International Journal of Rock Mechanics & Mining Sciences 86 (2016) 80–90

Contents lists available at ScienceDirect

International Journal of Rock Mechanics & Mining Sciences

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

Identifying rock blocks based on exact arithmetic

Yinhe Zheng, Lu Xia, Qingchun Yu *

School of Water Resources and Environment, China University of Geosciences, Beijing 100083, China

ARTICLE INFO

Article history:
Received 9 November 2015
Received in revised form
28 January 2016
Accepted 30 March 2016
Available online 14 April 2016

Keywords:
Rock block identification
Exact arithmetic
Geometrical errors
Element block

ABSTRACT

This paper presents a block identification method for jointed rock masses that is based on exact arithmetic. Block structures are constructed accurately without introducing rounding errors, and therefore the robustness of the block identification algorithm is guaranteed. A rational number type is defined and the basic arithmetic operations of the rational numbers are implemented exactly. Thus, the degenerated geometries can be handled without special measures and accurate modeling results can be obtained even if the fractures are densely packed. In order to ensure the efficiency of the proposed method, floating-point filters and lazy evaluation strategy are used in this study to avoid unnecessary exact computations. The fractures can be either planer or curved and the only limitation to the shape and size of the resulting blocks is the available memory of the host system. Several examples are given to demonstrate the capability and performance of this method.

1. Introduction

Identification of the rock blocks presented in a rock mass structure is an important aspect of analyzing the mechanical and hydraulic behavior of jointed rock masses. Fast and robust algorithms for rock block identification can be used to assist in studying various phenomena such as in situ block size distribution, block morphology and fluid conductivity. The identified blocks can be the input of other discrete analysis models such as the discrete element method (DEM)[1] or discontinuous deformation analysis (DDA).[2] Initial studies of block identification algorithms are carried out for the idealized case of planar and infinite fractures.[2–4] This assumption imposes a critical restriction on the shape of blocks because only convex blocks can be formed by infinite fractures. The work of Lin et al.[5] introduced a topological based algorithm that is capable to detect polyhedral blocks formed by finite-sized fractures. Further refinements of this algorithm advance its ability of processing more complex discrete fracture network (DFN).[6–11] Yu et al.[12] proposed a block identification methods based on the element-block assembling approach, in which the complex-shaped blocks were represented as assemblages of convex element-blocks. In recent years, this method has been applied to various engineering projects[13–15] and developed to detect blocks formed by both planar and curved fractures.[16]

The resulting block structures are becoming increasingly realistic and complex as the complexity of DFN handled by the block identification algorithm increases. Meanwhile, the robustness of the block identification program is being challenged by the computational errors introduced in various stages of the algorithms. Although most block identification algorithms are relatively robust to handle these errors when dealing with a small number of blocks and fractures, there is no guarantee that these algorithms will return accurate results when the number of blocks increases. Actually, the numerical calculations are usually performed inaccurately with finite precisions. Without a proper error management scheme, the block identification algorithms may unexpectedly terminate if the computational errors accumulate.

There are very little discussions about the computational errors and the methods required to accommodate them when dealing with a large and complex DFN. In the published literatures, most block identification methods are presented in an 'idealized form' with little mention of this issue.[11] Generally, the geometrical predicates and structures calculated in these methods are assumed to be correct throughout the algorithm. However, this assumption may not be valid especially when dealing with degenerated geometries, such as extremely short edges or sharp angles (Fig. 1). Elmouttie et al.[10,11] emphasized the importance of addressing robustness issues in block identification algorithms and presented some measures to fix topological errors. However, the proposed methods mainly deal with the geometrical issues, such as fixing unintended geometrical features (e.g. holes or cracks) and incorrect accounting for orientations,[11] while the numerical issues like handling computational inaccuracies and special-cases are not involved. These numerical issues may derive from either the input

* Corresponding author.
E-mail addresses: zhengyinhe@cugb.edu.cn (Y. Zheng), xialu@cugb.edu.cn (L. Xia), yuqch@cugb.edu.cn (Q. Yu).

http://dx.doi.org/10.1016/j.ijrmms.2016.03.020
1365-1609/

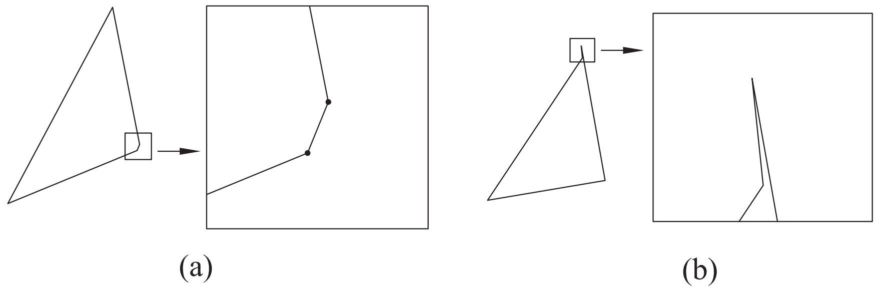

Fig. 1. Example of degenerate geometries: (a) an extremely short edge and (b) sharp angle.

data or the rounding operations of each calculation and may introduce great threat to the robustness of the block identification algorithm.

In this paper, a block identification method based on exact arithmetic is introduced. A rational number type is defined as the ratio of two multi-precision integers and five basic arithmetic operations (namely, addition, subtraction, multiplication, division and comparison) on rational numbers are implemented exactly. The algorithm used to identify blocks is based on the element-block assembling approach, which involves cutting the modeling domain into convex element-blocks with temporarily extended fractures and combining these element-blocks when the fractures are restored to their original size. The presented method is capable of constructing block systems containing hundreds of thousands of arbitrary shaped blocks formed by both planar and curved fractures. In this method, block structures are constructed accurately without introducing computational errors and the degenerated geometries, such as extremely short edges or sharp angles, can be handled without special measures. Accurate results are guaranteed even if the fractures are arranged in highly degenerated patterns. Several examples are given to validate the proposed algorithm and show its efficiency.

2. Exact arithmetic framework

All block identification methods implemented using computational geometry algorithms are sensitive to numerical errors. An effective approach for solving this issue is to use the exact arithmetic, which is rapidly developing in recent years.[17,18] This chapter gives a short discussion about the tolerance management scheme commonly used in most block identification algorithms to handle computational errors and presents an exact arithmetic framework based on a newly defined rational number type. Under this framework, the block structures can be constructed without computational errors and there is no need to contain additional numerical security measures in the block identification algorithm.

2.1. Tolerance management scheme

When working with finite precision arithmetic, unexpected results may be returned if two numbers are compared directly. The numbers that are equal in theory may turn out to be slightly different due to round-off errors. For instance, when we are calculating the intersection point of two lines and then checking whether this point lies on these two lines, the round-off errors may lead to a failure of the check.

A simple and commonly used solution for the above mentioned problem in most block identification algorithms is to set a small tolerance value μ, specifically, two numbers x and y are treated as equal if $|x-y|<\mu$. This approach, also known as tolerance management, might work in most situations, but its validity can never be known for sure.[18] The round-off errors may accumulate with the execution of the program and unpredictable behaviors will occur if the tolerance management fails, especially when handling degenerated cases, such as extremely short edges and sharp angles (Fig. 1). As the number of fractures increases, the degenerated cases will become a common phenomenon and the robustness of the block identification program is hard to guarantee.

Moreover, the well-established techniques to control round-off errors in numerical analysis is not applicable in block identification algorithms. Most of these techniques apply to numerical computation, such as finding numerical solutions of equations or calculating the eigenvalues of matrices. However, the block identification algorithms mainly focus on building geometrical structures, such as points and polygons, rather than computing numbers. Unlike the reliable numerical analysis methods, there is no established theory to evaluate the propagation of errors in geometrical computation.

2.2. Rational number type

Numbers are complex objects in computing. When building numerical models for rock blocks, the default arithmetic mode is usually the floating-point (FP) mode, which is dominant in scientific computing and have been implemented using hardware with IEEE standard. However, the floating-point numbers are represented using fixed precisions, which means that the calculations performed in FP mode are inexact. In order to perform exact arithmetic, a rational number type is defined in this study to represent the coordinates of points instead of floating-point numbers.

A rational number r is defined as the ratio of two integers, i.e. a numerator a and denominator b:

$$r = \frac{a}{b} \tag{1}$$

where a and b are required to be relatively prime to each other and $b > 0$. The number r can then be denoted as $r(a, b)$. Number zero is represented as $z(0, 1)$. Note that the map from integer pairs to rational numbers is not injective. In this study, a binary greatest common divisor (GCD)[19] algorithm is implemented to reduce integer pairs. Specifically, if a and b are integers, the greatest common divisor gcd (a, b) is defined as the largest integer that evenly divides both a and b. The rational number r corresponding to the integer pair (a, b) $(b > 0)$ can be reduced as:

$$r = r(a/gcd(a, b), b/gcd(a, b)) \tag{2}$$

Nevertheless, even with the help of GCD, the integers representing the numerators or denominators may still be very large in exact computation. Therefore, an arbitrary precision integer package such as the Boost Multiprecision Library or the GNU Multiple Precision (GMP) Arithmetic Library is necessary to avoid the overflow. Under the definition presented above, all the floating-point numbers can be converted to rational numbers without losing accuracy, but not vice versa.