Constrained Optimization Homework

Chiara Scagliola
Politecnico di Torino
Student Id : s303037
s303037@studenti.polito.it

Domenico Mereu
Politecnico di Torino
Student Id : s302151
s302151@studenti.polito.it

***Abstract*—In this report we are going to describe the results obtained by minimizing the constrained optimization problem described by equation (3) in [1]. The results are obtained by applying the projected gradient method using backtrack as a line search strategy using Exact Derivatives and Finite Differences.**

I. Method

Projected gradient descent is an optimization algorithm that is used to find the minimum of a function that is subject to constraints. It works by iteratively updating the variables of the function in the direction of the negative gradient, while projecting the updated variables back onto the feasible region χ, defined by the constraints. The main difference from the regular Steepest Descent is, of course, the constraint $x \in \chi$, this is crucial for the algorithm as it drastically changes the structure of the stopping criteria. While in the Unconstrained case it is indeed necessary to verify stationarity to declare convergence, in the Constrained Case there is actually no guarantee that the optimal point for the constrained problem, by definition feasible, is stationary with respect to the objective function.

II. Problem

The problem requires minimization of the convex function $f : \mathbb{R}^n \to \mathbb{R}$ defined as:

$$f(x) = \sum_{i}^{n} i x_i^2 \tag{1}$$

Subject to the following constraints

$$-5.12 \leq x_i \leq 5.12 \quad \forall i = 1, ..., n \tag{2}$$

$f(x)$ has a single local minimum, that is also a global minimum, since it is a convex, lower bounded function. This suggests that regardless of the starting guess, the iterative method will always converge to the same point. The minimum is mathematically $x_{min} = (0, ..., 0)^T$, where $f(x_{min}) = 0$. The gradient was computed and reported in equation (3).

$$\frac{\partial f}{\partial x_i} = 2ix_i; \nabla f(x) = (2x_1, 4x_2, ..., 2nx_n)^T \tag{3}$$

The constraints (2) define a box-shaped feasible set χ , for that it is possible to define a projector P_χ as follows:

$$P_\chi x = \begin{cases} +5.12 & \text{if} \quad x_i > +5.12 \\ -5.12 & \text{if} \quad x_i < -5.12 \\ x_i & \text{otherwise} \end{cases} \tag{4}$$

the problem was also tested with the following other constrains

$$X = [1, 5.12]^n \tag{5}$$

$$X = [-5.12, 5.12] \times [1, 5.12]^{n-1} \tag{6}$$

$$X = [-5.12, 5.12]^{n/2} \times [1, 5.12]^{n/2} \tag{7}$$

III. Exact gradient descent

In this section we show the parameters chosen and results obtained using the projected gradient method with exact derivatives. The problem has been solved in $\mathbb{R}^{10^3}$, $\mathbb{R}^{10^4}$, $\mathbb{R}^{10^5}$.

A. Configuration of parameters

In the projected gradient method there are two main scalar parameter γ_k and α_k. The first parameter determines the step length before projection the second one multiplies the direction $p_k = (\hat{x}_k - x_k)$, toward the projected point $\hat{x}_k = P_\chi x_k$. In these tests $\gamma_k = 1$ while α_k is calculated adopting backtracking, testing the value of α_k against Armijo conditions. The stopping criterion is

$$\|x_{k+1} - x_k\| \leq \delta \vee \|\nabla f(k_k)\| \leq \tau \tag{8}$$

In the experiments δ and τ are both fixed at 10^{-8}. As initial guess we have adopted $x_0 = (5, ..., 5)^T \in \mathbb{R}^n$. The usual value $c = 10^{-4}$ was used to apply the Armijo condition. In addition, the backtracking iterations were configured as follows. The value of $\alpha_k^{(0)} = 1$ is updated in accordance with the rule $\alpha_k^{(j+1)} = \rho\alpha_k^{(j)}$, with a maximum of 250 backtracking iterations. Different ρ values are tested. In Table I the parameter setting is shown.

TABLE I
Configuration of parameters

maximal number of iteration	10^4
τ: tolerance of $\|\nabla f(k_k)\|$	10^{-8}
δ: tolerance of $\|x_{k+1} - x_k\|$	10^{-8}
maximal backtracks	250
ρ: contraction factor	$0.5, 0.8, 0.9$
c1	10^{-4}

B. *Results*

The results were obtained in accordance with the parameter configuration mentioned above. The experiments focused on studying how the parameter ρ in the backtracking algorithm affects the behavior of the method. The results are shown in Table II.

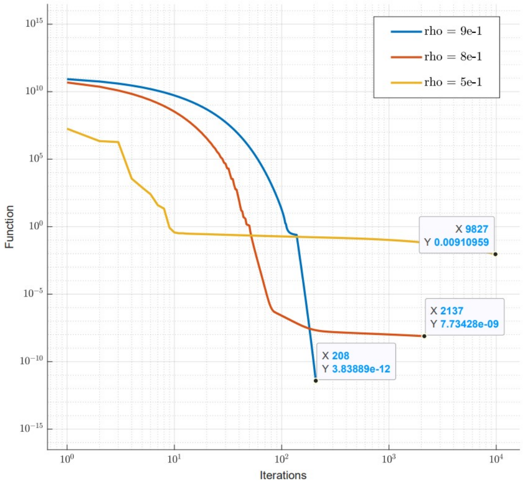

Fig. 1. Function value at each iteration for $n = 10^5$ of all the considered ρ

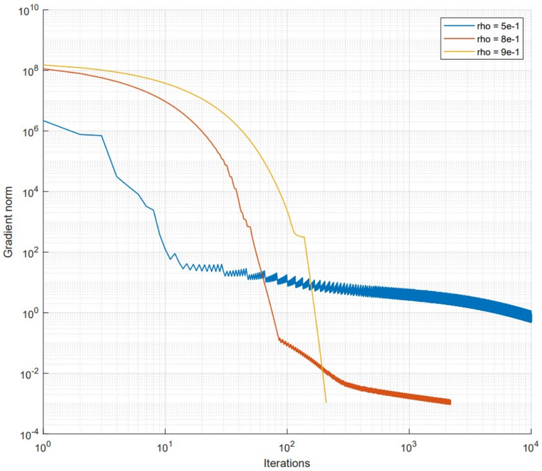

Fig. 2. Gradient norm value at each iteration for $n = 10^5$ of all the ρ

Interestingly, the contraction factor ρ plays a crucial role in the performance of the algorithm. Its impact is evident by considering the trends in $\mathbb{R}^{10^5}$. With ρ equal to 0.5, the function goes down to a certain point, as can be seen in Fig.1, and then the rate of descent decreases significantly. This trend can also be observed in the graph of the gradient norm Fig.2 where it is also possible to see an oscillating behavior, this oscillation can be explained as if the solution started to bounce along the walls of the function without being able to find the step length needed to go deeper.

TABLE II
RESULTS

$\rho = 0.5$				
n	k	$f(x_k)$	$\lVert\nabla f(x_k)\rVert$	elapsed time (s)
10^3	2559	$5.463{\cdot}10^{-12}$	$1.021{\cdot}10^{-5}$	0.481
10^4	10^4	$2.406{\cdot}10^{-8}$	$9.416{\cdot}10^{-4}$	20.68
10^5	10^4	0.0091	0.561	176.5
$\rho = 0.8$				
10^3	1646	$4.641{\cdot}10^{-12}$	$9.939{\cdot}10^{-6}$	0.446
10^4	6411	$2.322{\cdot}10^{-10}$	$9.254{\cdot}10^{-5}$	30.35
10^5	2494	$7.734{\cdot}10^{-9}$	0.0011	85.4752
$\rho = 0.9$				
10^3	217	$2.294{\cdot}10^{-14}$	$9.573{\cdot}10^{-6}$	0.0757
10^4	203	$2.498{\cdot}10^{-13}$	$9.985{\cdot}10^{-5}$	1.529
10^5	208	$3.839{\cdot}10^{-12}$	0.0010	10.245

With ρ equal to 0.8, the number of internal iterations in the backtracking algorithm increases as more alpha candidates are tested before the Armijo condition is satisfied. ρ equal to 0.9 has a surprising behavior, reducing alpha by 10% at each backtracking iteration allows the algorithm to find an optimal solution with a very fast gradient descent without oscillations. This result comes at the cost of a substantial increase in iterations of the backtracking algorithm. Table III shows the average number of backtracking iterations for the three tested values of ρ. In Fig.3 it is also possible to observe the behavior of the number of backtraking iterations required, before the armijo condition is satisfied, as the external iterations progress.

TABLE III
MEAN OF BACKTRACKING ITERATIONS

n	bt-iteration $\rho = 0.5$	bt-iteration $\rho = 0.8$	bt-iteration $\rho = 0.9$
10^3	9.95	30.68	56.06
10^4	12.99	41.13	69.04
10^5	16.52	51.00	81.17

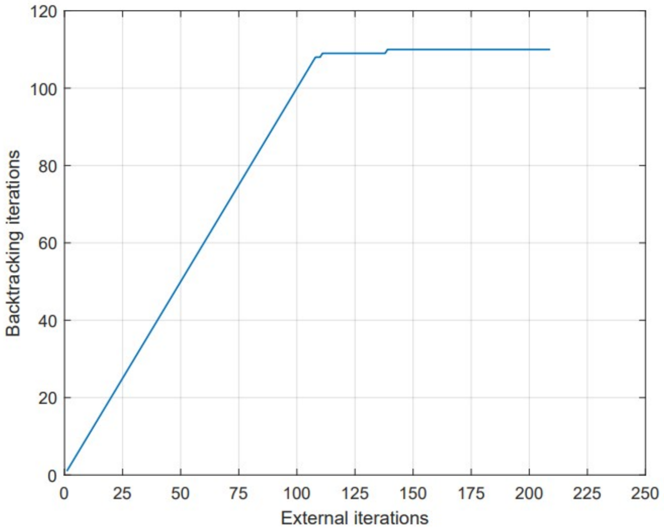

Fig. 3. Backtracking iteration for $n = 10^5$ and ρ = 0.9

For constraints (5), (6) and (7) we decided to use ρ equal to 0.5, all other parameters were left unchanged. Among the three previously tested contraction factors it is the parameter that works best in this situation. One possible explanation is related to the fact that the three new constraints exclude the global solution from the feasible set by imposing a solution that does not require deep search. As can be seen from the Table IV-C in a few iterations the algorithm finds the best solution within the feasible set in all the dimensions tested.

TABLE IV
RESULTS OBTAINED WITH THE CONSTRAINTS (5) (6) AND (7)
ρ = 0.5
x^* = BEST FEASIBLE SOLUTION, $\hat{x}$ = SOLUTION FOUND

constraint (5)			
n	k	$\|x^* - \hat{x}\|_2$	elapsed time (s)
10^3	2	0	0.0016
10^4	2	0	0.0030
10^5	2	0	0.0158
constraint (6)			
n	k	$\|x^* - \hat{x}\|_2$	elapsed time (s)
10^3	3	0	0.0013
10^4	3	0	0.0031
10^5	3	0	0.0187
constraint (7)			
n	k	$\|x^* - \hat{x}\|_2$	elapsed time (s)
10^3	3	0	0.0014
10^4	3	0	0.0040
10^5	3	0	0.0198

IV. Numerical Differentiation

In this section we will address solving the problem expressed in equation (1) using both forward and central difference methods to approximate the gradient, and compare the results with the ones obtained with the exact gradient. For both central and forward differences we used $h = 10^{-l}\,\|x\|_2$ for $l \in [2, 4, 6, 8, 10, 12]$. To have comparable results, we decided to adopt only $\rho = 0.9$. The different results are presented with respect to the space in which the minimization has been performed.

A. *Forward differences*

In figure 4 it is possible to observe the number of iterations required to satisfy the stop criteria as the parameter h changes in $\mathbb{R}^{10^3}$ with Forward differences. It is evident how the trend is similar to the behavior of the relative error between the actual gradient and the approximated one shown in figure 5. There is a descent trend inversion when $h = 10^{-10}$. Beyond this value the error begins to rise. When reducing the value of h we approximate better the gradient. Unfortunately when h is too small the numerical error becomes significant enough to destabilize the result. The results of the various descents are summarized in Table V. Behaviour varies depending on the size of the space. R3 seems to be the dimension most sensitive to numerical error, having a slight worsening with $h = 10^{-12}$. As the size increases, the algorithm needs the gradient approximation to be as correct as possible in order to avoid reaching the maximum number of iterations or in the case of $\mathbb{R}^{10^5}$ stopping prematurely due to steps being too close together. $h = 10^{-10}$ seems on average to be the best parameter. The closer we get to a correct approximation of the gradient, the more the algorithm behaves as in the case of using the exact gradient.

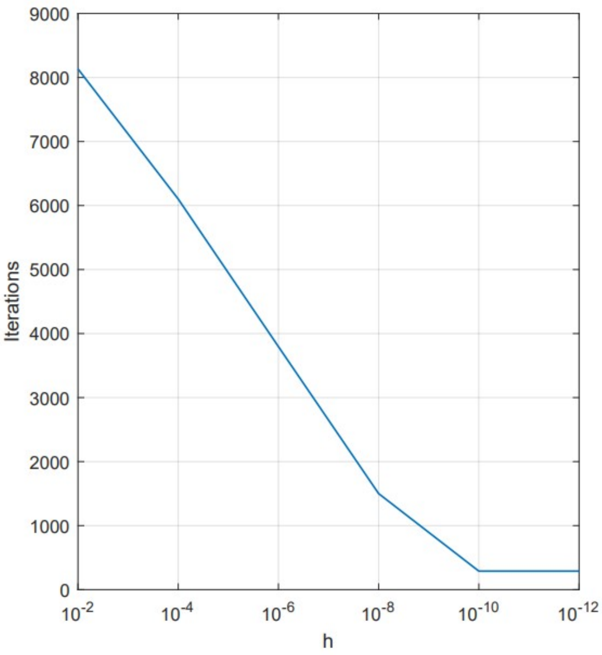

Fig. 4. Number of iterations as h changes, $n = 10^3$ and ρ = 0.9

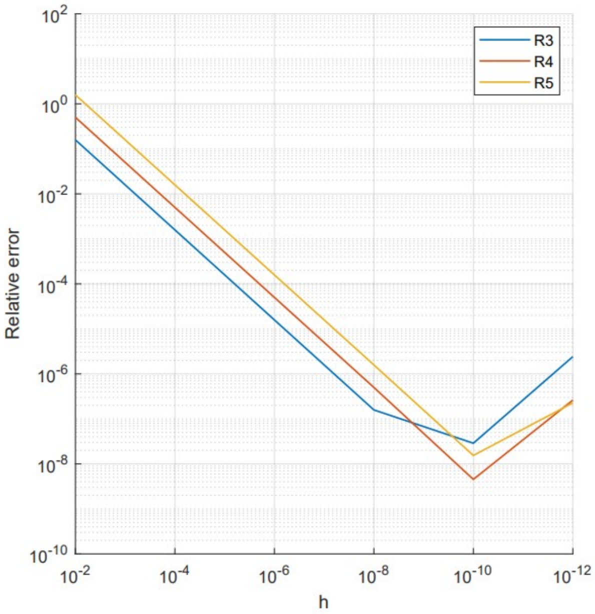

Fig. 5. Relative error in all test spaces R and all suggested values of h

TABLE V
RESULTS FORWARD DIFFERENCES

$\mathbb{R}^{10^3}$						
h	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
k	8133	6101	3796	1500	289	289
$\|\nabla f(x_k)\|$	$5.430 \cdot 10^{-5}$	$1.040 \cdot 10^{-5}$	$1.033 \cdot 10^{-5}$	$9.573 \cdot 10^{-6}$	$9.573 \cdot 10^{-6}$	$9.577 \cdot 10^{-6}$
$f(x_k)$	$1.178 \cdot 10^{-12}$	$1.469 \cdot 10^{-12}$	$1.316 \cdot 10^{-12}$	$2.334 \cdot 10^{-14}$	$2.294 \cdot 10^{-14}$	$2.295 \cdot 10^{-14}$
$\mathbb{R}^{10^4}$						
h	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
k	10000	10000	893	203	203	203
$\|\nabla f(x_k)\|$	32.348	0.00527	$9.538 \cdot 10^{-5}$	$9.985 \cdot 10^{-5}$	$9.985 \cdot 10^{-5}$	$9.983 \cdot 10^{-5}$
$f(x_k)$	0.039	$9.260 \cdot 10^{-9}$	$6.400 \cdot 10^{-11}$	$2.650 \cdot 10^{-13}$	$2.498 \cdot 10^{-13}$	$2.497 \cdot 10^{-13}$
$\mathbb{R}^{10^5}$						
h	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
k	1	181	777	209	209	209
$\|\nabla f(x_k)\|$	$1.500 \cdot 10^{8}$	5.158	0.00104	0.00104	0.00104	0.00104
$f(x_k)$	$8.437 \cdot 10^{10}$	$1.275 \cdot 10^{-04}$	$1.737 \cdot 10^{-09}$	$3.012 \cdot 10^{-12}$	$2.748 \cdot 10^{-12}$	$2.748 \cdot 10^{-12}$

TABLE VI
RESULTS CENTRAL DIFFERENCES

$\mathbb{R}^{10^3}$						
h	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
k	217	217	217	217	217	217
$\|\nabla f(x_k)\|$	$9.5734 \cdot 10^{-6}$	$9.5734 \cdot 10^{-6}$	$9.5734 \cdot 10^{-6}$	$9.5734 \cdot 10^{-6}$	$9.5734 \cdot 10^{-6}$	$9.579 \cdot 10^{-6}$
$f(x_k)$	$2.294 \cdot 10^{-14}$	$2.294 \cdot 10^{-14}$	$2.294 \cdot 10^{-14}$	$2.294 \cdot 10^{-14}$	$2.294 \cdot 10^{-14}$	$2.296 \cdot 10^{-14}$
$\mathbb{R}^{10^4}$						
h	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
k	203	203	203	203	203	203
$\|\nabla f(x_k)\|$	$9.985 \cdot 10^{-5}$	$9.985 \cdot 10^{-5}$	$9.985 \cdot 10^{-5}$	$9.985 \cdot 10^{-5}$	$9.985 \cdot 10^{-5}$	$9.984 \cdot 10^{-5}$
$f(x_k)$	$2.498 \cdot 10^{-13}$	$2.498 \cdot 10^{-13}$	$2.498 \cdot 10^{-13}$	$2.498 \cdot 10^{-13}$	$2.498 \cdot 10^{-13}$	$2.498 \cdot 10^{-13}$
$\mathbb{R}^{10^5}$						
h	10^{-2}	10^{-4}	10^{-6}	10^{-8}	10^{-10}	10^{-12}
k	209	209	209	209	209	209
$\|\nabla f(x_k)\|$	0.00104	0.00104	0.00104	0.00104	0.00104	0.00104
$f(x_k)$	$2.748 \cdot 10^{-12}$	$2.748 \cdot 10^{-12}$	$2.748 \cdot 10^{-12}$	$2.748 \cdot 10^{-12}$	$2.748 \cdot 10^{-12}$	$2.748 \cdot 10^{-12}$

B. *Central difference*

The results obtained for each of the problem dimensions can be seen in Table VI. It is clear that the error is much smaller when using the central difference approach compared to the forward difference method. When testing the descent results, it was found that the central difference method avoids premature stops in $\mathbb{R}^{10^5}$since it is able to provide more accurate approximations. The central difference approximation error has a theoretical error of the order of $O(h^2)$ on each partial derivative, implying more accurate approximate gradients. All tested parameters of h have a gradient approximation such that the algorithm behaves as in the case of using exact gradient. As can be seen by comparing the two tables V and VI, for h between 10^{-10} and 10^{-12} the two approaches behave in the same way since the error relative to the gradient approximation is almost the same as is evident from the Fig. 6

C. *Other constrains*

Here we show the results obtained with finite differences with respect to the constraints (5), (6), (7). To prevent the algorithm from exceeding the number of iterations, we raised the δ tolerance for the distance between two consecutive steps to $\delta = 10^{-4}$. Choice of δ impacts algorithm convergence and performance. Other tolerance levels may result in max number of iterations. In table VII we show the results obtained with the best parameter $h = 10^{-10}$ for forward differences and $h = 10^{-2}$ for central differences. Fig. 8 and 7 illustrate the effect of changing the value of h on the error of the solution found for constraint (6) using the two different approaches.

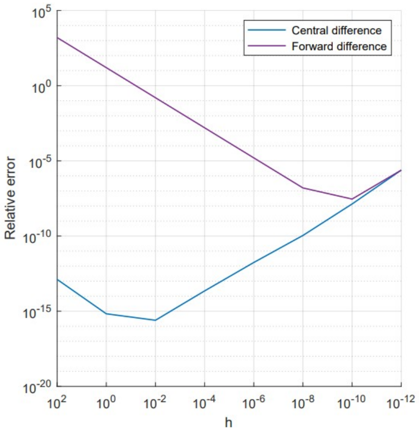

Fig. 6. Relative Error as h changes, $n = 10^3$

TABLE VII
RESULTS OBTAINED WITH THE CONSTRAINTS (5) (6) AND (7)
$\rho = 0.5$, $\delta = 10^{-4}$
x^* = BEST FEASIBLE SOLUTION, $\hat{x}$ = SOLUTION FOUND

Forward difference with $h = 10^{-10}$

constraint (5)			
n	k	$\lVert x^* - \hat{x} \rVert_2$	elapsed time (s)
10^3	2	0	0.022
10^4	2	0	0.041
10^5	2	0	0.072
constraint (6)			
n	k	$\lVert x^* - \hat{x} \rVert_2$	elapsed time (s)
10^3	3	$7.704 \cdot 10^{-8}$	0.0022
10^4	3	$1.270 \cdot 10^{-7}$	0.0096
10^5	3	$7.356 \cdot 10^{-8}$	0.0703
constraint (7)			
n	k	$\lVert x^* - \hat{x} \rVert_2$	elapsed time (s)
10^3	3	$2.816 \cdot 10^{-7}$	0.0015
10^4	3	$5.736 \cdot 10^{-8}$	0.0126
10^5	3	$6.291 \cdot 10^{-8}$	0.0797

Center difference with $h = 10^{-2}$

constraint (5)			
n	k	$\lVert x^* - \hat{x} \rVert_2$	elapsed time (s)
10^3	2	0	0.0075
10^4	2	0	0.052
10^5	2	0	0.086
constraint (6)			
n	k	$\lVert x^* - \hat{x} \rVert_2$	elapsed time (s)
10^3	3	$5.326 \cdot 10^{-15}$	0.0022
10^4	3	$8.881 \cdot 10^{-16}$	0.0121
10^5	3	$8.881 \cdot 10^{-16}$	0.0694
constraint (7)			
n	k	$\lVert x^* - \hat{x} \rVert_2$	elapsed time (s)
10^3	3	$8.881 \cdot 10^{-16}$	0.0015
10^4	3	0	0.0102
10^5	3	$1.776 \cdot 10^{-15}$	0.0702

D. Considerations on the initial point

To further examine the convergence of the algorithm, we conducted an experiment in which the average number of iterations required to terminate the algorithm was computed on one thousand random starting points. The results of this experiment are summarized in Table VIII. This study was conducted to investigate the role that the chosen starting point plays in the surprising efficiency of the algorithm in finding a solution that satisfies the constraints. The results of the experiment provide insight into the relationship between the starting point and the convergence of the algorithm. It has been observed that under constraint (5), the algorithm consistently arrives at the solution regardless of the initial starting point. However, under constraint (7), the effect of the starting point on the number of iterations required to reach the solution appears to be correlated with the size of the search space. This suggests that the choice of starting point may be more influential in determining the efficiency of the algorithm under certain conditions. Further investigation is necessary to fully understand the implications of these findings and to determine the optimal strategies for selecting starting points in different contexts.

V. Conclusion

To address the problem described in equation (1), various approaches were examined. One technique involved evaluating the effect of different contraction factors ρ on the descent rates in the backtracking algorithm. The results showed that using a value of ρ equal to 0.9 resulted in the most effective descents. Another approach involved replacing the exact gradient with approximations using finite differences. The results of the comparison between the two methods show that the central difference method performs better than the forward difference method in terms of accuracy and avoiding premature stops.

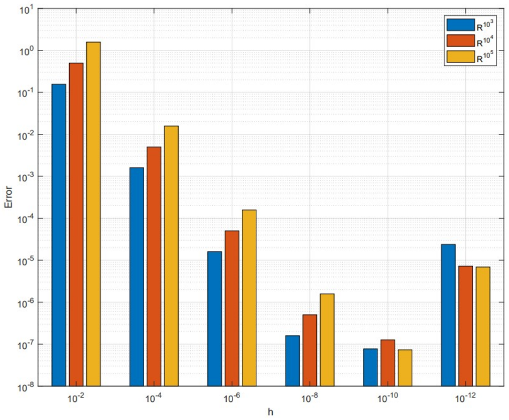

Fig. 7. Error as h changes, Forward differences, constrain (6)

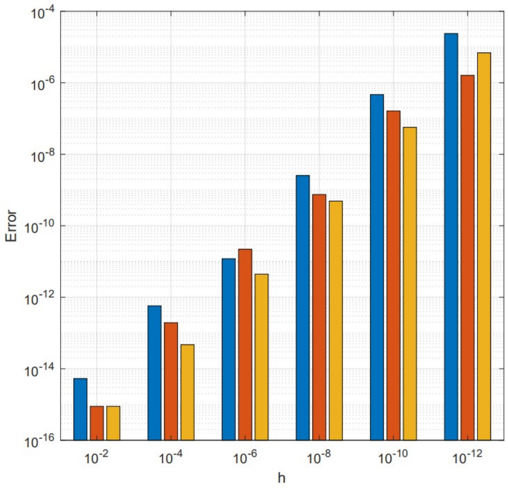

Fig. 8. Error as h changes, Central differences, constrain (6)

TABLE VIII
MEAN OF ITERATIONS ON 1000 SIMULATIONS

n	Constr. (5)	Constr. (6)	Constr. (7)
10^3	2	3	25.98
10^4	2	3	77.70
10^5	2	22.11	821.12