A study on the L-curve model and its applications

Seele Guo and Mingyu Yu

ABSTRACT. The L-curve model is a log-log plot of the residual norm versus the solution norm. It is a tool for selecting the optimal penalty parameter λ for Tikhonov regularization when solving the ill-posed inverse problems. In this project, we examine the properties and structures of the L-curve, including the singular value decomposition (SVD) and the curvature of the model. We also explore the existing algorithms for automatically selecting the optimal penalty parameter λ. Finally, we apply the L-curve model to the image deblurring model to provide the best approximations.

1. Introduction

Consider the least square problem $\min \|Ax - b\|_2$ for the matrix $A \in \mathbb{R}^{m\times n}, \operatorname{rank}(A) = r$. The singular value decomposition (SVD) of matrix A is $A = U\Sigma V^T$, where $U = (u_1, \ldots, u_m)$ consists of left singular vectors, $V = (v_1, \ldots, v_n)$ consists of right singular values, and $\{\sigma_j\}_{j=1}^r$ are the singular values of A.

By inserting the SVD to the least square problem, we retrieve

$$\begin{aligned}
\min \|Ax - b\|_2^2 &= \min \left\| \sum_{j=1}^r u_j \sigma_j v_j^T x - \sum_{j=1}^m u_j u_j^T b \right\|_2^2 \\
&= \min \left\| \sum_{j=1}^r u_j (\sigma_j v_j^T x - u_j^T b) + \sum_{j=r+1}^m u_j u_j^T b \right\|_2^2 \\
&= \min \left(\underbrace{\sum_{j=1}^r (\sigma_j v_j^T x - u_j^T b)^2}_{(1)} + \underbrace{\sum_{j=r+1}^m \left(u_j^T b\right)^2}_{(2)} \right).
\end{aligned} \tag{1.1}$$

We have no control over part (2) since it does not depend on x. Part (1) is minimized when it is set to 0. Therefore, by minimizing part (1), we retrieve

$$v_j^T \hat{x} = \frac{u_j^T b}{\sigma_j} \text{ for } j = 1, \ldots, n.$$

However, if the matrix A is ill-conditioned and some of its singular values gradually decay to 0, then the solution above is numerically unstable. Such algebraic problem $\min \|Ax - b\|$ is *ill-posed*. For an ill-posed problem $\min \|Ax - b\|_2$, the matrix A can be seen as underdetermined since some of its singular values are significantly close to 0. Hence, there are more than one x that obtain a perfrect fit, i.e., $\|y - Ax\|_2^2 = 0$. To rule out the redundant solutions, as well as to retrieve a numerically stable solution, we need to require further information and constraints. Such a process is known as regularization.

1.1. Tikhonov regularization. A popular regularization approach to the ill-posed problems is to require the ℓ^2-norm of the solution to be small. That is, given the default solution x_0, the constraint requires the minimization of the following quantity:

$$\Omega(x) = \|L(x - x_0)\|_2^2, \tag{1.2}$$

where the matrix L is a banded matrix with full row rank. This method is called Tikhonov regularization.

DEFINITION 1.1 ([Han99]). *Tikhonov regularized solution x_λ solves the following least square problem:*

$$\min \left\{ \|Ax - b\|_2^2 + \lambda^2 \|L(x - x_0)\|_2^2 \right\}, \tag{1.3}$$

The matrix L is set to the identity matrix I in its standard form.

The minimum of (1.3) is achieved if its derivative is set to 0. Formally,

$$\partial_x((Ax - b)^T(Ax - b) + \lambda^2(x - x_0)^T L^T L(x - x_0)) = 0.$$

Simplifying the above equation, we have the normal equation for the Tikhonov regularization,

$$(\lambda^2 L^T L + A^T A)x = \lambda^2 L^T L x_0 + A^T b. \tag{1.4}$$

1.2. The singular value decomposition. The main analysis tool for the Tikhonov regularization is the (compact) singular value decomposition. A general problem as in (1.3) can always be simplified to the case where the banded matrix L is the identity matrix [Han98, Chapter 2.3]. Therefore, in the rest of the text, we will assume that $L = I$ to considerably simplify our analysis of the problem. Let $A = U\Lambda V^T = \sum_{j=1}^r \sigma_j u_j v_j^T$, where $U = [u_1, u_2, \ldots, u_m]$, Λ is a m-by-n matrix whose nonzero is $\Lambda(j,j) = \sigma_j$ for $j = 1, 2, \ldots, r$, $V = [v_1, v_2, \ldots, v_n]$ and $r = \text{rank}(A)$. Let $\hat{x}_j = v_j^T x_\lambda$, $\hat{x}_{0,j} = v_j^T x_0$, and $\hat{b}_j = u^T b_j$. Plug in the SVD to equation (1.4) to retrieve

$$\sum_{j=1}^{r} (\lambda^2 + \sigma_j^2)\hat{x}_j v_j + \sum_{j=r+1}^{n} \lambda^2 \hat{x}_j v_j = \sum_{j=1}^{r} \left(\lambda^2 \hat{x}_{0,j} + \sigma_j^2 \frac{\hat{b}_j}{\sigma_j} \right) v_j + \sum_{j=r+1}^{n} \lambda^2 \hat{x}_{0,j} v_j \tag{1.5}$$

Therefore, the Tikhonov solution is given by

$$\begin{aligned} x_\lambda &= \sum_{j=1}^{r} \left(\frac{\lambda^2}{\lambda^2 + \sigma_j^2} \hat{x}_{0,j} + \frac{\sigma_j^2}{\lambda^2 + \sigma_j^2} \frac{\hat{b}_j}{\sigma_j} \right) v_j + \sum_{j=r+1}^{n} \hat{x}_{0,j} v_j \\ &= \sum_{j=1}^{r} \left(\frac{\lambda^2}{\lambda^2 + \sigma_j^2} \hat{x}_{0,j} + f_j \frac{\hat{b}_j}{\sigma_j} \right) v_j + \sum_{j=r+1}^{n} \hat{x}_{0,j} v_j \end{aligned} \tag{1.6}$$

The factors $f_j = \frac{\sigma_j^2}{\lambda^2 + \sigma_j^2}$ are known as the Tikhonov filter factors. In particular, if the current singular value σ_j is large, the regularized solution will ignore the corresponding element in the default solution $x_{0,j}$ when $j = 1, 2, \ldots, r$, while the current singular value σ_j is small, the filter f_j will increase the weight of the default solution.

1.3. The L-curve model. To simplify our calculation, we will assume that the default solution (the priori estimate) $x_0 = 0$ in the rest of the text, since when we usually have no priori estimate at first. And the Tikhonov solution in (1.6) is simplified to

$$x_\lambda = \sum_{j=1}^{r} f_j \frac{\hat{b}_j}{\sigma_j} v_j = \sum_{j=1}^{r} \frac{\sigma_j^2}{\lambda^2 + \sigma_j^2} \frac{u_j^T b}{\sigma_j} v_j. \tag{1.7}$$

which also can be write into the form of

$$x_\lambda = V c_\lambda \tag{1.8}$$

where c_λ is a vector of $\mathbb{R}^n$

$$c_\lambda = (f_1 \frac{\hat{b}_1}{\sigma_1}, f_2 \frac{\hat{b}_2}{\sigma_2}, \ldots, f_r \frac{\hat{b}_r}{\sigma_r}, 0, \ldots, 0)^T \tag{1.9}$$

The regularized solution norm is

$$\|x_\lambda\|_2^2 = \|Vc_\lambda\|_2^2 = \|c_\lambda\|_2^2 = \sum_{j=1}^{r} \left(f_j \frac{u_j^T b}{\sigma_j} \right)^2 \tag{1.10}$$

and the residual norm is

$$\|Ax_\lambda - b\|_2^2 = \|\Lambda c_\lambda - U^T b\|_2^2 = \sum_{j=1}^{r} \left((1 - f_j)\, u_j^T b\right)^2 + \sum_{j=r+1}^{m} \left(u_j^T b\right)^2 \tag{1.11}$$

It immediately follows that the solution norm $\|x_\lambda\|_2$ is a decreasing function of residual norm $\|Ax_\lambda - b\|_2$.

In order to choose an appropriate penalty parameter λ, we need to consider how to display regularized solution x_λ as a function of λ. A convinient way is to plot the residual norm $\|Ax_\lambda - b\|_2$ versus the regularized solution norm $\|x_\lambda\|_2$ [Han92].

DEFINITION 1.2 ([Han92]). *A continuous curve parameterized by the penalty parameter* λ

$$(\|Ax_\lambda - b\|_2, \|x_\lambda\|_2) \tag{1.12}$$

on the ***log-log*** *scale is called the L-curve in its standard form.*

The *general* L-curve is defined as $(\|Ax_\lambda - b\|_2, \|L(x_\lambda - x_0)\|_2)$ on the log-log scale, but as we discussed earlier, a general Tikhonov regularization problem can always be transformed to its standard form, so it suffices to work with the standard L-curve model. The L-curve model usefully solves the following constrained least square problems.

THEOREM 1.3 ([Han92]). *Any point* (δ, η) *on the curve* $(\|Ax_\lambda - b\|_2, \|x_\lambda\|_2)$ *solves*

$$\delta = \min \|Ax - b\| \textit{ subject to } \|x\| \leq \eta, \quad 0 \leq \eta \leq \|x_0\|,$$
$$\eta = \min \|x\| \textit{ subject to } \|Ax - b\| \leq \delta, \quad \delta_0 \leq \delta \leq \delta_\infty.$$

where

$$\delta_0 = \min_\lambda \|Ax_\lambda - b\|_2 = \|Ax_0 - b\|_2 \textit{, which implies } \delta_0^2 = \sum_{j=r+1}^{m} \left(u_j^T b\right)^2$$

$$\delta_\infty = \max_\lambda \|Ax_\lambda - b\|_2 = \|Ax_\infty - b\|_2 \textit{, which implies } \delta_\infty^2 = \sum_{j=1}^{m} \left(u_j^T b\right)^2$$

REMARK 1.4. Here x_0 means the limit of regularized solution x_λ when $\lambda \to 0$, not the default solution or the priori estimate. And x_∞ means the limit of regularized solution x_λ when $\lambda \to \infty$

It is significant to notice that

$$\begin{aligned} Ax_\lambda - b &= (A\bar{x}_0 - b) + (A\bar{x}_\lambda - A\bar{x}_0) + (Ax_\lambda - A\bar{x}_\lambda) \\ &= (A\bar{x}_0 - b) + A\underbrace{(\bar{x}_\lambda - \bar{x}_0)}_{\text{(i)}} + A\underbrace{(x_\lambda - \bar{x}_\lambda)}_{\text{(ii)}}, \end{aligned} \tag{1.13}$$

where $\bar{x}_0$ is the unregularized solution to the unperturbed problem, part (i) describes the regularization error, and part (ii) describes the perturbation error. If part (i) is large, then the residua norm is large. If part (ii) is large, then the regularized solution norm is large [Han92]. This suggests that a regularized parameter λ is optimal if it nears the corner of the L-curve.

2. Implementation of the L-curve model

The foundation of our analysis rests on an assumption termed the *Discrete Picard Condition*. This assumption forms a cornerstone of our analysis and will be recurrently used in subsequent discussions. We posit the following:

ASSUMPTION 2.1 ([Han99]). *The SVD coefficients $|u_i^T b|$ decay faster than σ_i*

This assumption ensures that the least square solution $x_0 = A^+ b$ to the regression when $\lambda = 0$ does not have a large norm, and also implies a critical corollary for the solution coefficients, represented as $|v_i^T x| = |u_i^T b/\sigma_i|$, which also decay under the *Discrete Picard Condition*. Later we will explore the implications and consequences of potential failures of this assumption, which will not only illuminate the limitations inherent in the current L-curve model but also inspire avenues for its improvement and refinement.

2.1. The shape of L-curve. To sketch the shape of the L-curve, which is also a smooth parametric curve theoretically, we will conduct a segmented analysis of it by the different value of λ.

It is easy to verify that as λ increases, f_i decrease and hence $\|x_\lambda\|_2$ decreases, while the residual norm $\|Ax_\lambda - b\|_2$ is increasing, which implies the L-curve is downwards.

When $\lambda \ll \sigma_r$, we have $f_j = \frac{\sigma_i^2}{\lambda^2+\sigma_i^2} \approx 1$, which implies the point in the L-curve is approximately by $(\log \|Ax_\lambda - b\|_2, \log \|x_\lambda\|_2) \approx (\log \|Ax_0 - b\|_2, \log \|x_0\|_2) = (\log \delta_0, \log \|x_0\|_2)$.

When the regularization parameter λ lies somewhere between σ_1 and σ_r, there are some filter factors close to zero and others not. Denote there are k filter factors close to one, then we have

$$\|x_\lambda\|_2^2 = \sum_{j=1}^{r} \left(f_j \frac{u_j^T b}{\sigma_j} \right)^2 \approx \sum_{j=1}^{k} \left(\frac{u_j^T b}{\sigma_j} \right)^2 \approx \sum_{j=1}^{r} \left(\frac{u_j^T b}{\sigma_j} \right)^2 = \|x_0\|_2^2 \tag{2.1}$$

where the last $\approx$ uses the corollary of *Discrete Picard Condition*, which implies the last $(r - k)$ terms contribute very little to the sum. And as we can see, the faster $|u_i^T b|$ decay faster than σ_i, the smaller the $\frac{u_j^T b}{\sigma_j}$ are, and hence the two sides around the last $\approx$ closer, which implies $\|x_\lambda\|$ varies less. And also we have

$$\|Ax - b\|_2^2 = \sum_{j=1}^{r} \left((1 - f_j)\, u_j^T b \right)^2 + \delta_0^2 \approx \sum_{j=k+1}^{r} \left(u_j^T b \right)^2 + \delta_0^2 \tag{2.2}$$

As λ increases from σ_r to σ_1, the regularization solution norm still varies little, but the residual norm grows from a tiny value to one approaches its upper bound. And finally the L-curve eventually bend down towards the horizontal axis, which happens when λ is comparable with the largest singular value σ_1.

When $\lambda \gg \sigma_1$, we have $f_j = \frac{\sigma_i^2}{\lambda^2+\sigma^2} \approx 0$, and hence the solution norm $\|x_\lambda\|_2 \xrightarrow{\lambda\to\infty} \|x_\infty\|_2 = 0$ and the residual norm $\|Ax_\lambda - b\|_2 \xrightarrow{\lambda\to\infty} \|Ax_\infty - b\|_2 = \delta_\infty$ as λ tends to infinity.

Intuitively, a good parameter λ is one that corresponds to a regularized solution near the "corner" of the L-curve because in this region there is a good compromise between achieving a small residual norm and keeping the solution norm small, as we shall see later.

2.2. The curvature of the L-curve. In this section, we will derive a convenient expression for the curvature of the L-curve, which plays a significant role in the understanding and the use of L-curve. The main theorem of this subsection is the following:

THEOREM 2.2 ([Han99]). *The curvature of L-curve κ is a function of λ and*

$$\kappa = 2\frac{\eta\rho}{\eta'} \cdot \frac{\lambda^2\eta'\rho + 2\lambda\eta\rho + \lambda^4\eta\eta'}{(\lambda^4\eta^2 + \rho^2)^{3/2}}, \tag{2.3}$$

where $\eta = \eta(\lambda) = \|x_\lambda\|_2^2$, and $\rho = \rho(\lambda) = \|Ax_\lambda - b\|_2^2$. The penalty parameter λ is achieved when the curvature κ is maximized.

PROOF. Let $\hat{\eta} = \log \eta$ and $\hat{\rho} = \log \rho$. Then L-curve is a plot of $\hat{\eta}/2$ versus $\hat{\rho}/2$. The curvature κ of the curve is given by

$$\kappa = \frac{2\left(\hat{\rho}'\hat{\eta}'' - \hat{\eta}'\hat{\rho}''\right)}{\left((\hat{\rho}')^2 + (\hat{\eta}')^2\right)^{3/2}} \tag{2.4}$$

Formula (2.4) gives the exact value of curvature, but it is not convenient for computing. Now we shall see we can turn the terms of $\hat{\eta}$ and $\hat{\rho}$ into η and ρ [Han99]. Denote $\beta_j = u_i^T b$. The first derivative of $\hat{\eta}$ and $\hat{\rho}$ are

$$\hat{\eta}' = \frac{d\hat{\eta}}{d\lambda} = \frac{\eta'}{\eta} \quad \text{and} \quad \hat{\rho}' = \frac{d\hat{\rho}}{d\lambda} = \frac{\rho'}{\rho}.$$

The derivates of η and ρ are

$$\eta' = -\frac{4}{\lambda}\sum_{j=1}^{r}(1-f_j)f_j^2\frac{\beta_i^2}{\sigma_i^2}, \qquad \rho' = \frac{4}{\lambda}\sum_{j=1}^{r}(1-f_j)^2 f_j \beta_i^2. \tag{2.5}$$

By noticing the following relation

$$\frac{f_j}{\sigma_j^2} = \frac{1}{\sigma_j^2 + \lambda^2} = \frac{1-f_j}{\lambda^2},$$

we may conclude that

$$\rho' = -\lambda^2 \eta'$$

The second derivative of $\hat{\eta}$ and $\hat{\rho}$ are given by

$$\hat{\eta}'' = \frac{d}{d\lambda}\frac{\eta'}{\eta} = \frac{\eta''\eta - (\eta')^2}{\eta^2}, \qquad \hat{\rho}'' = \frac{d}{d\lambda}\frac{\rho'}{\rho} = \frac{\rho''\rho - (\rho')^2}{\rho^2} = -2\lambda\eta' - \lambda^2\eta''. \tag{2.6}$$

Therefore, by substituting equations above into the formula (2.4), we recover the result in Theorem 2.2. □

Hansen also demonstrates the concavity of the curve by proving its negative curvature [Han99].

2.3. The method to choose the regularization parameter. The reason we use the corner to determine a regularization parameter lies in the fact that the corner represents a point where there is a fair balance between the regularization and perturbation errors since the corner separates the horizontal part of the curve from the vertical part.

Intuitively, if L-curve has a sharp corner, the abscissas of the left part of the curve are all very close to δ_0, while the ordinates of the right part of the curve are all very close to $\|x_\infty\|_2$, which implies the corner appears approximately at $(\delta_0, \|x_\infty\|_2)$. And now let us focus on the way to select the optimal parameter λ.

Through the theoretical analysis previously, we can directly parameterize the curvature of the L-curve as a function of λ, thus determining the maximum value of curvature and the λ_{opt} which corresponds to the corner point. on the L-curve. It is ideal since maximizing the curvature is much less costly than computing SVD and we even get the formula of curvature. However, in reality, sometimes due to large data volume or other reasons, we can only obtain a limited number of points on the L-curve, which inspires us to seek a method that uses a limited number of points on the L-curve to find the corner.

A highly reasonable approach is to use the discrete points to define a differentiable and smooth curve and consider it as the L-curve. We can accomplish it by either interpolation polynomials or cubic spline interpolation. Interpolation might not be able to tell us if the optimal parameter lies near the region where our selected points are concentrated, while the cubic spline curve does not have the desired local smoothing property, we can combine these two:

ALGORITHM 2.3 ([Han93]). *First perform a local smoothing of the L-curve points, in which each point is replaced by a new point obtained by fitting a low-degree polynomial to a few neighboring points. And then use the new smoothed points as control points for a cubic spline curve with knots* $1,\ldots,\ N+4$*, where* N *is the number of L-curve points. And finally locate the corner by computing the maximum curvature of the curve.*

There's yet another method to locate the point closest to the corner among all discrete points [Han08]. Initially, convert the solution norm and residual norm into a log-log scale. Next, define vectors between consecutive points and calculate their lengths. Normalize these vectors for subsequent computations and arrange them in descending order based on their lengths, prioritizing the longest ones first. This sorting aids in identifying potential corner points. There exist two algorithms to determine potential corner candidates: Angles and Global Behavior.

The Angles routine calculates the wedge products of consecutive vectors. By encountering a negative wedge product, it finds a potential corner, and the index of the corresponding point is considered a candidate.

The Global Behavior routine calculates angles between the normalized vectors and the horizontal axis. These angles are sorted, identifying vectors representing the horizontal and vertical components of the L-curve. Once we locate an intersection point representing the 'origin' of the L-curve by extending the horizontal and vertical parts of the L-curve in the opposite direction, we compute the distance of all L-curve points to this 'origin'. The point with the smallest Euclidean distance is then chosen as a potential corner candidate.

After gathering all potential corner candidates from the routines mentioned above, the next step involves refining them based on convexity. If no convex pruned L-curves are found, it implies that the selected points do not contain the corner. If so, the most suitable candidate is chosen as follows: evaluate the change in the solution norm relative to the change in the residual norm and look for a candidate where the increase in the solution norm is greater than or equal to the decrease in the residual norm. Then choose the rightmost corner candidate meeting this criterion (if available); otherwise, select the leftmost corner candidate. This approach finally determines the corner of the L-curve.

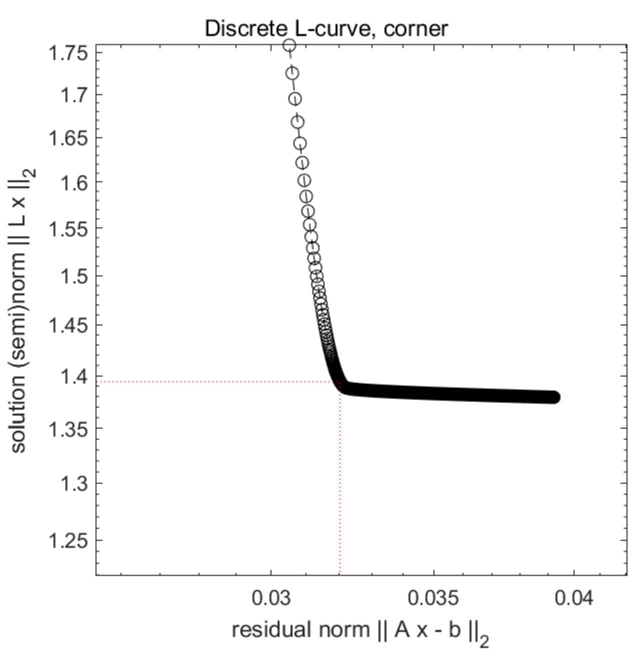

FIGURE 1. An L-curve example with selection of optimal parameter (in terms of iterations)

3. Application of the L-curve model: image deblurring

One application of the L-curve model is in image deblurring. Such a problem can be seen in the following form:

$$Hx + \varepsilon \approx \hat{H}x = b.$$

We are given a noisy image b and we want to recover the true image x. We are also given the mask matrix $\hat{H}$ that is ill-conditioned. For this project, the input image is blurred by applying the `fsepcial` function with the `motion` kernel in MATLAB. Such a function performs a filter to convolve with the input image and apply a linear motion of a camera. The image is blurred at an angle of 25 degrees along a line of 30 pixels. With such a kernel, we can construct a mask matrix $\hat{H}$. Then, the image deblurring problem can be solved by the Tikhonov regularization

$$\min \|\hat{H}x - b\|_2^2 + \lambda \|x\|_2^2 \tag{3.1}$$

The input image (Figure 2) has 200×200 pixels, and therefore the size of the corresponding mask matrix $\hat{H}$ is 40000×40000. Due to the restriction of our current computer power, we cannot form SVD for matrix $\hat{H}$. We use the normal equation to solve for x instead. A remark is that the images in MATLAB are saved in a three-dimensional array with the size of length$\times$width$\times$3. We transform the three-dimensional array to a matrix with height$\times$width rows and 3 columns. Therefore, the image matrix b and x have the same size, which is 40000×3.

FIGURE 2. The real image (left) and the blurred image (right)

We adopt the naive method for plotting the L-curve for this application due to the size of matrix $\hat{H}$. That is, we iterate through all $\lambda \in [0, 1/2]$ with a footstep of 10^{-4}, and we record the residual norm and the solution norm in each iteration. The corner point of the L-curve (Figure 3) suggests that $\lambda = 0.005$ provides the best approximation.

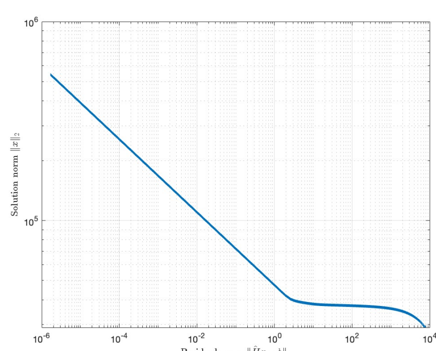

FIGURE 3. The L-curve for image deblurring

A practical challenge (as shown in this implementation) is that the L-curve may be too smooth to select the corner point. In this case, we need to manually compare the sum of residual and solution norms to make sure that $\lambda = 0.005$ provides the best approximation. A comparison of the chose of λ is shown in Figure 4.

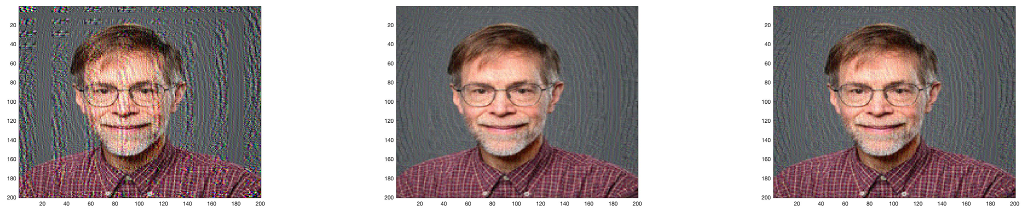

FIGURE 4. Recovered images with $\lambda = 0.001, 0.005$ (best), 0.01 respectively

4. The limitation of the L-curve model

When the Discrete Picard Condition fails, as we see in formula (2.1), the magnitude of change in $\|x_\lambda\|_2$ is larger, which means the L-curve is not that sharp, so the λ_{opt} we compute above may not be the most optimal parameter [Han99]. But when the corner of the L-curve is less sharp, perhaps we can consider finding other criteria beyond curvature to balance the trade-off between the solution norm and residual norm.

Additionally, all preceding analyses were mainly based on the assumption of disregarding noise and errors. If the data inherently include a degree of error or noise, the methods outlined above for parameter selection might not be applicable. There exist several references addressing this concern though an in-depth analysis of this aspect is outside of this paper. Simultaneously, the method to select optimal parameters is highly worth investigating.

5. MATLAB codes

LISTING 1. Implementation of image deblurring

```
%% blurring image
filter  =  fspecial ('motion', 30, 25);
im1 = imread('demmel.jpeg');
im2 = imfilter(im1, filter);
imwrite(im2, 'demmel_blurred.png');

%% set-up
[imblurd,h,w,H] = setup();

%% recording residual and sol norms
ress = zeros(1e4,1);
sols = zeros(1e4,1);
j=1;
for i=0:1e-4:0.5
    [curx, curres, cursol] = tikhonov(imblurd, H, i);
    ress(j) = curres; sols(j) = cursol;
    j = j+1;
end

%% approximation
im1 = tikhonov(imblurd, H, 0.005);
figure(1); to_image(im1,h,w);

%% plotting
figure(2);
loglog(ress(1:5000),sols(1:5000), 'LineWidth',3)
xlabel('Residual norm $\|\hat Hx-b\|_2$','Interpreter','latex')
ylabel('Solution norm $\|x\|_2$','Interpreter','latex')
grid on

%% functions
function [reshaped_blurred, height, width, Hmask] = setup()
    blurred = imread('demmel_blurred.png');
    height = size(blurred,1); width = size(blurred,2);
    K = fspecial('motion', 30, 25);
    Hmask = const_mask(K, height, width);
    reshaped_blurred = reshape(double(blurred),height*width,3);

    % get some idea from
    % https://sites.cc.gatech.edu/classes/AY2016/cs4476_fall/results/proj1/html/shiremath9/index.html
    function H = const_mask(kernel, height, width)
        [i,j,hij] = find(kernel);
        i = i-(size(kernel,1)+1)/2;
        j = j-(size(kernel,2)+1)/2;
        N = height*width;
        pixel_i = (1:height)'*ones(1,width);
```

```
        pixel_j = ones(height,1)*(1:width);
        pixel_k = reshape(1:(height*width), height, width);

        H = sparse(N,N);
        for cnt = 1:length(hij)
            hpixel_i = pixel_i+i(cnt);
            hpixel_j = pixel_j+j(cnt);
            hpixel_k = (hpixel_j-1)*height+hpixel_i;
            index = find(hpixel_i > 0 & hpixel_i <= height & hpixel_j > 0 & hpixel_j <= width);
            H_noise = sparse(pixel_k(index), hpixel_k(index), hij(cnt)*ones(length(index),1), N, N);
            H = H+H_noise;
        end
    end
end

function [x,res,sol] = tikhonov(b, H_hat, lambda)
    [m, n] = size(H_hat);
    [R, flag, P] = chol(H_hat'*H_hat+lambda^2*speye(n));
    curr = R'\(P'*H_hat'*b);
    curr = R\curr;
    x = P*curr;
    res = norm(H_hat*x-b);
    sol = norm(x);
end

function to_image(mat, height, width)
    reshaped = reshape(min(max(mat,0),255),height,width,3);
    image(reshaped/255);
end
```

LISTING 2. The method for optimal parameter lambda

```
num=10^3;
data=zeros(3,num);

% test data
% example 1
% A = [1, 1; 1, 1.001];
% b = [2.01;2];

% test data
% example 2
A=[0.16 0.10; 0.17 0.11;2.02 1.29];
b=[0.27;0.25;3.33];


% SVD decomposition of matrix A
[U, Sigma, V] = svd(A);
m = size(A, 1);
```

```
n = size(A, 2);
k = min([m n]);

% make A ill-posed if not
% Sigma(k,k)=0.001;
% A=U*Sigma*V'
% b=zeros(m,1);
% for i=1:1:k
%     b=b+Sigma(i,i)*U(:,i)*10^(1-i);
% end

tolerance = 1e-16;
D = diag(Sigma);
D = D(D > tolerance);
mm=min(D); % the smallest singular value
MM=max(D); % the biggest singular value
data(1,:)=linspace(90*mm,0.5*mm,num);
for i=1:1:num
    lambda=data(1,i);
    [x, residual] = ridgeRsion(U,Sigma,V,b,lambda);
    data(2,i)=norm(residual,2);
    data(3,i)=norm(x,2);
end
[k_corner,] = corner(data(2, :),data(3, :),1);
lambda_opt=data(1,k_corner)

function [x, residual] = ridgeRsion(U, Sigma, V,b,lambda)
% Ridge regression
% min_{x}||Ax-b||_2^2+\lambda||x||_2^2
% solution: x=argmin_{x}||Ax-b||_2^2+\lambda||x||_2^2
% residual: r=Ax-b
% [U, Sigma, V] = svd(A, "econ");
A=U*Sigma*(V');
m = size(U, 1);
n = size(V, 2);
k = min([m n]);
tolerance = 1e-16;
D = diag(Sigma);
D = D(D > tolerance);
r = size(D);
f=D.*D./(lambda^2.+D.*D);
x=A(1,:)'-A(1,:)';
for i=1:1:r
    x=x+f(i).*(U(:,i)'*b)*V(:,i)./D(i);
end
residual=A*x-b;
end

function [k_corner,info] = corner(rho,eta,fig)
%CORNER Find corner of discrete L-curve via adaptive pruning algorithm.
```

```
%This function corner is excerpted from https://www.mathworks.com/
   matlabcentral/fileexchange/52-regtools
```

References

[Han08] Per Christian Hansen. *Regtools*. https://www.mathworks.com/matlabcentral/fileexchange/52-regtools. Online. 2008.

[Han92] Per Christian Hansen. "Analysis of discrete ill-posed problems by means of the L-curve". In: *SIAM review* 34.4 (1992), pp. 561–580.

[Han93] Per Christian Hansen. "The Use of the L-Curve in the Regularization of Discrete Ill-Posed Problems". In: *SIAM Journal on Scientific Computing* 14.6 (1993), pp. 1487–1503.

[Han98] Per Christian Hansen. *Rank-Deficient and Discrete Ill-Posed Problems*. Society for Industrial and Applied Mathematics, 1998. DOI: 10.1137/1.9780898719697.

[Han99] Per Christian Hansen. "The L-curve and its use in the numerical treatment of inverse problems". In: (1999).

UNIVERSITY OF CALIFORNIA, BERKELEY

Email address: seele@berkeley.edu

UNIVERSITY OF CALIFORNIA, BERKELEY

Email address: miyu02@berkeley.edu