Solution of bound constrained nonlinear least squares problems with application to backcalculation of asphalt pavements

Lia B. G. Furtado [1], Evandro Parente Jr.[1], Elias S. Barroso[1], Lucas F. A. L. Babadopulos[1], Juceline B. S. Bastos[2]

[1]*Departamento de Engenharia Estrutural e Construção Civil, Universidade Federal do Ceará*
Campus do Pici, Bloco 733, 60440-900, Fortaleza, Ceará, Brasil
liagomes@alu.ufc.br, evandro@ufc.br, elias.barroso@ufc.br, babadopulos@ufc.br
[2]*Instituto Federal de Educação, Ciência e Tecnologia do Ceará*
Av. Treze de Maio, 2081, 60040-531, Fortaleza, Ceará, Brasil
juceline.santos@ifce.edu.br

Abstract. Backcalculation is a procedure used to estimate the material properties of pavement layers from results of non-destructive tests, as the Falling Weight Deflectometer. It is important to assess the quality of a pavement construction and/or to monitor its condition during its lifespan. The Finite Element Method can be used to evaluate pavement deflections, provided that the loading and the properties of each layer are known. Assuming linear elastic behavior and known Poisson's ratios, the backcalculation procedure consists in the determination of the elastic moduli that minimize the differences between the simulated and measured deflections. Thus, pavement backcalculation corresponds to the solution of a Nonlinear Least Squares problem, where the unknown parameters (elastic moduli) are strictly positive. This paper presents a simply approach to include bound constraints in the Gauss-Newton and the Levenberg–Marquardt methods to ensure convergence to physically meaningful solutions. The accuracy, robustness, and computational efficiency of the modified algorithms are compared in the backcalculation of asphalt pavements.

Keywords: Nonlinear Least Squares, Bound Constraints, Backcalculation, Asphalt Pavements.

1 Introduction

Experimental data can be represented by different models that depend on a set of parameters. Regression procedures consist in determine these parameters in order to fit the data with the model. The regression can be obtained by the minimization of the sum of squares of residues, which consists of the Nonlinear Least Squares (NLS) problem. Several methods can be used to solve this problem (Nocedal and Wright [1], Madsen et al. [2]).

An important NLS problem in engineering is pavement backcalculation, which consists of the determination of each pavement layer's elastic modulus. These moduli are the parameters of the finite element elastic model, that is used to predict the deflections of the pavement surface at defined radial distances of the load application point. This load is the same load applied in the Falling Weight Deflectometer (FWD) test, which measures deflections also on the pavement surface (Huang [3]). The difference between each simulated and measured deflection is the residue used in NLS function.

Although the model parameters have a clear physical meaning, optimization is a mathematical process that does not consider this aspect. Therefore, unrealistic results can be obtained in unconstrained minimization, as long as the objective function presents a local minimum. Since the model parameters should be in a physically acceptable range, it is necessary to include bound constraints in the solution of NLS problems. Furthermore, considering bounds on problem variables can improve the efficiency of optimization algorithms even if absurd results do not need to be avoided (Gill et al. [4]).

Different alternatives can be used to create algorithms to solve bound-constrained optimization problems, such as penalty, augmented Lagrangian, transformations, active set, and projection approaches (Gill et al. [4], Arora [5]). Due to its simplicity, a projection approach is adopted in this work to include bound constraints in the Gauss-Newton and Levenberg–Marquardt methods. The accuracy and robustness of the modified algorithms are

compared using a numerical example.

2 Numerical Methods for Nonlinear Least Squares Problems

The classical form of the Nonlinear Least Squares Problem (NLS) (Nocedal and Wright [1], Madsen et al. [2]) can be written as:

$$\min_{\mathbf{x}\in\mathbb{R}^n} f(\mathbf{x}) = \frac{1}{2}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2 = \frac{1}{2}\sum_{i=1}^{m} r_i^2 = \frac{1}{2}\mathbf{r}^T\mathbf{r}, \quad n \leq m, \quad \mathbf{x} = [x_1, ..., x_n]^T, \tag{1}$$

where y_i is the measured value of each point i, $\hat{y}_i$ is the corresponding simulated value that depends on the vector of unknown parameters ($\mathbf{x}$), and $\mathbf{r}$ is the residual vector.

This problem can be solved using different optimization algorithms. Due to their accuracy and efficiency, the Gauss-Newton and the Levenberg–Marquardt methods are applied in this work.

2.1 Gauss-Newton

The Gauss-Newton (GN) method is obtained by the application of the Newton method to the NLS problem. The Newton method is based on a quadratic approximation of the function to be minimized:

$$f(\mathbf{x}_{k+1}) \approx f(\mathbf{x}_k) + \mathbf{d}_k^T\mathbf{g}_k + \frac{1}{2}\mathbf{d}_k^T\mathbf{H}_k\,\mathbf{d}_k, \tag{2}$$

where k is the iteration number, $\mathbf{d}$ is the search direction, $\mathbf{g} = \nabla f(\mathbf{x})$ is the gradient, and $\mathbf{H} = \nabla^2 f(\mathbf{x})$ is the Hessian matrix. The minimization the approximate quadratic function yields:

$$\mathbf{g}_{k+1} \approx \mathbf{g}_k + \mathbf{H}_k\,\mathbf{d}_k = \mathbf{0}. \tag{3}$$

Thus, the search direction ($\mathbf{d}$) at each iteration k is evaluated solving the linear system:

$$\mathbf{H}_k\,\mathbf{d}_k = -\mathbf{g}_k. \tag{4}$$

If the Hessian is positive-definite then it can be easily shown that $\mathbf{d}_k$ is a descent direction ($\mathbf{d}_k^T\,\mathbf{g}_k < 0$) and the function value is reduced as we move in this direction.

The gradient and Hessian of the Sum of Squared Errors (SSE) function defined in eq. (1) are given by

$$\mathbf{g} = \mathbf{J}^T\mathbf{r}, \qquad \mathbf{H} = \mathbf{J}^T\mathbf{J} + \sum_{i=1}^{m} r_i\,\nabla^2 r_i, \tag{5}$$

where $\mathbf{J}$ is the Jacobian matrix:

$$\mathbf{J} = [J_{ij}] = \left[\frac{\partial r_i}{\partial x_j}\right] = \left[\frac{\partial \hat{y}_i}{\partial x_j}\right]. \tag{6}$$

The Gauss-Newton method is obtained neglecting the second term of the Hessian defined in eq. (5):

$$\mathbf{H} \approx \mathbf{J}^T\mathbf{J}. \tag{7}$$

Therefore, the search direction at each iteration is computed solving the linear system:

$$(\mathbf{J}_k^T\mathbf{J}_k)\,\mathbf{d}_k = -\mathbf{J}_k^T\mathbf{r}_k. \tag{8}$$

After the computation of the search direction, the new estimate of the parameter vector is computed as

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha\,\mathbf{d}_k, \tag{9}$$

where α is the step size along the search direction. The classical Gauss-Newton method considers $\alpha = 1$.

In this work, a backtracking line search is adopted in order to improve the algorithm robustness and efficiency. This algorithm starts with a unit step size ($\alpha_0 = 1$) and check the descent condition

$$f(\mathbf{x}_k + \alpha_l\,\mathbf{d}_k) < f(\mathbf{x}_k) + \alpha_l\,(\beta\,\mathbf{g}_k^T\mathbf{d}_k), \tag{10}$$

where l is the line search iteration and $\beta \in (0, 1)$ (Nocedal and Wright [1], Arora [5]). If this condition is satisfied, the step size is accepted, otherwise it is reduced:

$$\alpha_{l+1} = \eta\, \alpha_l \tag{11}$$

with $\eta \in (0, 1)$. The process is repeated until a sufficient function decrease is obtained. It is important to note that this approach allows for a unit step size at the solution, which is a condition for quadratic convergence of the Newton method. Furthermore, since the objective function is reduced at each iteration, the method will eventually converge to local minimum.

The iterative process described in eq. (9) can be stopped when:

$$\text{NRMSE} = \sqrt{\frac{1}{m}\sum_{i=1}^{m}\left(\frac{\hat{y}_i - y_i}{y_i}\right)^2} < tol_1 \quad \text{or} \quad \|\mathbf{g}\| < tol_2, \tag{12}$$

where NRMSE is the Normalized Root Mean Square Error, while and tol_1 and tol_2 are the convergence tolerances.

Using exact gradient and Hessian, the Newton method presents local quadratic convergence. On the other hand, failure can occur if the Hessian is singular. Furthermore, when the initial point $\mathbf{x}_0$ is far from the solution, the quadratic approximation may be not accurate, generating poor search directions, with slow improvement or even divergence.

The convergence rate of the Gauss-Newton method depends on how close eq. (7) approximates the true Hessian. Quadratic convergence can be obtained when the second term of the Hessian matrix defined in eq. (5) is small close to the solution, which can occur when the residual is very small ($r_i \approx 0$) or the Jacobian is affine with respect to the parameters ($\nabla^2 r_i = 0$) (Nocedal and Wright [1]).

2.2 Levenberg–Marquardt

The Levenberg–Marquardt (LM) method (Nocedal and Wright [1], Madsen et al. [2]) was proposed as a more robust alternative to the Gauss-Newton method. It is based on a modified form of eq. (8):

$$(\mathbf{J}_k^T \mathbf{J}_k + \lambda_k\, \mathbf{I})\, \mathbf{d}_k = -\mathbf{J}_k^T \mathbf{r}_k, \tag{13}$$

where the damping parameter ($\lambda > 0$) ensures that the coefficient matrix is positive definite, generating a descent search direction $\mathbf{d}_k$ at all iterations.

It is important to note that, for large damping factors, $\mathbf{d}_k \approx -\mathbf{g}_k/\lambda_k$, which is a short step in the steepest descent direction. This is a good option if the current iterate $\mathbf{x}_k$ is far from the solution. On the other hand, eq. (13) reduces to eq. (8) when λ_k is very small, generating the same search direction than the Gauss-Newton method, which is very good in the final iterations when $\mathbf{x}_k$ is close to the solution and Gauss-Newton can present quadratic convergence. Since the damping factor controls not only the search direction but also the step size, the Levenberg–Marquardt method is used without line searches. However, the new iterate is accepted only if it decreases the error:

$$\begin{cases} \mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{d}_k, & \text{if } f(\mathbf{x}_k + \mathbf{d}_k) < f(\mathbf{x}_k) \\ \mathbf{x}_{k+1} = \mathbf{x}_k, & \text{otherwise.} \end{cases} \tag{14}$$

Convergence is checked using eq. (12).

Several schemes have been proposed in the literature to update the damping factor (Madsen et al. [2], Transtrum and Sethna [6], Gavin [7]). The main idea is to begin with a relatively large factor λ. If the step computed by eq. (13) decreases the error, then λ is decreased. Otherwise, λ is increased. In this work, the damping factor is updated as:

$$\begin{cases} \lambda_{k+1} = \lambda_k/\gamma_1, & \text{if } f(\mathbf{x}_k + \mathbf{d}_k) < f(\mathbf{x}_k) \\ \lambda_{k+1} = \lambda_k\, \gamma_2, & \text{otherwise.} \end{cases} \tag{15}$$

where λ_0, γ_1, and γ_2 are positive real numbers.

3 Bound-Constrained NLS

Usually, the unknown model parameters ($\mathbf{x}$) have a clear physical meaning and a range of acceptable values. However, any value of $\mathbf{x}$ can be obtained in the solution of the NLS problem provided it minimizes the objective

function defined in Eq. (1). In many cases, unrealistic parameter values (e.g. negative elastic modulus) lead to errors in model evaluation causing convergence failure.

Thus, the unconstrained algorithms described previously need to be modified to include the bound constraints to ensure that the parameters are in the acceptable range. The bound-constrained NLS problem can be written as

$$\begin{cases} \min\limits_{\mathbf{x}\in\mathbb{R}^n} \; f(\mathbf{x}) = \dfrac{1}{2}\sum\limits_{i=1}^{m} r_i^2 = \dfrac{1}{2}\mathbf{r}^T\mathbf{r}, \;\; n \leq m \\ \text{s. t. } \; l_j \leq p_j \leq u_j, \;\; j = 1, \ldots, n \end{cases} \tag{16}$$

where l_j and u_j, are the lower and upper bounds, respectively. The algorithm used to solve this problem is an adaptation of a projection method for the solution of general optimization problems with bound constraints (Schwartz and Polak [8]).

The basic idea of projection methods is to identify the set of active variables, corresponding to the variables that are at their bounds at the solution (Schwartz and Polak [8], Bertsekas [9]). Once identified, these variables are kept fixed and the optimization algorithm changes only the free variables. One key element of this approach is the projection operator (P):

$$P\left(x_j\right) = \begin{cases} l_j, & \text{if } \; x_j \leq l_j \\ x_j, & \text{if } \; l_j < x_j < u_j \\ u_j, & \text{if } \; x_j \geq u_j \end{cases} \tag{17}$$

The projection operator is applied component-wise and guarantees that the variables (i.e. the model parameters) are kept within the feasible region ($l_j \leq x_j \leq u_j, \;\; j = 1$ to n) at all iterations.

After projection, the set of variables candidate to be active (A) is composed by the ones close or on its bounds:

$$A(\mathbf{x}) = \left\{ j = 1, \ldots, n \;\middle|\; \begin{array}{lll} l_j \leq x_j \leq l_j + \delta & \text{and} & g_j > 0, \text{ or} \\ u_j - \delta \leq x_j \leq u_j & \text{and} & g_j < 0 \end{array} \right\}, \tag{18}$$

where δ is a parameter defining if the variable is close to its bounds, which is given by

$$\delta = \min\{\varepsilon, \, \|\mathbf{w}(\mathbf{x})\|\}, \tag{19}$$

where $\varepsilon > 0$ and

$$\mathbf{w}(\mathbf{x}) = \mathbf{x} - P(\mathbf{x} - \mathbf{g}) \tag{20}$$

is the projected gradient, thus $\mathbf{w} = \mathbf{0}$ at the solution. It is important to note that only the variables whose steepest descent direction ($-g_j$) points to outside the feasible region are included in the active set. This procedure allows the correct identification of the active set after a finite number of iterations (Schwartz and Polak [8]).

The set of inactive variables (I) is given by:

$$I(\mathbf{x}) = \{j = 1, \ldots, n \mid j \notin A(\mathbf{x})\}. \tag{21}$$

Thus, at each iteration, the set of inactive variables $I_k = I(\mathbf{x}_k)$ is the complement of the set of active variables $A_k = A(\mathbf{x}_k)$.

In order to allow the variables close to bounds to move to the exact bounds, the search direction for active variables ($\mathbf{d}_A$) is the steepest descent direction (Schwartz and Polak [8]):

$$\mathbf{d}_A = -\mathbf{g}_A. \tag{22}$$

On the other hand, the search direction for inactive variables ($\mathbf{d}_I$) can be evaluated using any method for unconstrained NLS problems discussed previously. Furthermore, the modified stopping criteria is given by

$$\text{NRMSE} < tol_1 \quad \text{or} \quad \|\mathbf{g}_I\| \leq tol_2. \tag{23}$$

Thus, the gradient norm considers only the inactive variables.

The descent condition used in Gauss-Newton line search is modified to include the bound constraints (Arora [5], Schwartz and Polak [8]):

$$f(\mathbf{x}_k + \alpha_l \, \mathbf{d}_k) < f(\mathbf{x}_k) + \beta \left[\alpha_l \left(\mathbf{g}_k^T \mathbf{d}_k\right)_{I_k} + \left(\mathbf{g}_k^T \left(\mathbf{x}_{k+1} - \mathbf{x}_k\right)\right)_{A_k}\right]. \tag{24}$$

If the set A_k is empty, the term $\left(\mathbf{g}_k^T \left(\mathbf{x}_{k+1} - \mathbf{x}_k\right)\right)_{A_k}$ vanishes. If the set A_k contains some of the variables that are on their bounds, then $\left(\mathbf{x}_{k+1} - \mathbf{x}_k\right)_{A_k} = 0$ and the foregoing term again vanishes. In both cases, the descent condition of eq. (24) reduces to eq. (10). In other words, we find the step size to minimize f considering only the inactive (i.e. free) variables and keeping the active variables fixed at their bounds. Finally, if the variable x_j is close to its bound, the term $g_{j_k}^T \left(x_{j_{k+1}} - x_{j_k}\right) < 0$, which satisfies the descent condition, as for the free variables. Thus, the step size calculation criterion in eq. (24) allows for the variable to move closer to its bound.

4 Numerical Example

The algorithms to solve bound-constrained NLS problems were implemented in C++ language. A pavement backcalculation example is used to show how bound constraints can be useful to guarantee the convergence of the optimization process to realistic results. Table 1 shows the initial optimization parameters applied in this example.

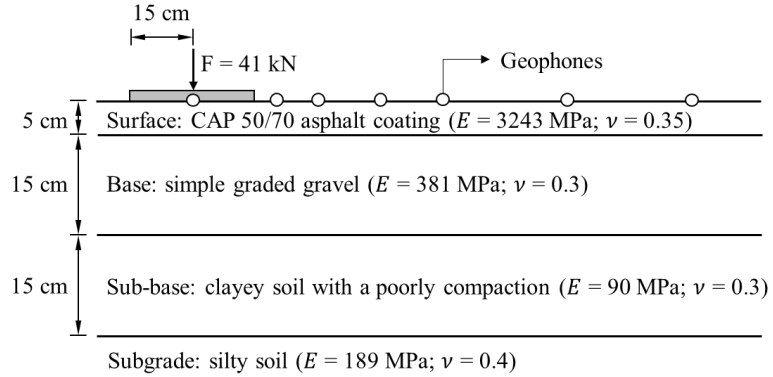
Table 1. Optimization parameters used in the numerical example.

k_{max}	tol_1	tol_2	l_{max}	β	η	λ_0	γ_1	γ_2	ε
200	10^{-8}	10^{-8}	10	10^{-4}	0.5	10^{-2}	10	10	10^{-2}

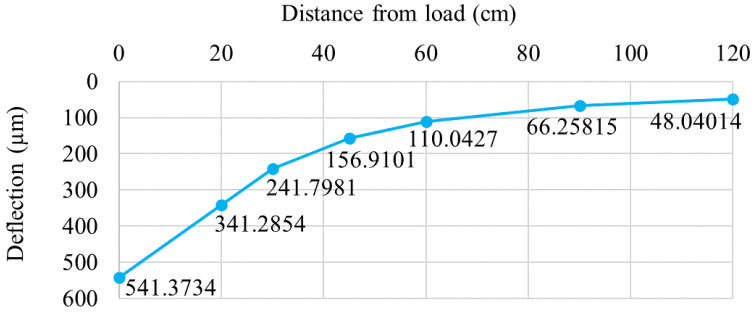
The Falling Weight Deflectometer (FWD) is a non-destructive testing device that applies a load to a circular plate in contact with the pavement surface through a falling mass and measures pavement surface deflections at specified radial distances from the plate center using geophones (Huang [3]). Provided that the pavement geometry and material properties are known, simulated deflections can be obtained by a finite element model (Barroso et al. [10]). Thus, the backcalculated properties correspond to the model parameters $\mathbf{x}$ that minimize the difference between the simulated (y) and the measured ($\hat{y}$) deflections.

Following the standard backcalculation approach, it is assumed that the thickness (h) and Poisson's ratio (ν) of each pavement layer is known and only the modulus of elasticity (E) of each layer needs to be backcalculated: $\mathbf{x} = [E_1, ..., E_n]^T$. Therefore, the number of pavement layers defines the size of the vector $\mathbf{x}$ (n), while the number of geophones in FWD defines the number of data points (m).

In this example, the pavement has 4 different layers, including the subgrade, and the FWD has 7 geophones. The pavement structure was obtained from Barroso et al. [10] and it is represented in Fig. 1a. Figure 1b shows the deflection basin obtained by FEM considering these data.



(a) Pavement structure.



(b) Deflection basin.

Figure 1. Pavement backcalculation problem.

Ten different randomly generated sets of seed moduli (i.e. starting points) were used to test the algorithms. The range of each seed modulus was determined following the recommendations of the Federal Highway Administration Research and Technology (FHWA) (Pierce et al. [11]) for each material type. Table 2 shows the modulus

range of the layers and the seed moduli.

Table 2. Seed moduli set.

Layer	E_{min} (MPa)	E_{max} (MPa)	Elastic seed moduli (MPa)									
			T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
Surface	750	15000	9315	5808	3099	8466	13741	10992	4052	758	7561	12265
Base	70	7000	143	4304	1953	5397	3870	814	2472	6529	6229	3115
Subbase	50	700	682	238	513	267	171	584	415	355	449	56
Subgrade	34	345	305	101	94	201	318	186	223	46	281	157

The seed moduli were tested for Gauss-Newton and Levenberg-Marquardt methods, totaling 20 backcalculation results, as shown in Table 3. Eleven of the twenty tests (55%) obtained the correct moduli: $\mathbf{x} = [3243, 381, 90, 189]^T$ MPa, while six tests presented solutions with negative elastic modulus. Other three tests did not converged in 200 iterations.

Table 3. Backcalculated moduli set.

Layer	Elastic backcalculated moduli (MPa)									
	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
Gauss-Newton										
Surface	**2867**	-	3243	**27198**	3243	3243	3243	**94**	-	3243
Base	**219**	-	381	**-54051**	381	381	381	**-3829**	-	381
Subbase	**-5041**	-	90	**15**	90	90	90	**630**	-	90
Subgrade	**132**	-	189	**506**	189	189	189	**149**	-	189
Levenberg-Marquardt										
Surface	3243	3243	**218**	3243	3243	3243	**-**	**218**	**218**	3243
Base	381	381	**2709**	381	381	381	**-**	**2709**	**2709**	381
Subbase	90	90	**-451**	90	90	90	**-**	**-451**	**-451**	90
Subgrade	189	189	**152**	189	189	189	**-**	**152**	**152**	189

On the other hand, considering a lower bound equal to 1 MPa, just to avoid solutions with negative moduli, all tests obtained the correct solution. This result shows the importance of using bound constraints not only to ensure physically acceptable solutions but also to increase the robustness of NLS algorithms.

Comparing the efficiency of the methods by the number of function evaluations, Gauss-Newton was more efficient in all tests, as shown in Table 4. Due to the large performance difference, the parameters used in Levenberg-Marquardt method were modified to improve its efficiency.

Table 4. Efficiency of NLS methods for $E_{min} = 1$ MPa.

Method	LM parameters			Number of function evaluations (NFE)								
	λ_0	γ_1	γ_2	T_1	T_2	T_3	T_4	T_5	T_6	T_7	T_8	T_9	T_{10}
Gauss-Newton	-			22	31	25	50	23	17	18	30	24	22
Levenberg-Marquardt	0.01	10	10	99	123	97	162	240	93	106	171	141	219
	1	10	10	100	127	90	160	238	91	104	169	139	217
	100	10	10	104	131	99	161	266	104	105	146	134	221
	1	9	11	83	95	69	133	206	66	69	156	159	197
	1	3	2	58	76	60	76	108	47	73	91	91	97

The initial values of parameters λ_0, γ_1 and γ_2 were chosen equal to 10^{-2}, 10 and 10, respectively, which are the values proposed by Marquardt [12] and used in the MATLAB implementation of this method (MathWorks [13]). Testing other values of λ_0, it is possible to state that this parameter does not have a significant influence on the efficiency of this problem, but $\lambda_0 = 1$ presented the best results.

On the other hand, keeping $\lambda_0 = 1$ but changing γ_1 and γ_2 to 9 and 11 respectively, values recommended by Gavin [7] and to 3 and 2 respectively, recommended by Madsen et al. [2], Transtrum and Sethna [6], it is possible to observe a significant improvement in the efficiency of the method. Using $\gamma_1 = 9$ and $\gamma_2 = 11$, NFE decreased by 16% on average, while with $\gamma_1 = 3$ e $\gamma_2 = 2$, the decrease was 44% on average (Table 4). Despite this improvement, the performance of Levenberg-Marquardt remained inferior to the Gauss-Newton method.

5 Conclusion

In this paper, efficient optimization algorithms for NLS problems were implemented and applied to the back-calculation of the elastic moduli of an asphalt pavement. In the case of classic (i.e. unconstrained) NLS algorithms, lack of convergence and convergence to unrealistic solutions occurred depending on the starting point (i.e. seed moduli).

On the other hand, when simple bounds were considered, the correct solution was obtained independent of the starting point. Therefore, the consideration of bound constraints is important not only to ensure convergence to physically acceptable solutions but also to increase the robustness of NLS algorithms.

The algorithms' efficiency was assessed using the number of function evaluations (i.e. finite element analyses) and the Gauss-Newton method was more efficient than the Levenberg-Marquardt method in this example. However, further research needs to be carried out on generalized this conclusion to other pavement structures.

Acknowledgements. The authors gratefully acknowledge the financial support provide by CNPq (Conselho Nacional de Desenvolvimento Científico e Tecnológico), CAPES (Coordenação de Aperfeiçoamento de Pessoal de Nível Superior) and FUNCAP (Fundação Cearense de Apoio ao Desenvolvimento Científico e Tecnológico).

Authorship statement. The authors hereby confirm that they are the sole liable persons responsible for the authorship of this work, and that all material that has been herein included as part of the present paper is either the property (and authorship) of the authors, or has the permission of the owners to be included here.

References

[1] J. Nocedal and S. J. Wright. *Numerical Optimization*. Springer Series in Operations Research and Financial Engineering. Springer, New York, NY, 1999.

[2] K. Madsen, H. Nielsen, and O. Tingleff. *Methods for Non-Linear Least Squares Problems (2nd ed.)*, 2004.

[3] Y. H. Huang. *Pavement analysis and design*. Pearson Prentice Hall, 2004.

[4] P. E. Gill, W. Murray, and M. H. Wright. *Practical Optimization*. Society for Industrial and Applied Mathematics, Philadelphia, PA, 2019.

[5] J. Arora. *Introduction to Optimum Design*. Academic Press, 2 edition, 2014.

[6] M. K. Transtrum and J. P. Sethna. Improvements to the levenberg-marquardt algorithm for nonlinear least-squares minimization, 2012.

[7] H. P. Gavin. *The Levenberg-Marquardt algorithm for nonlinear least squares curve-fitting problems*, 2020.

[8] A. Schwartz and E. Polak. Family of projected descent methods for optimization problems with simple bounds. *Journal of Optimization Theory and Applications*, vol. 92, pp. 1–31, 1997.

[9] D. P. Bertsekas. Projected newton methods for optimization problems with simple constraints. *SIAM Journal on Control and Optimization*, vol. 20, n. 2, pp. 221–246, 1982.

[10] E. S. Barroso, J. L. F. Oliveira, E. Parente Jr., L. F. A. L. Babadopulos, and J. B. S. Bastos. Efficient backcalculation procedure for asphalt pavements using the finite element method. *XLIII CILAMCE*, vol. , 2022.

[11] L. M. Pierce, J. E. Bruinsma, K. D. Smith, M. J. Wade, K. Chatti, and J. M. Vandenbossche. *Using Falling Weight Deflectometer Data with Mechanistic-Empirical Design and Analysis, Volume III: Guidelines for Deflection Testing, Analysis, and Interpretation*. Federal Highway Administration, 2017.

[12] D. W. Marquardt. An algorithm for least-squares estimation of nonlinear parameters. *Journal of the Society for Industrial and Applied Mathematics*, vol. 11, n. 2, pp. 431–441, 1963.

[13] MathWorks. Least-squares (model fitting) algorithms. `https://www.mathworks.com/help/optim/ug/least-squares-model-fitting-algorithms.html#f204`. Accessed: 2023-08-16, 2023.