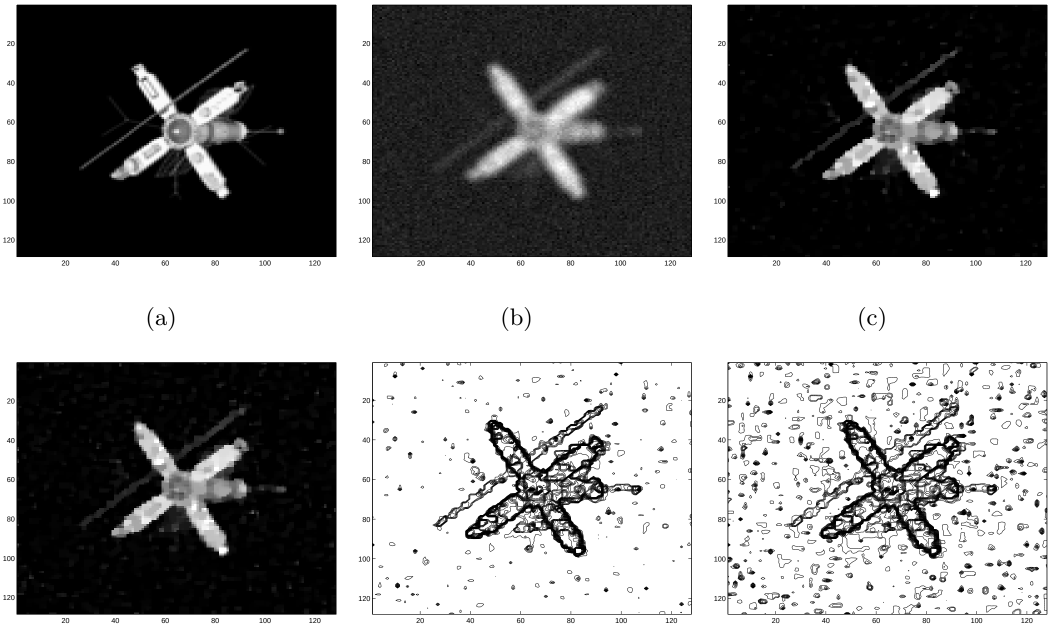


(a)

(b)

(c)

Figure 3.1: (a) Original License Plate image (128 × 128); (b) Blurred with Gaussian PSF 7 × 7, SNR 30dB; (c) TV deblurring results with NNCGM, $\beta = 0.4$.



(a) (b) (c)

(d) (e) (f)

Figure 3.2: Comparison of constrained and unconstrained deblurring. (a) Original Satellite image ($128 \times 128$); (b) Blurred and noisy data, Gaussian PSF of size $5 \times 5$, noise of SNR 15dB; (c) Non-negatively constrained deblurring result with $\beta = 0.4$, PSNR = 27.61dB; (d) Unconstrained CGM with $\beta = 0.4$ and with negative components set to 0, PSNR = 25.96dB; (e) Contour plot of the result in (c); (f) Contour plot of the result in (d).

which are largely independent of the inner iterations.

Fig. 3.3 and 3.4 compare the convergence of NNCGM, PN and AM for different SNRs of $-10$dB, 20dB, 50dB corresponding to high, medium and low level of noise respectively. A fixed Gaussian PSF of size $9 \times 9$ and a fixed $\epsilon$ of $10^{-2}$ were used. It is seen that the NNCGM method reaches a very high accuracy of KKT residual of the order of $10^{-6}$ and the convergence is quadratic eventually. An even higher accuracy can be achieved with only a few more iterations. The PN and AM methods become very slow in their progression after about 50 iterations. The total number of outer iterations for the NNCGM method stays below 70 even for a high noise level of $-10$dB.

Fig. 3.5 and 3.6 compare the convergence of NNCGM, PN and AM for varying Gaussian PSF size with a fixed SNR of 20 dB and a fixed $\epsilon$ of $10^{-2}$.

Fig. 3.7 and 3.8 compare the convergence of NNCGM, PN and AM for varying $\epsilon$ with fixed SNR of 20 dB and Gaussian PSF of size $9 \times 9$.

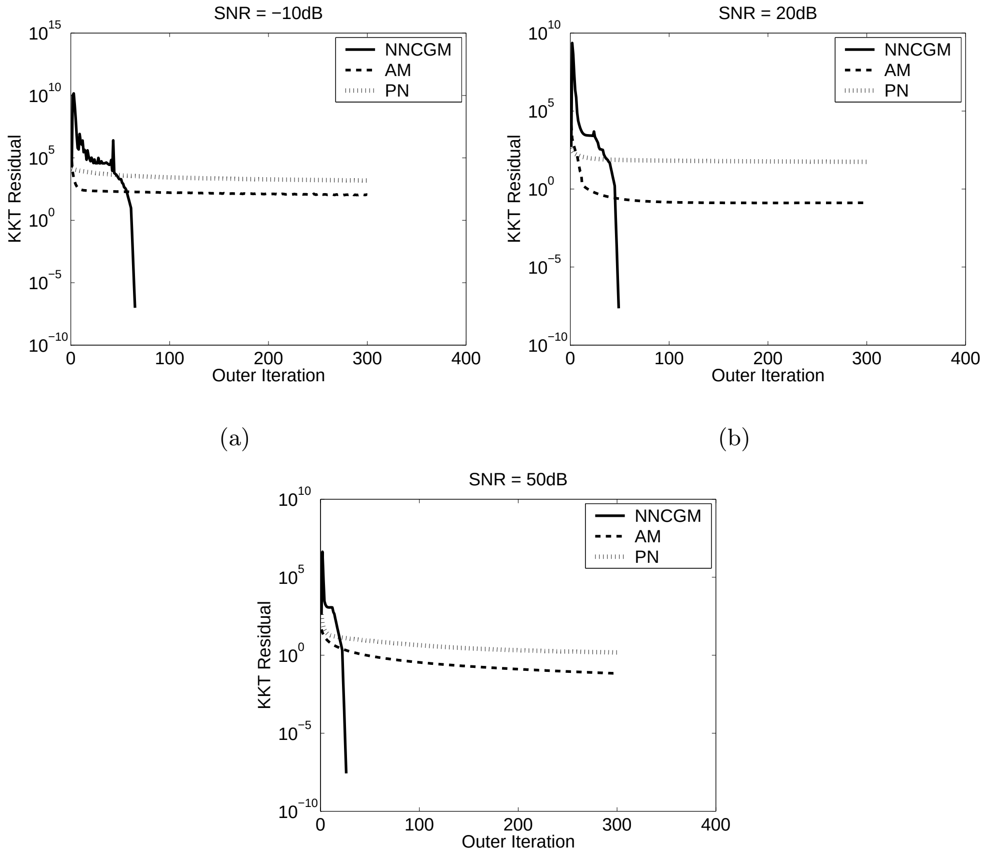
Fig. 3.9 and 3.10 compare the convergence of NNCGM, PN and AM for Gaussian blur and out-of-focus blur with a fixed SNR of 20dB, a fixed PSF size of $9 \times 9$ and a fixed $\epsilon$ of $10^{-2}$, for the License Plate and Satellite images respectively.

Tables 3.1 - 3.4 show the CPU timings in seconds and the *peak signal-to-noise ratio* (PSNR) in dB for the plots from Fig. 3.3-3.10. The PSNR, defined by

$$10 \log_{10} \left( \frac{255^2}{\frac{1}{mn} \|\text{original} - \text{reconstructed}\|^2} \right),$$

is a measure of image reconstruction error. Here, $m \times n$ are dimensions of the image. The larger the PSNR, the smaller the error will be. The figures in parentheses after each CPU timing for NNCGM refers to the total number of outer iterations required for each method. In all cases, we set the maximum number of iterations to 300, for both the PN and AM algorithms essentially stagnate after 300 iterations. The first sub-row in each row are License Plate data and the second sub-row are Satellite data. In each case, bold letters highlight the lowest CPU timings and the lowest reconstruction error (highest PSNR) among the three algorithms. All the algorithms were implemented in Matlab 7.2. CPU timings were measured on a Pentium D 3.2GHz processor with 4GB of RAM.

In most cases, the PN algorithm iterated for the maximum 300 iterations but did not

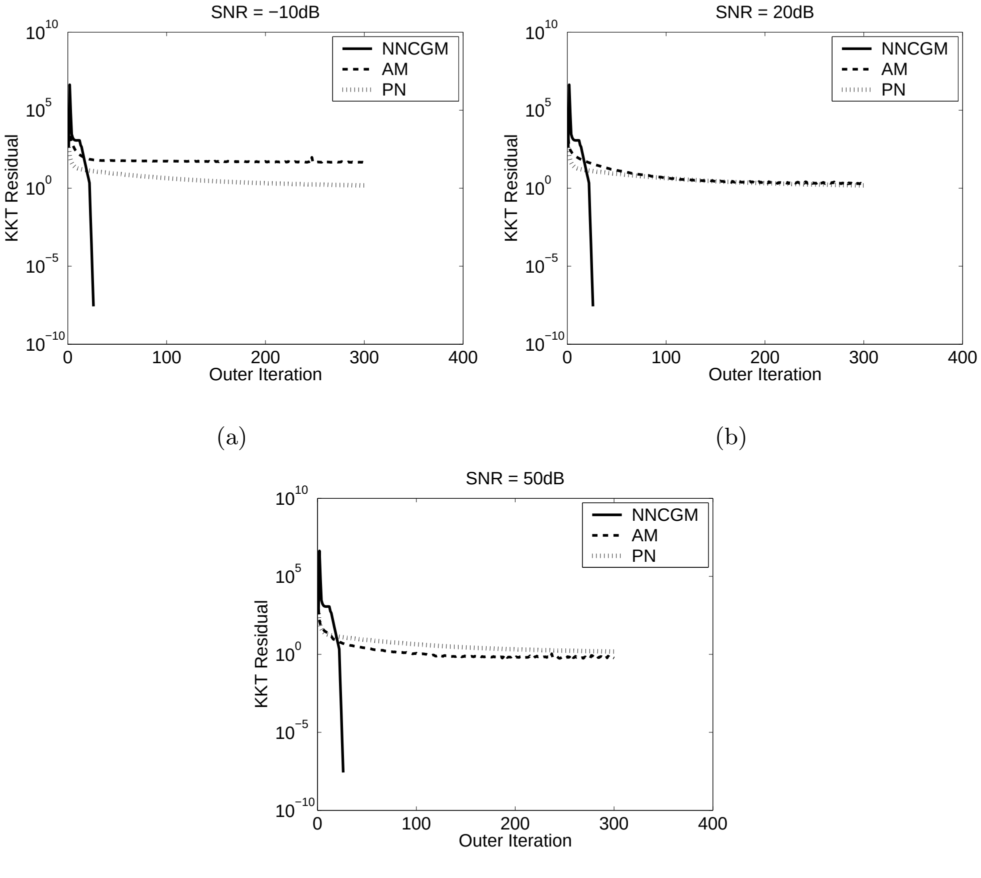


(a)

(b)

(c)

Figure 3.3: Convergence profiles of varying SNR for License Plate (a) −10 dB; (b) 20 dB; (c) 50 dB.

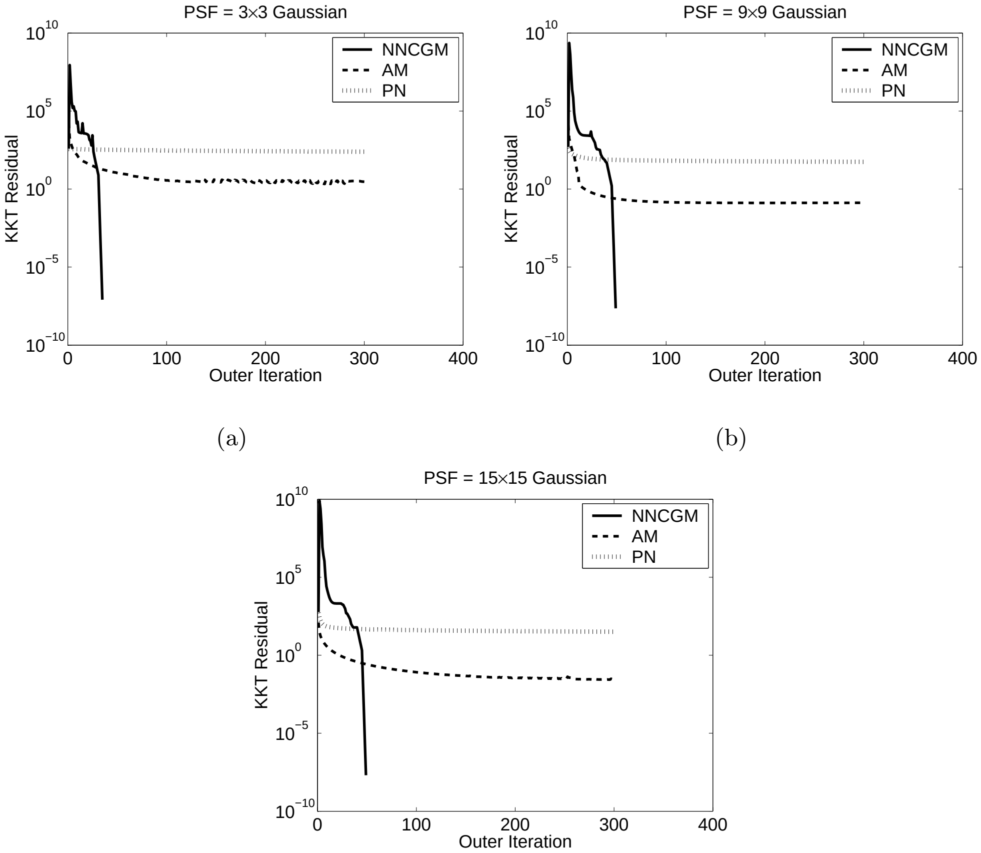


(a)

(b)

(c)

Figure 3.4: Convergence profiles of varying SNR for Satellite (a) −10 dB; (b) 20 dB; (c) 50 dB.

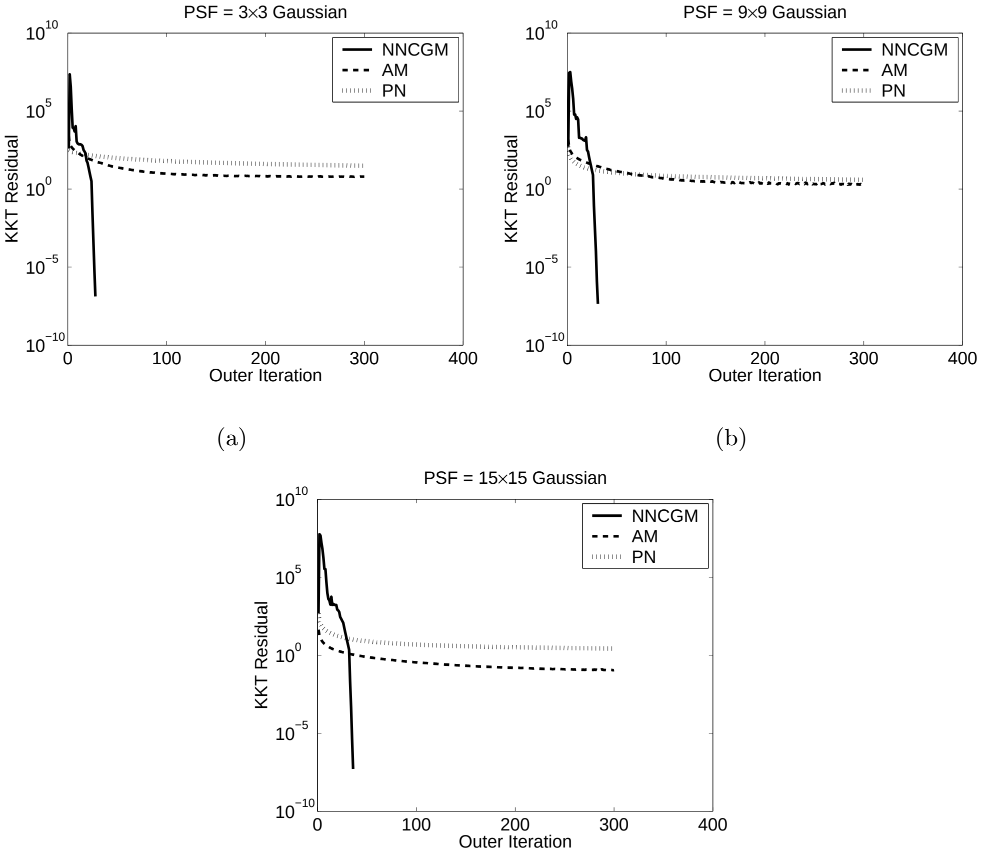


(a)

(b)

(c)

Figure 3.5: Convergence profiles of varying PSF for License Plate (a) Gaussian 3 × 3; (b) Gaussian 9 × 9; (c) Gaussian 15 × 15.

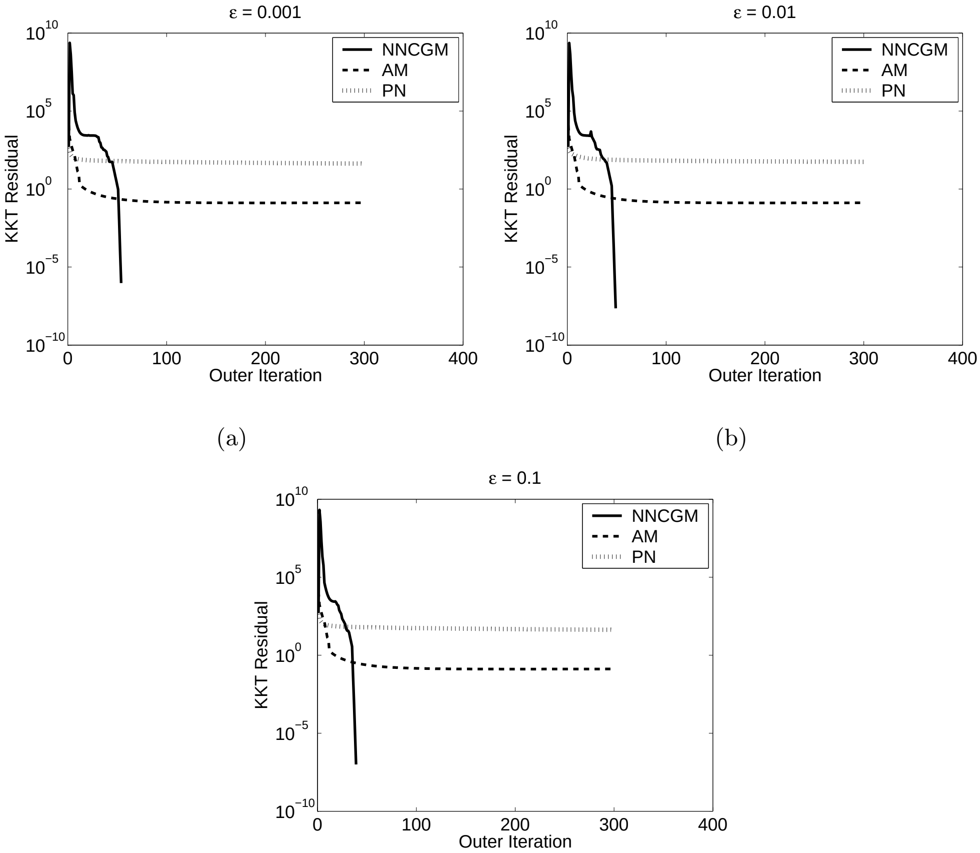


(a)

(b)

(c)

Figure 3.6: Convergence profiles of varying PSF for Satellite (a) Gaussian 3×3; (b) Gaussian 9 × 9; (c) Gaussian 15 × 15.

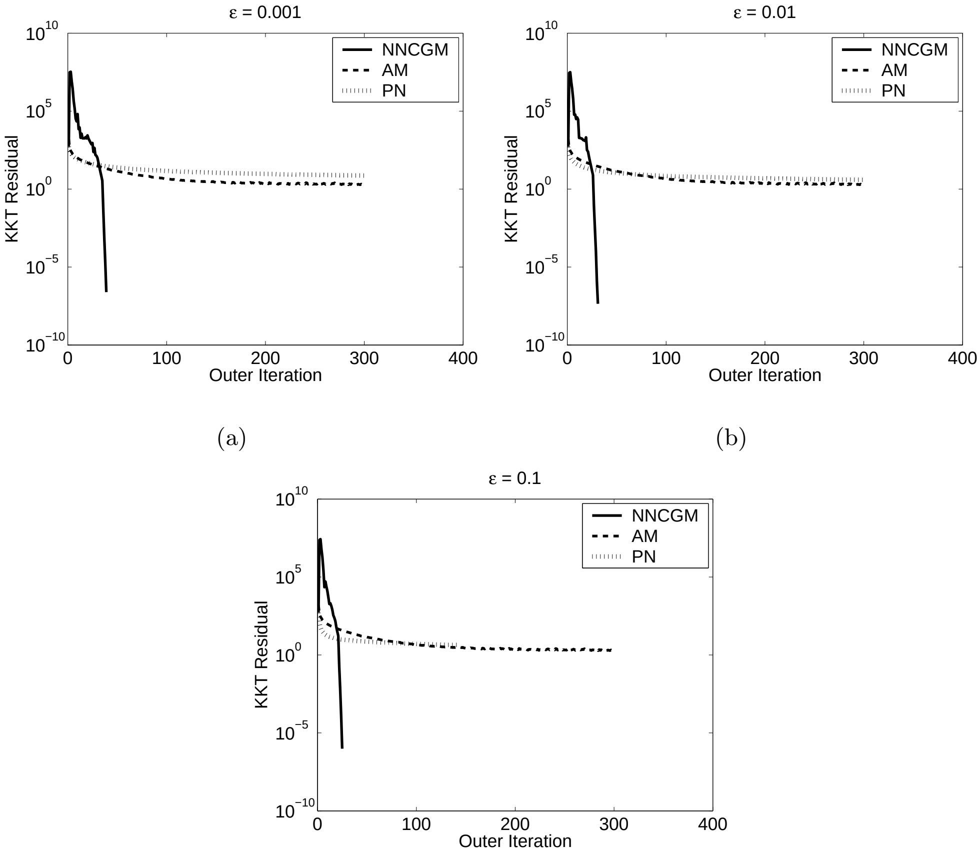


(a)

(b)

(c)

Figure 3.7: Convergence profiles of varying smoothing parameter $\epsilon$ for License Plate (a) $\epsilon = 0.001$; (b) $\epsilon = 0.01$; (c) $\epsilon = 0.1$.

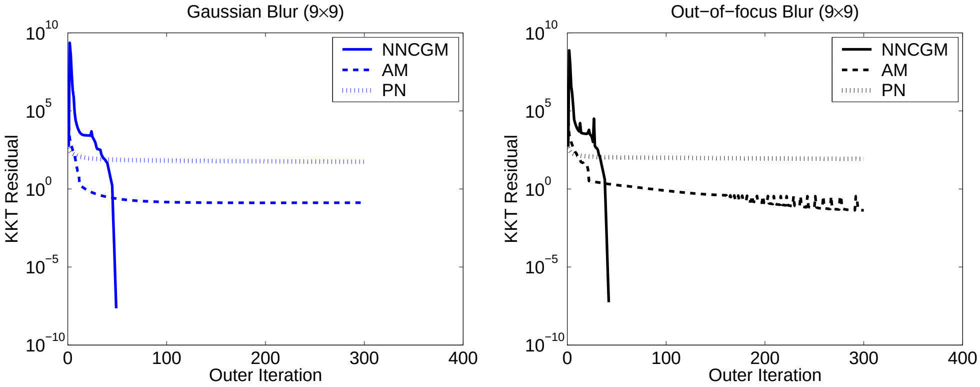


(a)

(b)

(c)

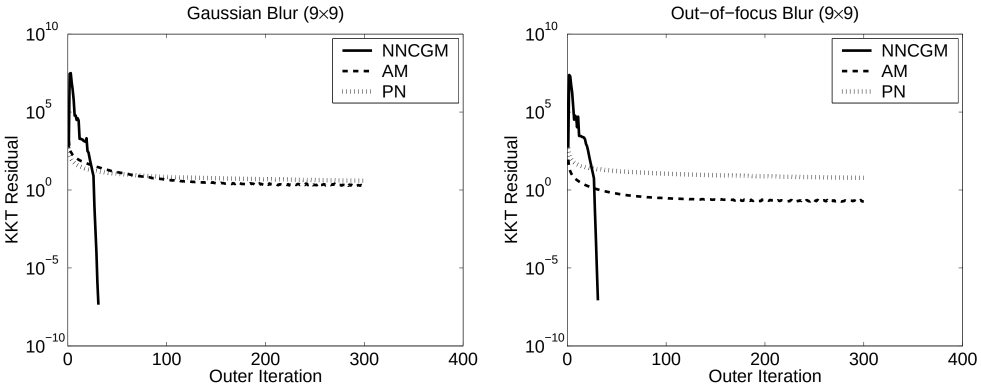
Figure 3.8: Convergence profiles of varying smoothing parameter $\epsilon$ for Satellite (a) $\epsilon = 0.001$; (b) $\epsilon = 0.01$; (c) $\epsilon = 0.1$.



(a)

(b)

Figure 3.9: Convergence profiles for different types of blur for License Plate (a) Gaussian; (b) Out-of-focus.



(a)

(b)

Figure 3.10: Convergence profiles for different types of blur for Satellite (a) Gaussian; (b) Out-of-focus.

reach even a moderate level of accuracy for the stopping criterion (the KKT residual). So, the CPU timings for the PN algorithm would have been much higher if we iterated until convergence. Moreover, it is well-known that primal-only methods such as PN have a small domain of convergence [7, 37]. We would have had to use a large value of $\epsilon$ to enlarge the domain of convergence to make PN converge. The NNCGM algorithm shows considerable robustness to the initial guess even for a small value of $\epsilon$ (see Fig. 3.11 below), therefore allowing for a greater accuracy. It is seen that the NNCGM algorithm outperforms the other two both in terms of CPU time and PSNR in most of the tested cases. NNCGM is also the only method that is able to achieve very high levels of accuracy within a reasonable time, and so we do not need the usual trade-off between time taken and accuracy. The AM algorithm, while having a larger domain of convergence, is too slow for practical purposes.

## 3.3 Robustness of NNCGM

In this section, we test the robustness of the NNCGM algorithm with respect to various parameters. Fig. 3.11 shows the convergence of NNCGM for extremely small values of $\epsilon$, a fixed SNR of 20dB and a Gaussian PSF of size $9 \times 9$ for the License Plate image. It is seen that even for small values of $\epsilon$, the rate of convergence is eventually quadratic. In contrast, PN diverges for such small values of $\epsilon$.

Fig. 3.12 shows the convergence of NNCGM for different values of $c$, a fixed SNR of

| SNR | Image | NNCGM | | PN | | AM | |
|---|---|---|---|---|---|---|---|
| | | CPU | PSNR | CPU | PSNR | CPU | PSNR |
| $-10$ | License Plate | 449.71 (65) | 11.88 | **369.4** (> 300) | **12.32** | 6630 (> 300) | 8.16 |
| | Satellite | **124.92** (62) | **29.64** | 150.64 (> 300) | 23.42 | 6360 (> 300) | 20.94 |
| 20 | License Plate | **192.28** (49) | **31.79** | 497.01 (> 300) | 29.96 | 3720 (> 300) | 30.82 |
| | Satellite | **49.46** (31) | **37.50** | 77.57 (> 300) | 36.31 | 5340 (> 300) | 32.22 |
| 50 | License Plate | **103.64** (26) | **37.79** | 309.95 (> 300) | 34.29 | 3570 (> 300) | 32.22 |
| | Satellite | **52.76** (27) | **38.47** | 96.17 (> 300) | 36.16 | 6040 (> 300) | 36.09 |

Table 3.1: CPU time in seconds and PSNR for varying SNRs. The figure in parentheses after the CPU timing is the total number of outer iterations (max. 300).

| PSF | Image | NNCGM | | PN | | AM | |
|---|---|---|---|---|---|---|---|
| | | CPU | PSNR | CPU | PSNR | CPU | PSNR |
| $3 \times 3$ | License Plate | **49.04** (35) | 37.29 | 382.22 (> 300) | 31.91 | 5280 (> 300) | **37.8** |
| | Satellite | **28.37** (42) | 45.53 | 148.15 (> 300) | 45.78 | 6050 (> 300) | **45.86** |
| $9 \times 9$ | License Plate | **205.26** (49) | **31.79** | 483.29 (> 300) | 29.91 | 3710 (> 300) | 30.82 |
| | Satellite | **48.82** (42) | **37.50** | 78.92 (> 300) | 36.31 | 5920 (> 300) | 36.06 |
| $15 \times 15$ | License Plate | **371.74** (49) | **28.25** | 614.15 (> 300) | 28.16 | 4200 (> 300) | 28.23 |
| | Satellite | 107.09 (42) | **35.54** | **84.31** (> 300) | 33.04 | 5210 (> 300) | 32.87 |

Table 3.2: CPU time in seconds and PSNR for different PSFs. The figure in parentheses after the CPU timing is the total number of outer iterations (max. 300).

| $\epsilon$ | Image | NNCGM | | PN | | AM | |
|---|---|---|---|---|---|---|---|
| | | CPU | PSNR | CPU | PSNR | CPU | PSNR |
| 0.001 | License Plate | **309.09** (54) | **31.79** | 356.84 ($>300$) | 29.8 | 3740 ($>300$) | 30.82 |
| | Satellite | **59.70** (42) | **37.5** | 128.87 ($>300$) | 34.79 | 5876 ($>300$) | 36.06 |
| 0.01 | License Plate | **206.17** (49) | **31.79** | 481.07 ($>300$) | 29.96 | 3749 ($>300$) | 30.82 |
| | Satellite | **48.81** (42) | **37.50** | 77.25 ($>300$) | 36.31 | 5280 ($>300$) | 36.06 |
| 0.1 | License Plate | **138.06** (39) | **31.8** | 352.42 ($>300$) | 29.8 | 3740 ($>300$) | 30.82 |
| | Satellite | 41.32 (42) | **37.51** | **30.85** ($>300$) | 35.31 | 5350 ($>300$) | 36.06 |

Table 3.3: CPU time in seconds and PSNR for different $\epsilon$'s. The figure in parentheses after the CPU timing is the total number of outer iterations (max. 300).

| Blur Type | Image | NNCGM | | PN | | AM | |
|---|---|---|---|---|---|---|---|
| | | CPU | PSNR | CPU | PSNR | CPU | PSNR |
| Gaussian | License Plate | **210.31** (49) | **31.79** | 478.35 (> 300) | 29.96 | 3650 (> 300) | 30.82 |
| | Satellite | **48** (42) | **37.50** | 75.67 (> 300) | 36.31 | 3670 (> 300) | 31.02 |
| Out-of-focus | License Plate | **162.70** (42) | **31.79** | 478.35(> 300) | 29.96 | 6096(> 300) | 36.06 |
| | Satellite | **41** (42) | 37.89 | 85.98 (> 300) | **38.62** | 6470 (> 300) | 37.61 |

Table 3.4: CPU time in seconds and PSNR for different blur types. All PSFs have size 9 × 9. The figure in parentheses after the CPU timing is the total number of outer iterations (max. 300).

20dB and a Gaussian PSF of size 9 × 9 for the License Plate image. The convergence plots are almost identical. This is in keeping with the theoretical results obtained in [2] for the PDAS algorithm.

NNCGM can be made significantly faster with the use of standard preconditioners. We tested the NNCGM method with an ILU preconditioner and a FBIP preconditioner. The best performance was seen with the FBIP preconditioner. Fig. 3.13 shows the number of inner CG iterations for each outer iteration for the FBIP preconditioner with varying PSF size, a fixed SNR of 20dB and a Gaussian PSF of size $9 \times 9$. These results are for the License Plate image. It is seen that the number of inner CG iterations shows a slight tendency to increase. But the number of CG iterations still remains low. Moreover, a very high accuracy is obtained so that there is no need to carry out further outer iterations. In this sense the slight increase in the number of CG iterations does not pose any serious problem.

There are three possible reasons for the increase in the number of inner CG iterations and the overall time taken to process as the PSF size increases. Firstly, as the system approaches convergence, the linear system Eq. (2.9) is increasingly ill-conditioned and thus the CG requires more iterations to solve the problem. Secondly, as the PSF size increases, the matrix $K$ has more bands. This means that the construction of the preconditioner takes more time. Thirdly, since we keep the bandwidth of the preconditioner fixed to 2 in most of the tests, the approximation error of the preconditioner becomes larger as the PSF size increases. This can be remedied by increasing the bandwidth of the preconditioner. However, it is not our purpose here to fine-tune the preconditioner performance.

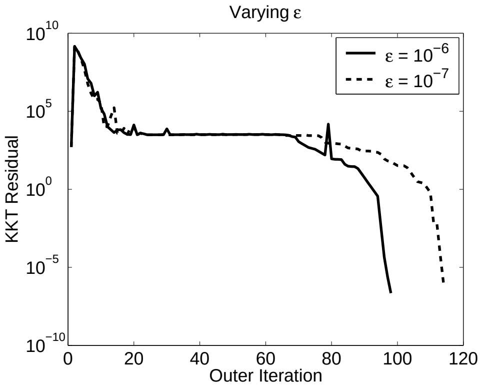


Figure 3.11: Convergence plots for NNCGM for very small values of $\epsilon$ for License Plate.

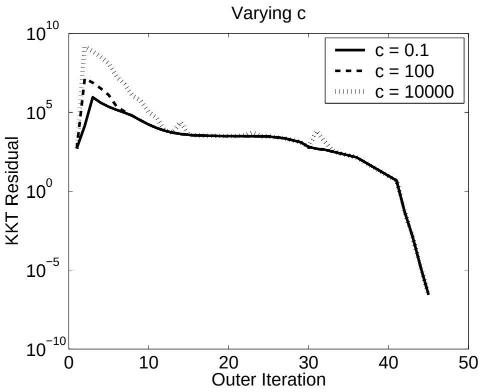
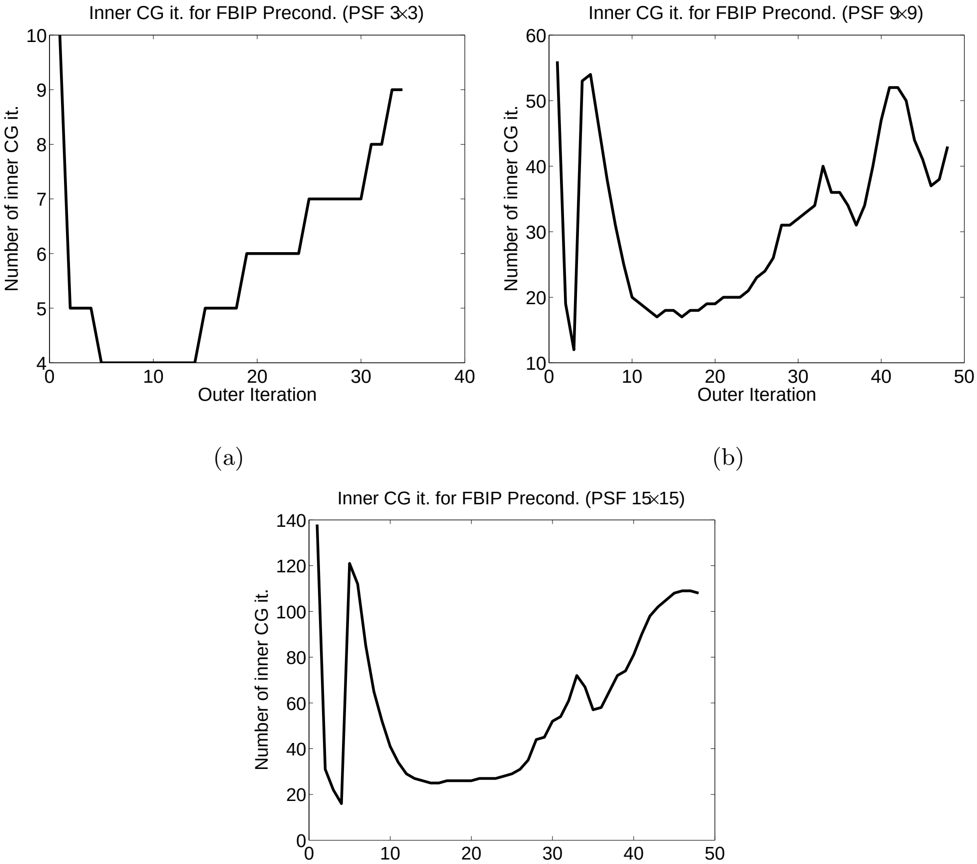


Figure 3.12: Convergence plots for NNCGM for varying $c$ for License Plate.



(a)

(b)

(c)

Figure 3.13: Inner CG iterations for FBIP preconditioner and PSF (a) Gaussian $3 \times 3$; (b) Gaussian $9 \times 9$; (c) Gaussian $15 \times 15$.

# Chapter 4

# Conclusion

Our experiments indicate that the NNCGM method is very robust in performance over a wide range of parameters. The rate of convergence is locally quadratic in all cases. The method is able to solve the non-negatively constrained TV deblurring problem to a high level of accuracy. Both Gaussian and out-of-focus blurs can be solved equally fast. The reduced system Eq. (2.9) has a simple structure and the standard FBIP preconditioner helps in significantly speeding up the processing time. We have also found in our tests that the method is highly robust to the initial guess of the image. In all the tests above, the initial guess was simply taken to be the observed image with negative components set to zero. Using the observed image itself gave very similar convergence results. Maintaining feasibility of the variables can be effectively done in a straightforward way whereas we found it difficult to do so with standard interior-point methods. All parameters can be left at their default values, except in the case of extremely small values of the regularization parameter $\epsilon$ (values below $10^{-6}$). In this case, the bandwidth of the FBIP preconditioner needs to be increased from the default value of 2 to overcome the highly ill-conditioning of the linear system Eq.(2.9). But we found that setting $\epsilon = 10^{-2}$ gives a good balance between quality and speed.

A topic for further research may be the inclusion of upper bounds on the original problem, since real-world images have gray levels that fall between the values of 0 and 255, for 8-bit images. Intuitively, it could be expected that this will further improve the accuracy of the deblurring reconstruction.

# Bibliography

[1] T. F. Chan, G. H. Golub, and P. Mulet, "A nonlinear primal-dual method for total variation-based image restoration," *SIAM J. Sci. Comput.*, vol. 20, no. 6, pp. 1964–1977, 1999.

[2] M. Hintermüller, K. Ito, and K. Kunisch, "The primal-dual active set strategy as a semismooth Newton's method," *SIAM J. Optim.*, vol. 13, no. 3, pp. 865–888, 2003.

[3] J. Starck, E. Pantin, and F. Murtagh, "Deconvolution in astronomy: A review," *Publications of the Astronomical Society of the Pacific*, vol. 114, pp. 1051–1069, 2002.

[4] S. M. Riad, "The deconvolution problem: An overview," *Proceedings of the IEEE*, vol. 74, no. 1, pp. 82–85, 1986.

[5] L. I. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259–268, 1999.

[6] Y. You and M. Kaveh, "Blind image restoration by anisotropic regularization," *IEEE Trans. Image Process.*, vol. 8, pp. 396–407, 1999.

[7] C. R. Vogel and M. E. Oman, "Iterative method for total variation denoising," *SIAM J. Sci. Comput.*, vol. 17, pp. 227–238, 1996.

[8] Y. Li and F. Santosa, "A computational algorithm for minimizing total variation in image reconstruction," *IEEE Trans. Image Process.*, vol. 5, pp. 987–995, 1996.

[9] T. F. Chan and J. Shen, "Mathematical models for local non-texture inpainting," *SIAM J. Appl. Math.*, vol. 62, pp. 1019–1043, 2001.

[10] S. Osher, A. Sole, and L. Vese, "Image decomposition and restoration using total variation minimization and the $H^{-1}$ norm," *Multiscale Model. Simul.*, vol. 1, no. 3, pp. 349–370, 2003.

[11] V. Kolehmainen, A. Vanne, S. Siltanen, S. Jarvenpaa, J. P. Kaipio, M. Lassas, and M. Kalke, "Parallelized Bayesian inversion for three-dimensional dental x-ray imaging," *IEEE Trans. Medical Imaging*, vol. 25, no. 2, pp. 218–228, 2006.

[12] M. Persson, D. Bone, and H. Elmqvist, "Total variation norm for three-dimensional iterative reconstruction in limited view angle tomography," *Phys. Med. Biol.*, vol. 46, pp. 853–866, 2001.

[13] N. Dey, L. Blanc-Féraud, C. Zimmer, C. Kam, J. Olivo-Marin, and J. Zerubia, "A deconvolution method for confocal microscopy with total variation regularization," in *Proc. IEEE Intl. Symposium on Biomedical Imaging: Macro to Nano*, vol. 2, 2004, pp. 1223–1226.

[14] A. N. Tikhonov and V. Y. Arsenin, *Solutions to Ill-posed Problems.* Washington, D.C.: Winston, 1977.

[15] T. F. Chan and C. K. Wong, "Total variation blind deconvolution," *IEEE Trans. on Image Process.*, vol. 7, no. 3, pp. 370–375, 1998.

[16] D. Krishnan, P. Lin, and X.-C. Tai, "An efficient operator splitting method for noise removal in images," *Commun. Comput. Phys.*, vol. 1, no. 5, pp. 847–858, 2006.

[17] M. Lysaker, S. Osher, and X.-C. Tai, "Noise removal using smoothed normals and surface fitting," *IEEE Trans. Image Process.*, vol. 13, no. 10, pp. 1345–1357, 2004.

[18] K. Ito and K. Kunisch, "An active set strategy based on the augumented Lagrangian formulation for image restoration," *ESAIM: Math. Model. Numer. Anal.*, vol. 33, pp. 1–21, 1999.

[19] T. Chan and K. Chen, "An optimization based total variation image denoising," *Multiscale Model. Simul.*, vol. 5, no. 2, pp. 615–645, 2006.

[20] D. Goldfarb and W. Yin, “Second-order cone programming methods for total variation based image restoration,” *SIAM J. Sci. Comput.*, vol. 27, no. 2, pp. 622–645, 2005.

[21] H. Fu, M. K. Ng, M. Nikolova, and J. L. Barlow, “Efficient minimization methods of mixed $l^2$-$l^1$ and $l^1$-$l^1$ norms for image restoration,” *SIAM J. Sci. Comput.*, vol. 27, no. 6, pp. 1881–1902, 2006.

[22] A. Chambolle, *Total Variation Minimization and a Class of Binary MRF Models.* Springer Verlag, 2005, vol. 3522, pp. 351–359, Lecture Notes in Computer Science, Proc. of EMMCVPR.

[23] T. Lü, P. Neittaanmäki, and X.-C. Tai, “A parallel splitting up method for partial differential equations and its application to navier-stokes equations,” *RAIRO Math Model. And Numer. Anal.*, vol. 26, pp. 673–708, 1992.

[24] J. L. Carter, “Dual methods for total variation-based image restoration,” Ph.D. dissertation, UCLA, April 2002.

[25] A. Chambolle, “An algorithm for total variation minimization and applications,” *J. Math. Imaging Vision*, vol. 20, pp. 89–97, 2004.

[26] M. Ng, L. Qi, Y. Yang, and Y. Huang, “On semismooth Newton’s methods for total variation minimization,” *J. Math. Imaging Vision*, to appear.

[27] M. Hintermüller and K. Kunisch, “Total bounded variation regularization as a bilaterally constrained optimisation problem,” *SIAM J. Appl. Math*, vol. 64, pp. 1311–1333, 2004.

[28] R. W. Schafer, R. M. Mersereau, and M. A. Richards, “Constrained iterative restoration algorithms,” *Proceedings of the IEEE*, vol. 69, no. 4, pp. 432–450, 1981.

[29] K. Ho, C. Beling, S. Fung, K. Chong, M. Ng, and A. Yip, “Deconvolution of coincidence Doppler broadening spectra using iterative projected Newton method with non-negativity constraints,” *Review of Scientific Instruments*, vol. 74, pp. 4779–4787, 2003.

[30] C. R. Vogel, *Computational Methods for Inverse Problems.* Philadelphia: SIAM, 2002.

[31] J. M. Bardsley and C. R. Vogel, "A nonnnegatively constrained convex programming method for image reconstruction," *SIAM J. Sci. Comput.*, vol. 25, pp. 1326–1343, 2004.

[32] M. Hanke, J. Nagy, and C. R. Vogel, "Quasi-Newton approach to nonnegative image restoration," *Linear Algebra and Its Applications*, vol. 316, pp. 223–236, 2000.

[33] C. R. Vogel, "Solution of linear systems arising in nonlinear image deblurring," in *Scientific Computing: Proceedings of the Workshop*, G. Golub, S. Lui, F. Luk, and R. Plemmons, Eds. Hong Kong: Springer, 1997, pp. 148–158.

[34] D. P. Bertsekas, "Projected Newton methods for optimization problems with simple constraints," *SIAM J. Control and Optim.*, vol. 20, no. 2, pp. 221–246, 1982.

[35] Y. Saad, *Iterative Methods for Sparse Linear Systems*, 2nd ed. Philadelphia: SIAM, 2003.

[36] D. Krishnan, P. Lin, and M. A. Yip, "A primal-dual active-set method for non-negativity constrained total variation deblurring problems," *To appear in IEEE Trans. Image Proc.*

[37] T. Chan, H. M. Zhou, and R. H. Chan, "Continuation method for total variation denoising problems," in *Proceedings to the SPIE Symposium on Advanced Signal Processing*, ser. Algorithms, Architectures, and Implementations, pp. 314-325, F. Luk, Ed., vol. 2563, 1995.

[38] I. Ekeland and R. Temam, *Convex Analysis and Variational Problems*, ser. Classics in Appl. Math. Philadelphia: SIAM, 1999, no. 28.

[39] S. Boyd and L. Vandenberghe, *Convex Optimization.* Cambridge University Press, 2004.

[40] F. Lin, M. Ng, and W. Ching, "Factorized banded inverse preconditioners for matrices with Toeplitz structure," *SIAM J. Sci. Comput.*, vol. 26, pp. 1852–1870, 2005.

[41] L. K. Saul, F. Sha, and D. D. Lee, “Statistical signal processing with nonnegativity constraints,” in *Proc. EuroSpeech 2003*, vol. 2, Geneva, Switzerland, 2003, pp. 1001–1004.

[42] L. Qi and J. Sun, “A nonsmooth version of Newton’s method,” *Mathematical Programming*, vol. 58, pp. 353–367, 1993.

# Appendix A

# Derivation of the Primal-Dual Program and the NNCGM Algorithm

## A.1 The Primal-Dual Program

The general primal problem under consideration (with the $l^2$ regularization parameter $\alpha$) is

$$\min_{u,u\geq 0} \frac{1}{2}\|Ku-f\|^2 + \frac{\alpha}{2}\|u\|^2 + \beta\|u\|_{TV},$$

which can be written as,

$$\min_{u} \left\{ \frac{1}{2}\|Ku-f\|^2 + \frac{\alpha}{2}\|u\|^2 + \beta\|u\|_{TV} + I(u \geq 0) \right\}, \tag{A.1}$$

where $I(u \geq 0) = 0$ if $u \geq 0$ for all components of $u$, $I(u \geq 0) = \infty$ if any of the components of $u$ are $< 0$. Using the Fenchel transform of the TV norm in Eq. (1.2), we have

$$\|u\|_{TV} = \sup_{p,|p_{i,j}|\leq 1} \langle u, -\text{div}p \rangle \tag{A.2}$$

where $p$ is the dual variable and div is the discrete divergence operator defined in Eq. (2.2) and Eq. (2.3) respectively. Substituting Eq. (A.2) into Eq. (A.1), we have

$$\min_{u} \left\{ \frac{1}{2}\|Ku-f\|^2 + \frac{\alpha}{2}\|u\|^2 + \beta \sup_{p,|p_{i,j}|\leq 1} \langle u, -\text{div}p \rangle + I(u \geq 0) \right\},$$

or

$$\min_u \max_p \left\{ \frac{1}{2}\|Ku - f\|^2 + \frac{\alpha}{2}\|u\|^2 + \beta\langle u, -\mathrm{div}p\rangle + I(u \geq 0) - I(|p_{i,j}| \leq 1) \right\},$$

where it is understood that $I(|p_{i,j}| \leq 1) = \infty$ if the Euclidean norm of any of the components of $p$ is greater than 1. By the convexity of the objective, we have

$$\max_p \min_u \left\{ \frac{1}{2}\|Ku - f\|^2 + \frac{\alpha}{2}\|u\|^2 + \beta\langle u, -\mathrm{div}p\rangle + I(u \geq 0) - I(|p_{i,j}| \leq 1) \right\}. \tag{A.3}$$

Consider the inner minimization for a fixed $p$. The Lagrangian for this problem is given by

$$L(u, \lambda) := \frac{1}{2}\|Ku - f\|^2 + \frac{\alpha}{2}\|u\|^2 + \beta\langle u, -\mathrm{div}p\rangle - \langle u, \lambda\rangle, \tag{A.4}$$

where $\lambda$ is the Lagrange multiplier for $u \geq 0$. Then,

$$\nabla_u L = (K^T K + \alpha I)u - K^T f - \beta \mathrm{div}p - \lambda.$$

Solving $\nabla_u L = 0$ gives

$$u = B(K^T f + \beta \mathrm{div}p + \lambda), \tag{A.5}$$

where $B := (K^T K + \alpha I)^{-1}$. Next, we plug-in Eq. (A.5) into Eq. (A.4) to obtain

$$L(\lambda) = -\frac{1}{2}\|B^{1/2}(K^T f + \beta \mathrm{div}p + \lambda)\|^2 + \frac{1}{2}\|f\|^2. \tag{A.6}$$

Now, we plug-in (A.6) into (A.3) and change $\min_u$ to $\max_\lambda$. After simplification, we get

$$\max_p \max_\lambda \left\{ -\frac{1}{2}\|B^{1/2}(K^T f + \beta \mathrm{div}p + \lambda)\|^2 + \frac{1}{2}\|f\|^2 - I(\lambda \geq 0) - I(|p_{i,j}| \leq 1) \right\}.$$

Equivalently, we arrive at the following dual problem:

$$\min_p \min_\lambda \left\{ \frac{1}{2}\|B^{1/2}(K^T f + \beta \mathrm{div}p + \lambda)\|^2 - \frac{1}{2}\|f\|^2 + I(\lambda \geq 0) + I(|p_{i,j}| \leq 1) \right\}. \tag{A.7}$$

## A.2 Optimality Conditions

The KKT for (A.7) is as follows

$$\begin{aligned}
-\beta \nabla B(\beta \mathrm{div}p + K^T f + \lambda) + \mu \odot p &= 0, & \text{(A.8)}\\
-B(\beta \mathrm{div}p + K^T f + \lambda) + u &= 0, & \text{(A.9)}\\
\mu \odot (|p|^2 - 1) &= 0, & \text{(A.10)}\\
u \odot \lambda &= 0, & \text{(A.11)}\\
\lambda, u &\geq 0, & \text{(A.12)}\\
|p_{i,j}| &\leq 1, \qquad \forall\ i, j. & \text{(A.13)}
\end{aligned}$$

Here, we identify the Lagrange multipliers for $\lambda \geq 0$ with $u$. This is because of Eq. (A.9) above, which is the same as Eq. (A.5). The equation $u \odot \lambda = 0$ is understood as component-wise multiplication. The equation $\mu \odot (|p|^2 - 1) = 0$ is understood as $\mu_{i,j}(|p_{i,j}|^2 - 1) = 0$ for all $i, j$. The expression $\mu \odot p$ is understood as $\mu_{i,j} p_{i,j}$ for all $i, j$. Instead of solving the above KKT system directly, we follow a technique developed in [1]. We solve the following equivalent system of equations:

$$p|\nabla u|_\epsilon - \nabla u = 0, \tag{A.14}$$

$$-\beta \mathrm{div} p - K^T f - \lambda + Au = 0, \tag{A.15}$$

$$\lambda - \max\{0, \lambda - cu\} = 0. \tag{A.16}$$

Here, $A := K^T K + \alpha I = B^{-1}$ and $c$ is an arbitrary positive constant. For notational convenience, we denote the left hand side of Eq. (A.14)-(A.16) by $F_1(p, u, \lambda)$, $F_2(p, u, \lambda)$ and $F_3(p, u, \lambda)$ respectively. To see the equivalency of the two systems, we note that:

1. Eq. (A.9) implies that $u = B(\beta \mathrm{div} p + K^T f + \lambda)$. Therefore, Eq. (A.8) can be reduced to $\mu \odot p = \beta \nabla u$. Taking the norm on both sides, along with the complementarity conditions Eq. (A.10), give us $\mu = \beta |\nabla u|$. Hence, we have $p|\nabla u| = \nabla u$ which is the same as $F_1 = 0$ (cf. Eq. (A.14)) when $\epsilon = 0$.

2. The equation $F_2 = 0$ (cf. Eq. (A.15)) is simply the Eq. (A.9) restated.

3. The equation $F_3 = 0$ (cf. Eq. (A.16)) is a standard technique to express Eq. (A.11) and Eq. (A.12) as a single equality constraint.

The backward construction of Eq. (A.8)-(A.12) from Eq. (A.14)-(A.16) can also be easily done. In addition, it can be seen that Eq. (A.13) is implied by $F_1 = 0$. Therefore, the two systems are equivalent (when $\epsilon = 0$). Notice that an $\epsilon$-regularization is added to the left-hand side of $F_1 = 0$ owing to the possibility that $\nabla u = 0$.

## A.3 The NNCGM Algorithm

In Eq. (A.14), an $\epsilon$-regularization has been added to overcome the non-differentiability of the term $|\nabla u|$, in $F_1(p, u, \lambda)$. The function $F_1$ is non-differentiable when $|\nabla u| = 0$. The equation (A.14) is required in the first place to maintain the nonlinear constraints $|p_{i,j}| \leq 1$.

The function $F_3(p, u, \lambda)$ (Eq. (A.16)) is similarly non-differentiable. This function is required to maintain the non-negativity constraints on $u$ (as well as to satisfy the KKT complementarity conditions). One approach to solving this problem could be to smooth this function by using a smoothing approximation as for $F_1(p, u, \lambda)$. This would then enable applying a smooth Newton's step for this method.

Instead, we propose to use a *semi-smooth* Newton's method based on slant differentiability of the max-function. This allows us to overcome the non-differentiability of $F_3(p, u, \lambda)$. Furthermore, local convergence results are also applicable. It may be possible to use a semi-smooth Newton's approach to handling the non-linear equation $F_1(p, u, \lambda) = 0$ as well, but we have not pursued this here. This may be a topic for further research.

The concept of slant differentiability as introduced in [2] allows the derivation of a semi-smooth Newton's method based on using a slanting function in place of the Jacboian (which does not exist for the system Eq. (A.14) - (A.16). The slanting function is defined as follows in [2]: Let $X$ and $Z$ be Banach spaces. The mapping $F : D \subset X \to Z$ is called slantly differentiable in the open subset $U \subset D$ if there exists a family of mappings $G : U \to \mathcal{L}(X, Z)$ such that

$$\lim_{h \to 0} \frac{\|F(x+h) - F(x) - G(x+h)h\|}{\|h\|} = 0.$$

If a bounded slanting function $G$ exists for a function $F$, Theorem 1.1 in [2] proves that the following Newton's iteration is locally quadratically convergent:

$$x^{k+1} = x^k - G(x^k)^{-1} F(x^k).$$

However, no global convergence results are available to us for this problem, though Theorem 3.3 of [42] could provide some leads to this problem. This is a question for further research. The numerical results shown in Section 3 confirm the local quadratic convergence. Our numerical experience also shows that the algorithm is very robust to initialization, and so there is some numerical evidence of global convergence.

Furthermore, it can be shown that the following matrix function given by a diagonal matrix $G_m(v)$ with diagonal elements

$$G_m(v)_{ii} = \begin{cases} 1 & \text{if } v_i > 0 \\ 0 & \text{if } v_i \leq 0 \end{cases}$$

serves as a (bounded) slanting function for the max-function. Together, the above results allow us to specify a semi-smooth Newton's method for the system (A.14) - (A.16).

Furthermore, it is shown in [2] that the semi-smooth Newton's method derived is equivalent to a primal-dual active set strategy. This is seen by setting the active and inactive sets to be those components of the max-function that are less than or equal to, or greater than 0, respectively. The corresponding Jacobian entries are taken from the above definition of $G_m$. The other two functions $F_1(p,u,\lambda)$ and $F_2(p,u,\lambda)$, are differentiable and so their Jacobians are defined. The process of deriving the PDAS method is shown below.

The semi-smooth Newton's update for the equations (A.14)-(A.16) is given by the following system:

$$\begin{bmatrix} |\nabla u|_\epsilon & -\left(I - \frac{p(\nabla u)^T}{|\nabla u|_\epsilon}\right)\nabla & 0 \\ -\beta \text{div} & A & -I \\ 0 & \frac{\partial F_3}{\partial u} & \frac{\partial F_3}{\partial \lambda} \end{bmatrix} \begin{bmatrix} \delta p \\ \delta u \\ \delta \lambda \end{bmatrix} = - \begin{bmatrix} F_1(p,u,\lambda) \\ F_2(p,u,\lambda) \\ F_3(p,u,\lambda) \end{bmatrix}. \tag{A.17}$$

Here, $|\nabla u|_\epsilon$ denotes a diagonal matrix such that $(|\nabla u|_\epsilon \delta p)_{i,j} = |(\nabla u)_{i,j}|_\epsilon (\delta p)_{i,j}$ and $\frac{p(\nabla u)^T}{|\nabla u|_\epsilon}\nabla$ denotes a matrix such that $(\frac{p(\nabla u)^T}{|\nabla u|_\epsilon}\nabla \delta u)_{i,j} = \frac{1}{|(\nabla u)_{i,j}|_\epsilon} p_{i,j} (\nabla u)^T_{i,j} (\nabla \delta u)_{i,j}$. The function $F_3$ is strongly semi-smooth and the derivatives $\frac{\partial F_3}{\partial u}$ and $\frac{\partial F_3}{\partial \lambda}$ are defined in the sense of slant differentiability which is a generalized derivative. Motivated by the PDAS algorithm in [2] and the relationship of the semi-smooth Newton's method to the PDAS method given there, we let

$$\mathcal{I}^k := \{i : \lambda_i^k - c u_i^k \leq 0\},$$

and

$$\mathcal{A}^k := \{i : \lambda_i^k - c u_i^k > 0\}.$$

These define the (predicted) complementary inactive and active sets respectively after the $k$-th Newton step. Let $D_\mathcal{I}$ and $D_\mathcal{A}$ be the down-sampling matrix with relation to the inactive set $\mathcal{I}$ and active set $\mathcal{A}$ respectively. To extract the components of $u$ in $\mathcal{I}$ and $\mathcal{A}$, we compute $u_\mathcal{I} = D_\mathcal{I} u$ and $u_\mathcal{A} = D_\mathcal{A} u$ respectively. The components $\lambda_\mathcal{I}$ and $\lambda_\mathcal{A}$ can be obtained similarly. Here, all the superscripts $^k$ are dropped for convenience. Let

$$\begin{aligned} A_{\mathcal{I}} &:= D_{\mathcal{I}} A D_{\mathcal{I}}^T, \\ A_{\mathcal{A}} &:= D_{\mathcal{A}} A D_{\mathcal{A}}^T, \\ A_{\mathcal{I}\mathcal{A}} &:= D_{\mathcal{I}} A D_{\mathcal{A}}^T, \\ A_{\mathcal{A}\mathcal{I}} &:= D_{\mathcal{A}} A D_{\mathcal{I}}^T. \end{aligned}$$

Using the active and inactive sets, the semi-smooth Newton step is given by

$$\begin{bmatrix} |\nabla u|_\epsilon & -\left(I - \frac{p(\nabla u)^T}{|\nabla u|_\epsilon}\right)\nabla D_{\mathcal{I}}^T & -\left(I - \frac{p(\nabla u)^T}{|\nabla u|_\epsilon}\right)\nabla D_{\mathcal{A}}^T & 0 & 0 \\ -\beta D_{\mathcal{I}}\mathrm{div} & A_{\mathcal{I}} & A_{\mathcal{I}\mathcal{A}} & -I & 0 \\ -\beta D_{\mathcal{A}}\mathrm{div} & A_{\mathcal{A}\mathcal{I}} & A_{\mathcal{A}} & 0 & -I \\ 0 & 0 & 0 & I & 0 \\ 0 & 0 & cI & 0 & 0 \end{bmatrix} \times$$

$$\begin{bmatrix} \delta p \\ \delta u_{\mathcal{I}} \\ \delta u_{\mathcal{A}} \\ \delta \lambda_{\mathcal{I}} \\ \delta \lambda_{\mathcal{A}} \end{bmatrix} = - \begin{bmatrix} F_1(p,u,\lambda) \\ D_{\mathcal{I}} F_2(p,u,\lambda) \\ D_{\mathcal{A}} F_2(p,u,\lambda) \\ D_{\mathcal{I}} F_3(p,u,\lambda) \\ D_{\mathcal{A}} F_3(p,u,\lambda) \end{bmatrix}.$$

Here, $\delta u_{\mathcal{I}} = D_{\mathcal{I}} \delta u$, etc. The fourth equation above reads

$$\delta \lambda_{\mathcal{I}} = -D_{\mathcal{I}}(\lambda - 0) = -\lambda_{\mathcal{I}},$$

which implies

$$\lambda_{\mathcal{I}}^{k+1} = \lambda_{\mathcal{I}}^k + \delta\lambda_{\mathcal{I}} = 0.$$

The fifth equation of the system reads

$$c\delta u_{\mathcal{A}} = -D_{\mathcal{A}}(\lambda^k - \lambda^k + cu^k) = -cu_{\mathcal{A}}^k,$$

so that

$$u_{\mathcal{A}}^{k+1} = u_{\mathcal{A}}^k + \delta u_{\mathcal{A}} = 0.$$

This implies that elements in $\mathcal{A}^k = \{i : \lambda_i^k - cu_i^k > 0\}$ are predicted as active variables in the $(k+1)$-st iteration. By setting $\delta u_{\mathcal{A}} = -u_{\mathcal{A}}^k$ and $\delta\lambda_{\mathcal{I}} = -\lambda_{\mathcal{I}}^k$, the Newton's update can be reduced to

$$\begin{bmatrix} |\nabla u|_\epsilon & -\left(I - \frac{p(\nabla u)^T}{|\nabla u|_\epsilon}\right)\nabla D_{\mathcal{I}}^T & 0 \\ -\beta D_{\mathcal{I}}\text{div} & A_{\mathcal{I}} & 0 \\ -\beta D_{\mathcal{A}}\text{div} & A_{\mathcal{A}\mathcal{I}} & -I \end{bmatrix} \begin{bmatrix} \delta p \\ \delta u_{\mathcal{I}} \\ \delta\lambda_{\mathcal{A}} \end{bmatrix}$$
$$= -\begin{bmatrix} F_1(p,u,\lambda) \\ D_{\mathcal{I}}F_2(p,u,\lambda) \\ D_{\mathcal{A}}F_2(p,u,\lambda) \end{bmatrix} + \begin{bmatrix} -\left(I - \frac{p(\nabla u)^T}{|\nabla u|_\epsilon}\right)\nabla D_{\mathcal{A}}^T u_{\mathcal{A}} \\ A_{\mathcal{I}\mathcal{A}}u_{\mathcal{A}} - \lambda_{\mathcal{I}} \\ A_{\mathcal{A}}u_{\mathcal{A}} \end{bmatrix}.$$

It is obvious that $\delta\lambda_{\mathcal{A}}$ can be expressed in terms of $\delta p$ and $\delta u_{\mathcal{I}}$:

$$\delta\lambda_{\mathcal{A}} = -\beta D_{\mathcal{A}}\text{div}\delta p + A_{\mathcal{A}\mathcal{I}}\delta u_{\mathcal{I}} + D_{\mathcal{A}}F_2(p,u,\lambda) - A_{\mathcal{A}}u_{\mathcal{A}}. \tag{A.18}$$

Therefore, the Newton's update can be further simplified to:

$$\begin{bmatrix} |\nabla u|_\epsilon & -\left(I - \frac{p(\nabla u)^T}{|\nabla u|_\epsilon}\right)\nabla D_{\mathcal{I}}^T \\ -\beta D_{\mathcal{I}}\text{div} & A_{\mathcal{I}} \end{bmatrix} \begin{bmatrix} \delta p \\ \delta u_{\mathcal{I}} \end{bmatrix}$$
$$= \quad -\begin{bmatrix} F_1(p,u,\lambda) \\ D_{\mathcal{I}}F_2(p,u,\lambda) \end{bmatrix} + \begin{bmatrix} -\left(I - \frac{p(\nabla u)^T}{|\nabla u|_\epsilon}\right)\nabla D_{\mathcal{A}}^T u_{\mathcal{A}} \\ A_{\mathcal{I}\mathcal{A}}u_{\mathcal{A}} - \lambda_{\mathcal{I}} \end{bmatrix}.$$

Note that the $(1,1)$-block $|\nabla u|_\epsilon$ is a diagonal matrix. Hence, we can express $\delta p$ in terms of $\delta u_{\mathcal{I}}$:

$$\delta p = \frac{1}{|\nabla u|_\epsilon}\left[\left(I - \frac{p(\nabla u)^T}{|\nabla u|_\epsilon}\right)\nabla(D_{\mathcal{I}}^T\delta u_{\mathcal{I}} - D_{\mathcal{A}}^T u_{\mathcal{A}}) - F_1(p,u,\lambda)\right]. \tag{A.19}$$

Eliminating $\delta p$ from the Newton's update, we get

$$\left[-\beta D_{\mathcal{I}}\text{div}\frac{1}{|\nabla u|_\epsilon}\left(I - \frac{p(\nabla u)^T}{|\nabla u|_\epsilon}\right)\nabla D_{\mathcal{I}}^T + A_{\mathcal{I}}\right]\delta u_{\mathcal{I}} = g(p,u,\lambda), \tag{A.20}$$

where

$$\begin{aligned} g(p,u,\lambda) \quad = \quad & -D_{\mathcal{I}}F_2(p,u,\lambda) + A_{\mathcal{I}\mathcal{A}}u_{\mathcal{A}} - \lambda_{\mathcal{I}} \\ & -\beta D_{\mathcal{I}}\text{div}\frac{1}{|\nabla u|_\epsilon}\left[\left(I - \frac{p(\nabla u)^T}{|\nabla u|_\epsilon}\right)\nabla D_{\mathcal{A}}^T u_{\mathcal{A}} + F_1(p,u,\lambda)\right]. \end{aligned}$$

The linear system (A.20) is non-symmetric and so PCG cannot be applied. Chan *et al.* [1] proposed in their paper to symmetrize the matrix so that PCG could be applied. Following this suggestion, we symmetrize the system as:

$$D_{\mathcal{I}}\left[-\beta\text{div}\frac{1}{|\nabla u|_\epsilon}\left(I - \frac{p(\nabla u)^T + (\nabla u)p^T}{2|\nabla u|_\epsilon}\right)\nabla + A\right]D_{\mathcal{I}}^T\delta u_{\mathcal{I}} = g(p,u,\lambda). \tag{A.21}$$

It can be shown that the system Eq. (A.21) is symmetric positive definite when $|p_{i,j}| \leq 1$ for all $i, j$. We remark that using the relationship between the PDAS and semi-smooth Newton's method allows us to simplify the original linear system Eq. (A.17) into the much smaller linear system Eq. (A.21). The simpler structure of the reduced system also facilitates the construction of effective preconditioners.

# Appendix B

# Default Parameters for Given Data

For the 3 algorithms whose performance we have compared, i.e. NNCGM, PN and AM, we used the parameters shown in Tables B.1, B.2 and B.3. We used the same values for all results shown in this work, except for the results for extremely small values of $\epsilon$, Fig. 3.11. In this case, the bandwidth of the FBIP preconditioner was increased from the default value of 2 to 3 and 4, respectively to overcome the highly ill-conditioning of the linear system (A.21).

Table B.1: Default Parameter Values (NNCGM)

| Parameter | Value |
|---|---|
| $\rho$ | 0.99 |
| PCG tolerance | $10^{-1}$ |
| FBIP Bandwidth (symmetric) | 2 |
| $c$ | $10^4$ |
| Outer loop tol. (KKT) | $10^{-6}$ |
| $\epsilon$ (reg.) | $10^{-2}$ |
| Max. $\sharp$ Outer Iterations | 300 |
| Max. $\sharp$ Line Search ($p$) | 40 |
| Max. $\sharp$ CG Iterations Per Inner Loop | 200 |
| $\alpha$ | 0 |
| Initial guess for $u$ | $u^0 = max(f, 0)$ |
| Initial guess for $p$ | $p_{i,j} = 0 \;\; \forall \; i, j$ |
| Initial guess for $\lambda$ | $\lambda_{i,j} = 0 \;\; \forall \; i, j$ |

Table B.2: Default Parameter Values (AM)

| Parameter | Value |
|---|---|
| PCG tolerance | $10^{-1}$ |
| Outer loop tol. (KKT) | $10^{-6}$ |
| Max. $\sharp$ Outer Iterations | 300 |
| Max. $\sharp$ CG Iterations Per Inner Loop | 200 |
| $\alpha$ | 0.008 |
| Initial guess for $u$ | $u^0 = max(f, 0)$ |
| Initial guess for $p$ | $p_{i,j} = 0 \;\; \forall \; i, j$ |
| Initial guess for $\lambda$ | $\lambda_{i,j} = 0 \;\; \forall \; i, j$ |

Table B.3: Default Parameter Values (PN)

| Parameter | Value |
|---|---|
| PCG tolerance | $10^{-1}$ |
| Outer loop tol. (KKT) | $10^{-6}$ |
| $\epsilon$ (reg.) | $10^{-2}$ |
| Max. $\sharp$ Outer Iterations | 300 |
| Max. $\sharp$ Line Search | 30 |
| Max. $\sharp$ CG Iterations Per Inner Loop | 200 |
| $\alpha$ | 0 |
| Newton step tolerance | $10^{-6}$ |
| Line search reduction fraction | 0.25 |
| Initial guess for $u$ | $u^0 = max(f, 0)$ |