

Section 1 : Derivatives Used for the Gradient Descent Approach

1 Overview

The objective is to model transcription rate of a target gene at a particular promoter as controlled by an enhancer region of DNA. The model is based on those in Reinitz et al. [2003] and Janssens et al. [2006]. The inputs to the model are the levels of transcription factors (TFs), divided into activators and quenchers, the DNA sequence of the enhancer region, and PWMs describing the DNA binding of the TFs. Each factor, a contributes two free parameters to the model: its maximum binding affinity, K_a , and its activating or quenching efficiency, C_a or E_a , respectively. The maximum transcription rate, R_0 , and the maximum energy barrier, G_0 , are also free parameters. We will refer to the free parameters of the model collectively as Θ .

The input data, D , contains a set of n pairs, (\mathbf{V}, v_t) , where each pair contains a vector, $\mathbf{V} = (v_1, v_2, \dots, v_n)$ listing the concentrations of all transcription factors, and v_t , the concentration of the target gene. The data contains a set of TF sites, $S^a = \{s_i = (a, j, s) | i = 1, \dots, n_a\}$. TF a has n_a sites, and each one specifies the TF, its position in the enhancer region, and its PWM score. The input data is thus $D = \{(\mathbf{V}, v_t)\} \cup S^a$.

The model predicts the rate of mRNA transcription at the promoter. It is based on a function, M , that predicts the “effective” number of activator molecules bound to the enhancer region. The rate of transcription, R , is assumed to be controlled by the amount an energy barrier, G_0 , is reduced by binding of M activator molecules. The scale factor by which each bound molecule reduces the barrier, G , is treated as a constant. Since mRNA degradation is assumed to be fast relative to transcription, mRNA level is assumed to be equivalent to mRNA rate for the purposes of model validation.

2 Objective Function

2.1 Function B

We minimize the mean squared error of the predicted transcription rate plus a penalty function, P , designed to enforce the constraint that the bound activator molecules should never reduce the energy barrier, G_0 , below zero. The minimization is with respect to Θ , the free parameters of the model. The objective function computes the mean of the the squared error in the predicted transcription rate, R , over the set of n observed points, $\{(\mathbf{V}, v_t)\}$. The objective function is

$$B(D, \Theta) = \frac{1}{n} \sum_{\{(\mathbf{V}, v_t)\}} \left(\frac{1}{2} (v_t - R(\mathbf{V}, S, \Theta))^2 \right). \quad (1)$$

Fixed Parameters	
A	:= Set of all activators
B	:= Set of all quenchers
$D = (\mathbf{V}, v_t)$:= set of measurements of TF levels, \mathbf{V} , and target gene level, v_t
$\mathbf{V} = (v_1, v_2, \dots, v_n)$:= one measurement of TF levels
$S^a = \{s_i = (a, j, s) i = 1, \dots, n_a\}$:= Set of all sites for TF a , where j is the position in the enhancer region, s is the PWM score of the site n_a is the number of sites for TF a
\hat{S}_a	:= Maximum possible PWM score for TF a
$S^A = \bigcup_{a \in A} S^a$:= Set of all sites for activators
$S^B = \bigcup_{b \in B} S^b$:= Set of all sites for quenchers
$S = S^A \cup S^B$:= Set of all sites for all TFs
$N^B = \sum_{b \in B} n_b$:= Total number of quencher sites
G	:= A constant (1.0), called Q in Janssens et al. [2006]
f^{AF}	:= A constant (0.99); Hill function in Reinitz et al. [2003]
Free Parameters	
$K^A = \{K_a a \in A\}$:= Activator maximal binding affinities
$K^B = \{K_b b \in B\}$:= Quencher maximal binding affinities
$K = K^A \cup K^B$:= Set of all maximal binding affinities
$C^A = \{C_a a \in A\}$:= Activator effectiveness parameters
$E^B = \{E_b b \in B\}$:= Quencher effectiveness parameters
R_0	:= Maximal transcription rate
G_0	:= Maximal energy barrier to transcription
Functions	
$B(D, \Theta)$:= Objective function to minimize.
$R(\mathbf{V}, S, \Theta)$:= Transcription rate function.
$P(\mathbf{V}, S, \Theta)$:= Penalty function.
$M(\mathbf{V}, S, \Theta)$:= Number of bound activators function.
$N(\mathbf{V}, S, \Theta)$:= Recruitment function.
$F(a, i, v_a)$:= Quenched occupancy function.
$Q^{(m,n)}(i)$:= Partial quenching function for position i .
$Q^{(1,N^B)}(i)$:= Quenching function for position i .
$d(i, j)$:= Distance-based quenching function
$f(a, i, v_a)$:= Occupancy function for TF a at position i .
$K(a, i)$:= Affinity of TF a at the position i
$T(x)$:= Transformation of $x \in \Theta$ in the unconstrained space
$T'(\hat{x})$:= Transformation of $\hat{x} \in \hat{\Theta}$ back to the normal space

Table 1: **Overview of the terminology used in the main document and here.**

2.2 Derivatives of B

$$\frac{\partial B}{\partial x} = \begin{cases} \frac{1}{n} \sum_{\{\mathbf{V}, v_t\}} ((v_t - R(\mathbf{V}, S, \Theta))^2 \frac{\partial R}{\partial x}) & \text{if } x = R_0, \\ \frac{1}{n} \sum_{\{\mathbf{V}, v_t\}} ((v_t - R(\mathbf{V}, S, \Theta))^2 \frac{\partial R}{\partial x} + \frac{\partial F}{\partial x}). & \text{if } x \in \Theta \cap R_0 \end{cases} \quad (2)$$

3 Transcription rate, R

3.1 Function R^c

The transcription rate model is originally defined by the function R^c , which is “capped” at R_0 ,

$$R^c(\mathbf{V}, S, \Theta) = \begin{cases} R_0 \exp(-(G_0 - GM(\mathbf{V}, S, \Theta))) & \text{if } GM < G_0 \\ = R_0 \exp(GM(\mathbf{V}, S, \Theta) - G_0), & \\ R_0 & \text{otherwise} \end{cases} \quad (3)$$

This is the R function used in the benchmark method, SA and GA.

3.2 Function R^s

For the gradient descent method, however, we use the sigmoid function to prevent R from adopting values larger than R_0 , called R^s ,

$$\begin{aligned} R^s(\mathbf{V}, S, \Theta) &= R_0 \operatorname{sigm}(-(G_0 - GM(\mathbf{V}, S, \Theta)) \cdot s_1 + s_2) \\ &= R_0 \frac{1}{1 + \exp((G_0 - GM(\mathbf{V}, S, \Theta)) \cdot s_1 - s_2)}, \end{aligned} \quad (4)$$

where $s_1 = 2.5$ and $s_2 = 2$ are scaling function to resemble R^c as closely as possible (see Fig.1).

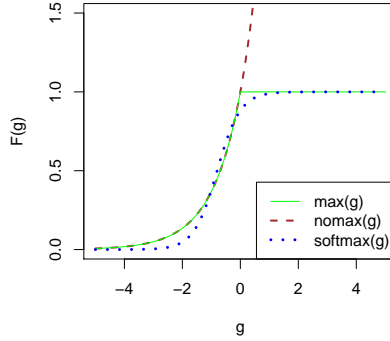


Figure 1: Comparison between original R and transformed R

The function M , which is described in the Section 4, does not depend G_0 or R_0 .

3.3 Derivatives of R^s

We need the derivatives of R^s with respect to each of the free parameters, $K \cup C^A \cup E^B$, G_0 and R_0 . The function M does not depend on the last two parameters, so there are three cases:

$$\frac{\partial R^s}{\partial x} = \begin{cases} \left(\begin{array}{l} (-\frac{\partial G_0}{\partial x}) \cdot s_1 R_0 \cdot \text{sigm}((GM(\mathbf{V}, S, \Theta) - G_0) \cdot s_1 + s_2) \cdot \\ (1 - \text{sigm}((GM(\mathbf{V}, S, \Theta) - G_0) \cdot s_1 + s_2)) \end{array} \right. & \text{if } x = G_0, \\ \frac{\partial R_0}{\partial x} \text{sigm}(-(G_0 - GM(\mathbf{V}, S, \Theta)) \cdot s_1 + s_2) & \text{if } x = R_0, \\ \left. \begin{array}{l} R_0 \cdot \text{sigm}((GM(\mathbf{V}, S, \Theta) - G_0) \cdot s_1 + s_2) \cdot \\ (1 - \text{sigm}((GM(\mathbf{V}, S, \Theta) - G_0)) \cdot s_1 + s_2) \cdot s_1 G \cdot \frac{\partial x}{\partial x} \end{array} \right) & \text{if } x \in K \cup C^A \cup E^B. \end{cases} \quad (5)$$

4 Effective number of bound activators, M

4.1 Function M

The function M describes the predicted, effective number of bound activators. It is defined in terms of the constant, f^{AF} , and function, N (Section 5), the number of sites that are available to recruit adapters.

$$M(\mathbf{X}, S, \Theta) = f^{AF} N(\mathbf{X}, S, \Theta). \quad (6)$$

4.2 Derivatives of M

Like M , the function N does not depend on free parameters R_0 or G_0 . So the non-zero partial derivatives are

$$\frac{\partial M}{\partial x} = f^{AF} \frac{\partial N}{\partial x}, \text{ for } x \in K \cup C^A \cup E^B. \quad (7)$$

5 Recruitment function, N

5.0.1 Function N

The amount of activators recruited by activator sites is modeled by function N . For each activator, $a \in A$, it sums the occupancy of all activator sites, after correcting for quenching, $F^A(a, i, v_a)$ (Section 6), and weights them by a 's effectiveness parameter, C_a . The recruitment function is

$$N(\mathbf{V}, S, \Theta) = \sum_{s \in S^A} C_a F(a, i, v_a), \quad (8)$$

where, in each term in the sum, a and i are determined from the site current site, $s = (a, i)$.

5.1 Derivatives of N

There are two non-zero cases,

$$\frac{\partial N}{\partial x} = \begin{cases} \sum_{i=1}^L F(a, i, v_a), & \text{if } x = C_a, \\ \sum_{s \in S^A} C_a \frac{\partial F(a, i, v_a)}{\partial x}, & \text{if } x \in K \cup E^B, \end{cases} \quad (9)$$

where, in the second case, the current site, $s = (a, i)$, determines the values of a and i for the current term of the sum.

6 Quenched occupancy function, F

6.0.1 Function F

The fractional occupancy of activator a at site i after quenching is modeled as the product of the unquenched occupancy function, f (Section 9), multiplied by the quenching function, Q (Section 7). The quenched occupancy function is

$$F(a, i, v_a) = f(a, i, v_a) Q^{(1, N^B)}(i). \quad (10)$$

6.0.2 Derivatives of F

The quenched occupancy depends only on the free parameters K_a , K^B and E^B . The function $f(a, i, v_a)$, assuming no competition for binding sites, depends only on the free parameter K_a . The function $Q^{(1, N^B)}(i)$ depends only on the sets of free parameters E^B and K^B , the effectiveness and maximal binding affinity parameters for the quenchers. So the non-zero partial derivatives of the quenched occupancy function, F , with respect to the free parameters of the model are

$$\frac{\partial F(a, i, v_a)}{\partial x} = \begin{cases} \frac{\partial f(a, i, v_a)}{\partial K_a} Q^{(1, N^B)}(i), & \text{if } x = K_a, \\ f(a, i, v_a) \frac{\partial Q^{(1, N^B)}(i)}{\partial x}, & \text{if } x \in K^B \cup E^B. \end{cases} \quad (11)$$

7 Quenching function, Q

7.0.3 Function Q

The quencher sites, S^B , are arbitrarily ordered from one to N^B . The effect of quenching sites depends on their occupancy, f (Section 9), scaled by a distance-based quenching function, $d(i, j)$ (Section 8, that depends on the distance between the position of the activator site, i , and the position of the quencher site, k). This, in turn, is scaled by the effectiveness factor, E_b . (The position and TF for site s_j are determined from the site itself, since $s_j = (b, k)$.)

We define the complete quenching function using the partial quenching function, which captures the effects of quencher sites s_m through s_n , on any activator site at position i , and is given by

$$Q^{(m,n)}(i) = \prod_{j=m}^n [1 - (d(i, k)E_b f(b, k, v_b))], \quad (12)$$

where k and b are determined for each j in the product from the values in site $s_j = (b, k)$. The complete quenching function considers the effects of all quencher sites, and is given by $Q^{(1, N^B)}(i)$.

Note that the (partial) quenching function is independent of the activator being quenched. It depends only on the position, i , in the DNA enhancer sequence. So it only needs to be computed once.

7.0.4 Derivatives of Q

The quenching function is the product of a series of partial quenching functions, each of which depends on one, particular quenching site, $s_j = (b, k)$. In cases where the current quencher site s_j is not overlapping with any other sites, each of these partial functions depends only on the two free parameters E_b and K_b , the effectiveness and binding affinity of the quencher b . The complete quenching function depends on the sets of free parameters E^B and K^B .

Using the chain rule, we can write the derivatives for the partial quenching functions, $Q^{(m,n)}(i)$ with respect to free variable x using the recursions

$$\frac{\partial Q^{(m,n)}(i)}{\partial x} = \frac{\partial Q^{(m,m)}(i)}{\partial x} Q^{(m+1,n)}(i) + Q^{(m,m)}(i) \frac{\partial Q^{(m+1,n)}(i)}{\partial x}, \quad (13)$$

for $x \in E^B \cup K^B$ and $1 \leq m < n \leq N^B$. The above equation covers the case when $m < n$. The base case of the recursion is when $m = n$. In that case, then $Q^{(m,m)}(i) = 1 - (d(i, k)E_b f(b, k, v_b))$, where the quencher site is $s_m = (b, k)$. So, the derivatives of Q when $m = n$ are

$$\frac{\partial Q^{(m,m)}(i)}{\partial x} = \begin{cases} -d(i, k) \frac{\partial E_b}{\partial x} f(b, k, v_b), & \text{if } x = E_b, \\ -d(i, k) E_b \frac{\partial f(b, k, v_b)}{\partial x}, & \text{if } x = K_b, \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

for $x \in E^B \cup K^B$ and $1 \leq m \leq N^B$.

Since we are only interested in computing the derivatives of the complete quenching function, $Q^{(1, N^B)}(i)$, we only need compute the partial functions $Q^{(m,n)}(i)$ for the cases where $n = N^B$ or $m = n$.

In cases where the current quencher site s_j is overlapping with other sites of activators or quenchers the $f2(a, i, v_a)$ function has to be used. This, however, does not only affect $\frac{\partial Q}{\partial K_b}$ and $\frac{\partial Q}{\partial E_b}$ of current quencher but effect the derivatives for all other K_a overlapping with the current quencher site.

$$\frac{\partial Q^{(m,m)}(i)}{\partial x} = \begin{cases} -d(i,k) \frac{\partial E_b}{\partial x} f2(b,k,v_b), & \text{if } x = E_b, \\ -d(i,k) E_b \frac{\partial f2(b,k,v_b)}{\partial x}, & \text{if } x = K_b, \\ f2(b,k,v_b), & \text{otherwise.} \end{cases} \quad (15)$$

for $x \in E^B \cup K^B$ and $1 \leq m \leq N^B$.

8 Distance-based Quenching Function, $d(i, j)$

8.0.5 Function $d(i, j)$

The distance-based quenching function, $d(i, j)$, captures the notion that the action of quenchers is local. It can be any function that takes two positive integers and returns a value between 0 and 1. We use the function in Janssens et al. [2006]. It is a symmetrical, trapezoidal impulse function centered around distance 0. It is defined as

$$d(i, j) = \begin{cases} 1, & \text{if } |i - j| < 100, \\ 0, & \text{if } |i - j| > 150, \\ 1 - \frac{150 - |i - j|}{50}, & \text{otherwise.} \end{cases} \quad (16)$$

8.0.6 Derivatives of $d(i, j)$

Not required, since $d(i, j)$ does not depend on the free parameters of the model.

9 Occupancy function, $f(a, i, v_a)$

9.0.7 Function $f(a, i, v_a)$

If there is no competition for binding sites, the occupancy of site $s = (a, i)$ is defined to be

$$f(a, i, v_a) = \frac{K(a, i)v_a}{1 + K(a, i)v_a}, \quad (17)$$

where v_a is the concentration of TF a , and $K(a, i)$ (Section 11) is the binding affinity of TF a at position i in the enhancer sequence.

9.0.8 Derivatives of $f(a, i, v_a)$

The function depends only on free parameter K_a via the binding affinity function $K(a, i)$. Therefore, the only non-zero derivative is

$$\frac{\partial f(a, i, v_a)}{\partial K_a} = \frac{v_a}{(1 + K(a, i)v_a)^2} \frac{\partial K(a, i)}{\partial K_a}. \quad (18)$$

10 Occupancy function considering overlapping sites, $f2(a, i, v_a)$

10.0.9 Function $f2(a, i, v_a)$

If there is competition for binding sites, the occupancy of site $s = (a, i)$ is defined to be

$$f2(a, i, v_a) = \frac{K(a, i)v_a}{1 + K(a, i)v_a + \sum_{j \in O_{K(a, i)}} K(b, j)v_b}, \quad (19)$$

where again v_a is the concentration of TF a , and $K(a, i)$ (Section 11) is the binding affinity of TF a at position i in the enhancer sequence. $O_{K(a, i)}$ are all sites overlapping with $K(a, i)$, where b can be any TF including a .

10.0.10 Derivatives of $f2(a, i, v_a)$

The derivative for K_a for the active site i is

$$\frac{\partial f2(a, i, v_a)}{\partial K_a} = \frac{\frac{\partial K(a, i)}{\partial K_a} v_a \left(1 + \sum_{j \in O_{K(a, i)}^{b \neq a}} K(b, j)v_b\right)}{\left(1 + K(a, i)v_a + \sum_{j \in O_{K(a, i)}} K(b, j)v_b\right)^2}, \quad (20)$$

where $O_{K(a, i)}^{b \neq a}$ are all sites overlapping with $K(a, i)$ whose TF is different from a .

However, there is a small fraction of the derivative for the K_a s of all other TFs overlapping with $K(a, i)$, which needs to be calculated for every $f2$.

$$\frac{\partial f2(a, i, v_a)}{\partial K_b} = \frac{-K(a, i)v_a \sum_{j \in O_{K(a, i)}^{b = a}} \frac{\partial K(b, j)}{\partial K_b} v_b}{\left(1 + K(a, i)v_a + \sum_{j \in O_{K(a, i)}} K(b, j)v_b\right)^2} \quad (21)$$

11 Binding affinity function, $K(a, i)$

11.0.11 Function

The binding affinity of TF a to a site in the enhancer region is assumed to be proportional to the exponent of the difference score of the site and the maximum possible score, \hat{S}_a . So the binding affinity function is

$$K(s, a) = K_a \exp(S(a, i) - \hat{S}_a). \quad (22)$$

The score, $S(a, i)$, is the PWM score of the site of TF a at position i in the enhancer sequence, and is part of the input data.

11.1 Derivatives of $K(a, i)$

The only non-zero derivative is with respect to K_a ,

$$\frac{\partial K(a, i)}{\partial K_a} = \exp(S(a, i) - \hat{S}_a). \quad (23)$$

12 Transformation to the unconstrained optimization space, T

12.1 Function T

The Reinitz model only tolerates certain parameter ranges of Θ . Constrained optimization using gradient descent is not very elegant. We therefore transform every free parameter in Θ into a space, where the optimization is unconstrained, using $T(x)$:

$$T(x) = -\log\left(\frac{r}{x} - 1\right) \cdot \frac{1}{b} \quad x \in \Theta \quad (24)$$

$$= \hat{x}, \quad (25)$$

where b is a problem-dependent scaling value, which is set to 0.01, and r is a scaling value set as the upper limit of the value range of x . The transformation to \hat{x} enables to optimize unconstrained.

13 Transformation to the normal space, T'

13.1 Function T'

To retrieve the real values for every x in Θ we use $T'(\hat{x})$:

$$T'(\hat{x}) = \frac{r}{1 + \exp(-b\hat{x})} \quad \hat{x} \in \hat{\Theta} \quad (26)$$

$$= x, \quad (27)$$

where b is a problem-dependent scaling value, which is set to 0.01, and r is a scaling value set as the upper limit of the value range of x . The transformation to \hat{x} enables to optimize unconstrained. No derivative is needed because this function is only called at the end of the optimization.

13.2 Derivatives of T'

In order to update the parameters we need to calculate the derivative of $T'(\hat{x})$ for parameter \hat{x} is

$$\frac{\partial T}{\partial \hat{x}} = \text{sigm}(\hat{x}b) \cdot (1 - \text{sigm}(\hat{x}b)) \cdot rb \hat{x} \in \hat{\Theta}, \quad (28)$$

$$\text{sigm}(x) = \frac{1}{1 + \exp(-x)}, \quad (29)$$

with, again, b is a problem-dependent scaling value, which is set to 0.01, and r is a scaling value set as the upper limit of the value range of x .

Section 2 : Performance of Gradient Descent Variants

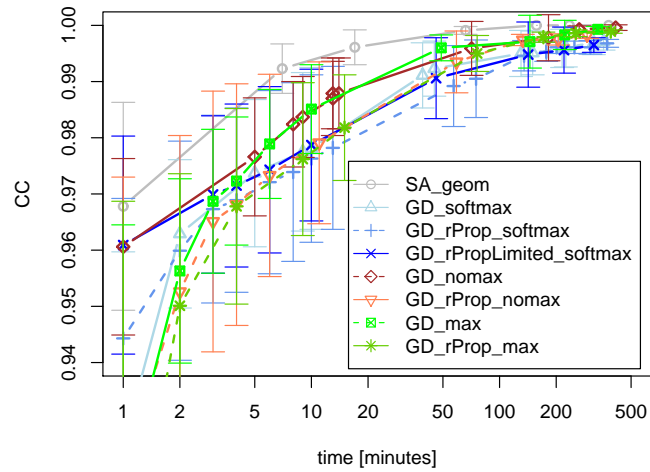


Figure 1: **Comparison of the different GD optimization methods** The figure shows the correlation coefficient (CC) achieved on the test set of each cross validation fold by the different GD variants in comparison with the SA_geom optimization method.

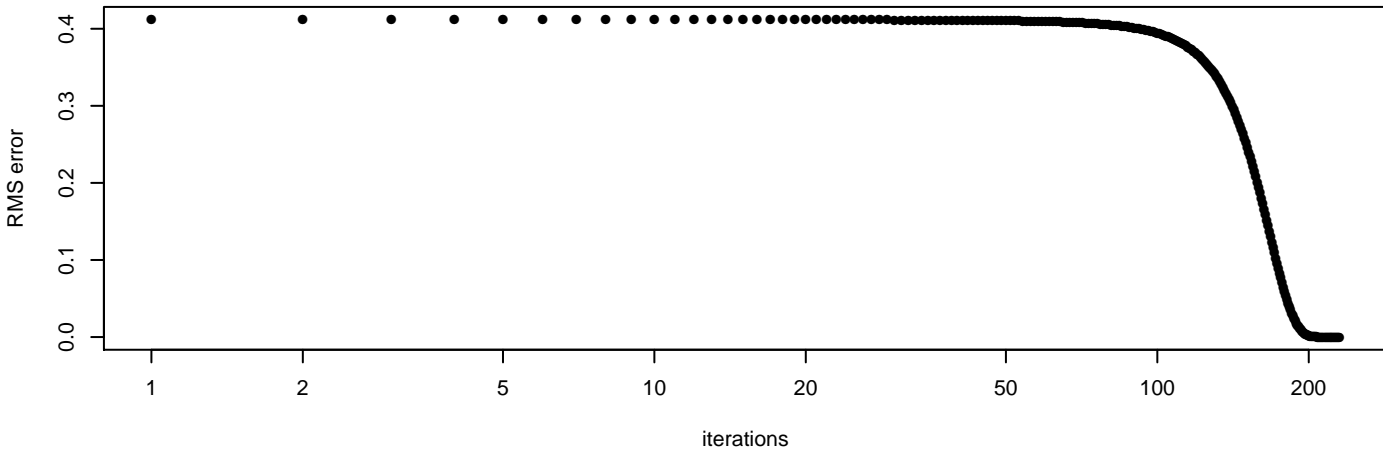
Section 3 : RMS Error and Parameter Distances for Different Levels of Perturbation

Convergence of a parameter set where each parameter is perturbed by * 1% * individually.

Gradient descent is executed with all parameters * held * at their optimal value except the perturbed one.

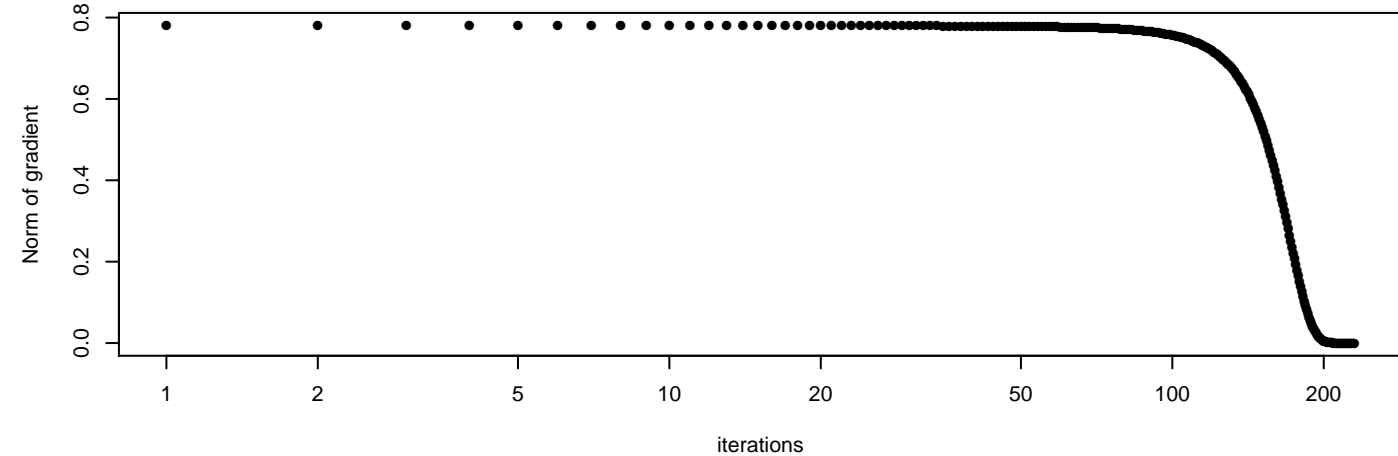
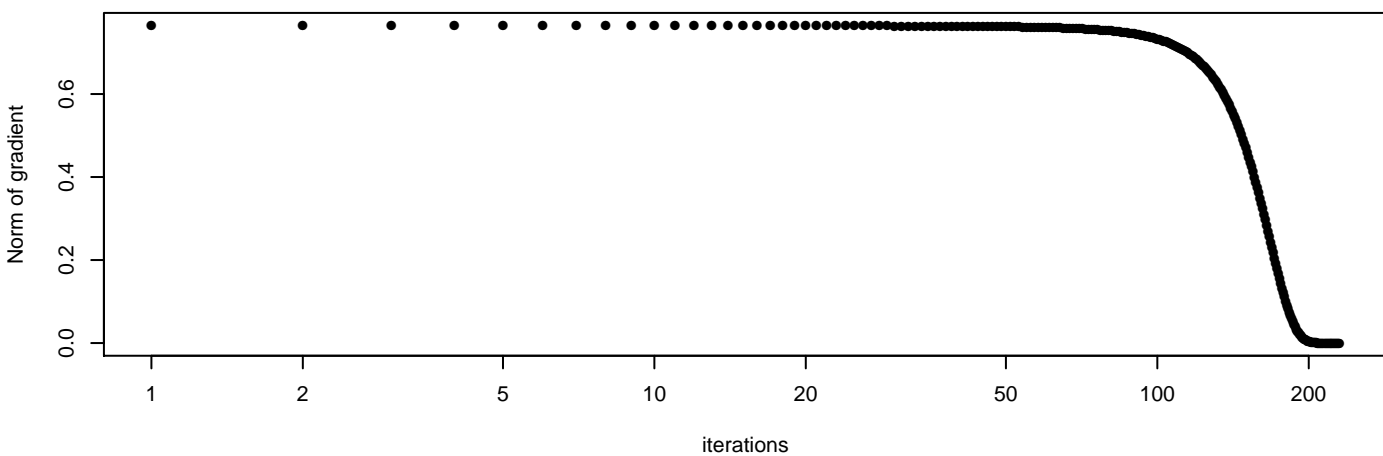
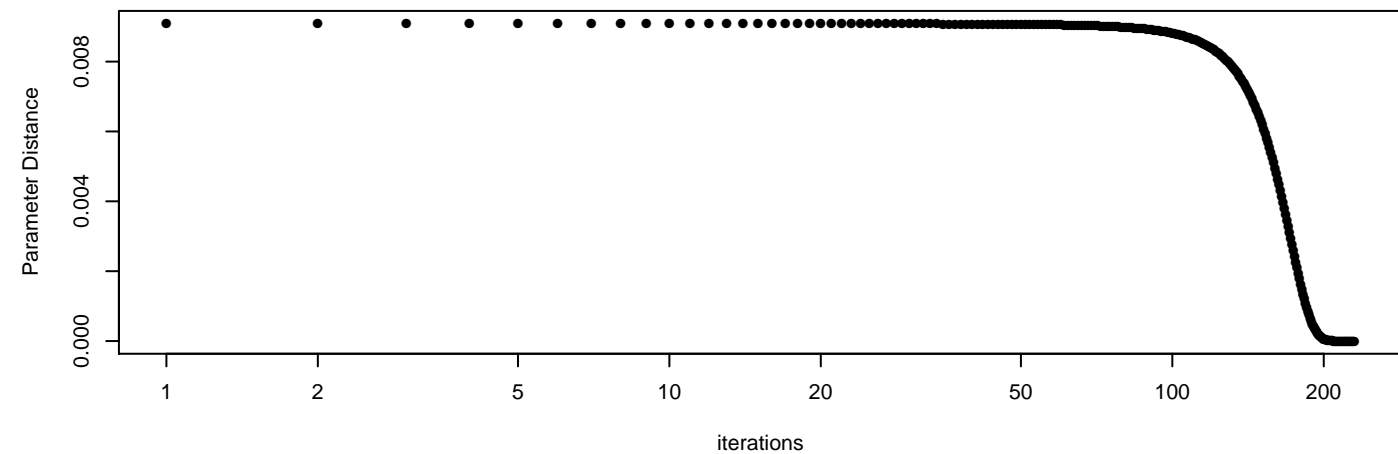
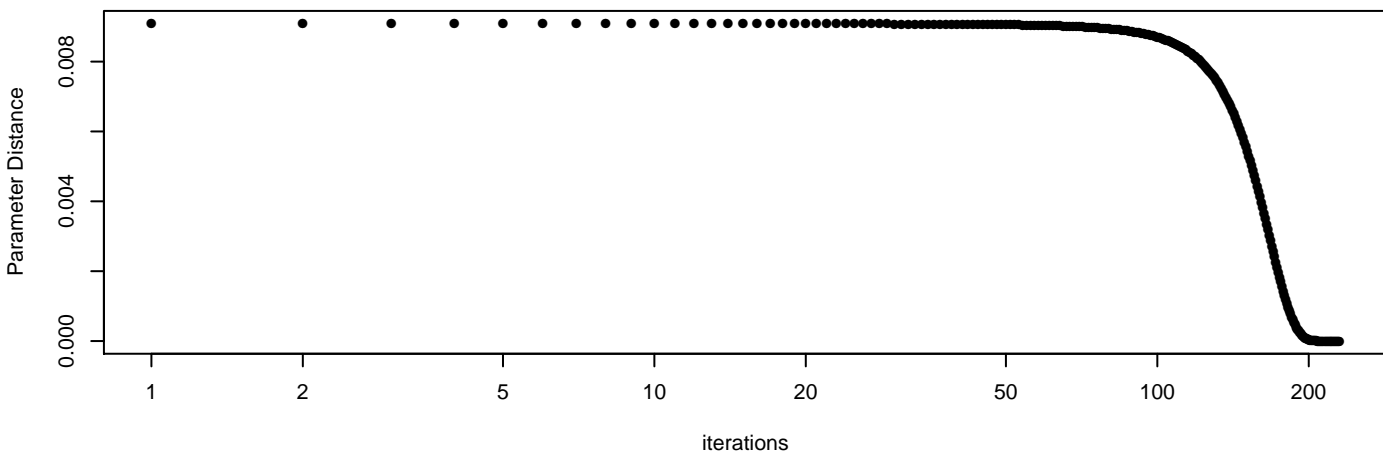
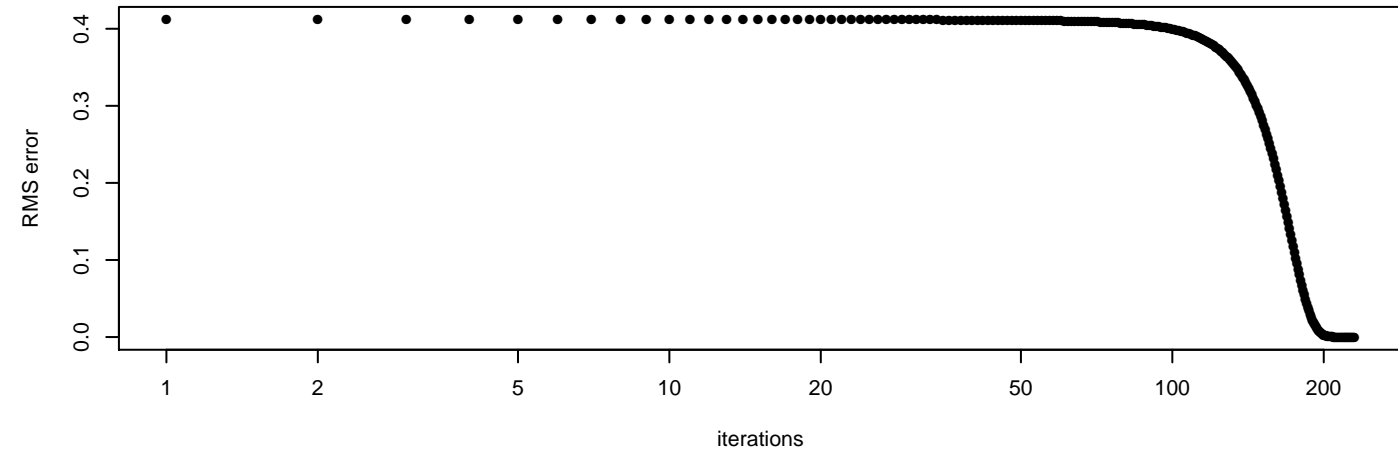
Gradient descent was ran in * normal * mode to take the value of the gradient.

Negative Perturbation

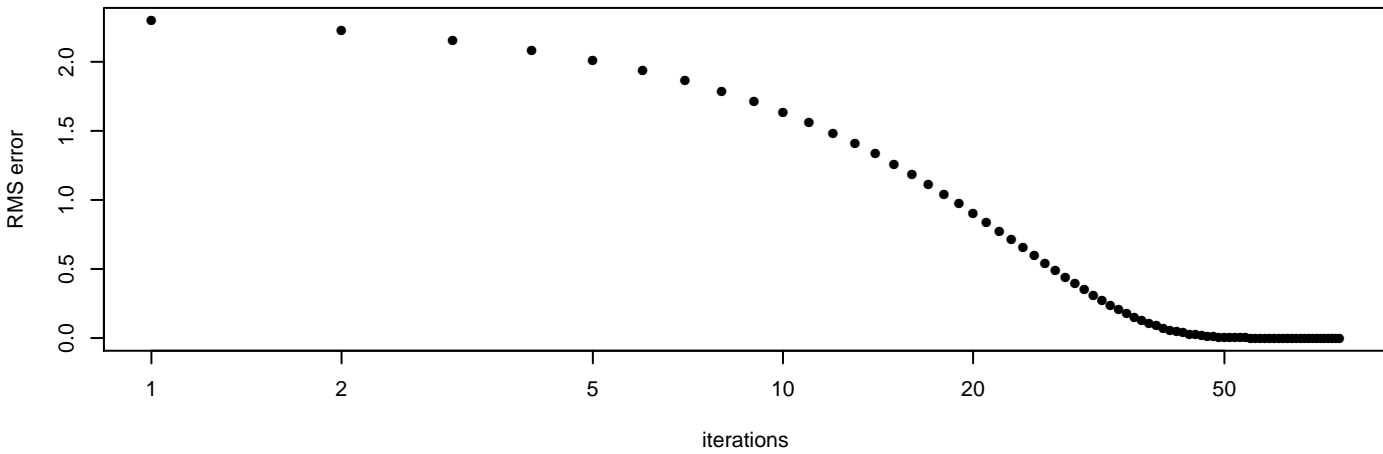


Parameter0

Positive Perturbation

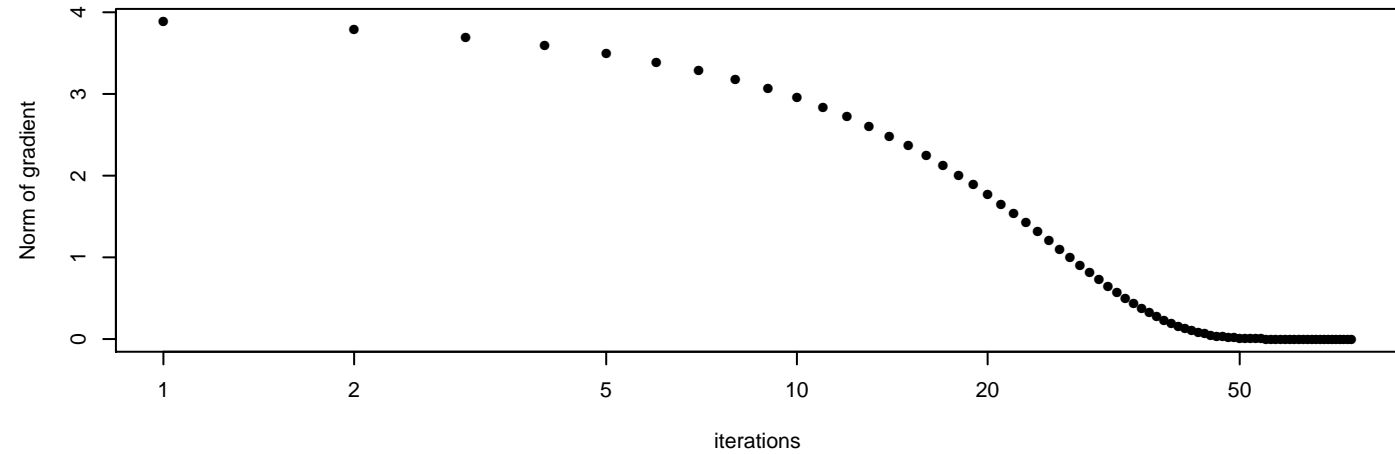
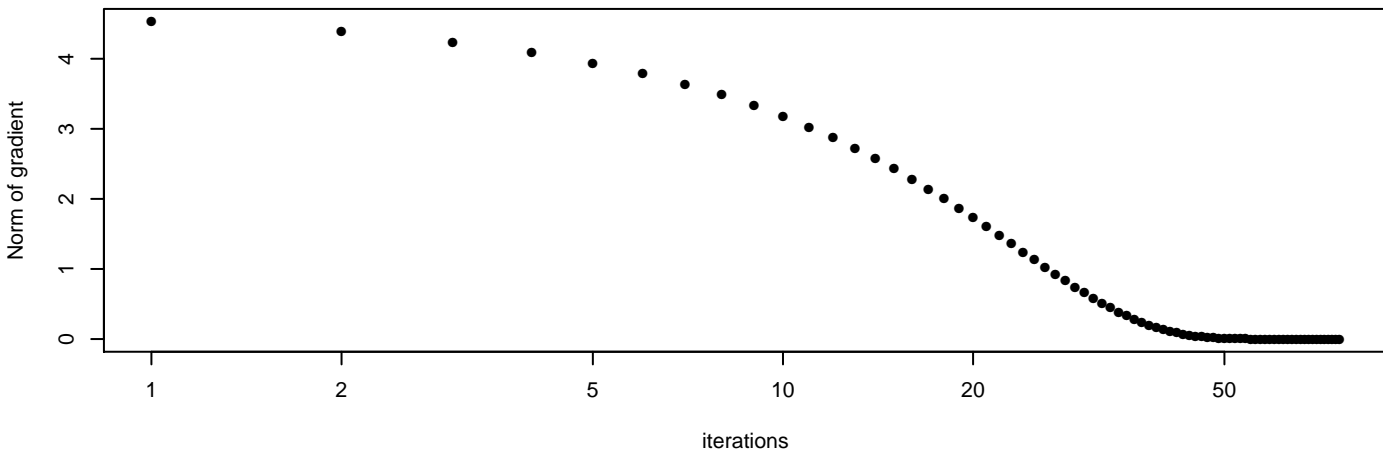
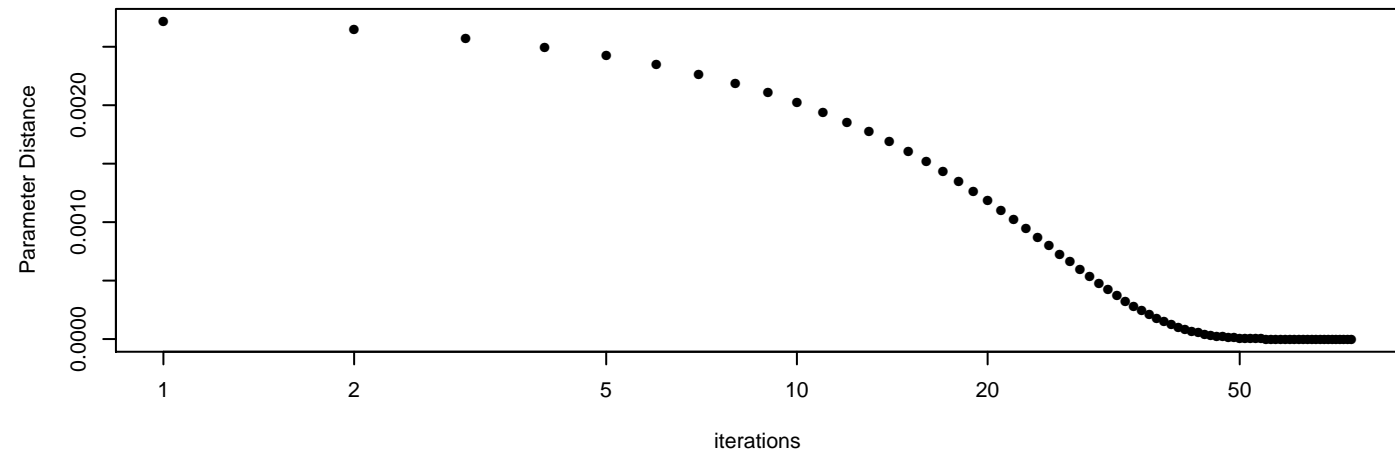
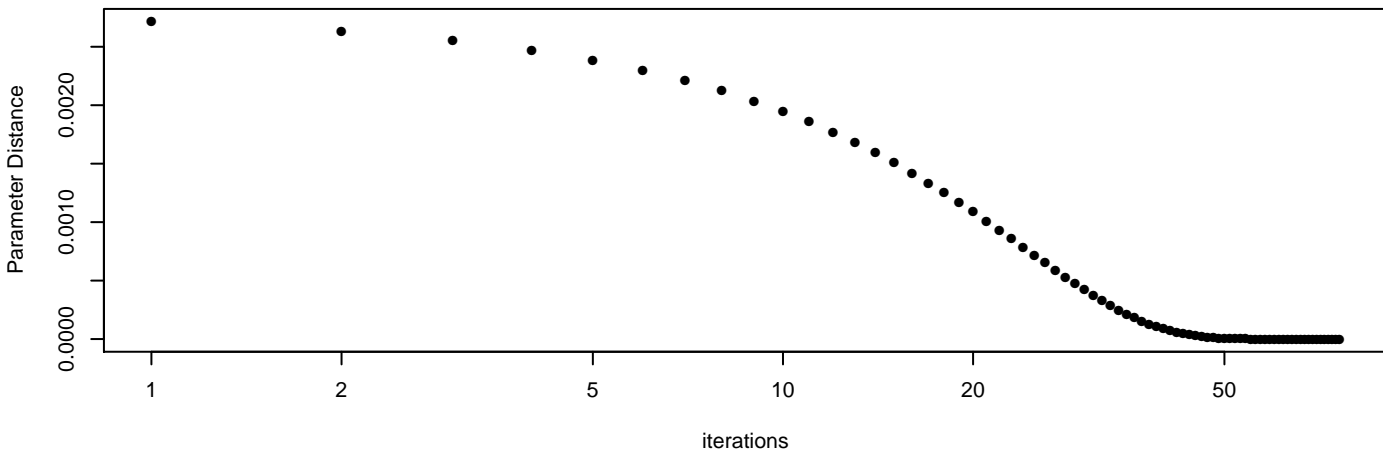
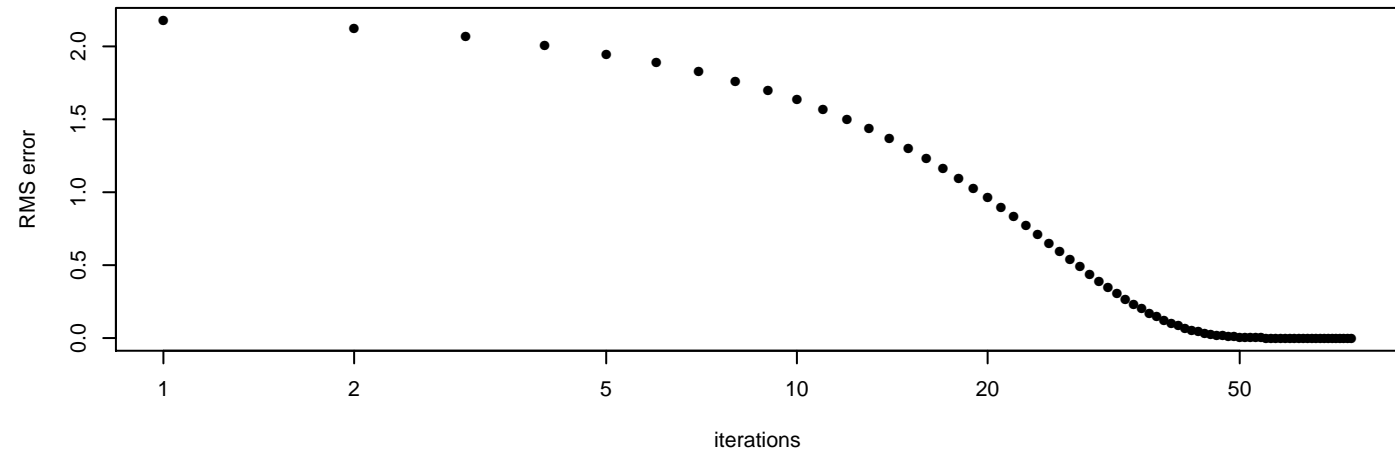


Negative Perturbation



Parameter1

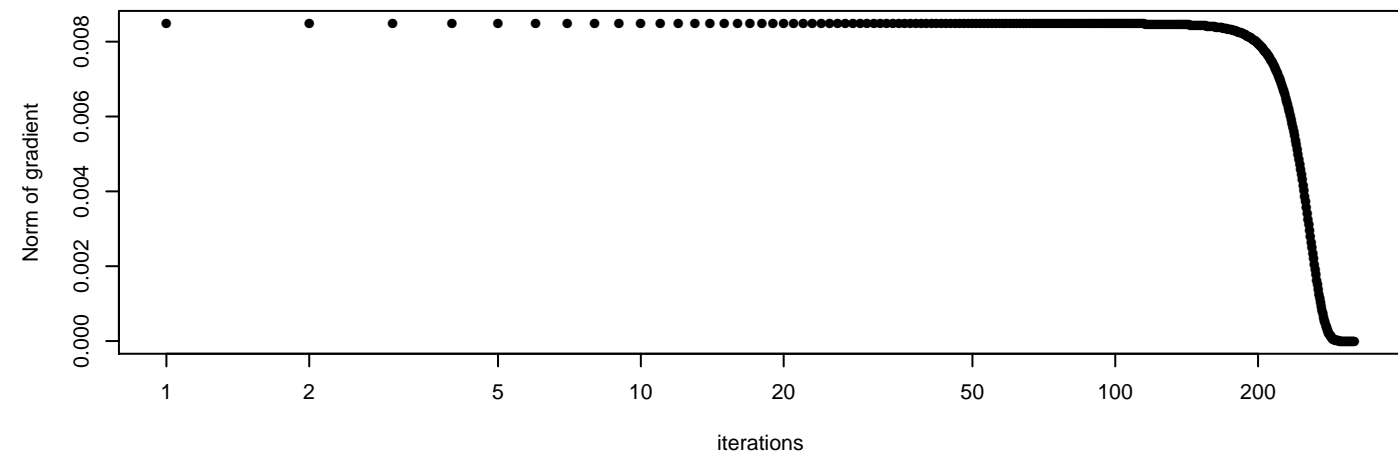
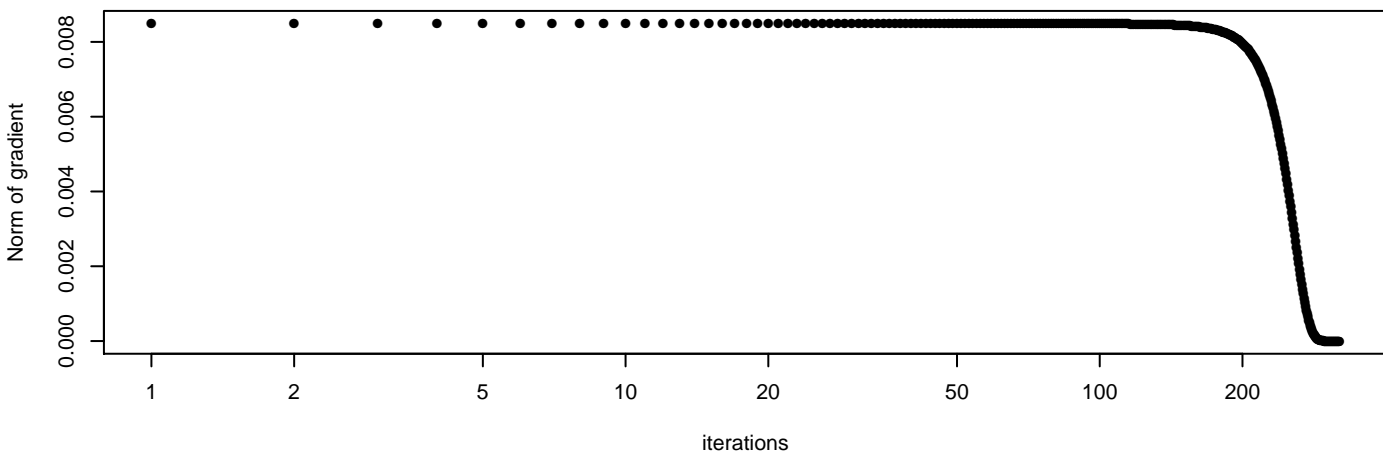
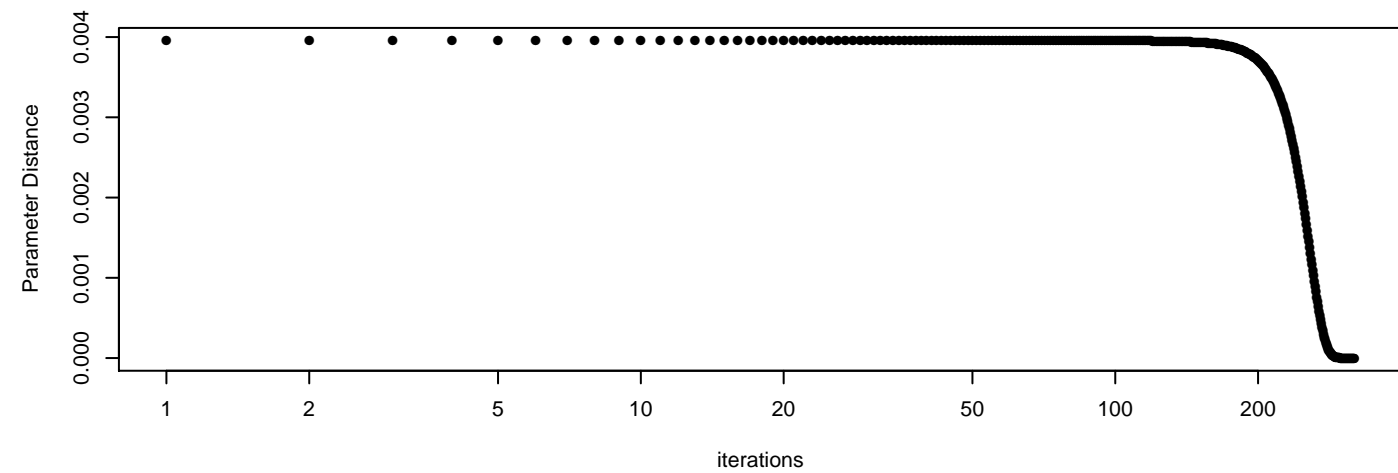
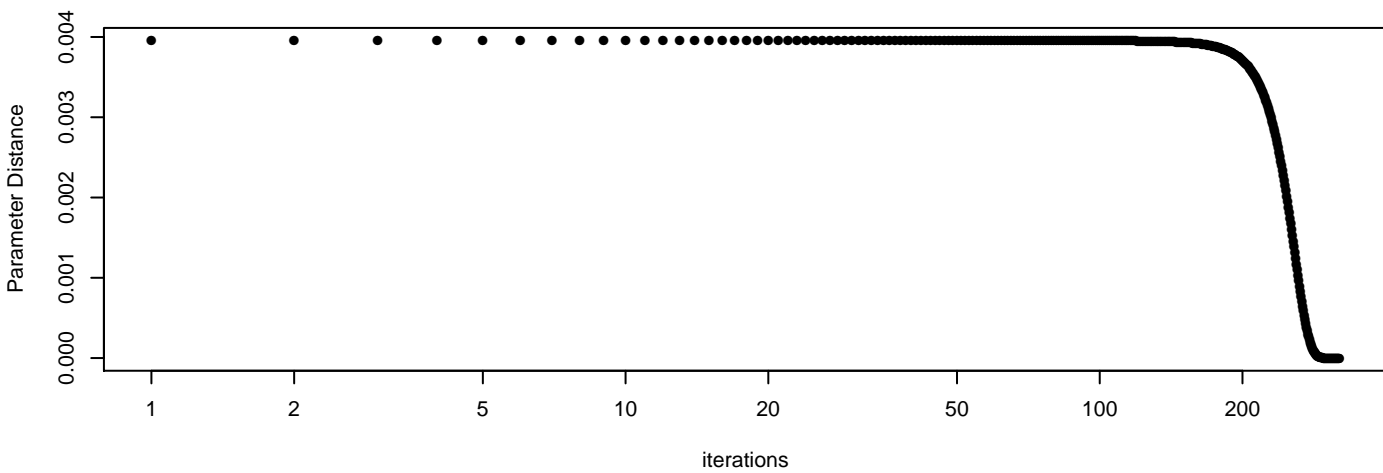
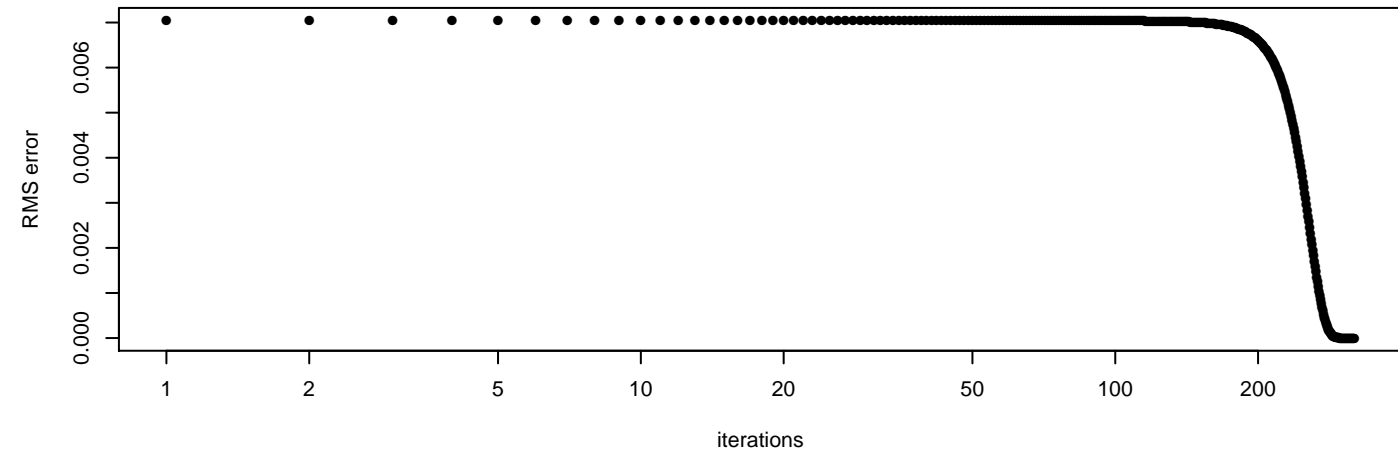
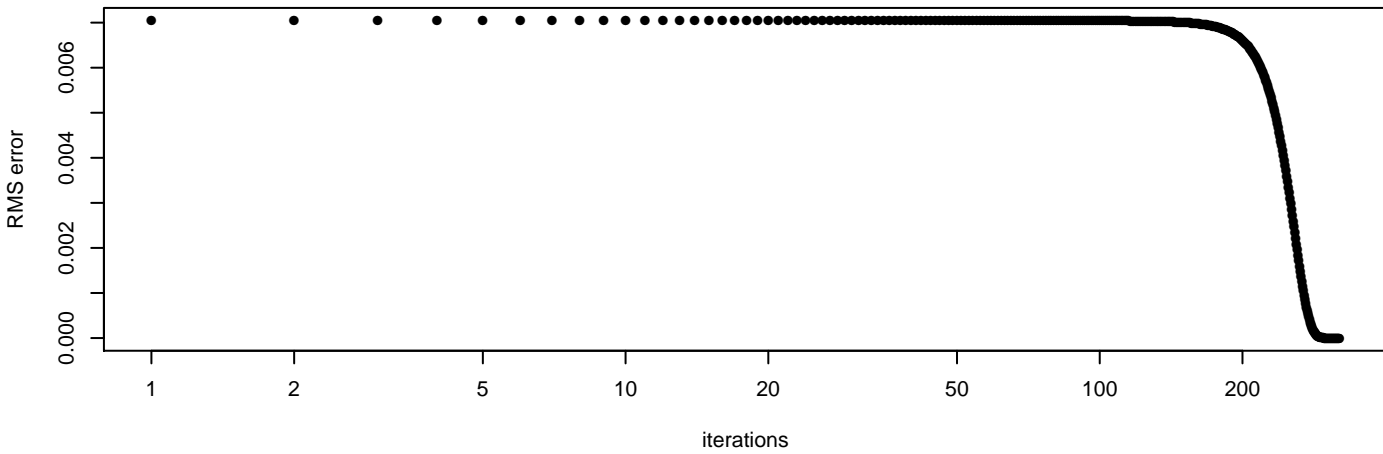
Positive Perturbation



Parameter10

Negative Perturbation

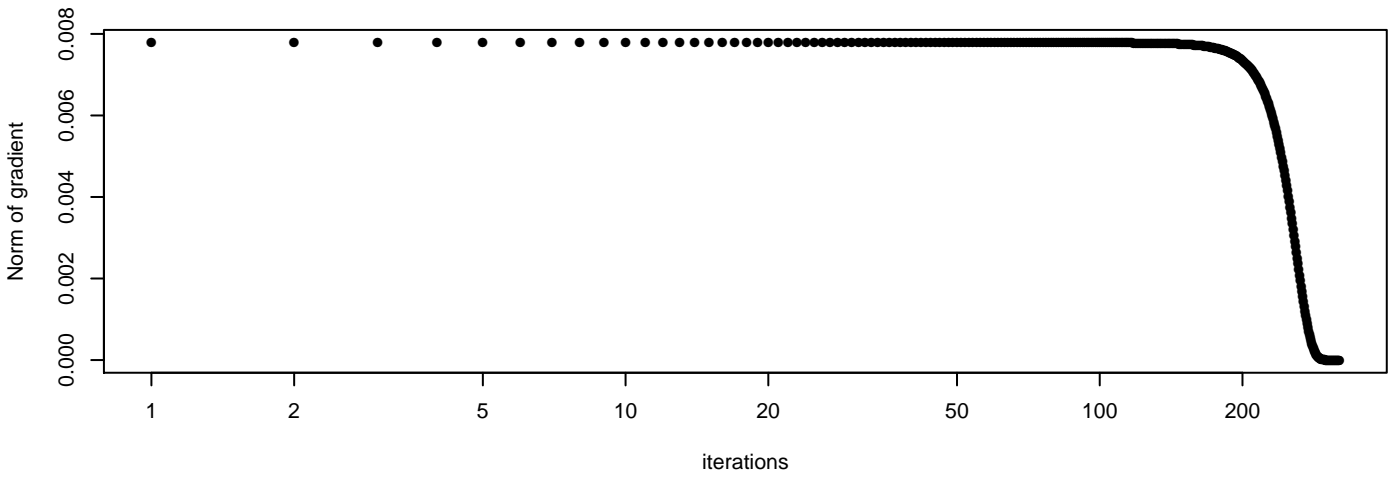
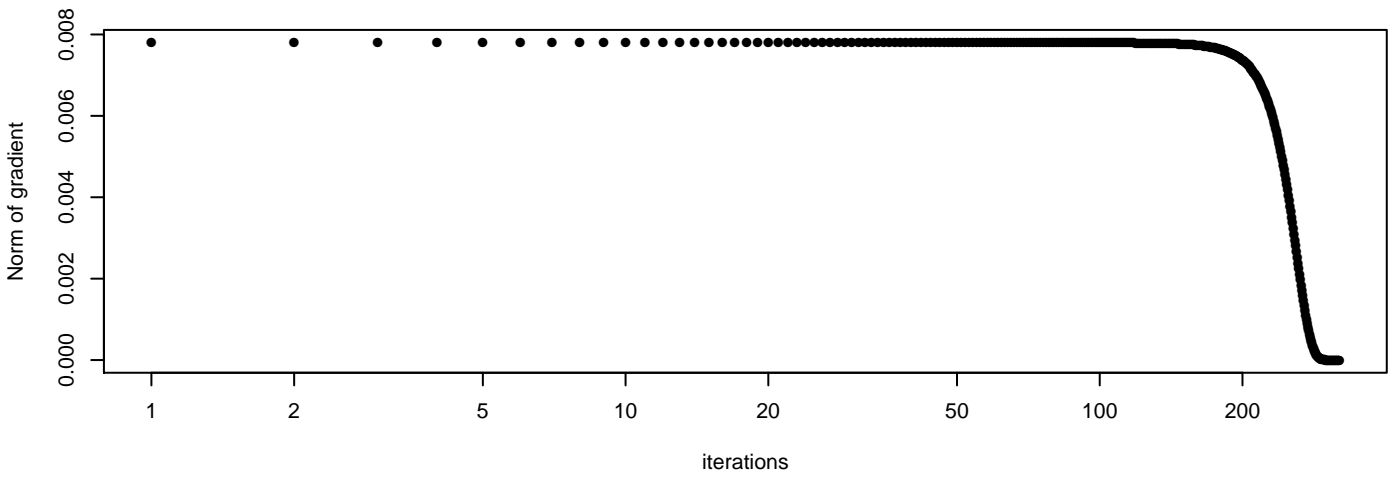
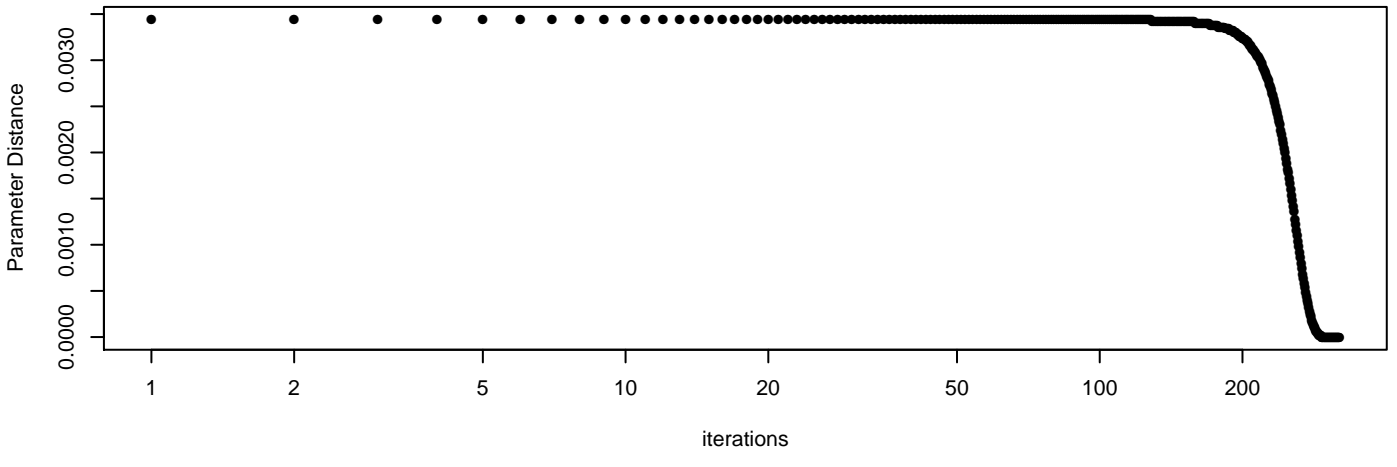
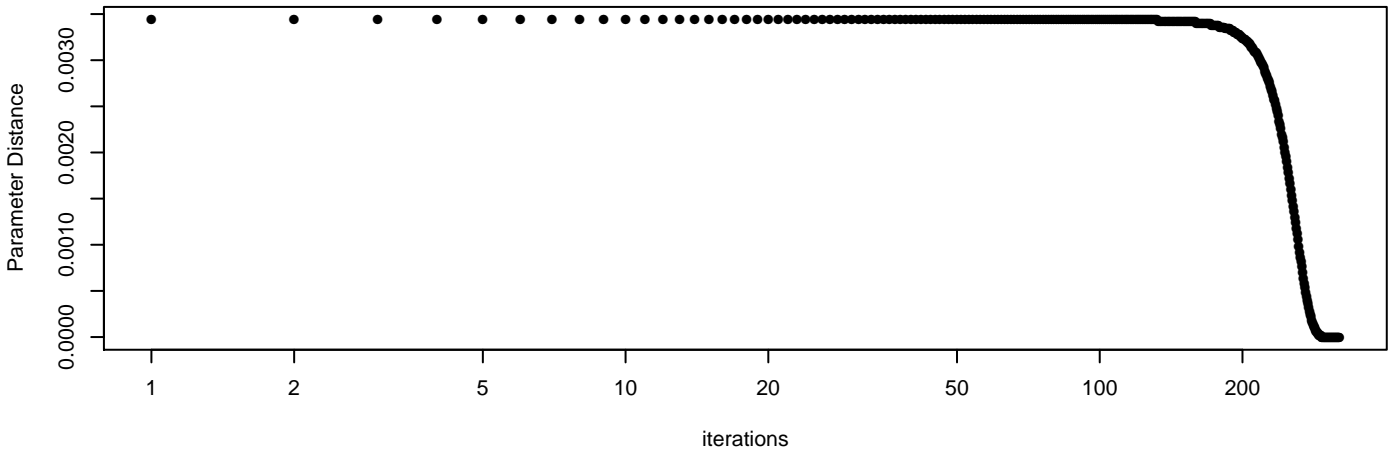
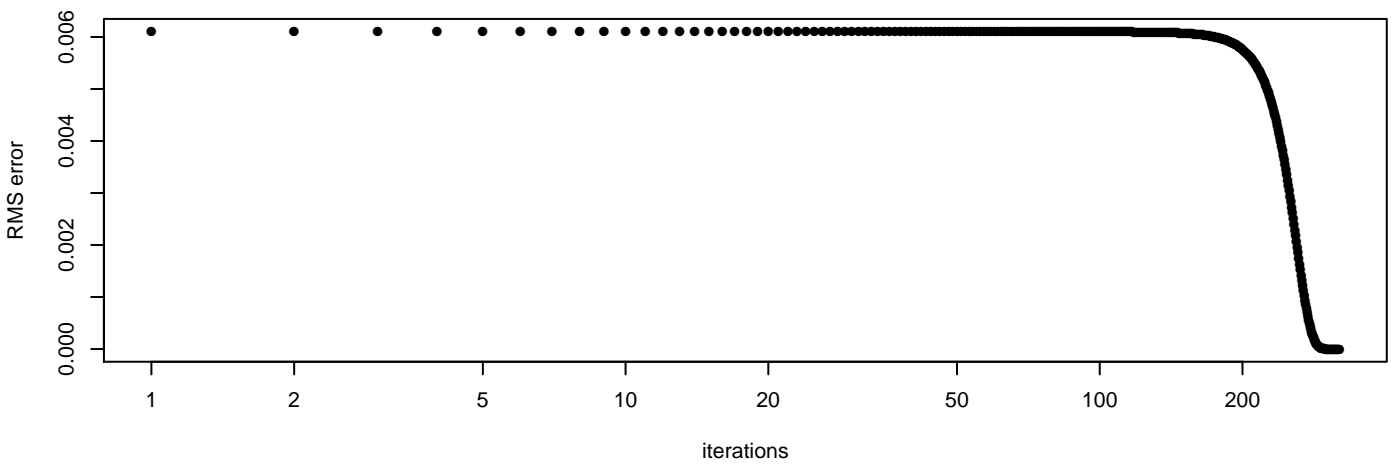
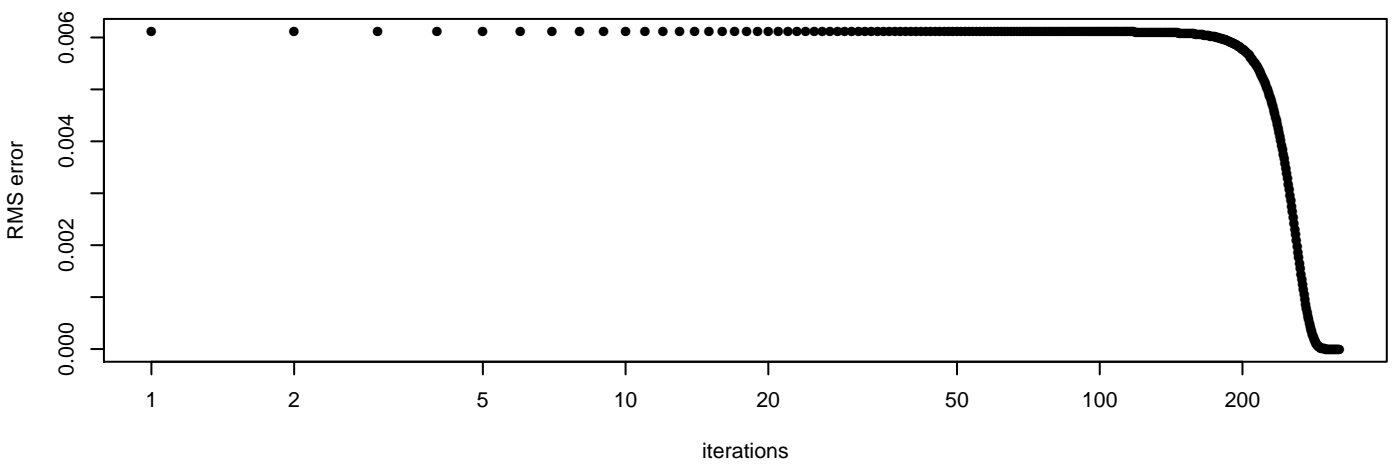
Positive Perturbation



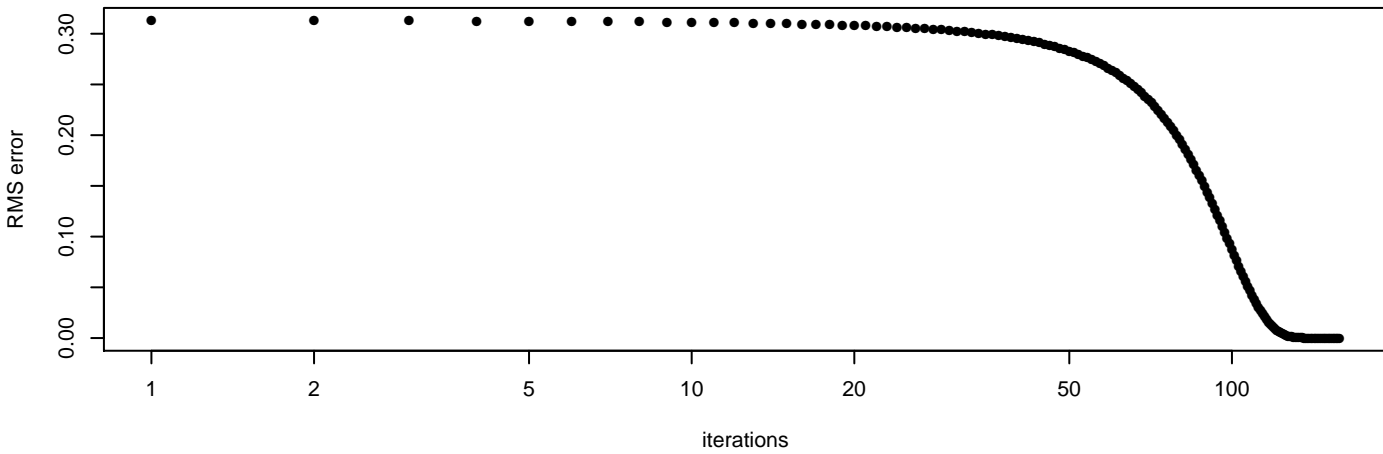
Parameter11

Negative Perturbation

Positive Perturbation

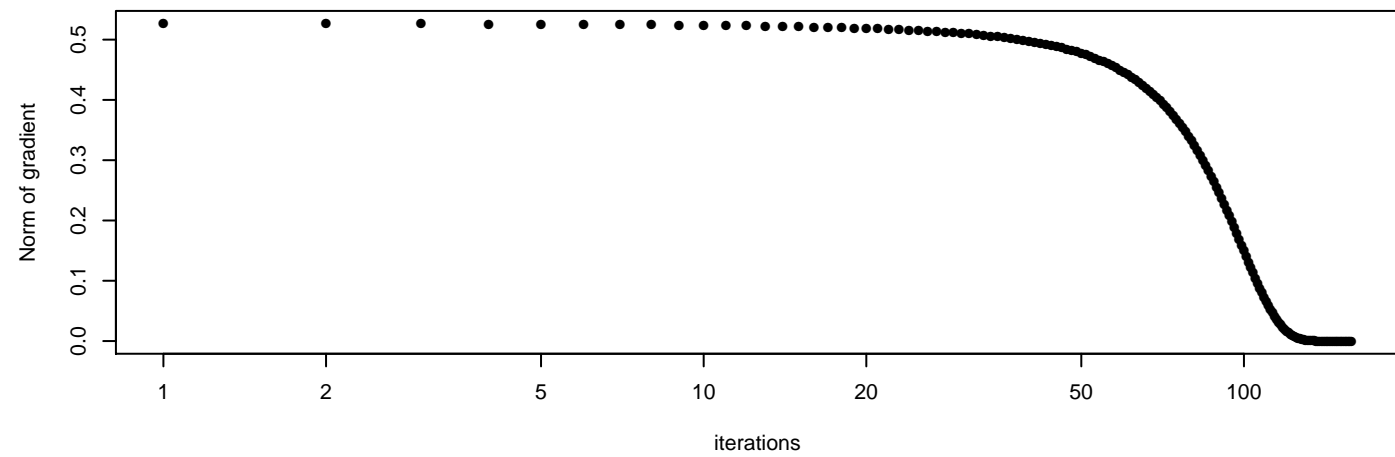
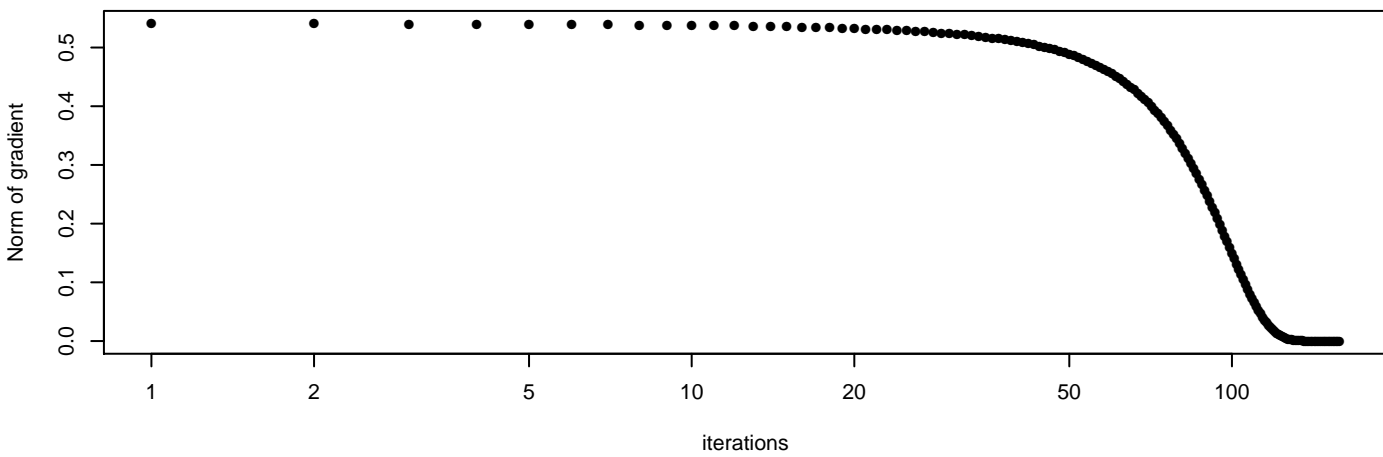
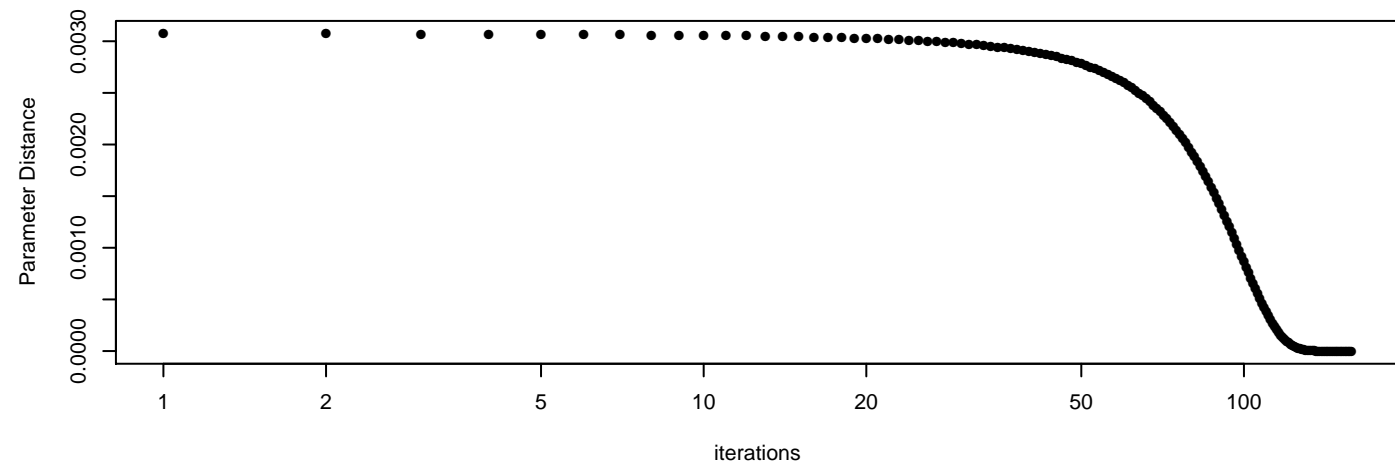
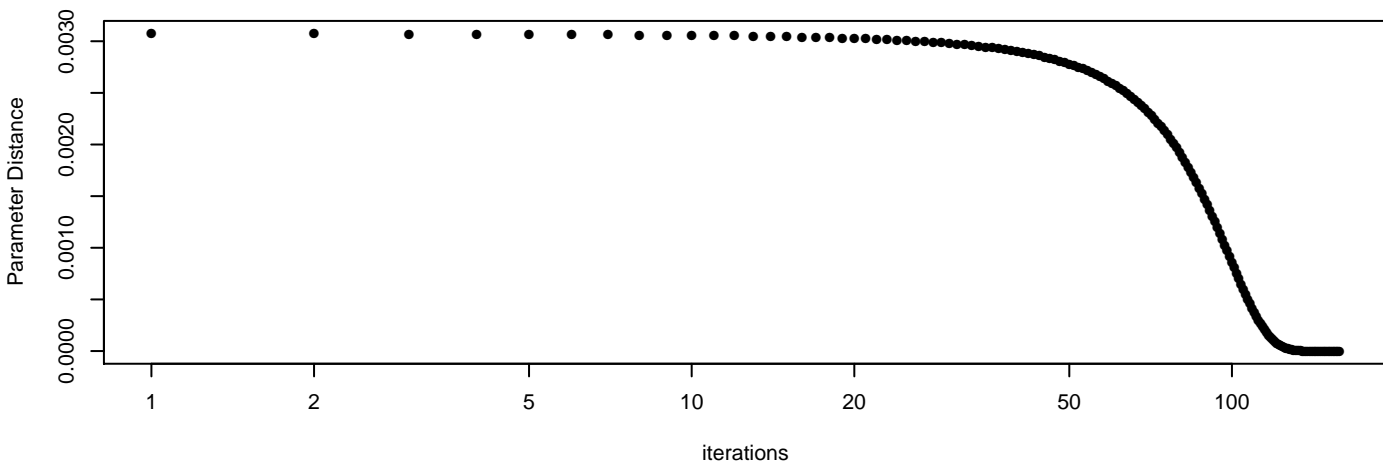
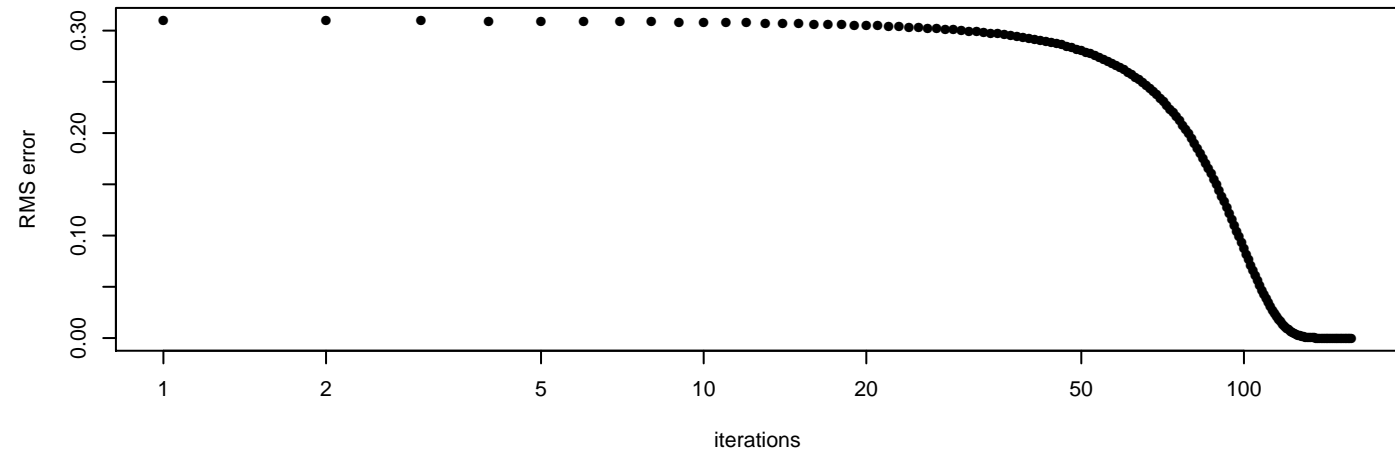


Negative Perturbation

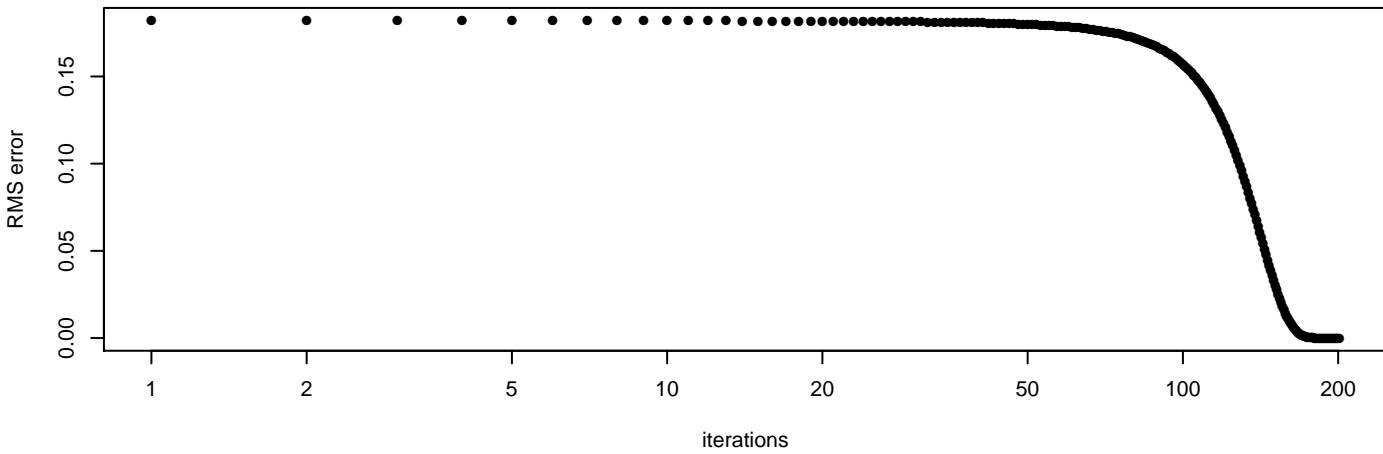


Parameter12

Positive Perturbation

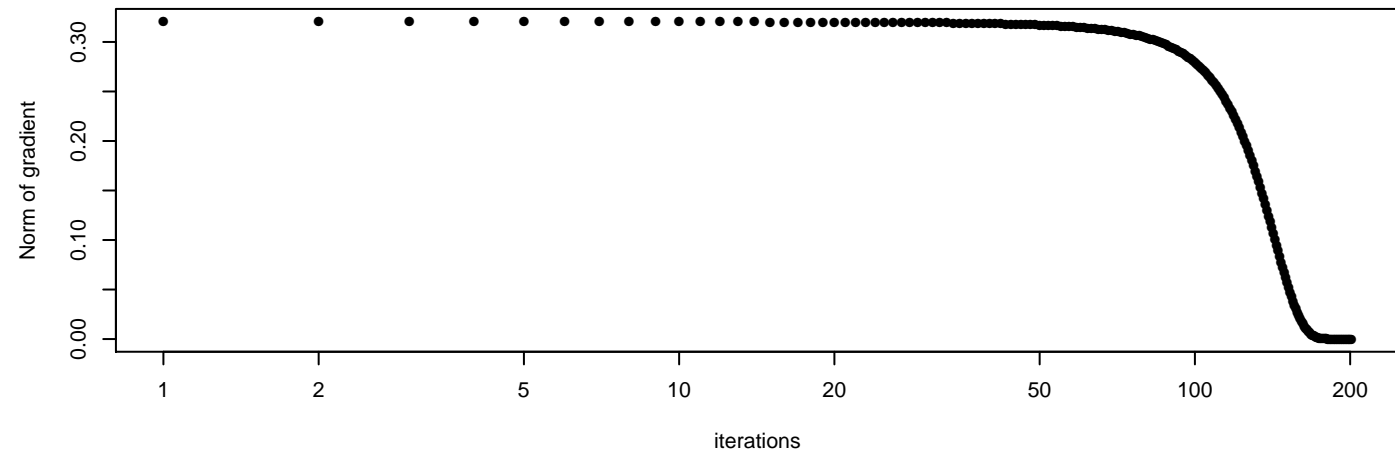
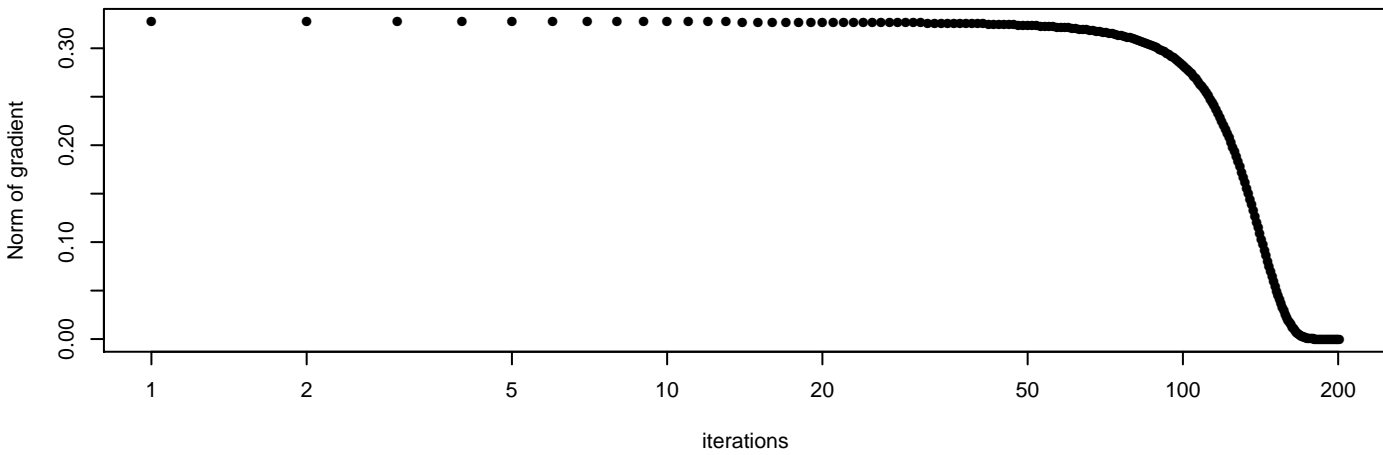
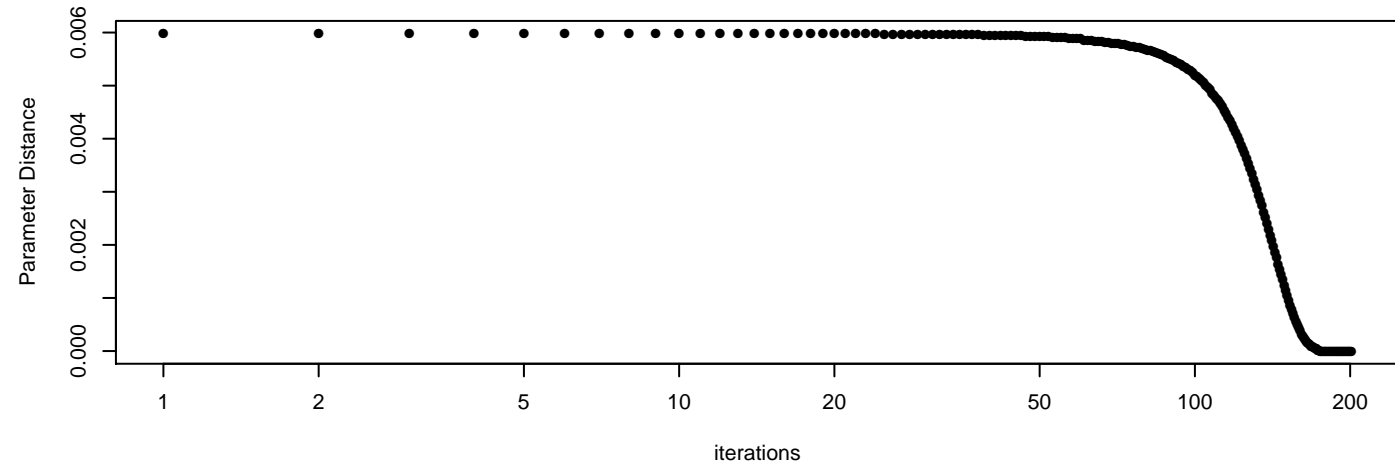
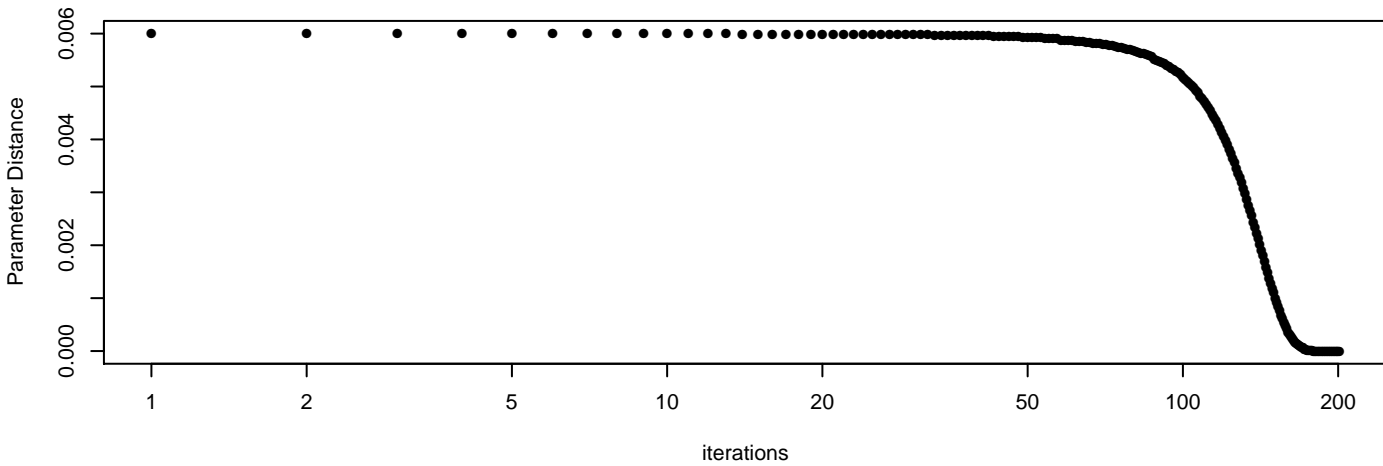
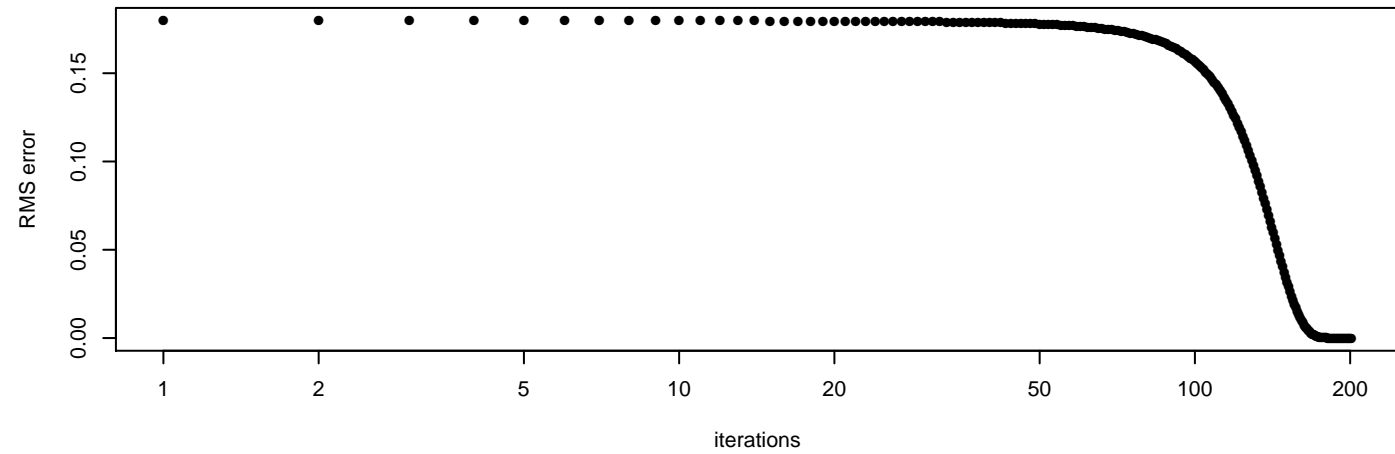


Negative Perturbation

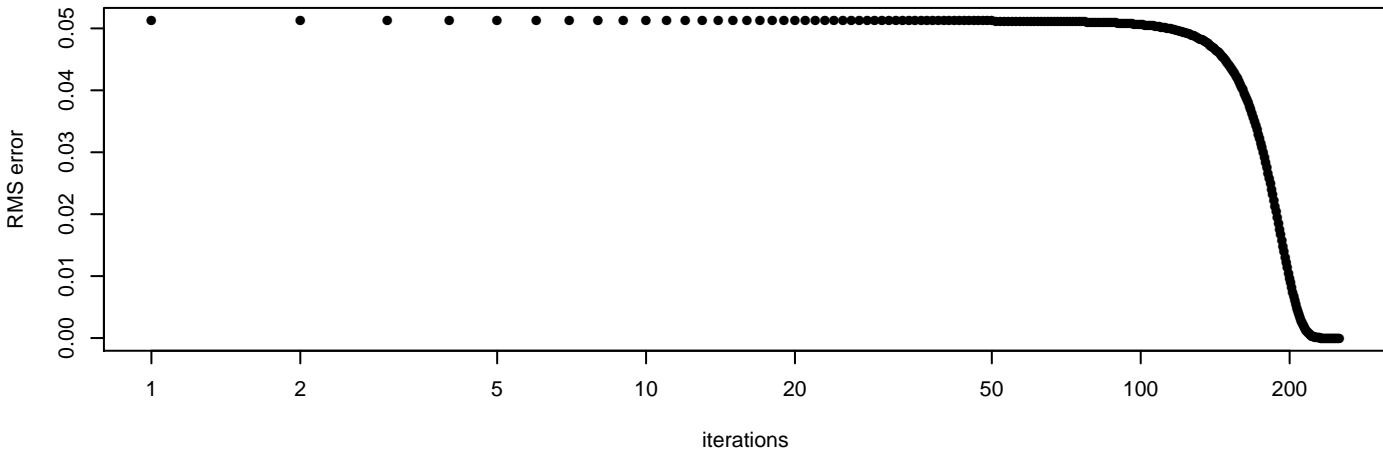


Parameter13

Positive Perturbation

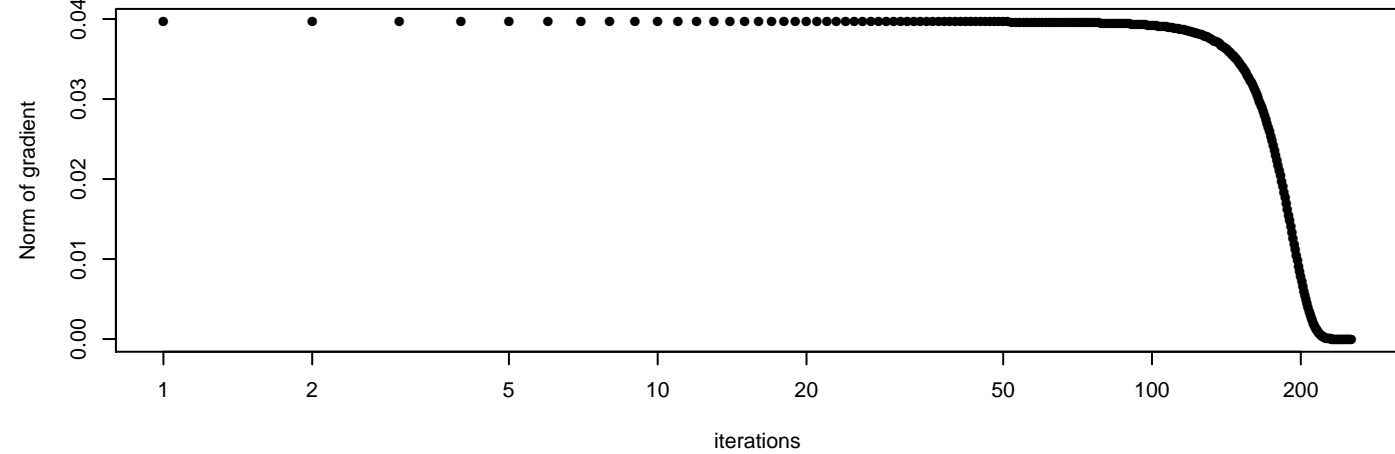
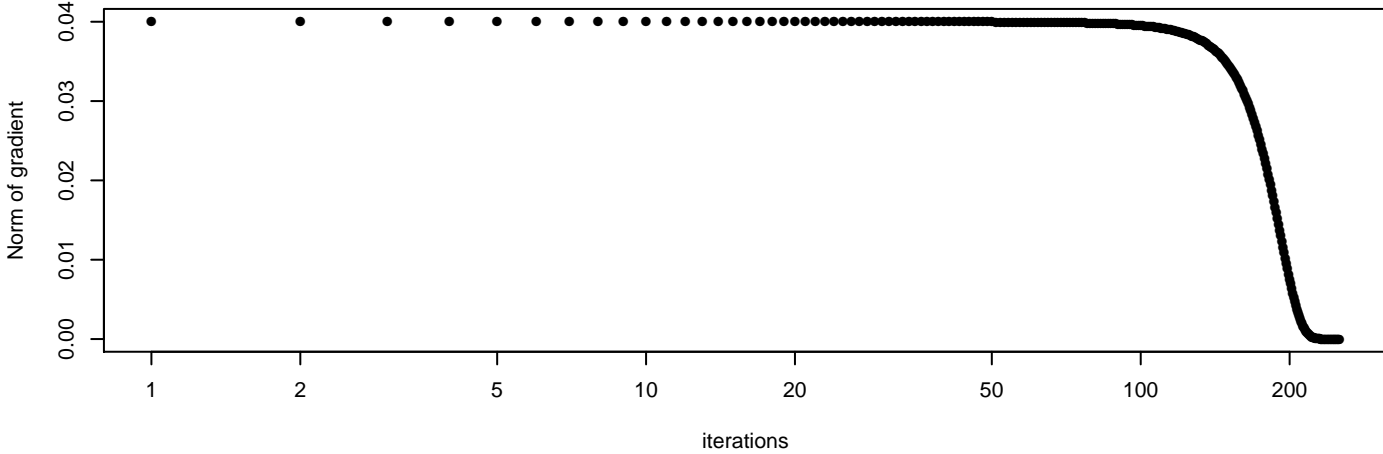
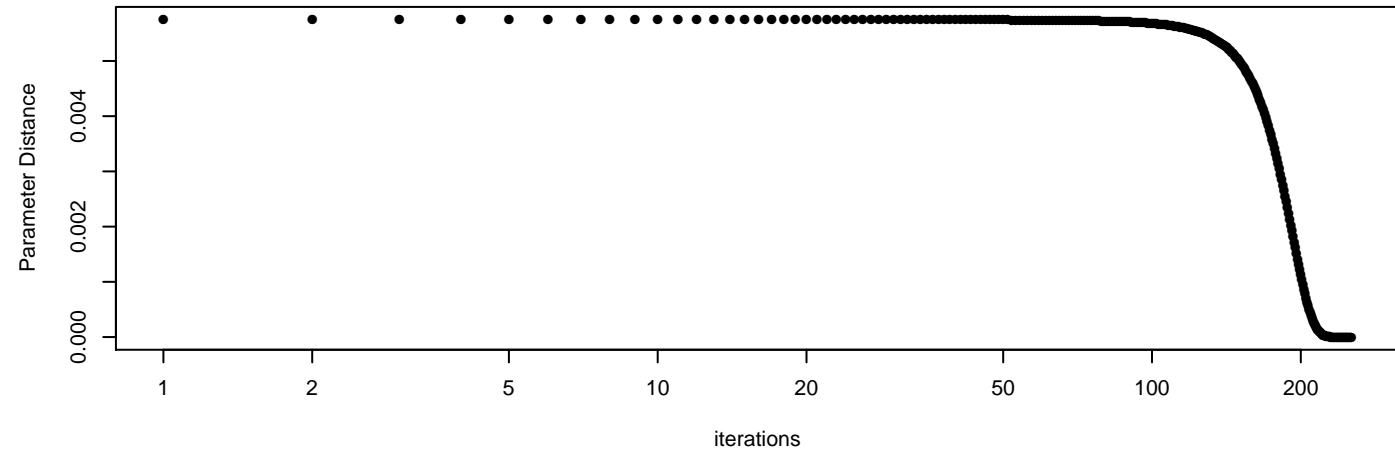
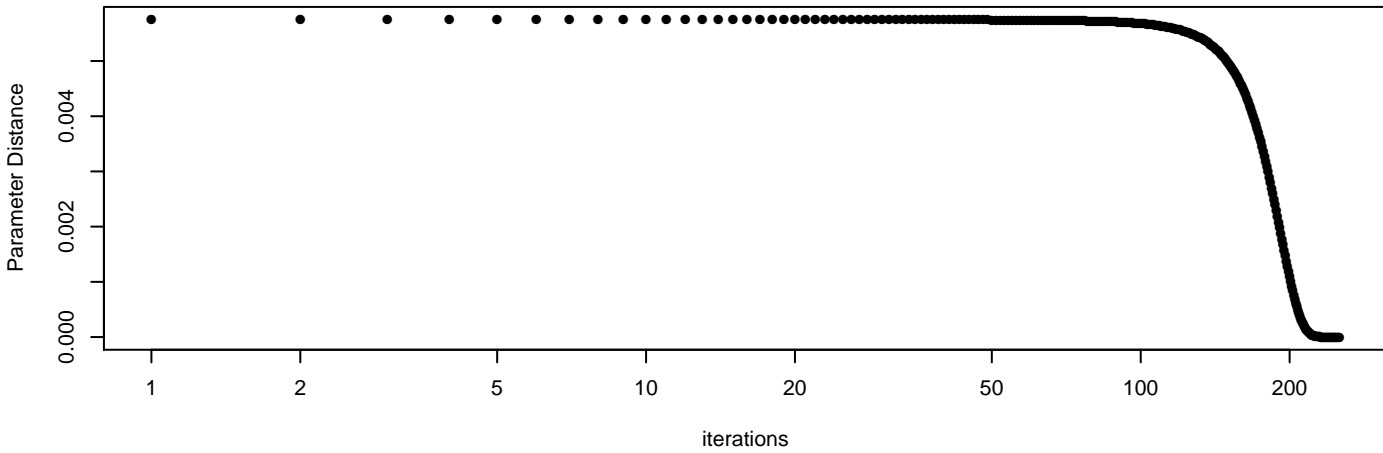
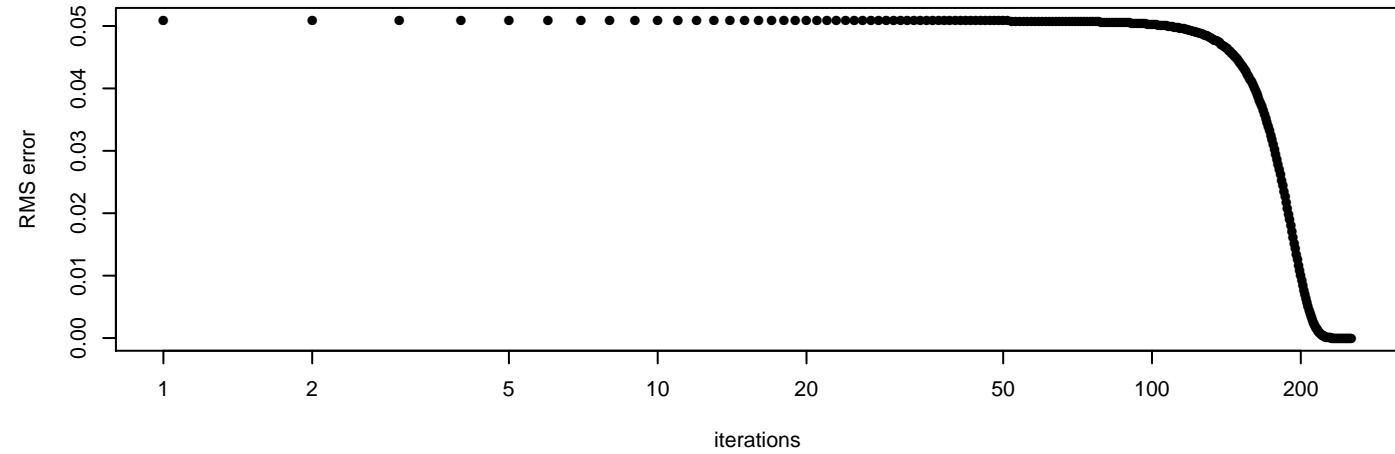


Negative Perturbation

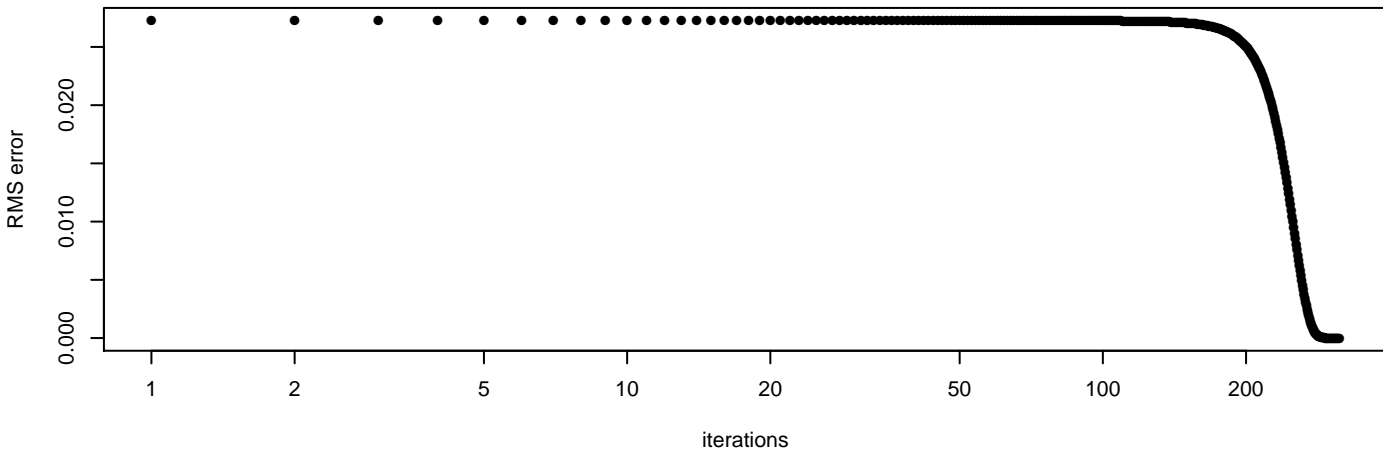


Parameter14

Positive Perturbation

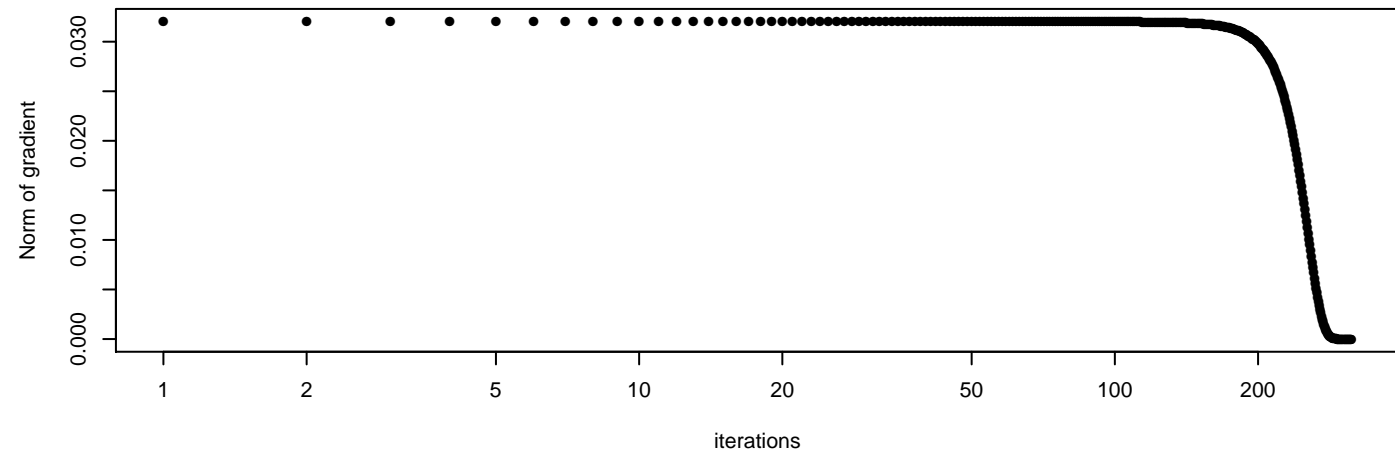
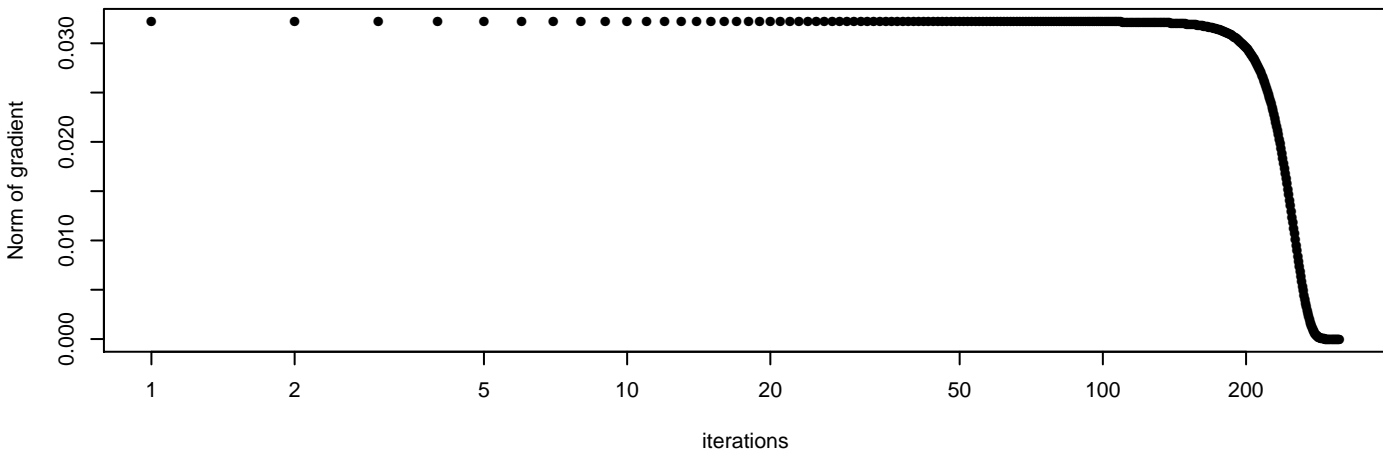
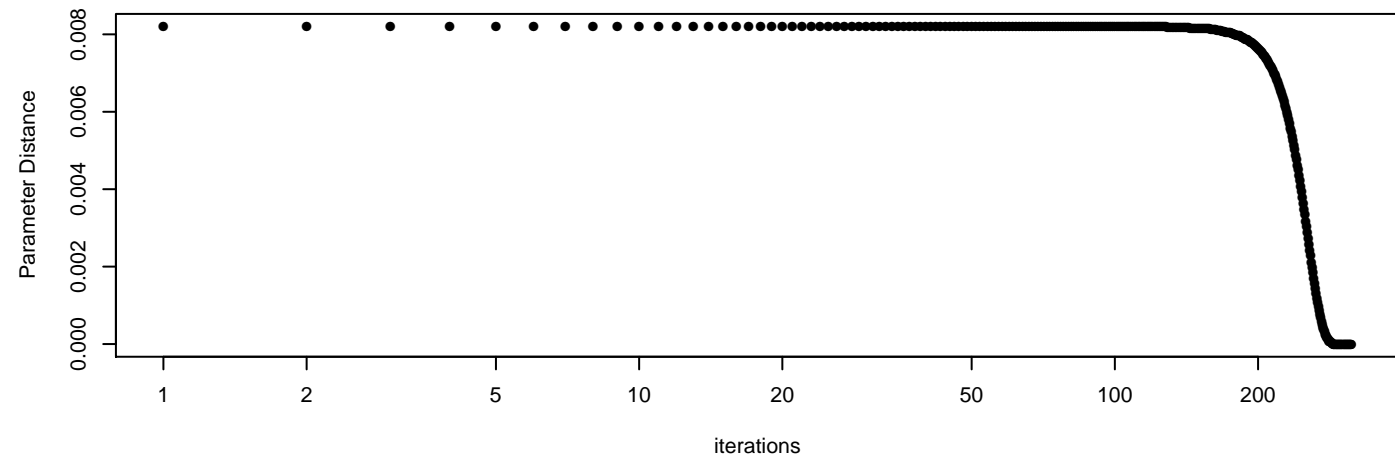
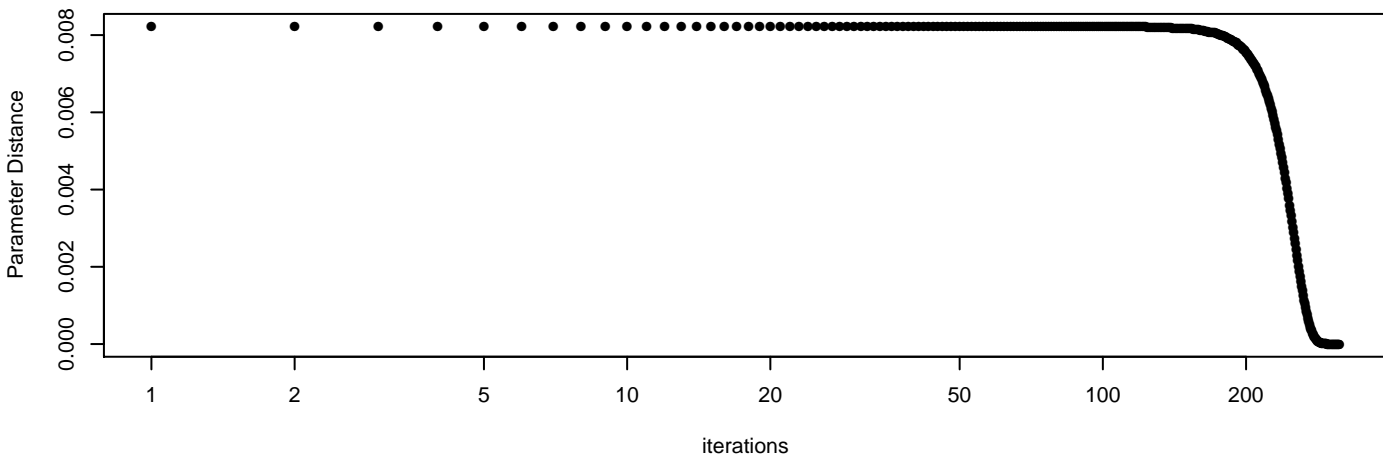
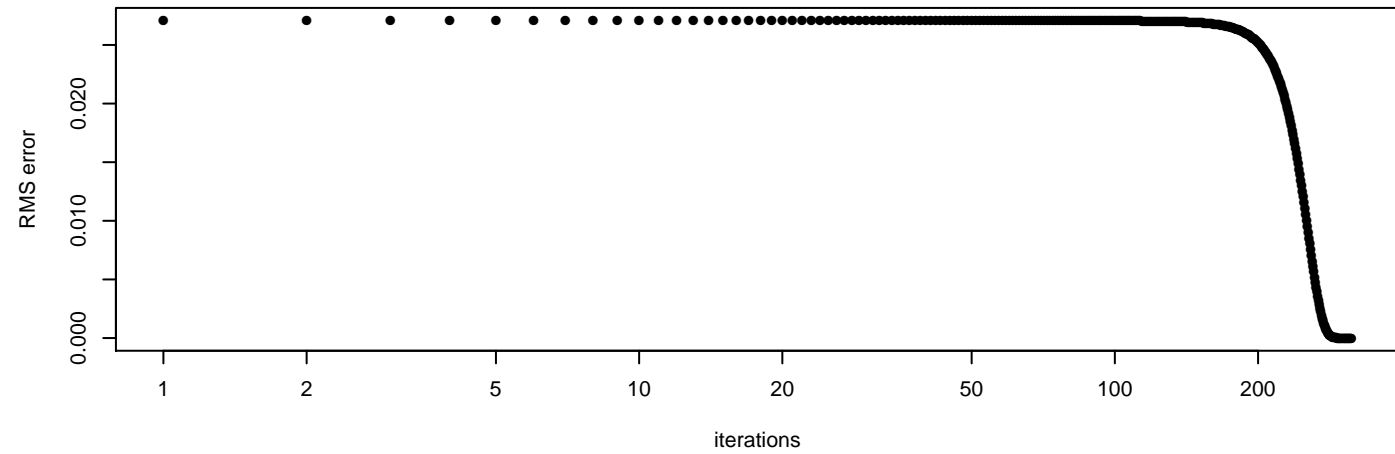


Negative Perturbation

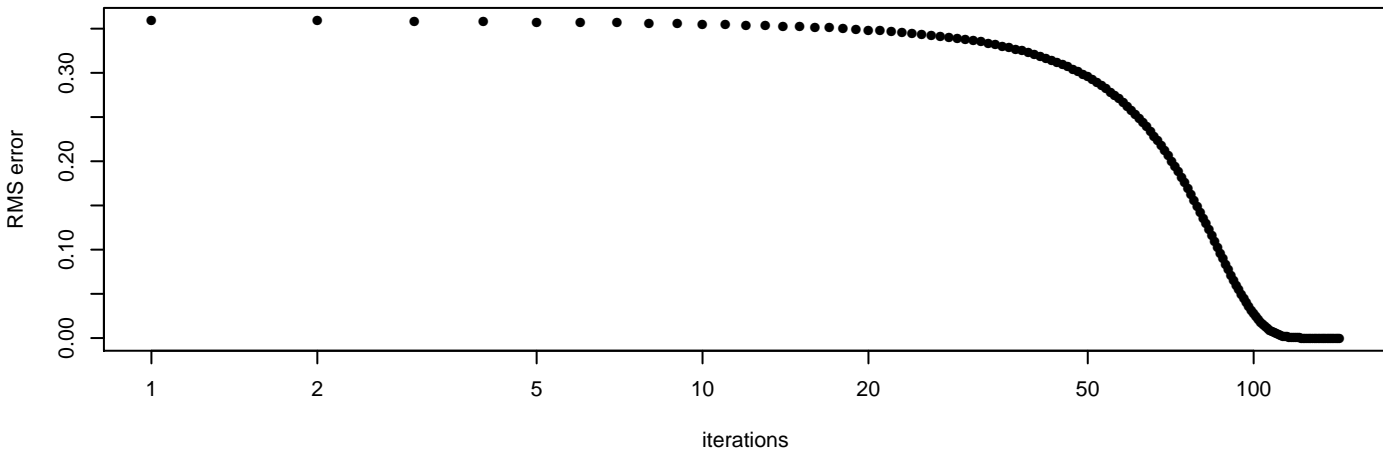


Parameter15

Positive Perturbation

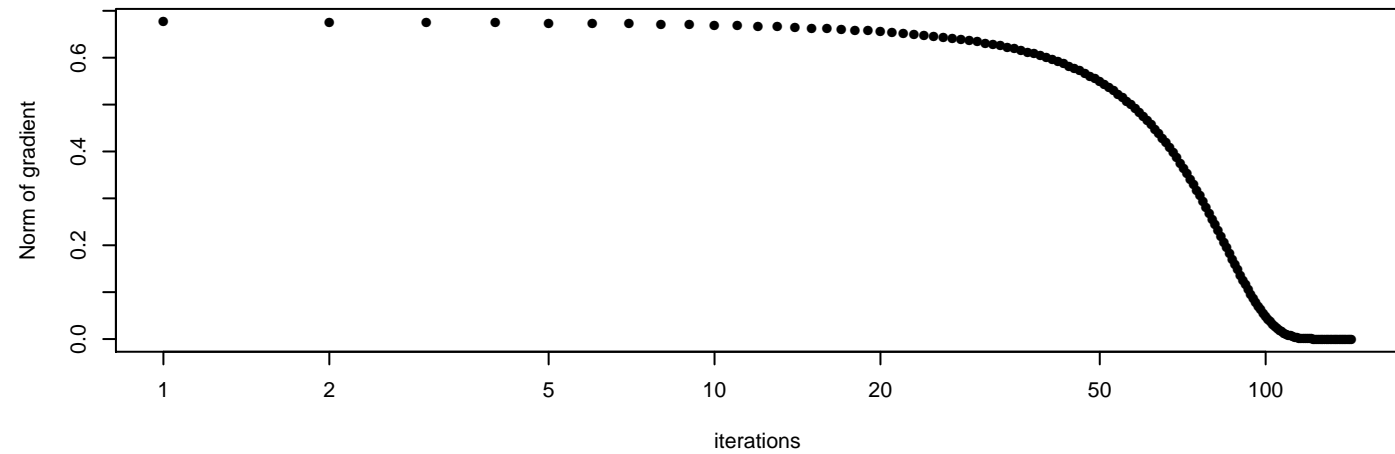
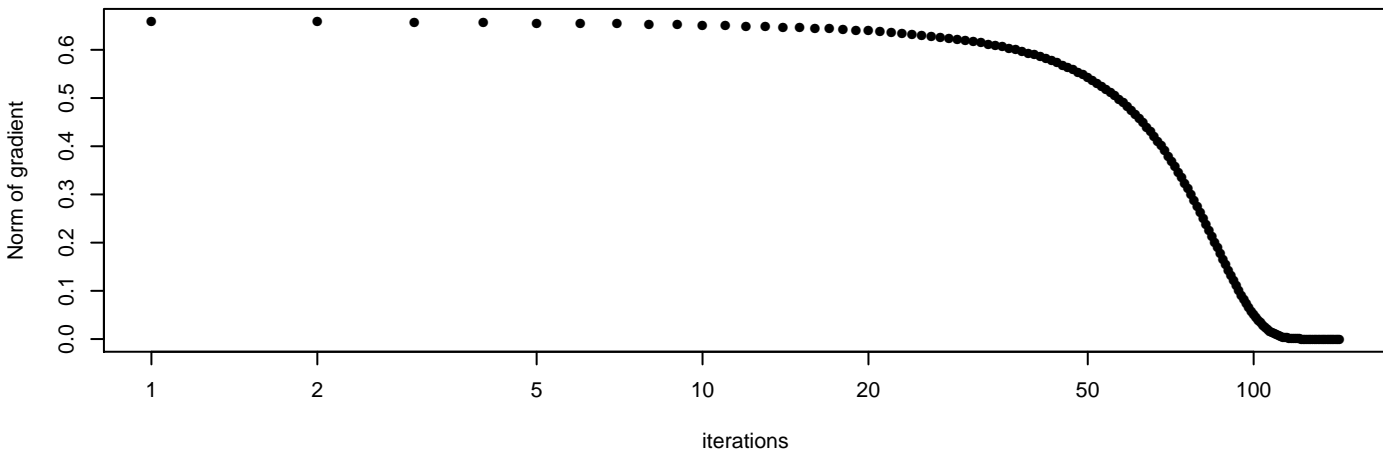
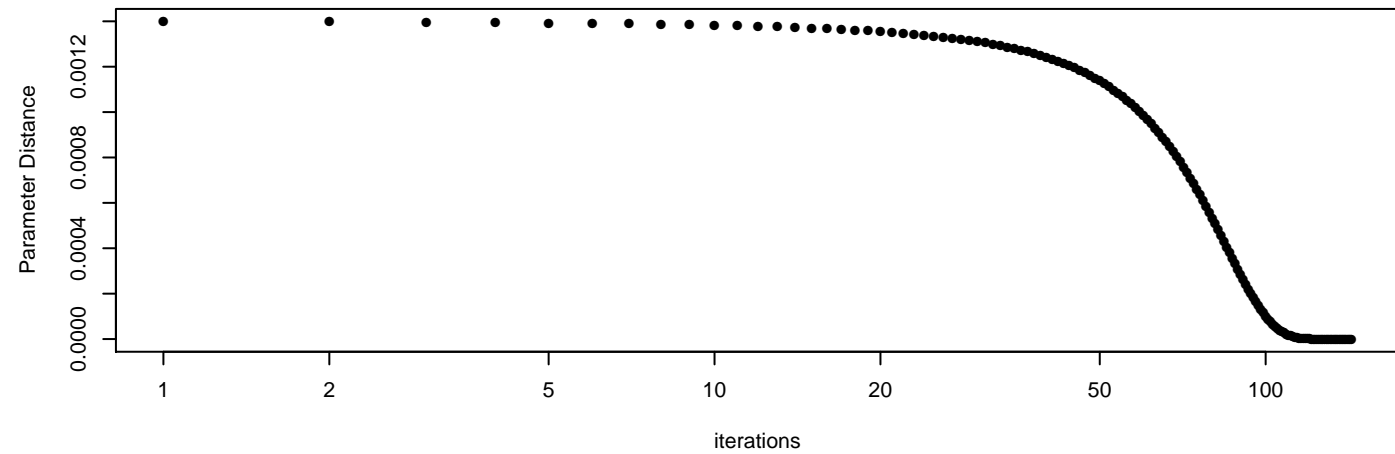
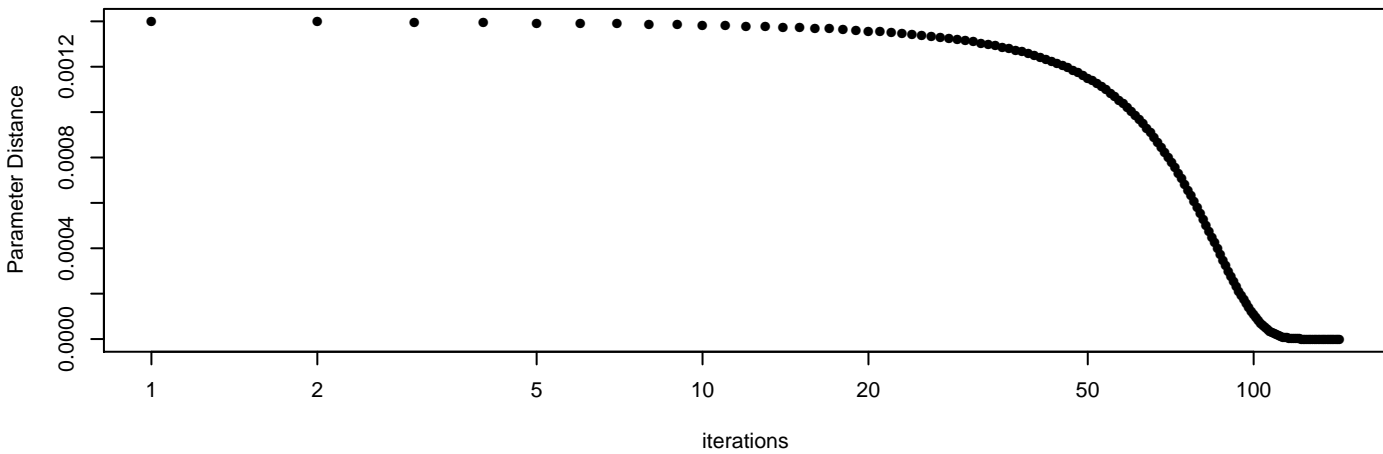
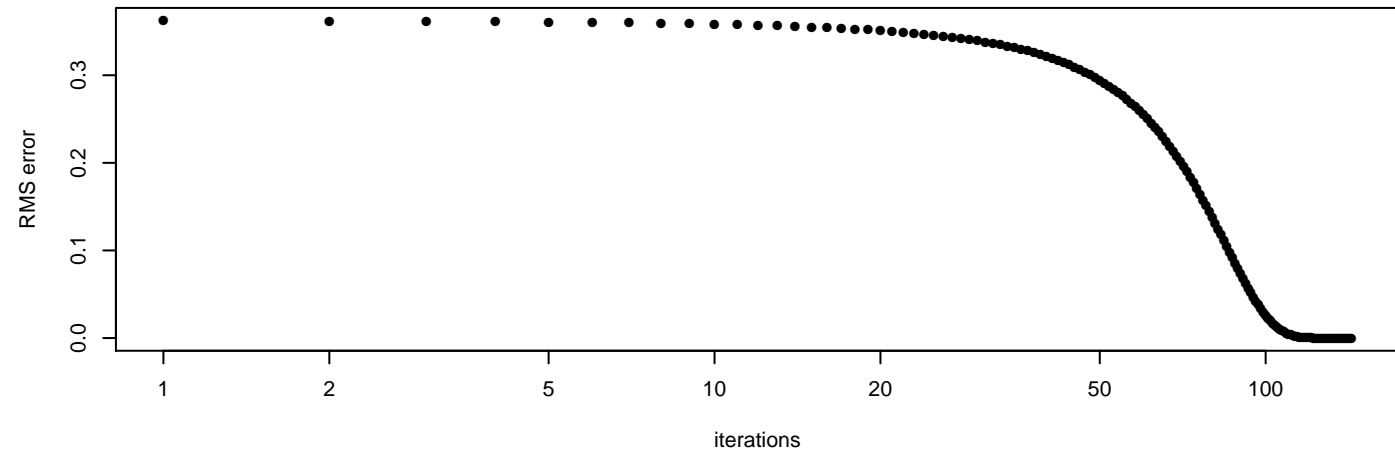


Negative Perturbation

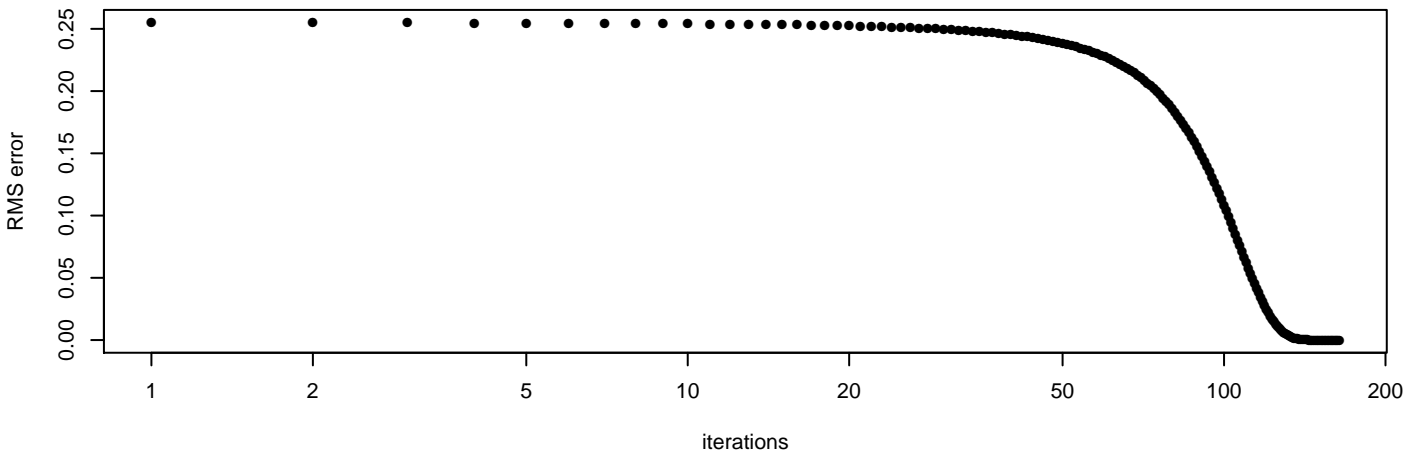


Parameter2

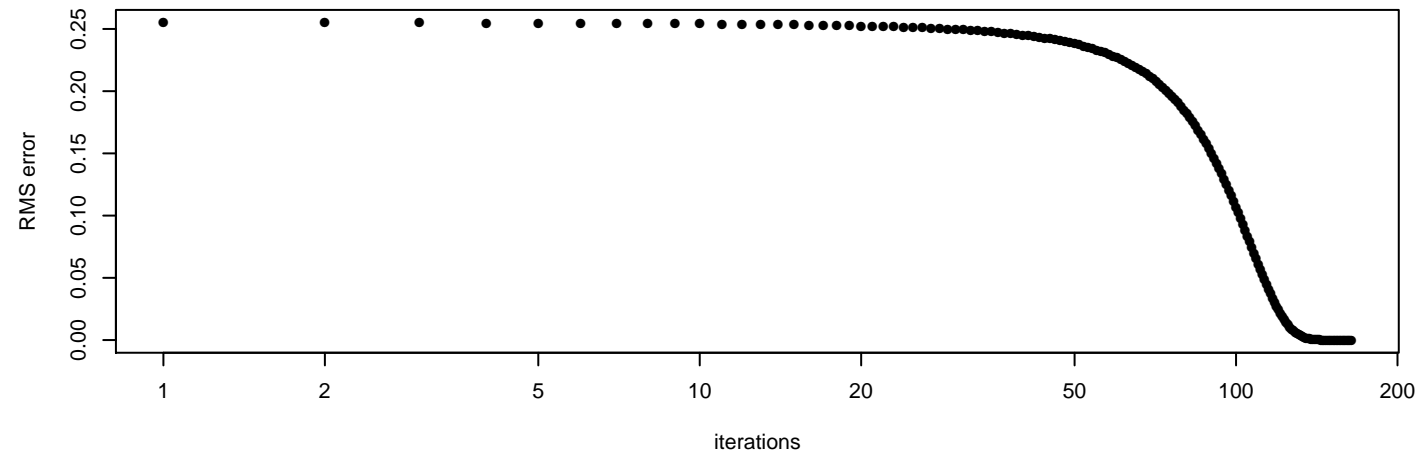
Positive Perturbation



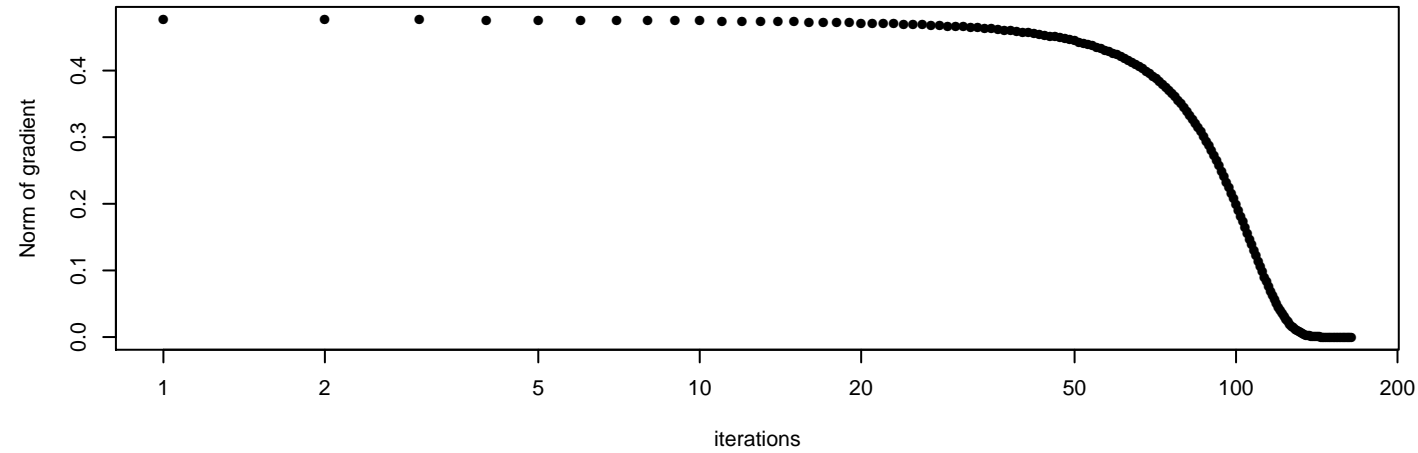
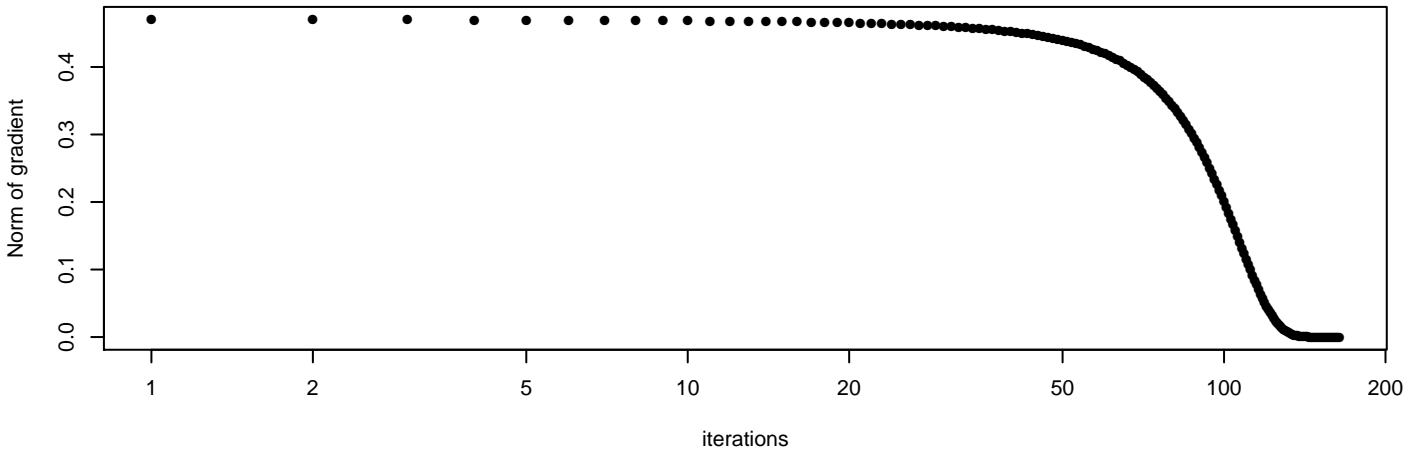
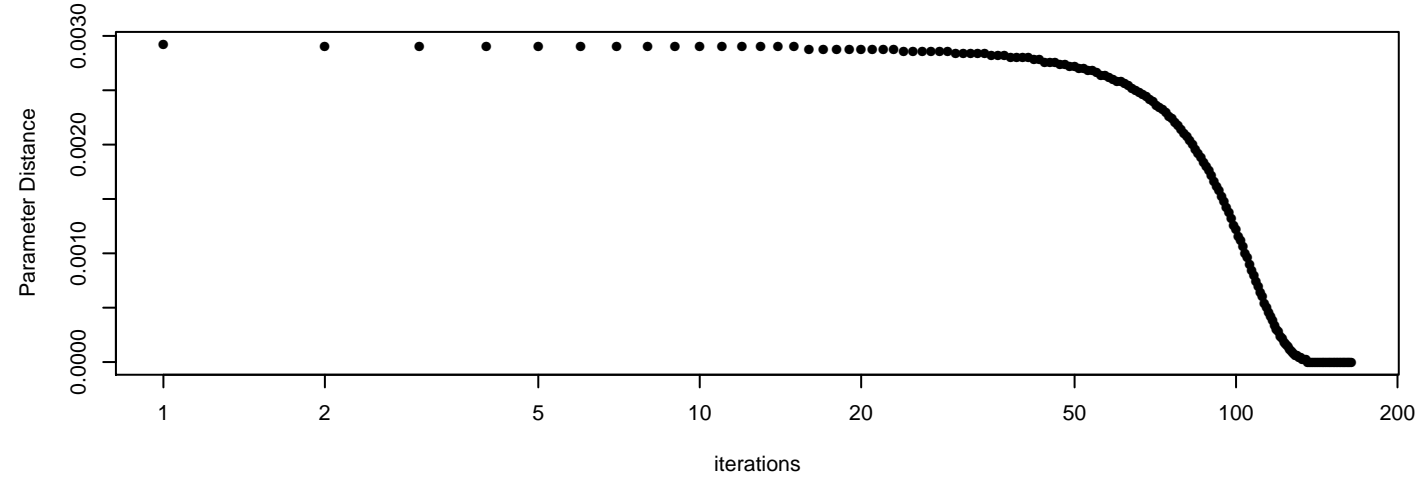
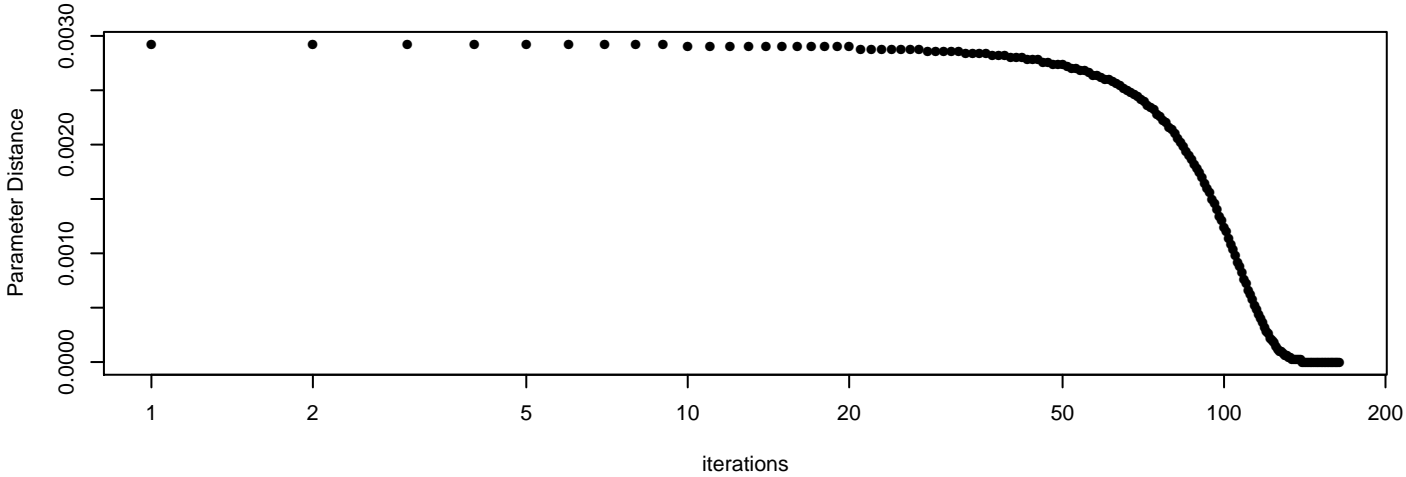
Negative Perturbation



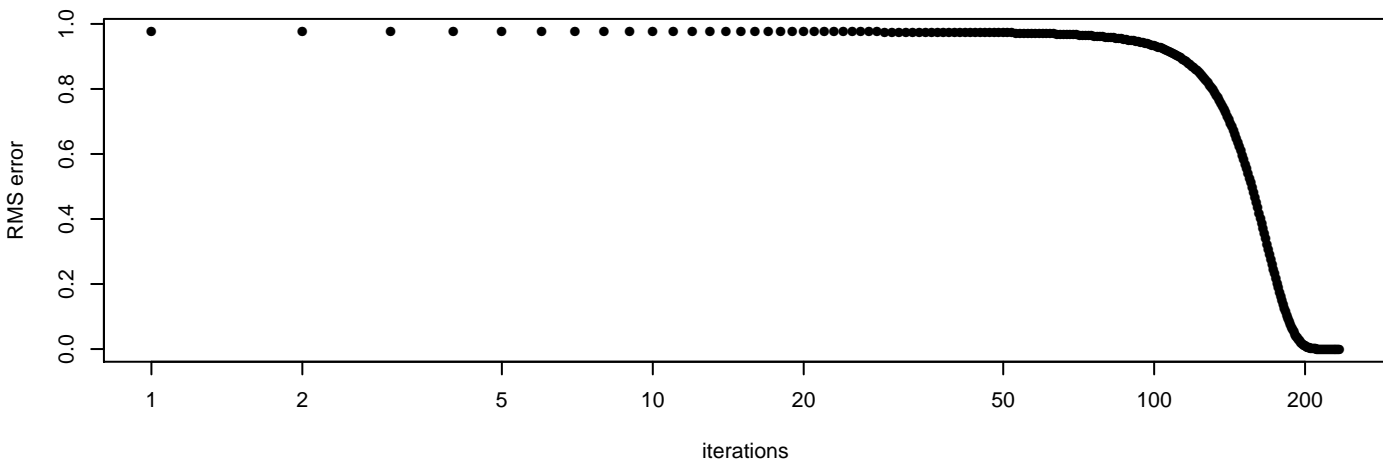
Parameter3



Positive Perturbation

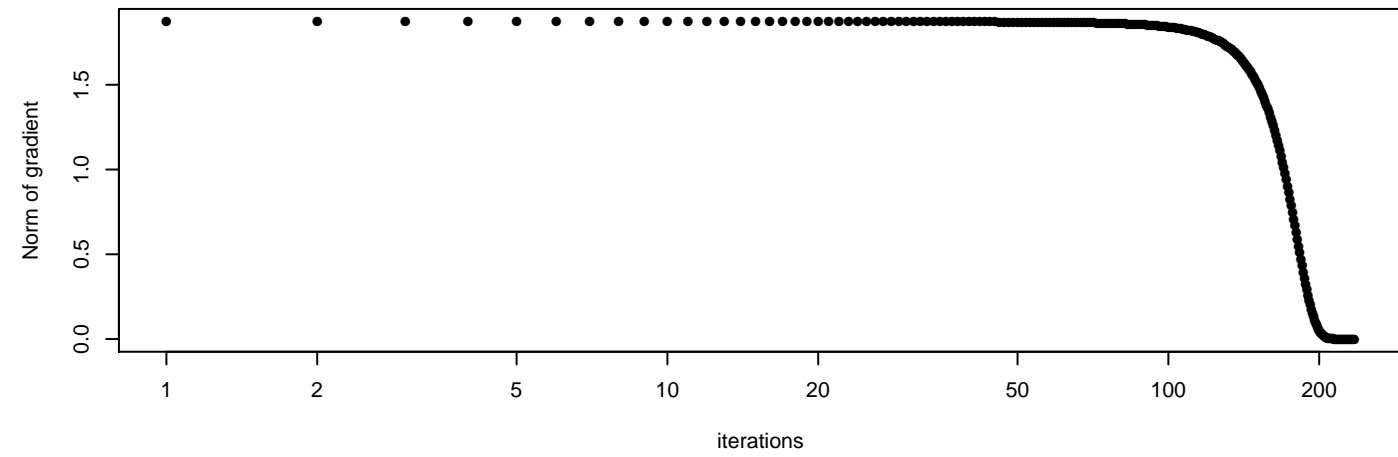
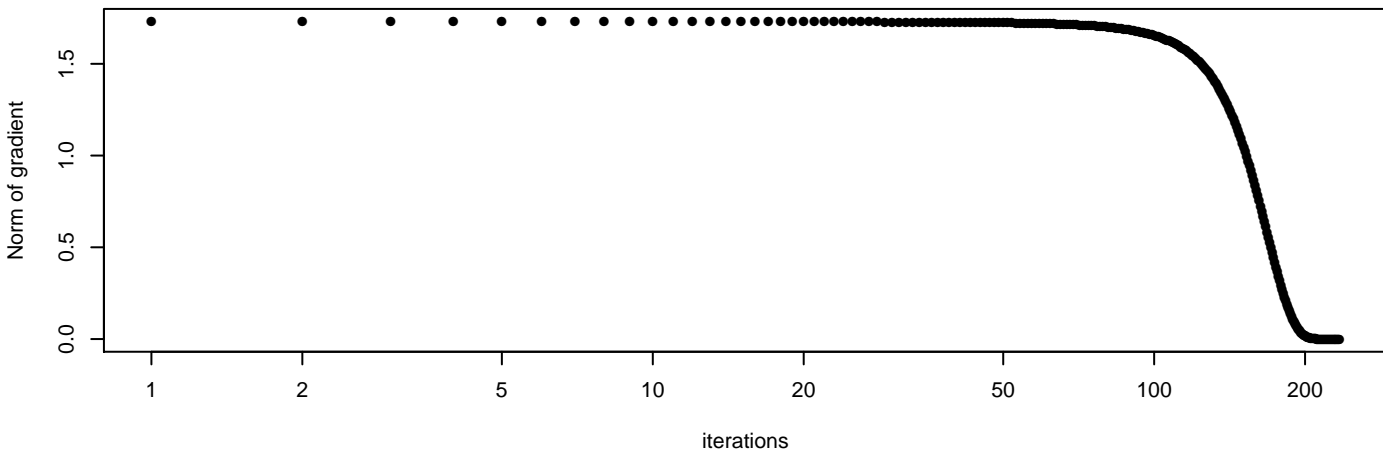
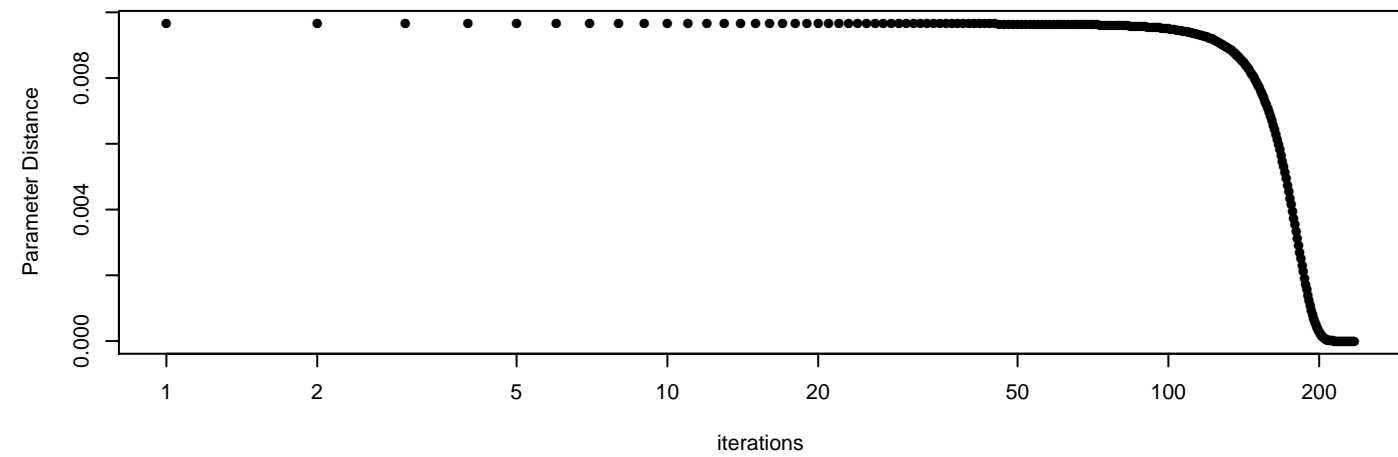
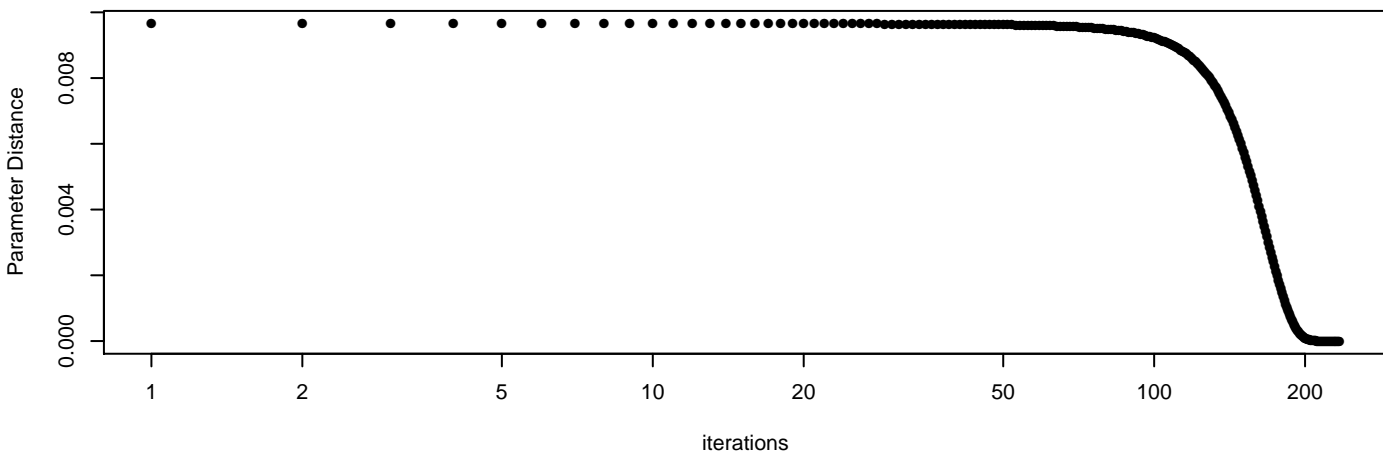
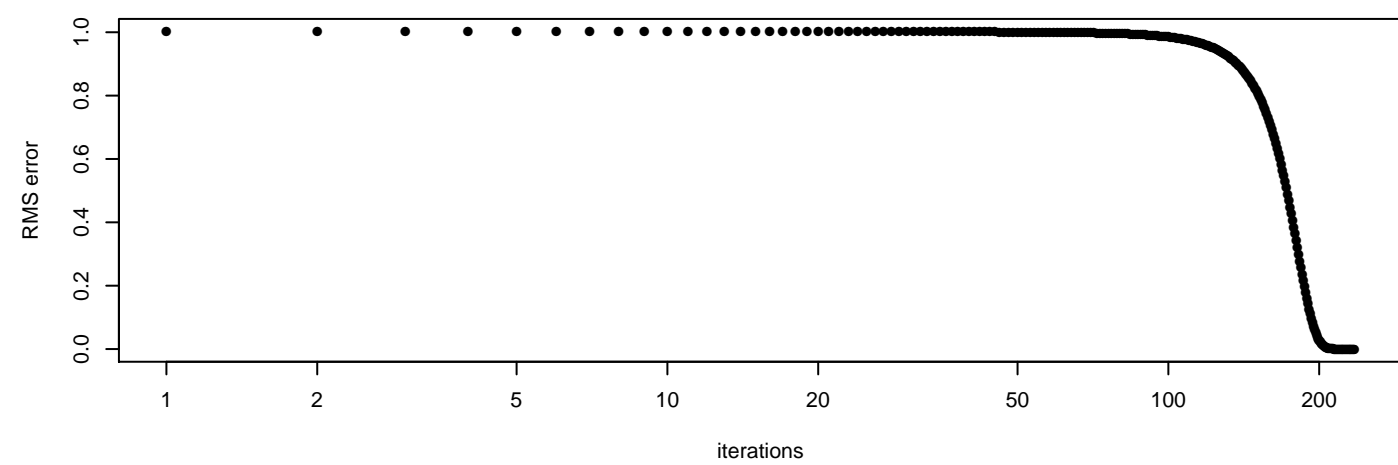


Negative Perturbation

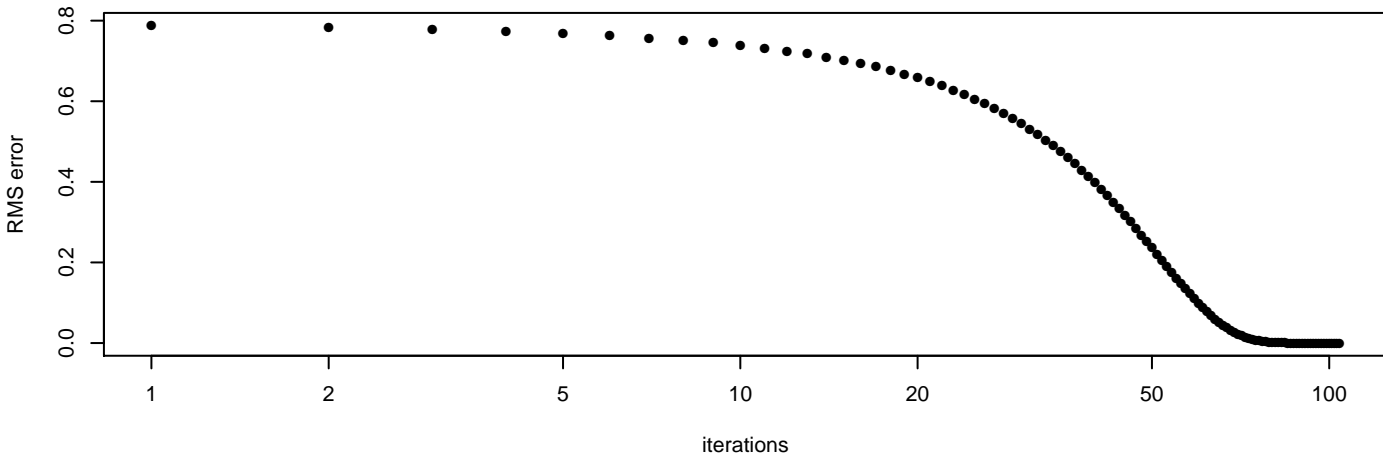


Parameter4

Positive Perturbation

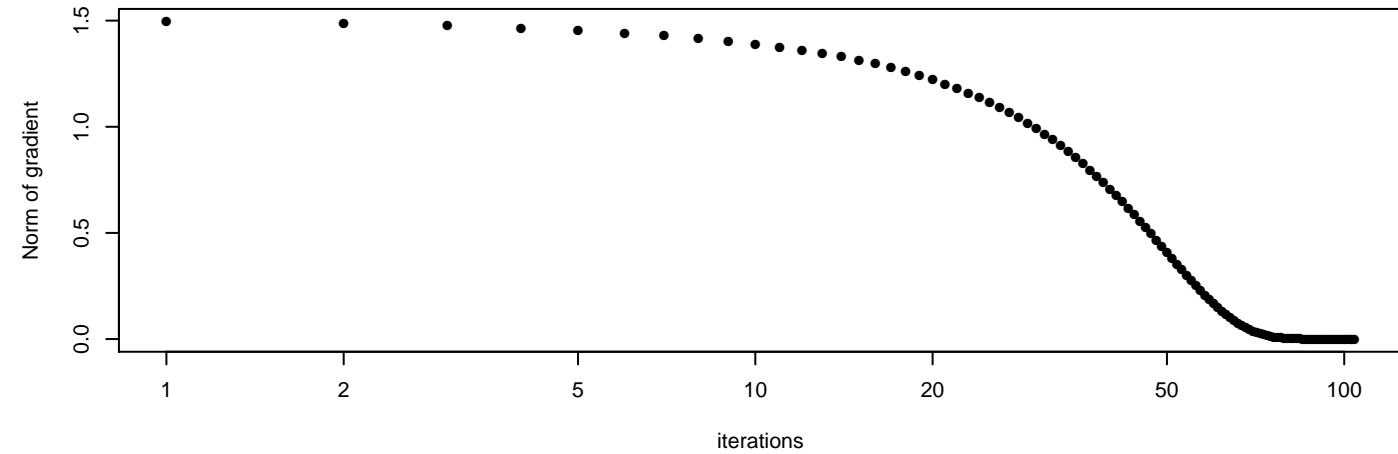
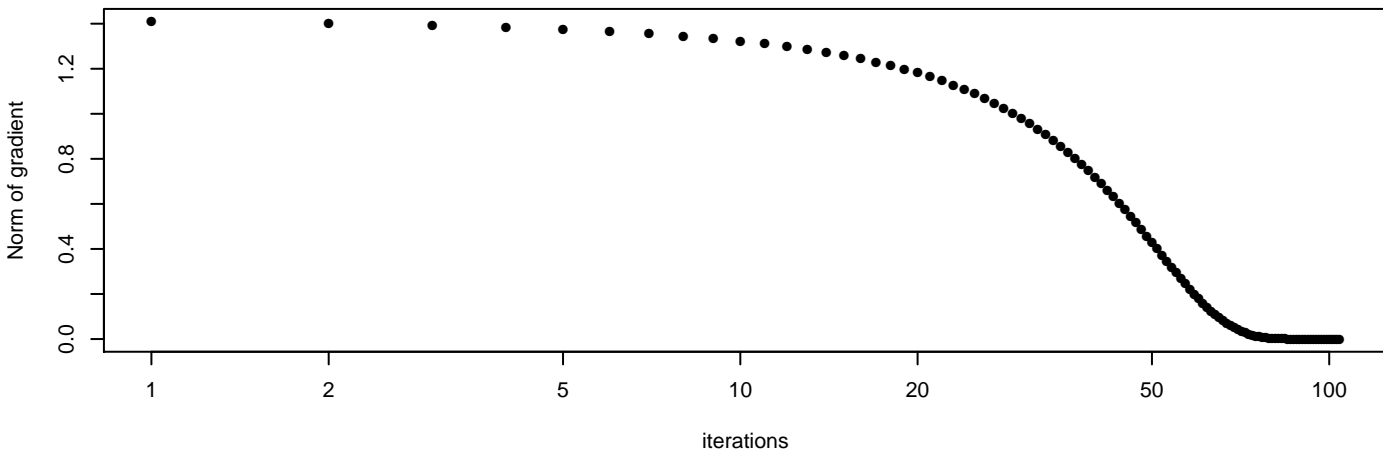
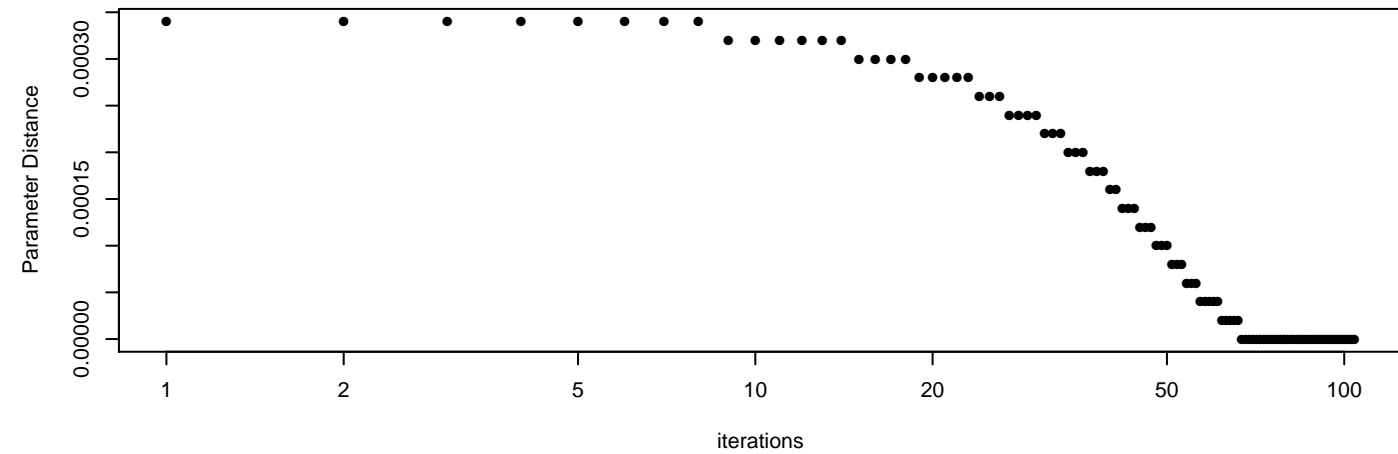
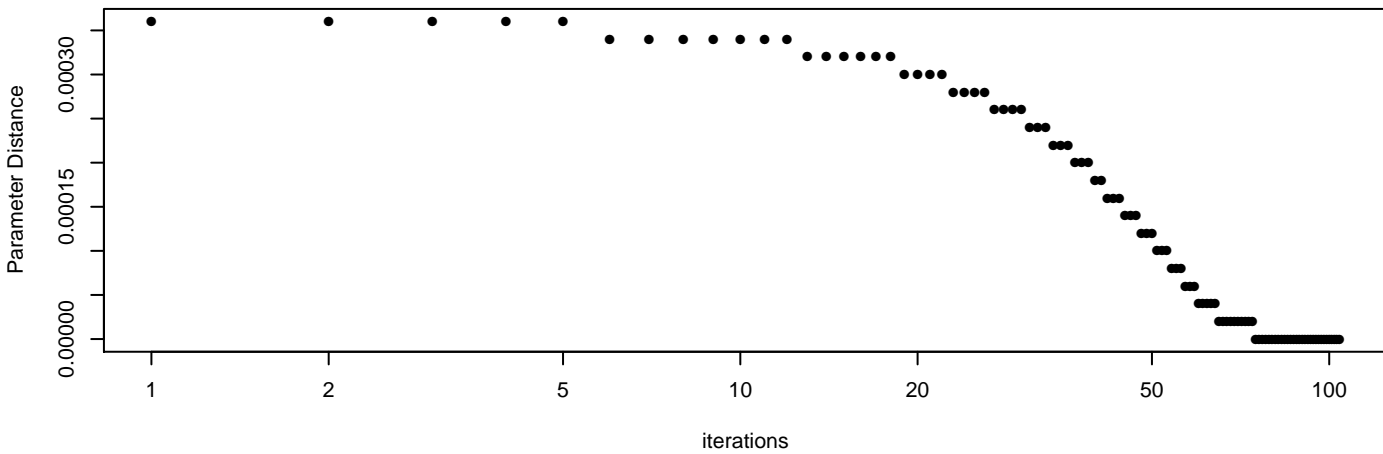
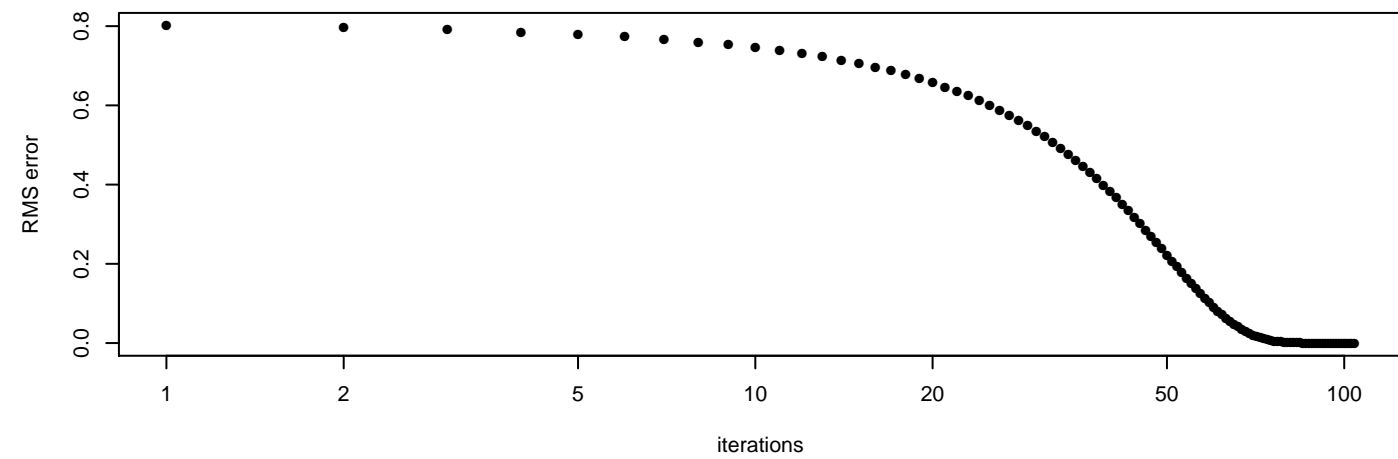


Negative Perturbation

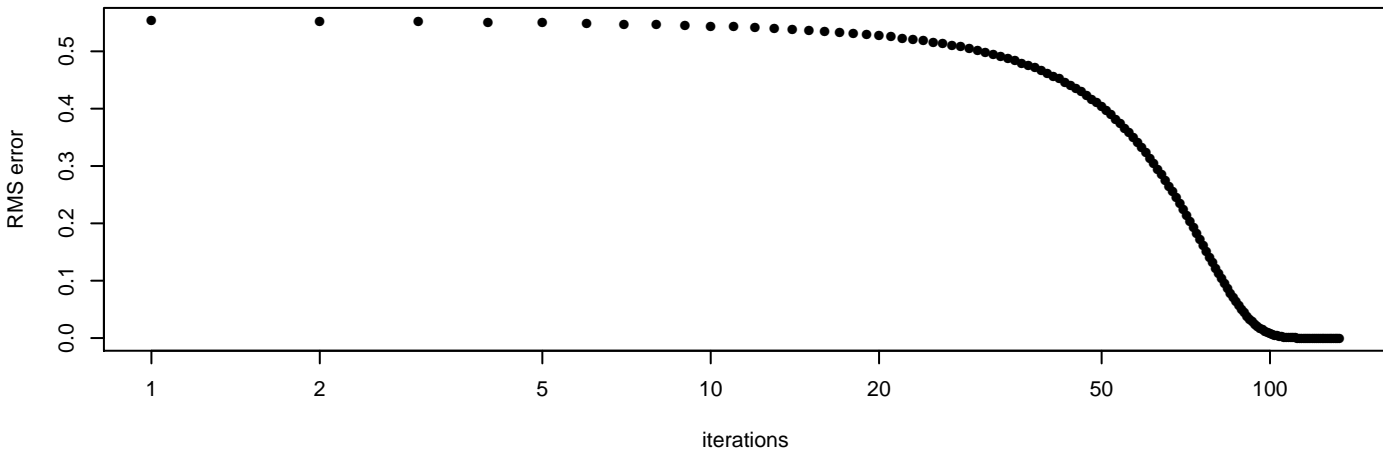


Parameter5

Positive Perturbation

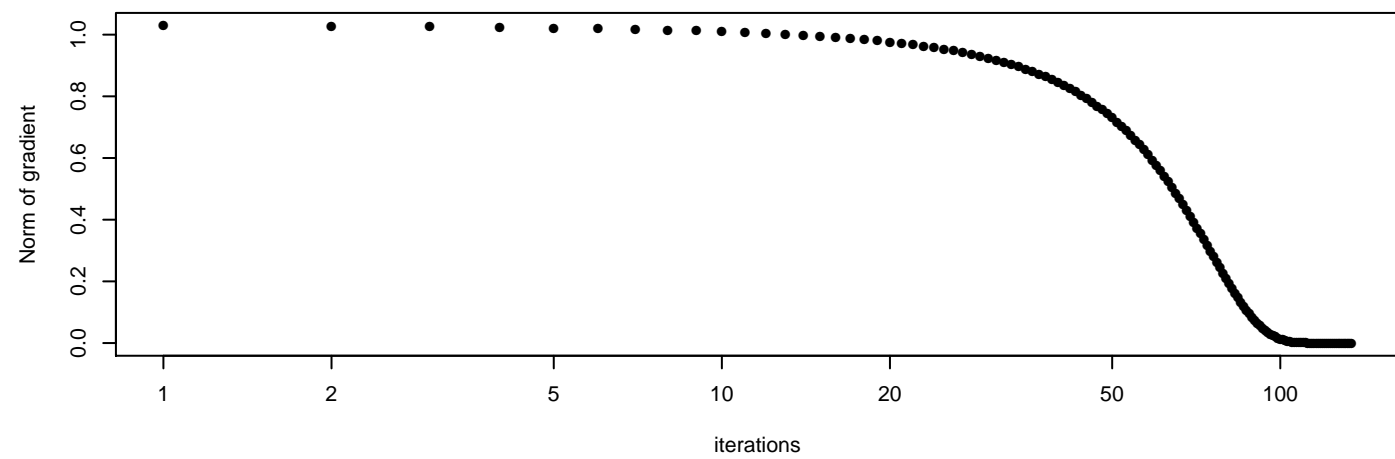
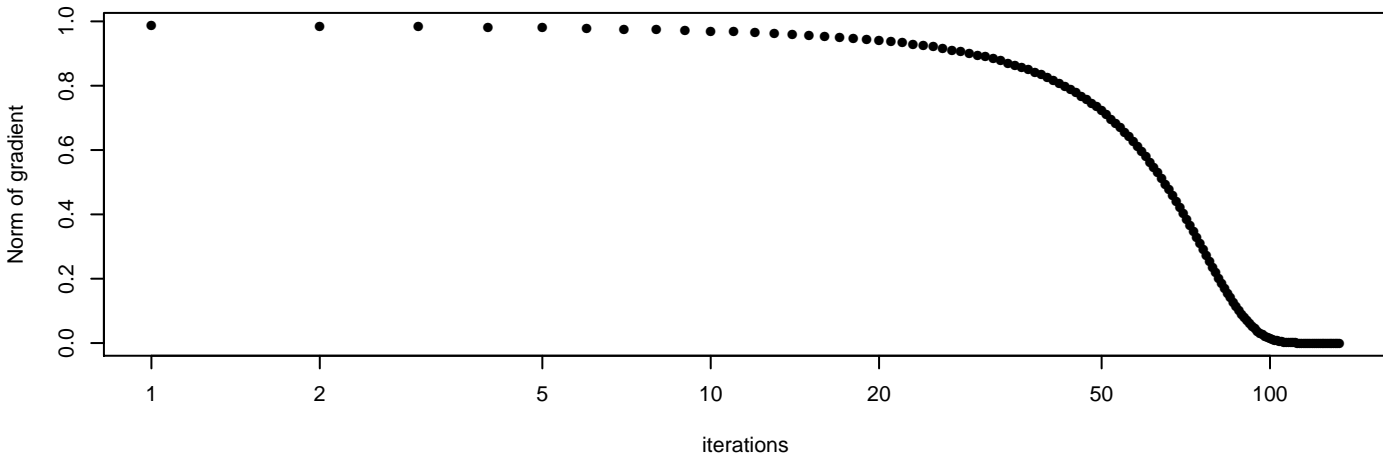
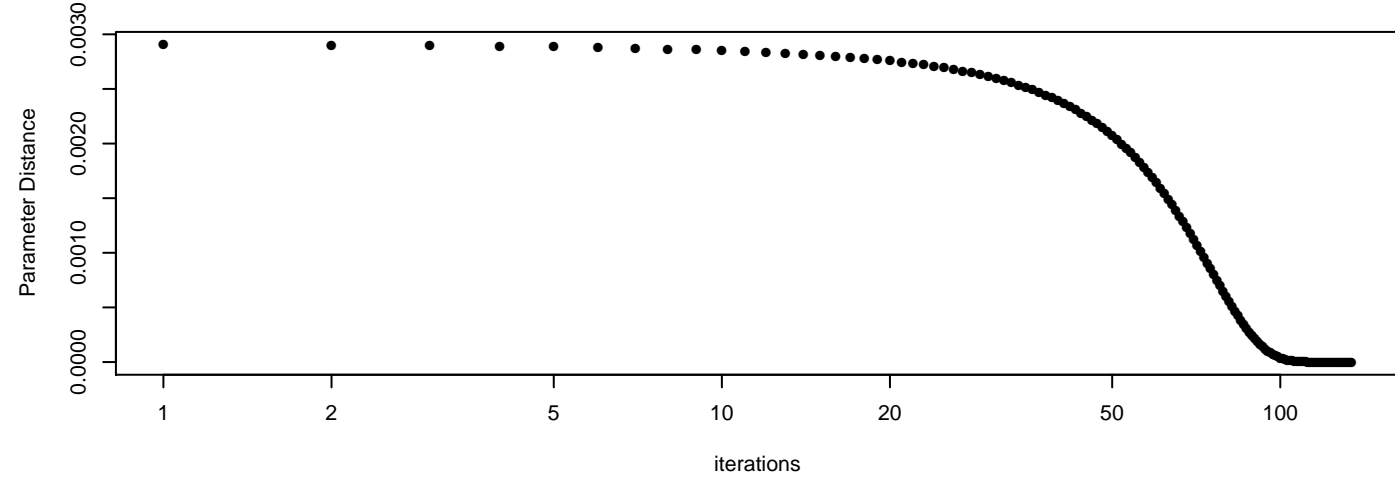
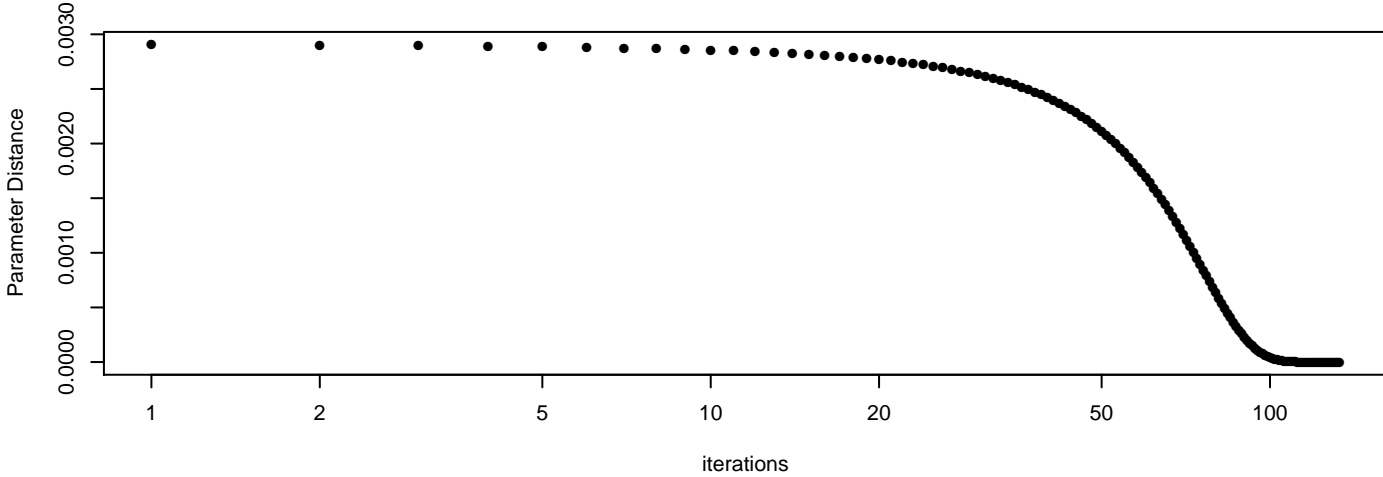
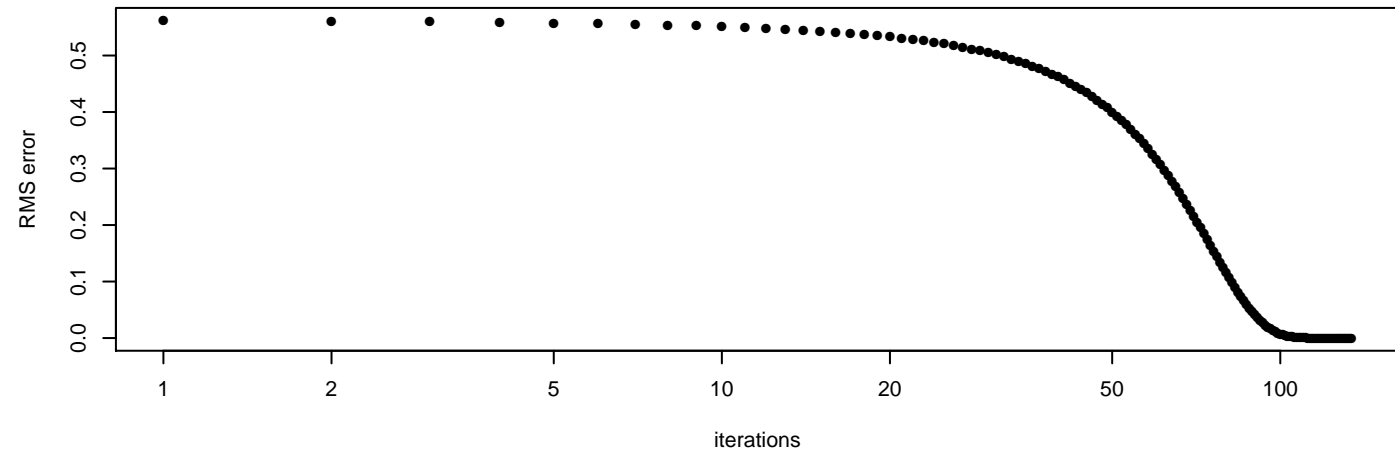


Negative Perturbation

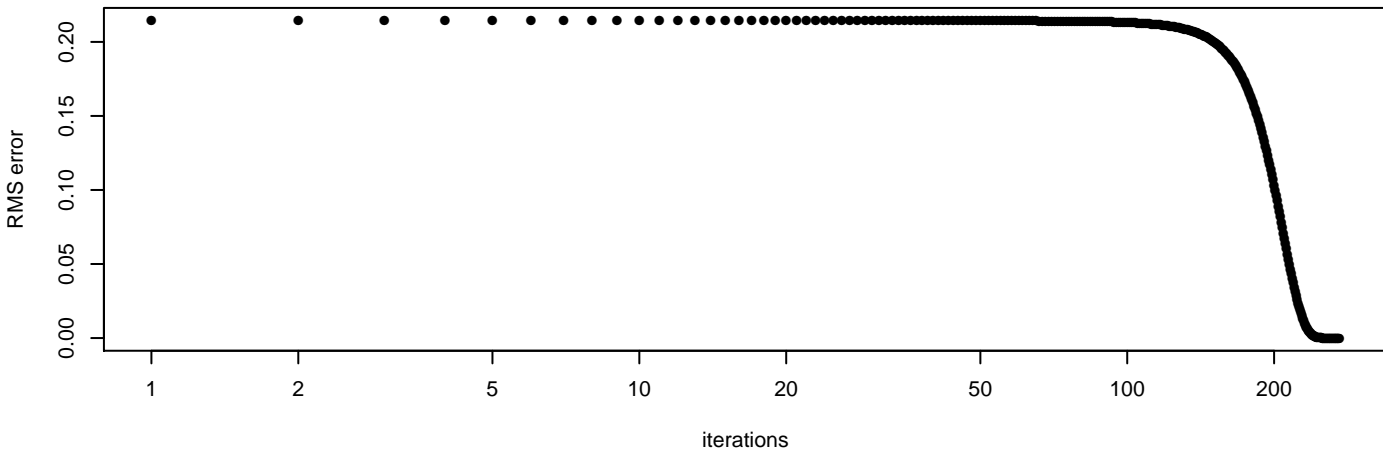


Parameter6

Positive Perturbation

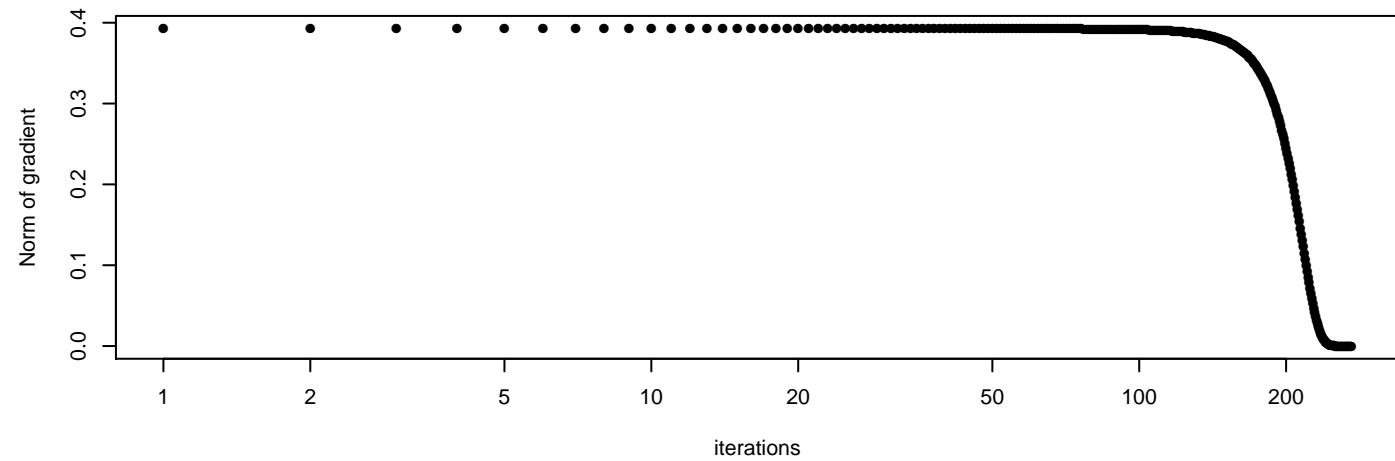
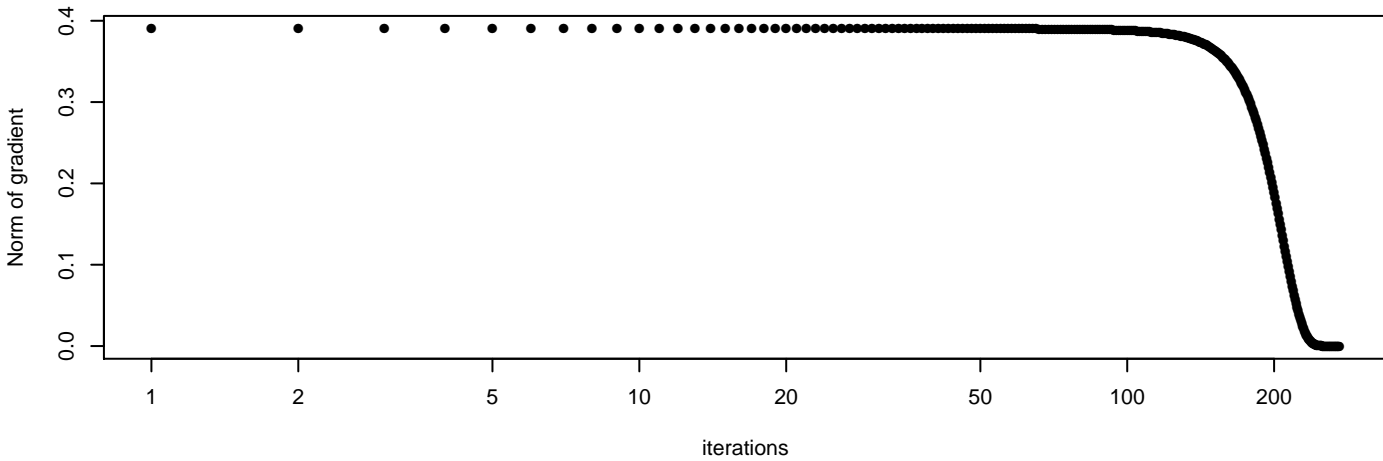
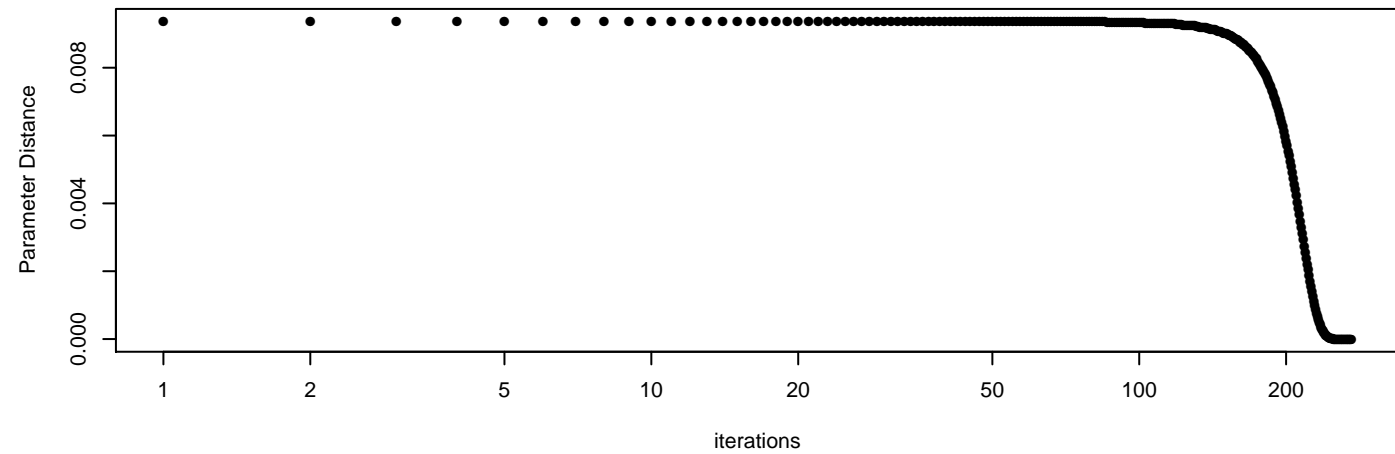
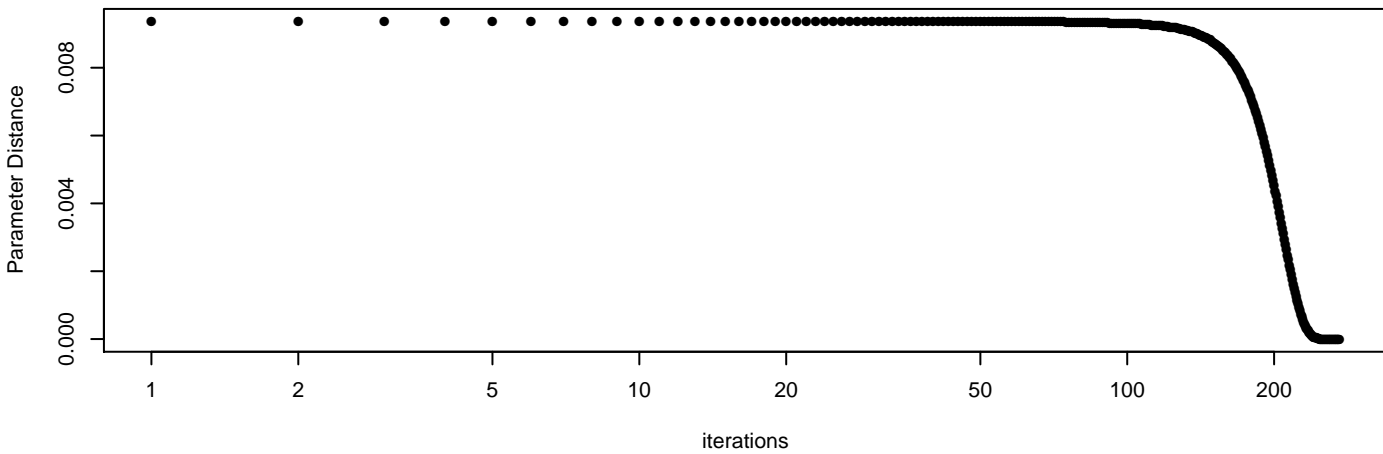
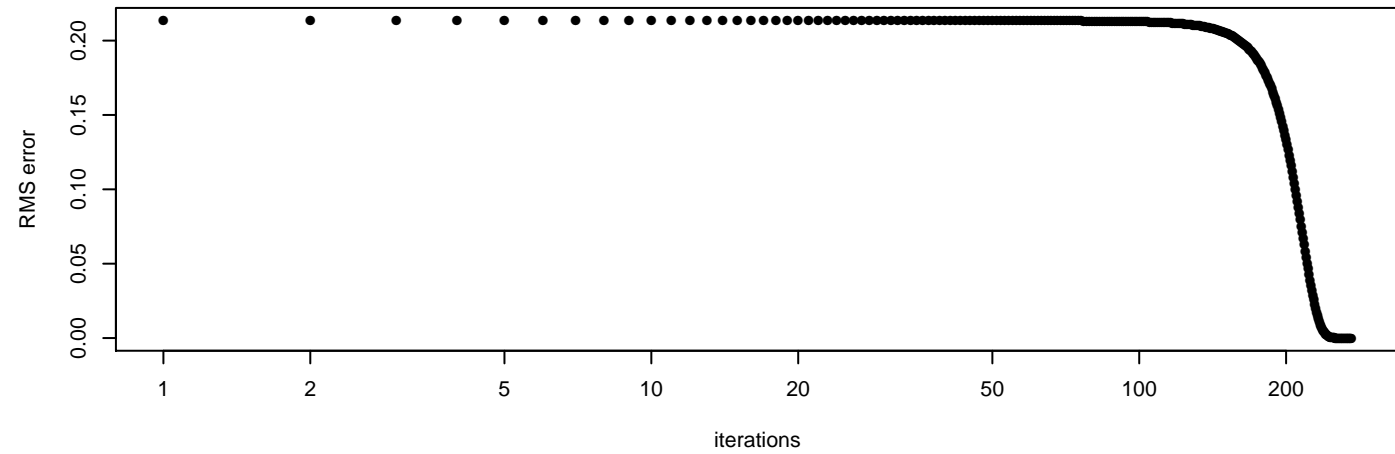


Negative Perturbation

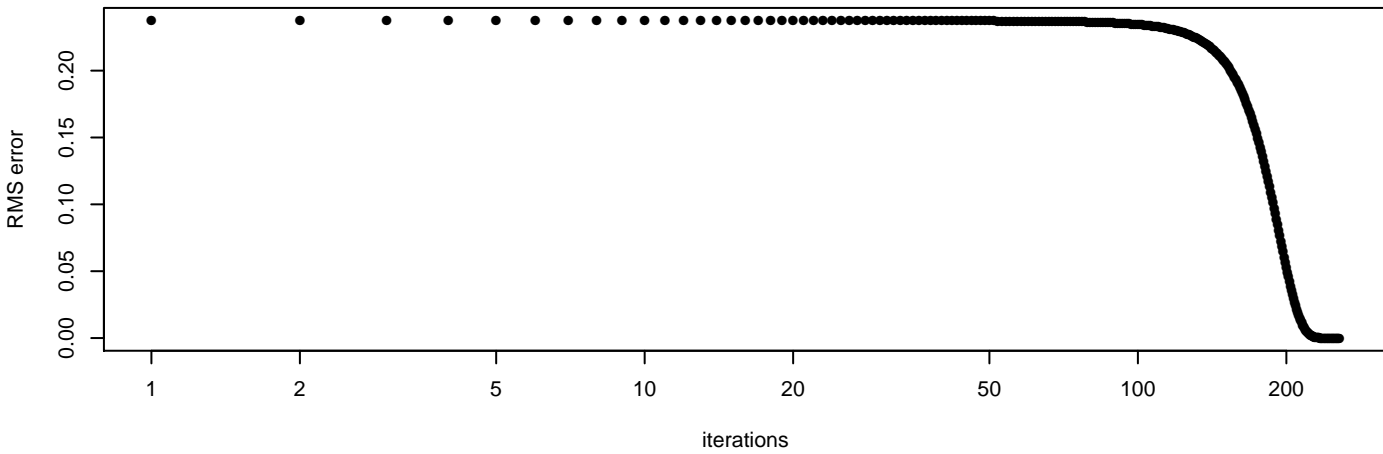


Parameter7

Positive Perturbation

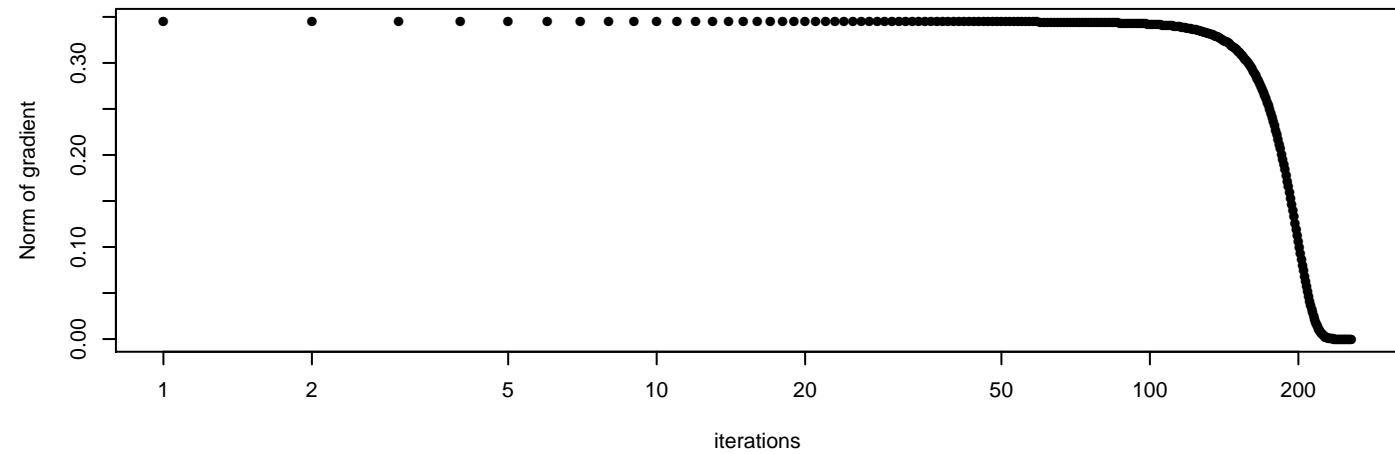
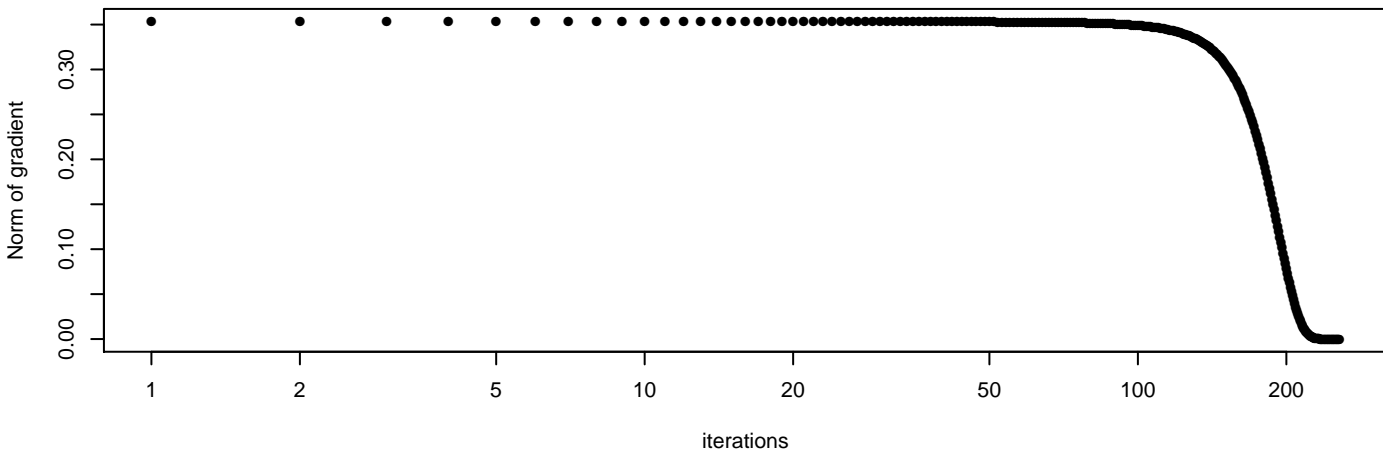
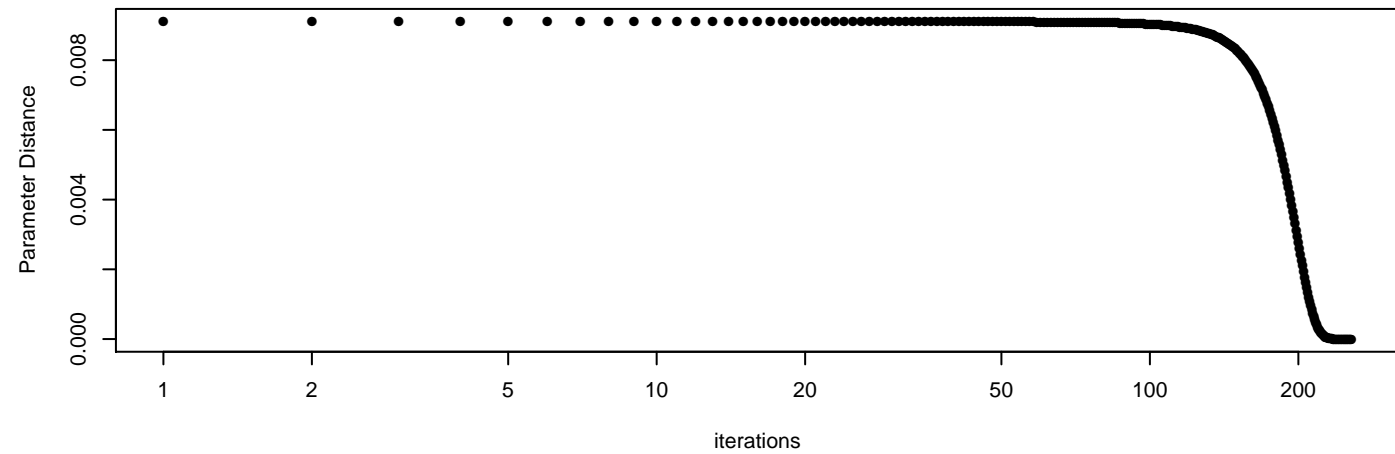
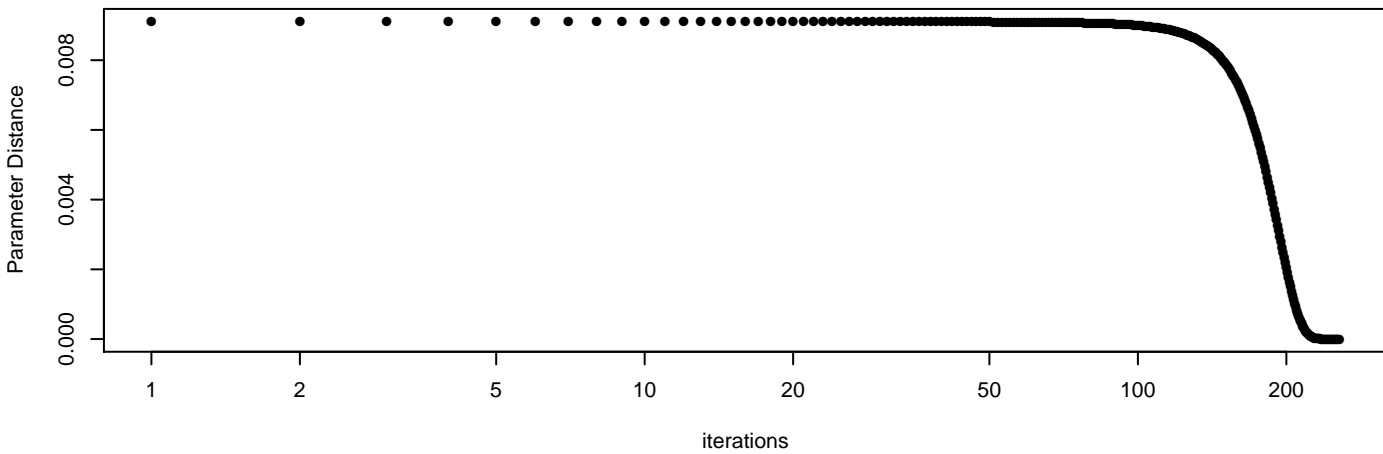
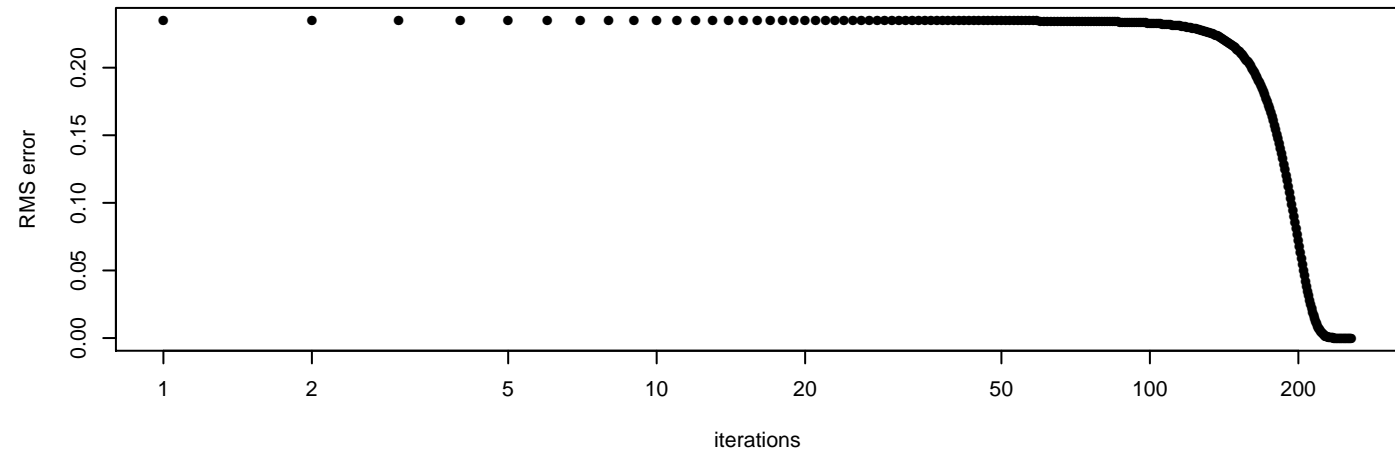


Negative Perturbation

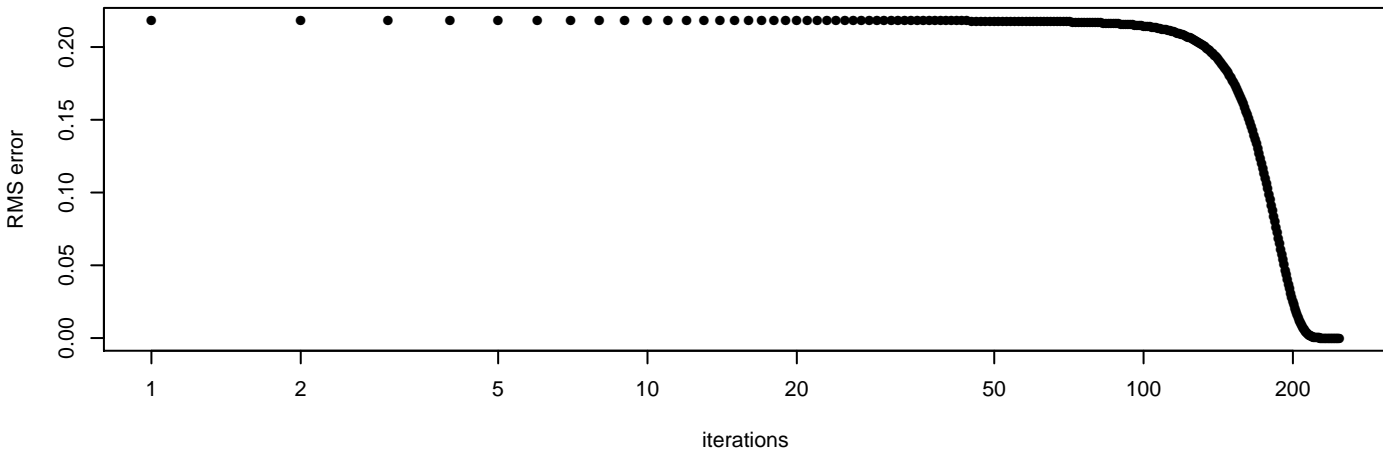


Parameter8

Positive Perturbation

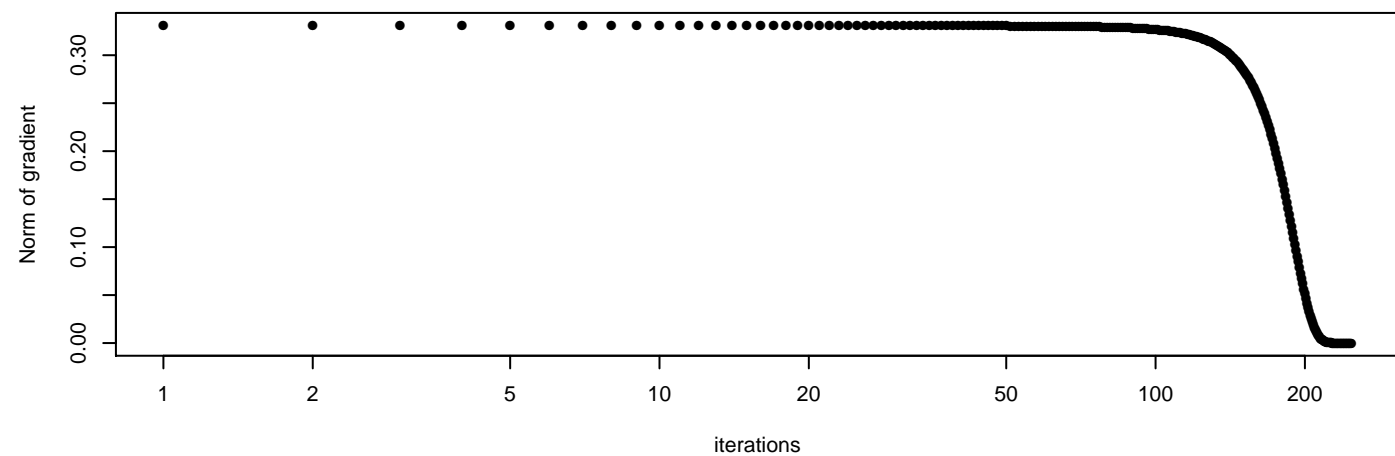
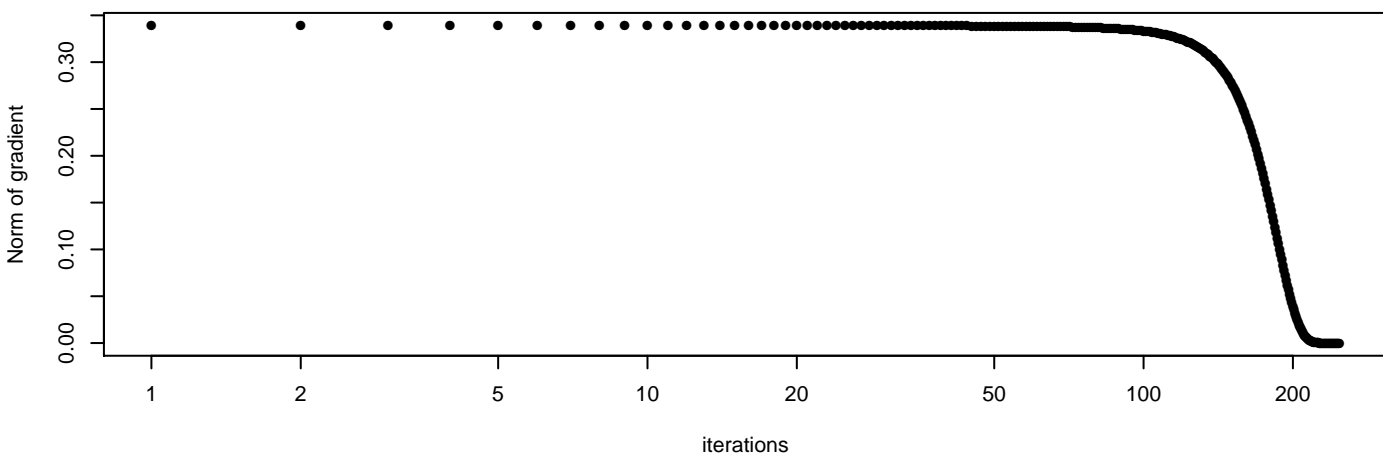
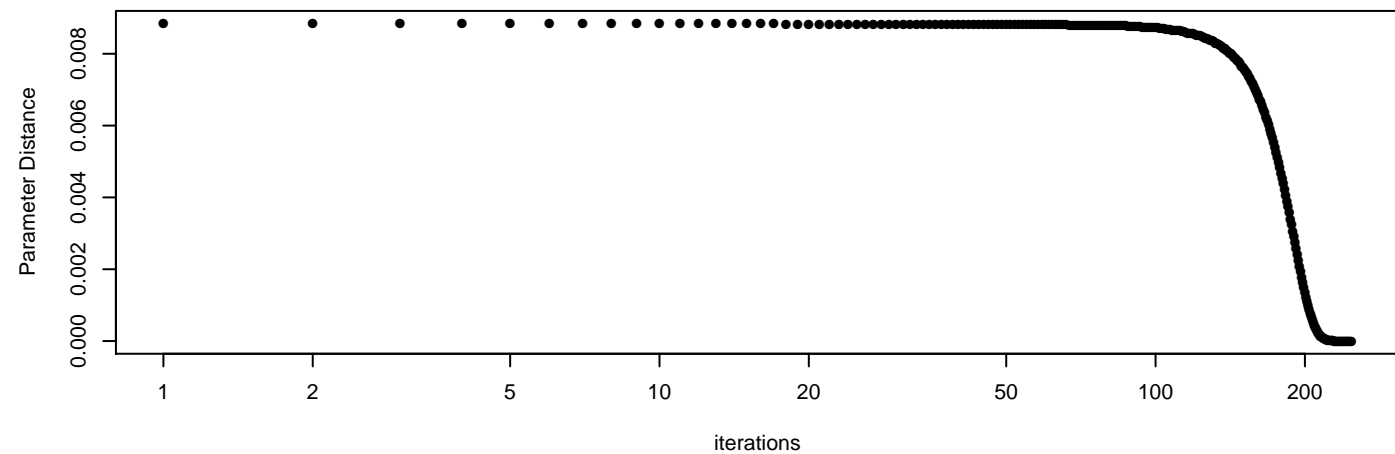
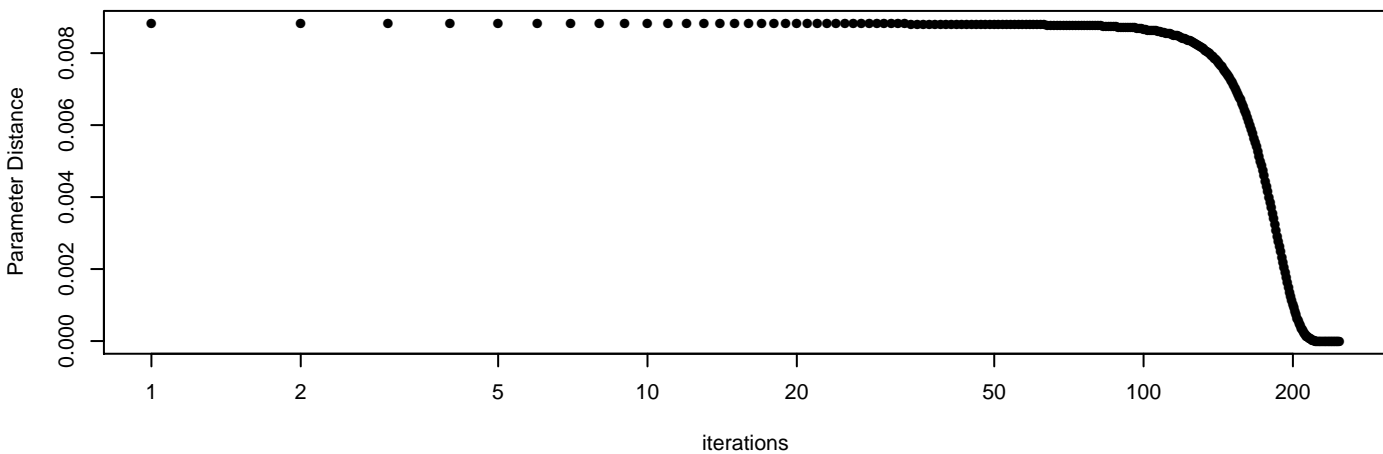
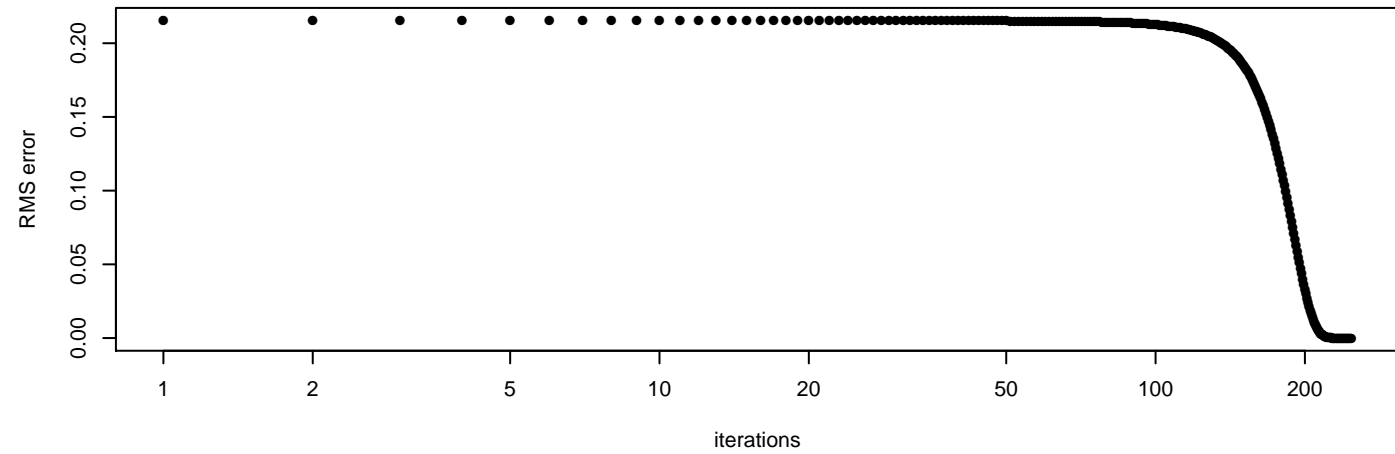


Negative Perturbation



Parameter9

Positive Perturbation

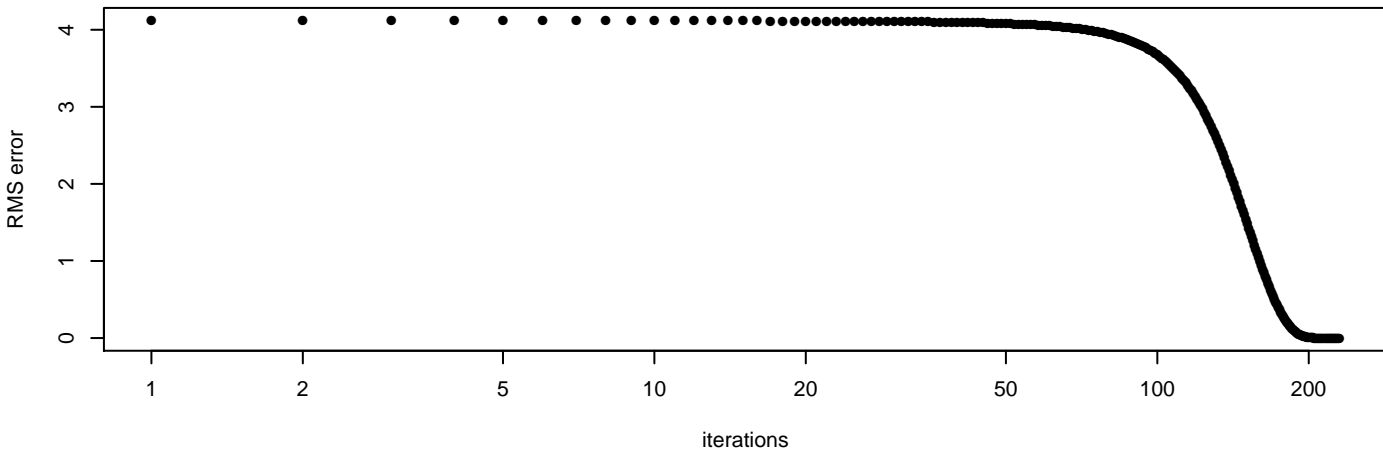


Convergence of a parameter set where each parameter is perturbed by * 10% * individually.

Gradient descent is executed with all parameters * held * at their optimal value except the perturbed one.

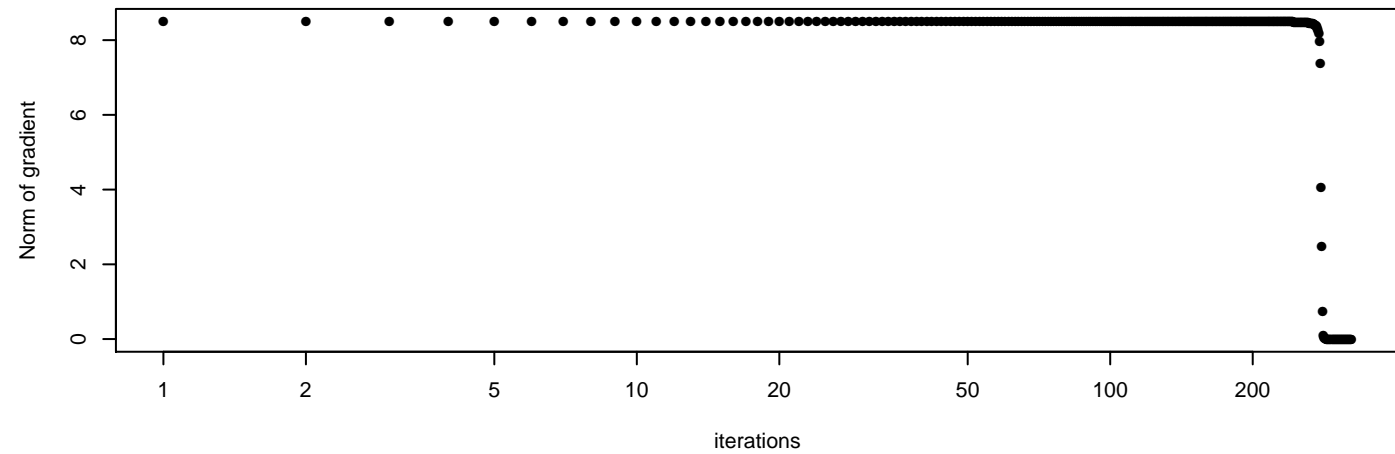
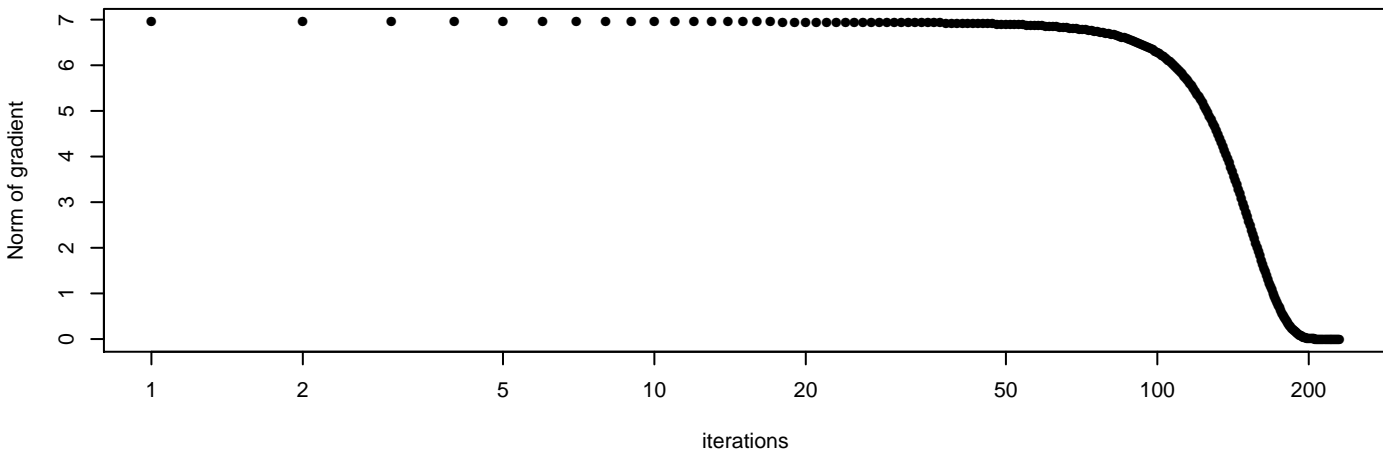
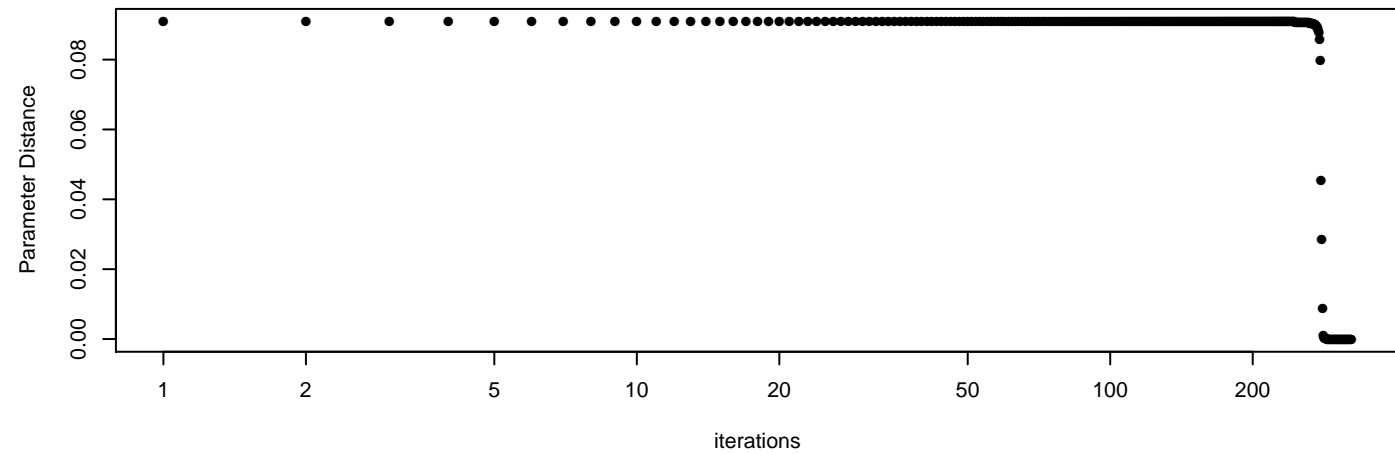
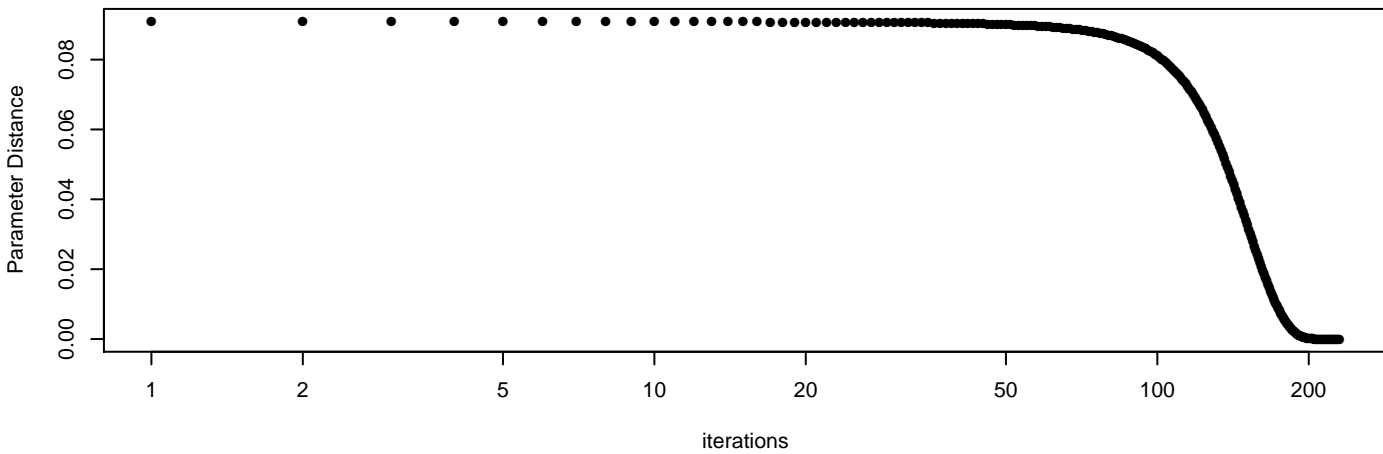
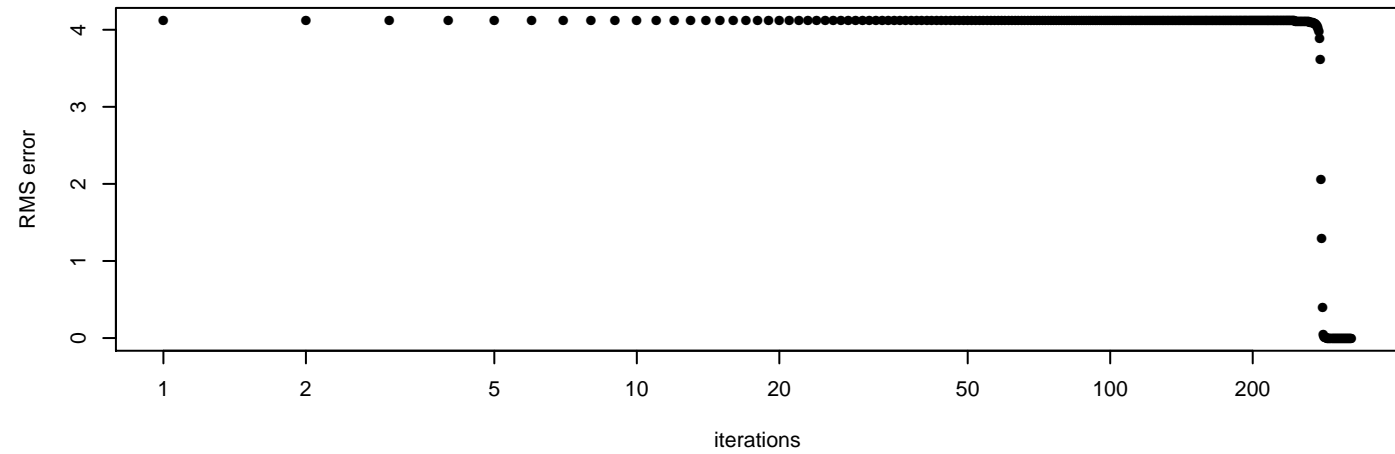
Gradient descent was ran in * normal * mode to take the value of the gradient.

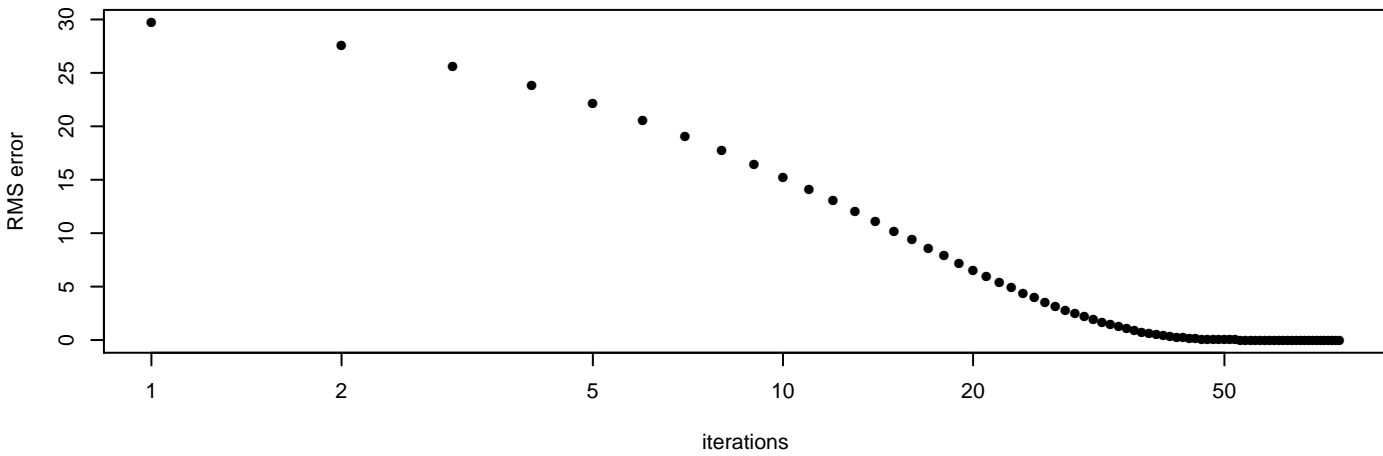
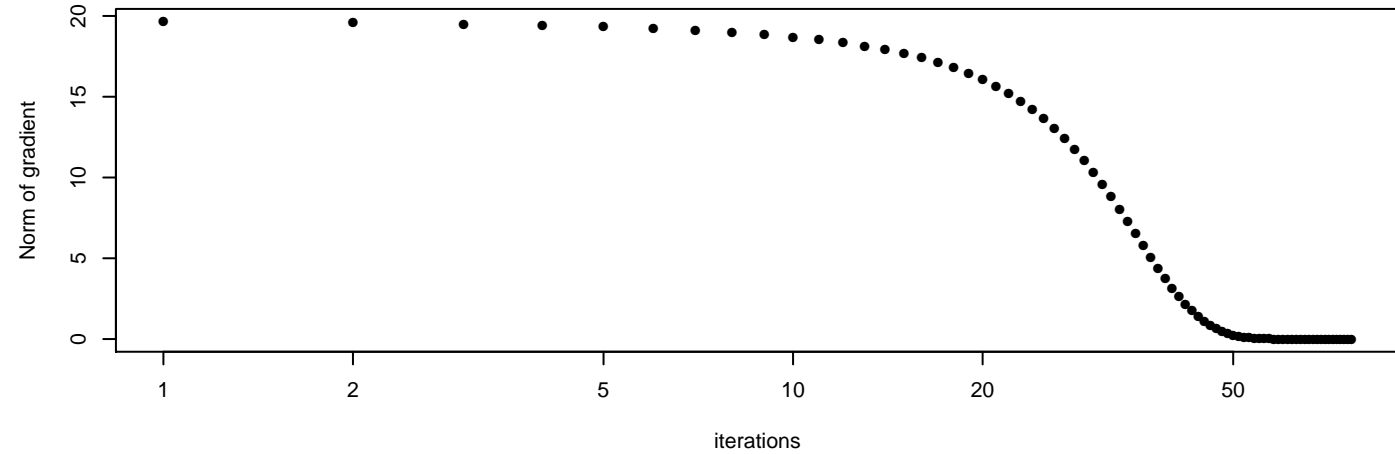
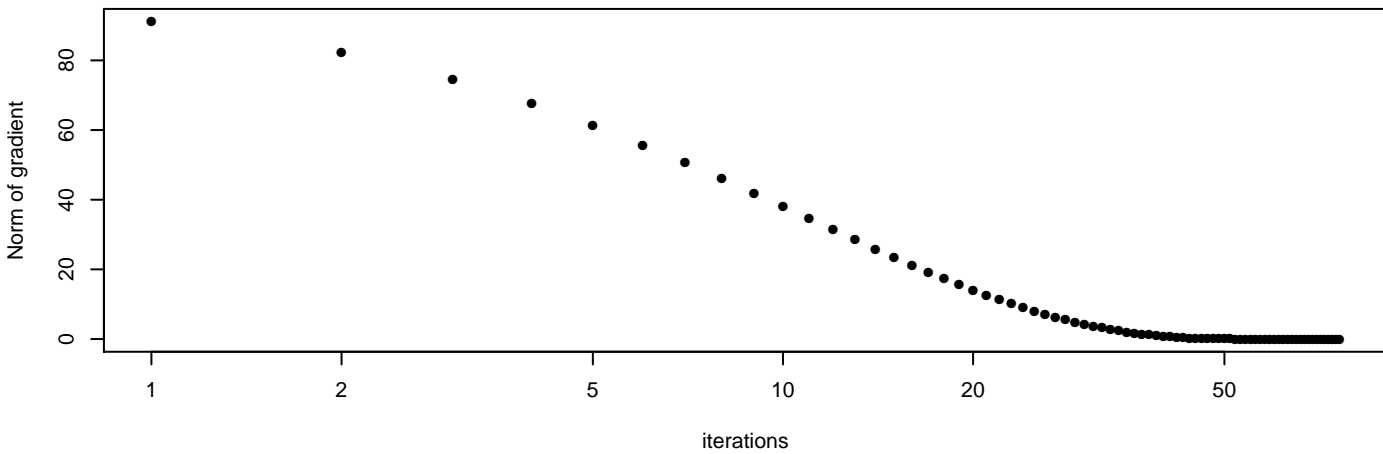
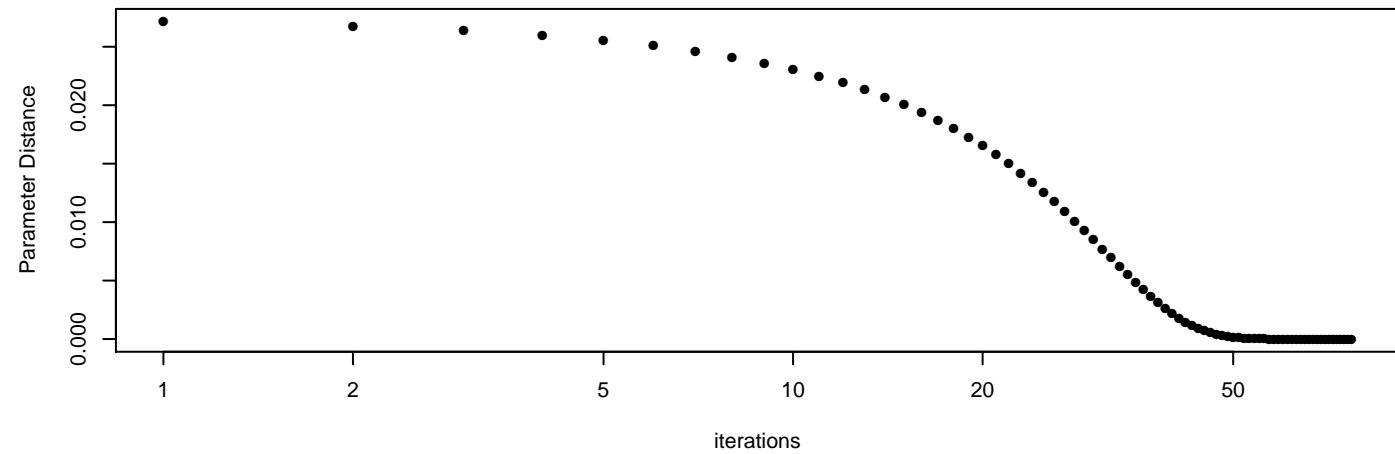
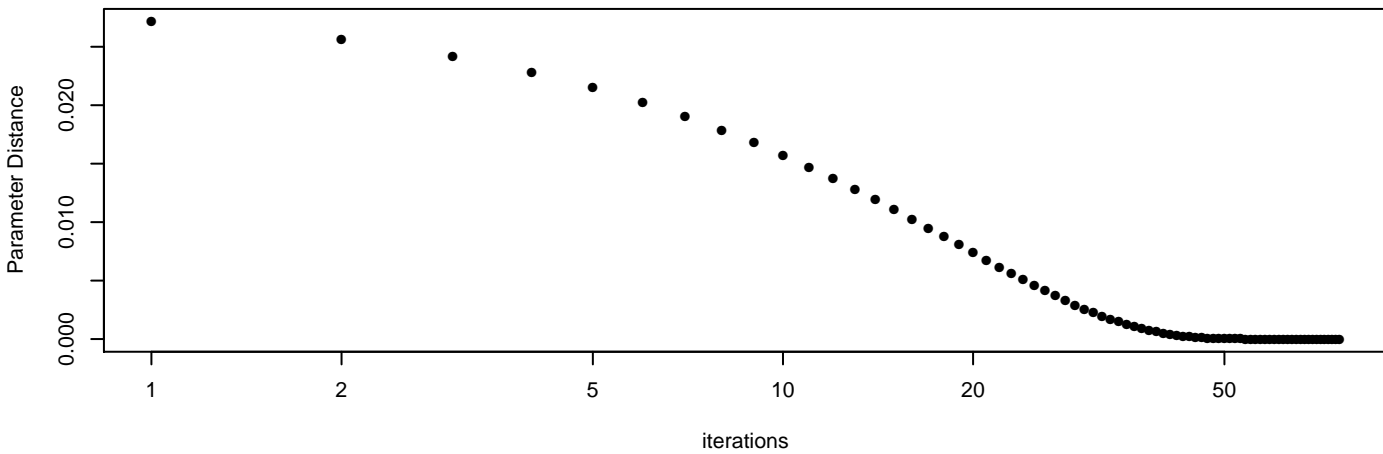
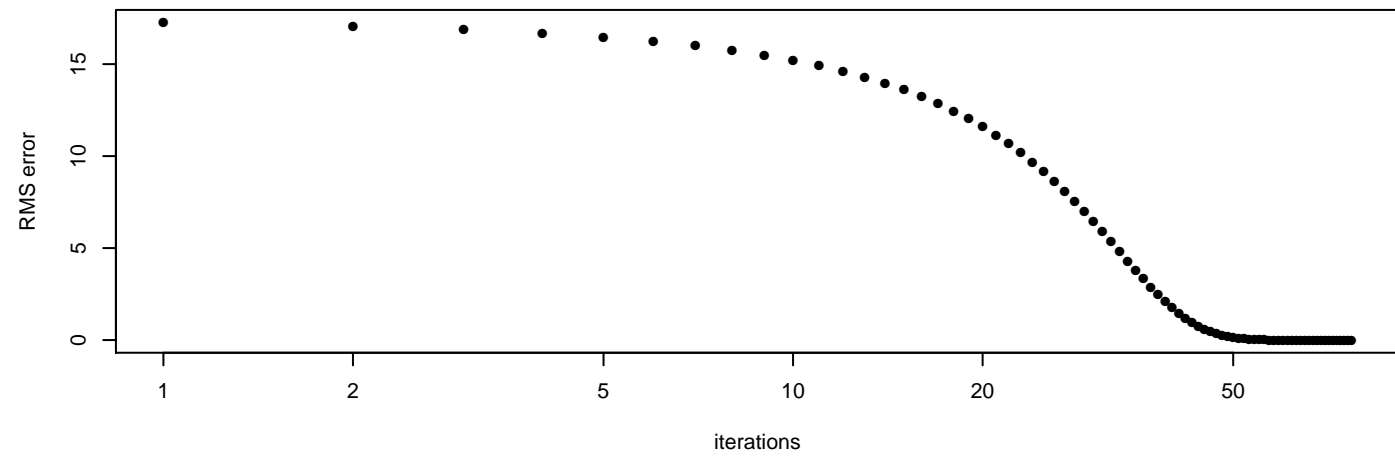
Negative Perturbation



Parameter0

Positive Perturbation

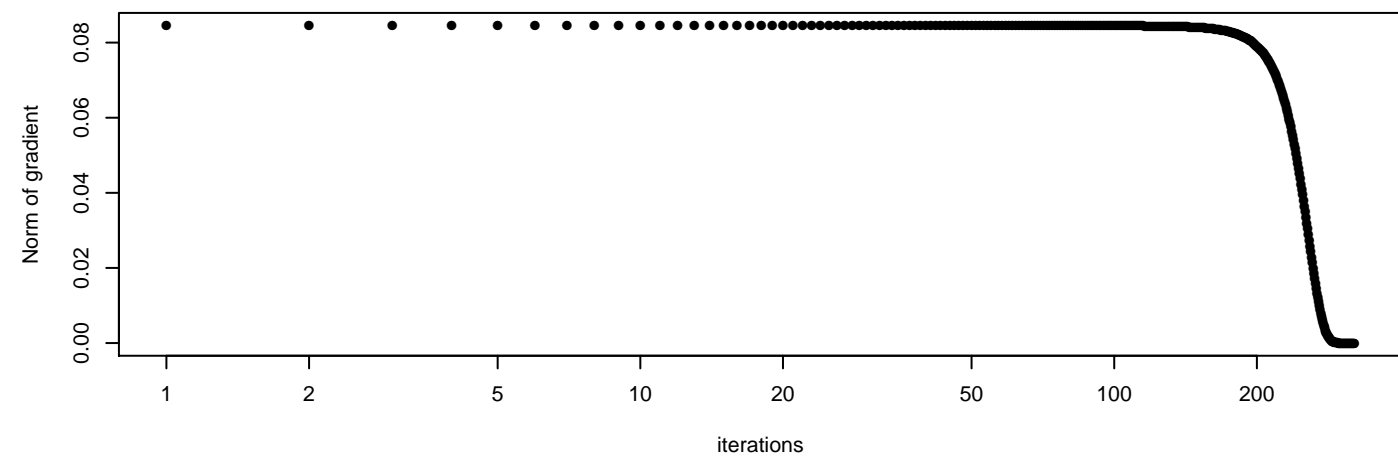
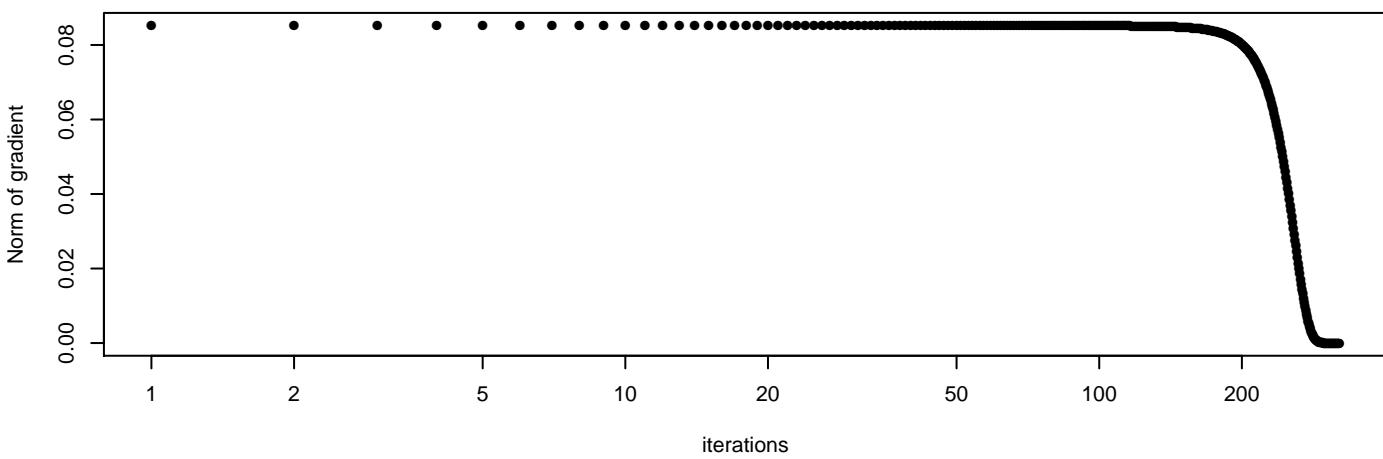
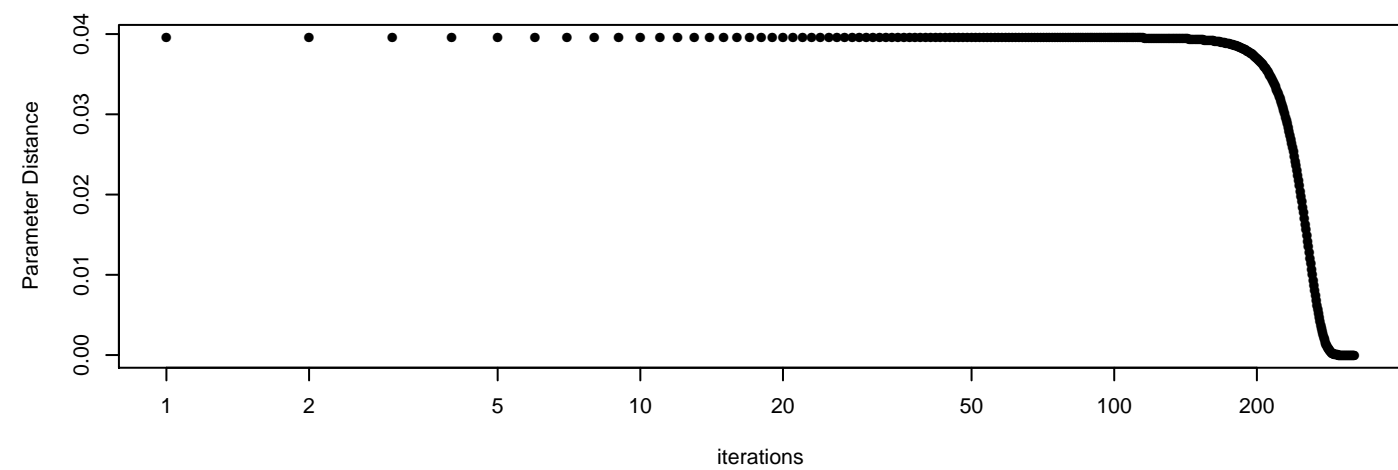
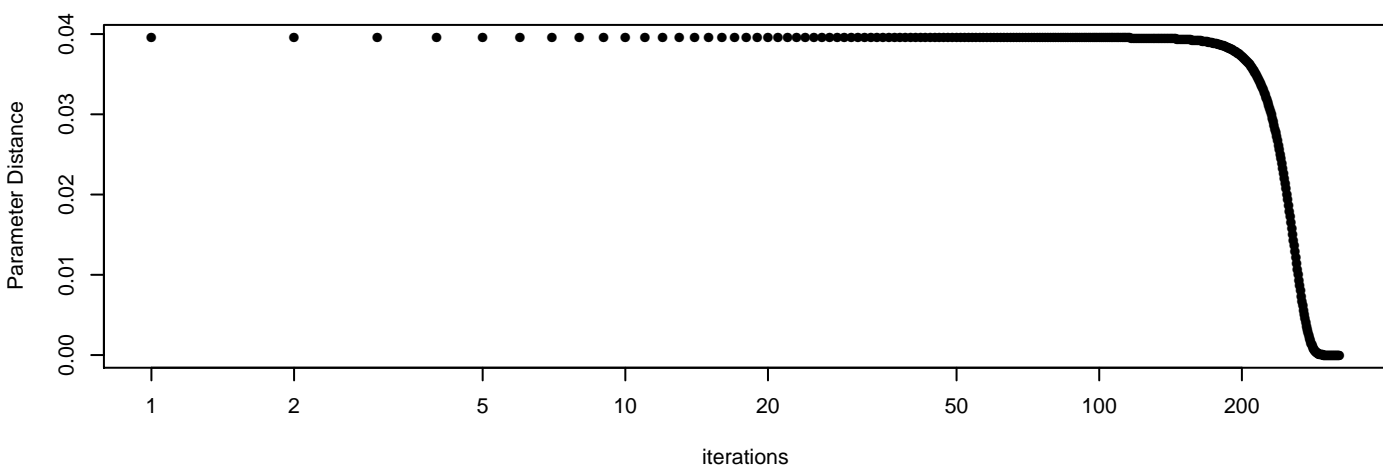
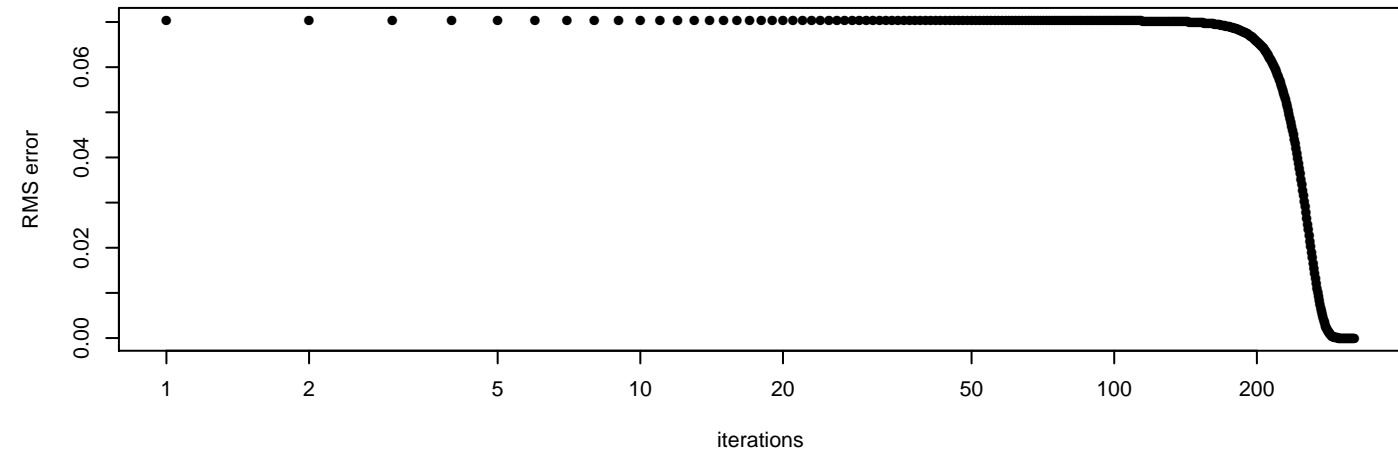
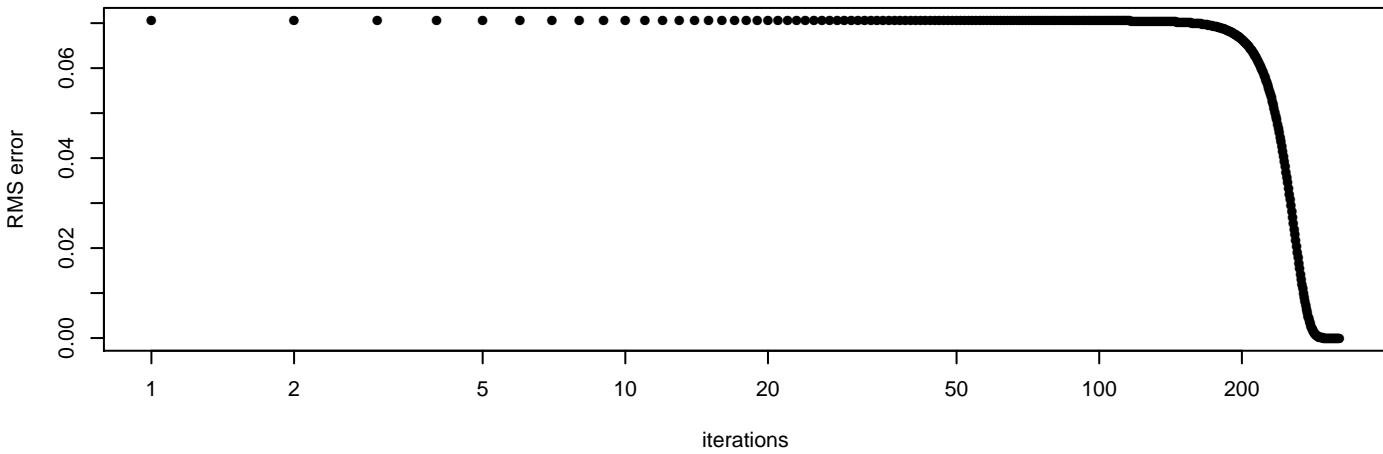


Negative Perturbation**Parameter1****Positive Perturbation**

Parameter10

Negative Perturbation

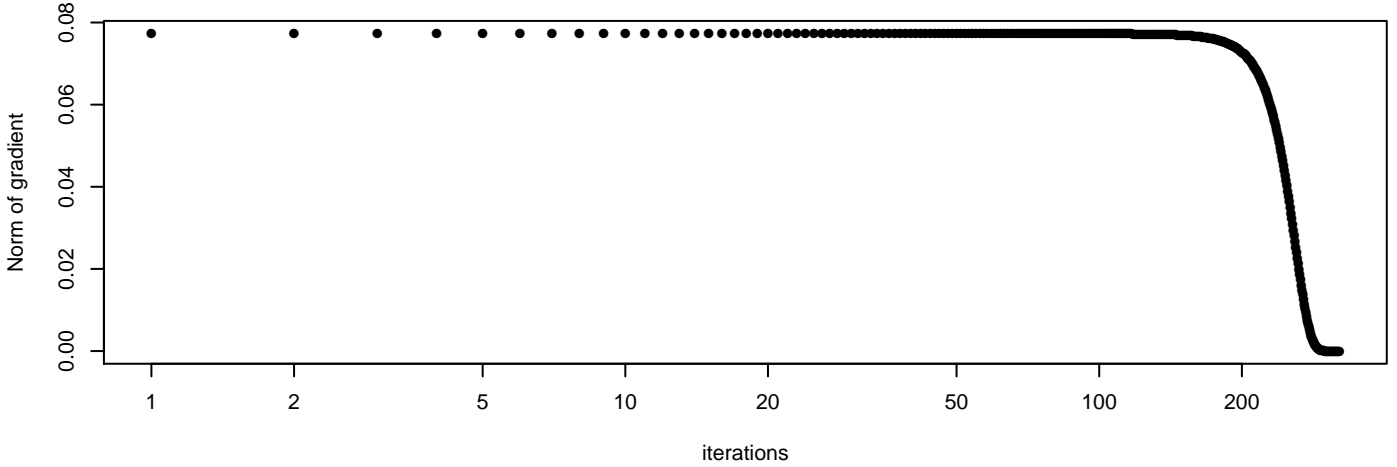
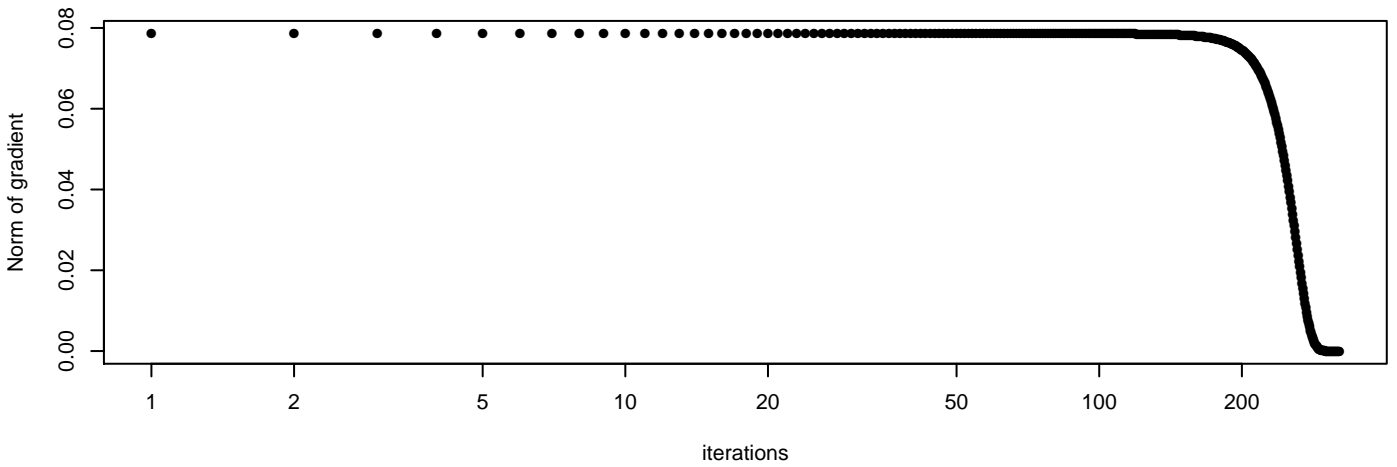
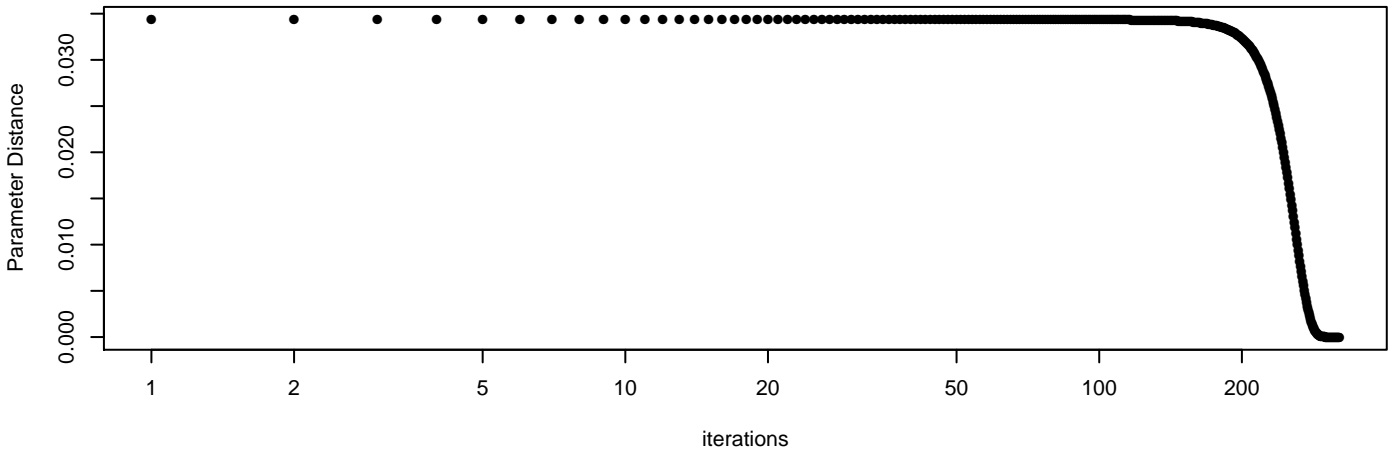
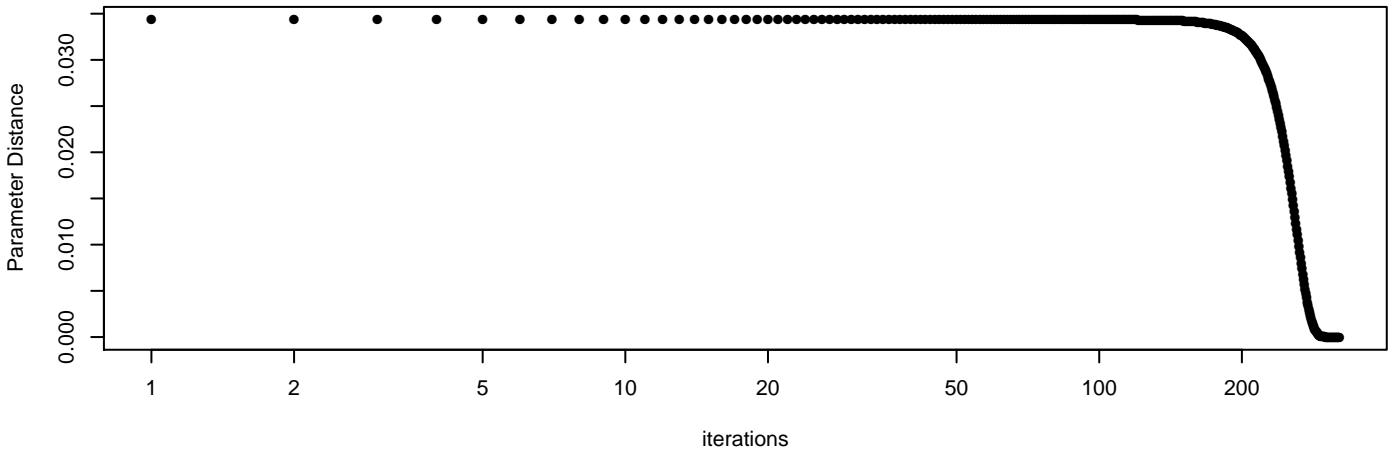
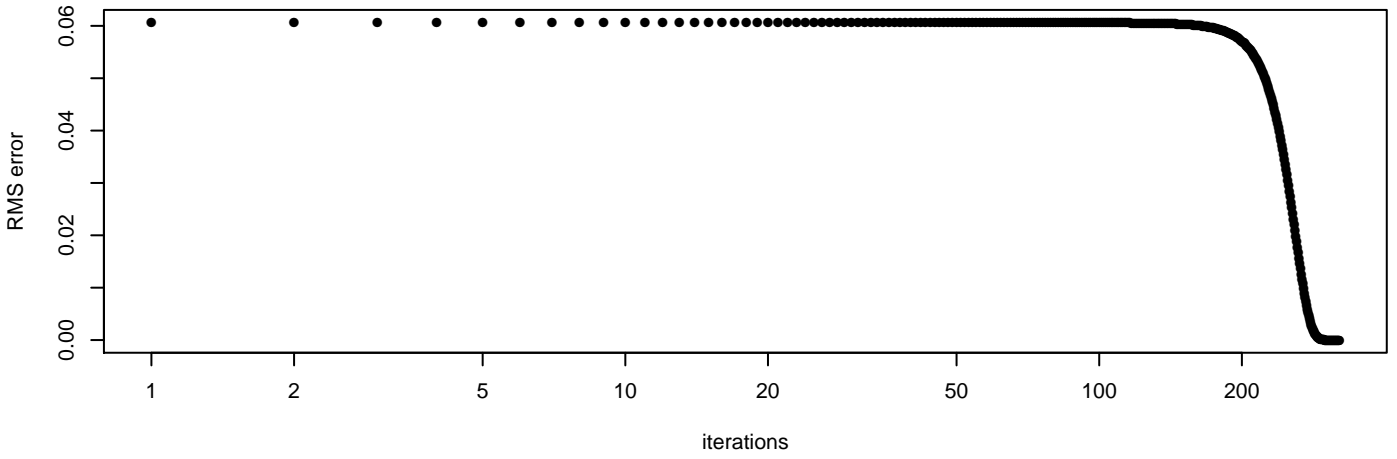
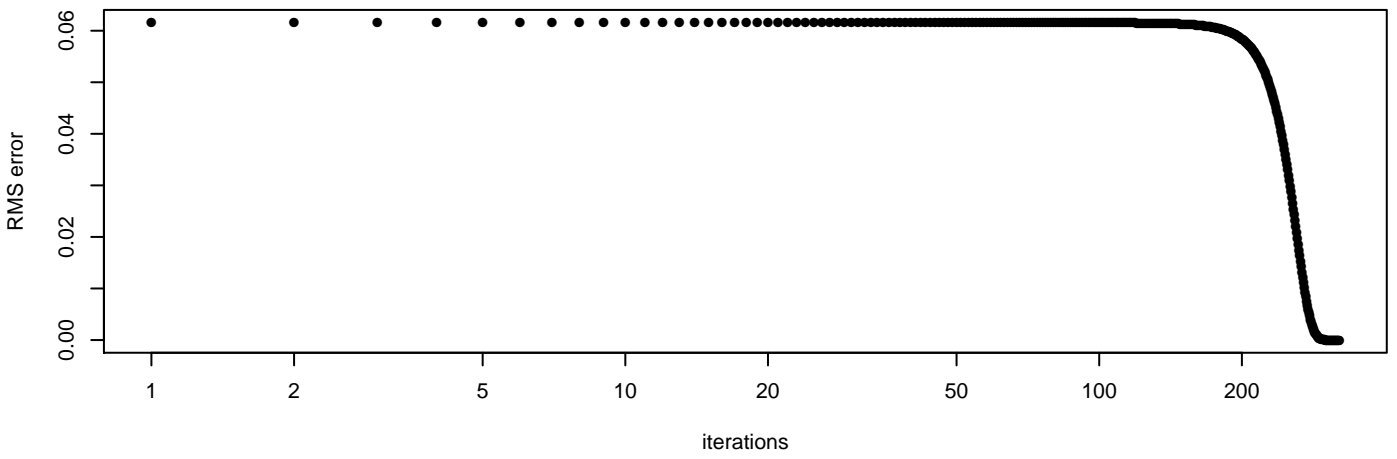
Positive Perturbation



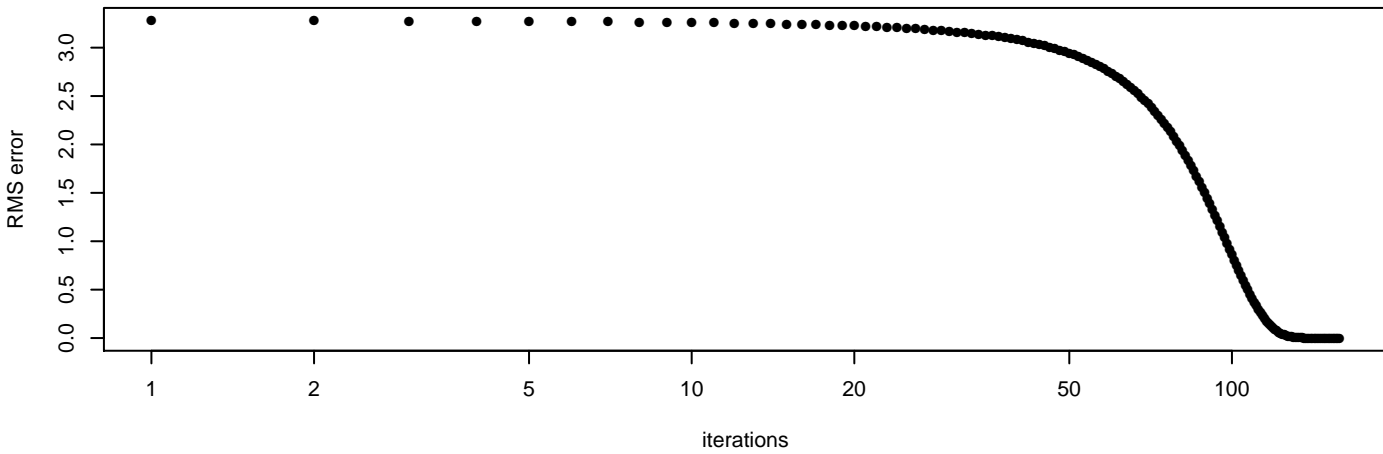
Parameter11

Negative Perturbation

Positive Perturbation

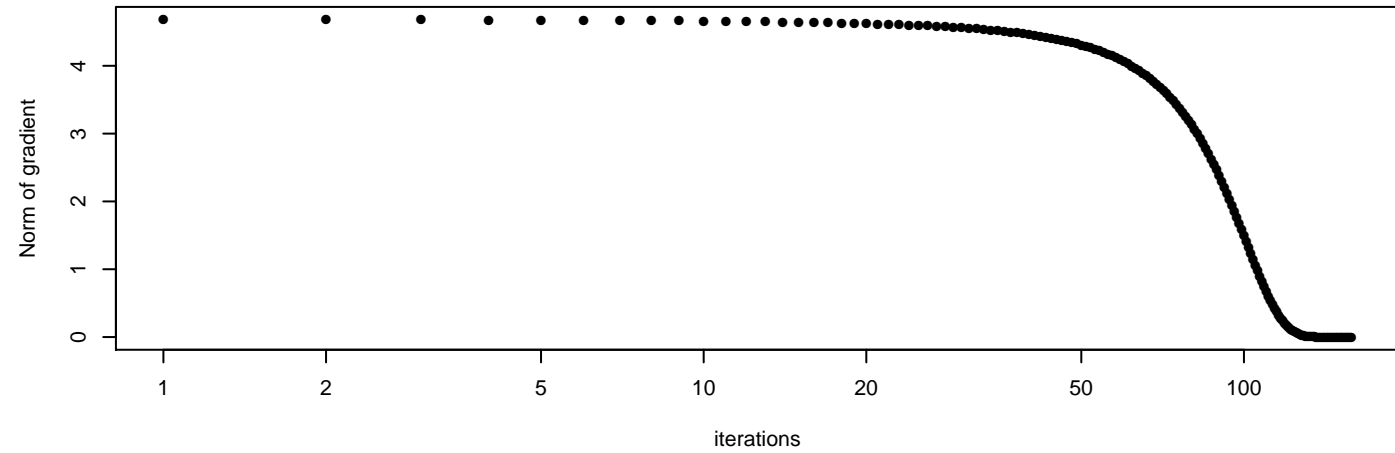
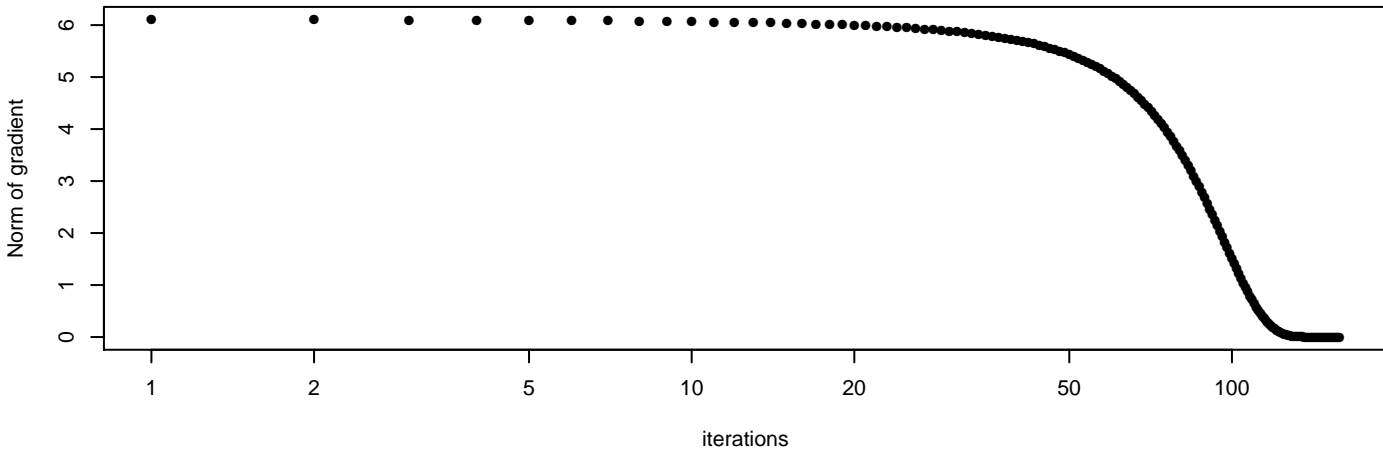
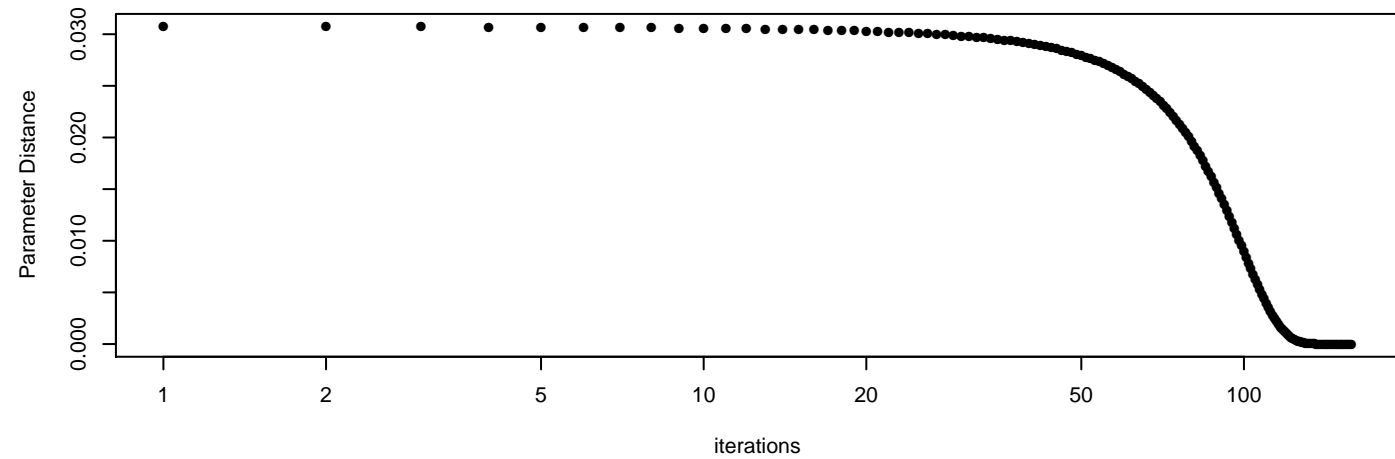
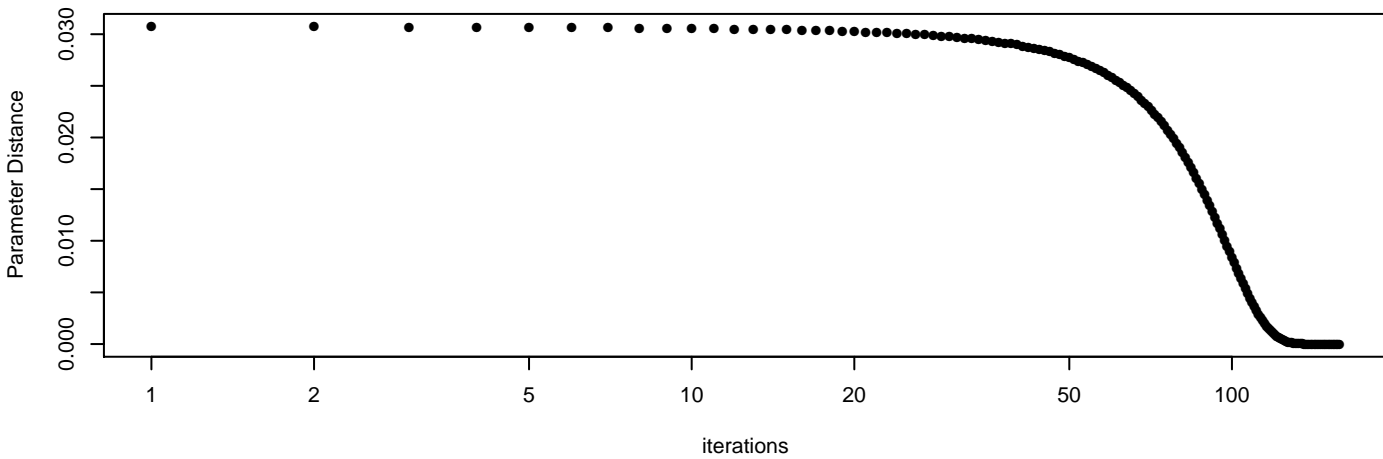
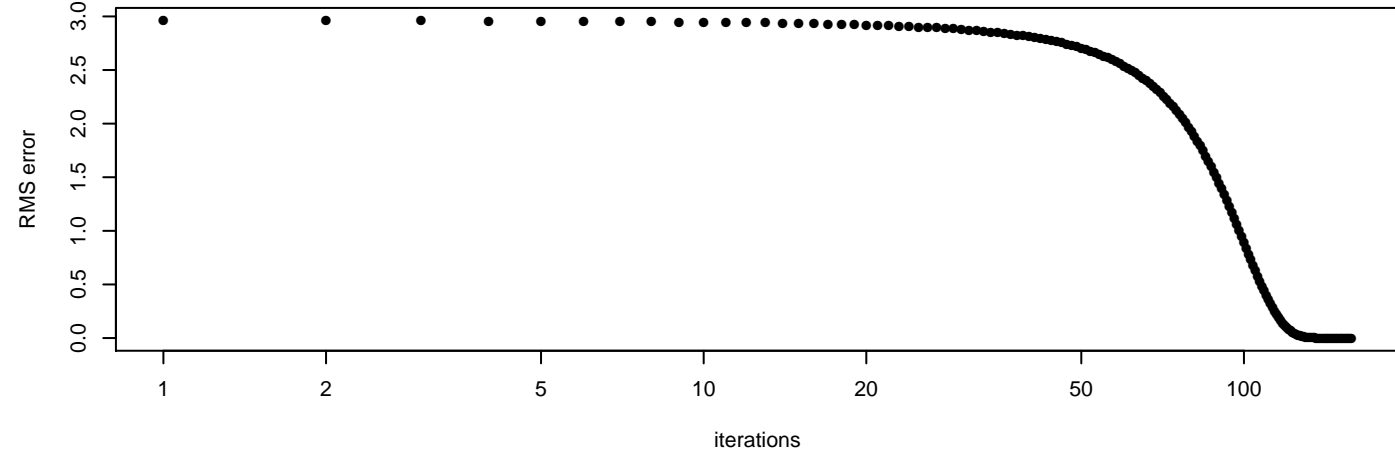


Negative Perturbation

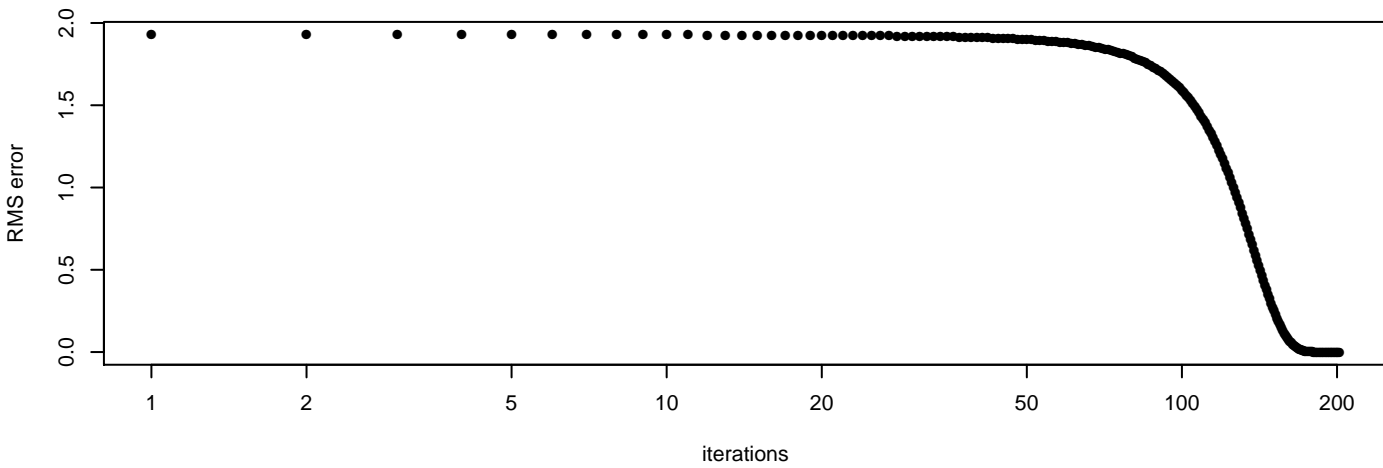


Parameter12

Positive Perturbation

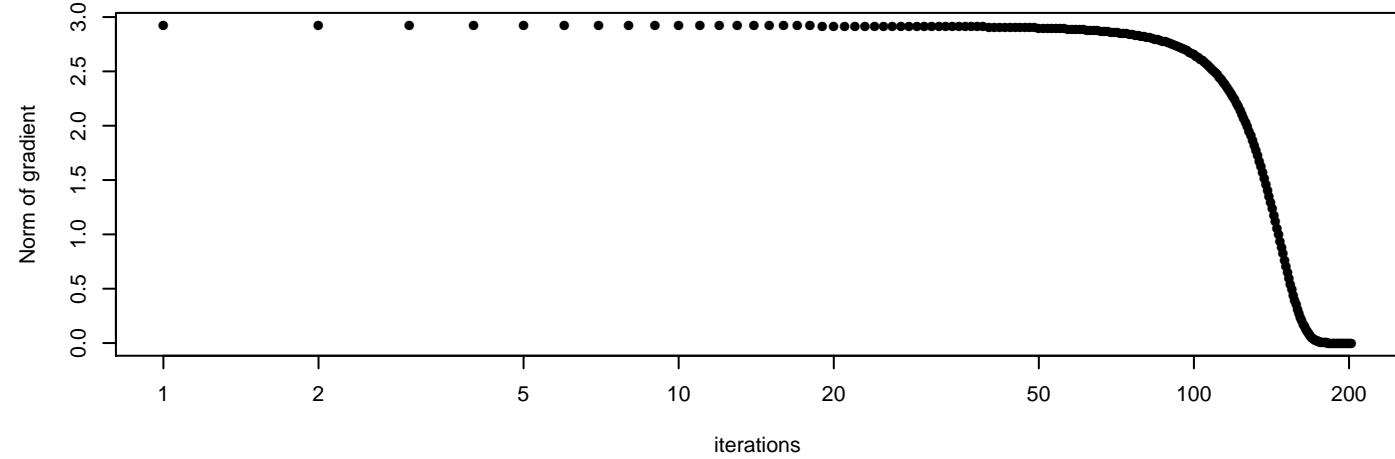
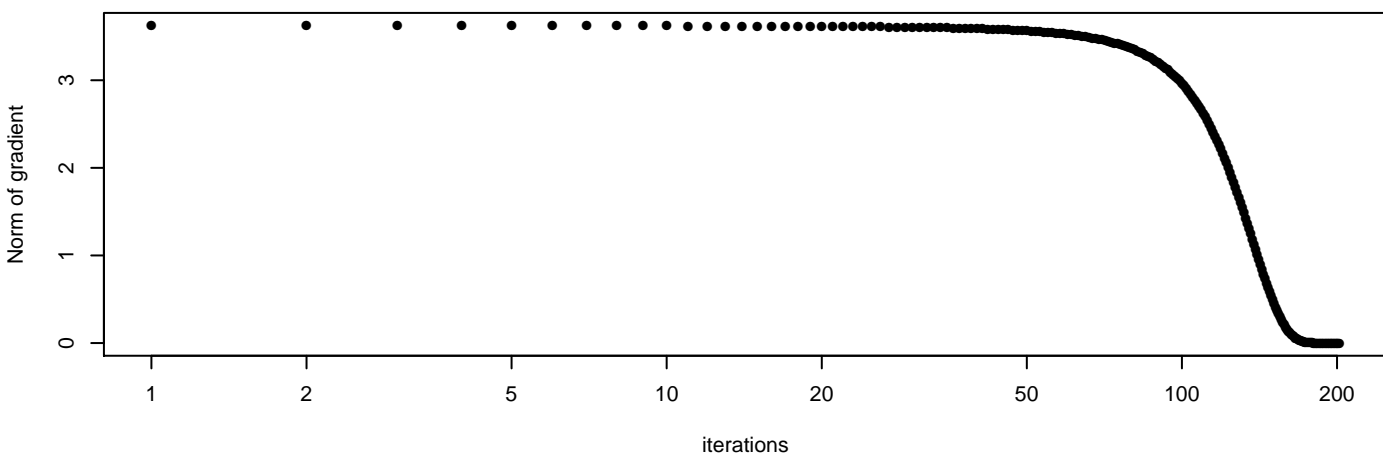
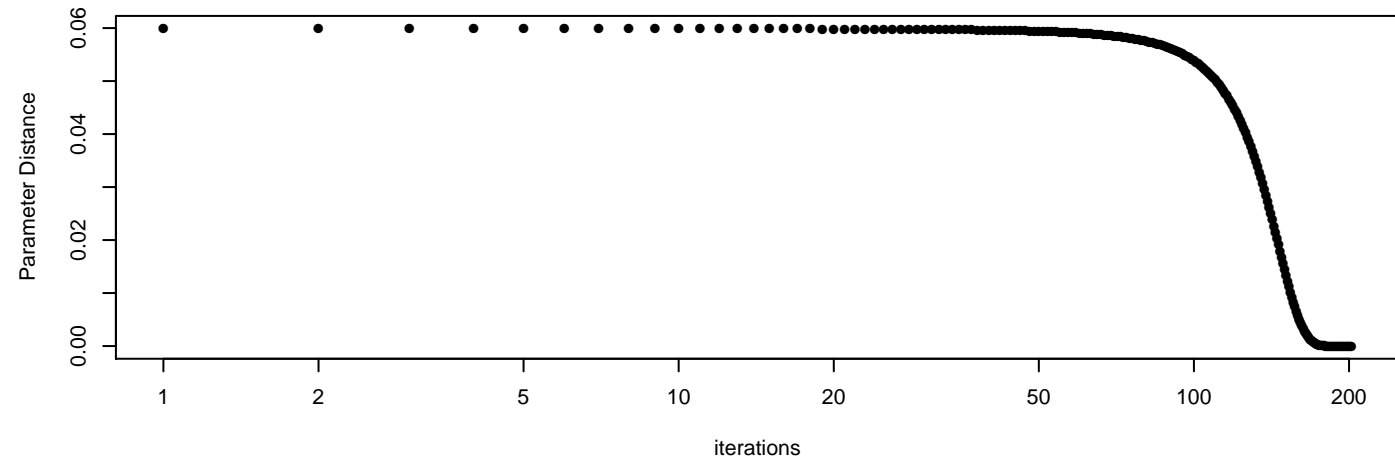
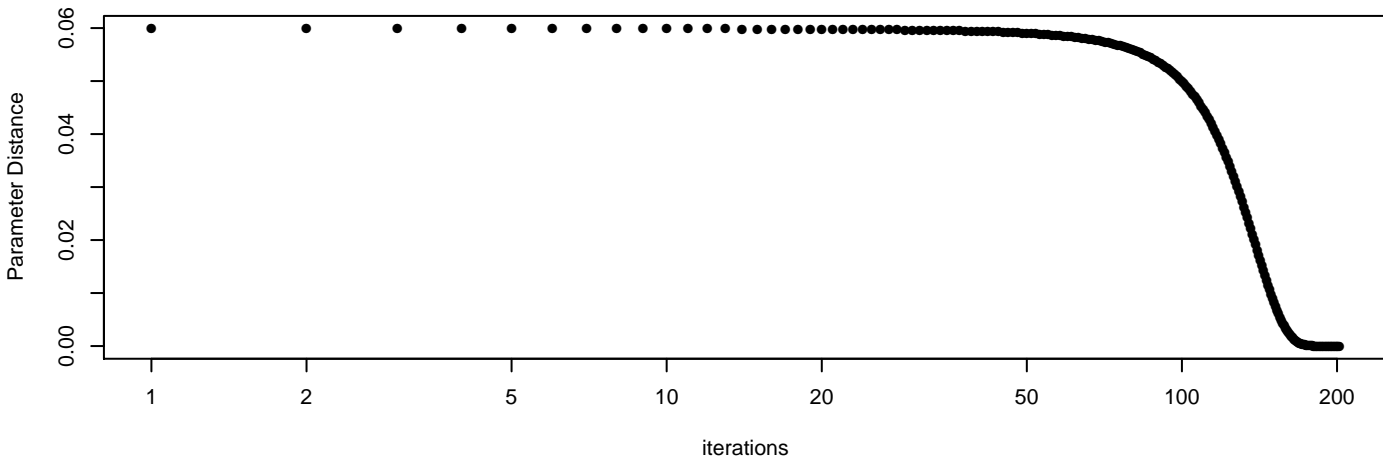
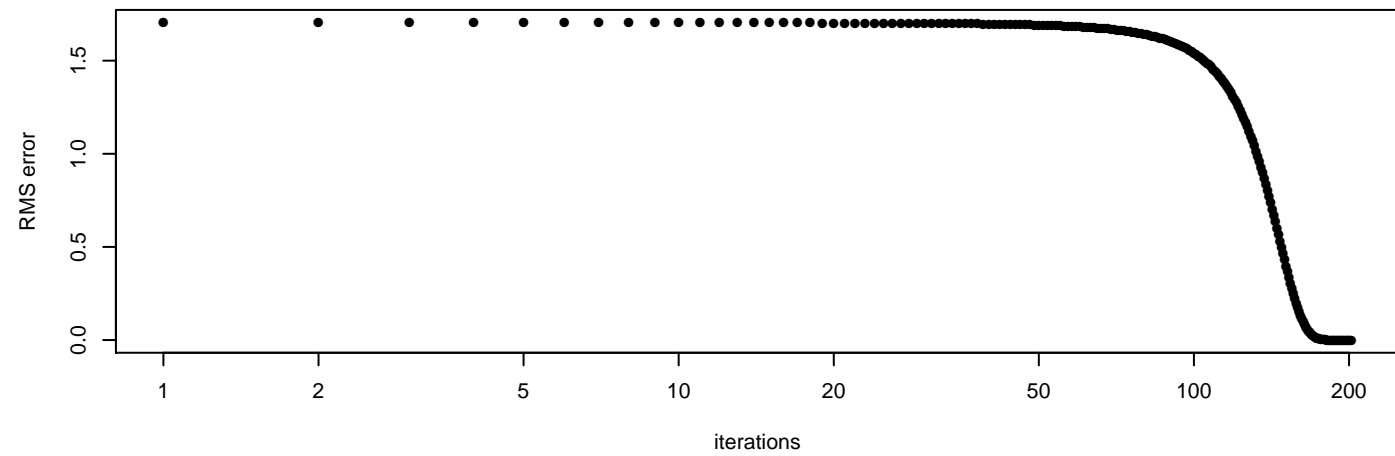


Negative Perturbation

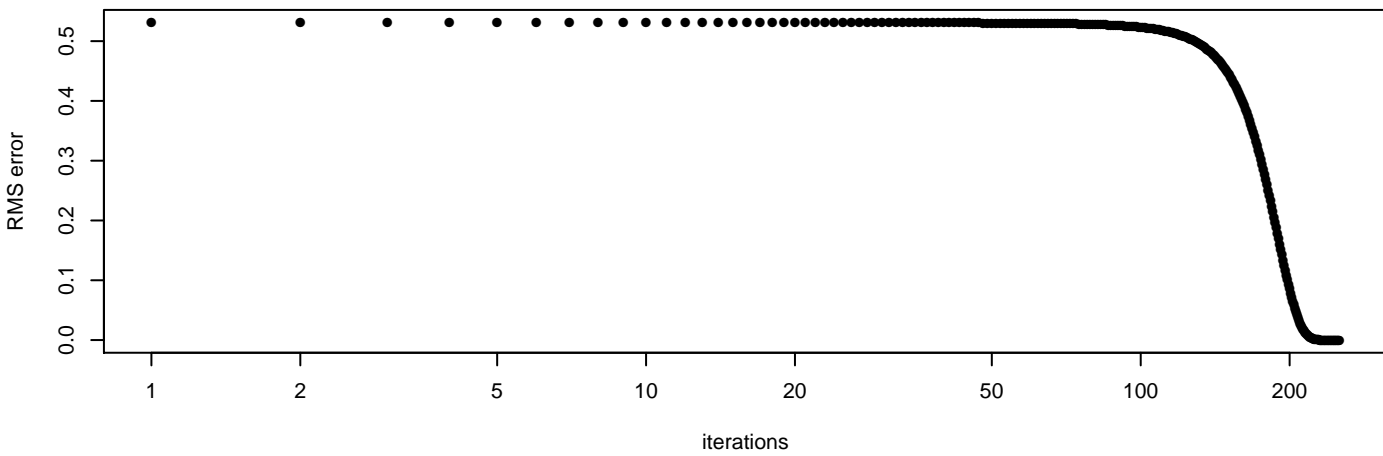


Parameter13

Positive Perturbation

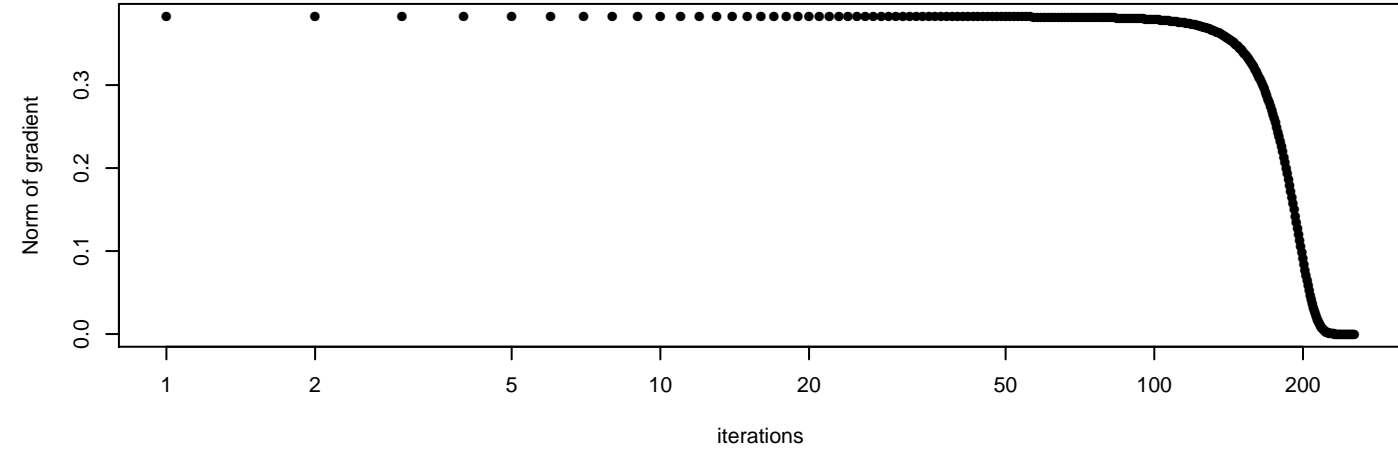
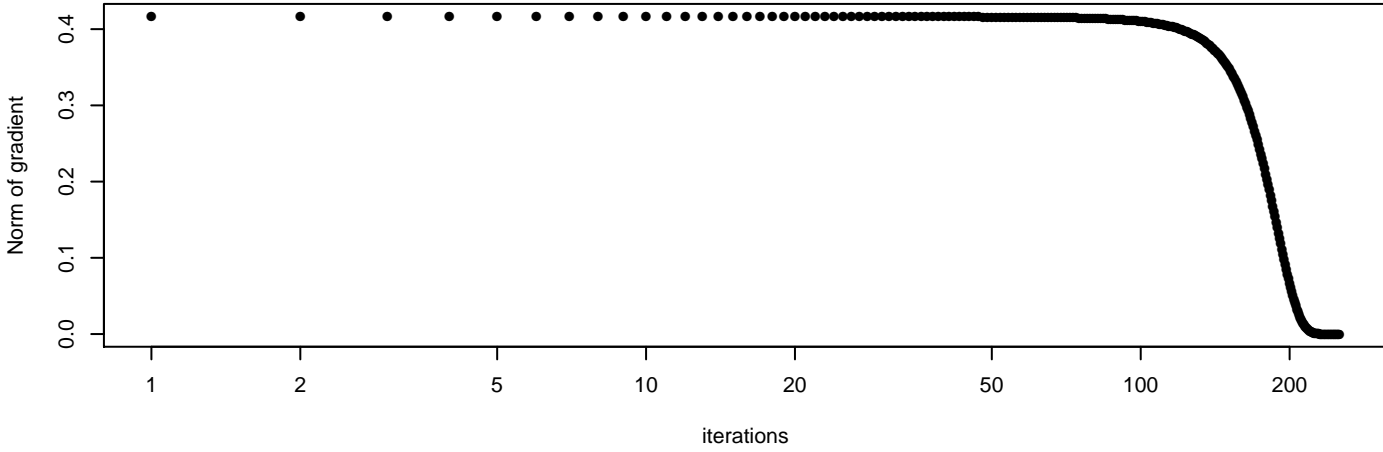
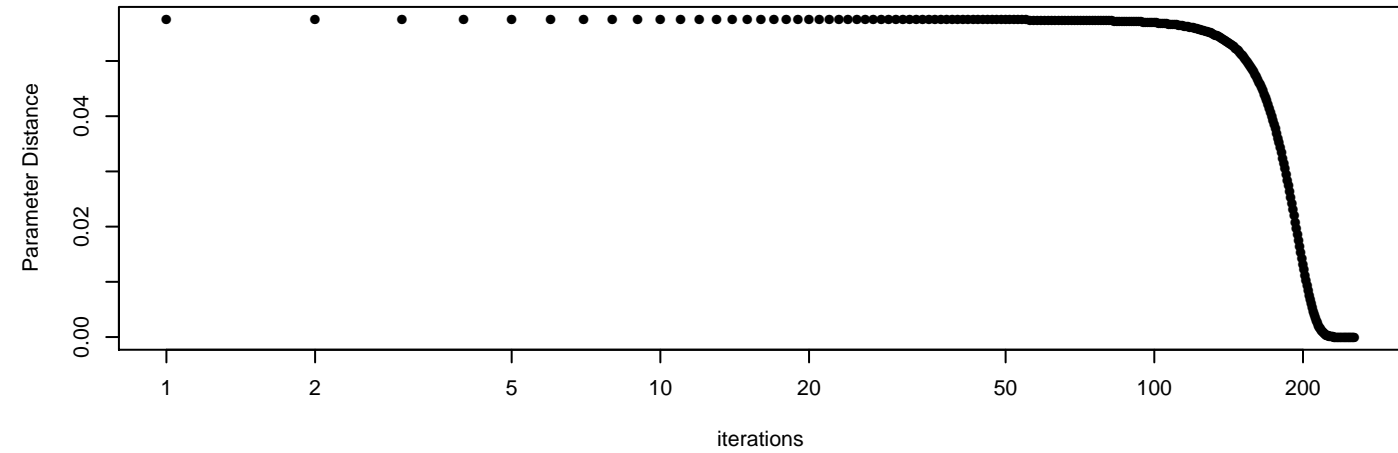
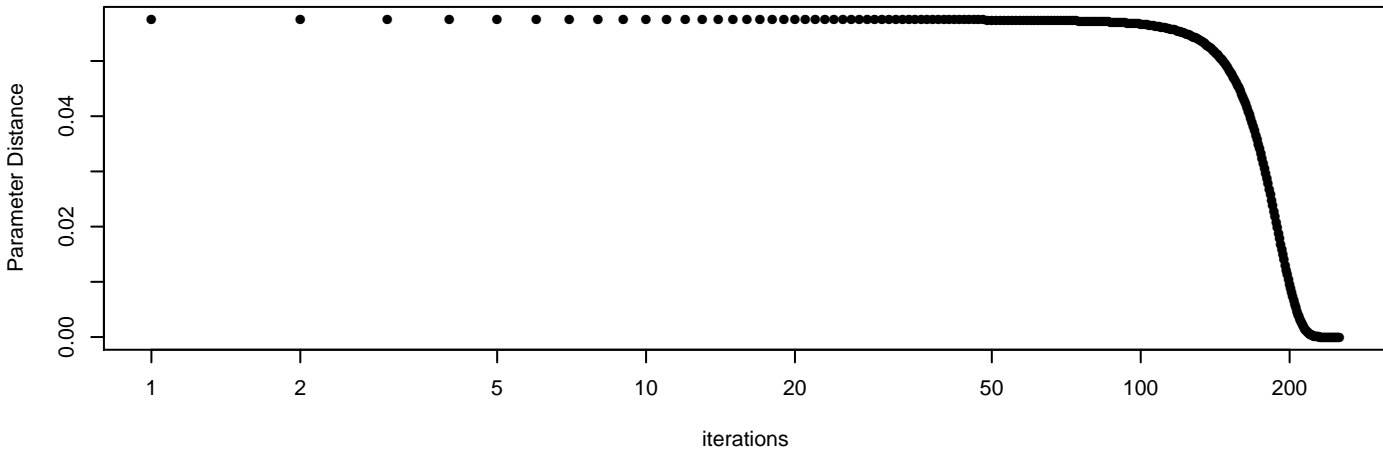
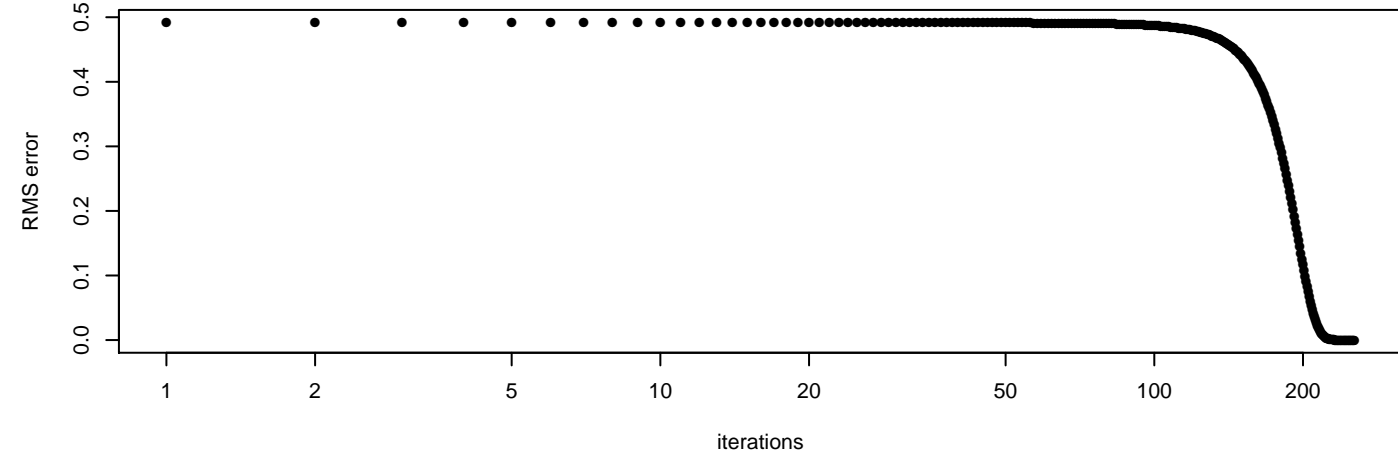


Negative Perturbation

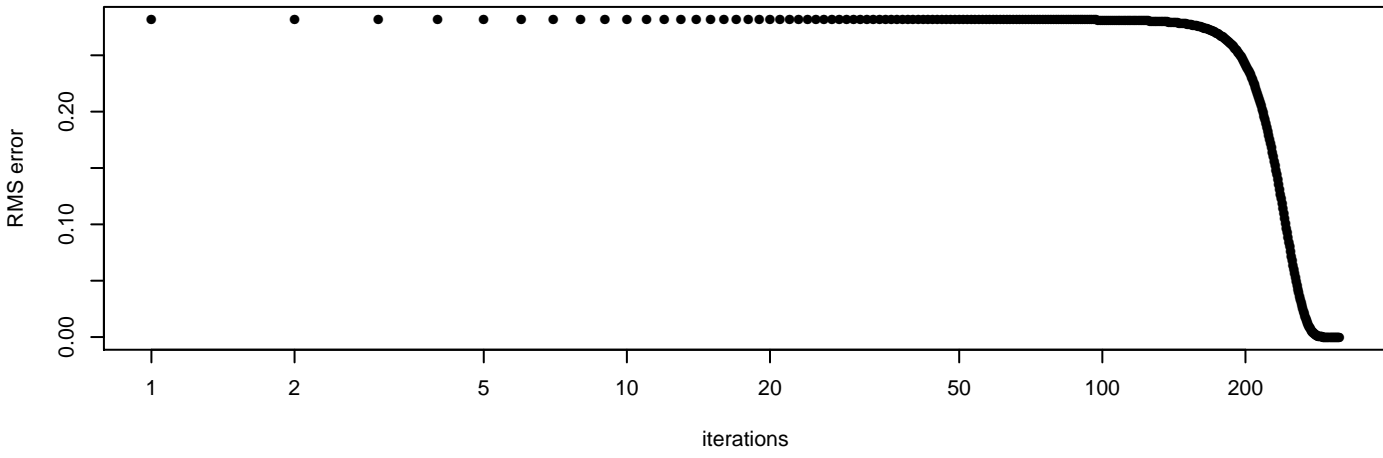


Parameter14

Positive Perturbation

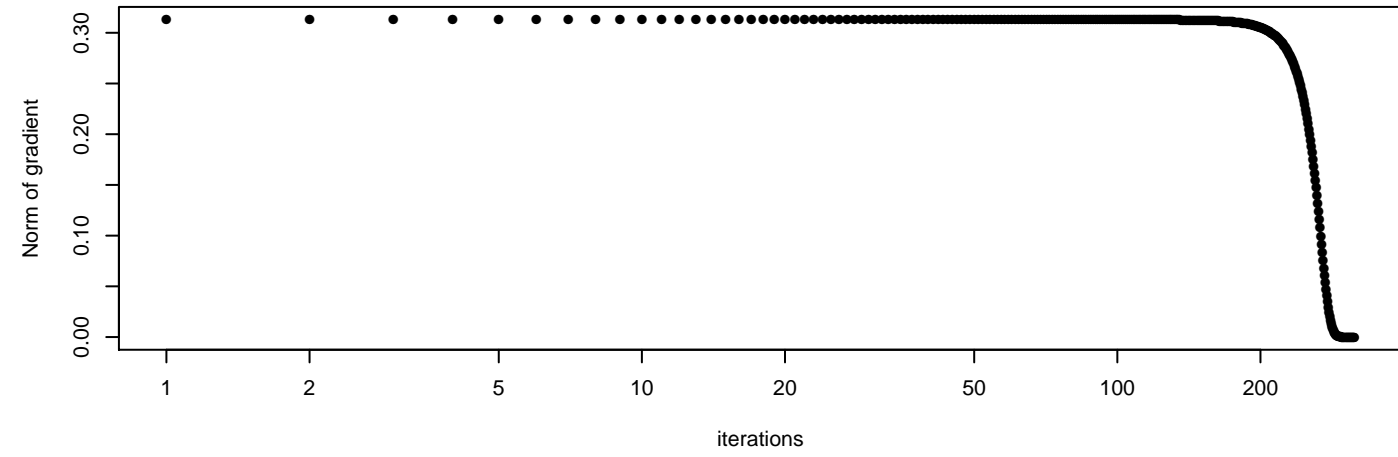
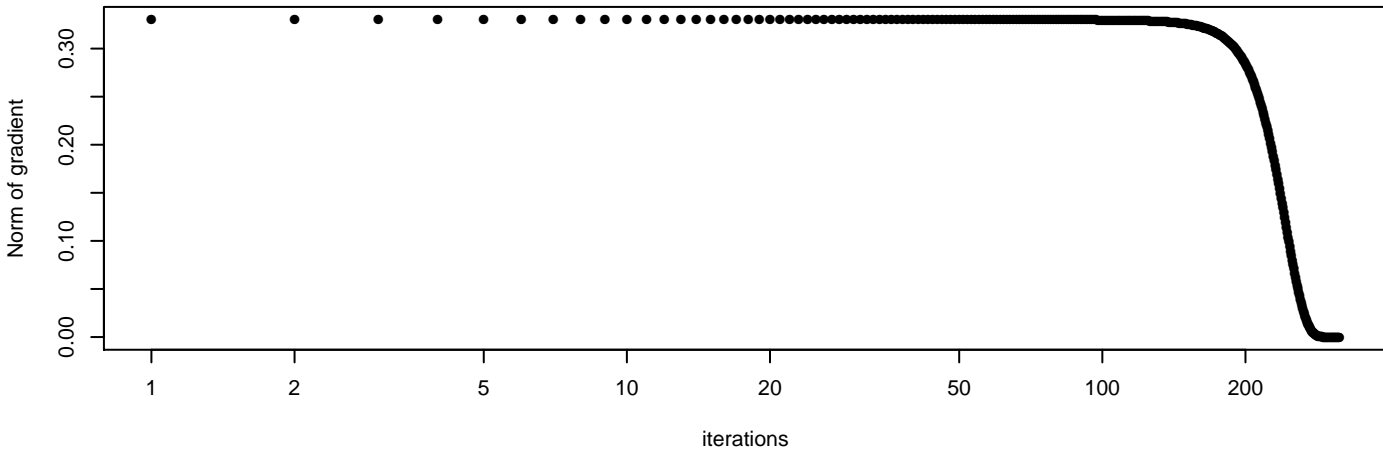
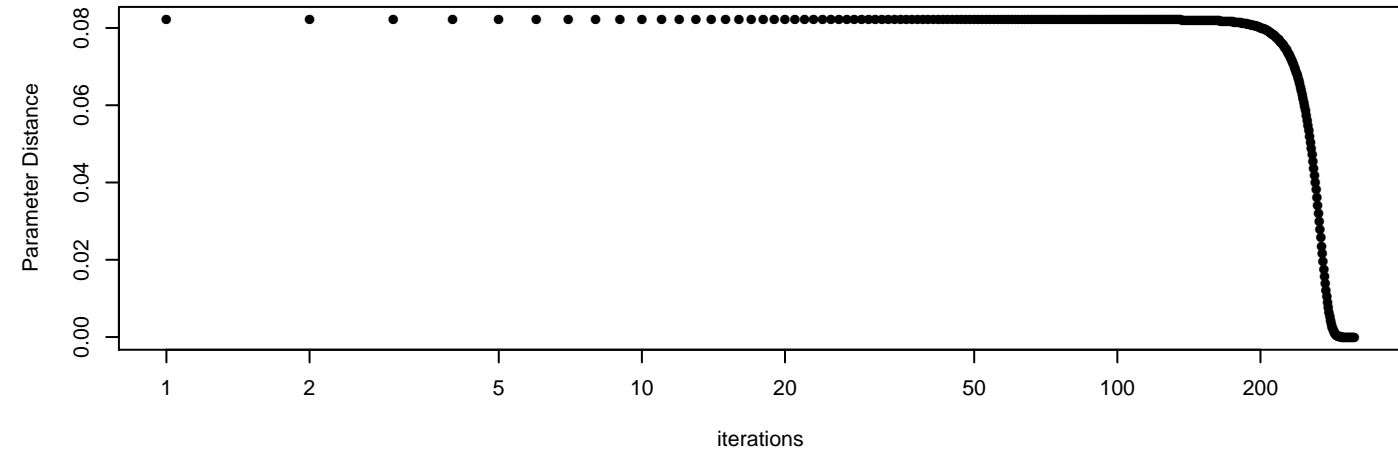
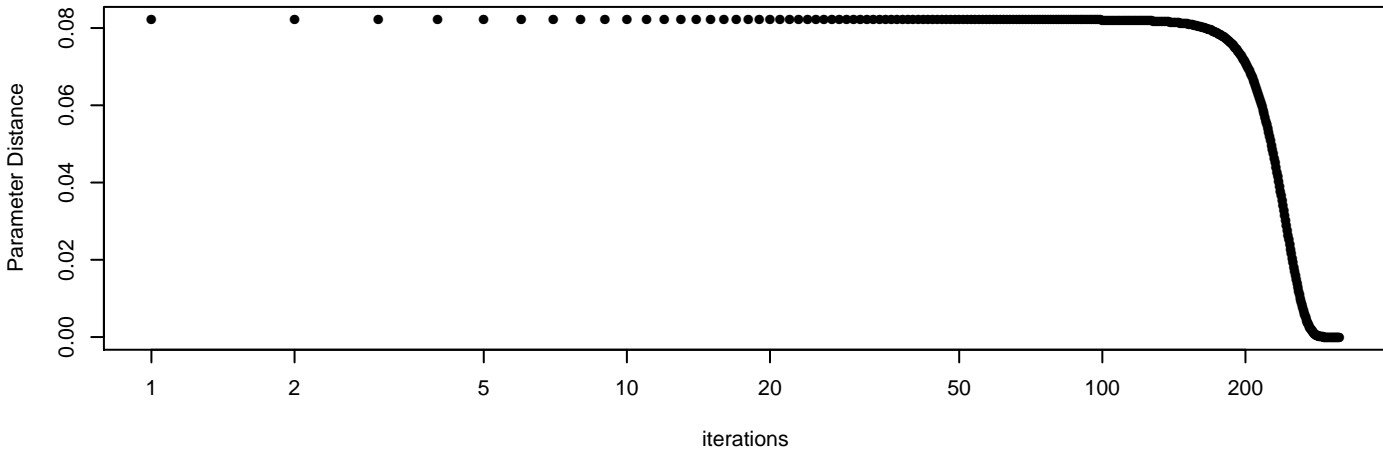
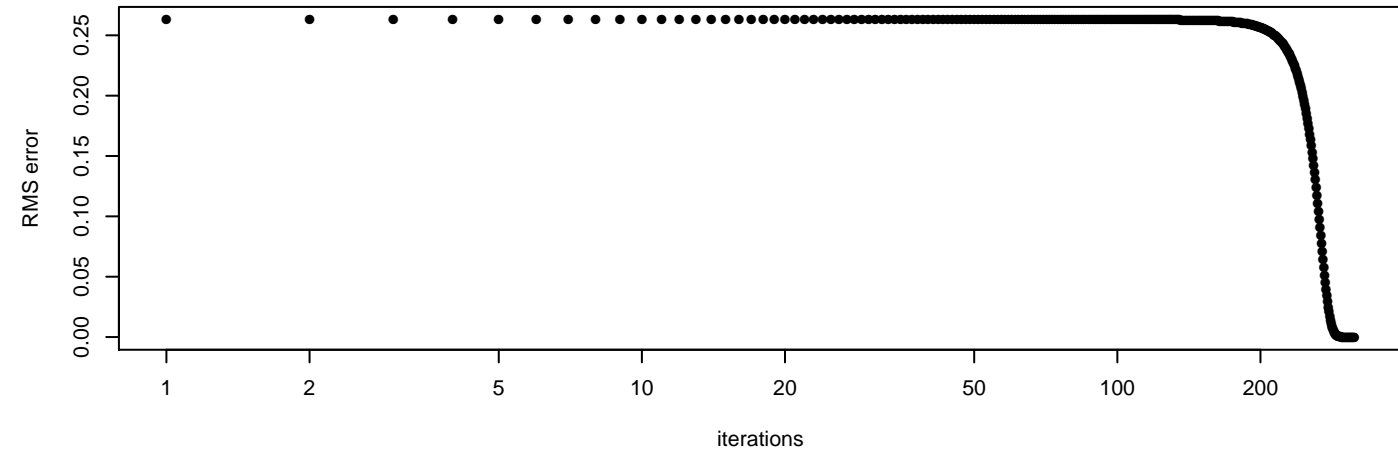


Negative Perturbation

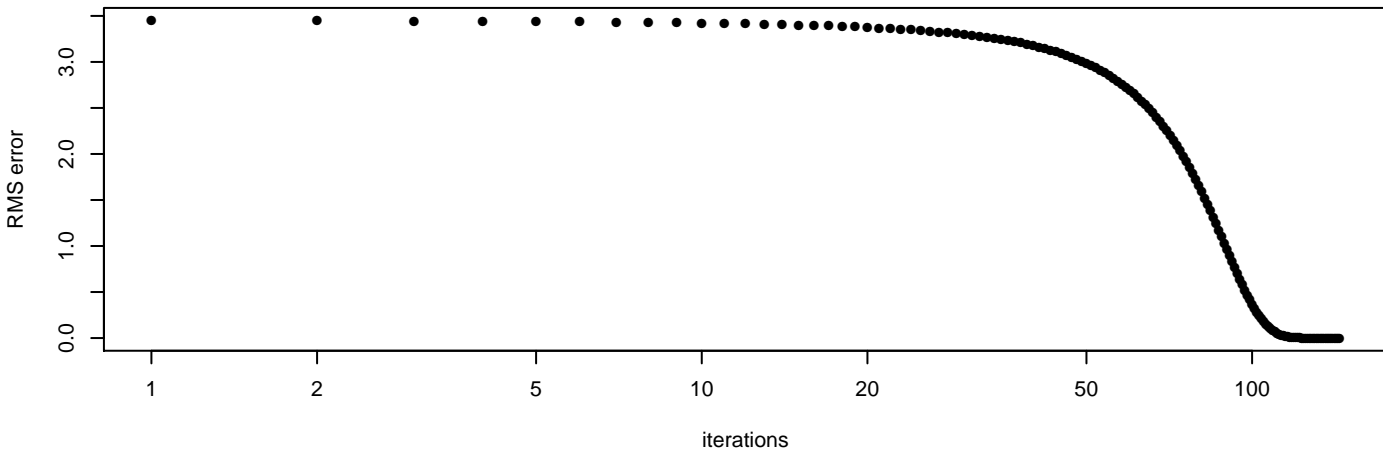


Parameter15

Positive Perturbation

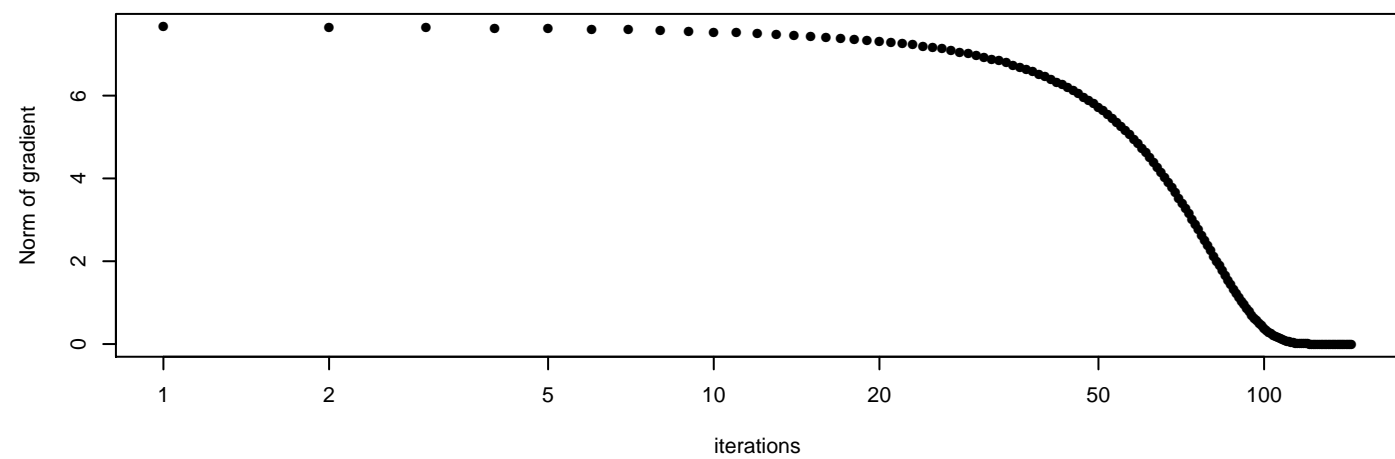
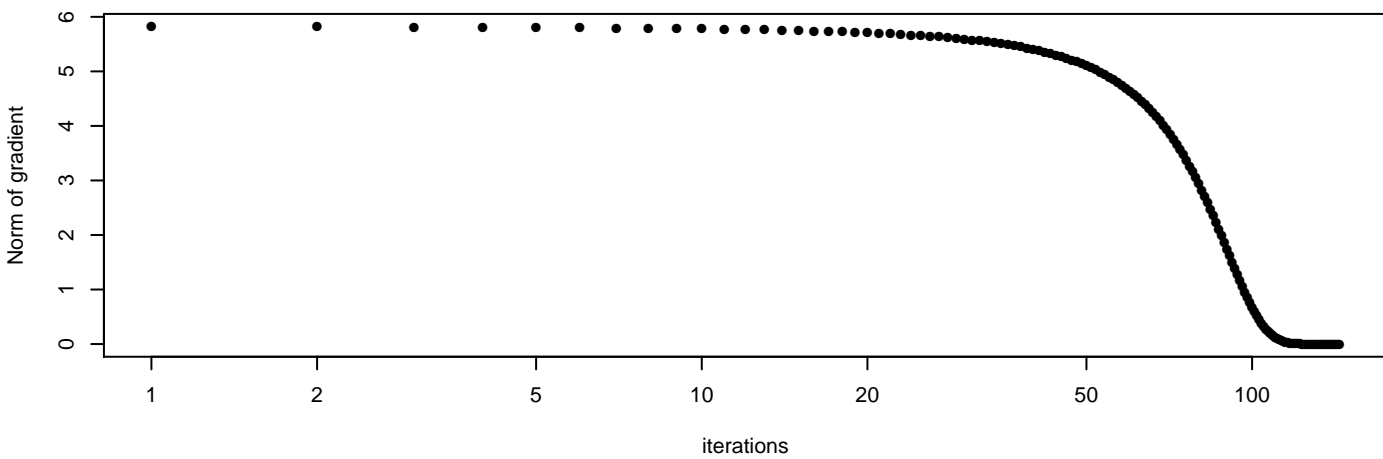
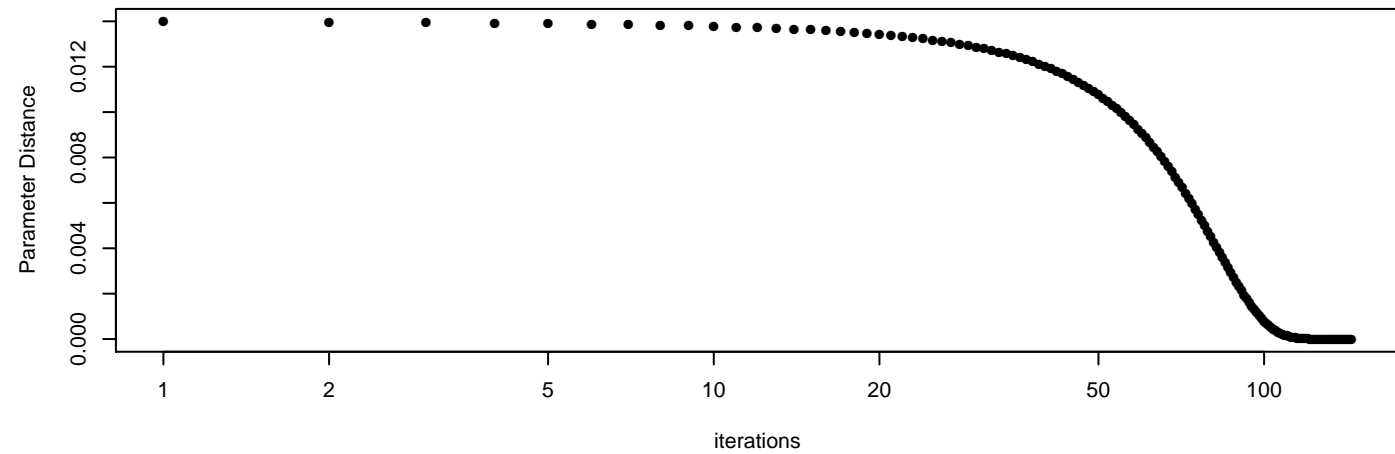
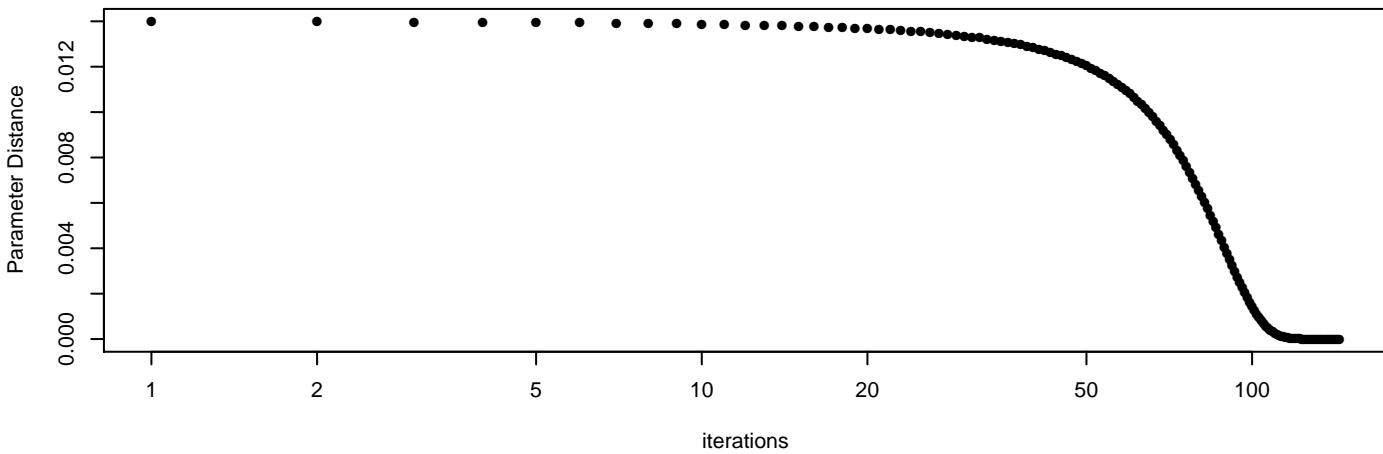
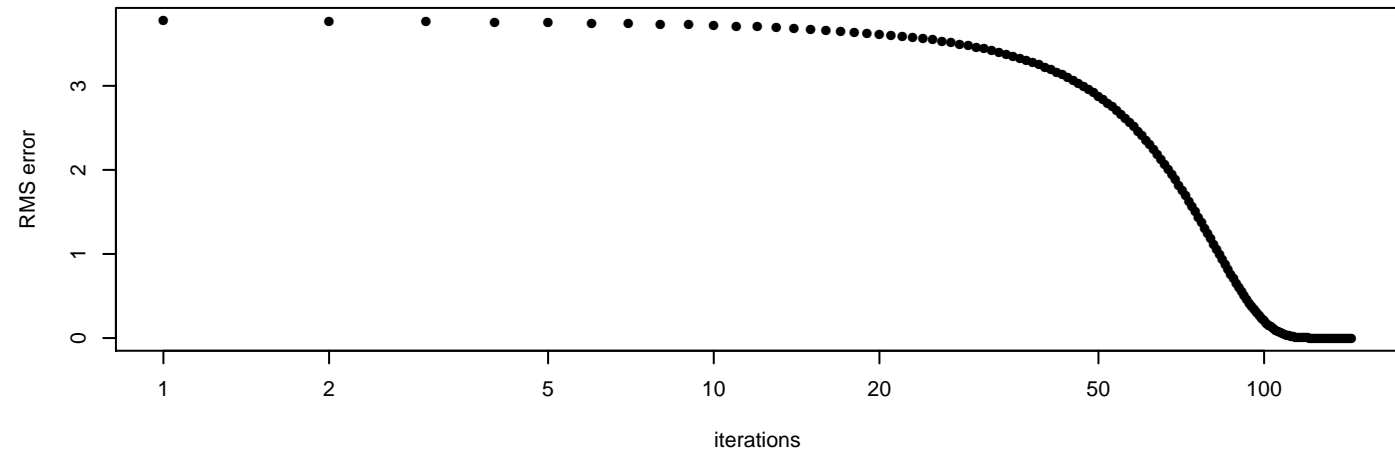


Negative Perturbation

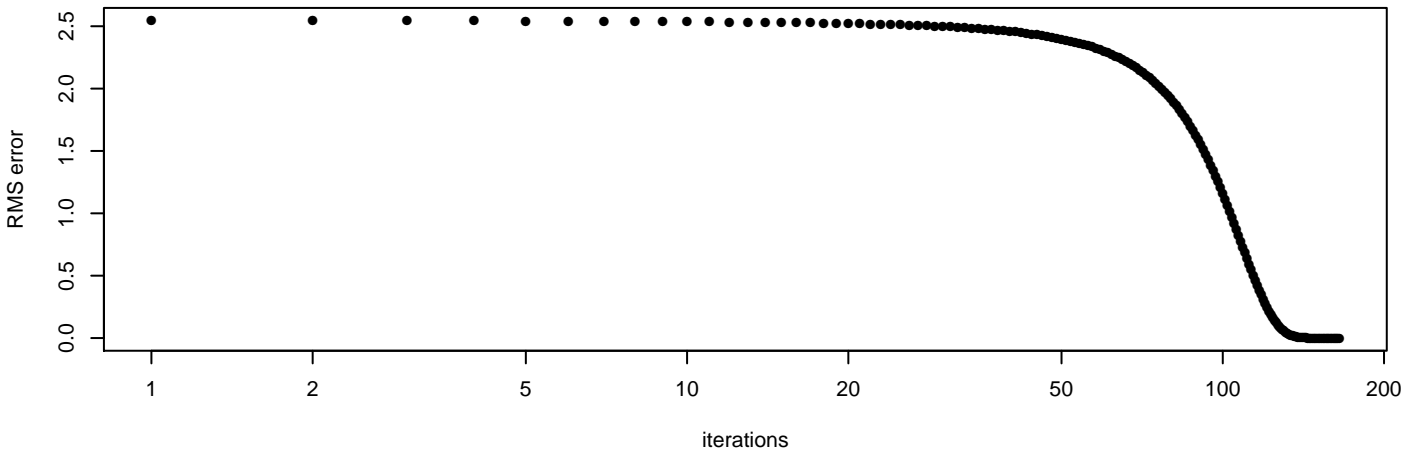


Parameter2

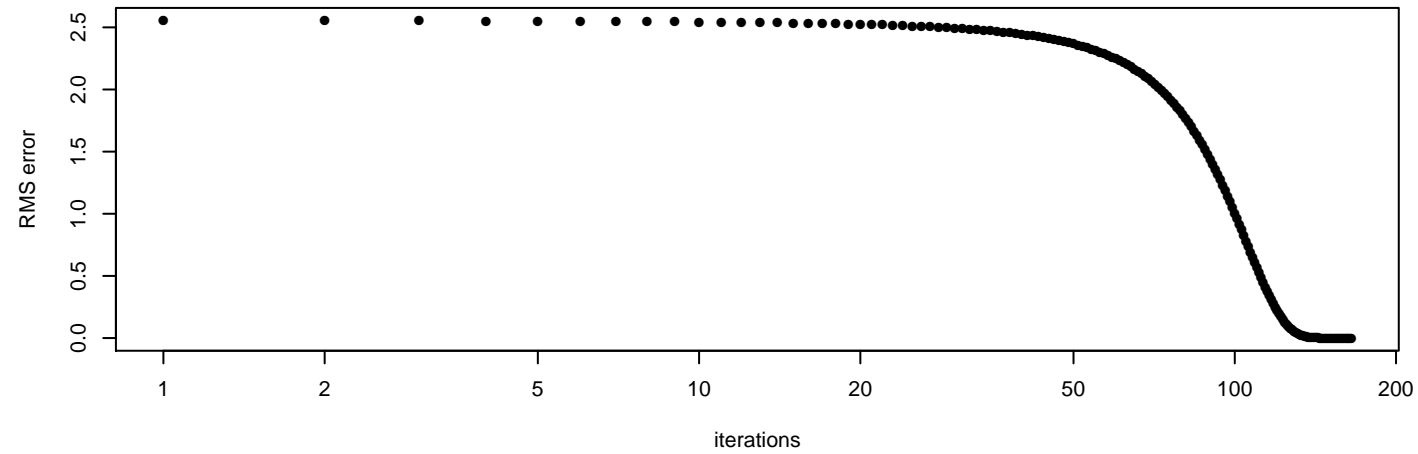
Positive Perturbation



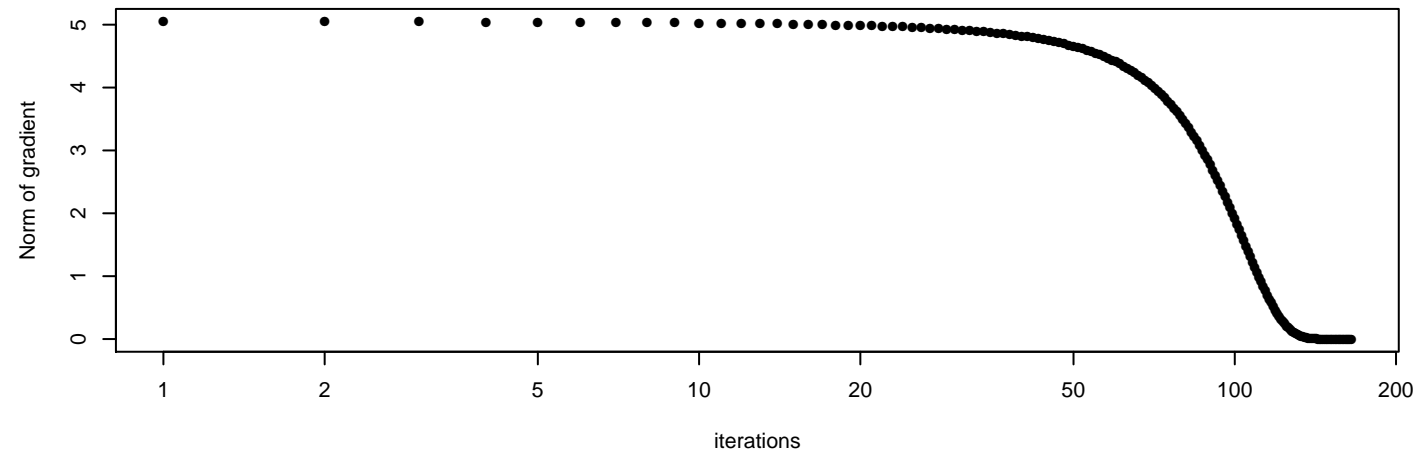
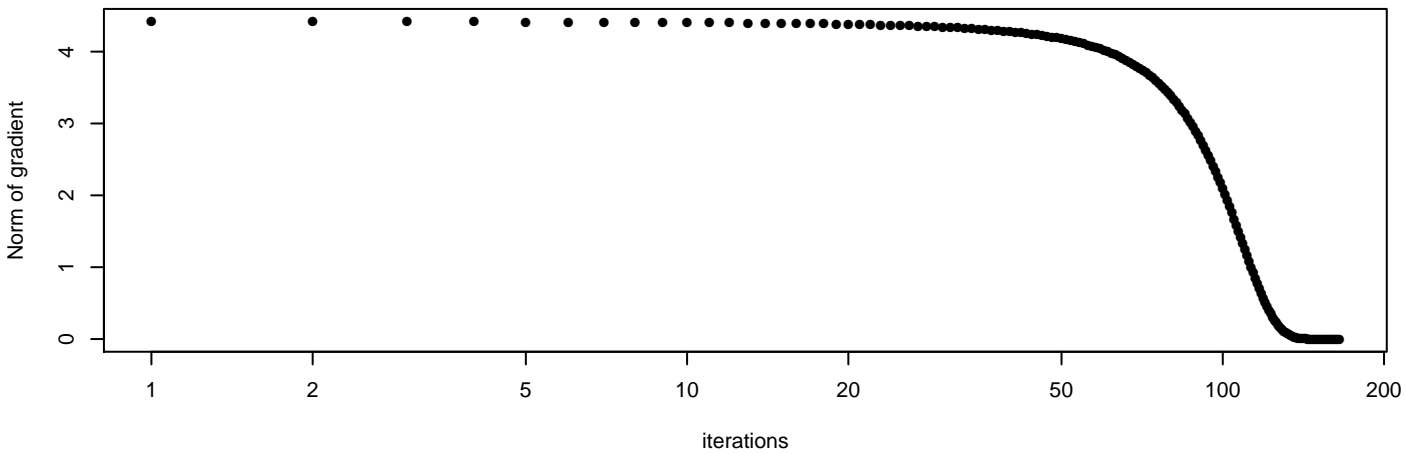
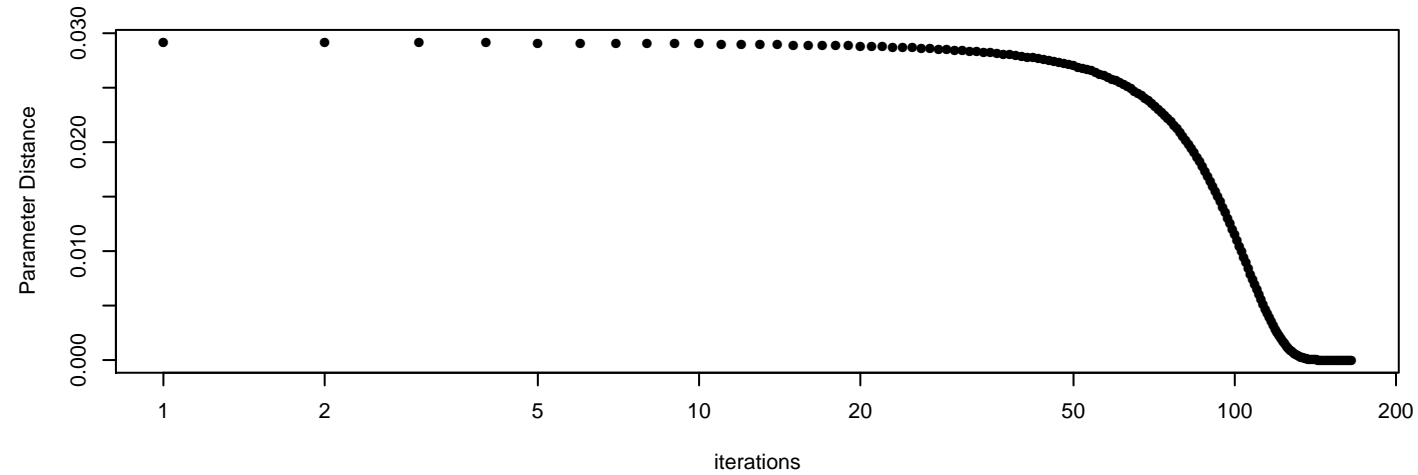
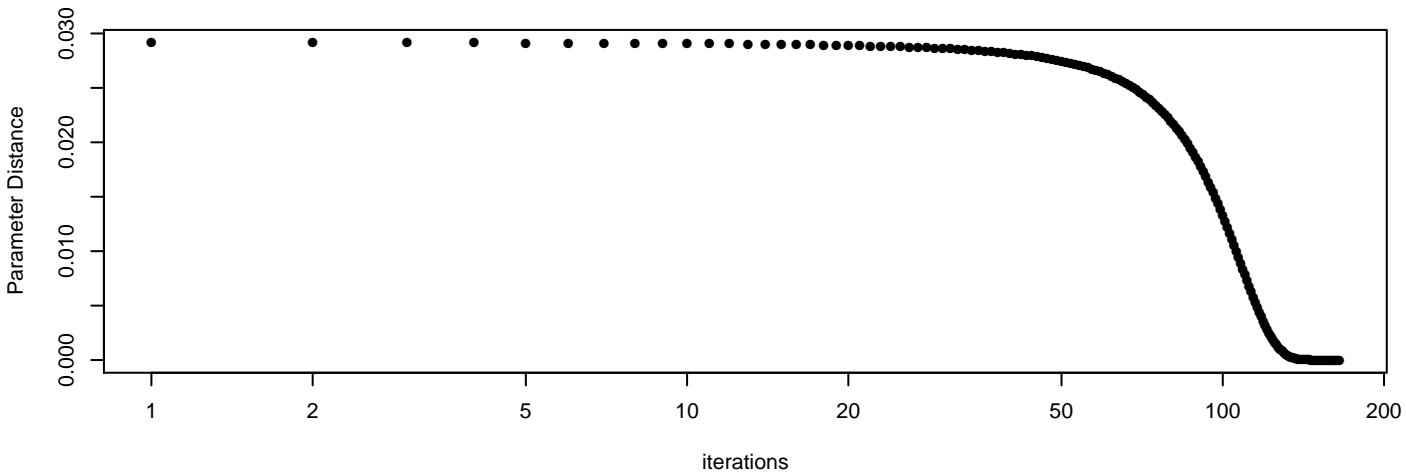
Negative Perturbation



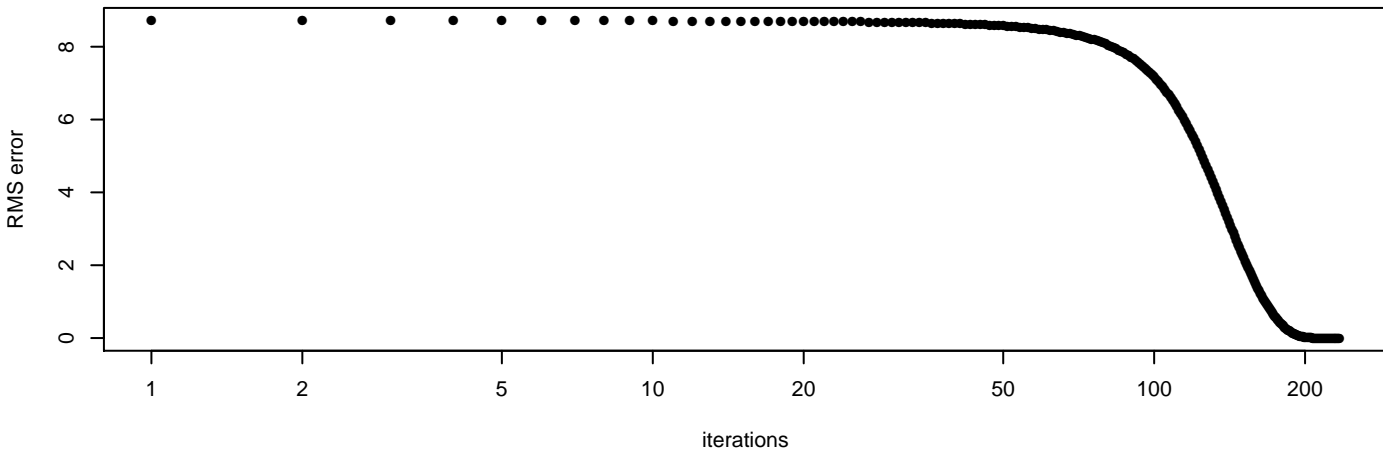
Parameter3



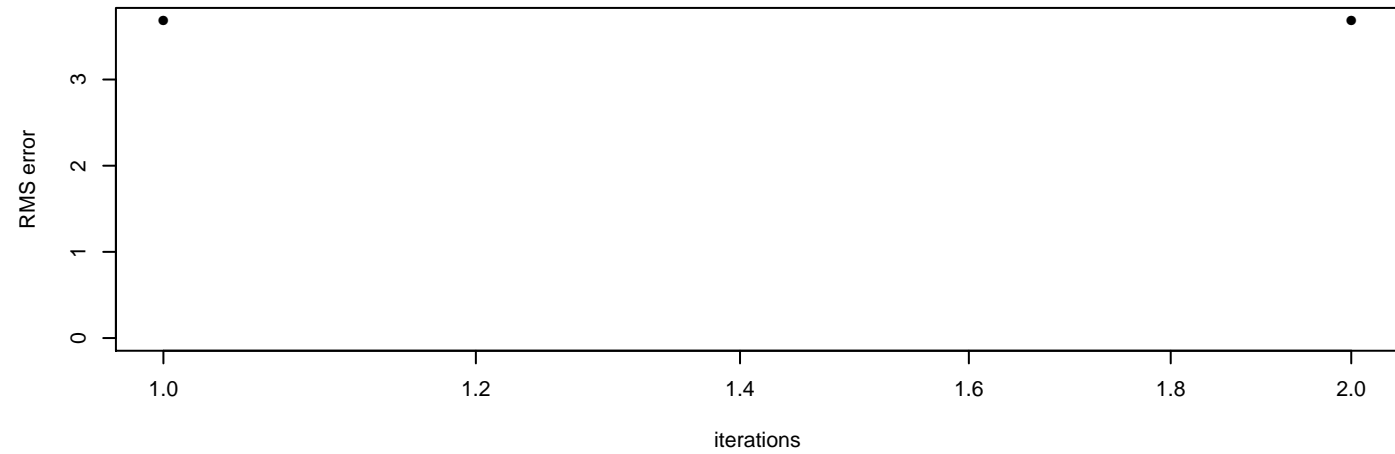
Positive Perturbation



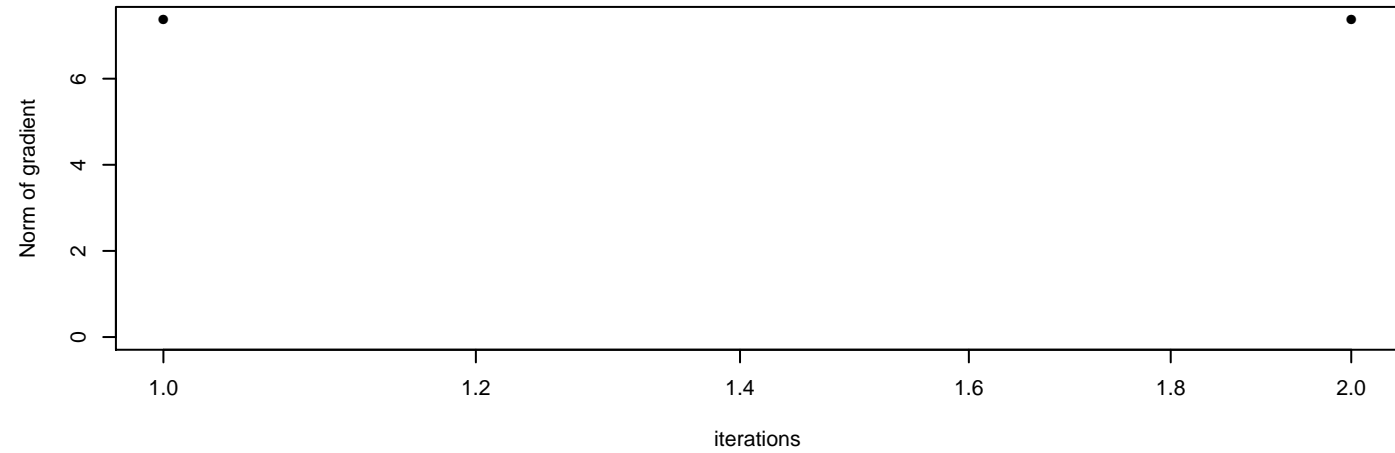
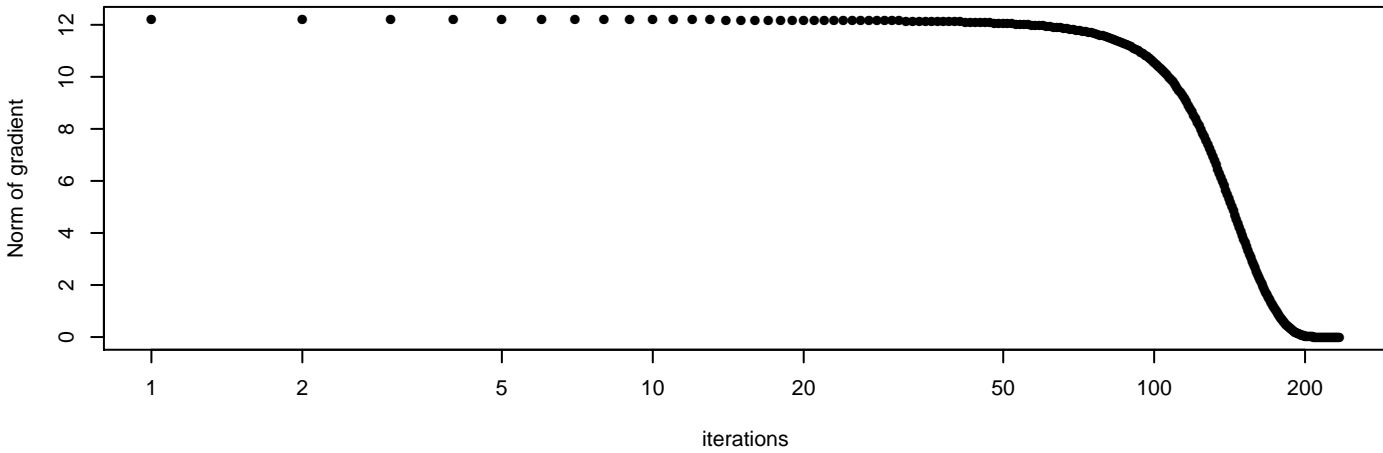
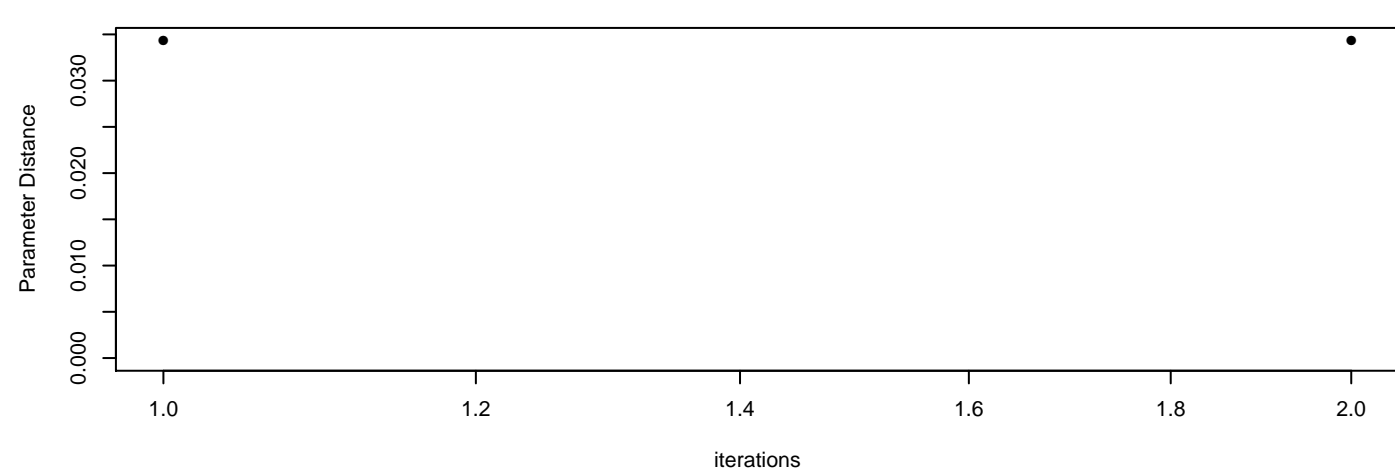
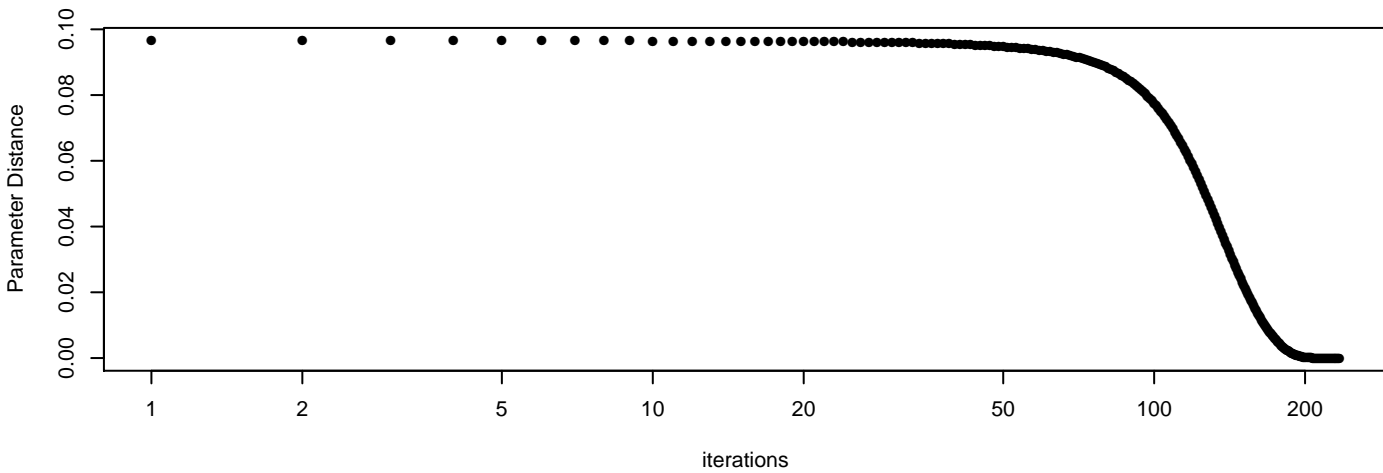
Negative Perturbation



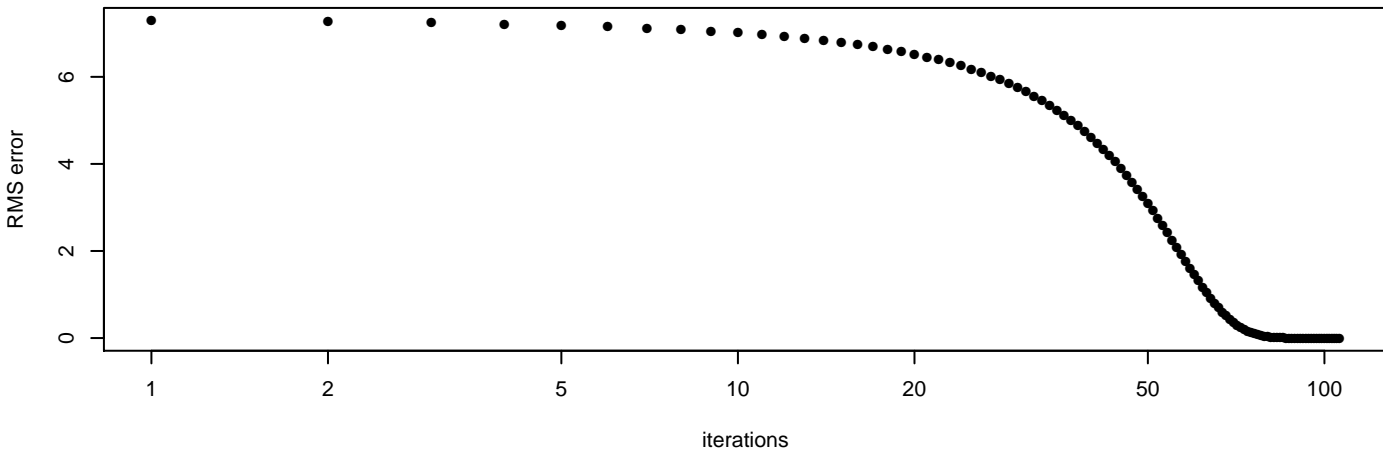
Parameter4



Positive Perturbation

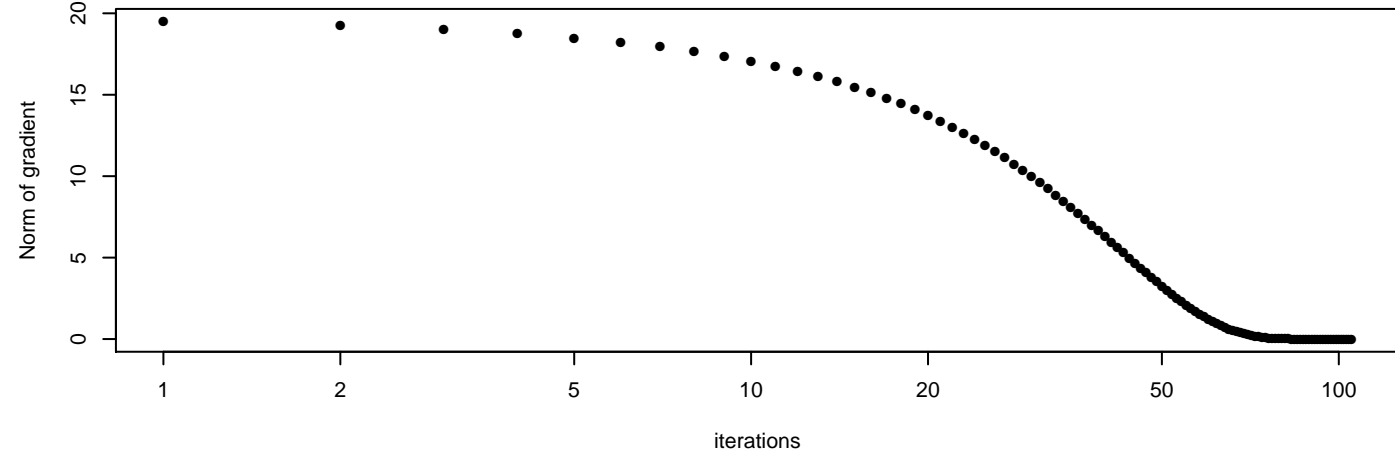
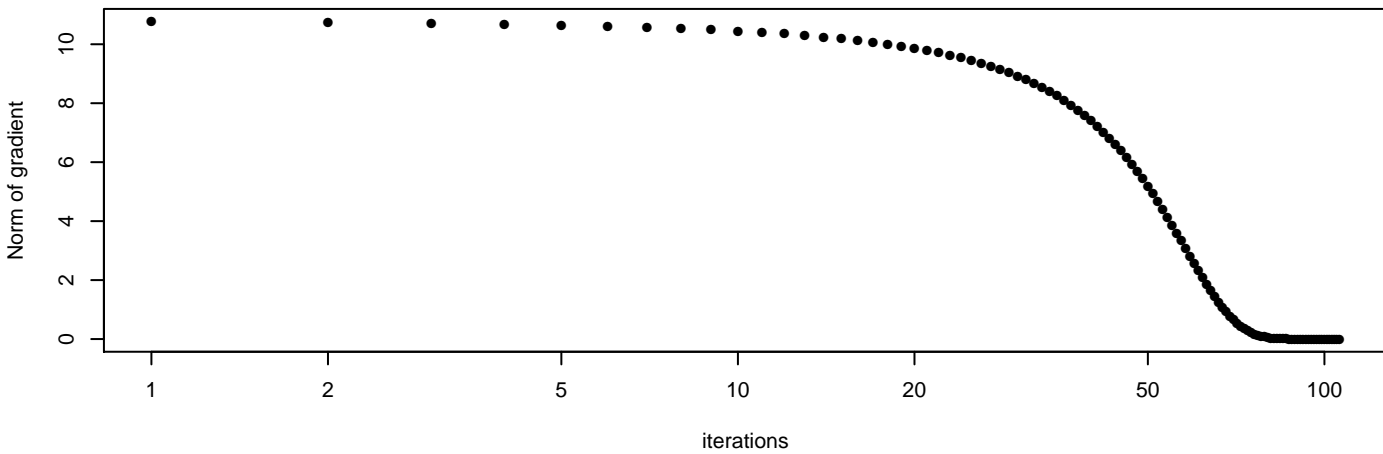
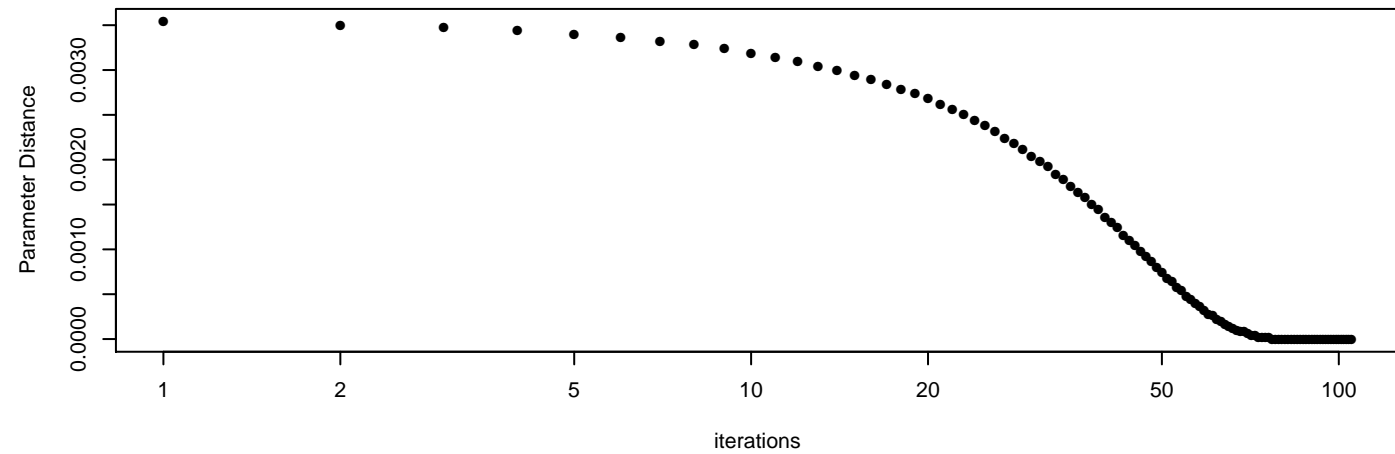
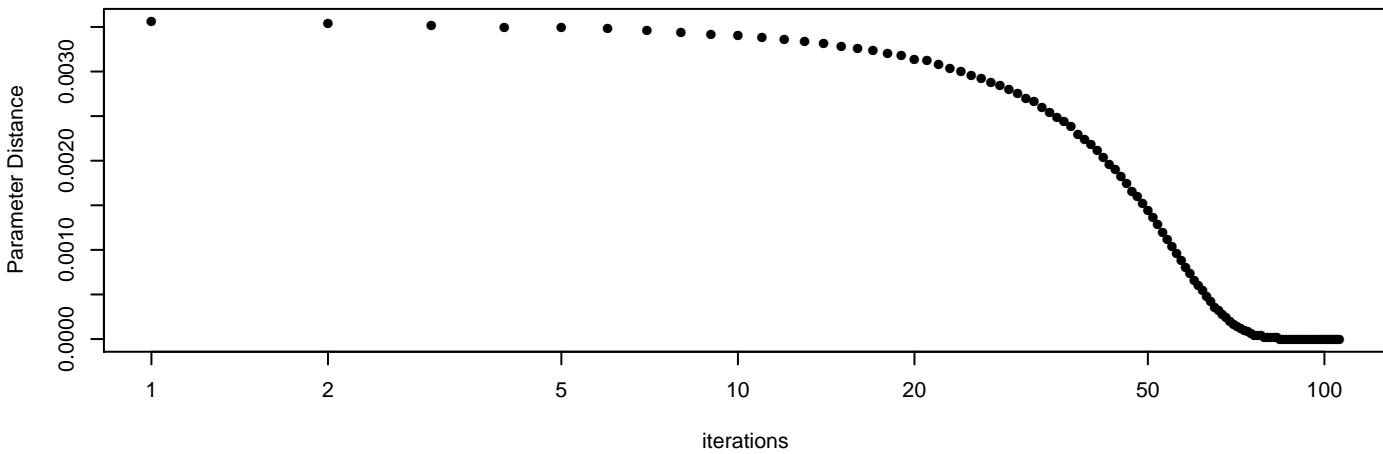
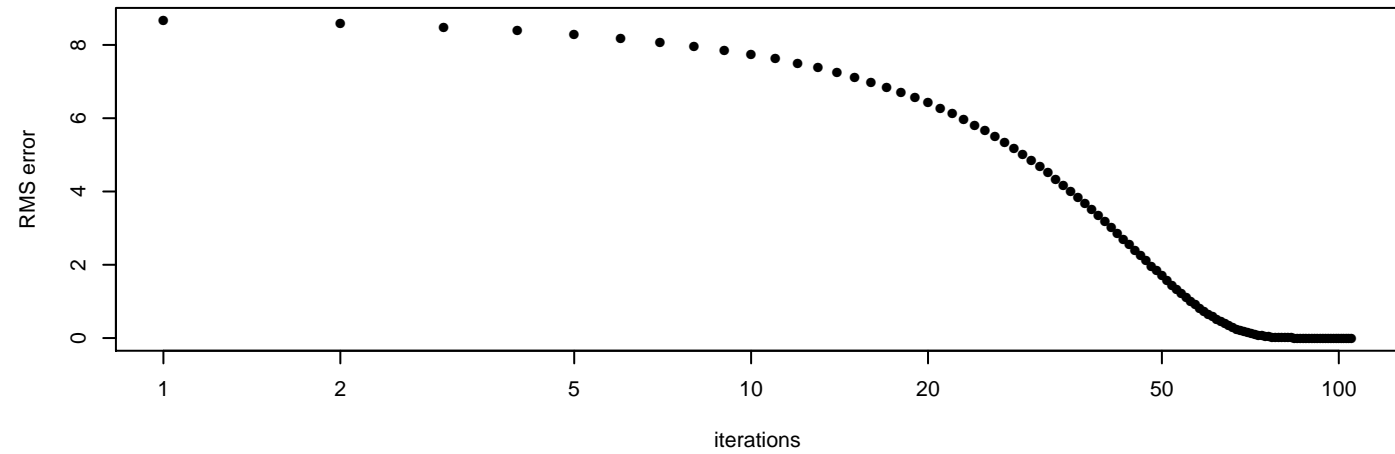


Negative Perturbation

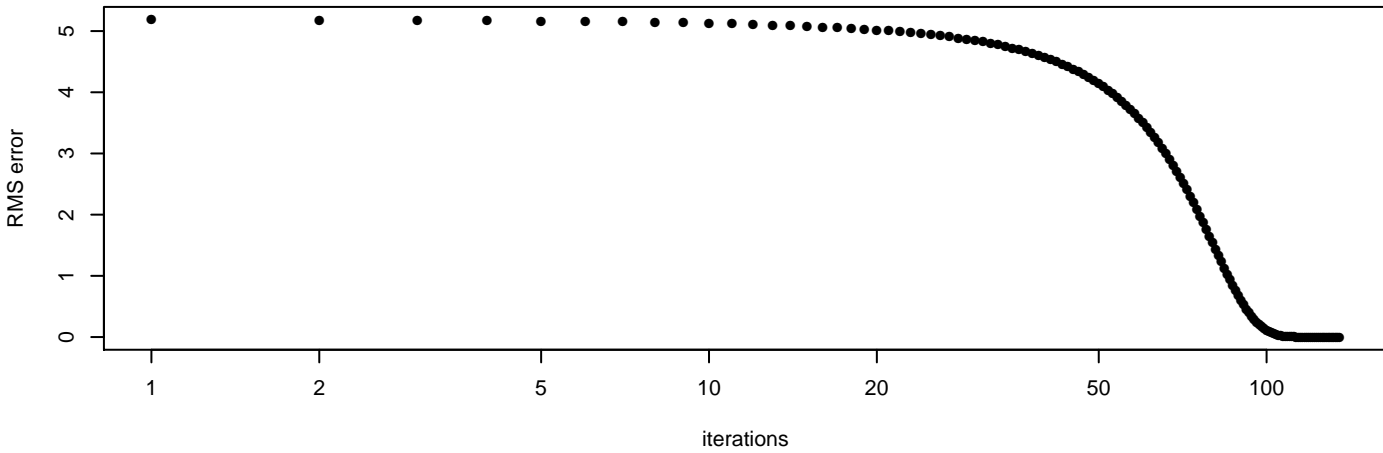


Parameter5

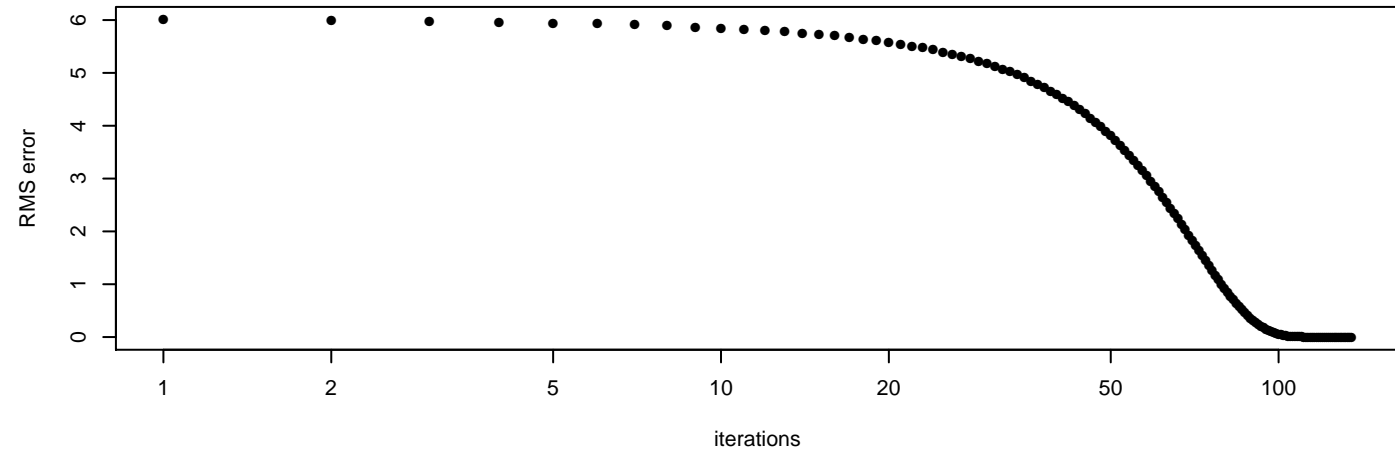
Positive Perturbation



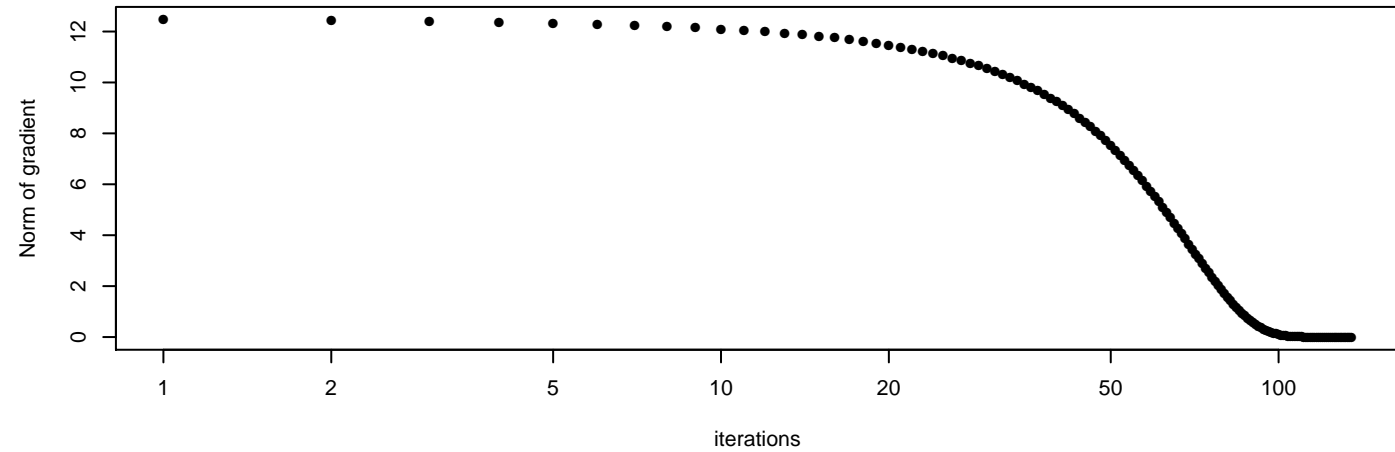
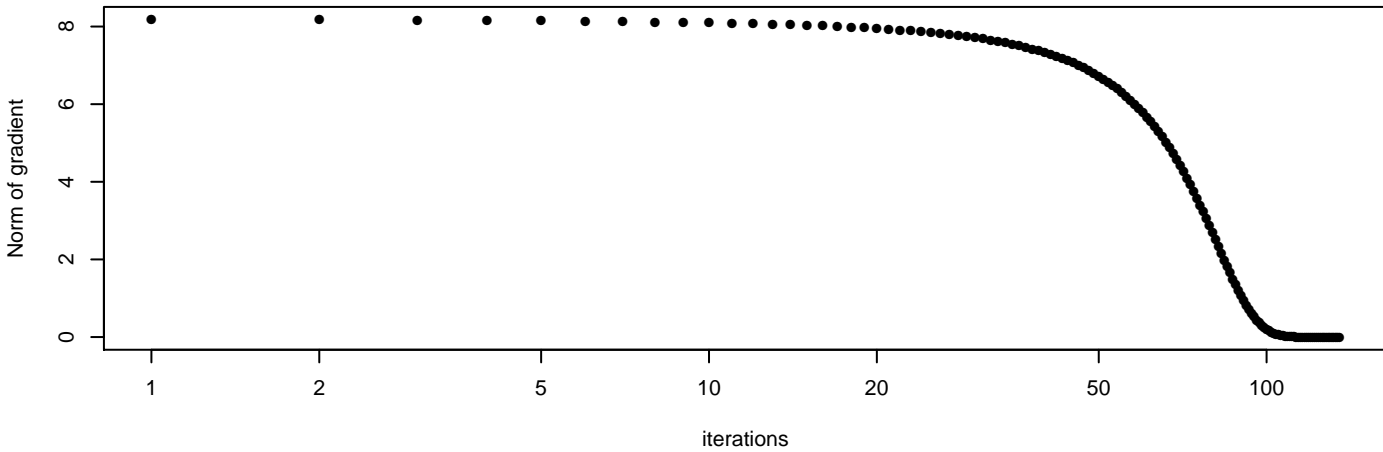
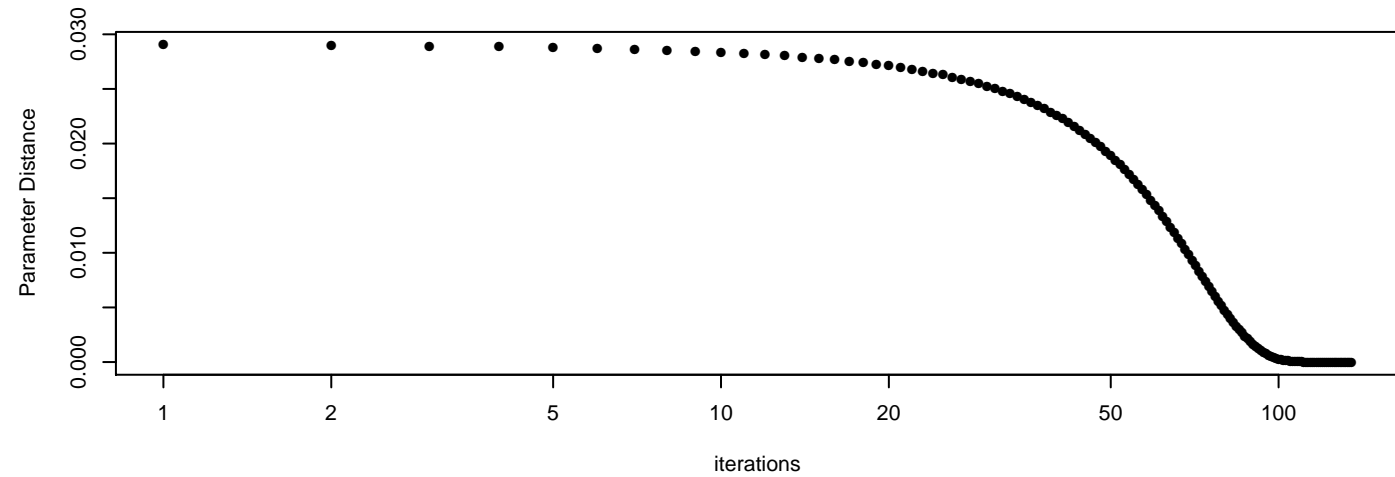
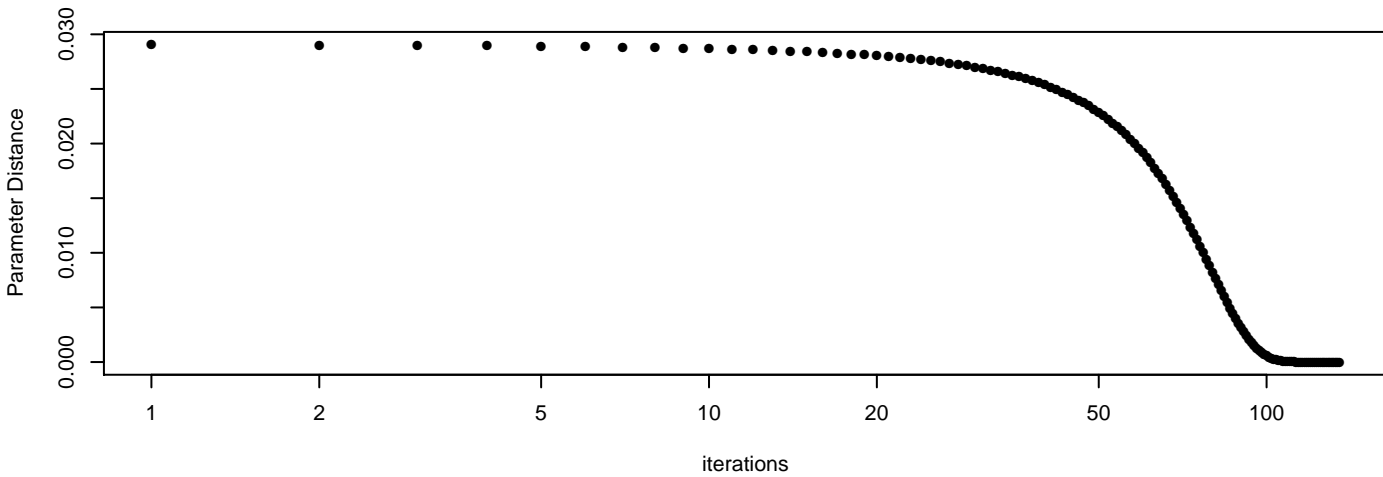
Negative Perturbation



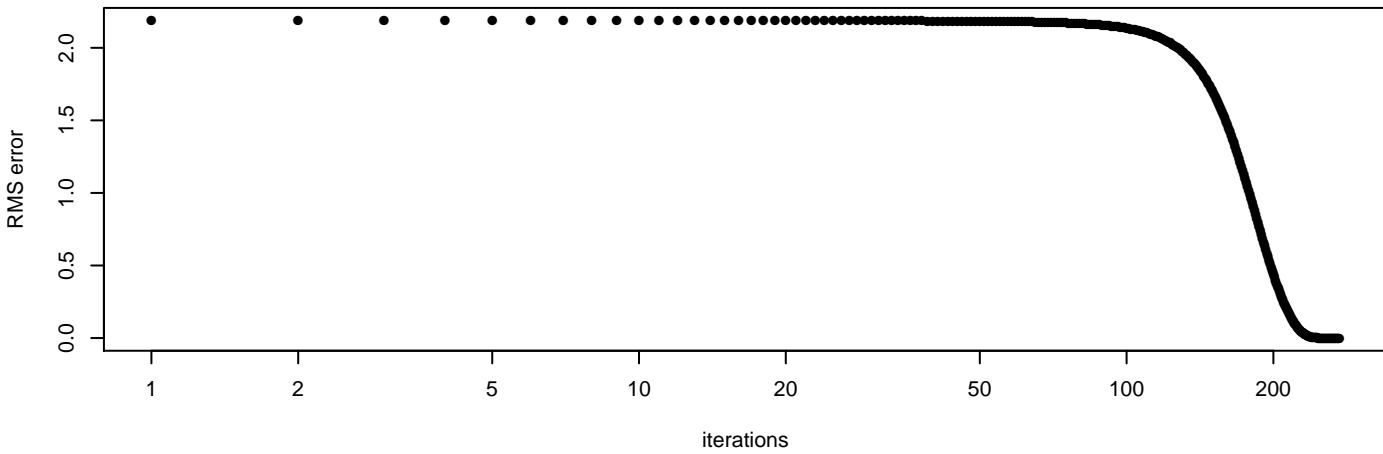
Parameter6



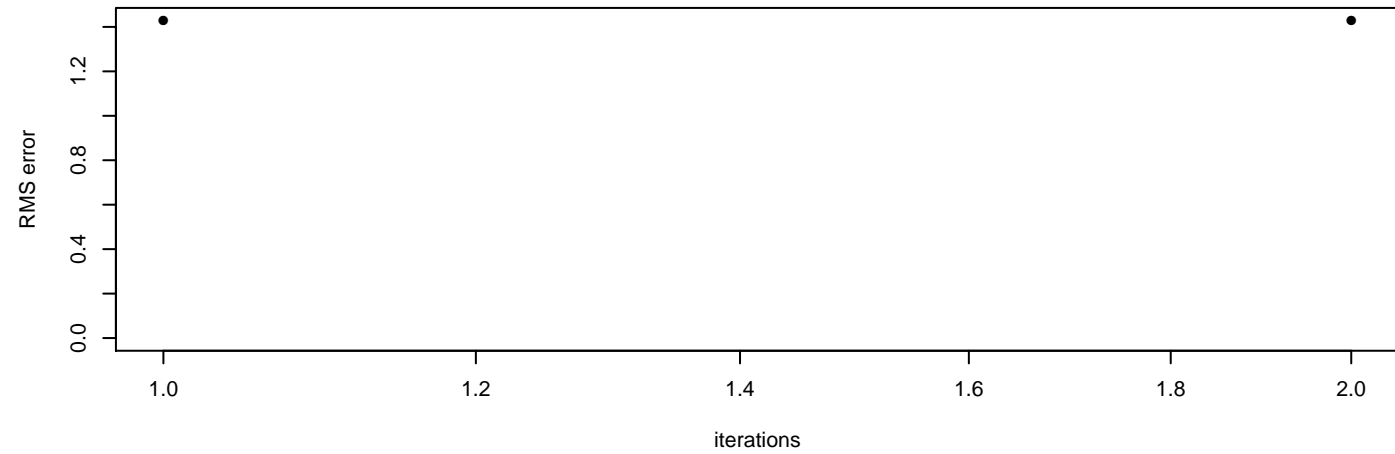
Positive Perturbation



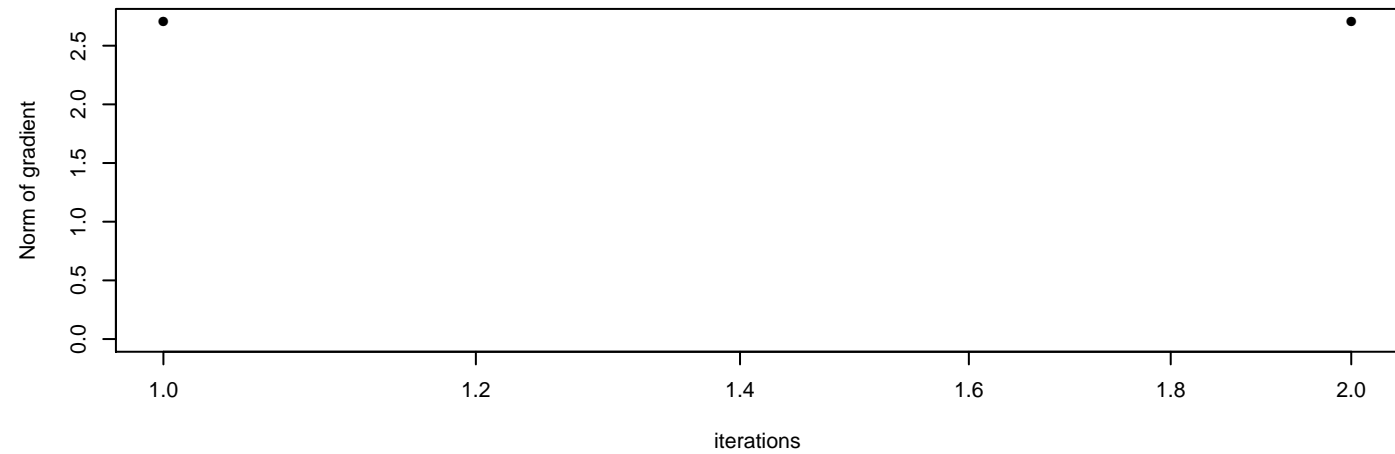
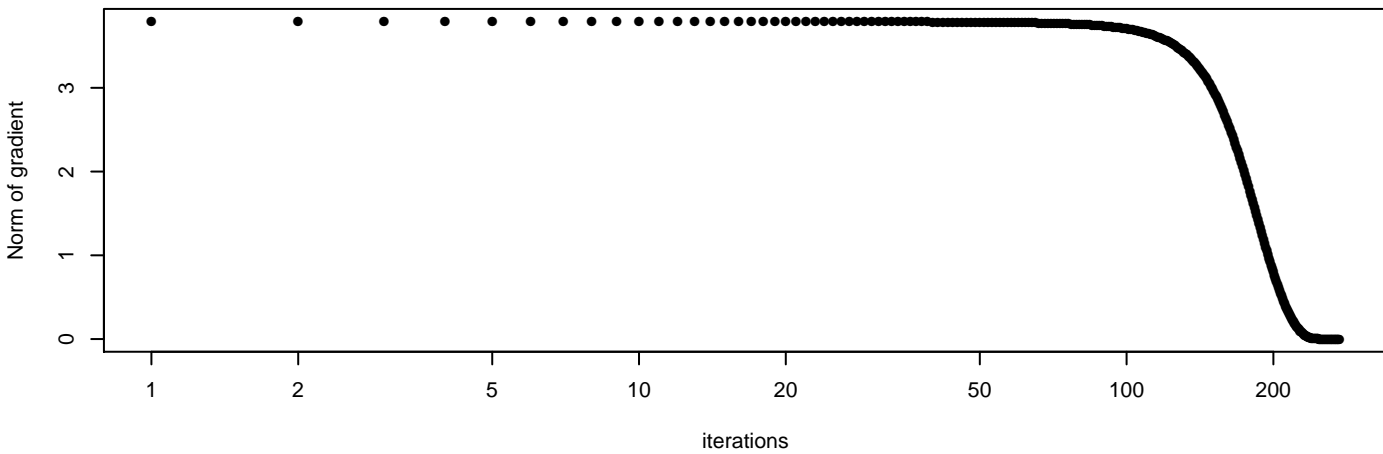
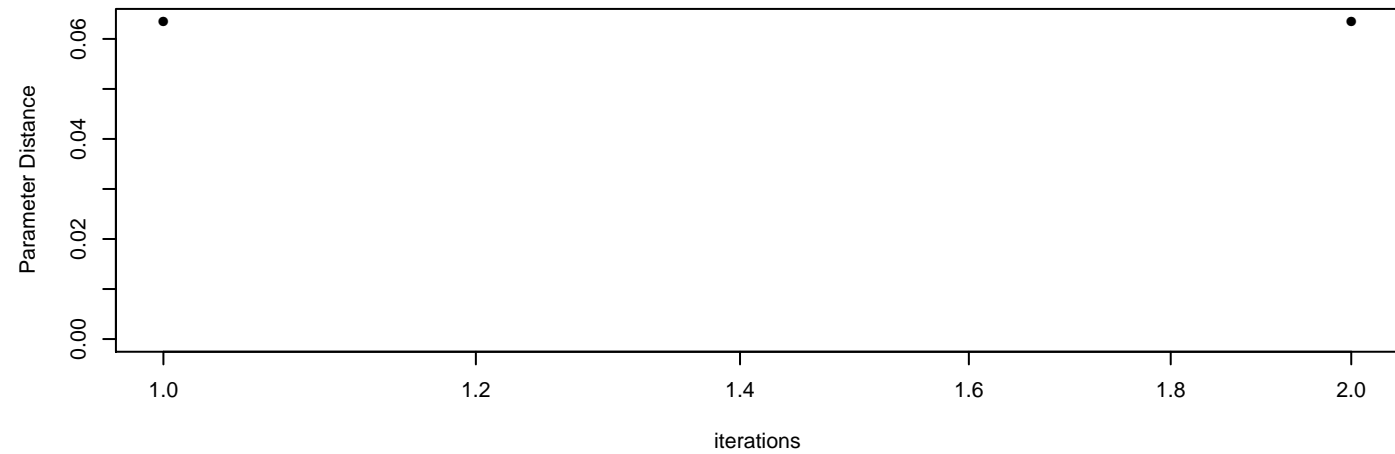
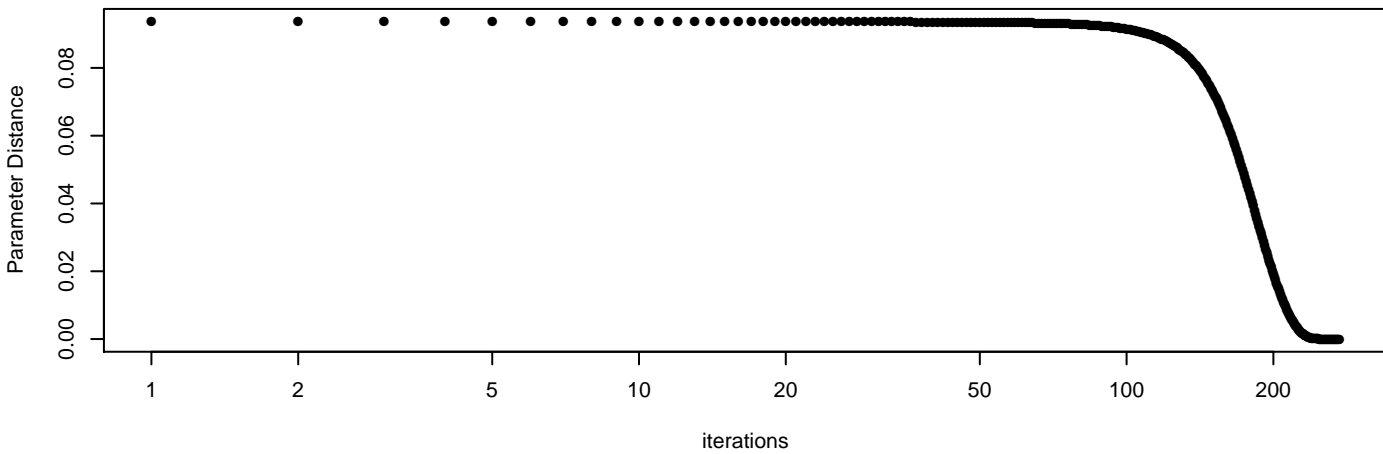
Negative Perturbation



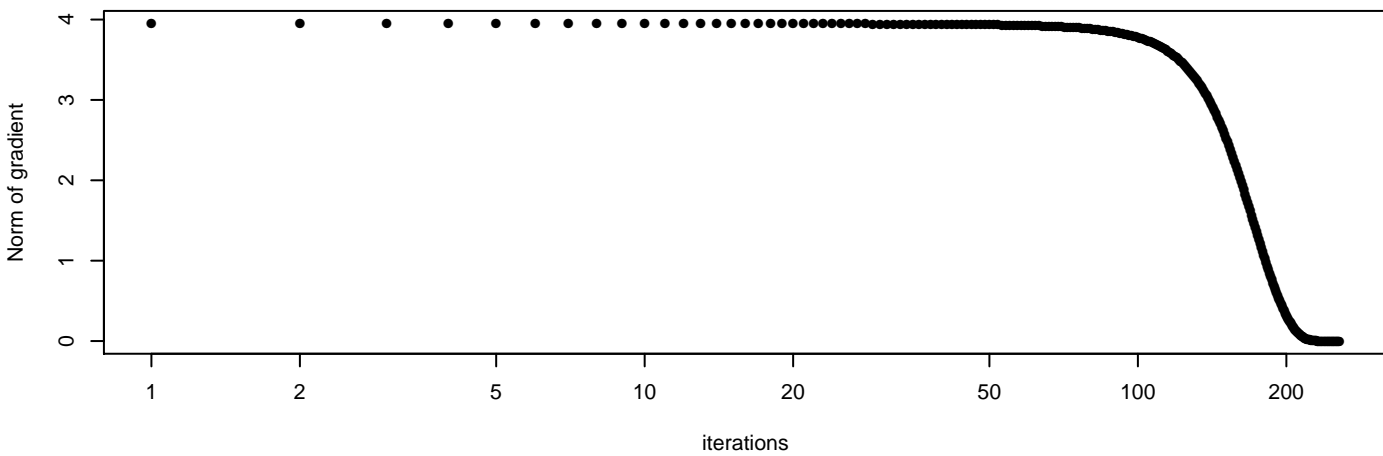
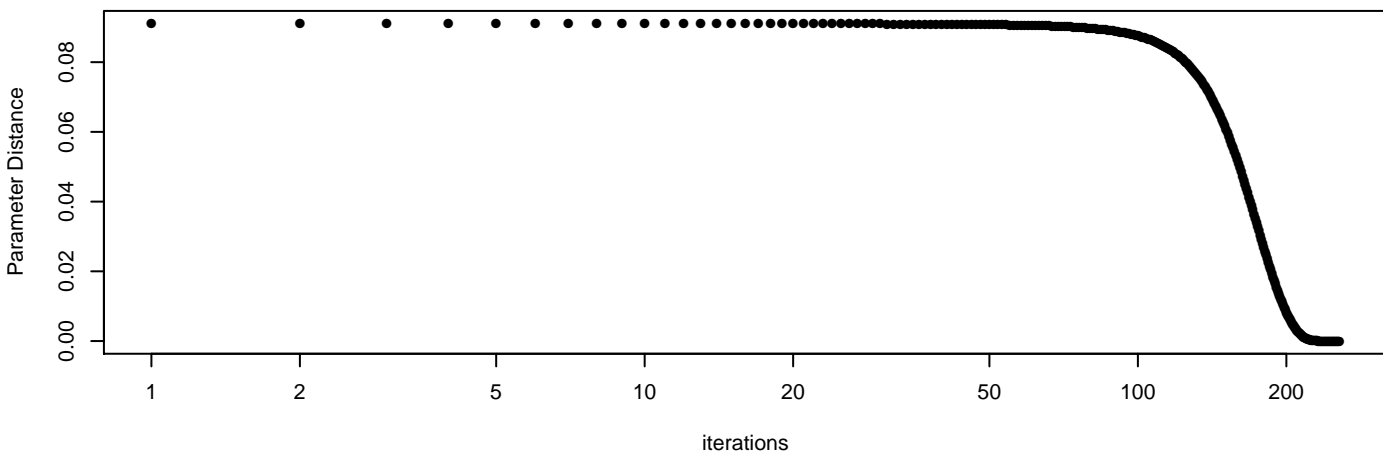
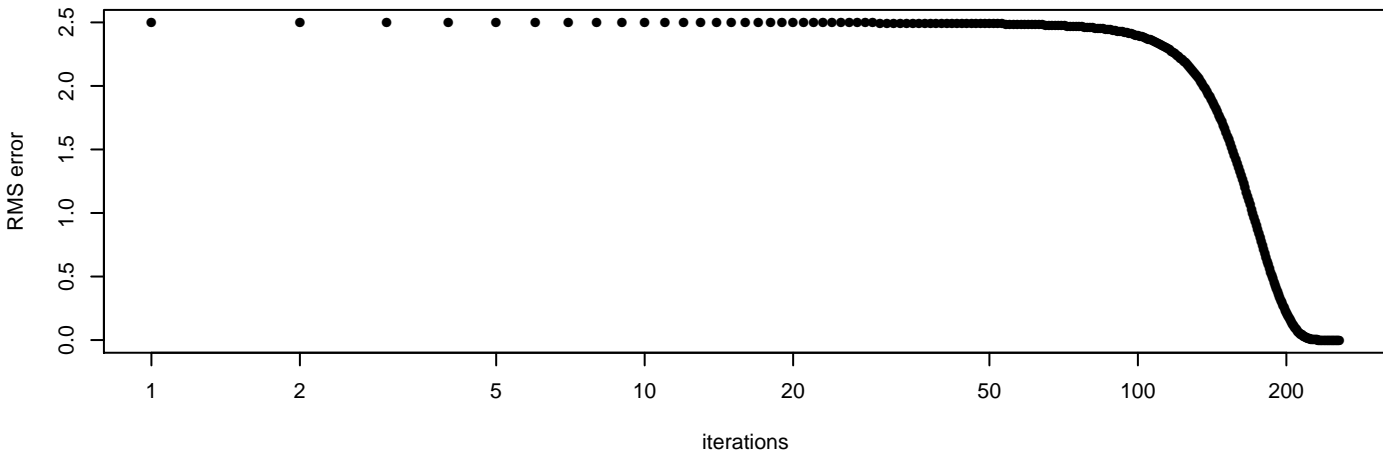
Parameter7



Positive Perturbation

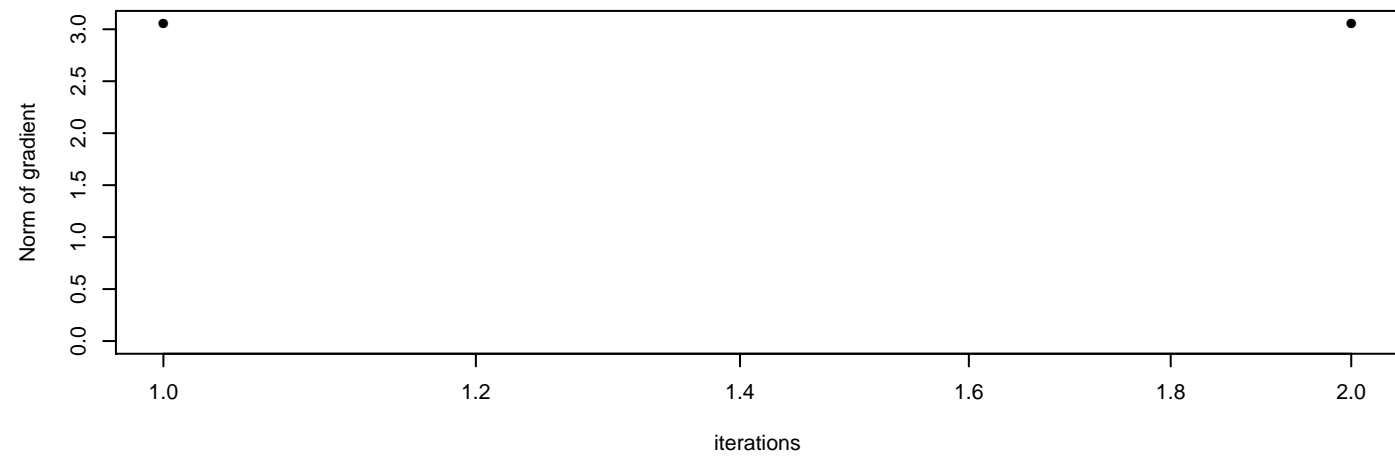
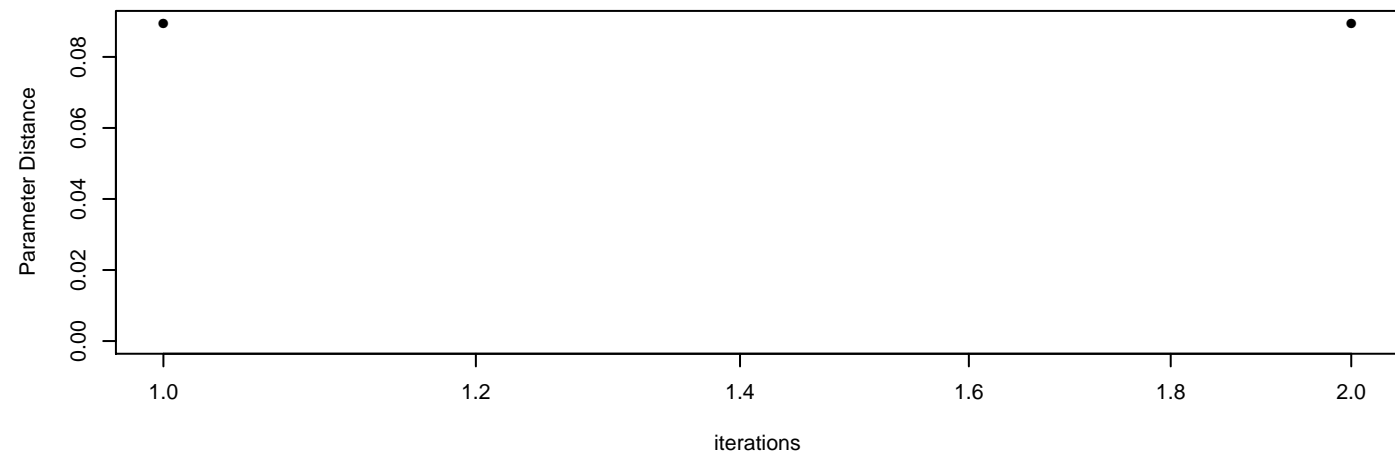
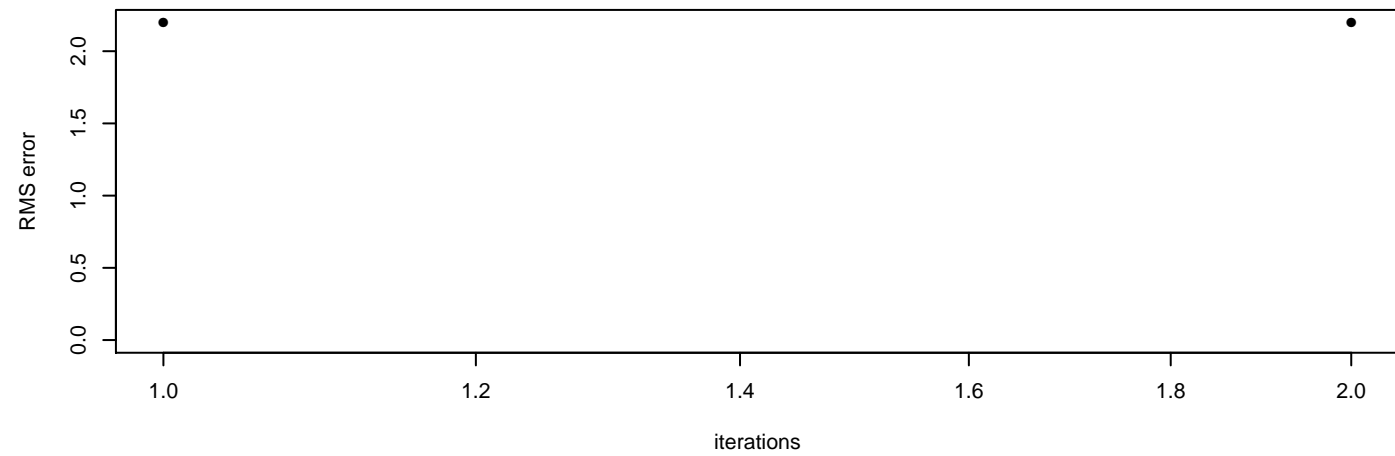


Negative Perturbation

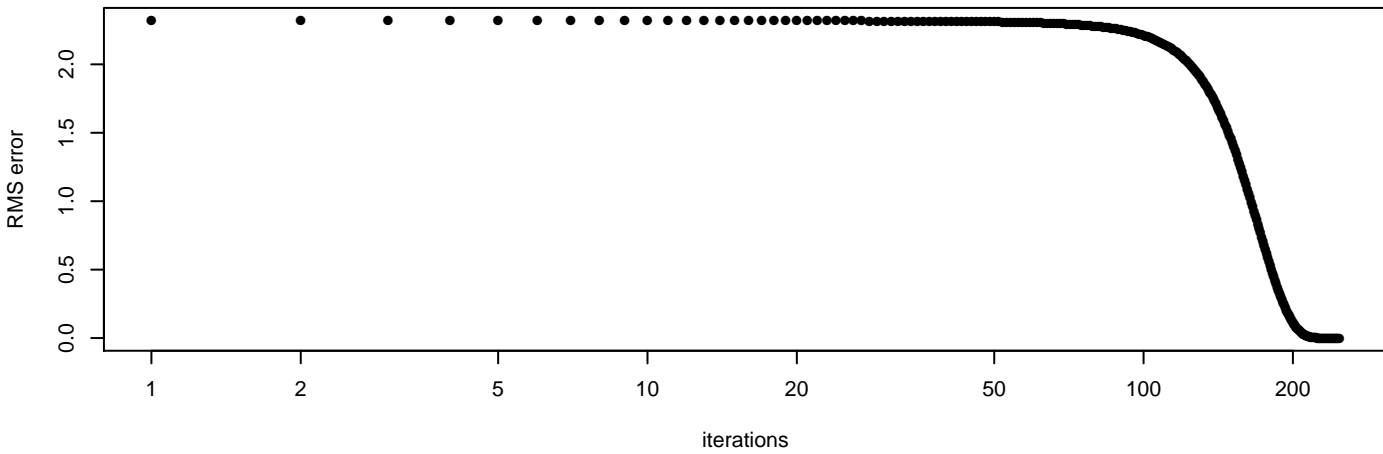


Parameter8

Positive Perturbation

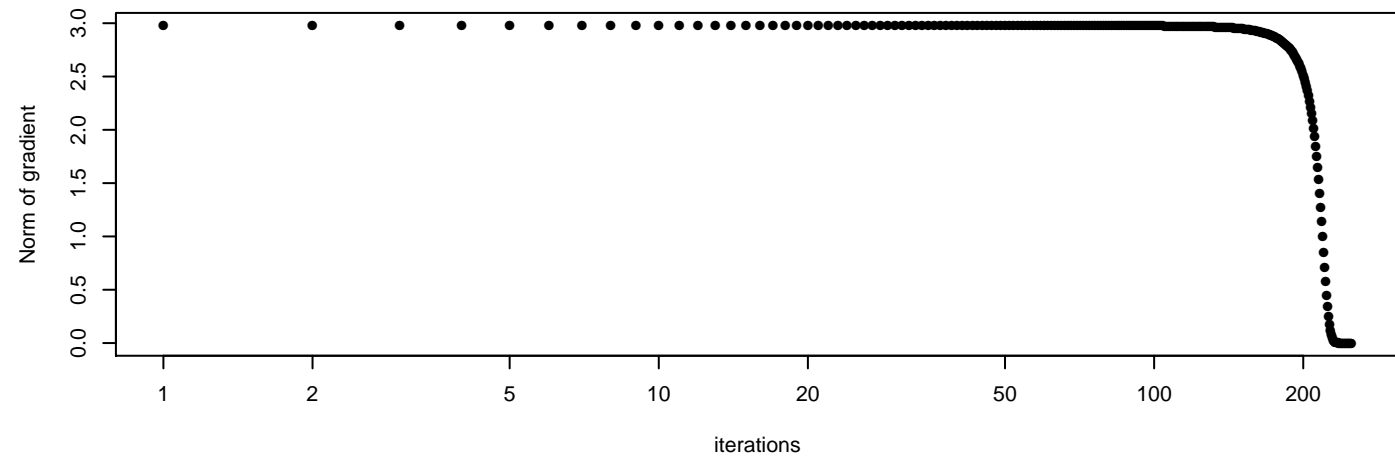
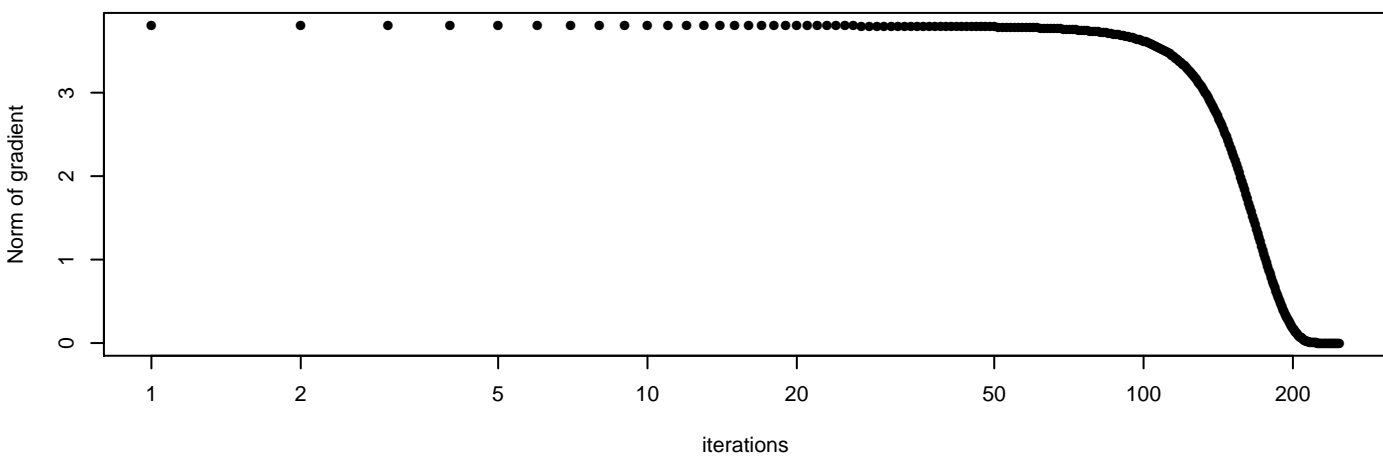
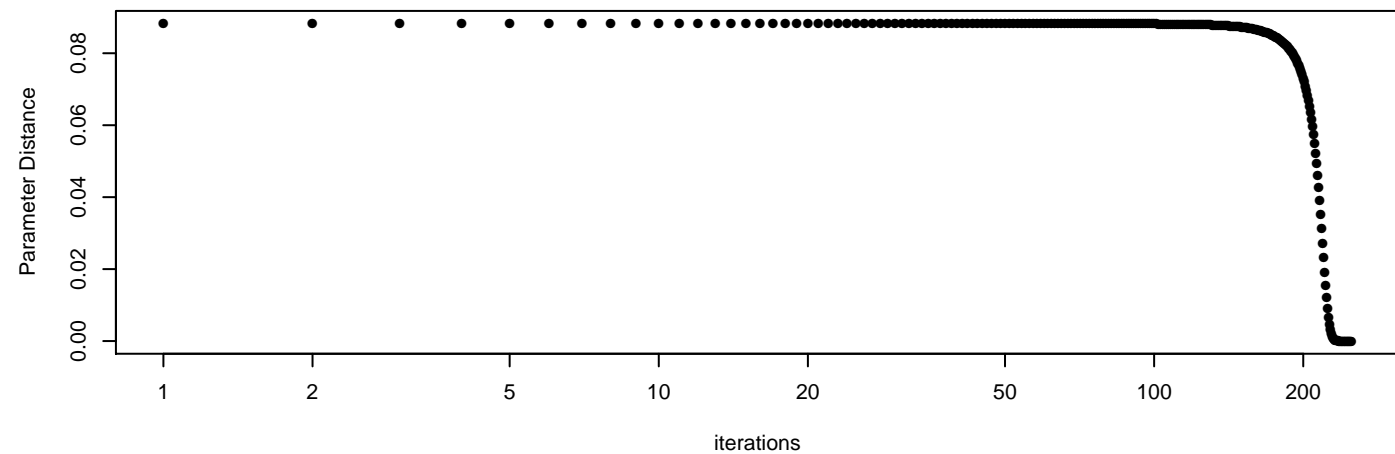
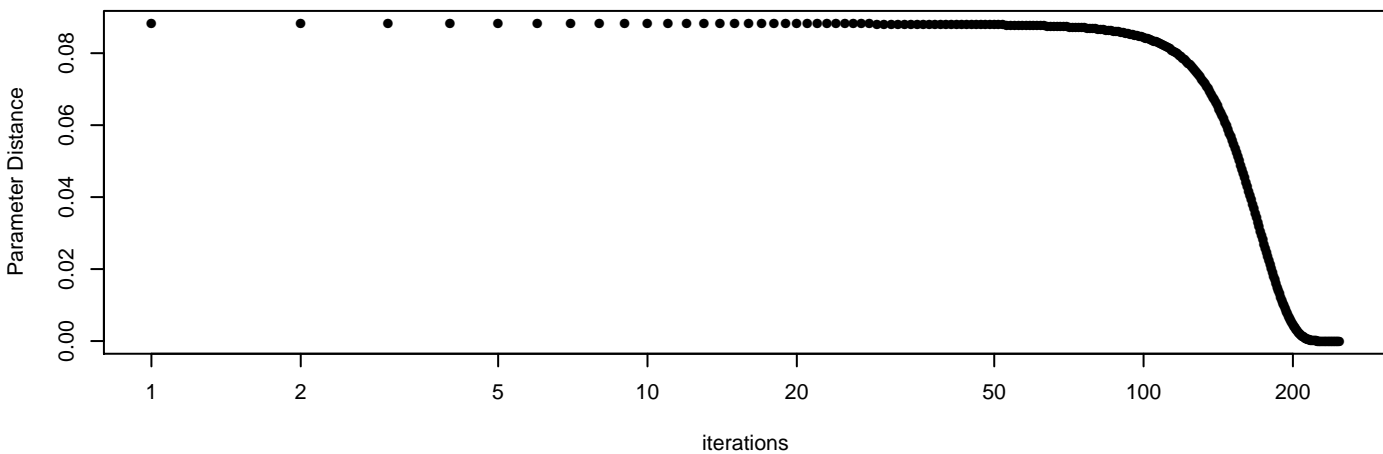
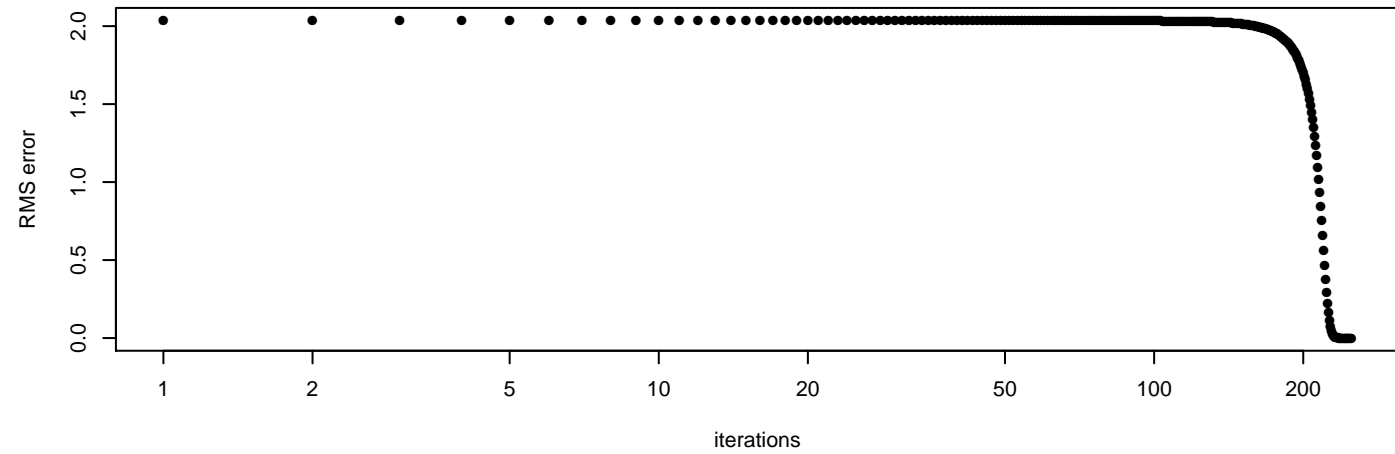


Negative Perturbation



Parameter9

Positive Perturbation

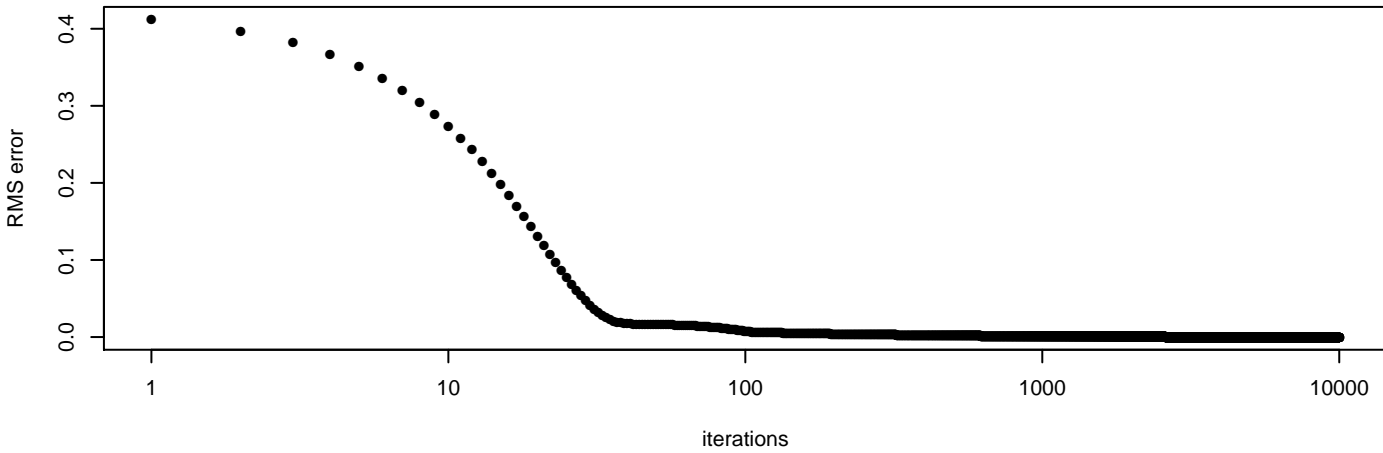


Convergence of a parameter set where each parameter is perturbed by * 1% * individually.

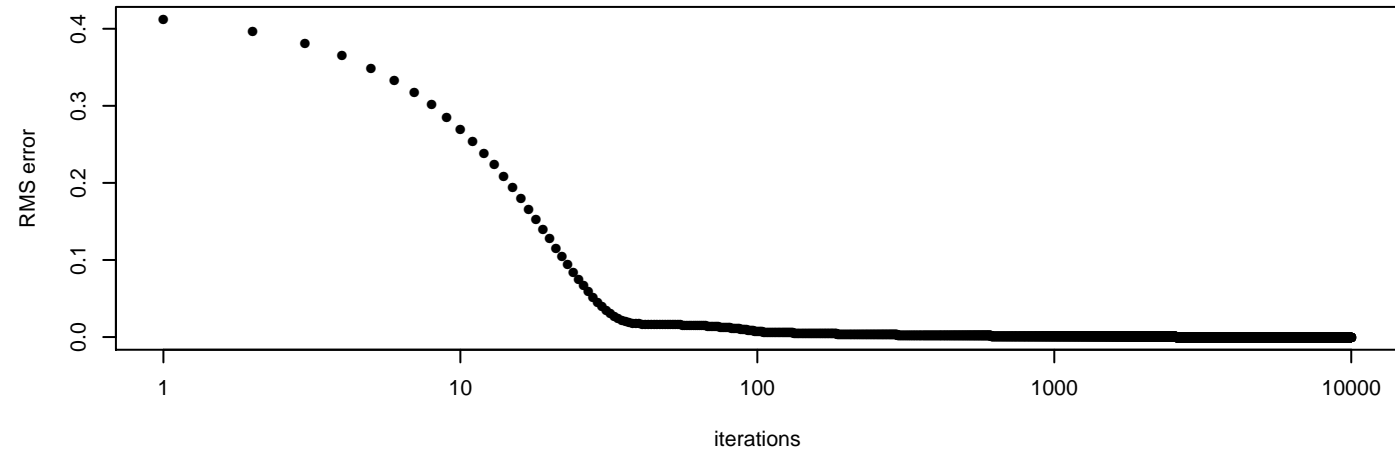
Gradient descent is executed with all parameters * free* to be changed by GD.

Gradient descent was ran in * normal * mode to take the value of the gradient.

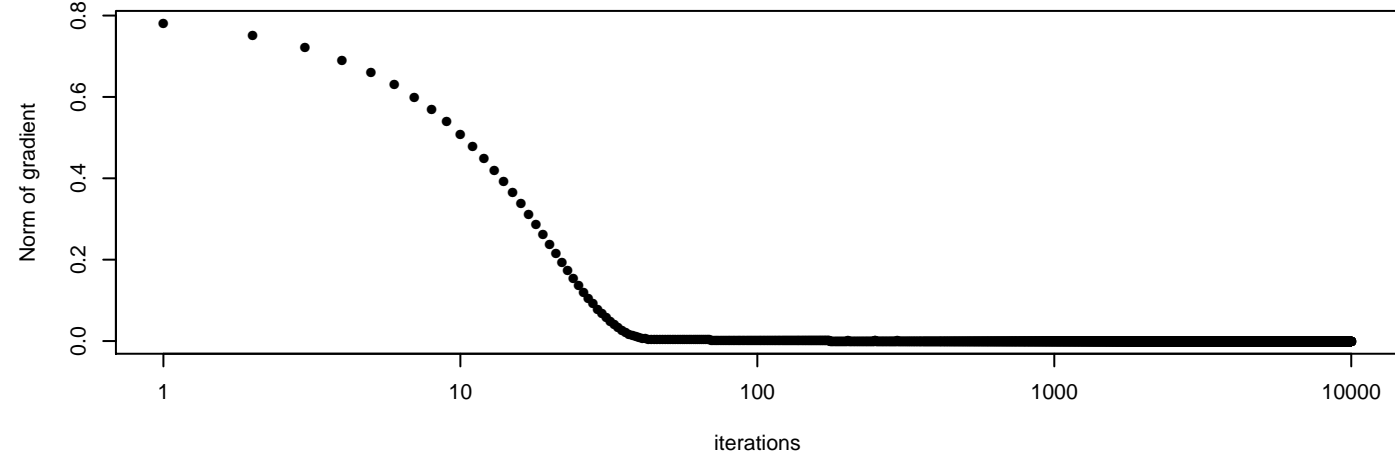
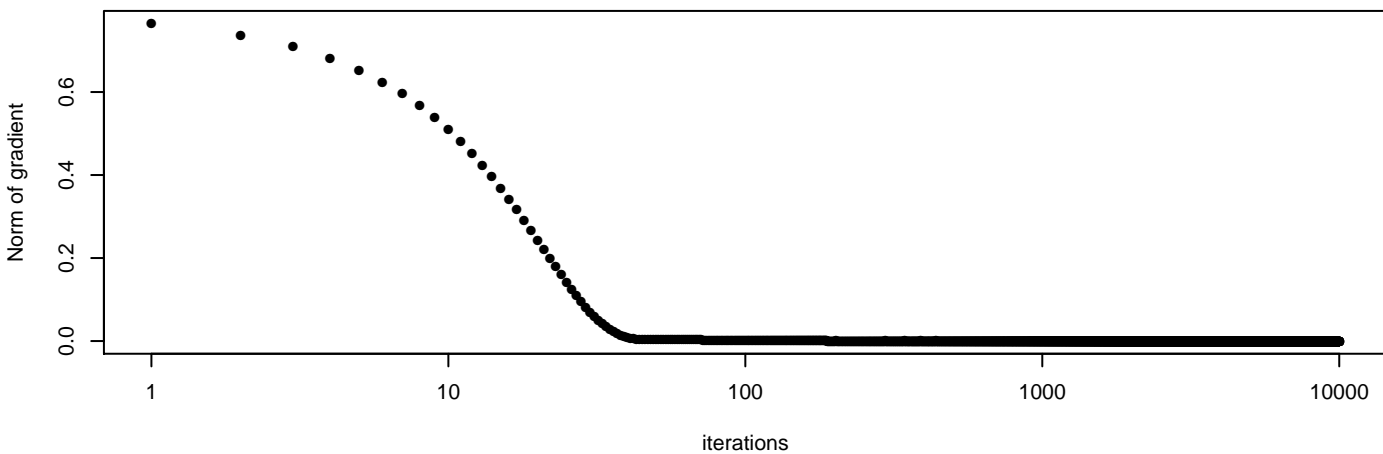
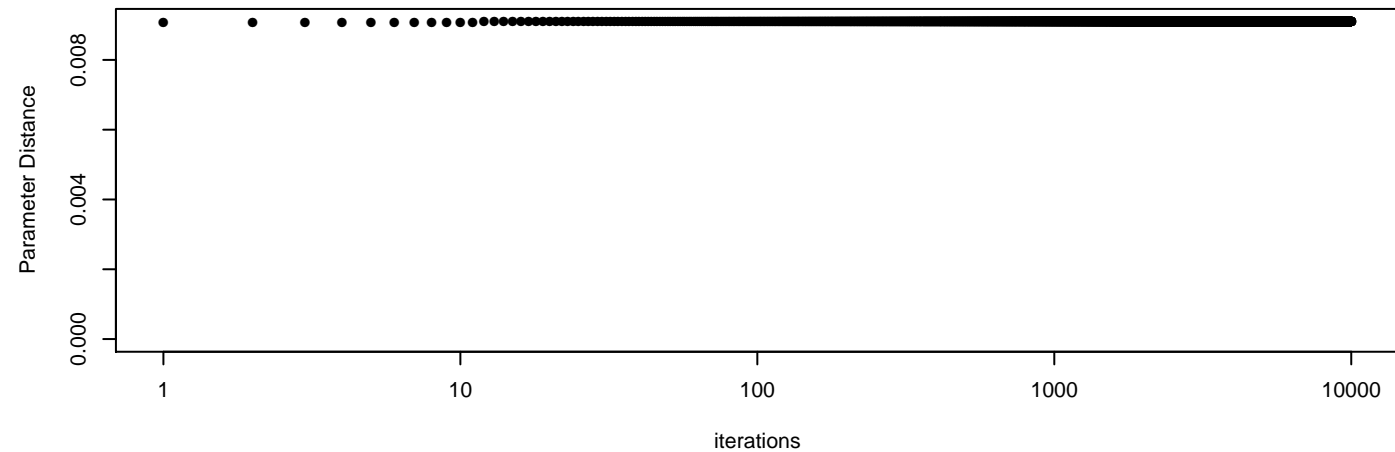
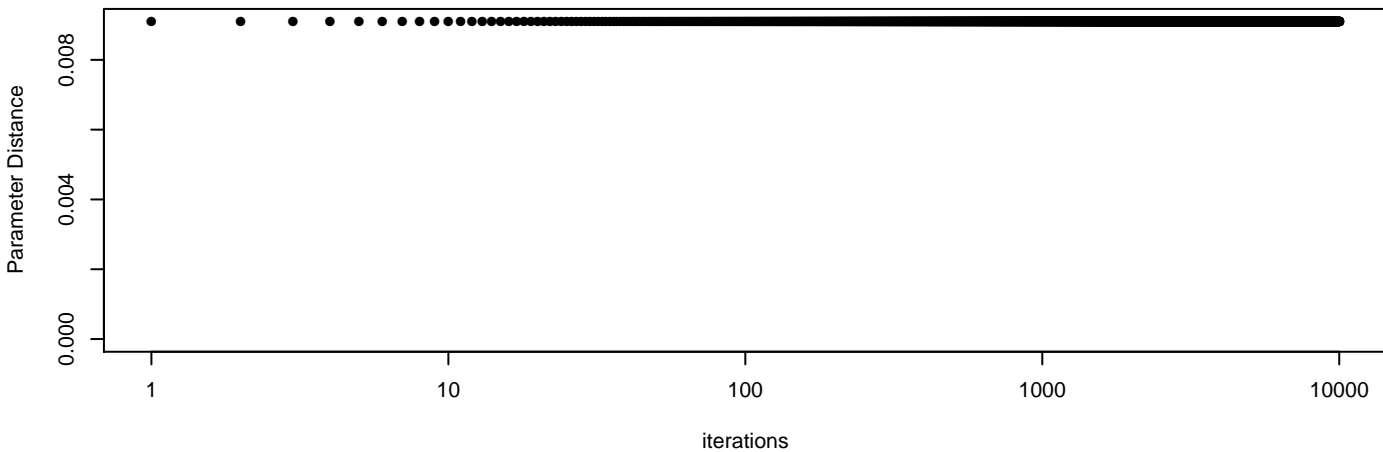
Negative Perturbation



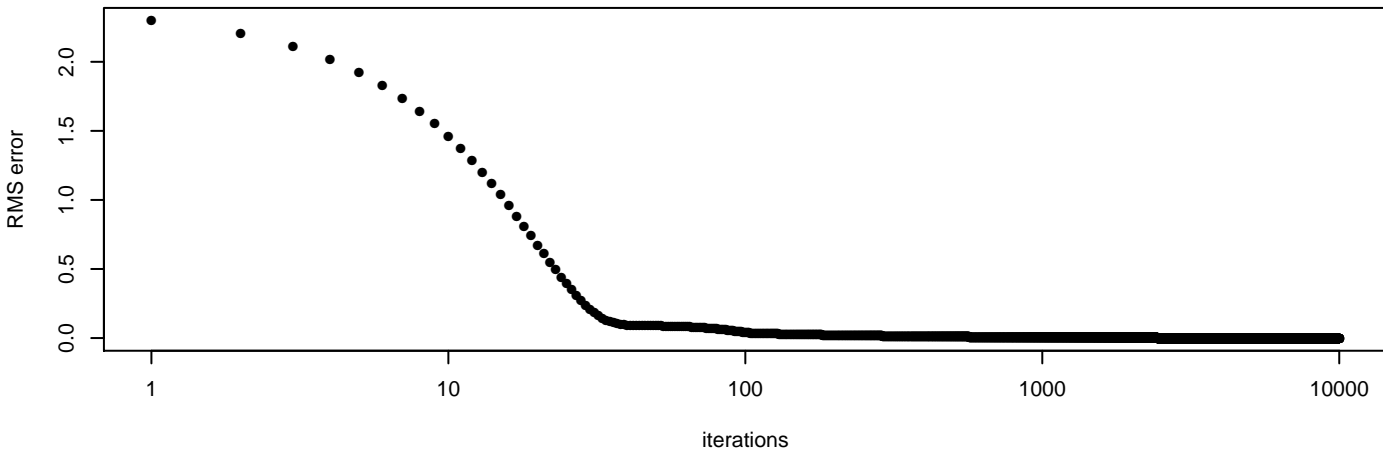
Parameter0



Positive Perturbation

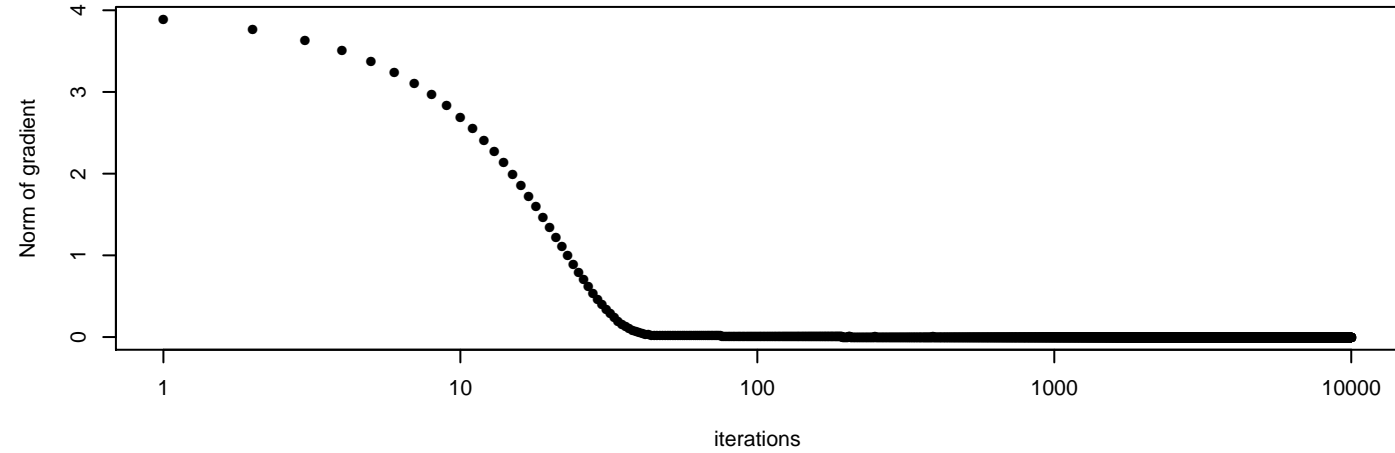
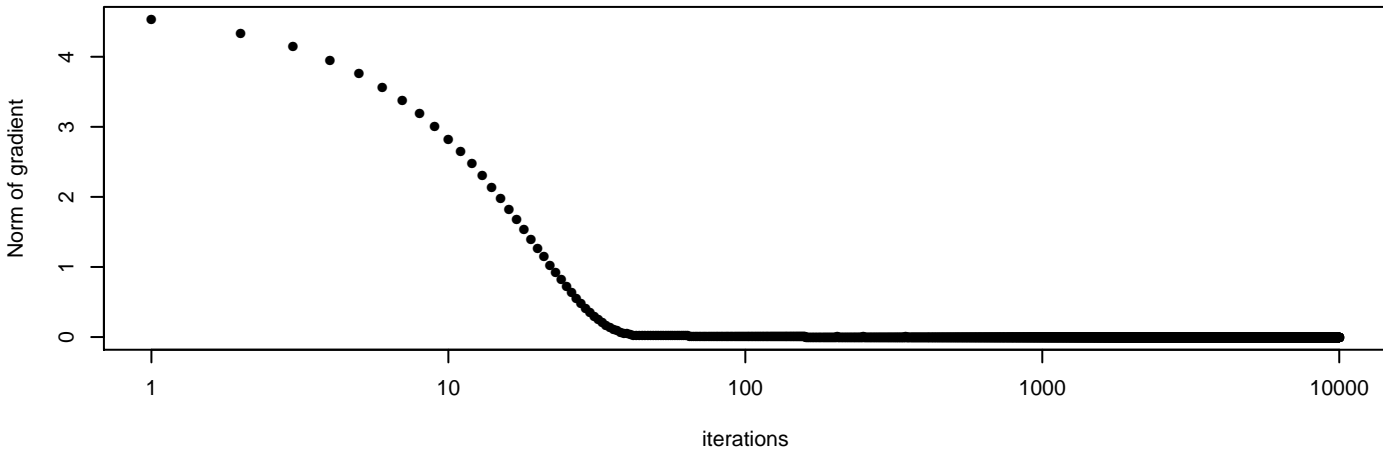
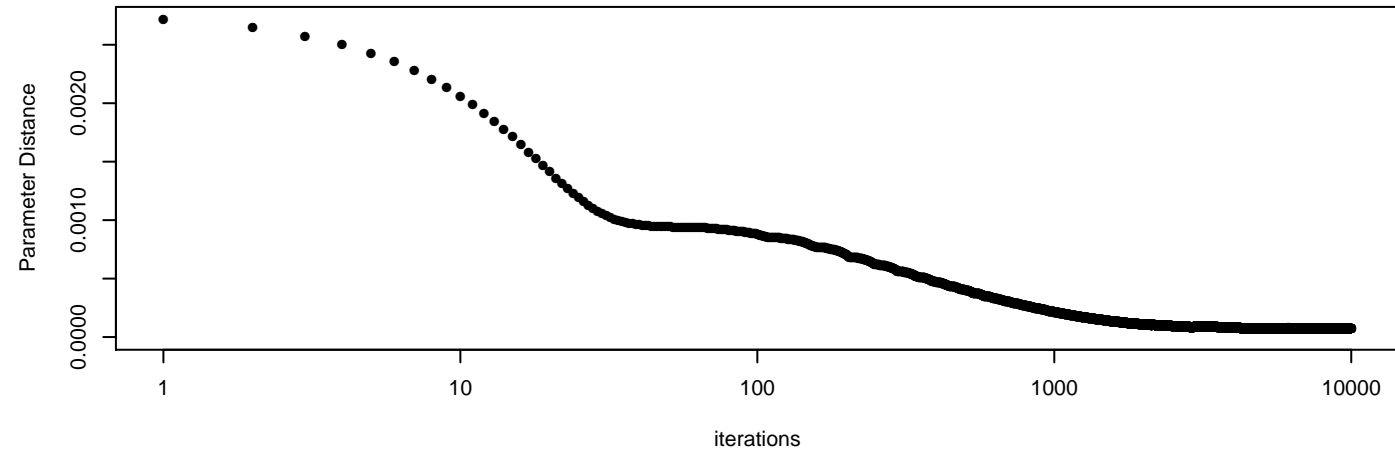
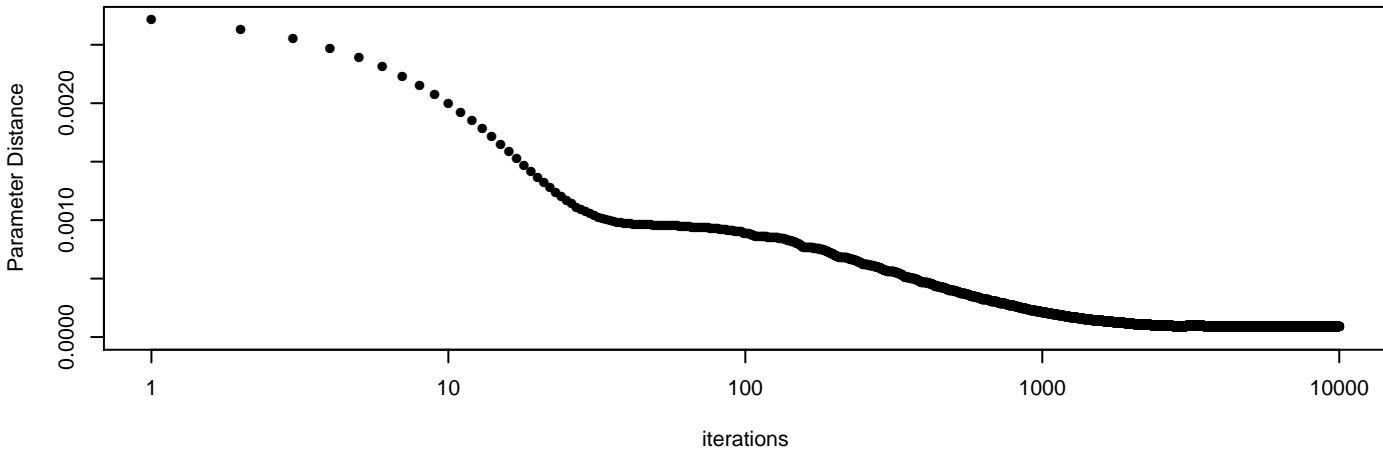
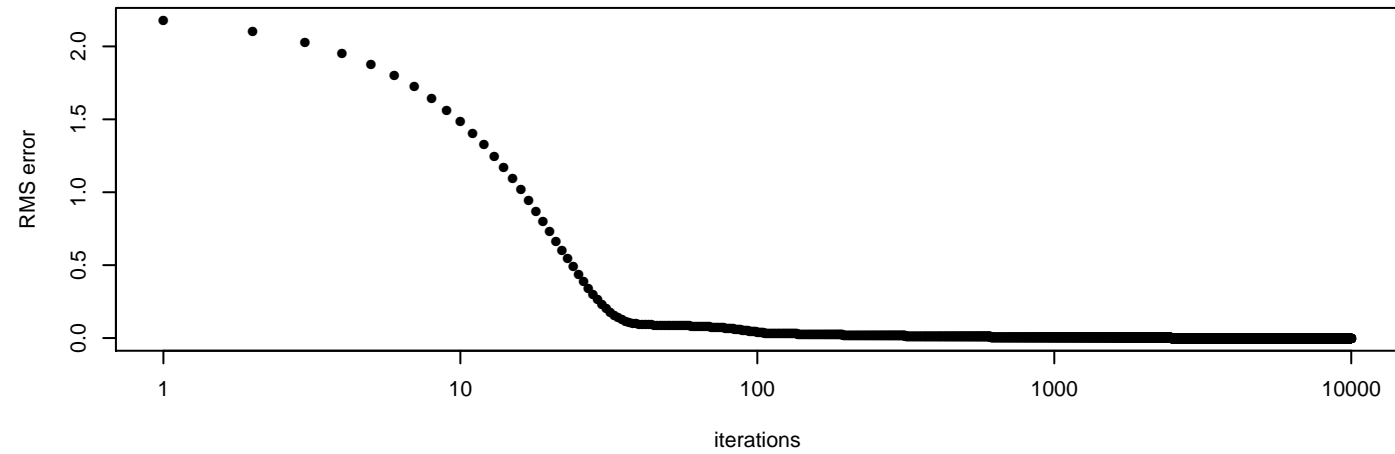


Negative Perturbation

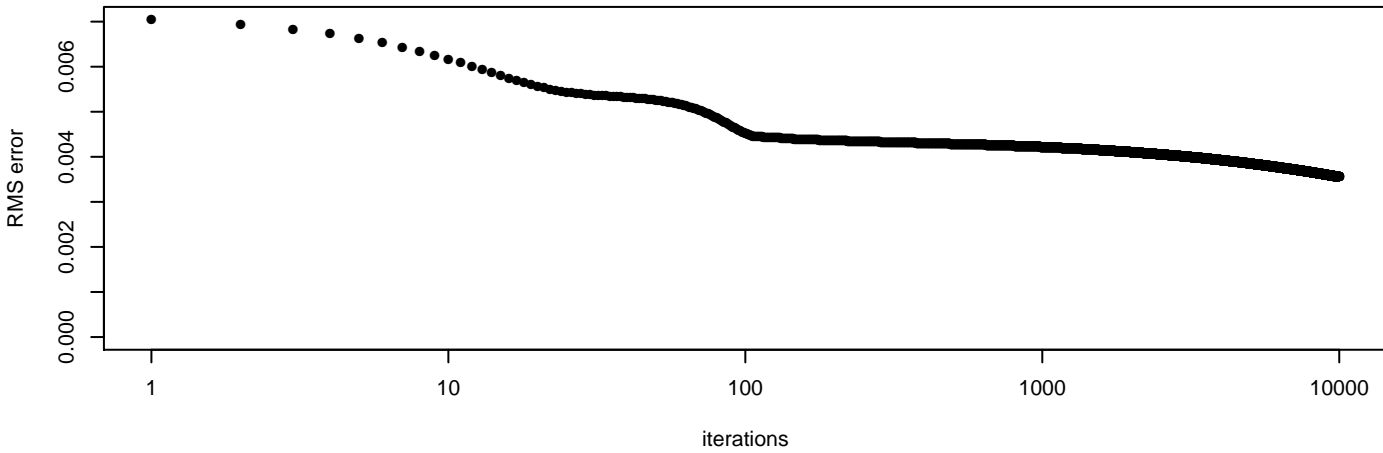


Parameter1

Positive Perturbation

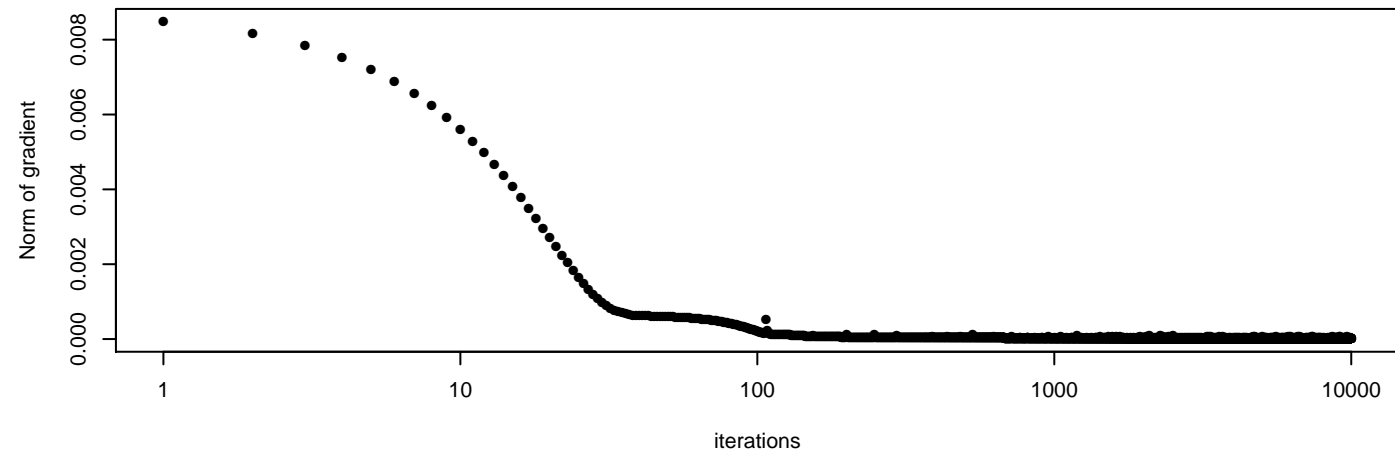
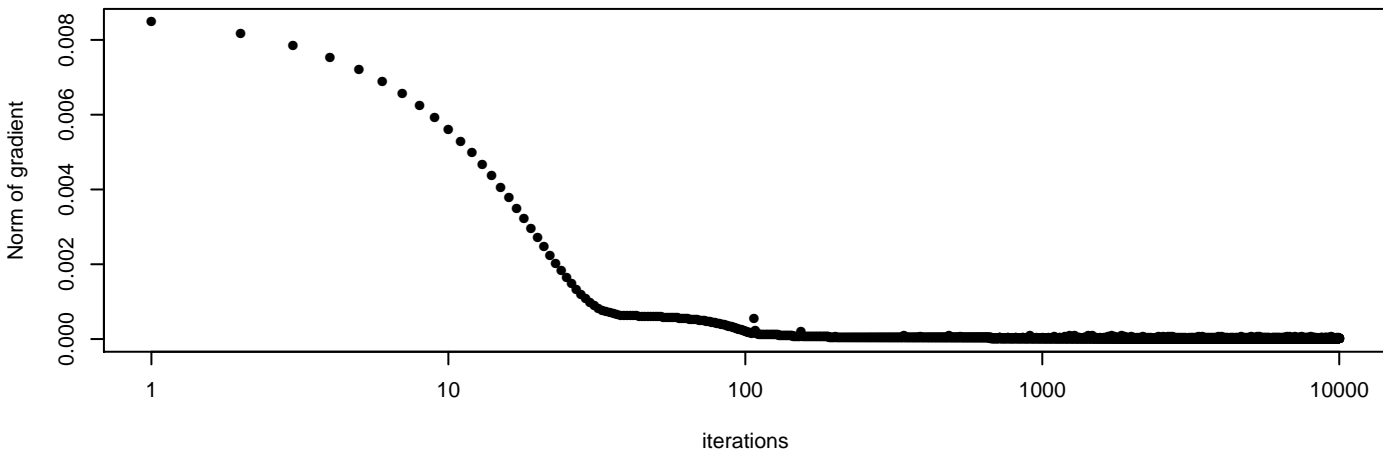
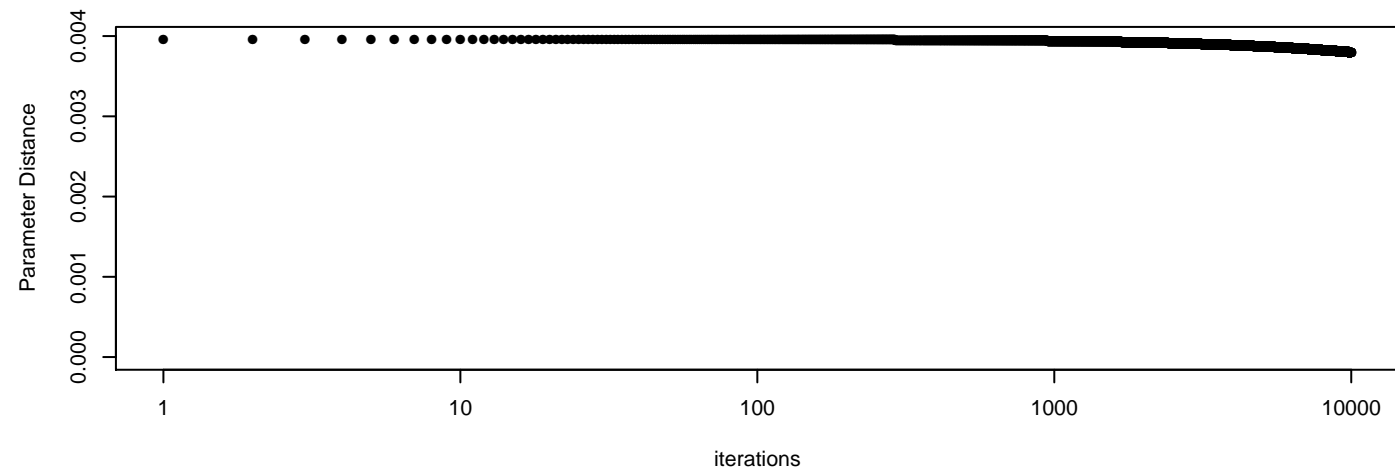
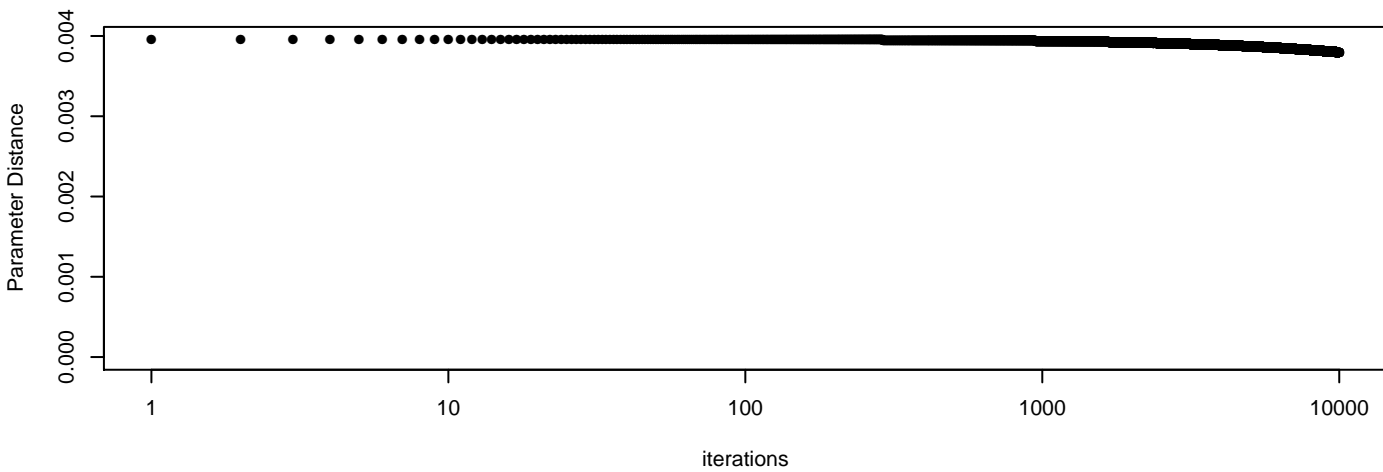
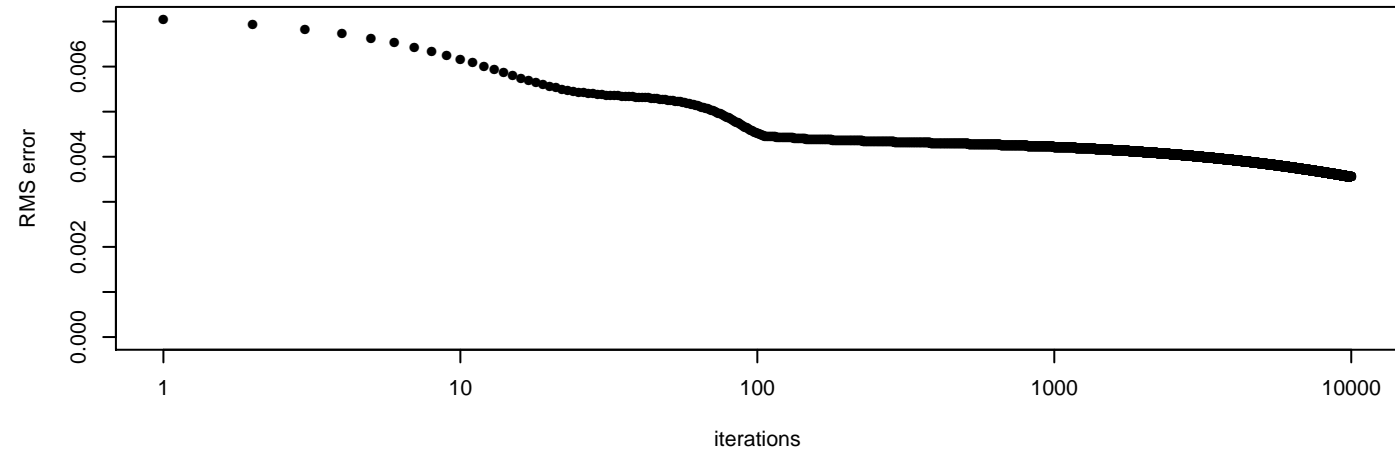


Negative Perturbation

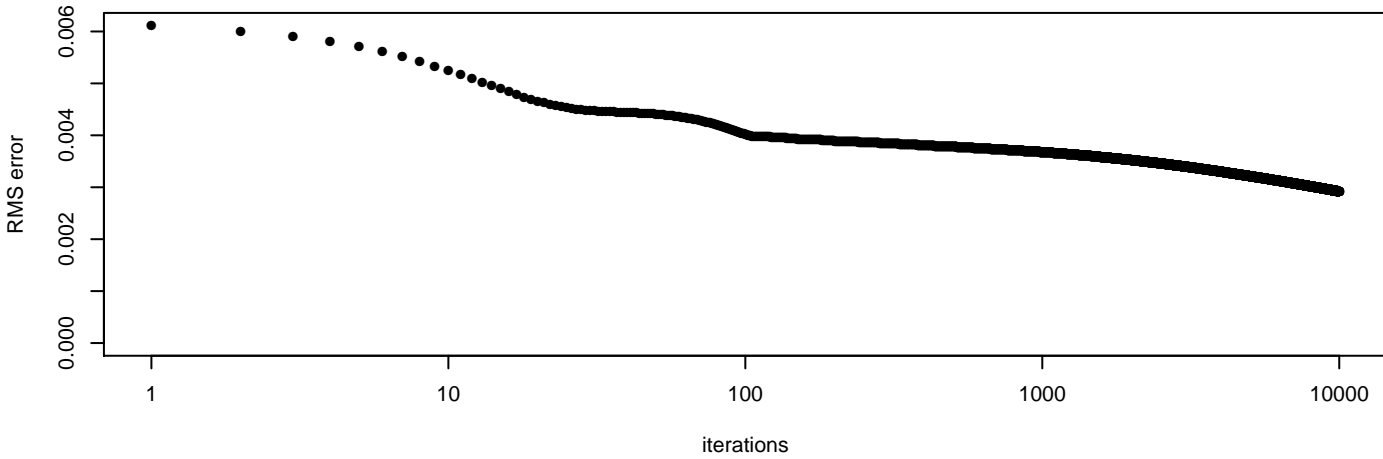


Parameter10

Positive Perturbation

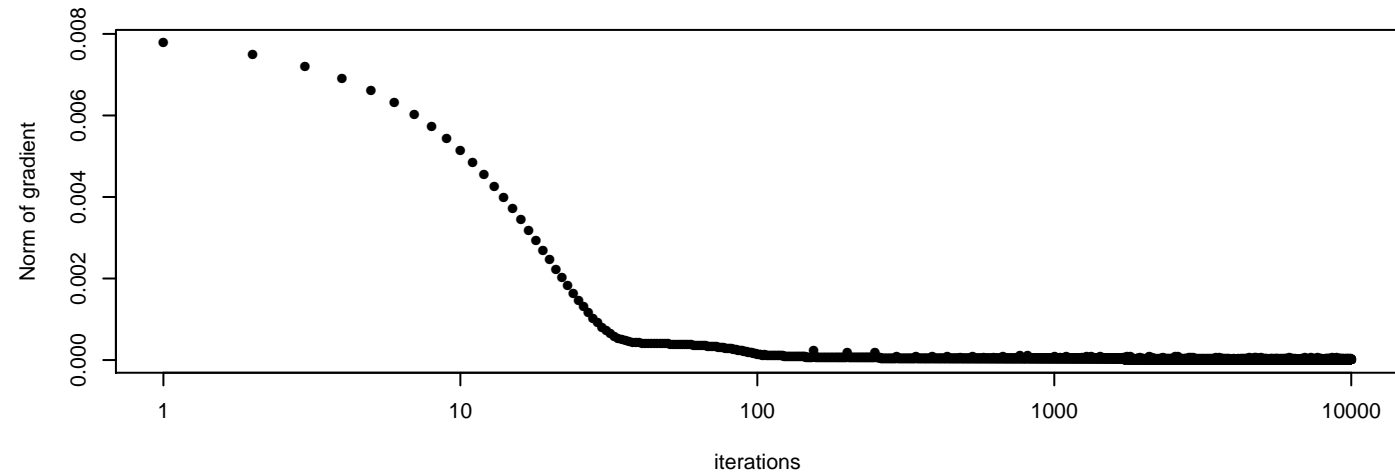
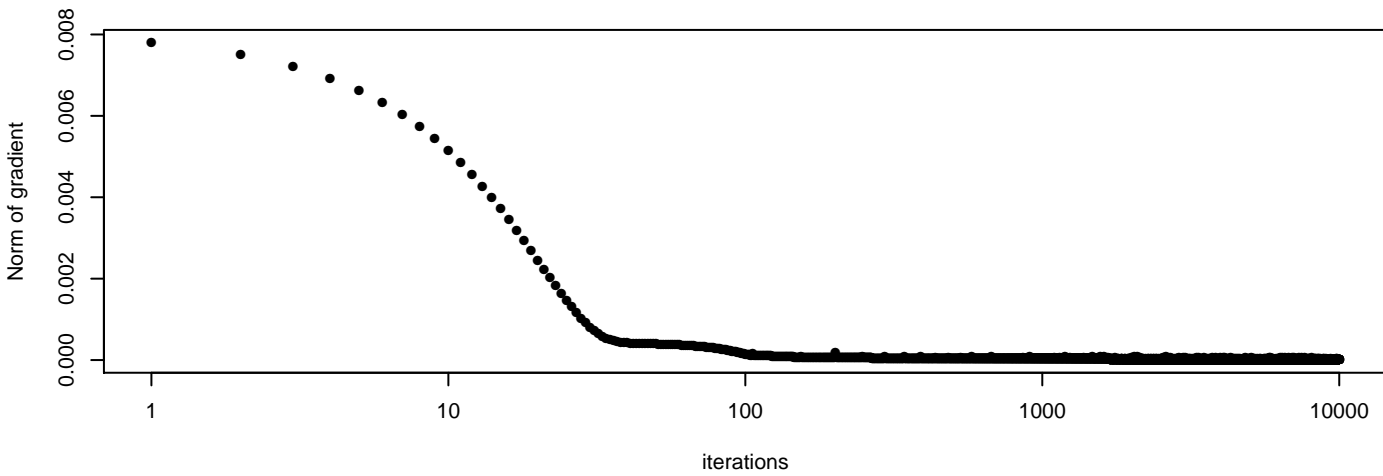
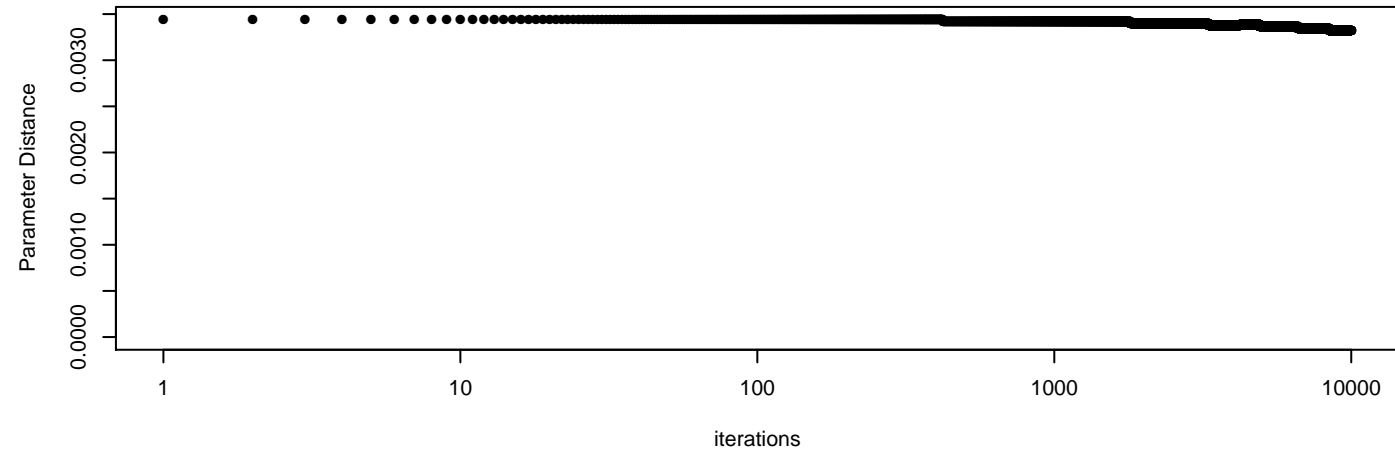
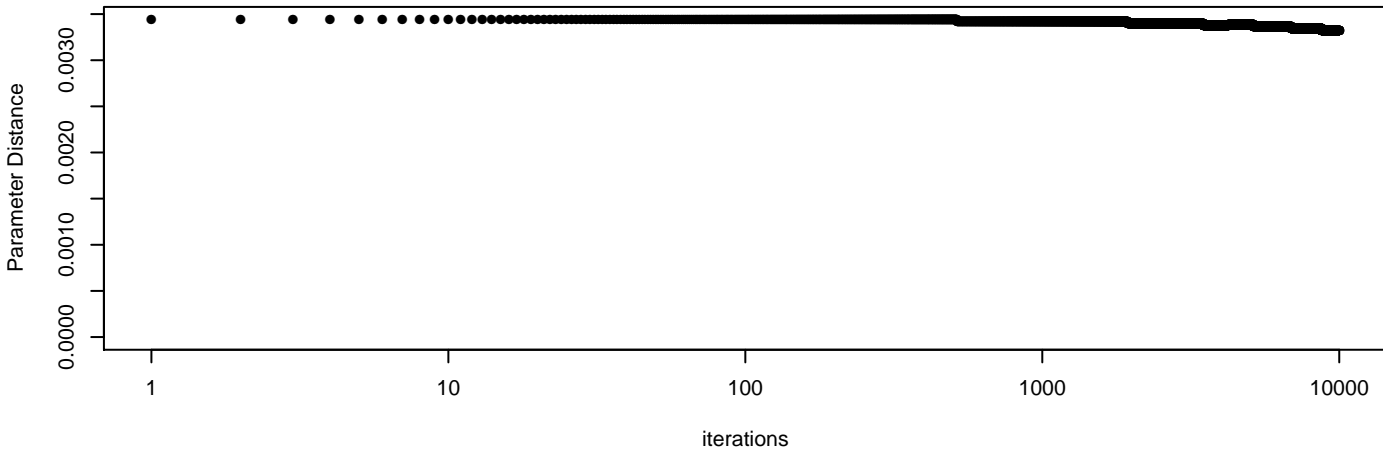
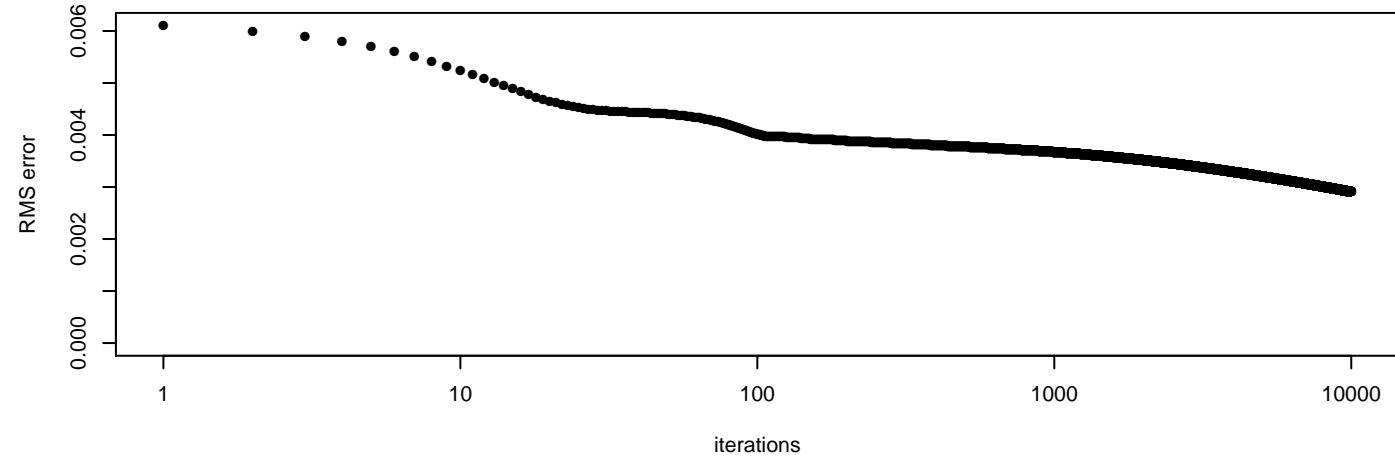


Negative Perturbation

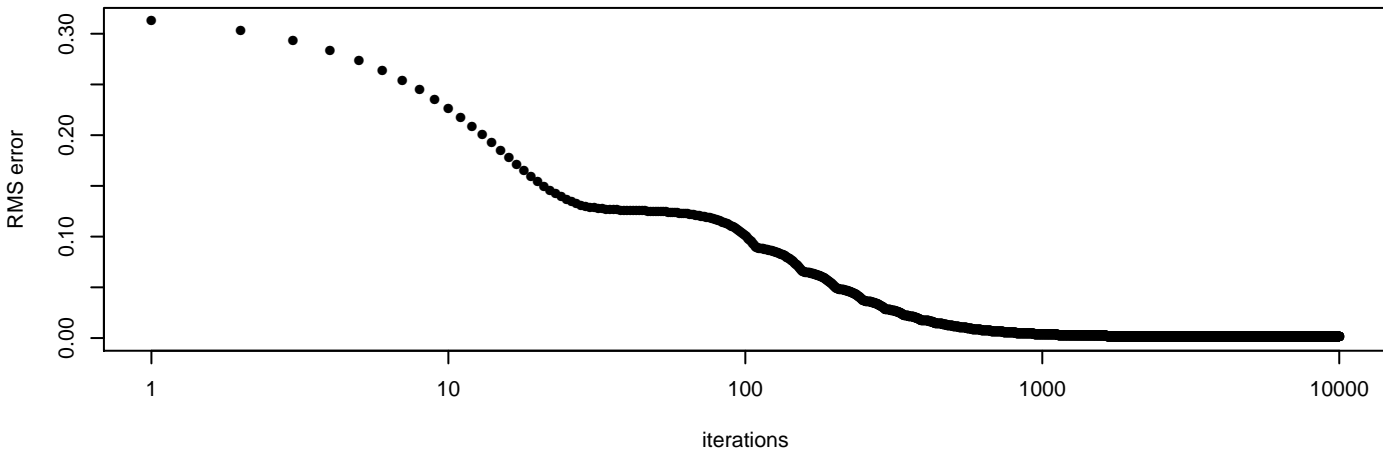


Parameter11

Positive Perturbation

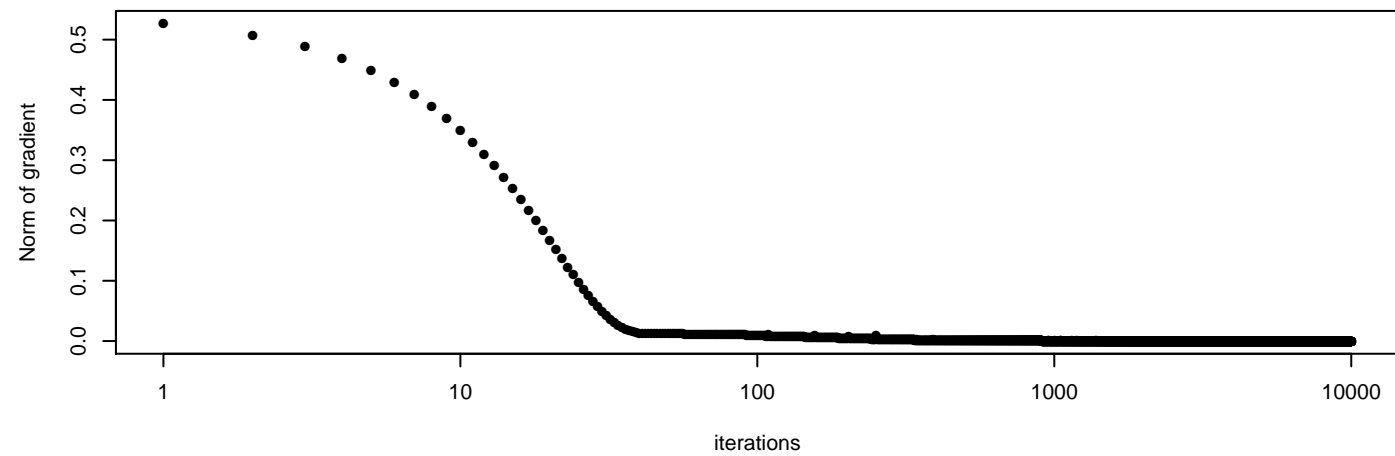
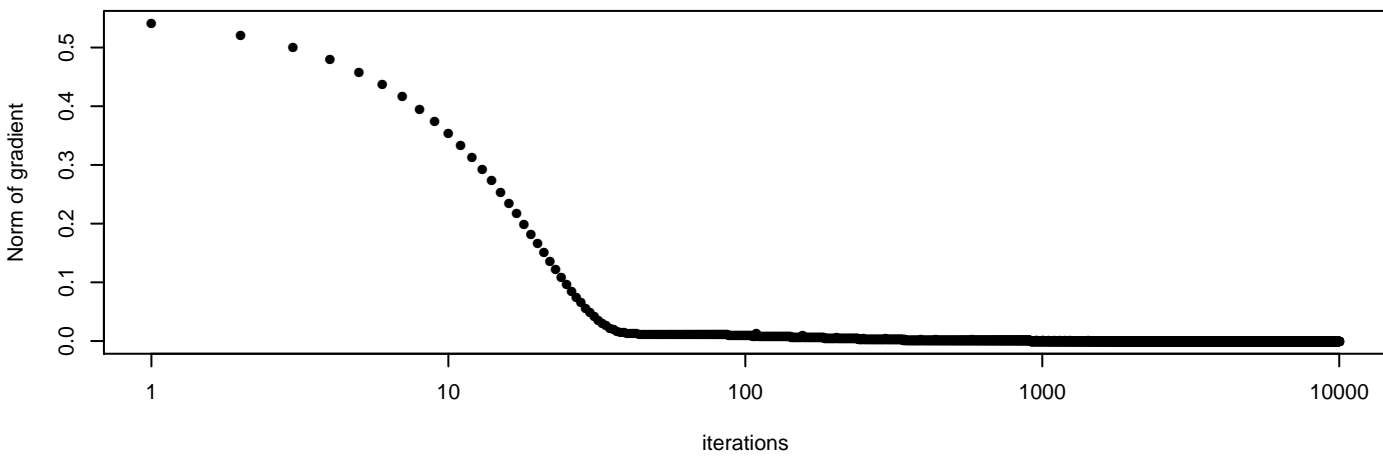
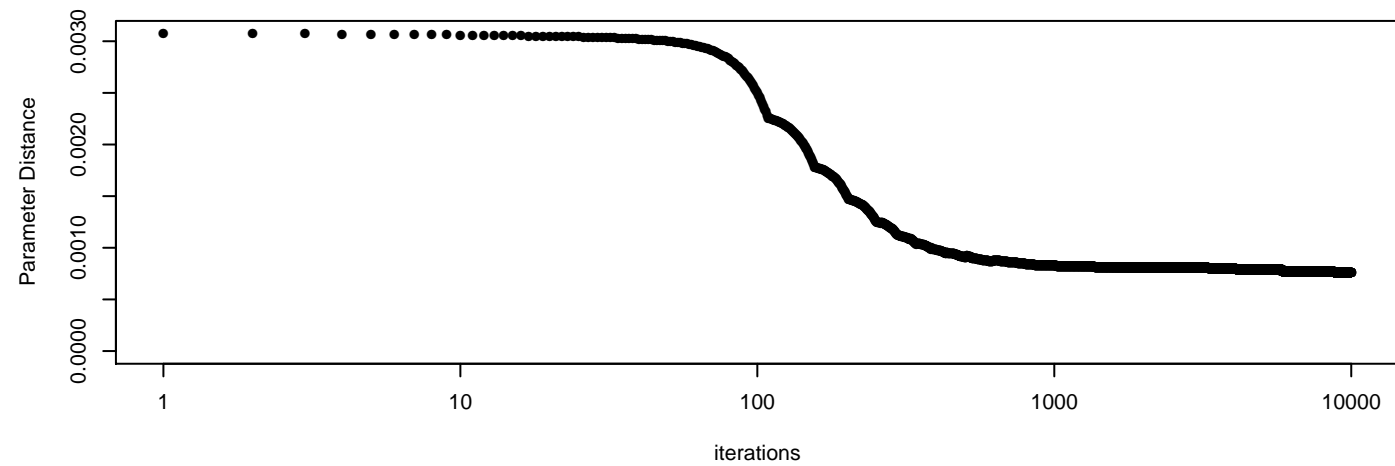
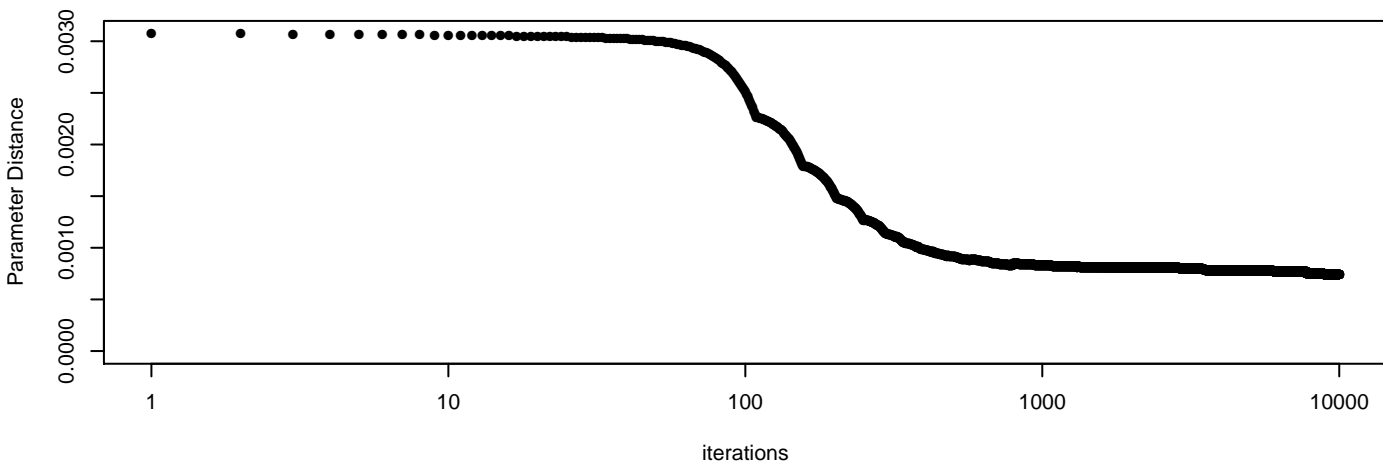
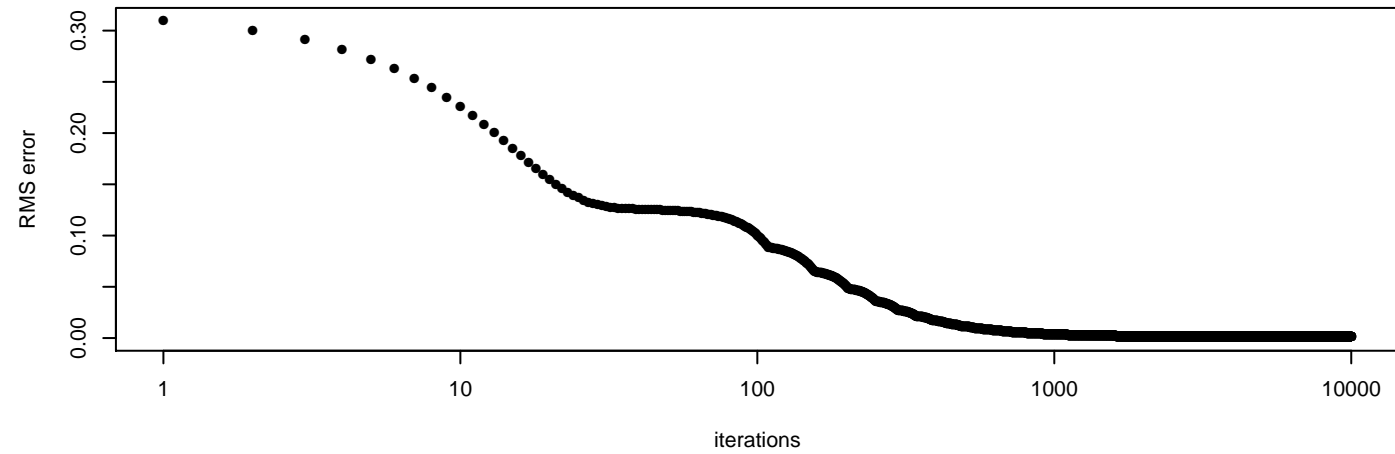


Negative Perturbation

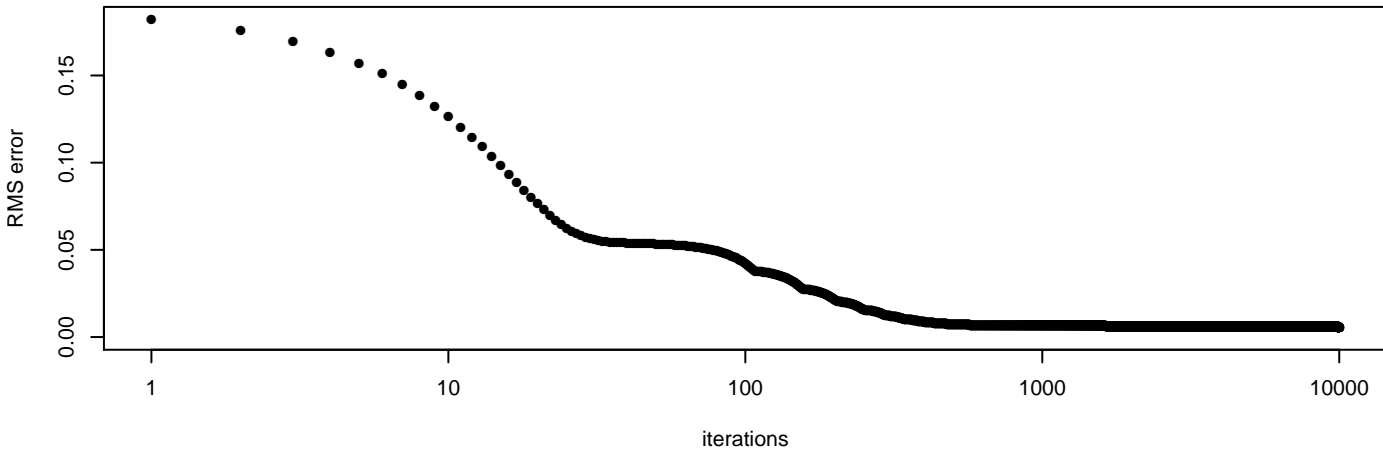


Parameter12

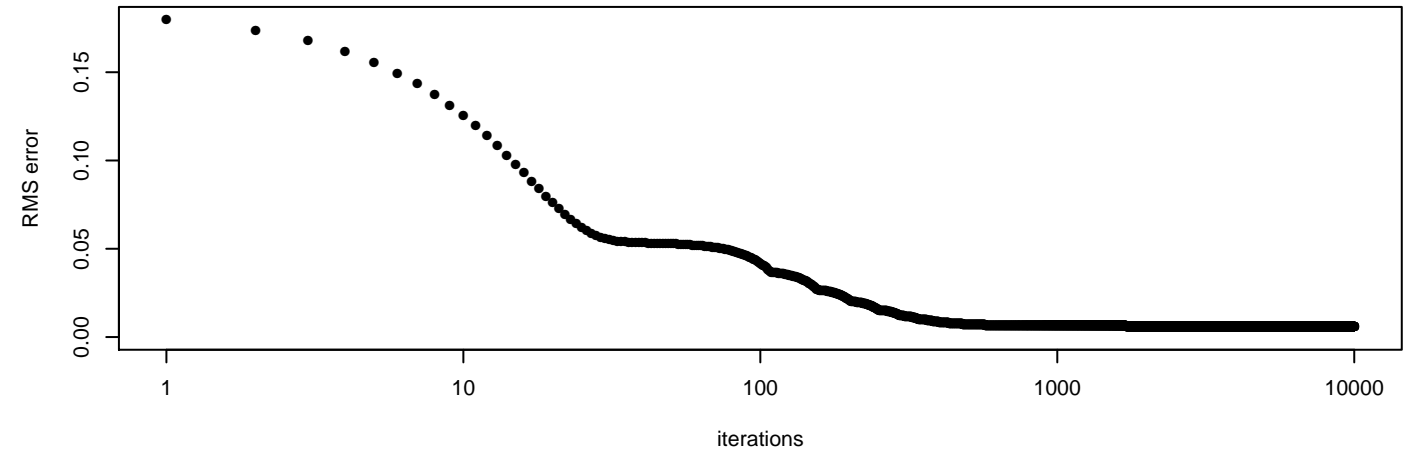
Positive Perturbation



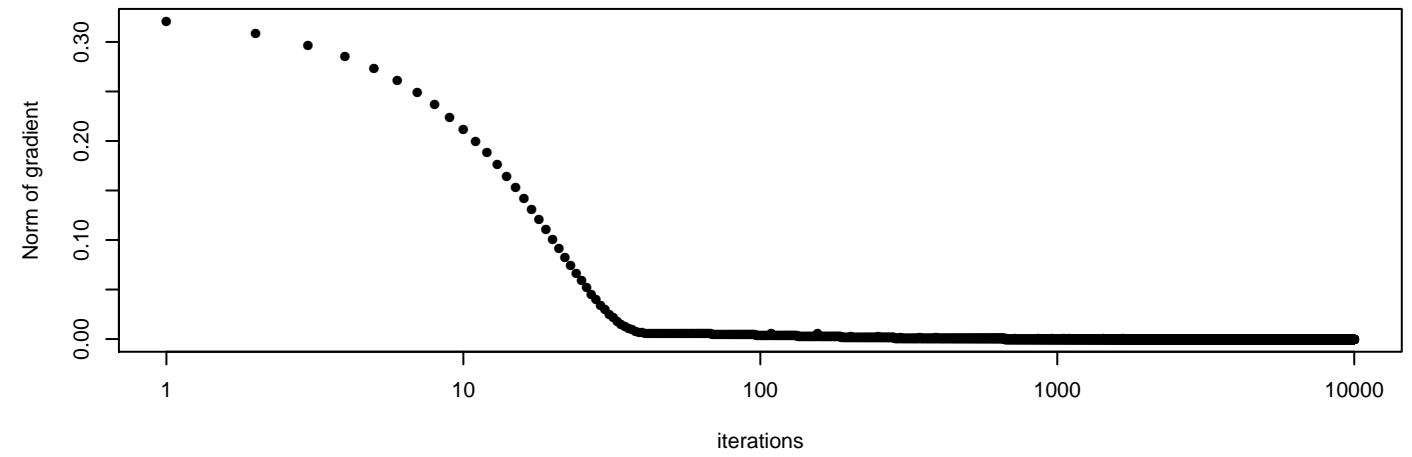
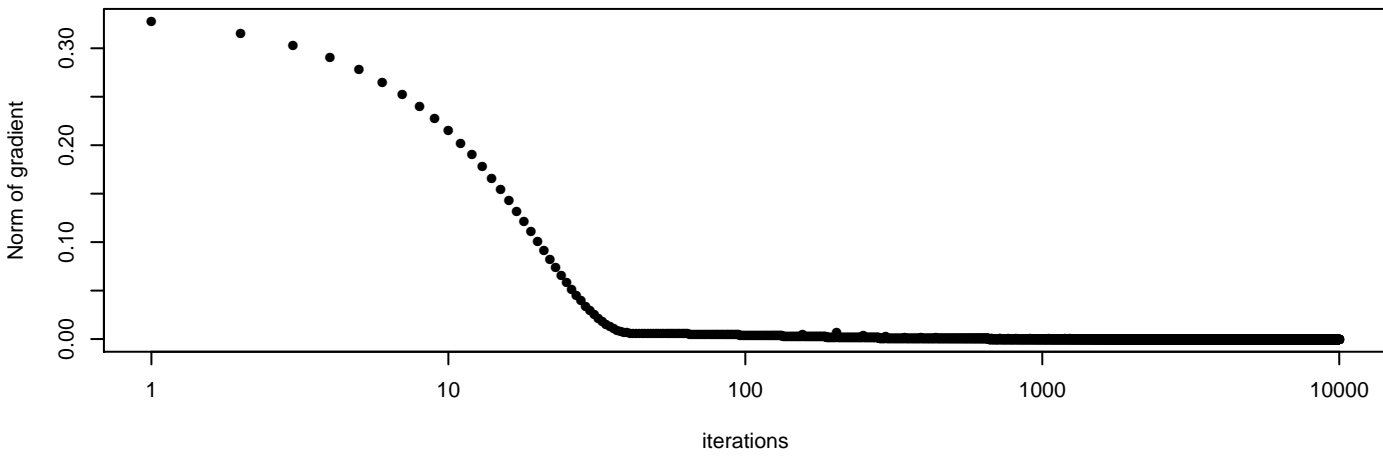
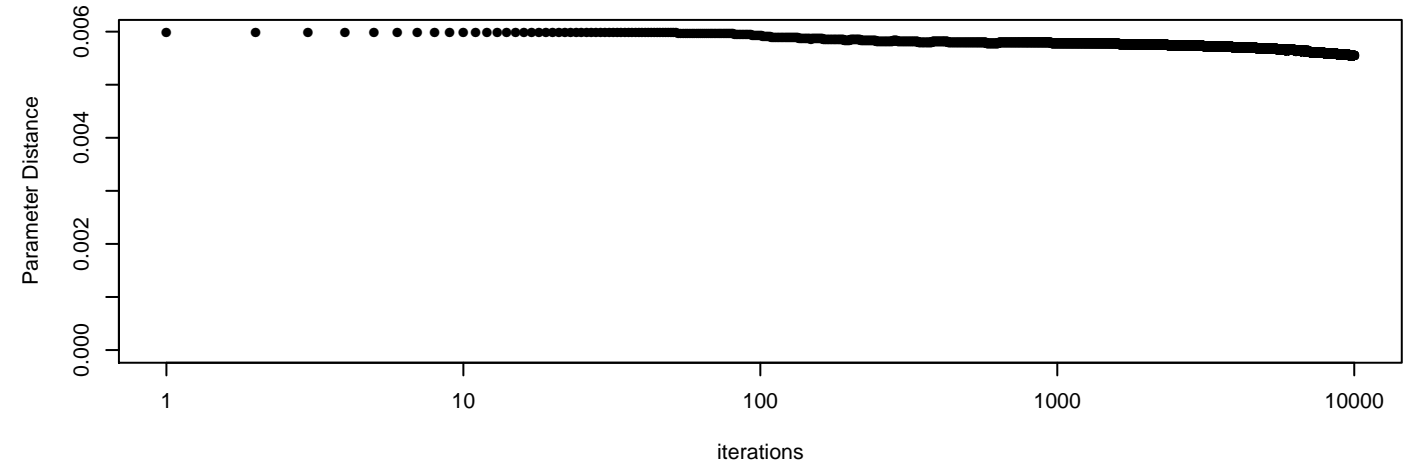
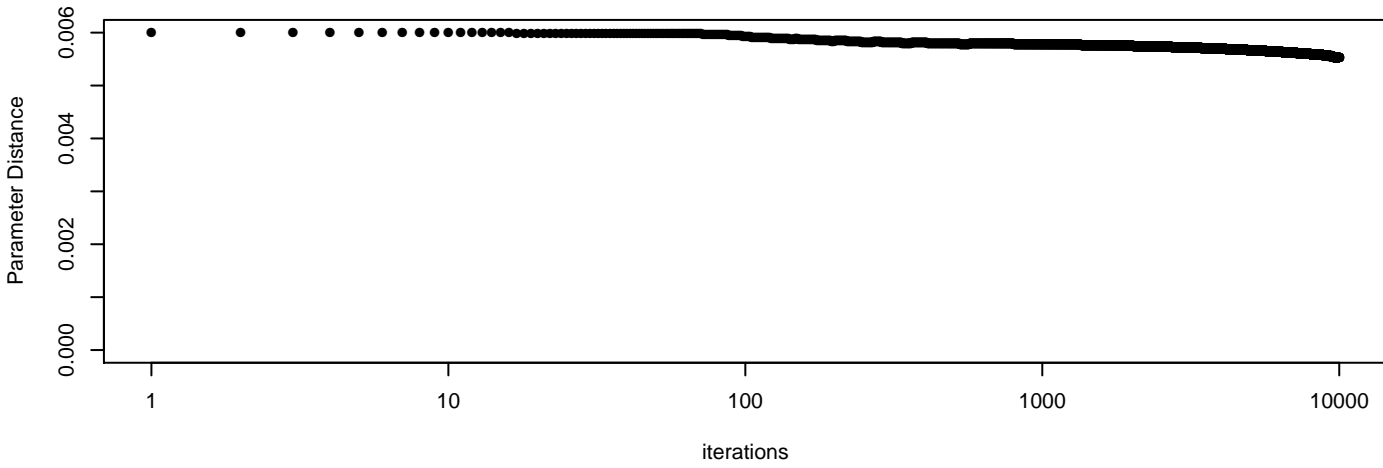
Negative Perturbation



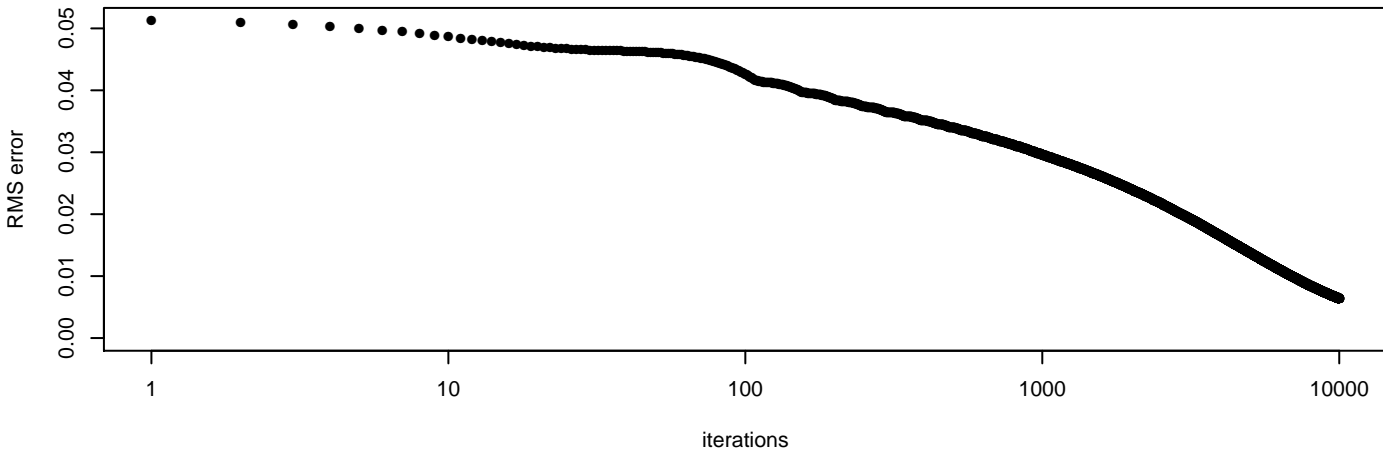
Parameter13



Positive Perturbation

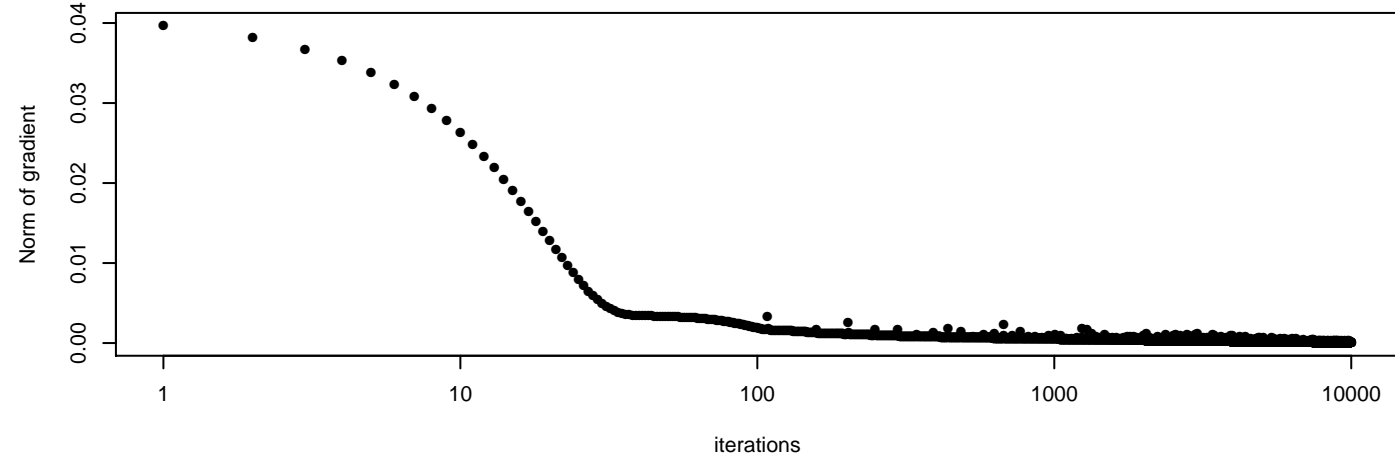
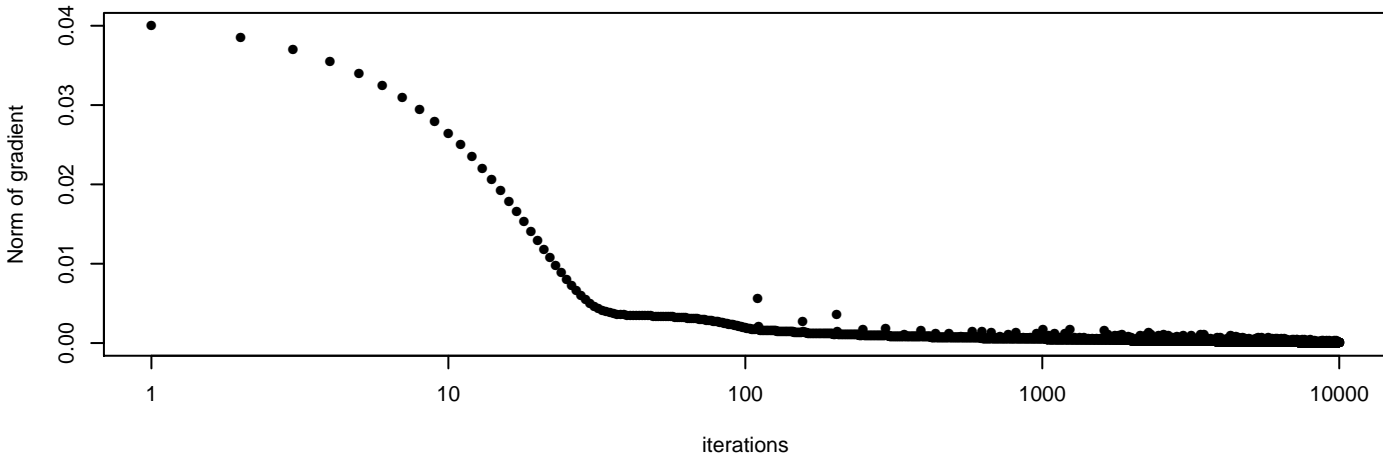
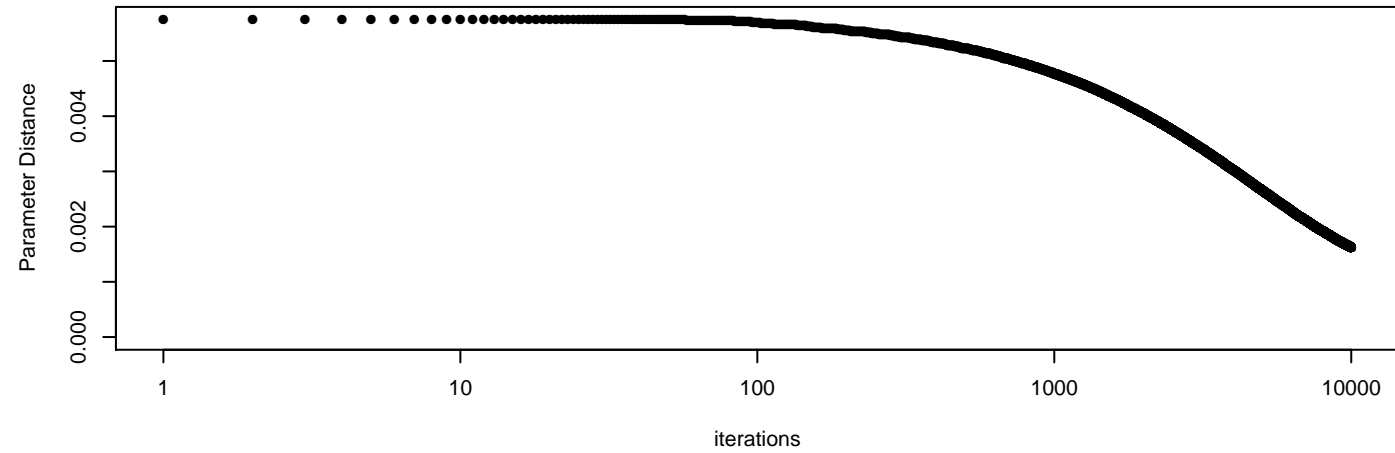
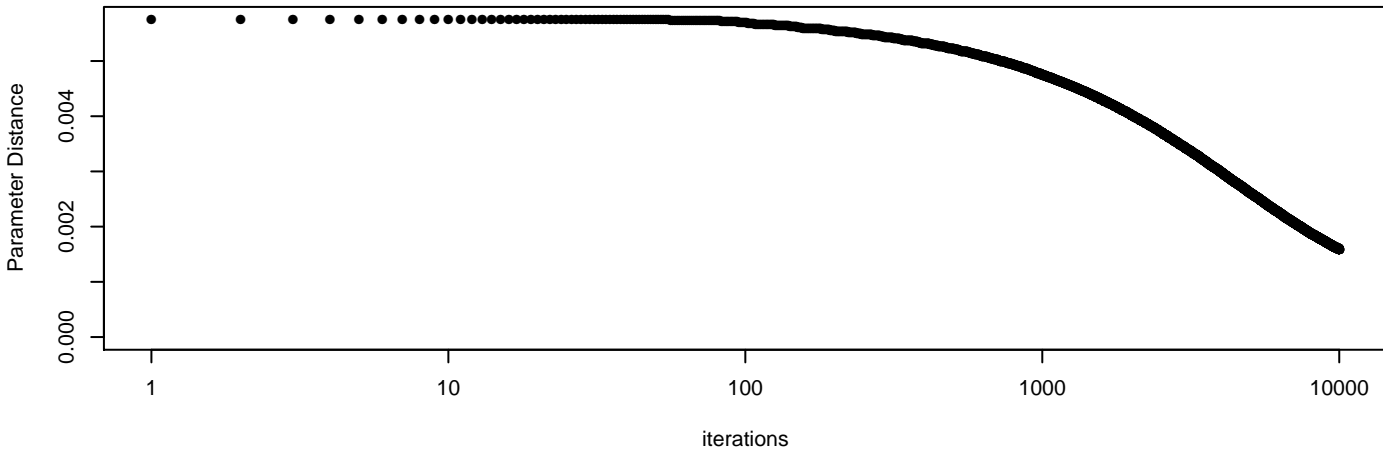
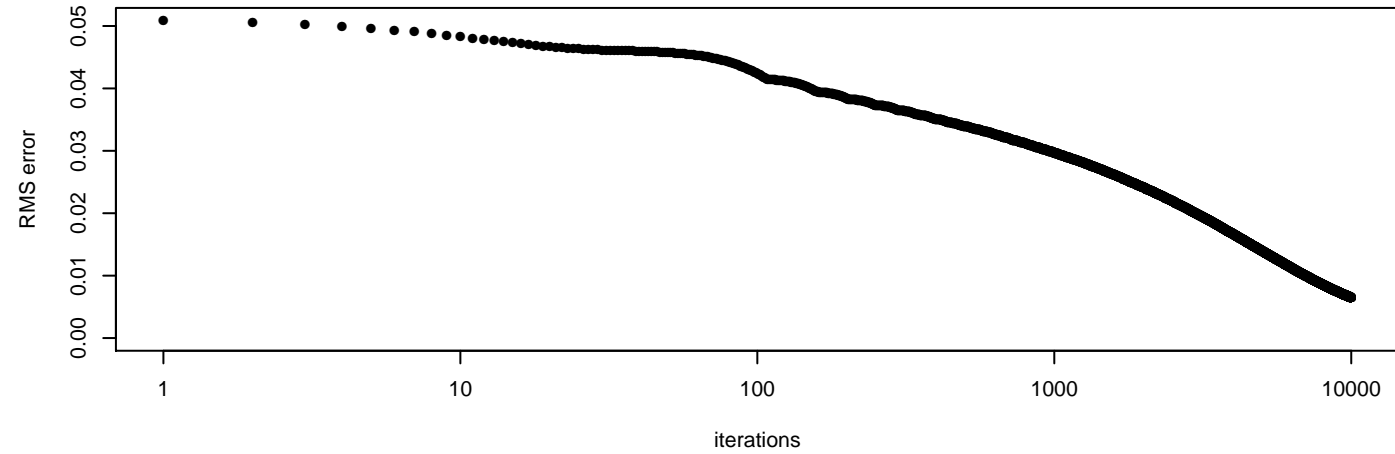


Negative Perturbation

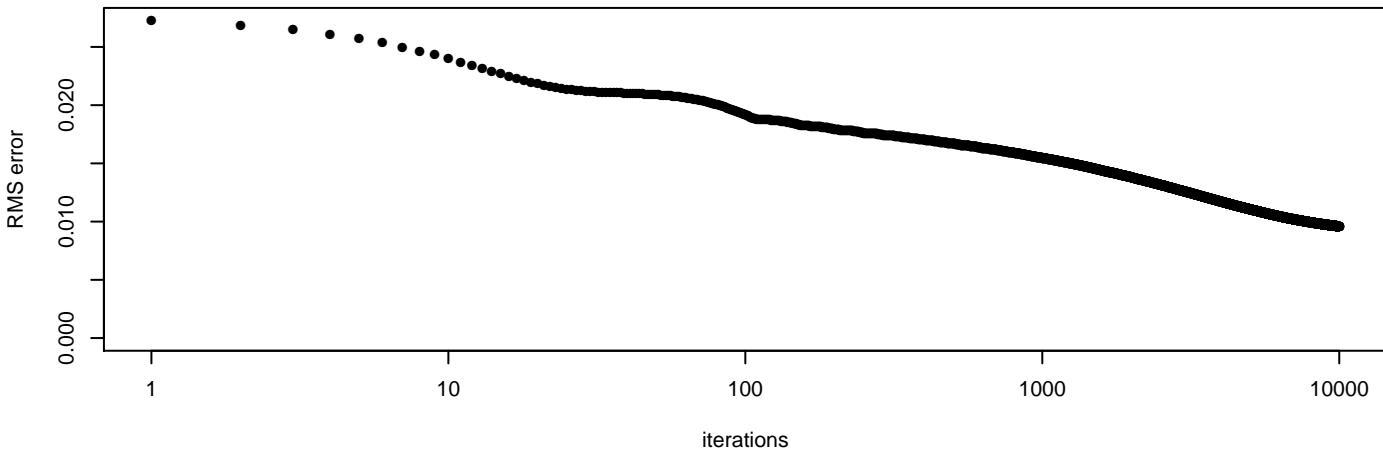


Parameter14

Positive Perturbation

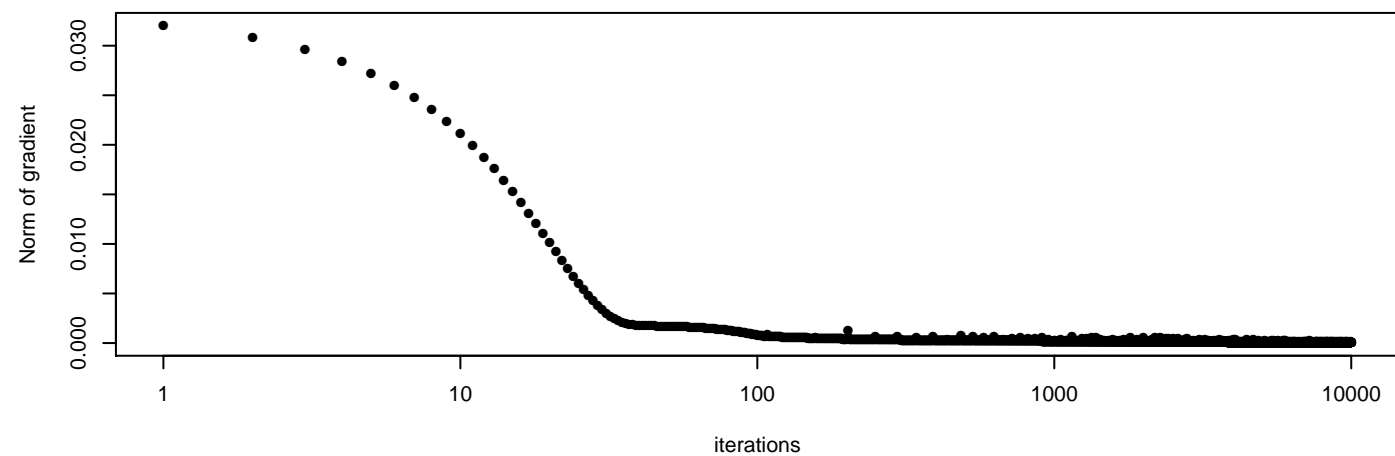
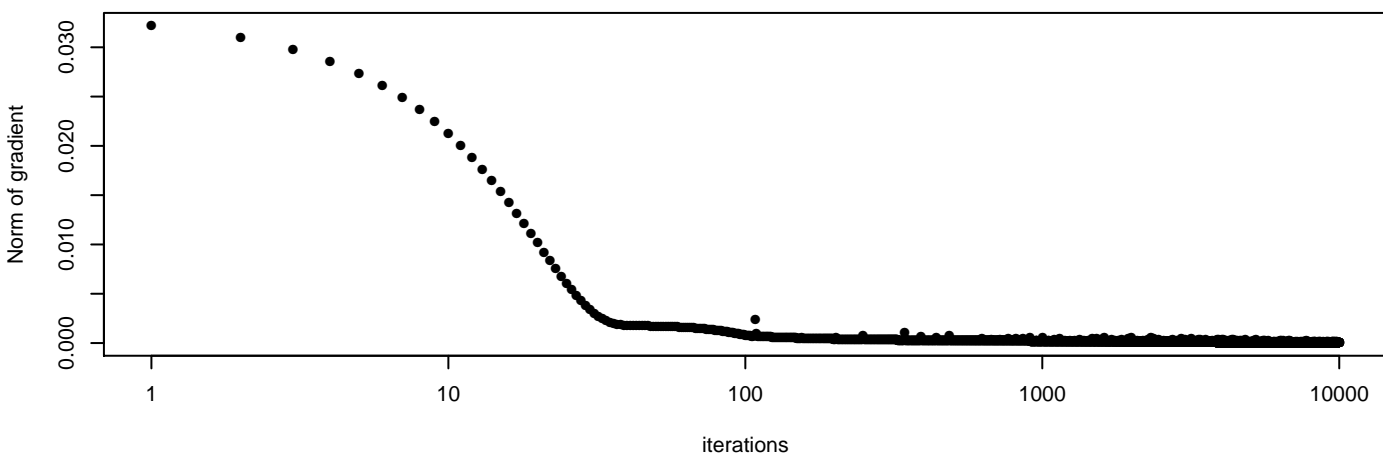
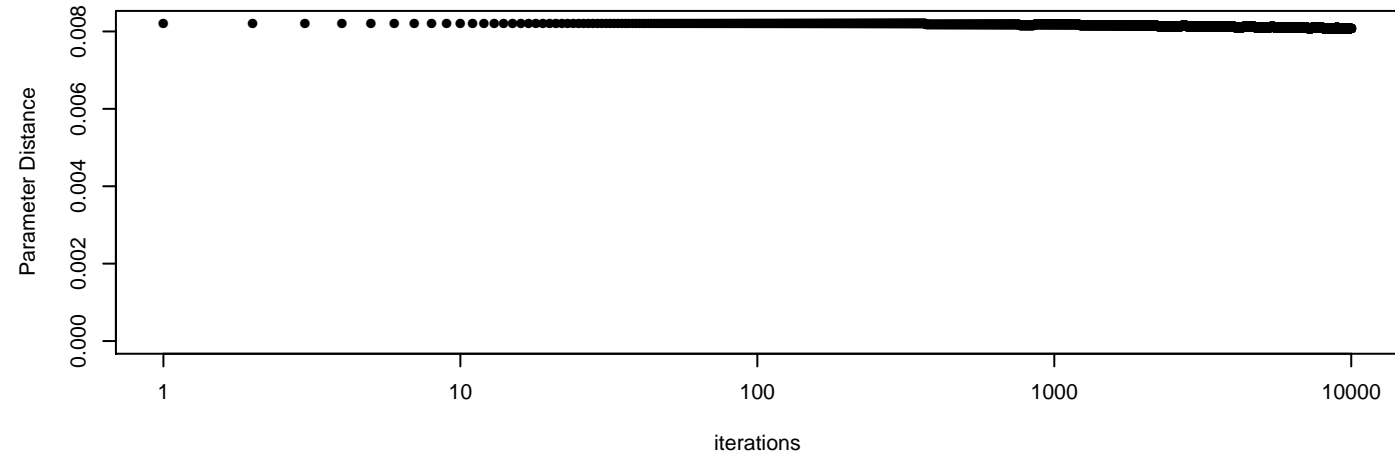
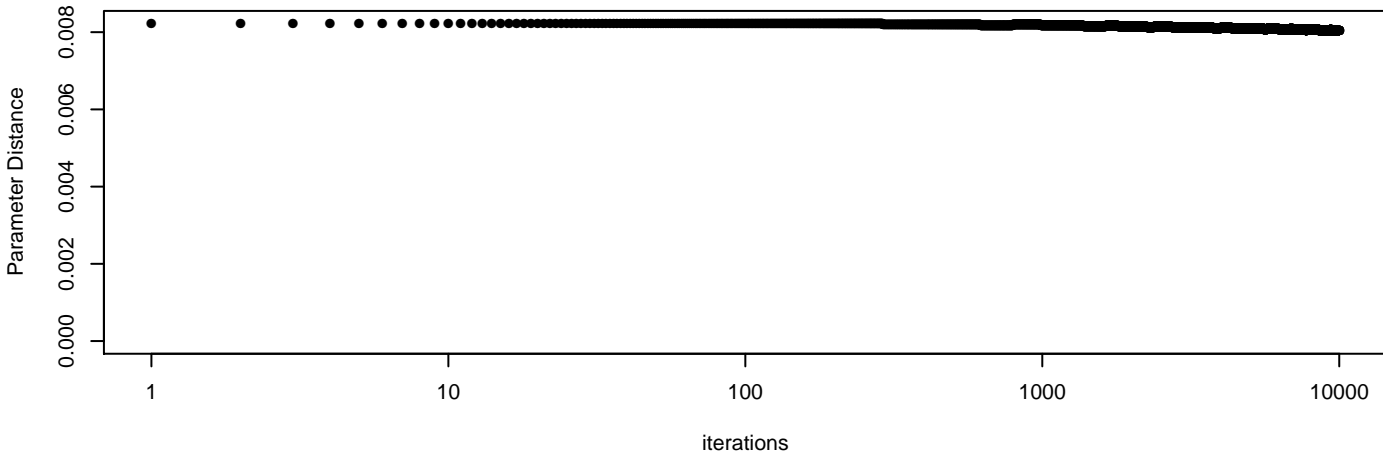
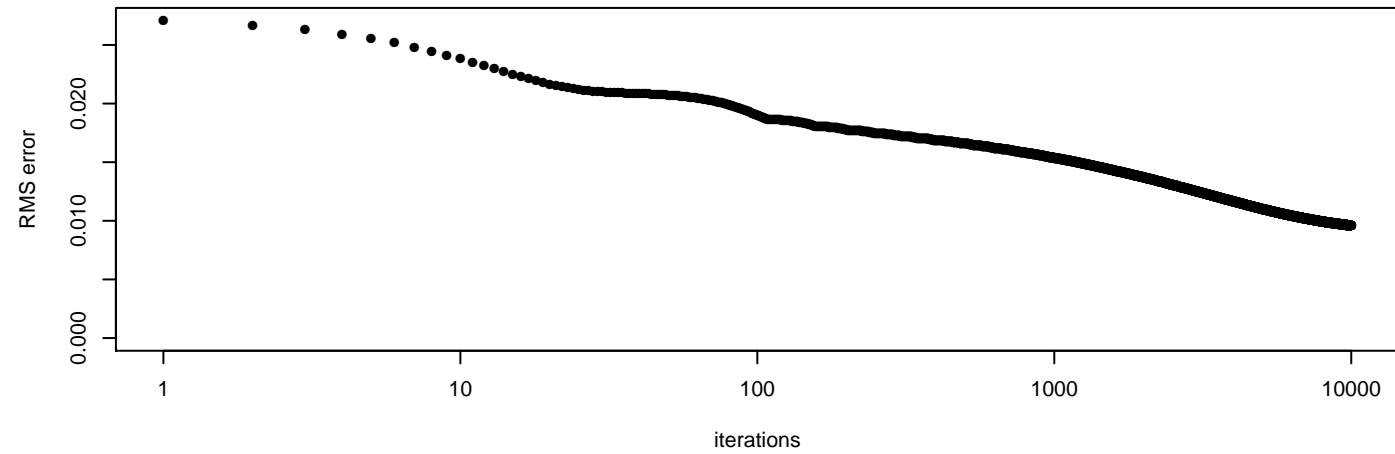


Negative Perturbation

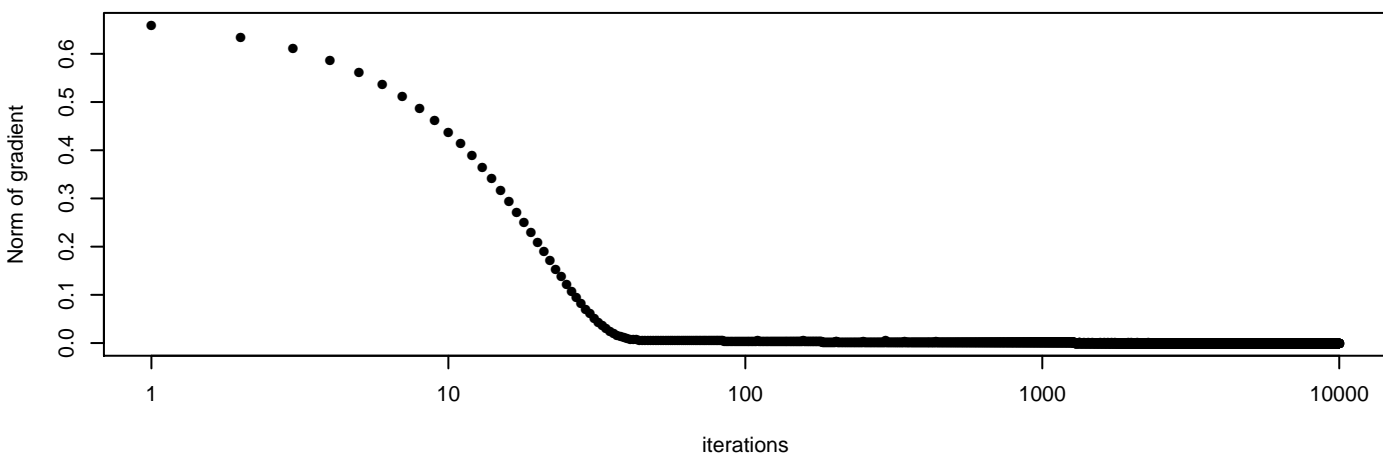
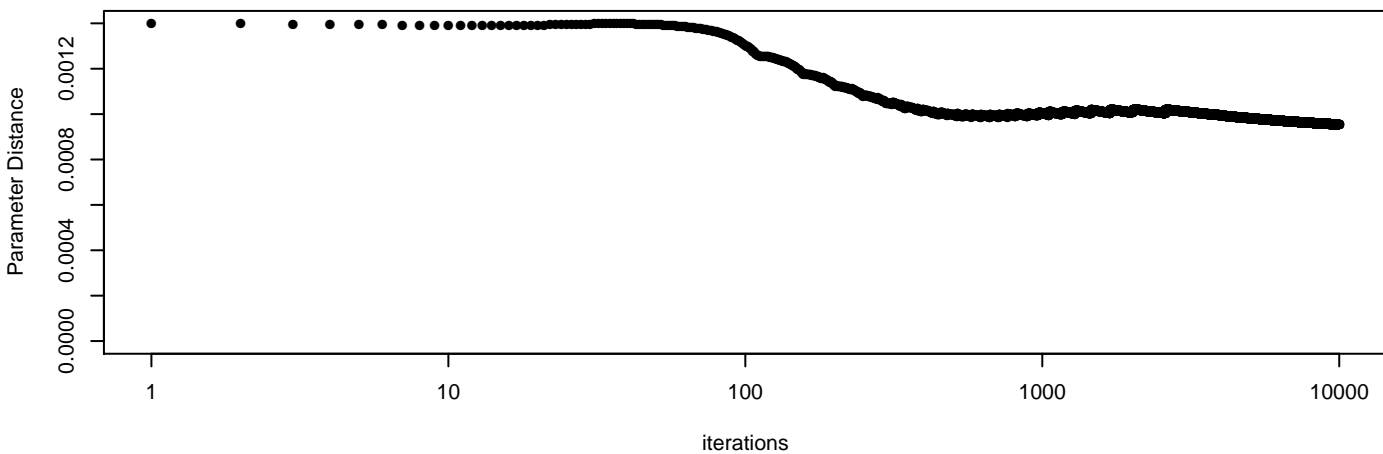
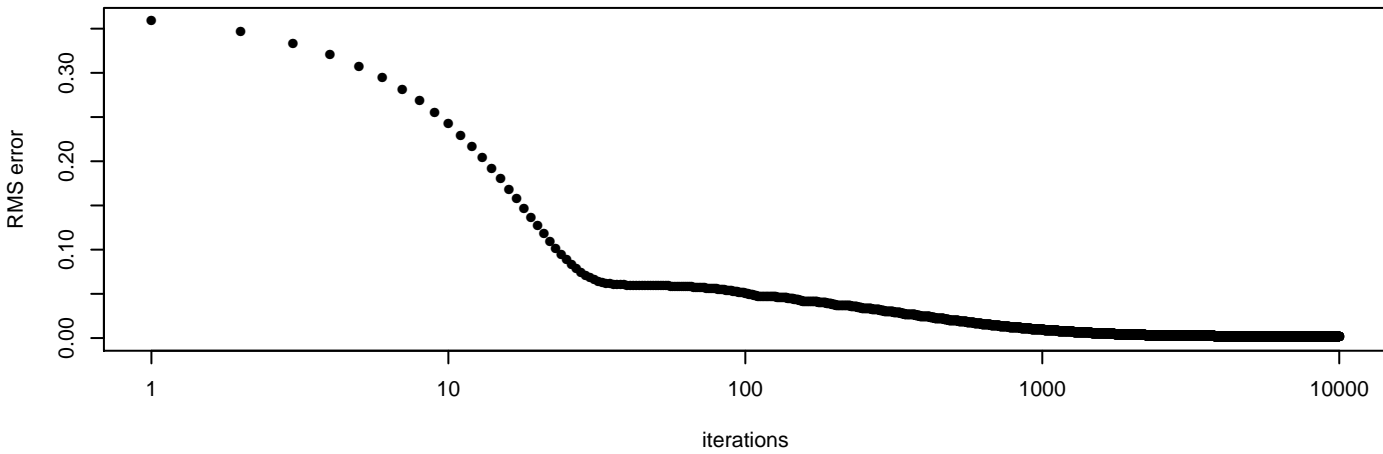


Parameter15

Positive Perturbation

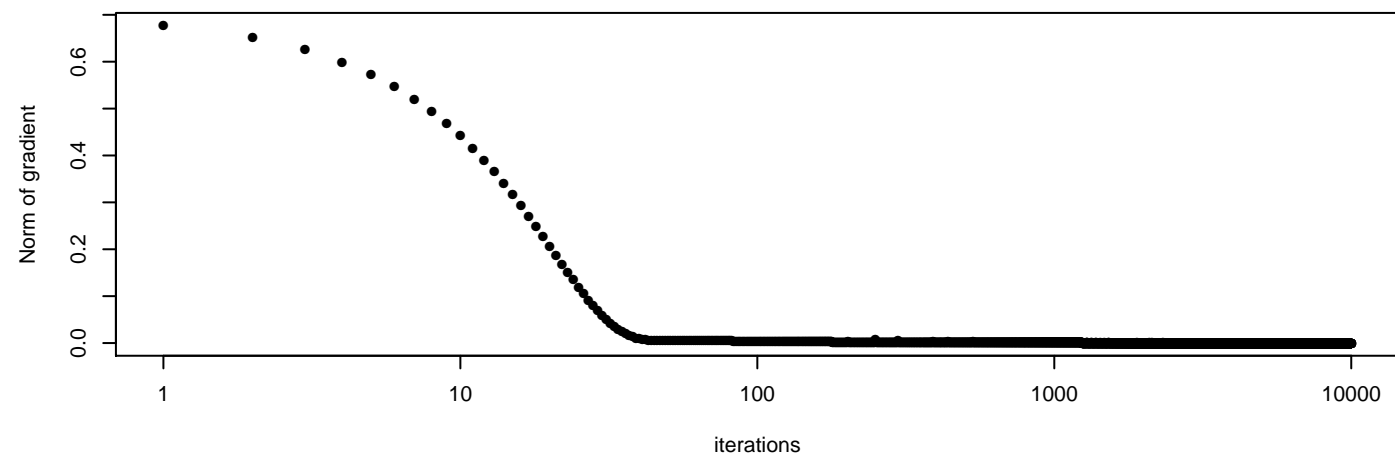
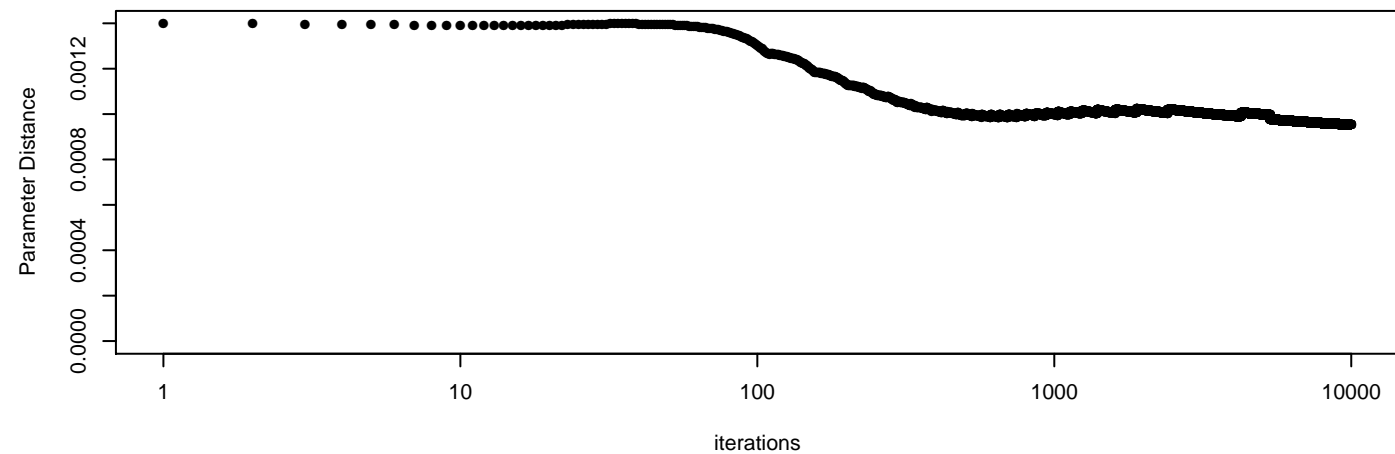
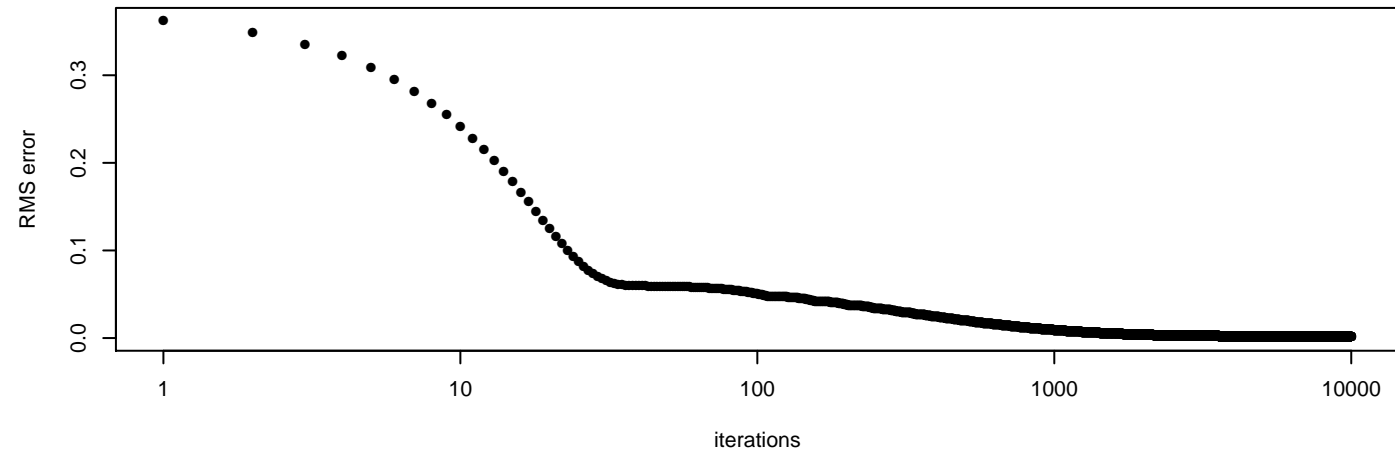


Negative Perturbation

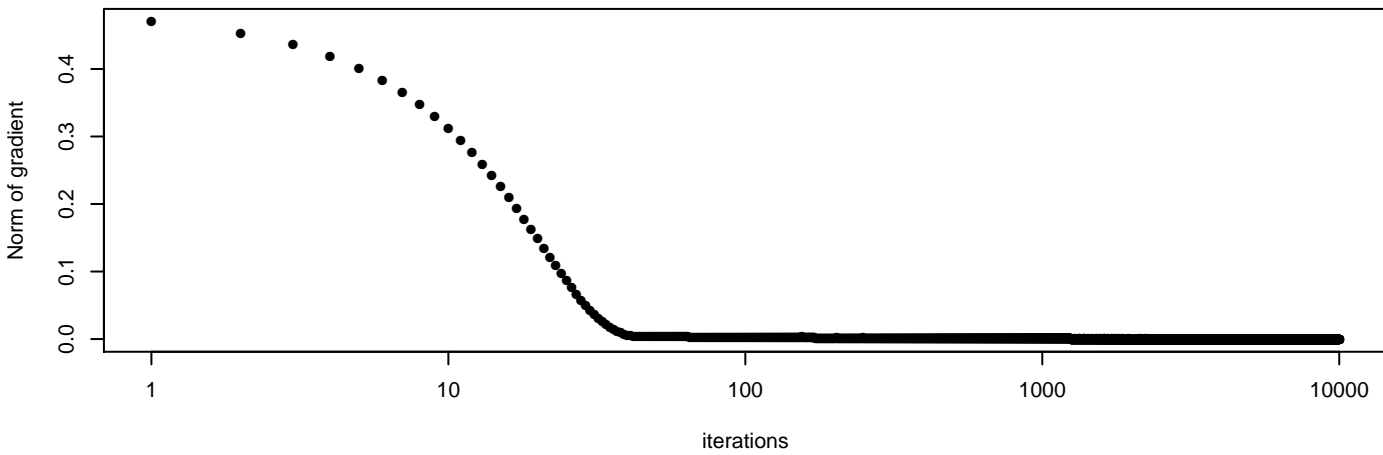
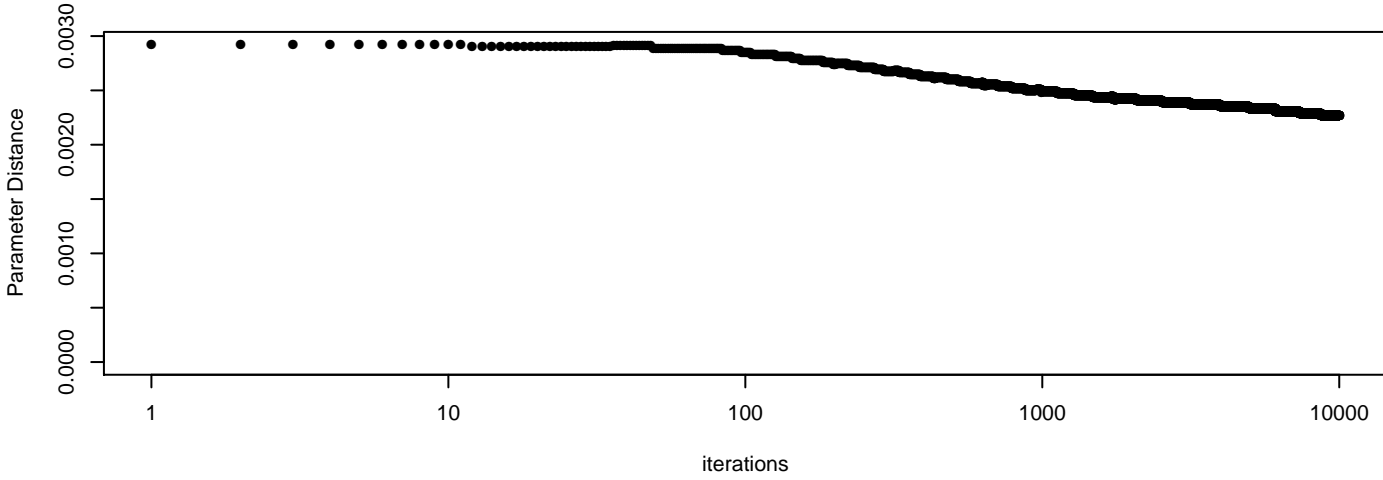
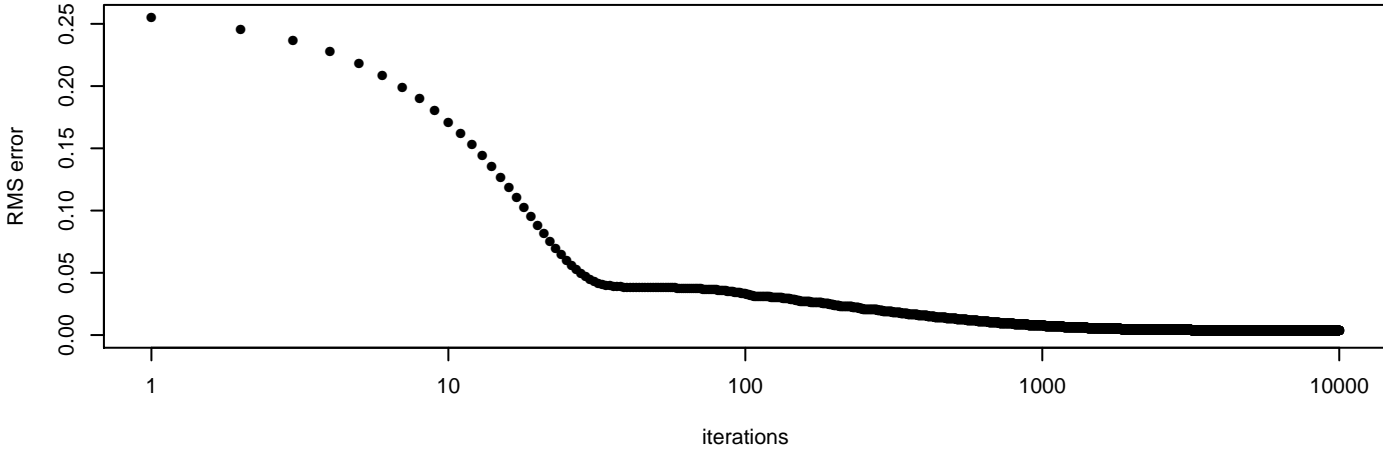


Parameter2

Positive Perturbation

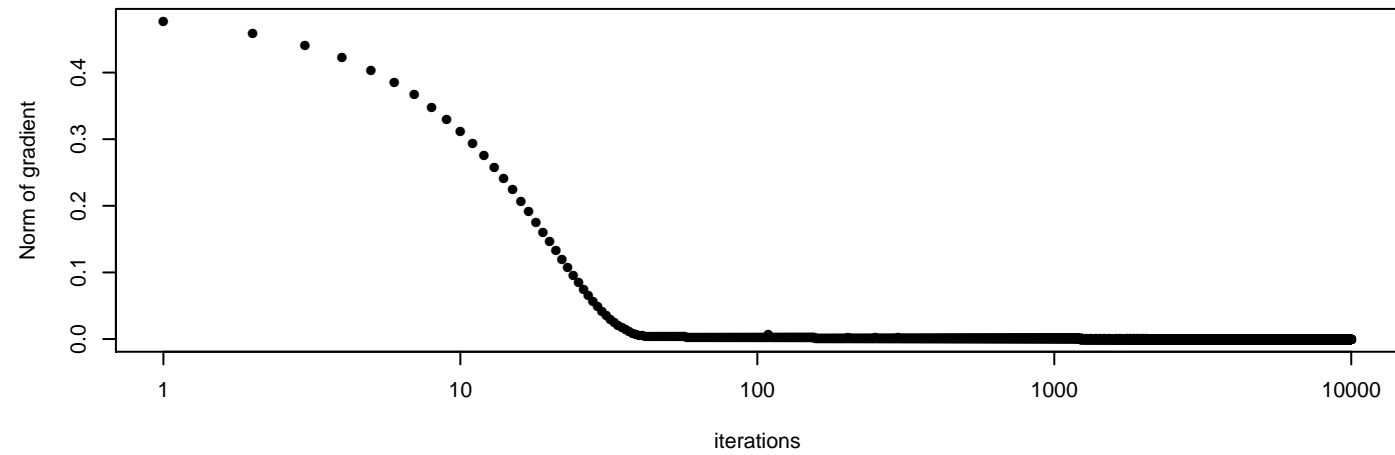
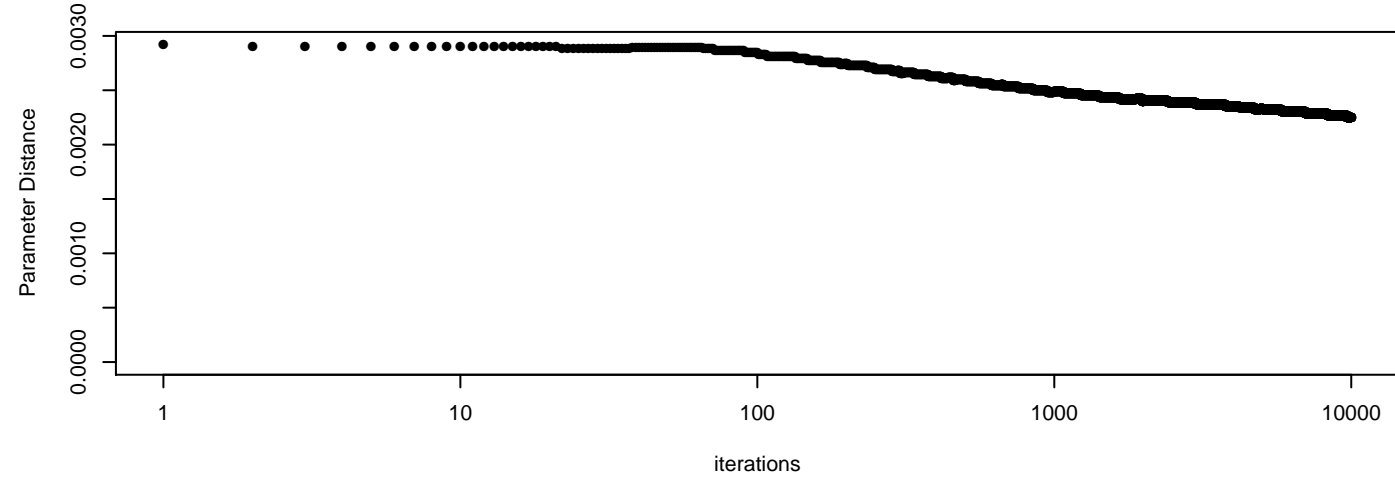
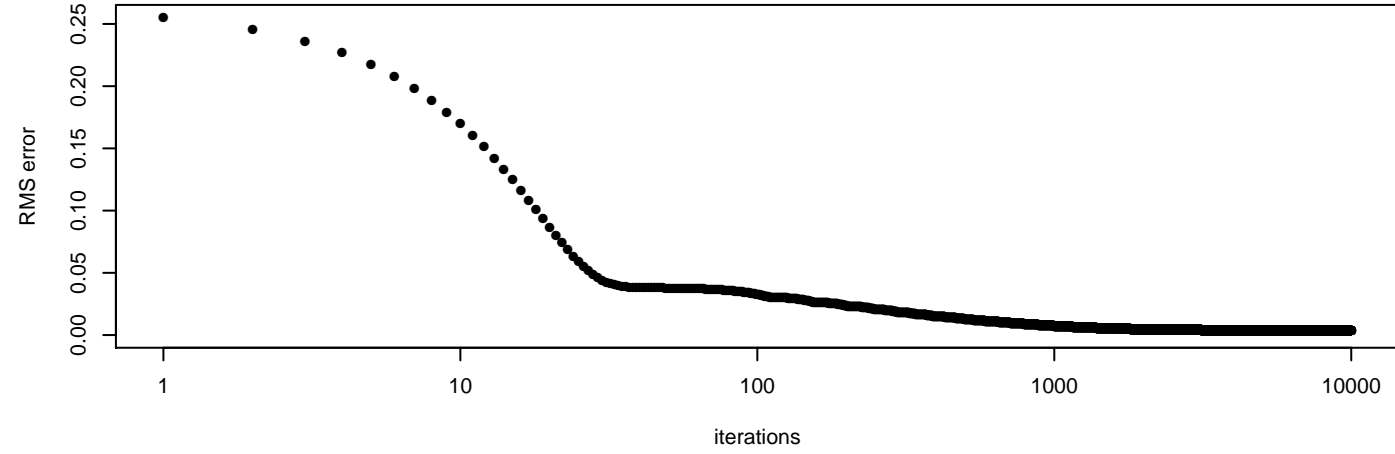


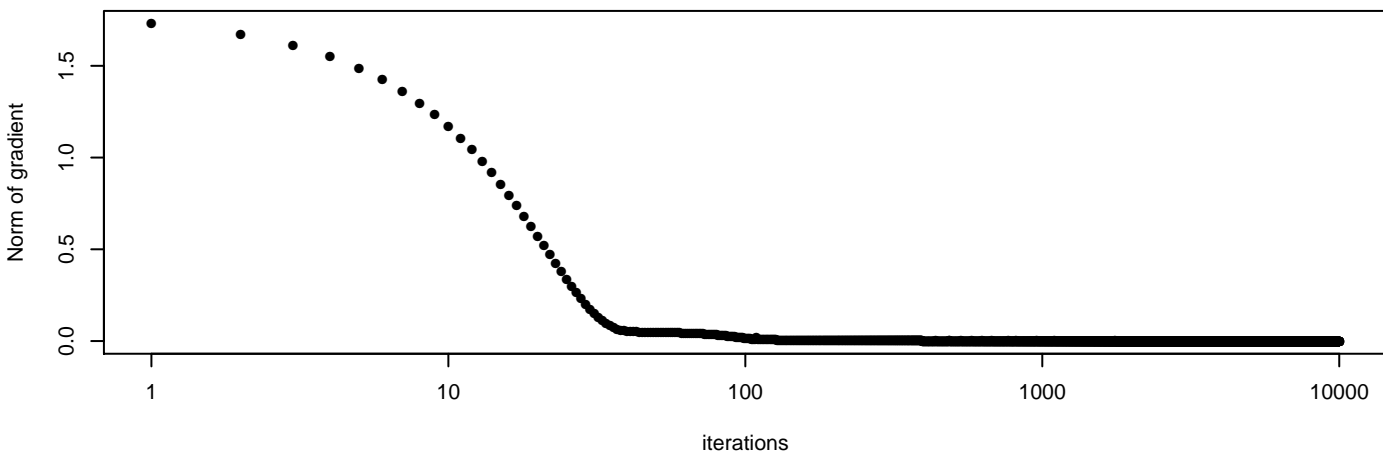
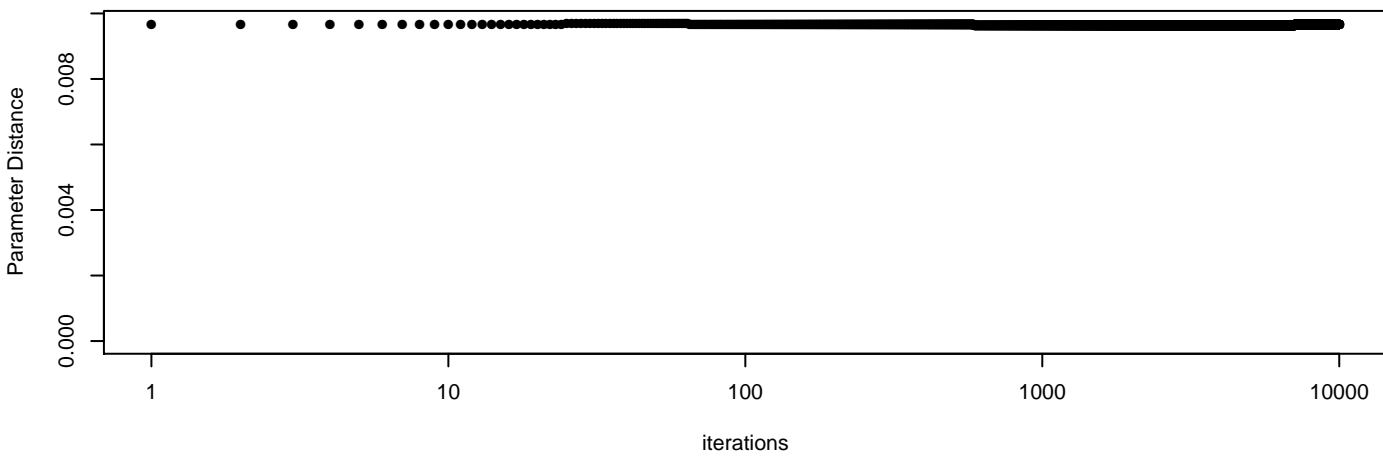
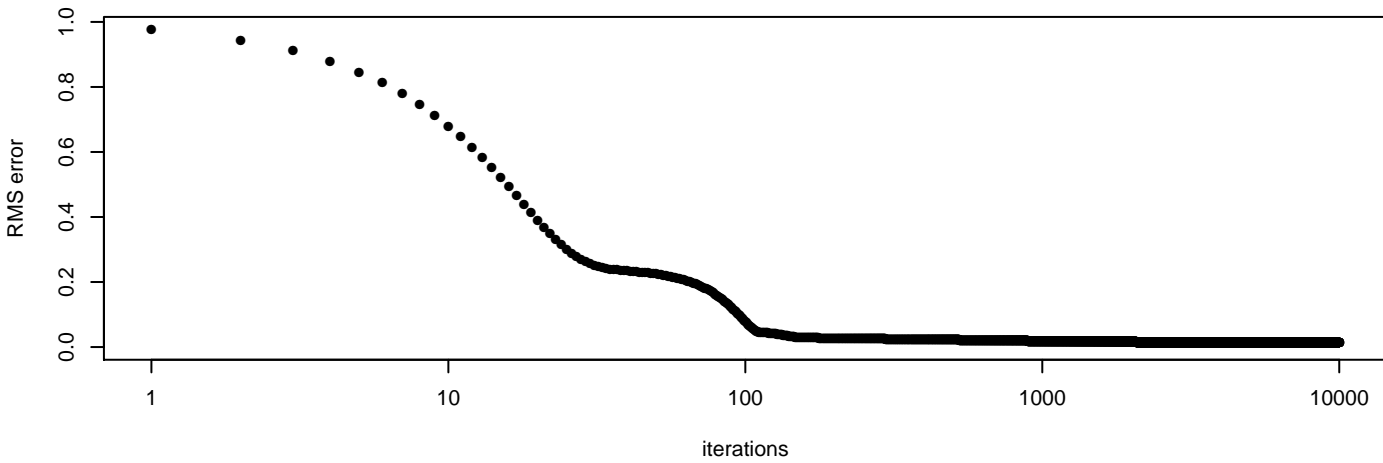
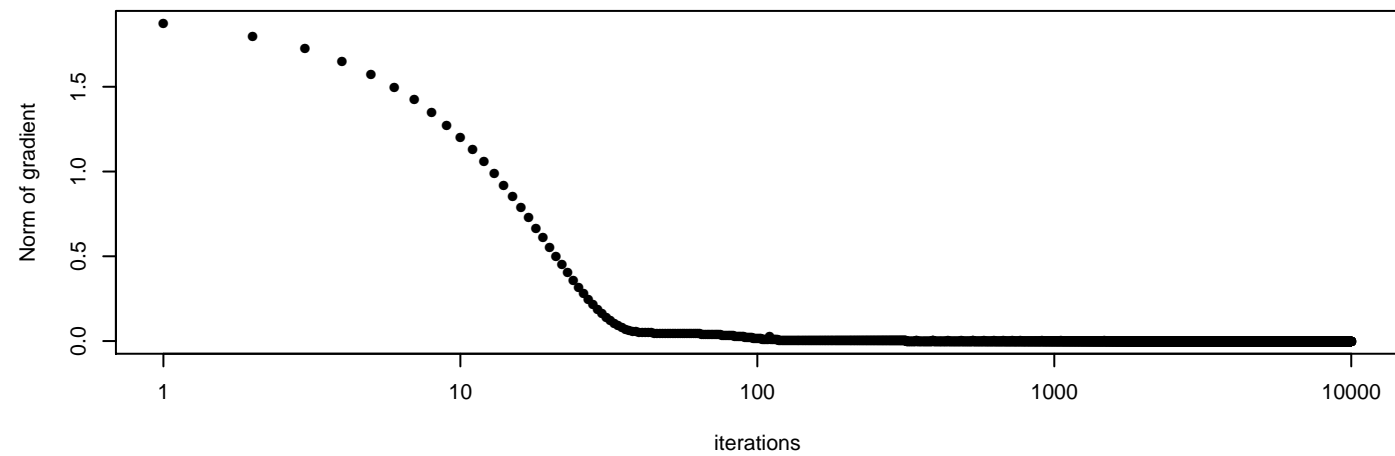
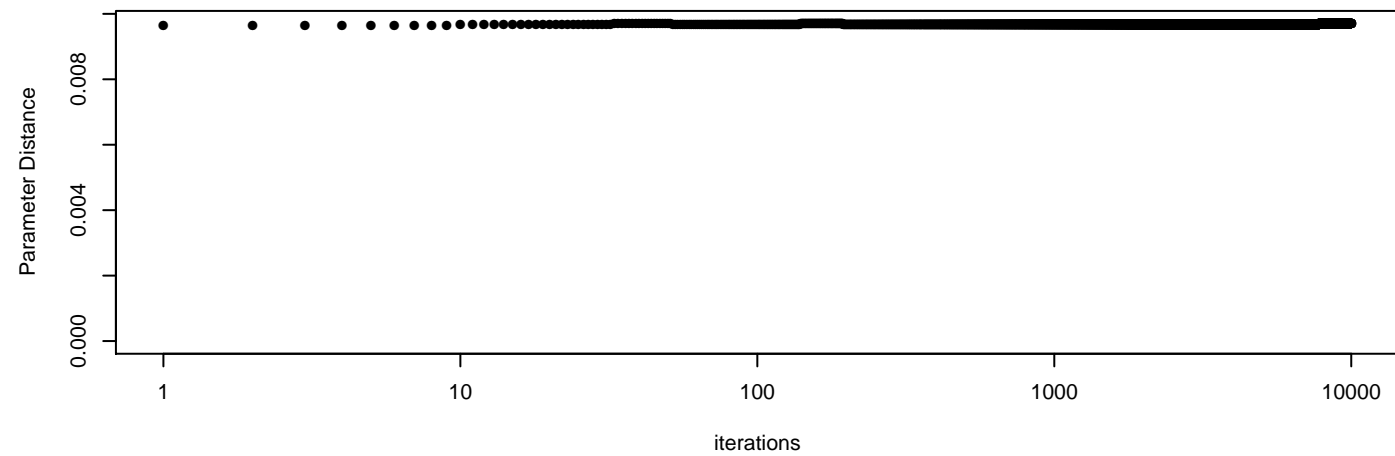
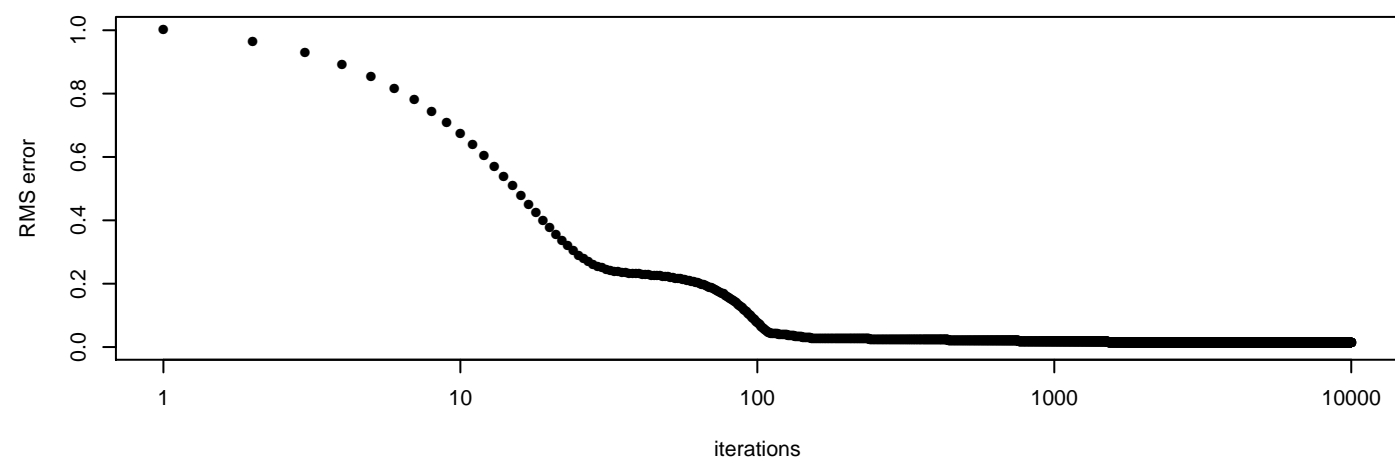
Negative Perturbation

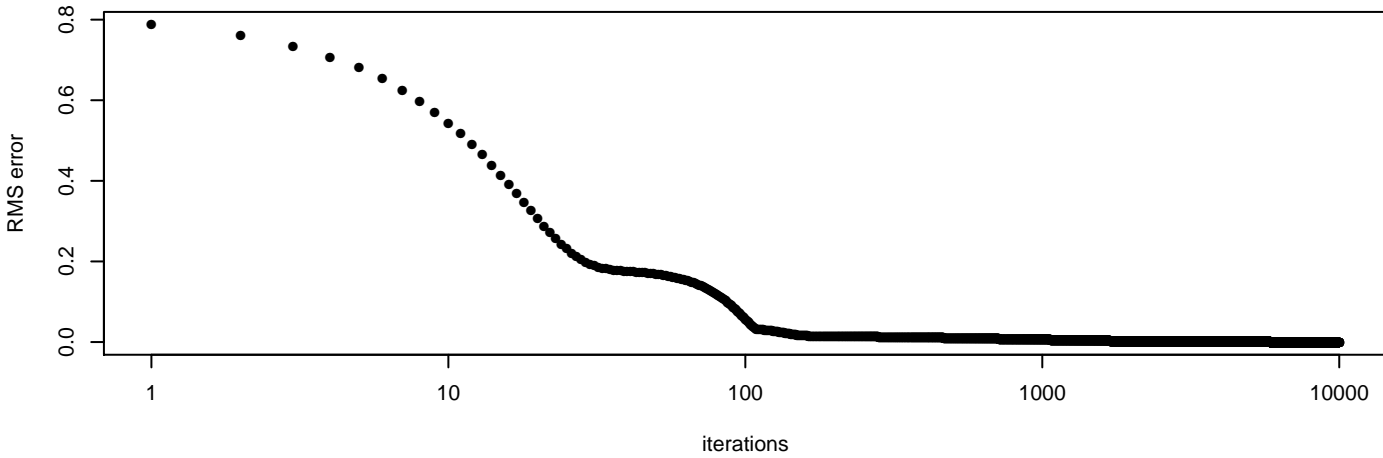
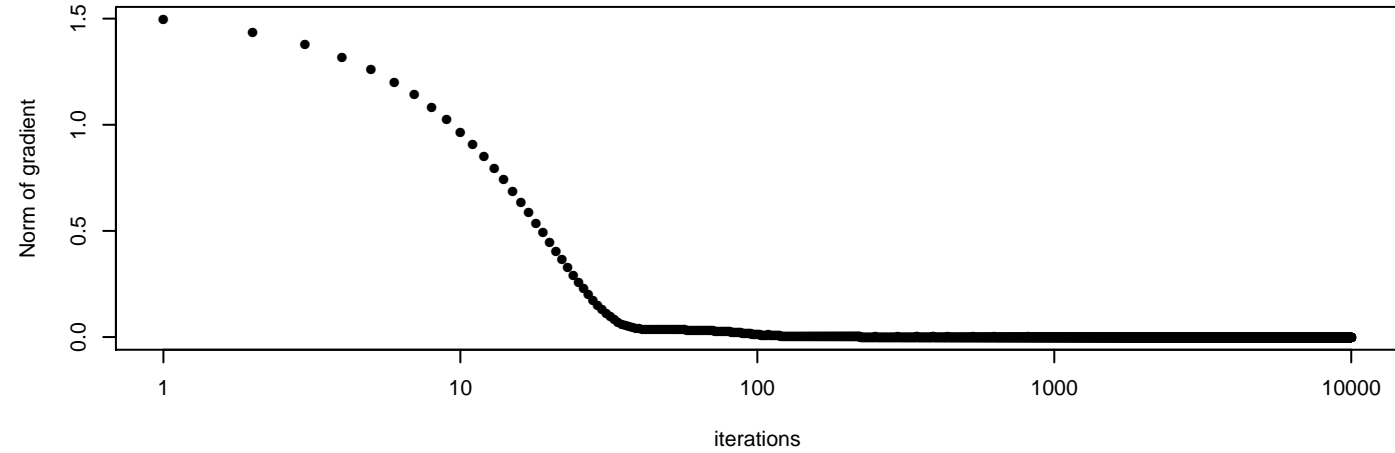
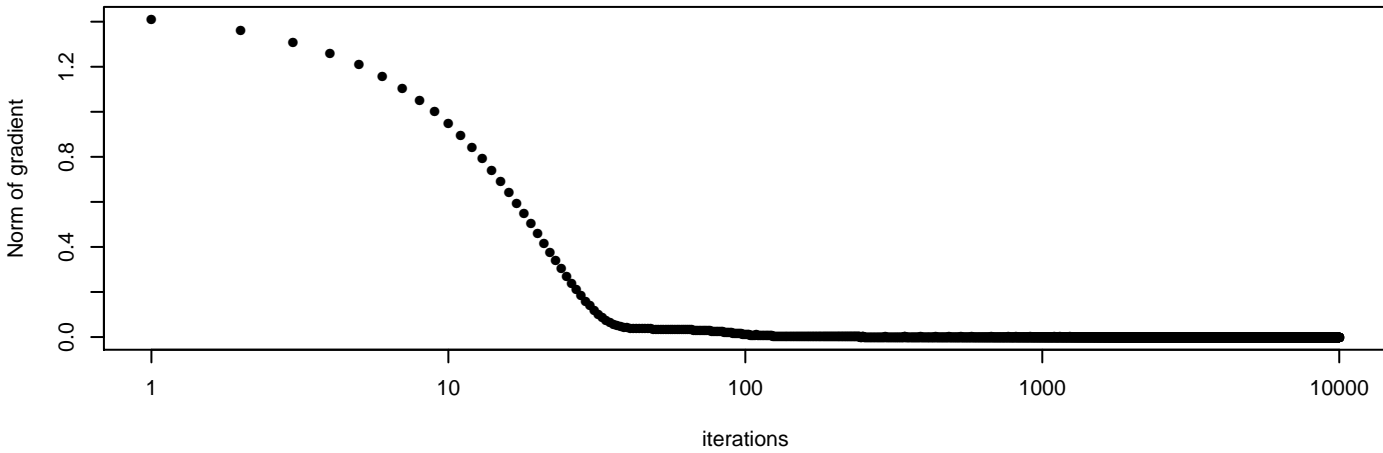
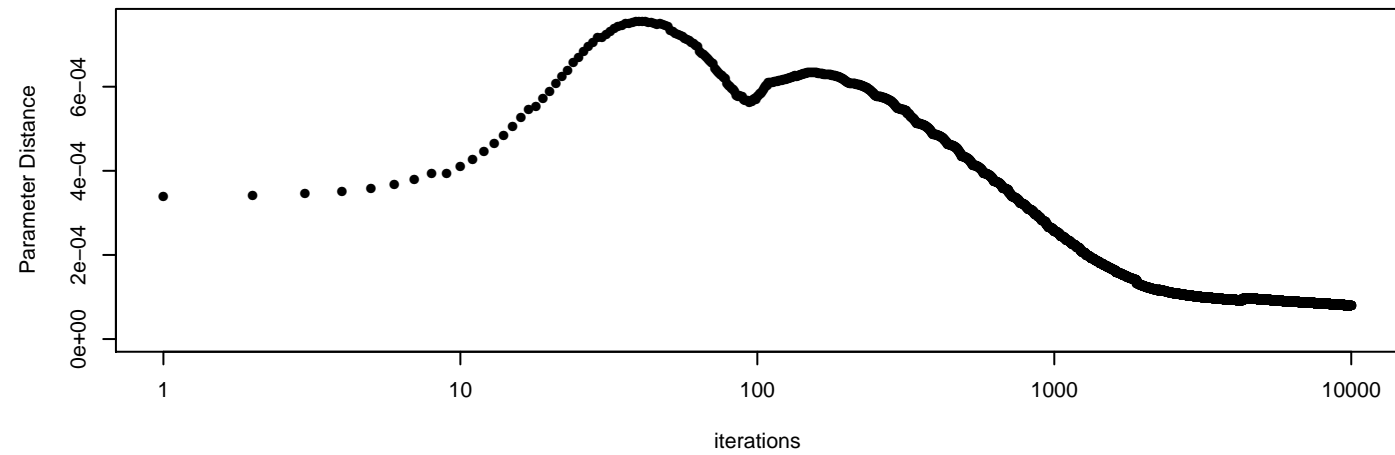
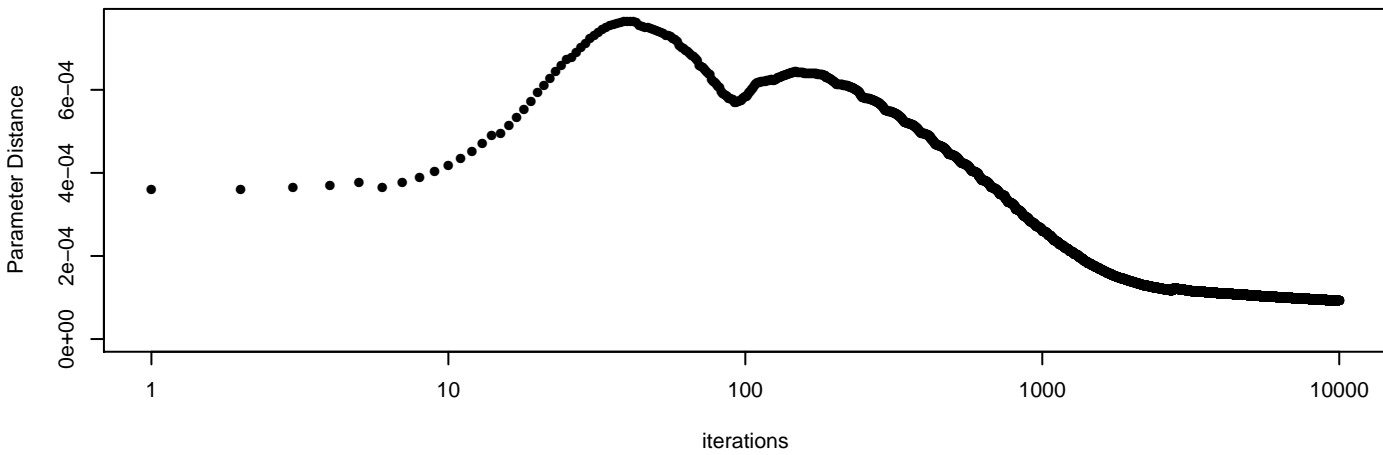
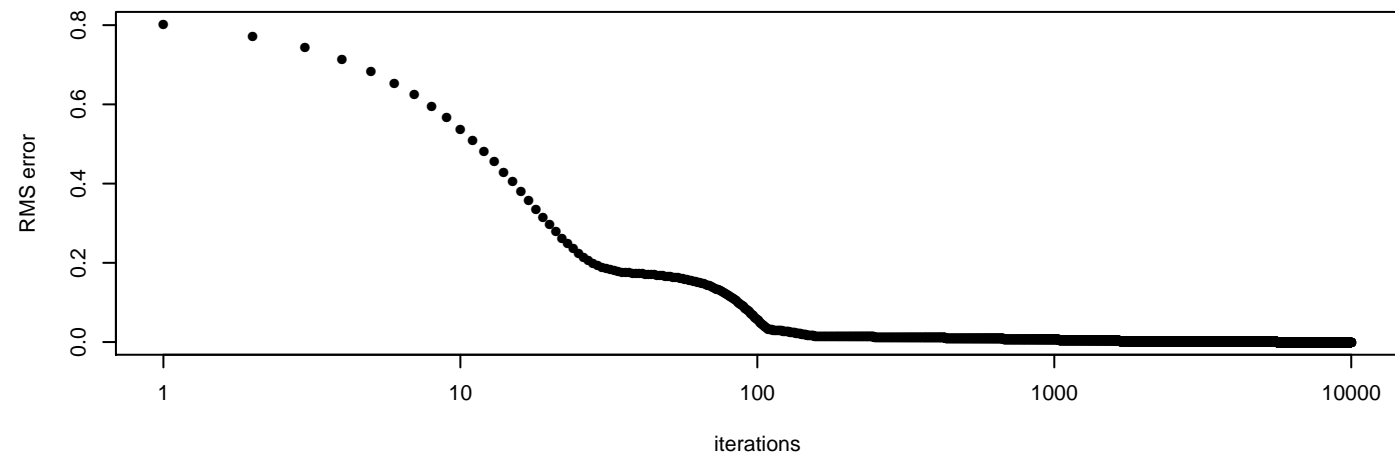


Parameter3

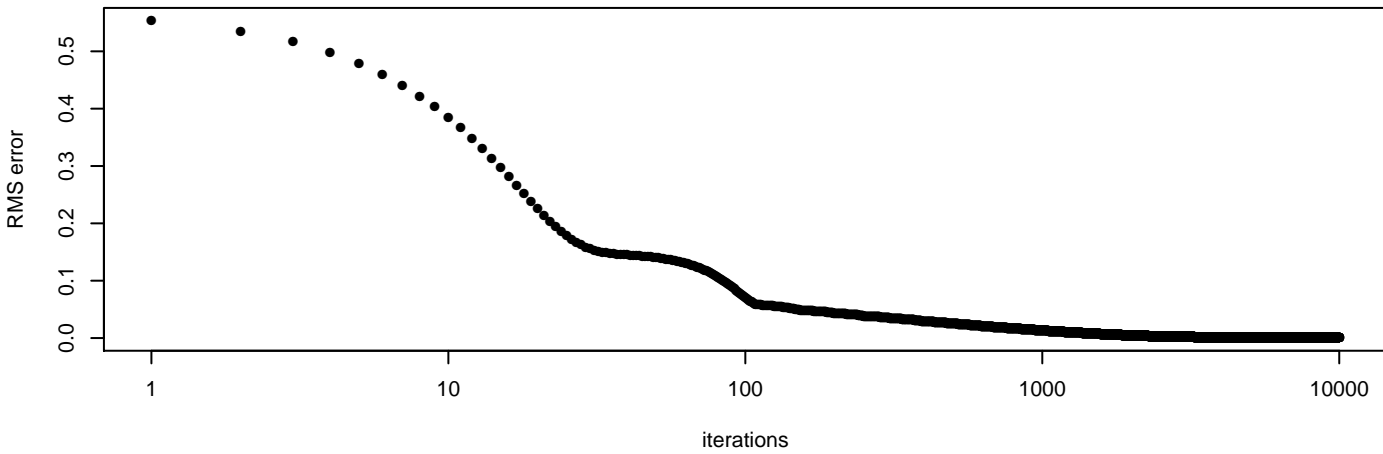
Positive Perturbation



Negative Perturbation**Parameter4****Positive Perturbation**

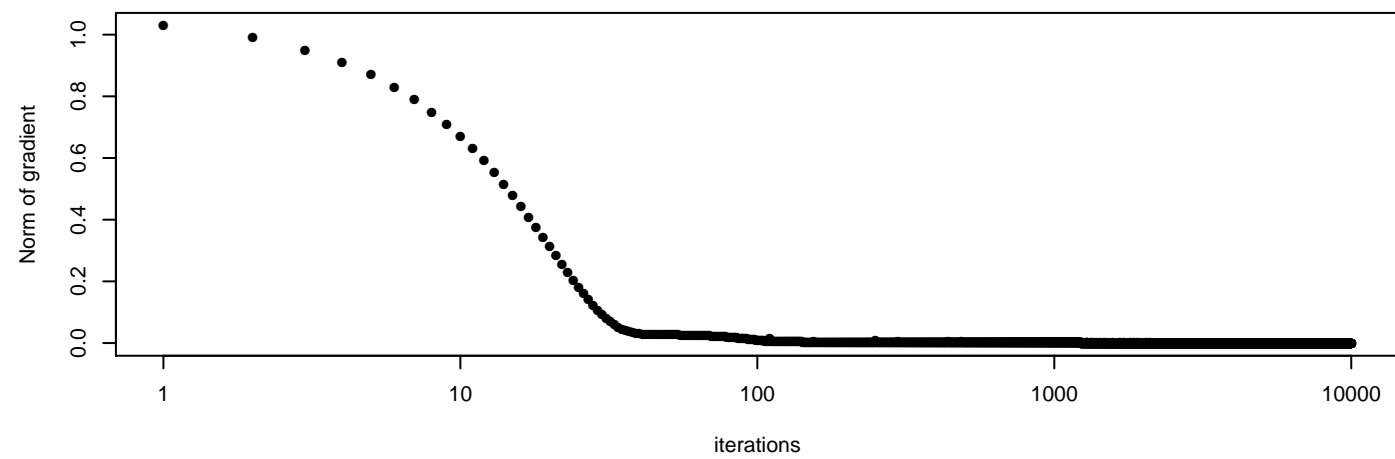
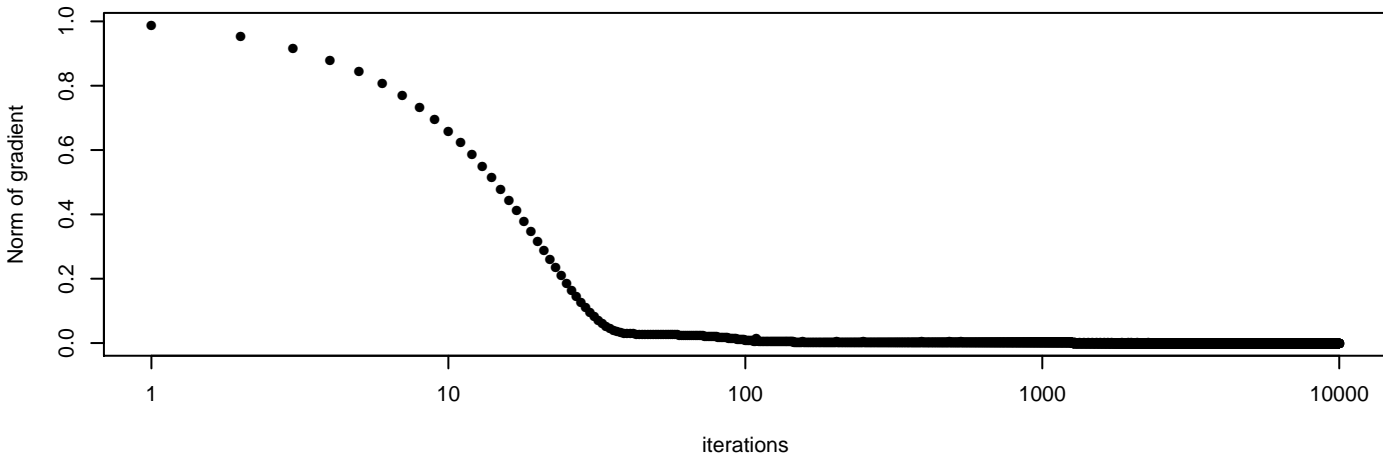
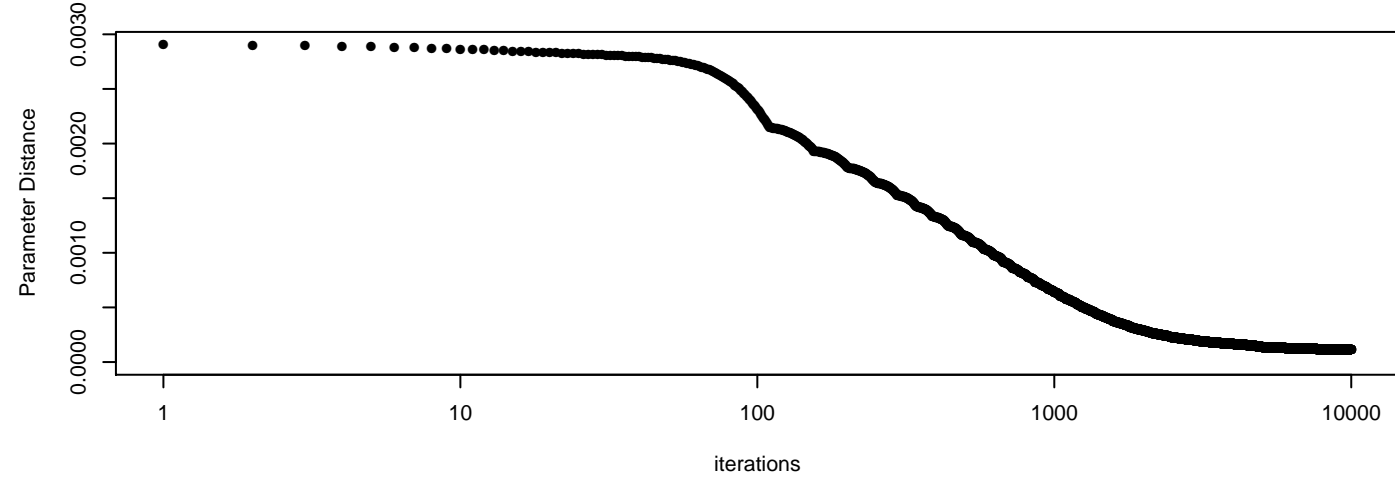
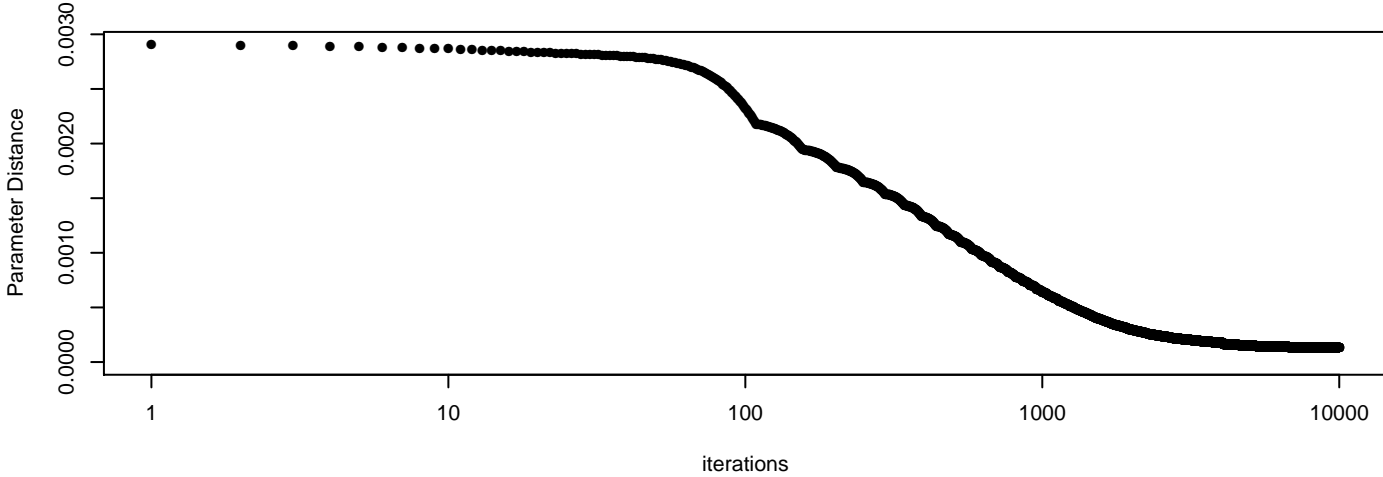
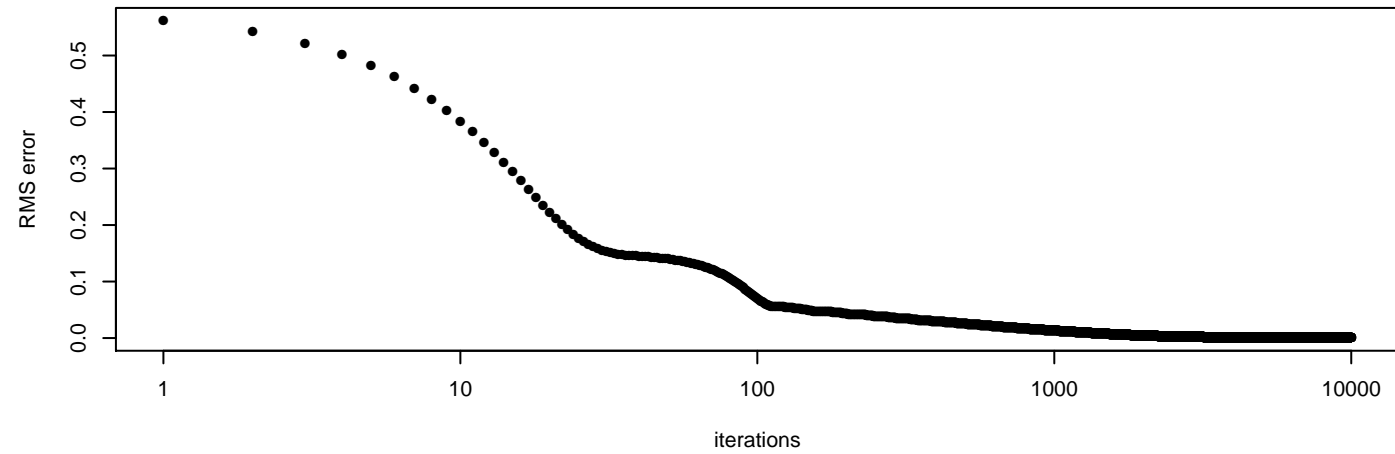
Negative Perturbation**Parameter5****Positive Perturbation**

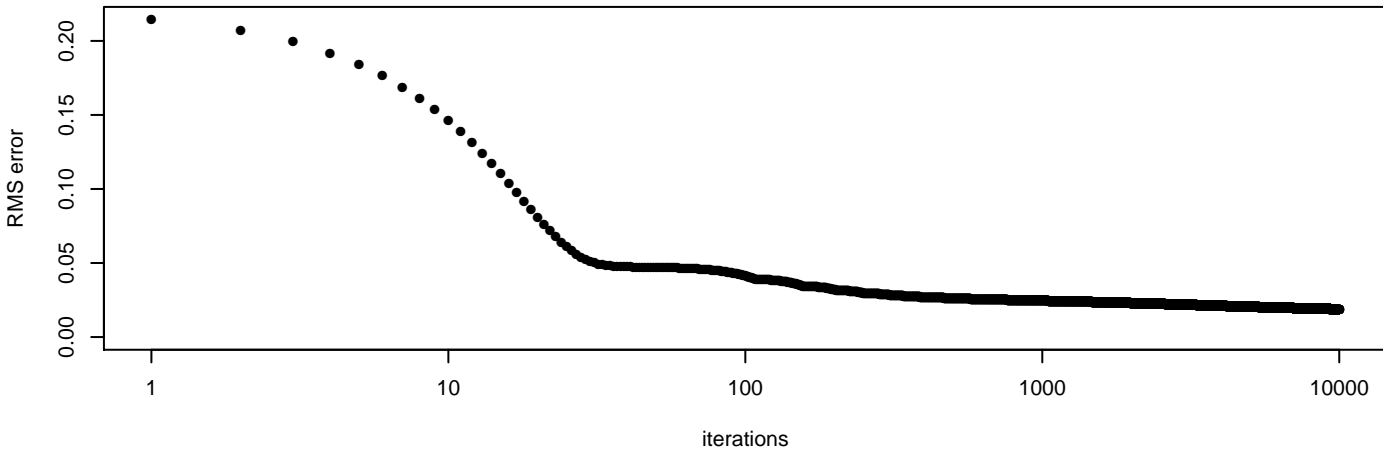
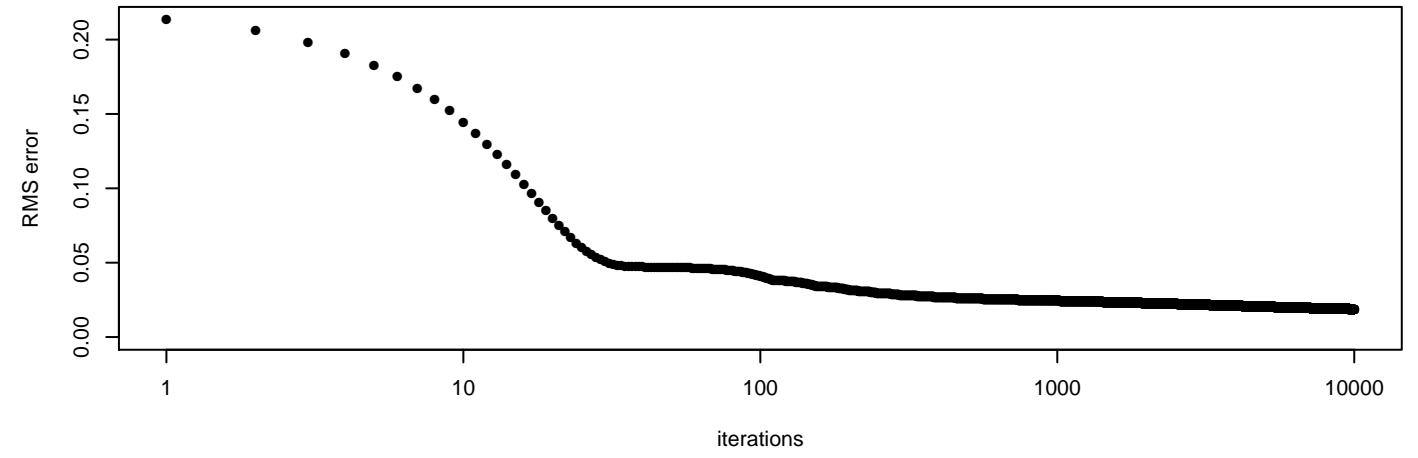
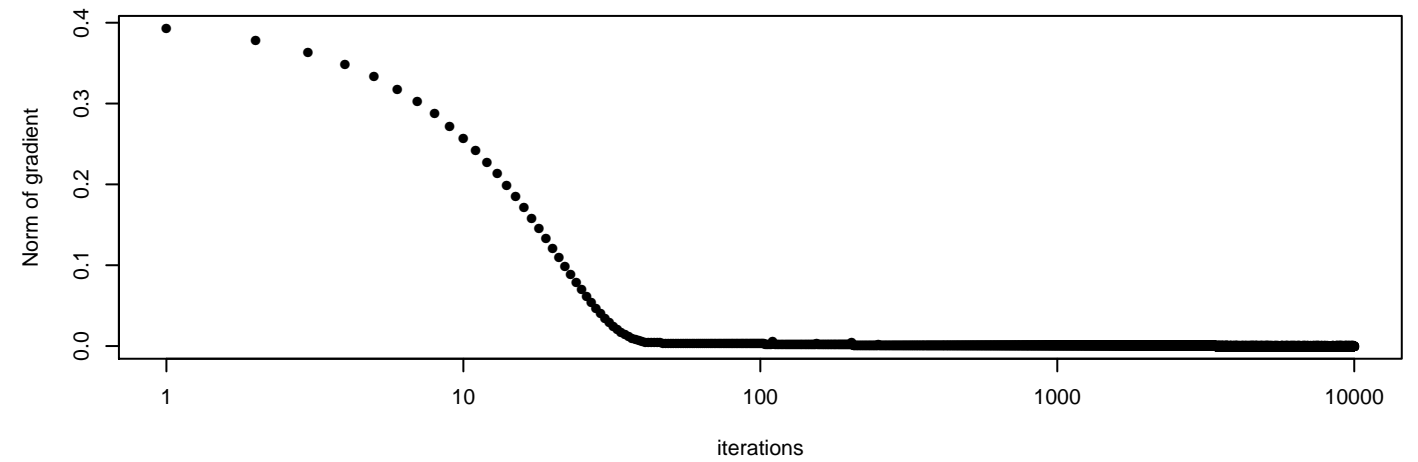
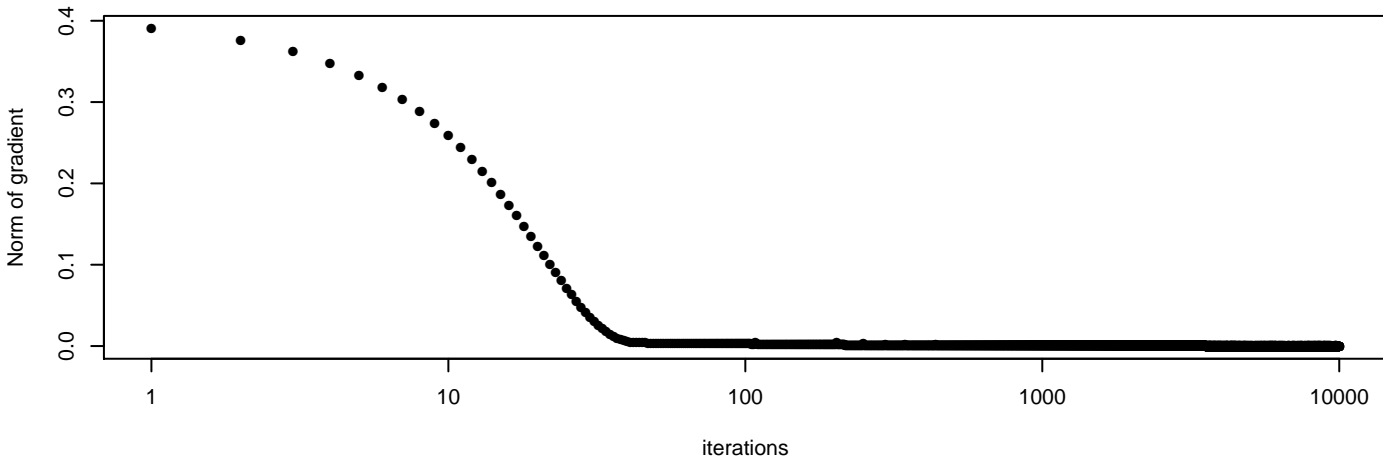
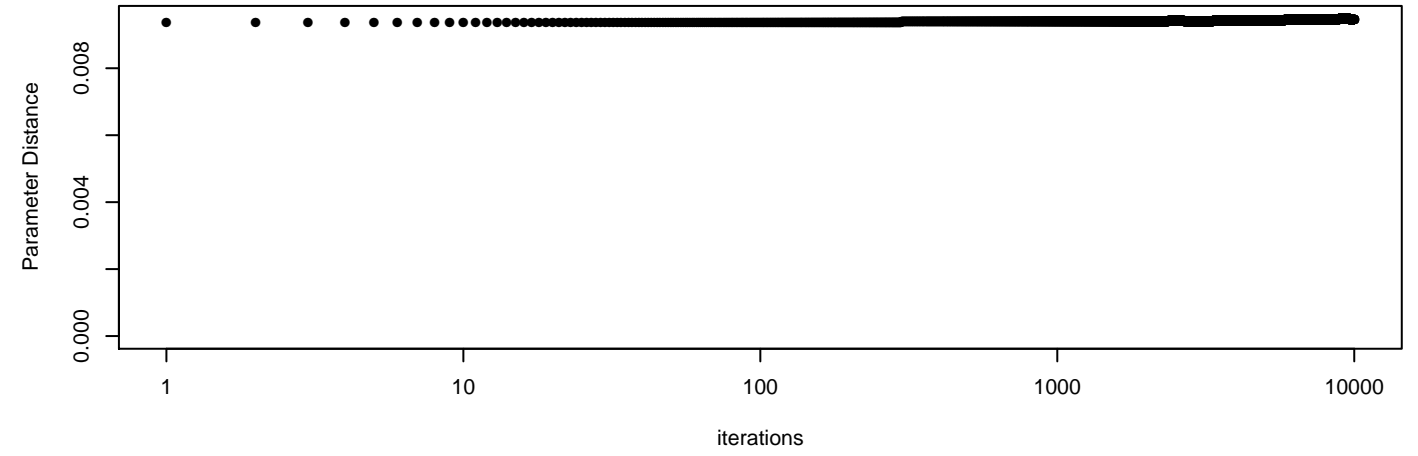
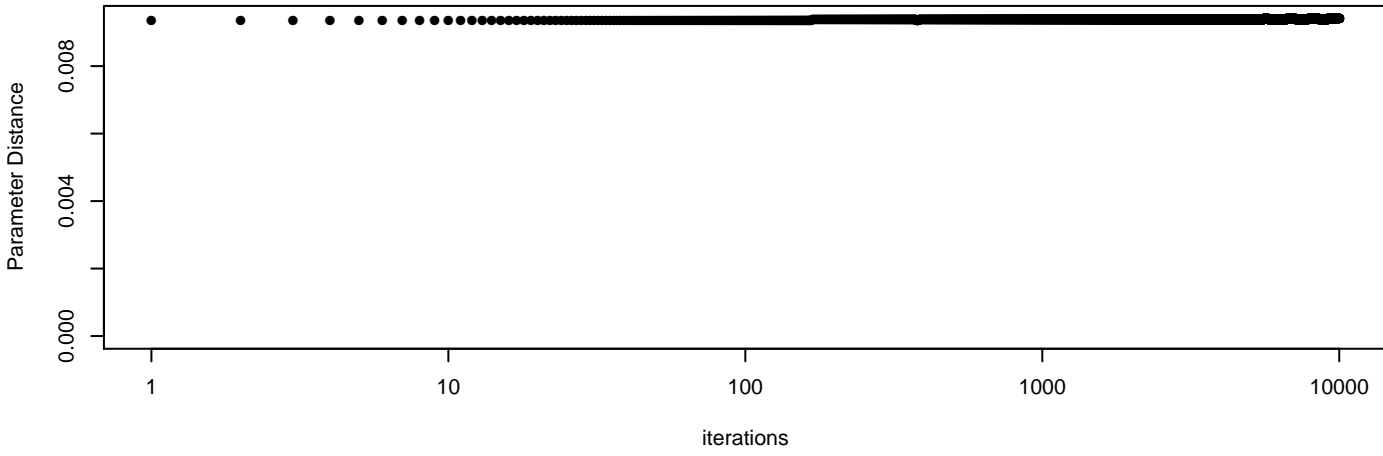
Negative Perturbation



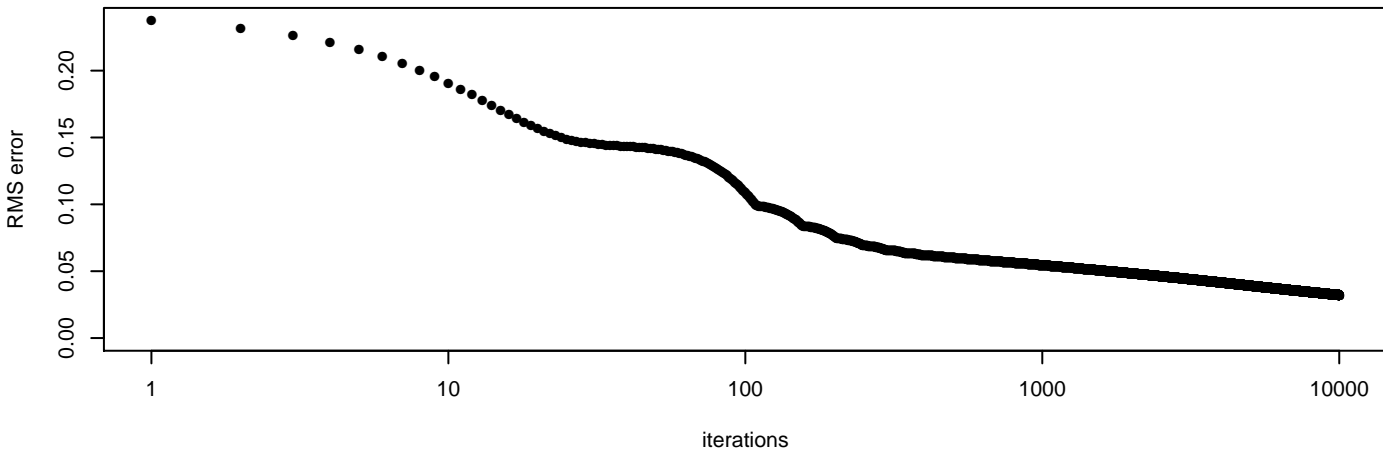
Parameter6

Positive Perturbation



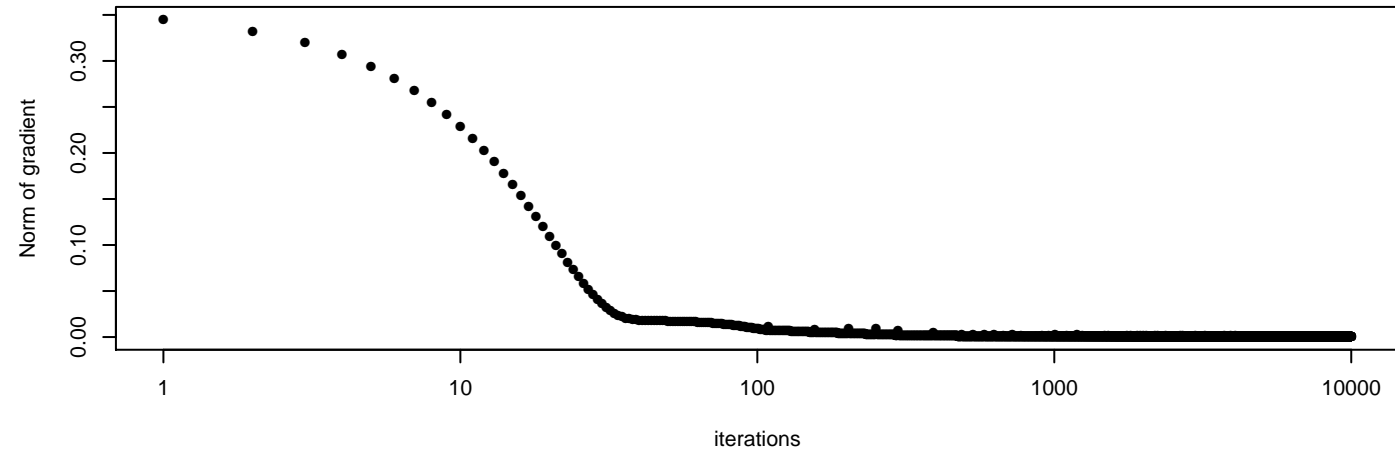
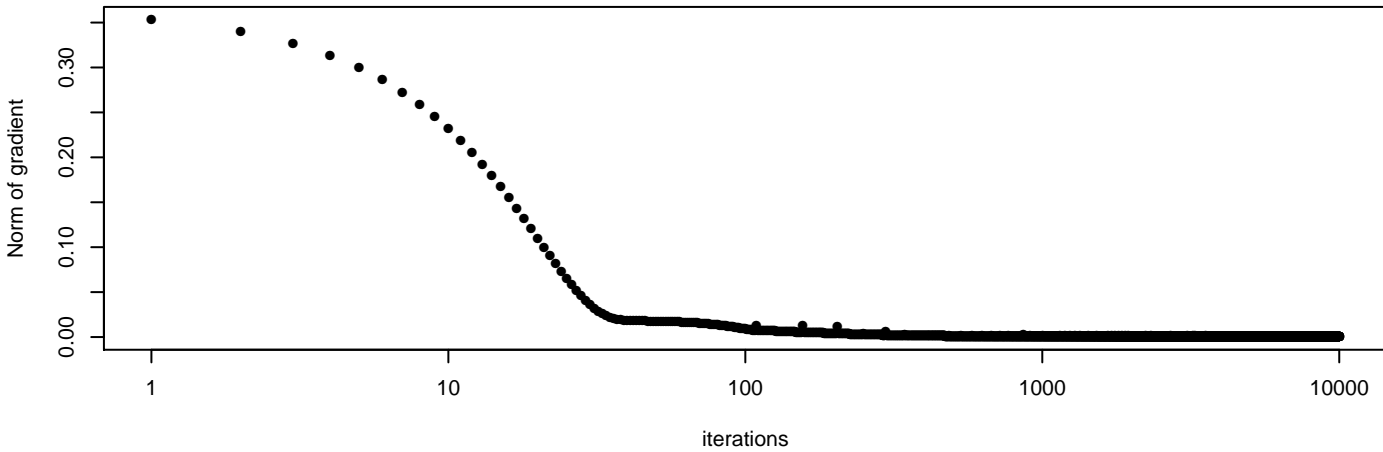
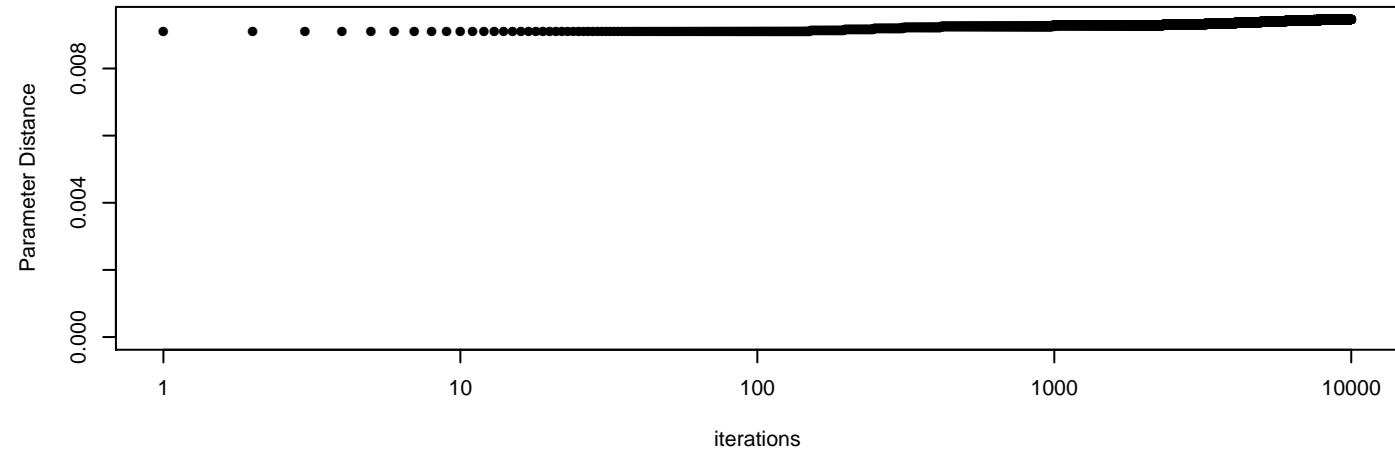
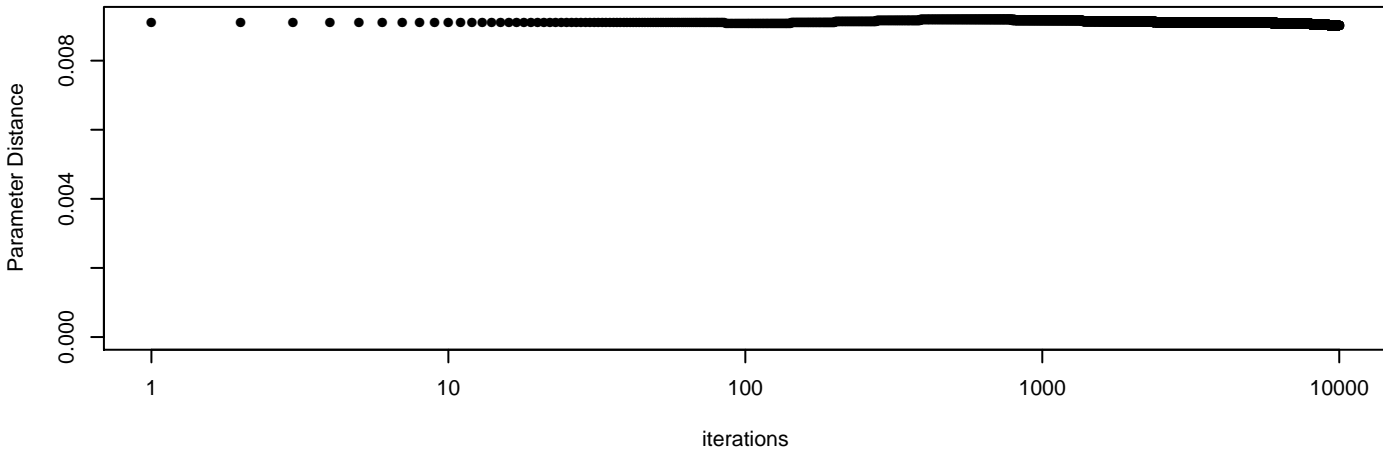
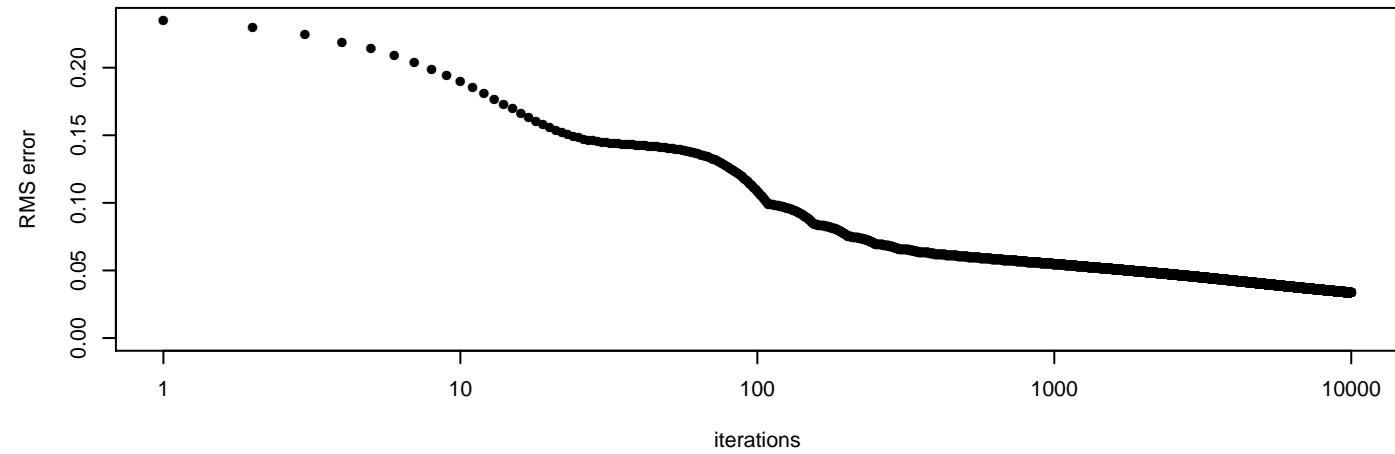
Negative Perturbation**Parameter7****Positive Perturbation**

Negative Perturbation

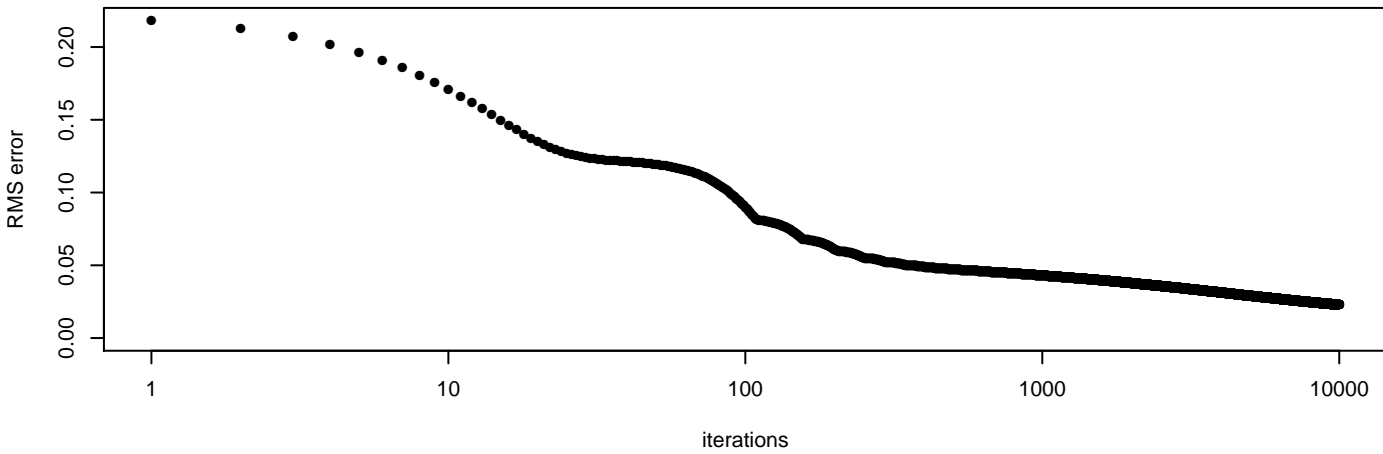


Parameter8

Positive Perturbation

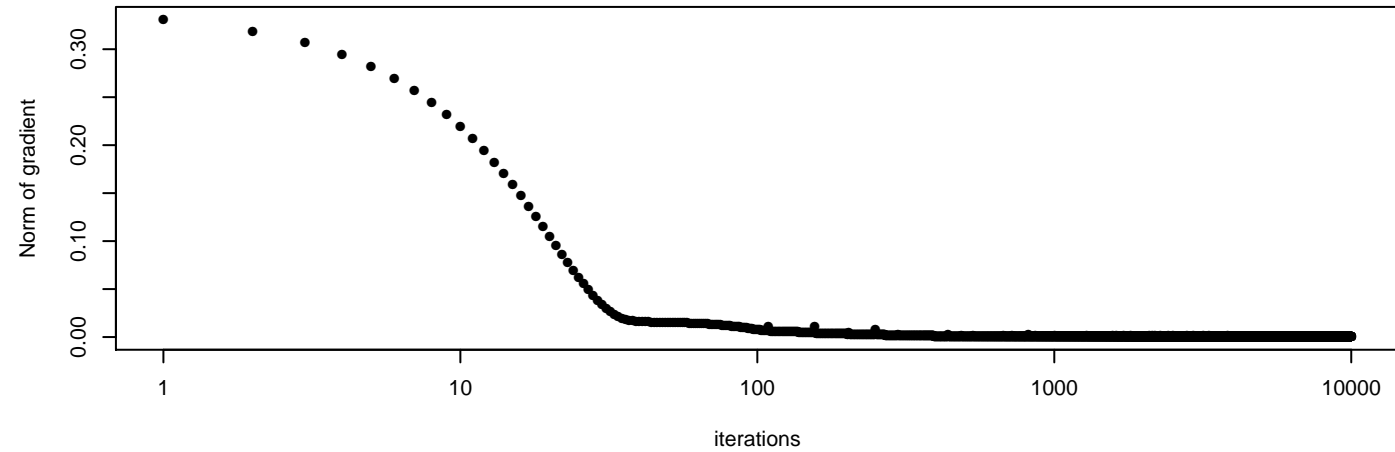
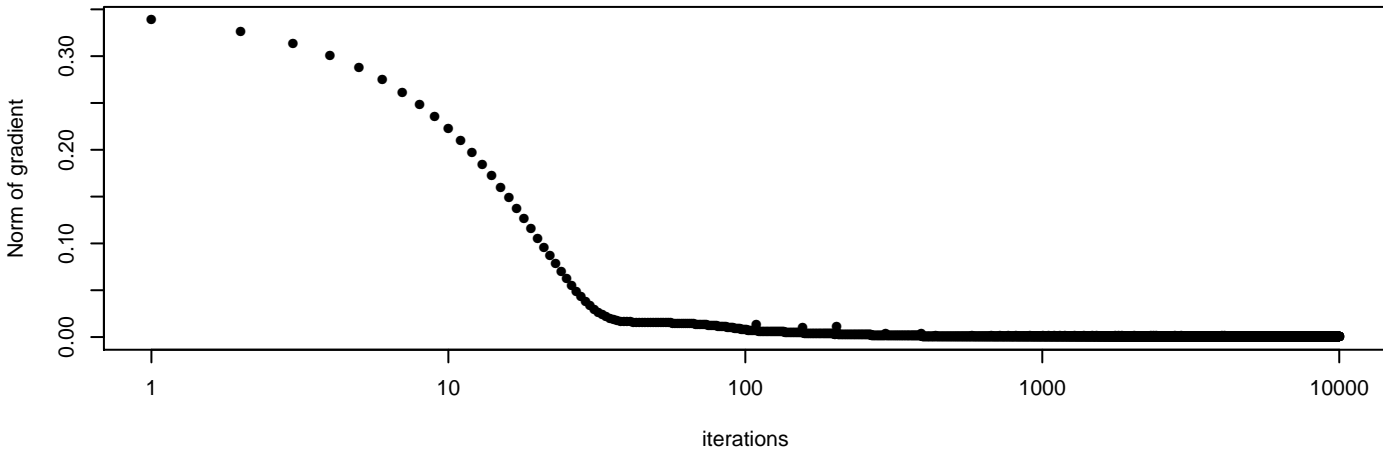
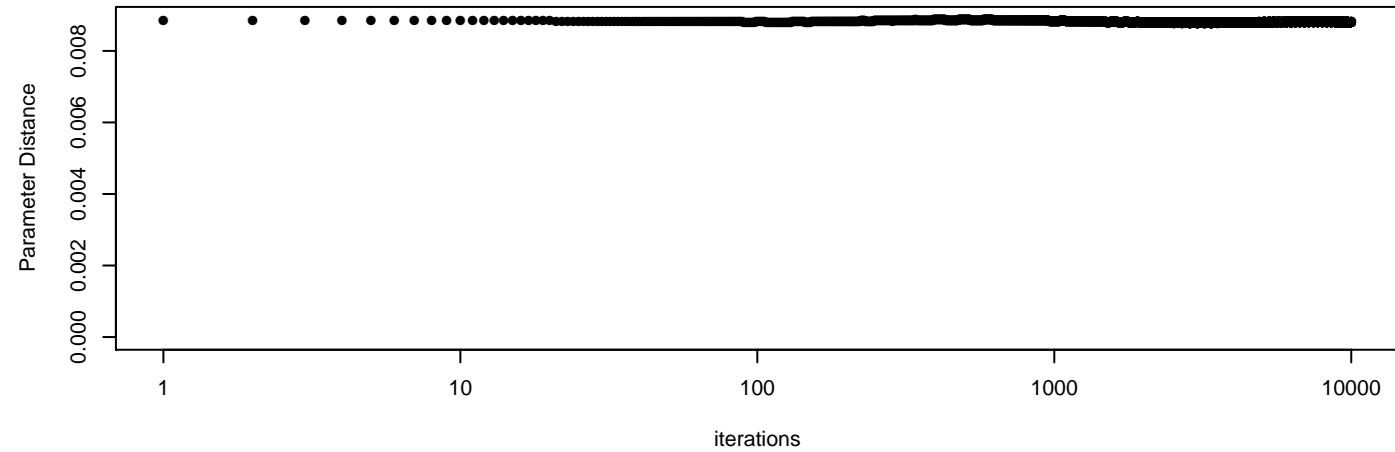
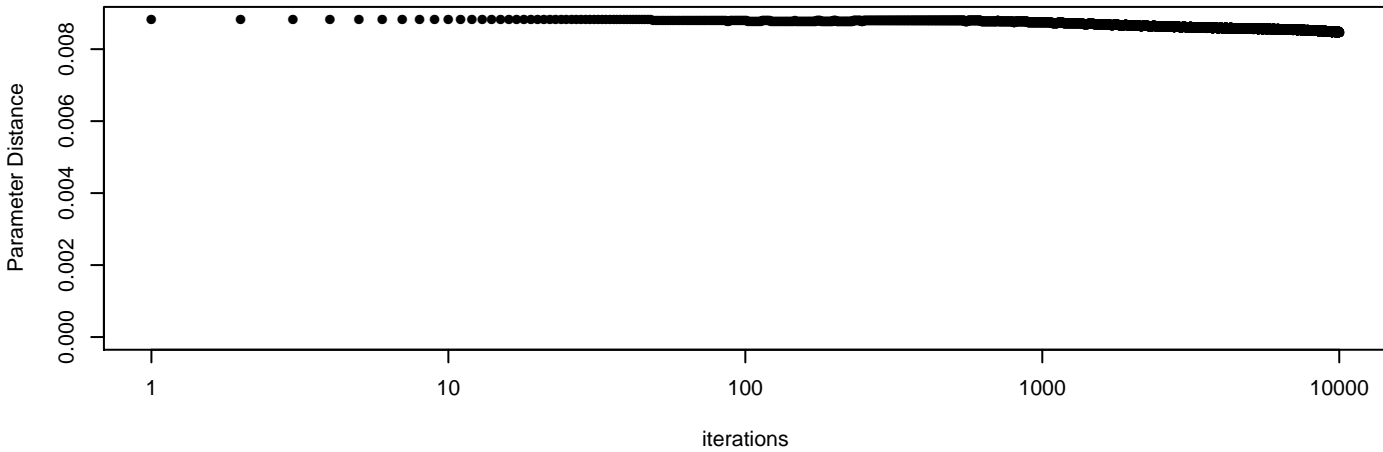
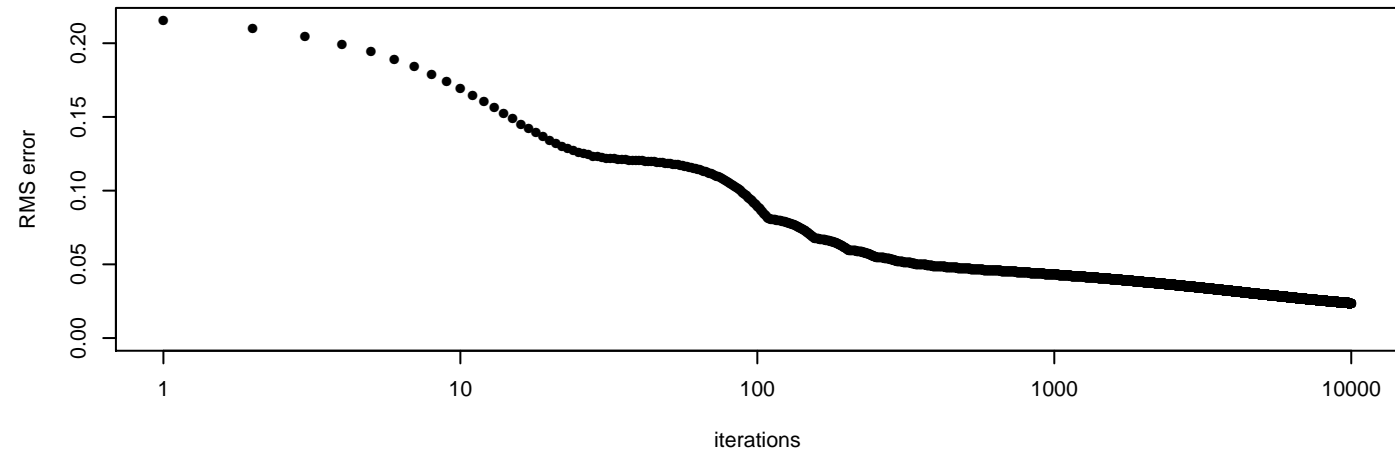


Negative Perturbation



Parameter9

Positive Perturbation

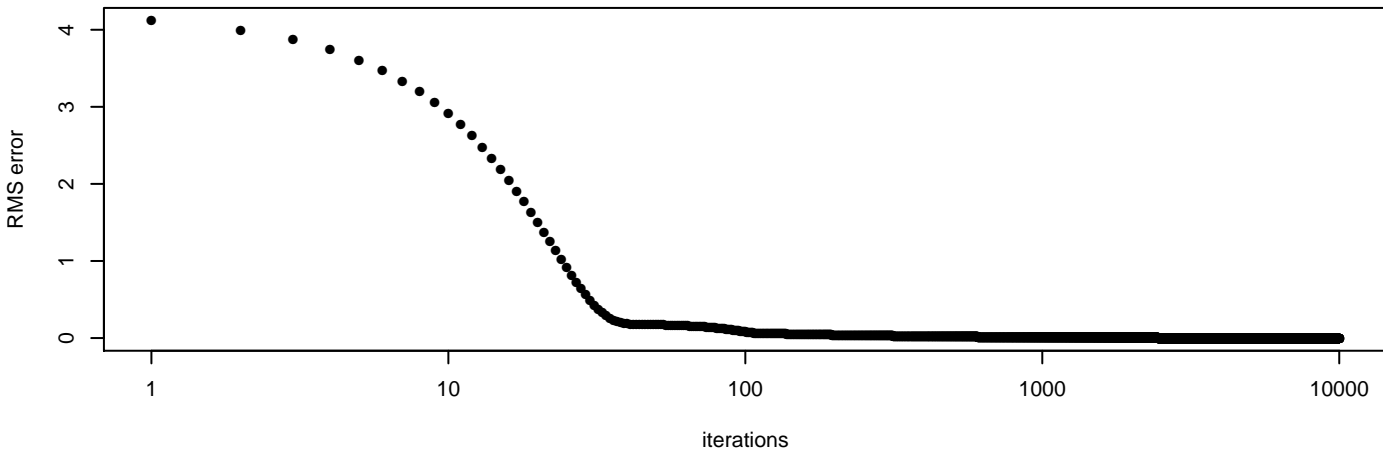


Convergence of a parameter set where each parameter is perturbed by * 10% * individually.

Gradient descent is executed with all parameters * free* to be changed by GD.

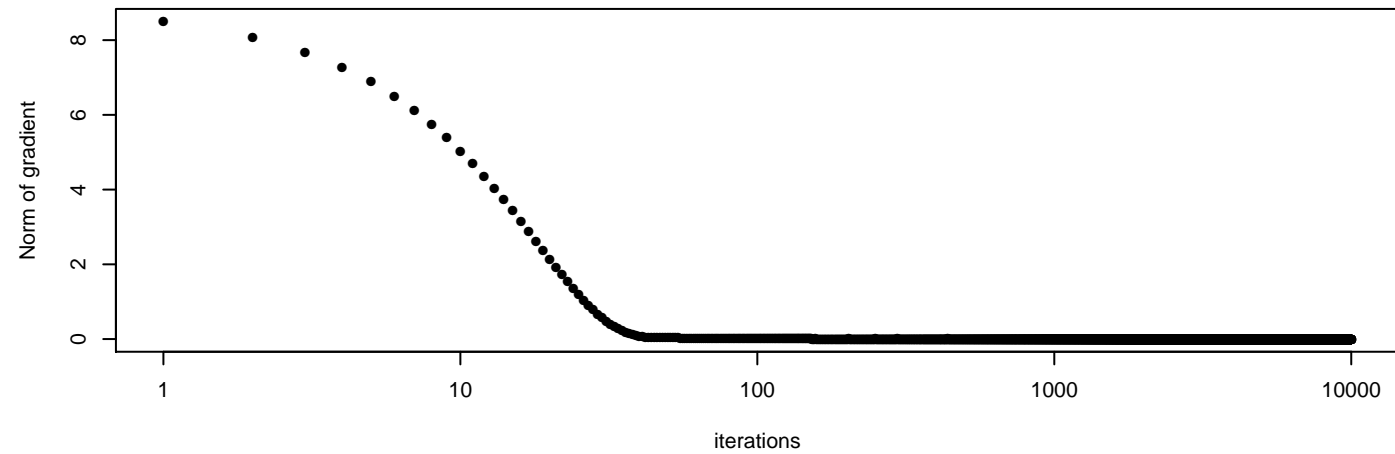
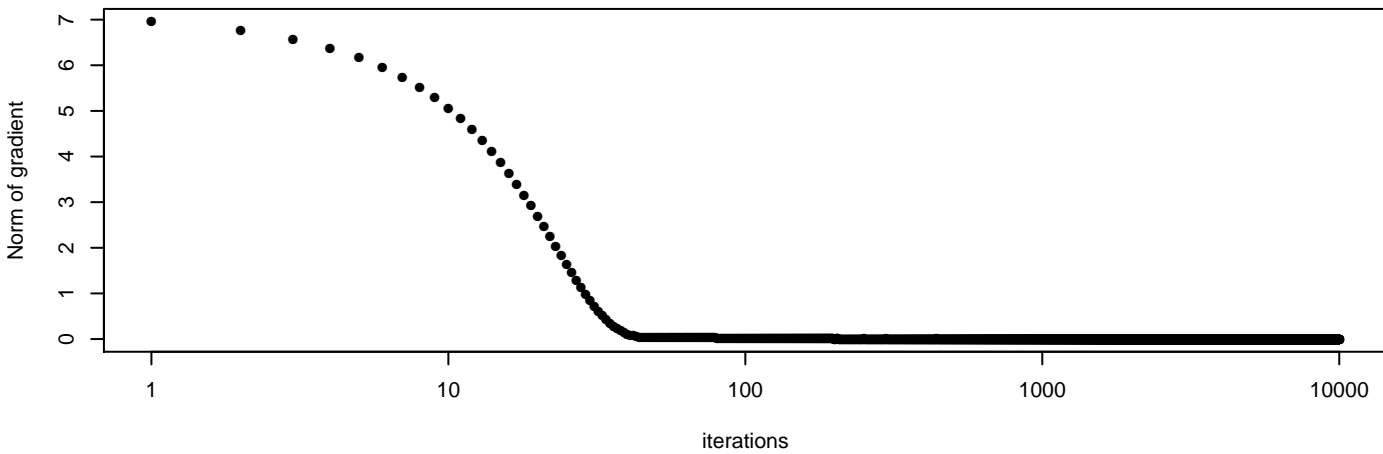
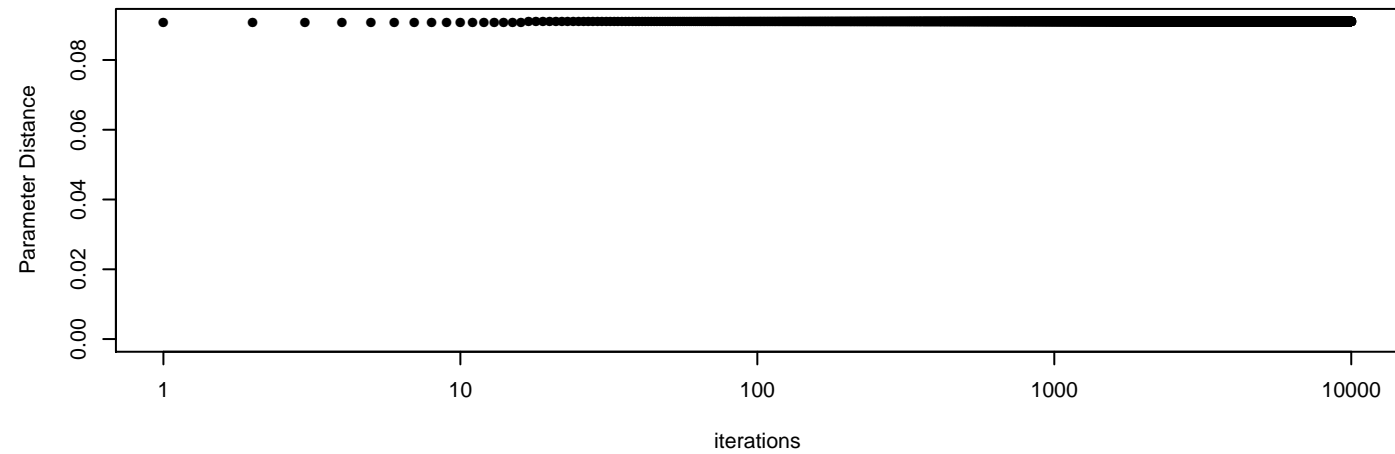
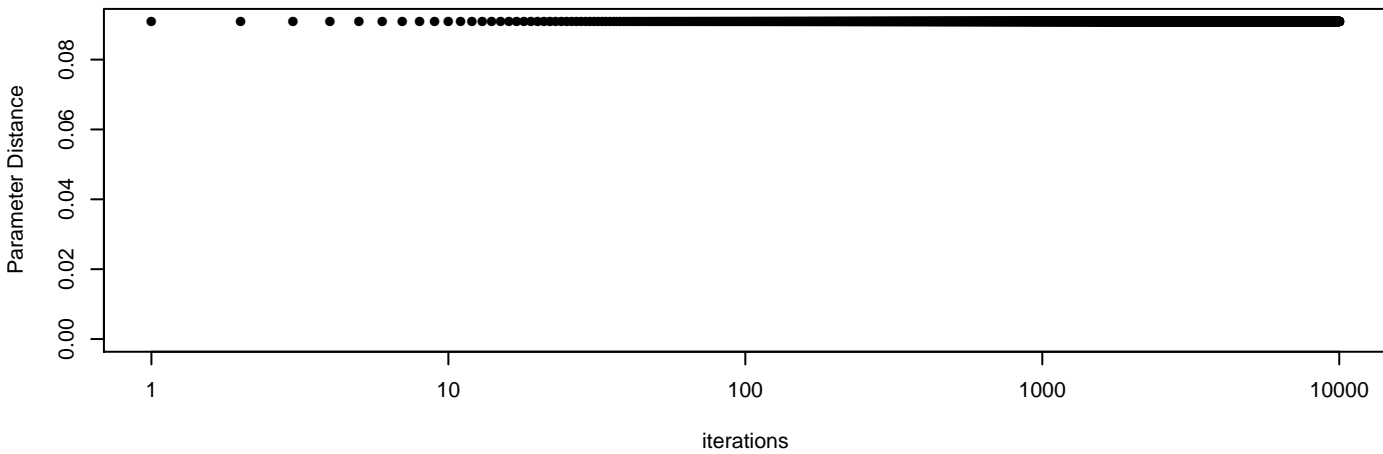
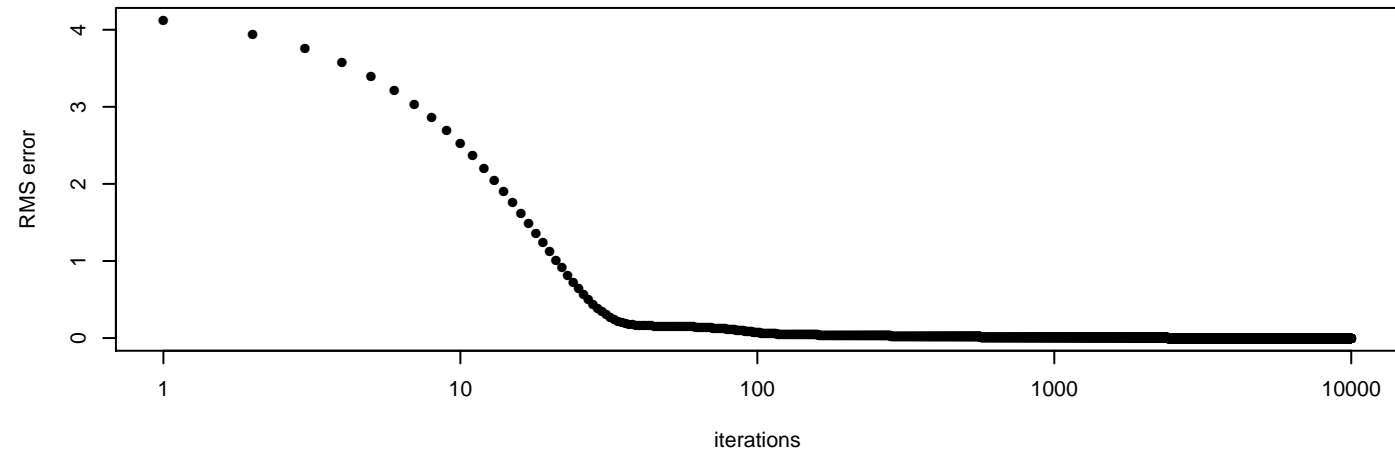
Gradient descent was ran in * normal * mode to take the value of the gradient.

Negative Perturbation

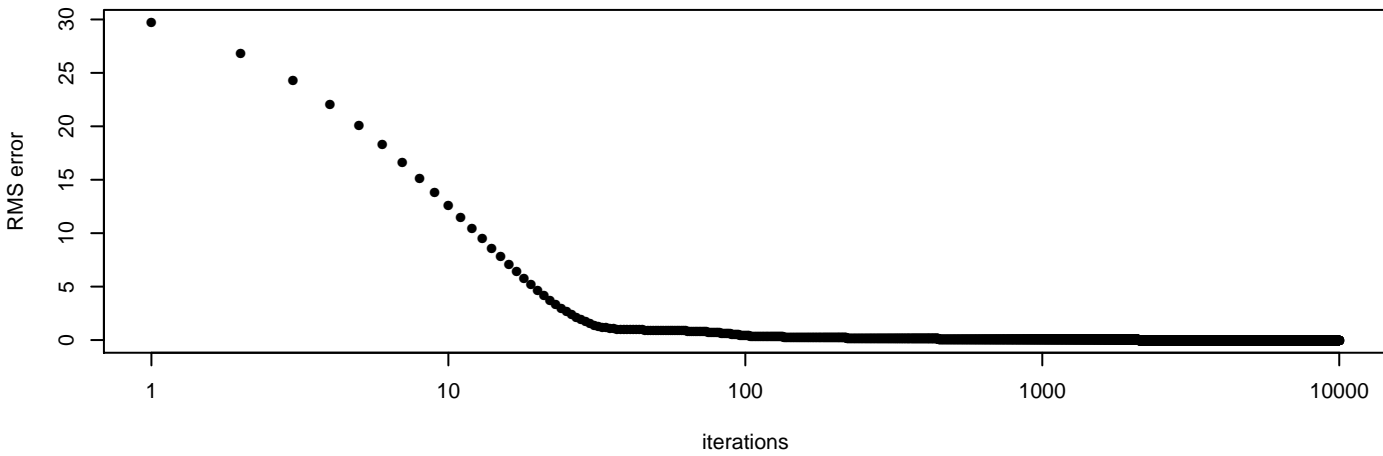


Parameter0

Positive Perturbation

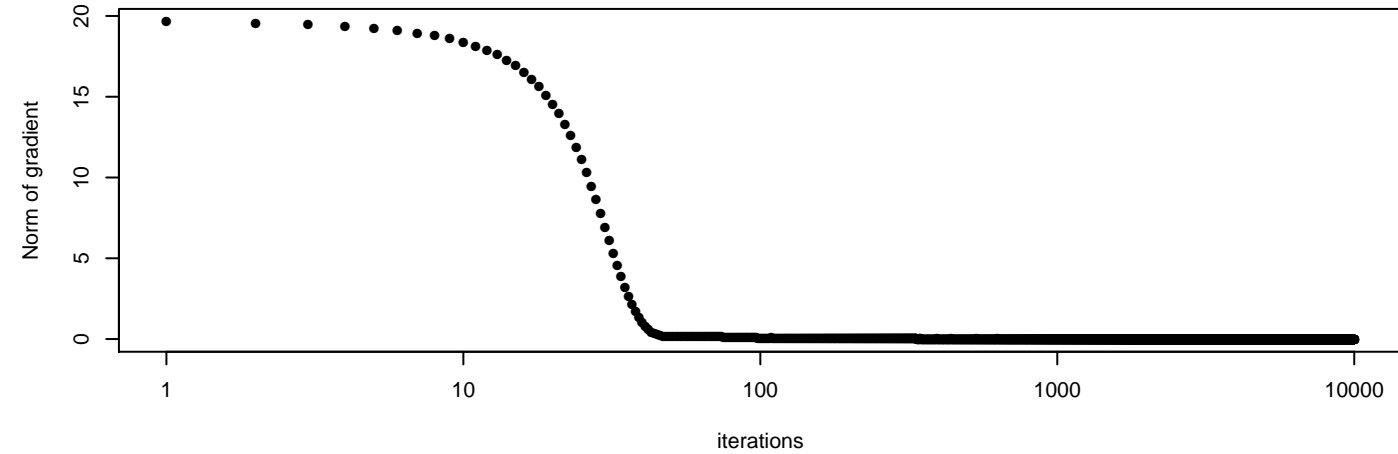
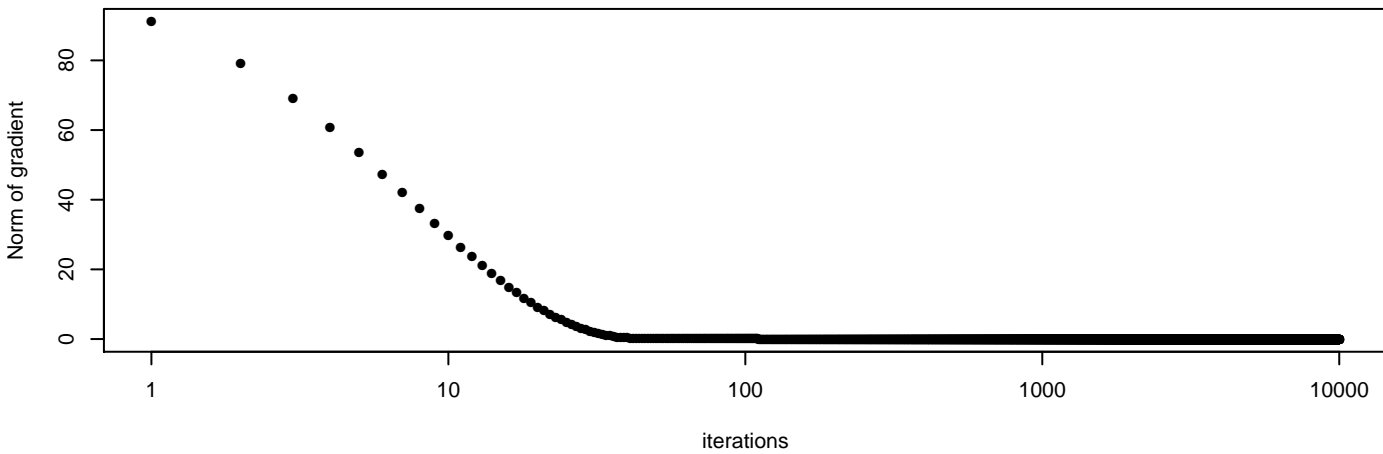
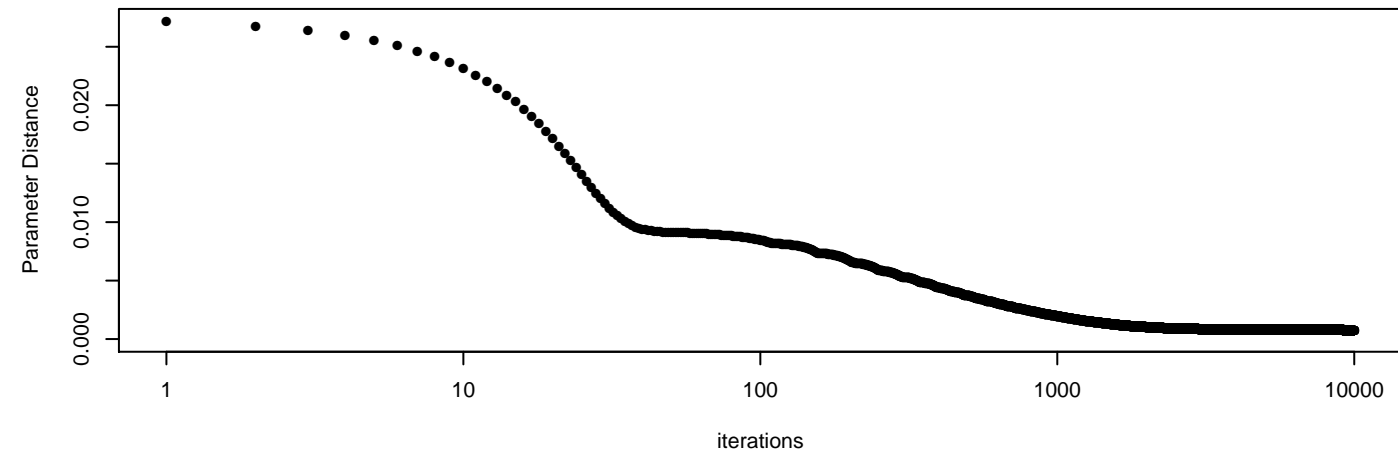
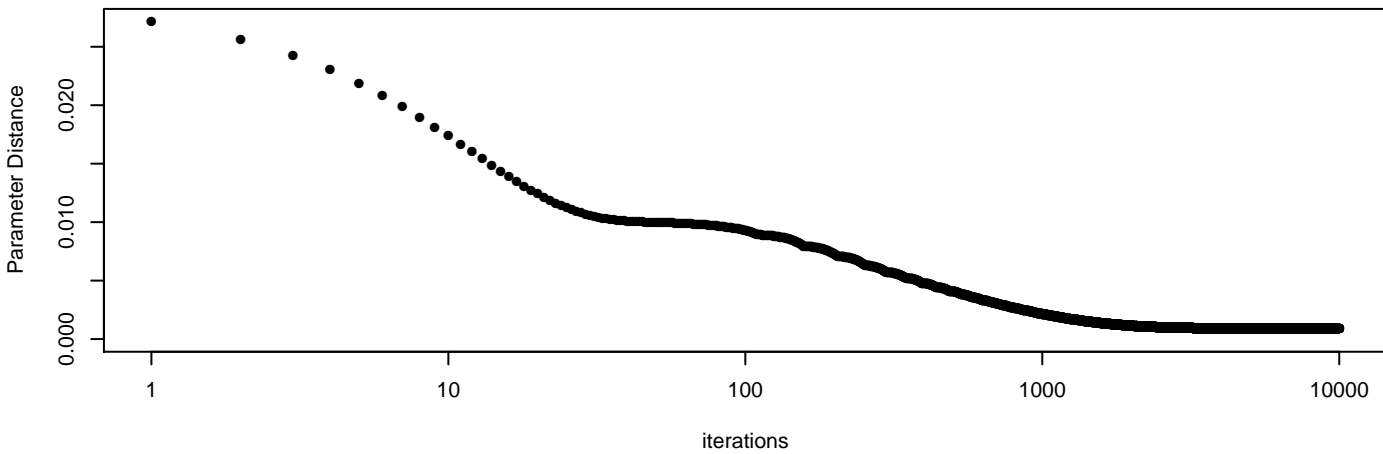
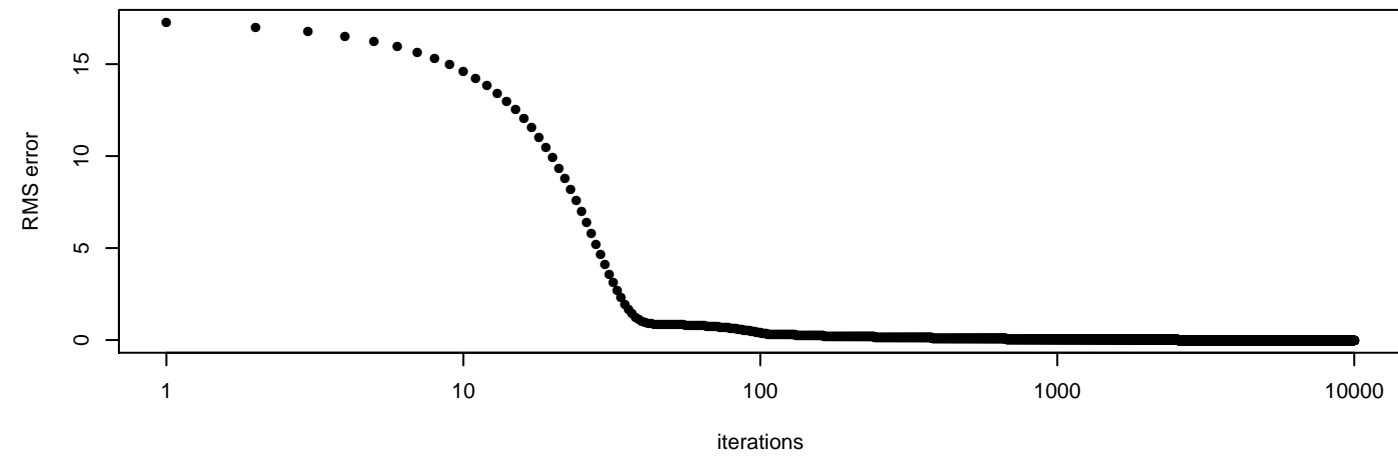


Negative Perturbation

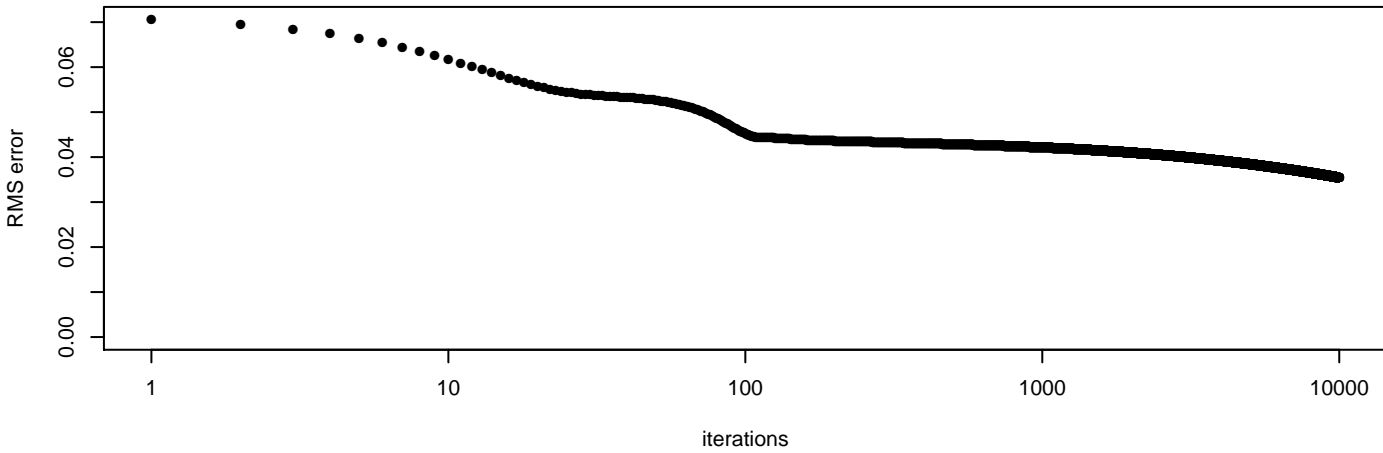


Parameter1

Positive Perturbation

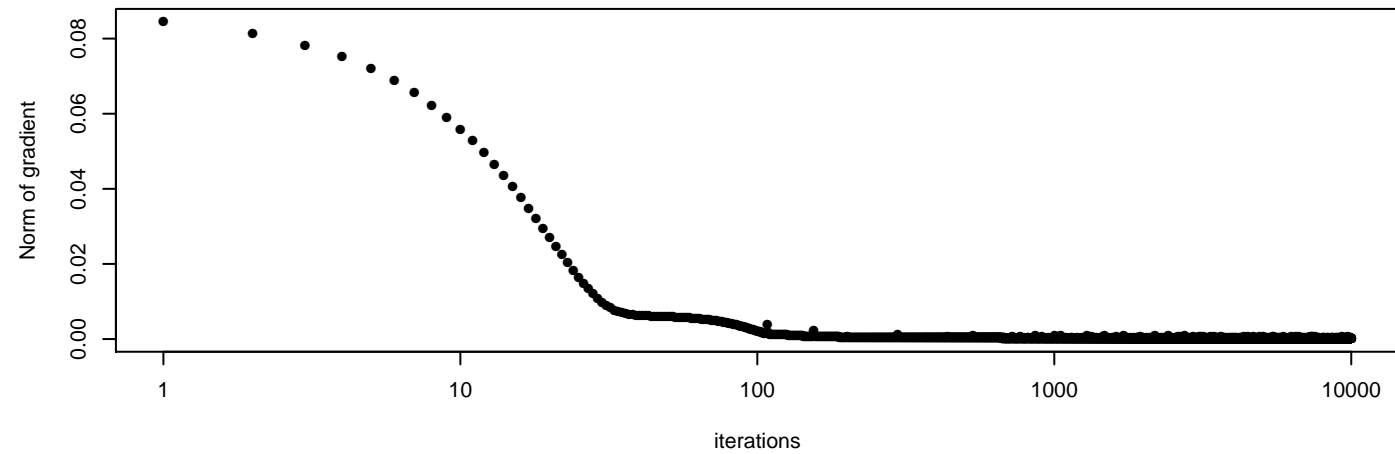
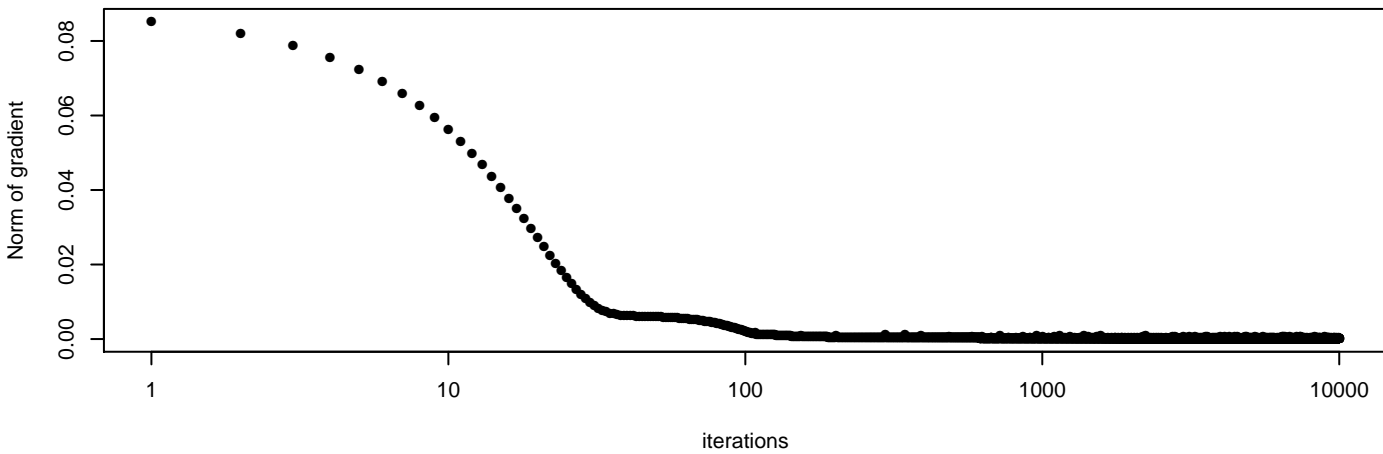
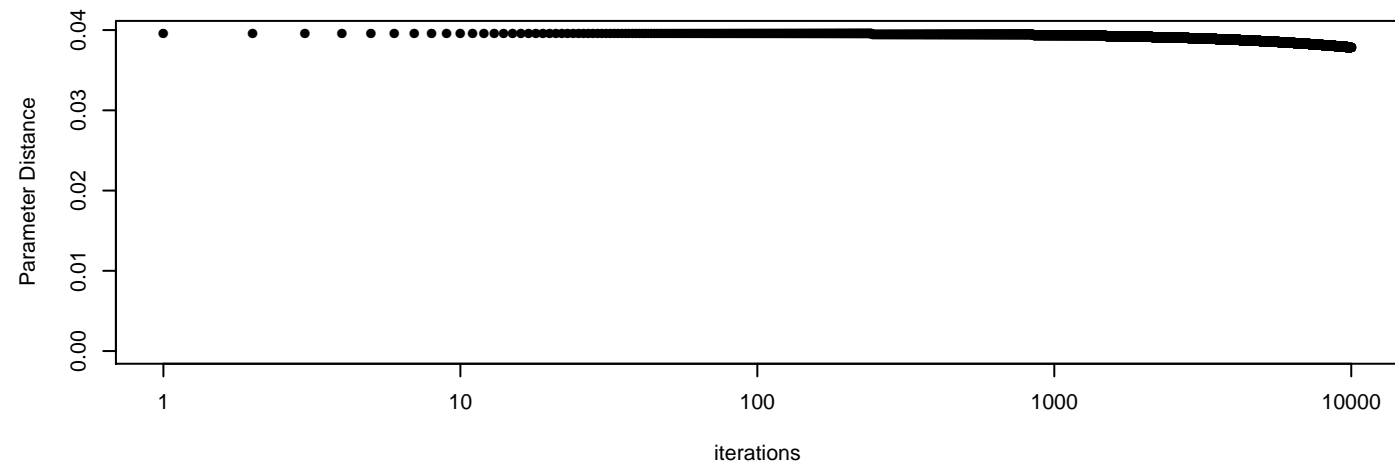
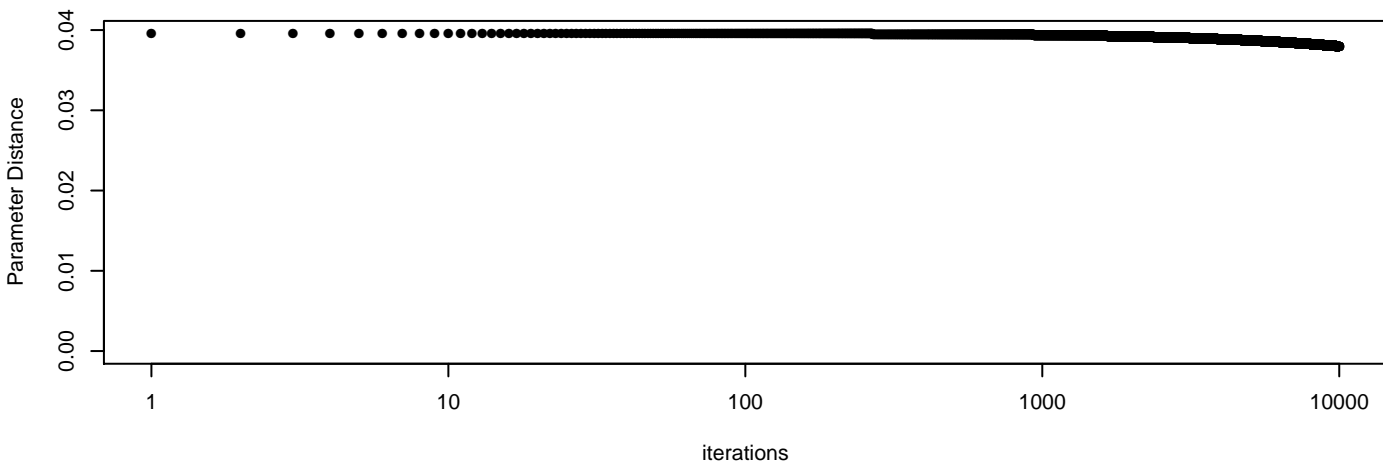
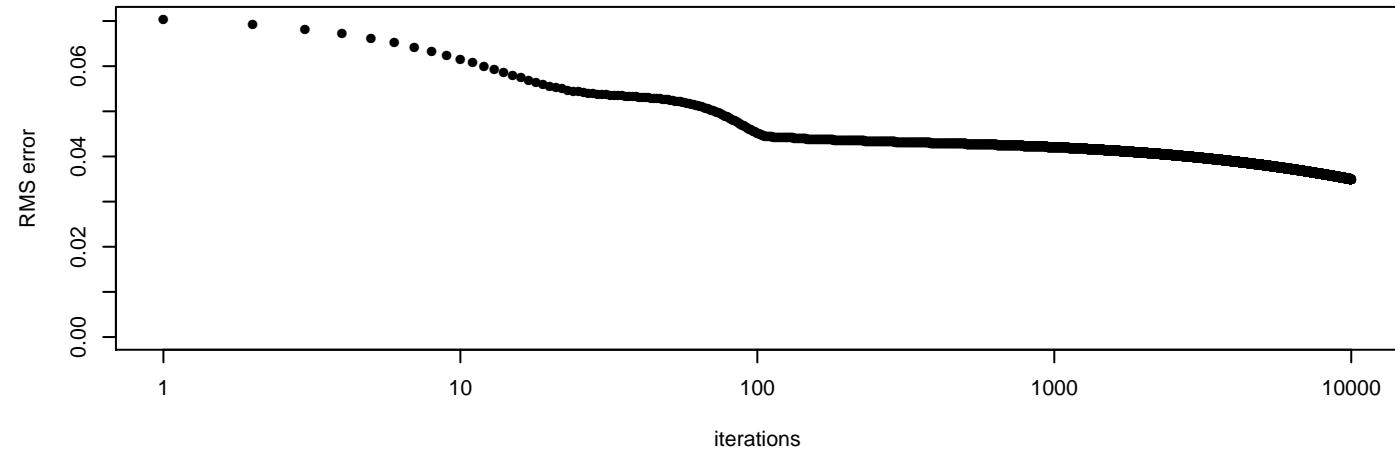


Negative Perturbation

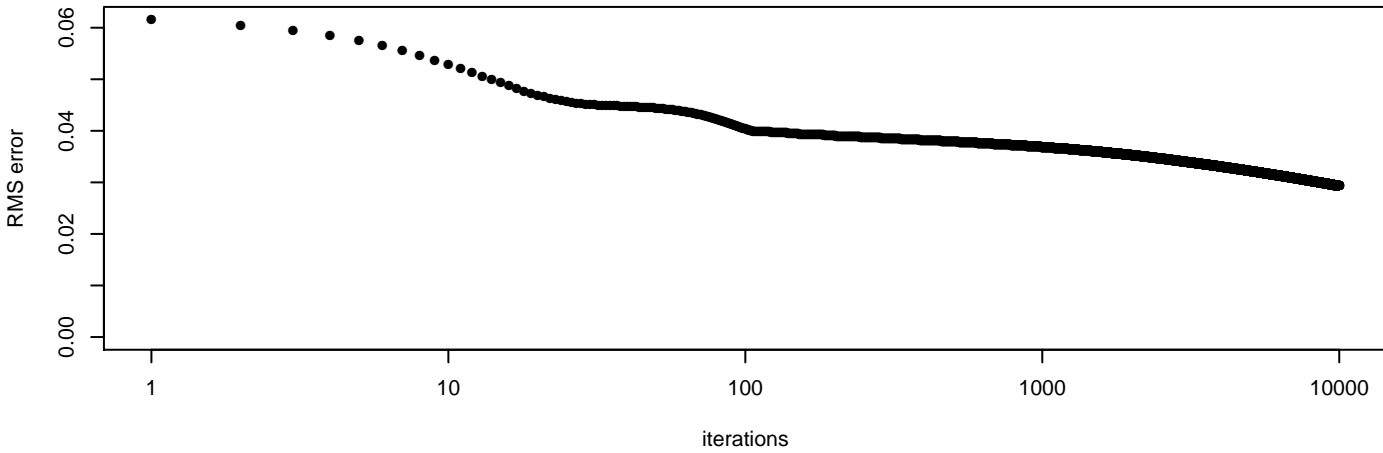


Parameter10

Positive Perturbation

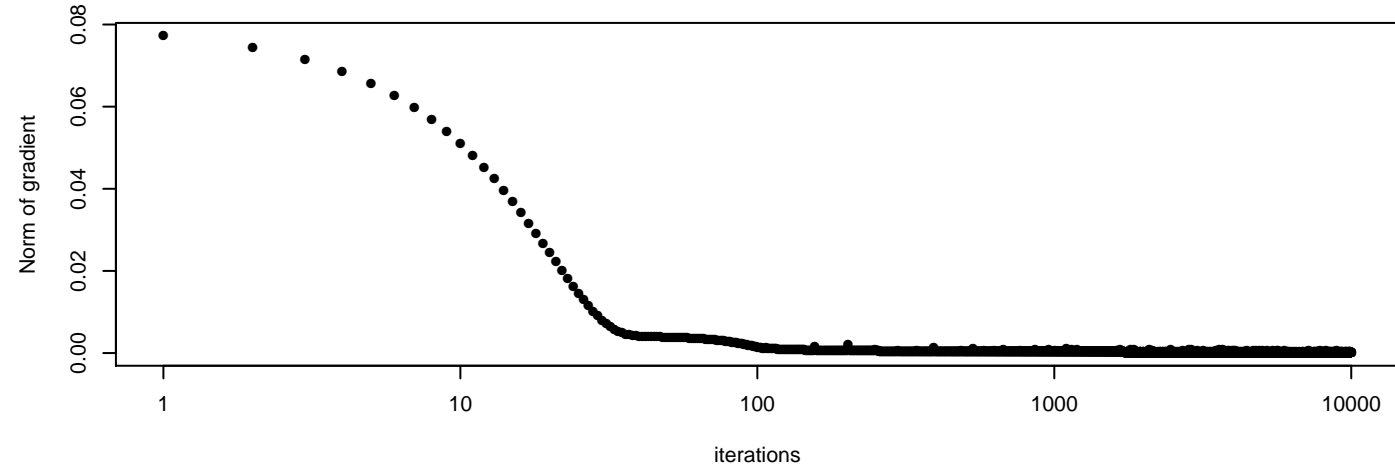
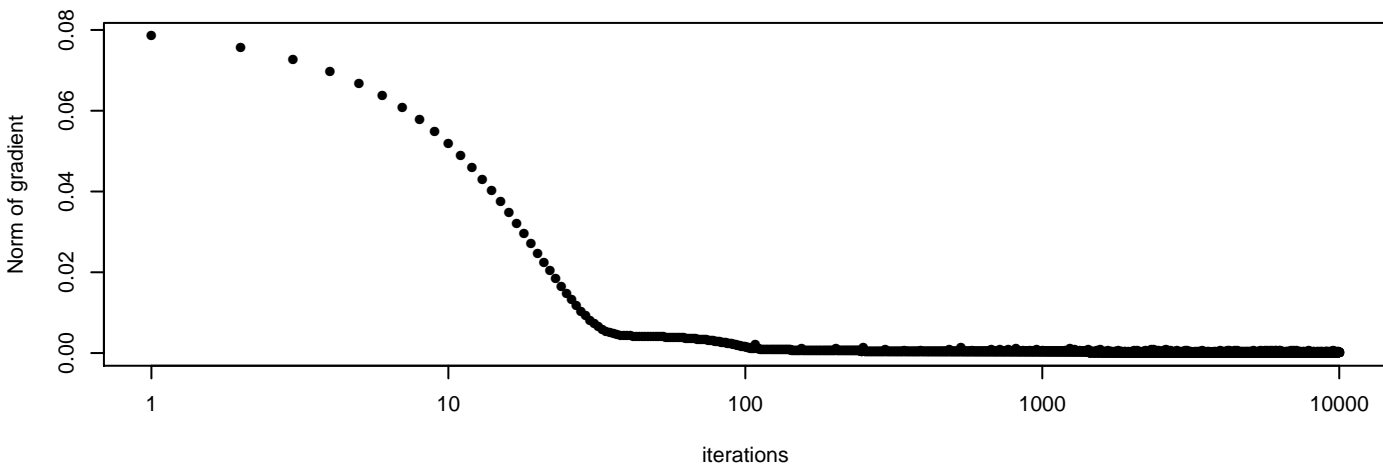
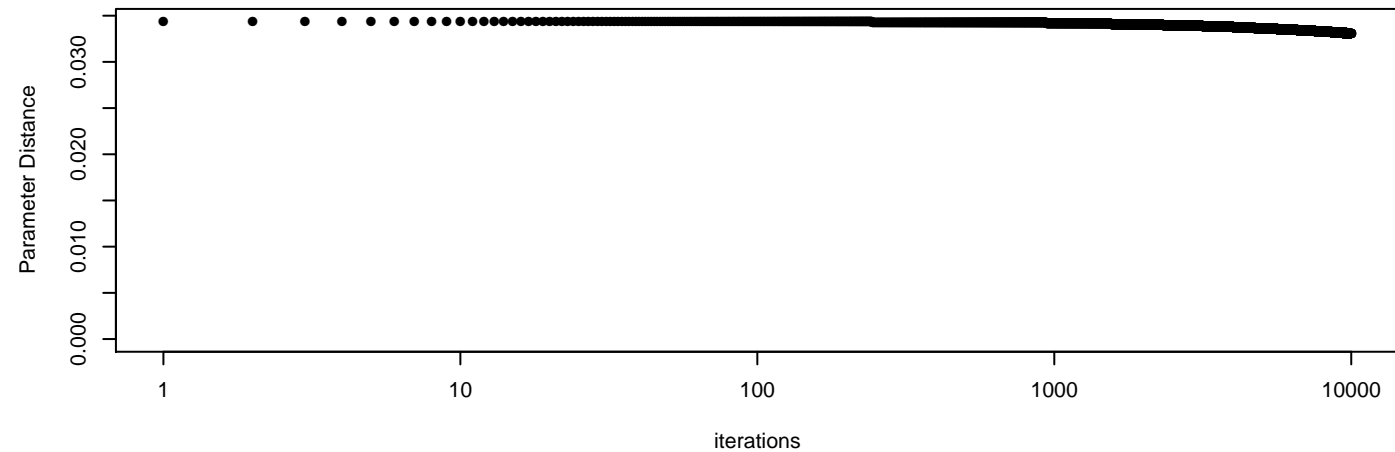
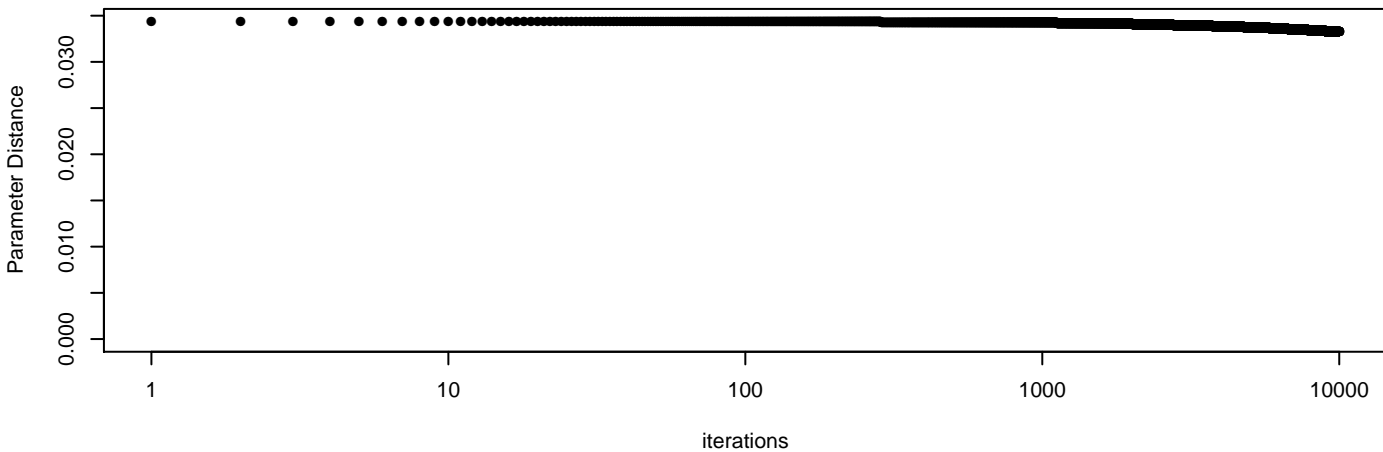
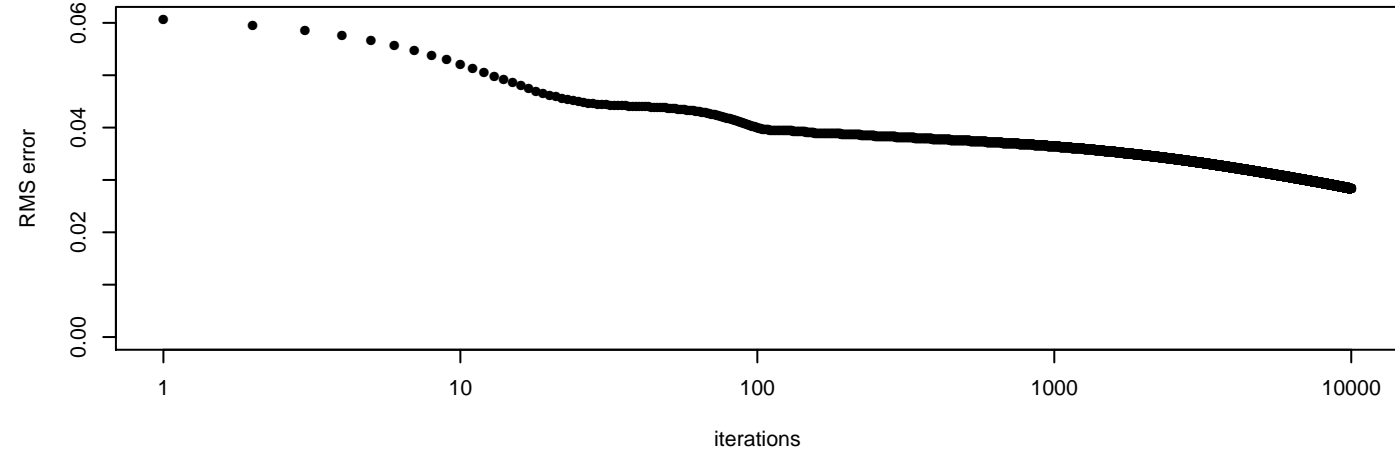


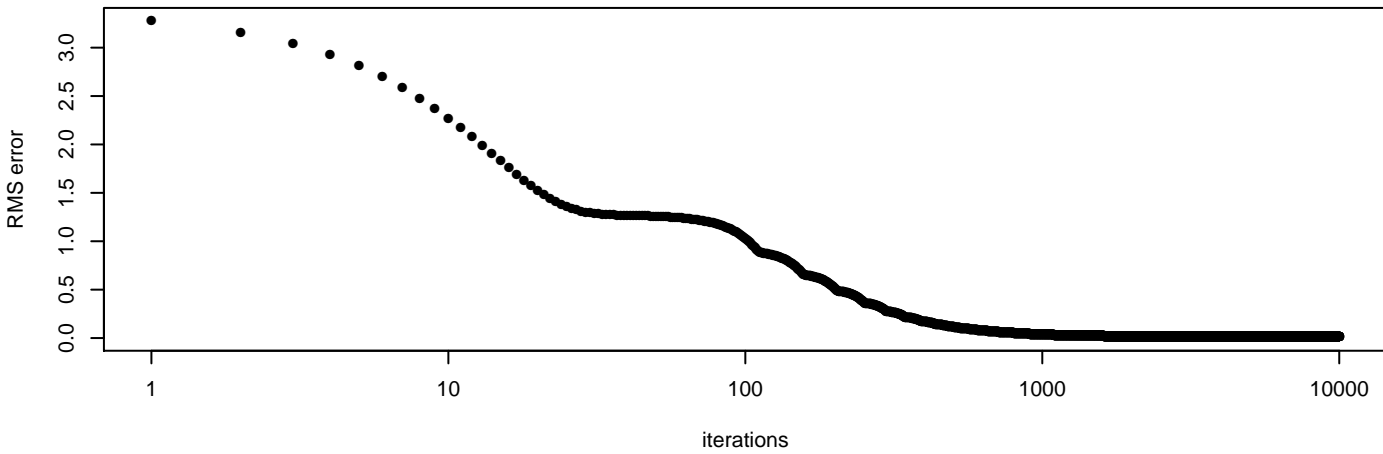
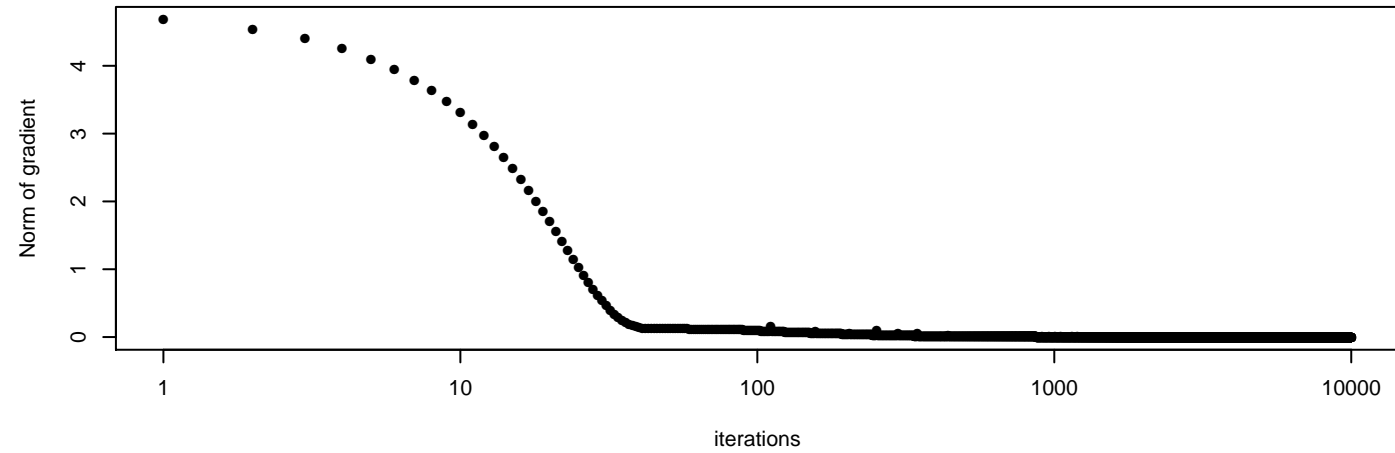
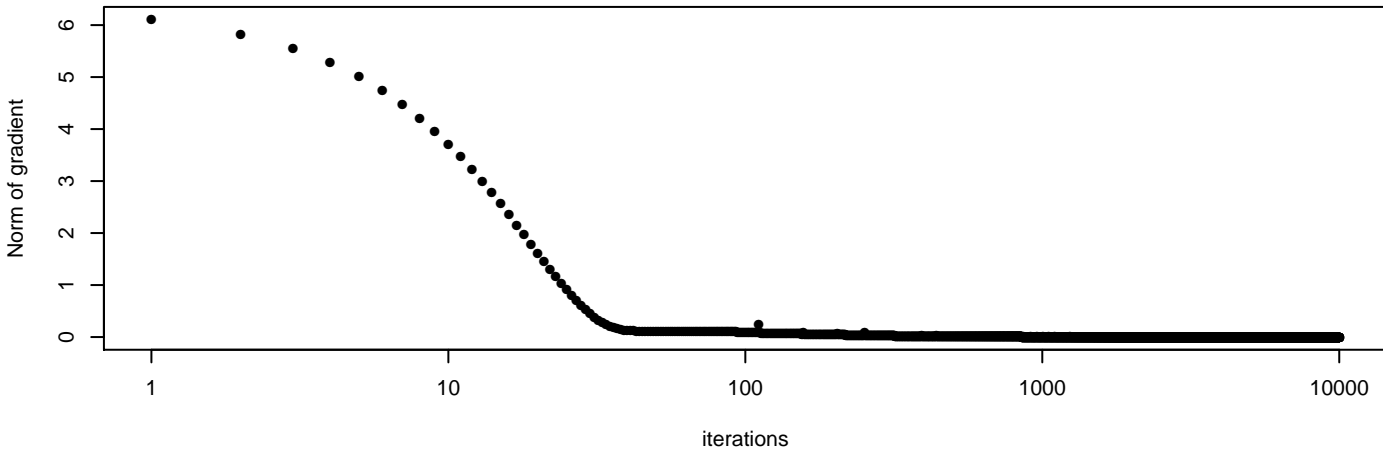
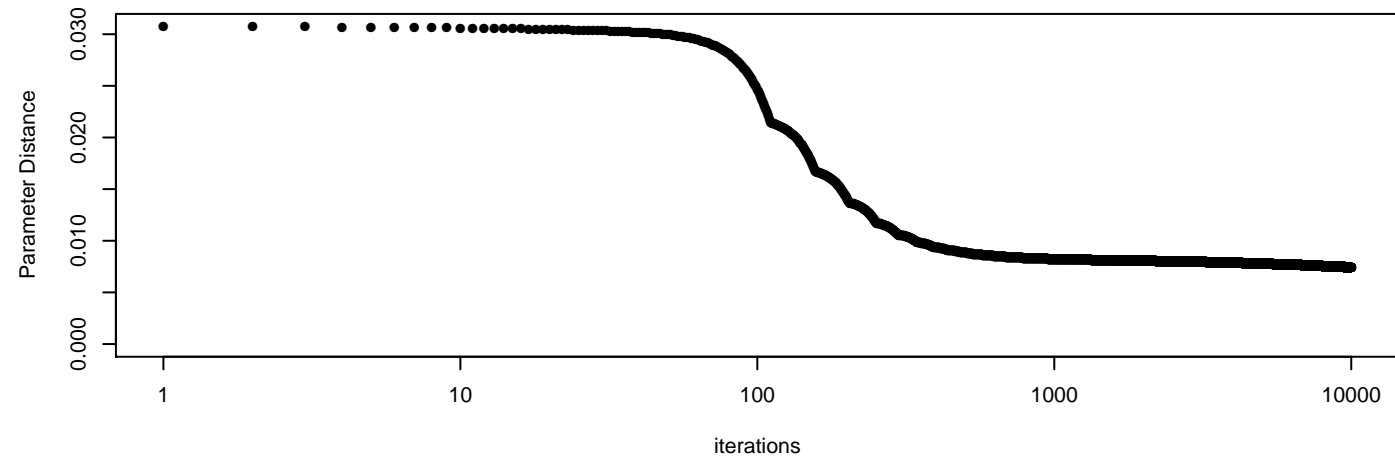
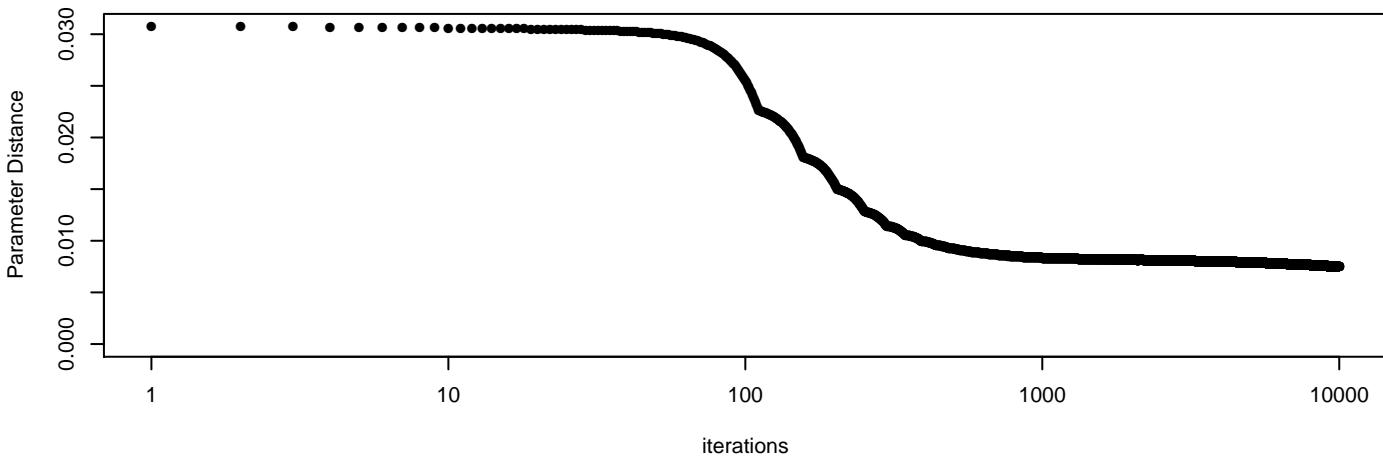
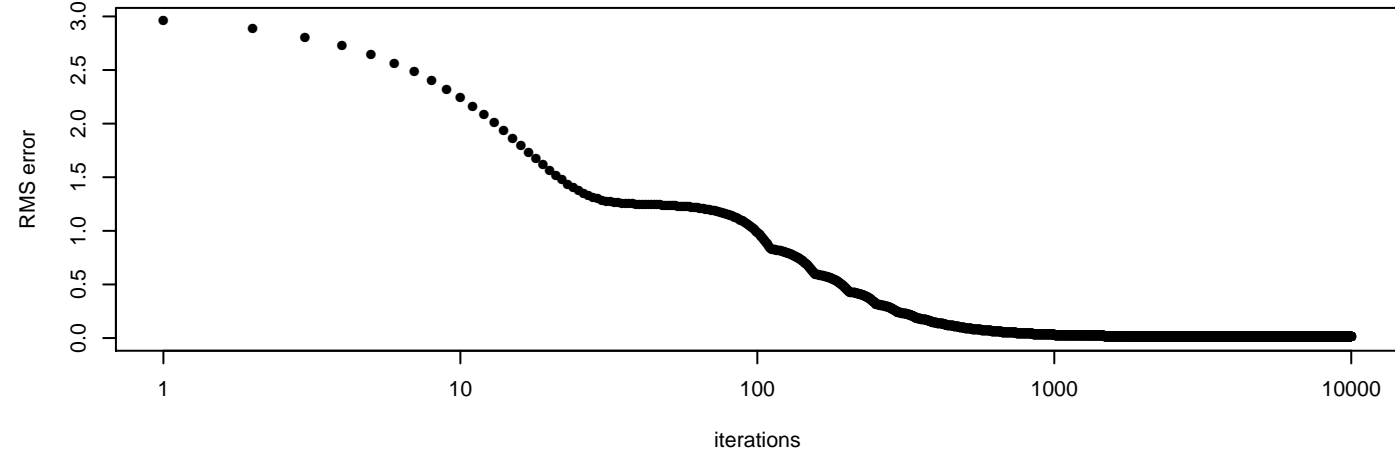
Negative Perturbation



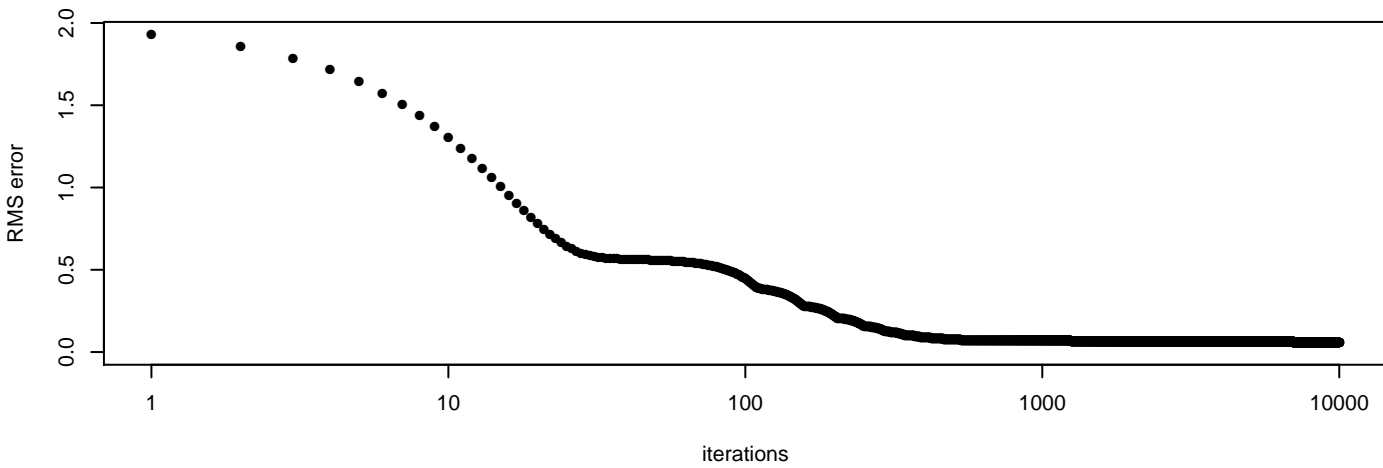
Parameter11

Positive Perturbation



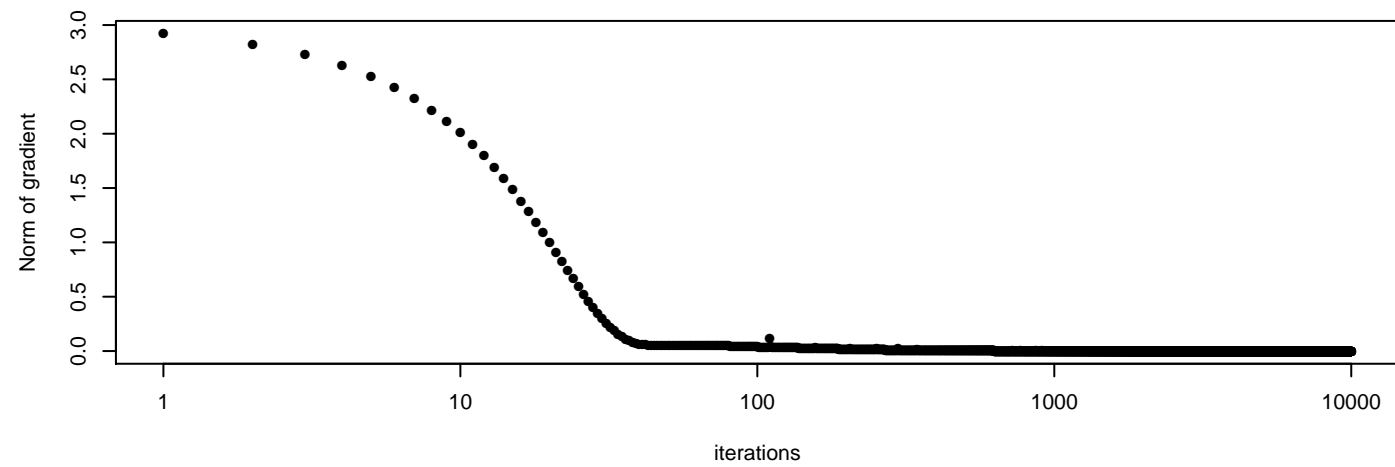
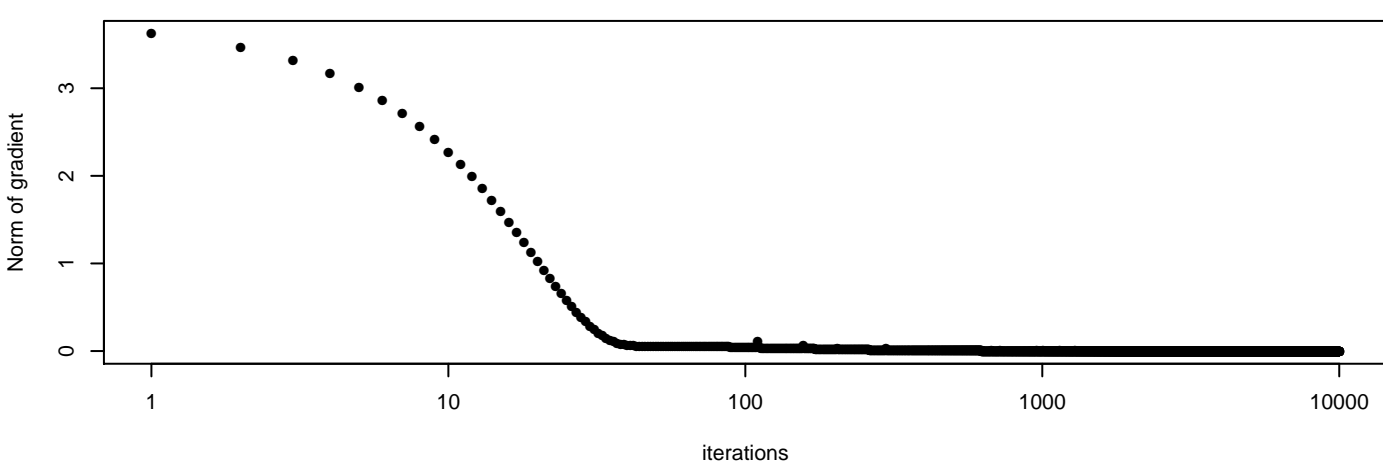
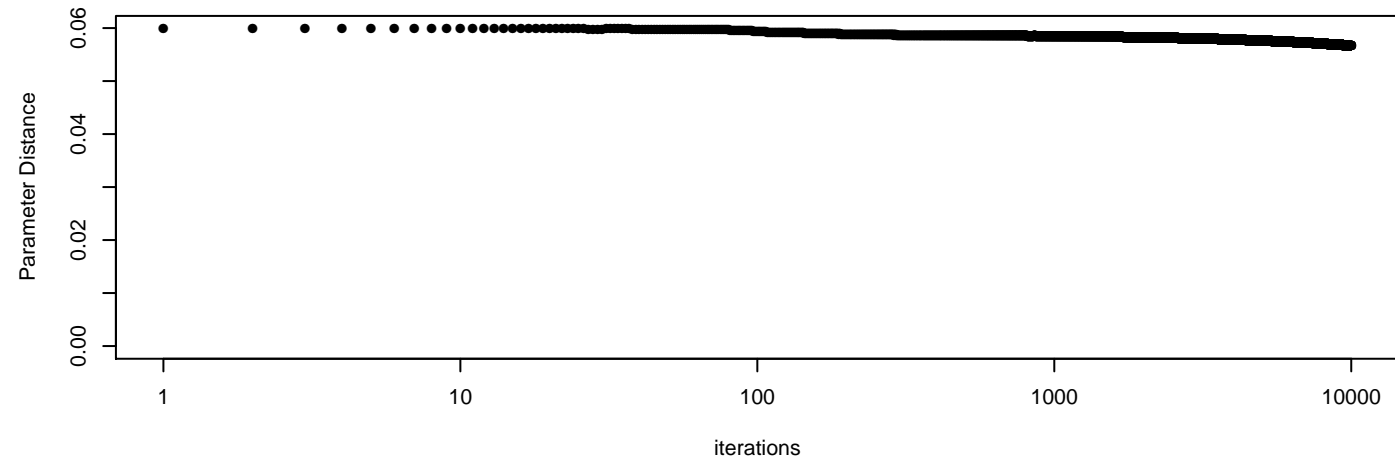
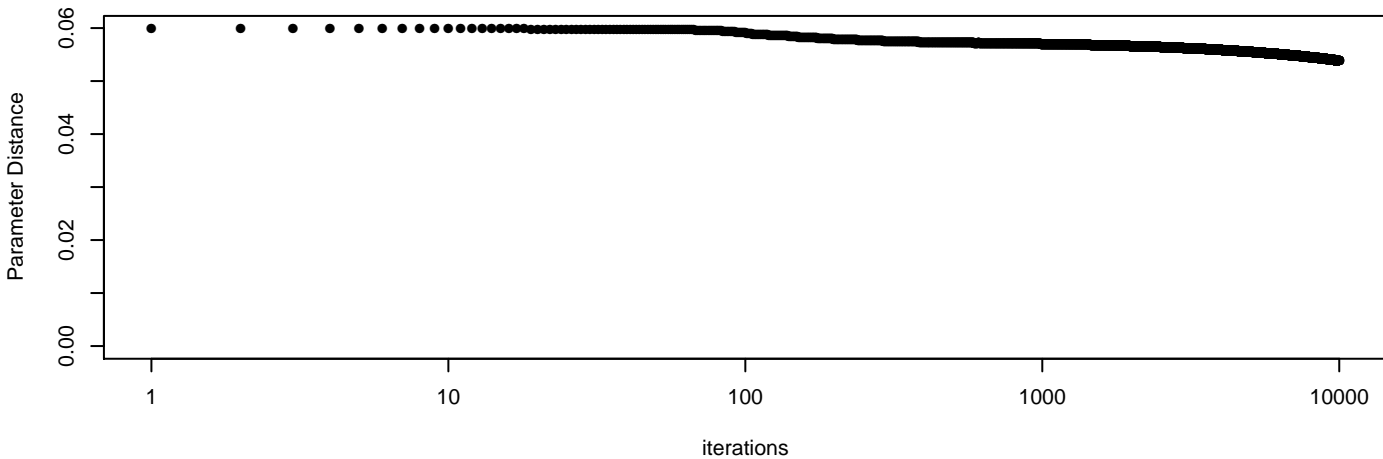
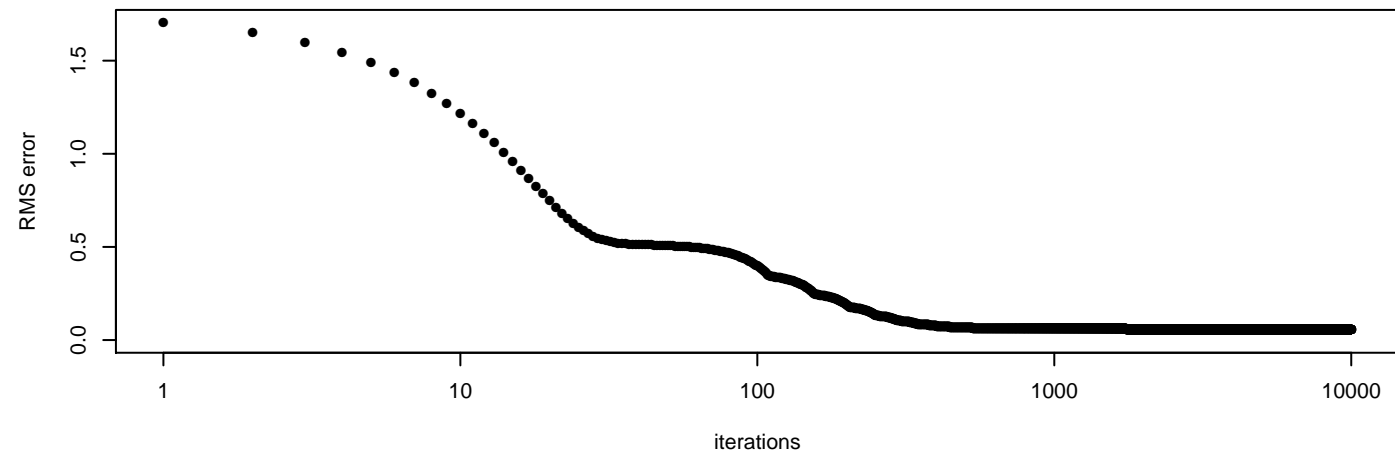
Negative Perturbation**Parameter12****Positive Perturbation**

Negative Perturbation

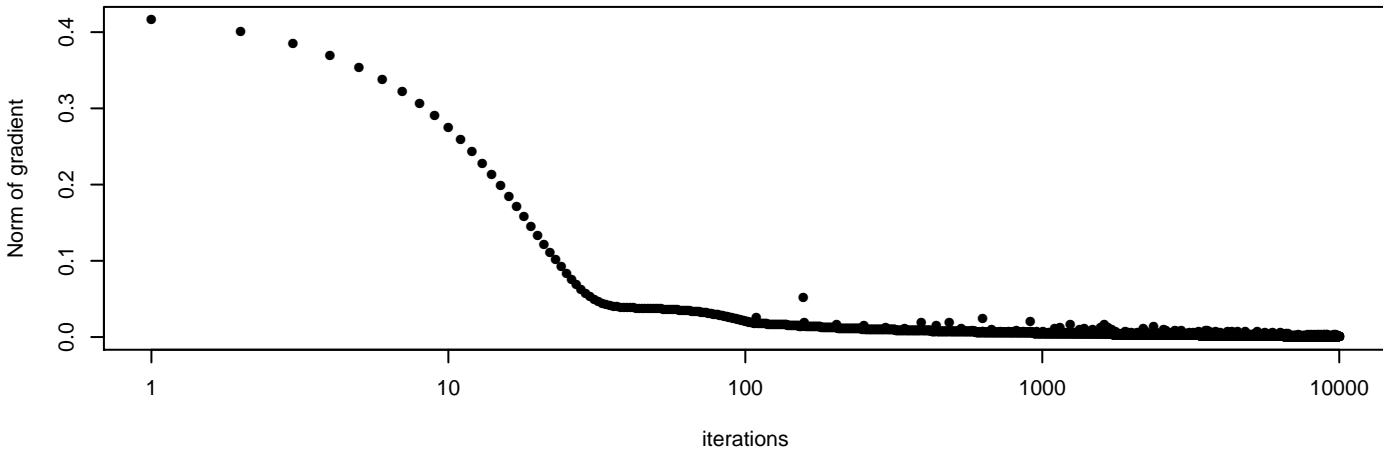
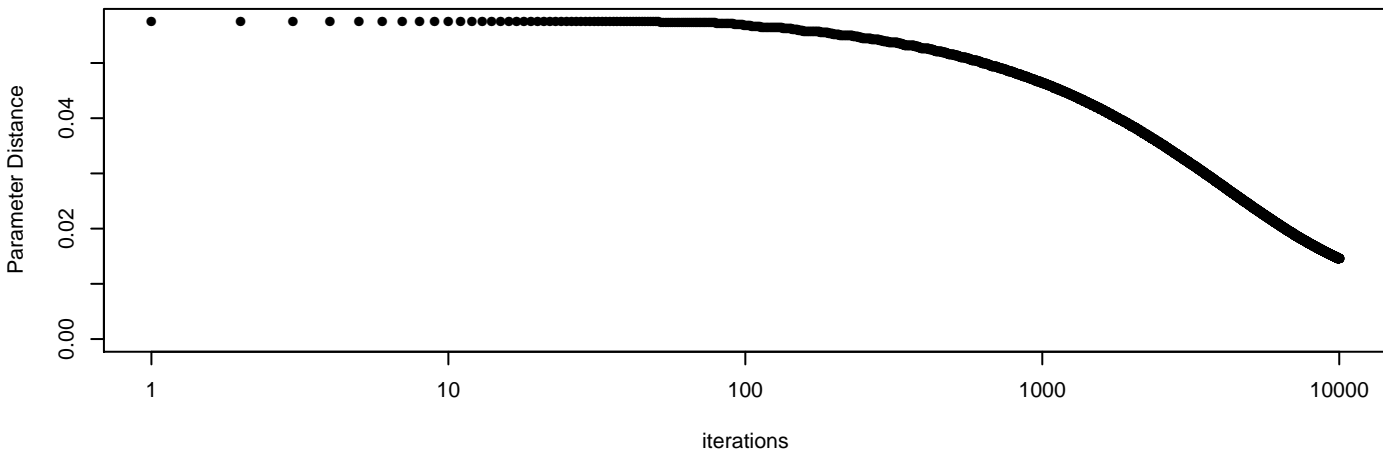
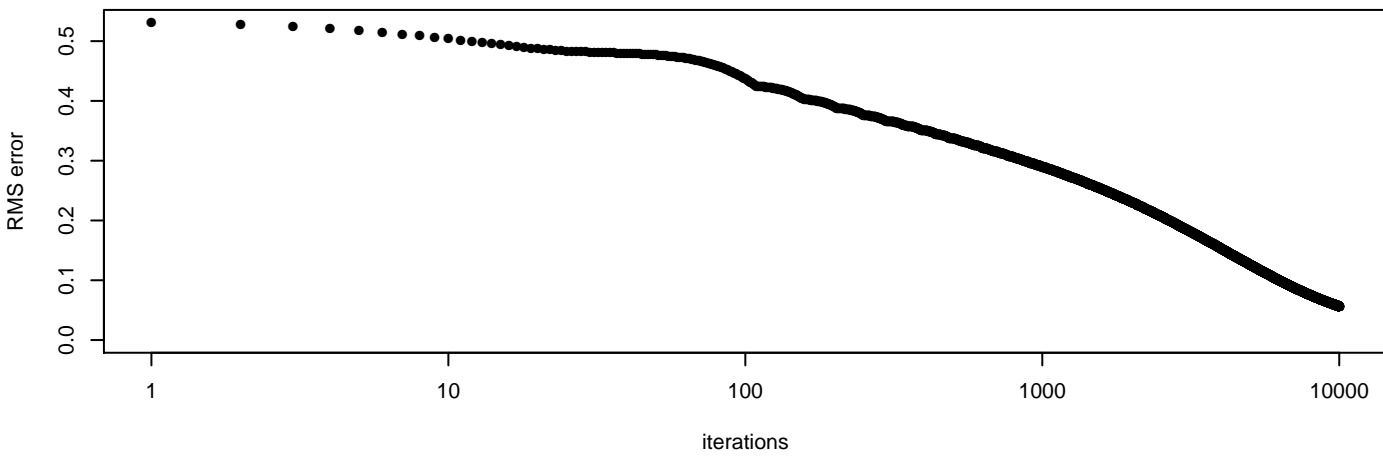


Parameter13

Positive Perturbation

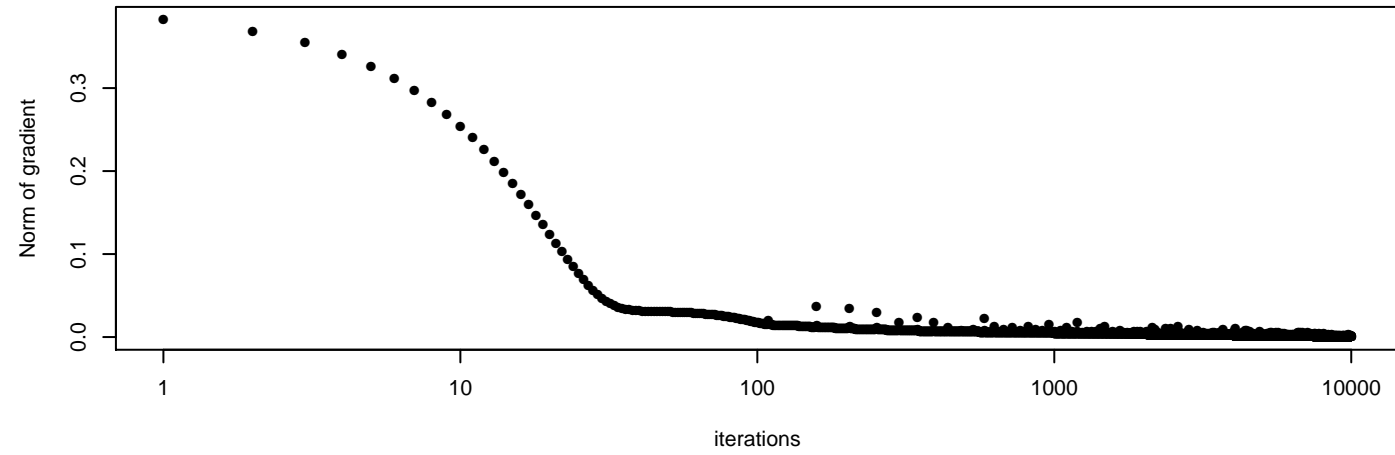
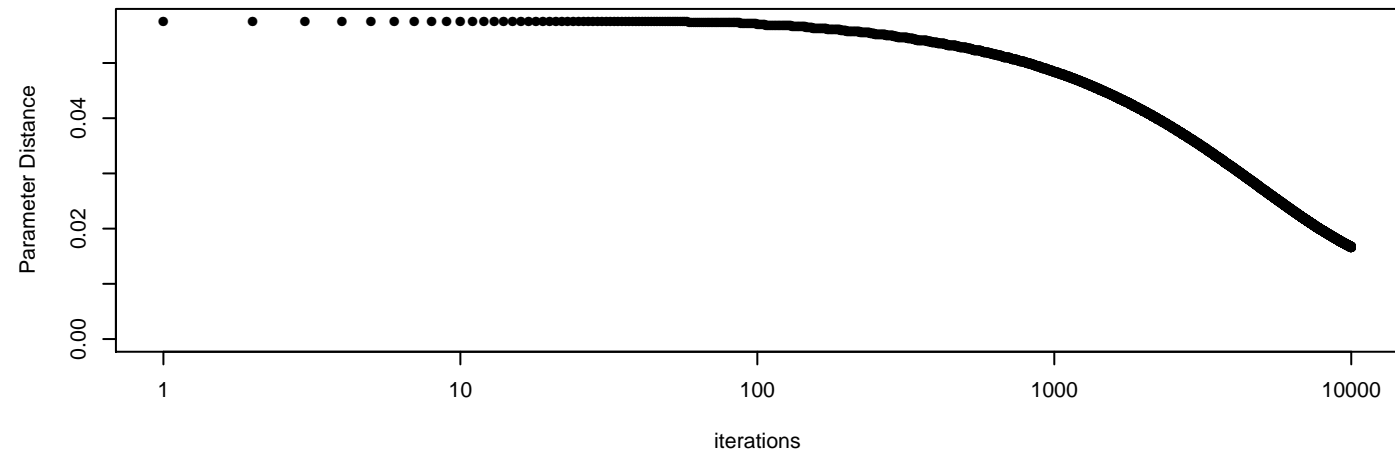
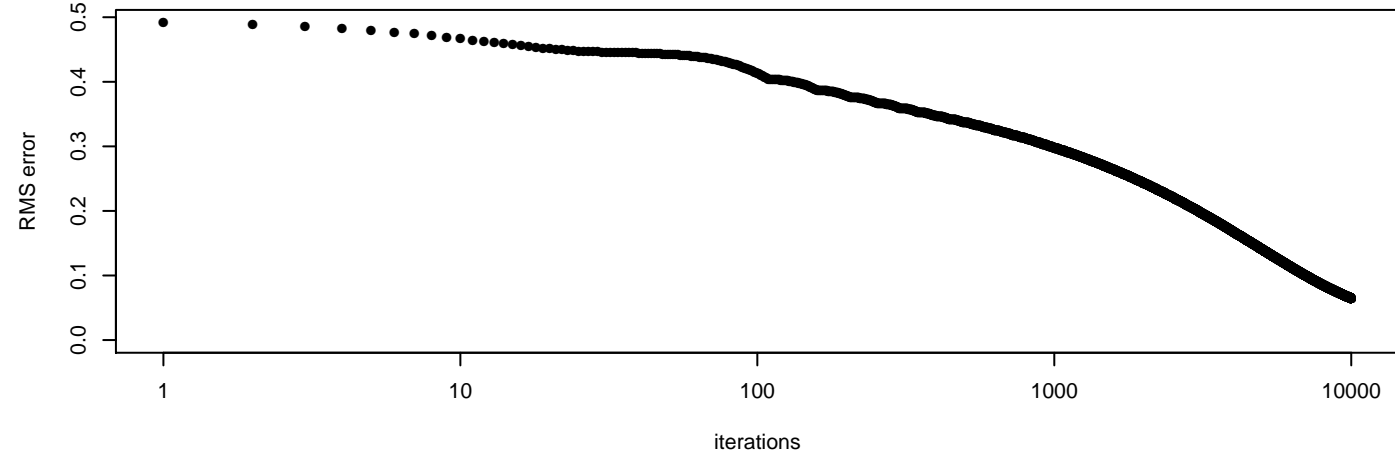


Negative Perturbation

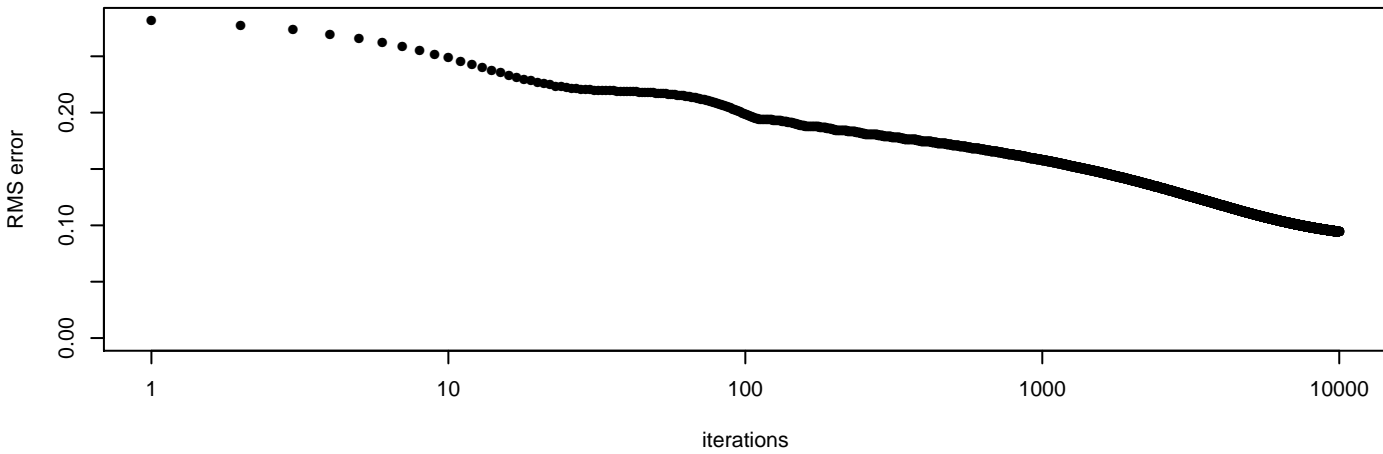


Parameter14

Positive Perturbation

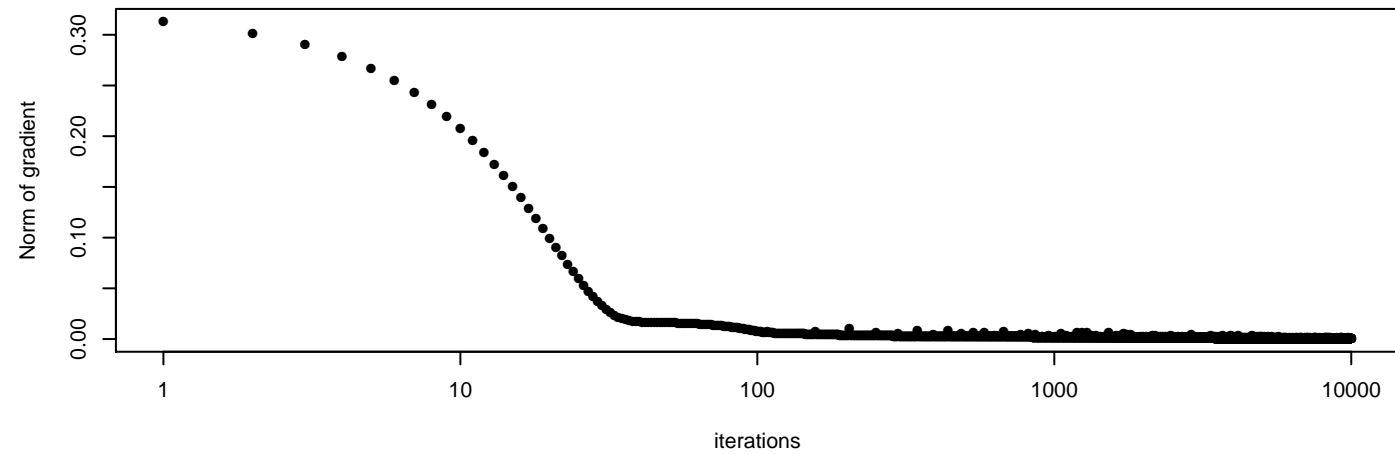
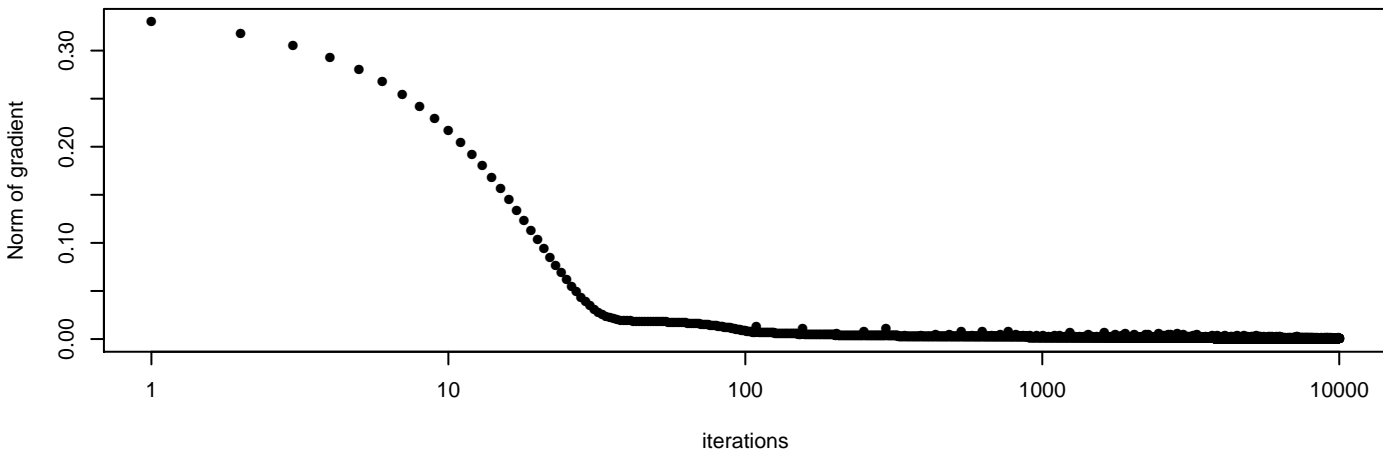
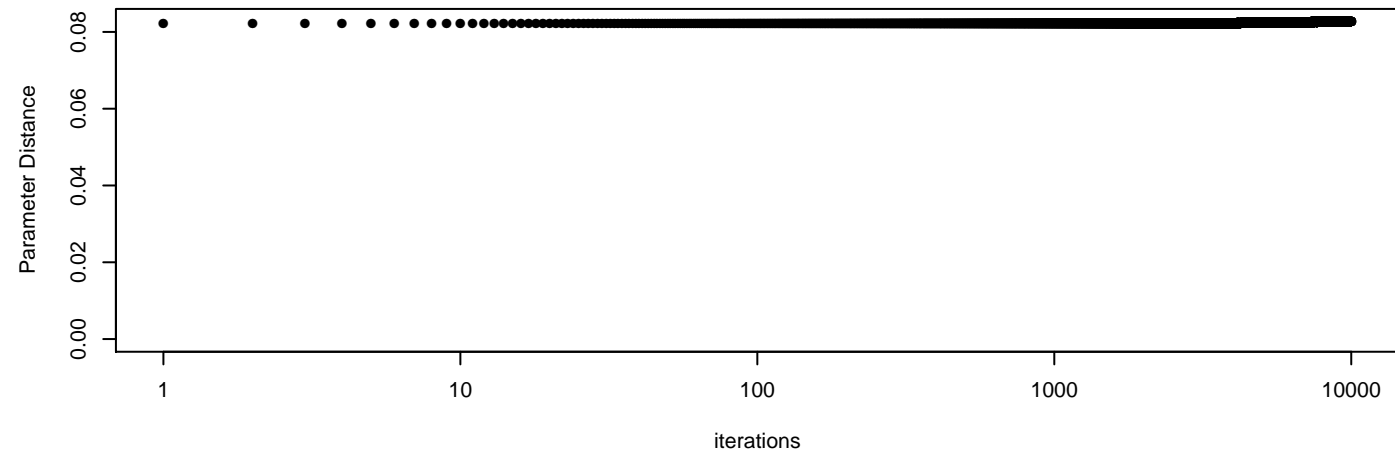
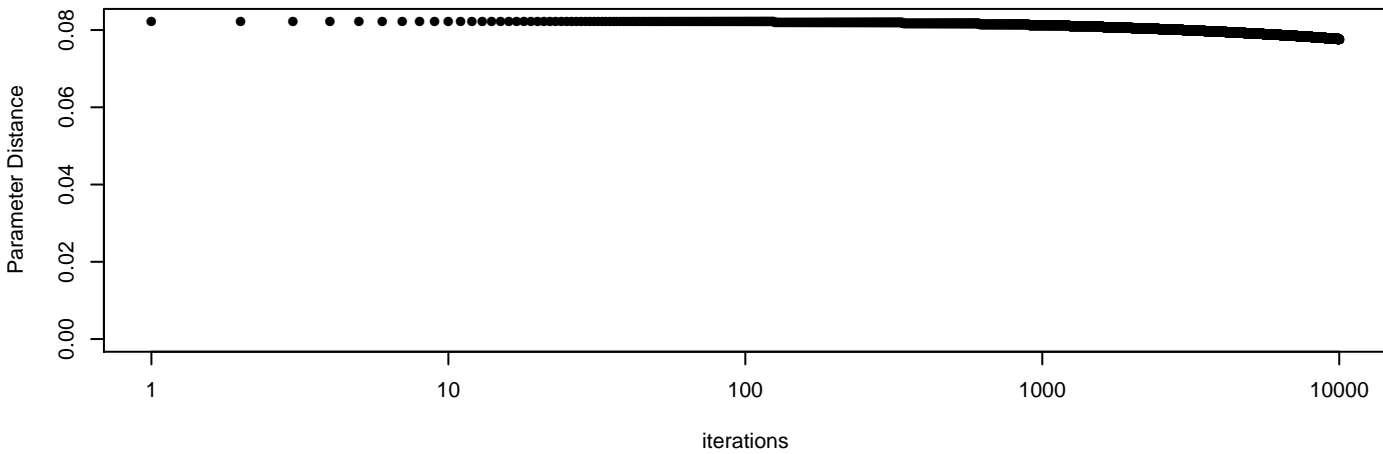
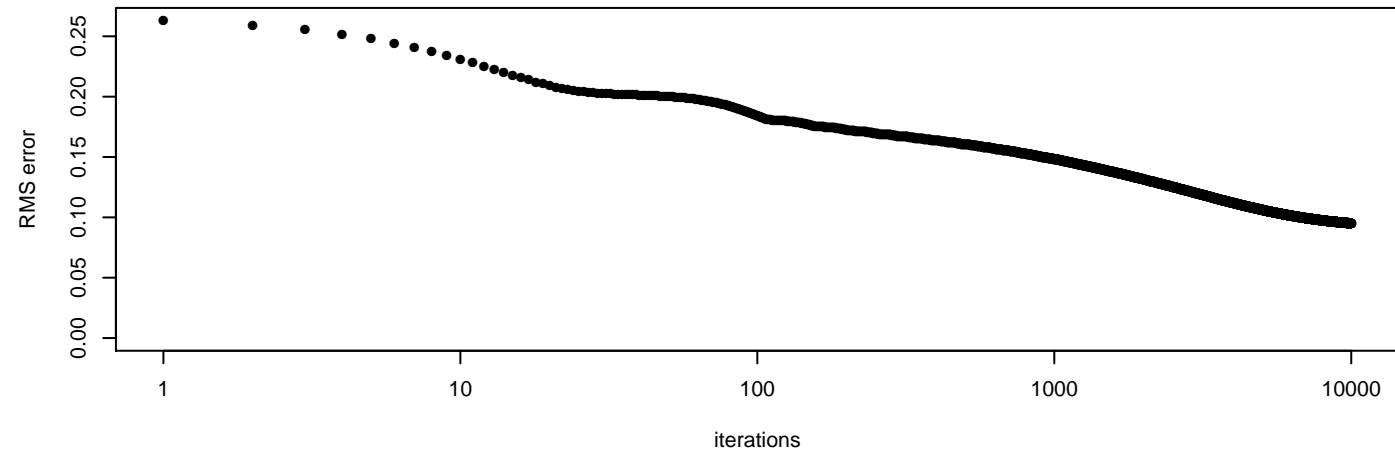


Negative Perturbation

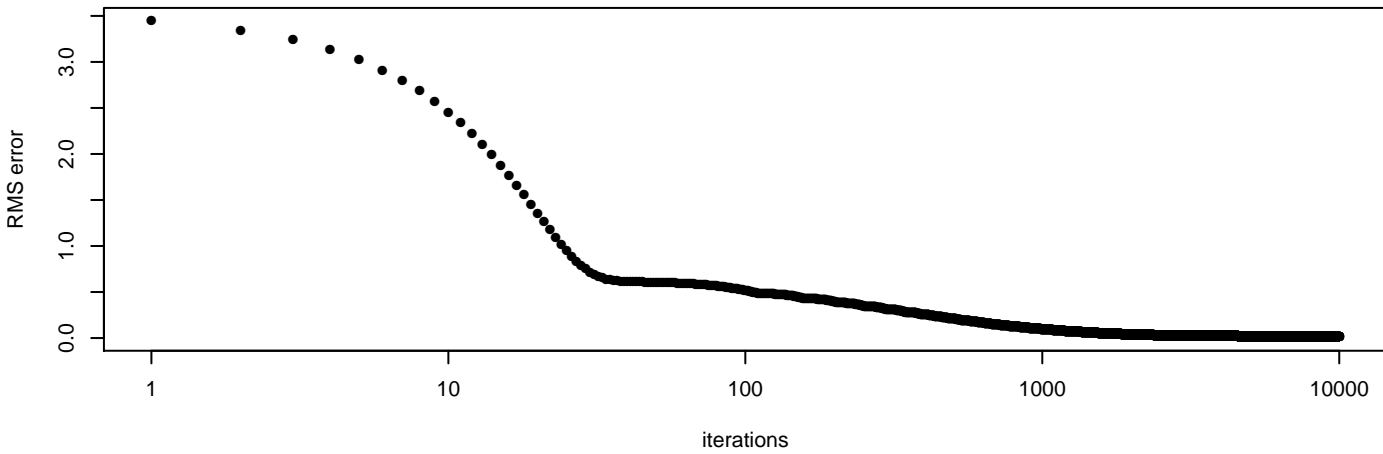


Parameter15

Positive Perturbation

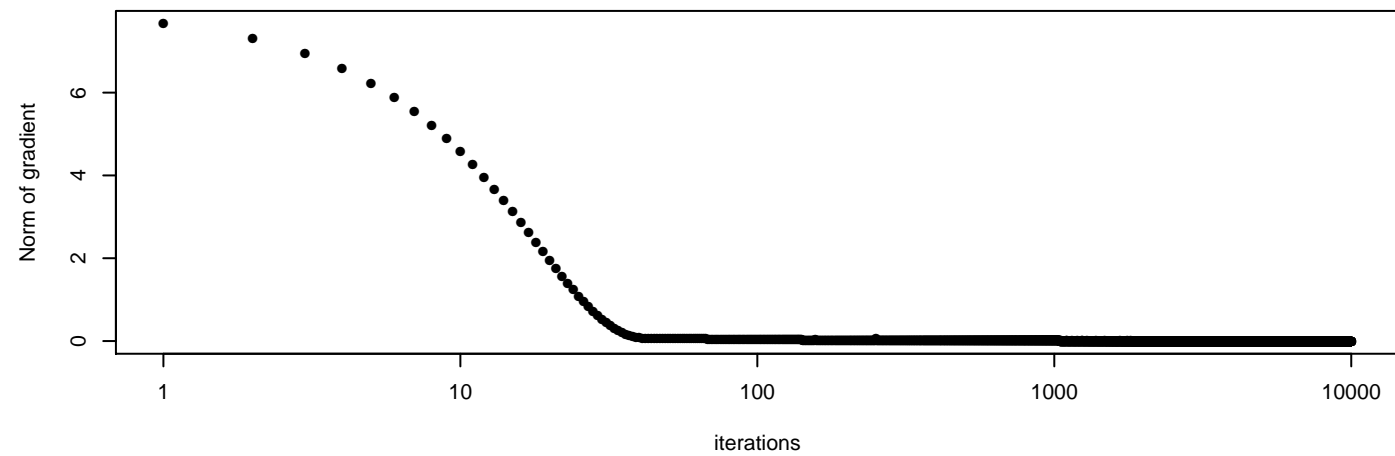
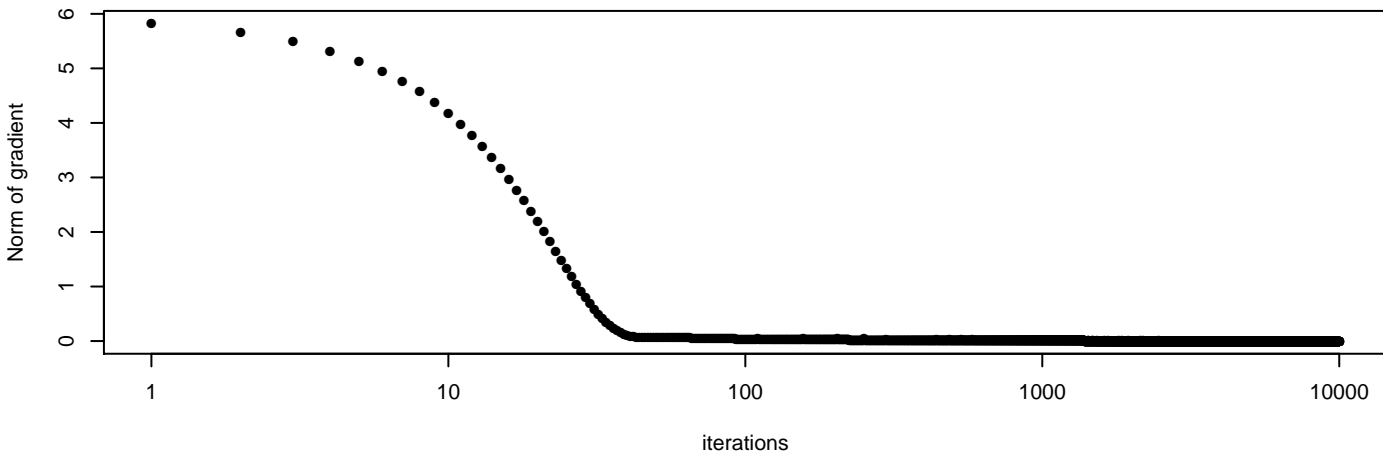
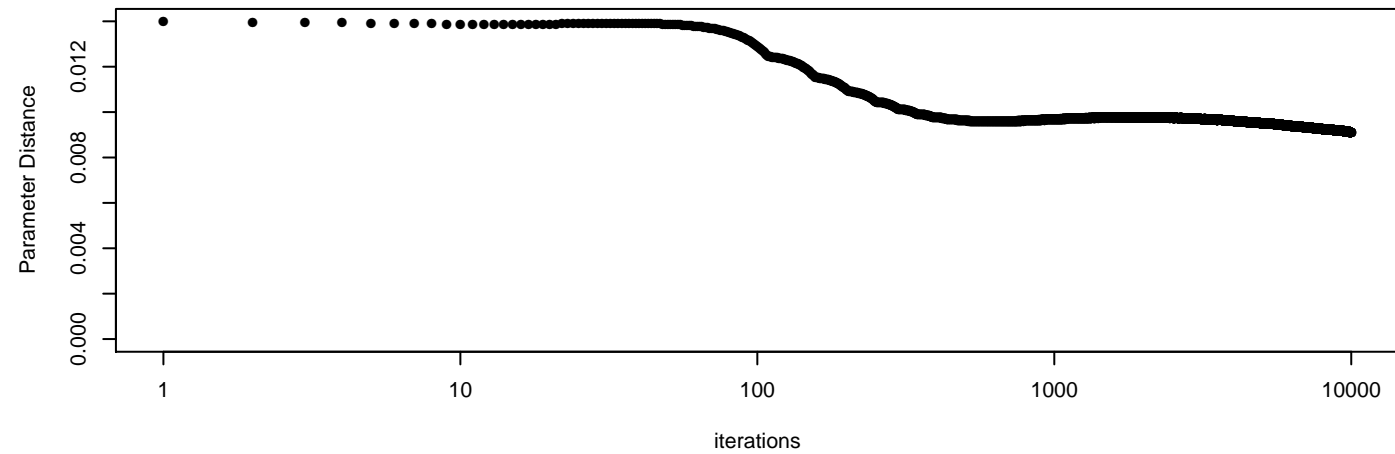
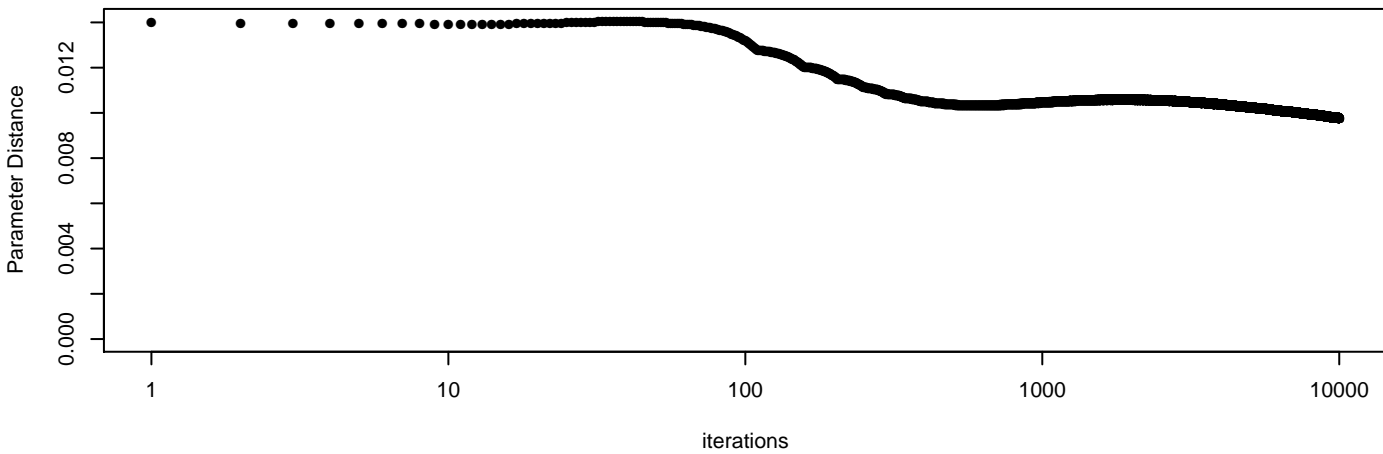
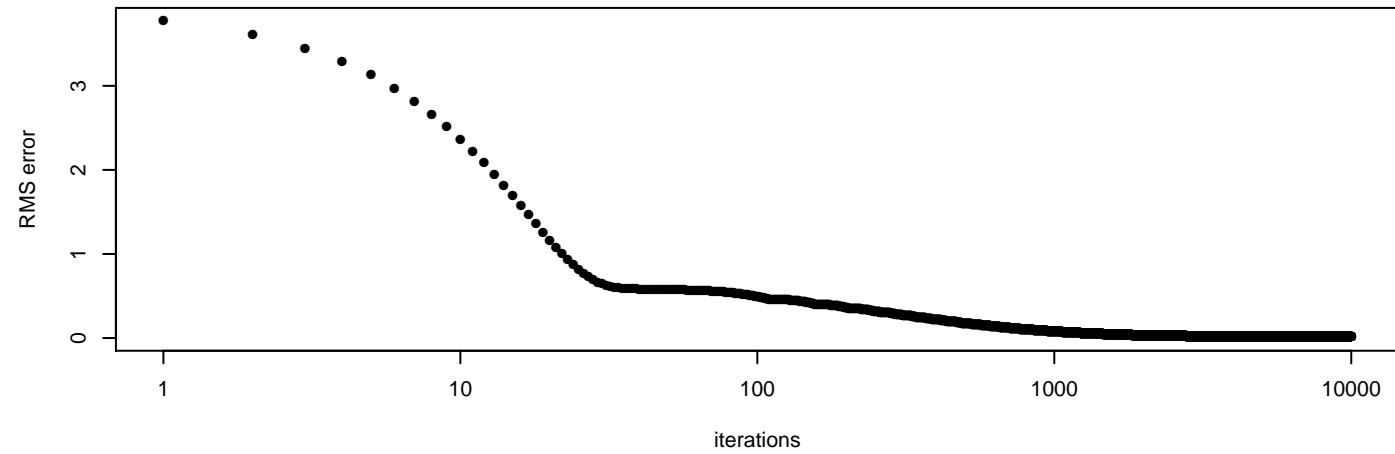


Negative Perturbation

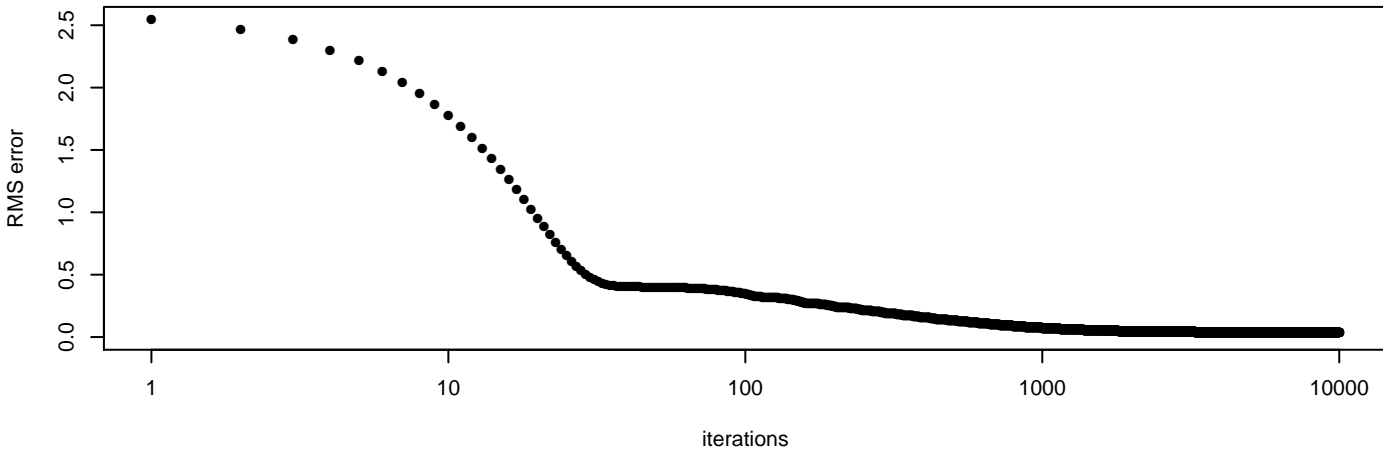


Parameter2

Positive Perturbation

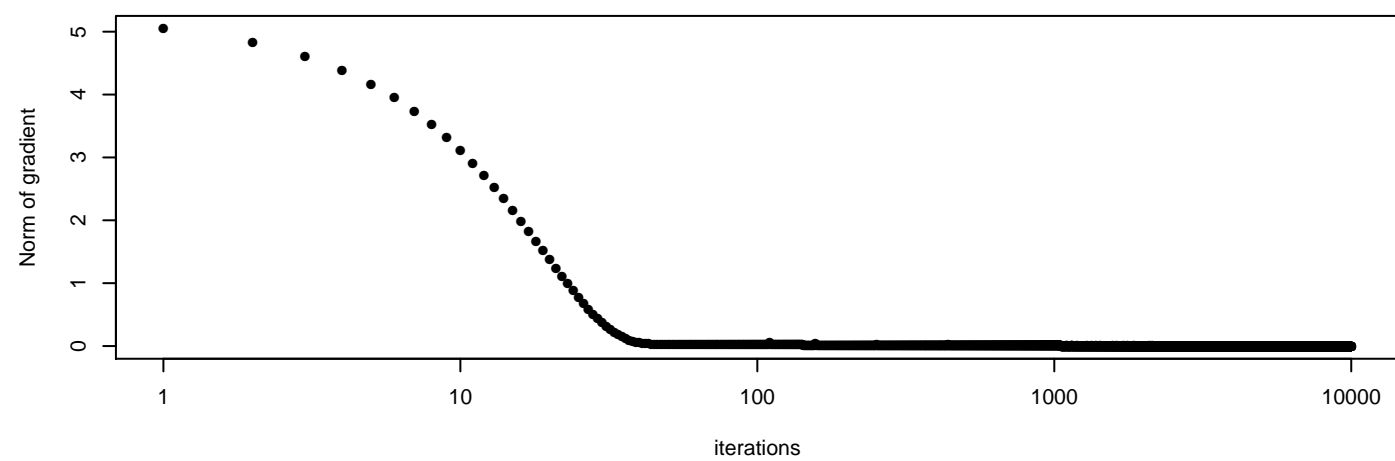
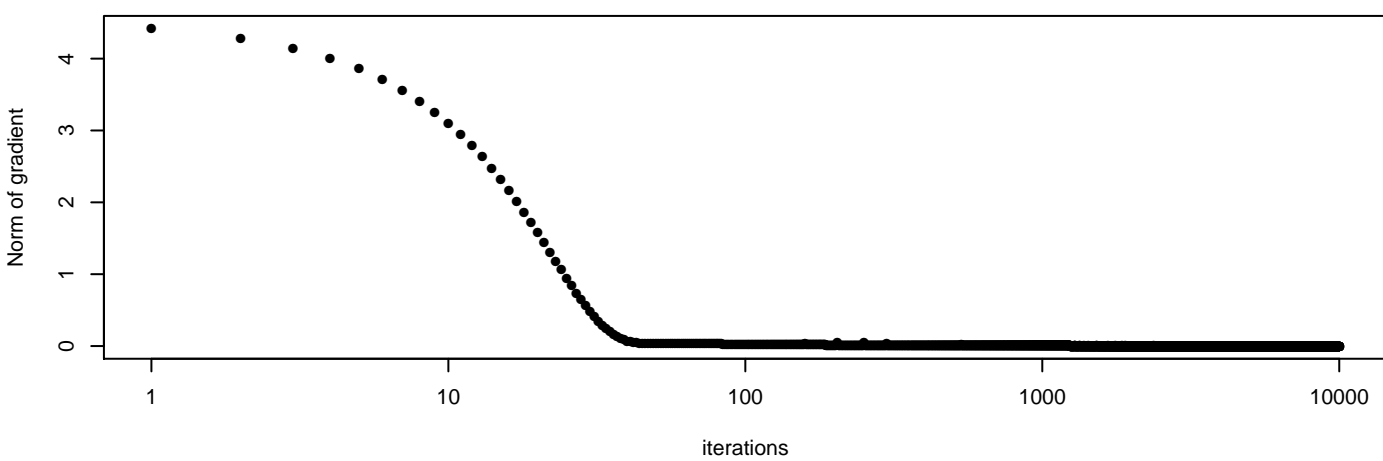
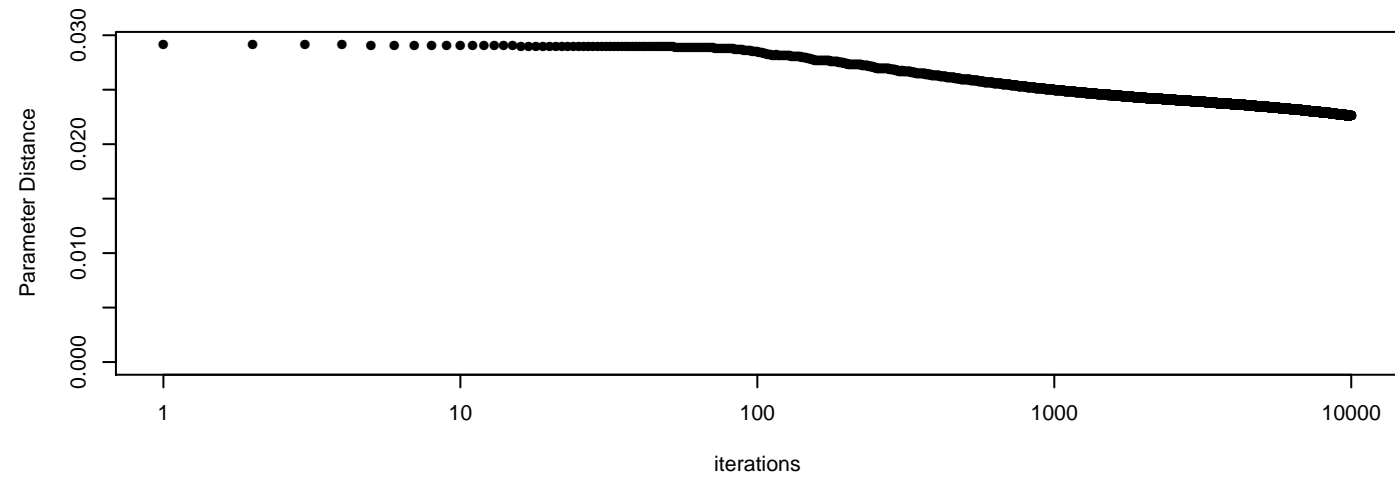
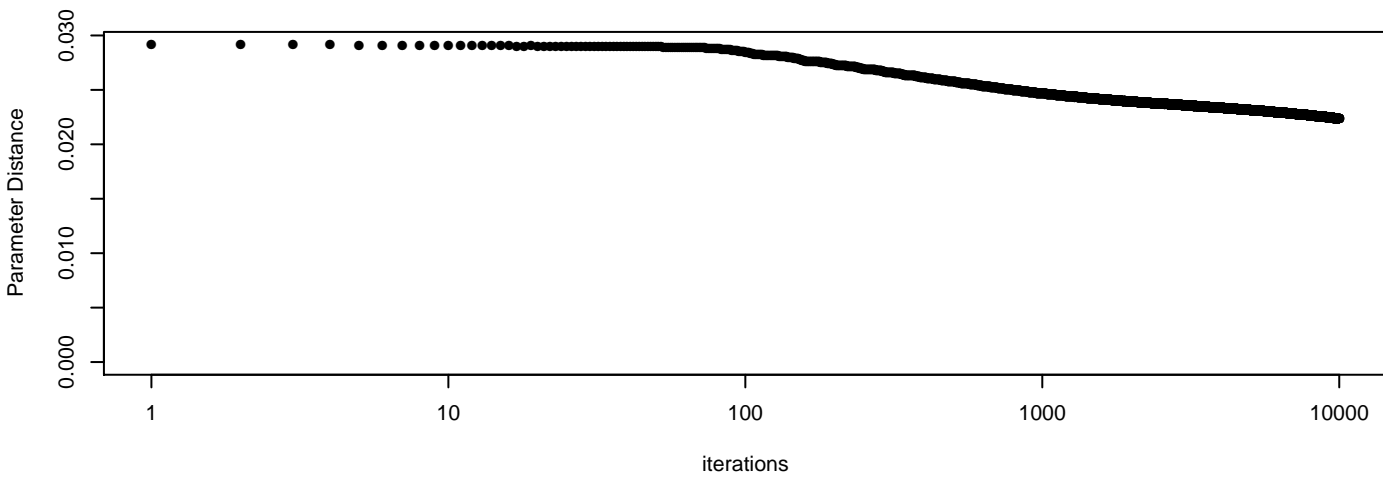
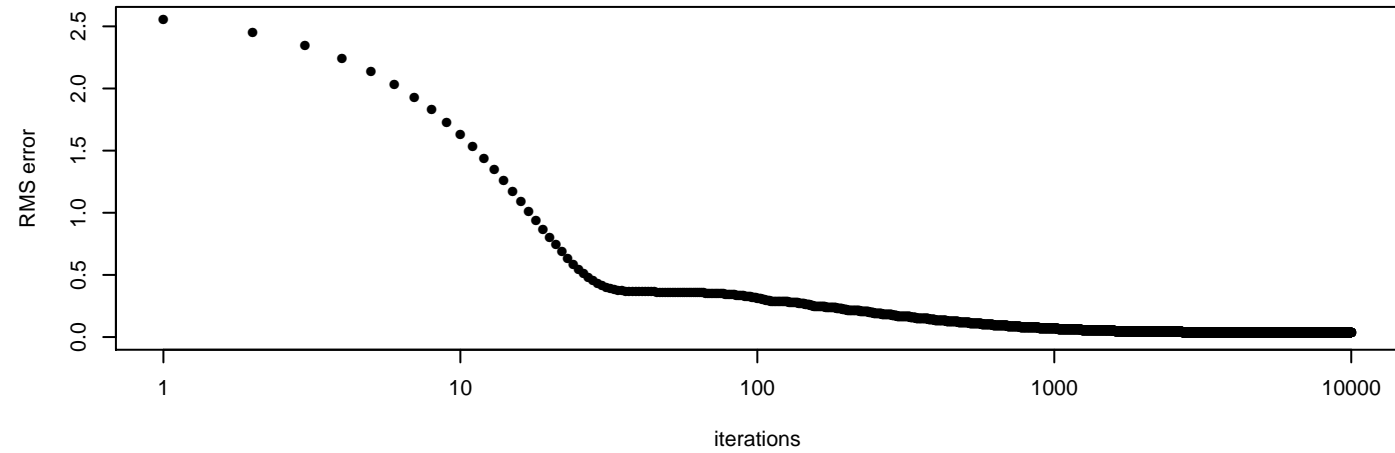


Negative Perturbation

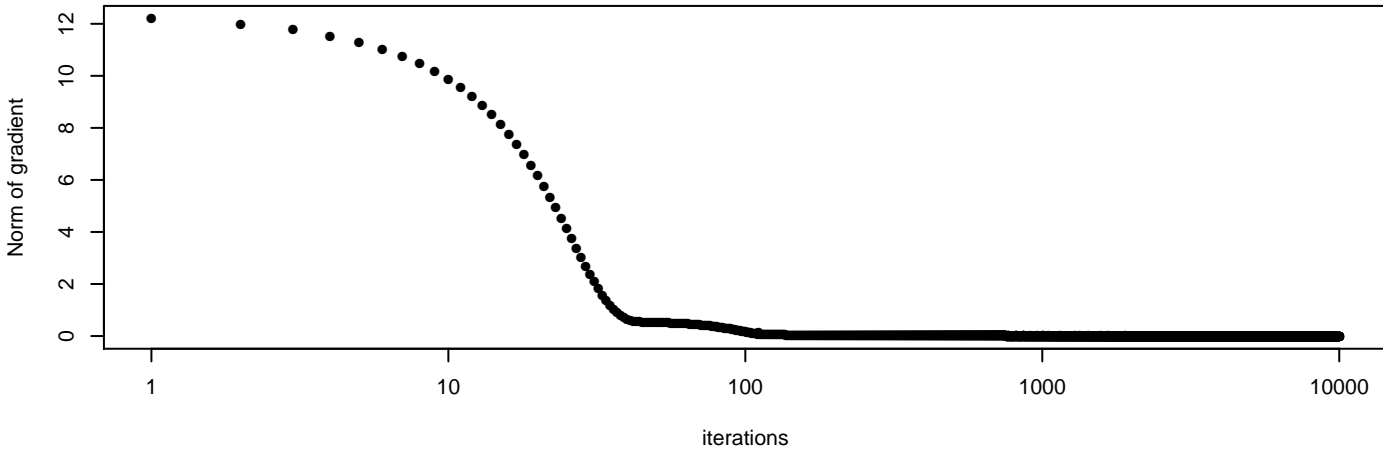
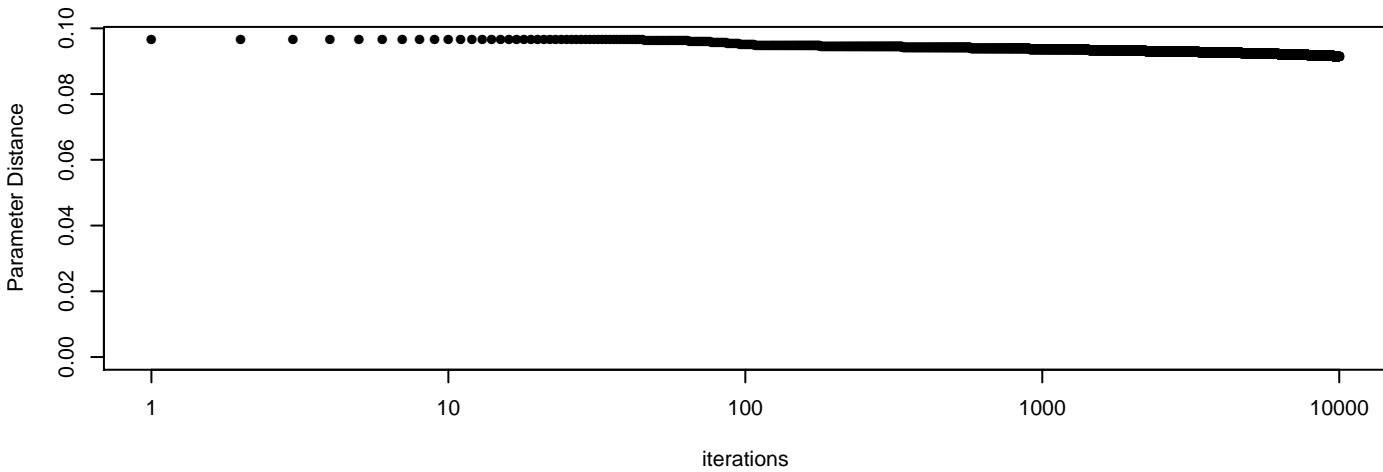
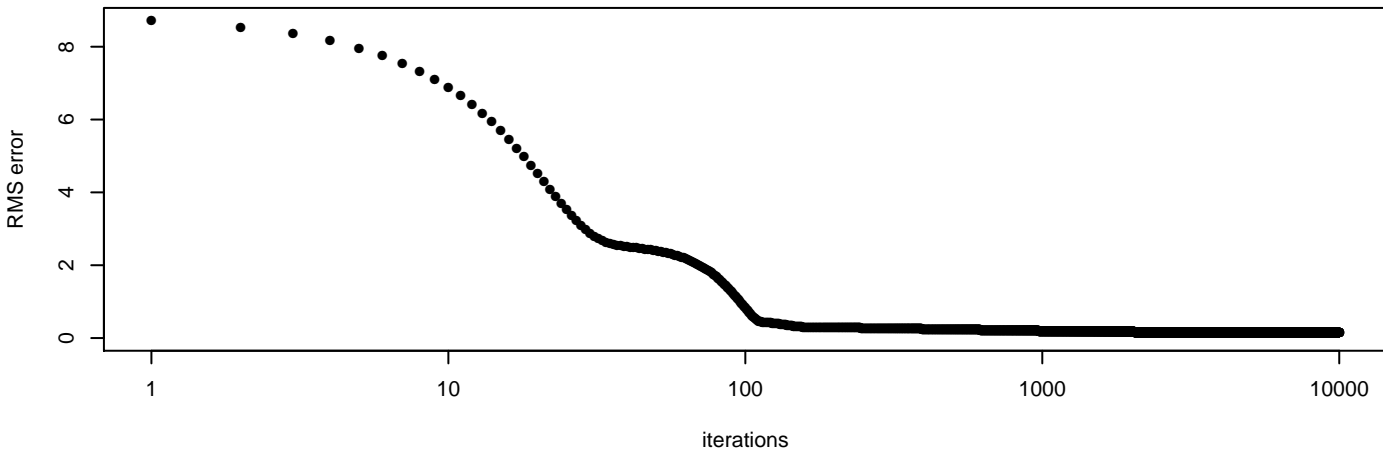


Parameter3

Positive Perturbation

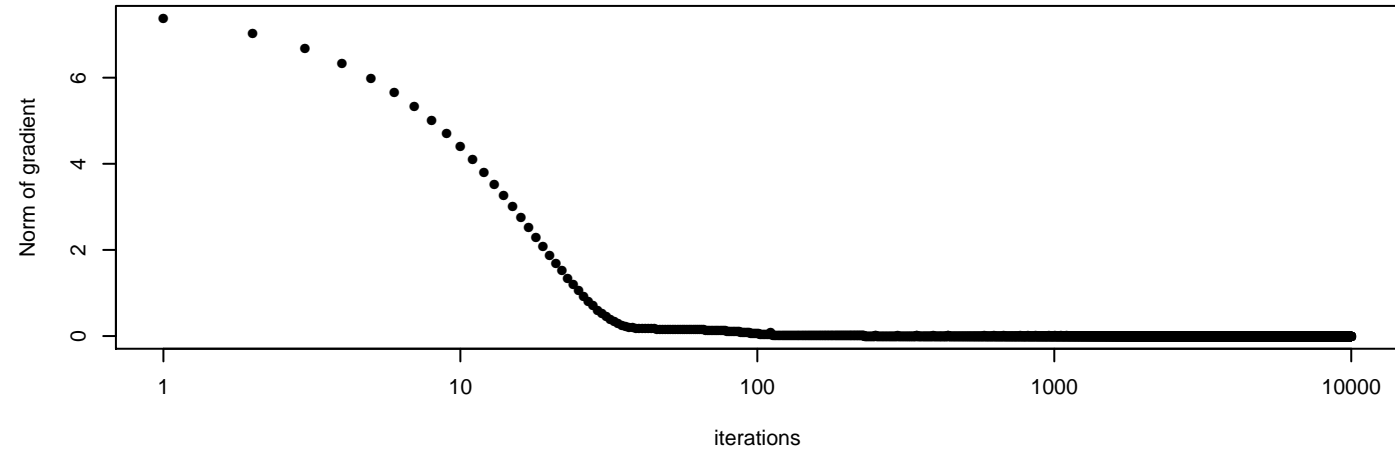
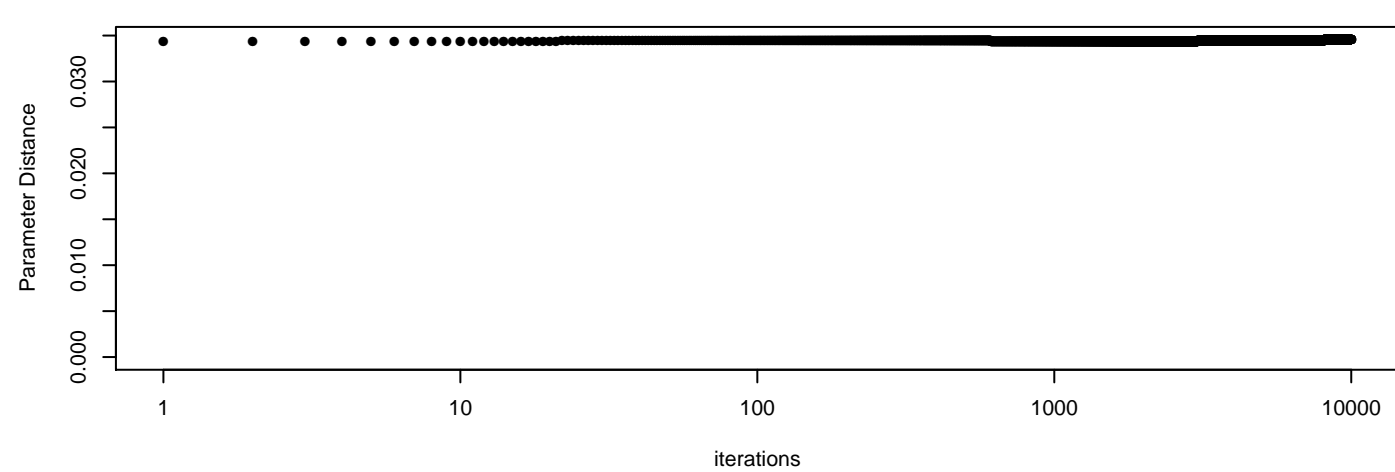
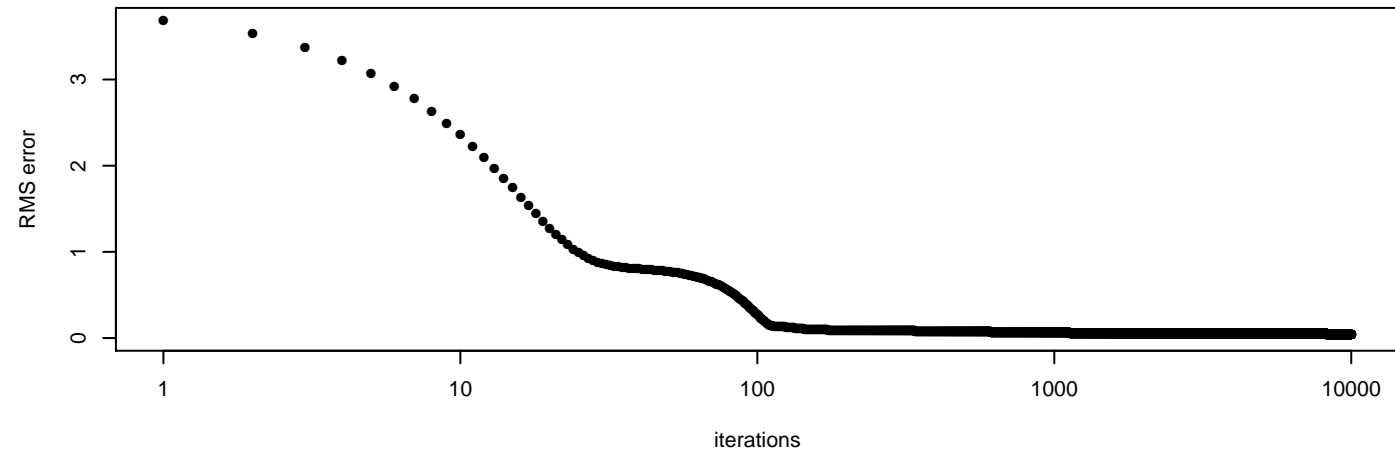


Negative Perturbation

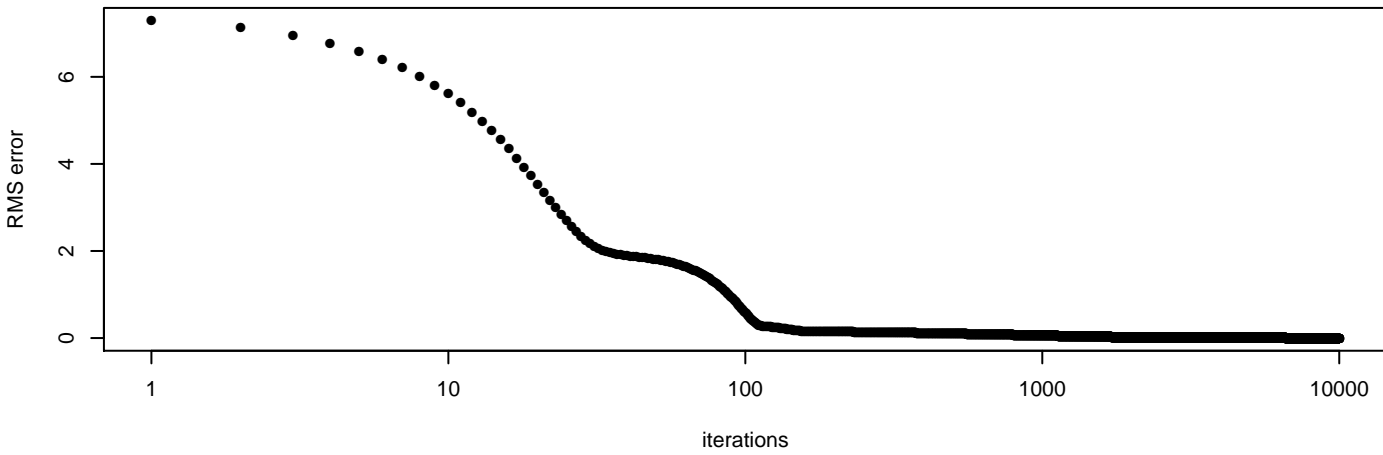


Parameter4

Positive Perturbation

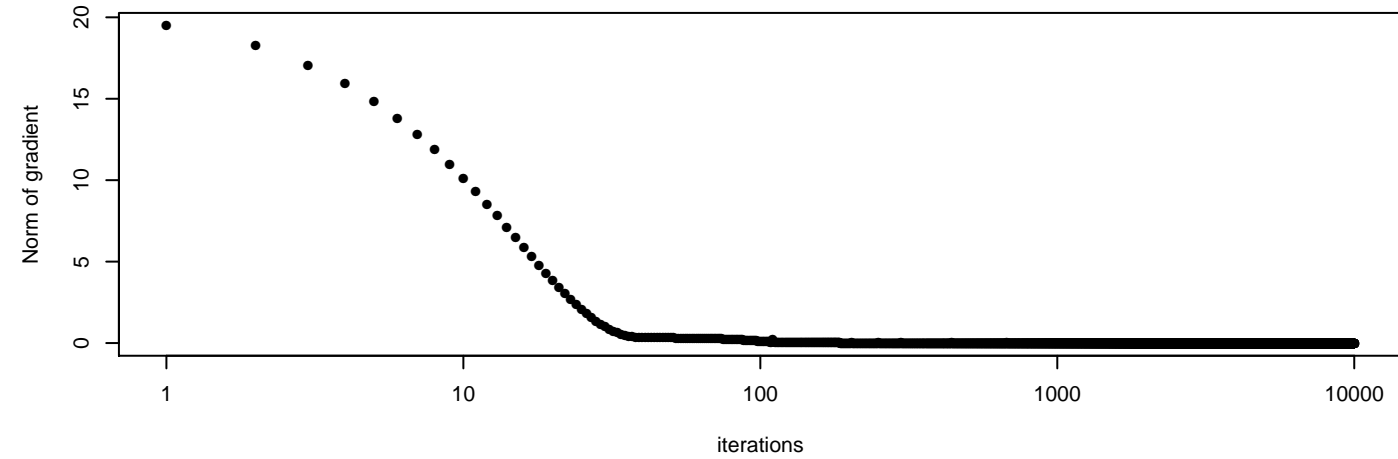
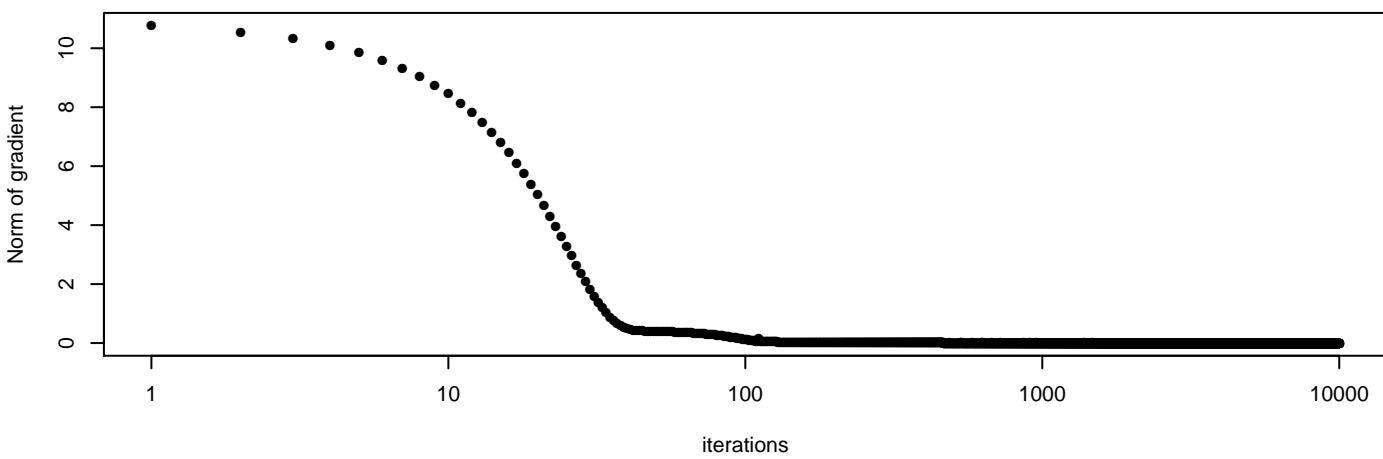
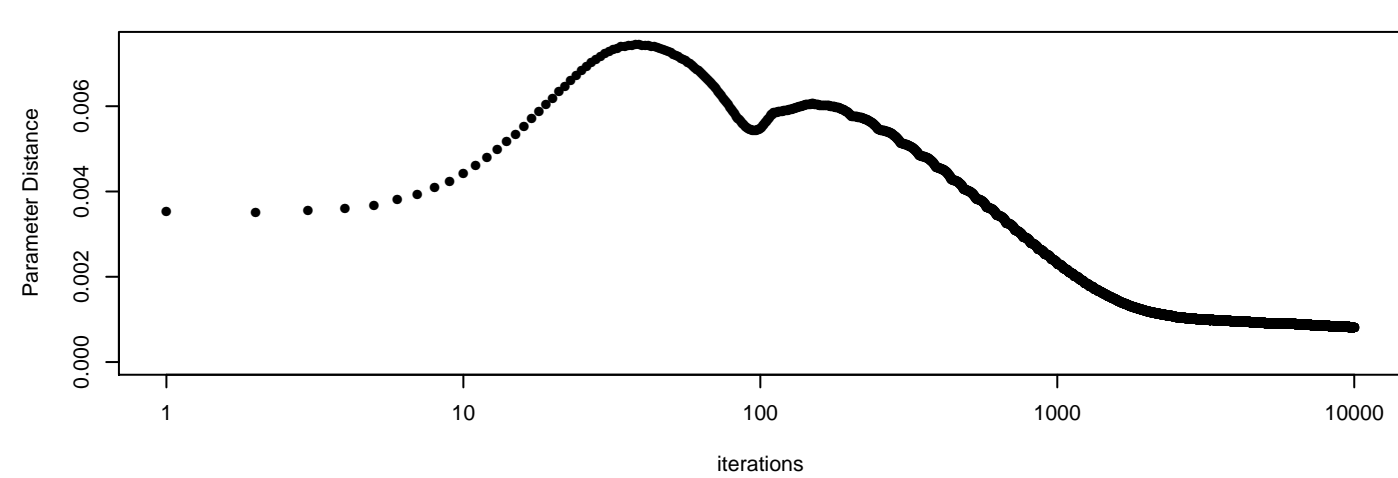
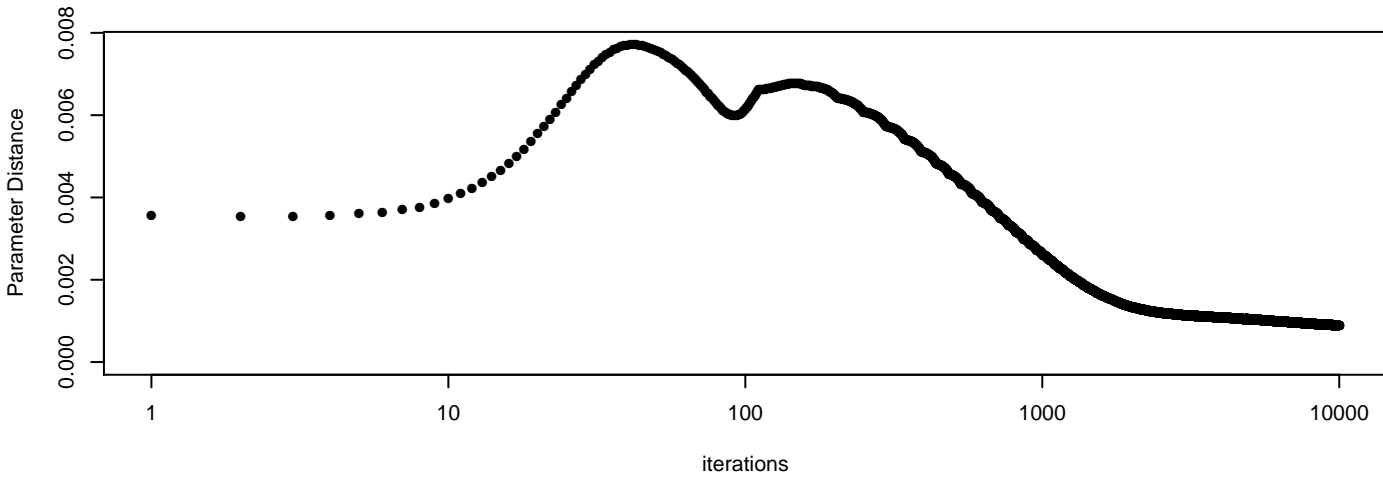
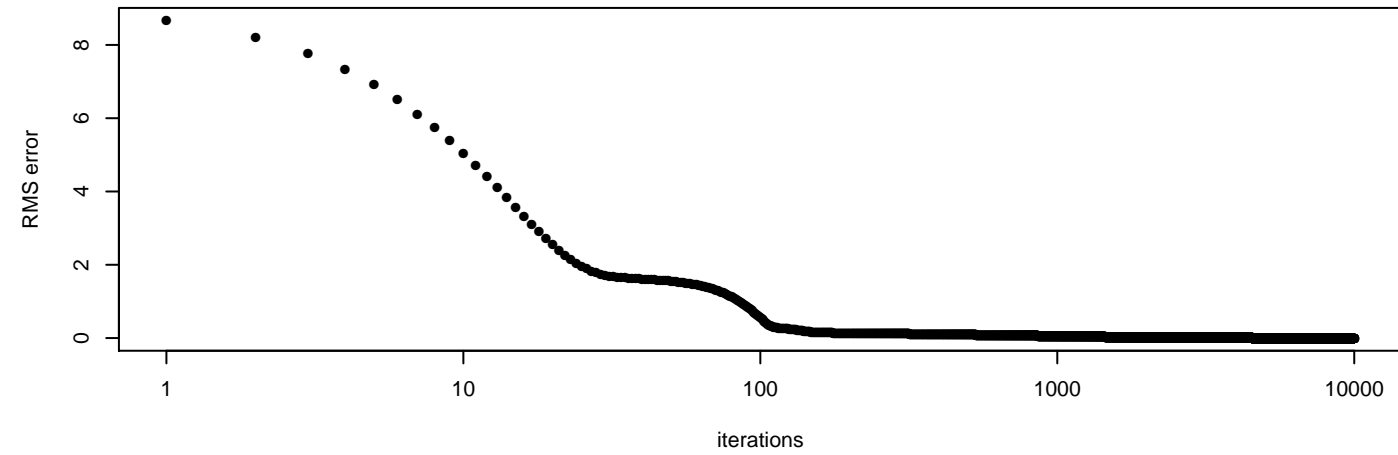


Negative Perturbation

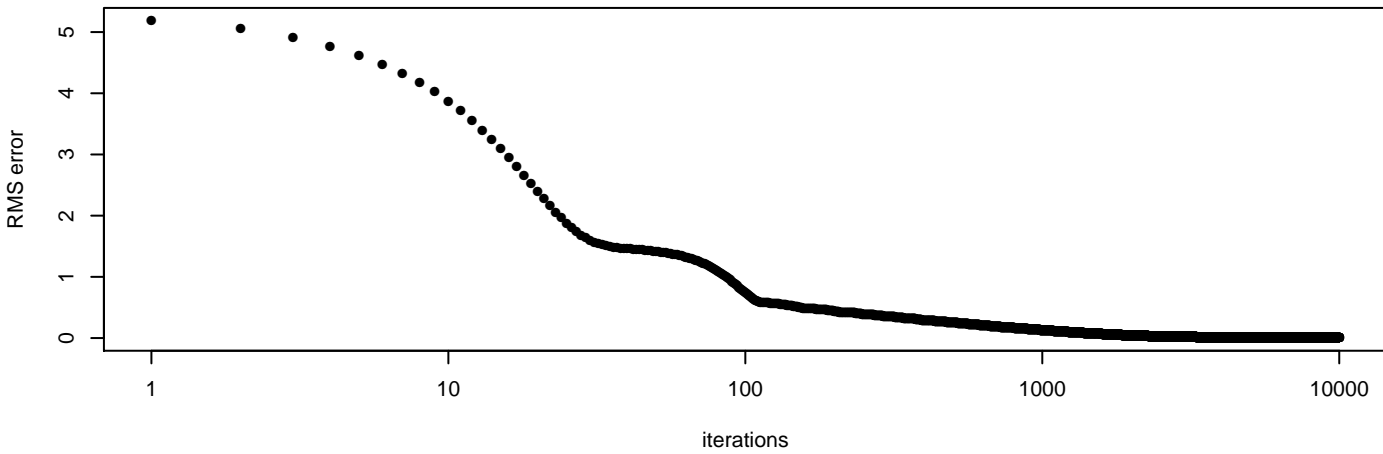


Parameter5

Positive Perturbation

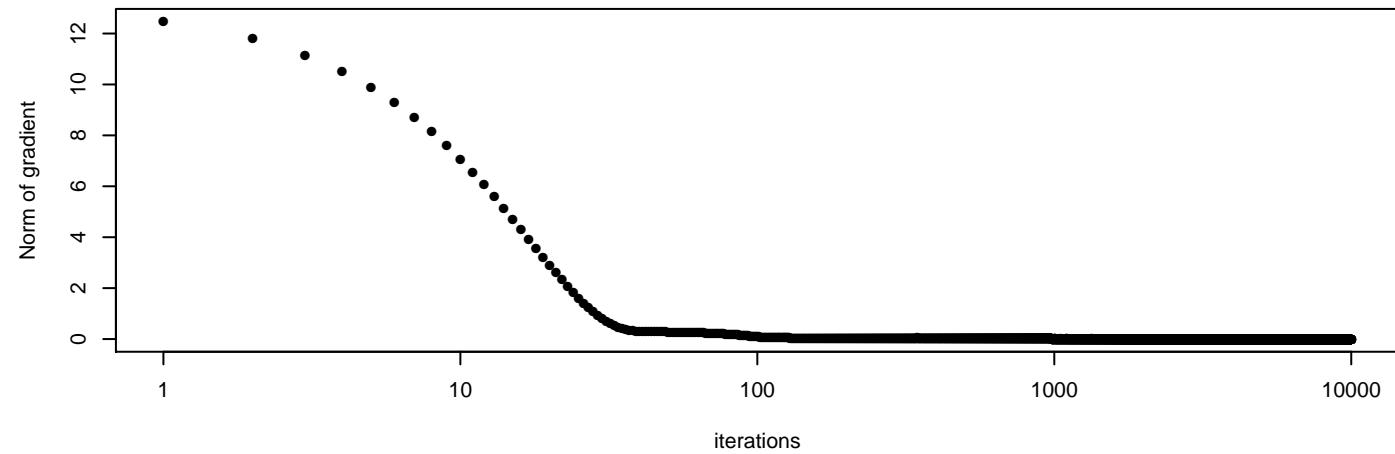
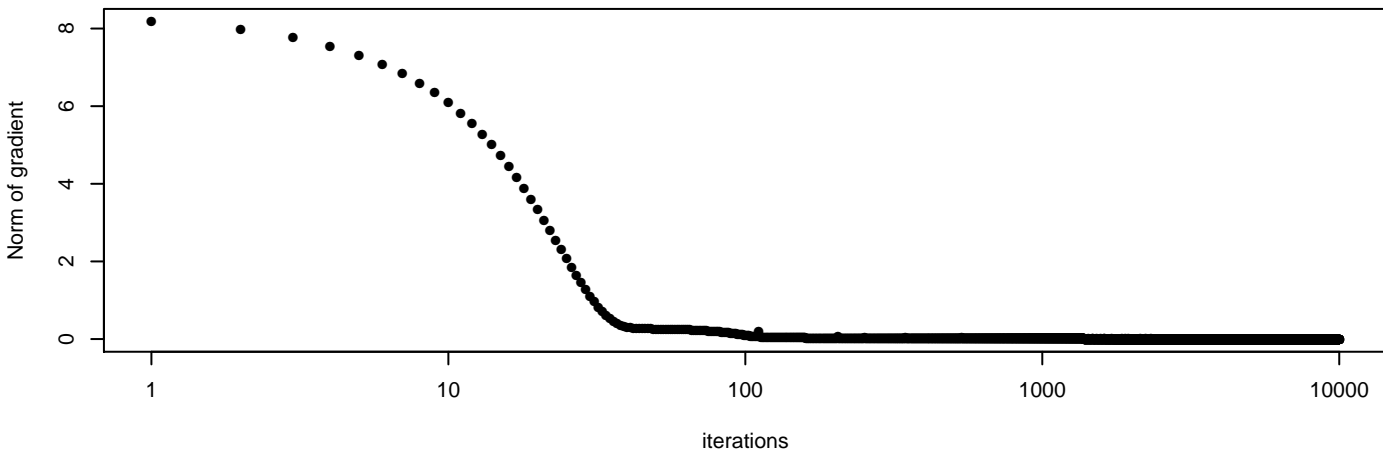
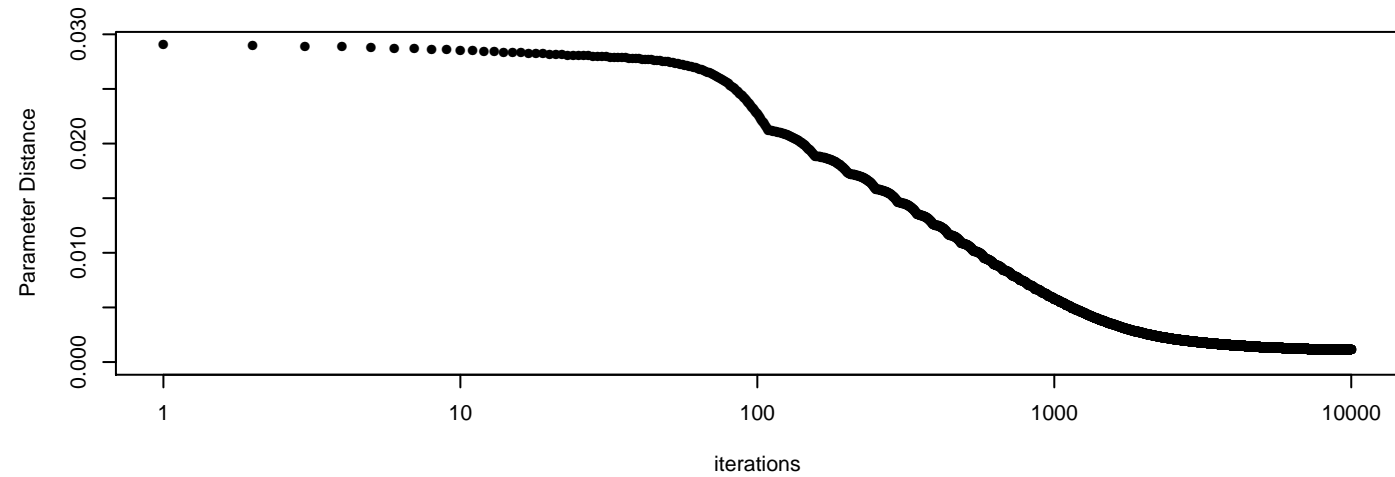
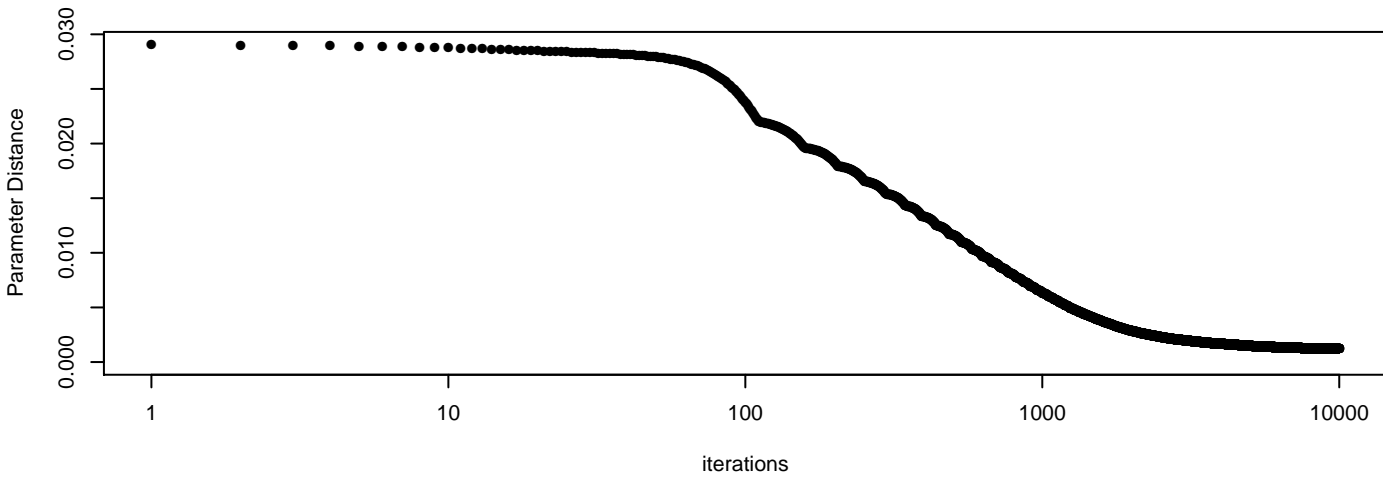
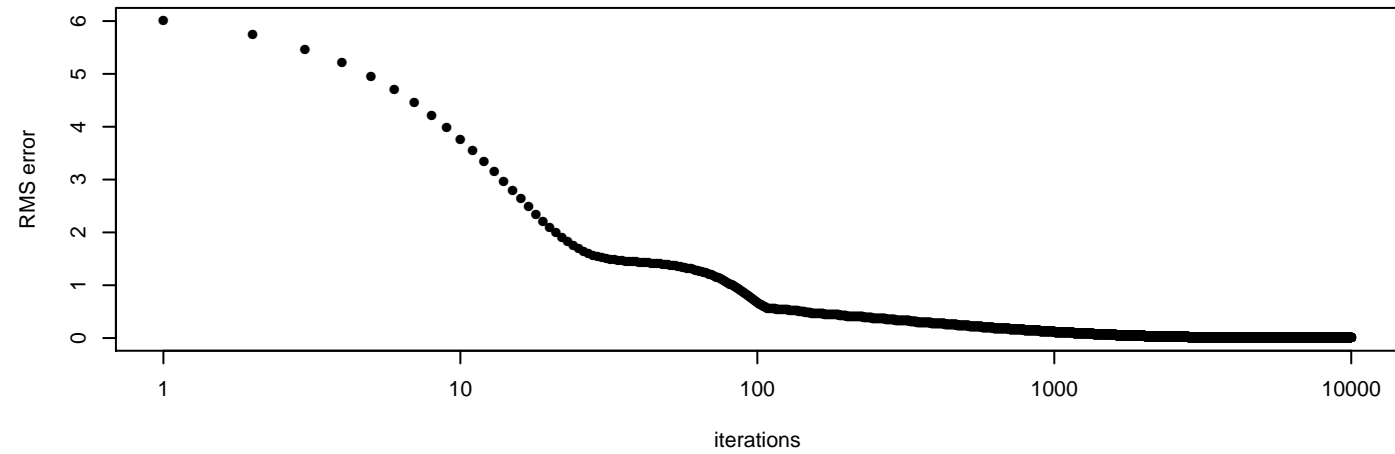


Negative Perturbation

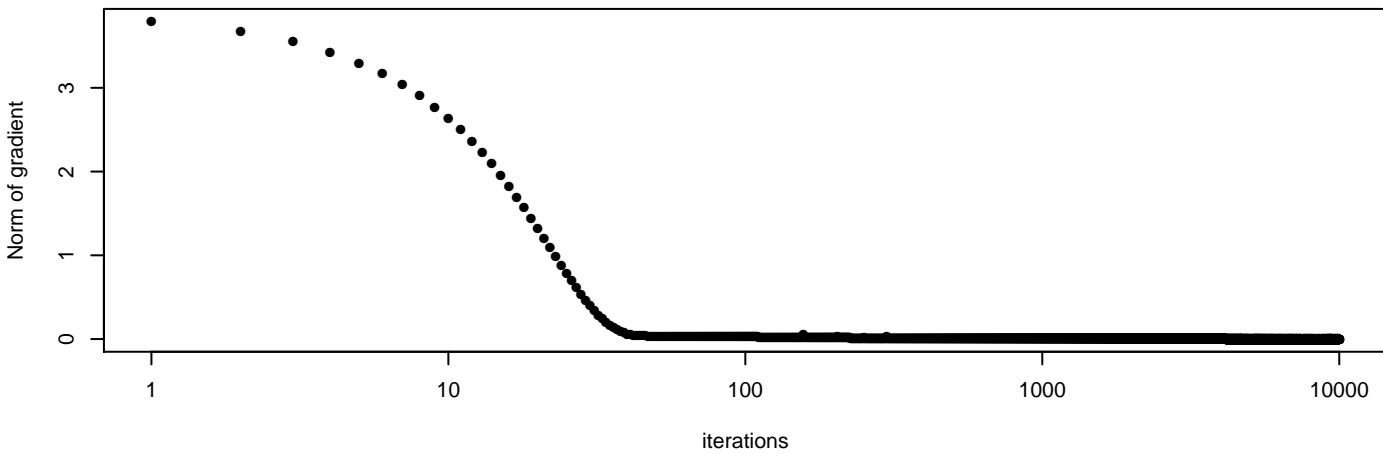
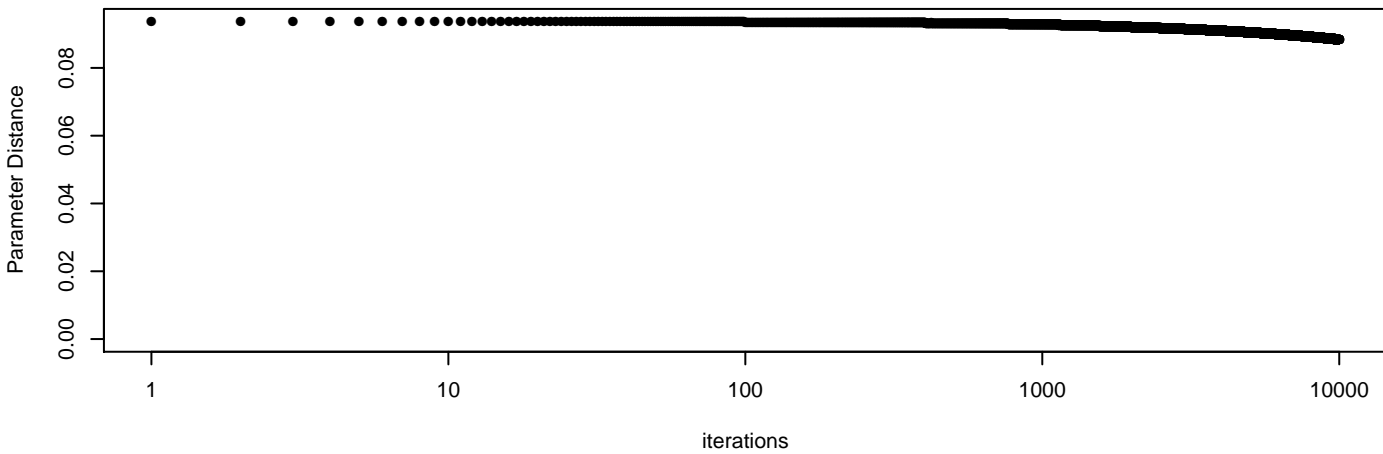
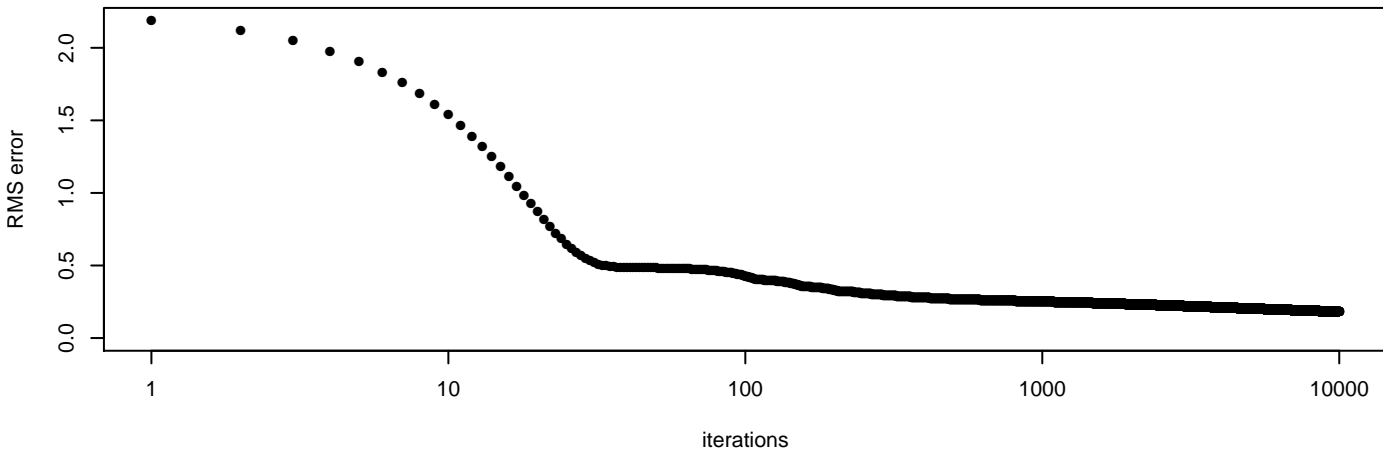


Parameter6

Positive Perturbation

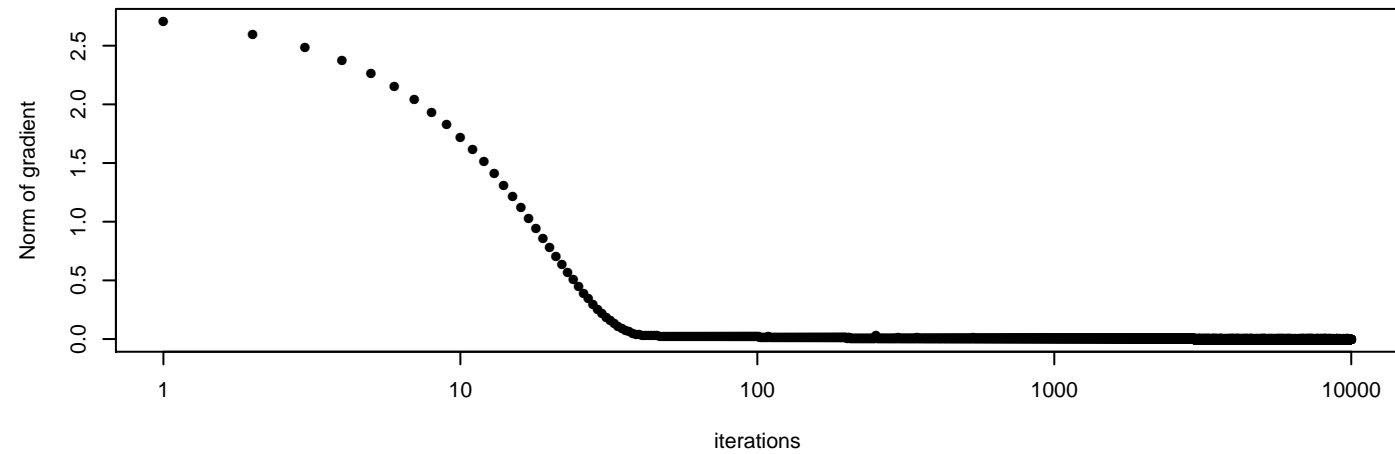
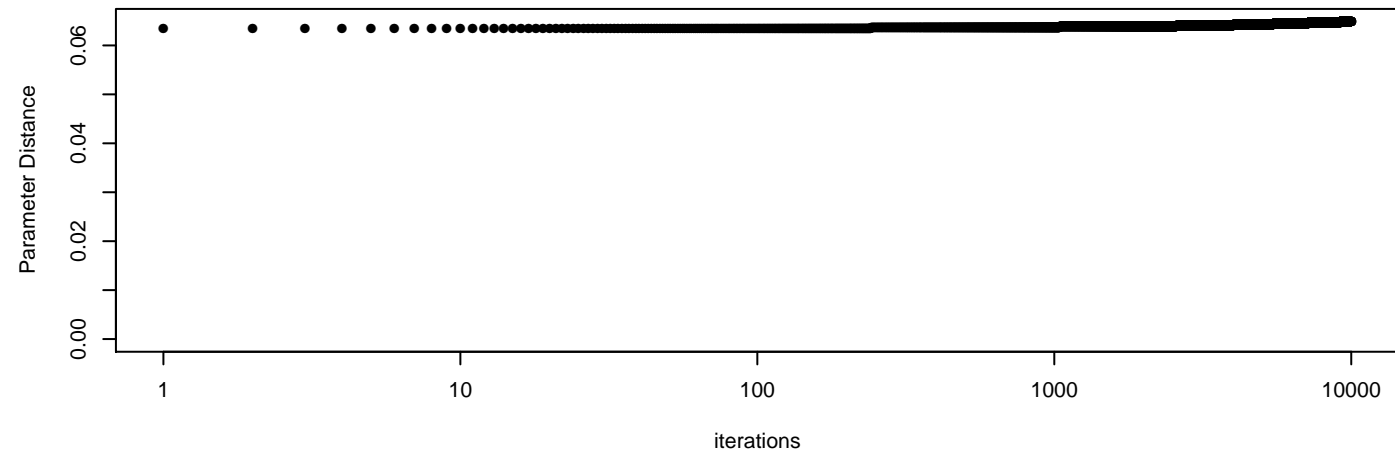
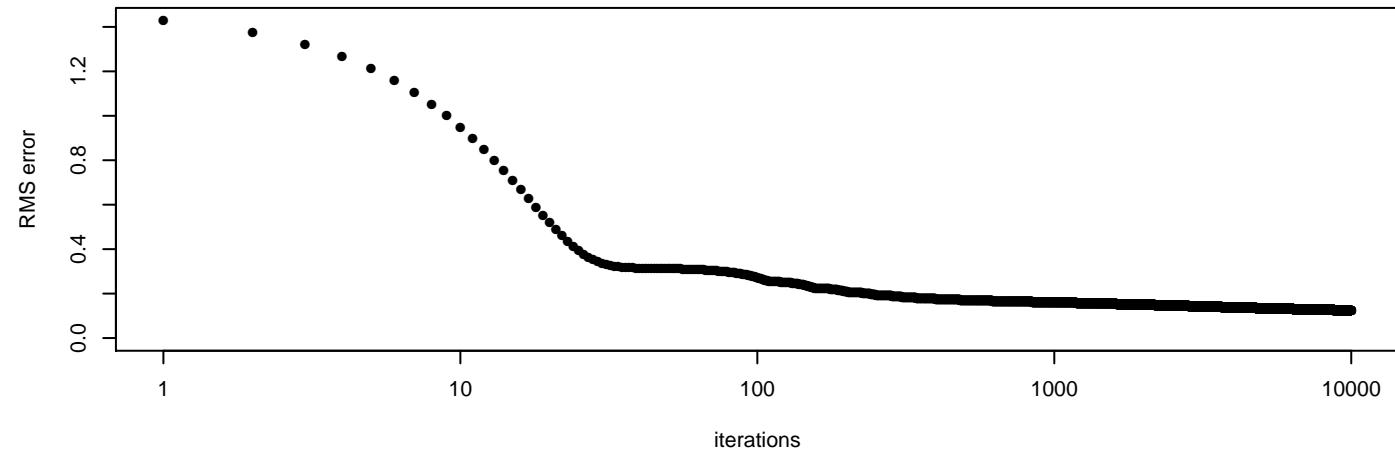


Negative Perturbation

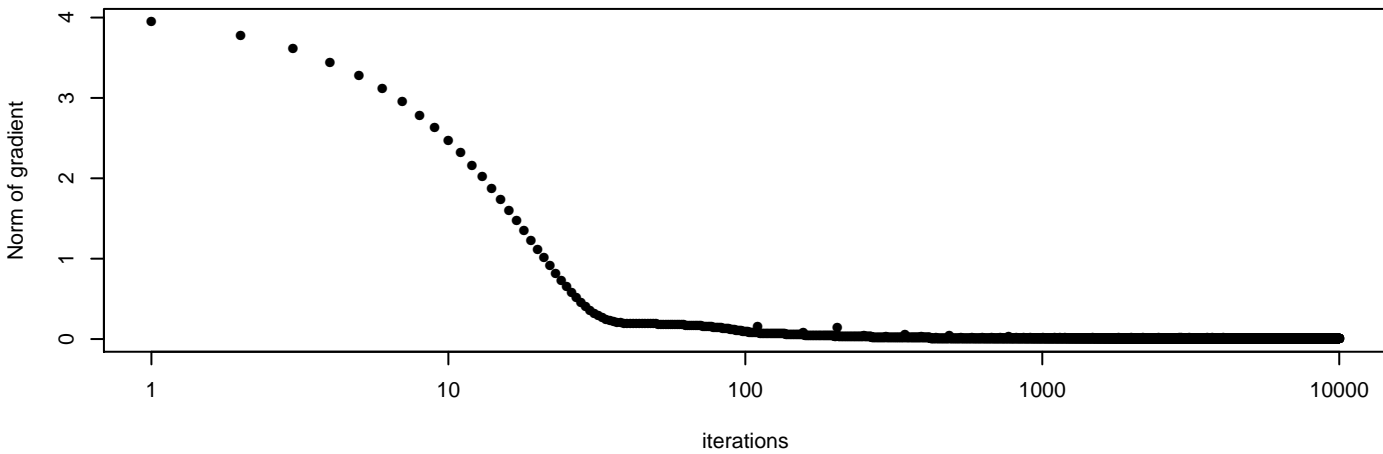
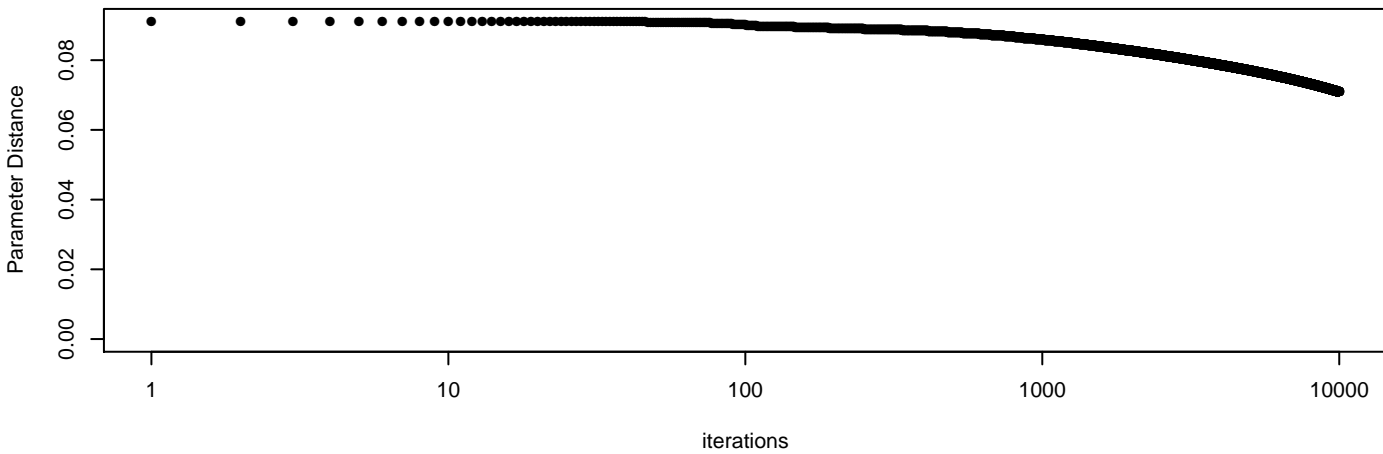
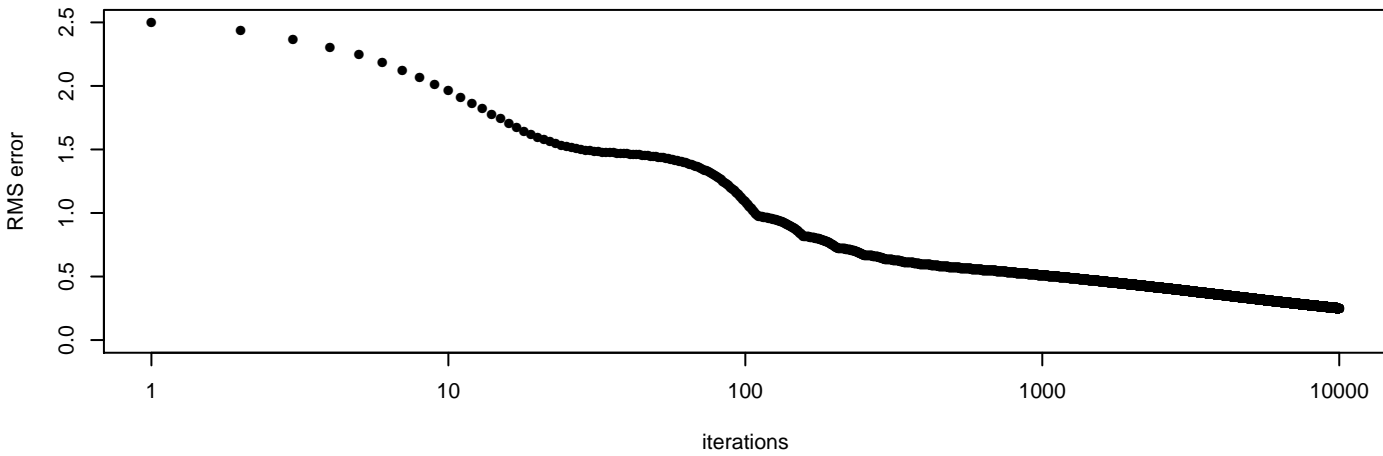


Parameter7

Positive Perturbation

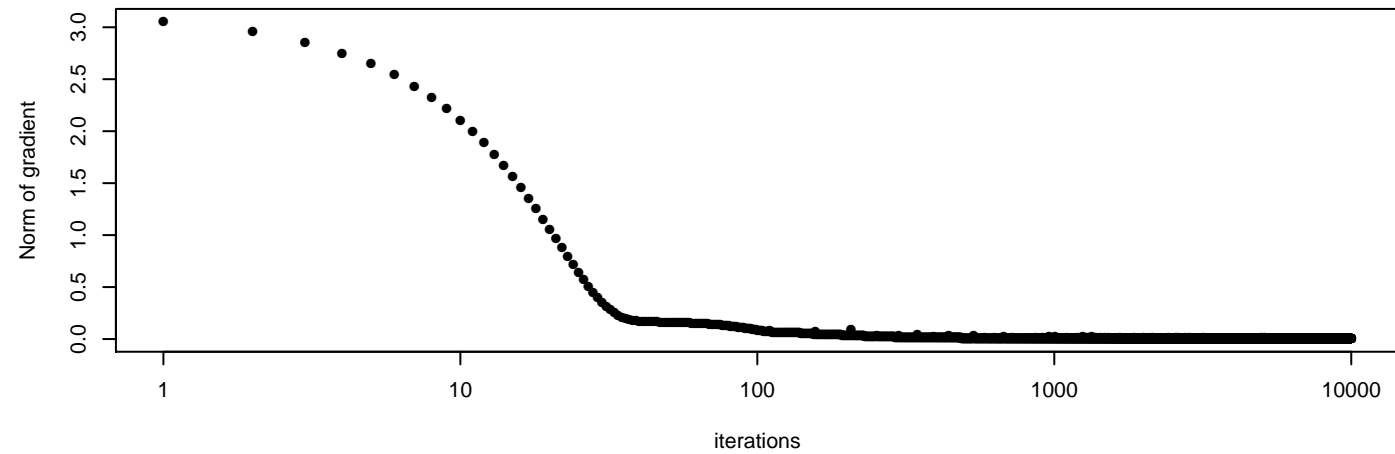
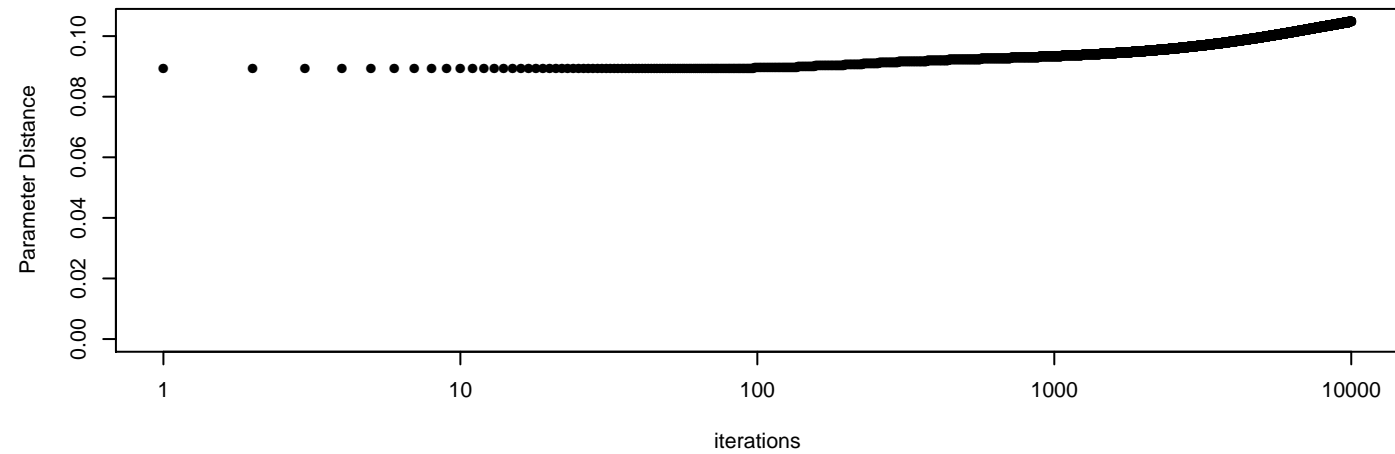
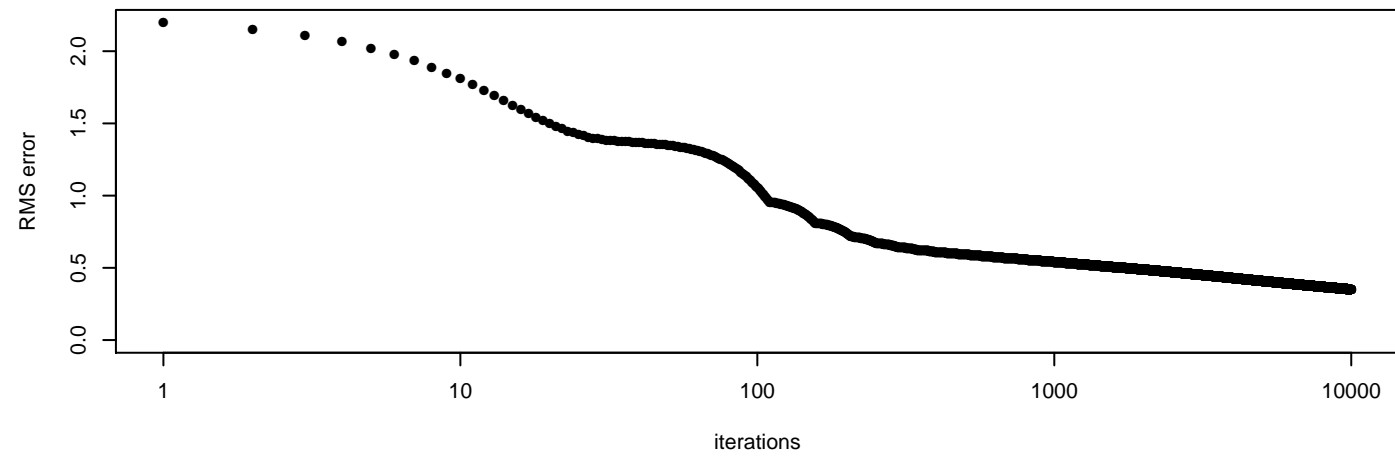


Negative Perturbation

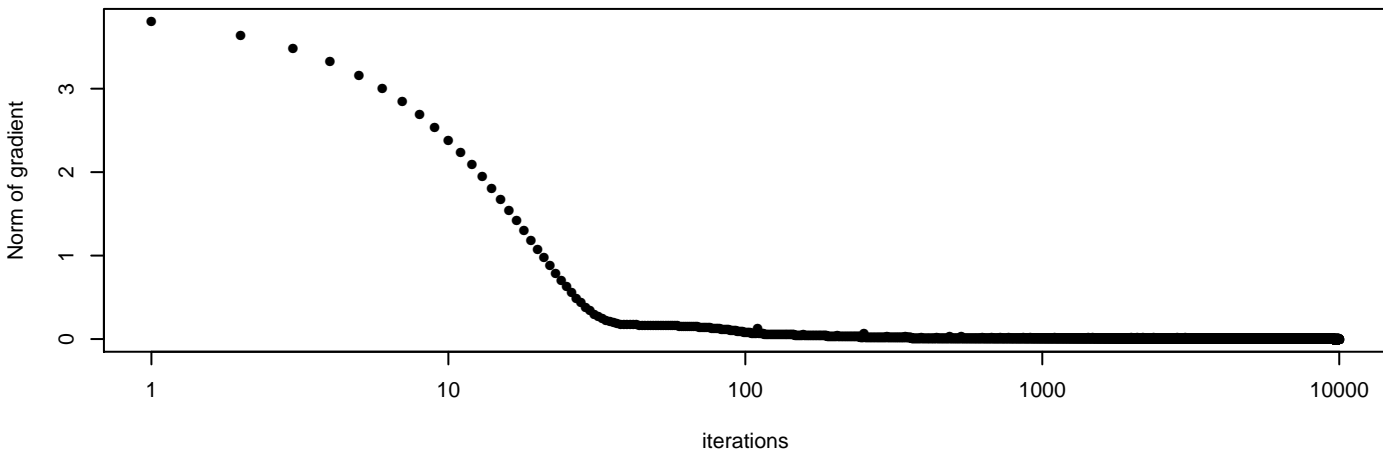
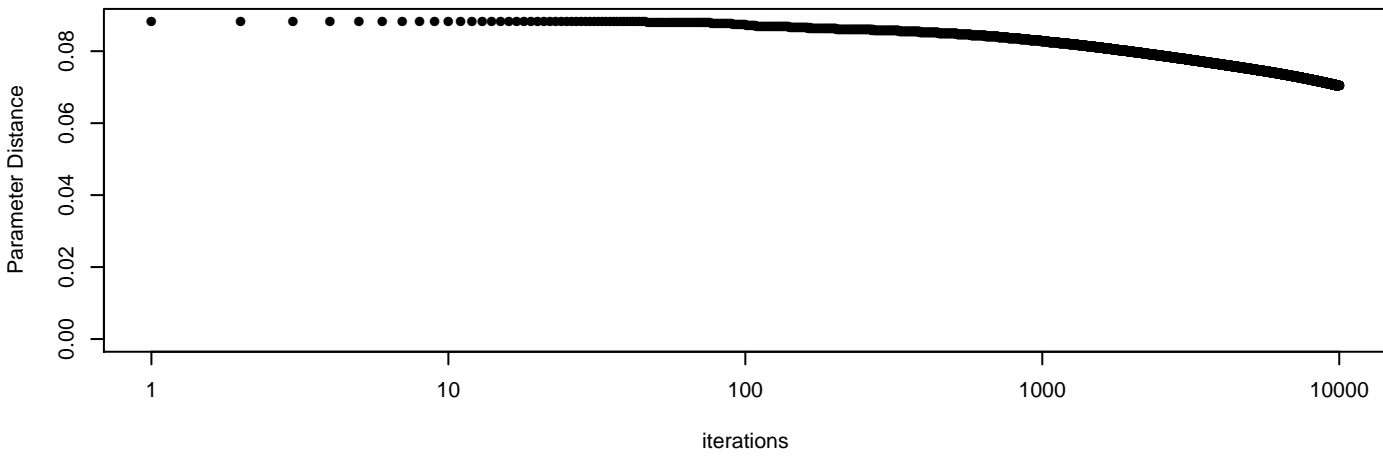
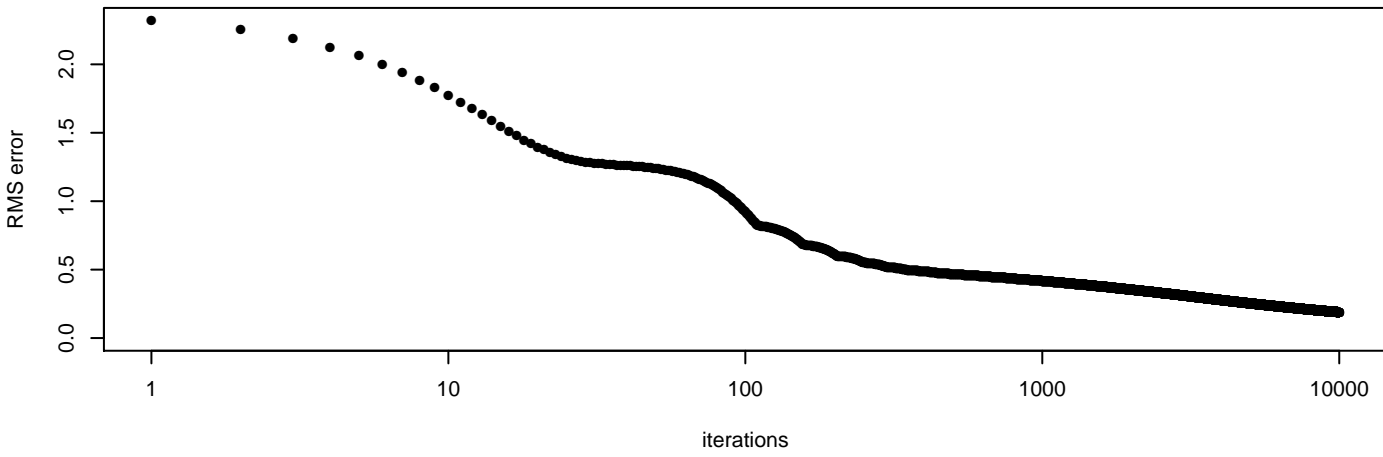


Parameter8

Positive Perturbation

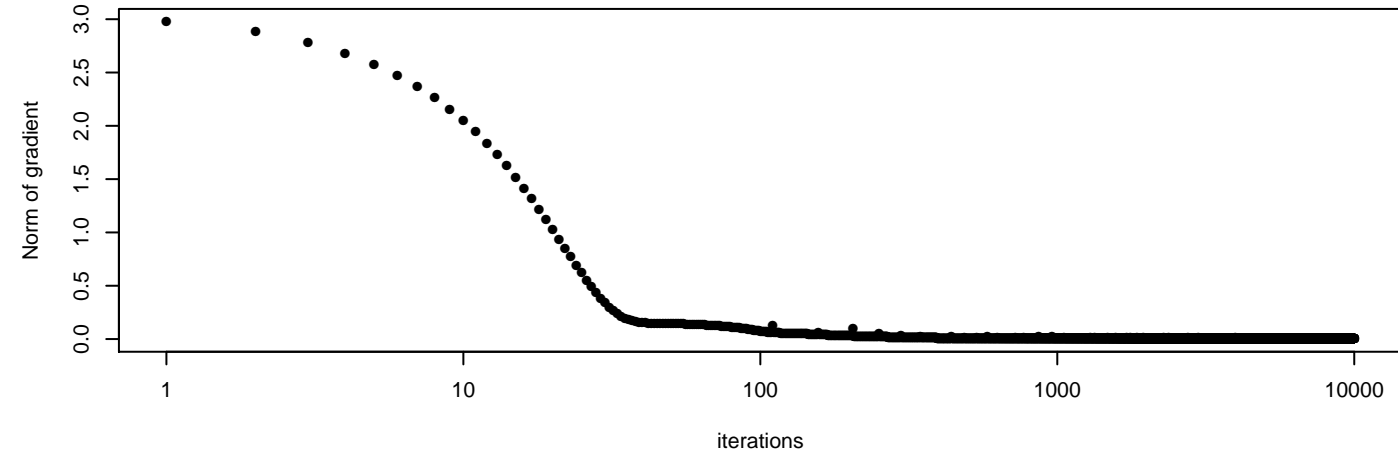
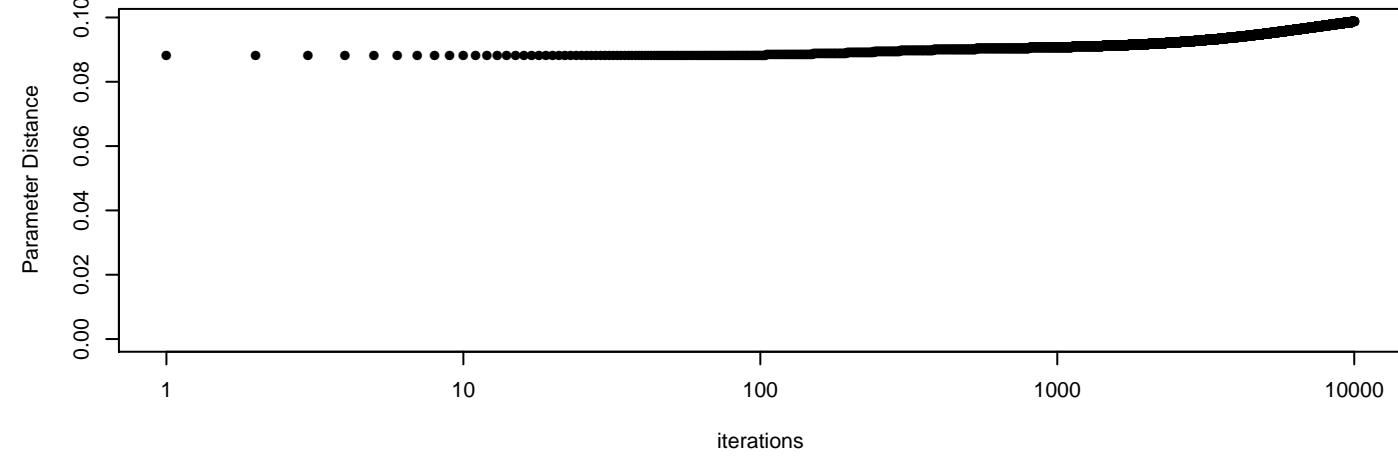
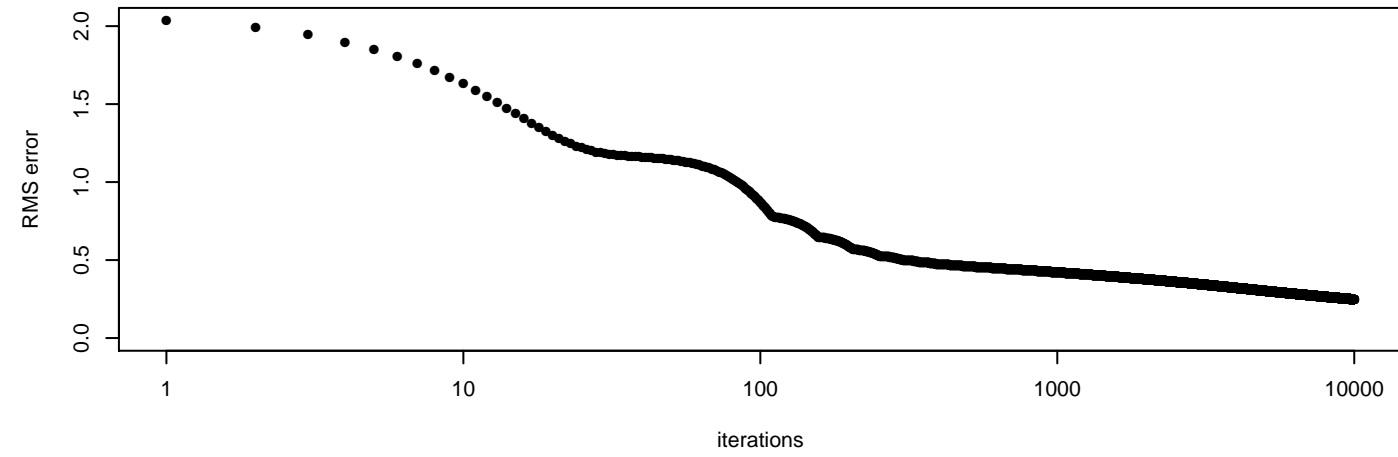


Negative Perturbation



Parameter9

Positive Perturbation



Section 4 : Sample Prediction of Short and Long Training Times of SA

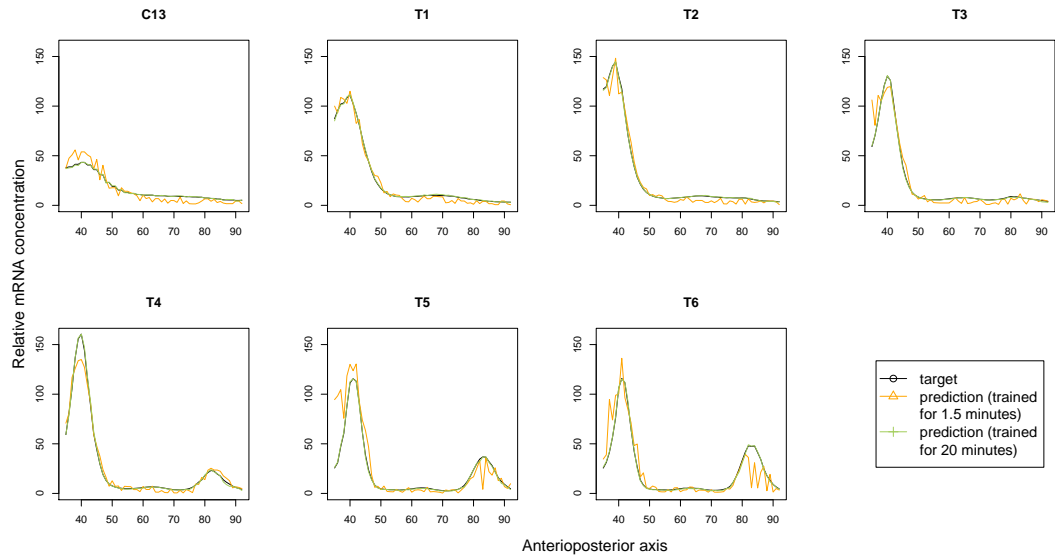


Figure 1: **Comparison predicted transcriptional output of a short and long trained model using SA** Each panel shows the target (circle) and the predicted transcriptional output for the seven developmental time points. The models are trained using SA_geom given a small and large iteration budget. The shortly trained model (triangle) achieved a CC of 0.974 after 1.5 minutes. The longer trained model (plus) has a CC of 1.0 and used 234 minutes.

Section 5 : Parameter Limits

<i>Parameter</i>	eve_stripe2_multitime		dros_singletime	
	<i>lower bound</i>	<i>upper bound</i>	<i>lower bound</i>	<i>upper bound</i>
R_0	166	255	1	5
G_0	$1 \cdot 10^{-6}$	20	$1 \cdot 10^{-6}$	5
K	$1 \cdot 10^{-6}$	0.05	$1 \cdot 10^{-6}$	1000
C	$1 \cdot 10^{-6}$	10	$1 \cdot 10^{-6}$	10
E	$1 \cdot 10^{-6}$	0.99999	$1 \cdot 10^{-6}$	1

Table 1: **Parameter ranges for Θ .** The parameter ranges are shown for the two datasets we use. Value ranges for eve_stripe2_multitime are obtained from Janssens et al. [2006] and adjusted to fit the data from dros_singletime. While SA and GA uses the lower and upper range, the unconstrained optimization method GD uses a parameter transformation that sets the lower bound to be 0 for all parameters.

Section 6 : Performance of Simulated Annealing Variants

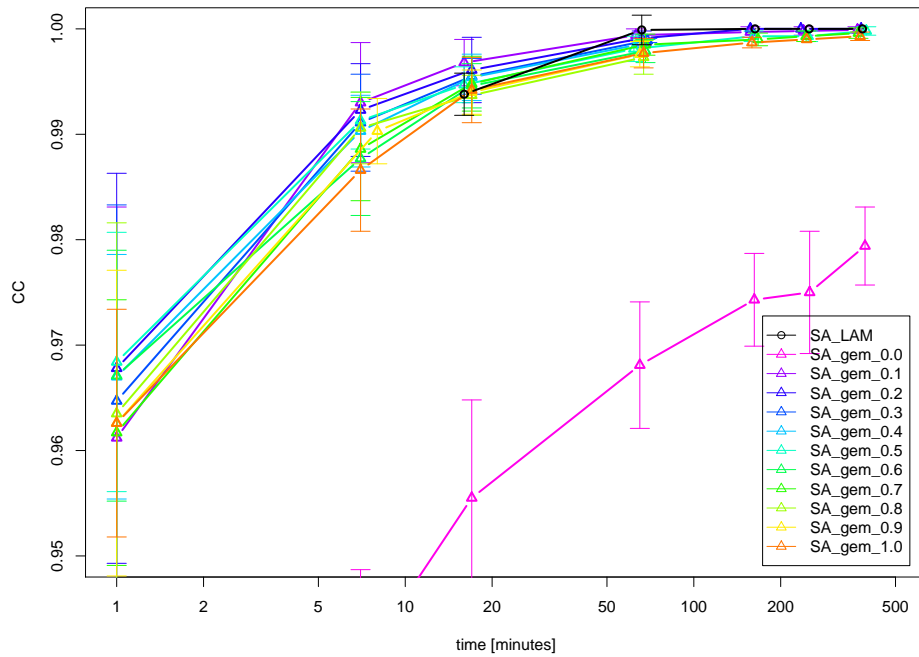


Figure 1: **Comparison of the different SA optimization methods** The figure shows the correlation coefficient (CC) achieved on the test set of each cross validation fold by the different SA variants. SA_LAM is the benchmark model introduced by Reinitz et al. [2003], the other SA variants use the geometric cooling schedule. SA_geom_0.0 has a neighbour function, which randomly samples new values within the range of each parameter. SA_geom_1.0 samples values out of a ball around the current parameter, which shrinks with the temperature and has a radius of 100% of the parameter range. The other variants reduce the radius of the ball by the indicated fraction.

Section 7 : Gradient Descent convergence behaviour from 100 random starting points

	$D(\theta^{(0)}, \hat{\theta})$	$D(\theta', \hat{\theta})$	$\ \nabla B(\theta')\ $	$B(\theta')$	<i>% RMS error improvement</i>
1	0.07	0.14	$3.8 \cdot 10^{-14}$	41.19	100
2	0.08	0.13	$4.6 \cdot 10^{-13}$	41.19	100
3	0.11	0.12	$3.2 \cdot 10^{-08}$	41.19	100
4	0.11	0.13	$2.4 \cdot 10^{-40}$	41.19	100
5	0.12	0.14	$4.5 \cdot 10^{-11}$	41.19	100
6	0.09	0.13	$3.6 \cdot 10^{-10}$	41.19	100
7	0.11	0.12	$1.2 \cdot 10^{-09}$	41.19	100
8	0.1	0.13	$2.4 \cdot 10^{-12}$	41.19	100
9	0.08	0.14	$2.3 \cdot 10^{-20}$	41.19	100
10	0.08	0.14	$4.7 \cdot 10^{-39}$	41.19	100
11	0.1	0.12	$7.8 \cdot 10^{-12}$	41.19	100
12	0.11	0.13	$9.6 \cdot 10^{-12}$	41.19	100
13	0.09	0.14	$3.7 \cdot 10^{-19}$	41.19	100
14	0.09	0.12	$5.5 \cdot 10^{-10}$	41.19	100
15	0.12	0.14	$8.0 \cdot 10^{-13}$	41.19	100
16	0.12	0.14	$4.5 \cdot 10^{-10}$	41.19	100
17	0.11	0.12	$3.3 \cdot 10^{-08}$	41.19	100
18	0.11	0.12	$6.5 \cdot 10^{-13}$	41.19	100
19	0.11	0.12	$1.0 \cdot 10^{-09}$	41.19	100
20	0.11	0.13	$7.6 \cdot 10^{-10}$	41.19	100
21	0.08	0.12	$7.8 \cdot 10^{-15}$	41.19	100
22	0.11	0.12	$1.5 \cdot 10^{-17}$	41.19	100
23	0.1	0.13	$9.1 \cdot 10^{-10}$	41.19	100
24	0.09	0.13	$9.7 \cdot 10^{-29}$	41.19	100
25	0.09	0.14	$4.0 \cdot 10^{-17}$	41.19	99.99
26	0.09	0.12	$3.3 \cdot 10^{-08}$	41.19	99.99
27	0.09	0.13	$3.7 \cdot 10^{-13}$	41.19	99.97
28	0.1	0.11	$3.0 \cdot 10^{-10}$	41.19	99.97
29	0.09	0.13	$3.7 \cdot 10^{-24}$	41.19	99.97
30	0.1	0.14	$3.5 \cdot 10^{-15}$	41.19	99.95
31	0.11	0.13	$5.1 \cdot 10^{-15}$	41.19	99.95
32	0.09	0.15	$1.8 \cdot 10^{-08}$	41.19	99.9
33	0.1	0.11	$7.0 \cdot 10^{-09}$	41.19	99.9
34	0.06	0.13	$6.7 \cdot 10^{-20}$	41.19	99.88
35	0.09	0.14	$4.8 \cdot 10^{-18}$	41.19	99.87
36	0.1	0.13	$2.4 \cdot 10^{-18}$	41.19	99.86
37	0.13	0.15	0.0022	28.52	99.84
38	0.1	0.14	$1.2 \cdot 10^{-11}$	41.19	99.84
39	0.1	0.09	$3.2 \cdot 10^{-06}$	32.41	99.8
40	0.1	0.13	$2.3 \cdot 10^{-08}$	41.19	99.79

	$D(\Theta^{(0)}, \hat{\Theta})$	$D(\Theta', \hat{\Theta})$	$\ \nabla B(\Theta')\ $	$B(\Theta')$	<i>% RMS error improvement</i>
41	0.13	0.15	$9.3 \cdot 10^{-12}$	41.19	99.79
42	0.08	0.04	0.01	1.26	99.77
43	0.09	0.13	$2.9 \cdot 10^{-30}$	41.19	99.73
44	0.08	0.05	0.015	1.48	99.72
45	0.08	0.13	$6.3 \cdot 10^{-25}$	41.19	99.68
46	0.11	0.13	$1.8 \cdot 10^{-14}$	41.19	99.67
47	0.09	0.13	$2.2 \cdot 10^{-24}$	41.19	99.63
48	0.11	0.14	$1.2 \cdot 10^{-17}$	41.19	99.61
49	0.11	0.16	0.007	28.55	99.59
50	0.09	0.13	$6.6 \cdot 10^{-19}$	41.19	99.59
51	0.1	0.12	$1.4 \cdot 10^{-18}$	41.19	99.59
52	0.1	0.08	0.014	1.94	99.59
53	0.1	0.06	0.018	2.19	99.58
54	0.09	0.11	0.027	11.2	99.58
55	0.08	0.07	0.015	3.07	99.57
56	0.1	0.08	0.038	2.85	99.54
57	0.1	0.12	$6.3 \cdot 10^{-17}$	41.19	99.54
58	0.11	0.08	0.043	4.89	99.51
59	0.09	0.05	0.018	1.89	99.39
60	0.11	0.15	2.2	33.04	99.34
61	0.1	0.08	0.018	2.44	99.06
62	0.11	0.15	0.0065	25.64	98.97
63	0.09	0.09	0.098	8.6	98.86
64	0.09	0.1	0.037	11.04	98.82
65	0.11	0.15	0.006	25.64	98.68
66	0.05	0.04	0.0029	0.61	98.52
67	0.1	0.15	0.0086	25.64	98.22
68	0.07	0.07	0.011	1.55	97.72
69	0.1	0.08	0.014	2.32	97.16
70	0.08	0.05	0.012	1.42	96.54
71	0.07	0.05	0.013	1.53	96.26
72	0.1	0.06	0.013	1.56	96.21
73	0.11	0.08	0.062	5.27	95.93
74	0.1	0.05	0.016	1.75	95.74
75	0.1	0.08	0.036	7.86	95.7
76	0.1	0.05	0.02	2.19	94.59
77	0.11	0.07	0.016	2.28	94.47
78	0.13	0.12	0.047	12.36	94.37
79	0.11	0.08	0.02	2.52	93.89
80	0.1	0.08	0.012	2.52	93.73

	$D(\Theta^{(0)}, \hat{\Theta})$	$D(\Theta', \hat{\Theta})$	$\ \nabla B(\Theta')\ $	$B(\Theta')$	<i>% RMS error improvement</i>
81	0.12	0.07	0.032	2.67	93.52
82	0.12	0.08	0.024	2.68	93.4
83	0.1	0.07	0.027	2.76	93.3
84	0.13	0.1	0.028	2.86	93.05
85	0.11	0.07	0.035	2.81	92.68
86	0.1	0.07	0.036	2.99	91.84
87	0.1	0.07	0.023	3.61	91.22
88	0.11	0.09	0.083	6.11	89.99
89	0.1	0.06	0.055	4.16	89.89
90	0.13	0.1	0.018	4.21	89.76
91	0.09	0.08	0.016	3.52	89.66
92	0.08	0.05	0.021	2.84	89.65
93	0.11	0.09	0.021	4.39	89.33
94	0.11	0.1	0.028	4.42	89.21
95	0.11	0.08	0.06	4.59	88.81
96	0.12	0.09	0.023	4.36	88.67
97	0.13	0.11	0.052	5.6	86.41
98	0.11	0.09	0.047	5.61	86.39
99	0.1	0.09	0.032	7.31	82.26
100	0.11	0.1	0.041	8.04	66.21

Table 1: **Individual results of the convergence experiments from 100 random $\Theta^{(0)}$.** The table shows the result of the gradient descent optimization using the `no_max` model, no restarts, and a maximal iteration budget of 10000. The **second** column shows the distance between randomly generated starting point and known solution. The **third** column shows the distance between Θ' and the known solution. The norm of the gradient is shown in column **four** and the final error as well as the percentage RMS error improvement is shown in column **five** and **six**.

Section 8 : Performance of the genetic algorithm

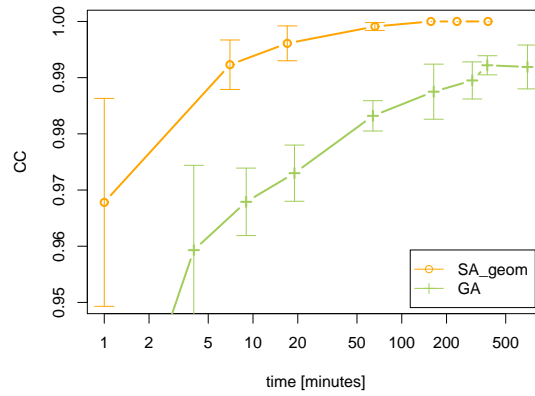


Figure 1: **Effectiveness of the GA optimization method.** The figure shows the cross-validated correlation coefficient (CC) as a function of run time (log scale) for the GA and SA_geom optimization methods.

Section 9 : Comparison real vs. synthetic data

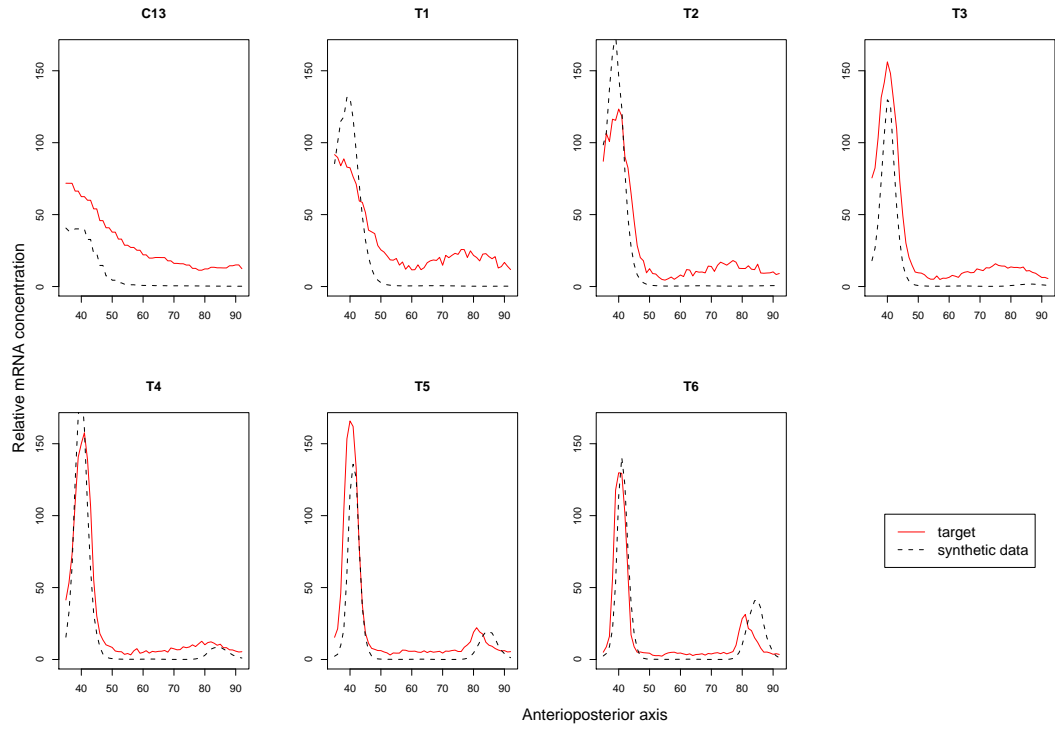
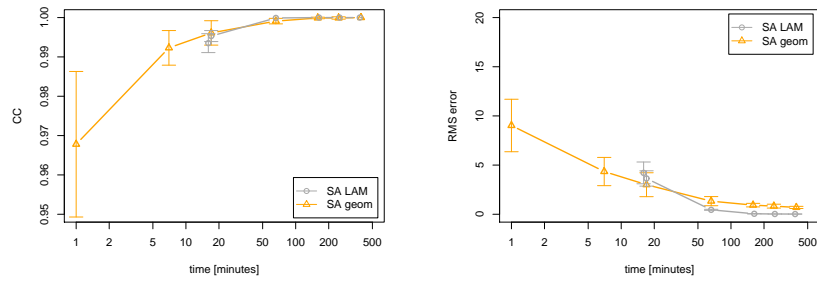


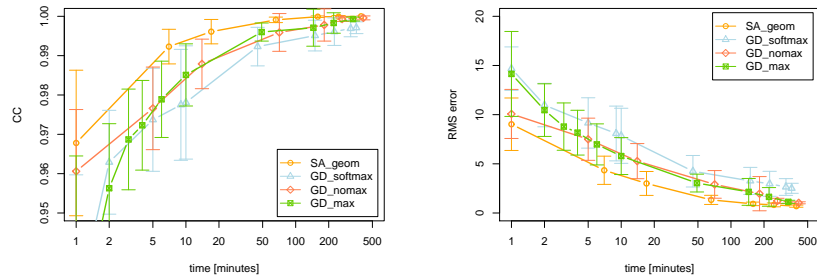
Figure 1: **Comparison between the synthetic and the real biological data.** The RMS error is 17.86 and the CC is 0.89 between synthetic and real data.

Section 10 : Comparison optimization methods using RMS error metric

a



b



c

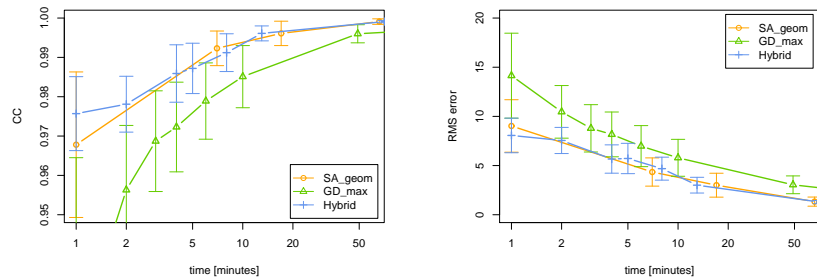


Figure 1: **Comparison of the different optimization methods using RMS error metric** The figure shows the RMS error $B(\theta')$ between prediction and target achieved on the test set of each cross validation fold by the different optimization methods. Panel **a** shows the comparison between SA_geom and SA_LAM. Note, SA_geom does not achieve zero error, but as demonstrated in the supplemental figure in Section 4 there is no visible difference between the target and the prediction made by SA_geom and hence the difference between SA_LAM and SA_geom in the RMS error metric are not relevant in practice. Panel **b** shows the performance of the GD variants and panel **c** shows the hybrid performance, respectively.

Section 11 : Comparison optimization methods for genes trained simultaneously

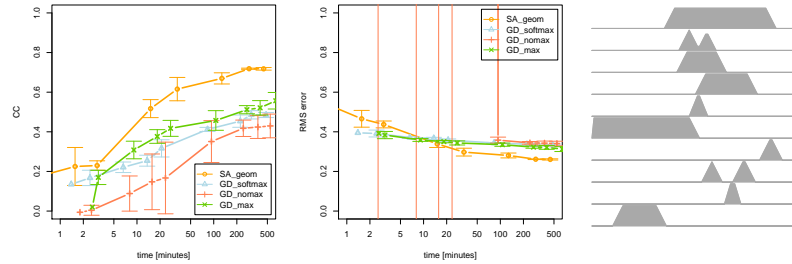


Figure 1: **Comparison of the different optimization methods for genes using real data and trained simultaneously** The figure shows the CC (left) and RMS error (middle) between prediction and target achieved on the test set of each cross validation fold by the different optimization methods. The training is done on several genes simultaneously, whose expression pattern is shown on the left (along the anterioposterior axis), “D_+4”, “h_stripe34”, “Kr_CD1_run”, “pdm2”, “run_stripe3”, “hb_anterior”, “cad_+14”, “eve_stripe_46”, “run_stripe5”, and “slp_-3”. Note, since gd_nomax is not capped the RMS error can become very large.

References

- Hilde Janssens, Shuling Hou, Johannes Jaeger, Ah-Ram Kim, Ekaterina Myasnikova, David Sharp, and John Reinitz. Quantitative and predictive model of transcriptional control of the *Drosophila melanogaster* even skipped gene. *Nat Genet*, 38(10):1159–1165, Oct 2006. doi: 10.1038/ng1886. URL <http://dx.doi.org/10.1038/ng1886>.
- John Reinitz, Shuling Hou, and David H. Sharp. Transcriptional Control in *Drosophila*. *Complexus*, 1:54–64, 2003.