Revista Minelor – Mining Revue
ISSN-L 1220-2053 / ISSN 2247-8590
vol. 31, issue 3, pp. 82-92, selected paper from the First International EnviroMining Virtual Conference 2025

GENERATION and VISUALIZATION of DFN (DISCRETE FRACTURE NETWORK) FROM ROCK MASS DISCONTINUITY SETS

Erkan ÜLKER[1], Alparslan TURANBOY[2],*
Görkem Rıdvan ARIK[3], İbrahim UYSAL[4]

[1]Konya Technical University, Computer Eng., Konya, Türkiye, eulker@ktun.edu.tr
[2]Necmettin Erbakan University, Energy Systems Eng., Konya, Türkiye, aturanboy@erbakan.edu.tr
[3]Konya Technical University, Computer Eng., Konya, Türkiye, gorkemarik2018@gmail.com
[4]Konya Technical University, Computer Eng., Konya, Türkiye, uysalibov@gmail.com

DOI: 10.2478/minrv-2025-0031
received – 24.01.2025 / revised – 11.02.2025
accepted – 18.02.2025 / published – 22.09.2025

Abstract: *This study aims to generate DFN (Discrete Fracture Network) from Rock Mass discontinuity sets and web-based 3D visualization. By providing accurate modeling of the fractures in the Ornamental Stone Quarry with visualization, it is expected to improve the analysis of discontinuity sets and increase efficiency in industrial processes. Nowadays, analysis of discontinuity sets in stone quarries is usually done with limited visual methods. This study will enable discontinuity sets to be more accurately identified and analyzed in 3D using DFN modeling methods. Thus, it will be possible to plan forward-looking goals such as more effective management of resources in mines, reducing waste and optimizing production processes. The method of this study does not include site studies, geophysical data and laboratory experiments to be conducted in natural stone quarries. It is accepted that detailed analysis was made on the rock samples obtained from the quarries and the geometry, orientation and properties of the cracks were determined. The data is combined using DFN modeling methods and 3D DFN models are created. Sample Data was obtained using a pre-registered database of a Commercial Firm. Additionally, a web-based interactive platform was developed and 3D DFN models were visualized on this platform.*

Keywords: *DFN, Web-based 3D visualization, Rock mass, Representative Prizm, Discontinuity, Marble Quarry*

1. Introduction

DFN is defined as a complex network of fractures found within subsurface rocks. This fracture network has significant effects on various geological processes such as the formation and flow behavior of hydrocarbon reservoirs, groundwater movement, underground mining, and underground storage. DFNs are a modeling approach that describes the geometry and properties of rock fractures. These fractures can be natural formations or human-made breaks. DFN modeling uses various data sources such as field studies, geophysical data, and laboratory experiments to determine the length, width, aperture, and orientation of fractures within the network.

Modeling DFNs enables a more accurate understanding of subsurface flow and reservoir behavior. Especially in hydrocarbon reservoirs, DFN modeling is used to predict the flow properties of reservoirs and plays an important role in reservoir engineering studies. Additionally, DFNs are used to understand the movement of groundwater resources. DFN modeling is a crucial tool in areas such as the protection of groundwater resources, assessing the risk of groundwater contamination, and groundwater management. In conclusion, the concept of Discrete Fracture Network (DFN) refers to a modeling approach that defines the complex network of fractures found in subsurface rocks and is used in understanding geological processes and resource management.

The generation of Discrete Fracture Networks (DFNs) from rock mass discontinuity sets and their 3D visualization is a critical area in geomechanics and hydrogeology. Recent studies have explored various methodologies and applications in this domain. A study by Sewnun et al. [1] reviews structural data

[*] Corresponding author: Erkan Ülker, prof. Ph.D. eng., Konya Technical University, Computer Eng., Konya, Türkiye, eulker@ktun.edu.tr

vol. 31, issue 3 / 2025
pp. 82-92

collection methodologies for DFN modeling, emphasizing the statistical representation of rock mass discontinuity systems through stochastic generation of discontinuity sets. Another research by Hyman et al. [2] introduces dfnWorks, a DFN framework for modeling subsurface flow and transport, highlighting its capabilities in simulating complex fracture networks. In the realm of 3D visualization, Rocscience [3] has advanced blocky rock mass modeling with DFNs, providing tools for detailed 3D representations of fracture networks. Additionally, the Petrel fracture modeling software [4] offers flexibility in both stochastic and geomechanically fracture characterization, supporting discrete and implicit fracture modeling. These studies contribute to the understanding and advancement of DFN generation and 3D visualization techniques.

The generation of Discrete Fracture Networks (DFNs) is pivotal in modeling and analyzing fractured rock masses. Several specialized software platforms have been developed to facilitate DFN generation, each offering unique features and capabilities. DFN.lab [5] is a modular computational suite designed to handle three-dimensional DFN models, encompassing generation, simulation, and analysis of connectivity, flow, mechanical, and transport properties. Developed by the Fractory, a collaboration between the French National Centre for Scientific Research (CNRS), the University of Rennes, and Itasca Consultants, DFN.lab is implemented in C++ for high performance, with a Python API for user accessibility. A distinctive feature of DFN.lab is its capacity to manage multiscale heterogeneities at both the fracture and network scales, allowing for local variations in hydraulic properties and the modeling of fracture patches clogged by mineralization. Additionally, DFN.lab includes genetic generation models as alternatives to classical Poisson models, providing a more nuanced approach to DFN generation [5]. dfnWorks developed at Los Alamos National Laboratory and is a parallelized computational suite for generating three-dimensional DFNs and simulating flow and transport within them. The suite comprises several components: dfnGen, dfnFlow and dfnTrans. **dfnGen**: Utilizes the Feature Rejection Algorithm for Meshing (FRAM) methodology to stochastically generate three-dimensional DFNs, integrating with the LaGriT meshing toolbox to create high-quality computational mesh representations. **dfnFlow**: Simulates flow through the network using the massively parallel subsurface flow and reactive transport finite volume code PFLOTRAN. **dfnTrans**: Adopts a Lagrangian approach to simulate transport through the DFN, determining pathlines and solute transport. The suite is designed for high-performance computing environments, enabling the study of flow and transport in fractured media across scales ranging from millimeters to kilometers. [2]. Itasca's DFN module [6] offers an efficient toolset for generating and manipulating fractures within rock masses. Key features include Import/export capabilities for fractures from Itasca and Fracman files, Addition of deterministic fractures and generation of stochastic fractures, Computation of clusters and connectivity properties, Fracture simplification methods, Definition of mechanical properties for fractures and their assignment to model components, and Visualization tools for fractures, outcrop/tunnel trace maps, and stereonets. These features facilitate the creation of site-descriptive fracture sets, calibration of fracture models based on 2D/1D data, derivation of DFN statistical properties, and structural analysis of fractures in relation to resulting mechanical properties [6]. PySimFrac [7] is an open-source Python library designed for generating three-dimensional synthetic fracture realizations, integrating with fluid simulators, and performing analysis. It allows users to specify one of three fracture generation techniques (Box, Gaussian, or Spectral) and perform statistical analyses, including autocorrelation, moments, and probability density functions of fracture surfaces and apertures. PySimFrac includes integration examples with pore-scale simulators and the discrete fracture network simulator dfnWorks, facilitating accurate characterization of fluid transport in geologic media [7]. In addition to the previously discussed software platforms, several other tools are available for generating Discrete Fracture Networks (DFNs): Fracman [8], FracSim3D [9] and Geodict [10]. FracMan is a comprehensive software suite developed by Golder Associates for modeling and analyzing fractured rock masses. It offers capabilities for DFN generation, visualization, and simulation of flow and transport processes within fracture networks. FracMan [8] is widely used in various fields, including hydrogeology, petroleum engineering, and mining. FracSim3D [9] is a software tool designed for the stochastic modeling of three-dimensional fracture networks. It allows users to generate DFNs based on statistical distributions of fracture properties and analyze their impact on rock mass behavior. FracSim3D is utilized in geotechnical engineering and hydrogeological studies to assess the influence of fractures on fluid flow and mechanical stability. GeoDict [10] is a versatile software package for modeling and analyzing porous media and fractured materials. It includes modules for generating DFNs, simulating flow and transport processes, and evaluating mechanical properties. GeoDict is employed in various research areas, such as material science, geosciences, and petroleum engineering. These software tools provide additional options for researchers and engineers involved in DFN modeling, each offering unique features tailored to specific applications and industries.

 vol. 31, issue 3 / 2025
pp. 82-92

The aim of this paper is to develop software that facilitates the generation of DFN (Discrete Fracture Network) from discontinuity sets and their 3D visualization through a web-based platform in the marble industry. This software will evaluate statistical approaches for DFN modeling, utilizing field data, and aims to enable a more accurate and detailed analysis of discontinuity sets.

The study is conducted to address a problem encountered in the marble industry. Current methods for analyzing discontinuity sets are generally based on limited data, which creates constraints in terms of accuracy and precision. The purpose of this paper is to create a more accurate DFN model by evaluating fracture network data obtained using field data through statistical approaches, ultimately aiming to enhance production efficiency in marble quarries.

The specific objectives of this paper include the collection and evaluation of field data, the application of statistical methods, the creation of the DFN model, and the development of a web-based visualization interface. The software will enable more effective management of marble quarries, optimize resource use, reduce waste, and optimize energy consumption.

2. Methodology

This section explains the analytical/experimental solution methods, technical/scientific approaches, and the steps followed to achieve the study objectives. Statistical methods used in marble quarries and marble mining are applied for various purposes. Here are some common statistical methods and their objectives.

- Descriptive Statistics: Descriptive statistics are used to summarize and describe data. For example, basic characteristics such as the sizes, weights, and densities of marble blocks can be statistically summarized. This method is used to provide an overall understanding of the dataset and to present basic information for visualization,
- Hypothesis Testing: Hypothesis testing is used to verify or reject a statistical claim on a sample. In marble mining studies, hypothesis testing can be applied to topics such as marble quality or the characteristics of marbles from different quarries,
- Regression Analysis: Regression analysis is used to model the relationship between one variable and others. In marble quarries, regression analysis can be applied to determine factors affecting marble quality or predict the strength properties of blocks,
- Variance Analysis: Variance analysis is used to statistically compare different groups or conditions. For example, variance analysis can be applied to compare the production efficiencies of different marble quarries or to assess the impact of different cutting methods on the quality of marble blocks.

Discrete Fracture Network (DFN) is a method used for the analysis and modeling of discontinuity sets. DFN creates a network model to define the geometric and hydraulic properties of fracture systems. This model includes the distribution, size, orientation, and connections of fractures. DFN is used in marble quarries to determine the distribution and properties of fractures and to understand factors influencing water movement or mining processes.

There is a connection between DFN and statistical methods. When creating a DFN, field data are used, and these data can be analyzed with statistical methods. For example, parameters such as the distribution and size of fracture systems can be determined through statistical analysis. Additionally, statistical methods like regression analysis or variance analysis can be used to predict the hydraulic or mechanical properties of DFN. DFN can be considered as a tool where statistical methods are applied. Statistical methods play a significant role in obtaining the necessary data for creating and analyzing the DFN model and determining its parameters. In this way, by using both DFN and statistical methods together, marble quarries can be better understood, optimized, and their efficiency can be increased.

The main studies and general method details of the paper include the following steps.

1. Data Collection:

Obtaining field data from a partner company
Collecting data using appropriate methods such as rock sampling, geophysical measurements or imaging techniques.

2. Data Analysis:

Evaluating the collected field data using statistical analysis, Summarizing the dataset using descriptive statistics
Applying statistical methods such as hypothesis testing and regression analysis
Statistically determining the geometry, size, distribution and connections of fractures

3. DFN Generation:

Using parameters obtained from statistical analysis to create the DFN model

vol. 31, issue 3 / 2025
pp. 82-92

Determining the network structure of fractures and defining the relationships
Performing calibration and validation processes for the DFN model

4. Web-Based 3D Visualization:

Visualizing the generated DFN model on a web-based platform
Developing a user interface that allows users to interactively explore and analyze the DFN
Realistically visualizing the geological structure of marble quarries and providing scaling and transformation options.

5. Validation and Optimization:

Conducting validation processes for the generated DFN model
Comparing the model with field data and evaluating its compatibility
Making necessary corrections to improve and optimize the DFN model

These methods include the analytical and experimental solution methods to achieve the objectives of the paper. The data collection process involves field studies, and the use of data obtained from the partner company. Data analysis includes the statistical evaluation of the dataset and the determination of DFN parameters. DFN generation involves using the parameters to create and calibrate the DFN model. Web-based 3D visualization ensures the presentation of the DFN model on a user-friendly platform. Validation and optimization processes aim to validate and improve the generated model.

This platform provides a web-based platform for 3D visualization of mining site data, enabling users to analyze structural information captured through various survey methods like scanline, televiewer, and lidar. The frontend, developed with React, allows users to upload data and interact with 3D models. The backend, built in .NET with MongoDB and PostgreSQL, handles data storage and computational tasks using C++ DLLs for efficiency. 3D rendering is achieved by compiling C++ code to WebAssembly with Emscripten, enabling high-performance, real-time visualization through WebGL. This architecture ensures accessible, interactive analysis of geological data.

2.1. Architecture

The basic architecture of the module is shown in Figure 1. All users of this software are referred to as "Customer" because the software is being developed for a private company based in Türkiye, and the term "Customer" is preferred instead of "User".

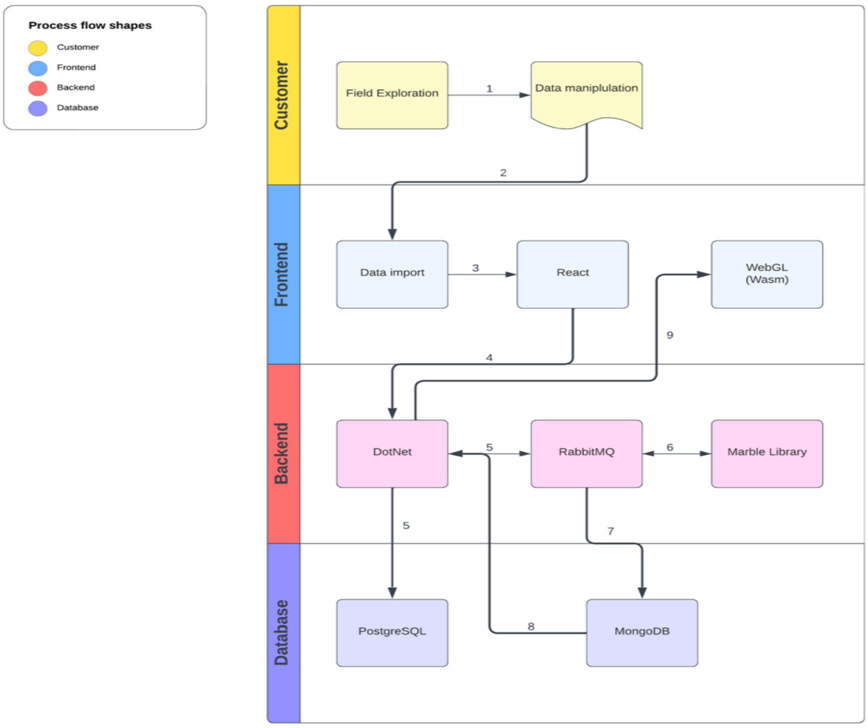

Fig.1. Basic software architecture used in the module

In the "Customer" phase, essential field data is gathered through various exploration techniques. Each method captures different types of information about the geological structure of the mining site:
Scanline: This technique involves observing and measuring rock layers or fractures along a specific line on the surface. It helps identify the size, orientation, and spacing of cracks or joints in the rock.

vol. 31, issue 3 / 2025
pp. 82-92

Jointset (Drilling): Jointset data is collected by drilling into the ground to understand the natural cracks and layers below the surface. It provides a 3D perspective on rock structures at different depths.
Lidar (GPR): Light Detection and Ranging (Lidar) and Ground Penetrating Radar (GPR) use laser pulses or radio waves to map the surface and subsurface of the area, capturing fine details of the terrain and any underlying structures.
Resistivity: This method measures how much the ground resists electrical current, which can indicate different types of rocks or soil layers.
Magnetometry: Magnetometry measures magnetic fields, which helps identify rock types and mineral content based on their magnetic properties.
Seismic: Seismic methods use sound waves to detect variations in rock layers and locate discontinuities or faults beneath the surface.

Each type of data is either entered forms by the customer or uploaded directly to the system. This collected data provides a foundational input for the platform and will later be used to create detailed 3D models of the mining site.

To enhance understanding, an example of scanline data has been included. This visual example illustrates how surface measurements capture information about fractures or joints, providing a clearer perspective on the data collection process. Visuals like scanline data examples help users better grasp the types of information gathered from the field. An example of Scanline Information is shown in Figure 2.

Scanline Information		**Outcrop information**					
No. 1		Location: Austria			Height: 3 m		
Name: CZ		Dip direction: 346			Width: 20 m		
Date: 1.5.2022		Dip angle: 85			Excavation method: natural		
Trend of Scanline: 170		Overhanging slope (yes/no): no			Condition of exposure: weathered		
Plunge of Scanline: 10		Rock type: Granodiorit gneiss			Comments: no		
Intersection distance [m]	**Dip direction [°]**	**Dip angle [°]**	**Semi-trace length above [m]**	**Semi-trace length below [m]**	**Termination above**	**Termination below**	**Comments**
0	270	86	2.5	1.4	2	1	Joint
0.1	272	75	3.4	1.5	2	1	Joint
0.6	10	88	1.5	1.0	3	2	Joint
0.9	135	8	0.9	0.8	3	2	Joint
1.1	262	75	1.2	1.3	2	2	Joint
1.3	82	12	1.0	1.1	2	3	Joint

Fig.2. Sample data format used for collecting field data

2.2. Frontend

The "Frontend" step is where users interact with the platform through a web-based interface, developed primarily with React. React's component-based architecture allows for modular, reusable pieces of code, which are essential in building a highly interactive and dynamic interface. This structure enables users to easily upload geological data, navigate through various options, and view 3D models, all within a smooth and responsive interface.

For styling and UI consistency, shadcn components are integrated. Shadcn provides a set of pre-built, customizable components that follow modern design principles. By using shadcn, the platform maintains a clean and cohesive visual design across different sections, enhancing usability and creating a professional appearance that matches industry standards.

To manage data and state across the application, Redux is used. Redux plays a critical role in organizing and synchronizing data flow, especially important in an application with complex interactions like 3D visualization. When a user uploads new data, all inputs are sent to the backend for processing. The backend performs necessary calculations, and when the user wants to visualize intersections, for instance, between discontinuities and the field, the relevant data is sent back to the frontend. This data then flows through React and reaches the WebAssembly (Wasm) component, where it is rendered on the screen.

Key frontend functionalities include:
Data Upload: Users can input geological data in various formats, such as forms or text files, which are sent to the backend for processing. React and Redux ensure that the file upload and data submission processes are smooth and provide immediate feedback to users.

 vol. 31, issue 3 / 2025
pp. 82-92

3D Rendering: Previously, all calculations for rendering 3D models were performed on the client side using Three.js. However, as the calculations required high processing power, we opted to refactor this process by compiling our calculations in C++ and creating a dynamic-link library (DLL) for use with our .NET backend. This shift means that we now use our custom library solely for the visualization component, leveraging the strengths of WebAssembly (Wasm) for rendering. WebAssembly allows us to run code written in languages like C++ in the browser, providing near-native performance. It cannot be used directly in browsers without a compilation step because browsers only understand JavaScript natively. Emscripten, a toolchain for compiling C and C++ code to Wasm, handles this translation, converting our C++ code into a format that can be executed in web environments. By utilizing Emscripten, we can directly leverage the computational power of C++ while rendering through WebGL in the browser, ensuring both high performance and flexibility in our graphical capabilities. The decision to forgo building a graphics library from scratch stems from the limitations of Three.js in meeting our specific computational needs. The architecture of Three.js would have required us to introduce intermediate steps to access the necessary data for our calculations, complicating the workflow. Instead, by using Emscripten to compile our existing C++ code into Wasm, we maintain full control over our design and achieve a high- performance interface tailored to our requirements.

Through this combination of React, shadcn, and Redux, the frontend provides a powerful, user-friendly environment. It enables users of varying technical backgrounds to effectively interact with geological data, preparing them for deeper analysis and exploration in the subsequent stages of the platform.

2.3. Backend

The backend of the platform is developed using .NET and employs a Multi-Tenant Architecture. This architecture is essential for managing multiple users and their respective datasets efficiently, ensuring that user data is securely isolated while still allowing for shared resources. This design is particularly beneficial for scalability and flexibility, accommodating a growing number of users without compromising performance.

The backend handles several critical operations, including user authentication, data management, and the integration of computational processes necessary for visualization. It ensures that user sessions are secure and that only authenticated users can access their specific data stored within the PostgreSQL database. The authentication system is designed to provide a seamless experience, allowing users to log in with minimal friction while maintaining robust security measures.

Our backend infrastructure is hosted on Hetzner Cloud, utilizing a range of technologies that are all containerized using Docker. This approach enables rapid deployment and management of services while ensuring that the environment remains consistent across different stages of development, testing, and production. The key components of our backend include:

Data Import:

This functionality allows users to input geological data into the system through the frontend interface. Once the user submits their data, it is securely transmitted to the backend, where it is processed and validated before being stored in the PostgreSQL database. The backend is responsible for managing various data formats and ensuring that data integrity is maintained. This includes checking for errors, handling data types, and converting inputs into a format suitable for storage and further processing.

Plot Generation:

During the visualization step, the backend utilizes computed polyhedron, Discrete Fracture Network (DFN), and maximum query (maxQ) data to perform calculations related to efficiency metrics. This process involves aggregating and analyzing the data to derive insights that can aid in geological assessments and decision-making. These calculations are crucial for understanding the characteristics of the geological formations being studied.

Visualization Processes (using C++ compiled as DLL):

Currently, for debugging purposes, when a user selects a represented prism for visualization, the backend directly invokes the DLL code to retrieve the corresponding visual data and send it to the frontend. This method allows for immediate testing of the DLL code, ensuring that the visualizations produced are accurate and effective. However, in the future, we plan to enhance this process significantly.

When a user inputs data, the relevant visualization calculations will be performed via RabbitMQ using multiple sub- channels. These channels will handle different aspects of the visualization process, such as represented prisms of the site, discontinuities, polyhedron, maxQ, etc. Once these calculations are completed, the graphical data, including vertices and indices, will be serialized and stored in MongoDB. This storage

vol. 31, issue 3 / 2025
pp. 82-92

method is chosen because it provides greater flexibility and ease of serialization compared to traditional relational databases. When a user wishes to view the visualized data, the backend will retrieve the relevant data from MongoDB and send it to the frontend for rendering. This workflow not only streamlines the process but also allows for efficient management of complex visual data, ensuring that users can access up-to-date and relevant visualizations based on their input.

All backend services, including RabbitMQ for calculation queuing and PostgreSQL for database management, are encapsulated within Docker containers. This containerization not only simplifies deployment but also allows for easy scaling of individual components based on demand. Additionally, this architecture facilitates easier updates and maintenance, ensuring that we can quickly adapt to evolving user needs and technological advancements.

By employing this robust backend architecture, we ensure that our platform delivers a secure, efficient, and responsive experience for users interacting with geological data.

2.4. Databases

In our application architecture, we utilize two distinct databases PostgreSQL and MongoDB to manage data effectively. Each database plays a vital role in ensuring that data is organized, secure, and accessible, catering to the needs of both the backend processing and the frontend visualization.

PostgreSQL

PostgreSQL serves as the primary relational database management system (RDBMS) in our backend. Renowned for its stability and powerful features, PostgreSQL is designed to handle complex queries and transactions, making it an ideal choice for managing user data. After users authenticate into the application, their data—such as user profiles, access rights, and settings—is securely stored in PostgreSQL.

One of the key advantages of using PostgreSQL is its ability to enforce data integrity through relationships and constraints. This ensures that the user data remains consistent and reliable throughout the application lifecycle. The structured nature of PostgreSQL allows for efficient querying and data retrieval, facilitating smooth interactions with the frontend. For instance, when users request their data or interact with different application features, PostgreSQL quickly provides the necessary information.

MongoDB

In parallel with PostgreSQL, we also employ MongoDB, a document-oriented NoSQL database that allows for flexible and scalable data storage. Unlike traditional relational databases, MongoDB is designed to handle unstructured or semi-structured data, making it an excellent fit for our visualization requirements.

The graphical data generated from the visualization processes, such as vertices and indices, is stored in MongoDB. This data is produced by calculations performed in the backend using a C++ DLL, which processes geometric computations and creates a polyhedron model. The decision to store vertices and indices in MongoDB stems from the need for flexibility and efficiency. Storing this data in a document-oriented format allows for easier serialization and faster retrieval when users request visual representations of geological data.

In the backend, once the necessary calculations are completed, the relevant graphical data is saved in MongoDB. This allows for efficient querying and retrieval of the visual data, enabling users to access and visualize complex geological structures seamlessly. When a user selects a specific geological feature for visualization, the backend retrieves the corresponding vertex and index data from MongoDB and sends it to the frontend. Here, the data is rendered in real- time, providing users with an interactive experience.

This dual-database approach, combining the strengths of PostgreSQL and MongoDB, enhances our application’s performance and scalability. While PostgreSQL manages structured user data with strong integrity constraints, MongoDB efficiently handles dynamic graphical data, ensuring that users can visualize their geological models accurately and promptly. Together, these databases create a robust data management system that supports the application’s overall functionality and user experience.

Marble Library

The Marble Library is an essential component of our application, implemented in C++ and utilizing the CMake build system. It compiles into a dynamic link library (DLL) for Windows and a shared object (SO) file for Linux servers, which are integrated with the .NET backend. This integration facilitates seamless data exchange and enhances the application's performance and usability.

The primary function of the Marble Library is to generate and process geometric data tailored for rendering in WebGL. The output consists of vertices and indices, with each vertex containing detailed information such as position, color, and normal data. The library can handle single objects or complex

 vol. 31, issue 3 / 2025
pp. 82-92

structures, like polyhedrons, which may contain thousands of individual components, each with their corresponding vertex and index data. The RPs that form the general shape of the quarry are presented in Figure 3.

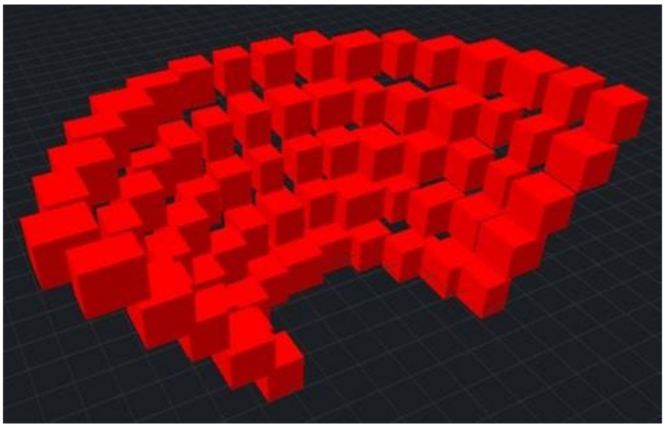

Fig.3. Representative prisms image obtained from the field

Key Operations in the Marble Library are as follows:

CreateRP: This function is responsible for creating the visual representation of a represented prism (RP). It takes several parameters, including:

Position: The 3D coordinates indicating where the RP is located in space,

Size: Dimensions of the RP, which define its scale,

Rotation: The orientation of the RP in 3D space. This information is crucial for accurately visualizing the RP in relation to other elements in the scene.

By processing these parameters, CreateRP generates the necessary geometric data that will be used in subsequent operations. Individual Representative Prism geometry and its image is presented Figure 4.

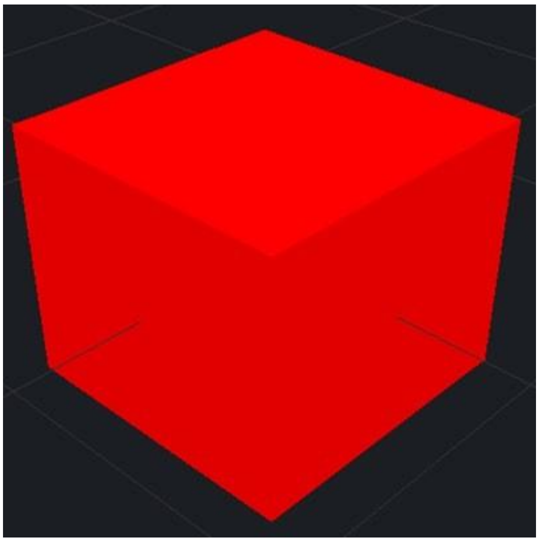

Fig.4. Individual Representative Prism geometry and its image

CreateDisc: This function models the intersections of discontinuities within the RP. It takes the following inputs:

Position: The location of the RP, which remains consistent with the input from CreateRP, **Size**: Again, the dimensions of the RP, ensuring that the discs are properly scaled, **Rotation**: The RP's orientation, which affects how the discs are aligned with the RP,

Disc Data: An array containing the positions of individual discs, along with their dip angles and dip directions. These parameters define the geometric features of the discontinuities.

The output of this function includes the visual representation of how the discs intersect with the RP, which is crucial for visualizing geological features. Individual Representative Prism Wireframe geometry and 3D image with discontinuities are presented in Figure 5.

vol. 31, issue 3 / 2025
pp. 82-92

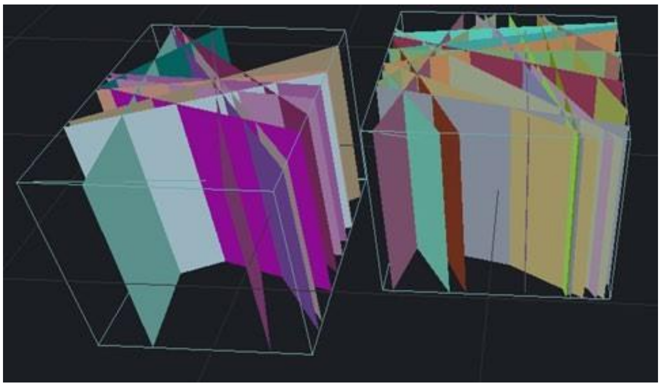

Fig.5. Individual Representative Prism Wireframe geometry and 3D image with discontinuities

CreatePolyhedron: Similar in function to CreateDisc, this operation processes the same input parameters but focuses on returning a polyhedral representation of the RP. It accounts for:

Intersection Planes: Rather than just identifying where the discs intersect, CreatePolyhedron provides a more complex visualization that includes the fragmentation of the RP caused by the discontinuities, **Volume Information**: For each fragment produced, this function also calculates and returns the volume, which is essential for backend analysis and geological assessments.

The data produced is instrumental in visualizing how the RP interacts with geological features. Solid Model Geometry and 3D Image of the Individual Representative Prism with Discontinuities are presented in Figure 6.

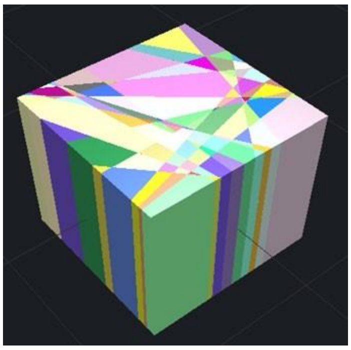

Fig.6. Solid Model Geometry and 3D Image of the Individual Representative Prism with Discontinuities

CreateDFN: This function generates new discontinuities based on the input parameters from CreateDisc. It utilizes:

Fisher Constant: A parameter used to influence the distribution of the generated discontinuities,
Distribution Type: Specifies the statistical model used for generating the new discontinuities,
Mean and Standard Deviation: These statistical parameters guide the variability and central tendency of the generated data,
Seed: A value used for random number generation, ensuring that the results can be replicated if necessary
Probability Parameter: Used to determine the likelihood of generating specific types of discontinuities based on the defined distribution.

This function enhances the complexity of the geological model by creating additional layers of discontinuities based on existing data. Discontinuity network image obtained after applying DFN on an RP is presented in Figure 7.

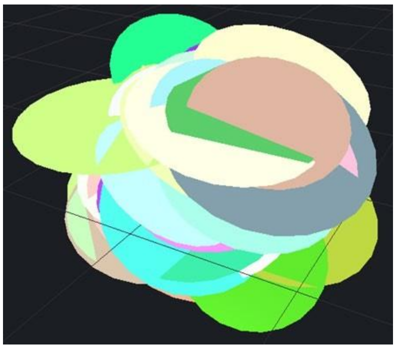

Fig.7. Discontinuity network image obtained after applying DFN on an RP

vol. 31, issue 3 / 2025
pp. 82-92

FreeData: Managing memory is a critical aspect of working with C++. Given the size and complexity of the data structures involved, this function plays a vital role in memory management. The FreeData function:

Ensures that memory allocated for large arrays is properly deallocated after the data has been transferred to the .NET backend,

Prevents memory leaks by releasing the memory addresses that are no longer needed once the processing is complete. This is crucial for maintaining application stability and performance.

3. Discussion

Marble is a natural stone widely used in construction, interior design, and architectural projects. It is extracted and processed in marble quarries using a country's natural resources. Marble mining can contribute to a nation's economy and has the potential to create employment opportunities.

The market size of marble may vary depending on the growth of the global construction industry, demand for interior design, and the volume of architectural projects. Due to its quality, color, and pattern diversity, marble is in demand across different market segments.

The employment size refers to the number of workers employed in marble quarries and processing plants. Workers in marble quarries are involved in excavation, cutting, and extraction processes, while those in processing plants participate in processes such as cutting, polishing, and marketing marble.

The marble industry is primarily a sector used in construction, interior design, and architectural projects. Computer software and computer science play a significant role in advancing technological levels.

Developments in this field increase efficiency in many industries, automate processes, and provide innovative solutions. A project that integrates the marble industry with the computer sector can further enhance this technological level and provide various benefits to the industry. For example, the development of a software module for DFN (Discrete Fracture Network) generation from discontinuity sets and web-based 3D visualization in the marble sector integrates computer software and computer science into the industry's technological framework. This study could provide many advantages to the industry by automating the traditional discontinuity set analysis and presenting it more effectively through 3D visualization.

The potential contributions of this paper to the technological level include:

- **Automation and increased efficiency**: The software can automatically analyze discontinuity sets, accelerating workflows and increasing efficiency. This results in time and cost savings in production processes.
- **Statistical analysis and predictive capabilities**: This study has the potential to improve DFN generation by utilizing statistical approaches. This could provide the ability to predict future discontinuities in marble quarries and detect potential issues in advance.
- **3D visualization**: Web-based 3D visualization allows users to better understand and visually analyze discontinuities in marble quarries. This improves decision-making processes and reduces errors.
- **Data management and analytics**: This paper can utilize advanced data management and analytics methods to collect, store, and analyze field data. This supports activities such as database creation, data mining, and trend analysis.

Such technological advancements can create more efficient and intelligent workflows in the marble industry. Additionally, they can enhance the sector's competitiveness, expand collaboration opportunities, and create new prospects.

However, before the software module proposed in this paper is directly implemented, it is crucial to conduct a detailed economic and technical evaluation, identify potential risks, and ensure strong communication and coordination with collaboration partners. This will help establish a solid foundation for the widespread impact and economic feasibility of the paper.

5. Conclusion

This study successfully demonstrates the generation and 3D web-based visualization of Discrete Fracture Networks (DFNs) from rock mass discontinuity sets. By leveraging DFN modeling techniques, the research addresses key challenges in the ornamental stone quarrying industry, including the accurate analysis of discontinuity sets and the optimization of industrial processes.

The proposed methodology integrates statistical approaches to evaluate fracture network data derived from site measurements and pre-existing datasets. The resulting DFN models provide a more detailed and precise representation of fracture geometries, orientations, and characteristics, enabling improved resource management, waste reduction, and production efficiency in quarries.

Furthermore, the development of an interactive, web-based visualization platform enhances accessibility and usability, allowing stakeholders to explore complex fracture networks in a user-friendly environment.

This platform not only supports decision-making in quarry operations but also serves as a foundation for further research and development in the field of geomechanics and hydrogeology.

In conclusion, this study contributes to the advancement of DFN modeling and visualization techniques, offering a scalable and practical solution for the ornamental stone industry. Future work may focus on incorporating real-time data integration, expanding the application to other geological settings, and exploring the potential of machine learning for automated fracture characterization.

This paper is not only an academic and scientific contribution but also a study with industrially focused outcomes and widespread impact. Marble quarries are a significant industry due to their large consumption of natural resources and energy. The DFN modeling methods and web-based visualization platform to be developed because of this study will ensure more efficient use of resources in quarries, reduce waste, and optimize energy consumption. Furthermore, the results of this paper can be applied to other industries facing similar challenges in the marble sector.

In conclusion, this paper aims to generate DFNs from discontinuity sets in the marble industry and provide 3D web-based visualization. The paper introduces significant innovations in terms of analysis and efficiency. The methodology includes field studies, data analysis, and DFN modeling techniques. The management of this paper will be carried out by a multidisciplinary team, and the results will provide valuable outcomes and a widespread impact in the marble industry.

Acknowledgement

This work is carried out within the scope of the work package called "3D Graphics Representation of Blocks and Mapping/Contouring Objectives" of the project called "Artificial Intelligence and Combined Survey Techniques for Stone Quarries Optimization (AI-COSTSQO)" supported by EU ERA-MIN2. [Project number 122N037 in Türkiye, 256 in ERA-MIN].

References

[1] **Sewnun N., du Plessis A., Sellers E.,** 2022
A review of structural data collection methodologies for discrete fracture network modelling. Proceedings of the 9th International Conference on Deep and High Stress Mining

[2] **Hyman J.D., Karra S., Makedonska N., Gable C.W., Painter S.L., Viswanathan H.S.,** 2015
dfnWorks: A discrete fracture network framework for modeling subsurface flow and transport, Computers & Geosciences

[3] **Rocscience** (n.d.)
Advancing the frontiers of blocky rock mass modelling with discrete fracture networks

[4] **Schlumberger** (n.d.)
Petrel fracture modeling

[5] **Le Goc R., Pinier B., Darcel C., Lavoine E., Doolaeghe D., de Simone S., Davy P.,** 2019
DFN.lab: software platform for Discrete Fracture Network models. American Geophysical Union Fall Meeting

[6] **Itasca Consulting Group, Inc.** (n.d.)
Discrete Fracture Network (DFN). Itasca Software 9.0 documentation.

[7] **Guiltinan E., Santos J.E., Purswani P., Hyman J.D.,** 2023
PySimFrac: A Python Library for Synthetic Fracture Generation, Analysis, and Simulation, arXiv preprint arXiv:2309.13849

[8] **Golder Associates** (n.d.)
FracMan. Retrieved from YouTube

[9] ***
FracSim3D. Retrieved from https://www.fracsim3d.com

[10] **Math2Market GmbH** (n.d.)
GeoDict. Retrieved from https://www.geodict.com