Application of Adjoint Operators in Gradient Computations

James Folberth and Stephen Becker

Abstract—**When using first-order optimization algorithms, it is often the case that the user must supply the gradient of the differentiable terms in the objective function. We consider two example problems that have a Euclidean error term involving a linear operation on the problem variables. The gradient of the Euclidean error term involves both the linear operator and its adjoint, which, in our examples, is not known in the literature. The first example is an image deblurring problem, where the linear operation is multi-stage wavelet synthesis. Our formulation of the adjoint holds for a variety of boundary conditions, which allows the formulation to generalize to a larger class of problems. The second example is a blind channel estimation problem taken from the convex optimization literature; the linear operation is convolution, but with a slight twist. In each example, we show how the adjoint operator can be applied efficiently.**

Prerequisites

The reader should be familiar with linear algebra, wavelets, and basic Fourier analysis. Knowledge of first-order iterative optimization algorithms is beneficial for motivating the fast adjoint computation, but should not be necessary to understand the adjoint computation itself. We begin by briefly motivating the need for a fast adjoint operator and then turn toward the two examples.

I. A Model Optimization Problem

In this lecture note we consider as examples a few variants of the standard optimization problem

$$\min_x \frac{1}{2}\|\mathcal{A}x - b\|_2^2 + \lambda\|x\|_1, \tag{1}$$

where $\mathcal{A}$ is a linear operator, $\|\cdot\|_2$ is the ℓ_2 norm, and $\|\cdot\|_1$ is the ℓ_1 norm. Let $g(x) = \frac{1}{2}\|\mathcal{A}x - b\|_2^2$ and $h(x) = \lambda\|x\|_1$. Notice that $g(x)$ is convex and differentiable, and $h(x)$ is convex but non-differentiable. The scalarization parameter λ controls the relative importance of the approximation error $g(x)$ and the sparsity heuristic $\|x\|_1$. Loosely speaking, larger λ should yield a sparser solution.

Even though $h(x)$ is non-differentiable, there exist many efficient algorithms for solving the above problem and its variants. Popular methods include the proximal gradient method and its variants, which include iterative shrinkage thresholding and projected gradient. Crucial to the speed of these methods is an efficient proximal mapping (prox-operator) for the non-differentiable term $h(x)$. The proximal mapping is defined as

$$\mathrm{prox}_h(x) = \mathrm{argmin}_y \left(h(y) + \frac{1}{2}\|y - x\|_2^2\right).$$

The proximal mapping can be thought of as finding the point minimizing $h(y)$ plus a simple quadratic model about the point x. For $h(y) = \|y\|_1$, the proximal mapping is the shrinkage ("soft-threshold") operator and takes the following particularly simple form:

$$\mathrm{prox}_h(x) = \begin{cases} x_i - 1 & x_i \geq 1 \\ 0 & |x_i| < 1 \\ x_i + 1 & x_i \leq 1. \end{cases}$$

The main step of a simple proximal gradient method is

$$x^{(k)} = \mathrm{prox}_{t_k h}\left(x^{(k-1)} - t_k \nabla g(x^{(k-1)})\right),$$

with step size $t_k > 0$. The reader may note that this update resembles the simple gradient descent update for minimizing $g(x)$, with the exception of the proximal mapping which handles the non-differentiable term $h(x)$. It is clear that an efficient proximal mapping is desired, as the proximal mapping must be evaluated at each step of the simple proximal gradient method.

The gradient of the differentiable term $g(x)$ is found to be

$$\nabla g(x) = \mathcal{A}^*(\mathcal{A}x - b).$$

It is during the evaluation of the gradient $\nabla g(x)$ that we need efficient means of computing the action of $\mathcal{A}$ and its adjoint $\mathcal{A}^*$. As we will see in the next sections, the operator $\mathcal{A}$ tends to have a well known, efficient forward and (pseudo)inverse transform (e.g. discrete cosine transform, discrete wavelet transform). For unitary and orthogonal operators, the adjoint is identically the inverse. For non-unitary and non-orthogonal operators, however, applying the adjoint efficiently may pose a problem. The following examples show cases where a fast adjoint operation is possible.

II. Example 1: An Image Deblurring Problem

Digital images can be blurred through a variety of means. For example, the optical system can be out of focus and atmospheric turbulence can cause blurring of astronomical images. The goal of image deblurring is to recover the original, sharp image by posing the blurring process in a mathematical model.

J. Folberth and S. Becker are with the Department of Applied Mathematics, University of Colorado at Boulder, Boulder, CO, 80309 USA

In this example, we use the formulation used in [1]. It is assumed that the blurring action is known, and the deblurring problem is posed as an optimization problem, where the optimization variables are the wavelet coefficients of the recovered image. Let $\mathcal{R}$ be a known blurring operator. For instance, $\mathcal{R}$ could represent a Gaussian point spread function (PSF) under symmetric (reflexive) boundary conditions [7]. Again, this model assumes that we have a good model of the blurring operator.

Let $\mathcal{W}$ represent a multi-level wavelet synthesis operator with suitable boundary conditions. Let b be the observed, blurry image. Let $\mathcal{A} = \mathcal{R}\mathcal{W}$ be the linear operator that synthesizes an image from wavelet coefficients x and then blurs the synthesized image under the blurring operator $\mathcal{R}$. The image deblurring task is now readily posed as the optimization problem (1), where we seek to recover sparse wavelet coefficients that accurately reconstruct the blurred image:

$$\min_x \frac{1}{2}\|\mathcal{R}\mathcal{W}x - b\|_2^2 + \lambda\|x\|_1, \tag{2}$$

From the sparse wavelet coefficients in an optimal solution $\hat{x}$, we can synthesize the deblurred image via $\mathcal{W}\hat{x}$. This formulation is sometimes called the *synthesis* formulation, since we synthesize an image from the wavelet coefficients in the optimization variable x.

The gradient of the differentiable term $g(x) = \frac{1}{2}\|\mathcal{R}\mathcal{W}x - b\|_2^2$ is

$$\nabla g(x) = \mathcal{W}^*\mathcal{R}^*(\mathcal{R}\mathcal{W}x - b).$$

In the case of a blurring PSF, we can apply $\mathcal{R}$ and $\mathcal{R}^*$ efficiently in the Fourier domain via the FFT [1], [7]. Wavelet software packages (e.g. MATLAB Wavelet Toolbox [4]) implement fast discrete wavelet analysis and synthesis, which correspond to $\mathcal{W}^\dagger$ and $\mathcal{W}$ (we use $\mathcal{A}^\dagger$ to denote the Moore-Penrose pseudoinverse of $\mathcal{A}$). However, the adjoint wavelet transform is not implemented in software packages and is not discussed in the literature.

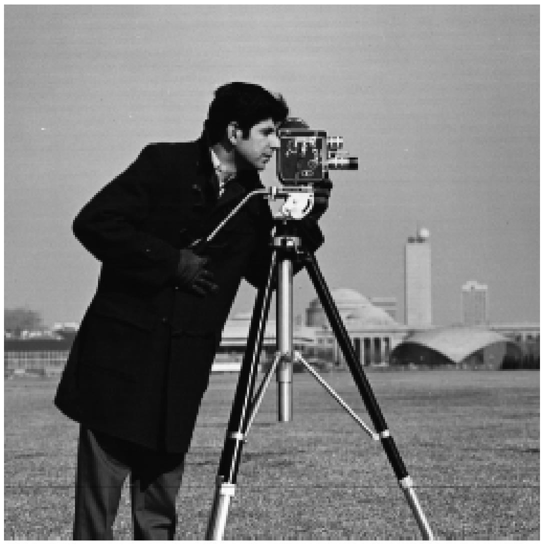
Fig. 1. The original, unblurred cameraman image. To create the blurred image, the $\mathcal{R}$ is applied to the original image and Gaussian noise is added.

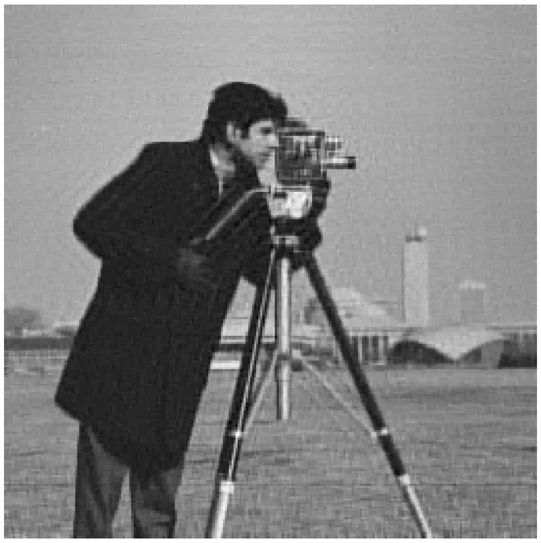
Fig. 2. Deblurred image after 200 iterations of FISTA with $\mathcal{W}$ a three-stage Haar transform with symmetric boundary conditions.

A. The adjoint for orthogonal wavelets

There is hope, however! For orthogonal wavelets (e.g. Haar and more generally Daubechies wavelets), the pseudoinverse $\mathcal{W}^\dagger$ is identically the adjoint $\mathcal{W}^*$, subject to matching boundary conditions, so the adjoint operator can be evaluated via the analysis operator $\mathcal{W}^\dagger$. Thus, one can compute the gradient as

$$\nabla g(x) = \mathcal{W}^\dagger\mathcal{R}^*\left(\mathcal{R}\mathcal{W}x - b\right).$$

Beck and Teboulle did exactly this with a three-stage discrete Haar wavelet transform [1]. For an example image, consider the cameraman image, shown in Figure 1. We prescribe symmetric boundary conditions on the image, which will affect both $\mathcal{W}$, $\mathcal{R}$, and their adjoints. Symmetric boundary conditions (compared to periodic or zero-padded boundary conditions) produce significantly less edge effects, and so are a natural choice. Luckily, for orthogonal wavelets, the adjoint of $\mathcal{W}$ with symmetric boundary conditions is identically the inverse $\mathcal{W}^\dagger = \mathcal{W}^{-1}$ with symmetric boundary conditions, so one can use existing wavelet libraries without hassle, e.g. [4].

Figure 2 shows the deblurred image after 200 iterations of FISTA, a fast proximal gradient method, developed by Beck and Teboulle [1]. The wavelet synthesis operator $\mathcal{W}$ is taken to be a three-stage Haar discrete wavelet transform with symmetric boundary conditions. In this case, the operator $\mathcal{W}$ is orthogonal, so $\mathcal{W}^* = \mathcal{W}^\dagger$, which is the three-stage wavelet reconstruction operation with symmetric boundary conditions. So, coding up the adjoint transform is a simple matter of performing the analysis operation with suitable boundary conditions.

B. The adjoint for biorthogonal wavelets

For biorthogonal wavelets, we no longer have $\mathcal{W}^* = \mathcal{W}^\dagger$. Since we have relaxed ourselves to biorthogonal wavelets, it is perhaps too much to ask that the adjoint of the primal wavelet synthesis operator involves only the primal wavelets. Let us recall briefly the pertinent aspects of frames of Hilbert spaces. These facts are described in more detail in [6].

Let $\mathcal{H}$ be a Hilbert space and take $f \in \mathcal{H}$ to be an arbitrary vector in the Hilbert space. Let $\{\phi_n\}_{n\in\Gamma}$, where Γ is an index set, be a set of vectors in $\mathcal{H}$. The set of vectors $\{\phi_n\}_{n\in\Gamma}$ is called a frame of $\mathcal{H}$ if there exists constants $B \geq A > 0$ such that

$$\forall f \in \mathcal{H}, \quad A\|f\|^2 \leq \sum_{n\in\Gamma} |\langle f, \phi_n\rangle|^2 \leq B\|f\|^2.$$

If the set $\{\phi_n\}_{n\in\Gamma}$ is a frame and is linearly independent, we call it a Riesz basis.

We assume henceforth that the set $\{\phi_n\}_{n\in\Gamma}$ is a frame of $\mathcal{H}$ with bounds $B \geq A > 0$. We can define the frame analysis operator Φ via

$$\forall n \in \Gamma, \quad \Phi f[n] = \langle f, \phi_n\rangle.$$

The frame analysis operator computes the expansion coefficients of f in the frame $\{\phi_n\}_{n\in\Gamma}$. The frame synthesis operator is Φ^*, and constructs a vector in $\mathcal{H}$ given expansion coefficients. Since $\{\phi_n\}_{n\in\Gamma}$ is assumed to be a frame, $\Phi^*\Phi$ is invertible, and we may define the Moore-Penrose pseudoinverse $\Phi^\dagger$, which implements reconstruction, as $\Phi^\dagger = (\Phi^*\Phi)^{-1}\Phi$.

Reconstruction with the pseudoinverse of the frame operator can be thought of as synthesis with a dual frame. The dual frame analysis operator and dual frame vectors are defined via

$$\forall n \in \Gamma, \quad \tilde{\Phi} f[n] = \langle f, \tilde{\phi}_n\rangle, \quad \text{where } \tilde{\phi}_n = (\Phi^*\Phi)^{-1}\phi_n.$$

Then the dual frame synthesis operator satisfied $\tilde{\Phi}^* = \Phi^\dagger$. Notice that we also have $\Phi^* = \tilde{\Phi}^\dagger$; this relation provides the basic idea for computing the adjoint of the discrete wavelet synthesis operator $\mathcal{W}$. Ignoring boundary effects and selecting the biorthogonal wavelets so that $\Phi^\dagger = \mathcal{W}$, we have the following:

$$\begin{aligned}\mathcal{W}^* &= \left(\Phi^\dagger\right)^* = \left((\Phi^*\Phi)^{-1}\Phi^*\right)^* = \Phi\,(\Phi^*\Phi)^{-1} \\ &= (\Phi^*)^\dagger = \left(\tilde{\Phi}^\dagger\right)^\dagger \\ &= \tilde{\Phi},\end{aligned}$$

where we have used various properties of the pseudoinverse. Thus, ignoring boundary conditions, the adjoint of wavelet synthesis is dual wavelet analysis! Note that in the case of orthogonal wavelets, $\Phi^*\Phi$ is the identity, the dual frame is identically the primal frame, and $\mathcal{W}^* = \tilde{\Phi} = \Phi = \mathcal{W}^\dagger$, exactly as before.

It now remains to include boundary conditions. A standard method for imposing boundary conditions on finite, discrete signals is to extend the signal to match those boundary conditions and perform the wavelet operations on the extended signal, assuming zero boundary conditions for the extended signal; this is the approach used in MATLAB's Wavelet Toolbox [4]. If we let $\mathcal{E}$ be the extension operator and $\mathcal{W}^\dagger_{\text{zpd}}$ be the wavelet analysis operator for extended signals under zero boundary conditions, we can write wavelet analysis as

$$\mathcal{W}^\dagger = \mathcal{W}^\dagger_{\text{zpd}}\mathcal{E}.$$

This separation of signal extension and wavelet operations provides a nice framework for determining the adjoint:

$$\mathcal{W} = \mathcal{E}^\dagger\mathcal{W}_{\text{zpd}}, \quad \mathcal{W}^* = \mathcal{W}^*_{\text{zpd}}(\mathcal{E}^\dagger)^*.$$

Consider a signal $y[n]$, $n = 0, ..., N-1$. Let L_p be the length of the wavelet analysis filters (one can zero-padd a shorter filter to length L_p). The double sided zero-padded extension of $y[n]$ puts $L_p - 1$ zeros on each end of the signal:

$$\underbrace{0, ..., 0}_{L_p-1}, y[0], ..., y[N-1], \underbrace{0, ..., 0}_{L_p-1}.$$

We can write down the linear operator $\mathcal{E}_{\text{zpd}}$ as a matrix that performs this operation on $y[n]$.

$$\mathcal{E}_{\text{zpd}} = \begin{bmatrix} 0_{(L_p-1)\times N} \\ I_{N\times N} \\ 0_{(L_p-1)\times N}\end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 1 & 0 \\ 0 & 0 & \cdots & 0 & 1 \\ 0 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & 0 \end{bmatrix}.$$

Once we have this explicit form of $\mathcal{E}_{\text{zpd}}$, we can find the adjoint of its pseudoinverse. In this case we have that $\left(\mathcal{E}^\dagger_{\text{zpd}}\right)^* = \mathcal{E}_{\text{zpd}}$. Although the zero-padded extension is simple to work with, it is not usually a reasonable extension mode for practical applications. A better extension mode is the half-point symmetric extension, which is the default extension mode in MATLAB's Wavelet Toolbox [4].

The double sided half-point symmetric extension reflects the signal about its boundaries in the following manner:

$$\underbrace{y[L_p-1], ..., y[0]}_{\text{Left extension}}, y[0], ..., y[N-1], \underbrace{y[N-1], ..., y[N+L_p-2]}_{\text{Right extension}}.$$

As in the zero-padded case, we can readily form a matrix that performs this operation on $y[n]$.

$$\mathcal{E}_{\text{sym}} = \left[\begin{array}{ccccc}
 & & \iddots & & \\
 & 1 & & & \\
1 & & & & \\
1 & & & & \\
 & 1 & & & \\
 & & \ddots & & \\
 & & & 1 & \\
 & & & & 1 \\
 & & & & 1 \\
 & & & 1 & \\
 & & \iddots & &
\end{array}\right].$$

The adjoint of the pseudoinverse can be computed in closed form and essentially amounts to rescaling the nonzero entries of $\mathcal{E}_{\text{sym}}$. Assuming $N > 2(L_p - 1)$, which usually occurs in practice, the adjoint of the pseudoinverse of $\mathcal{E}_{\text{sym}}$ is

$$\left(\mathcal{E}_{\text{sym}}^{\dagger}\right)^{*} = \left[\begin{array}{ccccccccc}
 & \iddots & & & & & & & \\
1/2 & & & & & & & & \\
1/2 & & & & & & & & \\
 & \ddots & & & & & & & \\
 & & 1/2 & & & & & & \\
 & & & 1 & & & & & \\
 & & & & \ddots & & & & \\
 & & & & & 1 & & & \\
 & & & & & & 1/2 & & \\
 & & & & & & & \ddots & \\
 & & & & & & & & 1/2 \\
 & & & & & & & & 1/2 \\
 & & & & & & & \iddots &
\end{array}\right].$$

If $N \leq 2(L_p - 1)$, the form of the pseudoinverse is slightly different, with some of the $1/2$ terms becoming $1/3$; both cases are considered in our implementation.

Figure 3 shows the deblurred cameraman image after 200 iterations of FISTA using a three-stage CDF 9/7 transform with symmetric boundary conditions.

III. Example 2: Blind Channel Estimation

Another application where an interesting adjoint is involved in the gradient computation is in blind channel estimation. In blind channel estimation, a single source sends an unknown signal over multiple channels with unknown response. Observers collect the output of each channel and collectively attempt to determine the source signal and the channel impulse response from each channel. Let s be the unknown source signal and h_i the channel impulse response of the ith channel. Then the output of the ith channel is

$$x_i = h_i * s.$$

In this form, we must solve a multitude of nonlinear equations in the components of s and each h_i.

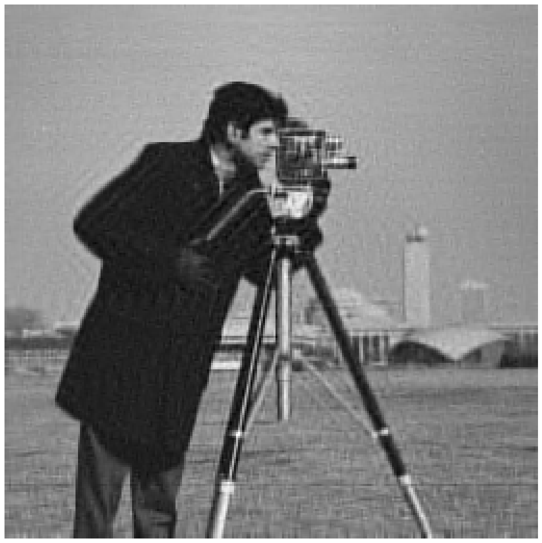

Fig. 3. Deblurred image after 200 iterations of FISTA with $\mathcal{W}$ a three-stage CDF 9/7 transform with symmetric boundary conditions.

References

[1] A. Beck and M. Teboulle, "A fast iterative shrinkage-thresholding algorithm for linear inverse problems," *SIAM Journal on Imaging Sciences*, 2009.

[2] P. L. Combettes and J.-C. Pesquet, "A Douglas-Rachford splitting approach to nonsmooth convex variational signal recovery," *IEEE Journal of Selected Topics in Signal Processing*, 2007.

[3] A. Skodras, C. Chistopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Magazine*, 2001.

[4] *MATLAB and Wavelet Toolbox Release R2015b*, The MathWorks, Inc., Natick, Massachusetts, 2015.

[5] G. Strang and T. Nguyen, *Wavelets and Filter Banks*. Wellesley-Cambridge Press, 1996.

[6] S. Mallat, *A Wavelet Tour of Signal Processing: The Sparse Way*. Elsevier, 2009.

[7] P. C. Hansen, J. G. Nagy, and D. P. O'Leary, *Deblurring Images: matrices, spectra, and filtering*. SIAM, 2006.

[8] S. Becker, J. Bobin, and E. J. Candès, "NESTA: A fast and accurate first-order method for sparse recovery," *SIAM Journal on Imagine Sciences*, 2011.

James Folberth Biography text here.

Stephen Becker Biography text here.