

Guide for inversion of noisy magnetic field using FFT

Eitan Levin

Alexander Y. Meltzer

August 29, 2016

In this note, we explain how to use the code packages `MagInverter2D` and `MagInverter1D` to invert noisy measurements of the magnetic field to obtain the underlying current density. For the theory behind the method, see [3]. In what follows, we assume that we are given:

- `d` - the thickness of the sample,
- `h` - an estimate of the height above the surface of the sample,
- `px` - the pixel size in `px` as a length-2 vector with `px(1)` being the pixel size along the first dimension and `px(2)`, along the second dimension. If the pixels are squares (`px(1)=px(2)`), the value in `px` can be a scalar.
- `B` - the noisy measurement of the normal component of the magnetic field,
- `mu_0` - the value of the permeability of free space. Note that this value sets the units for the lengths, the magnetic field, and the current density.
- `lambdaArr` - an array of values for the Tikhonov regularization parameter λ . To choose an appropriate value for λ , we minimize some function by first sampling it on a relatively sparse grid given by `lambdaArr` and then applying the `fminbnd` solver on intervals containing each local minimum on the grid to refine their value, finally choosing the global minimum.

For the purposes of the examples used in [3] and the demonstrations shown in this note, we assume lengths (`h`,

d , px) are given in microns, magnetic fields (B) are given in Gauss, the current density is given in $mA/\mu m^2$ and $\mu_0 = 4\pi$.

1 Reconstruction of two-dimensional measurements using the direct and stream function methods

To reconstruct currents using the FFT-based methods, we use the functions `reconstructCurrentsFFT_direct` and `reconstructCurrentsFFT_stream` for the direct and stream function methods, respectively. Their syntax is identical apart from the function name:

```
[jx, jy] = reconstructCurrentsFFT_direct(mu_0, h, d, px, data, lambdaArr, plotF, imposeRefBC, p);
```

where we have three options:

- `plotF` - a boolean option to plot the projected SURE function sampled on `lambdaArr`. Note that since it was shown in [3] that the projected SURE method consistently either outperforms or performs similarly to GCV, only the projected SURE method is used in this function.
- `imposeRefBC` - a boolean option specifying whether or not to impose reflections. If the underlying current distribution is known to be fully contained in the measurement window (i.e., the field was measured on a window larger than the sample and contains its entirety), boundary conditions are not needed and the user should set `imposeRefBC=0`. If, however, the current distribution extends beyond the measurement window (as is usually the case), boundary conditions are essential and the user should set `imposeRefBC=1`.
- `p` - a layer of width `ceil(p*size(data))` adjacent to the boundary to be excluded from the projected SURE parameter-choice method to preclude potential edge effects. The default value if `p` is not supplied or if `p` is left empty is `p=.1`.

To demonstrate the above code, we use the function `generateExamplesFFT` which generates the examples used in [3] by using:

```
[JxN,JyN,dataC] = generateExamplesFFT(flag,mu_0,h,d,px);
```

where `flag` $\in \{1, 2, 3\}$ is the example number from [3], the output `JxN`, `JyN` are matrices representing the true x - and y - current densities and `dataC` represents the clean data. Specifically, we generate the second example used in [3] at a height $h = 1 \mu m$ above a sample of width $d = 0.05 \mu m$ and square pixels with side length $1 \mu m$ by setting in the above code `flag=2; mu_0=4*pi; h=1; d=.05; px=1`. We then add white Gaussian noise to the clean data of variance $\sigma^2 = 10^{-2} \max |B|$ to generate the noisy measurement as `data = dataC + 1e-2*max(abs(dataC(:)))*randn(size(dataC))`; A view from above of the clean and noisy data is presented in Fig. 1. We then set `lambdaArr = logspace(-12,3,1e3)` to be our sparse λ array and reconstruct the currents using the direct method, impose boundary conditions by setting `imposeRefBC=1` and use `p=0.1` to reconstruct the currents. The resulting reconstruction along with the true current density, is given in Fig. 2, with the produced error being

$$MSD = \frac{\|j_{x;true} - j_x\|^2 + \|j_{y;true} - j_y\|^2}{\|j_{x;true}\|^2 + \|j_{y;true}\|^2} = 0.0124.$$

We also show the projected SURE function used to estimate the regularization parameter in Fig. 3, generated by setting the plot option `plotF=1`.

If we are interested in comparing the MSD values of the solution obtained by the GCV and projected SURE parameter-choice methods against the optimal one, we can use the functions `reconstructCurrentsFFT_direct_MSD` and `reconstructCurrentsFFT_stream_MSD` for the direct and stream function methods, respectively, with the syntax

```
[Gmin,GPmin,Mmin,alphaG,alphaGP,alphaM] ...
= reconstructCurrentsFFT_direct_MSD(mu_0,h,d,px,data,lambdaArr,plotF,imposeRefBC,p,JxN,JyN)
```

where the additional inputs are the true x - and y - current densities, and the outputs are:

- `Gmin,GPmin` - the MSD value obtained by using the GCV and projected SURE parameter-choice methods, respectively.
- `Mmin` - the minimal MSD value.

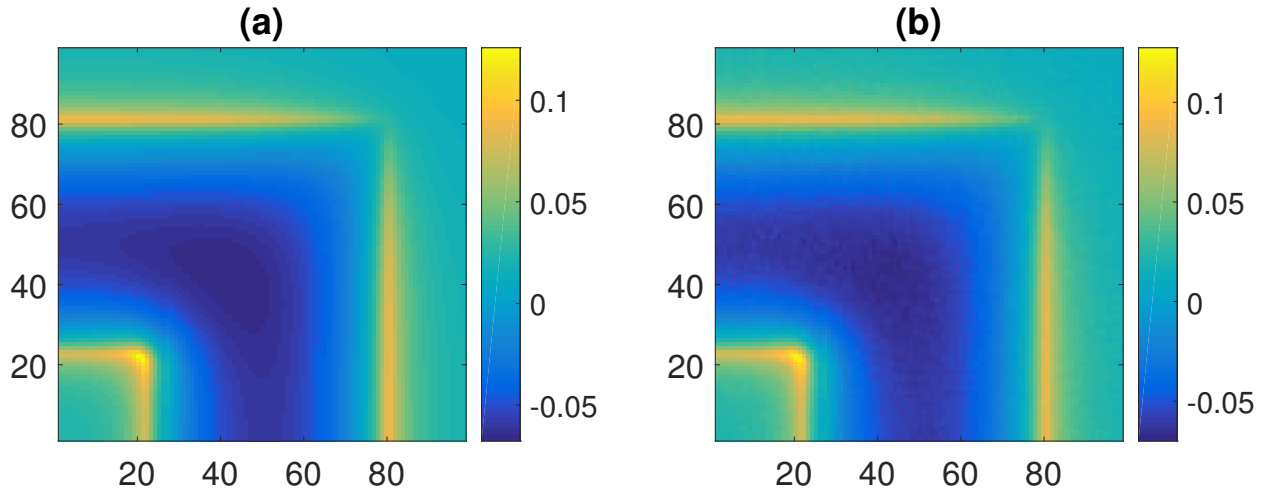


Figure 1: Normal component of the magnetic field for second example (a) without noise, (b) with noise of variance $\sigma^2 = 10^{-2} \max |B_z|$.

- `lambdaG, lambdaGP, lambdaM` - the values of λ found by GCV, projected SURE and the one minimizing the MSD, respectively.

The same syntax applies also to `reconstructCurrentsFFT_stream_MSD` that uses the stream function method. Note that setting `plotF=1` here will generate plots of GCV, P-SURE and MSD versus λ , with appropriate labels for the y-axis to distinguish between them. Note also that here, we must specify `p` and cannot leave it empty. For the example in Fig. 1, we got `Gmin=0.0123`, `GPmin = 0.0124`, `Mmin=0.0123`.

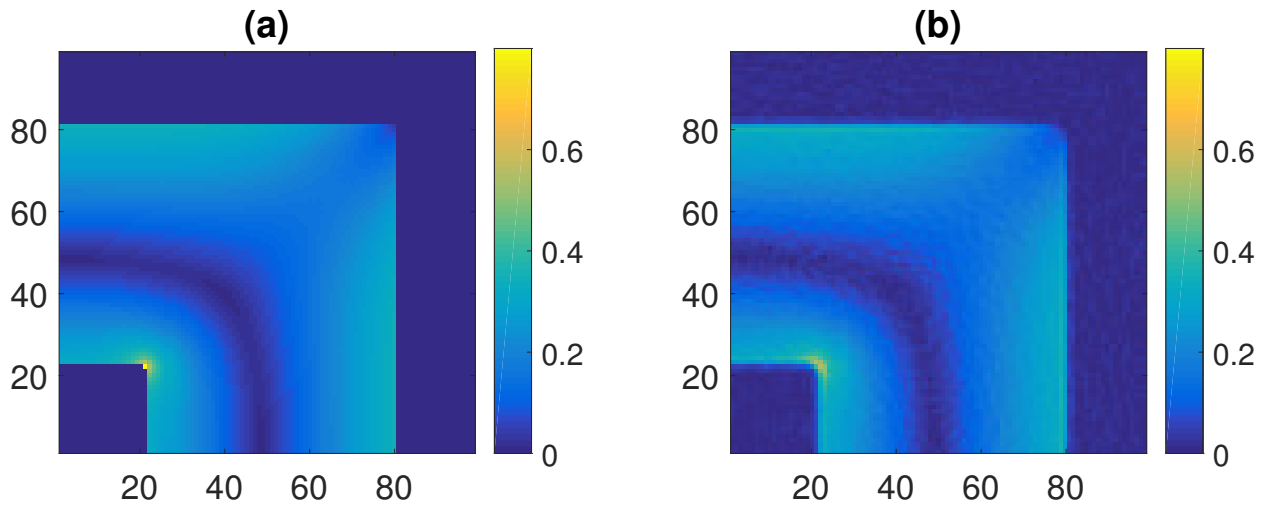


Figure 2: Absolute value of the current densities $|\mathbf{J}| = \sqrt{j_x^2 + j_y^2}$ for (a) the true distribution, (b) the distribution reconstructed from the noisy data Fig. 1.

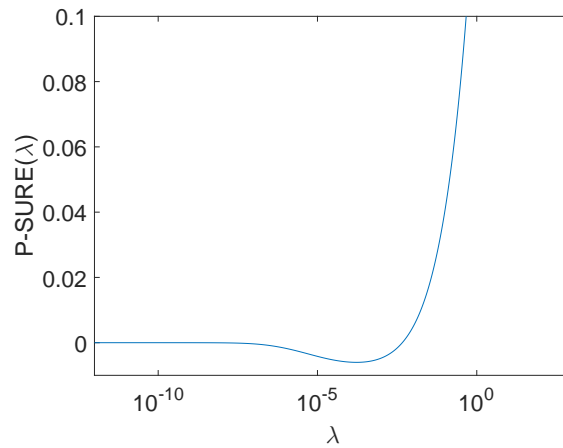


Figure 3: Projected SURE method for the noisy data in Fig. 1.

2 Reconstruction of one-dimensional current distributions from one-dimensional measurements

For a one dimensional problem, we assume that there is only current flowing in the x direction $\mathbf{J} = j_x(y)\hat{x}$ in the thin film, and that we have a noisy measurement of the z component of the field on a portion of the y axis $B_z = B_z(y)$. Note that we have to assume the current is degenerate along the x - axis for the continuity equation $\nabla \cdot \mathbf{J} = \frac{\partial j_x}{\partial x} = 0$ to hold, and that since j_x is degenerate along x , so is B_z . In such a case, the field can be written as the convolution

$$B_z(y) = \int_{-\infty}^{\infty} K_1(x, y, z, d) * j_x(y) dx = W_1(y, z, d) * j_x(y),$$

where the kernel $K_1(x, y, z, d)$ is defined in Eq.(3.30) in [3] while its x integral is given by

$$W_1(y, z, d) = \frac{\mu_0}{2\pi} \text{sign}(y) \left[\arccos \left(\frac{|y|}{\sqrt{y^2 + (z + d/2)^2}} \right) - \arccos \left(\frac{|y|}{\sqrt{y^2 + (z - d/2)^2}} \right) \right].$$

If we use the concept of surface currents as in [3], the resulting one-dimensional kernel would be

$$Z_1(y, z) = \frac{\mu_0}{2\pi} \frac{y}{y^2 + z^2}.$$

For one dimensional problems we do not use FFT because the small scale of the problem and the fact that the continuity equation is automatically satisfied and does constrain our solution enables us to directly solve the equation without assuming any boundary conditions on the problem (such as the periodic boundary conditions imposed by FFT or the reflexive ones used in [3]). The theory behind the algorithms used here is described in [2]. To invert a noisy measurement we use:

```
[lambda,j,fm] = reconstructCurrents1D(mu_0,h,d,px,b,lambdaArr,imposeRefBC,plotF,method);
```

where the variable names have identical interpretation to the 2D case. The additional input `method` can take on four value - 'SS', 'DFA', 'GCV', 'SURE' - and specifies the parameter-choice method to be used for the reconstruction. If `method` is left unspecified or empty, the default method is 'SS'. As an example of a 1D problem, consider the current distribution and its clean and noisy field shown in Fig. 4. To generate this example, we took the true current density to be the true solution of the `shaw` test problem from

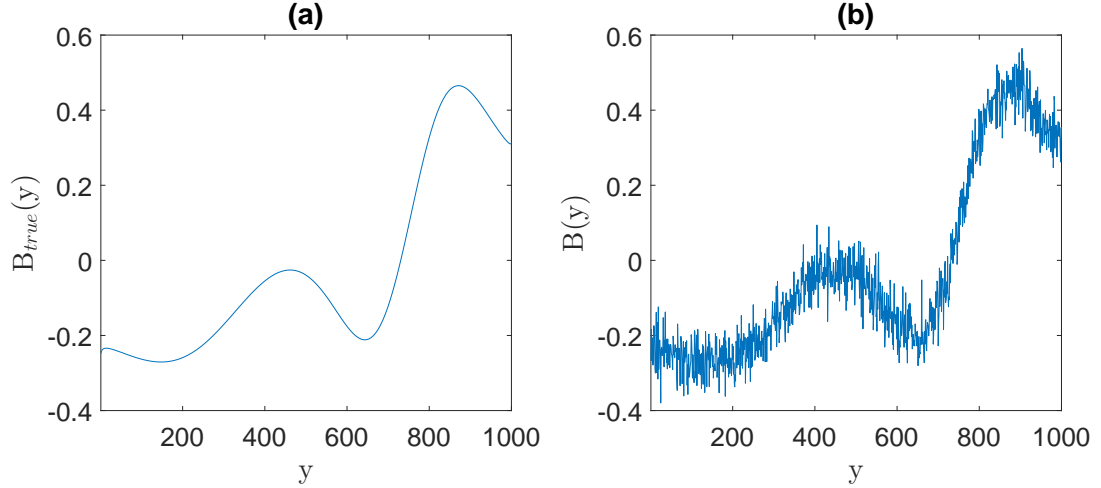


Figure 4: (a) Clean and (b) noisy magnetic fields for one-dimensional example.

Hansen's *Regularization Toolbox* in [1] such that $[\tilde{~}, \tilde{~}, j_true] = \text{shaw}(1e3);$. We then generate the coefficient matrix A as $A = \text{generateTomoMatrix1D}(\mu_0, h, d, px, \text{length}(j_true), 0);$ and compute the true data as $b_true = A*j_true;$. Finally, we contaminate the data with white Gaussian noise of variance $\sigma^2 = 10^{-2} \max |b_true|$ as $b = b_true + \text{sqrt}(1e-2)*\max(\text{abs}(b_true))*\text{randn}(\text{size}(b_true));$. Using the same parameter values as before, we reconstruct the currents without boundary conditions and with the default `method`. The solution is shown in Fig. 5b along with the relative error the solution produced of $\|j_true - j_\lambda\|^2 / \|j_true\|^2 = 8.4104e-04.$

References

- [1] P. C. HANSEN, *REGULARIZATION TOOLS: a Matlab package for analysis and solution of discrete ill-posed problems*, Numer. Algorithms, 6 (1994), pp. 1–35, doi:10.1007/BF02149761, <http://dx.doi.org/10.1007/BF02149761>.
- [2] E. LEVIN AND A. MELTZER, *Estimation of the regularization parameter in linear discrete ill-posed problems using the Picard parameter*, Submitted, (2016), <http://arxiv.org/abs/1607.00938>, arXiv:1607.00938.
- [3] A. MELTZER, E. LEVIN, AND E. ZELDOV, *Reconstruction of two-dimensional currents in a thin film from magnetic field measurement*, Preprint.

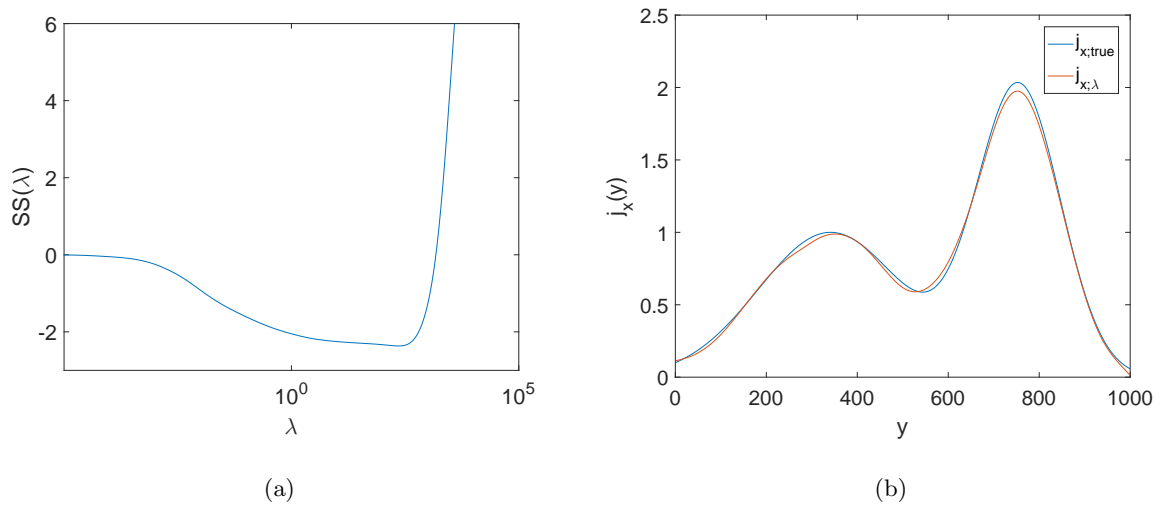


Figure 5: (a) The minimum value of the SS function versus height values λ for the data in Fig. 4. (b) True and reconstructed solution for the data in Fig. 4.