Mathematics for Imaging and Signal Processing

Assignment: Generalized Tikhonov Regularization & Error Analysis

A.A. 2025/2026

Context

In many imaging applications, we aim to recover a sharp image f from a blurred and noisy observation g. This inverse problem is modeled as:

$$g = Af + w \tag{1}$$

where A represents the convolution with a Point Spread Function (PSF) K, and w is additive noise. Since the direct inversion is ill-posed, we employ Regularization techniques. In this assignment, you will implement Generalized Tikhonov Regularization in the frequency domain, comparing different penalty terms (L^2, H^1, H^2) and analyzing the error components.

Here, the H^1 penalty enforces smoothness by penalizing the gradient, i.e. it involves $\|\nabla f\|_{L^2}$. The H^2 penalty enforces stronger smoothness by penalizing second derivatives, i.e. it involves the collection $\{\partial^2_{ij} f\}_{i,j}$ through $\|\partial^2 f\|_{L^2}$ (equivalently, the squared L^2-norm of the Hessian entries).

1 Part 1: The Forward Problem (Simulation)

Reference: Lecture Notes, Section 1.6 and 1.3.

1. **Image Selection:** Choose a sharp grayscale image $f^(0)$ (you can call it whatever you want to) (e.g., 256×256 pixels).
2. **Blur Simulation:** Implement the Transfer Function $\hat{K}(\omega)$ for at least **one** of the following physical models:
   - **Gaussian blur** (see the implementation guidelines)
   - **Linear Motion Blur:** defined by the Sinc function (Eq. 21 in notes).
   - **Out-of-Focus Blur:** defined by the Bessel function J_1 (Eq. 23 in notes).

   Hint: Ensure the DC component ($\omega = 0$) is centered correctly if using ***`fftshift`***. (on this point, read guidelines in the next pages)
3. **Noise Addition:** Generate the observed data $g = \mathcal{F}^{-1}(\hat{K}\hat{f}) + w$ by adding Gaussian noise. Simulate two scenarios:
   - High SNR (≈ 40 dB).
   - Low SNR (≈ 20 dB).

2 Part 2: Generalized Tikhonov Regularization

Reference: Lecture Notes, Section 5.

Implement the generalized reconstruction filter in the Fourier domain:

$$\hat{f}_\mu(\omega) = \frac{\hat{K}^*(\omega)}{|\hat{K}(\omega)|^2 + \mu|P(\omega)|^2}\hat{g}(\omega) \tag{2}$$

Compare the following three regularization strategies by varying the penalty polynomial $P(\omega)$:

A. Standard Tikhonov (L^2 Regularization)

Set $|P(\omega)|^2 = 1$. This corresponds to minimizing the energy of the solution $||f||^2$. It is effectively a "soft" thresholding of the frequencies based on the SNR.

B. First-Order Regularization (H^1 Penalty)

Set $|P(\omega)|^2 = |\omega|^2 = \omega_1^2 + \omega_2^2$. This penalizes the gradient magnitude $||\nabla f||^2$, promoting piecewise smooth solutions.
(Reference: Lecture Notes, Eq. 108. Note that in the lectures notes we are in 1D. Since, here, we are in 2D, $|\omega|^2 = \omega_1^2 + \omega_2^2$.)

C. Second-Order Regularization (H^2 Penalty)

Set $|P(\omega)|^2 = |\omega|^4 = (\omega_1^2 + \omega_2^2)^2$. This penalizes the second derivative (Laplacian), forcing the solution to be very smooth and strongly suppressing high-frequency noise.
(Reference: Lecture Notes, Eq. 110. Note that in the lectures notes we are in 1D. Since, here, we are in 2D, $|\omega|^4 = (\omega_1^2 + \omega_2^2)^2$.)

3 Part 3: Spectral Windowing (Comparison)

Reference: Lecture Notes, Section 6.1.

Compare the Tikhonov approach with a hard spectral cutoff (Truncated SVD equivalent). Implement the **Rectangular Window**:

$$\hat{f}_\Omega(\omega) = \hat{W}_\Omega(\omega)\frac{\hat{g}(\omega)}{\hat{K}(\omega)}, \quad \hat{W}_\Omega(\omega) = \begin{cases} 1 & \text{if } |\omega| < \Omega \\ 0 & \text{if } |\omega| \geq \Omega \end{cases} \tag{3}$$

Observe the difference in artifacts (ringing/Gibbs phenomenon) between the hard cutoff and the smooth Tikhonov filter.

4 Part 4: The Bias-Variance Trade-off

Reference: Lecture Notes, Section 4.

The total error of the reconstruction is composed of two competing terms:

$$\text{Total Error}^2(\mu) = \underbrace{||(R_\mu A - I)f_{true}||^2}_{\text{Approximation Error (Bias)}} + \underbrace{||R_\mu w||^2}_{\text{Noise Error (Variance)}} \tag{4}$$

For the H^1 regularization case:

1. Define a range of μ values (e.g., logarithmically spaced from 10^{-6} to 1).

2. Compute the **Bias** and **Variance** terms separately for each μ.
3. **Plot** the three curves (Bias, Variance, Total Error) on a log-log or semi-log scale.
4. Verify that the minimum of the Total Error corresponds to the intersection region of the Bias and Variance curves (the optimal trade-off).

Deliverables

1. **Source Code:** Python or MATLAB scripts.
2. **Report:**
   - Visual comparison of L^2, H^1, H^2 reconstructions (zoom in on edges).
   - The Bias-Variance trade-off plot.
   - A brief discussion on how the smoothness order (H^1 vs H^2) affects edge preservation and noise suppression.

Technical Appendix: Implementation Guidelines

Working in the frequency domain requires careful handling of the discrete grid and the zero-frequency component (DC). Below are the recommended steps to avoid common pitfalls.

Algorithm 1: Setup and Frequency Grid

Standard FFT implementations (Python `numpy.fft`, MATLAB `fft2`) place the zero frequency at index $(0,0)$. However, physical formulas (like the Gaussian or Bessel functions) assume the zero frequency is at the center of the domain. We recommend generating a centered grid first, and using `ifftshift` to adapt it to the FFT format.

Algorithm 1 Creating the Frequency Mesh

$N, M \leftarrow$ dimensions of image f ▷ Assume N, M are even
Create 1D ranges centered at 0:
$x\_range \leftarrow [-N/2, \ldots, N/2-1]$
$y\_range \leftarrow [-M/2, \ldots, M/2-1]$
Create 2D Meshgrid:
$\omega_X, \omega_Y \leftarrow \text{meshgrid}(x\_range, y\_range)$
Compute Frequency Magnitude (Radius):
$|\omega| \leftarrow \sqrt{\omega_X^2 + \omega_Y^2}$
Example: Constructing the Kernel (e.g., Gaussian):
$\hat{K}_{centered} \leftarrow \exp(-\frac{|\omega|^2}{2\sigma_{blur}^2})$
Shift for FFT compatibility:
$\hat{K} \leftarrow \text{ifftshift}(\hat{K}_{centered})$ ▷ Now zero freq is at (0,0)

Algorithm 2: The Forward Problem (Blur + Noise)

Once the kernel $\hat{K}$ is in the correct format, convolution becomes point-wise multiplication.

Algorithm 2 Blurring and Adding Noise

$\hat{f} \leftarrow \text{fft2}(f_{true})$
$\hat{g}_{clean} \leftarrow \hat{f} \cdot \hat{K}$ ▷ Point-wise multiplication
$g_{clean} \leftarrow \text{real}(\text{ifft2}(\hat{g}_{clean}))$ ▷ Return to spatial domain
Add Noise:
$\sigma_{noise} \leftarrow 10^{-\text{SNR}_{dB}/20} \cdot \text{std}(g_{clean})$ ▷ Calculate sigma based on desired SNR
$noise \leftarrow \text{random\_normal}(0, \sigma_{noise}, \text{shape} = (N, M))$
$g \leftarrow g_{clean} + noise$
$\hat{g} \leftarrow \text{fft2}(g)$ ▷ Spectrum of noisy data for reconstruction

---

Algorithm 3: Regularized Reconstruction

Implementing the generalized Tikhonov formula requires defining the penalty term $P(\omega)$ on the centered grid, then shifting it.

Algorithm 3 Generalized Tikhonov Filter

Define Penalty (on centered grid):
if Method is L^2 **then**
  $P_{sq} \leftarrow \text{ones}(N, M)$
else if Method is H^1 **then**
  $P_{sq} \leftarrow \omega_X^2 + \omega_Y^2$ ▷ Squared Gradient Magnitude
else if Method is H^2 **then**
  $P_{sq} \leftarrow (\omega_X^2 + \omega_Y^2)^2$ ▷ Squared Laplacian Magnitude
end if
Shift Penalty to match FFT:
$\hat{P}_{sq} \leftarrow \text{ifftshift}(P_{sq})$
Compute Filter:
$Denominator \leftarrow |\hat{K}|^2 + \mu \cdot \hat{P}_{sq}$
$\hat{f}_\mu \leftarrow \frac{\text{conj}(\hat{K}) \cdot \hat{g}}{Denominator}$
$f_\mu \leftarrow \text{real}(\text{ifft2}(\hat{f}_\mu))$