Development of a GPGPU-parallelized FDEM based on Extrinsic Cohesive Zone Model with Master-Slave Algorithm

Yutaro Maeda, Division of Global Architecture, Graduate School of engineering, Osaka University, Suita, 565-0871, JAPAN

Sho Ogata*, Center for Future Innovation, Graduate School of engineering, Osaka University, Suita, 565-0871, JAPAN

Daisuke Fukuda*, Division of Sustainable Resources Engineering, Faculty of Engineering, Hokkaido University, Sapporo, 060-8628, JAPAN

Hongyuan Liu, School of Engineering, College of Sciences and Engineering, University of Tasmania, Hobart, Tasmania 7001, AUSTRALIA

Toru Inui, Division of Global Architecture, Graduate School of engineering, Osaka University, Suita, 565-0871, JAPAN

*Corresponding author, E-mail address:

ogata@civil.eng.osaka-u.ac.jp

d-fukuda@eng.hokudai.ac.jp

Abstract

This paper proposes a novel general-purpose graphic-processing-units (GPGPU) parallel computing approach to an extrinsic cohesive zone model (ECZM) - based combined finite-discrete element method (FDEM) for simulating rock fracturing. The proposed GPGPU-parallelized ECZM-FDEM incorporates a master-slave algorithm as an alternative to the complex adaptive remeshing process, which is usually used in ECZM but has prevented it from being parallelized using GPGPU. Numerical experiments of the Brazilian test and uniaxial compression test of rocks are conducted to compare the proposed ECZM-FDEM with a GPGPU-parallelized FDEM using the intrinsic cohesive zone model (ICZM-FDEM). Results show that the proposed method can not only overcome the accuracy degradation of calculated stresses and deformations that is inevitable in ICZM-FDEM but also reasonably simulate rock fracturing. Moreover, the proposed GPGPU-parallelized ECZM-FDEM achieves a maximum relative speed-up of 13 times over GPGPU-parallelized ICZM-FDEM due to efficient contact calculations and larger stable time steps. Thus, the proposed ECZM-FDEM is more physically sound and more computationally efficient compared with ICZM-FDEM, which may contribute to the further developments of FDEM.

Keywords:

Rock fracturing, 2-D FDEM, Extrinsic Cohesive Zone Model (ECZM), Intrinsic Cohesive Zone Model (ICZM), GPGPU parallel computation

1. Introduction

Reasonable numerical modelling of rock fracturing process is necessary for a variety of geotechnical applications in civil, mining, and energy fields. Recently, hybrid numerical methods that incorporate the advantages of both continuous-based and discontinuous-based methods have received significant attention. The combined finite-discrete element method (FDEM) (Munjiza, 2004) is one of the most popular hybrid methods, which combines the finite element method (FEM) and the discrete element method (DEM), and has been applied to various rock fracture problems (e.g. An et al., 2017; Elmo and Stead, 2010; Fukuda et al., 2019; Guo et al., 2016; Hamdi et al., 2014; Knight et al., 2020; Lisjak et al., 2014; Lisjak et al., 2018; Mahabadi et al., 2012; Rock field, 2023; Vlachopoulos and Vazaios, 2018; Yan et al., 2022a). FDEM is based on the explicit time integration scheme and can simulate the deformation process of continuous rocks, the transition process from a continuum to a discontinuum (i.e., fracture initiations and propagations in rocks), and the contact process between material surfaces including newly created macroscopic fracture surfaces (i.e., discontinuous deformation process). Thus, FDEM is suitable for the simulations of various engineering applications involving in highly non-linear problems which are characterized by the series of complex rock fracturing processes. Historically, two representative FDEM codes, i.e., open-source “Y-code" (Munjiza,

2004) and commercial "ELFEN code" (Rock field, 2023) were developed. Since then, various FDEM codes have been developed including "HOSS" code (e.g. Knight et al., 2020; Rougier et al., 2014), "Irazu" code (e.g. Geomechanica, 2023; Lisjak et al., 2018), “MultiFracS code” (Yan et al., 2022a, 2022b), "Solidity" code (e.g. Guo et al., 2016; Latham et al., 2012), "Y-Geo" code (e.g. Mahabadi et al., 2012; Tatone and Grasselli, 2015), and the authors’ "Y-HFDEM code" (e.g. Liu et al., 2015; Fukuda et al., 2019, 2020a) in alphabetical order among others, and the applications of these codes to rock fracture problems have been reported (see Knight et al., 2020 for a comprehensive review on the history of recent FDEM developments).

To model rock fracturing process, almost all the FDEM codes, except for ELFEN, in the recent literature utilizes the cohesive zone model (CZM) (Barenblatt, 1962; Dugdale, 1960) by separating the boundaries or inside of continuum elements and inserting cohesive elements at separated portions. In CZM, rock fracturing is modeled by the softening of cohesive tractions acting on the initially zero-thickness cohesive elements according to their relative opening and sliding, i.e., traction-separation law. Note that ELFEN models material softening by degrading tensile strengths associated with the increments of inelastic extensional strains under the assumption that quasi-brittle fracture is mainly extensional in nature, and thus does not use CZM (Klerck, 2000; Klerck et al., 2004). In ELFEN, when the degraded tensile strength reaches zero, a discrete fracture is introduced. For CZM implementations, there are three main approaches: Intrinsic CZM (ICZM), Extrinsic CZM (ECZM) (Fukuda et al., 2020b; Pandolfi and Ortiz, 2002; Papoulia et al., 2003; Zhang et al., 2007), and discontinuous Galerkin-based CZM (DGCZM) (Nguyen, 2014). Note that the so-called universal CZM (UCZM) has been proposed and implemented in HOSS code. However, since the details of the UCZM have not been publicly available in the journal papers, UCZM is not reviewed here. Hereafter, we only focus on reviewing ICZM, ECZM and DGCZM. Although all these approaches are the same in terms of the post-peak behavior in the traction-separation law, they are different in terms of the timing of inserting the cohesive elements and corresponding implementations.

Table. 1. Classification of FDEM codes according to parallelization and CZM scheme.

Parallelization schemes		**CZM schemes**	
		ICZM-based	**ECZM-based**
CPU-based	**MPI**	Lukas et al., 2014 (Y-based) Lei et al., 2014 (HOSS※)	–
	Shared Memory	Xiang et al., 2016 (Solidity)	–
GPGPU-based		Lisjak et al., 2018 (Irazu) Fukuda et al., 2019 (Y-HFDEM) Liu et al., 2019 (Y-based) Liu et al., 2021, 2022(Y-based) Yan et al., 2019, 2022a (MultiFracS)	–

※ UCZM is also available.

In ICZM, cohesive elements are inserted at all boundaries of continuum elements from the start of the simulation even if no damage has occurred. Since there is no need to update mesh connectivity (i.e., adaptive remeshing) during the simulation, it is easy to implement parallel computation schemes to enhance computing performances. In fact, the majority of the existing FDEM codes are based on ICZM (hereafter, ICZM-FDEM), and have been actively accelerated by parallel computations. Table. 1 lists the FDEM codes that have incorporated various parallel computations up to present, including FDEM codes based on ECZM (hereafter, ECZM-FDEM) which are to be discussed later in this section. As shown in Table 1, various parallel schemes have been implemented for ICZM-FDEM. Among them, the CPU-based parallelization schemes are implemented for FDEM using multiple CPUs, which includes FDEM based on relatively large-scale parallel computations using message-passing interface (MPI) (Lei et al., 2014; Lukas et al. 2014) and relatively small-scale parallel computations using shared memory programming such as OpenMP (Xiang et al., 2016). However, since the CPU-based parallelization requires multiple CPU cores, a massive computing system such as a supercomputer with enormous resources are needed to achieve high performance parallel computing. On the other hand, several cases of FDEM parallelization have incorporated general-purpose graphic-processing-units (GPGPU), which have many cores within a single GPGPU card and can be installed in a personal computer (PC) or a workstation (e.g., Fukuda et al., 2019;

Lisjak et al., 2018; Liu et al., 2019; Liu et al., 2021, 2022; Yan et al., 2022a). GPGPU-based parallelization has the advantages of a relatively lower-cost setup and cheaper energy consumption as compared to CPU-based parallelization. With this background, the ICZM-FDEM based on GPGPU parallel computation has been actively developed. However, ICZM-FDEM has a serious drawback since an artificial elastic response constrained by a finite stiffness (i.e., cohesive penalty) must be introduced to reasonably handle the continuous deformation process of rocks before any crack initiations. This drawback makes the stable time step Δt in the ICZM-FDEM analysis become much smaller than that in the explicit FEM without cohesive elements. Moreover, it causes the problem of increasing the bulk compliance of the modelled rocks since the cohesive elements can open/slide even in the intact/continuous deformation regime (see the discussion in Fukuda et al., 2020b in the case of rock dynamics problems). Furthermore, because the cohesive elements are inserted in the whole domain from the beginning of the simulation, one controversial issue arises: when should the contact processing (i.e., the contact detection and contact force calculation) by DEM be initiated for the continuum elements located inside the solid body? One of the solutions is the brute-force contact activation approach (BCAA), in which the contact is processed for all separated continuum elements from the start of the simulation and which is used in many ICZM-FDEM codes. However, Fukuda et al. (2021) and Mohammadnejad et al. (2020) pointed out BCAA was not only physically unreasonable but also required enormous computational costs, which may still be too much even with the parallel computation. On the contrary, the adaptive contact activation approach (ACAA) utilized by Guo (2014) and semi-ACAA (Fukuda et al., 2021) only activate the contact processing at locations where the cohesive elements are completely/partially broken (i.e., physical cracks appear). Through this way, the shortcomings of BCAA mentioned above are solved, and ACAA and semi-ACAA can succeed in reducing the computational cost significantly compared to BCAA. In summary, ICZM-FDEM has the extremely attractive aspect of easy parallelization but also suffers from the drawbacks mentioned above.

One solution to these problems inherent in ICZM-FDEM is ECZM-FDEM, in which, the cohesive elements are inserted on the boundaries of the continuum elements only when and where the given failure criteria are satisfied. In this sense, ECZM-FDEM is simply a pure FEM without any cohesive elements at the intact/continuous deformation regime of rocks. Therefore, the problems of too small stable time step and increasing bulk compliance inherent in ICZM-FDEM never occur. Moreover, because the contact calculations are unnecessary inside the solid body before the cohesive elements are inserted, the concept of BCAA does not appear in ECZM-FDEM, while only ACAA or semi-ACAA is applicable. Thus, the excessive computation costs inherent in ICZM-FDEM are overcome in ECZM-FDEM. However, the developments and applications of ECZM-FDEM are extremely limited to present and most of them are based on the complex adaptive remeshing required when inserting cohesive elements during the calculation in ECZM-FDEM (e.g. Fukuda et al., 2020b; Rock field, 2023) including ECZM-based pure FEM (Fukuda et al., 2020b; Pandolfi and Ortiz, 2002; Papoulia et al., 2003; Zhang et al., 2007). For example, Fukuda et al. (2020b) implemented a three-dimensional (3-D) ECZM-FDEM and applied it to model the dynamic tensile fracture tests of rocks utilizing sequential Fortran 90. However, the remeshing algorithm used in that code is sequential and it has been challenging or impossible to directly extend it to the GPGPU parallelization. Even in the reported cases of ECZM-based pure FEM, only MPI parallel implementations (Espinha et al., 2013; Dooley et al., 2009) have been reported. In sum, as shown in Table. 1, there are no applications of GPGPU parallel computation to ECZM-FDEM till this moment.

In addition to ECZM for the possible solution to the inherent issues in ICZM-FDEM, DGCZM "weakly enforces the continuity of the displacements across cohesive elements at the undamaged state which are active in ICZM-FDEM" using two control parameters θ_{DG} and α (refer to Nguyen, 2014 for the explanation of these two parameters), and thus the GPGPU parallel implementation of DGCZM is very easy as that of ICZM. However, there has been no research on the application of DGCZM to FDEM, and thus further research is needed. Besides, as implied by the point "weakly enforce the continuity", a slight reduction in the precision of modelling the continuous deformation process using DGCZM is inevitable compared to ECZM although DGCZM may bring about a significant improvement over ICZM for the modelling of the continuous behavior.

In view of the literature review above, it should be significantly valuable if GPGPU-parallel ECZM-FDEM can be developed with any relative ease. To achieve it, this study attempts to extend the concepts proposed in Cai et al. (2023), Dooley et al. (2009), Maeda et al. (2022) and Woo et al. (2014 & 2019) to GPGPU parallel computation. Dooley et al. (2009) proposed a 2-dimentional (2-D) ECZM-based pure FEM in MPI parallel computing framework, in which all boundaries of continuum elements are physically separated as those in ICZM at the onset of the simulation. In this method, all nodes at the same location (hereafter “detached nodes”) are conceptually tied with each other, which includes not only the initial FEM nodes but also these nodes generated from the insertion of cohesive elements. Suppose a FEM node “i” before the insertion of any cohesive elements is detached into a group of N detached nodes “i_0”~”i_{N-1}” (hereafter “node group”) generated after the insertion of cohesive elements. One detached node is then considered as the representative node, i.e. master node (M-node) while the all other detached nodes are set as copy nodes, i.e. slave nodes (S-nodes) in each node group. After that, pure FEM simulation can be achieved by assembling the masses and nodal forces of all S-nodes in the same node group into the corresponding M-node and solving the equations of motion for this M-node. In this way, the cohesive elements can be completely dormant before any crack initiation. Dooley et al. (2009) further demonstrated that the cohesive elements could be adaptively inserted/activated by updating the relation between M-node and S-node (M-S relation) “i_0”~”i_{N-1}” in the same node group. Unfortunately, the updating algorithm of the M-S relation (hereafter M-S algorithm) was not sufficiently described in Dooley et al. (2009)

although it was the most crucial part of the proposed method. Later, Woo et al. (2014 & 2019) proposed a method very similar to ECZM, which was called as the selective activation of ICZM. As in Dooley et al. (2009), this method constrains the detached nodes belonging to the same node group to a representative node (multi-point constraints, i.e., MPCs) to realize a calculation accuracy equivalent to pure FEM. Then, the MPCs surrounding the continuum element boundary where fracture is "likely" to occur are released. However, this method has not been parallelized. In addition, neither Dooley et al. (2009) nor Woo et al. (2014 & 2019) focused on FDEM. More recently, Cai et al. (2023) and Maeda et al. (2022) extended the M-S algorithm originally proposed by Dooley et al. (2009) to FDEM to develop ECZM-FDEM for rock mechanics applications. However, only sequential computations are implemented in both Cai et al. (2023) and Maeda et al. (2022) while a GPGPU parallel implementation of ECZM-FDEM based on the M-S algorithm has not been achieved yet. In view of the background reviewed above, this paper proposes a GPGPU-parallelized ECZM-FDEM on the basis of the M-S algorithm in Cai et al. (2023), Dooley et al. (2009) and Maeda et al. (2022) without adaptive remeshing and implements it in Y-HFDEM code to simulate rock fracturing process.

The remaining of this paper is organized as follows. Section 2 describes the methodology and numerical implementation of GPGPU-parallelized ECZM-FDEM with the M-S algorithm. Section 3 verifies the GPGPU-parallelized ECZM-FDEM by applying it to simulate the rock fracturing process in the conventional laboratory tests. The obtained results are discussed through various comparative studies between the GPGPU implementations of both 2-D ICZM-FDEM (Fukuda et al., 2019) and 2-D ECZM-FDEM with M-S algorithm. Section 4 concludes the achievements of this study and points out the issues for future study.

2. GPGPU-based ECZM-FDEM with Master-Slave Algorithm

2.1. ECZM-FDEM with Master-Slave Algorithm

In this study, the GPGPU-parallelized ECZM-FDEM is realized in Y-HFDEM code (Liu et al., 2015; Fukuda et al., 2019) by newly incorporating the powerful algorithm that can fully consider the features of ECZM without complicated adaptive remeshing. This is achieved by utilizing the GPGPU-parallelized ICZM-FDEM utility which is already available in Y-HFDEM code (Fukuda et al., 2019). This paper only focuses on 2-D problems and its 3-D extension is considered as our future task. Although all simulations in this paper are conducted under the plane strain condition, it should be emphasized that the proposed M-S algorithm is applicable under both the plane strain and the plane stress conditions. The tensile and compressive stresses are regarded as positive and negative, respectively, which holds true throughout the paper unless otherwise stated.

FDEM has to deal with the following three important processes in order to simulate rock fracturing, (i) continuous deformation of the intact rock, (ii) transition from continuum to discontinuum (i.e. fracture initiation and propagation) and (iii) contact between solid surfaces including newly generated discontinuities upon rock fracturing. FDEM models the three processes (i)~(iii) through continuum mechanics, non-linear fracture mechanics based on CZM (Barenblatt, 1962; Dugdale, 1960) and contact mechanics (Munjiza, 2004), respectively. The difference between the traditional ICZM-FDEM and the proposed ECZM-FDEM with M-S algorithm (hereafter, MS-ECZM-FDEM) lies mainly in the handling of CZM and the process of assembling nodal forces from S-nodes to their M-nodes, which will be explained below.

In MS-ECZM-FDEM, the nodal masses and nodal forces are calculated through the computation of the aforementioned processes (i)~(iii) in each time step. The core idea of MS-ECZM-FDEM is that the all the continuum elements (which are 3-node triangle elements (TRI3s) in this study) are already detached by the insertion of initially zero-thickness 4-node cohesive elements (CE4s) at the onset of the simulation which is exactly same as the ICZM-FDEM (see Figs. 1 and 2). Thus, the proposed developments can be easily implemented into any existing ICZM-FDEM codes such as open source Y-code (Munjiza, 2004) and GPGPU-based Y-HFDEM code (Fukuda et al., 2019). The nodes generated upon the insertion of CE4s are "detached nodes" as mentioned in Section 1. In addition, Fig. 3 shows the concept of "node group" mentioned in Section 1, which consists of the detached nodes originally belonging to the same FEM node before the insertion of CE4.

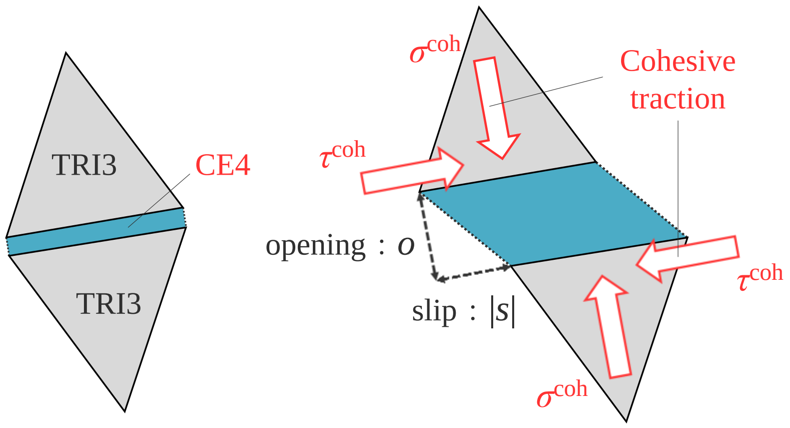

Fig. 1. Topological relation of TRI3s and CE4 (left) and overview of cohesive tractions with respect to opening and slip of CE4 in CZM (right).

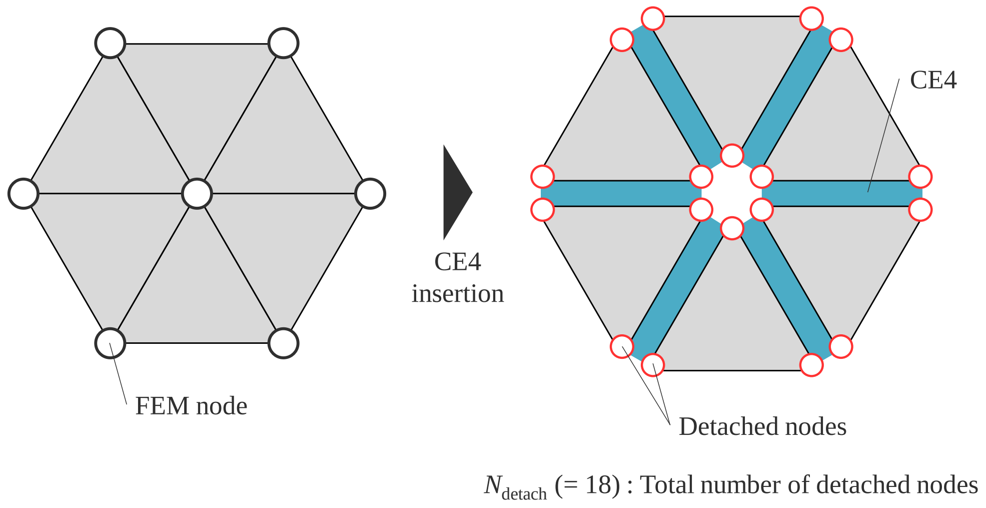

Fig. 2. The concept of detached nodes.

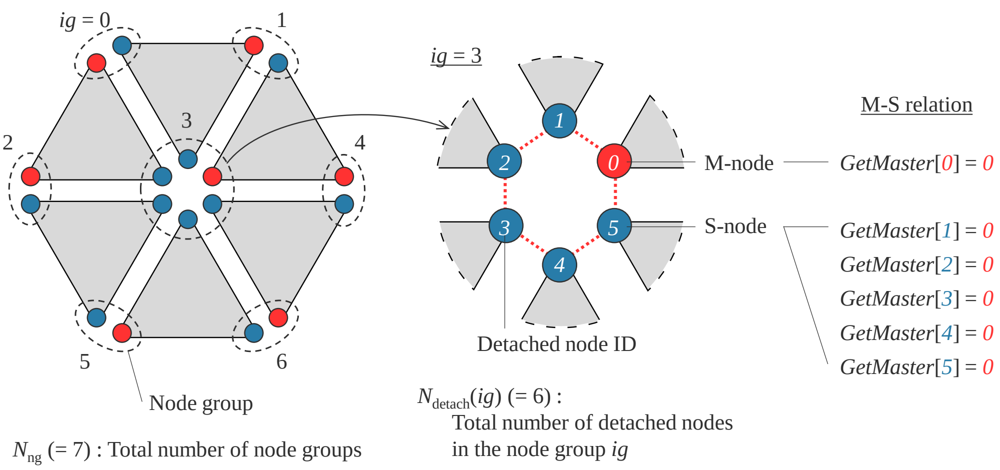

Fig. 3. The concept of node groups and M-S relation.

With these concepts in mind, we first explain how the aforementioned processes (i) and (iii) are modelled in the framework of MS-ECZM-FDEM while the treatment of the process (ii) is provided later. Let us first explain the case that the target problem only consists of multiple discrete bodies "without any fractures", in which each discrete body is purely continuous and only the continuous deformation of TRI3s in each discrete body (process (i)) and the contact between the discrete bodies (process (iii)) are involved. Since all nodes are detached in the MS-ECZM-FDEM, let N_{detach} be the total number of detached nodes in the target system. It should be noted that N_{detach} is the same as the total number of nodes in the case of ICZM-FDEM. Let us consider a single node group "*ig* (= 0,1,2, ····, N_{ng}-1)" where N_{ng} is the total number of node groups in the system, and N_{ng} is exactly same as the total number of FEM nodes before the insertion of CE4s. The node group *ig* consists of $N_{\mathrm{detach}}(ig)$ detached nodes where $N_{\mathrm{detach}}(ig)$ can be readily available from any existing ICZM-FDEM codes. To realize the pure continuous deformation within each discrete body, we apply the following M-S algorithm. In each node group *ig*, a single detached node is considered as the M-node while all other detached nodes in the same node group *ig* are assigned as the S-nodes to the M-node of *ig* (see Fig. 3). In terms of implementations, we first introduce a key data structure for the M-S node relation named "*GetMaster*[*i*]" for each detached node *i* (= 0,1,2, ····, N_{detach}-1) which literally stores the information of the M-node of the detached node *i* (see Fig. 3). The following rules are then assigned. If a detached node *i* satisfies the condition "*GetMaster*[*i*]= *i*", *i* is the M-node. Otherwise, if a detached node *i* satisfies the condition "*GetMaster*[*i*]= *j* ($j \neq i$)", *i* is the S-node whose M-node is *j*. The construction of "*GetMaster*" is the key to the successful implementation of MS-ECZM-FDEM. For the stage involving with no fracturing, "*GetMaster*" can be readily constructed using the existing ICZM-FDEM code. In addition, at the onset of the FDEM simulation, initial and current nodal coordinates as well as initial nodal velocities are set to be same between a M-node and their S-nodes in each node group. For the aforementioned process (i), this study assumes that each TRI3 obeys the isotropic hyper-elastic solid with viscous damping (see. Eqs. (2) and (3) in Fukuda et al., 2020a) under plane-strain condition and Cauchy stress tensor σ_{ij} is computed in each TRI3. Then, σ_{ij} is converted to the equivalent nodal force $\mathbf{f}_{int}$ [N], and $\mathbf{f}_{int}$ is assembled to each detached node in the TRI3. When each TRI3 is processed, lumped nodal mass **M** [kg] is also computed and is assembled to detached nodes in the TRI3. It must be noted that, at this stage, the assembling of nodal force is processed on the basis of each detached node and we can directly utilize the existing ICZM-FDEM code without any modification. For the aforementioned process (iii), the contact between two discrete bodies are handled by that between the elements, i.e., TRI3s in this study, used to discretize the two discrete bodies based on the potential contact force theory proposed by Munjiza (2004). When any overlap between two TRI3s is detected, the exact overlapping shape is computed.

Correspondingly, the repulsive normal contact forces are computed based on the contact potential, which is determined from the overlapping area (see Munjiza (2004) for full detail), along with the contact friction force based on the Coulomb type friction law. Then, the computed contact force is converted to the equivalent nodal force $\mathbf{f}_{con}$ [N], which is assembled to each detached node by directly utilizing the existing ICZM-FDEM code without any modification. If any external load (such as water pressure or gravity) is involved in the target problem, the equivalent nodal force $\mathbf{f}_{ext}$ [N] is assembled to each detached node in the same way.

Finally, using the aforementioned M-S relation "*GetMaster*", $\mathbf{M}$, $\mathbf{f}_{int}$, $\mathbf{f}_{ext}$ and $\mathbf{f}_{con}$ of detached nodes are assembled to their M-nodes, and the resultant equation of motion only for each M-node is solved in the explicit time integration scheme as given by Eq. (1):

$$\mathbf{M}\frac{\partial^2\mathbf{u}}{\partial t^2} = \mathbf{f}_{ext} - \mathbf{f}_{int} + \mathbf{f}_{con} \quad (1)$$

where $\mathbf{u}$ [m] is the nodal displacement and t [s] is the time. After the update of nodal information for M-nodes, the updated nodal velocity and current nodal coordinates of each M-node are copied to their S-nodes for the following timesteps to utilize the existing ICZM-FDEM code without any modification. In this way, the processes (i) and (iii) can be simulated as if no active CE4s exist in the system and each discrete body behaves as pure continuum as that in pure FEM. Using this approach, the issue of the increase of the bulk artificial compliance in the ICZM-FDEM can be completely overcome. Moreover, since the CE4s become completely dormant at this stage (Hereafter, these dormant CE4s before the onset of crack initiations are simply called "dormant CE4s"), we only need to consider TRI3s on the surface of each discrete body as potential contact candidates subjected to the contact detection. Thus, the concept of BCAA utilized in FDEM literature can be completely avoided. As long as each discrete body is intact without any fracturing, the M-S relation "*GetMaster*" does not need to be updated. The important remaining tasks are "how the dormant CE4s are activated to model the crack initiation", "how the M-S relation *GetMaster* is updated upon crack initiation" and "how the ECZM is implemented in FDEM" in the framework of GPGPU computing, which are explained in the remainder of this section.

The activation timing of the dormant CE4s is determined based on the normal and shear tractions acting on the boundary of two TRI3s where the target dormant CE4 is located. Note that the dormant CE4 is exactly the line element coinciding with the boundary of the TRI3s. Thus, we interpolate the Cauchy stress tensor $\boldsymbol{\sigma}$ on the boundary of TRI3 by taking the average of those in the surrounding two TRI3s. Let $\mathbf{n}$ denote the outward unit normal vector of the boundary of TRI3, the normal traction ($\sigma_{\mathrm{n}} = (\boldsymbol{\sigma}\mathbf{n}) \cdot \mathbf{n}$) and shear traction ($\tau_{\mathrm{n}} = \|\boldsymbol{\sigma}\mathbf{n} - \sigma_{\mathrm{n}}\mathbf{n}\|$) acting on the boundary can be calculated. When these values reach either the given tensile strength or Mohr-Coulomb shear strength set at the boundary of TRI3, tensile or shear failures occur, respectively, and the dormant CE4 is activated. The failure criteria are presented as follows:

$$\begin{cases} F_1 \equiv \sigma_{\mathrm{n}} - f_{\mathrm{t}} \\ F_2 \equiv |\tau_{\mathrm{n}}| - \langle c - \sigma_{\mathrm{n}} \tan\phi \rangle \end{cases} \tag{2}$$

where f_{t} [Pa], c [Pa] and ϕ [degree] are tensile strength, cohesion, and internal friction angle of a CE4, respectively. Through Eq. (2), crack initiation, i.e., the activation of the dormant CE4, is assumed to occur when either $F_1 \geq 0$ or $F_2 \geq 0$ is satisfied.

Once any dormant CE4s are activated, the M-S relation "*GetMaster*" must be adaptively updated. We follow the M-S algorithm introduced by the authors (Maeda et al., 2022). As depicted in Fig. 4, the counterclockwise search around the reference axis pointing the positive x direction is performed for the node group ig (=0,1,2, ···, N_{ng}-1). The detached nodes in each node group are sorted counterclockwise based on the geometric center of TRI3s to which each detached node member in the same node group belongs. Starting the search from one of the dormant CE4s (CE4_0 in Fig. 4) which has just satisfied the aforementioned failure criteria, the first encountered detached node immediately after passing across the CE4_0 is considered as the first M-node (M0 in Fig. 4). Then, before the searching passes across the other activated CE4 (CE4_1 in Fig. 4), all the encountered detached nodes are assigned to S-nodes (S0 in Fig. 4) to M0. Then, in the similar manner to M0, the first encountered detached node immediately after passing across the CE4_1 is considered as the second M-node (M1 in Fig. 4). In this way, any number of activations of the dormant CE4s which satisfy the failure criteria in the corresponding time step can be handled with ease. Besides, this computation can be localized to each node group. Thus, by updating the M-S relation in each node group, very complex topological change due to fracturing can be automatically traced without using complex adaptive remeshing as in Fukuda et al. (2020b) (see Fig. 3 therein) and corresponding data structure for manipulating remeshing, which makes the parallelization of the algorithm very easy. Thus, the advantage of the applied M-S algorithm is not only limited to the localization of processing by avoiding remeshing but also its ease of implementation and saving memory usage.

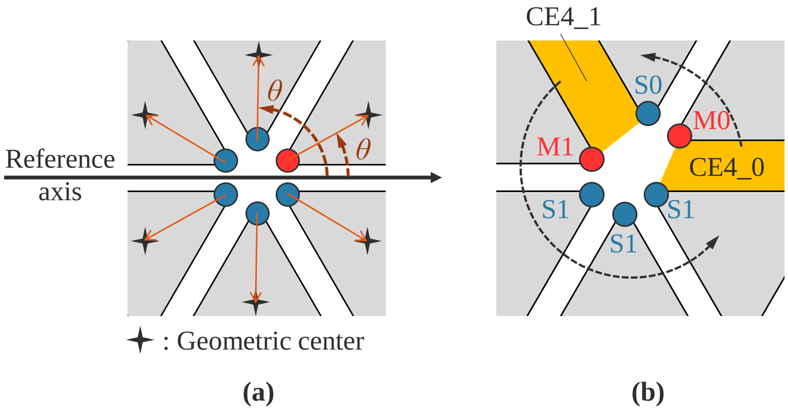

Fig. 4. M-S relation construction by counterclockwise search. **(a)** before failure **(b)** after failure.

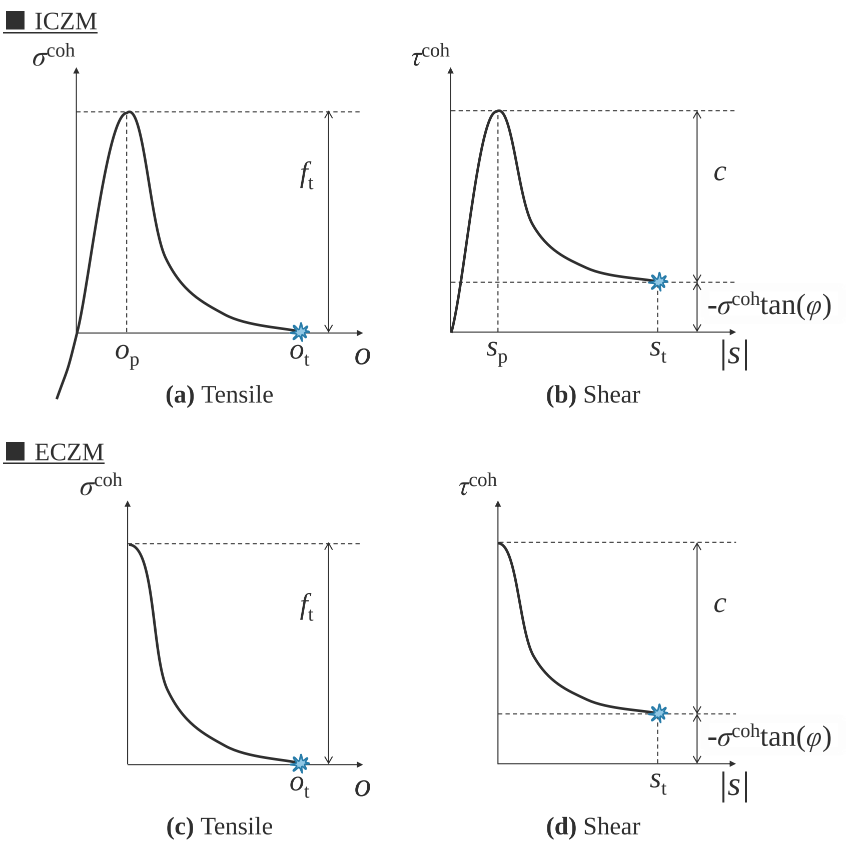

Fig. 5. Tensile/Shear traction-separation law. **(a)(b)** ICZM **(c)(d)** ECZM.

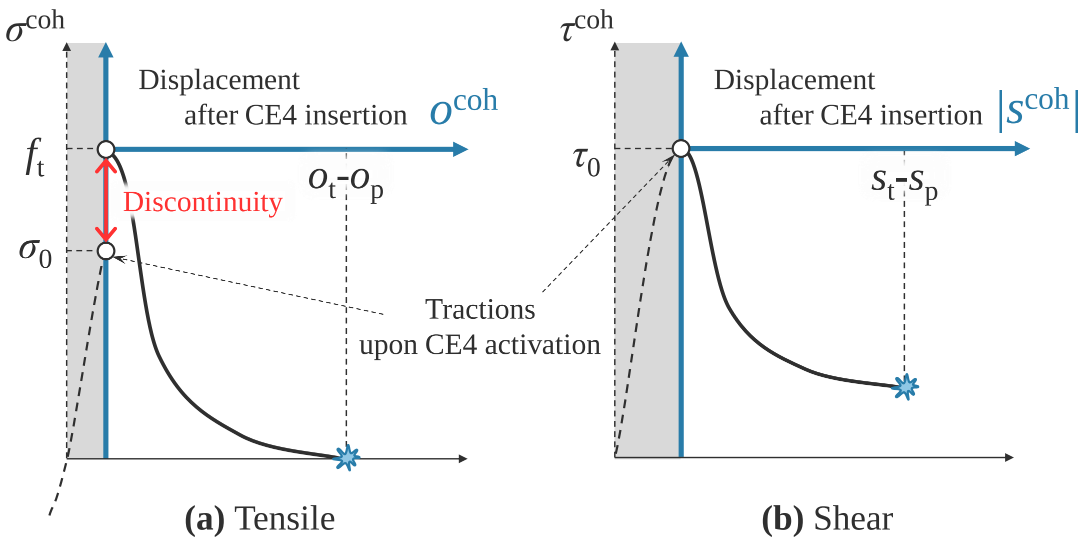

Fig. 6. Tensile/Shear traction-separation law of a conventional ECZM in case only the shear failure criterion is met (F_1<0 and F_2≥0).

The final important task for the successful implementation of the MS-ECZM-FDEM is how the ECZM is implemented. Before explaining that, one technical challenge must be pointed out. Figure 5 (a)(b) and (c)(d) show the traction-separation laws for CE4 in the case of ICZM and ECZM, respectively, where σ^{coh} [Pa] and τ^{coh} [Pa] are normal and shear cohesive tractions, respectively, according to the relative displacements of the two faces constituting a CE4 (o [m]: opening amount of CE4 in which opening is positive and $|s|$ [m]: sliding amount of CE4) (Fig. 1). On one hand, in the case of ICZM, pure mode I, pure mode II and mixed mode I-II fracturing can be modeled with ease (see Mahabadi et al., 2012; Fukuda et al., 2019) although the artificial compliance increase becomes the issue due to the existence of artificial elastic regime, i.e. $o<o_p$ and $|s|<s_p$ where o_p and s_p [m] are the "artificial" elastic limits of o and $|s|$, respectively (Munjiza

et al., 1999). On the other hand in the case of ECZM, Fig. 5 (c)(d) can be correct only when the dormant CE4 is activated with both $F_1 \geq 0$ and $F_2 \geq 0$ being simultaneously satisfied in Eq. (2). However, depending on the loading type in each target problem and due to the nature of unstructured mesh utilized in almost all the modern FDEM codes, the simultaneous satisfaction of the conditions $F_1 \geq 0$ and $F_2 \geq 0$ is very rare, and the cases "$F_1 \geq 0$ and $F_2 < 0$" and "$F_1 < 0$ and $F_2 \geq 0$" are rather encountered frequently. In fact, in most of the previous research using ECZM-based FEM (Cai et al., 2023; Dooley et al., 2009; Zhang et al., 2007), the cohesive tractions at the activation on a CE4 are set to the input strength values regardless of the failure modes. Figure 6 shows the softening curves of a conventional ECZM where only the shear failure criterion is met by $\tau_n = \tau_0$ ($F_1 < 0$ and $F_2 \geq 0$). In such a case, as shown in Fig. 6, although the normal traction $\sigma_n = \sigma_0$ upon the CE4 activation is less than the tensile strength, the normal cohesive traction starts from the tensile strength, and thus a time-discontinuity in the stress state before and after CE4 activation should occur. This time-discontinuity should also occur in the case where only the tensile failure criterion is met ($F_1 \geq 0$ and $F_2 < 0$). Particularly, the time-discontinuity may be significant in the case that the normal traction σ_0 is a compressive stress when CE4 is activated due to shear failure only. Dooley et al. (2009) assume that shear failure does not occur in such a compressive stress field, which, however, is not a reasonable assumption for the rock engineering applications targeted in this study. Note that Fig. 5 (a)(b) and (c)(d) in this paper are similar to Fig. 1(d) and (e), respectively in Cai et al. (2023). However, Cai et al. (2023) did not consider this time-discontinuity in their ECZM-FDEM at all. Based on this consideration, the following approach is taken as a remedy for alleviating this time-discontinuity issue although further study is needed for completely solving this issue. To this end, the traction-separation law (Fig. 5 (a)(b)) utilized in the ICZM-FDEM implementation is utilized. In the ICZM, to calculate σ^{coh} and τ^{coh}, the constitutive laws based on the tensile and shear softening laws, i.e. tensile/shear softening curves, are applied according to Eqs. (3) ~ (9) (e.g., Fukuda et al. (2019)). Note that in this traction-separation law, three integration points are adopted in each CE4 and the contribution of the integration points to the nodal forces is set according to Munjiza (1999) (See Eqs. (19) and (20) therein):

$$\sigma^{\mathrm{coh}} = \begin{cases} \dfrac{2o}{o_{\mathrm{p}}} f_{\mathrm{t}} & \text{if } \ o \le 0 \\ \left[\dfrac{2o}{o_{\mathrm{p}}} - \left(\dfrac{o}{o_{\mathrm{p}}} \right)^2 \right] f(D) f_{\mathrm{t}} & \text{if } \ 0 \le o \le o_{\mathrm{p}} \\ f(D) f_{\mathrm{t}} & \text{if } \ o_{\mathrm{p}} \le o \le o_{\mathrm{t}} \end{cases} \tag{3}$$

$$\tau^{\mathrm{coh}} = \begin{cases} \left[\dfrac{2|s|}{s_{\mathrm{p}}} - \left(\dfrac{|s|}{s_{\mathrm{p}}} \right)^2 \right] \times \left\langle f(D) c - \sigma^{\mathrm{coh}} \tan\phi \right\rangle & \text{if } \ 0 \le |s| \le s_{\mathrm{p}} \\ \left\langle f(D) c - \sigma^{\mathrm{coh}} \tan\phi \right\rangle & \text{if } \ s_{\mathrm{p}} \le |s| \le s_{\mathrm{t}} \end{cases} \tag{4}$$

$$f(D) = \left[1 - \frac{\alpha + \beta - 1}{\alpha + \beta} \exp\left(D \frac{\alpha + \gamma\beta}{(\alpha + \beta)(1 - \alpha - \beta)} \right) \right] \times \left[\alpha (1 - D) + \beta (1 - D)^{\gamma} \right] \tag{5}$$

$$o_p = \frac{2hf_{\mathrm{t}}}{P} \tag{6}$$

$$s_{\mathrm{p}} = \frac{2hc}{P} \tag{7}$$

$$G_{\mathrm{f\,I}} = \int_{o_\mathrm{p}}^{o_\mathrm{t}} \sigma^{\mathrm{coh}}(o)\mathrm{d}o \tag{8}$$

$$G_{\mathrm{f\,II}} + W_{\mathrm{res}} = \int_{s_\mathrm{p}}^{s_\mathrm{t}} \{\tau^{\mathrm{coh}}(|\,s\,|)\}\mathrm{d}|\mathrm{s}| \tag{9}$$

where o_t [m] and s_t [m] are the critical values of o and $|s|$, respectively; $<>$ is Macaulay brackets; h [m] is the representative length of a CE4; P [Pa] is the cohesive penalty; $G_{\mathrm{f\,I}}$ [J/m^2] and $G_{\mathrm{f\,II}}$ [J/m^2] in Eqs. (8) and (9) are the Mode I and Mode II fracture energies consumed during the generation of tensile and shear failures, respectively; W_{res} [J/m^2] is the amount of work per area of CE4 given by the residual stress term in the Mohr-Coulomb shear strength model illustrated in Eq. (9); $f(D)$ [-] ($0 \leq f \leq 1$) is the softening function that determines the softening curve and approximates the experimental stress-displacement curves obtained from the literature (Evans and Marathe, 1968); D ($0 \leq D \leq 1$) [-] is the damage variable indicating the damage degree of CE4; α, β, γ [-] are the parameters that determine the curve shape of softening curves. Note that the states with $o \leq o_\mathrm{p}$ or $0 \leq |s| \leq s_\mathrm{p}$ represent the artificial elastic state with $D = 0$, those with $o_\mathrm{p} \leq o \leq o_\mathrm{t}$, $s_\mathrm{p} \leq |s| \leq s_\mathrm{t}$ represent the damaged state (strain softening state with $0<D<1$) where a CE4 can be regarded as a micro fracture, and the state with $o_\mathrm{t} \leq o$ or $s_\mathrm{t} \leq |s|$ indicates the CE4 is broken with $D = 1$, deactivated, and turned into a macro fracture, and TRI3s surrounding the CE4 are treated as the surface of the discrete bodies. Besides, when the above-mentioned damage variable D is computed, the tension-induced damage D_o [-] and the shear-induced damage D_s [-], are computed as follows:

$$D = \min\left(1, \sqrt{D_\mathrm{o}^{\,2} + D_\mathrm{s}^{\,2}}\right) \tag{10}$$

$$D_{\mathrm{o}} = \min\left(1, \frac{o - o_{\mathrm{p}}}{o_{\mathrm{t}}}\right) \text{ if } o_{\mathrm{p}} \le o \le o_{\mathrm{t}} \text{ , otherwise } 0 \tag{11}$$

$$D_{\mathrm{s}} = \min\left(1, \frac{|s| - s_{\mathrm{p}}}{s_{\mathrm{t}}}\right) \text{ if } s_{\mathrm{p}} \le |s| \le s_{\mathrm{t}} \text{ , otherwise } 0 \tag{12}$$

No damage recovery is assumed to occur, and the damage variable D adopts the maximum value between the previous steps and the current step. In addition, the unloading process (decrease in o, $|s|$) and reloading process (increase in o, $|s|$) during material softening ($o_{\mathrm{p}} \le o \le o_{\mathrm{t}}$, $s_{\mathrm{p}} \le |s| \le s_{\mathrm{t}}$) are also modeled by the following equations (Camacho and Ortiz, 1996; Fukuda et al., 2019).

$$\sigma^{\mathrm{coh}} = \frac{o}{o_{\max}} f(D) f_{\mathrm{t}} \text{ if } 0 \le o \le o_{\max} \textit{ and } o_{\max} > o_{\mathrm{p}} \tag{13}$$

$$\tau^{\mathrm{coh}} = \frac{|s|}{s_{\max}} \left\langle f(D)c - \sigma^{\mathrm{coh}} \tan\phi \right\rangle \text{ if } 0 \le |s| \le s_{\max} \textit{ and } s_{\max} > s_{\mathrm{p}} \tag{14}$$

where $o_{\max}$ [m] and $s_{\max}$ [m] are the maximum values of o and $|s|$, respectively, which the CE4 experiences during the FDEM simulation.

Based on the above ICZM-based formulation, the countermeasure adopted by Maeda et al. (2022) is applied in this study. The boundary tractions (σ_{n}, τ_{n}) = (σ_0 (Eq. (15)), τ_0 (Eq. (16))) acting on the dormant CE4 upon the activation timing, when the failure criterion (Eq. (2)) is satisfied, are stored as shown in Eqs. (15) and (16), respectively:

$$\sigma_0 = \min(\sigma_{\mathrm{n}}, f_{\mathrm{t}}) \tag{15}$$

$$\tau_0 = \min(|\tau_{\mathrm{n}}|, \langle c - \sigma_{\mathrm{n}} \tan\phi \rangle) \tag{16}$$

Next, we define the following nominal opening/sliding displacements for the newly activated CE4 based on the boundary tractions (σ_0, τ_0) as follows:

$$o^{\mathrm{nominal}} = \frac{2h\sigma_0}{P} \tag{17}$$

$$s^{\text{nominal}} = \frac{2h\tau_0}{P} \quad (18)$$

Then, the effective opening/sliding (o, s) of the activated CE4 used in softening functions of cohesive tractions are defined as the sum of the nominal opening/sliding displacements (o^{nominal}, s^{nominal}) and actual geometrical opening/sliding displacements (o^{coh}, s^{coh}) which occur after the CE4 activation as shown in Eqs. (19) and (20):

$$o = o^{\text{nominal}} + o^{\text{coh}} \quad (19)$$

$$|s| = \left|s^{\text{nominal}} + s^{\text{coh}}\right| \quad (20)$$

It should be noted that the values of (o^{coh}, s^{coh}) at the activation timing of the dormant CE4 are zero because the CE4 has no gap. Then, these (o, $|s|$) are used in the above Eqs. (3) ~ (14). This approach is expected to alleviate the time-discontinuity issue to some extent and a similar concept has been used in the literature (e.g. Woo et al., 2019 and Fig.4 therein). However, with this approach, it must be noted that, upon the activation of CE4s, the effect of cohesive penalty P takes part in the FDEM simulation and thus the stabile time step becomes almost same as that in ICZM-FDEM since then while the time step can be taken relatively larger before the first CE4's activation. Furthermore, it is essential to emphasize that the complete elimination of the time-discontinuity in local nodal forces before and after the activation of CE4s remains a challenge. As discussed by Chen et al. (2019), the core issue lies in the reliance on failure judgments based on the stress states extrapolated from bulk elements (TRI3s in the present implementation), while the nodal forces contributed by the interpolated stresses and cohesive tractions remain independent. In their comprehensive review, Chen et al. (2019) examined the existing methods aimed at mitigating or resolving this time-discontinuity issue and concluded that these methods are too exceedingly complex to implement. Correspondingly, they proposed a promising node-based approach aiming to achieve continuous transitions, in which, both the failure judgment and cohesive force calculation are performed directly at each node. However, it is worth noting that the application of the node-based approach is limited to simple crack patterns, e.g., progressive debonding or delamination in composite structures and thus cannot be readily applied to address the complex fracture problems involving with shear failures under compressions within the framework of the Mohr-Coulomb shear failure model. Consequently, the resolution of this problem within our model is a subject for future research. Additionally, another time-discontinuity issue arises when a cohesive element reaches complete damage ($D = 1$) under a compressive stress field, i.e., transitioning from a state governed by the penalty of the cohesive elements to one governed by the contact penalty associated with contact interaction. This challenge is encountered in all CZM-based FDEM in spite of ICZM or ECZM. As a potential solution of to address this issue, Deng et al. (2021) proposed a smooth transition approach by assigning individual normal stiffness to each contact couple. However, our current paper only focuses on the implementation of MS-ECZM in the framework of GPGPU although we appreciate the time-discontinuity issue, which is regarded as a top priority of our future development. The computed cohesive tractions (σ^{coh}, τ^{coh}) are converted to the equivalent nodal force $\mathbf{f}_{\mathrm{coh}}$ [N], which is assembled to each detached node by utilizing the existing ICZM-FDEM code with very minor modification according to Eqs. (15) ~ (20), and further assembled to their M-nodes. Thus, upon fracturing, the resultant equation of motion for M-nodes is given as Eq. (21):

$$\mathbf{M}\frac{\partial^2 \mathbf{u}}{\partial t^2} = \mathbf{f}_{\mathrm{ext}} - \mathbf{f}_{\mathrm{int}} + \mathbf{f}_{\mathrm{con}} + \mathbf{f}_{\mathrm{coh}} \tag{21}$$

After the nodal information is updated for M-nodes, the nodal velocity and current nodal coordinates are copied to their S-nodes for the following timesteps to fully utilize the existing ICZM-FDEM code.

2.2. GPGPU parallel implementation

As mentioned in Section 2.1, the proposed MS-ECZM-FDEM can utilize the existing ICZM-FDEM code including the ICZM-based GPGPU-parallelized Y-HFDEM code with very minor modification. The only notable differences lie in the treatment of the judgement of the failure (Eq. (2)) and resultant update in the M-S relation (Fig. 4 (b)). Moreover, the processing of the update in the M-S relation is highly localized and suitable for parallel computation while it is challenging to parallelize the adaptive remeshing used in Fukuda et al. (2020b), Pandolfi and Ortiz (2002) and Yamamoto et al. (1999) especially in terms of GPGPU parallelization. This section implements the proposed algorithm through the GPGPU parallel computation using computing unified device architecture (CUDA) C/C++. The GPGPU parallel computing uses the following abstractions: threads, blocks, and grids (Fig. 7). In the GPGPU devices, parallel processing is performed in many threads using kernel functions. The threads are just execution units of kernel functions, and each thread performs operations similar to sequential computation. The blocks are the groups that manage several threads and allow memory sharing and synchronization among the threads within a block. Furthermore, the grids are the groups of blocks. This hierarchical management of threads enables parallel processing. Figure 8 shows the flow chart of the proposed MS-ECZM-FDEM with GPGPU parallel computing. In Fig. 8, the blue letters are the specific processes for the MS-ECZM-FDEM while the rest processes are the same as those used in the ICZM-based GPGPU-parallelized 2-D Y-HFDEM code. Therefore, the proposed MS-ECZM-FDEM can be realized by simply adding the processes shown in blue in Fig. 8 to the existing ICZM-FDEM codes. For the part which is common with the ICZM-FDEM, the interested readers are referred to the detailed explanations in Fukuda et al. (2019) and only the newly implemented portions are explained here. In Fig. 8, the process enclosed by the dashed line is the parallel computation in the GPGPU device. First, before entering the parallel computation by the GPGPU devices, the angle θ between the reference axis and the vector connecting each detached node and the geometric center of the TRI3, to which it belongs, are calculated for all the detached nodes, which in each group is sorted in the counterclockwise manner in order of decreasing θ (Fig. 4 (a)). After this sorting, the initial M-S relation *"GetMaster"* (Fig. 3) is constructed by simply selecting the detached node with the smallest θ as M-node while other detached nodes as S-nodes to this M-node in each node group. Note that these processes are handled in a host computer and processed by sequential computations only once at the onset of the simulation, which takes negligible computational time. The information of the relation between the sorted detached node IDs in each node group and *GetMaster* are transferred to the global GPGPU memory, and the subsequent computations are completely processed on GPGPU except for the output timing when the computed data from the GPGPU are transferred back to the host computer to generate the output files for visualization by the opensource software Paraview (Ayachit, 2015).

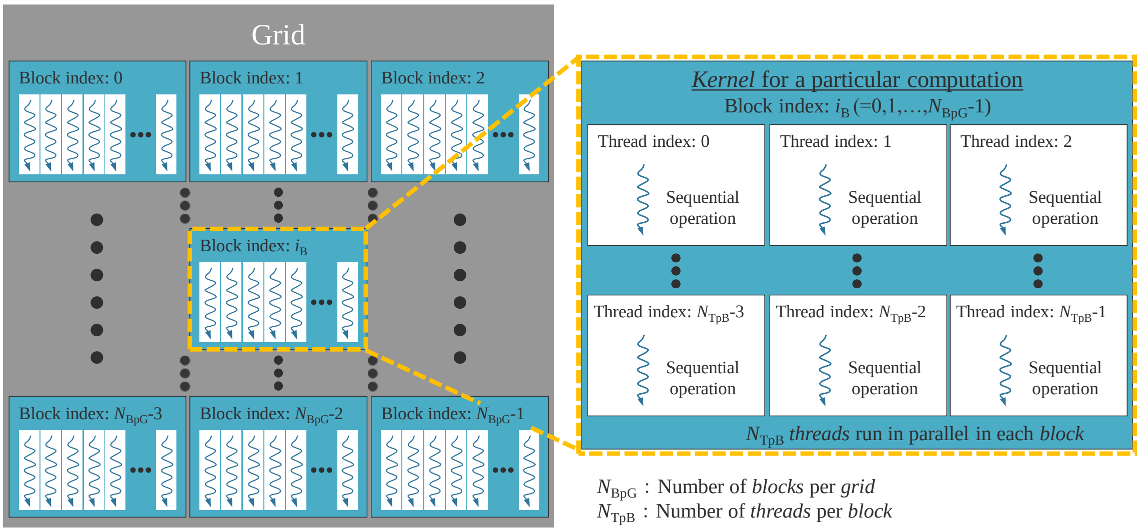

Fig. 7. The concept of GPGPU programming. (Fukuda et al., 2019).

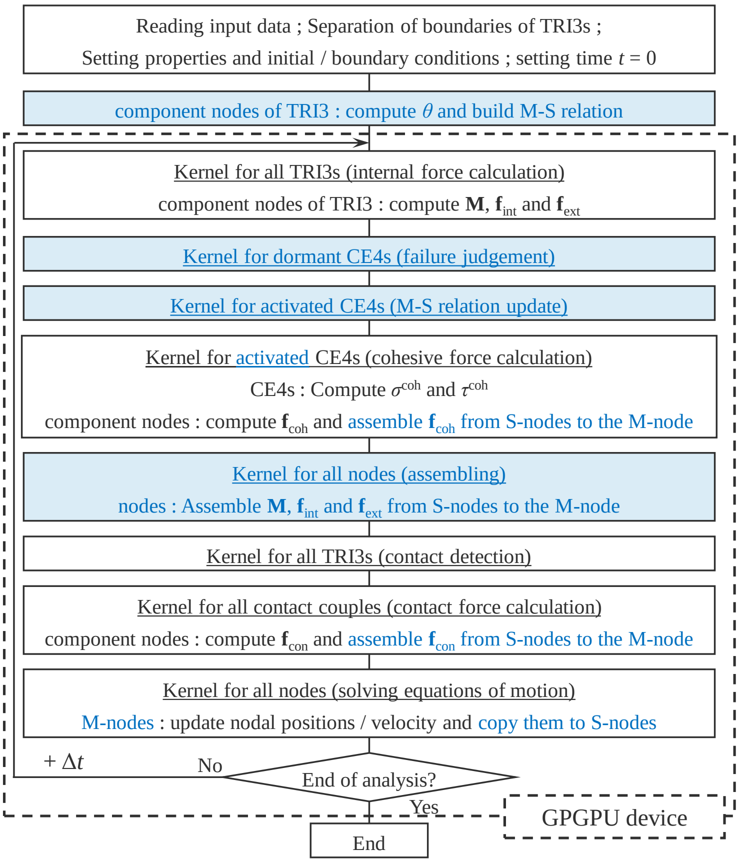

Fig. 8. Flow chart of proposed MS-ECZM-FDEM with GPGPU parallel computing.

As shown in Fig. 9, after calculating the Cauchy stress tensor σ_{ij} (for computing $\mathbf{f}_{\mathbf{int}}$) from the parallel processing of each TRI3s, the boundary stress tractions (σ_n,τ_n) acting on all the dormant CE4s are computed in parallel by assigning each CUDA thread to each dormant CE4 and failure judgment based on the boundary tractions (Eq. (2), Fig. 9 (a)). Note that, during the computations, an 1-dimentional (1-D) integer array "*i1CE4IDs*" and an 1-D Boolean array "*b1CE4states*" are prepared to store all the dormant and active CE4 IDs and state (active or dormant) of the CE4s, respectively. When any newly activated CE4s are identified, *i1CE4IDs* is sorted using *b1CE4states* as key in parallel following the parallel radix soring algorithm

(Satish et al., 2009) in the way that all the dormant CE4s' IDs are placed in the heading part of the *i1CE4IDs* array. In this way, the load balance between each CUDA thread in each CUDA block can be maximized for judging the failure of dormant CE4s and computation of cohesive tractions (σ^{coh}, τ^{coh}) for active CE4s.

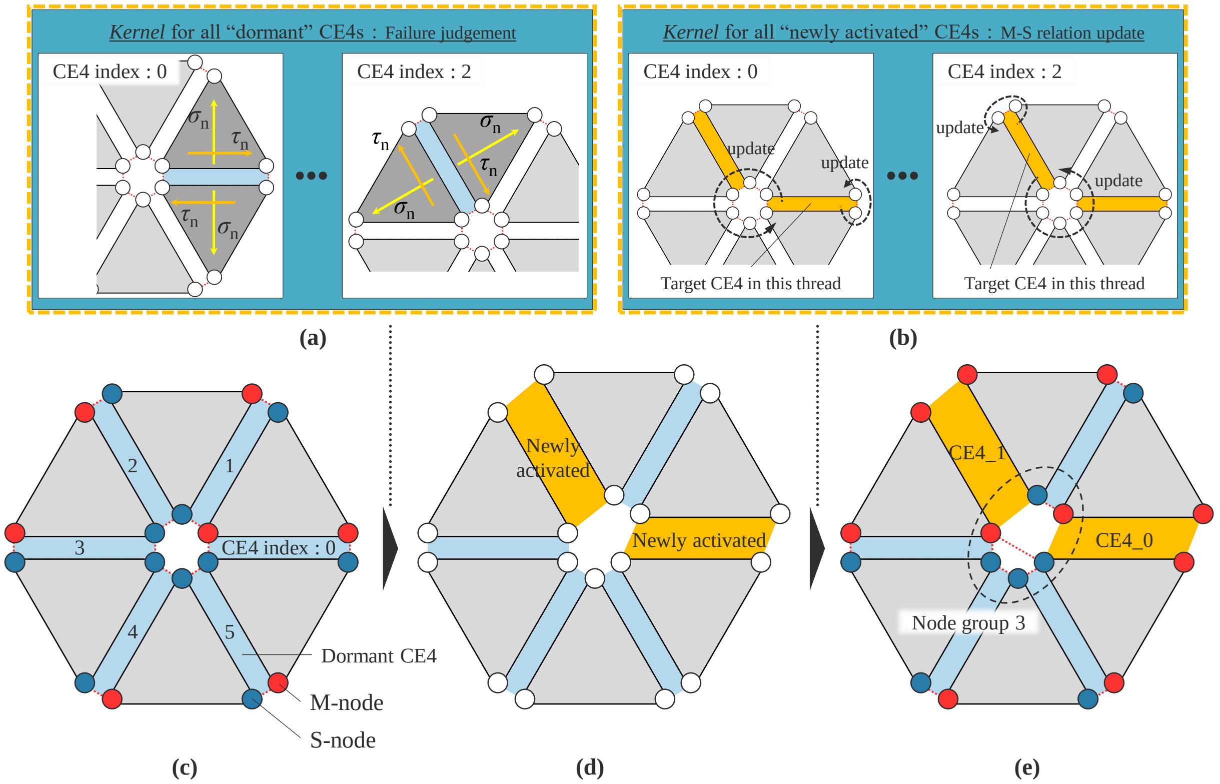

Fig. 9. The GPGPU parallel computation processes for CE4s in MS-ECZM-FDEM. **(a)** failure judgement, **(b)** M-S relation update, **(c)** before failure judgement, **(d)** after failure judgement and **(e)** after M-S relation update.

For the update in the M-S relations "*GetMaster*", each CUDA thread is assigned to each newly activated CE4s that has just satisfied the failure judgement in this step. Note that each newly activated CE4 consists of two node groups and thus each thread partially updates the M-

S relations in each node group. Figure 9 (b)(d)(e) shows the updating process of the M-S relations when two adjacent CE4s (CE4_0 and CE4_1 in Fig. 4 (b)) sharing the node group (*ig* = 3 in Fig. 3) are newly activated, which is the same as shown in Fig. 4 (b). For the node group No. 3, the threads for CE4_0 and CE4_1 partially update the M-S relation of the node group No. 3. For the thread processing CE4_0 when processing the node group No. 3, the first detached node, i.e., the right red node in the node group No. 3 is considered as an M-node and the counterclockwise search registers all the detached nodes until passing over other newly and already activated CE4s (CE4_1 in this case). Likewise, for the thread processing CE4_1 when processing the node group No. 3, the first detached node, i.e., the left red node in the node group No. 3 is considered as an M-node and the counterclockwise search registers all the detached nodes until passing over other newly and already activated CE4s (CE4_0 in this case). Therefore, the M-S update can be processed completely in parallel.

After computing the nodal masses and nodal forces for each detached node based on ICZM-FDEM manner, they are assembled from the S-nodes to their M-node through *GetMaster*, which is processed by assigning CUDA thread to each node group. Then, the nodal coordinates and nodal velocities of the M-nodes are updated based on the equations of motion (Eq. (1) for intact regime and Eq. (21) after fracture initiation) by assigning CUDA thread to each M-node. Finally, the updated information is copied from the M-nodes to their S-nodes through *GetMaster* by assigning CUDA thread to each node group.

3. Validation of GPGPU-parallelized MS-ECZM-FDEM

Numerical experiments of the fundamental laboratory rock mechanics tests, i.e., Brazilian tensile strength (BTS) and uniaxial compressive strength (UCS) tests under quasi-static loading, are conducted to validate the GPGPU-parallelized 2-D MS-ECZM-FDEM code developed in Section 2. In the remainder of this section, the MS-ECZM-FDEM is simply called as ECZM-FDEM. The results from the modellings of the BTS and UCS tests using GPGPU parallelized ECZM-FDEM and ICZM-FDEM codes are compared in terms of (I) accuracy of continuous deformation, (II) stable time step and (III) contact activation timing. It is important to note that this paper is not intended to delve into the evaluation of computational efficiency comparison between GPGPU parallelization and sequential computation since it has already been extensively discussed in our former publication (Fukuda et al., 2019). Instead, this paper focuses on the comparison between the GPGPU-based 2-D ECZM-FDEM and ICZM-FDEM codes, which should provide sufficient insights into their computational performance.

3.1. Overview of numerical models for BTS and UCS tests

Figure 10 shows the 2-D FDEM models for modelling the BTS and UCS tests. The diameter and height of the models are 30 [mm] × 30 [mm] for the BTS test and 30 [mm] × 60 [mm] for the UCS test, respectively. These FDEM models are discretized by TRI3s using unstructured mesh and average element size h for both models is 90 [μm]. The number of TRI3s included in the BTS and UCS tests are 187,061 and 475,810, respectively. Siliceous mudstone is considered as a target rock. The input parameters for the FDEM simulations for the BTS and UCS tests are set as shown in Table. 2 assuming this rock is isotropic and homogeneous. The density, Young's modulus, Poisson's ratio, tensile strength, cohesion, and internal friction angle are set based on Aoyagi et al. (2018) while other parameters are determined based on trial and error. Dynamic relaxation scheme (Munjiza, 2004) is used to approximately achieve the quasi-static condition and correspondingly damping coefficient $\eta = 2h\sqrt{\rho E}$ (see. Eqs. (2) and (3) in Fukuda et al. (2020b)) is assigned to each TRI3. The constant velocity of 0.05 [m/s] is assigned to each of the upper and lower platen and apparent strain rates are 3.3 [1/s] for BTS test and 1.7 [1/s] for UCS test. The time step Δt is set to be 1.0 [$\times 10^{-9}$ s/step]. It is known that a higher loading rate has a significant effect on the simulation results under quasi-static conditions such as the BTS and UCS tests (Mohammadnejad et al. (2020)). However, the loading rate set in this study has been confirmed to be appropriate because the stress-strain curve for the UCS test in Fig. 15, which is described below, does not show any fluctuation observed in Mohammadnejad et al. (2020). Furthermore, as a method of determining whether the loading rate setting can reasonably simulate quasi-static loading conditions, it is widely accepted that, if the ratio of total kinetic energy to total strain energy of a rock specimen is less than 0.05 at intact regime, the response obtained from the analysis does not include dynamic effects (Rojak et al., 2021; Siswanto et al., 2016). From the preliminary simulation, we confirmed that the above energy ratio through all the simulations in this analysis is much smaller than 0.05 and thus we considered that the loading rate is adequately small (Maeda et al., 2022).

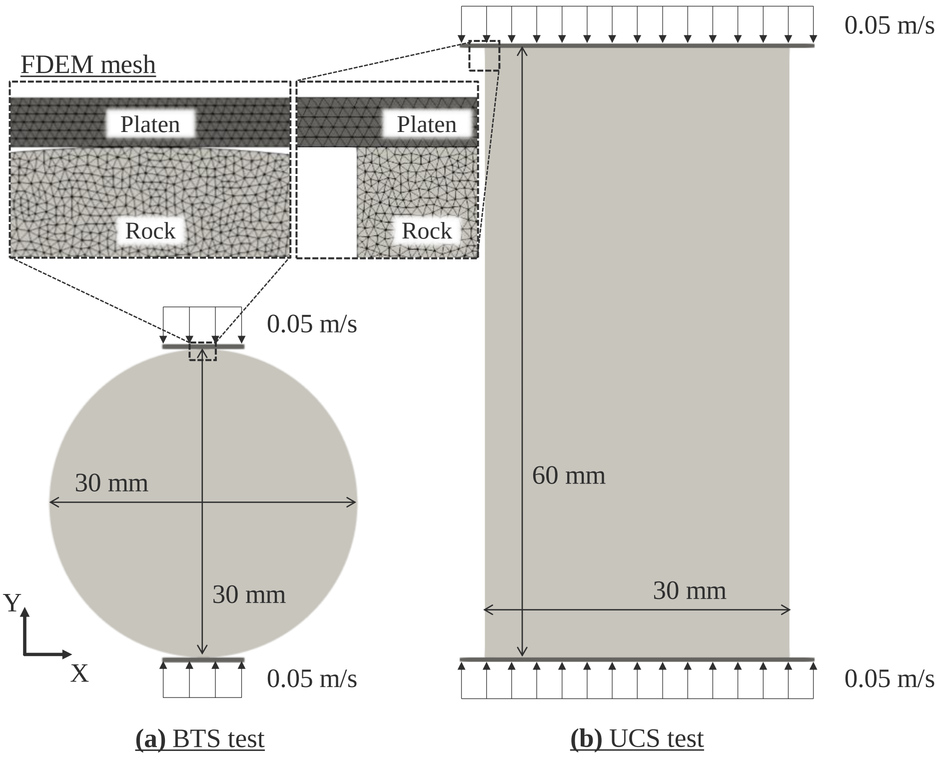

Fig. 10. Numerical models for FDEM simulations of **(a)** BTS and **(b)** UCS tests.

Table. 2. Input parameters for FDEM simulations of BTS and UCS tests.

Parameter		Value
Density	ρ [kg/m^3]	1840
Young's modulus	E [GPa]	1.82
Poisson's ratio	v [-]	0.17
Tensile strength	f_t [MPa]	1.83
Cohesion	c [MPa]	4.81
Internal friction angle	φ [°]	26
Mode I fracture energy	$G_{f\,I}$ [J/m^2]	16
Mode II fracture energy	$G_{f\,II}$ [J/m^2]	160
Cohesive penalty	P [GPa]	182
Normal contact penalty	P_{n_con} [GPa]	18.2
Friction coefficient of rock-platen surfaces	μ_{fric} [-]	0.1
Friction coefficient of rock fracture surfaces	μ_{fric} [-]	0.5

3.2. Verification and validation of GPGPU-parallelized ECZM-FDEM through comparative study

For comparison purpose, three different types of FDEM simulations, i.e., "ECZM-FDEM", "ICZM-FDEM", and "FDEM without CZM", are considered using the same numerical model set up in Section 3.1. The reason why "FDEM without CZM" is also considered is that each discrete body (rock and loading plates in the current case) behaves as purely continuum without any fracturing and thus can be considered as a benchmark for checking the precision of the FDEM computation for the continuous deformation of the rock at the intact regime, in which the rock behaves purely in FEM manner. Note that "FDEM without CZM" is achieved by deactivating the failure judgement (see Secion 2.1 and Eq. (2)) in the ECZM-FDEM to verify that the developed GPGPU-based ECZM-FDEM code works well without the activation of dormant CE4s by setting extremely large strength parameters.

In addition, different contact activation schemes, i.e., BCAA and (semi-)ACAA, are possible for ICZM-FDEM as mentioned in Section 1 while (semi-)ACAA is only possible for "ECZM-FDEM". To facilitate a fair comparison between ICZM-FDEM and ECZM-FDEM, a case of ICZM-FDEM with semi-ACAA is considered. It is worth noting that, in the literature related to FDEM developments and applications, ICZM-FDEM with BCAA, not semi-ACAA, is currently the most widely used approach. In this section, four cases in Table. 3 are considered for the following investigations. Figure 11 shows the difference between Case 2 (BCAA) and Cases 1, 3 and 4 for the case of BTS and UCS test models at the beginning of the FDEM simulations. The red area in Fig. 11 is the activated area for contact calculations (i.e. contact detection and contact force calculations), and the blue area is the part in which the contact calculations are initially deactivated. As shown in Fig. 11, the difference between the cases with BCAA (Case 2) and the semi-ACAA (Cases 1 and 3) is that the all the TRI3s are subjected to the contact calculations from the onset of the simulations in the BCAA while only the TRI3s on the rock-platen surfaces and those surrounding the CE6s just entering the shear softening regime are subjected to the contact calculations in the case of semi-ACAA. Please refer to Fukuda et al. (2021) for the detailed discussion of the BCAA and semi-ACAA. Since the ECZM does not involve with fracturing until first crack initiation occurs, it is evident that the BCAA concept is just unreasonable and not needed at all. Note that the because of no fracturing involved in Case 4, no adaptive contact activation is needed. The GPGPU-parallelization is applied in all cases.

Table. 3. Definition of four cases considered for the comparison.

	FDEM Type	**Contact activation scheme**	**TRI3s subjected to contact detection**
Case 1	ECZM-FDEM	Semi-ACAA	TRI3s on the rock-platen surfaces and newly created fractures
Case 2	ICZM-FDEM	BCAA	All the TRI3s in the system
Case 3	ICZM-FDEM	Semi-ACAA	TRI3s on the rock-platen surfaces and newly created fractures
Case 4	FDEM without CZM	Activated at the onset of the FDEM simulation	TRI3s on the rock-platen surfaces

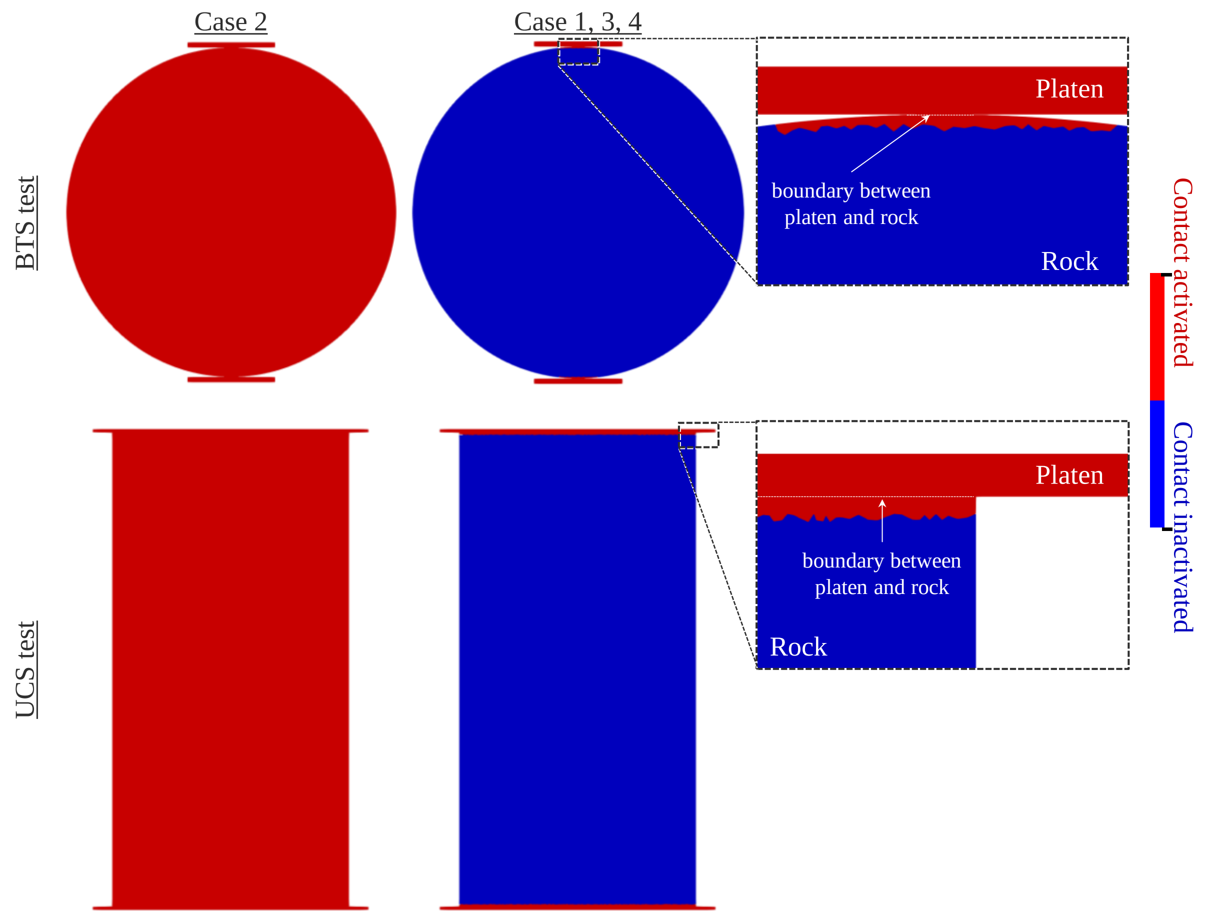

Fig. 11. Illustration explaining the different contact activation scheme in four cases in Table. 3.

Figure 12 shows the fracture process simulated from the BTS tests in the Cases 1, 2 and 3, which shows the spatial distribution of the damage state D in Eq. (10) at each loading stage with respect to nominal axial strain. The nominal axial strain ε_a is defined by the total axial displacement of both platens divided by the diameter of the rock disc. The contours in Fig. 12 represent the logarithm of the damage variable D, i.e. $\log_{10}D$, and $\log_{10}D \geq -3$ is visualized. A common trend observed in all cases is that microcracks of the order of $-3 \leqq \log_{10}D \leqq -2$ occur near the loading platens, and the macroscopic fractures ($\log_{10}D = 0$) that lead to the failure of the specimen progress vertically in the central part of the specimen. Although the final failure patterns are slightly different in each case, the characteristics of the resultant failure trends are consistent with those from the typical BTS tests of rocks. Thus, the GPGPU-implementation of adaptively activating dormant CE4s in the framework of MS-ECZM-FDEM is verified, and the result of Case 1 shows the almost similar fracture process as the conventional ICZM-FDEM (Cases 2 and 3). From the results of the Case 2 (ICZM-FDEM with BCAA), it is noticeable that more microcracks are observed near the loading platens compared to those of the Case 3 (ICZM-FDEM with semi-ACAA). The difference between the two cases may be due to the dual-force (i.e., the combination of cohesive traction and contact force) in the BCAA, which acts on the inside of the rock part even in the intact regime as discussed by Fukuda et al. (2021). This may enhance the micro-cracking. In contrast, Case 1 (ECZM-FDEM with semi-ACAA), which can avoid this dual-force issue, shows similar microcracks as Case 3.

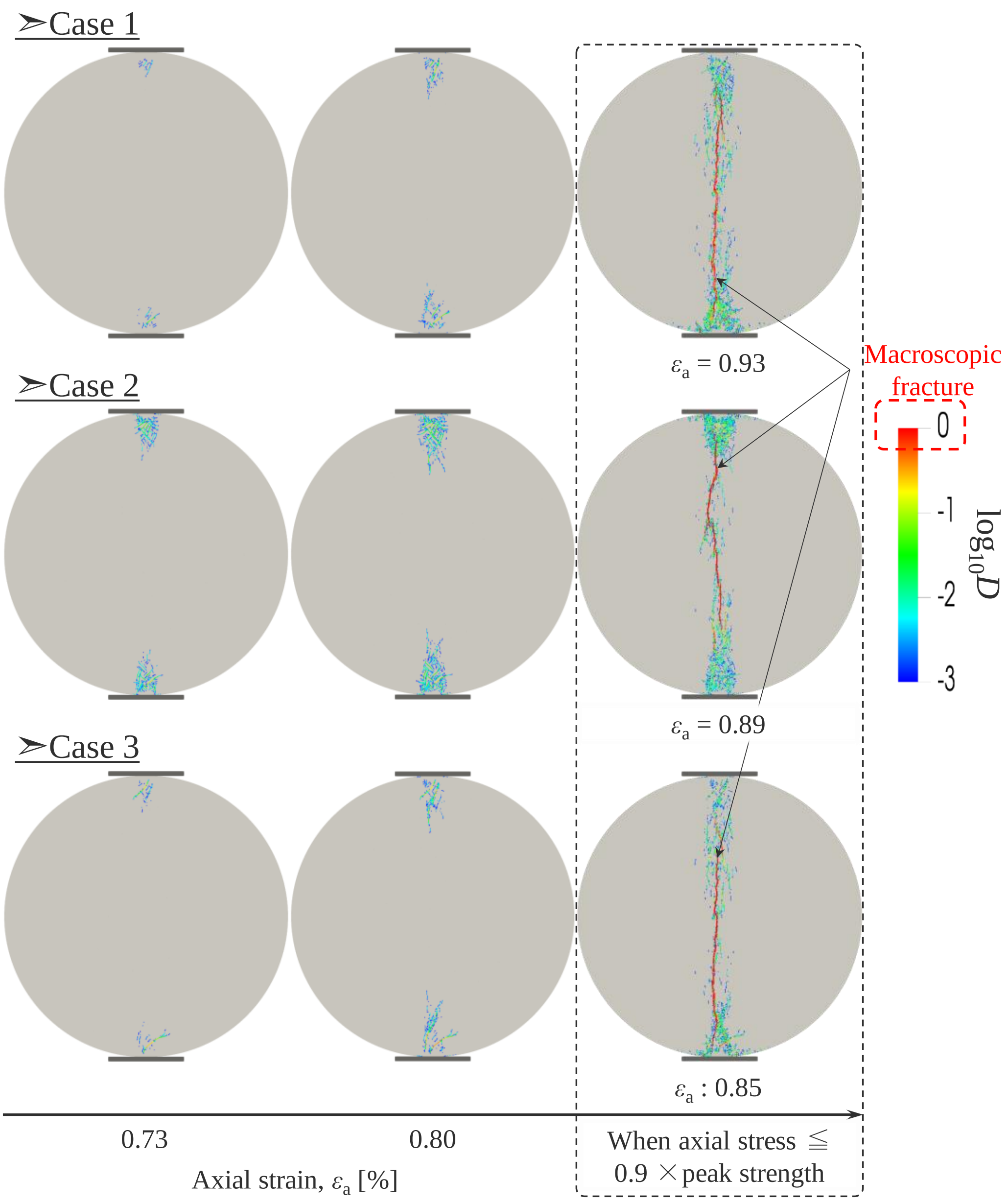

Fig. 12. Comparisons of fracture processes in BTS test for Cases 1, 2 and 3 in Table. 3.

Figure 13 plots the stress distribution along the loading diametrical line at a time when the nominal axial strain (contraction: positive) is small with the level of 0.2 % for Cases 1-4. Figure

13 (a) shows the observation line for stress monitoring. Figure 13 (b) plots the σ_{xx} and σ_{yy} along the observation line while Fig. 13 (c) shows an enlarged view of Fig. 13 (b). Since this stage corresponds to intact rock well before the failure stage, the stress distribution in Cases 1-3 should agree that in Case 4 (FDEM without CZM). In fact, the results of Case 1 (ECZM-FDEM) in Fig. 13 (b) clearly show that both σ_{xx} and σ_{yy} are in perfect agreement with the benchmark Case 4 (FDEM without CZM). On the other hand, the results of ICZM-FDEM, i.e., Cases 2 and 3, show some deviation from FDEM without CZM. This deviation is clearly due to the deterioration of the accuracy in computing continuous deformation and stress calculation by inserting active CE4s at the onset of the simulation in the ICZM. To quantitively evaluate these deviations, the average values of σ_{xx} for the TRI3s existing within a range of ±5 mm from the center of the specimen, and the average values of σ_{yy} for the TRI3s existing within a range of ±1 mm from the center of the specimen are calculated for Cases 1~4, and the deviation of the average stress [%] of Cases 1~3 against the benchmark Case 4 is investigated. For Cases 2 and 3, i.e., ICZM-FDEM, this deviation for σ_{xx} is found to be 1.6 % for BCAA and 2.0 % for semi-ACAA, respectively, while 1.4 % for BCAA and 1.8 % for semi-ACAA for σ_{yy}. In contrast, Case 1 (ECZM-FDEM) shows no deviation for both σ_{xx} and σ_{yy}. Thus, it can be concluded that the ICZM-FDEM involves with approximately 1.4~2 % deterioration of stress calculation accuracy, and the advantage of ECZM-FDEM is evident.

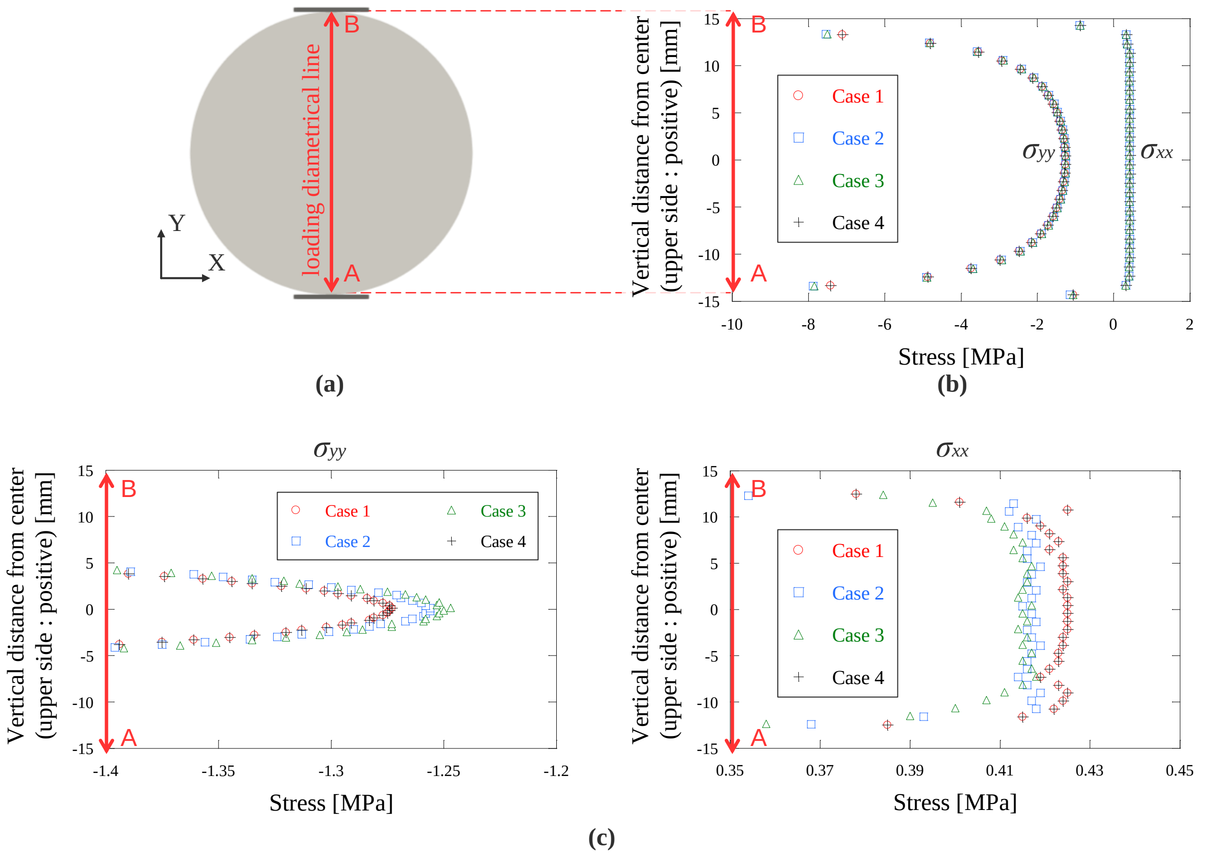

Fig. 13. Comparisons of stress distributions along the loading diametrical line of the specimen in FDEM simulations of BTS test between Cases 1, 2, 3 and 4 defined in Table. 3. **(a)** Schematic of loading diametrical line, **(b)** Stress distribution along the loading diametrical line, **(c)** Enlarged view of σ_{yy} (left) and σ_{xx} (right) in **(b)**.

Figures 14 and 15 show the fracture process and axial stress-strain curve for Cases 1, 2 and 3 in UCS test for selected axial strain levels. In Fig. 15 (contraction: positive), dotted line indicates the stress-strain line for benchmark Case 4 to verify the initial continuous behavior of Cases 1~3 at intact regime. Note that the apparent axial strain ε_a in these figures is obtained by dividing the total displacement at the upper and lower platens by the height of the specimen.

The contour in Fig. 14 represents the spatial distribution of $\log_{10}D$ for $\log_{10}D \geq -3$. For the intact deformation regime before the commencement of non-linear behavior near the peak in the stress-strain curve (Fig. 15), Cases 1-3 show more or less good agreement with FDEM without CZM. However, if these three cases are compared in terms of tangent modulus at the 50 % peak strength, the corresponding deviations of Cases 1 (ECZM), 2 (ICZM with BCAA) and 3 (ICZM with semi-ACAA) against Case 4 are 0 %, 1.2 % and 1.7 %, respectively, which again shows the advantage of ECZM-FDEM in improving the precision of continuous deformation at the intact regime. In addition, to check the precision of continuous deformation at the intact regime from the viewpoint of the stress distribution, Fig. 16 shows the stress distribution for σ_{yy} along the loading line at a time when the apparent axial strain ε_a is small with the level of 0.67 % for Cases 1-4. Figure 16 (a) shows the observation line for stress monitoring. Figure 16 (b) plots the σ_{yy} along the observation line. The figure shows that, as in the stress-strain curve, Case 1 (ECZM-FDEM) is in perfect agreement with the benchmark Case 4 (FDEM without CZM), while Cases 2 and 3, i.e., ICZM-FDEM, deviate from the benchmark Case 4. Furthermore, an obvious difference in curve shape is observed between ECZM-FDEM and ICZM-FDEM: while the curve shape of ECZM-FDEM is smooth, that of ICZM-FDEM is highly fluctuated, indicating a clearly discontinuous stress distribution, even at intact regime. This can be due to a geometrical inconsistency in the mesh, in other words, a physical gap caused by the relative displacement of CE4s between TRI3s even for the continuous deformation regime in the ICZM.

The geometric inconsistency causes local numerical instability. These results for stress-strain curve and stress distribution confirm the high accuracy of the calculations in ECZM-FDEM at intact conditions. These results for stress-strain curve and stress distribution confirm the high precision of the calculations in ECZM-FDEM at intact regime. To the best of knowledge of the authors, there are not any valid theoretical solutions or benchmark analyses available for delineating any arbitrary fracture initiation, propagation, and interaction in heterogeneous rocks. Consequently, it is very challenging, if not impossible, to compare the accuracy between the simulations from ECZM-FDEM and ICZM-FDEM after the onset of fracturing. However, it is anticipated that the less noisy stress distribution from ECZM-FDEM should result in more rational crack initiation and propagation.

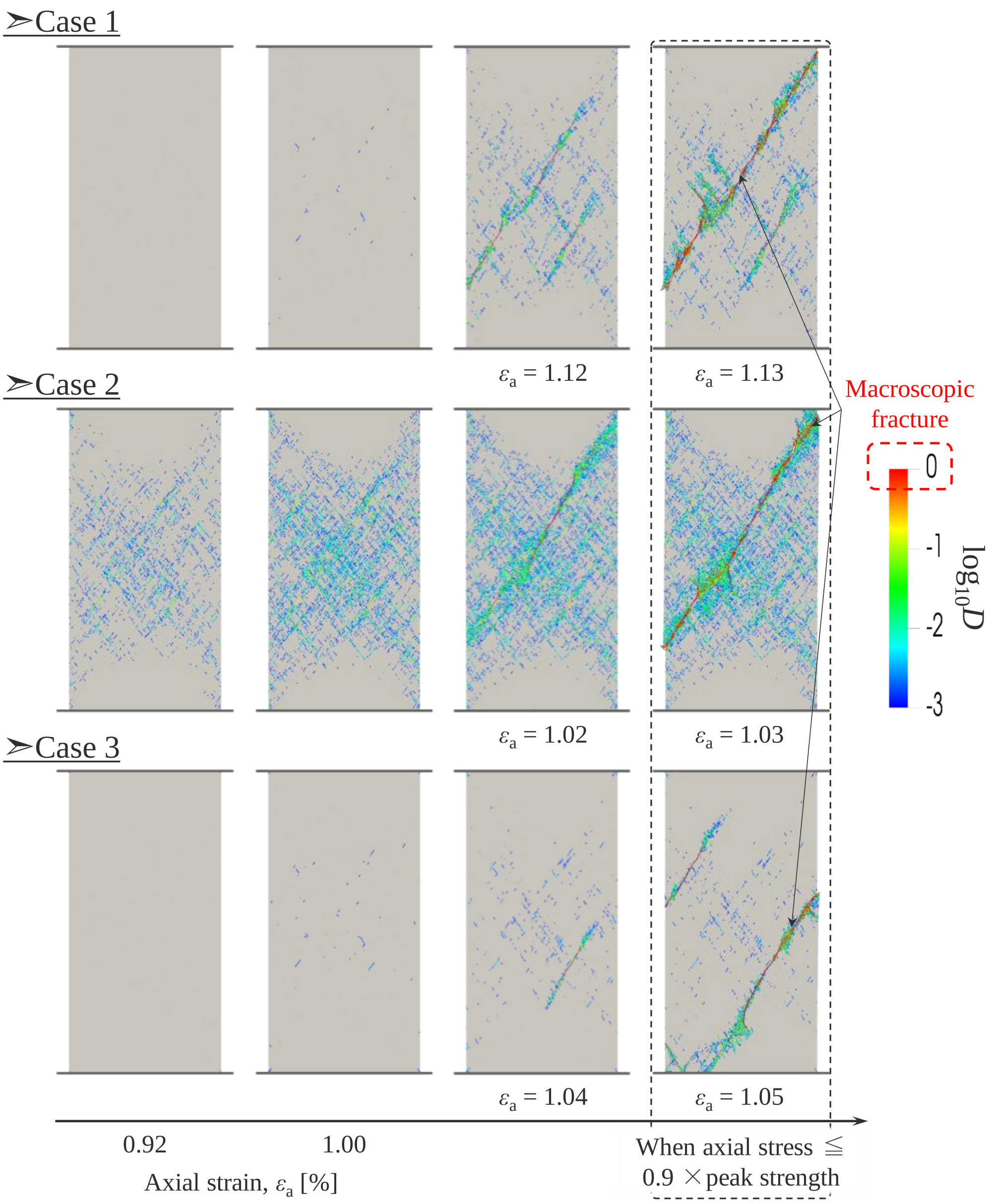

Fig. 14. Comparisons of fracture processes in UCS test for Cases 1, 2 and 3 in Table. 3.

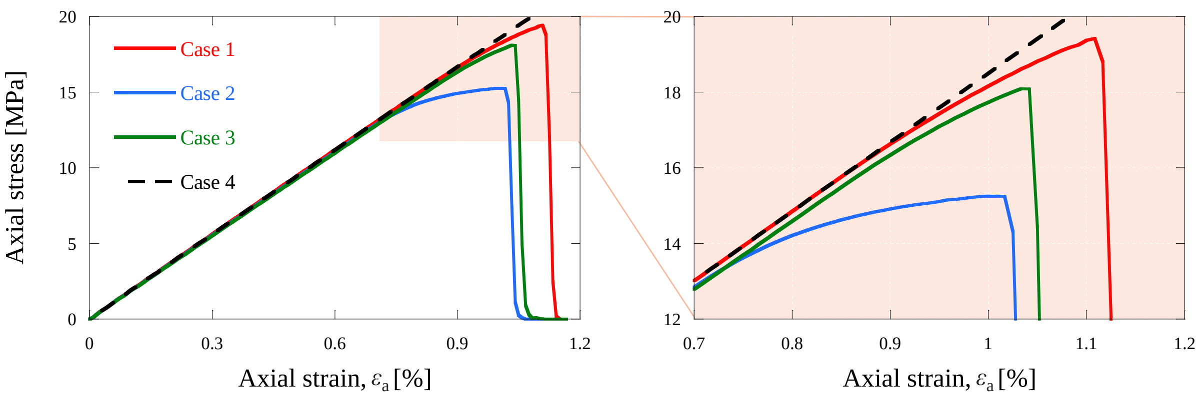

Fig. 15. Comparisons of axial stress-axial strain curve in UCS test for Cases 1, 2, 3 and 4 in

Table. 3.

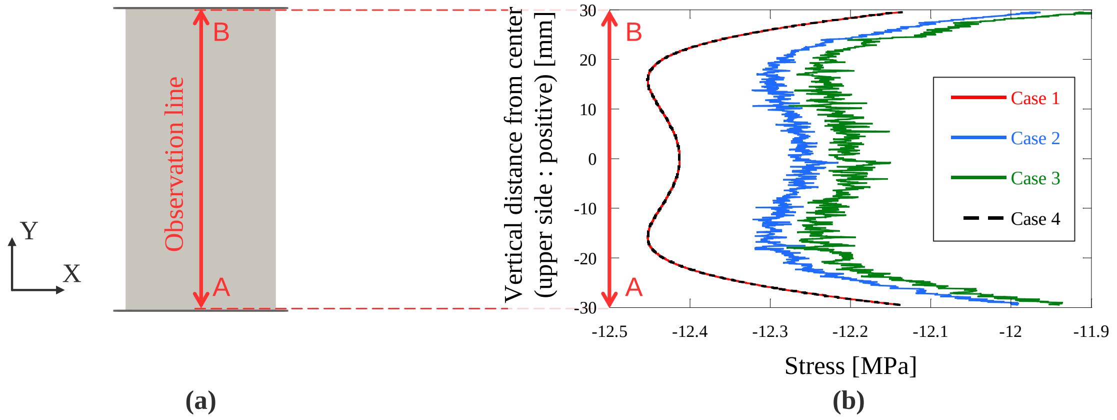

Fig. 16. Comparisons of stress distributions for σ_{yy} along the observation line of the specimen in FDEM simulations of UCS test between Cases 1, 2, 3 and 4 defined in Table. 3. (a) Schematic of observation line, **(b)** Stress distribution for σ_{yy} along the observation line.

In the pre-peak stage of Fig. 15, it can be observed that all Cases 1-3 show a nonlinear behavior accompanied by a decrease in slope near the peak, which is due to the transition from continuum to discontinuum by the commencement of softening of CE4s, which is evident from Fig. 14 when ε_a = 0.92 [%]. Similar to the trend observed in the BTS test, the amount of generated microcracks at the same level of ε_a varies between each case, and only ICZM (BCAA) shows more significant number of microcracks (see the results of ε_a = 0.92 [%] ~1.00[%]). Again, this could also be attributed to the effect of dual-force, which may enhance the local noise/fluctuation in stress, and much rapid decrease in the stiffness of bulk rock. For the failure process in Fig. 14, at the third axial strain level from the left, Cases 2 and 3, i.e., ICZM-FDEM show localized fracture growth from a single location, while Case 1 (ECZM-FDEM) shows fracture growth from multiple locations. Again, since CE4s can open/slide even at intact regime in the ICZM-FDEM, if a CE4 is softened, the CE4's deformation in ICZM can strongly enhances the deformation direction of surrounding CE4s than the case of ECZM. Consequently, fracturing around the softened CE4 may easily occur, propagate, and localize, along the deformation direction of the softened CE4s in the ICZM. On the other hand, in ECZM-FDEM, dormant CE4s cannot open/slide at the intact regime and the failure criteria for the commencement of softening (Eq. (2)) are based on the stress fields. Thus, if a CE4 is softened, stress concentration occurs in the surrounding TRI3s, leading to the satisfaction of the failure criteria in surrounding TRI3s' boundary (i.e., dormant CE4). In this process, the normal/shear stresses on surrounding dormant CE4s due to stress concentration does not necessarily correspond to the deformation direction of the softened CE4. In other words, the deformation direction of the softened CE4 does not directly affect the surrounding CE4s. Therefore, the deformability or localization along the direction of the softening CE4 is less likely to occur compared to ICZM-FDEM. This explanation is also supported by the fact that, at the peak stage of Fig. 15, the peak strength and the strain at that time in Case 1 (ECZM-FDEM) are higher than in Cases 2 and 3. ICZM-FDEM can be considered as a model in which the failure progresses more easily than ECZM-FDEM, due to the aforementioned numerical instability caused by geometric inconsistency and localization of fracturing. In contrast, the ECZM-FDEM has more stable nature, the rock model based on ECZM-FDEM behaves much stronger than ICZM-FDEM. For the resultant failure patterns, all Cases 1~3 in Fig. 14 show the formation of a shear band, and its inclination is of the same degree, which also verifies the developed ECZM-FDEM code and validates its applicability. However, it is evident that the calibrated input parameters in the ICZM-FDEM cannot be directly used in the ECZM-FDEM.

Furthermore, the computational speeds among Cases 1, 2 and 3 are compared, although it is difficult to simply compare because of the different timing of the fracturing in each case. To this end, by keeping the time step Δt = 1.0 [$\times 10^{-9}$ s], the concept of t_{250}, which indicates the runtime required for completing every 250 [μs] (=250,000 timesteps) of the above FDEM simulation of the UCS model, is introduced and it is monitored for these three cases. Since the t_{250} for Case 2 (ICZM-FDEM (BCAA)) is the longest, it is used as the reference. Then, transient relative speed-up is defined as (t_{250} for ICZM-FDEM (BCAA)/(t_{250} for ECZM-FDEM (semi-ACAA)) and (t_{250} for ICZM-FDEM (BCAA)/(t_{250} for ICZM-FDEM (semi-ACAA)) for Cases 1 and 3 against Case 2, respectively. Figure 17 shows the comparison of transient relative speed-up with respect to the simulation progress in analysis time, in which 0 % and 100 % is adjusted to t = 0 μs and t = 7000 μs, respectively. To check how the damage state D in the rock part in the above UCS model affects the computational performance, this figure also shows the evolution of the maximum damage D, i.e. ($\max(\log_{10}D)$), among all the CE4s for each case. Firstly, the ICZM-FDEM (semi-ACAA) shows approximately six times relative-speed up against ICZM-FDEM (BCAA) during the early stages of the FDEM simulation of UCS model before the onset of damaging. Then, this relative speed-up ratio reduces close to 1 when $\max(\log_{10}D) = 0$ is satisfied which is due to significant number of TRI3s newly added to the contact force calculation. This behavior is very similar to what is reported to dynamic UCS modelling in Fukuda et al. (2021). On the other hand, the ECZM-FDEM (semi-ACAA) shows approximately four times relative speed-up against ICZM-FDEM (BCAA) before the onset of damaging. Therefore, ECZM-FDEM (semi-ACAA) is less computationally efficient than the ICZM-FDEM (semi-ACAA). This is attributed to the unique processing in the MS-ECZM-FDEM such as the failure judgement, update of M-S relation and assembling the information of S-nodes to their M-nodes. The ECZM-FDEM (semi-ACAA) code has still the potential for improvements in terms of the computational efficiency. For instance, there is room to enhance the performance of the current code in the context of failure judgment. Currently, this judgment is conducted against all TRI3 boundaries, irrespective of the stress levels. It is possible to further enhance the computational performance by implementing an algorithm of processing the failure judgment only for the boundaries whose stress levels are closer to satisfy the failure criteria. This is regarded as another task for future study. With the increase in the number of activated CE4s due to damaging, the computational efficiency of ECZM-FDEM (semi-ACAA) becomes closet to that of ICZM-FDEM (semi-ACAA) since less failure judgement is needed and more contact force calculation becomes the most computationally demanding part of the FDEM calculations. Then, the performance of the ECZM-FDEM (semi-ACAA) becomes closer to that of ICZM-FDEM (BCAA) after the point of maximum ($\log_{10}D$) = 0. Therefore, it can be concluded that the ECZM-FDEM (semi-ACAA) is slightly slower than that ICZM-FDEM (semi-ACAA) as long as the same time step Δt is used. The total runtime is identified as 17,209 [s], 13,141 [s] and 67,976 [s] for ECZM-FDEM (semi-ACAA), ICZM-FDEM (semi-ACAA) and ICZM-FDEM (BCAA), respectively. However, since the ECZM-FDEM (semi-ACAA) does not involve in any CE4s before the onset of the fracturing, the stable time step for the ECZM-FDEM (semi-ACAA) can be taken larger than that for the ICZM-FDEM (semi-ACAA). Thus, this factor is investigated in Section 3.3 in more detail.

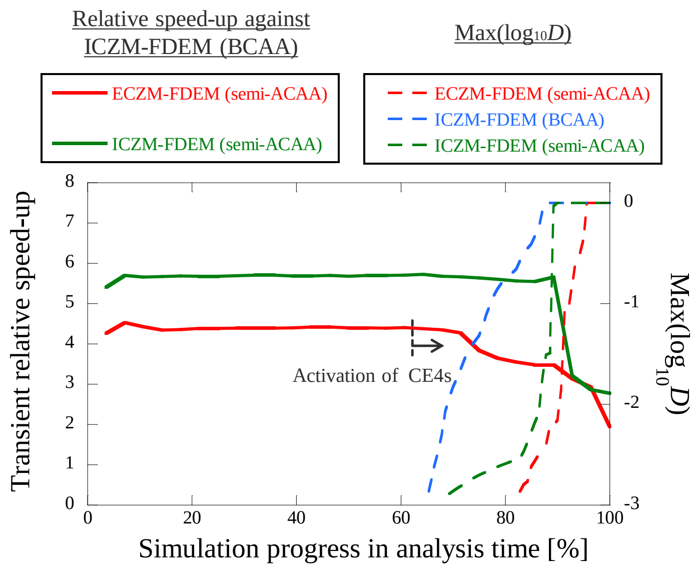

Fig. 17. Transient relative speed-up of ECZM-FDEM (semi-ACAA) and ICZM-FDEM (semi-ACAA) against ICZM-FDEM (BCAA), and change of max($\log_{10}D$) using UCS test model.

3.3. Effect of stable time step on computing performance

According to our estimation of the maximum stable time step based on the material properties and mesh geometries (Mohammadnejad et al., 2020) and preliminary analyses using various values of time step Δt, it was identified that the stable time step Δt_{crit} for the ICZM-FDEM (semi-ACAA) and ICZM-FDEM (BCAA) can be taken approximately 2.0 [$\times 10^{-9}$ s/step] for the entire simulation. On the other hand, Δt_{crit} for the ECZM-FDEM (semi-ACAA) can be taken approximately 6.0 [$\times 10^{-9}$ s/step] as long as no CE4s are activated by the crack initiation, i.e., in the intact regime. Note that in the preliminary analyses of this study, it was confirmed that, if the values of Δt were set larger than Δt_{crit} for each case, the FDEM calculations showed the spurious mode soon after the onset of the simulation, resulting in unrealistic stress level (> GPa) at various locations. However, to deal with time-discontinuity due to the activation of dormant CE4s in the ECZM-FDEM (semi-ACAA), cohesive penalty's effect takes part in the FDEM simulation, which results in the stable time step becomes more or less same as that in the ICZM-FDEM with semi-ACAA and BCAA since the crack initiation occurs. By taking this advantage of larger stable time step of ECZM-FDEM (semi-ACAA) in the intact regime, the comparison of transient relative speed-up is made similar to Fig. 17. To that end, ECZM-FDEM (semi-ACAA) simulation is conducted using Δt =6.0 [$\times 10^{-9}$ s] and Δt =2.0 [$\times 10^{-9}$ s] before and after the first activation of CE4, respectively, while the constant Δt =2.0 [$\times 10^{-9}$ s] is used for ICZM-FDEMs with semi-ACAA and BCAA. Here, the runtime t_{180} required to calculate for every 180 [μs] is considered in the above-mentioned UCS model and is monitored. In the case of ECZM-FDEM (semi-ACAA) before the first activation of CE4, t_{180} is equivalent to 6.0 [$\times10^{-9}$ s] $\times$ 30,000 timesteps. On the other hand, t_{180} is equivalent to 2.0 [$\times10^{-9}$ s] $\times$ 90,000 timesteps for ICZM-FDEMs with semi-ACAA and BCAA as well as for ECZM-FDEM (semi-ACAA)) after the first activation of CE4. Note that the first activation of CE4 is observed between 720,000 steps and 750,000 steps in the case of ECZM-FDEM (semi-ACAA). Since the t_{180} for Case 2 (ICZM-FDEM (BCAA)) is the longest, it is again used as the reference. Then, transient relative speed-up is defined as (t_{180} for ICZM-FDEM (BCAA)/(t_{180} for ECZM-FDEM (semi-ACAA)) and (t_{180} for ICZM-FDEM (BCAA)/(t_{180} for ICZM-FDEM (semi-ACAA)) for Cases 1 and 3 against Case 2, respectively, which is shown in Fig. 18. The horizontal axis in Fig. 18 is the simulation progress in analysis time, in which 0 % and 100 % is adjusted to t = 0 μs and t = 7000 μs, respectively. Figure 18 clearly shows that the relative speedup of ECZM-FDEM (semi-ACAA) against ICZM-FDEM (BCAA) is about 13 times before the first activation of CE4. This relative speed-up of ECZM-FDEM (semi-ACAA) against ICZM-FDEM (BCAA) is about 3 times faster than the case in which the constant Δt is used in ECZM-FDEM (semi-ACAA) as shown in Fig. 17 since the value of Δt used in ECZM-FDEM (semi-ACAA) is taken to be 3 times larger than that used in ICZM-FDEM. After the first activation of CE4 and the reduction in Δt, the relative speed-up of the ECZM-FDEM (semi-ACAA) decreased to about 4 times against ICZM-FDEM (BCAA), which is lower than that of ICZM-FDEM (semi-ACAA) against

ICZM-FDEM(BCAA) and similar to the trend in Fig. 17. Then, the total simulation time is identified as 4,527 [s], 5,246 [s] and 29,953 [s] for ECZM-FDEM (semi-ACAA), ICZM-FDEM (semi-ACAA) and ICZM-FDEM (BCAA), respectively. Thus, the total runtime of ECZM-FDEM (semi-ACAA) with variable Δt becomes the smallest among the three cases. Thus, if the simulation time for the intact regime occupies relatively larger part of the entire simulation, which tends to be true for many quasi-static loading scenarios especially for hard rocks such as granite, ECZM-FDEM (semi-ACAA) can achieve better performance than ICZM-FDEM thanks to the larger stable time step. Therefore, not only the improvement of the calculation precision in the intact deformation regime but also the improvement of the computational efficiency is achieved by introducing the proposed GPGPU-based ECZM-FDEM while the obtained fracture pattern can be still reasonable.

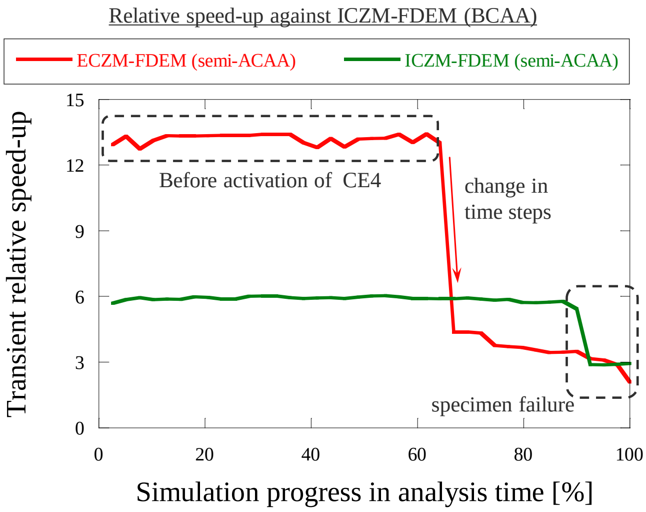

Fig. 18. Transient relative speed-up of ECZM-FDEM (semi-ACAA) with variable time step and ICZM-FDEM (semi-ACAA) against ICZM-FDEM (BCAA), and change of max($\log_{10}D$) using UCS test model.

4. Conclusions

In this study, GPGPU-parallelized 2-D ECZM-FDEM was proposed by applying the M-S algorithm as an alternative method for the complex adaptive remeshing traditionally adopted in ECZM-FDEM. The proposed MS-ECZM-FDEM algorithm was explained in detail and implemented in the GPGPU-parallelized Y-HFDEM code. Then, the developed code was applied to numerically model the BTS and UCS tests of siliceous mudstone under quasi-static loading for verification and validation. Furthermore, the results from the numerical modellings of the BTS and UCS tests using GPGPU-parallelized ECZM-FDEM and ICZM-FDEM were compared against each other in terms of several aspects, which are summarized below:

- During the continuous deformation stage, the accuracy of ICZM-FDEM is compromised against FDEM without cohesive elements by 1.6 % ~ 2.0 % and 1.4 % ~ 1.8 % for the distribution of normal stresses (σ_{xx}, σ_{yy}), respectively, inside the specimen in the BTS modelling, and by 1.2 % ~ 1.7 % for the tangent modulus in the stress-strain curve in the UCS modelling. However, ECZM-FDEM showed no degradation in accuracy compared with FDEM without cohesive elements. Moreover, for the spatial stress distribution before fracturing in the UCS modelling, ECZM-FDEM showed a smooth distribution while ICZM-FDEM showed a noisy and disturbed distribution.
- In terms of the stable time step, ECZM-FDEM can set the time step in the intact regime

about three times larger compared to ICZM-FDEM with the cohesive penalty being 100 times of Young's modulus of the rock. Furthermore, by taking advantage of larger stable time step in ECZM-FDEM, the total runtime of ECZM-FDEM became smaller than that of ICZM-FDEM in the case of modelling the UCS test. This study targeted at the soft rock and the relative speed-up in the total runtime was about 6 times faster. It is expected, if hard rock is modelled, a much more significant times of relative speed-up can be achieved the runtime spent in modelling the intact regime of the hard rock is much longer.

- As for the contact activation, ECZM-FDEM removes the need for the BCAA, which has been prevalent in the ICZM-FDEM community but extremely computationally intense. Instead, ECZM-FDEM implements the more computationally efficient semi-ACAA and ACAA for the contact activation. It is confirmed that a relative speed-up of about 13 times of ECZM-FDEM based on the semi-ACAA can be achieved against ICZM-FDEM based on BCAA in the intact regime together with the improvement about the aforementioned stable time step. Howevver, after the clack initiation, the stable time step of ECZM-FDEM becomes more or less same as that of ICZM-FDEM due to the effect of larger cohesive penalty to deal with time-discontinuity issue inherent in ECZM.

Thus, the proposed GPGPU-parallelized ECZM-FDEM with the M-S algorithm is confirmed to provide effective and valuable improvements over the conventional GPGPU-parallelized ICZM-FDEM for the numerical modelling of rock fracturing process. However, only 2-D model is considered in this study and further study is need to extended the proposed GPGPU-parallelized ECZM-FDEM with the M-S algorithm to 3-D model in order to reasonably model rock fracture problems. Although the proposed counter-clockwise searching around each node group cannot be directly used in the 3-D case, the M-S algorithm adopted this study is expected to pave the way to realize the 3-D implementation of GPGPU parallelized ECZM-FDEM.

Acknowledgement

This work was partially funded by JST FOREST Program (Grant Number JPJFR216Y, Japan) and JSPS KAKENHI (Grant Numbers 22H01589, 20K14826), and JFE 21st Century Foundation, Hitachi Global Foundation, JGC-S SCHOLARSHIP FOUNDATION to S.O., and JSPS KAKENHI (Grant Number 22K04996) to D.F. Their support is gratefully acknowledged.

References

An, H.M., Liu, H.Y., Han, H., Zheng, X., Wang, X.G., 2017. Hybrid finite-discrete element modelling of dynamic fracture and resultant fragment casting and muck-piling by rock blast. Computers and Geotechnics 81, 322-345.

Aoyagi, K., Ishii, E., 2018. A method for estimating the highest potential hydraulic conductivity in the excavation damaged zone in mudstone. Rock Mech Rock Eng. 52, 385-401.

Ayachit, U., 2015. The Paraview Guide: A Parallel Visualization Application.

Barenblatt, G.I., 1962. The mathematical theory of equilibrium cracks in brittle fracture. Adv. Appl. Mech. 7, 55-129.

Cai, W., Gao, K., Ai, S., Wang, M., Feng, Y.T., 2023. Implementation of extrinsic cohesive zone model (ECZM) in 2D finite-discrete element method (FDEM) using node binding scheme. Comput. Geotech. 159, 105470.

Camacho, G.T., Ortiz, M., 1996. Computational modelling of impact damage in brittle materials. Int. J. Solids. Struct. 33, 2899-2938.

Chen, S., Mitsume, N., Gao, W., Yamada, T., Zang, M., Yoshimura, S., 2019. A nodal-based extrinsic cohesive/contact model for interfacial debonding analyses in composite structures. Compt. Struct. 215, 80-97.

Deng, P., Liu, Q., Huang, X., Liu, Q., Ma, Hao., Li, Weiwei., 2021. Acquisition of normal contact stiffness and its influence on rock crack propagation for the combined finite-discrete element method (FDEM). Eng. Fract. Mech. 242, 107459.

Dooley, I., Mangala, S., Kale, L., Geubelle, P., 2009. Parallel simulations of dynamic fracture using extrinsic cohesive elements. J. Sci. Comput. 39, 144-165.

Dugdale, D.S., 1960. Yielding of steel sheets containing slits. J. Mech. Phys. Solids. 8, 100-104.

Evans, R., Marathe, M., 1968. Microcracking and stress-strain curves for concrete in tension. Mater. Struct. 1, 61-64.

Elmo, D., Stead, D., 2010. An integrated numerical modelling–discrete fracture network approach applied to the characterisation of rock mass strength of naturally fractured pillars. Rock Mech Rock Eng. 43, 3-19.

Espinha, R., Park, K., Paulino, G.H., Celes, W., 2013. Scalable parallel dynamic fracture simulation using an extrinsic cohesive zone model, Comput. Methods Appl. Mech. Eng. 266, 144-161.

Fukuda, D., Mohammadnejad, M., Liu, H., Dehkhoda, S., Chan, A., Cho, S.H., Min, G.J., Han, H., Kodama, J., Fujii, Y., 2019. Development of a GPGPU-parallelized hybrid finite-discrete element method for modeling rock fracture. Int. J. Numer. Anal. Methods Geomech. 43, 1797-1824.

Fukuda, D., Mohammadnejad, M., Liu, H., Zhang, Q., Zhao, J., Dehkhoda, S., Chan, A., Kodama, J., Fujii, Y., 2020a. Development of a 3D hybrid finite-discrete element simulator based on GPGPU-parallelized computation for modelling rock fracturing under quasi-static and dynamic loading conditions. Rock Mech Rock Eng. 53, 1079-1112.

Fukuda, D., Nihei, E., Cho, S.H., Oh, S., Nara, Y., Kodama, J., Fujii, Y., 2020b. Development of a numerical simulator for 3-D dynamic fracture process analysis of rocks based on hybrid FEM-DEM using extrinsic cohesive zone model. Materials transactions. 61, 1767-1774.

Fukuda, D., Liu, H., Zhang, Q., Zhao, J., Kodama, J., Fujii, Y., Chan, A.H.C., 2021. Modelling of dynamic rock fracture process using the finite-discrete element method with a novel and efficient contact activation scheme. Int. J. Rock Mech. Min. Sci. 138, 104645.

Geomechanica Inc., 2023. Irazu Software. https://www.geomechanica.com/software/ (accessed 28 Feb 2023).

Guo, L., 2014. Development of a three-dimensional fracture model for the combined finite-discrete element method. Imperial College London, London.

Guo, L., Xiang, J., Latham, J.P., Izzuddin, B., 2016. A numerical investigation of mesh sensitivity for a new three-dimensional fracture model within the combined finite-discrete element method. Eng Fract Mech. 151, 70-91.

Hamdi, P., Stead, D., Elmo, D., 2014. Damage characterization during laboratory strength testing: a 3D-fnite-discrete element approach. Comput. Geotech. 60, 33-46.

Klerck, P.A., 2000. The finite element modelling of discrete fracture in quasi-brittle materials, Ph.D thesis. University of Wales Swansea.

Klerck, P.A., Sellers, E.J., Owen, D.R.J., 2004. Discrete fracture in quasi-brittle materials under compressive and tensile stress states. Comput. Methods Appl. Mech. Engrg. 193, 3035-3056.

Knight, E.E., Rougier, E., Lei, Z., Euser, B., Chau, V., Boyce, S.H., Gao, K., Okubo, K., Froment, M., 2020. HOSS: an implementation of the combined finite-discrete element method. Comput. Part. Mech. 7, 765-787.

Latham, J., Guo, L., Wang, X., Xiang, J., 2012. Modelling the evolution of fractures using a combined FEM-DEM numerical method. 12th ISRM Congress.

Lei, Z., Rougier, E., Knight, E.E., Munjiza, A., 2014. A framework for grand scale parallelization of the combined finite discrete element method in 2d. Comput. Part. Mech. 1, 307-31.

Lisjak, A., Figi, D., Graselli, G., 2014. Fracture development around deep underground excavations: Insights from FDEM modelling. J. Rock Mech. Geotech. Eng. 6, 493-505.

Lisjak, A., Mahabadi, O.K., He, L., Tatone, B.S.A., Kaifosh, P., Haque, S.A., Grasselli, G. 2018. Acceleration of a 2D/3D finite-discrete element code for geomechanical simulations using General Purpose GPU computing. Comput. Geotech. 100, 84-96.

Liu, H.Y., Kang, Y.M., Lin, P., 2015. Hybrid finite-discrete element modelling of geomaterials fracture and fragment muck-piling. Int. J. Geotech. Eng. 9, 115-131.

Liu, H., Liu, Q., Ma, H., Fish, J., 2021. A novel GPGPU-parallelized contact detection algorithm for combined finite-discrete element method. Int. J. Rock Mech. Min. Sci. 144, 104782.

Liu, H., Ma, H., Liu, Q., Tang, X., Fish, J., 2022. An efficient and robust GPGPU-parallelized contact algorithm for the combined finite-discrete element method. Comput. Methods Appl. Mech. Eng. 395, 114981.

Liu, Q., Wang, W., Ma, Hao., 2019. Parallelized combined finite-discrete element (FDEM) procedure using multi-GPU with CUDA. Int. J. Numer. Anal. Methods Geomech. 44, 208-238.

Lukas, T., Schiava D'Albano, G.G., Munjiza, A., 2014. Space decomposition based parallelization solutions for the combined finite-discrete element method in 2D. J. Rock Mech. Geotech. Eng. 6, 607-615.

Mohammadnejad, M., Fukuda, D., Liu, H., Dehkhoda, S., Chan, A., 2020. GPGPU-parallelized 3D combined finite-discrete element modelling of rock fracture with adaptive contact activation approach. Comput. Part. Mech. 7, 849-867.

Munjiza, A., Andrews, K.R.F., White, J.K., 1999. Combined single and smeared crack model in combined fnite-discrete element analysis. Int. J. Numer. Meth. Eng. 44, 41-57.

Munjiza, A., 2004. The Combined Finite-Discrete Element Method. Wiley.

Mahabadi, O.K., Lisjak, A., Munjiza, A., Graselli, G., 2012. Y-Geo: New combined finite-discrete element numerical code for geomechanical applications. Int. J. Geomech. 12, 676-688.

Maeda, Y., Ogata, S., Fukuda, D., Inui, T., Yasuhara, H., Kishida, K., 2022. Fundamental study on fracture process analysis of rocks under quasi-static loading based on hybrid FEM-DEM using extrinsic cohesive zone model. Journal of the Society of Material Science, Japan. 71, 206-213.

Nguyen, V.P., 2014. Discontinuous Galerkin/extrinsic cohesive zone modeling: Implementation caveats and applications in computational fracture mechanics. Eng. Fract. Mech. 128, 37-

68.

Pandolfi, A., Ortiz, M., 2002. An efficient adaptive procedure for three-dimensional fragmentation simulations. Eng. Compt. 18, 148-159.

Papoulia, K.D., Sam, C.H., Varasis, S.A. 2003. Time continuity in cohesive finite element modeling. Int. J. Numer. Meth. Eng. 58, 679-701.

Rockfield, 2023. Elfen Advanced Software. https://www.rockfieldglobal.com/software/elfen-advanced/ (accessed 27 Feb 2023).

Rojak, J., Nosewicz, S., Thoeni, K., 2021. 3D formation of the deformable discrete element method. Int. J. Num. Methods Eng. 122, 3335-3367.

Rougier, E., Knight, E.E., Broome, S.T., Sussman, A.J., Munjiza, A., 2014. Validation of a three-dimensional finite-discrete element method using experimental results of the split Hopkinson pressure bar test. Int. J. Rock Mech. Min. Sci. 70, 101-108.

Satish, N., Harris, M., Garland, M., 2009. Designing efficient sorting algorithms for manycore GPUs. in Proc. *23rd IEEE International Parallel and Distributed Processing Symposium*, May 2009.

Siswanto, W. A., Nagentrau, M., Tobi, A.L.M., Tamin, M.N. 2016. Prediction of plastic deformation under contact condition by quasi-static and dynamic simulations using explicit finite element analysis. J. Mech. Sci. Technol. 30, 5093-5101.

Tatone, B.S.A., Grasseilli, G., 2015. A calibration procedure for two-dimensional laboratory-scale hybrid finite–discrete element simulations. Int. J. Rock Mech. Min. Sci. 75, 56-72.

Vlachopoulos, N., Vazaios, I., 2018. The Numerical Simulation of Hard Rocks for Tunnelling Purposes at Great Depths: A Comparison between the Hybrid FDEM Method and Continuous Techniques. Adv. Civ. Eng. 2018, 3868716.

Woo, K., Peterson, W.M., Cairns, D.S., 2014. Selective activation of intrinsic cohesive elements, J. Appl. Math. Phys. 2, 1061-1068.

Woo, K., Cairns, D.S., 2019. Selective activation of intrinsic cohesive elements for fracture analysis of laminated composites. Compos. Struct. 210, 310-320.

Xiang, J., Latham, J.P., Farsi, A., 2016. Algorithms and capabilities of solidity to simulate interactions and packing of complex shapes. Proceedings of the 7th international conference on discrete element methods. 188, 139-149.

Yamamoto, M., Ichijo, T., Morooka, K., Kaneko, K., 1999. Experimental and theoretical study on smooth blasting with electronic delay detonators. International Journal for Blasting and Fragmentation. 3, 3-24.

Yan, C., Jiao, Y.Y., Zheng, H., 2021. A three-dimensional heat transfer and thermal cracking model considering the effect of cracks on heat transfer. Int. J. Numer. Anal. Methods Geomech. 43, 1825-1853.

Yan, C., Wei, D., Wang, G., 2022a. Three-dimensional finite discrete element-based contact heat transfer model considering thermal cracking in continuous–discontinuous media. Comput.

Methods Appl. Mech. Eng. 388, 114228.

Yan, C., Xie, X., Ren, Y., Ke, W., Wang, G., 2022b. A FDEM-based 2D coupled thermal-hydro-mechanical model for multiphysical simulation of rock fracturing. Int. J. Rock Mech. Min. Sci. 149, 104964.

Zhang, J.Z., Paulino, G.H., Celes, W., 2007. Extrinsic cohesive zone modelling of dynamic fracture and microbranching instability in brittle materials. Int. J. Numer. Methods Eng. 72, 893-923.

Nomenclature

Abbreviations

1-D	1-dimentional
2-D	2-dimentional
3-D	3-dimensional
GPGPU	general-purpose graphic-processing-units
MPI	message-passing interface
PC	personal computer
CUDA	computing unified device architecture
FEM	finite element method
DEM	discrete element method

FDEM	combined finite-discrete element method
CZM	cohesive zone model
ICZM	intrinsic cohesive zone model
ECZM	extrinsic cohesive zone model
DGCZM	Galerkin-based cohesive zone model
UCZM	universal cohesive zone model
ICZM-FDEM	intrinsic cohesive zone model - based combined finite-discrete element method
ECZM-FDEM	extrinsic cohesive zone model - based combined finite-discrete element method
BCAA	brute-force contact activation approach
ACAA	adaptive contact activation approach
M-node	master node
S-node	slave node
M-S relation	relation between M-node and S-node
M-S algorithm	updating algorithm of the M-S relation
MPCs	multi-point constraints
TRI3s	3-node triangle elements
CE4s	4-node cohesive elements

BTS Brazilian tensile strength

UCS uniaxial compressive strength