Inverse Problems **23** (2007) 1611–1633 doi:10.1088/0266-5611/23/4/014

# Local regularization for *n*-dimensional integral equations with applications to image processing

**Changjun Cui**[1], **Patricia K Lamm**[2] **and Thomas L Scofield**[3]

[1] Deloitte & Touche LLP, Suite 600, 600 Renaissance Center, Detroit, MI 48243-1895, USA
[2] Department of Mathematics, Michigan State University, E Lansing, MI 48824-1027, USA
[3] Department of Mathematics and Statistics, Calvin College, Grand Rapids, MI 49546, USA

E-mail: ccui@deloitte.com, lamm@math.msu.edu and scofield@calvin.edu

Received 27 February 2007, in final form 31 May 2007
Published 6 July 2007
Online at stacks.iop.org/IP/23/1611

**Abstract**

We examine the method of local regularization for the solution of linear first-kind integral equations on $\mathbb{R}^n$. We provide a theoretical analysis of the method and prove that the regularized solutions converge to the true solution as the level of error in perturbed data goes to zero. We also develop an iterative numerical algorithm based on this theory and describe its implementation. Our testing with a number of examples shows that local regularization tends to perform better than a classical method we call Tik-CG (a method based on a conjugate gradient algorithm with stopping criteria applied to standard Tikhonov regularization) when performance is measured in terms of relative error in solutions and/or in perceived sharpness of images. Unfortunately, this improvement can come at a cost as testing shows that the local regularization algorithm tends to be slower than the Tik-CG approach when applied to 2D images. As we illustrate with our numerical results, however, a compromise can be found by using the converged Tik-CG image as the starting value for the iterative local regularization method.

## 1. Introduction

In this paper we consider the problem of finding $u$ solving the equation

$$\mathcal{A}u = f \tag{1.1}$$

where for $\Omega = [0,\ 1]^n \subset \mathbb{R}^n$, $\mathcal{A}$ is the bounded linear operator on $L^2(\Omega)$ given by

$$\mathcal{A}u(\boldsymbol{x}) = \int_\Omega k(\boldsymbol{x}, \boldsymbol{s})u(\boldsymbol{s})\,\mathrm{d}\boldsymbol{s}, \qquad \text{a.a. } \boldsymbol{x} \in \Omega, \tag{1.2}$$

with $k \in L^\infty(\Omega \times \Omega)$ satisfying

$$|k(\boldsymbol{x}, \boldsymbol{s}) - k(\boldsymbol{y}, \boldsymbol{s})| \leqslant L_k(\boldsymbol{s})\|\boldsymbol{x} - \boldsymbol{y}\|^{\mu_k} \qquad \text{a.a. } \boldsymbol{x}, \boldsymbol{y}, \boldsymbol{s} \in \Omega, \tag{1.3}$$

  

for $\mu_k > 0$ and $L_k \in L^2(\Omega)$. Here $\|\cdot\|$ denotes the Euclidean norm in $\mathbb{R}^n$. We assume that equation (1.1) has solutions and let $\bar{u} \in L^2(\Omega)$ denote its (unique) minimum norm solution. Clearly $f \in L^\infty(\Omega)$; we will further assume that $\bar{u} \in L^\infty(\Omega)$.

Equation (1.1) can be used as a mathematical model of an inverse problem, i.e., given data $f$ which represent a desired or an observed effect in real world problems, determine the cause which is represented by a physically relevant solution (often a generalized solution) $\bar{u}$ of equation (1.1).

Although a number of important inverse problems are covered by our theory, here we will be particularly interested in the *blurred image reconstruction problem* (see, e.g., [2, 5], and the references therein). That is, let $\bar{u}(\boldsymbol{x})$ represent grey-level values of an image at location $\boldsymbol{x} = (x_1, x_2) \in \Omega$, where $\Omega = [0, 1] \times [0, 1] \subset \mathbb{R}^2$. One model of image blurring is given by equation (1.1) with operator $\mathcal{A}$ in (1.2) defined using the Gaussian (convolution) kernel

$$k(\boldsymbol{x}, \boldsymbol{y}) = \frac{\beta}{\pi} \mathrm{e}^{-\beta\|\boldsymbol{x}-\boldsymbol{y}\|^2}, \tag{1.4}$$

for $\beta > 0$ a blurring parameter. The image reconstruction problem then is to approximate $\bar{u}$ (the exact image) solving (1.1) given an estimate of $\beta$ and data $f$, where $f$ denotes the blurred image grey level subject to observational and measurement errors.

For $\mathcal{A}$ with a non-closed range, problem (1.1) is an ill-posed problem due to the fact that $\mathcal{A}^{-1}$ (or the generalized inverse $\mathcal{A}^+$) is unbounded [13]. In a practical setting this means that in the usual situation where our data are imprecisely measured, the approximation $\bar{u}^\delta = \mathcal{A}^{-1} f^\delta$ (or $\bar{u}^\delta = \mathcal{A}^+ f^\delta$) may be far from the desired solution $\bar{u}$ even if the perturbed data $f^\delta$ satisfy $\|f - f^\delta\| \leqslant \delta$, for $\delta > 0$ small. Regularization methods overcome this difficulty by providing an alternate method for the construction of $\bar{u}^\delta$ from perturbed data $f^\delta$, ensuring that $\bar{u}^\delta \to \bar{u}$ as the noise level $\delta \to 0$ and that $\bar{u}^\delta$ depends continuously on data.

Classical regularization techniques such as Tikhonov regularization can produce an overly smooth solution in situations where sharp or discontinuous features in the true solution are precisely those of interest (such as occurs in the image reconstruction problem). Alternative regularization methods, e.g., bounded variation regularization [1, 4, 11, 12, 25], have been proposed to improve upon Tikhonov regularization in such situations; however these methods come with their own drawbacks. One problem is that the methods often are derived by replacing the usual Tikhonov penalty term with a nondifferentiable penalty function. This leads to nontrivial difficulties in practical implementation because standard optimization packages cannot be used. Thus regularization methods of this type can be costly and associated with slow speeds of convergence. Other approaches add a (nonquadratic) penalty term in an appropriate norm in a Besov space, and couple the analysis with wavelet expansions [9, 10].

An alternative approach is the method of *local regularization*, a differentiable optimization-based method (as will be seen in the next section). We will briefly motivate this idea by defining a local regularization parameter $\boldsymbol{r} = (r_1, r_2, \ldots, r_n)$ where $r_i \in \left(0, \frac{1}{2}\right)$, and by considering a translated version of equation (1.1),

$$\mathcal{A}\bar{u}(\boldsymbol{x} + \boldsymbol{\rho}) = f(\boldsymbol{x} + \boldsymbol{\rho}), \tag{1.5}$$

for a.a. $\boldsymbol{x}$, $\boldsymbol{\rho}$ satisfying

$$\boldsymbol{\rho} = (\rho_i)_{i=1}^n, \quad \rho_i \in [-r_i, r_i] \qquad \text{and} \qquad \boldsymbol{x} = (x_i)_{i=1}^n, \quad x_i \in [r_i, 1 - r_i]. \tag{1.6}$$

That is, for a.a. $\boldsymbol{x}$, $\boldsymbol{\rho}$ satisfying (1.6), we have

$$\int_{B(\boldsymbol{x},\boldsymbol{r})} k(\boldsymbol{x} + \boldsymbol{\rho}, \boldsymbol{s})\bar{u}(\boldsymbol{s})\,\mathrm{d}\boldsymbol{s} + \int_{\Omega\setminus B(\boldsymbol{x},\boldsymbol{r})} k(\boldsymbol{x} + \boldsymbol{\rho}, \boldsymbol{s})\bar{u}(\boldsymbol{s})\,\mathrm{d}\boldsymbol{s} = f(\boldsymbol{x} + \boldsymbol{\rho}), \tag{1.7}$$

where $B(\boldsymbol{x}, \boldsymbol{r}) \subset \Omega$ is the rectangular cube centred at $\boldsymbol{x}$ defined by

$$B(\boldsymbol{x}, \boldsymbol{r}) = \left\{ \boldsymbol{y} = (y_i)_{i=1}^n \in \mathbb{R}^n \,|\, |y_i - x_i| \leqslant r_i \right\}.$$

Making a change of integration variable in the first term in (1.7) we obtain, for a.a. $\boldsymbol{x}, \boldsymbol{\rho}$ satisfying (1.6),

$$\int_{B(\mathbf{0},\boldsymbol{r})} k(\boldsymbol{x}+\boldsymbol{\rho}, \boldsymbol{x}+\boldsymbol{s})\bar{u}(\boldsymbol{x}+\boldsymbol{s})\,\mathrm{d}\boldsymbol{s} + \int_{\Omega\setminus B(\boldsymbol{x},\boldsymbol{r})} k(\boldsymbol{x}+\boldsymbol{\rho}, \boldsymbol{s})\bar{u}(\boldsymbol{s})\,\mathrm{d}\boldsymbol{s} = f(\boldsymbol{x}+\boldsymbol{\rho}), \tag{1.8}$$

where we note that each term on the left-hand side of equation (1.8) represents the action of the operator $\mathcal{A}$ on the $\boldsymbol{x}$-dependent 'local part' and the 'global part' of $\bar{u}$, respectively. This idea suggests a decomposition of the operator $\mathcal{A}$ into 'global' and 'local' parts, for each $\boldsymbol{x} \in \Omega$. This splitting of $\mathcal{A}$ is the basis of the local regularization method and allows us to apply specific regularization strategies using only the local part of $\mathcal{A}$.

It is worth noting that, in addition to the possibility of a better reconstruction of localized or sharp features in the true solution, a variation of the local regularization method motivated above leads to a fast *sequential* numerical algorithm when applied to a Volterra equation such as

$$\int_0^t k(t,s)u(s)\,\mathrm{d}s = f(t), \qquad \text{a.a. } t \in (0,1).$$

Indeed local regularization is well-known for its ability to retain the causal structure of the original Volterra problem, in contrast to classical methods such as Tikhonov regularization [7, 8, 17–19, 21–24].

The subject of this paper is the solution of the non-Volterra problem (1.1), and in this case an *iterative* local regularization procedure is needed rather than a sequential method. Intuitively the procedure works like this: an initial guess is made for the solution on the entire domain, then the solution is iteratively updated on small subdomains by solving a local regularization problem while assuming the solution off the small subdomain (i.e. the global part) is fixed at its previously set value.

Such an iterative local regularization method was first introduced in [20] for one-dimensional non-Volterra problems, although with a slightly different structure. In [20] the translated equation (1.5) was defined for *all* $\boldsymbol{x}$ in the interior of $\Omega$; however, in order for the resulting equation to make sense, the value of $\boldsymbol{r}$ had to depend on $\boldsymbol{x}$ with $\boldsymbol{r}$ decreasing as $\boldsymbol{x}$ approached the boundary of $\Omega$. While such an approach does allow for a variable regularization parameter throughout the entire domain $\Omega$ (which has some benefits), this idea does not easily extend to $n$-dimensional problems because of difficulties at corners. Instead, we have taken a different approach here with a constant regularization parameter $\boldsymbol{r}$ associated with a given fixed size of the local regularization region; we have allowed for variable local regularization through the use of a second (variable) regularization parameter $\alpha$ which controls the amount of regularization being imposed on each local region.

Although the underlying structure of the theory we introduce here differs from that of [20], some of the convergence arguments carry over with little change. To abbreviate our presentation we will outline the main points of the construction and convergence theory in sections 2 and 3 below, referring the reader to [6] for complete details. Finally, in sections 4 and 5 we illustrate how numerical methods suggested from the theory can be used to solve some sample 2D image reconstruction problems. We compare our findings to a classical iterative regularization method (i.e., the conjugate gradient method with stopping criteria, applied to classical Tikhonov regularization). For 2D problems we find that the local regularization method is slower than the classical method but that in many cases it does a better job of resolving sharp features in images.

## 2. Basic definitions

We will make clear the local regularization ideas motivated above by defining the required spaces, operators and regularization parameters.

We will let $\boldsymbol{r} = (r_i)_{i=1}^n \in \mathbb{R}^n$, for $r_i \in \left(0, \frac{1}{2}\right)$, denote the regularization parameter associated with the size of the local regularization region, with

$$\|\boldsymbol{r}\| \equiv \max_{i=1,2,\ldots,n} r_i,$$

and make the definitions

$$[\boldsymbol{r}, \mathbf{1} - \boldsymbol{r}] = \{\boldsymbol{x} \in \mathbb{R}^n | r_i \leqslant x_i \leqslant 1 - r_i, \text{ for } i = 1, 2, \ldots, n\},$$
$$[-\boldsymbol{r}, \boldsymbol{r}] = \{\boldsymbol{\rho} \in \mathbb{R}^n | - r_i \leqslant \rho_i \leqslant r_i, \text{ for } i = 1, 2, \ldots, n\},$$

and define the $\boldsymbol{r}$-dependent product space $\boldsymbol{\Omega}_{\boldsymbol{r}} \subset \Omega \times \left(-\frac{1}{2}, \frac{1}{2}\right)^n$ via

$$\boldsymbol{\Omega}_{\boldsymbol{r}} = \{(\boldsymbol{x}, \boldsymbol{\rho}) | \boldsymbol{x} \in [\boldsymbol{r}, \mathbf{1} - \boldsymbol{r}], \boldsymbol{\rho} \in [-\boldsymbol{r}, \boldsymbol{r}]\}.$$

We will also use $\alpha$ to designate a second regularization parameter,

$$\alpha \in \Lambda \equiv \{\alpha \in L^\infty(\Omega) | \alpha_{\min} \equiv \inf_{\boldsymbol{x} \in \Omega} \alpha(\boldsymbol{x}) > 0\}.$$

This parameter will control the amount of regularization applied in local regularization regions, with the dependence on $\boldsymbol{x}$ allowing for the possibility of more or less regularization in different parts of the domain $\Omega$.

In order to better handle the translated variables $f$ and $\bar{u}$ in equations (1.5) and (1.8) we make the following definitions. Let

$$F_{\boldsymbol{r}}(\boldsymbol{x})(\boldsymbol{\rho}) = f(\boldsymbol{x} + \boldsymbol{\rho}), \qquad U_{\boldsymbol{r}}(\boldsymbol{x})(\boldsymbol{\rho}) = \bar{u}(\boldsymbol{x} + \boldsymbol{\rho}), \qquad \text{a.a. } (\mathbf{x}, \boldsymbol{\rho}) \in \boldsymbol{\Omega}_{\boldsymbol{r}}.$$

Then for $f, \bar{u} \in L^\infty(\Omega)$, it follows that $F_{\boldsymbol{r}}, U_{\boldsymbol{r}} \in \mathcal{X}_{\boldsymbol{r}}$, where

$$\mathcal{X}_{\boldsymbol{r}} = L^2((\boldsymbol{r}, \mathbf{1} - \boldsymbol{r}); L^2(-\boldsymbol{r}, \boldsymbol{r}))$$

is a Hilbert space with the weighted norm

$$\|\varphi\|_{\boldsymbol{r}}^2 \equiv \frac{1}{2^n r_1 r_2 \cdots r_n} \int_{[\boldsymbol{r}, \mathbf{1}-\boldsymbol{r}]} \int_{[-\boldsymbol{r}, \boldsymbol{r}]} |\varphi(\boldsymbol{x})(\boldsymbol{\rho})|^2 \, \mathrm{d}\boldsymbol{\rho} \, \mathrm{d}\boldsymbol{x},$$

$\varphi \in \mathcal{X}_{\boldsymbol{r}}$, and associated inner product. Given $\alpha \in \Lambda$ we will sometimes use an equivalent norm for $\varphi \in \mathcal{X}_{\boldsymbol{r}}$,

$$\|\varphi\|_{\boldsymbol{r},\alpha}^2 \equiv \frac{1}{2^n r_1 r_2 \cdots r_n} \int_{[\boldsymbol{r}, \mathbf{1}-\boldsymbol{r}]} \alpha(\boldsymbol{x}) \int_{[-\boldsymbol{r}, \boldsymbol{r}]} |\varphi(\boldsymbol{x})(\boldsymbol{\rho})|^2 \, \mathrm{d}\boldsymbol{\rho} \, \mathrm{d}\boldsymbol{x}.$$

It will also be useful to define $\bar{F}_{\boldsymbol{r}}, \bar{U}_{\boldsymbol{r}} \in \mathcal{X}_{\boldsymbol{r}}$ via

$$\bar{F}_{\boldsymbol{r}}(\boldsymbol{x})(\boldsymbol{\rho}) = f(\boldsymbol{x}), \qquad \bar{U}_{\boldsymbol{r}}(\boldsymbol{x})(\boldsymbol{\rho}) = \bar{u}(\boldsymbol{x}), \qquad \text{a.a. } (\mathbf{x}, \boldsymbol{\rho}) \in \boldsymbol{\Omega}_{\boldsymbol{r}},$$

and to define $F_{\boldsymbol{r}}^\delta$ and $\bar{F}_{\boldsymbol{r}}^\delta \in \mathcal{X}_{\boldsymbol{r}}$ in the expected way from $f^\delta$. It is not hard to show that

$$\|U_{\boldsymbol{r}}\| \leqslant \|\bar{u}\|_\infty, \qquad \|\bar{U}_{\boldsymbol{r}}\| \leqslant \|\bar{u}\|_\infty,$$

with similar bounds for the other quantities defined above [6].

Eventually we will be looking at the behaviour of regularized solutions as the regularization parameter $\boldsymbol{r}$ approaches $\mathbf{0}$, a fact which makes the use of parameter-dependent spaces $\mathcal{X}_{\boldsymbol{r}}$ somewhat problematic. It will be convenient to have a reference (parameter-free) space, $\mathcal{X} \equiv L^2(\Omega; X)$, to which these spaces may be mapped. Here $X = L^2((-\Delta, \Delta)^n)$ for some fixed $\Delta > 0$, with usual norm $|\cdot|_X$ and usual inner product $\langle \cdot, \cdot \rangle_X$; throughout we will simplify computations by taking $\Delta = 1$. The norm on $\mathcal{X}$ is the expected one,

$$\|\tilde{\varphi}\|_{\mathcal{X}}^2 \equiv \int_\Omega |\tilde{\varphi}(\boldsymbol{x})|_X^2 \, \mathrm{d}\boldsymbol{x} = \int_\Omega \int_{(-\Delta, \Delta)^n} |\tilde{\varphi}(\boldsymbol{x})(\boldsymbol{\rho})|^2 \, \mathrm{d}\boldsymbol{\rho} \, \mathrm{d}\boldsymbol{x},$$

for $\tilde{\varphi} \in \mathcal{X}$, with the associated inner product $\langle \cdot, \cdot \rangle_{\mathcal{X}}$.

To move from the parameter-dependent space $\mathcal{X}_r$ to $\mathcal{X}$ we will use the mapping $E_r$, where for $\varphi \in \mathcal{X}_r$, $\boldsymbol{\rho} \in [-\Delta, \Delta]^n$ and $\boldsymbol{x} \in \Omega$, we have $E_r\varphi \in \mathcal{X}$ with

$$E_r\varphi(\boldsymbol{x})(\boldsymbol{\rho}) \equiv \begin{cases} \varphi(\boldsymbol{x})(\rho_1 r_1, \rho_2 r_2, \ldots, \rho_n r_n), & \boldsymbol{x} \in (\boldsymbol{r}, \mathbf{1} - \boldsymbol{r}),\ \boldsymbol{\rho} \in (-\Delta, \Delta)^n, \\ 0, & \text{otherwise.} \end{cases}$$

It follows that $\|E_r\varphi\|_{\mathcal{X}}^2 = 2^n\|\varphi\|_r^2$, for $\varphi \in \mathcal{X}_r$.

The decomposition of the operator $\mathcal{A}$ into $\boldsymbol{x}$-dependent 'local' and 'global' parts is accomplished in part by the operators $A_r$ and $B_r$, respectively. We define $A_r : \mathcal{X}_r \mapsto \mathcal{X}_r$ and $B_r : L^2(\Omega) \mapsto \mathcal{X}_r$ by

$$A_r\varphi(\boldsymbol{x})(\boldsymbol{\rho}) = \int_{[-\boldsymbol{r},\boldsymbol{r}]} k(\boldsymbol{x}+\boldsymbol{\rho}, \boldsymbol{x}+\boldsymbol{s})\varphi(\boldsymbol{x})(\boldsymbol{s})\,\mathrm{d}\boldsymbol{s}, \qquad \text{a.a. } (\mathbf{x}, \boldsymbol{\rho}) \in \boldsymbol{\Omega}_r,$$

$$B_r\eta(\boldsymbol{x})(\boldsymbol{\rho}) = \int_{\Omega\setminus[\boldsymbol{x}-\boldsymbol{r},\boldsymbol{x}+\boldsymbol{r}]} k(\boldsymbol{x}+\boldsymbol{\rho}, \boldsymbol{s})\eta(\boldsymbol{s})\,\mathrm{d}\boldsymbol{s}, \qquad \text{a.a. } (\mathbf{x}, \boldsymbol{\rho}) \in \boldsymbol{\Omega}_r,$$

for $\varphi \in \mathcal{X}_r$ and $\eta \in \boldsymbol{L}^2(\Omega)$. Both operators are bounded linear, with operator norms satisfying

$$\|A_r\| \leqslant 2^n r_1 r_2 \cdots r_n \|k\|_\infty, \qquad \|B_r\| \leqslant \|k\|_\infty,$$

respectively [6]. To complete the decomposition of $\mathcal{A}$ we define $C_r : \mathcal{X}_r \mapsto \mathcal{X}_r$ via

$$C_r \equiv A_r + B_r T_r, \tag{2.1}$$

where the operator $T_r$ assumes the role of mapping quantities in $\mathcal{X}_r$ into variables defined on the original domain $\Omega$. An example of the operator $T_r$ particularly well-suited for numerical computations may be found in section 4. (See, for example, equation (4.3) and the discussion following it.) To precisely define $T_r$ here, we let $\ell \in X^*$ be fixed and normalized so that $\ell(\mathbf{1}) = 1$, where $\mathbf{1} \in X$ is given by $\mathbf{1}(\boldsymbol{\rho}) = 1$ for a.a. $\boldsymbol{\rho} \in (-\Delta, \Delta)^n$. We will denote by $\gamma_\ell$ that unique nonzero element of $X$ satisfying

$$\ell(v) = \langle v, \gamma_\ell \rangle_X, \qquad v \in X,$$

where the construction of $\ell$ gives $\int_{[-\Delta,\Delta]^n} \gamma_\ell(\rho)\,\mathrm{d}\rho = 1$. We then define $T \in \mathcal{L}(\mathcal{X}, L^2(\Omega))$ for $\tilde{\varphi} \in \mathcal{X}$ by

$$T\tilde{\varphi}(\boldsymbol{x}) \equiv \ell(\tilde{\varphi}(\boldsymbol{x})), \qquad \text{a.a. } \boldsymbol{x} \in \Omega,$$

and $T_r \in \mathcal{L}(\mathcal{X}_r, L^2(\Omega))$ via

$$T_r \equiv T E_r,$$

where the operator norms of $T_r$ and $T$ satisfy $\|T_r\| \leqslant \sqrt{2^n}\|T\|$. It is worth noting that for $\boldsymbol{x} \in \Omega$,

$$T_r\bar{U}_r(\boldsymbol{x}) = \begin{cases} \bar{u}(\boldsymbol{x}), & \boldsymbol{x} \in (\boldsymbol{r}, \mathbf{1} - \boldsymbol{r}), \\ 0, & \text{otherwise,} \end{cases}$$

so that

$$\|T_r\bar{U}_r - \bar{u}\|_{L^2(\Omega)}^2 = \int_{\Omega\setminus(\boldsymbol{r},\mathbf{1}-\boldsymbol{r})} |\bar{u}(\boldsymbol{x})|^2\,\mathrm{d}\boldsymbol{x} \leqslant (3^n - 1)\|\boldsymbol{r}\|_\infty \|\bar{u}\|_\infty^2.$$

Finally, from the properties of $A_r$, $B_r$ and $T_r$, we have that $C_r$ defined by (2.1) satisfies $C_r \in \mathcal{L}(\mathcal{X}_r)$.

*2.1. The local regularization problem $\mathcal{P}^{\delta}_{\boldsymbol{r},\alpha}$*

**Definition 2.1.** *Let $f^{\delta} \in L^{\infty}(\Omega)$ be given satisfying $\|f - f^{\delta}\|_{\infty} < \delta$, and let $\boldsymbol{r}$ and $\alpha$ be defined as above. Problem $\mathcal{P}^{\delta}_{\boldsymbol{r},\alpha}$ is that of finding $\varphi^{\delta}_{\boldsymbol{r},\alpha} \in \mathcal{X}_{\boldsymbol{r}}$ such that*

$$\varphi^{\delta}_{\boldsymbol{r},\alpha} = \arg\min_{\varphi \in \mathcal{X}_{\boldsymbol{r}}} \left\{ \left\| C_{\boldsymbol{r}}\varphi - F^{\delta}_{\boldsymbol{r}} \right\|^2_{\boldsymbol{r}} + \|\varphi\|^2_{\boldsymbol{r},\alpha} \right\}.$$

The following theorem follows from the classical Tikhonov regularization theory.

**Theorem 2.1.** *Let $\boldsymbol{r}$, $\alpha$ and $f^{\delta}$ satisfy the conditions stated in definition 2.1. Then there exists a unique solution $\varphi^{\delta}_{\boldsymbol{r},\alpha} \in \mathcal{X}_{\boldsymbol{r}}$ of problem $\mathcal{P}^{\delta}_{\boldsymbol{r},\alpha}$. Both $\varphi^{\delta}_{\boldsymbol{r},\alpha}$ and $\eta^{\delta}_{\boldsymbol{r},\alpha} \equiv T_{\boldsymbol{r}}\varphi^{\delta}_{\boldsymbol{r},\alpha} \in L^2(\Omega)$ depend continuously on $F^{\delta}_{\boldsymbol{r}} \in \mathcal{X}_{\boldsymbol{r}}$ and thus on data $f^{\delta} \in L^{\infty}(\Omega)$.*

## 3. Convergence

Our main convergence result is as follows:

**Theorem 3.1.** *Let $\{\delta_k\}_{k=1}^{\infty} \subseteq \mathbb{R}^+$ with $\delta_k \to 0$ as $k \to \infty$. Let $\{\boldsymbol{r}_k\}_{k=1}^{\infty}$ and $\{\alpha_k\}_{k=1}^{\infty} \subset \Lambda$ be given such that $\|\boldsymbol{r}_k\| \in \left(0, \frac{1}{2}\right)$ and $\|\boldsymbol{r}_k\|, \|\alpha_k\|_{\infty} \to 0$ as $k \to \infty$. Assume further that there is $M > 0$ such that*

*(i)* $\delta_k^2/\alpha_{k,\min} \to 0$,
*(ii)* $\|\boldsymbol{r}_k\|^n/\delta_k \leqslant M$,
*(iii)* $\|\alpha_k\|_{\infty}/\alpha_{k,\min} \to 1$,

*as $k \to \infty$. For each $k = 1, 2, \ldots$, let $f_{\delta_k} \in L^{\infty}(\Omega)$ be given with $\|f - f^{\delta_k}\| < \delta_k$, let $\varphi^{\delta_k}_{\boldsymbol{r}_k,\alpha_k} \in \mathcal{X}_{\boldsymbol{r}_k}$ denote the solution of problem $\mathcal{P}^{\delta_k}_{\boldsymbol{r}_k,\alpha_k}$ associated with $f^{\delta_k}$, and let*

$$\eta_k \equiv T_{\boldsymbol{r}_k}\varphi^{\delta_k}_{\boldsymbol{r}_k,\alpha_k}. \tag{3.1}$$

*Then*

$$\eta_k \to \bar{u} \quad \text{in } L^2(\Omega),$$

*as $k \to \infty$, where $\bar{u}$ is the solution of the original problem (1.1).*

**Remark 3.1.** We note that, as in [20], assumptions (i) and (ii) in the main convergence theorem above correspond to similar conditions in the classical regularization theory in requiring that the regularization parameters $\boldsymbol{r}_k$ and $\alpha_k$ converge to zero at a rate relative to the level $\delta_k$ of noise in the problem. For assumption (iii) we observe that $\alpha_k$ may be spatially varying as long as there is noise in the problem (i.e., when $\delta_k > 0$); however as $\delta_k$ nears zero and $k$ is sufficiently large, $\alpha_k$ is very close to its max and min values, both of which are approaching zero as $k \to \infty$.

We will simplify notation henceforth by writing $\varphi_k \equiv \varphi^{\delta_k}_{\boldsymbol{r}_k,\alpha_k}$, $\mathcal{P}_k \equiv \mathcal{P}^{\delta_k}_{\boldsymbol{r}_k,\alpha_k}$, $\mathcal{X}_k \equiv \mathcal{X}_{\boldsymbol{r}_k}$, $F_k \equiv F_{\boldsymbol{r}_k}$, $F^{\delta}_k \equiv F^{\delta_k}_{\boldsymbol{r}_k}$, $U_k \equiv U_{\boldsymbol{r}_k}$, $\bar{U}_k \equiv \bar{U}_{\boldsymbol{r}_k}$, $E_k \equiv E_{\boldsymbol{r}_k}$, $A_k \equiv A_{\boldsymbol{r}_k}$, and so on. The proof of theorem 3.1 requires the following intermediate lemma.

**Lemma 3.1.** *Let $\{\delta_k\}_{k=1}^{\infty} \subseteq \mathbb{R}^+$ and $\{\alpha_k\}_{k=1}^{\infty} \subseteq \Lambda$. Let $\{\boldsymbol{r}_k\}_{k=1}^{\infty}$ satisfy $\|\boldsymbol{r}_k\| \in \left(0, \frac{1}{2}\right)$ and assume that there exists $M > 0$ such that*

*(i)* $\delta_k^2/\alpha_{k,\min} \leqslant M$,
*(ii)* $\|\boldsymbol{r}_k\|^n/\delta_k \leqslant M$,
*(iii)* $\|\alpha_k\|_{\infty}/\alpha_{k,\min} \leqslant M$,

*as $k \to \infty$. For each $k = 1, 2, \ldots$, let $f^{\delta_k} \in L^\infty(\Omega)$ be given with $\|f - f^{\delta_k}\|_\infty < \delta_k$, and let $\varphi_k \in \mathcal{X}_k$ denote the solution of Problem $\mathcal{P}_k$ associated with $f^{\delta_k}$, with $\eta_k \equiv T_k\varphi_k \in L^2(\Omega)$.*

*Let $\tilde{\varphi}_k \equiv E_k\varphi_k \in \mathcal{X}$. Then there is $\tilde{\varphi} \in \mathcal{X}$ and a subsequence of $\{\tilde{\varphi}_k\}$ which converges weakly in $\mathcal{X}$ to $\tilde{\varphi}$. That is, relabelling the subsequential indices,*

$$\tilde{\varphi}_k \rightharpoonup \tilde{\varphi} \text{ in } \mathcal{X} \qquad \text{as} \quad k \to \infty.$$

*In addition, $\eta \in L^2(\Omega)$ defined by*

$$\eta \equiv T\tilde{\varphi}$$

*is such that (using the same relabelling of indices as above)*

$$\eta_k \rightharpoonup \eta \quad \text{in } L^2(\Omega) \text{ as } k \to \infty.$$

*Further, if $\delta_k \to 0$, $\|\boldsymbol{r}_k\| \to 0$, and $\|\alpha_k\|_\infty \to 0$ as $k \to \infty$, then $\eta$ is a solution of $\mathcal{A}u = f$ and $\tilde{\varphi}$ is a solution of $\tilde{A}\psi = f$, for $\tilde{A} \in \mathcal{L}(\mathcal{X}, L^2(\Omega))$ defined by $\tilde{A} = \mathcal{A}T$.*

**Proof.** Making only straightforward changes [6] in lemma 2.1 in [20], we have that

$$\left\|C_k\varphi_k - F_k^\delta\right\|_{\boldsymbol{r}_k}^2 + \|\varphi_k\|_{\boldsymbol{r}_k,\alpha}^2 \leqslant C\left[\left(r_{k1}^2 r_{k2}^2 \cdots r_{kn}^2\|k\|_\infty^2 + \|\alpha_k\|_\infty\right)\|\bar{u}\|_\infty^2 + \delta_k^2\right],$$

for some $C > 0$ independent of $\boldsymbol{r}_k$, $\alpha_k$ and $\delta_k$. Then

$$\begin{aligned}\|E_k\varphi_k\|_{\mathcal{X}}^2 &\leqslant \frac{2^n}{\alpha_{k,\min}}\left(\left\|C_k\varphi_k - F_k^\delta\right\|_{\boldsymbol{r}_k}^2 + \|\varphi_k\|_{\boldsymbol{r}_k,\alpha_k}^2\right)\\ &\leqslant \frac{2^n C}{\alpha_{k,\min}}\left[\left(r_{k1}^2 r_{k2}^2 \cdots r_{kn}^2\|k\|_\infty^2 + \|\alpha_k\|_\infty\right)\|\bar{u}\|_\infty^2 + \delta_k^2\right],\end{aligned}$$

so that $\|\tilde{\varphi}_k\|_{\mathcal{X}} = \|E_k\varphi_k\|_{\mathcal{X}}$ is uniformly bounded for all $k = 1, 2, \ldots$. The statements of the lemma regarding the weak subsequential convergence of $\tilde{\varphi}_k$ and $\eta_k$ follow from the fact that $\mathcal{X}$ is a Hilbert space and from the observation that $\eta_k = T_k\varphi_k = TE_k\varphi_k = T\tilde{\varphi}_k$ for $k = 1, 2, \ldots$.

We now define $\bar{A}_k \in \mathcal{L}(L^2(\Omega), \mathcal{X}_k)$ for $k = 1, 2, \ldots$ via

$$\bar{A}_k u(\boldsymbol{x})(\boldsymbol{\rho}) = \mathcal{A}u(\boldsymbol{x}), \qquad \text{a.a. } (\mathbf{x}, \boldsymbol{\rho}) \in \boldsymbol{\Omega}_{\boldsymbol{r}_k}$$

for $\mathcal{A}$ the original operator defined in (1.2) and $u \in L^2(\Omega)$. Then

$$\begin{aligned}\|\bar{A}_k\eta - \bar{F}_k\|_{\boldsymbol{r}_k}^2 &= \frac{1}{2^n r_{k1}r_{k2}\cdots r_{kn}}\int_{[\boldsymbol{r}_k, \mathbf{1}-\boldsymbol{r}_k]}\int_{[-\boldsymbol{r}_k, \boldsymbol{r}_k]} |\mathcal{A}\eta(\boldsymbol{x}) - f(\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{\rho}\,\mathrm{d}\boldsymbol{x}\\ &\to \int_\Omega |\mathcal{A}\eta(\boldsymbol{x}) - f(\boldsymbol{x})|^2 \,\mathrm{d}\boldsymbol{x} = \|\mathcal{A}\eta - f\|_{L^2(\Omega)} \qquad \text{as} \quad k \to \infty,\end{aligned}$$

where it remains to show that $\|\bar{A}_k\eta - \bar{F}_k\|_{\boldsymbol{r}_k} \to 0$ as $k \to \infty$ in order to conclude that $\eta$ solves $\mathcal{A}u = f$. But

$$\|\bar{A}_k\eta - \bar{F}_k\|_{\boldsymbol{r}_k} \leqslant \sum_{i=1}^5 T_i^k$$

where

$$\begin{aligned}&T_1^k = \left\|A_k\varphi_k + B_kT_k\varphi_k - F_k^\delta\right\|_{\boldsymbol{r}_k}, \qquad T_2^k = \|A_k\varphi_k\|_{\boldsymbol{r}_k},\\ &T_3^k = \|\bar{A}_k\eta - B_kT_k\varphi_k\|_{\boldsymbol{r}_k}, \qquad T_4^k = \|F_k^\delta - F_k\|_{\boldsymbol{r}_k}, \qquad T_5^k = \|F_k - \bar{F}_k\|_{\boldsymbol{r}_k}.\end{aligned}$$

Arguments like those in the proof of lemma 3.2 of [20] may be used to argue $T_i^k \to 0$ as $k \to \infty$ for $i = 1, 2, 4$ and 5. In what follows we show that $T_3^k \to 0$ as $k \to \infty$. We have that

$$\begin{aligned}T_3^k &= \|\bar{A}_k\eta - B_k\eta_k\|_{\boldsymbol{r}_k}\\ &\leqslant \|\bar{A}_k(\eta - \eta_k)\|_{\boldsymbol{r}_k} + \|(\bar{A}_k - B_k)\eta_k\|_{\boldsymbol{r}_k},\end{aligned} \tag{3.2}$$

where the first term in (3.2) satisfies

$$\begin{aligned}\|\bar{A}_k(\eta-\eta_k)\|_{\boldsymbol{r}_k}^2 &= \int_{[\boldsymbol{r}_k,\mathbf{1}-\boldsymbol{r}_k]} |\mathcal{A}(\eta-\eta_k)(\boldsymbol{x})|^2\,\mathrm{d}\boldsymbol{x}\\ &\leqslant \|\mathcal{A}(\eta-\eta_k)\|_{L^2(\Omega)}^2 \to 0,\end{aligned}$$

as $k\to\infty$ from the compactness of $\mathcal{A}$ on $L^2(\Omega)$. For the second term in (3.2), we have for a.a. $(\mathbf{x},\boldsymbol{\rho})\in\boldsymbol{\Omega}_{\boldsymbol{r}_k}$,

$$\begin{aligned}|(\bar{A}_k-B_k)\eta_k(\boldsymbol{x})(\boldsymbol{\rho})|^2 &= \left|\int_\Omega k(\boldsymbol{x},\boldsymbol{s})\eta_k(\boldsymbol{s})\,\mathrm{d}\boldsymbol{s}-\int_{\Omega\setminus B(\boldsymbol{x},\boldsymbol{r}_k)} k(\boldsymbol{x}+\boldsymbol{\rho},\boldsymbol{s})\eta_k(\boldsymbol{s})\,\mathrm{d}\boldsymbol{s}\right|^2\\ &\leqslant 2\left|\int_\Omega (k(\boldsymbol{x},\boldsymbol{s})-k(\boldsymbol{x}+\boldsymbol{\rho},\boldsymbol{s}))\eta_k(\boldsymbol{s})\,\mathrm{d}\boldsymbol{s}\right|^2\\ &\quad+2\left|\int_{B(\boldsymbol{x},\boldsymbol{r}_k)} k(\boldsymbol{x}+\boldsymbol{\rho},\boldsymbol{s})\eta_k(\boldsymbol{s})\,\mathrm{d}\boldsymbol{s}\right|^2\\ &\leqslant 2\|\eta_k\|^2(\|\boldsymbol{\rho}\|^{2\mu_k}\|L_k\|^2+2^n\|k\|_\infty^2 r_{k1}r_{k2}\cdots r_{kn}),\end{aligned}$$

and thus

$$\|(\bar{A}_k-B_k)\eta_k\|_{\boldsymbol{r}_k}^2 \leqslant \|\eta_k\|^2\big(2\|L_k\|^2\|\boldsymbol{r}_k\|^{2\mu_k}+2^{n+1}r_{k1}r_{k2}\cdots r_{kn}\|k\|_\infty^2\big),$$

where $\eta_k$ is uniformly bounded since $\|\eta_k\|=\|T_k\tilde{\varphi}_k\|\leqslant\sqrt{2^n}\|T\|\|\tilde{\varphi}_k\|$. □

**Proof of theorem 3.1.** In what follows we will show that $\tilde{\varphi}=\tilde{U}$ where $\tilde{\varphi}$ is given in lemma 3.1 and $\tilde{U}\in\mathcal{X}$ is defined by

$$\tilde{U}(\boldsymbol{x})(\boldsymbol{\rho}) = \frac{\bar{u}(\boldsymbol{x})\gamma_\ell(\boldsymbol{\rho})}{|\gamma_\ell|_X^2}, \qquad \text{a.a. } \boldsymbol{\rho}\in[-\Delta,\Delta]^n, \quad \boldsymbol{x}\in\Omega. \tag{3.3}$$

Indeed, using arguments similar to those in the proof of theorem 3.1 in [20] it is not difficult to see that $\tilde{U}$ is the minimum norm solution of $\tilde{A}\psi=f$. If we can then show that $\|\tilde{\phi}\|_{\mathcal{X}}\leqslant\|\tilde{U}\|_{\mathcal{X}}$, it will follow that $\tilde{\varphi}=\tilde{U}$. In fact, since $\tilde{\phi}_k=E_k\phi_k$, it follows that

$$\begin{aligned}\|\tilde{\phi}_k\|_{\mathcal{X}}^2 &\leqslant \frac{2^n}{\alpha_{k,\min}}\big(\|C_k\phi_k-F_k^\delta\|_{\boldsymbol{r}_k}^2+\|\phi_k\|_{\boldsymbol{r}_k,\alpha_k}^2\big)\\ &\leqslant \frac{2^n}{\alpha_{k,\min}}\big(\|C_k\tilde{V}_k-F_k^\delta\|_{\boldsymbol{r}_k}^2+\|\tilde{V}_k\|_{\boldsymbol{r}_k,\alpha_k}^2\big),\end{aligned} \tag{3.4}$$

where $\tilde{V}_k\in\mathcal{X}_k$ is given for $(\boldsymbol{x},\boldsymbol{\rho})\in\Omega_{\boldsymbol{r}_k}$ by $\tilde{V}_k(\boldsymbol{x})(\boldsymbol{\rho})=\tilde{U}(\boldsymbol{x})(\rho_1/r_{k1},\rho_2/r_{k2},\ldots,\rho_n/r_{kn})$, with

$$\|\tilde{V}_k\|_{\boldsymbol{r}_k}^2 \leqslant \frac{1}{2^n|\gamma_\ell|_X^2}\int_{[\boldsymbol{r}_k,\mathbf{1}-\boldsymbol{r}_k]}\bar{u}^2(\boldsymbol{x})\,\mathrm{d}\boldsymbol{x}. \tag{3.5}$$

Thus

$$\begin{aligned}T_k\tilde{V}_k(\boldsymbol{x}) &= \int_{[-\Delta,\Delta]^n} E_k\tilde{V}_k(\boldsymbol{x})(\boldsymbol{\rho})\gamma_\ell(\boldsymbol{\rho})\,\mathrm{d}\boldsymbol{\rho}\\ &= \begin{cases}\bar{u}(\boldsymbol{x}), & \boldsymbol{x}\in(\boldsymbol{r}_k,\mathbf{1}-\boldsymbol{r}_k),\\ 0, & \text{otherwise},\end{cases}\end{aligned}$$

and

$$\begin{aligned}\|C_k\tilde{V}_k-F_k^\delta\|_{\boldsymbol{r}_k}^2 &= \|A_k\tilde{V}_k+B_kT_k\tilde{V}_k-F_k^\delta\|_{\boldsymbol{r}_k}^2\\ &\leqslant 4\|A_k(\tilde{V}_k-U_k)\|_{\boldsymbol{r}_k}^2+2\|F_k-F_k^\delta\|_{\boldsymbol{r}_k}^2\\ &\leqslant 8\|A_k\|^2\big(\|\tilde{V}_k\|_{\boldsymbol{r}_k}^2+\|U_k\|_{\boldsymbol{r}_k}^2\big)+2\delta_k^2\\ &\leqslant C\big(r_{k1}^2r_{k2}^2\cdots r_{kn}^2+\delta_k^2\big),\end{aligned}$$

for $C = C(\bar{u}, k)$ independent of $\boldsymbol{r}$; here we have used the fact that $\|A_k U_k + B_k T_k \tilde{V}_k - F_k\|_{\boldsymbol{r}_k}^2 = \|A_k U_k + B_k \bar{u} - F_k\|_{\boldsymbol{r}_k}^2 = 0$ [6]. We thus have

$$\|\tilde{\phi}\|_{\mathcal{X}}^2 \leqslant \liminf \|\tilde{\phi}_k\|_{\mathcal{X}}^2 \tag{3.6}$$

$$\begin{aligned} &\leqslant \limsup \frac{2^n}{\alpha_{k,\min}} \left[ C\left(r_{k1}^2 r_{k2}^2 \cdots r_{kn}^2 + \delta_k^2\right) + \frac{\|\alpha_k\|_\infty}{2^n |\gamma_\ell|_X^2} \int_{[\boldsymbol{r}_k, \mathbf{1}-\boldsymbol{r}_k]} \bar{u}^2(\boldsymbol{x})\, \mathrm{d}\boldsymbol{x} \right] \\ &= \|\bar{u}\|_{L^2(\Omega)}^2 / |\gamma_\ell|_X^2 \\ &= \|\tilde{U}\|_{\mathcal{X}}^2, \end{aligned} \tag{3.7}$$

under the assumptions of the theorem, so that $\|\tilde{\phi}\|_{\mathcal{X}} \leqslant \|\tilde{U}\|_{\mathcal{X}}$. By uniqueness of the minimum norm solution of $\tilde{A}\psi = f$, it follows that $\tilde{\phi} = \tilde{U}$.

All inequalities between (3.6) and (3.7) must then be equalities, so

$$\|\tilde{\phi}_k\|_{\mathcal{X}} \to \|\tilde{\phi}\|_{\mathcal{X}} \qquad \text{as} \quad k \to \infty,$$

and in fact the weak convergence of $\tilde{\phi}_k$ to $\tilde{\phi}$ is strong convergence. In addition,

$$\eta_k = T\tilde{\phi}_k \to T\tilde{\phi} = T\tilde{U} = \bar{u}, \qquad \text{as} \quad k \to \infty,$$

so that $\eta = \bar{u}$. Standard arguments can be used to extend the subsequential convergence to full sequential convergence. This completes the proof of theorem 3.1. □

## 4. Numerical implementation

In this section we develop a discrete implementation of the local regularization method presented in the previous sections, and describe several numerical examples. The algorithms have been developed in $n \geqslant 1$ dimensions; specific examples are provided in the case $n = 2$.

### *4.1. The discrete local regularization problem*

Let $N \in \mathbb{N}$ and assume $\Omega$ is divided into $N^n$ 'cells', where each cell has edges of length $h = \frac{1}{N}$. A more general mesh through $\Omega$, where $N_1$ subintervals are used along the first axis, $N_2$ along the second and so on, is also possible and does not significantly alter what follows. We define mesh points $\boldsymbol{x}^{(\boldsymbol{j})} = \left(x_1^{(j)}, \ldots, x_n^{(j)}\right) = h\boldsymbol{j}$, for each $\boldsymbol{j}$ in the set $J = \{0, 1, \ldots, N\}^n$. We further define midpoints $\hat{\boldsymbol{x}}^{(\boldsymbol{j})} = \boldsymbol{x}^{(\boldsymbol{j})} - (h/2)\mathbf{1}$, for $\boldsymbol{j} \in \{1, \ldots, N\}^n$, $\boldsymbol{\rho}^{(\boldsymbol{\ell})} = h\boldsymbol{\ell}$, for $\boldsymbol{\ell} \in \mathbb{Z}^n$, and $\hat{\boldsymbol{\rho}}^{(\boldsymbol{\ell})} = \boldsymbol{\rho}^{(\boldsymbol{\ell})} + (h/2)\mathbf{1}$, for $\boldsymbol{\ell} \in \mathbb{Z}^n$, where $\mathbf{1} = (1, 1, \ldots, 1)$. In our discretization we will employ indicator functions in the various cells, given by

$$\chi_j(\boldsymbol{x}) = \begin{cases} 1, & \text{if} \quad x_k^{(j-1)} < x_k \leqslant x_k^{(j)}, \quad \text{for each } 1 \leqslant k \leqslant n, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\hat{\chi}_\ell(\boldsymbol{\rho}) = \begin{cases} 1, & \text{if} \quad \hat{\rho}_k^{(\ell-1)} < \rho_k \leqslant \hat{\rho}_k^{(\ell)}, \quad \text{for each } 1 \leqslant k \leqslant n, \\ 0, & \text{otherwise.} \end{cases}$$

The regularization parameter $\alpha(\boldsymbol{x})$ is replaced with a discrete approximation

$$\alpha(\boldsymbol{x}) = \sum_{j \in \boldsymbol{J}} \alpha_j \cdot \chi_j(\boldsymbol{x}), \qquad \boldsymbol{x} \in \Omega.$$

The other regularization parameter $\boldsymbol{r}$ is chosen as $\boldsymbol{r} = h\boldsymbol{i}_r + (h/2)\mathbf{1}$, where $\boldsymbol{i}_r = (i_1, \ldots, i_n)$ (fixed) has integer components such that $1 \leqslant i_k < N/4$ for each $k$.

We also need a discrete approximation of the space $\mathcal{X}_r$. We first define the sets

$$J_r = \{\boldsymbol{j} = (j_1, \ldots, j_n) | i_k + 1 \leqslant j_k \leqslant N - i_k, \text{an integer}\}, \tag{4.1}$$

and

$$L_r = \{\boldsymbol{\ell} = (\ell_1, \ldots, \ell_n) | -i_k \leqslant \ell_k \leqslant i_k, \text{an integer}\}. \tag{4.2}$$

Then let

$$\mathcal{X}_r^N = \left\{ \varphi : \varphi(\boldsymbol{x})(\boldsymbol{\rho}) = \sum_{\boldsymbol{j} \in J_r} \sum_{\boldsymbol{\ell} \in L_r} c_{\boldsymbol{j}\boldsymbol{\ell}} \chi_{\boldsymbol{j}}(\boldsymbol{x}) \hat{\chi}_{\boldsymbol{\ell}}(\boldsymbol{\rho}) \right\}.$$

If $g \in \mathcal{X}_r$, then for $\boldsymbol{j} \in J_r$, $g(\hat{\boldsymbol{x}}^{(\boldsymbol{j})})(\cdot) \in L^2(-\boldsymbol{r}, \boldsymbol{r})$, and

$$\begin{aligned} \|g(\hat{\boldsymbol{x}}^{(\boldsymbol{j})})\|_2^2 &= \int_{[-\boldsymbol{r}, \boldsymbol{r}]} |g(\hat{\boldsymbol{x}}^{(\boldsymbol{j})})(\boldsymbol{\rho})|^2 \, \mathrm{d}\boldsymbol{\rho} \\ &= \sum_{\boldsymbol{\ell} \in L_r} \int_{[\hat{\boldsymbol{\rho}}^{(\boldsymbol{\ell}-\mathbf{1})}, \hat{\boldsymbol{\rho}}^{(\boldsymbol{\ell})}]} |g(\hat{\boldsymbol{x}}^{(\boldsymbol{j})})(\boldsymbol{\rho})|^2 \, \mathrm{d}\boldsymbol{\rho} \\ &\doteq \sum_{\boldsymbol{\ell} \in L_r} |g(\hat{\boldsymbol{x}}^{(\boldsymbol{j})})(\hat{\boldsymbol{\rho}}^{(\boldsymbol{\ell})})|^2 h^n. \end{aligned}$$

Setting $R_{\boldsymbol{j}} = [\boldsymbol{x}^{(\boldsymbol{j}-\mathbf{1})}, \boldsymbol{x}^{(\boldsymbol{j})}] \cap [\boldsymbol{r}, \mathbf{1} - \boldsymbol{r}]$ for $\boldsymbol{j} \in J_r$, we have

$$\begin{aligned} \|g\|_r^2 &= \frac{1}{2^n r_1 \cdots r_n} \int_{[\boldsymbol{r}, \mathbf{1}-\boldsymbol{r}]} \int_{[-\boldsymbol{r}, \boldsymbol{r}]} |g(\boldsymbol{x})(\boldsymbol{\rho})|^2 \, \mathrm{d}\boldsymbol{\rho} \, \mathrm{d}\boldsymbol{x} \\ &= \frac{1}{2^n r_1 \cdots r_n} \int_{[\boldsymbol{r}, \mathbf{1}-\boldsymbol{r}]} \|g(\boldsymbol{x})\|_2^2 \, \mathrm{d}\boldsymbol{x} \\ &= \frac{1}{2^n r_1 \cdots r_n} \sum_{\boldsymbol{j} \in J_r} \int_{R_{\boldsymbol{j}}} \|g(\boldsymbol{x})\|_2^2 \, \mathrm{d}\boldsymbol{x} \\ &\doteq \frac{1}{2^n r_1 \cdots r_n} \sum_{\boldsymbol{j} \in J_r} \|g(\hat{\boldsymbol{x}}^{(\boldsymbol{j})})\|_2^2 m(R_{\boldsymbol{j}}) \\ &\doteq \frac{h^n}{2^n r_1 \cdots r_n} \sum_{\boldsymbol{j} \in J_r} m(R_{\boldsymbol{j}}) \sum_{\boldsymbol{\ell} \in L_r} |g(\hat{\boldsymbol{x}}^{(\boldsymbol{j})})(\hat{\boldsymbol{\rho}}^{(\boldsymbol{\ell})})|^2, \end{aligned}$$

where $m(\cdot)$ denotes the 'volume' ($n$-dimensional Lebesgue measure) of a region in $\mathbb{R}^n$. In fact, when $g \in \mathcal{X}_r^N$ this is a strict equality (i.e., no approximations), and thus we define norms on $\mathcal{X}_r^N$ via

$$\|\varphi\|_{N,\boldsymbol{r}}^2 = \frac{h^n}{2^n r_1 \cdots r_n} \sum_{\boldsymbol{j} \in J_r} m(R_{\boldsymbol{j}}) \sum_{\boldsymbol{\ell} \in L_r} |\varphi(\hat{\boldsymbol{x}}^{(\boldsymbol{j})})(\hat{\boldsymbol{\rho}}^{(\boldsymbol{\ell})})|^2, \qquad \text{and}$$

$$\|\varphi\|_{N,\boldsymbol{r},\alpha}^2 = \frac{h^n}{2^n r_1 \cdots r_n} \sum_{\boldsymbol{j} \in J_r} m(R_{\boldsymbol{j}}) \sum_{\boldsymbol{\ell} \in L_r} |\varphi(\hat{\boldsymbol{x}}^{(\boldsymbol{j})})(\hat{\boldsymbol{\rho}}^{(\boldsymbol{\ell})})|^2 \alpha(\hat{\boldsymbol{x}}^{(\boldsymbol{j})}).$$

Now we need to calculate the operators $T_r$, $A_r$ and $B_r$. For $0 < c \leqslant 1$, let

$$\gamma_{\boldsymbol{l}}(\boldsymbol{\rho}) = \begin{cases} \frac{1}{c^n}, & \text{if} \quad \rho_k \in (-c, 0) \quad \text{for each } k, \\ 0, & \text{otherwise.} \end{cases} \tag{4.3}$$

Then, for $\varphi \in \mathcal{X}_{\boldsymbol{r}}^N$, $\boldsymbol{x} \in [\boldsymbol{r}, \mathbf{1} - \boldsymbol{r}]$, and $c \in (0, 1)$ small enough so that $\max_k cr_k \leqslant h/2$, we have

$$\begin{aligned} T_{\boldsymbol{r}}\varphi(\boldsymbol{x}) &= T E_r \varphi(\boldsymbol{x}) \\ &= \int_{(-\Delta,\Delta)^n} \gamma_l(\boldsymbol{\rho}) \cdot \varphi(\boldsymbol{x})(\rho_1 r_1, \ldots, \rho_n r_n)\, \mathrm{d}\boldsymbol{\rho} \\ &= \frac{1}{c^n} \int_{-c}^{0} \cdots \int_{-c}^{0} \varphi(\boldsymbol{x})(\rho_1 r_1, \ldots, \rho_n r_n)\, \mathrm{d}\boldsymbol{\rho} \\ &= \frac{1}{c^n r_1 r_2 \cdots r_n} \int_{-cr_1}^{0} \cdots \int_{-cr_n}^{0} \varphi(\boldsymbol{x})(\boldsymbol{\rho})\, \mathrm{d}\boldsymbol{\rho} \\ &= \frac{1}{c^n r_1 \cdots r_n} \int_{-cr_1}^{0} \cdots \int_{-cr_n}^{0} \sum_{\boldsymbol{j} \in J_r} \sum_{\boldsymbol{\ell} \in L_r} c_{\boldsymbol{j}\boldsymbol{\ell}}\, \chi_{\boldsymbol{j}}(\boldsymbol{x})\, \hat{\chi}_{\boldsymbol{\ell}}(\boldsymbol{\rho})\, \mathrm{d}\boldsymbol{\rho} \\ &= \sum_{\boldsymbol{j} \in J_r} c_{\boldsymbol{j}\mathbf{0}}\, \chi_{\boldsymbol{j}}(\boldsymbol{x}) \\ &= \varphi(\boldsymbol{x})(\mathbf{0}). \end{aligned} \tag{4.4}$$

To calculate $A_{\boldsymbol{r}} \colon \mathcal{X}_{\boldsymbol{r}}^N \to \mathcal{X}_{\boldsymbol{r}}^N$, we fix $\boldsymbol{j} \in J_r$ and $\boldsymbol{\ell} \in L_r$. Then

$$\begin{aligned} A_{\boldsymbol{r}}\varphi(\hat{\boldsymbol{x}}^{(j)})(\hat{\boldsymbol{\rho}}^{(\ell)}) &= \int_{[-\boldsymbol{r},\boldsymbol{r}]} k(\hat{\boldsymbol{x}}^{(j)} + \hat{\boldsymbol{\rho}}^{(\ell)}, \hat{\boldsymbol{x}}^{(j)} + \boldsymbol{s}) \varphi(\hat{\boldsymbol{x}}^{(j)})(\boldsymbol{s})\, \mathrm{d}\boldsymbol{s} \\ &= \int_{[-\boldsymbol{r},\boldsymbol{r}]} k(\boldsymbol{x}^{(j+\ell)}, \hat{\boldsymbol{x}}^{(j)} + \boldsymbol{s}) \left( \sum_{\boldsymbol{p} \in J_r} \sum_{\boldsymbol{q} \in L_r} c_{\boldsymbol{p}\boldsymbol{q}}\, \chi_{\boldsymbol{p}}(\hat{\boldsymbol{x}}^{(j)}) \hat{\chi}_{\boldsymbol{q}}(\boldsymbol{s}) \right) \mathrm{d}\boldsymbol{s} \\ &= \sum_{\boldsymbol{q} \in L_r} c_{\boldsymbol{j}\boldsymbol{q}} \int_{[-\boldsymbol{r},\boldsymbol{r}]} k(\boldsymbol{x}^{(j+\ell)}, \hat{\boldsymbol{x}}^{(j)} + \boldsymbol{s})\, \hat{\chi}_{\boldsymbol{q}}(\boldsymbol{s})\, \mathrm{d}\boldsymbol{s} \\ &= \sum_{\boldsymbol{q} \in L_r} c_{\boldsymbol{j}\boldsymbol{q}} \int_{[\hat{\boldsymbol{\rho}}^{(q-1)}, \hat{\boldsymbol{\rho}}^{(q)}]} k(\boldsymbol{x}^{(j+\ell)}, \hat{\boldsymbol{x}}^{(j)} + \boldsymbol{s})\, \mathrm{d}\boldsymbol{s} \\ &= \sum_{\boldsymbol{q} \in L_r} c_{\boldsymbol{j}\boldsymbol{q}} \int_{[\boldsymbol{x}^{(j+q-1)}, \boldsymbol{x}^{(j+q)}]} k(\boldsymbol{x}^{(j+\ell)}, \boldsymbol{v})\, \mathrm{d}\boldsymbol{v} \\ &= \sum_{\boldsymbol{q} \in L_r} c_{\boldsymbol{j}\boldsymbol{q}} \Delta_{\boldsymbol{j}+\boldsymbol{\ell}, \boldsymbol{j}+\boldsymbol{q}}, \end{aligned} \tag{4.5}$$

where

$$\Delta_{\boldsymbol{p},\boldsymbol{q}} = \int_{[\boldsymbol{x}^{(q-1)}, \boldsymbol{x}^{(q)}]} k(\boldsymbol{x}^{(p)}, \boldsymbol{v})\, \mathrm{d}\boldsymbol{v}. \tag{4.6}$$

The identity in (4.5) shows that our discretization leads to a finite-dimensional linear (matrix-like) representation of the operator $A_{\boldsymbol{r}}$. For instance, in the case $n = 1$ (where we adopt the convention of writing $x_j$ instead of $x^{(j)}$), let $i$ be the one-dimensional analogue of the vector $\boldsymbol{i}$, and $i + 1 \leqslant j \leqslant N - i$, $-i \leqslant \ell \leqslant i$. Then

$$A_r\varphi(\hat{x}_j)(\hat{\rho}_\ell) = (\mathbf{A}_j \cdot \mathbf{c}_j)_\ell, \tag{4.7}$$

where we define the matrix $\mathbf{A}_j$ and the vector $\mathbf{c}_j$ as:

$$\mathbf{A}_j = \left(\Delta_{j+\ell_1,j+\ell_2}\right)_{-i\leqslant\ell_1,\ell_2\leqslant i} = \begin{pmatrix} \Delta_{j-i,j-i} & \Delta_{j-i,j-i+1} & \cdots & \Delta_{j-i,j+i} \\ \Delta_{j-i+1,j-i} & \Delta_{j-i+1,j-i+1} & \cdots & \Delta_{j-i+1,j+i} \\ \vdots & \vdots & \ddots & \vdots \\ \Delta_{j+i,j-i} & \Delta_{j+i,j-i+1} & \cdots & \Delta_{j+i,j+i} \end{pmatrix}_{(2i+1)\times(2i+1)},$$

$$\mathbf{c}_j = \begin{pmatrix} c_{j,-i} & c_{j,-i+1} & \cdots & c_{j,i} \end{pmatrix}^T \in \mathbb{R}^{2i+1}.$$

Cases in which $n > 1$ are essentially the same, though it would be more natural to think of expression (4.5) as something like a tensor product. This is because, for each fixed $\boldsymbol{p}$, $(\Delta_{\boldsymbol{p},\boldsymbol{q}})$ is a $n$-dimensional array. To get to a matrix expression like (4.7), we arrange the $\Delta_{\boldsymbol{p},\boldsymbol{q}}$ (fixed $\boldsymbol{p}$) as a single row in $\mathbf{A}_j$, and treat the components of $\mathbf{c}_j$ similarly.

Now we calculate $B_r T_r \colon \mathcal{X}_r^N \to \mathcal{X}_r^N$. According to the definition of $B_r$ and (4.4), we have, for $\boldsymbol{j} \in J_r$ and $\boldsymbol{\ell} \in L_r$,

$$\begin{aligned} B_r T_r \varphi(\hat{\boldsymbol{x}}^{(\boldsymbol{j})})(\hat{\boldsymbol{\rho}}^{(\boldsymbol{\ell})}) &= \int_{\Omega\setminus B(\hat{\boldsymbol{x}}^{(\boldsymbol{j})},\boldsymbol{r})} k(\hat{\boldsymbol{x}}^{(\boldsymbol{j})} + \hat{\boldsymbol{\rho}}^{(\boldsymbol{\ell})}, \boldsymbol{s}) T_r \varphi(\boldsymbol{s})\, \mathrm{d}\boldsymbol{s} \\ &= \sum_{\boldsymbol{\mu}\in J_r} c_{\boldsymbol{\mu}\mathbf{0}} \int_{\Omega\setminus B(\hat{\boldsymbol{x}}^{(\boldsymbol{j})},\boldsymbol{r})} k(\boldsymbol{x}^{(\boldsymbol{j}+\boldsymbol{\ell})}, \boldsymbol{s})\ \chi_{\boldsymbol{\mu}}(\boldsymbol{s})\, \mathrm{d}\boldsymbol{s}. \end{aligned} \tag{4.8}$$

While, as in the case of $A_r\varphi(\hat{\boldsymbol{x}}^{(\boldsymbol{j})})(\hat{\boldsymbol{\rho}}^{(\boldsymbol{\ell})})$, such values may be described via matrix multiplication, because of the hole in the region of integration, any generic description of the matrix necessarily is broken up by the presence of zeroed entries. For instance, in the case $n = 1$ we have, for $i + 1 \leqslant j \leqslant N - i$ and $-i \leqslant \ell \leqslant i$,

$$\begin{aligned} B_r T_r \varphi(\hat{x}_j)(\hat{\rho}_\ell) &= \left(\int_0^{x_j-(i+1)h} + \int_{x_j+ih}^1\right) k(\hat{x}_j + \hat{\rho}_\ell, s) \cdot \left(\sum_{\mu=i+1}^{N-i} c_{\mu 0}\chi_\mu(s)\right) \mathrm{d}s \\ &= \sum_{\mu=i+1}^{N-i} \int_0^{x_{j-i-1}} k(x_{j+\ell}, s) c_{\mu 0}\chi_\mu(s)\, \mathrm{d}s + \sum_{\mu=i+1}^{N-i} \int_{x_{j+i}}^1 k(x_{j+\ell}, s) c_{\mu 0}\chi_\mu(s)\, \mathrm{d}s \\ &= \sum_{\mu=i+1}^{j-i-1} \int_0^{x_{j-i-1}} k(x_{j+\ell}, s) c_{\mu 0}\chi_\mu(s)\, \mathrm{d}s \\ &\quad + \sum_{\mu=j+i+1}^{N-i} \int_{x_{j+i}}^1 k(x_{j+\ell}, s) c_{\mu 0}\chi_\mu(s)\, \mathrm{d}s, \end{aligned} \tag{4.9}$$

where a summation whose lower index exceeds the upper one is to be interpreted as zero. The case $n = 1$, then, has three subcases, depending on the value of $j$:

*Case 1.* If $j < 2(i + 1)$, then $0 \leqslant j - i - 1 < i + 1$, making the first sum vacuous. Hence

$$\begin{aligned} B_r T_r \varphi(\hat{x}_j)(\hat{\rho}_\ell) &= \sum_{\mu=j+i+1}^{N-i} \int_{x_{j+i}}^1 k(x_{j+\ell}, s) c_{\mu 0}\chi_\mu(s)\, \mathrm{d}s \\ &= \sum_{\mu=j+i+1}^{N-i} \int_{x_{\mu-1}}^{x_\mu} k(x_{j+\ell}, s) c_{\mu 0}\, \mathrm{d}s \\ &= \sum_{\mu=j+i+1}^{N-i} c_{\mu 0}\Delta_{j+\ell,\mu}. \end{aligned} \tag{4.10}$$

In this case, if we define the vector $\bar{\mathbf{c}}$ as

$$\bar{\mathbf{c}} = \begin{pmatrix} c_{i+1,0} & c_{i+2,0} & \cdots & c_{N-i,0} \end{pmatrix}^T \in \mathbb{R}^{N-2i},$$

then

$$B_r T_r \varphi(\hat{x}_j)(\hat{\rho}_\ell) = (\mathbf{B}_j \cdot \bar{\mathbf{c}})_\ell, \tag{4.11}$$

where $\mathbf{B}_j$ is the $(2i+1) \times (N-2i)$ matrix given by

$$\begin{aligned} \mathbf{B}_j &= (\mathbf{0}_{(2i+1)\times j} | \, (\Delta_{j+\ell,\mu})_{(2i+1)\times(N-2i-j)}) \\ &= \begin{pmatrix} 0 & \cdots & 0 & \Delta_{j-i,j+i+1} & \Delta_{j-i,j+i+2} & \cdots & \Delta_{j-i,N-i} \\ 0 & \cdots & 0 & \Delta_{j-i+1,j+i+1} & \Delta_{j-i+1,j+i+2} & \cdots & \Delta_{j-i+1,N-i} \\ \vdots & & \vdots & \vdots & \vdots & & \vdots \\ 0 & \cdots & 0 & \Delta_{j+i,j+i+1} & \Delta_{j+i,j+i+2} & \cdots & \Delta_{j+i,N-i} \end{pmatrix}. \end{aligned} \tag{4.12}$$

*Case 2*. If $2(i+1) \leqslant j \leqslant N-2i-1$, then from (4.9) we get

$$\begin{aligned} B_r T_r \varphi(\hat{x}_j)(\hat{\rho}_\ell) &= \sum_{\mu=i+1}^{j-i-1} \int_{x_{\mu-1}}^{x_\mu} k(x_{j+\ell}, s) c_{\mu 0} \, \mathrm{d}s + \sum_{\mu=j+i+1}^{N-i} \int_{x_{\mu-1}}^{x_\mu} k(x_{j+\ell}, s) c_{\mu 0} \, \mathrm{d}s \\ &= \left( \sum_{\mu=i+1}^{j-i-1} + \sum_{\mu=j+i+1}^{N-i} \right) c_{\mu 0} \Delta_{j+\ell,\mu}. \end{aligned} \tag{4.13}$$

Here (4.11) holds with

$$\mathbf{B}_j = (\mathbf{S}_j | \mathbf{0}_{(2i+1)\times(2i+1)} | \mathbf{T}_j), \tag{4.14}$$

where

$$\mathbf{S}_j = \begin{pmatrix} \Delta_{j-i,i+1} & \cdots & \Delta_{j-i,j-i-1} \\ \Delta_{j-i+1,i+1} & \cdots & \Delta_{j-i+1,j-i-1} \\ \vdots & & \vdots \\ \Delta_{j+i,i+1} & \cdots & \Delta_{j+i,j-i-1} \end{pmatrix} \qquad \text{and}$$

$$\mathbf{T}_j = \begin{pmatrix} \Delta_{j-i,j+i+1} & \cdots & \Delta_{j-i,N-i} \\ \Delta_{j-i+1,j+i+1} & \cdots & \Delta_{j-i+1,N-i} \\ \vdots & & \vdots \\ \Delta_{j+i,j+i+1} & \cdots & \Delta_{j+i,N-i} \end{pmatrix}$$

are $(2i+1) \times (j-2i-1)$ and $(2i+1) \times (N-2i-j)$ matrices respectively.

*Case 3*. If $j > N-2i-1$, then $j+i+1 > N-i$. Dropping the vacuous second sum in (4.9), we get

$$B_r T_r \varphi(\hat{x}_j)(\hat{\rho}_\ell) = \sum_{\mu=i+1}^{j-i-1} c_{\mu 0} \Delta_{j+\ell,\mu}. \tag{4.15}$$

In this case, (4.11) holds with

$$\begin{aligned} \mathbf{B}_j &= ((\Delta_{j+\ell,\mu})_{(2i+1)\times(j-2i-1)} \mid \mathbf{0}_{(2i+1)\times(N-j+1)}) \\ &= \begin{pmatrix} \Delta_{j-i,i+1} & \Delta_{j-i,i+2} & \cdots & \Delta_{j-i,j-i-1} & 0 & \cdots & 0 \\ \Delta_{j-i+1,i+1} & \Delta_{j-i+1,i+2} & \cdots & \Delta_{j-i+1,j-i-1} & 0 & \cdots & 0 \\ \vdots & \vdots & & \vdots & \vdots & & \vdots \\ \Delta_{j+i,i+1} & \Delta_{j+i,i+2} & \cdots & \Delta_{j+i,j-i-1} & 0 & \cdots & 0 \end{pmatrix}. \end{aligned} \tag{4.16}$$

We note that for the discrete algorithms developed below we will require, for a general (non-convolution) kernel, $\prod_{k=1}^{n} N(N-2i_k) = \mathcal{O}(N^{2n})$ (possibly) different $\Delta_{p,q}$ values. Even in $n = 2$ dimensions this is too costly for current computers when $N$ is of reasonable size. For kernels of convolution type $k(\boldsymbol{x}, \boldsymbol{y}) = K(\boldsymbol{x} - \boldsymbol{y})$, however, there are many repeat values. In the case $n = 2$, for instance, the matrix of $\Delta_{p,q}$-values has a block Toeplitz with Toeplitz blocks structure and, following [26], we exploit this structure in our algorithms in order to reduce the number of computed $\Delta_{p,q}$-values to $(2N-1)^2 = \mathcal{O}(4N^2)$ from $\mathcal{O}(N^4)$.

The last quantity we require is $F_r^{N,\delta}$, our discrete data. By definition,

$$F_r^{N,\delta}(\hat{\boldsymbol{x}}^{(j)})(\hat{\boldsymbol{\rho}}^{(\ell)}) = f^{\delta}(\hat{\boldsymbol{x}}^{(j)} + \hat{\boldsymbol{\rho}}^{(\ell)}) = f^{\delta}(\boldsymbol{x}^{(j+\ell)}), \qquad \text{for} \quad \boldsymbol{j} \in J_r, \boldsymbol{\ell} \in L_r.$$

For fixed $\boldsymbol{j}$ in the $n = 1$ case, we are led naturally to define the vector

$$\bar{\mathbf{f}}_j = (f^{\delta}(x_{j-i}) \quad f^{\delta}(x_{j-i+1}) \quad \cdots \quad f^{\delta}(x_{j+i}))^T \in \mathbb{R}^{2i+1}. \tag{4.17}$$

When $n > 1$, it would be more natural to think of our data as a multidimensional array in $\mathbb{R}^{(2i_1+1)\times\cdots\times(2i_n+1)}$. Nevertheless, to stay within a matrix-multiplication framework, we 'straighten out' the contents of that array into a single (column) vector, much in the same way *Matlab* does when a multi-indexed array *array*(:, :, . . . , :) is accessed with just one index *array*(:).

Finally we can define the discrete format of the local regularization problem $\mathcal{P}_{r,\alpha}^{\delta}$ given in definition 2.1: for $\varphi \in \mathcal{X}_r^N$

$$\begin{aligned}
\left\| A_r\varphi + B_r T_r\varphi - F_r^{N,\delta} \right\|_{N,r}^2 + \|\varphi\|_{N,r,\alpha}^2 &= \frac{h^n}{2^n r_1 \cdots r_n} \sum_{j\in J_r} m(R_j) \sum_{\ell\in L_r} \left( |\varphi(\hat{\boldsymbol{x}}^{(j)})(\hat{\boldsymbol{\rho}}^{(\ell)})|^2 \alpha(\hat{\boldsymbol{x}}^{(j)}) \right. \\
&\quad \left. + \left| A_r\varphi(\hat{\boldsymbol{x}}^{(j)})(\hat{\boldsymbol{\rho}}^{(\ell)}) + B_r T_r\varphi(\hat{\boldsymbol{x}}^{(j)})(\hat{\boldsymbol{\rho}}^{(\ell)}) - F_r^{N,\delta}(\hat{\boldsymbol{x}}^{(j)})(\hat{\boldsymbol{\rho}}^{(\ell)}) \right|^2 \right) \\
&= \frac{h^n}{2^n r_1 \cdots r_n} \sum_{j\in J_r} m(R_j) \sum_{\ell\in L_r} \left( |(\mathbf{A}_j \cdot \mathbf{c}_j + \mathbf{B}_j \cdot \bar{\mathbf{c}} - \bar{\mathbf{f}}_j)_\ell|^2 + (c_{j\ell})^2 \alpha(\hat{\boldsymbol{x}}^{(j)}) \right) \\
&= \frac{h^n}{2^n r_1 \cdots r_n} \sum_{j\in J_r} H_j(\mathbf{c}_j; \bar{\mathbf{c}}),
\end{aligned} \tag{4.18}$$

where $H_j(\mathbf{c}_j; \bar{\mathbf{c}})$ is defined as

$$H_j(\mathbf{c}_j; \bar{\mathbf{c}}) = m(R_j) \sum_{\ell\in L_r} \left( |(\mathbf{A}_j \cdot \mathbf{c}_j + \mathbf{B}_j \cdot \bar{\mathbf{c}} - \bar{\mathbf{f}}_j)_\ell|^2 + (c_{j\ell})^2 \alpha(\hat{\boldsymbol{x}}^{(j)}) \right).$$

In order to describe the relaxation type of minimization method we are going to use to solve our discrete regularization problem, we introduce the following notations. Fix $\boldsymbol{m} \in J_r$, and define

$$\mathcal{J}_m(\mathbf{c}_m) = H_m(\mathbf{c}_m; \bar{\mathbf{c}}).$$

For $\boldsymbol{j} \neq \boldsymbol{m} \in J_r$, note that $H_j(\mathbf{c}_j; \bar{\mathbf{c}})$ depends on $\mathbf{c}_m$ only through the component $c_{m\mathbf{0}}$ in $\bar{\mathbf{c}}$. So it is valid to define

$$\hat{\mathcal{J}}_m(c_{m\mathbf{0}}) = \sum_{j\in J_r, j\neq m} H_j(\mathbf{c}_j; \bar{\mathbf{c}}).$$

Then

$$\frac{2^n r_1 \cdots r_n}{h^n} \left( \left\| A_r\varphi + B_r T_r\varphi - F_r^{N,\delta} \right\|_{N,r}^2 + \|\varphi\|_{N,r,\alpha}^2 \right) = \mathcal{J}_m(\mathbf{c}_m) + \hat{\mathcal{J}}_m(c_{m\mathbf{0}}).$$

Using notations $\mathcal{J}_m(\mathbf{c}_m)$ and $\hat{\mathcal{J}}_m(c_{m\mathbf{0}})$, we introduce the following iterative relaxation-type minimization algorithm for the discrete regularization problem:

*Local regularization algorithm 1*

(1) Initialize vectors $\mathbf{c}_j$, $\boldsymbol{j} \in J_r$.
(2) Do for $\boldsymbol{m} \in J_r$:
  (a) Holding the previously determined values of $\mathbf{c}_j$, $\boldsymbol{j} \neq \boldsymbol{m}$, find $\bar{\boldsymbol{\beta}} \in \mathbb{R}^{2i_1 \times \cdots \times 2i_n}$ solving
$$\min\{\mathcal{J}_{\boldsymbol{m}}(\boldsymbol{\beta}) + \hat{\mathcal{J}}_{\boldsymbol{m}}(\boldsymbol{\beta}_{\mathbf{0}}) : \boldsymbol{\beta} = (\beta_{\boldsymbol{\ell}})_{\boldsymbol{\ell} \in L_r} \in \mathbb{R}^{2i_1 \times \cdots \times 2i_n}\}.$$
  (b) Set $\mathbf{c}_{\boldsymbol{m}} = \bar{\boldsymbol{\beta}}$.
(3) Go to step 2.

Local regularization algorithm 1 is that which comes directly from a discretization of the integral equations in sections 2 and 3. In practice it has been observed that nearly identical results are obtained if the algorithm is simplified considerably. Such simplifications (given in the local regularization algorithms 2 and 3 below) lead to significantly faster numerical computations.

*Local regularization algorithm 2*

(1) Initialize numbers $c_j$, $\boldsymbol{j} \in J_r$.
(2) Do for $\boldsymbol{m} \in J_r$:
  (a) Holding the previously determined values of $c_j$, $\boldsymbol{j} \neq \boldsymbol{m}$, find $\bar{\boldsymbol{\beta}} \in \mathbb{R}^{2i_1 \times \cdots \times 2i_n}$ solving
$$\min\{\mathcal{J}_{\boldsymbol{m}}(\boldsymbol{\beta}) : \boldsymbol{\beta} = (\beta_{\boldsymbol{\ell}})_{\boldsymbol{\ell} \in L_r} \in \mathbb{R}^{2i_1 \times \cdots \times 2i_n}\},$$
  or, equivalently, solving
$$\min_{\boldsymbol{\beta} \in \mathbb{R}^{2i_1 \times \cdots \times 2i_n}} \|\mathbf{A}_{\boldsymbol{m}} \cdot \boldsymbol{\beta} + \mathbf{B}_{\boldsymbol{m}} \cdot \bar{\mathbf{c}} - \bar{\mathbf{f}}_{\boldsymbol{m}}\|^2 + \alpha(\boldsymbol{x}^{(\boldsymbol{m})})\|\boldsymbol{\beta}\|^2,$$
  where $\bar{\mathbf{c}}$ is composed of the numbers $c_j$, $j \in J_r$.
  (b) Set $c_{\boldsymbol{m}} = \bar{\beta}_{\mathbf{0}}$.
(3) Go to step 2.

Each time step 3 is reached in algorithm 2, one might say that $\bar{\mathbf{c}}$ represents a new iterate, a new approximation to the true solution of (1.1) and (1.2). To be more explicit, going into step 2 there is a state $\bar{\mathbf{c}}_{\text{old}}$, and we reach step 3 having made adjustments to each $c_{\boldsymbol{m}}$, $\boldsymbol{m} \in J_r$ in order to reach a new state $\bar{\mathbf{c}}_{\text{new}}$. Let $\boldsymbol{\epsilon} = \bar{\mathbf{c}}_{\text{new}} - \bar{\mathbf{c}}_{\text{old}}$, the vector containing only these adjustments. Before entering step 2 again, one can set $\bar{\mathbf{c}}_{\text{old}}$ equal to $\bar{\mathbf{c}}_{\text{new}}$ and $\boldsymbol{\epsilon} = \mathbf{0}$. Then, as we step through the $\boldsymbol{m} \in J_r$, $\boldsymbol{\epsilon}$ fills up sequentially with nonzero entries.

As we shall see in the next subsection, algorithm 2 is costly ($\mathcal{O}(N^4)$) to implement. By far the greatest cost comes from computing the matrix–vector product ($\mathbf{B}_{\boldsymbol{m}} \cdot \bar{\mathbf{c}}$) which, for each $\boldsymbol{m} \in J_r$, is $\mathcal{O}(N^2)$. If we let the matrix $\mathbf{A}$ denote the full, unregularized matrix operator that results from discretizing the operator in (1.2)—a matrix whose $\boldsymbol{p}$th row consists of the $\Delta_{\boldsymbol{pq}}$ with $\boldsymbol{q}$ running through all elements of $J = \{1, 2, \ldots, N\}^n$—then one can fairly easily extract out the product ($\mathbf{B}_{\boldsymbol{m}} \cdot \bar{\mathbf{c}}$) for any given $\boldsymbol{m} \in J_r$ from the product ($\mathbf{A} \cdot \bar{\mathbf{c}}$). To see this, let $\mathbf{L}_{\boldsymbol{m}}$ denote the operator that extracts the $(2\boldsymbol{i}_r + \mathbf{1})$-width neighbourhood surrounding entry $\boldsymbol{m}$. Then
$$\mathbf{A}_{\boldsymbol{m}} \cdot \mathbf{L}_{\boldsymbol{m}}(\bar{\mathbf{c}}) + \mathbf{B}_{\boldsymbol{m}} \cdot \bar{\mathbf{c}} = \mathbf{L}_{\boldsymbol{m}}(\mathbf{A} \cdot \bar{\mathbf{c}}).$$
Note that there is an abuse of notation here in that, as we have defined it, $\bar{\mathbf{c}}$ is not of the appropriate size for the product ($\mathbf{A} \cdot \bar{\mathbf{c}}$) to make sense. Our practice is to pad $\bar{\mathbf{c}}$ with zeros in the 'border entries.' (It is possible to pad these entries with local solution values; see [20] for this type of approach.) For instance, in $n = 1$ dimension with $N = 8$ and $i = 2$, if $\bar{\mathbf{c}} = (3, 1, 2, 5)$ then in order to make a well-defined product ($\mathbf{A} \cdot \bar{\mathbf{c}}$) we first pad $\bar{\mathbf{c}}$ with $i = 2$ zeros at each end to get the vector $(0, 0, 3, 1, 2, 5, 0, 0)$. So, our equation above really should be written as
$$\mathbf{A}_{\boldsymbol{m}} \cdot \mathbf{L}_{\boldsymbol{m}}(\bar{\mathbf{c}}) + \mathbf{B}_{\boldsymbol{m}} \cdot \bar{\mathbf{c}} = \mathbf{L}_{\boldsymbol{m}}(\mathbf{A} \cdot \mathbf{P}(\bar{\mathbf{c}})),$$
where $\mathbf{P}$ is this padding operator (linear).

Now, let $\boldsymbol{m} \in J_r$ be fixed, designating an intermediary stage of carrying out step 2 of algorithm 2. Let the changes made thus far to the previous iterate $\bar{\mathbf{c}}_{\text{old}}$ up to stage $\boldsymbol{m}$ be contained in $\boldsymbol{\epsilon}$, and let $\bar{\mathbf{c}}$ be the currently-developing iterate, so $\bar{\mathbf{c}} = \bar{\mathbf{c}}_{\text{old}} + \boldsymbol{\epsilon}$. Then

$$\begin{aligned}\mathbf{B}_m \cdot \bar{\mathbf{c}} &= \mathbf{L}_m(\mathbf{A} \cdot \mathbf{P}(\bar{\mathbf{c}})) - \mathbf{A}_m \cdot \mathbf{L}_m(\bar{\mathbf{c}}) \\ &= \mathbf{L}_m(\mathbf{A} \cdot \mathbf{P}(\bar{\mathbf{c}}_{\text{old}})) - \mathbf{A}_m \cdot \mathbf{L}_m(\bar{\mathbf{c}}_{\text{old}}) + \mathbf{B}_m \cdot \boldsymbol{\epsilon}. \end{aligned} \tag{4.19}$$

Note that one can compute $\mathbf{A} \cdot \mathbf{P}(\bar{\mathbf{c}}_{\text{old}})$ before entering into step 2. Employing the block Toeplitz with Toeplitz blocks (BTTB) structure of $\mathbf{A}$, this computation is $\mathcal{O}(N^2 \log N)$. Computing $(\mathbf{B}_m \cdot \bar{\mathbf{c}})$ via (4.19) would not represent any savings, however, as $(\mathbf{B}_m \cdot \boldsymbol{\epsilon})$ is roughly the same cost to compute as $(\mathbf{B}_m \cdot \bar{\mathbf{c}})$. But knowing that the entries of $(\mathbf{B}_m \cdot \bar{\mathbf{c}})$ nearest to the local region surrounding entry $\boldsymbol{m}$ are the most influential in determining the solution of the minimization problem of Step 2, we replace $(\mathbf{B}_m \cdot \boldsymbol{\epsilon})$ with a reduced calculation. Namely, we fix an integer $L > 1$, and keep only the parts of $\boldsymbol{\epsilon}$ in an $(2L\boldsymbol{i}_r + \mathbf{1})$-width neighbourhood of $\boldsymbol{m}$ along with the corresponding parts of $\mathbf{B}_m$ which act on these parts of $\boldsymbol{\epsilon}$. The algorithm using this approximation of $(\mathbf{B}_m \cdot \bar{\mathbf{c}})$ will be called *local regularization algorithm 3*. In principle, it seems the size of $L$ might have to grow with $N$. In practice we have found $L \leqslant 3$ to be generally sufficient for most problems, at least when $N$ is small enough that the problem itself does not get out of hand for today's computers. This means that the products $(\mathbf{B}_m \cdot \bar{\mathbf{c}})$ are dominated by the $\mathcal{O}(N^2 \log N)$ calculation of $(\mathbf{A} \cdot P(\bar{\mathbf{c}}))$, and may continue to be as $N$ grows.

### *4.2. Operation counts in two dimensions*

As we saw in the last section, for each $\boldsymbol{m} \in J_r$, algorithm 2 calls for the solution to the minimization problem

$$\min_{\boldsymbol{\beta}} \|\mathbf{A}_m \cdot \boldsymbol{\beta} + \mathbf{B}_m \cdot \bar{\mathbf{c}} - \bar{\mathbf{f}}_m\|_2^2 + \alpha(\hat{\boldsymbol{x}}^{(m)})\|\boldsymbol{\beta}\|_2^2. \tag{4.20}$$

As is well known (see, for instance, [13]), the solution to this minimization problem is

$$\operatorname*{argmin}_{\boldsymbol{\beta}} \|\mathbf{D}_\mathbf{m}\boldsymbol{\beta} - \mathbf{g}_\mathbf{m}\|_2^2,$$

where

$$\mathbf{D}_m = \begin{pmatrix} \mathbf{A}_m \\ \sqrt{\alpha(\hat{\boldsymbol{x}}^{(m)})}\,\mathbf{I} \end{pmatrix}, \qquad \mathbf{g}_m = \begin{pmatrix} \bar{\mathbf{f}}_m - \mathbf{B}_m \cdot \bar{\mathbf{c}} \\ \mathbf{0} \end{pmatrix},$$

and, in the case $n = 2$, $\mathbf{I}$ represents the $(2i_1+1)(2i_2+1) \times (2i_1+1)(2i_2+1)$ identity matrix, and $\mathbf{0}$ a zero vector of length $(2i_1+1)(2i_2+1)$. One approach to solving this least-squares problem is to apply Gaussian elimination ($LU$-decomposition) to the associated *normal equations*

$$\mathbf{D}_m^* \mathbf{D}_m \boldsymbol{\beta} = \mathbf{D}_m^* \mathbf{g}_m. \tag{4.21}$$

(In general, one would solve (4.20) via more stable methods like those of Householder or Givens—see, e.g., chapter 9 of [13]—but given the relatively small dimension of the system, the direct solution of the normal equations (4.21) by elimination should not introduce instabilities. The operation counts for these other methods are of the same order.) Since

$$\mathbf{D}_m^* \mathbf{D}_m = \mathbf{A}_m^* \mathbf{A}_m + \alpha(\hat{\boldsymbol{x}}^{(m)})\mathbf{I},$$

and $\mathbf{A}_m$ is constant (not varying with $\boldsymbol{m}$) for convolution kernels, it is the case that the computation of $\mathbf{A}_m^* \mathbf{A}_m$, which represents most of the work in computing $\mathbf{D}_m^* \mathbf{D}_m$, is performed only once (not once for each $\boldsymbol{m} \in J_r$). For algorithm 2, we use standard operation counts from linear algebra to get the following breakdown for those subprocesses which are carried out anew for *each* $\boldsymbol{m} \in J_r$, assuming $i_k \ll N$ for all $k$:

| Subprocess | Flops (leading order term) |
|---|---|
| Compute $\bar{\mathbf{f}}_j - \mathbf{B}_j \cdot \bar{\mathbf{c}}$ | $2(2i_1+1)(2i_2+1)N^2$ |
| Compute $\mathbf{D}_j^* \mathbf{g}_j = \mathbf{A}_j^*(\bar{\mathbf{f}}_j - \mathbf{B}_j \cdot \bar{\mathbf{c}})$ | $2(2i_1+1)^2(2i_2+1)^2$ |
| Decompose $\mathbf{D}_j^* \mathbf{D}_j$ into $LU$ | $(2/3)[(2i_1+1)(2i_2+1)]^3$ |
| Solve $L\boldsymbol{y} = \mathbf{D}_j^* \boldsymbol{g}_j$ by forward substitution | $[(2i_1+1)(2i_2+1)]^2$ |
| Solve $U\beta = \boldsymbol{y}$ by backward substitution | $[(2i_1+1)(2i_2+1)]^2$ |

If each $i_k \ll N$, then the most significant term above is the calculation of $\bar{\mathbf{f}}_j - \mathbf{B}_j \cdot \bar{\mathbf{c}}$, which (to leading order) requires $\mathcal{O}(8i_1 i_2 N^2)$ floating point operations. Considering that during one iteration of the local Tikhonov method such a calculation takes place for each $\boldsymbol{j} \in J_r$, which in the case $n = 2$ has $(N - 2i_1)(N - 2i_2)$ elements, this means that each local Tikhonov iteration requires $\mathcal{O}(8i_1 i_2 N^4)$ floating point operations. By comparison, algorithm 3 employs $\mathcal{O}(N^2 \log N)$ floating point calculations.

The above represents a big savings over standard Tikhonov regularization, for which the computational work is of order $\mathcal{O}(N^6)$ operations in general, and $\mathcal{O}(N^4)$ if the Toeplitz structure is taken into account (see p 233 of [13], and the references therein).

As a computational savings, Tikhonov regularization can be coupled with an iterative method such as the conjugate gradient method, with a regularized stopping criterion. We will henceforth refer to this numerical approach as Tik-CG. Following [26], the assumption of a convolution kernel means that the Tikhonov matrix $\mathbf{A}$ has special structure. This fact may be exploited so that matrix–vector products involving $(\mathbf{A}^*\mathbf{A} + \alpha\mathbf{I})$, which account for the greatest cost in Tik-CG, may be carried out using two-dimensional fast Fourier transforms at a much greater efficiency than standard matrix–vector products. In the case $n = 2$ this means $\mathcal{O}(N^2 \log_2(N))$ floating point operations (to leading order) as opposed to $\mathcal{O}(N^4)$.

Comparing the estimates for Tik-CG with those for our local Tikhonov method, Tik-CG is potentially less costly (and, in practice, this is borne out). In $n = 1$ dimension one is unlikely to notice the difference. Even in $n = 2$ dimensions, we think our examples in the next section demonstrate that the regularization properties of the local Tikhonov method may, at least some of the time, warrant using the method over Tik-CG. And, local Tikhonov offers the possibility of varying the amount of regularization throughout different regions of the domain, by letting $\alpha(\cdot)$ vary over grid points. A few *ad hoc* examples with variable regularization appear in [6, 20] in a one-dimensional setting, but this remains a direction for future research.

### *4.3. Numerical examples*

We display numerical examples in $n = 2$ dimensions. (For examples in one dimension, see [6].) The Gaussian kernel function $k$ is given by (1.4) and the corresponding operator $\mathcal{A}$ represents the blurring operator. Several choices of $\beta$ are used. In each case, a true solution $\bar{u}$ was pre-selected, and the noisy data $f^\delta$ used in the regularization process are a random perturbation of $f = \mathcal{A}\bar{u}$, with the latter computed using quadrature. All calculations are conducted using Matlab.

In all cases the original image $\bar{u}$ takes values in $[0, 1]$ (after rescaling of the functional representation of $\bar{u}$, if needed). Approximations of $\bar{u}$ via local regularization or the conjugate gradient approach to Tikhonov regularization may actually take values outside $[0, 1]$, however, so we display the approximate graphical images in the same way that a camera would, i.e., by setting values smaller than 0.0 to 0.0 (compression of blacks) and larger than 1.0 to 1.0 (blowing-out of whites). When looking at the actual values of the approximate solutions in

the examples which follow, we frequently observed that the local regularization method demonstrated significantly less compression and/or blow-out than the classical method, suggesting that local regularization appears to exert tighter controls over the overall image.

We make comparisons between local regularization and the coupling of Tikhonov regularization with the conjugate gradient method (Tik-CG). It is perhaps desirable to make a comparison also with edge-sharpening methods such as bounded variation (BV) methods, but since local Tikhonov and Tik-CG both reside in the class of differentiable optimization techniques (and BV methods do not), the comparison between the two seems more apt. In all examples below, we have run numerous tests with different values of the regularization parameters $\alpha$ (for Tik-CG) and $\boldsymbol{r}$ and $\alpha$ (for local regularization) to determine which parameters were 'best' in the sense of providing an approximate solution with smallest relative error. In what follows we only present the results obtained with these 'best' parameters in our findings for both Tik-CG and local regularization. Tik-CG solutions are computed employing block circulant extension preconditioning as described in [26], and indicated in that reference as being similar to the Toeplitz approximate inverse conditioners of [15], and using code adapted from routines made available by C Vogel at his website[4]. For the local regularization solutions, algorithm 3 is employed, generally cycling through until the relative error changed little between iterates. In the case of local Tikhonov, relative error is really meaningful only on the $(N - 2i_1) \times (N - 2i_2)$ subarray. So, the stated relative errors for both Tik-CG and local Tikhonov solutions have been computed on this subarray. Because the values off this subarray have been set (arbitrarily) to 0.0 for local regularization, a black border appears around each image. Alternatively, the values could have been set to 1.0 around the border (giving a better visual presentation for most examples); however, testing showed that values of the solution on the internal subarray changed little in this case.

**Example 4.1.** The true solution, a grey box on a white background, is displayed at top left in figure 1, along with the blurred $u$ containing 15% noise, the Tik-CG and local Tikhonov solutions. Here, our blurring operator has kernel (1.4) with $\beta = 100$. For the Tik-CG solution, we have displayed the method's 4th iterate with the 'best' value of $\alpha = 0.1$, which required approximately 0.34 s of computation time. For the local Tikhonov solution, 'best' parameters were found to be $\alpha = 0.0075$ (constant), $L = 1$, and $\boldsymbol{i}_r = [2\ 2]$. In this case, the third iterate is displayed, which took 7.24 s of computation time. Roughly the same local Tikhonov solution is found in a single iteration (1.99 s) when the Tik-CG solution is taken as the initial state.

**Example 4.2.** Here the true solution consists of vertical black stripes on a white background. The blurred $u$ comes from a Gaussian kernel (1.4) with $\beta = 175$, and contains 10% relative error. These are displayed in figure 2 along with the Tik-CG and local Tikhonov solutions. For the Tik-CG solution, the 'best' parameter is $\alpha = 0.105$; the resulting fourth iterate is displayed, requiring 0.31 s of computation time. For the local Tikhonov solution, the best parameters are $\alpha = 0.0075$, $L = 1$ and $\boldsymbol{i}_r = [2\ 2]$. The third iterate is displayed, which took 7.27 s of computation time. Roughly the same local Tikhonov solution is found in a single iteration (2.48 s) when the Tik-CG solution is taken as the initial state.

**Example 4.3.** Here the true solution consists of a diagonal cross on a white background, where a grey stripe passes over a black one. In figure 3(a), $N = 64$, $\beta = 100$, and the blurred $u$ contains 10% relative error. The Tik-CG solution once again is the fourth iterate, coming from the best parameter value $\alpha = 0.095$. In the local Tikhonov solution, the best parameters

[4] See http://www.math.montana.edu/∼ vogel/Book/Codes/Ch5.

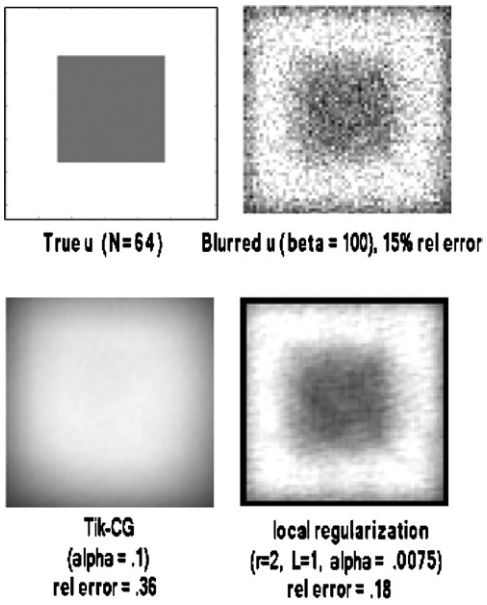


**Figure 1.** Example 4.1.

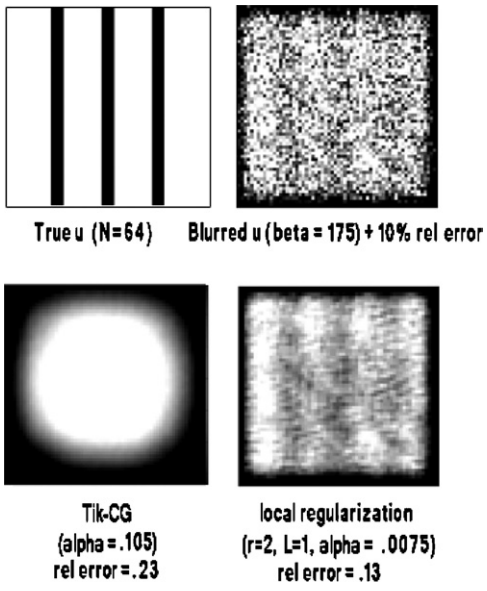


**Figure 2.** Example 4.2.

are $\alpha = 0.0075$ (constant), $L = 4$, and $i_r = [2\ 2]$. The third iterate is displayed, and the computation times are roughly the same as in examples 4.1 and 4.2.

In figure 3(b), $N = 128$ and $\beta = 250$. The blurred $u$ contains 15% relative error. For the Tik-CG solution, the best parameter value is $\alpha = 0.1$ and, again, we display the fourth iterate. For the local Tikhonov solution, the best parameter values are $\alpha = 0.005$, $L = 1$ and

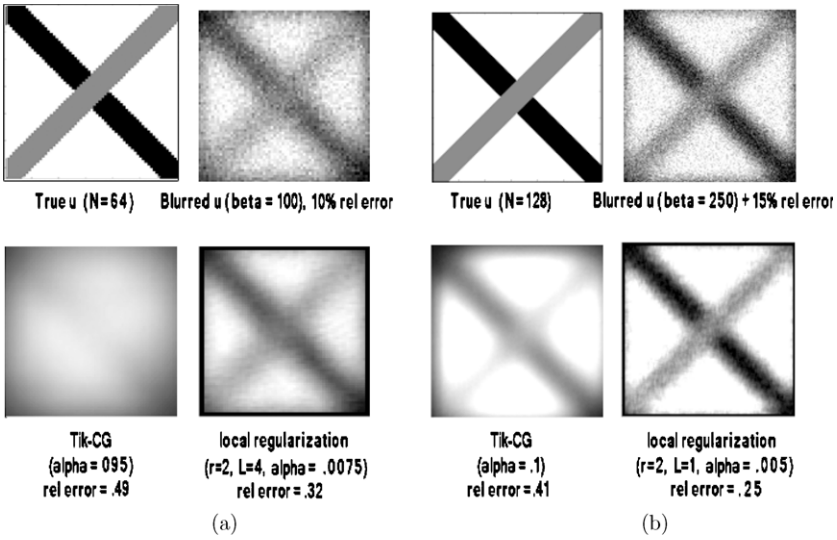


**Figure 3.** Example 4.3. (a) $N = 64$, $\beta = 100$, $\delta = 10\%$. (b) $N = 128$, $\beta = 250$, $\delta = 15\%$.

$\boldsymbol{i}_r = [2\ 2]$. For both Tik-CG and local Tikhonov, computation times are a little more than doubled over those of figure 3(a). In the case that a Tik-CG solution was used as an initial state for local Tikhonov iteration, the single computation took 8.16 s.

**Example 4.4.** In this case we have constructed a true solution like the simulated satellite image found in [26]. In [26], this image was blurred in a manner that models the atmospheric distortion found in images from outer space taken with ground-based telescopes. Such realism comes at the price of producing blurs using a discretized blurring operator and matrix multiplication, so we have chosen instead to generate blurred data $f^\delta$ using the integral operator and Gaussian kernel, as was done in all prior examples. In figure 4(a) $\beta = 100$, and relative noise of 10% has been added to this 64-by-64 image. For Tik-CG we have graphed the 23rd iterate, employing best $\alpha = 0.004$, which took about 0.6 s of computation time. For local Tikhonov, the computation time was about 25 s to get through the fourth iterate (shown), using best parameters $\alpha = 0.002$, $L = 4$ and $\boldsymbol{i}_r = [3\ 3]$.

The true image in figure 4(b) is 128-by-128. Here, $\beta = 250$ was used in the blurring operator.

In all but example 4.4, it seems that local Tikhonov outperforms Tik-CG, often quite markedly, at a cost of requiring greater computation time. The image of example 4.4 is somewhat different from the others in that it has a black background. Nevertheless, local Tikhonov performs almost as well. Example 4.3 is interesting in that it is not visually obvious from the blurred image that the background should be white instead of grey. In figure 3(b) it may be argued that Tik-CG does a better job in restoring the background to white (which it does by blowing-out white values), but local Tikhonov produces the more visually satisfying rendering of the original.

**Example 4.5.** Finally, in this example we illustrate the *a priori* parameter choice rule given in theorem 3.1 as applied to the image given in example 4.1. Using $\hat{\delta} = 85$, $\hat{\alpha} = .032$, and $\hat{\boldsymbol{r}} = (4, 4)$, we define the sequences

$$\delta_k = 2^{-k}\hat{\delta}, \quad \alpha_k = 2^{-k}\hat{\alpha}, \qquad \boldsymbol{r}_k = 2^{-k/2}\hat{\boldsymbol{r}}, \qquad \text{for} \quad k = 0, 1, \ldots,$$

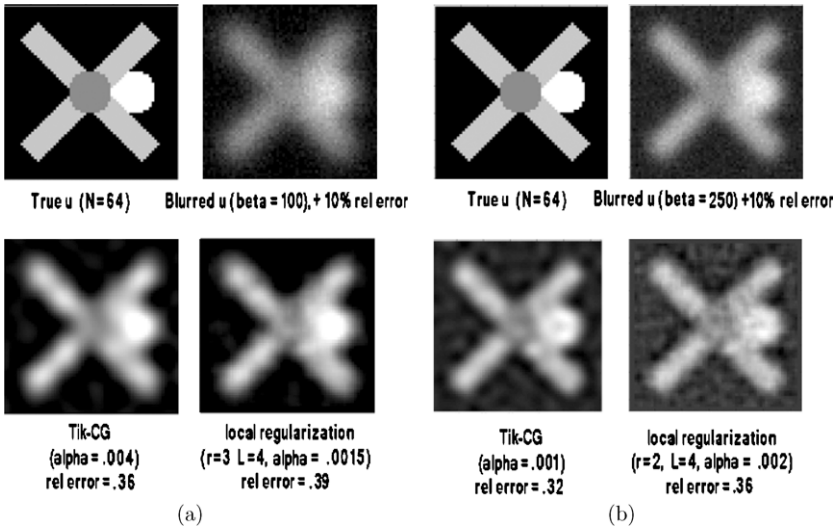


**Figure 4.** Example 4.4. (a) $N = 64$, $\beta = 100$, $\delta = 10\%$. (b) $N = 128$, $\beta = 250$, $\delta = 10\%$.

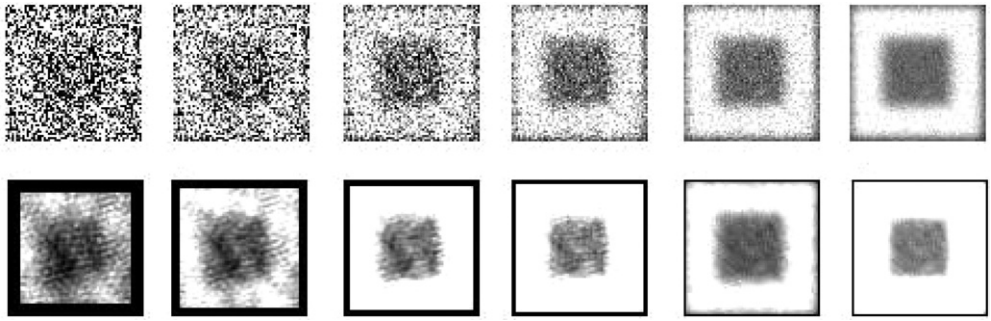

**Figure 5.** Example 4.5.

so that $\delta_k$, $\|\boldsymbol{r}_k\|$, $\alpha_k \to 0$ as $k \to \infty$. Further, as required by theorem 3.1, we have $\delta_k^2/\alpha_k \to 0$ as $k \to 0$ and $\|\boldsymbol{r}_k\|^2/\delta_k$ remains bounded as $k \to \infty$. In the top row of figure 5 we show the blurred images $f^{\delta_k}$ for $k = 0, 1, \ldots, 5$ (left to right), where $\|f - f^{\delta_k}\| \approx \delta_k$. In the bottom row of figure 5 we show the results of local regularization applied to the noisy data $f^{\delta_k}$ using regularization parameters $\alpha_k$, $\boldsymbol{r}_k$, for $k = 0, 1, \ldots, 5$ (left to right). As in example 1, we use $\beta = 100$ and $N = 64$ in each case; the true image may be found in figure 1. The black border around the deblurred images decreases as $k$ increases and $\boldsymbol{r}_k$ decreases, and is more evident in this example because of our choice to set unregularized areas to the value 0.0 (black); as discussed earlier, an alternate choice would have given less obvious bordering.

Finally, because of the discretization, the regularization parameter $\boldsymbol{r}_k$ in this example must be transformed (via rounding) to the *integer* pair $\boldsymbol{i}_{\boldsymbol{r}_k}$, with the limitation (again due to the discretization) that $\boldsymbol{i}_{\boldsymbol{r}_k} = [1\ 1]$ for $k \geqslant 4$. This limitation means that over-regularization actually occurs in the discretizations of images for $k > 5$ since $\boldsymbol{i}_{\boldsymbol{r}_k} = [1\ 1]$ is too large for its non-integer counterpart $\boldsymbol{r}_k$. (In fact, this over-regularization is already starting to be seen in the bottom right image in figure 5, the case of $k = 5$.)

## 5. Conclusion

We have extended the ideas of [20] for the local regularization of 1D integral equations in such a way as to facilitate a theory of local regularization for general 2D integral equations. Our convergence theory shows that regularized solutions converge to the true solution of the original problem with appropriate choices of the regularization parameters $r$ and $\alpha$ as the level $\delta$ of noise goes to zero. Numerical implementation illustrates that in many examples local regularization out-performs other regularization methods based on differentiable optimization schemes when it comes to relative errors in solutions and perceived sharpness/resolution of images. Future extensions of this work include the problem of selection of regularization parameters, a difficult problem in local regularization. Indeed, only recently has this question been answered for the problem of selection of the parameter $r$ in the 1D Volterra problem [3], so there is hope for the extension of these ideas to the 2D non-Volterra problem. Even more challenging is the subject of the selection of a *variable* regularization parameter $\alpha$, also the subject of future study.

## Acknowledgments

This work was supported for the second author in part by the National Science Foundation under contract numbers NSF DMS-0104003 and DMS-0405978. Parts of the research were undertaken while the second author was in residence at the Institute for Pure and Applied Mathematics (IPAM) at UCLA and the third author was in residence at the Institute for Mathematics and its Applications (IMA) at the University of Minnesota. We wish to thank both institutes for providing the financial support and hospitable environment conducive to productive research. Finally, the third author wishes to acknowledge the support of Calvin College for a research fellowship which greatly facilitated the completion of this work.

## References

[1] Acar R and Vogel C R 1994 Analysis of bounded variation penalty methods for ill-posed problems *Inverse Problems* **10** 1217–29
[2] Bertero M and Boccacci P 1998 *Introduction to Inverse Problems in Imaging* (Bristol: Institute of Physics Publishing)
[3] Brooks C D and Lamm P K 2007 A discrepancy principle for parameter selection in the local regularization of linear Volterra inverse problems *Preprint*
[4] Chambolle A and Lions P-L 1997 Image recovery via total variation minimization and related problems *Numer. Math.* **76** 167–88
[5] Chalmond B 2003 *Modeling and Inverse Problems in Imaging Analysis* (*Applied Mathematical Sciences* vol 155) (New York: Springer)
[6] Cui C 2005 Local regularization methods for $n$-dimensional first-kind integral equations *PhD Thesis* Department of Mathematics, Michigan State University
[7] Cinzori A and Lamm P K 2000 Future polynomial regularization of ill-posed Volterra equations *SIAM J. Numer. Anal.* **37** 949–79
[8] Dai Z and Lamm P K 2007 Local regularization for the nonlinear autoconvolution equation *SIAM J. Numer. Anal.* submitted
[9] Daubechies I, Defrise M and De Mol C 2004 An iterative thresholding algorithm for linear inverse problems with a sparsity constraint *Commun. Pure Appl. Math.* **57** 1413–57
[10] De Mol C and Defrise M 2002 A note on wavelet-based inversion algorithms *Contemp. Math.* **313** 85–96
[11] Dobson D C and Vogel C R 1997 Convergence of an iterative method for total variation denoising *SIAM J. Numer. Anal.* **34** 1779–971
[12] Dobson D C and Santosa F 1996 Recovery of blocky images from noisy and blurred data *SIAM J. Appl. Math.* **56** 1181–98
[13] Engl H W, Hanke M and Neubauer A 1996 *Regularization of Inverse Problems* (Dordrecht: Kluwer)

[14] Groetsch C W 1984 *The Theory of Tikhonov Regularization for Fredholm Equations of the First Kind* (Boston, MA: Pitman)
[15] Hanke M and Nagy J G 1994 Toeplitz approximate inverse preconditioner for banded Toeplitz matrices *Numer. Algorithms* **7** 183–99
[16] Hansen P C Manual for *Regularization Tools Version 3.1*, a MATLAB package for analysis and solution of discrete ill-posed problems (http://www2.imm.dtu.dk/~pch/Regutools/)
[17] Lamm P K 2000 A survey of regularization methods for first-kind Volterra equations *Surveys on Solution Methods for Inverse Problems* ed D Colton, H W Engl, A Louis, J R McLaughlin and W Rundell (New York: Springer) pp 53–82
[18] Lamm P K 2005 Full convergence of sequential local regularization methods for Volterra inverse problems *Inverse Problems* **21** 785–803
[19] Lamm P K 1995 Future-sequential regularization methods for ill-posed Volterra equations: Applications to the inverse heat conduction problem *J. Math. Anal. Appl.* **195** 469–94
[20] Lamm P K 2003 Variable-smoothing local regularization methods for first-kind integral equations *Inverse Problems* **19** 195–216
[21] Lamm P K and Dai Z 2005 On local regularization methods for linear Volterra equations and nonlinear equations of Hammerstein type *Inverse Problems* **21** 1773–90
[22] Lamm P K and Eldén L 1997 Numerical solution of first-kind Volterra equations by sequential Tikhonov regularization *SIAM J. Numer. Anal.* **34** 1432–50
[23] Lamm P K and Scofield T L 2000 Sequential predictor-corrector methods for the variable regularization of Volterra inverse problems *Inverse Problems* **16** 373–99
[24] Lamm P K and Scofield T L 2001 Local regularization methods for the stabilization of linear ill-posed problems of Volterra type *Numer. Funct. Anal. Opt.* **22** 913–40
[25] Rudin L I, Osher S and Fatemi E 1992 Nonlinear total variation based noise removal algorithms *Physica* D **60** 259–68
[26] Vogel C 2002 *Computational Methods for Inverse Problems* (Philadelphia, PA: SIAM) pp 65–92