L.N.Y. Wong, Z. Liu, K.K.C. Tse, S.H. Cheung and L. Yu. (2023). A Computational Algorithm for Calculating Fracture Index of Core Runs. *Rock Mechanics and Rock Engineering*.

https://doi.org/10.1007/s00603-023-03422-z

Published in June 2023

A computational algorithm for calculating fracture index of core runs

Louis Ngai Yuen Wong[1], Zihan Liu[1]*, Keith Ki Chun Tse[1], Sai Hung Cheung[2], Lequan Yu[3]

[1]Department of Earth Sciences, The University of Hong Kong, Hong Kong, China

[2]Department of Civil Engineering, The University of Hong Kong, Hong Kong, China

[3]Department of Statistics and Actuarial Science, The University of Hong Kong, Hong Kong, China

Corresponding author: Zihan Liu (zihanliu@connect.hku.hk)

Highlights:

- The definition of fracture index in various standards and guidelines is reviewed.
- An algorithm for calculating the fracture index of core runs is proposed.
- The algorithm's application is demonstrated through three examples, yielding satisfactory results.
- Subjective judgement in manual logging accounts for the differences between algorithmic and traditional manual methods.

Abstract: Fracture Index (FI), which represents the count of fractures over an arbitrary length of core with similar intensity of fracturing, provides insight into the fracture state of rock masses. Manual interpretation of core is not only time-consuming but also prone to human error. To address these challenges, this study develops a computational algorithm that automatically calculates the FI of core runs based on fracture spacing data. The variance of fracture spacing values is selected as the key evaluating indicator. The algorithm automatically groups fractures appearing in core imagery based on uniformity of fracture spacing. Upon counting the number of fractures in each group over a certain length of core run obtained from the optimal grouping scheme, the algorithm outputs the FIs automatically. The algorithm's performance has been effectively demonstrated on synthetic data and three real-life core datasets. Algorithm results show good agreement with those produced by logging geologists and listed in

standard human-generated reports. The occasional discrepancies are attributed to the inherent subjective nature associated with the manual logging.

Keywords: Computational Algorithm, Fracture Index, Joint Spacing, Core Run

List of Symbols

$\mathbf{A}^k$	An array recording the information of each subarray
b_i	The position of the last member of each subarray in **S**
c_n	The counter of small spacing members
D^2	Variance
F	Fracture array
FI	Fracture Index
G_max	Maximum group number
G_min	Minimum group member number
k	Total number of groups
L	Core run length
m	Counter of grouping scheme
n_f	Number of discontinuities
n	Total number of members in input fracture spacing array
n_c	Critical number
s_c	Critical spacing
S	Input fracture spacing array
$\mathbf{S}^i$	Subarray of input fracture spacing array
$\mathbf{S_C}$	An array recording subarrays divided by NI segments
$\mathbf{S_{HF}}$	An array recording HF segments
$\mathbf{S_{NI}}$	An array recording NI segments
$\bar{x}$	Average spacing of the array
X	An array recording abscissa of each fracture
$\mathrm{X_{HF}}$	An array recording abscissae of fractures in HF segments
$\mathrm{X_{NI}}$	An array recording abscissae of fractures in NI segments

1 Introduction

Ground investigation (GI) is an important means of assessing land and groundwater conditions in civil engineering projects. Inaccurate GI data may result in inadequate and unsafe design that poses significant human and economic risk. Sinking boreholes in the ground is the most common GI technique. Soil/rock samples recovered from boreholes are logged manually by geologists with reference to international or national standards and guidelines.

In addition to rock type identification, logging geologists have to spend considerable time determining the fracture state of the rock core. Fracture state reflects the structural integrity and thus the engineering performance of the site's bedrock. Calculation of fracture state requires mechanical measurement of the position and inclination (relative to the core's axis) of each fracture identified. The most commonly reported quantitative fracture state indices are Total Core Recovery (TCR), Solid core recovery (SCR), Rock Quality Designation (RQD), and Fracture Index (FI). FI has also been known as fracture frequency (Vučemilović et al. 2021) and fracture intensity (Sen 2014). All these fracture state indices provide an estimate of the degree of fracturing of rock masses, and they are often collected together to constrain and verify discontinuity state. For example, Sen (2014) proposed a geomechanical classification method based on RQD and FI to characterize local zones of heterogeneities within a rock mass. A number of studies have interpreted relationship between FI and RQD (He et al. 2021; Vali and Arpa 2013). Vučemilović et al. (2021) proposed a corrected fracture index that considers the reduction of SCR in core run portions that lack solidity or were non-intact. The corrected fracture index is inversely proportional to the reduction in SCR. This allows for inclusion of crushed or weathered zones or soil layers/pockets in FI calculations. Of these, TCR, SCR, and RQD quantify the integrity of the rock mass through an overall estimation of the core run. However, they fail to record local joint spacing variation within a core run (Sonmez et al. 2022; He et al. 2020; Winn and Wong, 2019), while FI can capture the position of local changes in fracture state. A number of computer-aided algorithms have been proposed to automatically calculate these fracture state indices, with most focusing on RQD, but limited targeting FI. For instance, Olson et al. (2015) developed an RQD calculation algorithm based on 3D representations of fractures obtained using a 3D laser profile along the core. However, this method is only

applicable to fresh, intact physical cores and not to core images, which are easily stored and transferred. Saricam and Ozturk (2018) achieved the semi-automatic computation of RQD through classic digital image analysis using a shadow-based method. To eliminate the significant user biases associated with digital image processing techniques, Alzubaidi et al. (2022) proposed using computer vision and machine learning-based algorithms for drill core characterization to determine RQD assisted by a convolutional neural network model.

This study focuses on FI and begins with a review of FI definitions in published standards and guidelines. To the best of our knowledge, FI first appeared in a formal standard in 1981 as recommended by the British Standard Institution (BSI 1981). In BS5930 (2015), FI is defined as:

Count of the number or spacing of fractures over an ***arbitrary length of core of similar intensity*** *of fracturing recorded as minimum/mode/maximum. Commonly reported as Fracture Spacing (If, mm) or as Fracture Index (Fl, number of fractures per metre). Where core is non-intact in the ground, the abbreviation Nl may be used.*

The following two standards/guidelines provide similar (but not identical) definitions.

Geoguide 3 “Guide to Rock and Soil Descriptions” published by the HKSAR Government (GEO 2017) provides the following definition.

Fracture Index, FI (No./m run), is the number of clearly identifiable fractures per metre run of intact core pieces, measured over core lengths of ***reasonably uniform character****. This index does not necessarily apply to whole core runs. If there is a marked change in fracture frequency during a core run, the fracture index should be calculated for each part of the run separately. The term 'non-intact' (NI) should be used when the core is fragmented. Additional detail can be given by quoting the maximum, mean and minimum length of core pieces recovered for any core length of reasonably uniform character.*

In the EU standard ISO 14689 (2017), FI is defined as: *Fracture index (If) is the spacing*

between natural fractures along the core in zones of ***uniform character****, not per core run. The If can usefully be given as minimum, mode and maximum values within the zone.*

The Engineering Geology Field Manual published by the U.S. Department of the Interior (US Bureau of Reclamation 2001) defines *fracture frequency* as *the number of fractures occurring within a* ***unit length****. The number of natural fractures is divided by the length and is reported as fractures per foot or fractures per meter*.

FI is not defined in the ISRM suggested methods for quantitative description of discontinuities in rock masses (ISRM, 1978). However, frequency, a concept similar to FI, is defined as:
Frequency (F) defined as the number of natural discontinuities intersecting a unit length of recovered core, should be ***counted for each metre of core****. Since the orientation of the discontinuities is not considered at this stage, it is clear that differently orientated holes will usually produce different results.*

The National Standard for Engineering Classification of Rock Mass (GB50218-2014) used in China states that *the number of fractures included in a* ***one-metre-long core segment*** *is a quantitative indicator of the integrity degree of a rock mass* (based on Chinese translation).

To summarize, the literature review suggests that FI (or fracture frequency) is commonly defined in two approaches - (1) based on measurements over arbitrary core lengths of reasonably uniform character (BS5930, 2015; ISO 14689, 2017), which involves the subjective judgement of such core length, or (2) counting for each metre of core directly (ISRM, 1978; GB50218-2014). Notice that the U.S. Bureau of Reclamation (2001) does not explicitly state how to count the fractures, but *"the number of natural fractures is divided by the length and is reported as fractures per foot or fractures per meter*" implicitly implies that the counting is not necessarily performed for each metre of core, in particular when the core boxes are longer than a metre. Comparing the two approaches, the "counting for each metre of core" approach is more objective, while such regular division does not necessarily correspond to the change of fracture state in the rock mass. By contrast, the "arbitrary core lengths of

reasonably uniform character" approach can more accurately reflect the position associated with the local change of fracture state in the rock mass, but it involves subjective judgment that may introduce human errors in the process. To address this issue, developing an automatic calculation based on computer-aided algorithms is warranted.

The main purpose of this study is to develop an algorithm for automatically assigning fractures into groups (sections) of reasonably uniform character and calculating the FI value of each section automatically. The goal is to minimize uncertainties or error introduced by human judgement. The NI segments, as well as the position of the fractures and the associated fracture spacing values, are assumed to have been obtained by other means. Such fracture image recognition and ranging algorithms associated with fracture spacing data of core runs have been addressed by a broad range of literature (Jiang et al. 2022; Zhou et al. 2022), which is out of the scope of this paper.

2 Algorithm Setup

FI can be expressed as follows

$$\mathrm{FI} = \frac{n_\mathrm{f}}{L} \tag{1}$$

where n_f is the number of fractures, and L is the core run length, which is of arbitrary core length having similar intensity of fracturing (BS5930, 2015) or a core length of reasonably uniform character (GEO, 2017).

All the fractures on the core run are recorded in a one-dimension array $\mathbf{F} = \{f_1, f_2, f_3, \cdots, f_i, \cdots, f_n\}$, where n is the total number of fractures. The distance from each fracture to the previous fracture located at a shallower depth is included in a one-dimension array $\mathbf{S} = \{s_1, s_2, s_3, \cdots, s_i, \cdots s_n\}$. As such, each spacing value is associated with the fracture beneath it such that, for example, s_1 and f_1 form a pair of data.

2.1 Preprocessing

The raw input fracture spacing array **S** requires preprocessing before the main calculation procedures. First, the terminal fracture of the core run is often mechanically-induced as part of the drilling process and thus does not represent a natural feature. Since only natural fractures should be counted for FI calculation, the last member, s_n,

of the input fracture spacing array **S** is eliminated in preprocessing. Second, in conventional logging, the term 'non-intact' (NI) can be assigned to portions of the core run assessed as fragmented and lacking structural integrity (Norbury et al. 1986; GEO 2017). In the proposed method, NI segments are assumed to have been identified by an AI-based computer vision method or other methods. An NI segment (containing multiple fractures, e.g., fractures located at x_d, x_e, x_f, x_g, x_h in Fig. 1) will be simplified and represented by two spacing values namely the original fracture spacing of its first fracture, i.e., s_d in Fig. 1, and the physical length of the entire NI segment, i.e., s_H , which is the distance between the fractures located at x_d, and x_h in Fig. 1. These two values will be multiplied minus one for special labeling and be included in the raw input fracture spacing data. GI reports often list FI values larger than 20 simply as '>20' (Vali and Arpa 2013; Vučemilović et al. 2021). Here, the term 'highly fractured (HF)' is assigned to these segments with FI larger than 20. These NI segments and HF segments divide the entire core run into different segments. Each segment contains a number of sub-segments associated with different FIs. Thus, before identifying core lengths of reasonably uniform character, it is vital to divide the entire core run, i.e., the input fracture spacing array **S**, by the number of NI segments and HF segments. The raw input fracture spacing array **S** records the location of NI segments, while the other preliminary step entails identifying HF segments.

Fig.1 A sketch of transforming NI segment data

In the proposed computer algorithm, the HF segments are identified mathematically. In the input fracture spacing array **S**, an HF segment represents a core portion containing multiple, consecutive, very closely spaced fractures. Here, two parameters are introduced.

(1) s_c (cm) is critical spacing. When a spacing value s_i is equal to or less than s_c, the fracture f_i immediately behind at a deeper core depth will be flagged as a possible member of an HF segment. According to Eq. (1), FI equal to 20 can yield an average spacing of 5 cm. Therefore, s_c is set to 5 cm. If s_c exceeds this value, some normal fractures will be identified as members of HF segments, thus increasing the physical length of corresponding HF segments and decreasing the FI value of adjacent normal segments.

(2) N_c is critical number. When the number of consecutive members equal to or less

than s_c in **S** is equal to or greater than n_c (Fig. 2), these consecutive spacing members are assigned to an HF segment. These spacing members will be grouped and recorded as subarrays ($\mathbf{S_{HF}}^1$, $\mathbf{S_{HF}}^2$, $\mathbf{S_{HF}}^3$…) in $\mathbf{S_{HF}}$.

For a GI project, n_c is calibrated by trial and error until the HF segments corresponding to the GI reports are identified correctly from the input fracture spacing data. After a number of trials, $n_c = 3$ is found to be generally optimal. Larger n_c values may cause procedures to overlook HF segments with few fractures and overestimate FI values of adjacent segments.

After the HF segments are identified, the input fracture spacing array is divided by these HF and NI segments into a number of subarrays ($\mathbf{S_C}^1$, $\mathbf{S_C}^2$, $\mathbf{S_C}^3$…) for input into subsequent calculations. The subarrays are entered into the array $\mathbf{S_C}$ as rows. Figure 2 shows the schematic relationship of **S**, $\mathbf{S_C}$, $\mathbf{S_{HF}}$, and $\mathbf{S_{NI}}$. If the core run does not contain any HF or NI segments, **S** equals $\mathbf{S_C}$. A flowchart of the preprocessing procedures is shown in Figure 3.

Fig. 2 Schematic diagram of **S**, $\mathbf{S_C}$, $\mathbf{S_{HF}}$, and $\mathbf{S_{NI}}$

Fig. 3 Flow chart of preprocessing steps

2.2 Grouping scheme

Successful determination of FI requires accurate identification of core lengths of reasonably uniform character. This requires dividing the fracture array $\mathbf{F} = \{f_1, f_2, f_3, \cdots, f_i, \cdots, f_n\}$ into different groups with similar spacing values for each group. The above problem can be rephrased as dividing the spacing array $\mathbf{S} = \{s_1, s_2, s_3, \cdots, s_i, \cdots s_n\}$ into k subarrays $\{\mathbf{S}^1, \mathbf{S}^2, \mathbf{S}^3, \cdots, \mathbf{S}^k\}$. Each subarray $\mathbf{S}^i = \{x_1, x_2, x_3, \cdots, x_a\}$ contains a_i members for which the average spacing of the subarray is $\bar{x}_i = (1/a_i)\sum_{j=1}^{a_i} x_j$. A variance assesses the dispersion of spacing values within each subarray. The variance D_i^2 of $\mathbf{S}^i$ can be expressed as follows:

$$D_i^2 = \frac{\sum_{j=1}^{a_i}\left(x_j - \bar{x}_i\right)^2}{a_i}. \tag{2}$$

The optimal grouping scheme should minimize the sum of the variances of all subarrays.

$$f(k)=\sum_{i=1}^{k} D_i^2 \tag{3}$$

Note also that the first member of each subarray ($\mathbf{S_C}^1$, $\mathbf{S_C}^2$, $\mathbf{S_C}^3$…) represents the distance of the first fracture of that subarray from the last member of the previous subarray. The method thereby excludes the first spacing value in each subarray to prevent this member from biasing the variance and the overall optimization of the grouping scheme.

GI reports show that a core box often contains only a limited number of FI values. Fractures are grouped so that a group does not contain too few fractures. The algorithm approximates these practices by using two parameters described as follows.

(1) $G_{\min}$ is minimum group member quantity and represents the lower bound of the number of fractures present in a segment.

(2) $G_{\max}$ is maximum group quantity and constrains the number of different FI values to be computed over a certain core run.

Adopting these two parameters ensures a reasonable grouping of fractures while avoiding excessive computations. $G_{\max}$ is related to $G_{\min}$ as follows:

$$G_{\max}=\left\lfloor \frac{n}{2G_{\min}} \right\rfloor +1 \tag{4}$$

where the expression in brackets ($\lfloor\cdots\rfloor$) is the floor function and n is the total number of fractures. A small $G_{\min}$ will result in a large $G_{\max}$. The algorithm will then group the fractures into more small subarrays.

After manually assigning $G_{\min}$ and calculating $G_{\max}$ based on $G_{\min}$, the first step is determining the total number of groups $k, 1 \le k \le G_{\max}$, that will be divided along the entire core run. A traversal algorithm is utilized to group the fractures. The procedures of determining one single grouping scheme under the k-groups condition are described as follows. The algorithm determines the length of each subarray in sequence and extracts members from the input spacing array **S**, from left to right. The length ranges (fracture count) of the first and second subarrays are $\left[G_{\min}, n-1-G_{\min}(k-1)\right]$ and $\left[G_{\min}, n-1-G_{\min}(k-2)\right]$, respectively. After the former subarray is determined, procedures determine the length of subsequent subarrays sequentially. After all the subarrays are determined, the algorithm calculates the variance of each subarray

separately (Eq. (2)) and then takes a sum to obtain the total variance of the grouping scheme (Eq. (3)). Procedures then construct a two-dimensional array $\mathbf{A}^{k}$ (Eq. (5)) to record the position of the last member of each subarray in **S** from left to right. The last member of the row is the total variance of the grouping scheme $f(k)$.

$$\mathbf{A}^{k}=\begin{bmatrix} & & \vdots & \\ b_1^m & b_2^m & \cdots & b_{k-1}^m & f^m(k) \\ b_1^{m+1} & b_2^{m+1} & \cdots & b_{k-1}^{m+1} & f^{m+1}(k) \\ & & \vdots & \end{bmatrix} \tag{5}$$

where b is the position of the last member of each subarray in **S**, k represents the total number of groups as stated above, and m records the count of grouping schemes.

This concludes the execution of a single grouping scheme for k-groups. The same procedures are repeated to obtain results for other possible grouping schemes given k-groups with total variance for each scheme generated as output. The algorithm can then identify the grouping scheme offering the minimum total variance.

The total number of groups (k) can affect results. Calculate all grouping schemes under different conditions of total number of groups, respectively, namely dividing the input array into one to $G_{\max}$ groups along with total variance for each of these grouping schemes and k ($1 \le k \le G_{\max}$) items of minimum variance will be obtained. Then, the grouping scheme that minimizes the total variance is obtained through comparison, which is named as the optimal grouping scheme. The optimal grouping scheme provides us with the core lengths of reasonably uniform character since the fracture spacings in each subarray under this grouping scheme are similar. Figure 4 represents the grouping algorithm procedures.

Fig. 4 Flow chart showing the grouping scheme

2.3 Output

Following completion of the grouping process, the algorithm calculates the FI for each subarray of the optimal grouping scheme over the entire core run. The number of fractures or numerator of Eq. (1) is obtained by counting the number of members in each subarray. Since each member of each subarray is the distance of each fracture to the former one at a shallower depth, summing each member gives the length of each subarray. It should be noted that the length calculation given above simply approximates the length of each subarray. Additional steps refine the output as follows.

(1) The spacing separating the last fracture of the former subarray and the first fracture of the following subarray requires special consideration (Fig. 5). Standard GI reports often divide this spacing into two equal parts and allocate them separately to former and latter subarrays or segments. In keeping with professional practice, the algorithm performs a similar operation. After the approximate length of each subarray is determined, add half of the first spacing of the following subarray to the approximate length and subtract half of the first spacing of the subarray from the approximate length.

(2) There are two special situations (Fig. 5):

   a. If the subarray is the first subarray of the optimal grouping scheme, only add half of the first spacing of the following subarray to obtain the length of the first subarray.

   b. If the subarray is the last subarray of the optimal grouping scheme, only subtracted half of the first spacing of the subarray to obtain the length of the last subarray.

Fig. 5 Diagram on special consideration of core lengths

Incorporating the fine-tuning procedures as outlined above, the entire procedures of FI calculation are shown in Figure 6.

Fig. 6 Flow chart for FI calculation procedures

To present and visualize the results of the calculation results, an *X*-axis along the length of the core run is established where the upper (shallower) end of the core run is taken as the origin. To obtain the abscissa of any fracture f_i, all the spacings before or above the fracture, i.e., $\{s_1, s_2, s_3, \cdots, s_i\}$, are summed and recorded as its abscissa $x_i = \sum_{j=1}^{i} s_j$ in array **X** (Fig. 7).

The HF and NI segments require special treatment. The abscissa information of fractures present in HF and NI segments is extracted from $\mathbf{S_{HF}}$ and $\mathbf{S_{NI}}$, respectively. Then, these abscissae from **X** are coded into new arrays $\mathbf{X_{HF}}$ and $\mathbf{X_{NI}}$. Finally, delete the members corresponding to the HF segments and NI segments in **X** (Fig. 7).

Fig. 7 Diagram of **X**, $\mathbf{X_{HF}}$, and $\mathbf{X_{NI}}$

After the optimal grouping scheme is obtained, the array $\mathbf{A}^k$ (Eq. 5) can yield the position of each group in combination with respective abscissae from **X** for each fracture. On a visualization bar, fractures are represented by vertical lines, and different groups (subarrays) are represented by different colors. The last fracture (the end of the entire core run) is shown in black. HF and NI segments are shown as black rectangles. FI (rounded to one decimal place) is shown in the center below each segment.

2.4 Validation cases

The algorithm is implemented in our self-coded MATLAB program for its high efficiency in matrix calculation and powerful data visualization. Before testing the algorithm with real core run data, we tested it with three synthetic arrays of fracture spacing data. Case 1 consisted of a number of fractures with regular spacings as follows: $\mathbf{S_I}$ = {50, 50, 50, 50, 50, 50, 30, 30, 30, 30, 30, 10, 10, 10, 10, 10, 10, 40, 40, 40, 40, 40, 15, 15, 15, 15, 15, 15, 15} (unit: cm). The algorithm grouped the fractures into five segments (coloured differently in Figure 8(a)). The fractures within each group possessed the same spacings.

Case 2 included fractures with regular spacings as well as several smaller spacings representing a locally highly fractured core. Synthetic data were as follows: $\mathbf{S_{II}}$ = {50, 50, 50, 50, 50, 50, 30, 30, 30, 30, 30, 10, 10, 10, 10, 10, 10, 40, 5, 5, 5, 5, 15, 15, 15, 15, 15, 15, 15} (unit: cm). Figure 8(b) shows that the algorithm identified the HF segment accurately and grouped the remaining fractures based on uniform fracture spacing values.

Case 3 consisted of fractures with random, non-identical spacings and several small spacings representing a locally highly fractured core as follows: $\mathbf{S_{III}}$ = {50, 45, 65, 55, 40, 53, 31, 35, 38, 39, 28, 11, 15, 10, 16, 17, 14, 42, 5, 4.9, 4, 1, 12, 13, 14, 15, 11, 13, 18} (unit: cm). Figure 8(c) shows that the method identifies HF segment and groups other fractures into four groups. Within each group, the fractures showed similar spacings. These cases demonstrate the stability and accuracy of the algorithm.

Table 1 Parameters for three synthetic datasets

Fig. 8 Results of synthetic fracturing spacing data for (a) Case 1 - regular spacings (b) Case 2 - regular spacings with an HF segment (c) Case 3 - random non-identical spacings with an HF segment

3 Application and Discussion

The proposed FI calculation algorithm was also tested on real borehole data from localities around Hong Kong. The boreholes were located in Fo Tan Cottage Area (Case 1), Tung Chung (Case 2), and Kwun Tong (Case 3). The three core runs ranged in depth from 17.70 – 21.15 m, 14.06 – 17.49 m, and 114.26 – 120.45 m, respectively. The main lithology was granite. Case 1 contains 22 fractures, Case 2 contains 26 fractures, and Case 3 contains 25 fractures. Fracture inclination was not always perpendicular to the core axis but appeared as a curvilinear or irregular trace. As a simplification, the centre of each fracture is taken to represent the position of the fracture. For the three cases, G_{min} was set to three to give calculated G_{max} of four, five, and five for the respective core runs. The critical number, n_c, was set to three and s_c was set to 5 cm. Table 2 summarises the parameters for each of the three cores. By utilizing the above algorithms, the FIs of the three core runs are calculated and described below.

Table 2 Parameters for three real-world core runs from sites around Hong Kong

Case 1 (Fig. 9)

Figure 9a shows the algorithm-determined segments and FIs along with those listed in GI reports. Values are presented in separate rows similar to the rows shown in the core box image. Figure 9b shows the fracture grouping results with the location of each fracture within the core run and FIs calculated by the proposed method. These are presented in a continuous, single row for side-by-side comparison with manual GI results and core imagery.

The black block (HF segment) described in the GI report and that recognized by the algorithm occupy nearly the same position within the core run, indicating that HF segments can be satisfactorily identified by the proposed method. The algorithm estimated a FI of 4.9 for the first portion of the core run while GI reports list a FI of 4.4 (in blue). The difference between these two FIs is relatively small. Besides, the position of the top and bottom end obtained by the proposed algorithm and those stated in the GI report are almost identical. A few discrepancies appear in later (deeper) parts of the core run. The algorithm grouped the last six fractures into two parts (Fig. 9b) to give two FIs (2.2, 10.1), while the GI report divides the core run into three parts with different FIs (1.5, 13.5, 0.9). The FIs of the proposed method (2.2 and 10.1) are quite close to that of the GI report, i.e., 1.5 and 13.5.

Fig. 9 Real-world Case 1 (a) segment and FI results compared with core box photo (b) fracture grouping and FI results compared with GI report results

Case 2 (Fig. 10)

The algorithm groups the first five fractures (in blue) together with FI=4.3. By contrast, the GI report divides the same core segments into three groups and assigned different FIs (2.6, 12.0, and 1.3). These are followed by an orange segment indicating closely-spaced fractures for which algorithm results show general accordance with results in the GI report. Upon going deeper, an HF segment is obtained by the proposed method. In contrast, no HF segment was logged in the GI report. Visual examination of the core box photo indeed shows a locally fragmented region at a depth of around 16 m. This phenomenon suggests the versatility of the proposed method for identifying HF segments. Algorithmic assessment of the last two segments (purple and green) generally matched the assessment given in the GI report, although the respective FIs differ.

Fig. 10 Real-world Case 2 (a) segments and FI results compared with core box photo (b) fracture grouping and FI results compared with those in GI reports

Case 3 (Fig. 11)

The last segment assessed by the algorithm was a NI segment already identified by the AI method and included in the joint spacing array as input data. The algorithm gave results that comported with those given in the GI report. At shallower depths of 114.26 – 116.06 m, the GI report lists only a single segment (in blue) with FI = 7.6 but ignored the HF segment in the second row of the core box (Fig.11a). The algorithm instead grouped the shallower part (114.26 – 116.06 m) into three segments within the HF segment in the core box image.

Fig. 11 Real-world Case 3 (a) segment and FI results compared with core box photo (b) fracture grouping and FI results compared with those in GI reports

The above three cases illustrate the role of the subjective judgement exercised by logging geologists. The proposed grouping scheme, which is solely based on the mathematical manipulation of fracture spacing values, possesses high stability and

repeatability. It should be noted that the precondition of the proposed method is that the NI segments and the associated fracture spacing values should have been obtained by other means. The accuracy of the output strongly depends on the input data. If the NI segments or fracture positions are incorrectly identified, the errors will be transmitted and propagated to the proposed algorithm to generate irrational grouping schemes and consequently inaccurate FI values.

4 Conclusions

The Fracture Index (FI), a count of the number of fractures over an arbitrary length of core of similar intensity of fracturing, is a parameter of great importance for rock mass classification, which provide clues on the fracture state of a rock mass. Conventional determination of an arbitrary length of uniform fracture spacing character is subject to human judgement and possibly a human error. To address this issue, this study develops an algorithm to calculate FIs of core runs automatically based on fracture spacing data.

The algorithm takes a fracture spacing array with pre-identified non-intact (NI) segments as input data. In the preprocessing of fracture spacing data, highly-fractured (HF) segments are first identified. Iterative computation of fracture spacing variance in groups allows the algorithm to identify the fracture grouping scheme with the least overall variance. The optimal fracture grouping scheme assigns fractures in core runs to different continuous segments such that each has a reasonably uniform character with respect to fracture spacing. After counting the number of fractures in each group of known length, the algorithm calculates the FI for each group. The algorithm was tested on synthetic data and three real-life core runs. Results show that the algorithm generated core run segmentation and FI results are in general accordance with those listed in associated GI reports. The main differences between them are examined, which are attributed to the differences between the subjective judgement exercised by the logging geologists and the algorithm operating strictly according to mathematical rules.

Acknowledgments

This publication was made possible by the research funding from the Construction Industry Council [CICR/01/22]. Its contents are solely the responsibility of the authors and do not necessarily represent the official views of the Construction Industry Council. The authors also acknowledge the support from the General Research Fund (No. 17206822) of the Research Grants Council (Hong Kong). The second and third authors acknowledge the Postgraduate Scholarship from the University of Hong Kong.

Compliance with Ethical Standards

Conflict of interest: The authors declare that they have no conflict of interest.

References

Alzubaidi F, Mostaghimi P, Si G, Swietojanski P, Armstrong RT (2022) Automated rock quality designation using convolutional neural networks. Rock Mech Rock Eng 55(6): 3719-3734. Doi: 10.1007/s00603-022-02805-y

BSI (1981) BS 5930: Code of Practice for Ground Investigations. London, UK.

GB50218-2014 (2014) Standard for Engineering Classification of Rock Masses. China Planning Publishing House. Beijing, China. (in Chinese).

He M, Zhang Z, Li N (2021) Prediction of Fracture Frequency and RQD for the Fractured Rock Mass Using Drilling Logging Data. B Eng Geol Environ 80(6): 4547-4557. Doi: 10.1007/s10064-021-02240-3

He MM, Li N, Yao XC, Chen YS (2020) A new method for prediction of rock quality designation in borehole using energy of rotary drilling. Rock Mech Rock Eng 53: 3383-3394. Doi: 10.1007/s00603-020-02091-6

ISO 14689 (2017) Geotechnical Investigation and Testing — Identification, Description and Classification of Rock. ISO.

Jiang Q, Jia M, Bi L, Zhuang Z, Gao K (2022) Development of a Core Feature Identification Application Based On the Faster R-Cnn Algorithm. Eng Appl Artif Intel 115: 105200. Doi: 10.1016/j.engappai.2022.105200

Norbury DR, Child GH, Spink TW (1986) A Critical Review of Section 8 (Bs 5930) - Soil and Rock. Geol Soc Eng Geol Spec Publ 2: 331-342. Doi: 10.1144/GSL.1986.002.01.57

Olson L, Samson C, McKinnon SD (2015) 3-D laser imaging of drill core for fracture detection and rock quality designation. Int J Rock Mech Min 73: 156-164. Doi: 10.1016/j.ijrmms.2014.11.004

Saricam T, Ozturk H (2018) Estimation of RQD by digital image analysis using a shadow-based method. Int J Rock Mech Min 112: 253-265. Doi: 10.1016/j.ijrmms.2018.10.032

Sen Z (2014) Rock Quality Designation-Fracture Intensity Index Method for Geomechanical Classification. Arab J Geosci 7(7): 2915-2922. Doi: 10.1007/s12517-013-0975-5

Sonmez H, Ercanoglu M, Dagdelenler G (2022) A novel approach to structural anisotropy classification for jointed rock masses using theoretical rock quality designation formulation adjusted to joint spacing. J Rock Mech Geotech 14(2): 329-345. Doi: 10.1016/j.jrmge.2021.08.009

US Bureau of Reclamation (2001) Engineering Geology Field Manual, 2nd ed. US Department of the Interior, Bureau of Reclamation. US Government Printing Office, Washington, DC.

Vali B, Arpa G (2013) Finding the Relationship Between RQD and Fracture Frequency in the Different Ok Tedilithologies. Procedia Earth Planet Sci 6: 403-410. Doi: 10.1016/j.proeps.2013.01.053

Vučemilović H, Mulabdić M, Miščević P (2021) Corrected Rock Fracture Parameters and Other Empirical Considerations for the Rock Mechanics of Rock Masses of Doha, Qatar. Geotech Geol Eng 39(4): 2823-2847. Doi: 10.1007/s10706-020-01658-y

Winn K, Wong LNY (2019) Quantitative GSI determination of Singapore's sedimentary rock mass by applying four different approaches. Geotech Geol Eng 37: 2103-2119. Doi: 10.1007/s10706-018-0748-8

Zhou Y, Wong LNY, Tse KKC (2022) Novel Rock Image Classification: The Proposal and Implementation of HKUDES_Net. Rock Mech Rock Eng. Doi: 10.1007/s00603-022-03003-6

Table 1 Parameters for three synthetic datasets

Parameters	Number of Fractures	G_{min}	G_{max}	n_c	s_c (cm)
Case 1	29	3	5	3	5
Case 2	29	3	5	3	5
Case 3	29	3	5	3	5

Table 2 Parameters for three real-world core runs from sites around Hong Kong

Parameters	Depth (m)	Number of Fractures	G_{min}	G_{max}	n_c	s_c (cm)
Case 1	17.70 – 21.15	22	3	4	3	5
Case 2	14.06 – 17.49	26	3	5	3	5
Case 3	114.26 – 120.45	25	3	5	3	5

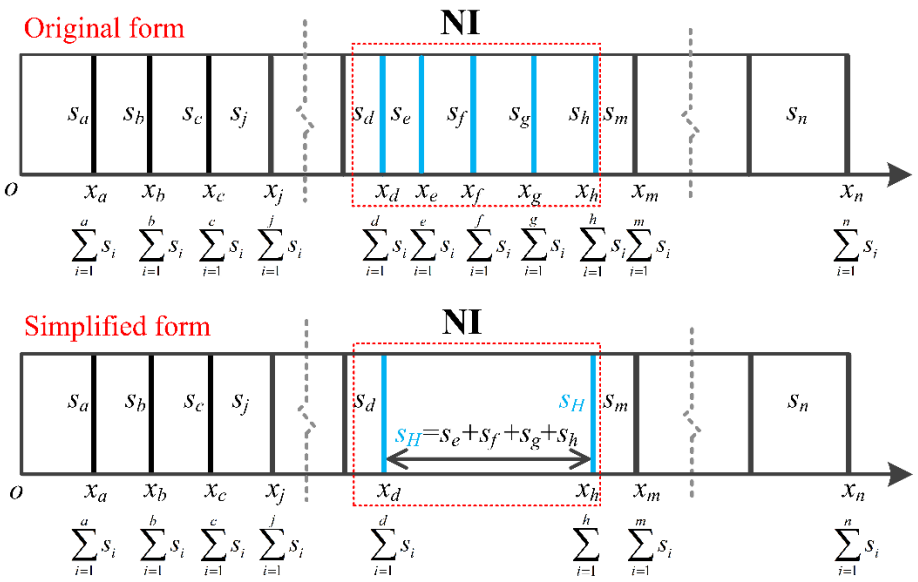


Fig.1 A sketch of transforming NI segment data

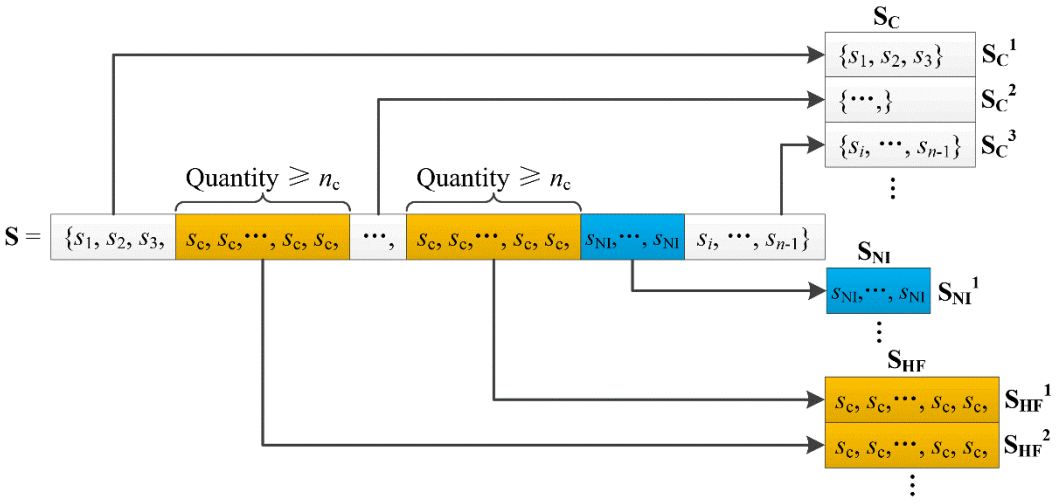


Fig. 2 Schematic diagram of **S**, $\mathbf{S_C}$, $\mathbf{S_{HF}}$, and $\mathbf{S_{NI}}$

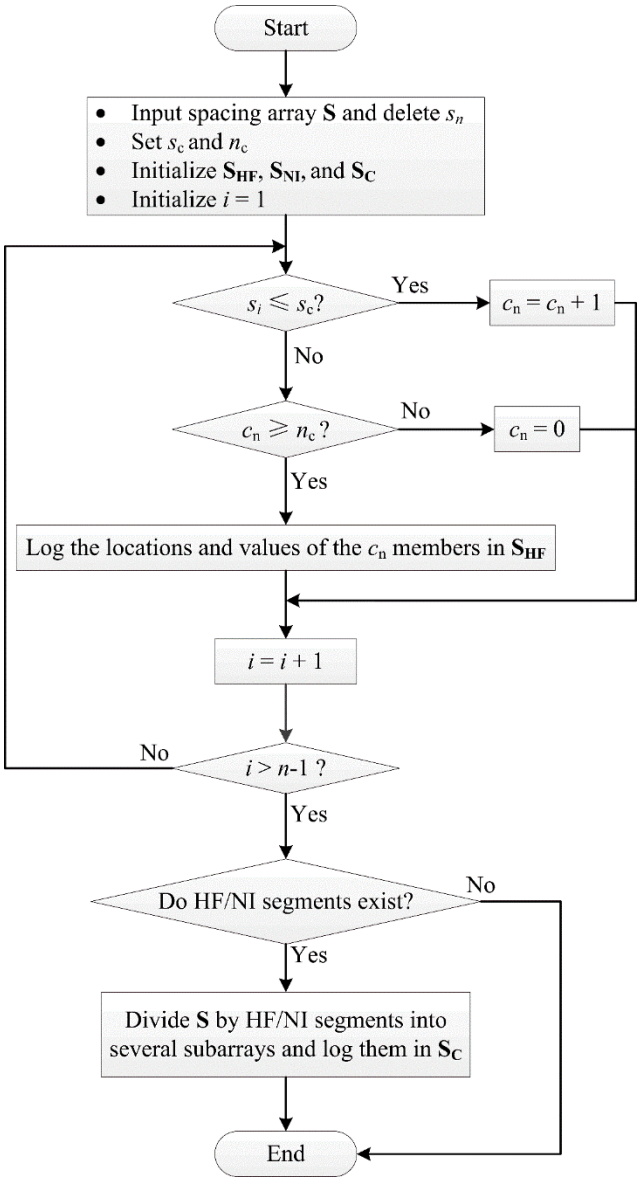


Fig. 3 Flow chart of preprocessing steps

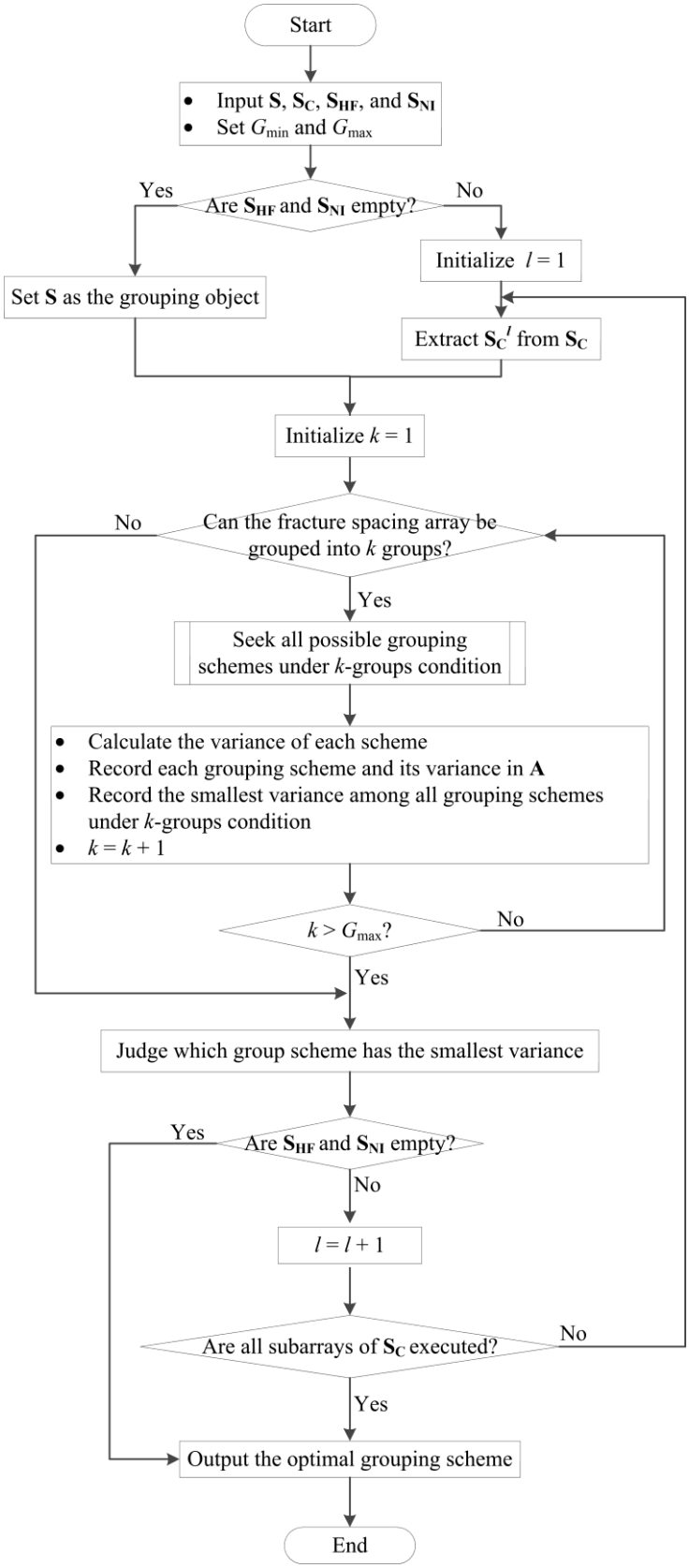


Fig. 4 Flow chart showing the grouping scheme

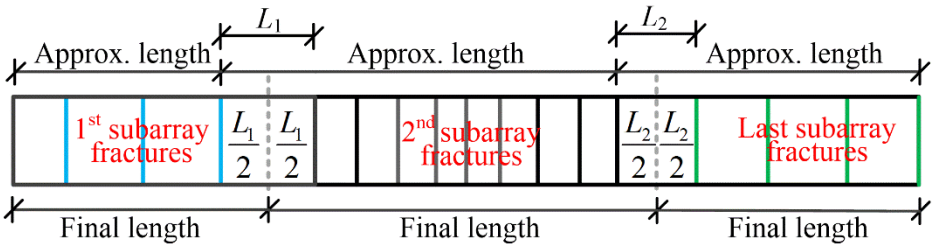


Fig. 5 Diagram on special consideration of core lengths

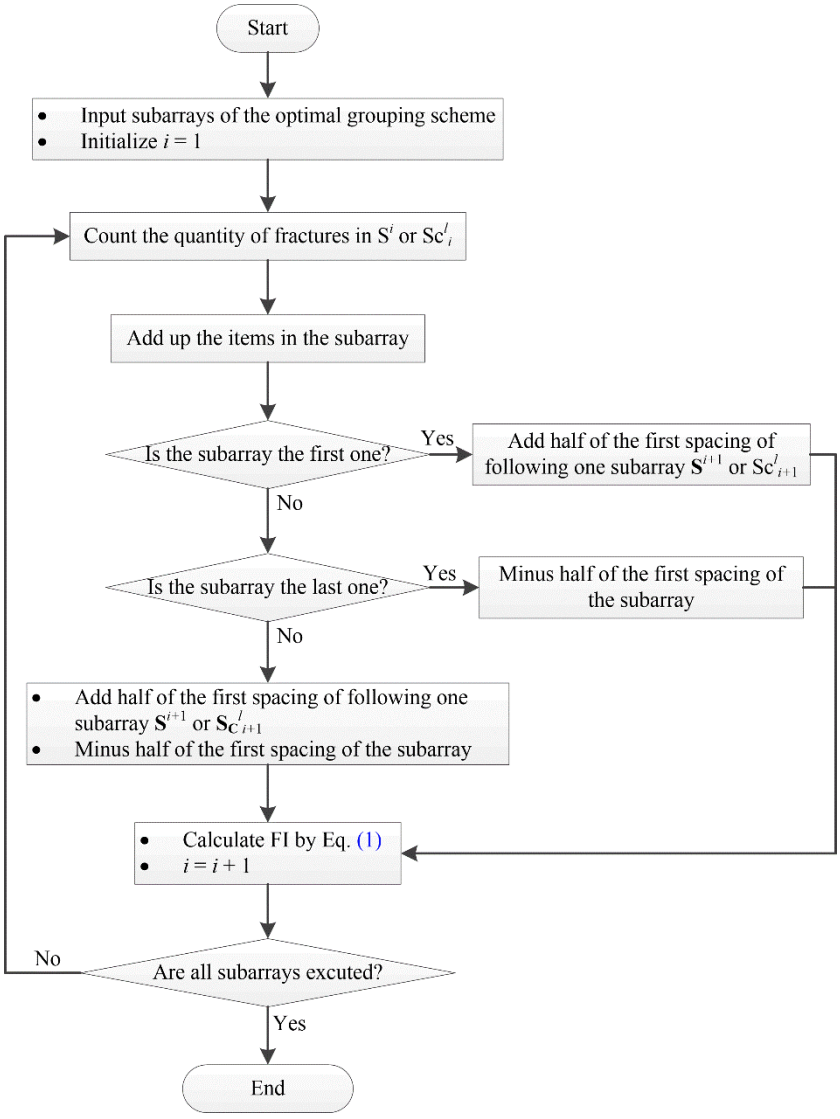


Fig. 6 Flow chart for FI calculations procedures

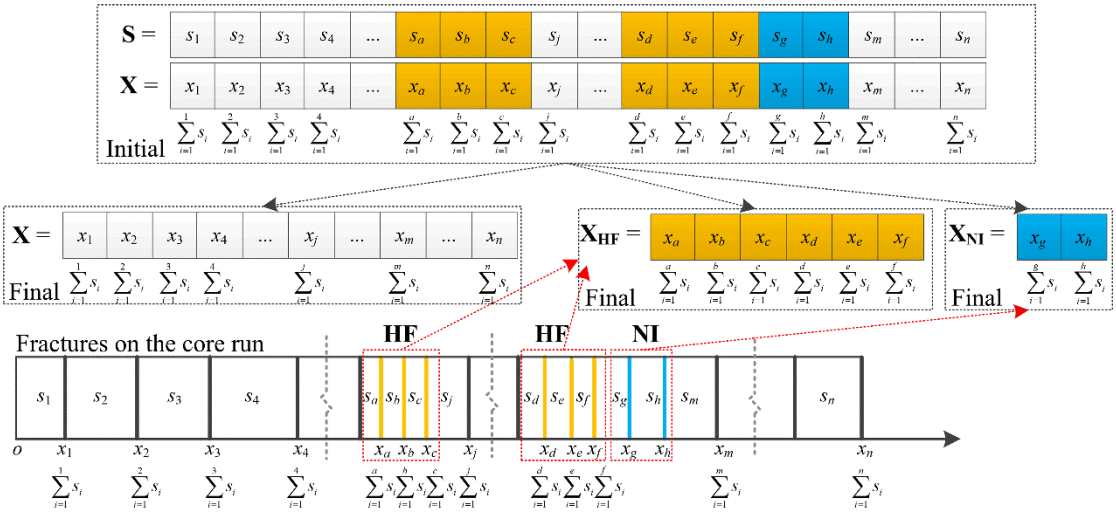


Fig. 7 Diagram of **X**, $\mathbf{X_{HF}}$, and $\mathbf{X_{NI}}$

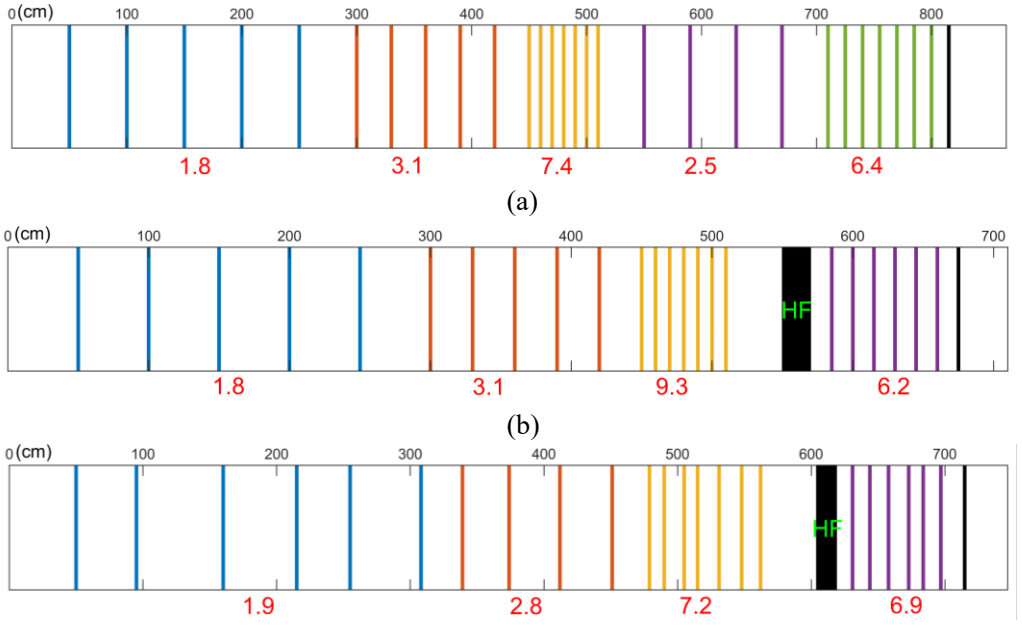


Fig. 8 Results of synthetic fracturing spacing data for (a) Case 1 - regular spacings (b) Case 2 - regular spacings with an HF segment (c) Case 3 - random non-identical spacings with an HF segment

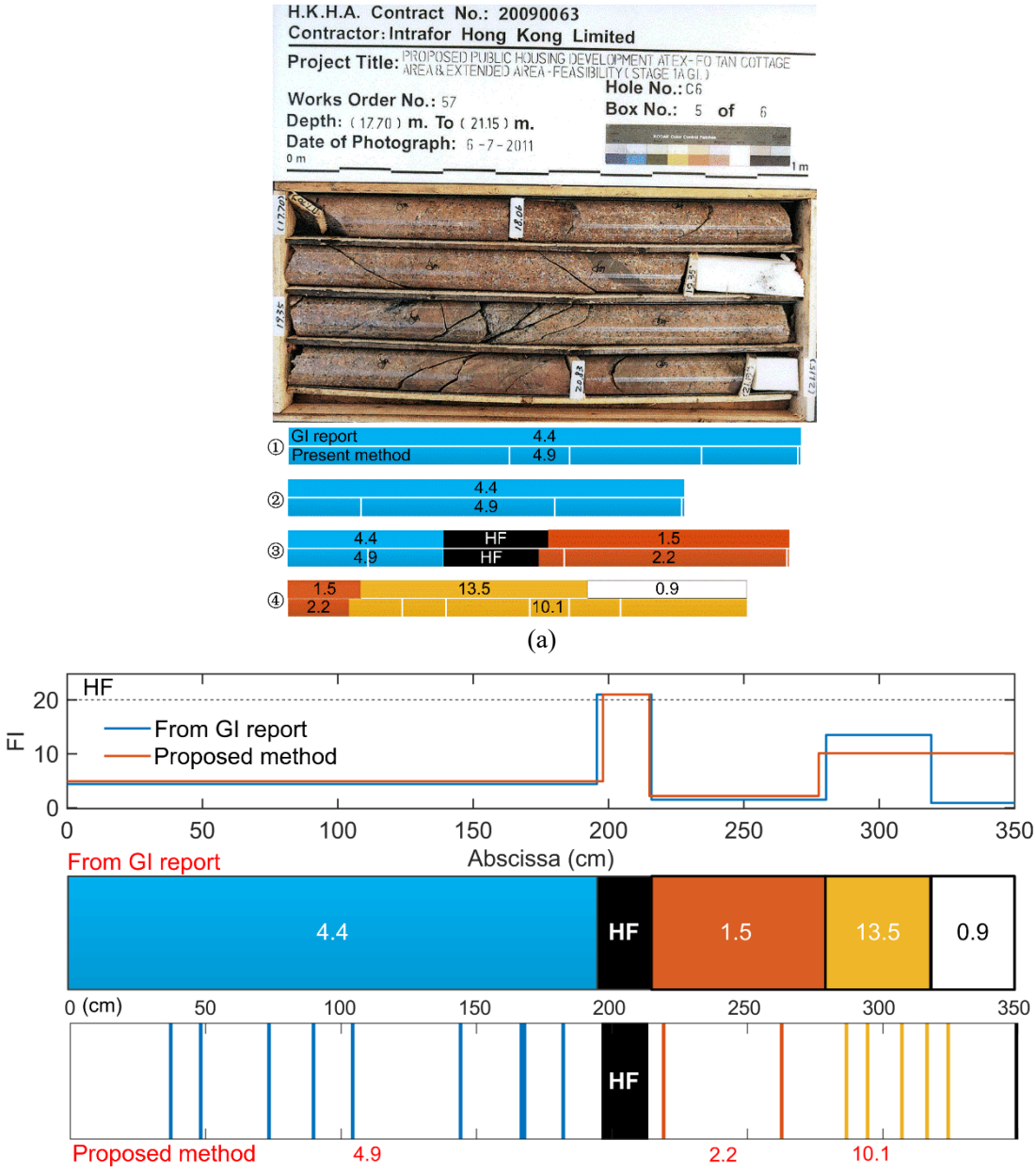
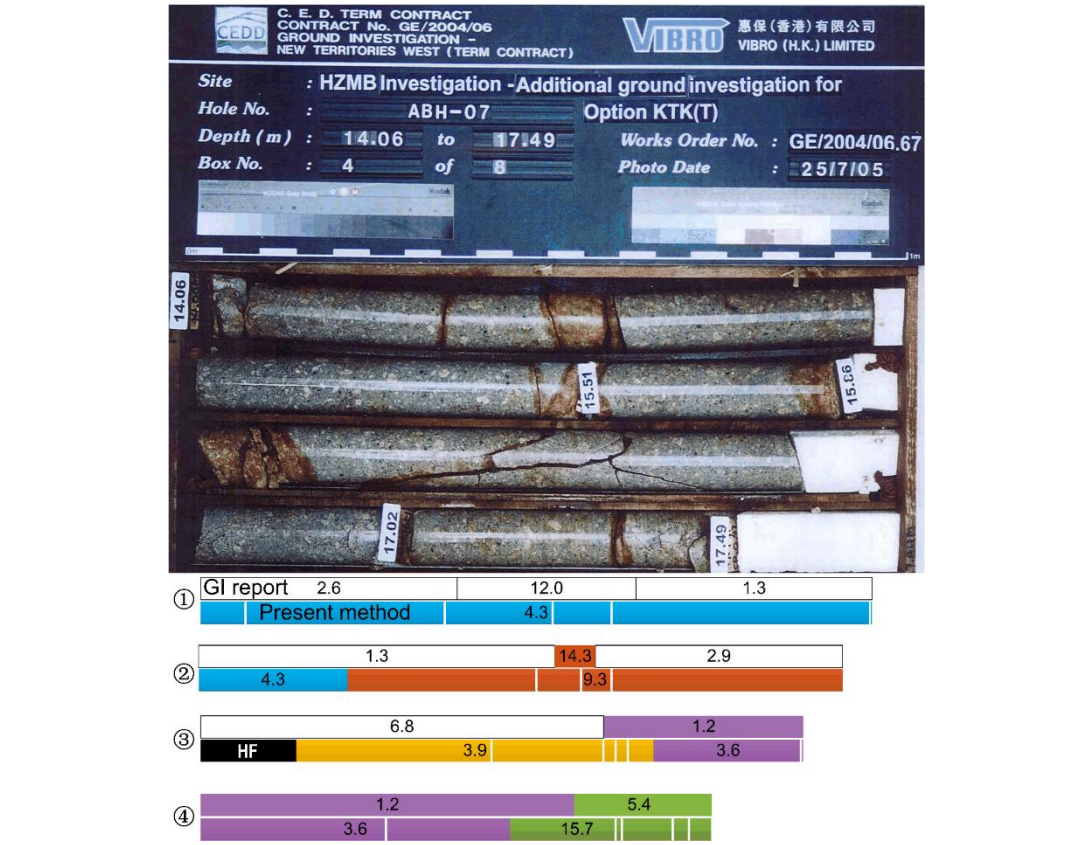
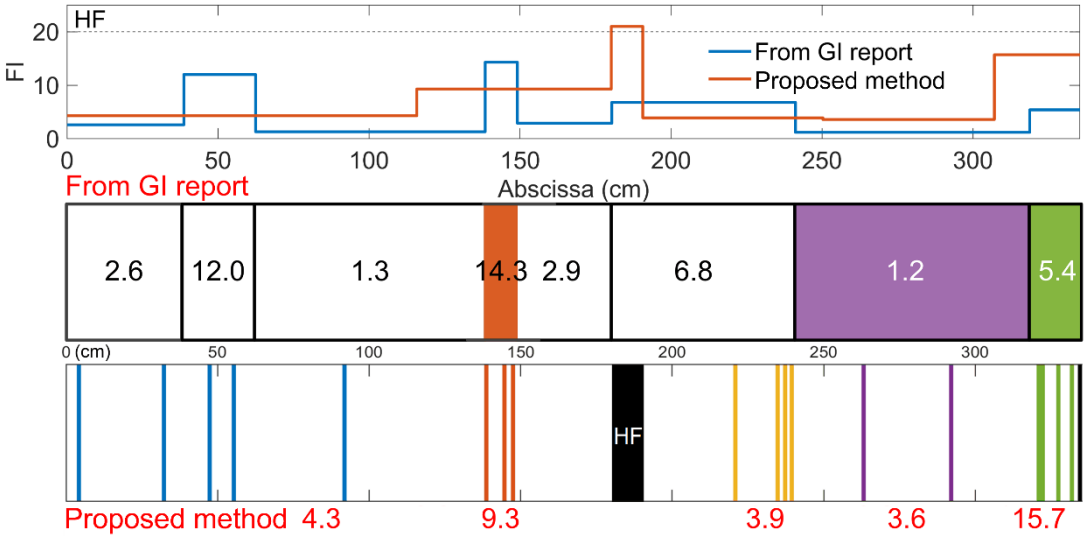


Fig. 9 Real-world Case 1 (a) segment and FI results compared with core box photo (b) fracture grouping and FI results compared with GI report results



(a)



(b)

Fig. 10 Real-world Case 2 (a) segments and FI results compared with core box photo (b) fracture grouping and FI results compared with those in GI reports

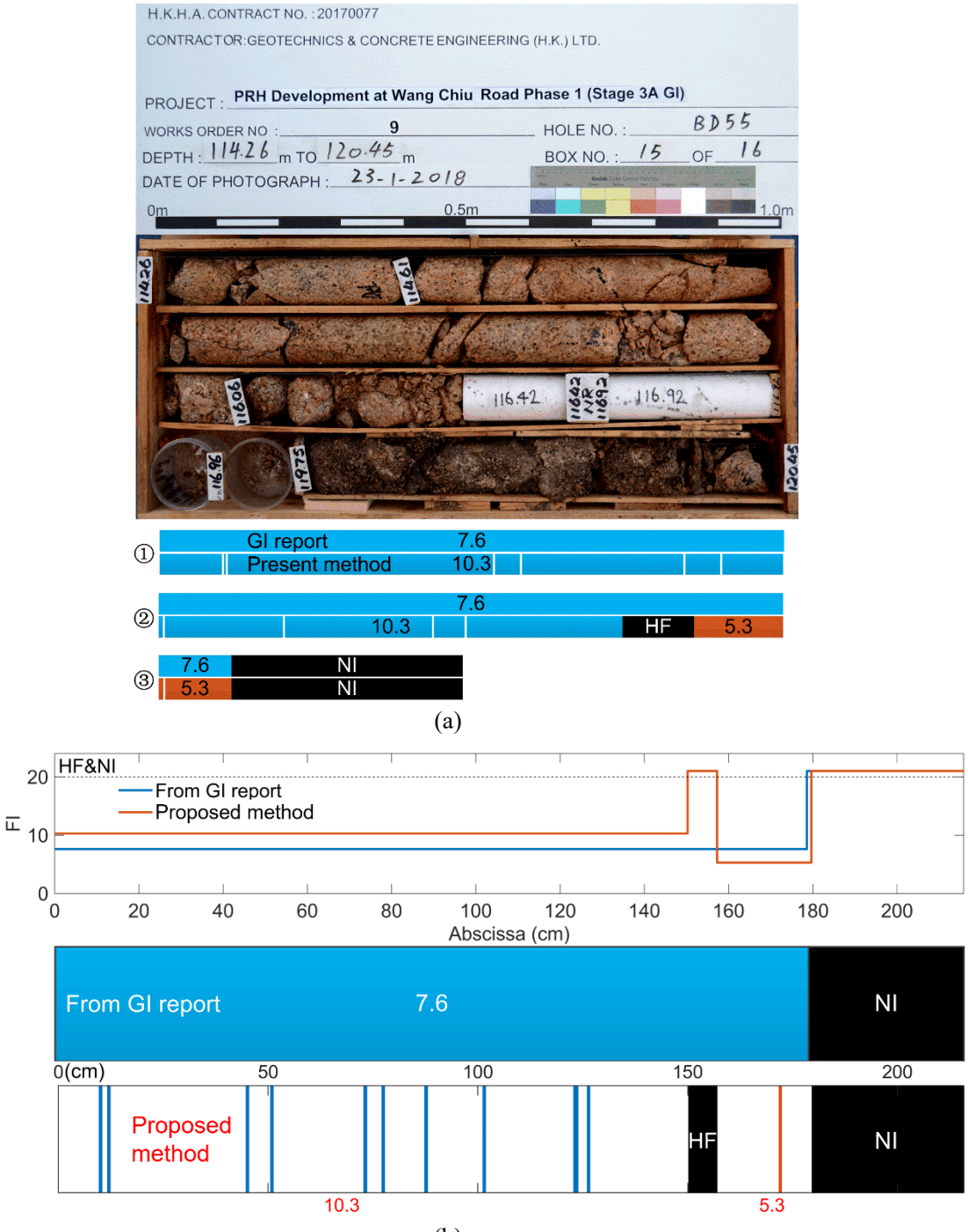


Fig. 11 Real-world Case 3 (a) segment and FI results compared with core box photo (b) fracture grouping and FI results compared with those in GI reports