The PP-TSVD Algorithm for Image Restoration Problems

Per Chr. Hansen, Michael Jacobsen, Jan M. Rasmussen, and Heino Sørensen

Department of Mathematical Modelling, Technical University of Denmark
Building 321, DK-2800 Lyngby, Denmark
Email: `pch@imm.dtu.dk`
Home page: `http://www.imm.dtu.dk/~pch`

Abstract. The PP-TSVD algorithm is a regularization algorithm based on the truncated singular value decomposition (TSVD) that computes piecewise polynomial (PP) solutions without any a priori information about the locations of the break points. Here we describe an extension of this algorithm designed for two-dimensional inverse problems based on a Kronecker-product formulation. The 2-D version of the PP-TSVD algorithm is formulated such that it is suited for large-scale problems. We illustrate its use in connection with deblurring of digital images with sharp edges.

1 Introduction

In this work we focus on discretizations of linear inverse problems in the form of square systems $A\,z = b$ or overdetermined least squares systems $\min \|A\,z - b\|_2$. These systems, which we denote *discrete ill-posed problems*, represent a wealth of applications of inverse problems; see, e.g., [6, §1.2].

Our main "tool" from linear algebra for analysis as well as computations is the singular value decomposition (SVD) of the coefficient matrix A. If we assume that A is $m \times n$ with $m \geq n$, then the SVD takes the form

$$A = \sum_{i=1}^{n} u_i\,\sigma_i\,v_i^T\,, \tag{1}$$

where u_i and v_i are the left and right singular vectors which are orthonormal, and σ_i are the singular values which are nonnegative and appearing in non-decreasing order. In terms of the SVD, discrete ill-posed problems are characterized by having a coefficient matrix A whose singular values decay gradually to zero (in practice: until they hit a level determined by the machine precision).

Standard methods for regularization of discrete ill-posed problems are *Tikhonov regularization*

$$\min\left\{\|A\,z - b\|_2^2 + \lambda^2\,\|z\|_2\right\} \tag{2}$$

and *truncated SVD* (TSVD)

$$\min \|z\|_2 \quad \text{subject to} \quad \|A_k\,z - b\|_2 = \min, \tag{3}$$

where A_k is the TSVD matrix of rank k given by

$$A_k = \sum_{i=1}^{k} u_i\, \sigma_i\, v_i^T\,. \tag{4}$$

It is well know that the Tikhonov and TSVD solutions are smooth, in the sense that they are continuous and tend to represent minimum energy. By replacing the norm $\|z\|_2$ in the above equations with the seminorm $\|L_p\, z\|_2$, where the matrix L_p is a discrete approximation to the pth derivative operator, we can produce solutions with, say, maximum flatness (for $p = 1$) or minimum roughness (for $p = 2$) — but the solutions remain continuous and smooth.

While smooth solutions are desirable in many applications, they are undesirable in other applications where the solutions are known to have discontinuities or steep gradients. Hence, other regularization algorithms must be used to compute such regularized solutions. One approach is to replace the 2-norm $\|z\|_2$ in Tikhonov's method with the norm $\|L_1\, z\|_1$, i.e., the 1-norm of the first spatial derivative of the solution. This is called *total variation* (TV) regularization, and it is able to produce solutions with very steep gradients; see [14] for details. Due to the non-differentiability of the 1-norm, specialized optimization algorithms [15] must be used to compute the TV solutions.

A different approach, which is a modification of the TSVD method and which is based solely on tools from linear algebra, was proposed in [8] and [9]. The key idea is to replace the 2-norm $\|z\|_2$ in the TSVD method with the seminorm $\|L_p\, z\|_1$, where L_p is again an approximation to the pth derivative operator:

$$\min \|L_p\, z\|_1 \quad \text{subject to} \quad \|A_k\, z - b\|_2 = \min\,. \tag{5}$$

The change to the 1-norm has a dramatic effect on the computed solutions. As proved in [9], the solutions to (5) consist of polynomial pieces, and the degree of the polynomials is $p-1$. Moreover, the number of break points is at most $k-p$, where k is the TSVD truncation factor in (4). The method is called the *PP-TSVD method*, and it has been used, e.g., in helioseismology [3] and inversion of gravity and magnetic data [11].

The original PP-TSVD algorithm was specifically developed for 1-D problems, and the extension to 2-D problems may not be obvious. The purpose of this work is to describe how the PP-TSVD is extended to treat 2-D inverse problems. Along this line, we also discuss important implementation details necessary to make the 2-D version useful for problems of realistic size. First we discuss some general issues in the treatment of discretizations of 2-D inverse problems in §2. Then we describe various discrete techniques for computing the 1-norm of the derivatives of the solution in §3. Next, in §§4–5 we describe the PP-TSVD algorithm and its implementation for large-scale problems. Various properties of 1-D and 2-D PP-TSVD solutions are discussed in §6, and finally we present some small image reconstructions in §7.

2 Discretization of 2-D Problems

In order to simplify our presentation, we limit our discussion to *2-D deconvolution problems whose variables separate*, i.e., we work with a 2-D first-kind Fredholm integral equation of the generic form

$$\int_0^1 \int_0^1 \kappa(x - x')\, \omega(y - y')\, f(x', y')\, dx'\, dy' = g(x, y), \tag{6}$$

where κ and ω are functions. An example of such a problem is image deblurring with a Gaussian point spread function, for which

$$\kappa(t) = \omega(t) = \frac{1}{\sqrt{2\pi}\sigma} \exp\left(-\frac{1}{2}\left(\frac{t}{\sigma}\right)^2\right),$$

and which is used as a model for out-of-focus blur as well as atmospheric turbulence blur [1].

Assume now, also for ease of presentation, that we use the midpoint quadrature rule to discretize the integral equation. Thus, we approximate the integral over y' with a sum of n terms,

$$\int_0^1 \omega(y - y')\, f(x', y')\, dy' \approx n^{-1} \sum_{i=1}^{n} \omega(y - y'_\ell)\, \tilde{f}(x', y'_\ell) = \phi(x', y),$$

where y'_ℓ are the quadrature points and $\tilde{f}$ represents the approximate solution that we compute. Next, we approximate the integral over x' with another sum,

$$\int_0^1 \kappa(x - x')\, \phi(x', y)\, dx' \approx n^{-1} \sum_{i=1}^{n} \kappa(x - x'_k)\, \phi(x'_k, y) = \psi(x, y),$$

where x'_k are also quadrature points. Finally, we use collocation in the n^2 points (x_i, y_j),

$$\psi(x_i, y_j) = g(x_i, y_j), \qquad i, j = 1, \ldots, n,$$

which eventually will lead to a system of n^2 linear equations in the n^2 unknowns $\tilde{f}(x_i, y_j)$.

To derive this system we introduce the following four $n \times n$ matrices A, $\bar{A}$, F and G with elements

$$A_{ij} = n^{-1}\kappa(x_i - x'_k), \qquad \bar{A}_{j\ell} = n^{-1}\omega(y_j - h'_\ell)$$

$$F_{k\ell} = \tilde{f}(x'_k, y'_\ell), \qquad G_{ij} = g(x_i, y_j),$$

where all indices are in the range $1, \ldots, n$. Note that A and $\bar{A}$ consist of samples of the functions κ and ω, respectively, while F and G consist of samples of the approximate solution $\tilde{f}$ and the right-hand side g, respectively.

We now define an $n \times n$ matrix Φ that corresponds to the y'-integration, with elements

$$\Phi_{kj} = \phi(x'_k, y_j) = n^{-1} \sum_{\ell=1}^{n} \omega(y_j - y'_\ell)\, \tilde{f}(x'_k, y'_\ell), \qquad j,k = 1,\ldots,n,$$

and by carefully studying the indices of the above expression it follows that Φ can be written as

$$\Phi = F\, \bar{A}^T .$$

Similarly, we define an $n \times n$ matrix Ψ that corresponds to the x'-integration, with elements

$$\Psi_{ij} = \psi(x_i, y_j) = n^{-1} \sum_{k=1}^{n} \kappa(x_i - x'_k)\, \phi(x'_k, y_j), \qquad i,k = 1,\ldots,n,$$

and this matrix can be written as

$$\Psi = A\, \Phi = A\, F\, \bar{A}^T .$$

Collocation now corresponds to the requirement $\Psi = G$. We have thus shown that the discretization of the 2-D deconvolution problem leads to the linear relation $A\, F \bar{A}^T = G$ between the discrete solution F and the discrete data G.

Finally, to arrive at a standard system of linear equations, we introduce the Kronecker product and the "vec" notation. The Kronecker product $\bar{A} \otimes A$ of two $n \times n$ matrices A and $\bar{A}$ is defined as the $n^2 \times n^2$ matrix

$$\bar{A} \otimes A = \begin{pmatrix} \bar{a}_{11}A & \bar{a}_{12}A & \cdots & \bar{a}_{1n}A \\ \bar{a}_{21}A & \bar{a}_{22}A & \cdots & \bar{a}_{2n}A \\ \vdots & \vdots & & \vdots \\ \bar{a}_{n1}A & \bar{a}_{n2}A & \cdots & \bar{a}_{nn}A \end{pmatrix}$$

If X is an $n \times n$ matrix with column partitioning $X = (x_1, \ldots, x_n)$, then we define the vector $\mathrm{vec}(X)$ of length n^2 as

$$\mathrm{vec}(X) = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix} .$$

The Kronecker product and the "vec" notation are connected via the following important relation

$$\left(\bar{A} \otimes A\right)\, \mathrm{vec}(X) = \mathrm{vec}\left(A\, X\, \bar{A}^T\right) . \tag{7}$$

Thus we see that the discretized linear system that we derived above can be written in the following two alternative forms

$$A\, F\, \bar{A}^T = G \qquad \Longleftrightarrow \qquad \left(\bar{A} \otimes A\right)\, \mathrm{vec}(F) = \mathrm{vec}(G). \tag{8}$$

The rightmost form is a conventional system of linear algebraic equations with an $n^2 \times n^2$ structured coefficient matrix. It is by applying the original PP-TSVD algorithm to this problem that we shall derive the 2-D version of the algorithm.

3 Derivative Operators

In this section we derive some useful expressions for discretizations of derivative operators. We start with 1-D problems and a function $f = f(x)$ given in the interval $[0,1]$. Moreover, we assume that the n-vector z consists of equidistant samples of the function f, e.g., $z_i = f(ih)$, $i = 1,\ldots,n$, where $h = 1/n$ is the grid spacing. Then approximations to discretizations of the first and second derivative f' and f'' are given by $h^{-1}L_1 z$ and $h^{-2}L_2 z$, respectively, where the two matrices L_1 and L_2 are given by

$$L_1 = \begin{pmatrix} -1 & 1 & & \\ & \ddots & \ddots & \\ & & -1 & 1 \end{pmatrix}, \qquad L_2 = \begin{pmatrix} 1 & -2 & 1 & & \\ & \ddots & \ddots & \ddots & \\ & & 1 & -2 & 1 \end{pmatrix}, \tag{9}$$

and where L_1 is $(n-1)\times n$ and L_2 is $(n-2)\times n$. These definitions of L_1 and L_2 ensure that the two matrices have nontrivial null spaces whose basis vectors are exact discretizations of the functions that span the null spaces of the derivative operators:

$$L_1 \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = 0, \qquad L_2 \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} = 0, \qquad L_2 \begin{pmatrix} 1 \\ \vdots \\ n \end{pmatrix} = 0.$$

Then it follows that a numerical approximation to the 1-norm of f can be computed as

$$\|f\|_1 = \int_0^1 |f(x)|\, dx \simeq h\|z\|_1$$

while numerical approximations to the 1-norm of the derivatives f' and f'' are given by

$$\|f'\|_1 \simeq \|L_1 z\|_1, \qquad \|f''\|_1 \simeq h^{-1}\|L_2 z\|_1.$$

For 2-D problems, we assume that the matrix F consists of samples of an underlying function $f = f(x,y)$ in the domain $[0,1]\times[0,1]$, sampled on a regular $n \times n$ mesh with grid spacing $h = 1/n$ in both directions. Using the matrices L_1 and L_2 defined in (9) we then obtain the following numerical approximation to the 1-norm of f:

$$\|f\|_1 = \int_0^1 \int_0^1 |f(x,y)|\, dx\, dy \simeq h^2 \sum_{i=1}^n \sum_{j=1}^n |f_{ij}| = h^2 \|\mathrm{vec}(F)\|_1$$

while numerical approximations to the 1-norms of the first and second partial derivatives are given by

$$\left\| \frac{\partial f}{\partial x} \right\|_1 \simeq h\|\mathrm{vec}(F\, L_1^T)\|_1 = h\|(L_1 \otimes I)\mathrm{vec}(F)\|_1$$

$$\begin{aligned}
\left\|\frac{\partial f}{\partial y}\right\|_1 &\simeq h\|\mathrm{vec}(L_1\, F)\|_1 = h\|(I \otimes L_1)\mathrm{vec}(F)\|_1 \\
\left\|\frac{\partial^2 f}{\partial x^2}\right\|_1 &\simeq \|\mathrm{vec}(F\, L_2^T)\|_1 = \|(L_2 \otimes I)\mathrm{vec}(F)\|_1 \\
\left\|\frac{\partial^2 f}{\partial y^2}\right\|_1 &\simeq \|\mathrm{vec}(L_2\, F)\|_1 = \|(I \otimes L_2)\mathrm{vec}(F)\|_1 \\
\left\|\frac{\partial^2 f}{\partial x \partial y}\right\|_1 &\simeq \|\mathrm{vec}(L_1\, F\, L_1^T)\|_1 = \|(L_1 \otimes L_1)\mathrm{vec}(F)\|_1 .
\end{aligned}$$

Here and throughout the manuscript, I denotes the $n \times n$ identity matrix. Another useful result concerns the sum of 1-norms:

$$\left\|\frac{\partial f}{\partial x}\right\|_1 + \left\|\frac{\partial f}{\partial y}\right\|_1 \simeq h \left\| \begin{pmatrix} L_1 \otimes I \\ I \otimes L_1 \end{pmatrix} \mathrm{vec}(F) \right\|_1 \tag{10}$$

$$\left\|\frac{\partial^2 f}{\partial x^2}\right\|_1 + \left\|\frac{\partial^2 f}{\partial y^2}\right\|_1 \simeq \left\| \begin{pmatrix} L_2 \otimes I \\ I \otimes L_2 \end{pmatrix} \mathrm{vec}(F) \right\|_1 . \tag{11}$$

These relations follow immediately from the definition of the vector 1-norm.

The matrices L_1 and L_2 can also be used to compute other quantities that involve derivatives. For example, to approximate $\nabla^2 f = \frac{\partial^2 f}{\partial x^2} + \frac{\partial^2 f}{\partial y^2}$ we can first compute approximations $L_2\, F$ and $F\, L_2^T$ to the second partial derivatives. However, we cannot immediately add these two matrices because their dimensions are incompatible (they are $(n-2) \times n$ and $n \times (n-2)$, respectively). One solution is to "peel off" the first and last columns of $L_2\, F$ and the first and last rows of $F\, L_2^T$, and then add the two $(n-2) \times (n-2)$ matrices.[1]

Another useful quantity that we can approximate is the total variation (TV) of f, defined as

$$J_{\mathrm{TV}}(f) = \int_0^1 \int_0^1 \left(\left(\frac{\partial f}{\partial x}\right)^2 + \left(\frac{\partial f}{\partial y}\right)^2 \right)^{1/2} dx\, dy. \tag{12}$$

The quantity $\left((\partial f/\partial x)^2 + (\partial f/\partial y)^2\right)^{1/2}$ is known as the gradient magnitude in the image processing literature [13, §4.3.2]. The TV functional gives information about the discontinuities in the image. For example, if f takes on a constant value in a region Ω_0 and takes on another constant value on the complementary of Ω_0, then $J_{\mathrm{TV}}(f)$ is the length of the boundary of Ω_0 multiplied by the magnitude of the jump [14]. Various aspects of the TV function and TV regularization are discussed in [4] and [14]. To approximate $J_{\mathrm{TV}}(f)$, define the $(n-1) \times n$ matrix

$$M = \frac{1}{2}\,|L_1| = \frac{1}{2} \begin{pmatrix} 1 & 1 & & \\ & \ddots & \ddots & \\ & & 1 & 1 \end{pmatrix}$$

[1] The resulting matrix is identical, except for a scaling factor, to the inner points in the matrix computed by means of Matlab's del2 function.

and the two $(n-1) \times (n-1)$ matrices

$$A_x = M\,F\,L_1^T, \qquad A_y = L_1 F\,M^T.$$

Then we have found experimentally that a reasonable numerical approximation to the TV of f is given by

$$J_{\mathrm{TV}}(F) = \sum_{i=1}^{n}\sum_{j=1}^{n}\left((A_x)_{ij}^2 + (A_y)_{ij}^2\right)^{1/2}.$$

We note that some authors choose to approximate $J_{\mathrm{TV}}(f)$ by

$$\widetilde{J}_{\mathrm{TV}}(f) = \int_0^1\int_0^1\left(\left|\frac{\partial f}{\partial x}\right| + \left|\frac{\partial f}{\partial y}\right|\right) dx\,dy = \left\|\frac{\partial f}{\partial x}\right\|_1 + \left\|\frac{\partial f}{\partial y}\right\|_1 \tag{13}$$

which is linear in the two derivatives and which, in turn, is approximated numerically by the quantity in Eq. (10).

4 2-D PP-TSVD Regularization Algorithms

Having realized that discretized 2-D problems have the same general structure as 1-D problems, cf. (8), it is obvious that we can derive 2-D versions of all the 1-D regularization algorithms simply by replacing the coefficient matrix with the Kronecker product $\bar{A} \otimes A$, the right-hand side with $\mathrm{vec}(G)$, and the solution with $\mathrm{vec}(F)$. Moreover, in the constraints, we replace the solution's seminorm with one of the functionals defined in the previous section. Of particular interest here are the functionals that can be written in terms of a matrix times $\mathrm{vec}(F)$. This excludes the TV function, but not the approximation in (10).

Thus, we define the 2-D version of the PP-TSVD method as follows:

$$\min \|\mathcal{L}\,\mathrm{vec}(F)\|_1 \qquad \text{subject to} \qquad \|\mathcal{A}_k\,\mathrm{vec}(F) - \mathrm{vec}(G)\|_2 = \min, \tag{14}$$

where the matrix $\mathcal{A}_k$ is the truncated SVD of the Kronecker product $\bar{A} \otimes A$, and the matrix $\mathcal{L}$ is one of the matrices

$$I \otimes I, \qquad \begin{pmatrix} L_p \otimes I \\ I \otimes L_p \end{pmatrix}, \quad p = 1, 2, \ldots\,. \tag{15}$$

To obtain more insight into the 2-D PP-TSVD method, we use the fact that the SVD of the Kronecker product $\bar{A} \otimes A$ can be conveniently expressed in terms of the SVDs of the two matrices. Assume that the SVDs of A and $\bar{A}$ are given by

$$A = U\,\Sigma\,V^T = \sum_{i=1}^{n} u_i\,\sigma_i\,v_i^T, \qquad \bar{A} = \overline{U}\,\overline{\Sigma}\,\overline{V}^T = \sum_{i=1}^{n} \overline{u}_i\,\overline{\sigma}_i\,\overline{v}_i^T.$$

Then it follows from the properties of Kronecker products that

$$\begin{aligned}\bar{A} \otimes A &= \left(\overline{U} \otimes U\right)\left(\overline{\Sigma} \otimes \Sigma\right)\left(\overline{V} \otimes V\right)^T \\ &= \sum_{i=1}^{n}\sum_{i=1}^{n}\left(\overline{u}_i \otimes u_i\right)\left(\overline{\sigma}_i\,\sigma_i\right)\left(\overline{v}_i \otimes v_i\right)^T,\end{aligned} \tag{16}$$

which, except for the ordering, constitutes the SVD of $\bar{A} \otimes A$. Hence, the matrix $\mathcal{A}_k$ is given by

$$\mathcal{A}_k = \sum_{(i,j)\in\mathcal{K}}^{n} (\overline{u}_i \otimes u_j)\ (\overline{\sigma}_i\, \sigma_j)\ (\overline{v}_i \otimes v_j)^T ,$$

where $\mathcal{K}$ denotes the set of indices (i,j) corresponding to the largest k values of $\overline{\sigma}_i\, \sigma_j$.

We can also use expression (16) to define a slightly different 2-D PP-TSVD method, namely, by replacing $\mathcal{A}_k$ with the matrix

$$\widehat{\mathcal{A}}_\ell = \sum_{i=1}^{\ell}\sum_{j=1}^{\ell} (\overline{u}_i \otimes u_j)\ (\overline{\sigma}_i\, \sigma_j)\ (\overline{v}_i \otimes v_j)^T = \bar{A}_\ell \otimes A_\ell,$$

obtained by including the largest ℓ singular values of A and $\bar{A}$. Using the relation $\|\mathrm{vec}(F)\|_2 = \|F\|_\mathrm{F}$, this variation of the PP-TSVD method takes the form

$$\min \|\mathcal{L}\, \mathrm{vec}(F)\|_1 \qquad \text{subject to} \qquad \|A_\ell\, F\, \bar{A}_\ell^T - G\|_\mathrm{F} = \min . \tag{17}$$

Both versions of the 2-D PP-TSVD method yield regularized solutions; they differ in the way that they select which SVD components to be included in the solution.

5 Large-Scale Implementation

The original 1-D version of the PP-TSVD algorithm was based on the fact that the PP-TSVD solution $z_{L,k}$ can be written as

$$z_{L,k} = z_k - V_k^o w_k, \qquad z_k = \sum_{i=1}^{k} \frac{u_i^T b}{\sigma_i}\, v_i, \tag{18}$$

where z_k is the TSVD solution, i.e. the solution to (3). Moreover, the matrix $V_k^o = (v_{k+1}, \ldots, v_n)$ consists of the last $n-k$ right singular vectors, and the vector w_k is the solution to the linear ℓ_1-problem

$$\min \|(L\, V_k^o)\, w - L\, z_k\|_1 . \tag{19}$$

This algorithm thus requires two main computations: the *full SVD* of the matrix A and the solution to the unconstrained ℓ_1-problem. Note that this formulation gives insight into the PP-TSVD solution: its first k SVD components are identical to the TSVD components while its last $n-k$ SVD components, which lie in the null space of A_k, are chosen so as to minimize the 1-norm of the vector $L\, z$.

Although convenient for small- and medium-size problems, the original algorithm is not suited for large-scale problems where it is inconvenient, or impossible, to compute the full SVD — even if the Kronecker product formulation is used. Large-scale SVD algorithms, such as those based on Lanczos bidiagonalization, compute only (approximations to) the principal SVD components. Thus, we need an algorithm that avoids the explicit use of the matrix V_k^o.

The solution to this problem is to replace the unconstrained ℓ_1-problem with a related linearly constrained ℓ_1-problem. Let $V_k = (v_1, \ldots, v_k)$ consist of the first k right singular vectors, and rewrite the PP-TSVD solution as

$$z_{L,k} = z_k - y_k, \tag{20}$$

where $y_k = V_k^o w_k$ and therefore lies in the range of V_k^o. Hence y_k is orthogonal to the range of V_k, and it follows that y_k can be computed as the solution to the linearly constrained ℓ_1-problem

$$\min \|L\, y - L\, z_k\|_1 \qquad \text{subject to} \qquad V_k^T\, y = 0. \tag{21}$$

We have implemented this approach in Matlab in the function pptsvd, which is available via the home page[2] for the REGULARIZATION TOOLS package [5]. The principal SVD components are computed by means of the built-in Matlab function svds (which uses Lanczos bidiagonalization with implicit restarts), and to solve the linearly constrained ℓ_1-problem we implemented the classical algorithm by Barrodale and Roberts [2].

In a future project, we wish to explore the possibilities for using the Lanczos vectors of the bidiagonalization process without transforming them into approximate SVD vectors. Specifically, assume that the mth iteration of the Lanczos bidiagonalization process provides the three matrices $U^{(m)}$, $B^{(m)}$, and $V^{(m)}$ such that

$$A\, V^{(m)} = U^{(m)}\, B^{(m)},$$

where $U^{(m)}$ and $V^{(m)}$ have orthonormal columns and $B^{(m)}$ is bidiagonal. Then the iterative LSQR algorithm [12], which is based on Lanczos bidiagonalization, computes a solution which is formally given by

$$\tilde{z}_m = V^{(m)} \left(B^{(m)}\right)^\dagger \left(U^{(m)}\right)^T b,$$

and which can be considered as a (rough) approximation to a TSVD solution z_k for some k [6, §6.4]. Moreover, for the same k the columns of $V^{(m)}$ span an approximation to the subspace spanned by the first k right singular vectors $v_1, \ldots .v_k$. Hence, it is worth exploring the possibility of replacing z_k and V_k in (20) and (21) with $\tilde{z}_m$ and $V^{(m)}$.

It is also worth exploring the possibilities for using more recent and faster algorithms for solving the constrained ℓ_1-problem in (21), as well as exploring whether information from previous values of k can be used to warm-start the algorithm when the solution for a new k-value is computed.

[2] The URL is `http://www.imm.dtu.dk/~pch/Regutools/regutools.html`.

6 Properties of the 2-D PP-TSVD Solutions

For 1-D problems it is easy to quantify the appearance of the PP-TSVD solutions $z_{L,k}$, see [9, §2.2] for details. A key observation is that the vector $L_p\, z_{L,k}$ is precisely the residual vector in the linear ℓ_1-problem (19), and this residual vector will have many zero elements. The maximum number of nonzero elements equals the difference between the row and column dimensions of the coefficient matrix in (19). Hence, if the $(n-p) \times n$ matrix L_p is an approximation to the $p$the derivative operator and if $k > p$, then $L_p\, z_{L,k}$ will have at most $k-p$ nonzero elements. And since $z_{L,k}$ can be considered as samples of the pth integral of the function represented by $L_p\, z_{L,k}$, it follows that $z_{L,k}$ itself represents a piecewise polynomial of degree $p-1$ with at most $k-p$ break points. If $k < p$ then the ℓ_1-problem is underdetermined and $z_{L,k}$ is identical to the MTSVD solution [10] which is known to be continuous and smooth.

Consider now the 2-D problems and assume that $\mathcal{L}$ is one of the matrices in Eq. (15) with L_p being an approximation to the pth derivative operator. Then the coefficient matrix $\mathcal{L}\, V_k^o$ in (19) is either $n^2 \times (n^2-k)$ for $p=0$, or $2n(n-p) \times (n^2-k)$ for $p>0$, ensuring that the ℓ_1-problem is overdetermined, and the number of nonzero entries in the residual vector $\mathcal{L}\, z_{\mathcal{L},k}$ corresponding to the PP-TSVD solution $z_{\mathcal{L},k}$ is at most k for $p=0$, and at most $n(n-2p)+k$ for $p>0$. Our experience is that the number of nonzeros in the ℓ_1-residual vector is much smaller.

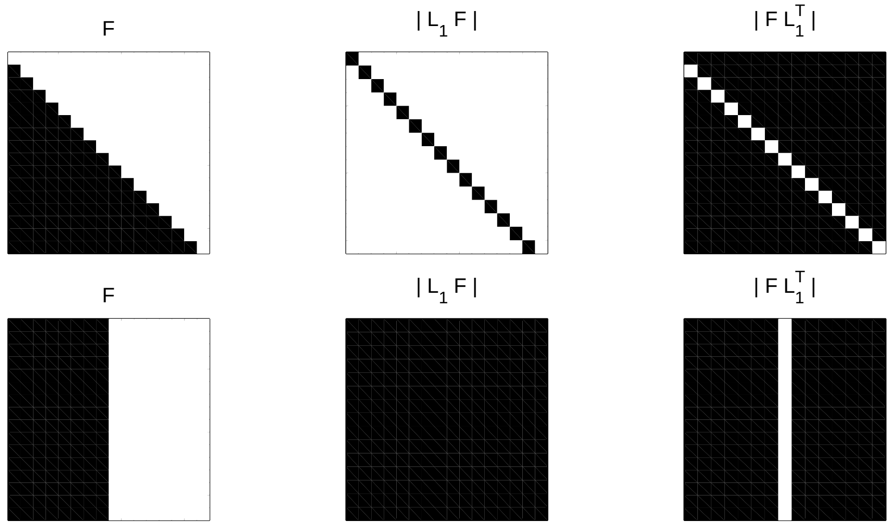

Fig. 1. A skew edge gives a higher value of $\|\mathcal{L}\,\mathrm{vec}(F)\|_1$ than a vertical (or horizontal) edge.

In order to illustrate this, consider the case $p = 1$ and $\mathcal{L}$ given by

$$\mathcal{L} = \begin{pmatrix} L_1 \otimes I \\ I \otimes L_1 \end{pmatrix} \tag{22}$$

which leads to regularized solutions that minimize the *approximate* TV functional $\widetilde{J}_{TV}(F)$, cf. (10) and (13). Clearly, horizontal or vertical edges in F lead to smaller values of

$$\|\mathcal{L}\,\mathrm{vec}(F)\|_1 = \|\mathrm{vec}(L_1\,F)\|_1 + \|\mathrm{vec}(F\,L_1^T)\|_1$$

than skew edges; for example, if

$$F = \mathsf{triu}(\mathsf{ones}(n))$$

then $\|\mathrm{vec}(L_1\,F)\|_1 = \|\mathrm{vec}(F\,L_1^T)\|_1 = n - 1$, while if

$$F = [\mathsf{zeros}(n, n/2), \mathsf{ones}(n, n/2)]$$

then $\|\mathrm{vec}(L_1\,F)\|_1 = 0$ and $\|\mathrm{vec}(F\,L_1^T)\|_1 = n$; see Fig. 1. Hence, $\|\mathcal{L}\,\mathrm{vec}(F)\|_1$ will be made small by constructing F as a blocky image consisting of horizontal or vertical rectangles with the same intensity — and such images are precisely the ones that are produced by the 2-D PP-TSVD algorithm with regularization term $\|\mathcal{L}\,\mathrm{vec}(F)\|_1$. And since $\|\mathcal{L}\,\mathrm{vec}(F)\|_1$ can be considered as an approximation to the TV of the image, we see that with this regularization term the PP-TSVD produces regularized solutions which are related to the TV regularized images (which are also found to be blocky, cf. [4]).

To take the illustration further, consider an 8×8 image F consisting of a white 3×4 rectangle on a black background, with $\|\mathcal{L}\,\mathrm{vec}(F)\|_1 = 14$. Now add Gaussian blurring to F and compute regularized images F_k by means of the 2-D PP-TSVD algorithm with $k = 1, \ldots, 16$. These 16 solutions are shown in Fig. 2 along with the corresponding values of $\|\mathcal{L}\,\mathrm{vec}(F_k)\|_1$. We see that as k increases, the image becomes more complex and $\|\mathcal{L}\,\mathrm{vec}(F_k)\|_1$ increases until the original image F is recovered for $k = 16$. We stress that no noise was added in this experiment — the purpose is solely to illustrate the reconstruction capabilities of the 2-D PP-TSVD algorithms with $\mathcal{L}$ given by (22).

It is interesting to inspect the 2-D PP-TSVD solutions more closely for various values of p, in order to understand how "piecewise polynomial" should be interpreted in two dimensions. Figure 3 shows surface plots of four 16×16 regularized solutions F corresponding to the four choices of $\mathcal{L}$ listed in Table 1. We see that for $\mathcal{L}$ equal to the $n^2 \times n^2$ identity matrix, the regularized solution F consists of delta functions, and for $\mathcal{L} = L_i \otimes L_i$, $i = 1, 2, 3$ each row or column of F is a piecewise polynomial of degree $i - 1$.

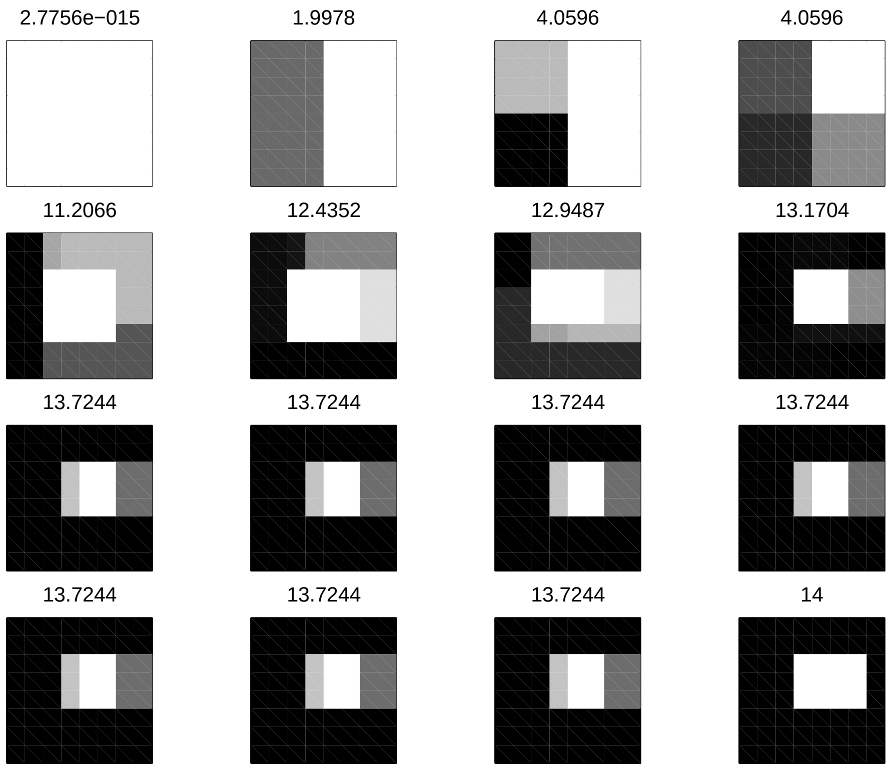

Fig. 2. PP-TSVD reconstructions F_k for $k = 1, \ldots, 16$. The numbers on top of each image are the corresponding values of $\|\mathcal{L}\,\mathrm{vec}(F_k)\|_1$ with $\mathcal{L}$ given by (22).

Matrix $\mathcal{L}$	Characterization of solutions
$I \otimes I$	Delta functions
$\begin{pmatrix} L_1 \otimes I \\ I \otimes L_1 \end{pmatrix}$	Piecewise constant functions
$\begin{pmatrix} L_2 \otimes I \\ I \otimes L_2 \end{pmatrix}$	Piecewise linear functions
$\begin{pmatrix} L_3 \otimes I \\ I \otimes L_3 \end{pmatrix}$	Piecewise quadratic functions

Table 1. The four choices of $\mathcal{L}$ used to illustrate 2-D PP-TSVD solutions.

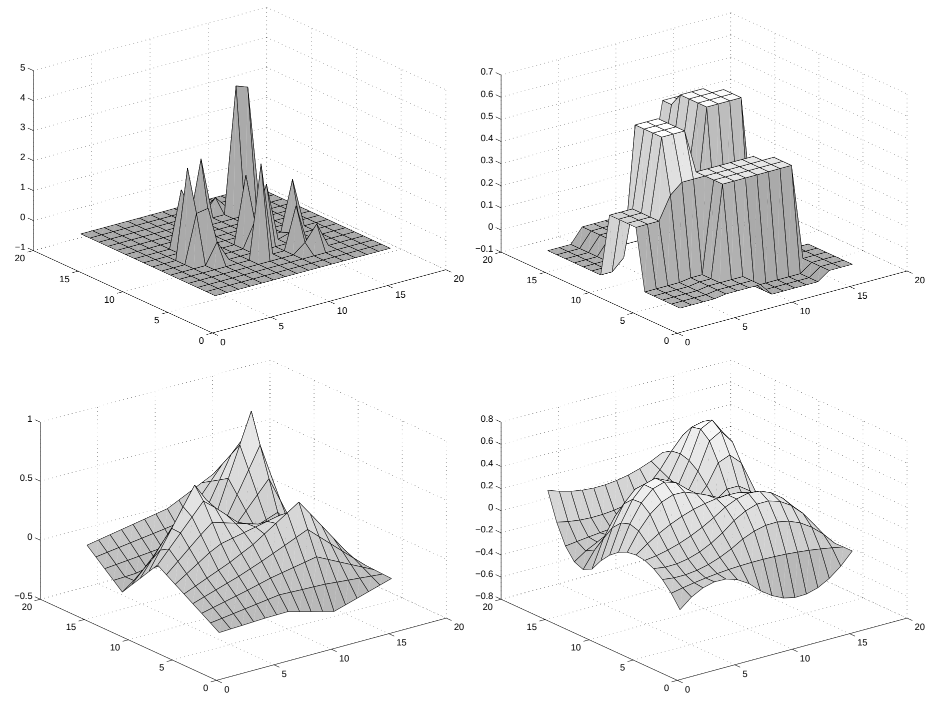

Fig. 3. Regularized solutions for the four choices of $\mathcal{L}$ in Table 1.

7 Image Reconstructions

We conclude with a few illustrations of the reconstruction capabilities of the 2-D PP-TSVD algorithm. Our test problem is the image deblurring problem blur included in Version 3 of REGULARIZATION TOOLS [7], and the image size is 16×16 (which is similar to the test problem used in [6, §7.7.2].

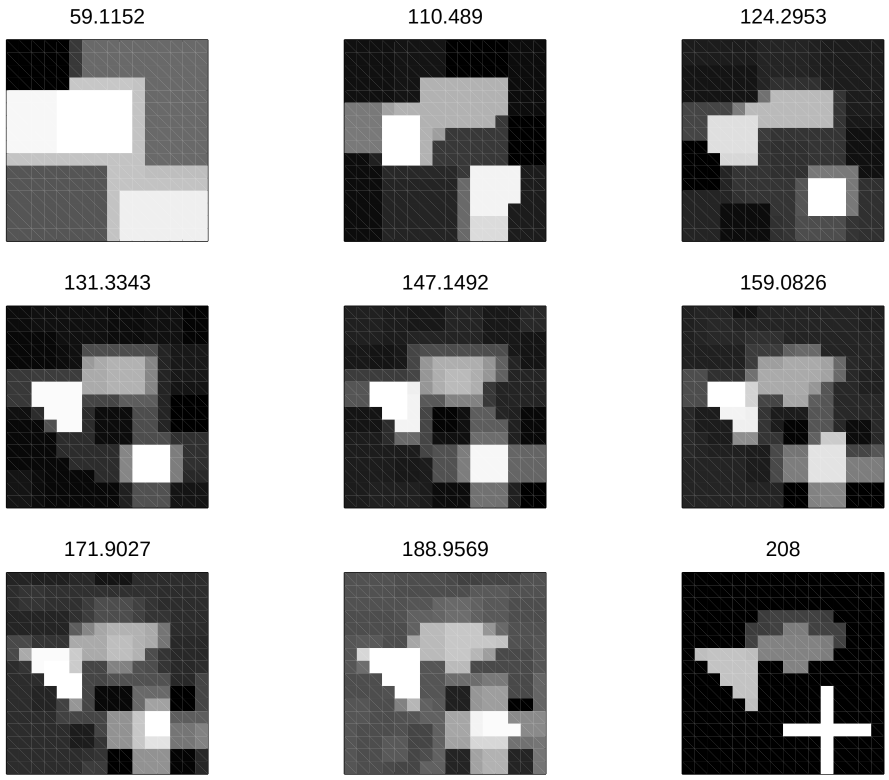

Fig. 4. Reconstructed 16×16 images F_k in the noise-free test problem for $k = 10, 20, \ldots, 90$, along with the quantities $\|\mathcal{L}\,\mathrm{vec}(F_k)\|_1$. Perfect reconstruction is achieved for $k = 90$.

First we use a noise-free test problem to demonstrate that the 2-D PP-TSVD algorithm is indeed capable of reconstructing images with sharp edges — in contrast to the classical regularization algorithms that produce smooth solutions. Figure 4 shows PP-TSVD reconstructions for $\mathcal{L}$ given by (22) and for $k = 10, 20, \ldots, 90$. We see that $\|\mathcal{L}\,\mathrm{vec}(F_k)\|_1$ increases monotonically with k until we achieve perfect reconstruction for $k = 90$. All the reconstructions are blocky and the number of image blocks increases with k.

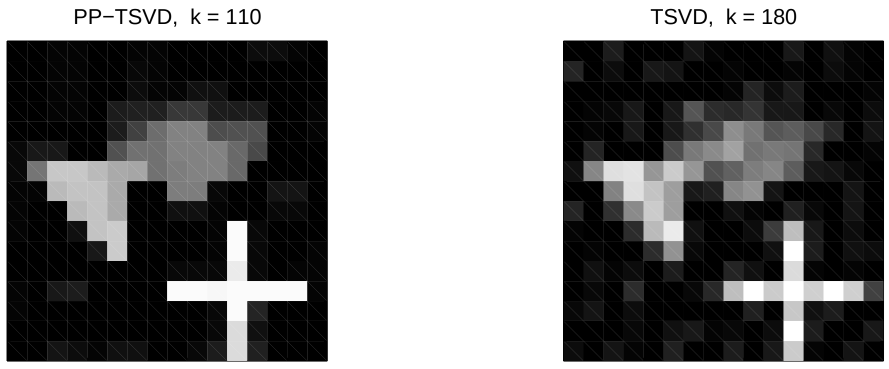

Fig. 5. The optimal PP-TSVD and TSVD solutions to $A\,F\,\bar{A}^T = G + E$ with additive Gaussian noise E and blurred signal-to-noise ratio $\|G\|_F/\|E\|_F = 10$.

We now add Gaussian noise E to the blurred image G, and we choose a noise level such that the blurred signal-to-noise ratio is $\|G\|_F/\|E\|_F = 10$. Figure 5 shows the "optimal" PP-TSVD solution F_k (i.e., the one that minimizes $\|F^{\rm exact} - F_k\|_F$), which is achieved for $k = 110$, along with the optimal TSVD solution achieved for $k = 180$. Clearly, the PP-TSVD algorithm is much better at reconstructing the sharp edges in the image than the TSVD algorithm.

The choice of the regularization parameter k is a complicated matter that lies outside the scope of this work.

References

1. M. R. Banham and A. K. Katsaggelos, *Digital image restoration*, IEEE Signal Proc. Magazine, 14 (1997), pp. 24–41.
2. I. Barrodale and F. D. K. Roberts, *An efficient algorithm for discrete approximation with linear constraints*, SIAM J. Numer. Anal., 15 (1978), pp. 603–611.
3. T. Corbard, G. Berthomieu, J. Provost, and P. Morel, *Inferring the equatorial solar tachocline from frequency splittings*, Astron. Astrophys., 330 (1998), pp. 1149–1159.
4. D. C. Dobson and F. Santosa, *Recovery of blocky images from noisy and blurred data*, SIAM J. Appl. Math., 56 (1996), pp. 1181–1198.
5. P. C. Hansen, *Regularizations Tools: A Matlab package for analysis and solution of discrete ill-posed problems*, Numer. Algo., 6 (1994), pp. 1–35.
6. P. C. Hansen, *Rank-Deficient and Discrete Ill-Posed Problems. Numerical Aspects of Linear Inversion*, SIAM, Philadelphia, 1998.
7. P. C. Hansen, *Regularization Tools, Version 3.0 for Matlab 5.2*, Numer. Algo., to appear.

8. P. C. Hansen and K. Mosegaard, *Piecewise polynomial solutions to linear inverse problems;* in B. H. Jacobsen, K. Mosegaard, and P. Sibani (Eds.), *Inverse Methods*, Lecture Notes in Earth Sciences 63, Springer, Berlin, 1996.
9. P. C. Hansen and K. Mosegaard, *Piecewise polynomial solutions without a priori break points*, Num. Lin. Alg. Appl., 3 (1996), pp. 513–524.
10. P. C. Hansen, T. Sekii, and H. Shibahashi, *The modified truncated SVD method for regularization in general form*, SIAM J. Sci. Stat. Comput., 13 (1992), pp. 1142–1150.
11. T. K. Nielsen, *Linear inversion of gravity and magnetic data using truncated SVD and piecewise polynomial TSVD;* in B. H. Jacobsen (Ed.), *Proc. Fifth Interdisciplinary Inversion Workshop*, Aarhus, 1997.
12. C. C. Paige and M. A. Saunders, *LSQR: An algorithm for sparse linear equations and sparse least squares*, ACM Trans. Math. Software, 8 (1982), pp. 195–209.
13. M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis and Machine Vision*, Chapman & Hall, London, 1993.
14. C. R. Vogel, *Nonsmooth regularization;* in H. W. Engl, A. K. Louis, and W. Rundell (Eds.), *Inverse Problems in Geophysical Applications*, SIAM, Philadelphia, 1995.
15. C. R. Vogel and M. E. Oman, *Iterative methods for total variation denoising*, SIAM J. Sci. Comput., 17 (1996), pp. 227–238.