# GeneralBlock: A C++ Program for Identifying and Analyzing Rock Blocks Formed by Finite-Sized Fractures

Lu Xia, Qingchun Yu, Youhua Chen, Maohua Li, Guofu Xue, Deji Chen



# GeneralBlock: A C++ Program for Identifying and Analyzing Rock Blocks Formed by Finite-sized Fractures

Lu Xia[1], Qingchun Yu[1,*], Youhua Chen[2], Maohua Li[2], Guofu Xue[2], Deji Chen[2]

[1]School of Water Resources & Environment Science, China University of Geosciences, Beijing 100083,China.
yuqch@cugb.edu.cn
[2]Changjiang Conservancy Commission, Yichang 443003,China.

**Abstract.** GeneralBlock is a software tool for identifying and analyzing rock blocks formed by finite-sized fractures. It was developed in C++ with a friendly user interface, and can analyze the blocks of a complex-shaped modeling domain, such as slopes, tunnels, underground caverns, or their combinations. The heterogeneity of materials was taken fully into account. Both the rocks and the fractures can be heterogeneous. The program can either accept deterministic fractures obtained from a field survey, or generate random fractures by stochastic modeling. The program identifies all of the blocks formed by the excavations and the fractures, classifies the blocks, and outputs a result table that shows the type, volume, factor of safety, sliding fractures, sliding force, friction force, cohesion force, and so on for each block. It also displays three-dimensional (3D) graphics of the blocks. With GeneralBlock, rock anchors and anchor cables can be designed with the visual assistance of 3D graphics of blocks and the excavation. The anchor, cables, and blocks are shown within the same window of 3D graphics. The spatial relationship between the blocks and the anchors and cables is thus very clear.

**Keywords:** GeneralBlock · finite-sized fracture · rock block identification and analysis

## 1 Introduction

A computer program for identifying and analyzing rock blocks formed by finite-sized fractures is a useful tool in many problems involving fractured rocks. If a rock is slightly fractured, the blocks are infrequent in the rock mass. In this case the number, dimensions and locations of the blocks are a major concern. The rock blocks, which are commonly buried in the rock mass under natural conditions, may be exposed by artificial excavations and may fall into the openings formed by the excavation. Many authors have discussed the possibilities of the occurrence of unstable blocks caused by the intersection between fractures and excavations [1-4]. On the other hand, if a rock is heavily fractured, the rock may be visualized as an assemblage of isolated rock blocks. In this case, the mechanical behavior of the rock is often simulated as the behavior of a system of rock blocks [5-7].

Because the geometry of the blocks exerts a strong influence on the mechanical properties of the rocks, and the geometry of the blocks is determined by the geometry

adfa, p. 1, 2011.

of the fractures, many authors have studied the relationship between the geometries of fractures and rock blocks. Basically, their approaches may be divided into two types. The major difference between the two methods is whether or not the shape of the fractures has a definition prior to block identification. In the first method, the shapes of the fractures are defined before block identification; this approach was used by many authors [8-12], and is adopted in GeneralBlock. In this method, the fractures can cross each other and terminate in intact rock; an individual fracture can be contributive, noncontributive, or partly contributive. The second method uses a block generation language [13-14], whereby the fractures always form fully connected networks and all of the fractures are completely contributive.

GeneralBlock is written in Visual C++ 6.0 using the OpenGL library. In the program, the modeling domain or the excavation may be of arbitrarily complex shape. The rock and fractures may be heterogeneous. The fractures may be either deterministic fractures obtained from a field survey or random fractures generated by stochastic modeling. The program identifies all blocks formed by the excavations and the fractures, classifies the blocks, and outputs a result table that shows the type, volume, factor of safety, sliding fractures, sliding force, friction force, cohesion force, and so on for each block. It also outputs three-dimensional (3D) graphics of each block. With GeneralBlock, anchors and cables may be added, and are shown with the blocks within the same window of 3D graphics, making the spatial relationship between the blocks, anchors, and cables very clear. The user can therefore edit rock anchors and anchor cables with the visual assistance of 3D graphics of blocks and excavation faces.

## 2 Algorithm

Some authors have discussed the issues in block identification. In initial studies, the fractures were assumed to be infinitely large, and hence, the blocks were limited to convex shapes. Lin et al. [8] and Ikegawa and Hudson [9] presented their block identification approaches for finite extended fractures based on topological concepts. In these approaches, the intersections of the fractures are calculated first to define the sets of vertices, edges, and faces. Then, the sets were regularized to discard any isolated and dangling vertices, edges, and faces. Finally, the blocks were identified through boundary-chain operations of the closed surfaces and the Euler–Poincare formula for a polyhedron. Similar methods have also been discussed in detail [10][12][15].

GeneralBlock was developed based on the procedure presented by Yu et al. [16]. It decomposes a complex block into a finite number of convex element blocks during the identification process. The adoption of the concept of an element block makes it possible to represent a complex block using an assemblage of simple convex blocks, and thus, to transform the difficult calculations involved in identifying the blocks around complex excavations into simple calculations using convex geometry. In the procedure, the complexity of the modeling domains and the heterogeneity of a material are taken fully into account.

In the procedure, a complex modeling domain is initially divided into convex subdomains, and each subdomain is further decomposed into convex blocks with infinite fractures. Then, the fractures are restored to finite dimensions, and the domain is reassembled by combining the subdomains into a full domain. Thus, the procedure is comprised of the following major steps:

1. Subdividing the modeling domain into a finite number of convex subdomains;
2. Removing the non-contributive fractures;
3. Decomposing the subdomains into element blocks with the fractures, which are temporarily taken as infinitely large at this stage;
4. Restoring the infinite fractures to finite;
5. Assembling the modeling domain.

# 3 User Interface

The solution of a block identification and analysis problem with GeneralBlock usually includes the following major steps.

1. Open a new project. This creates a new directory in which to store the data for the problem. All ensuing data for this problem will be stored in this directory.
2. Define the modeling domain.
3. Input deterministic fractures and/or generate stochastic fractures.
4. Calculate and observe the traces of the fractures.
5. Filter fractures.
6. Identify blocks and analyze their stability.
7. Display analysis results.
8. Design rock anchors and anchor cables.

## 3.1 Defining the Model Domain

The first step in block identification and analysis is defining the model domain. A major process in the definition is to decompose the problem domain into a number of convex subdomains. The subdivision of a problem domain is similar to the subdivision of a model domain into elements in the FEM, and the geometry and the associated data structure of a subdomain are also similar to those of an element in the FEM technique. The subdomains, surface polygons, and vertices must be numbered uniquely, and no overlap among the subdomains or surface polygons is allowed.

Manually decomposing a problem domain into subdomains is not difficult, even for a problem domain with a very complex shape. However, it is not simple to code a computer program to automatically decompose different-shaped rock masses. Thus, GeneralBlock provides graphical user interfaces for the subdivision of some common rock structures, including slopes, tunnels, and underground caverns. For these common structures, the user only enters a few data values through a graphical window to define the rock masses. The subdivision of the domain is performed automatically by GeneralBlock, and the user is not required to be aware of it.

### 3.1.1 The Window Defining the Model Domain.

GeneralBlock provides graphical interfaces for three kinds of structures: slope, tunnel, and underground cavern. Accurately speaking, a tunnel is a kind of underground cavern. As an example, figure 1 shows the actual underground powerhouse cavern 311.3 m long and 71 m high in the Three Gorges Project. In the lower part of the powerhouse, the distance between the two vertical sidewalls is 31 m, and in the upper part the span of the crown is 32.6 m.

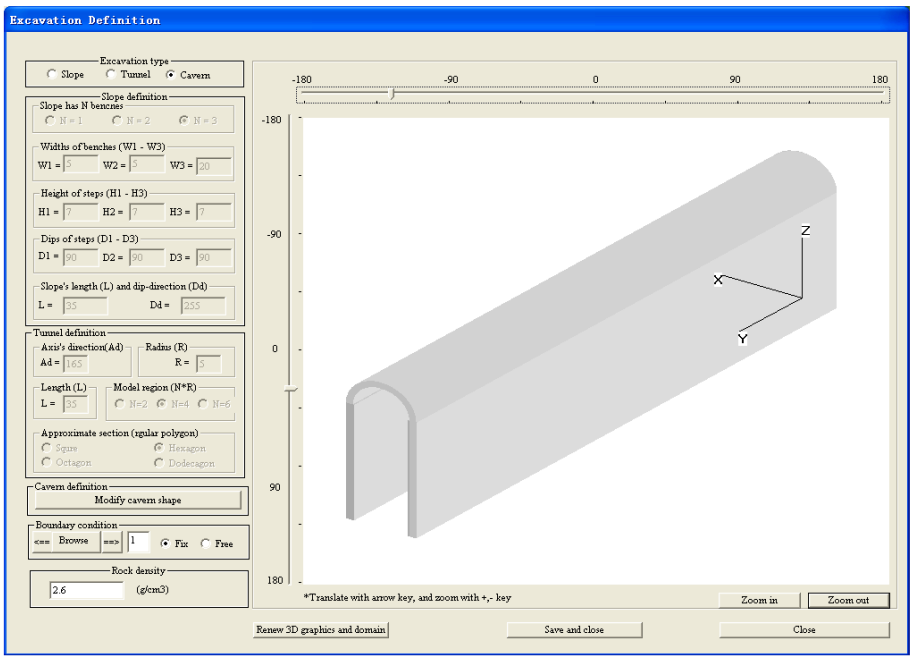


**Fig. 1.** The window defining the model domain (the example of the underground powerhouse cavern in the Three Gorges Project defined by GeneralBlock)

### 3.1.2 The Format of "model_domain.dat"

The data defining the model domain and sub-domains are all stored in an ASCII file named "model_domain.dat". This file can be edited by a text editor. The user can prepare it and have GeneralBlock read it to perform block analyses.

The rock masses and the excavations of study domains may have various shapes, e.g., slopes, tunnels, and their combinations. However, all of the data arising from domain definitions and subdivisions are stored in this file, and all of them always have the same format. This file includes the following data.

1. Direction of x-axis (in deg), excavation type (slope = 0; tunnel = 1; cavern = 3; complex = 4);
2. Number of the nodes to define the domain, number of the faces to define the domain;

3. Number of the nodes to define each of the faces;
4. An index for each of the faces;
5. An index to indicate the mechanic property of each face (common = 0; fix = 1; free = 2);
6. A vertex list for each of the face;
7. Three-dimensional coordinates for each vertex;
8. The number of sub-domains, rock density of each sub-domain;
9. The number of faces for each sub-domain;
10. Face list for each sub-domain.

### 3.2 Entering and Stochastic Modeling of Fractures

For GeneralBlock, the fractures are divided into two types: deterministic fractures and stochastic fractures. Deterministic fractures usually arise from excavation mappings; their geometric and mechanical parameters are determined through various measurements. Stochastic fractures are usually obtained through random modeling; their parameters are generated from a stochastic model. Deterministic fractures are often larger in scale than stochastic ones. Although deterministic and stochastic fractures may arise from different sources, they have the same defining method: each fracture is defined by nine parameters, including the x, y, and z coordinates of the disc center, dip direction, dip angle, radius, aperture, cohesion coefficient, and friction angle. In block analysis, the two kinds of fractures are treated in the same way.

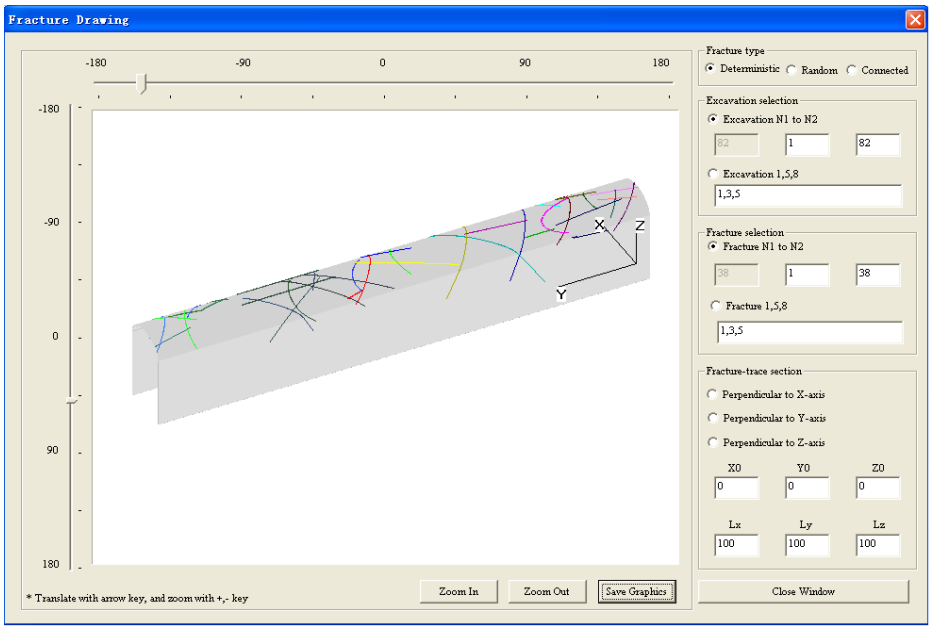


**Fig. 2.** The window for calculating and displaying the trace of fractures (traces of main faults in vaults on excavation of underground powerhouse in the Three Gorges Project)

After inputting deterministic fractures or simulating stochastic fractures, filtering fractures is necessary. The objective of fracture filtering is to find and eliminate the fractures that are evidently non-contributive to block formation. And fractures below the size specified by user may be neglected in block identification. For most practical engineering, the blocks close to the excavations are usually of more concern than more distant ones. Thus, it is often the case that only the fractures near the excavations are taken into account. In GeneralBlock, the distance of a fracture from an excavation is an integer. If the fracture intersects directly with the excavation, the distance is zero; if the fracture does not intersect directly with the excavation but is connected indirectly to the excavation through one fracture, the distance is one; the rest may be deduced by analogy.

The remaining fractures after fracture filtering are stored in the file "contributive_fracture_xyzabr.dat". GeneralBlock reads all fractures in this file when performing block identification.

Calculating and showing the traces of fractures should be done next. A trace is an intersection line between a fracture and a rock surface, e.g., an excavation or a natural exposure. Calculating the traces of fractures on specified excavations and displaying them through 3D graphics is a useful function of GeneralBlock. In practical rock engineering, when a fault is encountered by a borehole or an exploration tunnel, the engineering geologist is often confronted with the problem of predicting where the fault would be reencountered by other excavations.

Figure 2 shows the example traces shown in this figure are the traces of main faults in vaults on excavation of underground powerhouse in the Three Gorges Project.

### 3.3 Identifying Blocks and Displaying Results

The calculation of block identification completes the following tasks: block identification, volume calculation, block classification, removability analysis, sliding fracture determination, and force calculation (sliding force, friction, cohesion, and support).After the completion of block identification, the user can investigate the results.

Figure 3 shows the window displaying the results of block analysis. All of the blocks are listed in the table and sequenced according to their volumes. The type, volume, safety factor, sliding fracture, sliding force, friction, cohesion, and the support forces provided by rock bolts and anchor cables of each block are displayed in the table. All data of block results save in the file "Result_table.dat".

## 4 Conclusion

GeneralBlock has been developed for identifying and analyzing the rock blocks formed by complex excavations and finite-sized fractures. It is a useful software tool in many ways for the study of fractured rocks. It may be used to identify and analyze the stability of rock blocks arising from rock excavation. It may also be used to investigate the size and geometry of rock blocks, two important factors affecting the mechanics of fractured rocks.

GeneralBlock was developed in C++ with a friendly user interface. In the program, the modeling domain or the excavation can be very complex in shape, such as slopes, tunnels, underground caverns, or their combinations. The heterogeneity of the materials was taken fully into account. Both the rocks and the fractures can be heterogeneous. The fractures can be either deterministic fractures obtained from a field survey, or random fractures generated by stochastic modeling. The program: identifies all of the blocks formed by the excavations and the fractures; classifies the blocks and outputs a result table that shows the type, volume, factor of safety, sliding fractures, sliding force, friction force, cohesion force, and so on, of each block; and outputs three-dimensional (3D) graphics of each block. With GeneralBlock, rock anchor and anchor cable design can be performed with the visual assistance of 3D graphics of blocks and excavation faces. The anchors, cables, and blocks are shown within the same window of 3D graphics. Consequently, the spatial relationships between the blocks and the anchors and cables are very easy to understand.

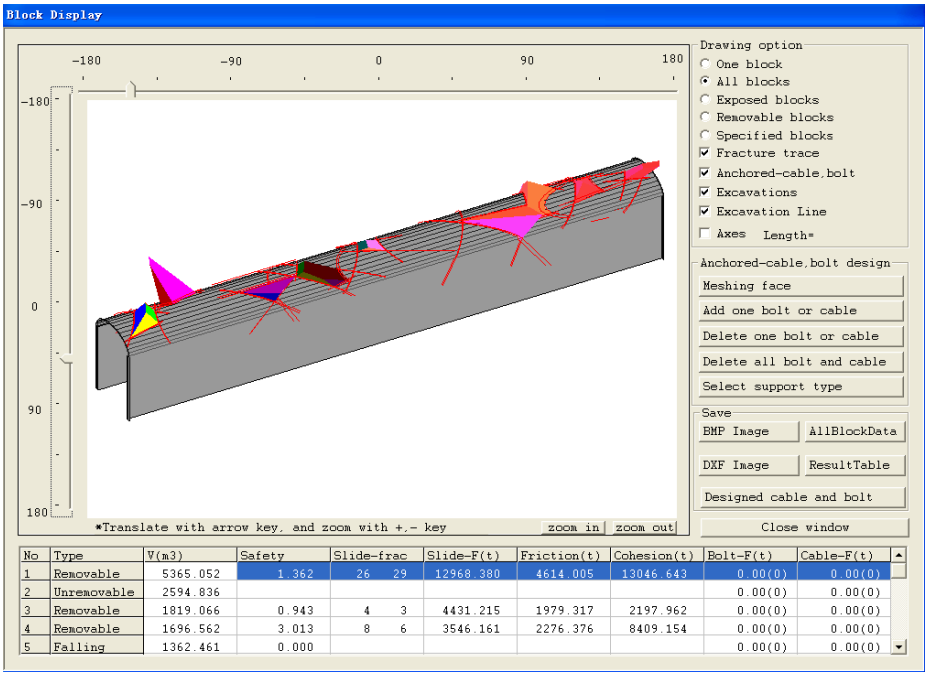


**Fig. 3.** The window displaying the results of block analysis (3D graphics of blocks in vaults of underground powerhouse in the Three Gorges Project using GeneralBlock)

## Acknowledgements

This study was financially supported by the National Natural Science Foundation of China (Grant No.40372134, No.40772208, No.41272387) and the Fundamental Research Funds for the Central Universities (Grant No.2652011305).

## References


1. Warburton, P.M. (1981) Vector stability analysis of an arbitrary polyhedral rock block with any number of free faces. International Journal of Rock Mechanics and Mining Science and Geomech. Abstracts 18, 415–427.
2. Goodman, R.E. and Shi, G. (1985) Block theory and its application to rock engineering. Prentice-Hall, Englewood Cliffs, NJ.
3. Lin, D. and Fairhurst, C. (1988) Static analysis of the stability of three-dimensional blocky systems around excavation in rock. International Journal of Rock Mechanics and Mining Science and Geomech. Abstracts 25, 139–147.
4. Hoek, E., Kaiser, P.K. and Bawden, W.F. (1995) Support of underground excavation in hard rock. Balkema, Rotterdam.
5. Cundall, P.A. (1988) Formulation of a three-dimensional distinct element model—part 1: A scheme to detect and represent contacts in a system composed of many polyhedral blocks. International Journal of Rock Mechanics and Mining Science and Geomech. Abstracts 25, 107–116.
6. Shi, G. (1989) Discontinuous deformation analysis: A new numerical model for the statics and dynamics of block systems, PhD thesis, University of California, Berkeley, USA.
7. Wu, J.H., Ohnishi, Y. and Nishiyama, S. (2005) A development of the discontinuous deformation analysis for rock fall analysis. International Journal of Numerical and Analytical Methods in Geomechanics 29, 971–988.
8. Lin, D., Fairhurst, C. and Starfield A.M. (1987) Geometrical identification of three-dimensional rock block system using topological techniques. International Journal of Rock Mechanics and Mining Science and Geomech. Abstracts 24, 331–338.
9. Ikegawa, Y. and Hudson, J.A. (1992) A novel automatic identification system for three-dimensional multi-block systems. Engineering Computing 9, 169–179.
10. Jing, L. (2000) Block construction for three-dimensional discrete element models of fractured rocks. International Journal of Rock Mechanics and Mining Science and Geomech. Abstracts 37, 645–659.
11. Ohnishi, Y. and Yu, Q. (2001) 3D block analyses for fractured rock. In: Desai et al., editors. Computer methods and advances in geomechanics. Rotterdam: Balkema, 577–580.
12. Lu, J. (2002) Systematic identification of polyhedral blocks with arbitrary joints and faults. Comput Geotech 29, 49–72.
13. Heliot, D. (1988) Generating a blocky rock mass. International Journal of Rock Mechanics and Mining Science and Geomech. Abstracts 25, 127–138.
14. Empereur-Mot, L. and Villemin, T. (2003) OBSIFRAC: database-supported software for 3D modeling of rock mass fragmentation. Computers & Geosciences 29, 173–181.
15. Yu, Q. (2000) Analyses for fluid flow and solute transport in discrete fracture network, PhD thesis, Kyoto University, Kyoto, Japan.
16. Yu, Q., Ohnishi, Y., Xue, G. and Chen, D. (2009) A generalized procedure to identify three-dimensional rock blocks around complex excavations, International Journal of Numerical and Analytical Methods in Geomechanics 33, 355–375.