DEVELOPMENT OF A MATLAB FINITE ELEMENTS CODE FOR PLASTICITY: IMPLEMENTATION OF THE MOHR-COULOMB LAW

Gildas Yaovi AMOUZOU and Azzeddine SOULAIMANI

ABSTRACT

This work deals with the development of a two-dimensional (2D) elastoplastic finite elements code based on the Mohr-Coulomb law. This law remains the simplest and most used by engineers for geotechnical studies. For this purpose, two new numerical methods are developed to solve elastoplastic problems are proposed. The first deals with the resolution of the return-mapping algorithm and is based on a fixed point algorithm. In this method, the system of nonlinear equations for plasticity is rewritten in a form adapted to this fixed point algorithm which is converges in a few iterations. The second method relates to the computation of the consistent tangent matrix for elastoplasticity using the finite difference method and calling twice the return mapping algorithm. It is known that the analytical evaluation of the tangent matrix is usually very tedious, but with algorithm this difficulty is virtually avoided. A first validation test is performed by application to a uniaxial bar problem. The results obtained show that the numerical algorithms we developed prove to very efficient. Then, the algorithms are generalized to 2D plane strain. Also, an explicit form of the elastoplastic consistent tangent for the Mohr-Coulomb law has been developed and implemented for the purpose of comparison. An application to a rockfill dam has been conducted and the results obtained using the numerical and analytical algorithms are very close. Also, comparisons with the results obtained via the commercial software PLAXIS validate the proposed methods. It should be noted that the code developed (in Matlab) offers the possibility of simulating the layer by layer dam construction phases. With the developed algorithms, it is relatively easy to integrate other constitutive laws including the Hardening Soil Model law. Also, having the source code, parameters calibration and uncertainties analyses can be easily performed by coupling our code with existing optimization functions such those implemented in Matlab.