UNIVERSITY OF CAMBRIDGE

Practical Optimisation
Norm Approximation

5568D

30th November 2019

1 Problem 1

The first problem consists of investigating the common optimisation task:

$$\text{minimise} \quad ||\mathbf{Ax} - \mathbf{b}||,$$

where $\mathbf{A} \in R^{m \times n}$ and the $\mathbf{b} \in R^m$ are given as problem data, $\mathbf{x} \in \mathcal{R}^n$ is the variable and $||.||$ is the norm on $\mathcal{R}^m$. A vector $\mathbf{r} \in \mathcal{R}^m$ called the residual vector is defined as:

$$\mathbf{r} = \mathbf{Ax} - \mathbf{b}$$

In the first problem, a solution of $\mathbf{Ax} \approx \mathbf{b}$ is found by norm approximation. Firstly a definition of norm and its approximation is given. Then it is showed how this type of problem can be cast as a *linear programming(LP)* problem. After the theory has been defined some real data norm approximations will be studied.

1.1 Task a

A norm is non-negative object, real-valued number that measures 'how big' something is. The norm of $||\mathbf{Ax} - \mathbf{b}||$ is the same as the norm of $||\mathbf{r}||$. A l_p norm is defined as:

$$l_p : \quad ||\mathbf{r}||_p = (|r_1|^p + \ldots + |r_m|^p)^{1/p} \tag{1}$$

The main norms discussed in this report are l_1, l_2 and l_∞. The approximations of these norms are given by these equations:

$$\begin{aligned} l_1 : \quad & ||\mathbf{r}||_1 = |r_1| + \ldots + |r_m| & (2)\\ l_2 : \quad & ||\mathbf{r}||_2 = (r_1^2 + \ldots + r_m^2)^{1/2} & (3)\\ l_\infty : \quad & ||\mathbf{r}||_\infty = \max(|r_1|, \ldots |r_m|) & (4) \end{aligned}$$

l_2 norm is especially important as it is widely used method especially in finding the maximum likelihood of a data-set. Instead of looking at the residual vector $\mathbf{r}$ the original problem is considered and the l_2 norm is expanded:

$$||\mathbf{Ax} - \mathbf{b}||_2^2 = \mathbf{x}^T\mathbf{A}^T\mathbf{Ax} - 2\mathbf{b}^T\mathbf{Ax} + \mathbf{b}^T\mathbf{b} \tag{5}$$

The equation above represents an unconstrained quadratic function which is a convex optimisation problem. The problem of minimising this type of convex quadratic function is called *Least-Squares* approximation([1, Chapter 2]) and has an analytical solution which is derived below:

$$\begin{aligned} \mathbf{f}(\mathbf{x}) &= \mathbf{x}^T\mathbf{A}^T\mathbf{Ax} - 2\mathbf{b}^T\mathbf{Ax} + \mathbf{b}^T\mathbf{b}\\ \Delta\mathbf{f}(\mathbf{x}) &= 2\mathbf{A}^T\mathbf{Ax} - 2\mathbf{A}^T\mathbf{b} = 0 \qquad (6)\\ \mathbf{x} &= (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b} \end{aligned}$$

Instead of having one objective to minimise, one can introduce two objectives which act as trade-off between each other. One method of solving bi-criterion is using regularisation. Regularisation can be thought of as the prior knowledge of the optimum solution [1, Chapter 6]. One of the most common regularisation is called *Tikhonov Regularisation* which has the following quadratic optimisation problem:

$$\text{minimise} \quad ||\mathbf{Ax} - \mathbf{b}||_2^2 + \lambda||\mathbf{x}||_2^2 \tag{7}$$

This *Tikhonov Regularisation* problem has an analytical solution:

$$\mathbf{x} = (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{A}^T\mathbf{b} \tag{8}$$

In **Problem 3** details of the properties of the *Tikhonov Regularisation* will be discussed along with another method of regularisation

1.2 Task b

Now that the norms approximation have been defined. It is time to look at how these norm approximation can be transformed into *LP* problems. The main reason why it is desired to solve LP problem is the fact that there are very effective and fast methods of solving them such as the *simplex method* and *interior point method* [1, Chapter 1]. More specifically because the *LP* problems are convex problems, it can be differentiated and therefore one might implement very effective first-order or second-order methods.
Firstly, the l_1 norm can be cast as a *LP* problem if the form of the objective function is:

$$\begin{aligned} \text{minimise} \quad & \mathbf{1}^T \mathbf{y} \\ \text{subject to} \quad & -\mathbf{y} \preceq \mathbf{A}\mathbf{x} - \mathbf{b} \preceq \mathbf{y} \end{aligned} \tag{9}$$

where $\mathbf{y} \in \mathcal{R}^m$ and $\mathbf{x} \in \mathcal{R}^n$. This is called the sum of residuals and in the context of estimation, a robust estimator[1, Chapter 6]. To put the new expression into a standard form it can be expressed as:

$$\begin{aligned} \text{minimise} \quad & \underbrace{\begin{pmatrix} \mathbf{0} \\ \mathbf{1} \end{pmatrix}^T}_{\tilde{\mathbf{c}}^T} \underbrace{\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}}_{\tilde{\mathbf{x}}} \\ \text{subject to} \quad & \underbrace{\begin{pmatrix} \mathbf{A} & -\mathbf{I} \\ -\mathbf{A} & -\mathbf{I} \end{pmatrix}}_{\tilde{\mathbf{A}}} \underbrace{\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix}}_{\tilde{\mathbf{x}}} \preceq \underbrace{\begin{pmatrix} \mathbf{b} \\ -\mathbf{b} \end{pmatrix}}_{\tilde{\mathbf{b}}} \end{aligned} \tag{10}$$

which has a standard form of $\tilde{\mathbf{A}}\tilde{\mathbf{x}} \preceq \tilde{\mathbf{b}}$, where $\tilde{\mathbf{A}} \in \mathcal{R}^{2m \times (n+m)}$, $\tilde{\mathbf{b}} \in \mathcal{R}^{2m}$ and $\tilde{\mathbf{x}}$, $\tilde{\mathbf{c}} \in \mathcal{R}^{n+m}$.
Now, the l_∞ is cast as a *LP* problem if:

$$\begin{aligned} \text{minimise} \quad & y \\ \text{subject to} \quad & -y\mathbf{1} \preceq \mathbf{A}\mathbf{x} - \mathbf{b} \preceq y\mathbf{1} \end{aligned} \tag{11}$$

where $y \in \mathcal{R}$ and $\mathbf{x} \in \mathcal{R}^n$.This again can be expressed into a standard from:

$$\begin{aligned} \text{minimise} \quad & \underbrace{\begin{pmatrix} \mathbf{0} \\ 1 \end{pmatrix}^T}_{\tilde{\mathbf{c}}^T} \underbrace{\begin{pmatrix} \mathbf{x} \\ y \end{pmatrix}}_{\tilde{\mathbf{x}}} \\ \text{subject to} \quad & \underbrace{\begin{pmatrix} \mathbf{A} & -\mathbf{1} \\ -\mathbf{A} & -\mathbf{1} \end{pmatrix}}_{\tilde{\mathbf{A}}} \underbrace{\begin{pmatrix} \mathbf{x} \\ y \end{pmatrix}}_{\tilde{\mathbf{x}}} \preceq \underbrace{\begin{pmatrix} \mathbf{b} \\ -\mathbf{b} \end{pmatrix}}_{\tilde{\mathbf{b}}} \end{aligned} \tag{12}$$

which again has a standard form of $\tilde{\mathbf{A}}\tilde{\mathbf{x}} \preceq \tilde{\mathbf{b}}$, where $\tilde{\mathbf{A}} \in \mathcal{R}^{2m \times (n+1)}$, $\tilde{\mathbf{b}} \in \mathcal{R}^{2m}$ and $\tilde{\mathbf{x}}$, $\tilde{\mathbf{c}} \in \mathcal{R}^{n+1}$.
By transforming the norm-approximation into *LP* form a solution of the optimum value $\mathbf{x}^*$ can be found using different methods such as the *dual-simplex method* or the *interior point method*.

1.3 Task c

Using the *LP* standard form defined above, l_1,l_2 and l_∞ minimisation is performed on $||\mathbf{A}\mathbf{x} - \mathbf{b}||$ where different matrices ($\mathbf{A}$) and different vectors ($\mathbf{b}$) are provided. For l_1-norm and l_∞-norm a dual-simplex method is used while for the l_2-norm a linear solver method is implemented. The size of the problem grows each time exponentially by a factor of 4. In the *table* below a summary of the results of the norms minimisation is provided along with the computational time:

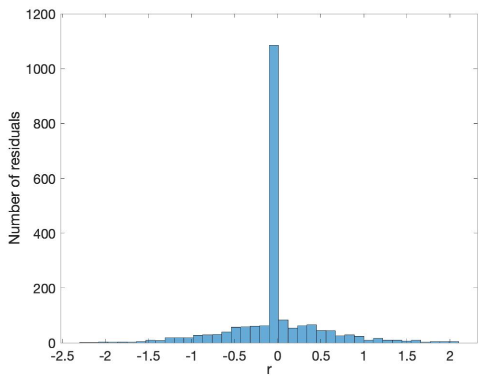
Data-set	$\min \|\|\mathbf{Ax}-\mathbf{b}\|\|_1$	$\min \|\|\mathbf{Ax}-\mathbf{b}\|\|_2$	$\min \|\|\mathbf{Ax}-\mathbf{b}\|\|_\infty$	l_1 runtime(s)	l_2 runtime(s)	l_∞ runtime(s)
$(\mathbf{A1}, \mathbf{b1})$	10.26	2.389	0.6081	0.0115	0.0002	0.0217
$(\mathbf{A2}, \mathbf{b2})$	33.61	4.404	0.5796	0.0364	0.0006	0.1049
$(\mathbf{A3}, \mathbf{b3})$	143.26	9.390	0.6135	1.8551	0.0054	0.8291
$(\mathbf{A4}, \mathbf{b4})$	277.18	12.91	0.5936	19.6887	0.0183	8.0418
$(\mathbf{A5}, \mathbf{b5})$	571.64	18.55	0.6035	246.0085	0.2063	88.8865

Table 1: Data containing the norm minimisation values and the running time

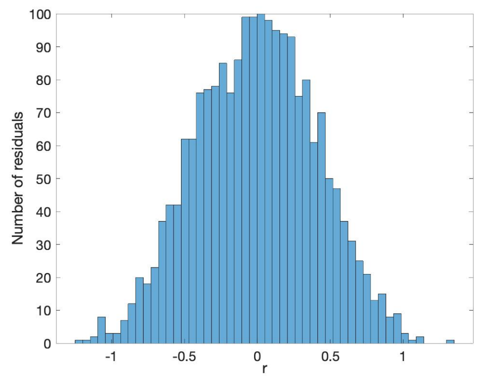
It is observed that values for minimisation of the norms l_1 and l_2 increase with the size of the problem while the solution of the l_∞ minimisation stays almost the same as the size of the problem increases. Looking at the timing required to solve the minimisation it is observed that the l_1 minimisation takes significantly longer. This is because of the fact the l_1 norm is not smooth and therefore it is required to search for the solution which takes significantly longer. It is also observed that l_∞ takes significantly longer than the l_2 but less than the l_1 norm. This is because in l_∞ minimisation it is required only to look at the largest element.

1.4 Task d

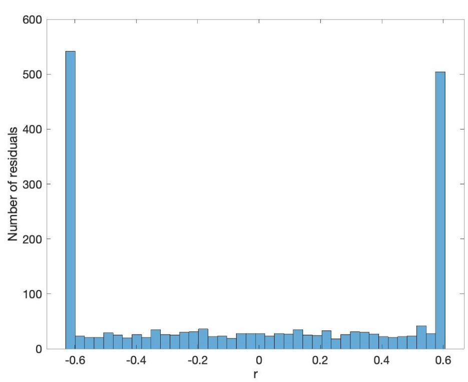
In **Task c** the norm minimisation values are given in *Table 1*. Besides the value itself and the running time, the residuals are another way of observing what the difference between different norm minimisation is.



(a) l_1 minimisation residuals



(b) l_2 minimisation residuals



(c) l_∞ minimisation residuals

Figure 1

The figures above give an insight into the behaviour of these norms. Looking at *Figure 1a* it is observed that l_1 norm minimisation puts a lot more weight on the small residuals while almost not taking into account the large residuals. In terms of regression, this could be interpreted as the outliers are not taken into account at all. On the other hand, by looking at *Figure 1b* it is observed that the l_2 norm minimisation fits a Gaussian centred at residual 0 showing that large residuals will also be taken into account but only by a small margin. In terms of regression, this could be interpreted as outliers are also taken into account. Lastly, looking at *Figure 1c* it is observed that the weights are put only on the large residuals. In terms of regression, this could be interpreted that only outliers are taken into account.

2 Problem 2

The second problem consists of looking at a different method of minimising the l_1 norm. The method studied will be a *First Order Interior Point Method with a Backtracking Line Search.* Firstly it is looked at how the problem can be transformed from an inequality constrained problem to an unconstrained problem. Then it is looked at developing an algorithm based on *Gradient Descent* to solve the problem. Lastly, the convergence of the algorithm is studied.

2.1 Task a

A typical optimisation problem is considered:

$$\begin{aligned} &\min_{x} f_0(x) \\ &\text{subject to} \quad f_i(x) \leq 0, \quad i = 1, \cdots, m \end{aligned} \tag{13}$$

where $f_0(x)$ and $f_i(x)$ are convex and twice differentiable functions for $i = 1, \cdots, m$. For this problem the objective will be to find the l_1-norm minimisation by performing gradient descent .Therefore the problem above can be expressed as in **Task b** of **Problem 1**:

$$\begin{aligned} &\min_{\tilde{\mathbf{x}}} \quad \tilde{\mathbf{c}}^T \tilde{\mathbf{x}} \\ &\text{subject to} \quad \tilde{\mathbf{A}}\tilde{\mathbf{x}} \preceq \tilde{\mathbf{b}} \end{aligned} \tag{14}$$

It is desired to move from inequality constrained problem to an equality constrained problem. This can be achieved by introducing a logarithmic barrier $\phi(x)$ which should have the following property:

$$\phi(x) = \begin{cases} 0 & u \leq 0 \\ \infty & u \geq 0 \end{cases} \tag{15}$$

Which mathematically can be approximated as as:

$$\phi(x) = -\frac{1}{t} \sum_{i=1}^{m} \log(-f_i(x)) \tag{16}$$

The function above is called a logarithmic barrier. For a single inequality the shape of the barrier can be seen below in *Figure 2*:

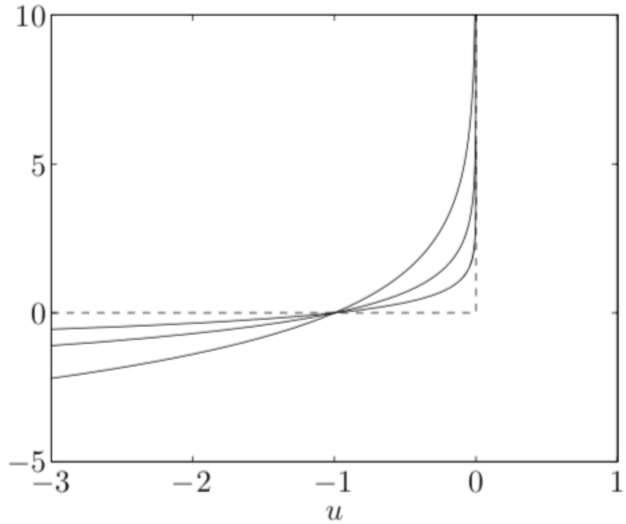


Figure 2: The dashed line shows the ideal barrier while the solid curves show $\phi(u) = -\frac{1}{t}\log(-u)$ for $t = 0.5, 1, 2$ [1, Chapter 11]

As observed the function is convex and differentiable. As t is increased, the approximation of the real barrier becomes more accurate. For the LP inequality problem defined in *Equation 14* above the logarithmic barrier is equal to:

$$\phi(\tilde{\mathbf{x}}) = -\frac{1}{t}\sum_{i=1}^{m+n}\log(\tilde{b}_i - \tilde{\mathbf{a}}_i^T\tilde{\mathbf{x}}) \tag{17}$$

where $\tilde{\mathbf{a}}_1^T, \cdots, \tilde{\mathbf{a}}_{m+n}^T$ are the rows of $\tilde{\mathbf{A}}$.Therefore a new objective for the current problem can be defined:

$$\begin{aligned} \min_{\tilde{\mathbf{x}}} \quad & \tilde{\mathbf{c}}^T\tilde{\mathbf{x}} + \phi(\tilde{\mathbf{x}}) \\ \text{subject to} \quad & \tilde{\mathbf{A}}\tilde{\mathbf{x}} = \tilde{\mathbf{b}} \end{aligned} \tag{18}$$

For $t > 0$ we define $\tilde{\mathbf{x}}^*(t)$ as the solution of this minimisation. The *central path* corresponds to the set of points $\tilde{\mathbf{x}}^*(t)$. Points on the central path are characterised by the fact that $\tilde{\mathbf{x}}^*(t)$ is strictly feasible and there exists $\mathbf{v} \in \mathcal{R}^p$ [1, Chapter 11] such that :

$$0 = t\nabla(\tilde{\mathbf{c}}^T\tilde{\mathbf{x}}^*(t)) + \nabla\phi(\tilde{\mathbf{x}}^*(t)) + \tilde{\mathbf{A}}^T\mathbf{v} \tag{19}$$

Therefore the central path for the l_1-norm minimisation can be defined as:

$$\text{central path} \quad f(\tilde{\mathbf{x}}) = \tilde{\mathbf{c}}^T\tilde{\mathbf{x}} - \sum_{i=1}^{m+1}\log(\tilde{b}_i - \tilde{\mathbf{a}}_i^T\tilde{\mathbf{x}}) \tag{20}$$

It is observed that this problem is convex and therefore differentiable. In order to build the minimisation algorithm the gradient cost needs to be calculated. It has the following expression:

$$\begin{aligned} &\text{gradient cost} \quad \nabla f(\tilde{\mathbf{x}}) = t\tilde{\mathbf{c}} + \tilde{\mathbf{A}}^T\mathbf{d} \\ &\text{where} \quad \mathbf{d} = \frac{1}{\tilde{\mathbf{b}} - \tilde{\mathbf{A}}\tilde{\mathbf{x}}} \end{aligned} \tag{21}$$

2.2 Task b

For the purpose of this problem a *Gradient Descent* method with a *Backtracking Line Search* is chosen as the algorithm to find the minimisation of l_1-norm. *Gradient Descent* represents a first order optimisation algorithm which uses the gradient of the of function as a step direction towards the minimum solution. It alternates between two steps: determining a descent direction $\Delta x = -\nabla f(x)$ and the selection of step size s [1, Chapter 9]. To choose the value of s a *Backtracking Line Search* method is used which has two parameters α, β with $0 \leq \alpha \leq 0.5$, $0 \leq \beta \leq 1$. The algorithm starts with a unit size step s and then reduces it by a factor β until the stopping condition $f(x+\Delta x) \leq f(x) + \alpha s \nabla f(x) \Delta x$ is satisfied. A visual representation of the *Backtracking Line Search* can be seen below:

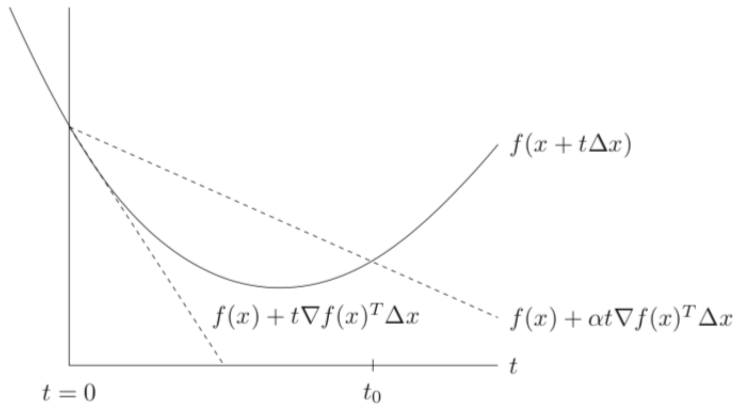


Figure 3: The curved line shows the restricted line over which the backtracking line search is done while the dashed lines represents the limits over which the search is done. The step t defined here corresponds to step s defined above [1, Chapter 9]

By using the equations defined in **Task a** the *Gradient Descent* algorithm with *Backtracking Line search* has the following steps:

Algorithm 1 Gradient Descent with Backtracking Line search for l_1-norm minimisation

$\tilde{\mathbf{x}} \leftarrow \tilde{\mathbf{x}}_0$
$\alpha \leftarrow 0.2$
$\beta \leftarrow 0.5$
$\epsilon \leftarrow 10^{-3}$
while $||\nabla f(\tilde{\mathbf{x}})||_2^2 > \epsilon$ **do**
  $s \leftarrow 1$
  while $f(\tilde{\mathbf{x}} - s\nabla f(\tilde{\mathbf{x}})) > f(\tilde{\mathbf{x}}) - \alpha s \nabla f(\tilde{\mathbf{x}})^T \nabla f(\tilde{\mathbf{x}})$ **do**
    $s \leftarrow \beta s$
  end while
  $\tilde{\mathbf{x}} \leftarrow \tilde{\mathbf{x}} - s\nabla f(\tilde{\mathbf{x}})$
end while

With *Algorithm 1* defined above, there two issues that are needed to be addressed. One is regarding the initial point $\tilde{\mathbf{x}}_0$ which needs to be a feasible point, i.e. $\tilde{\mathbf{A}}\tilde{\mathbf{x}}_0 \preceq \tilde{\mathbf{b}}$. To find this a point a phase 1 implementation has to be considered [1, Chapter 11]. This phase 1 has only one objective, to find a feasible solution of this inequality. The problem can be formed as LP problem which can be solved using the simplex method. This problem minimises the number of infeasible solutions.

$$\begin{aligned} \text{minimise} \quad & \mathbf{1}^T \mathbf{z} \\ \text{subject to} \quad & \tilde{\mathbf{A}}\tilde{\mathbf{x}} \preceq \tilde{\mathbf{b}} + \mathbf{z} \end{aligned} \tag{22}$$

The other issue that needs to be addressed is the fact that the increment $\tilde{\mathbf{x}}_{k+1}$ from $\tilde{\mathbf{x}}_k$ needs to stay in the feasible region. To ensure that $\tilde{\mathbf{x}}_{k+1}$ is feasible the following conditions is introduced[1, Chapter 11]:

while $\min(\tilde{\mathbf{b}} - \tilde{\mathbf{A}}(\tilde{\mathbf{x}} - s\nabla f(\tilde{\mathbf{x}})) < 0$ **do**
  $s \leftarrow \beta s$
end while

By applying *Algorithm 1* and ensuring that the two problems are avoided the l_1 minimisation algorithm results in the value of the central path equal to $f(\tilde{\mathbf{x}}^*) = 335.1928$, the gradient cost equal to $\nabla f(\tilde{\mathbf{x}}^*) = 7.98 \times 10^{-4}$ and the number of iterations required equal to $k_{total} = 2124$. The full algorithm can be seen in the *Appendix Figure 11* .

2.3 Task c

The convergence analysis of the of *Gradient Descent* with *Backtracking Line Search* is done by subtracting $f(\tilde{\mathbf{x}}^*)$ from all the values of $f(\tilde{\mathbf{x}}_k)$ calculated during the algorithm implementation. The results are displayed in the figure below:

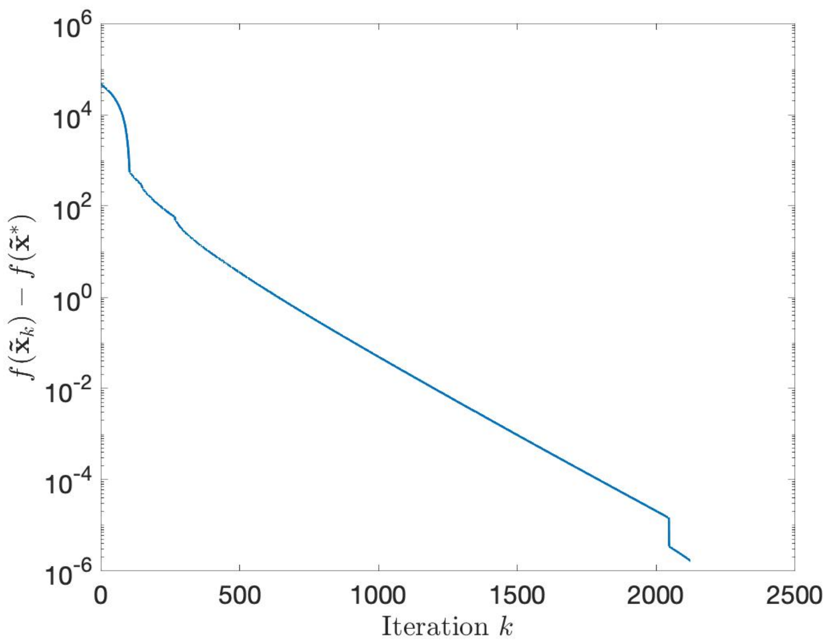


Figure 4: Convergence analysis for the minimisation of l_1-norm with Gradient Descent method with a Backtracking Line Search in which $f(\tilde{\mathbf{x}}^*) = 335.1928$

As observed from *Figure 4* above the convergence is fairly linear. This suggests that the backtracking is producing the expected results.

The true value of the minimisation $\tilde{\mathbf{c}}^T\tilde{\mathbf{x}} = 10641$ which is very far away from the l_1-norm minimisation found by the dual-simplex method in **Problem 1** which had $\tilde{\mathbf{c}}^T\tilde{\mathbf{x}} = 143.26$. This is because of the fact that the logarithmic barrier introduced is not ideal. If t is increased to $t = 8$ the new minimisation value of l_1 is $\tilde{\mathbf{c}}^T\tilde{\mathbf{x}} = 242$ which is closer to the true minimum. However by increasing the value t number of iterations increases considerably.

3 Problem 3

The third problem is concerning *Compressed Sensing* which is a problem in signal processing concerning sampling sparse signal at a lower rate than *Nyquist Frequency*. Firstly, the meaning of the problem l_1-*Regularised Least Squares* is looked at and understood. Then, a *Second Order Interior Point Method with Backtracking Line Search* is developed to solve these types of problems. Lastly, it is looked at applying this algorithm to recover a sparse signal and comparing the method of l_1-*Regularised Least Squares* with other methods which could be used.

3.1 Task a

In **Problem 3** new optimisation problem is introduced: l_1-*regularised least squares (LS)*. This problem can be expressed in the following way:

$$\min_{x} \quad ||\mathbf{Ax} - \mathbf{b}||_2^2 + \lambda||\mathbf{x}||_1 \tag{23}$$

where $\lambda > 0$ is a regularisation parameter,$\mathbf{b} \in \mathcal{R}^m$ and $\mathbf{x} \in \mathcal{R}^n$ and $\mathbf{A} \in \mathcal{R}^{m \times n}$. There are some important things to note in this problem. The best way to understand these things is compare this regularisation method to the *Tikhonov Regularisation* described in **Problem 1** [2].

- *Solution:*The l_1-*Regularised LS* always has a solution but it might not been unique. This is different compared to the *Tikhonov Regularisation* which has clear analytical solution.
- *Non-Linearity:*It can be observed that the solution of the *Tikhonov Regularisation* is a linear function of $\mathbf{b}$ which by contrast the l_1-*Regularised LS* yields a solution which is non-linear in $\mathbf{b}$.
- *Convergence*: In *Tikhonov Regularisation* the optimal solution tends to zero as $\lambda \to \infty$, while for l_1-*Regularised LS* the convergence occurs for $\lambda = ||2\mathbf{A}^T\mathbf{b}||_\infty$
- *Sparsity* the solution is observed by looking at the residuals of the norm minimisation for l_1 and l_2. As studied in **Problem 1**, it is expected that l_1-*Regularised LS* will have a large number of zeros while the *Tikhonov Regularisation* solution will have all of coefficients non-zero

To be able to perform an optimisation on this problem which does not require a high complexity it is desired to have a differentiable objective. Therefore it is required to transform the problem into a quadratic convex problem with linear inequality constraints. This is done by introducing a vector $\mathbf{u}$ which transforms the problem into a *LP* problem:

$$\begin{aligned} \min_{x} \quad & ||\mathbf{Ax} - \mathbf{b}|| + \lambda\mathbf{1}^T\mathbf{u} \\ \text{subject to} \quad & |\mathbf{x}| \preceq \mathbf{u} \end{aligned} \tag{24}$$

As observed the equations above corresponds to a quadratic convex problem which can be solved by an interior point method. Firstly, it is required to define a logarithmic barrier which will remove the inequality constrained:

$$\begin{aligned} \Phi(\mathbf{x}, \mathbf{u}) &= \underbrace{-\sum_{i=1}^{n} \log(u_i + x_i)}_{\mathbf{x} \succeq \mathbf{u}} - \underbrace{-\sum_{i=1}^{n} \log(u_i - x_i)}_{\mathbf{x} \preceq \mathbf{u}} \\ &= -\sum_{i=1}^{n} \log(u_i^2 - x_i^2) \\ &= -\mathbf{1}^T \log(\mathbf{u}^2 - \mathbf{x}^2) \end{aligned} \tag{25}$$

Using this barrier the central path of the problem can be defined as:

$$\phi_t(\mathbf{x}, \mathbf{u}) = t||\mathbf{Ax} - \mathbf{b}||_2^2 + t\lambda\mathbf{1}^T\mathbf{u} - \mathbf{1}^T \log(\mathbf{u}^2 - \mathbf{x}^2) \tag{26}$$

3.2 Task b

Now that the central path ϕ_t has been defined, it is required to find the gradient and the Hessian of ϕ_t. This is required to define a step increment in the direction that would minimise ϕ_t. The gradient $\mathbf{g}$ can be expressed as:

$$\mathbf{g}(\mathbf{x},\mathbf{u}) = \begin{pmatrix} \nabla_{\mathbf{x}}\phi_t(\mathbf{x},\mathbf{u}) \\ \nabla_{\mathbf{u}}\phi_t(\mathbf{x},\mathbf{u}) \end{pmatrix} = \begin{pmatrix} 2t\mathbf{A}^T\mathbf{A}\mathbf{x} - 2t\mathbf{A}^T\mathbf{b} + \frac{2\mathbf{x}}{(\mathbf{u}^2-\mathbf{x}^2)} \\ t\lambda\mathbf{1} - \frac{2\mathbf{u}}{(\mathbf{u}^2-\mathbf{x}^2)} \end{pmatrix} \tag{27}$$

Respectively, the Hessian $\mathbf{H}$ can be expressed as:

$$\begin{aligned} \mathbf{H}(\mathbf{x},\mathbf{u}) &= \begin{pmatrix} \nabla_{\mathbf{x}\mathbf{x}}\phi_t(\mathbf{x},\mathbf{u}) & \nabla_{\mathbf{x}\mathbf{u}}\phi_t(\mathbf{x},\mathbf{u}) \\ \nabla_{\mathbf{u}\mathbf{x}}\phi_t(\mathbf{x},\mathbf{u}) & \nabla_{\mathbf{u}\mathbf{u}}\phi_t(\mathbf{x},\mathbf{u}) \end{pmatrix} = \begin{pmatrix} 2t\mathbf{A}^T\mathbf{A} + \mathbf{D}_1 & \mathbf{D}_2 \\ \mathbf{D}_2 & \mathbf{D}_1 \end{pmatrix} \\ \mathbf{D}_1 &= \mathrm{diag}(\frac{2(\mathbf{u}^2+\mathbf{x}^2)}{(\mathbf{u}^2-\mathbf{x}^2)^2}) \\ \mathbf{D}_2 &= \mathrm{diag}(\frac{-4\mathbf{u}\mathbf{x}}{(\mathbf{u}^2-\mathbf{x}^2)^2}) \end{aligned} \tag{28}$$

There is more information which is needed to be addressed. As observed in **Problem 2** it was required to increase variable t in order to make the barrier ideal. In this problem the barrier plays an important factor as it determines the sparsity of the solution. As t increases the barrier will impose more restrictions on the central path such that the required solution has to become sparse. In order to progressively adjust for the sparsity an update step is introduced which increases t if the step size s is greater than minimum size step s_{min}, in which case :

$$\begin{aligned} &t = \mu * t \\ &\text{where} \quad \mu \approx 2 \end{aligned} \tag{29}$$

By using all the information above $\phi_t(\mathbf{x}\mathbf{u}), \mathbf{g}(\mathbf{x},\mathbf{u})$ and $\mathbf{H}(\mathbf{x},\mathbf{u})$ a *Newton Interior-Point* method with *Backtracking Line Search* algorithm can be implemented to find $\mathbf{x}^*$ which minimises the central path. The structure of the algorithm is summarised below with the following parameters:

Algorithm 2 Newton Method with Backtracking Line search for l_1-regularised Least Squares minimisation

---

$\mathbf{x} \leftarrow \mathbf{0}$
$\lambda_{max} \leftarrow ||2\mathbf{A}^T\mathbf{b}||_\infty$
$\lambda \leftarrow 0.01\lambda_{max}$
$t \leftarrow 1$
$\alpha \leftarrow 0.01$
$\beta \leftarrow 0.5$
$\mu \leftarrow 2$
$s_{min} \leftarrow 0.5$
$\epsilon \leftarrow 10^{-5}$
$MAXITER \leftarrow 5$

---

while $\sqrt{\mathbf{g}(\mathbf{x},\mathbf{u})^T\mathbf{H}(\mathbf{x},\mathbf{u})^{-1}\mathbf{g}(\mathbf{x},\mathbf{u})} > \epsilon$ **do**
  $s \leftarrow 1$
  $\Delta = -\mathbf{H}(\mathbf{x},\mathbf{u})^{-1}\mathbf{g}(\mathbf{x},\mathbf{u})$
  $\Delta_x = \Delta(1:n)$
  $\Delta_u = \Delta(n+1:end)$
  for $ITER \leftarrow 1 : MAXITER$ **do**
    $\mathbf{x}_{new} \leftarrow \mathbf{x} + s\Delta_x$
    $\mathbf{u}_{new} \leftarrow \mathbf{u} + s\Delta_u$
    if $\max([\mathbf{x}_{new} - \mathbf{u}_{new}; -\mathbf{x}_{new} - \mathbf{u}_{new}]) < 0$ **then**
      if $\phi_t(\mathbf{x}_{new},\mathbf{u}_{new}) - \phi_t(\mathbf{x},\mathbf{u}) \leq \alpha s \mathbf{g}(\mathbf{x},\mathbf{u})^T\Delta$ **then**
        break
      end if
    end if
    $s \leftarrow \beta s$
  end for
  $\mathbf{x} \leftarrow \mathbf{x}_{new}$
  $\mathbf{u} \leftarrow \mathbf{u}_{new}$
  if $s \geq s_{min}$ **then**
    $t \leftarrow \mu t$
  end if
end while

3.3 Task c

It is time now to define the problem. Compressed sensing is a technique that involves sampling a sparse signal at a lower frequency than what *Nyquist Sampling Theorem* states. The algorithm involves recovering the original signal from the sampled signal. This process is visualised in the figure below:

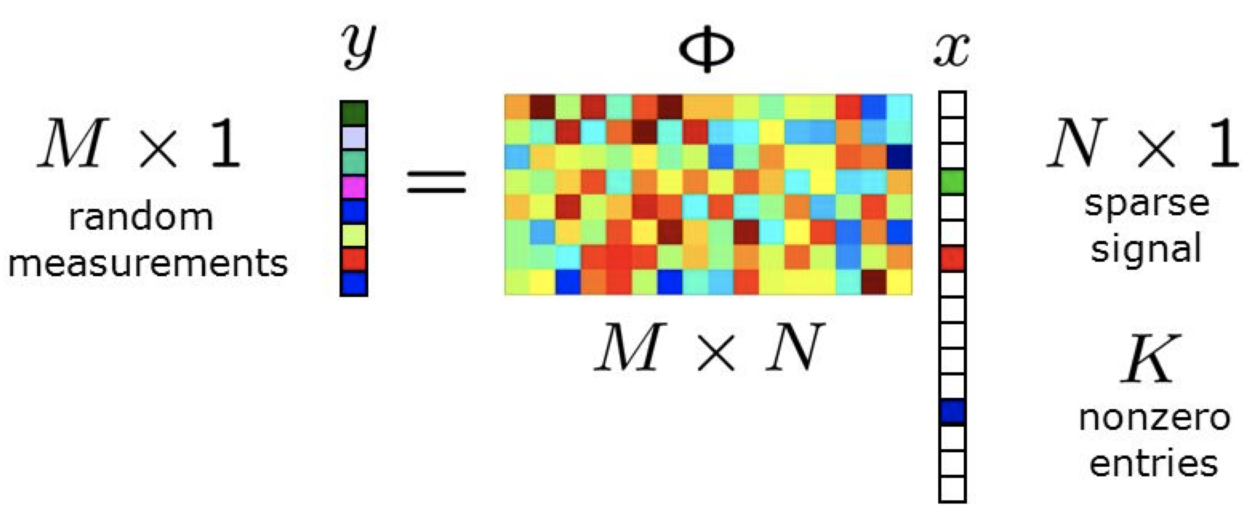


Figure 5: Compressed sensing theory with $\mathbf{x}$ being the sparse signal and $\mathbf{y}$ the sampled signal [3]

For the current problem that is investigated the original sparse signal is defined as $\mathbf{x}_0$ where $\mathbf{x}_0 \in \mathcal{R}^{256}$. The signal $\mathbf{x}_0$ consists of 10 spikes of amplitude ± 1. The signal can be observed in *Figure 6* below:

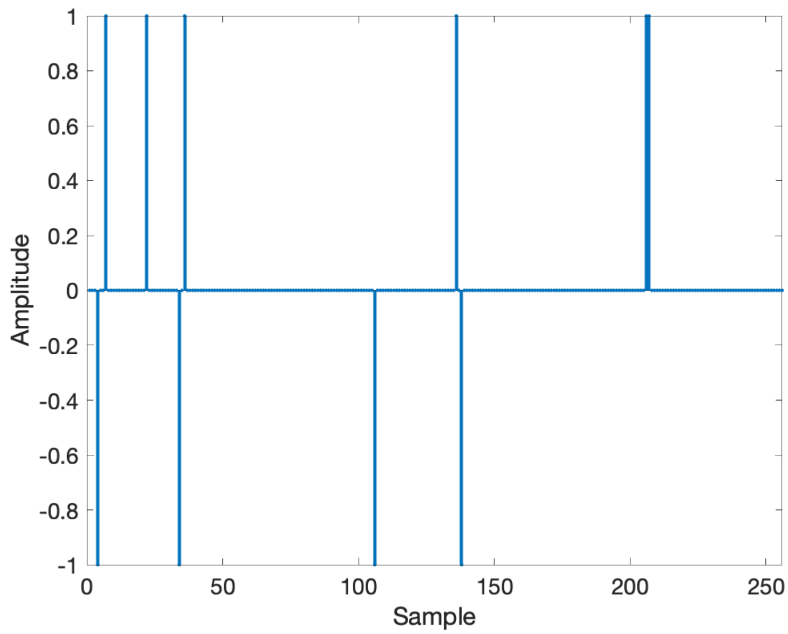


Figure 6: Original sparse signal $\mathbf{x}_0$

A measurement matrix is used to sample the sparse signal. For the problem that is looked at the matrix is $\mathbf{A}$, where $\mathbf{A} \in \mathcal{R}^{60\times 256}$. Using this matrix the sampled vector $\mathbf{b}$, where $\mathbf{b} \in \mathcal{R}^{60}$, can be found by $\mathbf{b} = \mathbf{A}\mathbf{x}_0$. The sampled vector is displayed in the figure below:

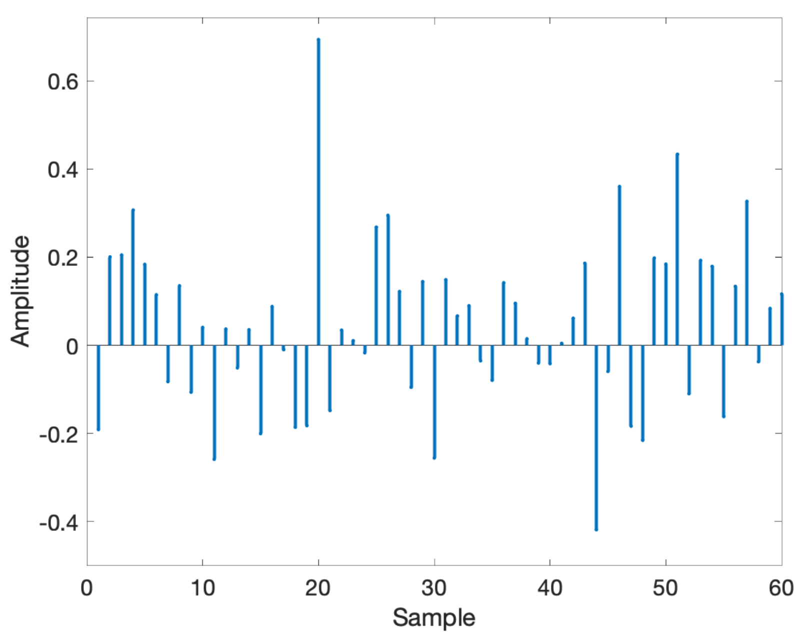


Figure 7: Compressed signal $\mathbf{b}$

The challenge of the problem lies in recovering the original signal $\mathbf{x}_0$ from $\mathbf{b}$. The sparsity of the problem can be taken advantage of and therefore the problem can be redefined as in **Task b**. As discussed and demonstrated in the previous task this problem can be solved with a *Newton Method* with a *Backtracking Line Search*. Therefore applying *Algorithm 2* to the current values of $\mathbf{b}$ and $\mathbf{A}$ results in the fact that the central math is minimised and an optimum value of $\mathbf{x}$ can be found as $\mathbf{x}^*$. This optimal vector $\mathbf{x}^*$ is displayed in the figure below:

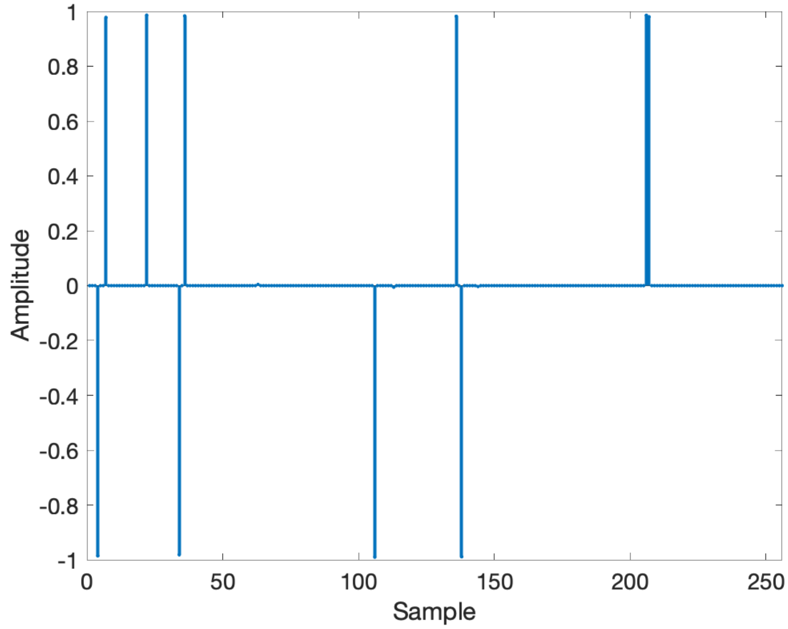


Figure 8: Recovered Signal $\mathbf{x}^*$

As observed the value obtained by applying the optimisation algorithm completely matches the original signal $\mathbf{x}_0$. It is observed that the amplitudes of the spikes are not exactly ± 1 but rather very close to them. As discussed before this is no surprise as the algorithm designed in **Task b** is made so that the recovered signal will be sparse. This is due to introducing the l_1-norm which forces most of the terms of $\mathbf{x}^*$ to be zero.

3.4 Task d

In this task, other methods of reconstructing the original signal are investigated. Firstly it is looked at the simple minimum energy reconstruction which means the that recovered signal is being found by the least squares solution:

$$\mathbf{x}^*_{\mathbf{LS}} = (\mathbf{A}^T\mathbf{A})^{-1}\mathbf{A}^T\mathbf{b}$$

The reconstructed signal $\mathbf{x}^*_{LS}$of is displayed in *Figure 9* below:

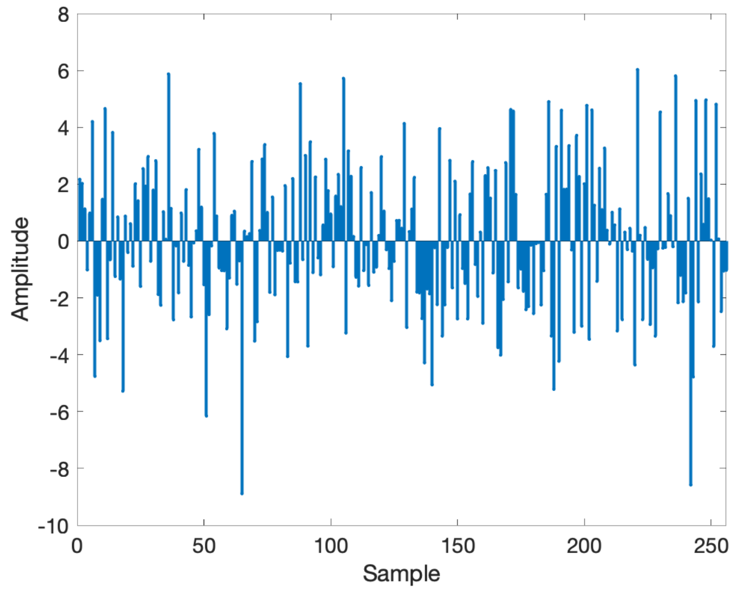


Figure 9: Minimum energy recovered signal $\mathbf{x}^*_{LS}$

As expected the recovered signal $\mathbf{x}^*_{LS}$ does not match the original signal $\mathbf{x}_0$. It is also observed that the amplitudes are very large. This is because the values far away from away from the origin are not penalised. Therefore one might think that *Tikhonov Regularisation* will be good idea as it would penalise those values. The solution of the *Tikhonov Regularisation* is expressed as:

$$\mathbf{x}^*_{TIK} = (\mathbf{A}^T\mathbf{A} + \lambda\mathbf{I})^{-1}\mathbf{A}^T\mathbf{b}$$

The signal $\mathbf{x}^*_{TIK}$ can be visualised in the figure below.

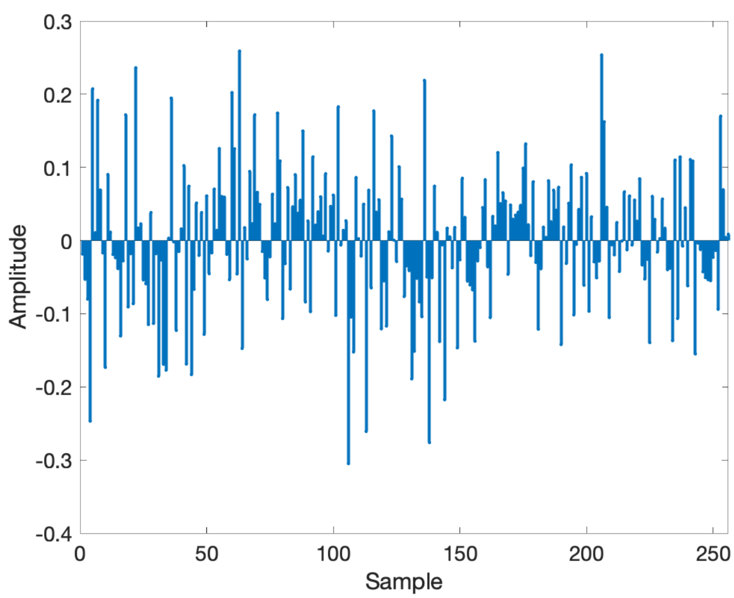


Figure 10: Recovered signal using Tikhonov regularisation $\mathbf{x}^*_{TIK}$

It is observed that the amplitudes are reduced. However, there is still a core problem, one of the sparsity of the solution which cannot be solved. This because minimising the least-squares without any l_1-norm barrier will mean that every element in $\mathbf{x}^*_{LS}$ or $\mathbf{x}^*_{TIK}$ will be non-zero. Therefore it is observed how essential a l_1-norm barrier is because it will ensure that the recovered signal will be sparse.

4 APPENDIX

```
beta=0.5;
alpha=0.01;
t=1;
nr=0;
x=x0;
function_values=[]
while norm(gradient(x,f,t,b,A))>0.0013
 s=1;
 nr = nr + 1;
 while min(b-A*(x-s*gradient(x,f,t,b,A))) < 0
     s = beta*s;
 end
 while objective_function(x-s*gradient(x,f,t,b,A),f,t,b,A)>objective_function(x,f,t,b,A)-alpha*s*norm(gradient(x,f,t,b,A))
   s = beta * s;
 end

  x = x - s*gradient(x,f,t,b,A);
  function_values=[function_values;objective_function(x,f,t,b,A)];

end
```

Figure 11: l_1-norm minimisation algorithm MATLAB implementation

```
%Hyperaramets Initialisation
alpha = 0.01;
beta = 0.5;
epsilon = 10^-4;
x_1 = zeros(size(x));
maxiter = 5;
smin=0.5;
mu = 2;

while sqrt(g(x_1,u)' * (H(x_1,u)\g(x_1,u))) > epsilon
    delta   = -H(x_1,u)\g(x_1,u);
    delta_x = delta(1:size(x_1));
    delta_u = delta(size(x_1)+1:end);
    s=1;
    for iter=1:maxiter
       x_new = x_1+s*delta_x;
       u_new = u+s*delta_u;
       f = [x_new-u_new;-x_new-u_new];
       if (max(f) < 0)
           if (F(x_new,u_new)-F(x_1,u) <= alpha*s*g(x_1,u)'*delta)
               break;
           end
       end
       s = beta*s;
    end
    x_1 = x_new;
    u = u_new;

    if s >= smin
        t=mu*t;
        setGlobalt(t);
    end
```

Figure 12: Compressed Sensing algorithm MATLAB implementation

References

[1] Boyd Stephen P. and Lieven. Vandenberghe. *Convex Optimization.* Cambridge University Press, 2012.

[2] M. Lustig S. Boyd S.-J. Kim, K. Koh and D. Gorinevsky. An interior-point method for large-scale l1-regularized least squares. *IEEE Journal on Selected Topics in Signal Processing*, 1(4):06–617, Dec 2007.

[3] Slideplayer. Compressive sensing a new approach to image acquisition and processing.