

# AN OPTIMIZED ELASTO-PLASTIC SUBGRADE REACTION FOR MODELING THE RESPONSE OF A NONLINEAR FOUNDATION FOR A STRUCTURAL ANALYSIS

Richard Paul RAY<sup>1\*</sup>

## Abstract

*Geotechnical and structural engineers are faced with a difficult task when their designs interact with each other. For complex projects, this is more the norm than the exception. In order to help bridge that gap, a method for modeling the behavior of a foundation using a simple elasto-plastic subgrade reaction was developed. The method uses an optimization technique to position 4-6 springs along a pile foundation to produce similar load deflection characteristics that were modeled by more sophisticated geotechnical finite element software. The methodology uses an Excel spreadsheet for accepting user input and delivering an optimized subgrade spring stiffness, yield, and position along the pile. In this way, the behavior developed from the geotechnical software can be transferred to the structural analysis software. The optimization is achieved through the solver add-in within Excel. Additionally, a beam on a nonlinear elastic foundation model is used to compute deflections of the optimized subgrade reaction configuration.*

## Address

<sup>1</sup> Department of Structural and Geotechnical Engineering, Széchenyi István University, Egyetem ter 1, H9026, Győr Hungary

\* **Corresponding author:** ray@sze.hu

## Key words

- Soil-structure interaction,
- pile foundation,
- nonlinear modeling.

## 1 INTRODUCTION

Structural and geotechnical engineers are faced with a wide array of software packages to help them in the design of structures and foundations. AxisVM and SAP2000 are typical structural design and analysis programs used every day throughout Europe and the U.S. To perform geotechnical design, Plaxis, Fine, FLAC, and MIDAS are typical choices. However, when faced with a complex issue of soil-structure interaction, a comprehensive modeling approach using general purpose finite element programs (NASTRAN, ABAQUS, DIANA) may be prohibitively expensive, or too much of an investment in time and otherwise profitable design talent. This paper describes a method that falls between a nonlinear geotechnical finite element model and a structural design model. The method allows the engineer to create sophisticated models for pile foundations under axial and lateral loads and replace them with a few elasto-plastic subgrade reactions (soil springs) applied at strategic locations on the structural pile to produce the same behavior. This allows the structural design

and analysis to proceed as usual with a more realistic representation of the soil reactions.

## 2 NONLINEAR FEM FOR THE GEOTECHNICAL MODELING OF PILE FOUNDATIONS

Creating realistic models of pile and pile group foundations has evolved into a very sophisticated and relatively inexpensive process for a geotechnical modeler. Using Plaxis or Midas, a modeler can simulate construction stages, consolidation, pile installation, and complicated load cases. While all of these processes cannot be reproduced with a subgrade reaction, a large percentage of the soil structure interaction that is key to dimensioning and arranging structural members can be faithfully modeled.

We have used this approach for bridge abutment designs and mid-rise office buildings. It can also serve as a first estimate for dynamic interactions. When concerns center around the first one or two modes

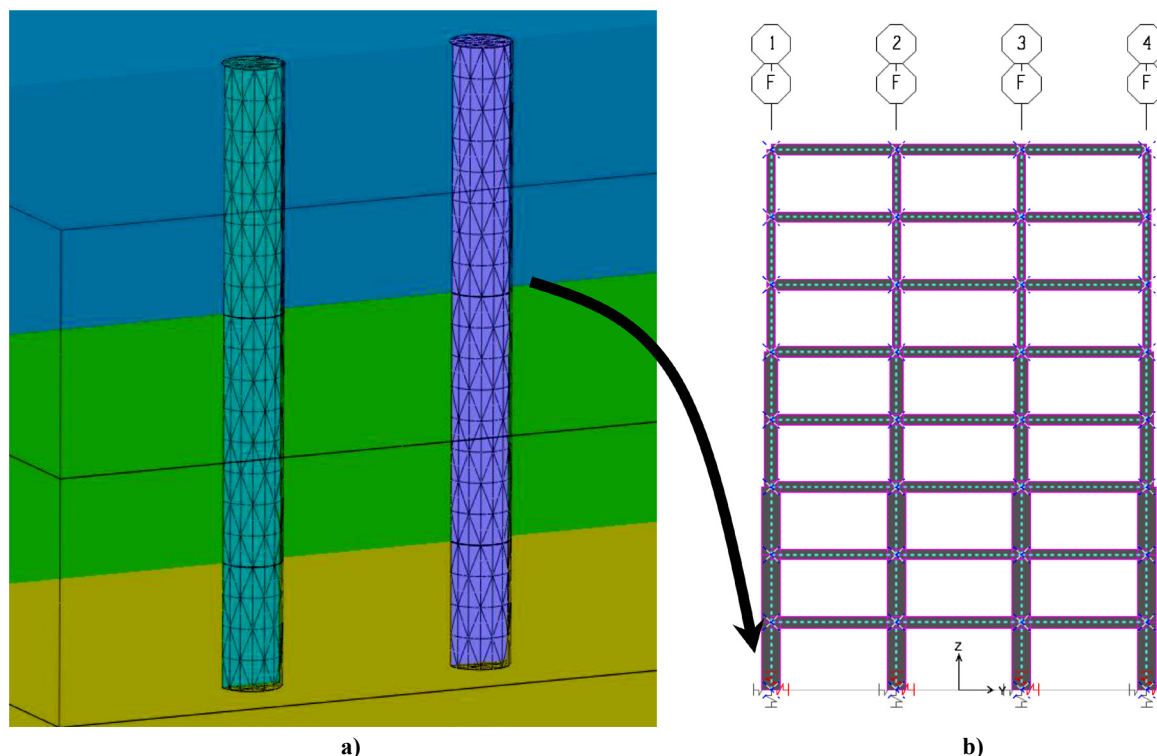


Fig. 1 Geotechnical model (a) and structural model (b) perspectives.

of vibrations in a structure, a subgrade model can produce fairly useful results.

### 3 METHODS

The initial problem consists of two design perspectives: geotechnical and structural (Fig. 1). The first step is to model the geotechnical system carefully using geotechnical software to study the effects of soil strength, stiffness, and layering as well as pile-soil interface conditions. The structural interaction can be modeled where needed as simple elastic frames or walls.

The structural section design and more refined details are left for later. A typical problem would be a medium-sized building supported by piles or small pile groups. The initial behavior studies examine the load deflection behavior of the piles and pile-cap system. The load-displacement and moment-rotation relationships at strategic locations throughout the building foundation are recorded for later use. These may be specific points on the pile cap or the tops of individual piles. As each load case is applied to the foundation systems, the resulting deflections and rotations are recorded and then stored on a spreadsheet for subsequent matching with a simpler spring system. One may consider the behavior of a single pile or the behavior of the group as a block. The degree of detail will depend on the demands of the specific design.

### 4 THE PILE MODEL

In order to simplify this discussion, a lateral load system will be presented. If both the axial and lateral loads are to be modeled, additional vertical springs can be placed at the same nodes, or new nodes can be added with only the vertical springs applied. This example will only fit the lateral displacement at the pile top ( $\delta$ ), but the rotation at the top ( $\theta$ ) may be another value to fit as well. The pile model consists of beam elements connected at the nodes. Each node can accept

a soil spring reaction that is either elastic, elastic-plastic, or generally non-linear. A schematic representation is shown in Fig. 2a.

This example shows four springs with the pile itself divided into three beam elements. One may use any number of springs and elements; however, 3-6 seem to be sufficient. The user can vary the parameters shown in the figure, with some constraints. Each spring may have a different stiffness ( $k_i$ ) and yield value ( $c_i$ ). Other spring models (hyperbolic, polynomial) can be applied as well. The springs can be placed at different depths ( $Z_i$ ) as long as they follow some order with  $K_1$  at the top and  $K_4$  at the bottom. Typically,  $K_1$  and  $K_4$  are constrained to be at the surface and pile tip respectively. For the model in Fig. 2, since  $Z_1=0$  and  $Z_4=L$ , the number of parameters to vary is reduced to 10 ( $Z_2, Z_3, k_1, c_1, k_2, c_2, k_3, c_3, k_4, c_4$ ). Usually, the length, section, and material properties of the pile itself are fixed.

The values of the horizontal force  $F_H$  versus the horizontal displacement  $\delta$ , at the pile top was already measured or calculated by the previous finite element simulation (i.e. Fig. 1a). This load-displacement curve is the target to be reproduced by moving the springs along the pile ( $Z_2, Z_3$ ) and varying the spring stiffnesses ( $k_1, k_2, k_3, k_4$ ) and yield strengths ( $c_1, c_2, c_3, c_4$ ). The optimum solution minimizes the error between the field data (or FEM model) and the optimized springs (Fig. 2b). The spring reactions are applied at the nodes of the beam elements. Naturally, if the springs are moved up or down, the beam elements will change their lengths correspondingly.

### 5 IMPLEMENTATION OF THE OPTIMIZATION METHOD

The method for selecting the best location and parameters for each spring is implemented in Microsoft Excel. The spreadsheet serves to define the input parameters and constraints. The Microsoft Excel Solver add-in performs the minimization process, and the Visual Basic code within the workbook module computes the nonlinear pile top deflection. Each of these components are described in the following sections.

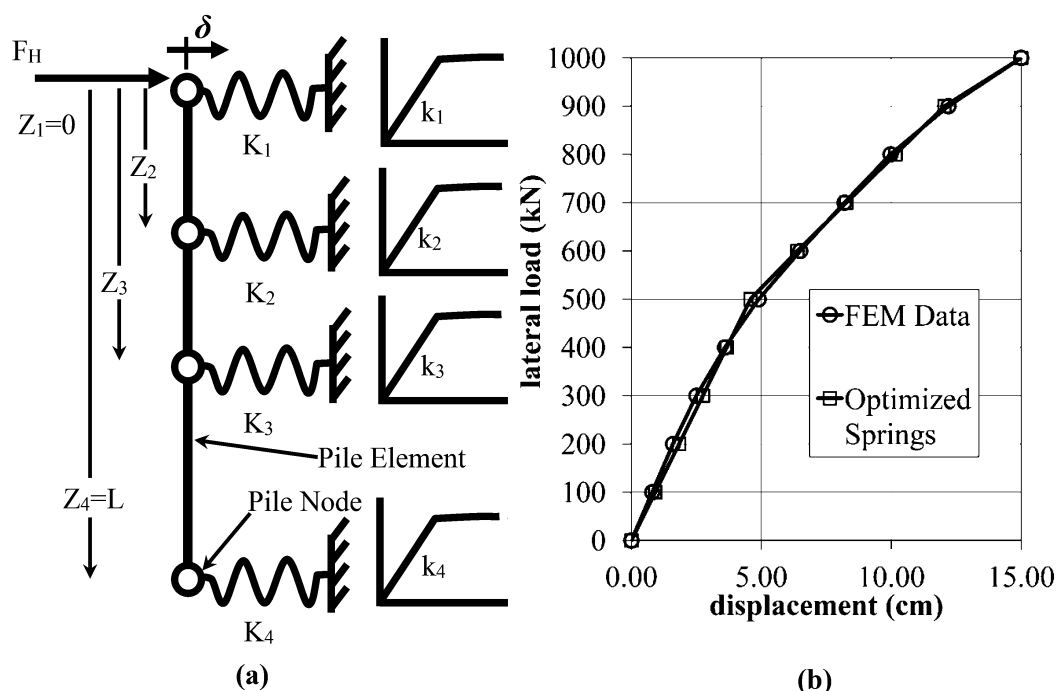


Fig. 2 Schematic representation of a pile with (a) optimized elasto-plastic springs; (b) load displacement curve from FEM fitted by a spring model.

The input/output spreadsheet consists of the input data, spring positions, spring parameters, and constraints. The problem consists of a system with six springs (Fig. 3). The left columns show the load-displacement behavior for the test/FEM calculations; the next column is the deflection, which is computed during the optimization process. The error column is the difference between the test and the

Lateral Load Curve Fitting Procedure				Node	Y	Z	Min	Max
Test	Test	Comp	Error	Num	(cm)	(cm)	Depth	Depth
Load	Defl	Defl	(yt-yc) <sup>2</sup>	1	0	0	0	0
(kN)	(mm)	(mm)	(mm <sup>2</sup> )	2	0	-125	-30	-220
0	0	0	0	3	0	-250	-155	-345
20	0.525	0.679	0.0238	4	0	-375	-280	-470
40	1.05	1.358	0.0950	5	0	-500	-405	-620
60	2.38	2.037	0.1174	6	0	-650	-650	-650
80	3.71	3.704	0.0000	Spring		K i	C i	
100	6.39	5.820	0.3246	Number		(MN/m)	(kN)	
120	9.07	7.937	1.2842	1	20	45		
140	12.69	12.240	0.2023	2	8	50		
160	16.32	16.641	0.1028	3	5	100		
180	20.85	21.041	0.0364	4	10	100		
200	25.37	25.441	0.0051	5	10	100		
Minimize Sum=>			2.1915	6	500	600		
				Toler		1E-06		
				N.Iter		10000		
						K i	C i	
				Spring	Min	Max	Min	Max
				1	5	200	10	500
				2	5	200	10	500
				3	5	200	10	500
				4	5	200	10	500
				5	5	200	10	500
				6	500	1000	500	600

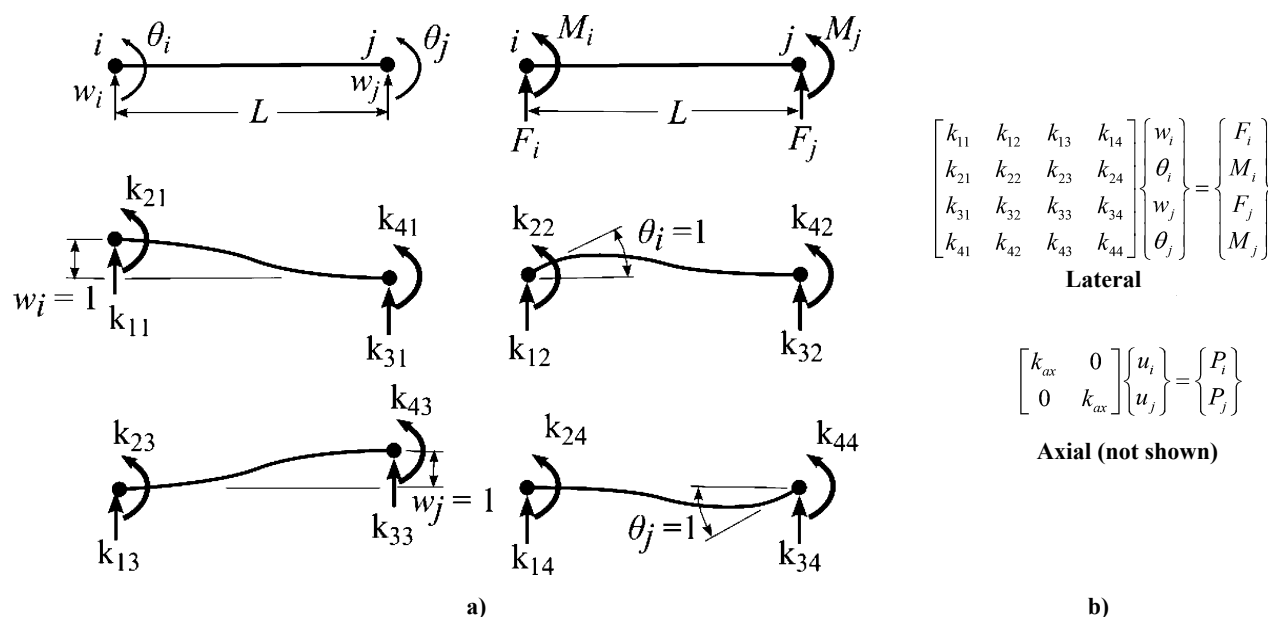
Fig. 3 Input/Output worksheet for the optimization problem for six springs. The first two columns are the measured or FEM data; the third column is the computed response; the fourth column is the square of the difference between columns 2 and 3. The right side sections are the location of the springs, the spring properties, and the constraints on those properties.

computed deformations. At the base of this column is the sum of the errors which is to be minimized by the solver program. On the right side of the spreadsheet are the depths of the nodes where the springs are attached. The Y-values will be varied subject to the minimum and maximum depths listed. Note that the top and bottom nodes (1 and 6) are fixed. Below the node locations are the spring parameters;  $K_i$  is the spring stiffness;  $C_i$  is the yield strength. These may be initial estimates since all 12 of these parameters are systematically varied during the solver's optimization. The constraints on the spring parameters are listed at the bottom and may reflect the soil strengths and stiffness based on the field data. Since they are not being used for the foundation design, the constraints are not as critical, the physical conditions should be reflected as nearly as possible.

Two other parameters are shown: Toler is the convergence tolerance, and N.Iter are the maximum iterations for performing the deflection computations. Usually the worksheet is color coded to reflect the purpose of each cell. Once the test load and test deflection values are entered, the user will set the constraint values for the spring locations, stiffnesses and yield values. The computed deflections are brought into the spreadsheet through a user-defined function call. The function has been programmed "behind" the spreadsheet in the Visual Basic Application (VBA) environment in Excel.

## 6 COMPUTING THE DEFLECTIONS

The deflection of a beam on an elastic foundation was first solved analytically by Hetenyi (1946). Since then there have been many variations of this concept applied to shallow foundations as well as piles (Bowles, 1997; Das, 2009). The procedure adapted here uses a simplified beam element where the end nodes have two degrees of freedom: the displacement in the horizontal direction and the rotation. Loads that are carried by the beam are simplified to horizontal forces and bending moments. If greater freedom is needed, the beam elements can model the vertical displacement and carry the axial loads as well. However for this problem, they are not needed and would only add to the computational demands of the problem.



**Fig. 4** Direct stiffness concept for the beam element formulation in bending. If an axial displacement is desired, two axial forces, two axial deflections, and one axial stiffness would be added to this matrix. For laterally loaded piles the diagram would be turned sideways.

The global stiffness matrix used in the formulation is generated from the direct stiffness method, which is common to many matrix structural analysis programs as well as more general finite element models. The individual element stiffness matrices are first computed as shown in Fig. 4. Then the global matrix is assembled the same way as with any other structural system as they are super-imposed on each other according to their nodal degrees of freedom.

The global stiffness matrix is then combined with the soil reaction matrix to obtain the equilibrium equations.

$$[\mathbf{K}_G]\{\mathbf{y}\} + [\mathbf{K}_s]\{\mathbf{y}\} = \{\mathbf{p}\} \quad (1)$$

where

- $[\mathbf{K}_G]$  is the global stiffness matrix of all the beam elements in the problem
- $[\mathbf{K}_s]$  is the spring stiffness matrix of the soil reactions (diagonal matrix)
- $\{\mathbf{y}\}$  is the horizontal deflection of the element nodes
- $\{\mathbf{p}\}$  is the vector of the horizontal loads and moments applied; there is usually just one load and perhaps one moment applied to the top node

For the linear reaction case,  $[\mathbf{K}_G]$  and  $[\mathbf{K}_s]$  can be added together to form a combined stiffness matrix, and the problem is solved like any other linear model. However, if the soil response is not linear, then the formulation looks like:

$$[\mathbf{K}_G]\{\mathbf{y}\} + [\mathbf{K}_s(\mathbf{y})]\{\mathbf{y}\} = \{\mathbf{p}\} \quad (2)$$

And since  $[\mathbf{K}_s(\mathbf{y})]$  is no longer constant, the matrices cannot be added. Instead, one must use a Newton-Raphson method to solve the non-linear simultaneous equations. It is the simplest multidimensional root-finding method and provides a very efficient means of converging to a root, if one has a sufficiently good initial guess. It can also spectacularly fail to converge, indicating (though not proving) that one's putative root does not exist nearby. In a general sense, one may cast equations in this form (Press et al., 2007):

$$F_i(x_0, x_1, \dots, x_{N-1}) = 0 \quad i = 0, 1, \dots, N-1 \quad (3)$$

We let  $\mathbf{x}$  denote the entire vector of values  $x_i$  and  $\mathbf{F}$  denote the entire vector of functions  $F_i$ . In the neighborhood of  $\mathbf{x}$ , each of the functions  $F_i$  can be expanded in a Taylor series:

$$F_i(\mathbf{x} + \delta\mathbf{x}) = F_i(\mathbf{x}) + \sum_{j=0}^{N-1} \frac{\partial F_i}{\partial x_j} \delta x_j + O(\delta\mathbf{x}^2) \quad (4)$$

The matrix of the partial derivatives appearing in equation (4) is the *Jacobian* matrix  $\mathbf{J}$ :

$$J_{ij} \equiv \frac{\partial F_i}{\partial x_j} \quad (5)$$

In the matrix notation equation 4 is

$$\mathbf{F}(\mathbf{x} + \delta\mathbf{x}) = \mathbf{F}(\mathbf{x}) + \mathbf{J} \cdot \delta\mathbf{x} + \mathbf{O}(\delta\mathbf{x}^2) \quad (6)$$

By neglecting the terms of order  $\delta\mathbf{x}^2$  and higher and by setting  $\mathbf{F}(\mathbf{x} + \delta\mathbf{x}) = 0$ , we obtain a set of linear equations for the corrections  $\delta\mathbf{x}$ , which move each function closer to zero simultaneously, namely:

$$\mathbf{J} \cdot \delta\mathbf{x} = -\mathbf{F} \quad (7)$$

Matrix equation 7 can be solved by an LU decomposition. The corrections are then added to the solution vector:

$$\mathbf{x}_{new} = \mathbf{x}_{old} + \delta\mathbf{x} \quad (8)$$

and the process is iterated to a convergence. In general it is a good idea to check the degree to which both the functions and variables have converged. Once either one of them reaches machine accuracy, the other will not change. For our problem, as written in the Visual Basic Application code within the spreadsheet, the following algorithm is used:

1. Populate stiffness matrix  $[\mathbf{KStiff}]$ , soil reaction matrix  $[\mathbf{KSoil}]$ , soil force function vector  $\{\mathbf{F}(\mathbf{Y})\}$ , derivative soil force vector  $\{\mathbf{DF}(\mathbf{Y})\}$ , load vector  $\{\mathbf{P}\}$  ( $\mathbf{KSoil}$  usually unit diagonal matrix)
2. Governing equation:  $[\mathbf{KStiff}]\{\mathbf{Y}\} + [\mathbf{KSoil}]\{\mathbf{F}(\mathbf{Y})\} = \{\mathbf{P}\}$

3. Solving by Newton-Raphson, guess initial values {Y0}
4. Compute [KStiff] {Y0} + [KSoil] {F(Y0)} - {P} = {Gmat}
5. Compute [KStiff] + [KSoil] {DF(Y0)} = [Fmat]
6. Solve for {DeltaY} by Gauss Elim. [Fmat] {DeltaY} = -{Gmat}
7. {YNew} = {Y0} + {DeltaY}
8. reiterate until {DeltaY} is small, solution is {YNew}

The method works for any soil reaction that can be differentiated either as an analytical derivative or a finite difference (the **DF(Y0)** term in the VBA listing above). Note that for an elasto-plastic response, the soil reaction and derivative are adjusted based on the yield value. When below the yield value, the derivative is  $k_s$  and above the yield value; the derivative is 0. If one wishes to use a polynomial function such as the Ramberg Osgood formulation or a Hyperbolic formulation for a soil response, the derivative function can either be programmed by substituting the analytical formula derivative or using a finite-difference slope computation. The hyperbolic formulation is shown in Eq. 9:

$$F_{soil} = \frac{y}{A + By} \quad dF_{soil} = \frac{A}{(A + By)^2} \quad (9)$$

where  $F_{soil}$  = soil reaction force (kN)

$y$  = deflection (m)

$A = 1/E_{soil}$  (Young's modulus, kPa)

$B = 1/\tau_{max}$  (maximum shear strength of soil, kN)

The reaction force would also need a value for the contributing area (1m x length); however, that would vary with the change in location and dimensions of the spring reaction system.

This process of computation is programmed as a user function in Excel VBA and is recalculated whenever inputs to the function change. Using the spreadsheet example above, there are 11 calls to the nonlinear solution (generating new values of the computed deflection) every time an optimization parameter is changed by the solver. The Toler and N.Iter spreadsheet cells set the tolerance for the convergence (Eq. 8) and limit the Newton-Raphson method to a finite number of iterations.

## 7 SOLVER

The solver add-in bundled with Microsoft System's Excel spreadsheet was developed by Frontline systems and incorporated into Excel in the mid-1990s (Fylstra et al., 1998). From the perspective of Operations Research/Management Science, the Solver is a somewhat low-end solution to optimization problems. However, the modest standing of the software is mainly due to its speed and capacity limitations, not the correctness or rigor of the methods used to perform the optimization. A comparison of the different solver "packages" is shown in Table 1.

## 8 SOLVING NONLINEAR PROBLEMS

When the Assume Linear Model box in the Solver Options dialog is cleared, the Excel Solver uses the generalized reduced gradient method (GRGM), as implemented in the GRG2 code (Lasdon et al., 1978), to solve the problem. Like other gradient-based methods, GRG2 is guaranteed to find a local optimum only for problems with continuously differentiable functions and then only in the absence of numerical difficulties (such as degeneracy or ill conditioning). The GRGM approach requires a great deal of computational finesse in that if one wishes to reach an optimum in a reasonable time, taking the steepest descent path does not often lead to a rapid solution. Nuances, both conceptual and computational in the algorithm are worth investigating for the interested reader (see Press et al., 2007, on conjugate gradient methods).

However, GRG2, compared to other nonlinear optimization methods, has a reputation for robustness when dealing with difficult problems where these conditions are not fully satisfied. The GRG approach used by the solver remained stable for almost all of the problems attempted in this study. Most of the solver difficulties were due to the instability of the *physical* problem, not the mathematical model representing it.

Perhaps the most notable characteristic of the optimization "system" is that it uses finite differences to determine the gradients of the functions and constraints. This is due to the nature of Excel; it is a numerical rather than a symbolic (such as Mathematica) system. In order for the optimization programming to interact with other Excel functions, all the problem manipulations had to be numeric.

**Tab. 1** Relative performance characteristics of various optimization solvers (Fylstra et al., 1998).

	Excel Built-In Solver	Premium Solver	Premium Solver Plus	Premium SolverPlatform
NLP Variables/ Constraints	200/100 + bounds	400/200 + bounds	400/200 + bounds	1000/1000 + bounds
LP Variables/ Constraints	200/unlimited	800/unlimited	800/unlimited	2000-16000/unlimited
Setup Performance	1x	1-50x	1-50x	1-50x
NLP Performance	1x	1x	1.5x	2-10x
LP Performance	1x	2-3x	2-3x	Large Scale
MIP Performance	1x	5-10x	25-50x	25-50x
Selection of optimizers	Fixed set	Fixed set	Fixed set	Multiple choices, field-installable
LP/QP Methods	Simplex w/bounds	Enhanced Simplex w/bounds	Enhanced Simplex, Dual Quadratic	Sparse Simplex, LU, Markowitz
MIP Methods	Branch & Bound	Enhanced Branch & Bound	Enhanced B&B, P&P, Dual Simplex	Enhanced B&B, P&P, Dual Simplex
NLP Methods	GRG2	GRG2	Enhanced GRG2	LSGRG, SQP, etc.
Reports	Standard: Answer, Limits, Sensitivity	Standard + Linearity, Feasibility	Standard + Linearity, Feasibility	Standard + Linearity, Feasibility



## 9 OPTIMIZATION APPLIED TO A PILE RESPONSE

There are several aspects of the specific optimization that are important. They include:

1. The variable to be optimized
2. The parameters to be manipulated to produce an optimum result
3. The constraints applied to the general problem (usually to the parameters in (2))
4. The functional relationship between the parameters and the system's response.

A sample of the spreadsheet used in the optimization is shown in Fig. 3. The variable to be optimized is indicated in the figure. It is the sum of the square of the error between the test deflections (column 2) and the computed deflections (column 3). This is exactly the same as a least-squares fit problem for trend lines in Excel. The goal of the optimization is to minimize this value.

The parameters that are allowed to vary are shown in the table, i.e., the Y-coordinate of the pile nodes, except for nodes 1 and 6. Node 1 is set on the surface, and node 6 represents the bottom of the pile. From the table, one can see that there are four Y-coordinate parameters that can be manipulated. The spring parameters ( $K_i, C_i$ ) are shown as well, with two parameters for each spring. For this problem, the solver has a total of 16 parameters (18 minus the 2 fixed locations) it can vary in order to find an optimum solution. If more springs are added, then there will be more possible variations.

All the parameters are constrained by the user. The spring depths are limited so that one spring does not get any closer than a set distance to its neighbor. This is set by the Excel formulas in the cells in columns 8 and 9 labeled "min depth" and "max depth". These constraints can change as the spring locations change so that the minimum allowed depth of spring 2 is -30 cm (30 cm from spring 1), and the maximum allowed depth is -220 (30 cm from spring 3). If spring 3 changes its depth during the optimization, then the maximum allowed depth for spring 2 would also change. Other constraints set by the user are for the stiffness and yield of each spring. There may be physical limits to the strength and stiffness of the soil one might encounter, so setting these constraints to reasonable values allows the user to better model the soil profile.

The solver does not guarantee that it will find the best solution (the global minimum) but only a satisfactory solution (the local minimum). However, one may obtain a global, or nearly global minimum, by carefully setting the initial values for the depth, stiffness, and yield in the spreadsheet before initiating the solver. Other issues with the solver's ability to find a minimum solution are often linked to "scaling factor" problems. These occur when the parameters have widely different magnitudes (different by factors of 1000-10,000); hence the units for the parameters were chosen so they all had similar numerical values.

## 10 RESULTS AND CONCLUSIONS

Figure 2b shows the results from a four-spring optimization problem. The values for goodness of fit ( $R^2$ ) can be computed using the data already generated. Once the springs and their locations are computed, they can be transferred to a structural analysis program and applied to the pile beam member that is integrated into the remaining structure. In this way, a structural designer has a more accurate soil response and will produce more realistic results for different loading configurations (Szép and Ray, 2013).

The impetus for this paper came about from a need to connect geotechnical modeling to structural design in a straightforward and understandable way. The results have been satisfying in that the structural design and analysis produce more accurate behavior with very little added effort. The reader should remember, however, that the model does not address many important geotechnical issues associated with building construction. It will certainly not replace careful geotechnical investigations, rigorous analysis, and experienced insight in the design process.

## REFERENCES

- Bowles, J. E. (1996)** *Foundation Analysis and Design*, McGraw-Hill, New York.
- Das, B. M. (2010)** *Principles of foundation engineering*, Cengage Learning.
- Fylstra, D. L. – Lasdon, L. – Watson, J. – Warren, A. (1998)** *Design and Use of the Microsoft Excel Solver*, INTERFACES, 28, 5, 29-55.
- Hetenyi, M. (1946)** *Beams on elastic foundation*, Ann Arbor Press, Ann Arbor Michigan.
- Lasdon, L. S. – Waren, A. D. – Jain, A. – Ratner, M. (1978)** *Design and testing of a generalized reduced gradient code for nonlinear programming*, ACM Transactions on Mathematical Software, 4, 1, 34-49.
- Press, W. H. – Teukolsky, S. A. – Vetterling, W. T. – Flannery, B. P. (2007)**. *Numerical recipes, 3rd ed.: The art of scientific computing*. Cambridge University Press.
- Szép, J. – Ray, R. P. (2013)** *Integrating Nonlinear Pile Behavior with Standard Structural Engineering Software*, Proceedings of 18<sup>th</sup> Int'l Conf on Soil Mech Geot Engrg, Paris 2869-72.