Advances in Engineering Software 121 (2018) 84–97

Contents lists available at ScienceDirect

Advances in Engineering Software

journal homepage: www.elsevier.com/locate/advengsoft

Research paper

ErosLab: A modelling tool for soil tests

Yin-Fu JIN[a,b], Zhen-Yu YIN[a,b,*]

[a] Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong
[b] Research Institute of Civil Engineering and Mechanics (GeM), UMR CNRS 6183, Ecole Centrale de Nantes, France

ARTICLE INFO

Keywords:
Laboratory tests
Geomechanics
Geotechnical engineering
Constitutive model
Interface
Software

ABSTRACT

The focus of this paper is ErosLab, a useful tool for the development, analysis and application of constitutive models developed to solve the modelling problems inherent in soil tests. The ErosLab is programmed in the way of admixture programming with C#, MATLAB and FORTRAN, offering a powerful environment for various kinds of modelling soil tests. The proposed tool has six important features: (1) a mechanical calculator; (2) the ability to cover various kinds of soil tests; (3) a number of soil models with a user extension interface; (4) multiple methods of loading control; (5) comprehensive and efficient debugging; and (6) visualisation with graphical displays. Furthermore, the entire graphical user interface and usage instructions for the tool are briefly illustrated in simple and practical terms. Finally, three case studies are presented in which ErosLab was used, to highlight its performance in modelling tests for different soils.

1. Introduction

Constitutive models play an important role in the design and construction of geotechnical engineering. To date, hundreds of different soil constitutive models, varying in view from micro to macro, have been proposed [1–10]. A range of results may be obtained depending on the selection of model, leading to different engineering decisions, which consequently alters the economy and risk level of problems. However, most engineers have failed to fully understand constitutive models and have invariably chosen a model based on their own preferences and experiences, hoping that a "one-size-fits-all" approach can solve all engineering problems. Some widely used models can sometimes result in significantly unreasonable predictions when applying to conventional engineering [11], as seen when the Mohr-Coulomb model was adopted to analyse an excavation [12] and when the modified cam-clay was employed to predict the long-term settlement of embankment [13–16]. A lack of proper understanding of the constitutive model has become one of the main risk factors in terms of accidents [17–21]. Therefore, it is essential that the merits and drawbacks of the selected model are completely understood before its application. In general, the quickest way to do this is to simulate laboratory tests. However, most engineers struggle with writing a computer program that can implement the soil model to achieve such a simulation. To address this, a tool that could model soil tests by providing a variety of constitutive models would be highly useful.

Previously, a range of practical tools in the field of geotechnical engineering have been developed. These offer an object-oriented design to simulate engineering issues using a variety of constitutive models, such as some commercial codes (ABAQUS [22], FLAC [23], PLAXIS [24] and COMSOL [25]), or open sources codes [26–29]. Of these, only PLAXIS has partial functions in the modelling of soil tests. However, the kinds of tests provided, and the loading control, are somewhat limited. This bolsters the case for the development of a tool that can offer a powerful environment for simulating various kinds of laboratory tests. Engineers could use this to understand a soil model without the need to reproduce its mode of operation, which is another area of difficulty.

In this paper, a modelling tool (ErosLab) for soil laboratory tests is developed and introduced. First, the different kinds of tests that can be used with the tool are briefly introduced. Second, its general framework is presented, including its mixed language programming and six main features. Third, its graphical user interface and usage instructions are illustrated. Finally, descriptions are given of the carrying out of three cases of parameter identification (first for modelling of sand behaviours, second for modelling of clay behaviours and the third for modelling of time effects of soil). The developed software can be freely downloaded from the following URL: http://www.geoinvention.com/en/news.asp?big=14.

Since the development of constitutive models is usually based on laboratory tests, developing this tool should be first helpful for the research purpose of constitutive modelling. Even though field-scale problems cannot be directly simulated, the debugging scheme in this tool includes complex loading combinations reflecting various in-situ conditions. Furthermore, the tool should also be helpful for the teaching purpose and basic training of constitutive modelling for students.

* Corresponding author at: Department of Civil and Environmental Engineering, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong.
E-mail address: zhenyu.yin@ec-nantes.fr (Z.-Y. YIN).

https://doi.org/10.1016/j.advengsoft.2018.04.003
Received 14 October 2017; Received in revised form 13 March 2018; Accepted 1 April 2018

2. Basic definitions

2.1. Stress analysis

The stress state of a single element can be described using six independent stress components. In constitutive model programming, the stress tensor is usually expressed as

$$\sigma_{ij} = [\sigma_{xx} \quad \sigma_{yy} \quad \sigma_{zz} \quad \sigma_{xy} \quad \sigma_{xz} \quad \sigma_{yz}]^T \tag{1}$$

The σ_m (or p) is defined as the average normal stress or mean effective stress:

$$\sigma_m = \frac{1}{3}(\sigma_{xx} + \sigma_{yy} + \sigma_{zz}) \tag{2}$$

The deviatoric stress tensor can be expressed as:

$$s_{ij} = \sigma_{ij} - \sigma_m\delta_{ij} = \begin{bmatrix} \sigma_{xx}-\sigma_m & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy}-\sigma_m & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz}-\sigma_m \end{bmatrix} = \begin{bmatrix} s_{xx} & s_{xy} & s_{xz} \\ s_{yx} & s_{yy} & s_{yz} \\ s_{zx} & s_{zy} & s_{zz} \end{bmatrix} = \begin{bmatrix} s_{11} & s_{12} & s_{13} \\ s_{21} & s_{22} & s_{23} \\ s_{31} & s_{32} & s_{33} \end{bmatrix} \tag{3}$$

The first, second and third invariants of the stress tensor are:

$$\begin{aligned} I_1 &= \sigma_{xx} + \sigma_{yy} + \sigma_{zz} \\ I_2 &= \begin{vmatrix} \sigma_{xx} & \tau_{xy} \\ \tau_{yx} & \sigma_{yy} \end{vmatrix} + \begin{vmatrix} \sigma_{yy} & \tau_{yz} \\ \tau_{zy} & \sigma_{zz} \end{vmatrix} + \begin{vmatrix} \sigma_{zz} & \tau_{zx} \\ \tau_{xz} & \sigma_{xx} \end{vmatrix} = \sigma_{xx}\sigma_{yy} + \sigma_{yy}\sigma_{zz} + \sigma_{zz}\sigma_{xx} - \tau_{xy}^2 \\ &\quad - \tau_{yz}^2 - \tau_{zx}^2 \\ I_3 &= \begin{vmatrix} \sigma_{xx} & \tau_{xy} & \tau_{xz} \\ \tau_{yx} & \sigma_{yy} & \tau_{yz} \\ \tau_{zx} & \tau_{zy} & \sigma_{zz} \end{vmatrix} = \sigma_{xx}\sigma_{yy}\sigma_{zz} + 2\tau_{xy}\tau_{yz}\tau_{zx} - \sigma_{xx}\tau_{yz}^2 - \sigma_{yy}\tau_{zx}^2 - \sigma_{zz}\tau_{xy}^2 \end{aligned} \tag{4}$$

While the three invariants of the deviatoric stress tensor are:

$$\begin{cases} J_1 = s_{xx} + s_{yy} + s_{zz} = 0 \\ J_2 = \frac{1}{2}s_{ij}s_{ji} = \frac{1}{2}(s_{xx}^2 + s_{yy}^2 + s_{zz}^2 + 2\tau_{xy}^2 + 2\tau_{xz}^2 + 2\tau_{yz}^2) \\ J_3 = s_{xx}s_{yy}s_{zz} = 2\tau_{xy}\tau_{yz}\tau_{xz} - \sigma_{xx}\tau_{yz}^2 - \sigma_{yy}\tau_{xz}^2 - \sigma_{zz}\tau_{xy}^2 \end{cases} \tag{5}$$

The deviatoric stress q can be calculated using the second invariant of the deviatoric stress tensor J_2.

$$q = \sqrt{3J_2} \tag{6}$$

In a triaxial test, the deviatoric stress q can be simplified to $q = |\sigma_a - \sigma_r|$, or $q = \sigma_a - \sigma_r$ to distinguish the compression or the extension conditions.

The lode angle θ can be calculated using the invariants of the deviatoric stress tensor as follows:

$$\cos 3\theta = \frac{3\sqrt{3}}{2}\frac{J_3}{J_2^{\frac{3}{2}}} \tag{7}$$

This works for a conventional triaxial compression test with $\sigma_2 = \sigma_3$, $b = 0$ and $\theta = 0°$; for a conventional triaxial extension test with $\sigma_2 = \sigma_1$, $b = 1$ and $\theta = 60°$; and when $\sigma_2 = (\sigma_1 + \sigma_3)/2$, $b = 0.5$ and $\theta = 30°$ Note that b is the parameter of intermediate principal stress, and is defined as $b = (\sigma_2 - \sigma_3)/(\sigma_1 - \sigma_3)$).

The principal stress σ_1, σ_2 and σ_3 can be obtained as follows,

$$\begin{cases} \sigma_1 = \frac{I_1}{3} + 2\sqrt{\frac{J_2}{3}}\cos\theta \\ \sigma_2 = \frac{I_1}{3} + \sqrt{\frac{J_2}{3}}(\cos\theta - \sqrt{3}\sin\theta) \\ \sigma_3 = \frac{I_1}{3} + \sqrt{\frac{J_2}{3}}(\cos\theta + \sqrt{3}\sin\theta) \end{cases} \quad \text{or} \quad \begin{cases} \sigma_1 = p + \frac{2}{3}q\cos\theta \\ \sigma_2 = p + \frac{2}{3}q\cos\left(\theta - \frac{2\pi}{3}\right) \\ \sigma_3 = p + \frac{2}{3}q\cos\left(\theta + \frac{2\pi}{3}\right) \end{cases} \tag{8}$$

2.2. Strain analysis

Under the small deformation condition, the strain tensor can be divided into deviatoric and spherical tensors as follows:

$$\varepsilon_{ij} = \begin{bmatrix} \varepsilon_x - \varepsilon_m & \frac{1}{2}\gamma_{xy} & \frac{1}{2}\gamma_{xz} \\ \frac{1}{2}\gamma_{yx} & \varepsilon_y - \varepsilon_m & \frac{1}{2}\gamma_{yz} \\ \frac{1}{2}\gamma_{zx} & \frac{1}{2}\gamma_{zy} & \varepsilon_z - \varepsilon_m \end{bmatrix} + \begin{bmatrix} \varepsilon_m & 0 & 0 \\ 0 & \varepsilon_m & 0 \\ 0 & 0 & \varepsilon_m \end{bmatrix} = e_{ij} + \varepsilon_m\delta_{ij}, \tag{9}$$

where γ is the engineering shear strain, e_{ij} is deviatoric strain tensor, and the mean strain ε_m is defined as $\varepsilon_m = (\varepsilon_x + \varepsilon_y + \varepsilon_z)/3$.

The general shear strain ε_d is defined as:

$$\varepsilon_d = \sqrt{\frac{2}{3}e_{ij}e_{ji}} \quad \text{and} \quad \varepsilon_d = \frac{2}{3}(\varepsilon_1 - \varepsilon_3) \quad \text{for a triaxial test } (\varepsilon_2 = \varepsilon_3) \tag{10}$$

The volumetric strain ε_v is (under the small deformation assumption):

$$\varepsilon_v = \frac{\Delta V}{V} = (1 + \varepsilon_1)(1 + \varepsilon_2)(1 + \varepsilon_3) - 1 \approx \varepsilon_1 + \varepsilon_2 + \varepsilon_3 \tag{11}$$

3. ErosLab tool

3.1. Mixed-language programming

Fig. 1 shows the schematic overview of the mixed-language programming for ErosLab. The tool is programmed using the admixture method, with Microsoft Visual C#, MATLAB and FORTRAN. The graphical user interface is programmed in C#, the post-processing (for plotting the figure, exporting the results, generating the report and reading the help documentation) is realised using MATLAB, and the constitutive models are programmed in FORTRAN. All MATLAB files are built as dynamic library files (*.dll) under the .NET Framework 4.0. The version of MATLAB used is MATLAB 2016b.

3.2. General structure of of ErosLab

The general structure of ErosLab is shown in Fig. 2 and the six main features are summarised in this section.

3.2.1. Provision of a mechanical calculator

The tool provides a practical mechanical calculator. For a given stress tensor σ_{ij}, the invariants of the stress tensor (I_1, I_2 and I_3), the invariants of the deviatoric stress tensor (J_1, J_2 and J_3), the principal stress (σ_1, σ_2 and σ_3), the mean stress p, the deviatoric stress tensor s_{ij}, the deviatoric stress q, the lode angle θ, and the directions of principal stresses (l, m and n) can be obtained. Furthermore, the transformation of coordinates can also be achieved. For a given strain tensor, the invariants of the strain tensor (I_1', I_2' and I_3'), the invariants of the deviatoric strain tensor (J_1', J_2' and J_3'), the principal strain (ε_1, ε_2 and ε_3), the mean strain ε_m, the deviatoric stress tensor e_{ij}, and the deviatoric strain

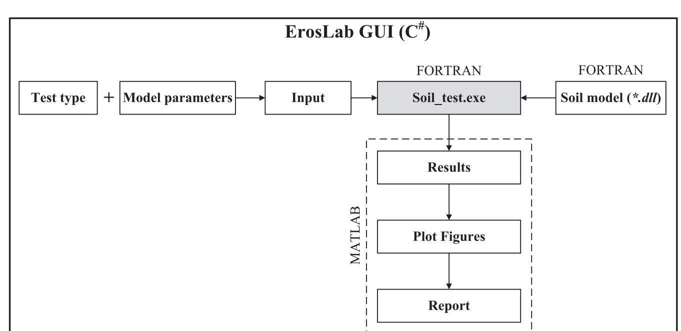

Fig. 1. Schematic overview of the mixed-language programming for ErosLab.