USING PENALTY FUNCTION METHOD FOR ANALYSIS OF DYNAMIC BEHAVIOUR OF VISCOELASTIC MATERIALS

Marina Griscenko, Janis Viba, Maris Eiduks
Riga Technical University, Latvia
marina.griscenko@rtu.lv, janis.viba@rtu.lv, maris.eiduks@gmail.com

Abstract. The present article investigates the application of the penalty function method in the analysis of behaviour of viscoelastic materials, particularly silicon rubbers, under dynamic loading. To perform the analysis, the mathematical model of the object was chosen, and the Matlab code for the mathematical model was developed. The dynamic force was applied to the given model. To find the most precise element describing the behaviour of elastic material from some set of available alternatives, the penalty method, which replaces a constrained optimization problem by a series of unconstrained problems, was used.

Keywords: elastic materials, penalty function, dynamic load.

Introduction

Silicon rubbers are materials with highly non-linear behaviour patterns. The hypothesis states that the penalty function algorithm for solving non-linear problems with linear lower level problems could be especially useful in elastic material modelling.

The dynamic load conditions analyzed in the given paper are the stress and strain, and it is assumed the silicon rubber material could be described as a viscoelastic material having the properties both of elasticity and viscosity. The mechanical model chosen to describe viscoelastic response under dynamic load is the three-element model (Standard Linear Solid).

Because viscoelastic material modelling under dynamic load resulted into graphs with highly non-linear behaviour, some chaotic behaviour analysis and error estimation were done.

The objective of the paper was to define, whether using the penalty function could be sophisticated problem-solving algorithm for analysis of behaviour of viscoelastic materials under dynamic loading. Matlab program and additional Matlab-based modeling system were used to meet the objective.

Materials and methods

The mechanical model chosen to describe viscoelastic response under dynamic load is the three-element model (Standard Linear Solid), where equations follow the same logics as the Maxwell model, but describe the dynamic load with more precision [1]. The Green strain tensor with random variables was used to define deformation of the viscoelastic material.

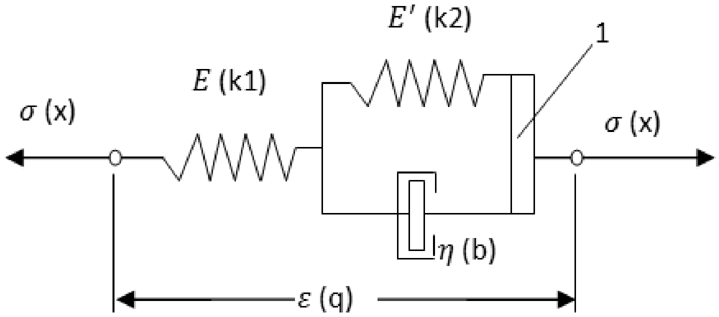

Fig. 1. **Diagram for mathematical model of viscoelastic material:** 1 – applied mass

The three-element model obtained from deformation tensors had to be adjusted for dynamic load programming, so the mass was added to the model. Denominations of variables in the three-element model were used for dynamic load equations and resulting oscillation of the system, substituting variables of load equations as shown in Fig. 1.

The model is described with mathematical expressions, where motion is described by the material coordinate and time t according to Lagrangian description [1].

The model yields for the following expressions:

$$E_1 = \frac{E\left(1+\omega^2\lambda\lambda'\right)}{1+\omega^2\lambda^2} = \frac{E+\omega^2\lambda^2 E'}{1+\omega^2\lambda^2},$$

$$E_2 = \frac{E\left(\lambda'-\lambda\right)\omega}{1+\omega^2\lambda^2} = \frac{\left(E'-E\right)\omega\lambda}{1+\omega^2\lambda^2},$$

$$\tan\delta = \frac{\left(\lambda'-\lambda\right)\omega}{1+\omega^2\lambda\lambda'} = \frac{\left(E'-E\right)\omega\lambda}{1+\omega^2\lambda^2 E'}, \tag{1}$$

where E_1, E_2 – modulus of Lagrangian stress tensor;

$\lambda = \eta \,/\, E$ – relaxation time constant;

η – material coefficient of viscosity, kg·(s·m)$^{-1}$;

ω – natural frequency, equals $2\pi f$ where f is frequency of strain oscillation, Hz;

δ – phase lag between stress and strain.

To state that the model indeed has non-linear behaviour patterns as described in theoretical examples [1], the Lagrangian stress tensors and phase lag between stress and strain were plotted in 3D space, and values of Lagrangian stress tensor E_2 were plotted in 2D space. The results that supported the statement are illustrated in Fig. 2.

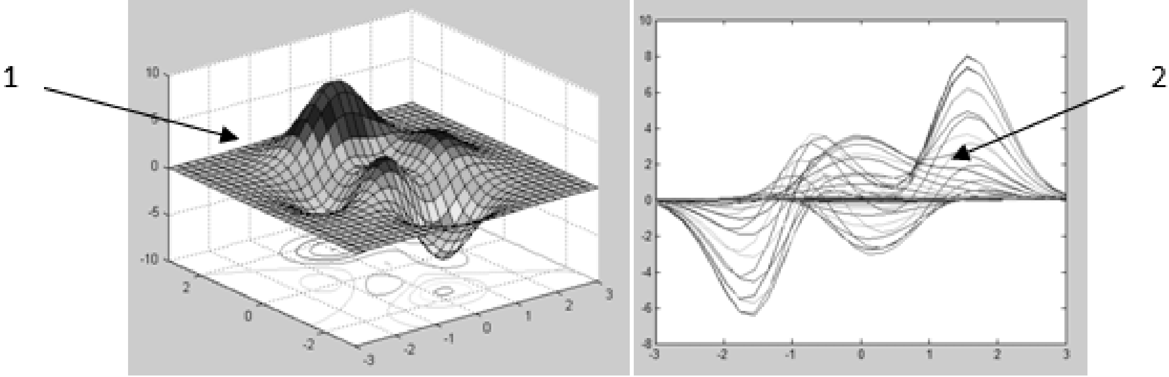

Fig. 2. **Plotted values of Lagrangian stress tensor:** 1 – E_1, E_2, phase lag in 3D; 2 – E_2 in 2D

The dynamic load and resulting oscillation of the system is described with the following non-linear equations [2], where original designations are already substituted with designations used for the mathematical model (2):

$$\begin{cases} m\ddot{\sigma} + E_2\left(\sigma-\varepsilon\right) + \eta\dot{\sigma} = P(t) \\ E_1\varepsilon + \tilde{F}\cdot sign\dot{\varepsilon} + E_2\left(\varepsilon-\sigma\right) = 0 \end{cases}, \tag{2}$$

where $\varepsilon = \varepsilon 0 \cdot \sin\left(t\omega\right)$ – strain;

$\sigma = \sigma 0 \cdot \sin\left(t\omega + \delta\right)$ – stress;

$\tilde{F} = F0 \cdot sign\left(\dot{\sigma}\right)$ – friction force, N.

For the Matlab code, which was created, using combinations of equations (1) and (2), the time period of 30 seconds was used. Then the simple penalty function of quadratic form as shown in equation (3) was designed, using as a penalty function the squared equation from the equation system (2).

$$P_N\left(x\right) = \sum_{i=1}^{m} g_i^2\left(x\right), \tag{3}$$

where $g_i^2\left(x\right)$ – chosen penalty function for oscillation equation.

One iteration was performed for manual penalty function calculation. As expected, the resulting plot did not differ from the original oscillation plot after the first iteration. Please, refer to Fig. 3 for graphical results before and after the first penalty function iteration.

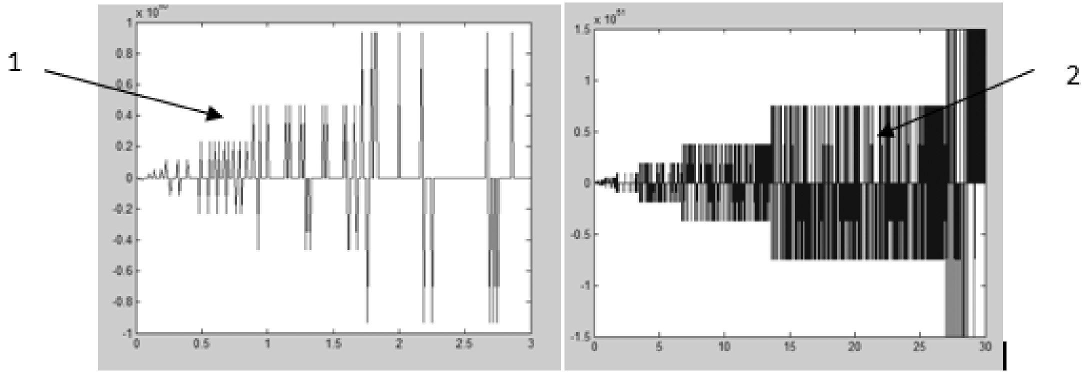

Fig. 3. **Plotted values for oscillation of the system:** 1 – plot for three seconds before and after penalty function application; 2 – plot for 30 seconds before and after penalty function application

Performing a large number of iterations manually would be time consuming and error-prone process, therefore, the penalty function was set with the help of special software CVX. CVX is a Matlab-based modeling system for convex optimization. CVX turns Matlab into a modeling language, allowing constraints and objectives to be specified using standard Matlab expression syntax [3-5].

Results and discussion

After 19 iterations (1 initial value and 18 optimisation steps) the numerical value for the optimisation problem was calculated to be +3.50379e-009 (Optimal value (cvx_optval)). Please, refer to Fig. 4 for illustration of the results. From the plot 2 of Fig. 4 it is seen that the final value found by the penalty method is very close to zero.

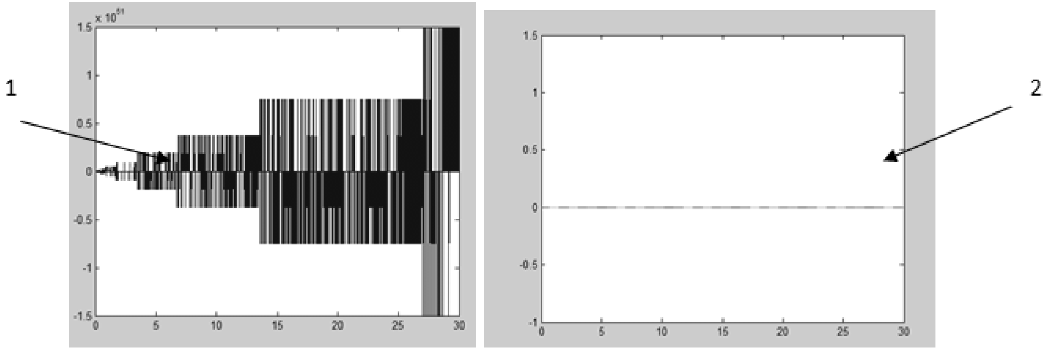

Fig. 4. **Plotted values for oscillation of the system:** 1 – plot for 30 seconds before penalty function application; 2 – plot for 30 seconds after penalty function application

CVX program allows following the value of oscillations after every iteration, so the method is much more convenient than manual calculation.

To avoid exceeding the scope of the paper, the full Matlab code is not quoted. Instead, only primal objective values resulting from the penalty method analysis are listed. Primal objective values for every iteration were calculated as follows: 0.00000000028760, 21.21292000000000, 26.33153000000000, 6.85262500000000, 0.87425990000000, 0.20348780000000, 0.02906845000000, 0.00907735000000, 0.00174014000000, 0.00050240120000, 0.00015596390000, 0.00004775169000, 0.00001845475000, 0.00000655635000, 0.00000155286600, 0.00000049112410, 0.00000008265245, 0.00000002262981, 0.00000000350379.

Standard error of mean for the resulting values was calculated to be 1.73, which is a relatively large value. The reason for the large error was the large value of the second and third iterations, which did not correlate to the values of other 17 iterations.

As the matrix function should correspond to the penalty function, there is high probability of error for the simple penalty function application using manual programming, and matrix calculation for penalty function formation is a challenge. In the example of viscoelastic material behaviour under dynamic load the problem of formation of matrix for the penalty function existed as well, because the penalty function, derived from equation (2) did not correspond to the function matrix, and some adjustments for manual calculation had to be done, which as a result made calculations less precise. Adjustments of penalty functions could lead to errors, if the final penalty function is small relative to the objective function. Starting the optimization process, it is usually not known in advance, whether the penalty function is a large value relative to a given problem and consequently an initial trial may produce a solution point that is not close enough to the feasible region. Such problem was observed during manual penalty function application. On the other hand, large values of the penalty function yielded to errors in matrix calculations, and errors in Matlab program coding.

Conclusion

The possibility to use the penalty function for viscoelastic material behaviour analysis under dynamic load was explored, and numeric values were obtained for initial oscillations and oscillations after optimisation.

The theory shows that all algorithms for constrained optimization are unreliable to some extent [3; 4], and the study performed in Matlab program showed, that limitation was also true for the penalty method. Although penalty methods are among the most powerful class of algorithms available for attacking general nonlinear optimization problems, in the particular numerical experiment the penalty function method did not allow to smooth the non-linear behaviour. Instead, the technique allowed finding with great precision a local minimum. Regarding the convexity characteristics of the objective function and constraints, finding the local minimum was still a challenging goal, so the algorithm could be used successfully for similar tasks.

The main complexity of using the penalty method for viscoelastic material behaviour was connected with a difficulty to adjust the penalty function matrix with the objective function matrix. However, the penalty function method appears to be useful for the sets of data obtained from the real-life experiments.

References

1. Haddad Y. M. Viscoelasticity of Engineering Materials. 1st edition. Springer, 1994. 408 p.
2. Цыфанский С.Л., Бересневич В.И., Лушников Б.В. Нелинейная вибродиагностика машин и механизмов. - Рига: РТУ, 2008
3. Venkataraman P. Applied Optimization with MATLAB Programming, John Wiley & Sons, 2009, 526 p.
4. CVX: Matlab Software for Disciplined Convex Programming, http://cvxr.com/cvx/