Computers and Geotechnics 65 (2015) 12–29

Contents lists available at ScienceDirect

Computers and Geotechnics

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

Research Paper

A new rock slicing method based on linear programming

C.W. Boon [a], G.T. Houlsby [a], S. Utili [b,*]

[a] University of Oxford, Department of Engineering Science, Parks Road, Oxford OX1 3PJ, United Kingdom
[b] University of Warwick, School of Engineering, Coventry CV4 7AL, United Kingdom [1]

ARTICLE INFO

Article history:
Received 5 June 2014
Received in revised form 10 October 2014
Accepted 13 November 2014

Keywords:
Rock slicing
Block generation
Distinct element method
Sequential subdivision
Rock mechanics
Linear programming

ABSTRACT

One of the important pre-processing stages in the analysis of jointed rock masses is the identification of rock blocks from discontinuities in the field. In 3D, the identification of polyhedral blocks usually involve tedious housekeeping algorithms, because one needs to establish their vertices, edges and faces, together with a hierarchical data structure: edges by pairs of vertices, faces by bounding edges, polyhedron by bounding faces.

In this paper, we present a novel rock slicing method, based on the subdivision approach and linear programming optimisation, which requires only a single level of data structure rather than the current 2 or 3 levels presented in the literature. This method exploits the novel mathematical framework for contact detection introduced in Boon et al. (2012). In the proposed method, it is not necessary to calculate the intersections between a discontinuity and the block faces, because information on the block vertices and edges is not needed. The use of a simpler data structure presents obvious advantages in terms of code development, robustness and ease of maintenance. Non-persistent joints are also introduced in a novel way within the framework of linear programming. Advantages and disadvantages of the proposed modelling of non-persistent joints are discussed in this paper. Concave blocks are generated using established methods in the sequential subdivision approach, i.e. through fictitious joints.

1. Introduction

Jointed rock masses are made up from numerous polyhedral rock blocks, whose faces are cut out by discontinuities in the rock field. The spatial distribution, size and orientation of these discontinuities are rarely regular and usually follow probabilistic distributions. As a result, the size and shape of each block in the jointed rock mass are different. For the purpose of distinct element modelling (DEM) or discontinuous deformation analysis (DDA), one has to invest significant effort to identify polyhedral blocks from the discontinuities (see Fig. 1), whose orientations are typically defined using their dip directions and dip angles (see Fig. 2).

Broadly, there exist two approaches in block generation algorithms. The first approach is based on subdivision, in which discontinuities are introduced sequentially [47,19,48,53]. Each discontinuity is introduced one-at-a-time (see Fig. 3a). If a discontinuity intersects a block, the parent block is subdivided into a pair of so-called child blocks. This process is repeated until all the discontinuities are introduced. The number of blocks increases as more "slices" are introduced, and a data structure of every block is maintained throughout the slicing process. The blocks generated through sequential subdivision are convex because a discontinuity has to terminate at the face of a neighbouring block. Concave blocks can, nonetheless, be generated through the use of clustering, which can be automated [48] or guided by specifying fictitious construction joints [47]; Fig. 4). Blocks subdivided by a construction joint are clustered together by imposing a kinematic constraint which prevents any relative movement between the two sides of the joint. Likewise, non-persistent joints, i.e. joints of finite sizes [14,51], can be modelled through clustering, specifying fictitious construction joints, or subdomains [19]; see Fig. 5). This is discussed again in further detail in a later paragraph. On the other hand, in the second approach ('face-tracing' based on simplicial homology theory), discontinuities are introduced all-at-once (see Fig. 3b). All the vertices and edges in the domain are first calculated from the intersections between the discontinuities. From these vertices and edges, there are ways by which the faces and polyhedra in the rock mass can be identified [33,23,28,34]. The necessary algorithms are, however, rather complex. The advantage of this approach is that convex and concave blocks are identified in the same manner. Non-persistent joints and dangling joints (see Fig. 6), i.e. joints which terminate inside intact rock without

* Corresponding author.
E-mail address: s.utili@warwick.ac.uk (S. Utili).
[1] Formerly at University of Oxford.

http://dx.doi.org/10.1016/j.compgeo.2014.11.007
0266-352X/

contributing to block formation [28], are also treated in the same manner as persistent joints, i.e. joints of infinite size. Depending on the type of mechanical analysis which is to be performed on the generated rock mass, these dangling joints may have to be removed; for instance, they have to be removed if either the distinct element method [10,24,25] or discontinuous deformation analysis [42] is used later on for analysis; but they do not need to be removed if fracturing has to be modelled, for instance employing the discrete-finite element method [38]. A summary of the two approaches is shown in Table 1.

This paper is about the sequential subdivision approach. In the case of a complex 3-D jointed rock mass, the generation of polyhedral blocks requires tedious and algorithmically complex updates of the data structure which is used to encapsulate the significant geometrical features of the mass. The number of faces, edges and vertices of the polyhedra in the jointed rock mass is unknown to the modeller, and they become known only at the end of the rock slicing procedure. Therefore, during block generation, the management of this triple-level data structure (faces, edges and vertices) requires careful implementation in a numerical code. Since computing resources, e.g. computing time and memory, is rarely a major concern in rock slicing algorithms by comparison to the simulation runtime of the physical problem considered (e.g. underground excavations, stability analysis of rock slopes, etc.), the choice of code implementation is dictated by factors such as the time needed for code development, ease of code maintenance, and robustness. Algorithms based on the subdivision approach are mainly concerned about the updating of the data structure every time a block is subdivided. A triple-level and a double-level hierarchical data structure have been proposed by Warburton [47] and Heliot [19] respectively for their rock slicing algorithms (see Fig. 7). In Warburton [47], the flow of the algorithm proceeds as follows: (i) intersections (new vertices) are identified and old edges are subdivided, (ii) new edges are identified from the old faces which cross the joint plane and also from their edges which cross the joint plane (not every pair of new vertices can form a new edge), (iii) faces and other data structure for the child blocks are updated (see Fig. 7a). Most of the algorithms proposed recently (e.g. [48] make use of the data structure proposed by Heliot [19]. In Heliot [19], every face of a polyhedron is indexed, and a vertex is assumed to result from the intersection of three planes (see Fig. 7b). Each vertex therefore consists of three indices. An intersection check is performed for every pair of vertices which have two indices in common (e.g. between vertex-146 and vertex-346). New vertices are created from the intersection, and their indices are identified. Old vertices are allocated to the new child blocks depending on whether they are on the positive or negative halfspace. The lists of faces and vertices are rebuilt for each child blocks.

The level of housekeeping (or bookkeeping) algorithms, which is required in a block generation computer code, depends on the choice of data structures. Heliot [19] has, for instance, made bookkeeping more manageable by reducing the original three-level data structure [47] to a two-level data structure consisting of only vertices and faces (see Fig. 7). In the rock slicing method presented in this paper, only a single data structure consisting of the block faces is used. It will be shown that this novel procedure makes block generation algorithmically simpler and numerically more robust. Whilst it is necessary to establish whether there is intersection between a block and a discontinuity, the exact intersections between the discontinuity and the block faces need not be calculated in our method. In other words, information on block vertices and edges are not necessary, so there is no longer the need to maintain a complex hierarchical data structure, and problems arising from rounding errors in the case of high vertex density can be avoided (c.f. [15]). According to the proposed novel mathematical treatment based on convex optimisation, the block faces of a polyhedron are defined by linear inequalities, the equation of a joint plane is defined by a linear equality constraint, and the geometrical boundary of a non-persistent joint by linear inequalities. Given a non-persistent joint and a polyhedron which are potentially intersecting, we establish whether there is actual intersection by checking if the optimisation problem defined by the linear equality constraint for the joint plane, the inequality constraints for the geometrical boundary of the non-persistent joint, and the inequality constraints for the polyhedron is feasible (i.e. whether the convex set is not empty). The problem is feasible if there is a point lying inside the interior region defined by the linear inequalities and at the same time satisfying the linear equality constraint [8], and not feasible if otherwise. To ascertain the existence of such a point, i.e. whether the problem is feasible, a linear program is run (illustrated in Section 2.3) to find the point with the largest

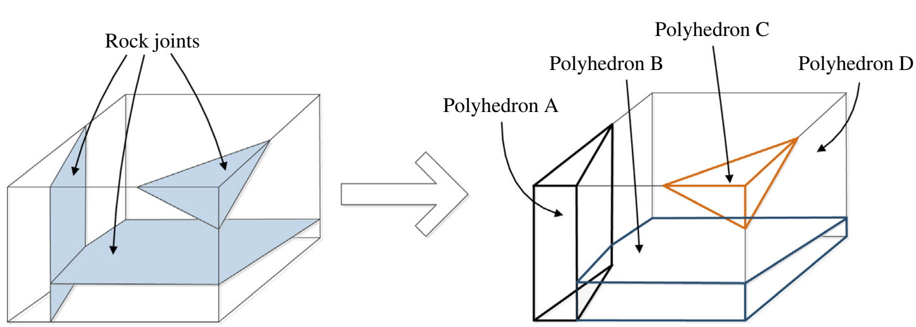

Fig. 1. Illustration of a simple set of rock slices, resulting in polyhedral rock blocks.

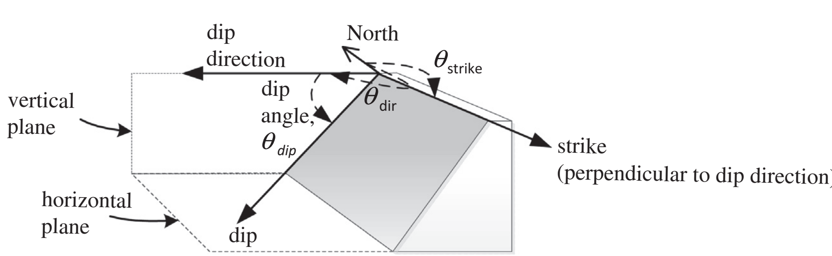

Fig. 2. Definition of strike, dip and dip direction according to Hoek et al. [20].