**Contemporary Engineering Sciences, Vol. 11, 2018, no. 16, 771 - 778**
**HIKARI Ltd, www.m-hikari.com**
**https://doi.org/10.12988/ces.2018.8121**

# Linear Search Optimization through the Armijo Rule Method

**German Correa Velez**

Department of Mathematics
Universidad Tecnológica de Pereira
Pereira, Colombia

**Fernando Mesa**

Department of Mathematics and GEDNOL
Universidad Tecnológica de Pereira
Pereira, Colombia

**Pedro Pablo Cárdenas Alzate**

Department of Mathematics and GEDNOL
Universidad Tecnológica de Pereira
Pereira, Colombia



**Abstract**

This paper shows the development of the minimization method in a variable called "Armijo Rule", a method that goes into the selected denomination of line search methods. The intermediate sections show the optimization methods based on direct search and search with descent directions in such a way that the reader has enough information to choose one method or another depending on their optimization problem, whether it is linear optimization with restrictions, non-linear optimization unlimited among others. In addition, a code implemented in Matlab of the Armijo Rule method is presented to find an optimal candidate for a quadratic function.

**Keywords:** Minimization method, Armijo rule, Linear search optimization

# 1 Introduction

In this document the terminology and explanation of Armijo's rule will be systematically displayed, a method used in the optimization and minimization of a variable that is also called "line search"; therefore we will show the development of Armijo's rule implemented in Matlab to obtain efficient and clear results, without infringing computational prohibitive times for minimization in one variable [1].

Optimization covers different areas of knowledge such as: mathematics, statistics, empirical science, computer science, economics. Mathematical optimization (or optimization or mathematical programming) is the selection of the best element (with respect to some criterion) of a set of available elements. In the simplest case, an optimization problem consists of finding the values of the variables of a function which maximize or minimize an objective function; systematically choosing input values (taken from an allowed set) and computing the value of the function. The generalization of optimization theory and techniques for other formulations comprises a large area of applied mathematics. In general, optimization includes the discovery of the "best values" of some objective function given a defined domain, including a variety of different types of objective functions and different types of domains. Pierre de Fermat and Joseph Louis Lagrange found formulas based on calculation to identify optimal values, while Isaac Newton and Carl Friedrich Gauss proposed iterative methods to approximate the optimum. Historically, the term linear programming to refer to certain optimization problems is due to George B. Dantzig, although much of the theory had been introduced by Leonid Kantorovich in 1939. Dantzig published the Simplex Algorithm in 1947 and John von Neumann developed the theory of duality in the same year. On the other hand, the term programming in this context does not refer to computer programming. Rather, the term comes from the use of the program by the United States Army when referring to the training and logistics planning proposal, which was the problem studied by Dantzig at that time [2].

# 2 Iterative methods

The iterative methods used to solve non-linear programming problems differ according to what they evaluate: Hessian, gradients, descent directions or only values of the objective function. While evaluating Hessians (H) and gradients

(G) improves the speed of convergence, such assessments increase the computational complexity (or computational cost) of each iteration. In some cases, the computational complexity may be excessively high.

An important criterion for optimizers is just the number of function evaluations required, as this is often a great computational effort in itself, usually much more effort than the optimizer itself, since it mostly has to operate on $N$ variables; the derivatives provide detailed information for the optimizers, but are even more costly to calculate, for example, approaching the gradient takes at least $N+1$ function evaluations. For the approximation of the second derivatives (grouped in the Hessian matrix) the number of function evaluations is of order $N^2$. Newton's method requires the second order derivatives, therefore for each iteration the number of function calls is of order $N^2$, but for the optimizer a simpler pure gradient is of order $N$. However, the gradient optimizers they usually need more iterations than Newton's algorithm. Being better with respect to the number of function calls depends on the problem itself [3]. Below are a series of optimization methods:

- Methods that evaluate Hessians (or approximate Hessians, using finite differences).
- Newton's method.
- Quadratic sequential programming: A Newton method based on restricted small-medium scale problems. Some versions can handle large problems.
- Methods that evaluate gradients or approximate gradients using finite differences (or even subgradients).
- Quasi-Newton methods: Iterative methods for medium-large problems (example $N < 1000$).
- Conjugated gradient methods: Iterative methods for large problems. (In theory, these methods end in a finite number of steps with quadratic objective functions, but this finite termination is not observed in practice in computers of finite precision.)
- Inner point methods: This is a great class of methods for restricted optimization. Some interior point methods use only information from the subgradient, and others require the evaluation of the Hessians.
- Descent of the gradient (alternatively, pronounced descent or sharp rise): a slow method of theoretical and historical interest, which has been renewed to find approximate solutions to enormous problems [4].

- Subgradient method: An iterative method for large Lipschitz functions locally using generalized gradients.
- Methods that evaluate only values of functions: If a problem is continuously differentiable, then the gradients can be approximated using finite differences, in which case a gradient-based method can be used.
- Interpolation methods Pattern search methods, which have better convergence properties than the Nelder-Mead heuristic.

# 3 Linear search or line search

In optimization (unrestricted), the tracking line search strategy is used as part of a line search method, to calculate how far one should move along a given search direction. It is an advanced strategy with respect to the classic Armijo method. It is a search method along a coordinate axis in which the search must be carried out in both directions of the axis. You must also take into account the fact that an address can be assigned so that $\alpha = 0$ represents a local minimum point of the function $g(\alpha) = F(xk + \alpha dk)$, for which you may not be able to find positive or negative values for $\alpha$ close to 0 for which $g(\alpha) < g(0)$. If want to use any derivative, the "finished" numerical procedures must define can discriminate the situation. The presented model is an Armijo type regression scheme, based on the condition of acceptability of the "parabolic" type [5].

## 3.1 Inaccurate Linear Search

In practice, the minimum of the linear search is not determined exactly. In this sense, it is desirable to sacrifice precision in the linear search with the purpose of favoring the general computation time. Remember that the minimum in a local search does not have to be the minimum of the function. The inaccuracy is usually introduced simply by finishing the linear search before it reaches convergence. The exact nature of the inaccuracy depends on:

- The search technique used.
- The stop criterion.

## 3.2 Rule of Armijo

This rule is a little special because it does not declare any term $\alpha$ as too small and in fact it is never extrapolated. It is chosen $0 < m_1 < 1$ and the cases are defined as:

- If $q(t) \leq q(0) + m_1 \alpha q'(0)$ step small enough to have descent), then the step is adequate and it is finished.
- If $q(t) > q(0) + m_1 \alpha q'(0)$, then $\alpha g = \alpha$ (step $\alpha$ is too large).

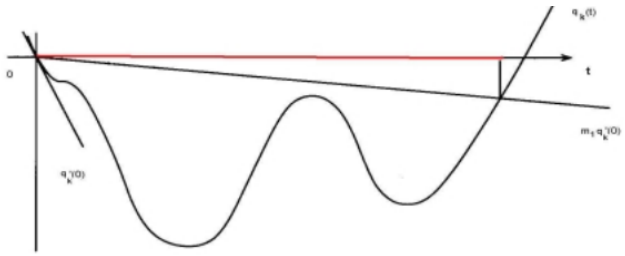


Figure 1: Armijo method.

# 4 Results

The objective of this work is to define a simple linear model of joints used in aeronautics and update this model efficiently. Industrial designers usually resort to semi-empirical models of linear joints to represent the behavior of the joints of a large aeronautical structure. Here, we intend to develop a one-dimensional linear articulation model that is capable of representing the behavior of each articulation of a large structure worldwide, while allowing a new nonlinear local analysis of the joints with greater load.

To solve the numerical difficulties found in some modeling situations, an update strategy based on the constitutive relationship error is proposed. Given that the updating efficiency is significantly affected by the relationships of the rigidities of the different parts of the model, the strategy consists in stiffening some parts of the model in order to control the updating accuracy and the convergence rate. The numerical results of a standard model and a rigid model illustrate the upgrade improvements allowed by the strategy.

Joints are often used in aeronautics because they facilitate assembly and maintenance tasks for manufacturers. However, joint properties such as machining quality, friction or preload are difficult to control during manufacture and lead to differences in the behavior of one fastener to another. These irregularities can cause overload and failure in joint locations during the overall load of the structure.

Therefore, the representation of the actual behavior of a single joint or a set of joints is a real problem in structural mechanics. From the point of view of computational mechanics, joints create a dilemma. When it comes to large structures such as aircraft, the joints are too small and too numerous for each to be modeled with a detailed 3D geometry: an Airbus aircraft uses more than a million bolts and several million rivets. However, you can not perform a realistic simulation without taking them into account. Due to these computational limits, industrial design is generally carried out using a two-scale method. First, simulations are performed using a linear representation of the overall structural level. Linear modeling consists of frame or plate elements (representing the structural parts of the airplane) connected by several types of springs (representing the joints). Most of the joint representations are based on semi-empirical models (Huth, 1986, Tate and Rosenfeld, 1946). This first simulation provides an estimate of the distribution of the joint loads in the structure. Finally, the most loaded joints are identified and new local analyzes are performed with non-linear 3D models to verify the damage criteria. In general, uncertain representation and sensitivity analysis are also carried out. At the global level, an important effort must be made in the quality of the estimated distribution of the joint loads. In this document, only the first part of the design is considered, that is, the estimation of a reliable distribution of loads that employs linear modeling.

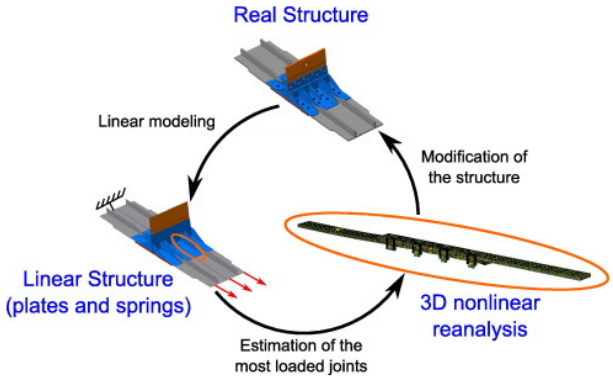


Figure 2: Application of the Armijo method.

The objective of model updating methods is to minimize the distance between the behavior of the real structure and that of a simulated structure by modifying the numerical model. Now, the gradient expression can be used to find a minimum of the constitutive relationship error using a minimization algorithm. A gradient descent is performed and optimized with a line search strategy based on Armijo's rule (Armijo, 1966). In each step k of the descent of the gradient, the optimization consists of finding $\alpha$.

**Example**.

Let $x^2$ be a known function and we will choose two border points of the function (-3) and (5) we will call them $A$ and $B$. Now applying

$$\alpha = A + \frac{(B - A)}{2}$$

with $\alpha = 1$. Therefore, we take the values $\lambda$ and $\mu$, i.e.,

$$\begin{aligned}
\mu &= \alpha + \epsilon \\
\lambda &= \alpha - \epsilon \\
\mu &= 1 + 0.05 = 1.05 \\
\lambda &= 1 - 0.05 = 0.95
\end{aligned}$$

Now, we evaluate $\lambda$ and $\mu$ in the function and we obtain $\mu = 1.1025$ and $\lambda = 0.9025$. As the value of $\mu$ is greater, we take $\mu$ as the new $B$ and iterate again. So we reduce our border points to almost half, giving the method faster. So, we have $A = -3$ and $B = 1.05$ and therefore $\alpha = -0.975$ and we get $\mu = -0.925$, $\lambda = -1.025$. We evaluate the values again in the function and we obtain $\mu = 0.8556$, $\lambda = 1.0506$ and so on.

| | | | |
|---|---|---|---|
| -3 | -1.0250 | -0.9250 | 1.0500 |
| -1.0250 | -0.0375 | 0.0625 | 1.0500 |
| -1.0250 | -0.5313 | -0.4313 | 0.0625 |
| -0.5313 | -0.2844 | -0.1844 | 0.0625 |
| -0.2844 | -0.1609 | -0.0609 | 0.0625 |
| -0.1609 | -0.0992 | 7.8125e-04 | 0.0625 |
| -0.0992 | -0.0684 | 0.0316 | 0.0625 |

Table 1: Results of the method.

# 5 Conclusion

It is vitally important to use software such as Matlab that facilitates the tasks of numerical methods, since doing them manually requires much more time. Efficiently apply line search optimization methods in a variable to determine the minimum point or optimize the function. Highlight definitions and the wide world of applications that optimize without restrictions whether in one variable or multiple variables, in different areas of knowledge. The application

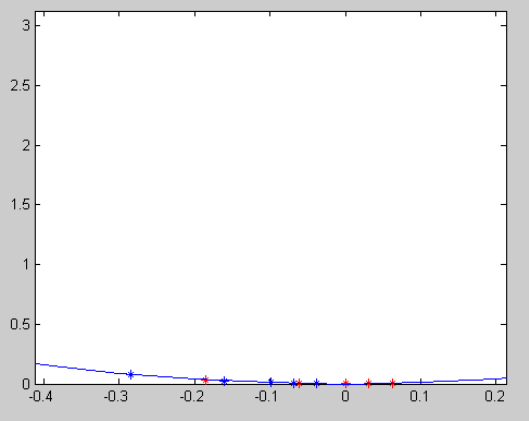


Figure 3: Application of the Armijo method.

of the code in Matlab helps us to be able to graphically see exactly the intercept points generated by Armijo's rule method, thus having a clear idea of the approach of the step, to reach the optimum of the function.

**Acknowledgements.** We would like to thank the referee for his valuable suggestions that improved the presentation of this paper and our gratitude to the Department of Mathematics of the Universidad Tecnológica de Pereira (Colombia) and the group GEDNOL.

# References

[1] A. Quarteroni, R. Sacco, F. Saleri, *Numerical Mathematics,* Springer, 2000.

[2] F. Bonnans, J.C. Gilbert, C. Lemarechal, C. Sagastizabal, *Optimisation Numérique: Aspects Théoriques et Practiques*, SMAI-Springer Verlag, 1997.

[3] D. Luenberger, *Programación Lineal y no Lineal*, Ed. 2, Addison-Wesley, 1989.

[4] J. Mathews and K. Fink, *Métodos Numéricos con Matlab*, Pearson, Prentice-Hall, 2007.

[5] L. Ridgway, *Numerical Methods*, Princeton University Press, 2011.

**Received: February 12, 2018; Published: April 9, 2018**