Chapter 3

(Linear Elastic Finite-Element Remeshing Algorithm)

### 3.1 Introduction

Many aspects of geotechnical engineering involve penetration into soil, including the cone penetration test (CPT) which measures a nearly continuous data profile as a cone is advanced through the soil profile. Figure 3.1 shows an example of CPT-measured cone tip resistance, sleeve friction, and pore water pressure (at the $u_2$ position) data profiles. These data are interpreted to estimate engineering soil properties or soil type. However, existing interpretation methods are predominantly empirically based and are therefore limited to a narrow range of soil types and conditions.

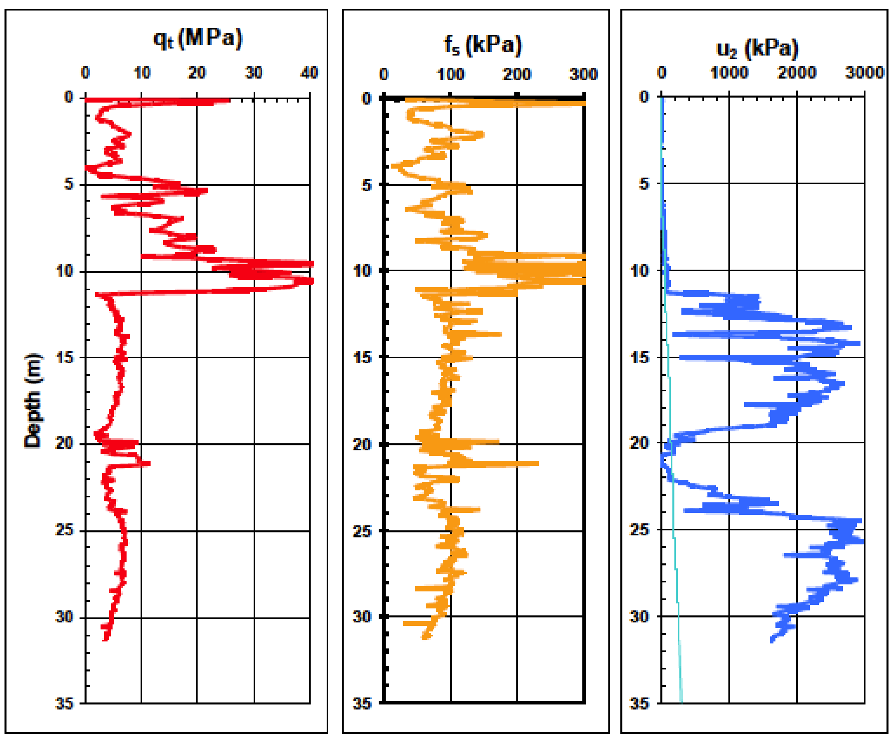


**Figure 3.1**: Example of CPT-measured site profile

Cone penetration simulations have led to more reliable, theory-based characterization methods of engineering properties. Advanced numerical analysis of penetration is challenging due to large deformations around the penetrometer that lead to severe element distortion and numerical instability. Large deformations have been overcome with Arbitrary Lagrangian Eulerian (ALE) algorithms where deformed model geometry (from Lagrangian calculations) is remeshed to a less deformed geometry, then model properties are remapped from the old mesh onto the new mesh with Eulerian calculations. A schematic of the ALE algorithm with a cone penetration model is illustrated in Figure 3.2. Published numerical analyses of cone penetration are still limited by either (a) using simple soil models that idealize soil behavior, (b) not capturing changing conditions with depth, or (c) idealizing soil as isotropic and uniform. Advancing CPT interpretation methods requires simulations of full penetration conditions with depth and with an advanced soil model.

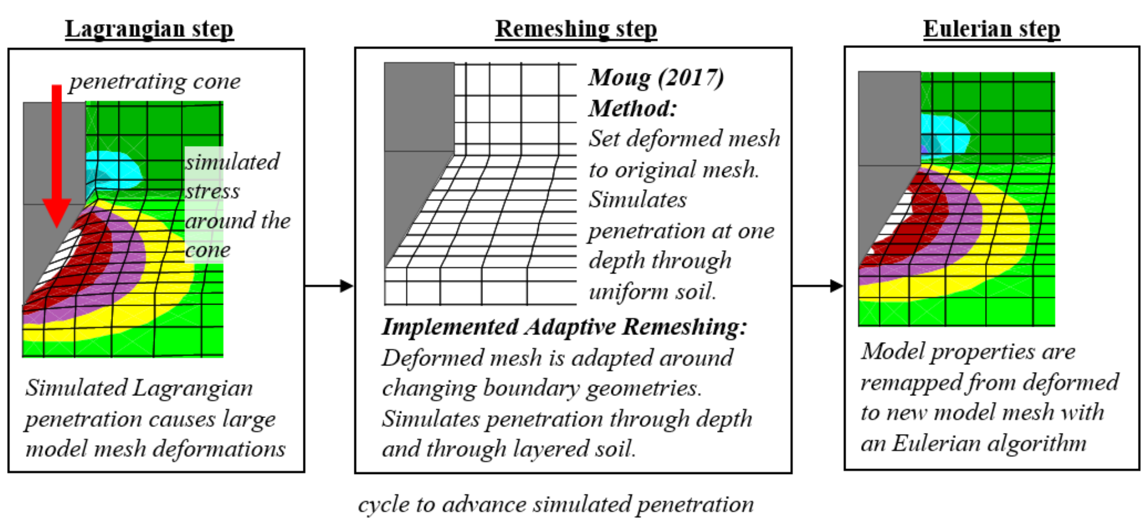


**Figure 3.2**: Schematic illustrating the ALE algorithm with adaptive remeshing

Moug (2017) presented a steady-state, ALE numerical model for simulating cone penetration at a single depth using the MIT-S1 constitutive model. MIT-S1 is a complex constitutive model that captures clayey and silty behavior well, including anisotropic loading of clay, which is especially important because of the complex anisotropic stress conditions that exist around the cone (Moug, et al. 2019). This numerical model accurately reflects complex soil behavior.

However, the model is limited to simulating conditions at a single depth and unable to model behavior over a range of depths as the cone advances through a soil profile. Also, the model is limited to modeling conditions within a uniform soil and unable to represent non-uniform soils, such as an interlayered soil profile. These limitations result from the implemented remeshing step, which resets the deformed mesh to the original undeformed mesh before performing Eulerian remapping calculations.

To overcome these limitations, an algorithm for an adaptive remeshing step is presented. In this remeshing step boundary nodes (e.g. nodes that delineate soil-structure or soil unit interfaces) are tracked, and interior node positions are systematically adapted based on linear-elastic relationships to boundary node displacements. This chapter presents the approach for adaptive remeshing, its verification, describes its implementation with the FLAC cone penetration model, and proposes future work for its application.

### 3.2 Adaptive remeshing approach

This adaptive remeshing step is formulated as an axisymmetric, linear elastic finite-element problem with known displacements at the boundary nodes of the grid: the displacement of the interieror nodes is determined based on linear-elastic relationships to displacement of the boundary nodes. The FLAC model grid node coordinates prior to deformation are used to define the finite-element model geometry, where each FLAC zone is treated as an element. From this geometry and assigned material stiffness properties, the axisymmetric grid's stiffness matrix is defined, according to standard axisymmetric finite-element equations. Note that these linear elastic material properties are assigned for the adaptive remeshing only and do not represent actual soil properties. The FLAC grid is represented by the linear system: $K\epsilon = \sigma$, where $K$ is the stiffness matrix, $\epsilon$ is a vector of radial and axial strains at each node, and $\sigma$ is a vector of radial and axial stresses at each node.

This linear system presents an inhomogeneous strong boundary condition problem in which $\epsilon$ is partially known (at the boundary nodes) and partially unknown (at the interior nodes), and similarly $\sigma$ is partially known (zero at the interior nodes) and partially unknown (at the boundary nodes). The solution for this system is obtained by applying the algorithm described by Bangerth (2013).

After solution, the strains at the interior nodes are known, from which the displacements at the interior nodes are calculated. The calculated displacements at the interior nodes are used to calculate their adapted coordinates. The adapted boundary nodes and interior nodes form the remeshed grid. The coordinates of this remeshed grid are read by FLAC to proceed with the Eulerian remapping step of the ALE cycle.

To illustrate the solution algorithm, consider solving the equation $Ku = f$ for the example finite-element mesh in Figure 3.3, where $K$ is the system's stiffness matrix, $u$ is a vector containing the nodal displacements, and $f$ is a vector containing the nodal forces. Assume that the displacements at the outer nodes (i.e. nodes 0, 2, 3, 4, 7, 8, and 9) are known, and the displacements at the interior nodes (i.e. nodes 1, 5, and 6) are unknown. Conversely, the forces at the exterior nodes are unknown, and the forces at the interior nodes are known to be zero.

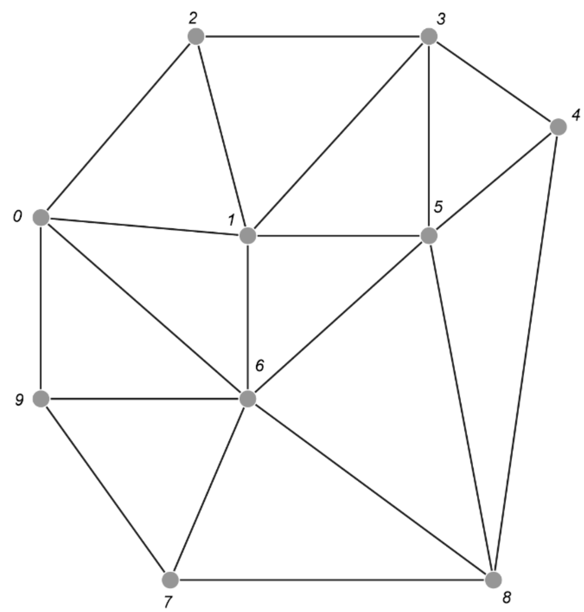


**Figure 3.3**: Example finite-element mesh with 10 nodes

Let:

$$G = \begin{bmatrix} g(x_0) \\ 0 \\ g(x_2) \\ g(x_3) \\ g(x_4) \\ 0 \\ 0 \\ g(x_7) \\ g(x_8) \\ g(x_9) \end{bmatrix}, \text{ where } g(x_i) = u_i.$$

The system to be solved is represented by equation 3.1:

$$\begin{bmatrix} k_{00} & k_{01} & k_{02} & k_{03} & k_{04} & k_{05} & k_{06} & k_{07} & k_{08} & k_{09} \\ k_{10} & k_{11} & k_{12} & k_{13} & k_{14} & k_{15} & k_{16} & k_{17} & k_{18} & k_{19} \\ k_{20} & k_{21} & k_{22} & k_{23} & k_{24} & k_{25} & k_{26} & k_{27} & k_{28} & k_{29} \\ k_{30} & k_{31} & k_{32} & k_{33} & k_{34} & k_{35} & k_{36} & k_{37} & k_{38} & k_{39} \\ k_{40} & k_{41} & k_{42} & k_{43} & k_{44} & k_{45} & k_{46} & k_{47} & k_{48} & k_{49} \\ k_{50} & k_{51} & k_{52} & k_{53} & k_{54} & k_{55} & k_{56} & k_{57} & k_{58} & k_{59} \\ k_{60} & k_{61} & k_{62} & k_{63} & k_{64} & k_{65} & k_{66} & k_{67} & k_{68} & k_{69} \\ k_{70} & k_{71} & k_{72} & k_{73} & k_{74} & k_{75} & k_{76} & k_{77} & k_{78} & k_{79} \\ k_{80} & k_{81} & k_{82} & k_{83} & k_{84} & k_{85} & k_{86} & k_{87} & k_{88} & k_{89} \\ k_{90} & k_{91} & k_{92} & k_{93} & k_{94} & k_{95} & k_{96} & k_{97} & k_{98} & k_{99} \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ u_2 \\ u_3 \\ u_4 \\ u_5 \\ u_6 \\ u_7 \\ u_8 \\ u_9 \end{bmatrix} = \begin{bmatrix} f_0 \\ f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \end{bmatrix}. \tag{3.1}$$

Then, an equivalent solution of equation 3.1 is obtained by using standard methods to bring unknowns to the left side and to solve the system in equation 3.2:

$$\begin{bmatrix} k_{11} & k_{15} & k_{16} \\ k_{51} & k_{55} & k_{56} \\ k_{61} & k_{65} & k_{66} \end{bmatrix} \begin{bmatrix} u_1 \\ u_5 \\ u_6 \end{bmatrix} = \begin{bmatrix} f_1 - \sum\limits_{i \in \{0, 2, 3, 4, 7, 8, 9\}} k_{1i} g(x_i) \\ f_5 - \sum\limits_{i \in \{0, 2, 3, 4, 7, 8, 9\}} k_{5i} g(x_i) \\ f_6 - \sum\limits_{i \in \{0, 2, 3, 4, 7, 8, 9\}} k_{6i} g(x_i) \end{bmatrix}. \tag{3.2}$$

### 3.3 Verification of adaptive remeshing algorithm

The linear elastic finite-element adaptive remeshing algorithm was implemented in Python. To verify the implemented finite-element solution, two linear elastic problems—unconfined axial compression and radial compression—with Poisson's ratio, $\nu$, = 0.3 and bulk modulus, K, = 10000 kPa were solved via FLAC and the finite-element solution, and the solutions were compared. Each of the deformed grids are plotted together in Figures 3.4 and 3.5 for unconfined axial compression and radial compression, respectively. As shown in Figures 3.4 and 3.5 the solutions exhibit a high degree of agreement, which indicates that the finite-element solution is properly implemented.

Note that the FLAC grid consists of rectangular zones defined by four nodes, but the finite-element formulation divides each of these zones into four triangles whose common vertex is in the middle of the rectangular zone. So, the finite-element representation of one FLAC zone, defined by four points, is defined by five points. This is the reason for the extra nodes within the FLAC zones in Figures 4 and 5.

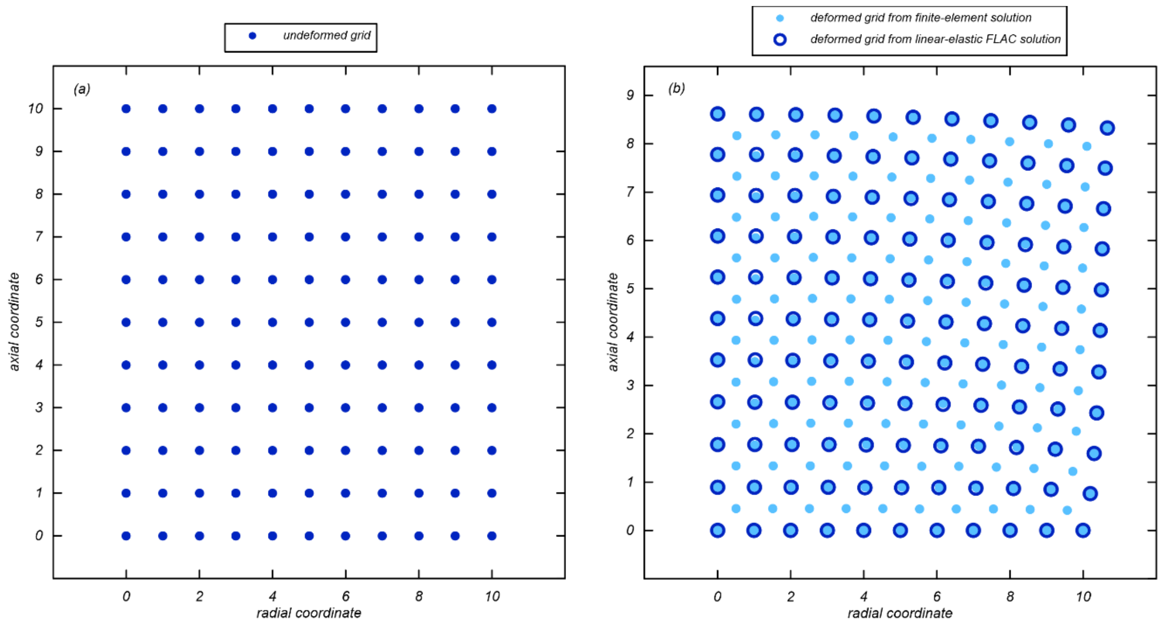


**Figure 3.4**: Comparison of undeformed grid coordinates and deformed (by unconfined axial compression) grid coordinates, obtained by solution in FLAC and the remeshing finite-element solution

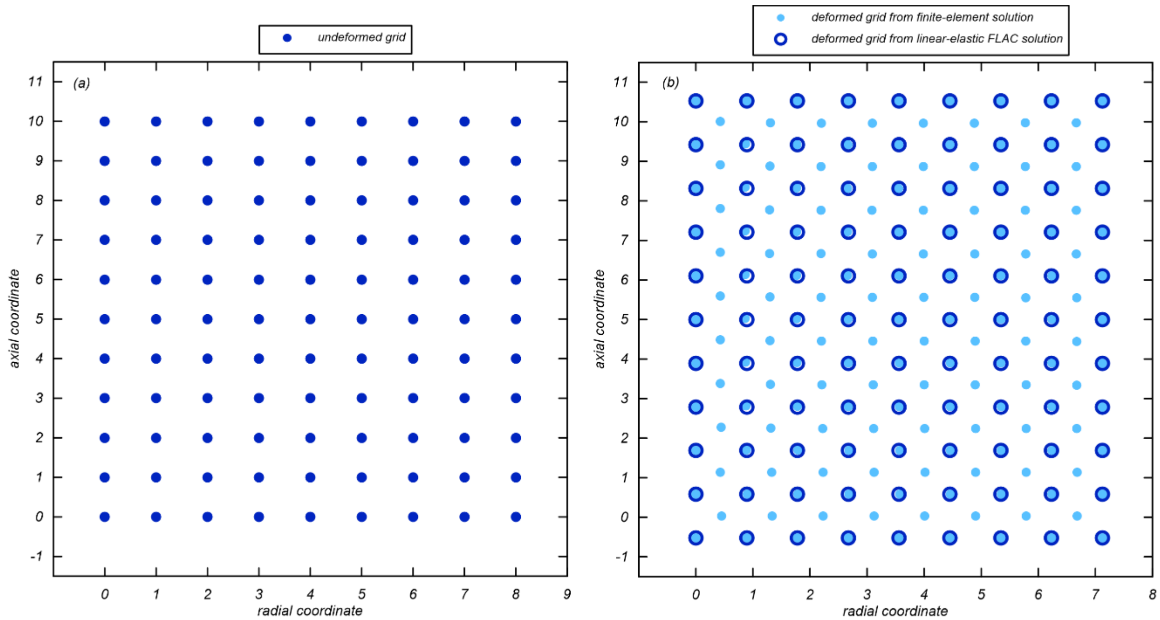


**Figure 3.5**: Comparison of undeformed grid coordinates and deformed (by radial compression) grid coordinates, obtained by solution in FLAC and the remeshing finite-element solution

### 3.4 Implementation with FLAC cone penetration model

The adaptive remeshing algorithm incorporates into the FLAC direct cone penetration model presented in Moug (2017). The direct penetration model with an ALE algorithm in Moug (2017) follows the steps of (1) simulate direct penetration for a set distance – now the geometry is considered deformed due to Lagrangian deformations, (2) reset gridpoints to the initial, undeformed geometry, and (3) remap model properties from the deformed to undeformed geometry with an Eulerian convective algorithm. This model simulates cone penetration by holding the cone at one point in the soil column, and simulating soil as flowing up past the stationary cone. The outcome is a steady state cone penetration model that simulates the stress, porewater pressure, and strain around a cone at one point in the soil column.

The adaptive remeshing algorithm is implemented into the original algorithm instead of step (2). The adaptive remeshing algorithm determines the adjustments to deformed gridpoints based on the deformations at boundary gridpoints. This will allow cone penetration to be simulated as a cone penetrating through the soil column, and across material boundaries (i.e., soil layers).

This implementation requires running the Python-implemented adaptive remeshing algorithm along with the FLAC cone penetration model. The "Axisymmetric_Displacement_Solution.py" file should be located in the same folder as the cone penetration model is run from. The cone penetration model with adaptive remeshing is run with the following steps:

(i) Run the Python executable file "Axisymmetric_Displacement_Solution.py" from Spyder.

(ii) Run the cone penetration model executable "CPT_ALE_adapt.fis" from a FLAC project.

These two steps should run the new cone penetration model that has been adjusted to integrate adaptable remeshing. The pseudo algorithm is outlined as:

- Initialize model geometry and initial conditions
- Solve to ensure the model is at static conditions
- While time is less than total solve time:
    - Write "undeformed" model coordinates to a .txt file "write_coordinates.txt"
    - Simulate cone penetration for a set penetration distance
    - Write "deformed" model coordinates to a .txt file "new_boundary_coordinates.txt". This file is written to have a column with

a flag = 1. This file is being continuously checked in the Python routine. If flag = 0, Python continues to check; if flag = 1, Python carries out the adaptive remeshing routine.

- Pause FLAC execution to let the Python script execute.

  *Note: the amount of time of the pause will depend on the computer speed and size of the model. We have found 20 seconds to be sufficient in initial simulations.*

- Python solves for internal gridpoint displacements based on "new_boundary_coordinates.txt" and assigned linear elastic material properties.
- Python outputs "adapted_coordinates.fis" file, and FLAC assigns new geometry from these coordinates.
- Remap model properties from deformed to adapted model geometry with Eulerian algorithms (Moug 2017).

### 3.5 Conclusions & Next Steps

This chapter describes the implementation of an adaptive remeshing scheme for an ALE algorithm. The algorithm is developed to be implemented with a direct axisymmetric cone penetration model in FLAC. The adaptive remeshing algorithm was validated by comparing its results of solving gridpoint displacements with results from a linear elastic solution in FLAC.

The next objective is to implement the adaptive remeshing approach in simulating cone penetration and validate this model. After validation, the model can be applied to model continuous cone penetration. A particularly interesting application of the new model would be modeling cone penetration through an interlayered soil profile to investigate thin-layer effects and behavior at and across soil layer interfaces.