User Guide
(MATLAB Version `R2019a`)

Amrina Ferdous
amrinaferdous@u.boisestate.edu
Boise State University, Boise, ID 83725

December 20, 2019

1 Introduction

This code produces a solution of an ill-posed nonlinear problem of the form:

$$\begin{aligned} u(x) &= g(z(w)) \qquad (1)\\ &= \int f(z(w))dw \qquad (2) \end{aligned}$$

where g is given and we are trying to recover continuous $z(w)$ with discrete observation of $u(x)$.

Considering discrete data with continuous parameter case, the authors of [1] use the following back-propagation to recover $z(w)$. They devised the following iterative algorithm that is continuous in w.

$$\begin{aligned} \hat{z}_{k+1}(w) = z_0 + \int dw' \sum_i \sum_j C_z(w,w') G_k^i(w')(S^{-1})^{ij}[d^j - g^j(\hat{\mathbf{z}}_k) \\ + \int dw'' G_k^j(w'').[\hat{z}_k(w'') - z_0(w'')] \qquad (3) \end{aligned}$$

We are coding the above equation (3). Here $G_k^i(w)$ is the derivative of $g(x^i, w, z_k(w))$, z_0 represents the prior parameter estimate and $C_z(w,w')$ is the prior covariance. The authors define $C_z(w,w')$ as follows:

$$C_z(w,w') = \sigma^2 exp[-\frac{1}{2}\frac{(w-w')^2}{\Delta^2}] \qquad (4)$$

Note that here $\mathbf{G}^T$ in the continuous setting is the more general adjoint G^*. That adjoint G^* is defined by $\langle Gr, t\rangle = \langle t, G^* g\rangle$ for all r, t in a Hilbert space [1].

2 User Interface

If you want to run the code, you need to run the `"frontier.m"` file only. Note that the `"frontier.m"` is calling the `"inv_DDCP.m"` file so that we could obtain our output `zhat` along with its validations and plot those. For more details about `zhat`, please see the Tarantola's paper [1] and `"inv_DDCP.m"` file where we have coded the main part of Tarantola's algorithm. More general, the name `"inv_DDCP.m"` was chosen to represent "inverse solution of discrete data and continuous parameter".

The user interface of obtaining the "inverse solution of discrete data and continuous parameter", $\hat{z}(w)$ (see equation (3)) between two media of different density using as data the anomaly is as follows:

```
zhat=inv_DDCP(w,x,d,sigma_d,sigma,theta,K,@Gfun,@ffun)
```

where

Output Variable :

`zhat` : Output vector $\hat{z}_{k+1}(w)$ (see equation (3)). Here we are recovering the $\hat{z}(w)$ at the $(K+1)$-th iteration.

Input Variables :

`w` : Discrete grid of the independent variable for the continuous parameter. A finer grid will produce more accurate approximation to $z(w)$. This w needs to be a column vector. For more details, please see the run.m file.

`x` : Location where data are observed. This x needs to be a column vector. For more details, please see the run.m file.

`d` : Data vector considering it as a single column floating point numbers in units. This data is imported from .txt file. In my case, the name has been specified as "data.txt". Please change your datafile path if needed.

`sigma_d` : Constant standard deviation of the data error.

`sigma` : Prior uncertainty of $z_0(w)$ in units. See σ in equation (4). Changing `sigma` is the reason of having different prior covariances.

`theta` : Spread in the prior covariance. See Δ in equation (4).

`K` : Number of iteration of the algorithm (3).

`Gfun` : Model which has been implemented as the above equation (3). $G_k^i(w)$ is the derivative of $g(x^i, w, z_k(w))$. Here "Gfun.m" file contains the model. Writing a different model in the "Gfun.m" file but keeping the "inv_DDCP.m" file unchanged would provide a different result. For example: if we change the value of the "Depth between the surface and the subsurface", then we can easily form a different Gfun and end up having a different result.

`ffun` : Integrand. See equation (1). Writing a different "ffun.m" file but keeping the "inv_DDCP.m" file unchanged would provide a different result. For example: if we change the value of the "Depth between the surface and the subsurface", then we can easily form a different ffun and end up having a different result.

3 Example

For an example, we are considering "the gravitational inverse problem of obtaining the shape of a frontier $z(w)$ between two media of different densities using as data the anomaly at the surface" [1]. It is the example of an ill-posed non-linear problem from [2].

- We have considered the Tarantola paper's

$$u(x) = \int_a^b log \frac{(x-w)^2 + H^2}{(x-w)^2 + [H - z(w)]^2} dw \tag{5}$$

  where $a = wmin$, $b = wmax$, H is the depth between the surface and the subsurface.

  Also, according to Tarantola's example, we have considered the following $g(x^i, w, z_k(w))$ and `ffun` i.e, f

$$g = \int f dw = \int log \frac{(x-w)^2 + H^2}{(x-w)^2 + [H - z(w)]^2} dw \tag{6}$$

- We have created a continuous true $z(w)$ called *ztrue* using the concept of the Tarantola paper [1] where

$$ztrue = \begin{cases} 0 & \text{if } 0 < w < 40 \\ 2.5\, e^{-5\frac{(w-50)^2}{m}} & \text{if } 40 < w < 60 \\ 0 & \text{if } 60 < w < 100. \end{cases}$$

- We form a true $u(x)$ (see equation (1)) called $utrue$ where $utrue = \int f(ztrue)dw$. Then we create our own data set d where

  $$d = u(x) + \varepsilon \tag{7}$$

  and ε follows a Gaussian distribution with mean 0 and standard deviation $\sqrt{0.001}$. Users can create their own data set d using the above equation (7) according to their particular problem set. Note that in the `"inv_DDCP"` folder we also have `"creatingNEWdata.m"` file so that user could create other different data sets according to my particular example. With new data d, our results will be different. To avoid replacement issue, we have chosen different names in `"creatingNEWdata.m"` file. If you are using different data with a different filename, then please change the filename in the "Loading data" command at our `"frontier.m"` file.

- We save our data set d as "data.txt" file which is a column vector and it is inside our `"inv_DDCP"` folder.

- We form the `zhat=inv_DDCP(w,x,d,sigma_d,sigma,theta,K,@Gfun,@ffun)` and plot this `"zhat"` (see Figure 1) considering the following values of our variables.

An example: Recovering the frontier

```
m= 100; %Points of integration grid of Tarantola paper's algorithm
wmin=0; % [km] Lower limit of the integration domain
wmax=100; % [km] Upper limit of the integration domain
w = linspace(wmin,wmax,m); % Defining the integration grid w
% Loading data
d= load("data.txt"); %[km] data vector (column vector)
n = length(d); %Total number of data
x = linspace(wmin,wmax,n)'; %Location where data are observed
sigma_d= √0.001; %[km] Constant standard deviation of the data error
sigma = 5; %[km] Prior uncertainty
theta = 1; %[km] Spread in the prior covariance
H = 10; %[km] Depth between the surface and the subsurface
K=10; %Number of iteration
```

Note that we have considered our `Gfun` i.e., $G_k^i(w)$ according to [1] where

$$G_k^i(w) = \frac{2(H - z_k(w))}{(x^i - w)^2 + [H - z_k(w)]^2} \tag{8}$$

The plot of our `zhat` looks like:

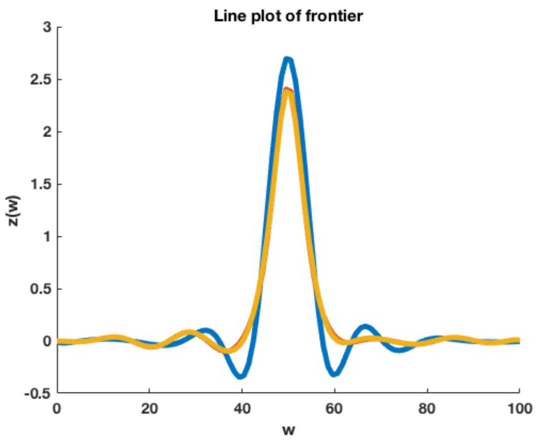

Figure 1: Example: Recovering the frontier $z(w)$.

Validation of my example

To validate our `zhat`, we have created a section in our `"frontier.m"` file. When users run this `"frontier.m"` file, they will see the plot of our resulting frontier, `zhat` along with its validations. In this file, we have formed a ground truth i.e., `ztrue` and plotted this `ztrue` with our estimated `zhat` to see the accuracy in z (see figure ??) . Similarly, we have created a `utrue`, and estimated a `uhat` and plotted this `utrue` with our estimated `uhat` (see figure ??). The file named as "`g_small_fun.m`" is also a part of our "`frontier.m`" file. We have formed this "`g_small_fun.m`" according to the equation (1). For more details, please read [1].

The plots to check the accuracy in z (1st graph) and prediction (2nd graph) are as follows:

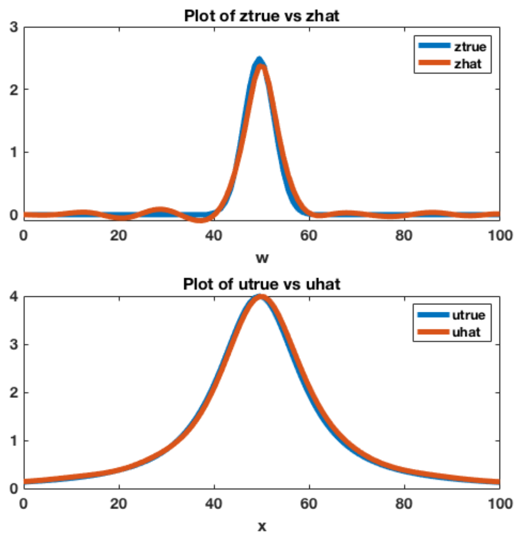

Figure 2: Accuracy checking in z (1st graph) and Accuracy checking in prediction (2nd graph)

In addition, we have calculated the “relative error in z" and the “relative error in prediction" as follows:

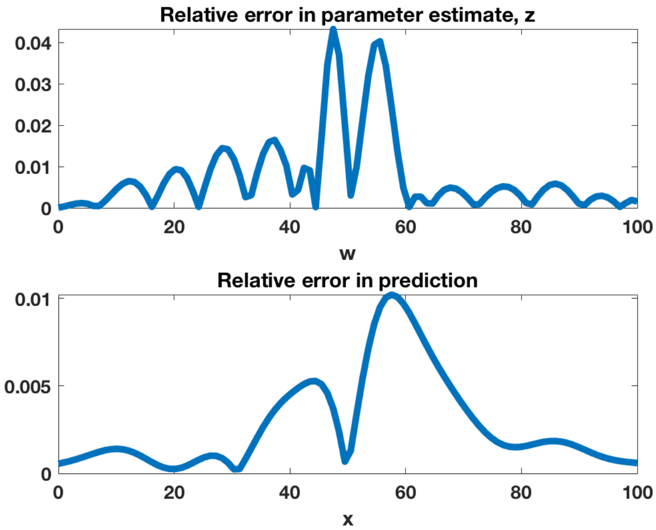

Figure 3: Relative Error in z (1st graph) and Relative Error in prediction (2nd graph)

References

[1] A. Tarantola, B. Valette. *Generalized Nonlinear Inverse Problems Solved Using the Least Squares Criterion*. Institut de Physique du Globe de Paris, 75005 Paris, France. Reviews of Geophysics and Space Physics, Vol. 20, No. 2, pages 219 – 232; May 1982.

[2] A. Tikhonov, V. Arsenine. *Methodes de resolution de problemes mal poses*. Editions MIR, Moscow, 1976.