NOTES FOR THE INVERSION CODE

BIRCH

1. Inversions for a scalar model

We assume that the relationship between the input travel-time maps d, the kernels K, the model m, and the noise n is given by

$$d_i(\boldsymbol{x}) = n_i(\boldsymbol{x}) + \iiint K_i(\boldsymbol{x}' - \boldsymbol{x}, z) m(\boldsymbol{x}', z) \, \mathrm{d}\boldsymbol{x}' \mathrm{d}z \ , \forall i \tag{1}$$

where $\boldsymbol{x} = (x, y)$ denotes horizontal position and z is height with $z = 0$ labeling the photosphere. Notice that this assumes translation invariance of the kernel functions. The index i labels the different travel-time maps (i.e., different distances and/or filters).

The goal of the inversion is to use knowledge of d and K and the statistics of n to estimate the model m. We will begin by applying the MCD approach (Jacobsen et al. 1999) to equation (1). In the horizontal Fourier domain we have:

$$d_i(\boldsymbol{k}) = n_i(\boldsymbol{k}) + (2\pi)^2 \int K_i^*(\boldsymbol{k}, z) m(\boldsymbol{k}, z) \, \mathrm{d}z \ , \forall i, \forall \boldsymbol{k} \tag{2}$$

where $\boldsymbol{k} = (k_x, k_y)$ is the horizontal wavevector and the superscript * denotes the complex conjugate.

Assume that at each $\boldsymbol{k}$, the z dependence of the model can be reasonably approximated as a sum over some basis functions $\phi(z)$ (e.g. b-splines)

$$m(\boldsymbol{k}, z) = \sum_{j=1}^{N} b_j(\boldsymbol{k}) \phi_j(z) \ , \tag{3}$$

where the choice of the set of basis functions will depend on the problem we are trying to solve.

Equation (2) is then

$$d_i(\boldsymbol{k}) = n_i(\boldsymbol{k}) + \sum_j M_{ij}^*(\boldsymbol{k}) b_j(\boldsymbol{k}) \ , \forall i, \forall \boldsymbol{k} \tag{4}$$

where the matrix M is given by the projection of the kernel of index i against the basis function of index j:

$$M_{ij}(\boldsymbol{k}) = (2\pi)^2 \int K_i^*(\boldsymbol{k}, z) \phi_j(z) \mathrm{d}z \tag{5}$$

1.1. **RLS inversion.** In general, the matrix M will be very badly conditioned and we can't just solve the equations directly. There are a number of options. Here we will use the RLS approach with a Tikhonov-type regularization (e.g. [Kosovichev, 1996], [S. Couvidat et al., 2004] in the context of local helioseismology). We will look to minimize a combination of the misfit between the model and data while simultaneously controlling some other aspect of the model (e.g., amplitude or smoothness). Define a cost function

$$X[\mathbf{b}] = \sum_i \frac{1}{\sigma_i^2}\left[d_i - \sum_j M_{ij} b_j\right]^2 + \lambda^2 \sum_{ij} b_i R_{ij} b_j \tag{6}$$

where λ is a regularization parameter, R is symmetric positive-definite matrix (more later on this), and σ_i is the noise estimate on measurement d_i (in reality, we need to treat the the full error covariance, this is a straightforward generalization and can be taken care of later). The first term in the above equation is the misfit between the model and the data. The second term is the "regularization term", which is a penalty against some property of the solution b. The parameter λ controls the relative importance of these two terms.

We minimize X by demanding $\partial_{b_k} X = 0, \forall k$. This gives

$$0 = -\sum_i \frac{1}{\sigma_i^2}\left[d_i - \sum_j M_{ij} b_j\right] M_{ik} + \lambda^2 \sum_j R_{kj} b_j \tag{7}$$

define for later the diagonal matrix $C_{ij}^{-1} = 1/\sigma_i^2$. Then the above reduces to the matrix equation

$$\left[M^T C^{-1} M + \lambda^2 R\right] b = M^T C^{-1} d \tag{8}$$

and

$$\left[M^H C^{-1} M + \lambda^2 R\right] b = M^H C^{-1} d \tag{9}$$

This is the equation that we want to solve at each $\boldsymbol{k}$. Note, M^H is the transpose and complex conjugate of M. Consider the case: $K = i$, where $i = \sqrt{-1}$, $d = 1$, and switching notation, $K = M$, $m = b$, then minimizing

$$\begin{aligned}
X^2 &= ||Km - d||_2^2 \\
&= (Km - d)^*(Km - d) \\
&= K^*Km^*m - K^*m^*d - Kmd^* + d^*d \\
&= m^*m - (-i)m^* - im + 1 \\
&= m = m_r + im_i \\
&= m^*m + im^* - im + 1 \\
&= m_r^2 + m_i^2 + 2m_i + 1 \\
&= m_r^2 + (m_i + 1)^2, m = -i, m_r = 0, m_i = -1
\end{aligned}$$

1.2. **The regularization matrix.**

1.2.1. *amplitude.* Suppose we want to control the amplitude of the solution $m(\boldsymbol{x}, z)$. We want the cost function be an approximation to

$$I[m] = \iiint [m(\boldsymbol{x}, z)]^2 \, \mathrm{d}\boldsymbol{x}\mathrm{d}z \ . \tag{10}$$

In the horizontal Fourier domain, it is the same

$$I[m] = \iiint [m(\boldsymbol{k}, z)]^2 \, \mathrm{d}\boldsymbol{k}\mathrm{d}z \ . \tag{11}$$

At each $\boldsymbol{k}$, we need to control $\int m(\boldsymbol{k}, z)^2 \mathrm{d}z$. Insert the expansion in terms of basis functions, and we get

$$I[\mathbf{b}] = \sum_{ij} b_i \left[\int \phi_i(z)\phi_j(z)\mathrm{d}z \right] b_j \tag{12}$$

which is in the form we want, and we see that R is given by

$$R_{ij} = \int \phi_i(z)\phi_j(z)\mathrm{d}z \tag{13}$$

1.2.2. *first derivative.*

1.2.3. *second derivative.*

1.3. **First Project for RLS inversion of scalar.** First thing to look at is given a matrix M, and set of data d, a set of errors σ, and regularization matrix R, solve equation (8) for a number of λ. Here we are working at a single $\boldsymbol{k}$ (each $\boldsymbol{k}$ has z-dimension only). Compute averaging kernels (see below) and error estimates for the output models (see below).

1.4. **Second project for RLS inversion of scalar.** Generalize the code from previous section to solve for many $\boldsymbol{k}$. we need to talk about formats for the input files, etc.

Solve equation (9) for a number of λ. Compute averaging kernels (see below) and error estimates for the output models (see below).

1.5. **Third project.** Generalize to a full noise covariance matrix.

$$\boldsymbol{x} = (x, y) \tag{14}$$

$$n_i(\boldsymbol{x}) \tag{15}$$

The noise covariance is an input file and it contains $C_{ij}(x, y)$. Here, i is an index into the different travel-time maps.

$$C_{ij}(x, y) = C_{ij}(\boldsymbol{x}) = E[n_i(\boldsymbol{x}')n_j(\boldsymbol{x}' + \boldsymbol{x})] \tag{16}$$

or

$$C(r_2 - r_1) = E[n(r_2)n(r_1)] \tag{17}$$

which is the expected input file for the noise covariance. From here, there are two options for transforming and inverting the noise covariance.

(1) $C^{(1)}(\boldsymbol{k}) = FFT[C(r_2 - r_1)]$
(2) $C^{(2)}(\boldsymbol{k}) = E[n^*(\boldsymbol{k})n(\boldsymbol{k})] = factor C^{(1)}(\boldsymbol{k})$

We follow the second option and find the factor,

$$\begin{aligned} C_{ij}(\boldsymbol{k}) &= E[n_i^*(\boldsymbol{k})n_j(\boldsymbol{k})] \\ &= \frac{1}{(2\pi)^4} \sum_{x_1} \sum_{x_2} E[n_i(x_1)n_j(x_2)] e^{ik\cdot(x_1 - x_2)} \\ &= \frac{N^2}{(2\pi)^2} \frac{1}{(2\pi)^2} \sum_x \Lambda_{ij}(x) e^{-ik\cdot x} \\ C_{ij}(n) &= \frac{N^2}{(2\pi)^2} \Lambda_{ij}(\boldsymbol{k}) \end{aligned}$$

The factor is $\frac{N^2}{(2\pi)^2}$

1.6. **Averaging Kernels.** Averaging kernels A describe the way in which the output model m is expected to be related to the actual (true) model $\tilde{m}$. The expected value of b is given by

$$E[b] = FM\tilde{b} \tag{18}$$

with $F = \left[M^H C^{-1} M + \lambda^2 R\right]^{-1} M^H C^{-1}$. The operator E denotes expected value (ensemble average) and $\tilde{b}$ is the true model. The above shows that even if we average together many inversion results, we don't in general recover the true expansion coefficients $\tilde{b}$. The relationship between the expectation for b and the true coefficients is $E[b] = A\tilde{b}$ with $A = FM$. The matrices A are the averaging kernels. In an inversion that we did not have to regularize, we would have the A equal to the identity (insert $\lambda = 0$ into the above)

1.7. **Noise Propagation.** Look for the formal error in b due to the noise n:

$$b_i - E[b_i] = F_{ij} n_j \tag{19}$$

so then, old equation,

$$\text{Cov } [b_i, b_j] = E[F_{ik} n_k, F_{jl}, n_l] = F_{ik} C_{kl} F_{jl} \tag{20}$$

New equation,

$$\text{Cov } [b_i, b_j] = E[F_{ik}^* n_k^*, F_{jl}, n_l] = F_{ik}^* C_{kl} F_{jl} \tag{21}$$

so the full error covariance of the output FCF^H. This is missing a factor since $C^{(2)}(\boldsymbol{k}) = E[n^*(\boldsymbol{k})n(\boldsymbol{k})] = \frac{N^2}{(2\pi)^2} C^{(1)}(\boldsymbol{k})$.

$$
\begin{aligned}
\left[M^H C^{-1} M + \lambda^2 R\right] b &= M^H C^{-1} d \\
b(k) &= F(k) n(k) \\
C(x') = E[b(x) b(x+x')] &= \left(\frac{2\pi}{N}\right)^4 \sum_{k_1} \sum_{k_2} F^*(k_1) F(k_2) E\left[n^*(k_1) n(k_2)\right] e^{ik_2(x+x') - ik_1 x} \\
E\left[n^*(k_1) n(k_2)\right] &= C(k_1) \delta_{k_1,k_2} \\
&= \left(\frac{2\pi}{N}\right)^4 \sum_k F^* C F e^{ikx'} \\
&= \left(\frac{2\pi}{N}\right)^2 \left(\frac{2\pi}{N}\right)^2 \sum_k F^* C F e^{ikx} \\
&= \left(\frac{2\pi}{N}\right)^2 FFT\left[F^* C F\right]
\end{aligned}
$$

Now, the full error covariance of the output is $\left(\frac{2\pi}{N}\right)^2 \left[F C F^H\right]$.

2. MCD RLS inversion for many model params

Now generalize to the case where the model is a set of values at each location $(\boldsymbol{x}, z)$. Examples include inversions for a flow field, or for multiple structure variables (e.g. sound speed, density, pressure).

References

B.H. Jacobsen, I. Møller, J.M. Jensen, and F. Effersø. Multichannel Deconvolution, MCD, in Geophysics and Helioseismology. *Phys. Chem. Earth*, 24:215–220, 1999.

A.G. Kosovichev. Tomographic Imaging of the Sun's Interior. *Astrophys. J. Lett.*, 461:L55, April 1996.

S. Couvidat, A.C. Birch, A.G. Kosovichev, and J. Zhao. Three-dimensional Inversion of Time-distance Helioseismology Data: Ray-path and Fresnel-zone Approximations. *Astrophys. J.*, 607:554–563, 2004.