Functions

- **blur1D.m:** Operator which applies Gaussian blur to a 1D signal.
- **E.m:** Evaluates the lower level objective function and its gradient.
- **Experiment2_script.m:** Code for the second experiment for the Recycling Krylov methods strand. It uses the computed optimal alpha to regularise a signal which was not in the training set and investigates how well it optimises it.
- **gd_convex.m:** Gradient descent for the lower-level objective function, which is convex. Contains the a-posteriori stopping criterion for strongly convex problems.
- **gd_non_convex_basic.m:** Gradient descent for the upper level objective function, which is non-convex. Uses an adaptive step size, via backtracking line search. The linear system in the gradient is solved using GMRES.
- **gd_non_convex_basic.m:** This is same as **gd_non_convex_basic.m:**, but has been modified such that the linear system in the gradient is solved using Recycled GMRES.
- **generate_signal_1D.m:** Creates a number of 1D signals with jumps at random locations. Maximum number of jumps is 10. Amplitude ranges between $[-2, 2]$. Input n the length of the signal and `total`, the number of signals required. The function outputs `signals`, a matrix which contain the individual signals in its columns, and `k_vals`, the number of jumps in each signal.
- **grad_tests.m:** Tests the lower and upper level gradients.
- **L_gmres.m:** Function which computes the value for the loss function and also its gradient. The linear system contained within the computation of the gradient is solved using GMRES. The function **solve_lower.m** is contained within this function, which computes the reconstructed images.
- **L_recycle.m:** Same function as **L_gmres.m**, except the linear system contained within the computation of the gradient is solved using Recycled GMRES.
- **L_val.m:** Function which only computes the value for the loss function.
- **main_bilevel_script.m:** The main script for the bilevel code. Run this script to run the main bilevel experiment, experiment 1. The parameters for the problem are set here and the training data is created using make_signals.m. The optimisation procedure is ran twice, one calling **solve_upper_basic.m** which uses GMRES to solve the linear system contained within the gradient, and once using **solve_upper_recycle.m** which uses Recycled GMRES to solve the linear system. The results are visualised.
- **make_signals.m:** Creates the training set of ground truths and corresponding noisy signals. Uses the function **generate_signal_1D.m** to generate the ground truth signals. Gaussian noise is added to this to create the noisy data.
- **rho.m**: This term is part of the regularisation term. This function outputs $\rho(x)$, and its first and second derivative, all evaluated at a given x.
- **solve_lower.m:** Generates a reconstruction, for a given value of α. This is achieved by running **gd_convex.m** to minimise the lower level objective, and obtaining u which minimises it.

- **solve_upper_basic.m:** Uses the functions **L_basic.m** and **L_val.m** to compute the objective function and its gradient and assigns them to a function handle. Uses **gd_non_convex_basic.m** to minimise the objective function.
- **solve_upper_recycle.m:** Uses the functions **L_recycle.m** and **L_val.m** to compute the objective function and its gradient and assigns them to a function handle. Uses **gd_non_convex_recycle.m** to minimise the objective function.