Civil Engineering Beyond Limits

www.acapublishing.com

RESEARCH ARTICLE

NumPy-based Calibration of Basic Hypoplastic Constitutive Models

Muzafer Kerem Ertek^{a*}, Gökhan Demir^a, Utku Köktan^a

^aDepartment of Civil Engineering, Faculty of Engineering, Ondokuz Mayıs University, 55139, Samsun, Turkey

Abstract

Constitutive modeling of soils is a crucial topic in geotechnics. Several constitutive models for soils can be found in material libraries of open-source or commercial geotechnical software packages, and these models can be based on various theories. Hypoplasticity as a relatively young theory is an alternative to elastoplasticity and consistently attracts new researchers. Contrary to elastoplasticity, hypoplasticity does not involve a priori defined yield surface, flow rule and plastic potential and arises from a simple tensorial function of the rate type. An exhaustive review of literature, however, points to the fact that for the calibration of these models, commercial symbolic mathematics software is mostly referred to and a calibration procedure based upon an open-source software which any individuals can easily make use of is missing. Therefore, an explicit procedure for calibration making use of NumPy, which is the main package for scientific computing with Python, following a concise summary for the theory of hypoplasticity is established. By doing so, it is expected to draw attention to take advantage of open-source packages that almost the majority of the scientific community utilizes increasingly.

Keywords: Hypoplasticity, Calibration, NumPy, Python, Constitutive modeling.

1. Introduction

Mathematical modeling of the behavior of soils may be one of the most challenging professions due to the complexity of its characteristics such that the majority of expansions in plasticity theory have been achieved owing to the exertions to describe the irreversible behavior of soils [1]. Inspired by the pioneering work of Truesdell (1955) [2] in which the proposed concept is later on called the hypoelasticity theory by Truesdell (1955) [3], many attempts have emerged to model the irreversible behavior of soils, e.g. by Valanis (1971) [4] and Kolymbas (1977) [5] with intent to establish alternative theories to plasticity. Hypoplasticity as a relatively young theory has mainly been the center of attraction after the tensorial equation proposed by Kolymbas (1987) [6] with the capability to represent apparent aspects of cohesionless soils. However, it took some more years for hypoplasticity to become a well-established theory since the processor had some drawbacks, which have been overcome constituting a framework with several restrictions [7].

Calibration has always been a supplementary fact in hypoplasticity research such that all the proposed models have been revealed more or less with a detailed explanation about parameter identification. The identification of material parameters depends on only one triaxial test in basic hypoplastic models while some of the extended hypoplastic models require granulometric properties of cohesionless soil for calibration. The fact that commercial software is not always easily possessed and benefiting open-source opportunities so far as possible is a growing tendency in the scientific community, this paper aims to emphasize that calibration can explicitly be made with Python complemented by NumPy package. The given code has the potential to be a part of a comprehensive program for research in constitutive modeling with hypoplasticity.

1.1 Notation

Tensor notation will be used throughout the paper. Second-order tensors will be represented by bold typeface, e.g. $\mathbf{T}$. The second-order unit tensor will be symbolized by $\mathbf{1}$ which also stands for the Kronecker delta (δ_{ij}). The trace of a tensor, e.g. trace of $\mathbf{D}$, will be denoted by $\text{tr}(\mathbf{D})$. Multiplication of two tensors, e.g. $\mathbf{T}$ and $\mathbf{D}$ will be represented as $\mathbf{T}\mathbf{D}$ ($T_{ij}D_{jk}$ in index notation). An asterisk as superscript stands for the deviatoric part of a tensor, e.g. $\mathbf{T}^* = \mathbf{T} - 1/3 \text{tr}(\mathbf{T})\mathbf{1}$. A tensor raised to a power, e.g. $\mathbf{T}^2$ represents the multiplication $\mathbf{T}\mathbf{T}$. Euclidian norm of a tensor will be represented by brackets $\|\cdot\|$, e.g. for $\mathbf{D}$ as $\|\mathbf{D}\|$.

2. Hypoplastic Constitutive Model

Wu and Kolymbas (1990) [8] define a hypoplastic constitutive model bearing in mind that there exists a tensorial (tensor-valued) function $\mathbf{H}$ such that

$$\dot{\mathbf{T}} = \mathbf{H}(\mathbf{T}, \mathbf{D}) \quad (1)$$

where $\mathbf{T}$, $\mathbf{D}$ and $\dot{\mathbf{T}}$ stand for the Cauchy stress, strain rate (or stretching) and the Jaumann (or Zaremba) stress rate tensors, respectively. Jaumann stress rate can be defined as follows:

$$\dot{\mathbf{T}} = \dot{\mathbf{T}} + \mathbf{T}\mathbf{W} - \mathbf{W}\mathbf{T} \quad (2)$$

where $\dot{\mathbf{T}}$ and $\mathbf{W}$ denote the material time derivative of Cauchy's stress and vorticity (spin) tensors, respectively. The definition of the above tensors can be found in Truesdell and Noll (1965) [9].

*Corresponding Author: mkerem.ertek@omu.edu.tr
(M.K. Ertek)

Eq. 1 does not yield robust constitutive equations due to the broadness of numerous possibilities. Hence, some restrictions have to be imposed to acquire a more pronounced formulation [7]. Two of the below restrictions are based on the general principles of continuum mechanics while the other depends on the experimental observations made by Goldscheider (1984) [10] by true triaxial tests on sand.

To leave natural time out of the constitutive model to enable rate-independence, $\mathbf{H}$ should be positively homogeneous of the first degree in $\mathbf{D}$:

$$\mathbf{H}(\mathbf{T}, \lambda \mathbf{D}) = \lambda \mathbf{H}(\mathbf{T}, \mathbf{D}) \quad (3)$$

where λ is an arbitrary and positive scalar. The above restriction is the first one applied to Equation (1). The second restriction is for ensuring the tangential stiffness is proportional to stress level and depends on the following findings by Goldscheider (1984) [10] which reveals that a proportional strain (or stress) path starting from a nearly stress-free and undistorted state leads to a proportional stress (or strain) path. Therefore, the function $\mathbf{H}$ must be homogeneous in $\mathbf{T}$ to fulfill this verity:

$$\mathbf{H}(\lambda \mathbf{T}, \mathbf{D}) = \lambda^n \mathbf{H}(\mathbf{T}, \mathbf{D}) \quad (4)$$

where λ is an arbitrary scalar and n stands for the order of homogeneity in stress. This restriction implies that the tangential stiffness is proportional to the n -th power of the stress level $(\text{tr} \mathbf{T})^n$ so that experiments conducted under different stress levels can be normalized by $(\text{tr} \mathbf{T})^n$ [7].

The third and the last restriction are a general requirement on the mathematical description of the material behavior that is objectivity [11]. The constitutive equation is objective when the function $\mathbf{H}$ satisfies.

$$\mathbf{H}(\mathbf{Q} \mathbf{T} \mathbf{Q}^T, \mathbf{Q} \mathbf{D} \mathbf{Q}^T) = \mathbf{Q} \mathbf{H}(\mathbf{T}, \mathbf{D}) \mathbf{Q}^T \quad (5)$$

where $\mathbf{Q}$ is an orthogonal tensor. The representation theorem for isotropic tensorial functions makes it possible to fulfill the objectivity requirement if the function $\mathbf{H}$ is constituted according to the representation theorem for a tensorial function of two symmetric tensors [9,12,13] given as

$$\begin{aligned} \dot{\mathbf{T}} = & \alpha_0 \mathbf{1} + \alpha_1 \mathbf{T} + \alpha_2 \mathbf{D} + \alpha_3 \mathbf{T}^2 + \alpha_4 \mathbf{D}^2 + \alpha_5 (\mathbf{T} \mathbf{D} + \mathbf{D} \mathbf{T}) \\ & + \alpha_6 (\mathbf{T}^2 \mathbf{D} + \mathbf{D} \mathbf{T}^2) + \alpha_7 (\mathbf{T} \mathbf{D}^2 + \mathbf{D}^2 \mathbf{T}) + \alpha_8 (\mathbf{T}^2 \mathbf{D}^2 + \mathbf{D}^2 \mathbf{T}^2) \end{aligned} \quad (6)$$

where the coefficients α_i ($i = 0, \dots, 8$) are scalar functions of invariants and joint invariants of $\mathbf{T}$ and $\mathbf{D}$:

$$\alpha_i = \alpha_i(\text{tr} \mathbf{T}, \text{tr} \mathbf{T}^2, \text{tr} \mathbf{T}^3, \text{tr} \mathbf{D}, \text{tr} \mathbf{D}^2, \text{tr} \mathbf{D}^3, \text{tr}(\mathbf{T} \mathbf{D}), \text{tr}(\mathbf{T}^2 \mathbf{D}), \text{tr}(\mathbf{T} \mathbf{D}^2), \text{tr}(\mathbf{T}^2 \mathbf{D}^2)) \quad (7)$$

In hypoplasticity, Eq. 1 can be decomposed into two parts called linear and nonlinear parts, one for the reversible and the other for the irreversible behavior, respectively. Within the framework of Eq. 1, the general formulation for the hypoplastic rate-independent constitutive equation proposed by Wu and Kolymbas (1990) [8] can be written as the sum of linear and nonlinear terms of the strain rate $\mathbf{D}$:

$$\dot{\mathbf{T}} = \mathbf{L}(\mathbf{T}, \mathbf{D}) + \mathbf{N}(\mathbf{T}, \mathbf{D}) \quad (8)$$

where $\mathbf{L}$ is linear in $\mathbf{D}$ and $\mathbf{N}$ is nonlinear in $\mathbf{D}$. Keeping in mind that the nonlinear dependence of $\mathbf{N}$ on $\mathbf{D}$ should also satisfy the restriction of rate-independence, Wu and Kolymbas (1990) [8] constructed the following convenient framework among other unsuitable alternatives by making use of the response envelopes introduced by Gudehus [14]:

$$\dot{\mathbf{T}} = \mathbf{L}(\mathbf{T}, \mathbf{D}) + \mathbf{N}(\mathbf{T}) \|\mathbf{D}\| \quad (9)$$

where $\|\mathbf{D}\|$ stands for the Euclidian norm of $\mathbf{D}$, i.e. $\|\mathbf{D}\| = \sqrt{\text{tr} \mathbf{D}^2}$. A detailed explanation for the preferability of Eq. 9 can be found in [7,15] employing response envelopes.

2.1. A Potential Basic Hypoplastic Model

Equations (6), (7) and (9) lead to numerous possibilities for equations generated with four terms, which is quite common [16] for hypoplastic models. One of these candidates is given in the equation below, but the performance of the model is only investigated for loose and dense Karlsruhe sand since proposing a hypoplastic model is out of the scope of this paper. Test data for triaxial tests at 100 kPa confining pressure are from Kolymbas and Wu (1990) [17]. Experimental and numerical results for comparison are given in Fig. 1 and Fig. 2 for triaxial compression tests.

$$\dot{\mathbf{T}} = c_1 \text{tr}(\mathbf{T} \mathbf{D}) \mathbf{1} + c_2 \text{tr}(\mathbf{D}) \mathbf{T} + c_3 (\mathbf{T} \mathbf{D} + \mathbf{D} \mathbf{T}) + c_4 (\mathbf{T} + \mathbf{T}') \|\mathbf{D}\| \quad (10)$$

2.1.1. Calibration of the Potential Model

The Inexistence of a calibration procedure on papers regarding constitutive modeling is unfortunately common. However, such an absence is beside the point for hypoplasticity research [18, 19]. Moreover, calibration can even be done with the results of only one triaxial test for basic hypoplastic models. Note that this is also valid for the pre-calibration of extended hypoplastic models where additional parameters need to be calibrated for the evolution of additional state variables, e.g. void ratio. The calibration procedure here will be explained via the hypoplastic model given in Eq. 10. Typical results of a triaxial test with two distinct points (e.g. at initial state and state at failure) say A and B, which will be benefited from is given in Fig. 3.

Figure 1. Triaxial compression test for loose Karlsruhe sand: axial strain vs. stress ratio (upper), axial strain vs. volumetric strain (lower)

Figure 2. Triaxial compression test for dense Karlsruhe sand: axial strain vs. stress ratio (upper), axial strain vs. volumetric strain (lower)

Figure 3. Typical results of a triaxial test

The stress and strain rate tensors for a triaxial test, i.e. for axisymmetric conditions, are as follows:

$$\mathbf{T} = \begin{pmatrix} T_1 & 0 & 0 \\ 0 & T_2 & 0 \\ 0 & 0 & T_2 \end{pmatrix}, \quad \mathbf{D} = \begin{pmatrix} D_1 & 0 & 0 \\ 0 & D_2 & 0 \\ 0 & 0 & D_2 \end{pmatrix} \quad (11)$$

For the points A and B, the above tensors have the following form ($T_{1A} = T_{2A}$ for hydrostatic stress state):

$$\mathbf{T}_A = \begin{pmatrix} T_{1A} & 0 & 0 \\ 0 & T_{2A} & 0 \\ 0 & 0 & T_{2A} \end{pmatrix}, \quad \mathbf{D}_A = \begin{pmatrix} D_{1A} & 0 & 0 \\ 0 & D_{2A} & 0 \\ 0 & 0 & D_{2A} \end{pmatrix} \quad (12)$$

$$\mathbf{T}_B = \begin{pmatrix} T_{1B} & 0 & 0 \\ 0 & T_{2B} & 0 \\ 0 & 0 & T_{2B} \end{pmatrix}, \quad \mathbf{D}_B = \begin{pmatrix} D_{1B} & 0 & 0 \\ 0 & D_{2B} & 0 \\ 0 & 0 & D_{2B} \end{pmatrix}$$

As can be seen from Fig. 3, the following values can easily be obtained accordingly:

$$E_A, \quad \beta_A = \arctan\left(\frac{D_{1A} + 2D_{2A}}{D_{1A}}\right), \quad (13)$$

$$\beta_B = \arctan\left(\frac{D_{1B} + 2D_{2B}}{D_{1B}}\right), \quad (T_{1B} - T_{2B})$$

The component D_1 of strain rate tensor should be negative since the considered test is a triaxial compression and the value of D_1 (also equal to D_{1A} and D_{1B}) can be taken equal to 1 owing to the rate-independence of Eq. 10. This leads to the values of D_{2A} and D_{2B} together with the second or third expressions in Eq. 13:

$$\begin{aligned} D_{2A} &= 0.5(\tan\beta_A - 1) \\ D_{2B} &= 0.5(\tan\beta_B - 1) \end{aligned} \quad (14)$$

The angles β_A and β_B are directly related to the Poisson ratios at points A and B (see Eq. 13):

$$\begin{aligned} \tan\beta_A &= \frac{D_{1A} + 2D_{2A}}{D_{1A}} = 1 - 2\nu_A \\ \tan\beta_B &= \frac{D_{1B} + 2D_{2B}}{D_{1B}} = 1 - 2\nu_B \end{aligned} \quad (15)$$

Substituting Eq. 15 in Eq. 14 results in

$$\begin{aligned} D_{2A} &= -\nu_A \\ D_{2B} &= -\nu_B \end{aligned} \quad (16)$$

With D_{2A} and D_{2B} defined, strain rate tensor $\mathbf{D}$ given as the second expression in Eq. 12 can be generated at points A and B as follows:

$$\mathbf{D}_A = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -\nu_A & 0 \\ 0 & 0 & -\nu_A \end{pmatrix} \quad (17)$$

$$\mathbf{D}_B = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -\nu_B & 0 \\ 0 & 0 & -\nu_B \end{pmatrix}$$

Since the vorticity tensor $\mathbf{W}$ is

$$\mathbf{W} = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (18)$$

due to the absence of rotation of principal stress and strain axes. This condition leads to the following equality (see Eq. 2):

$$\dot{\mathbf{T}} = \dot{\mathbf{T}} \quad (19)$$

Eventually, the stress rate tensors at points A and B can be written as:

$$\dot{\mathbf{T}}_A = \begin{pmatrix} E_A & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix} \quad (20)$$

$$\dot{\mathbf{T}}_B = \begin{pmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{pmatrix}$$

Eq. 12 can be substituted in Eq. 10 and the below explicit equation system is consequently obtained:

$$\begin{aligned} \dot{T}_{1A} &= c_1 \text{tr}(\mathbf{T}_A \mathbf{D}_A) + c_2 \text{tr}(\mathbf{D}_A) T_{1A} + c_3 (T_{1A} D_{1A} + D_{1A} T_{1A}) \\ &\quad + c_4 (T_{1A} + T_{1A}^*) \sqrt{\text{tr}(\mathbf{D}_A^2)} \\ \dot{T}_{2A} &= c_1 \text{tr}(\mathbf{T}_A \mathbf{D}_A) + c_2 \text{tr}(\mathbf{D}_A) T_{2A} + c_3 (T_{1A} D_{2A} + D_{2A} T_{1A}) \\ &\quad + c_4 (T_{1A} + T_{1A}^*) \sqrt{\text{tr}(\mathbf{D}_A^2)} \\ \dot{T}_{B1} &= c_1 \text{tr}(\mathbf{T}_B \mathbf{D}_B) + c_2 \text{tr}(\mathbf{D}_B) T_{1B} + c_3 (T_{1B} D_{1B} + D_{1B} T_{1B}) \\ &\quad + c_4 (T_{1B} + T_{1B}^*) \sqrt{\text{tr}(\mathbf{D}_B^2)} \\ \dot{T}_{2B} &= c_1 \text{tr}(\mathbf{T}_B \mathbf{D}_B) + c_2 \text{tr}(\mathbf{D}_B) T_{2B} + c_3 (T_{2B} D_{2B} + D_{2B} T_{2B}) \\ &\quad + c_4 (T_{2B} + T_{2B}^*) \sqrt{\text{tr}(\mathbf{D}_B^2)} \end{aligned} \quad (21)$$

Substituting Eq. 12, 15 and 18 in Eq. 19 leads to

$$\begin{aligned} E_A &= c_1 T_{1A} (-1 - 2\nu_A) + c_2 (-1 - 2\nu_A) T_{1A} + c_3 (-2T_{1A}) \\ &\quad + c_4 T_{1A} \sqrt{1 + 2\nu_A^2} \\ 0 &= c_1 T_{1A} (-1 - 2\nu_A) + c_2 (-1 - 2\nu_A) T_{1A} + c_3 (-\nu_A T_{1A}) \\ &\quad + c_4 T_{1A} \sqrt{1 + 2\nu_A^2} \\ 0 &= c_1 (-T_{1B} - 2T_{2B}\nu_B) + c_2 (-1 - 2\nu_B) T_{1B} + c_3 (-2T_{1B}) \\ &\quad + c_4 (T_{1B} + T_{1B}^*) \sqrt{1 + 2\nu_B^2} \\ 0 &= c_1 \text{tr}(-T_2 - 2T_2\nu_B) + c_2 (-1 - 2\nu_B) T_2 + c_3 (-2\nu_A T_2) \\ &\quad + c_4 (T_2 + T_2^*) \sqrt{1 + 2\nu_B^2} \end{aligned} \quad (22)$$

Eq. 20 can finally be written in the matrix form which can be solved for c_1, c_2, c_3 and c_4 using LU decomposition:

$$\begin{pmatrix} E_A \\ T_{1A} \\ 0 \\ 0 \\ 0 \end{pmatrix} = \begin{pmatrix} -1 - 2\nu_A & -1 - 2\nu_A & -2 & \sqrt{1 + 2\nu_A^2} \\ -1 - 2\nu_A & -1 - 2\nu_A & -\nu_A & \sqrt{1 + 2\nu_A^2} \\ -T_{1B} - 2T_{2B}\nu_B & (-1 - 2\nu_B) T_{1B} & -2T_{1B} & (T_{1B} + T_{1B}^*) \sqrt{1 + 2\nu_B^2} \\ -T_2 - 2T_2\nu_B & (-1 - 2\nu_B) T_2 & -2\nu_A T_2 & (T_2 + T_2^*) \sqrt{1 + 2\nu_B^2} \end{pmatrix} \begin{pmatrix} c_1 \\ c_2 \\ c_3 \\ c_4 \end{pmatrix} \quad (23)$$

3. Numpy-based Calibration

Kolymbas and Wu (1990) [17] performed a series of triaxial tests on loose and dense Karlsruhe sand, which will be used for the calibration routine given in this chapter. Triaxial test results for loose and dense Karlsruhe sand at 100 kPa confining pressure can be seen in Fig. 1 and 2, respectively. Test results for loose and dense Karlsruhe sand are given in Table 1.

Table 1. Triaxial test results for loose and dense Karlsruhe sand

	E_A (MPa)	β_A	β_B	$T_{1B} - T_{2B}$ (kPa)
Loose sand	10	-44.9	0	200
Dense sand	32	-44.91	29.95	470.53

At the very beginning of the calibration program, printing options are edited to get outputs with reasonable precision right after the NumPy library is imported:

```
import numpy as np
np.set_printoptions(precision=3)
np.set_printoptions(suppress=True)
```

Test results in Eq. 13 are asked from the user and T_{1B} component of tensor $\mathbf{T}_B$ is returned:

```
# EA, BetaA, BetaB, (T1-T2)max and confining
# pressure are defined by the user.
# (T1)max (T1B) is calculated => (T1-T2)max +
# confining pressure (t_c).
```

```
ea = float(input('Enter the initial value of\
tangential stiffness \'EA\' : '))
beta_a = float(input('Enter the value of Beta at\
point A \'BetaA\' : '))
beta_b = float(input('Enter the value of Beta at\
point B \'BetaB\' : '))
t1_t2_max = float(input('Enter the peak\
deviatoric stress \'(T1-T2)\' : '))
t_c = float(input('Enter the confining\
pressure in kPa : '))
t1_max = t1_t2_max + t_c
```

Since NumPy uses radians but not degrees, β_A and β_B are converted to radians from degrees and Poisson ratios at points A and B are calculated according to Eq. 15:

```
beta_a = np.deg2rad(beta_a)
beta_b = np.deg2rad(beta_b)
v_a = 0.5*(np.tan(beta_a)-1)
v_b = 0.5*(np.tan(beta_b)-1)
```

The stress and strain rate tensors (Eq. 12 and 15) at points A and B are generated as

```
t_a = np.array([[ -t_c, 0., 0.],[0., -t_c, 0.],\
[0., 0., -t_c]])
t_b = np.array([[ -t1_max, 0., 0.],[0., -t_c, 0.],\
[0., 0., -t_c]])
d_a = np.array([[ -1., 0., 0.],[0., -v_a, 0.],\
[0., 0., -v_a]])
d_b = np.array([[ -1., 0., 0.],[0., -v_b, 0.],\
[0., 0., -v_b]])
```

which lead to stress and strain rate tensors for loose sand such that

$$\mathbf{T}_A = \begin{pmatrix} -100. & 0 & 0 \\ 0 & -100. & 0 \\ 0 & 0 & -100. \end{pmatrix}, \quad \mathbf{D}_A = \begin{pmatrix} -1. & 0 & 0 \\ 0 & 0.002 & 0 \\ 0 & 0 & 0.002 \end{pmatrix}$$

$$\mathbf{T}_B = \begin{pmatrix} -300. & 0 & 0 \\ 0 & -100. & 0 \\ 0 & 0 & -100. \end{pmatrix}, \quad \mathbf{D}_B = \begin{pmatrix} -1. & 0 & 0 \\ 0 & 0.5 & 0 \\ 0 & 0 & 0.5 \end{pmatrix}$$

and for dense sand as

$$\mathbf{T}_A = \begin{pmatrix} -100. & 0 & 0 \\ 0 & -100. & 0 \\ 0 & 0 & -100. \end{pmatrix}, \quad \mathbf{D}_A = \begin{pmatrix} -1. & 0 & 0 \\ 0 & 0.002 & 0 \\ 0 & 0 & 0.002 \end{pmatrix}$$

$$\mathbf{T}_B = \begin{pmatrix} -470.53 & 0 & 0 \\ 0 & -100. & 0 \\ 0 & 0 & -100. \end{pmatrix}, \quad \mathbf{D}_B = \begin{pmatrix} -1. & 0 & 0 \\ 0 & 0.788 & 0 \\ 0 & 0 & 0.788 \end{pmatrix}$$

Constitutive Eq. 10 can be split into four terms, which will make ease to form matrices in Eq. 23:

Constitutive equation term by term

```
kron = np.identity(3) # Kronecker delta
term1a = np.trace(t_a.dot(d_a))*kron # tr(TD)1
term2a = np.trace(d_a)*t_a # tr(D)T
term3a = np.tensordot(t_a,d_a, axes=1)+\
np.tensordot(d_a,t_a,axes=1) # TD + DT
term4a = (t_a + t_a -1./3.*np.trace(t_a)*kron)*\
np.sqrt(np.trace(np.tensordot(d_a,d_a,\
axes=1))) # (T+T)||D||

term1b = np.trace(t_b.dot(d_b))*kron # tr(TD)1
term2b = np.trace(d_b)*t_b # tr(D)T
term3b = np.tensordot(t_b,d_b, axes=1)+\
np.tensordot(d_b,t_b,axes=1) # TD + DT
term4b = (t_b + t_b -1./3.*np.trace(t_b)*kron)*\
np.sqrt(np.trace(np.tensordot(d_b,d_b,\
axes=1))) # (T+T)||D||
```

Note that editing the above 8 lines of code is sufficient for calibration of other basic hypoplastic models. Relevant components of the above-generated terms can be gathered as the first matrix (so-called A) of the right-hand side in Eq. 23:

```
a = np.array([[term1a[0,0],term2a[0,0],\
term3a[0,0],term4a[0,0]],term1a[1,1],\
term2a[1,1],term3a[1,1],term4a[1,1]],\
[term1b[0,0],term2b[0,0],term3b[0,0],\
term4b[0,0]],term1b[1,1],term2b[1,1],\
term3b[1,1],term4b[1,1]])
```

leading to the matrix (say A_l) for loose sand as

$$A_l = \begin{pmatrix} 99.652 & 99.652 & 200. & -100. \\ 99.652 & 99.652 & -0.348 & -100. \\ 200 & 0 & 600 & -530.723 \\ 200 & 0 & -100. & -40.825 \end{pmatrix}$$

and to the matrix A_d for dense sand:

$$A_d = \begin{pmatrix} 99.686 & 99.686 & 200. & -100. \\ 99.686 & 99.686 & -0.314 & -100. \\ 312.911 & -271.113 & 941.06 & -1074.454 \\ 312.911 & -57.619 & -157.619 & 35.204 \end{pmatrix}$$

The left-hand side matrix of Eq. 23 is formed as follows:

```
b = np.array([ea/t_a[0,0], 0., 0., 0.]
```

Finally, Eq. 23 can be solved using the below linear algebra function of NumPy based on LU decomposition:

```
c = np.linalg.solve(a,b)
```

which leads to the material parameters given in Table 2.

Table 2. Material parameters obtained with NumPy for loose and dense Karlsruhe sand

	c_1	c_2	c_3	c_4
Loose sand	-39.514	-32.229	-49.913	-71.319
Dense sand	-76.87	-68.985	-159.749	-144.986

4. Conclusions

Utilization of open-source programming languages as well as open-source software being outcomes of these languages is actual in doing science. Python is an interpreted, high-level and object-oriented programming language widely used for academic purposes. For instance, open-source projects such as FeniCS[20] and Yade[21] have been developed via Python in conjunction with C++. This influence of programming languages in doing science will undoubtedly be extant. Correspondingly, it has been considered necessary to introduce an open-source programming language aided calibration of a basic hypoplastic model. The outcome is a code with less than 40 lines, which can be easily modified for other basic hypoplastic models by changing 8 lines of the code. This code is a part of ongoing research in which it is expected to extend to a more comprehensive program that simulates element tests to facilitate research for potential hypoplastic models.

Nomenclature

$\dot{T}$	: Jaumann stress rate tensor
T	: Cauchy's Stress tensor
$\dot{T}$	: Material time derivative of stress tensor
T_i	: Stress tensor at point i
D	: Strain rate tensor
D_i	: Strain rate tensor at point i
W	: Vorticity tensor
$L(.)$	: Linear tensorial function of its arguments
$N(.)$	: Nonlinear tensorial function of its arguments
Q	: An orthogonal tensor
$\ D\ $	: Euclidian norm of strain rate tensor
T^*	: Deviatoric part of the stress tensor
T_i	: i-th (ie. ii-th) component of stress tensor
D_i	: i-th (ie. ii-th) component of strain rate tensor
E_i	: Tangential stiffness at point i
β_i	: Angle between volumetric and axial strain at point i
ν_i	: Poisson ratio at point i

References

- [1] Puzrin, A.M., Constitutive Modelling in Geomechanics. Springer Berlin Heidelberg: Berlin, Heidelberg (2012).
- [2] Truesdell, C., The mechanical foundations of elasticity and fluid dynamics. *Journal of Rational Mechanics and Analysis* 1 (1952) 125-300.
- [3] Truesdell, C., Hypo-elasticity. *Journal of Rational Mechanics and Analysis* 4 (1955) 83-1020.
- [4] Valanis, K.C., A theory of viscoplasticity without a yielding surface. *Arch. Mech. Stos.* 23(4) (1971) 517-533.
- [5] Kolymbas, D., A rate-dependent constitutive equation for soils. *Mechanics Research Communications* 4:6 (1977) 367-372.
- [6] Kolymbas, D., A novel constitutive law for soils. *Proc. Int. Conf. on Constitutive Laws, Tucson* (1987) 319-323.
- [7] Wu, W., Kolymbas, D., Hypoplasticity then and now. In *Constitutive modelling of granular materials*. Springer, Berlin, Heidelberg (2000) 57-105.
- [8] Wu, W., Kolymbas, D., Numerical testing of the stability criterion for hypoplastic constitutive equations, *Mechanics of Materials* 9(3) (1990) 245-253.
- [9] Truesdell, C., Noll, W., The nonlinear field theories of mechanics, in S. Flugge (ed.) *Encyclopedia of Physics*, III/1, Springer, Berlin (1965).
- [10] Gudehus, M., True triaxial tests on dense sand, in: Gudehus, Darve and Vardoulakis (eds), *Proc. Int. Workshop on Constitutive Relations for Soils* (1984) 11-54.
- [11] Kolymbas, D., Wu, W. Introduction to Hypoplasticity. Modern approaches to plasticity. Kolymbas, D. (ed) Amsterdam: Elsevier (1993).
- [12] Smith, G.F., On isotropic functions of symmetric tensors, skew-symmetric tensors and vectors. *International Journal of Engineering Science*, 9(10) (1971) 899-916.
- [13] Wang, C.C., A new representation theorem for isotropic functions, parts I and II, *J. Rat. Mech. Anal.* 36 (1970) 166-223.
- [14] Gudehus, G., A comparison of some constitutive laws for soils under radially symmetric loading and unloading, in: Wittke (ed), *Proc. 3rd Int. Conf. on Numer. Methods in Geomech.*, (1979) 1309-1323.
- [15] Wu, W., Hypoplastizität als mathematisches Modell zum mechanischen Verhalten granularer Stoffe, *Universitat Karlsruhe, Veröffentlichung des Institutes für Bodenmechanik der Universität Fridericiana zu Karlsruhe, Heft 129* (1992).
- [16] Kolymbas, D., Computer-aided design of constitutive laws. *International Journal for Numerical and Analytical Methods in Geomechanics*, 15(8) (1991) 593-604.
- [17] Kolymbas, D., Wu, W., Recent results of triaxial tests with granular materials. *Powder Technology*, 60(2) (1990) 99-119.
- [18] Wu, W., Bauer, E., A simple hypoplastic constitutive model for sand. *International Journal for Numerical and Analytical Methods in Geomechanics*, 18(12) (1994) 833-862.
- [19] Kolymbas, D., Introduction to Hypoplasticity: Advances in Geotechnical Engineering and Tunnelling 1 (Vol. 1). CRC Press (1999).
- [20] <https://fenicsproject.org>
- [21] <https://yade-dem.org/doc/>