Received: 13 October 2023 | Revised: 12 March 2024 | Accepted: 18 April 2024

DOI: 10.1002/nme.7504

RESEARCH ARTICLE

WILEY

Matrix-based implementation and GPU acceleration of hybrid FEM and peridynamic model for hydro-mechanical coupled problems

Tao Ni[1,2] | **Jin Zhang**[3] | **Mirco Zaccariotto**[2,4] | **Ugo Galvanetto**[2,4] | **Bernhard A. Schrefler**[5]

[1]State Key Laboratory of Geohazard Prevention and Geoenvironment Protection, Chengdu University of Technology, Chengdu, China

[2]Industrial Engineering Department, University of Padova, Padova, Italy

[3]College of Civil and Transportation Engineering, Hohai University, Nanjing, China

[4]Center of Studies and Activities for Space (CISAS)-G. Colombo, University of Padova, Padova, Italy

[5]Department of Civil, Environmental and Architectural Engineering, University of Padova, Padova, Italy

Correspondence
Bernhard A. Schrefler, Department of Civil, Environmental and Architectural Engineering, University of Padova, via Marzolo 9, Padova 35131, Italy.
Email: bernhard.schrefler@dicea.unipd.it

Funding information
National Natural Science Foundation of China, Grant/Award Number: 42207226; Natural Science Foundation of Sichuan Province, Grant/Award Numbers: 2023NSFSC0808, 2022NSFSC1129; State Key Laboratory of Geohazard Prevention and Geoenvironment Protection Independent Research Project, Grant/Award Numbers: SKLGP2022Z018, SKLGP2021Z026; University of Padova, Grant/Award Numbers: NR.237212/23, NR.232492/23; MIUR, Italy, Grant/Award Number: PRIN-PNRRP2022HLHHB

Abstract

The hybrid finite element-peridynamic (FEM-PD) models have been evidenced for their exceptional ability to address hydro-mechanical coupled problems involving cracks. Nevertheless, the non-local characteristics of the PD equations and the required inversion operations when solving fluid equations result in prohibitively high computational costs. In this paper, a fast explicit solution scheme for FEM-PD models based on matrix operation is introduced, where the graphics processing units (GPUs) are used to accelerate the computation. An in-house software is developed in MATLAB in both CPU and GPU versions. We first solve a problem related to pore pressure distribution in a single crack, demonstrating the accuracy of the proposed method by a comparison of FEM-PD solutions with those of PD-only models and analytical solutions. Subsequently, several examples are solved, including a one-dimensional dynamic consolidation problem and the fluid-driven hydraulic fracture propagation problems in both 2D and 3D cases, to comprehensively validate the effectiveness of the proposed methods in simulating deformation and fracture in saturated porous media under the influence of hydro-mechanical coupling. In the presented numerical results, some typical strong dynamic phenomena, such as stepwise crack advancement, crack branching, and pressure oscillations, are observed. In addition, we compare the wall times of all the cases executed on both the GPU and CPU, highlighting the substantial acceleration performance of the GPU, particularly when tackling problems with a significant computational workload.

KEYWORDS

explicit time integration, finite element method, GPU acceleration, hydro-mechanical coupling, matrix operation, peridynamics

1 | INTRODUCTION

The interplay of multiple physics resulting from the interaction between porous media and fluids is of considerable interest to various engineering fields, including reservoir engineering, biomechanics, chemical engineering, and environmental engineering.[1–4] Comprehending and reliably forecasting the interrelated hydro-mechanical responses of fluid-saturated porous media hold crucial significance for a wide array of engineering application.[3] However, the study of hydro-mechanical (HM) coupling remains a challenging research endeavour due to its intricate nature. Analytical solutions for hydro-mechanical coupled problems are often limited,[5] while experimental investigations grapple with the challenge of accurately capturing the intricacies of the HM coupled system,[6] further compounded by their resource-intensive and time-consuming nature. Consequently, numerical simulations have taken precedence as the primary approach for exploring and gaining insight into HM coupled phenomena.[3,7,8] This necessitates the development of advanced numerical models and simulation techniques capable of accurately representing the interactions between multi-physics and delivering reliable predictions.[9]

Over the past few decades, a variety of complex numerical models have been developed. These models can be broadly categorized into three main groups: continuous, discrete, and hybrid approaches.[3,7,9] The hybrid finite element-peridynamic (FEM-PD) model firstly introduced in Reference 3 belongs to the last category of methods. The FEM-PD models have demonstrated exceptional capability in addressing hydro-mechanical coupled problems that encompass crack-related phenomena.[3,10–13] In the FEM-PD models, Peridynamics (PD), a novel and alternative non-local continuum theory founded on integral-differential equations with unique capabilities for simulating discontinuities in solids,[14,15] is employed to characterize the deformation of the solid skeleton and to capture the propagation of cracks, while finite element equations are used to describe the fluid flow within both intact and fractured porous media. Despite their unparalleled advantages in addressing crack propagation issues within the hydro-mechanical coupled context, the FEM-PD model does exhibit a common drawback shared by all PD-based approaches, namely, relatively low computational efficiency. This limitation primarily arises from the inherent non-local characteristics of the PD.

To overcome this shortcoming, two potential strategies can be employed: (i) reducing the computational costs and (ii) implementing parallel programming techniques. In order to reduce the overall computing cost of the PD portion, several different methods have been proposed to couple PD-based models to models based on classical mechanics at either continuous or discrete levels, as demonstrated in prior studies.[16–21] Furthermore, parallel programming techniques for PD models have been introduced in several research papers. In Reference 22, the implementation of bond-based peridynamic (BB-PD) models was integrated into the molecular dynamics package LAMMPS (Large-scale Atomic/Molecular Massively Parallel Simulator). The calculations were parallelized using the distributed-memory Message Passing Interface (MPI). The PD code, *EMU*, was created as a research code, as outlined in Reference 23. Subsequently, a scalable parallel code named *PDQ* was developed and introduced in Reference 24 for use with BB-PD models. The open-source PD code Peridigm, originally developed at Sandia National Laboratories, was initially released in 2011, as documented in Reference 25. Notably, Peridigm enables parallel calculations to be executed efficiently across multiple CPU cores. In the work presented in Reference 26, a parallel code for the BB-PD model was developed in Fortran, employing the Open Multi-Processing application programming interface (OpenMP). This implementation involved coupling the PD model with the finite element method to effectively reduce computational costs. In Reference 27, a parallel implementation of the BB-PD model was executed on the Sunway Taihulight Supercomputer. In Reference 28, an innovative approach to parallelization was introduced, involving asynchronous and task-based implementation of PD models, utilizing High Performance ParalleX (HPX) in the C++ programming language. Indeed, alongside the CPU-based (Central Processing Unit) parallelization techniques, there is a growing focus on GPU-based (Graphics Processing Unit) parallel programming in the realm of high-performance computing. In the work described in Reference 29, a PD code was implemented using GPUs to enable highly parallel computation. This was achieved by leveraging OpenACC, a programming standard designed to simplify parallel programming for heterogeneous CPU/GPU systems, which was developed collaboratively by Cray, CAPS, Nvidia, and PGI. In Reference 30, the Compute Unified Device Architecture (CUDA), which is a parallel architecture specifically designed for NVIDIA GPUs, was exploited to implement a PD model. In the study outlined in Reference 31, the use of OpenACC was adopted to expedite the calculation of the internal PD force density on CUDA-enabled GPU devices. An alternative approach for the GPU-based implementation of PD models involves the use of OpenCL, a framework designed for creating programs that can operate across heterogeneous platforms, encompassing both CPUs and GPUs, among others, and the relevant work can be found in Reference 32.

The general consensus is that vectorized calculations are typically faster than nested loop calculations, a conclusion supported by findings in Reference 31. Taking inspiration from this, we have introduced a matrix-based implementation approach for linearized ordinary state-based peridynamic (OSB-PD) models and have conducted a comparative study with a non-parallel loop-based scheme, as detailed in Reference 33. Based on that work, this paper introduces a fast explicit solution scheme for the hybrid FEM-PD models proposed in Reference 3, where both the FE and PD equations are addressed through explicit time integration algorithms. An in-house software is developed in MATLAB, including CPU and GPU versions. Then several typical numerical examples are solved by using the developed software to investigate its capability and effectiveness in simulating the hydro-mechanical coupled problems in saturated porous media.

The main contributions of this paper are outlined as follows:

- A fast fully explicit solution scheme is proposed for the implementation of hybrid FEM-PD model for hydro-mechanical coupled problems in saturated porous media;
- A detailed explanation of the method for determining the critical time increment, which guarantees numerical stability, is provided;
- An in-house FEM-PD software is developed in MATLAB, comprising both CPU and GPU versions;
- The developed software is successfully applied with remarkable high performance to simulate dynamic problems in saturated porous media under the influence of hydro-mechanical coupling.

The structure of this paper is as follows. In Section 2, a concise overview of the FE and OSB-PD equations pertaining to hydro-mechanical coupled problems in fracturing saturated porous media is provided. Section 3 describes the model discretization and numerical implementation of the proposed approach. In Section 4, several numerical examples are presented and discussed. Finally, Section 5 concludes the paper.

2 | MODEL DESCRIPTION

2.1 | Ordinary state-based peridynamic model for the fracturing saturated porous media

Shown as Figure 1, a solid body $\mathcal{B}$ governed by a PD model is assumed to be composed by a series of material points. Each of these material points represents a specific location within the body and occupies an infinitesimally small volume or region. Given that $\boldsymbol{x}$ represents a material point within the studied body, it is subject to interactions with all other points located within a specified neighborhood. The neighborhood can be marked as $\mathcal{H}_{\boldsymbol{x}}$, which is usually a circle in 2D and a sphere in 3D, being described by the radius δ (the *horizon*) and mathematically defined as:

$$\mathcal{H}_{\boldsymbol{x}} = \mathcal{H}(\boldsymbol{x}, \delta) = \{\|\boldsymbol{\xi}\| \leq \delta : \boldsymbol{x}' \in \mathcal{B}\} \tag{1}$$

where $\boldsymbol{x}'$ represents a point within $\boldsymbol{x}$'s neighbourhood. The vector $\boldsymbol{\xi}$ describes the relative position between the points $\boldsymbol{x}$ and $\boldsymbol{x}'$ in the initial configuration ($\mathcal{B}_0$):

$$\boldsymbol{\xi} = \boldsymbol{x}' - \boldsymbol{x} \tag{2}$$

and the symbol $\|\cdot\|$ denotes the Euclidean norm, thus, $\|\boldsymbol{\xi}\|$ represents the distance between the two points. In fact, within the framework of PD theory, a concept known as 'bond' is introduced to characterize the connection between points, and $\boldsymbol{\xi}$ represents the bond connecting the points $\boldsymbol{x}$ and $\boldsymbol{x}'$.

As shown in Figure 1, in the deformed configuration (denoted as $\mathcal{B}_t$), points $\boldsymbol{x}$ and $\boldsymbol{x}'$ experience displacements represented by $\boldsymbol{u}'$ and $\boldsymbol{u}'$, respectively. Consequently, the relative displacement vector between these two points is expressed as:

$$\boldsymbol{\eta} = \boldsymbol{u}' - \boldsymbol{u} \tag{3}$$

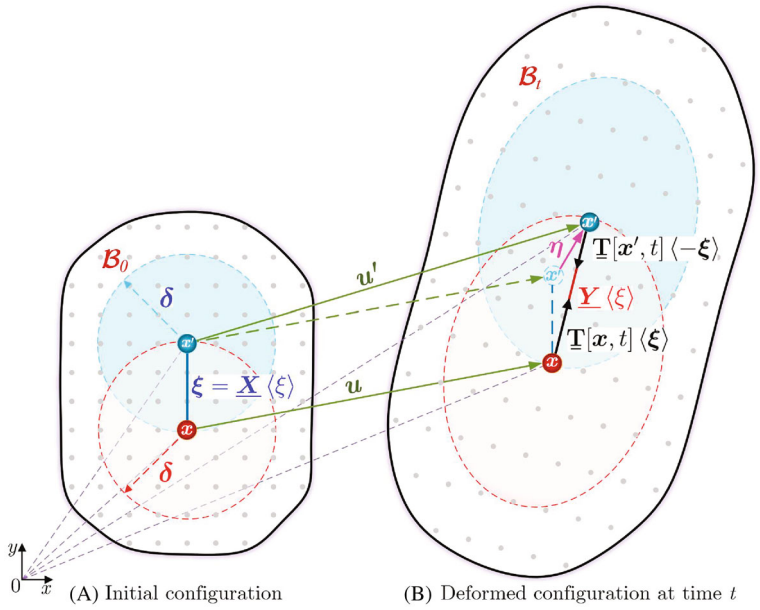


FIGURE 1 Schematic diagram of a PD model in its initial (left) and deformed (right) configurations.

Within the framework of OSB-PD theory, the reference vector state, denoted as $\underline{\boldsymbol{X}}\langle\boldsymbol{\xi}\rangle$, and the deformation vector state, denoted as $\underline{\boldsymbol{Y}}\langle\boldsymbol{\xi}\rangle$, are defined as:

$$\underline{\boldsymbol{X}}\langle\boldsymbol{\xi}\rangle = \boldsymbol{\xi} \text{and} \quad \underline{\boldsymbol{Y}}\langle\boldsymbol{\xi}\rangle = \boldsymbol{\xi} + \boldsymbol{\eta} \tag{4}$$

In addition, the reference position scalar state and deformation scalar state are defined as:

$$\underline{x} = \|\underline{\boldsymbol{X}}\| \quad \text{and} \quad \underline{y} = \|\underline{\boldsymbol{Y}}\| \ , \tag{5}$$

respectively, which represent the lengths of the bond in its initial and deformed states.

As in References 15,34, the strain energy density at point $\boldsymbol{x}$ in an isotropic elastic body can be expressed as:

$$W\left(\theta, \underline{e}^d\right) = \frac{\kappa\theta^2}{2} + \frac{15\mu}{2m}\int_{\mathcal{H}_x} \underline{w}\,\underline{e}^d\,\underline{e}^d dV_{x'} \tag{6}$$

where $dV_{x'}$ is the infinitesimal volume associated to point $\boldsymbol{x}'$. The parameters κ and μ represent the bulk and shear moduli of the material, respectively, and they are defined as:

$$\kappa = \frac{E}{3(1-2\nu)} \tag{7}$$

and

$$\mu = \frac{E}{2(1+\nu)}, \tag{8}$$

in which E and ν represent the Young's modulus and Poisson's ratio of the material.

The parameter θ represents the volume dilatation value at point $\boldsymbol{x}$, while $\underline{e}^d$ denotes the deviatoric extension state. Their definitions are provided as follows:[35–37]

$$\theta = \frac{A}{m}\int_{\mathcal{H}_x} \underline{w}\,\underline{x}\,\underline{e} dV_{x'} \tag{9}$$

and

$$\underline{e}^d = \underline{e} - \frac{\theta \underline{x}}{3}, \tag{10}$$

where A is a constant given as:

$$A = \begin{cases} 3\ , & \text{3D} \\ 2\ , & \text{plane strain} \end{cases}, \tag{11}$$

m represents the weighted volume, which is defined as:

$$m = \int_{\mathcal{H}_x} \underline{w}\, \underline{x}\, \underline{x} dV_{x'}, \tag{12}$$

where $\underline{w}$ denotes the influence function, and the various possible forms have been summarized in Reference 21. In this paper, for the sake of convenience, we adopt the value $\underline{w} = 1$. Furthermore, $\underline{e}$ is the extension scalar state, describing the longitudinal deformation of the bond, which is defined as:

$$\underline{e} = \underline{y} - \underline{x} \tag{13}$$

As derived in References 3,7, the hydro-mechanical coupled OSB-PD equation of motion for the fluid saturated porous media can be given as:

$$\rho \ddot{\boldsymbol{u}}(\boldsymbol{x}, t) = \int_{\mathcal{H}_x} \left\{ \underline{\mathbf{T}}[\boldsymbol{x}, p, t]\langle \boldsymbol{\xi} \rangle - \underline{\mathbf{T}}\left[\boldsymbol{x}', p', t\right]\langle \boldsymbol{\xi} \rangle \right\} dV_{x'} + \boldsymbol{b}(\boldsymbol{x}, t), \tag{14}$$

in which, ρ is the mass density of the two-phase porous media, which can be defined as:[38]

$$\rho = (1 - n)\rho_s + n\rho_w \tag{15}$$

where n is the porosity; ρ_s and ρ_w are the mass densities of the solid and fluid phases. $\ddot{\boldsymbol{u}}$ is the acceleration; $\boldsymbol{b}$ is the force density applied by an external force field. p and p' are the values of pore pressure at points $\boldsymbol{x}$ and $\boldsymbol{x}'$, respectively. $\underline{\mathbf{T}}[\boldsymbol{x}, p, t]\langle \boldsymbol{\xi} \rangle$ and $\underline{\mathbf{T}}\left[\boldsymbol{x}', p, t\right]\langle -\boldsymbol{\xi} \rangle$ are the total force density vector states of points $\boldsymbol{x}$ and $\boldsymbol{x}'$ in the deformed configuration at time t. The force density acting on $\boldsymbol{x}'$ due to point $\boldsymbol{x}$ is aligned with the deformed bond in the OSB-PD theory. Consequently, the total force density vector state $\underline{\mathbf{T}}[\boldsymbol{x}, p, t]\langle \boldsymbol{\xi} \rangle$ can be defined as:[3]

$$\underline{\mathbf{T}}[\boldsymbol{x}, p, t]\langle \boldsymbol{\xi} \rangle = \underline{t}^{tot} \cdot \underline{\boldsymbol{M}}\langle \boldsymbol{\xi} \rangle \tag{16}$$

where $\underline{\boldsymbol{M}}\langle \boldsymbol{\xi} \rangle$ is a unit state in the direction of $\underline{\boldsymbol{Y}}$ defined as:

$$\underline{\boldsymbol{M}}\langle \boldsymbol{\xi} \rangle = \frac{\underline{\boldsymbol{Y}}}{\|\underline{\boldsymbol{Y}}\|}, \tag{17}$$

which has the approximation $\underline{\boldsymbol{M}}\langle \boldsymbol{\xi} \rangle \approx \underline{\boldsymbol{X}}/\|\underline{\boldsymbol{X}}\|$ under the hypothesis of infinitesimal deformations. $\underline{t}^{tot}$ is called the total force density scalar state. The expression of $\underline{t}^{tot}$ in 3D case can be given as:

$$\underline{t}^{tot} = (K\theta - 3\alpha p)\frac{\underline{w}\, \underline{x}}{m} + G\underline{e}^d \frac{\underline{w}}{m} = \left[\left(K - \frac{G}{3}\right)\theta - 3\alpha p\right]\frac{\underline{w}\, \underline{x}}{m} + G\underline{e}\frac{\underline{w}}{m}, \tag{18}$$

while in the context of plane strain, the expression for $\underline{t}^{tot}$ is given as:

$$\underline{t}^{tot} = \left[\left(K - \frac{G}{3}\right)\theta - 2\alpha p\right]\frac{\underline{w}\, \underline{x}}{m} + G\underline{e}\frac{\underline{w}}{m} \tag{19}$$

In Equations (18) to (19), α is the Biot coefficient; the constants K and G represent positive values that are associated with material parameters, which are defined as follows:

$$K = \begin{cases} 3\kappa\ , & \text{3D} \\ 2\kappa - \frac{2}{3}\mu\ , & \text{plane strain} \end{cases} \tag{20}$$

$$G = \begin{cases} 15\mu\ , & \text{3D} \\ 8\mu\ , & \text{plane strain} \end{cases} \tag{21}$$

To describe the failure and crack propagation in solid materials, the use of bond-breakage criteria is indispensable in a PD model. Referring to Reference 39, a specific 'critical bond stretch' criterion has been developed for OSB-PD models in References 7,33, and it is also applied in this paper.

In the adopted failure criterion, a scalar variable ϱ is defined to assess the bond connection status, which is expressed as:[19,40]

$$\underline{\varrho}\langle\boldsymbol{\xi}\rangle = \begin{cases} 1\ , & \text{if } s\langle\boldsymbol{\xi}\rangle < s_c \\ 0\ , & \text{otherwise} \end{cases}, \tag{22}$$

where $s\langle\boldsymbol{\xi}\rangle$ represents the stretch value of bond $\boldsymbol{\xi}$, and it is defined as:

$$s\langle\boldsymbol{\xi}\rangle = \frac{\underline{e}\langle\boldsymbol{\xi}\rangle}{\underline{x}\langle\boldsymbol{\xi}\rangle}, \tag{23}$$

s_c is the critical stretch value, which is given as:

$$s_c = \begin{cases} \sqrt{\frac{G_c}{(9\kappa-15\mu)\Lambda+15\mu\Lambda'}}\ , & \text{3D} \\ \sqrt{\frac{G_c}{\left(4\kappa-\frac{28}{9}\mu\right)\Lambda+8\mu\Lambda'}}\ , & \text{plane strain} \end{cases}, \tag{24}$$

where the expressions of Λ and Λ' can be obtained as:

$$\Lambda = \begin{cases} \frac{125\delta}{1848}\ , & \text{3D} \\ \frac{1087\delta}{1250\pi^2}\ , & \text{plane strain} \end{cases} \tag{25}$$

and

$$\Lambda' = \begin{cases} \frac{5\delta}{24\pi}\ , & \text{3D} \\ \frac{4\delta}{5\pi}\ , & \text{plane strain} \end{cases} \tag{26}$$

In order to characterize the failure and cracks, the damage level φ_x at point $\boldsymbol{x}$ is defined and expressed as:

$$\varphi_x = 1 - \frac{\int_{\mathcal{H}_x} \underline{w}\langle\boldsymbol{\xi}\rangle\ \underline{\varrho}\langle\boldsymbol{\xi}\rangle \mathrm{d}V_{x'}}{\int_{\mathcal{H}_x} \underline{w}\langle\boldsymbol{\xi}\rangle \mathrm{d}V_{x'}}, \tag{27}$$

where φ_x is constrained within the range of $[0, 1]$, and the presence of cracks can be identified whenever $\varphi_x \geqslant 0.5$.

2.2 | Classical continuum mechanics-based governing equations for fluid flow in fractured saturated porous media

As illustrated in the fig. 4a of Reference 3, the entire domain of a fractured porous medium modeled by using the PD approach can be categorized into three regions based on the damage levels φ defined in Equation (27), represented as Ω_r, Ω_f and Ω_t. These regions are delimited by two threshold values, c_1 and c_2, which are defined as follows:

- The unbroken domain (reservoir domain), denoted as Ω_r, is identified when $\varphi \leq c_1$;
- The fracture domain, labeled as Ω_f, is recognized when $\varphi \geq c_2$;
- The transition domain, referred to as Ω_t, is characterized when $c_1 \leq \varphi \leq c_2$.

By employing Darcy's law to depict fluid flow in saturated porous media, the governing equation, that is, the mass balance equation, describing the fluid flow in the reservoir domain is provided as follows:[38]

$$s_r \frac{\partial p}{\partial t} + \alpha_r \frac{\partial \varepsilon_v}{\partial t} + \nabla \cdot \left[\frac{k_r}{\mu_w} (-\nabla p + \rho_w g) \right] = 0 \tag{28}$$

where s_r, α_r and k_r represent the storage coefficient, Biot coefficient and permeability of the media in the reservoir domain, respectively. Additionally, μ_w and ρ_w denotes the viscosity coefficient and density of the fluid, while ε_v signifies volumetric strain and g is gravity. The storage coefficient is defined as:[38]

$$s_r = \frac{(\alpha_r - n_r)(1 - \alpha_r)}{K_r} + \frac{n_r}{K_w}, \tag{29}$$

where K_r and K_w denote the bulk moduli of solid and fluid in the reservoir domain, respectively, and n_r represents the porosity.

Similarly, the governing equation for the fluid flow in the fracture domain can be given as:

$$s_f \frac{\partial p}{\partial t} + \alpha_f \frac{\partial \varepsilon_v}{\partial t} + \nabla \cdot \left[\frac{k_f}{\mu_w} (-\nabla p + \rho_w g) \right] = 0 \tag{30}$$

where s_f, α_f and k_f are the storage coefficient, Biot coefficient and permeability of the media in the fracture domain. For simplification purposes, as employed in Reference 3 it is assumed that the fracture domain is completely filled with fluid. Based on this assumption, the cubic law can be utilized to determine the permeability of the fracture domain:[1,41]

$$k_f = \frac{1}{12} a^2, \tag{31}$$

and a is the aperture of the macro crack. The cubic law is derived from the parallel plate model, representing a simplest model of flow through a rock fracture,[42] where the fracture is conceptualized as being confined between two smooth, parallel walls with a separation distance denoted as aperture a. Remarkably, this model stands as the sole geometric representation for which a precise determination of hydraulic conductivity is achievable. By comparing the expression of total volumetric flux within the fracture with the one-dimensional Darcy's law formula, the permeability coefficient of the fracture is obtained as Equation (31).[43]

Note that for the fracture domain, the porosity and Biot coefficient are assumed to be $n_f = 1$ and $\alpha_f = 1$, following the works of References 1,3. Additionally, the corresponding storage coefficient is considered as $s_f = \frac{1}{K_w}$.

The transition domain is introduced to facilitate a gradual transition between the fracture domain and the reservoir domain, as discussed in Reference 1. The governing equation for the fluid flow in the transition domain can be defined as:

$$s_T \frac{\partial p}{\partial t} + \alpha_T \frac{\partial \varepsilon_v}{\partial t} + \nabla \cdot \left[\frac{k_T}{\mu_w} (-\nabla p + \rho_w g) \right] = 0 \tag{32}$$

Utilizing the linear indicator functions χ_r and χ_f defined in Equations (29)–(31) of Reference 3, the relevant material parameters in the transition domain can be computed through interpolation between those in the reservoir and fracture domains. This interpolation yields the following expressions:

- Storage coefficient: $s_T = s_r\chi_r + s_f\chi_f$;
- Biot coefficient: $\alpha_T = \alpha_r\chi_r + \alpha_f\chi_f$;
- Porosity: $n_T = n_r\chi_r + n_f\chi_f$;
- Permeability:$k_T = k_r\chi_r + k_f\chi_f$.

In order to solve the governing equations, it is necessary to establish the initial and boundary conditions within the solution domain (Ω) and on the boundary (Γ). The initial conditions for the pore pressure field at time $t = 0$ are provided as[38]:

$$p = p_0 \quad \text{in } \Omega \quad \text{and} \quad \text{on } \Gamma \tag{33}$$

In addition, the Dirichlet boundary conditions are given by:[38]

$$p = p_c \quad \text{on } \Gamma_w \ , \tag{34}$$

and the Neumann boundary conditions for the fluid flow are defined as:[38]

$$\rho_w \frac{k}{\mu_w}(-\nabla p + \rho_w g)^{\mathrm{T}} \cdot \mathbf{n} = q_w \quad \text{on } \Gamma_w^q, \tag{35}$$

where q_w is the fluid source and $\mathbf{n}$ is the unit vector normal to the applied boundary.

3 | DISCRETIZATION AND NUMERICAL IMPLEMENTATION

In this section, we will use the three-dimensional case as an illustrative example to introduce the discretization and numerical implementation of the FEM-PD model.

3.1 | Discretization of the hydro-mechanical coupled PD equations

The discretized hydro-mechanical coupled OSB-PD equation of motion of the reference point $\boldsymbol{x}_i$ at time instant t is expressed as:[10,44]

$$\rho\ddot{\boldsymbol{u}}_i^t = \sum_{j=1}^{N_{\mathcal{H}_i}} \left\{\underline{\mathbf{T}}\left[\boldsymbol{x}_i, p_i, t\right]\left\langle \boldsymbol{\xi}_{ij}\right\rangle - \underline{\mathbf{T}}\left[\boldsymbol{x}_j, p_j, t\right]\left\langle -\boldsymbol{\xi}_{ij}\right\rangle\right\} \cdot V_j + \boldsymbol{b}_i^t \tag{36}$$

$$= \sum_{j=1}^{N_{\mathcal{H}_i}} \left\{\underline{\mathbf{T}}[\boldsymbol{x}_i, t]\left\langle \boldsymbol{\xi}_{ij}\right\rangle - \underline{\mathbf{T}}\left[\boldsymbol{x}_j, t\right]\left\langle -\boldsymbol{\xi}_{ij}\right\rangle\right\} \cdot V_j \tag{37}$$

$$- 3\alpha \sum_{j=1}^{N_{\mathcal{H}_i}} \left[p_i \frac{\underline{w}_{ij}\,\underline{x}_{ij}}{m_i}\underline{\boldsymbol{M}}\left\langle \boldsymbol{\xi}_{ij}\right\rangle - \ p_j \frac{\underline{w}_{ij}\,\underline{x}_{ij}}{m_j}\underline{\boldsymbol{M}}\left\langle -\boldsymbol{\xi}_{ij}\right\rangle\right] \cdot V_j + \boldsymbol{b}_i^t \tag{38}$$

where $\boldsymbol{x}_j$ represents a family node of $\boldsymbol{x}_i$, and $N_{\mathcal{H}_i}$ is the number of family nodes in $\boldsymbol{x}_i$'s neighbourhood. V_j corresponds to the volume associated with node $\boldsymbol{x}_j$. p_i and p_j represent the pore pressure values at nodes $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, respectively.

Equation (38) can also be reformulated as:[7]

$$\rho V_i \ddot{\boldsymbol{u}}_i^t = \sum_{j=1}^{N_{\mathcal{H}_i}} \mathbf{F}_{ssi}^{\xi_{ij}} - \sum_{j=1}^{N_{\mathcal{H}_i}} \mathbf{F}_{swi}^{\xi_{ij}} + \boldsymbol{b}_i^t V_i \tag{39}$$

where, $\mathbf{F}_{ssi}^{\xi_{ij}}$ and $\mathbf{F}_{swi}^{\xi_{ij}}$ represent the forces related to solid deformation and pore pressure acting on node $\boldsymbol{x}_i$ through the deformed bond $\boldsymbol{\xi}_{ij}$. Similarly, $\mathbf{F}_{ssj}^{\xi_{ij}}$ and $\mathbf{F}_{swj}^{\xi_{ij}}$ denote the forces applied to node $\boldsymbol{x}_j$. Based on Equation (18), these forces can be calculated by:

$$\begin{bmatrix} \mathbf{F}_{ssi}^{\xi_{ij}} \\ \mathbf{F}_{ssj}^{\xi_{ij}} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{ij}^{s\theta} \end{bmatrix} \begin{bmatrix} \theta_i \\ \theta_j \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{ij}^{s\underline{e}} \end{bmatrix} \underline{e}_{ij} \tag{40}$$

$$\begin{bmatrix} \mathbf{F}_{swi}^{\xi_{ij}} \\ \mathbf{F}_{swj}^{\xi_{ij}} \end{bmatrix} = \begin{bmatrix} \mathbf{K}_{ij}^{sw} \end{bmatrix} \begin{bmatrix} p_i \\ p_j \end{bmatrix} \tag{41}$$

where θ_i and θ_j are the volume dilatation values at nodes $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, and the contributions of the deformed bond $\boldsymbol{\xi}_{ij}$ to these two values can be computed by:

$$\begin{bmatrix} \theta_i^{\xi_{ij}} \\ \theta_j^{\xi_{ij}} \end{bmatrix} = \begin{bmatrix} \mathbf{C}_{ij}^{\theta} \end{bmatrix} \underline{e}_{ij} \tag{42}$$

$\underline{e}_{ij}$ is the extension scalar state value of bond $\boldsymbol{\xi}_{ij}$, which can be obtained by:

$$\underline{e}_{ij} = \begin{bmatrix} \mathbf{C}_{ij}^{\underline{e}} \end{bmatrix} \begin{bmatrix} \boldsymbol{U}_i \\ \boldsymbol{U}_j \end{bmatrix}, \tag{43}$$

where $\boldsymbol{U}_i = [U_{i1}, U_{i2}, U_{i3}]^T$ and $\boldsymbol{U}_j = [U_{j1}, U_{j2}, U_{j3}]^T$ represent the displacement vectors of nodes $\boldsymbol{x}_i$ and $\boldsymbol{x}_j$, respectively. $\left[\mathbf{K}_{ij}^{s\theta}\right]$, $\left[\mathbf{K}_{ij}^{s\underline{e}}\right]$, $\left[\mathbf{K}_{ij}^{sw}\right]$, $\left[\mathbf{C}_{ij}^{\theta}\right]$ and $\left[\mathbf{C}_{ij}^{\underline{e}}\right]$ are the bond-associated matrices, and their detailed expressions can be found in References 7,33.

Drawing from the principles outlined in the sect. 3.1 of Reference 33 and the appendix C of Reference 7, the assembly of all bond-associated matrices results in the global matrix representation of the equation of motion, given as:

$$\mathbf{M}\ddot{\boldsymbol{u}} + \mathbf{K}_{ss}\boldsymbol{u} - \mathbf{K}_{sw}\boldsymbol{p} = \mathbf{F}^{ext}, \tag{44}$$

where $\mathbf{M}$ denotes the mass matrix, typically selected in a lumped form for the PD-based approaches, with its main diagonal entries specified as:

$$M_{ii} = \rho V_i, \tag{45}$$

$\mathbf{K}_{ss}$ is the global stiffness matrix of OSB-PD model, which can be obtained by:[33,36]

$$\mathbf{K}_{ss} = \left(\mathbf{K}_{s\theta} \cdot \mathbf{C}_{\theta} + \mathbf{K}_{s\underline{e}}\right) \cdot \mathbf{C}_{\underline{e}}, \tag{46}$$

where the matrices $\mathbf{C}_{\underline{e}}$, $\mathbf{K}_{s\theta}$, $\mathbf{K}_{s\underline{e}}$ and $\mathbf{C}_{\theta}$, as well as $\mathbf{K}_{sw}$ in Equation (44), represent the global forms of the bond-associated matrices provided in eqs. (C.9) to (C.14) of Reference 7.

3.2 | FEM discretization of the governing equation for fluid flow

By applying the weighted residual method with an appropriate selection of weighting functions $\mathbf{w}^w$ to the governing equation and boundary conditions, we obtain the following weak form:[38]

$$\int_\Omega \left[\left(\mathbf{w}^w\right)^{\mathrm{T}} s \frac{\partial \mathbf{p}}{\partial t} + \left(\mathbf{w}^w\right)^{\mathrm{T}} \alpha \mathbf{m}^{\mathrm{T}} \mathbf{L} \frac{\partial \mathbf{u}}{\partial t} \right] d\Omega \tag{47}$$

$$- \int_\Omega \left[\left(\nabla \mathbf{w}^w\right)^{\mathrm{T}} \left(-\frac{\mathbf{k}}{\mu_w} \nabla \mathbf{p} + \frac{\mathbf{k}}{\mu_w} \rho_w \mathbf{g} \right) \right] d\Omega + \int_{\Gamma_w^q} \left[\left(\mathbf{w}^w\right)^{\mathrm{T}} \frac{q_w}{\rho_w} \right] d\Gamma = 0 \tag{48}$$

for the fluid flow process in the saturated porous media.

Subsequently, the finite element method can be employed to implement the spatial discretization of the governing equation presented in its weak form. When applying the Galerkin method with the shape functions $\boldsymbol{N}_u$ and $\boldsymbol{N}_p$ for the fields of displacement and pore pressure, the governing equations undergo the following transformation:

$$\int_\Omega \left[\boldsymbol{N}_p^{\mathrm{T}} s \boldsymbol{N}_p \frac{\partial \mathbf{p}}{\partial t} + \boldsymbol{N}_u^{\mathrm{T}} \alpha \mathbf{m}^{\mathrm{T}} \mathbf{L} \boldsymbol{N}_u \frac{\partial \mathbf{u}}{\partial t} \right] d\Omega \tag{49}$$

$$+ \int_\Omega \left[\left(\nabla \boldsymbol{N}_p\right)^{\mathrm{T}} \frac{\mathbf{k}}{\mu_w} \left(\nabla \boldsymbol{N}_p\right) \mathbf{p} - \left(\nabla \boldsymbol{N}_p\right)^{\mathrm{T}} \frac{\mathbf{k}}{\mu_w} \rho_w \mathbf{g} \right] d\Omega + \int_{\Gamma_w^q} \left[\boldsymbol{N}_p^{\mathrm{T}} \frac{q_w}{\rho_w} \right] d\Gamma = 0 \tag{50}$$

which can be rewritten in the following form:[38]

$$\mathbf{C}_{ww} \dot{\boldsymbol{p}} + \mathbf{K}_{ws} \dot{\boldsymbol{u}} + \mathbf{K}_{ww} \boldsymbol{p} = \mathbf{Q}_w \tag{51}$$

where $\mathbf{C}_{ww}$, $\mathbf{K}_{ws}$ and $\mathbf{K}_{ww}$ represent the storage, coupling and permeability matrices, and $\mathbf{Q}_w$ the source vector, respectively. These matrices and vector can be computed by:[38]

$$\mathbf{C}_{ww} = \int_\Omega \boldsymbol{N}_p^{\mathrm{T}} s \boldsymbol{N}_p \mathrm{d}\Omega \tag{52}$$

$$\mathbf{K}_{ws} = \int_\Omega (\boldsymbol{L}\boldsymbol{N}_u)^{\mathrm{T}} \alpha \mathbf{m}^{\mathrm{T}} \boldsymbol{N}_p \mathrm{d}\Omega \tag{53}$$

$$\mathbf{K}_{ww} = \int_\Omega \left(\nabla \boldsymbol{N}_p\right)^{\mathrm{T}} \frac{k}{\mu_w} \left(\nabla \boldsymbol{N}_p\right) \mathrm{d}\Omega \tag{54}$$

$$\mathbf{Q}_w = \int_\Omega \left(\nabla \boldsymbol{N}_p\right)^{\mathrm{T}} \frac{\mathbf{k}}{\mu_w} \rho_w \mathbf{g} d\Omega - \int_{\Gamma_w^q} \left[\boldsymbol{N}_p^{\mathrm{T}} \frac{q_w}{\rho_w} \right] d\Gamma \tag{55}$$

In the equations presented above, $\boldsymbol{L}$ is a differential operator defined as:

$$\boldsymbol{L} = \begin{bmatrix} \frac{\partial}{\partial x} & 0 & 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial z} \\ 0 & \frac{\partial}{\partial y} & 0 & \frac{\partial}{\partial x} & \frac{\partial}{\partial z} & 0 \\ 0 & 0 & \frac{\partial}{\partial z} & 0 & \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix}^{\mathrm{T}}, \tag{56}$$

and $\mathbf{m}$ is a vector defined as:

$$\mathbf{m} = [1, 1, 1, 0, 0, 0] \tag{57}$$

3.3 | Explicit time integration schemes for the dynamic solutions of the FEM-PD coupled model

The dynamic solution of the OSB-PD model can be achieved through the implementation of a modified explicit central difference time integration scheme, as elaborated in Reference 45. This scheme involves the integration of velocities using a forward difference method and displacements using a backward difference method. Consequently, the velocity and displacement at the $(n+1)^{th}$ time increment can be computed by:

$$\begin{aligned} \dot{\boldsymbol{u}}^{n+1} &= \dot{\boldsymbol{u}}^{n} + \Delta t \ddot{\boldsymbol{u}}^{n} \\ \boldsymbol{u}^{n+1} &= \boldsymbol{u}^{n} + \Delta t \dot{\boldsymbol{u}}^{n+1} \end{aligned} \tag{58}$$

where $\ddot{\boldsymbol{u}}^{n}$ denotes the acceleration at the n^{th} time increment, by considering the pore pressure, it can be determined through the application of Newton's second law:

$$\ddot{\boldsymbol{u}}^{n} = \mathbf{M}^{-1}\left(\mathbf{F}^{ext} + \mathbf{K}_{sw} \cdot \boldsymbol{p}^{n} - \mathbf{F}^{int}\right), \tag{59}$$

in which, $\mathbf{F}^{ext}$ and $\mathbf{F}^{int} = \mathbf{K}_{ss}\boldsymbol{u}^{n-1}$ represent the external and internal force vectors, respectively. $\boldsymbol{p}^{n}$ is the vector of pore pressure at the n^{th} time increment. $\mathbf{M}$ is the diagonal mass matrix, therefore, its inverse matrix is also in diagonal form, with the main diagonal entries being $1/M_{ii} = 1/(\rho V_i)$.

In addition to the consistent form presented in Equation (52), the storage matrix can also be represented in a diagonal form (marked as $\underline{\mathbf{C}}_{ww}$ for the later use), with its main diagonal entries being defined as:

$$C_{ii}^{ww} = sV_i \tag{60}$$

Based on that, the pore pressure field at the $(n+1)^{th}$ time increment can be updated by the using following explicit time integration scheme:

$$\boldsymbol{p}^{n+1} = \boldsymbol{p}^{n} + \underline{\mathbf{C}}_{ww}^{-1}\left[\Delta t \mathbf{Q}_{w}^{n} - \Delta t \mathbf{K}_{ww}\boldsymbol{p}^{n} - \mathbf{K}_{ws}(\boldsymbol{u}^{n} - \boldsymbol{u}^{n-1})\right] \tag{61}$$

where the inverse matrix of $\underline{\mathbf{C}}_{ww}$, i.e. $\underline{\mathbf{C}}_{ww}^{-1}$, is also in diagonal form, with the main diagonal entries being $1/C_{ii}^{ww} = 1/(sV_i)$.

The time increment Δt in Equations (58) and (61) is taken as a constant value here. For an undamped system using an explicit method, it is essential to select a time step smaller than the critical time increment (Δt_c) for the numerical stability. Since the governing equations for the solid domain and the fluid domain correspond to different-order time-dependent systems and are solved using distinct time integration schemes, the critical time increment required to ensure their respective numerical stability are typically different and should be estimated using separate methods.

According to Reference 46, the stable time increment for the solid domain of the PD model can be defined as:

$$\Delta t < \Delta t_c^{s} = \delta / c' \tag{62}$$

in which, $c' = \sqrt{(\lambda + 2\mu)/\rho}$ represents the dilatational wave speed, where λ and μ are the Lame's elastic constants of the material, and ρ is the mass density.

For the fluid domain of FE model, the stability condition of the adopted explicit integration scheme can be defined as:[47,48]

$$\Delta t \leq \Delta t_c^{f} = \frac{2}{\lambda_s} \tag{63}$$

where $\lambda_s = \left[\lambda_1, \lambda_2, \ldots, \lambda_{N_n}\right]$ represents the vector of eigenvalues of the system matrix, that is, $\underline{\mathbf{C}}_{ww}^{-1}\mathbf{K}_{ww}$. We then select the maximum eigenvalue, denoted as λ_{max}, from λ_s to satisfy the stability requirements:[48]

$$\Delta t \leq \Delta t_c^{f} = \frac{2}{\lambda_{\max}} \tag{64}$$

In addition, methods for evaluating the maximum eigenvalues and critical time increment in different cases can be found in Reference 49.

Remark 1. The numerical stability represents an inevitable issue encountered by explicit time integration schemes, where a relatively small-time step size must be generally used. This will significantly limit the applicability of explicit integration algorithms in solving engineering-scale problems. Due to its better stability characteristics, implicit time integration is more commonly employed in solving hydro-mechanical coupled problems.[50,51] However, when dealing with issues involving extremely large scales, implicit time integration schemes can be a time-consuming and memory-intensive problem due to the need to perform matrix inversion operations when solving large systems of linear equations. In Reference 12, an implicit time integration has been applied to FEM-PD model for solving hydraulic fracturing problems. Therefore, when employing implicit integration schemes, some available libraries can be used to solving large systems of linear equations leveraging CPU or GPU parallelism,[52] including PARDISO, MUMPS, PETSc, Trilinos, CUSP, MAGMA, cuSOLVER and penBLAS, et al.

3.4 | Implementation and GPU acceleration of the FEM-PD models in MATLAB

The staggered approach is used to solve the described hydro-mechanical coupled problems,[3,7,10] and the roadmap for the implementation of the software is shown in Figure 2. The rectangle blocks with a light-blue background represent pre-processing sections, those with a pink background denote solver sections, and the grey block designates the post-processing section. The rhombus blocks are conditional sections.

Due to the use of an explicit time integration scheme in solving fluid equations, when dealing with problems involving cracks and gradually updating the permeability matrix based on crack aperture, the maximum eigenvalue of the system matrix in the fluid domain will continuously change. This behavior significantly augments the intricacy of algorithm implementation, making the preservation of numerical stability a more challenging endeavor. Referring to the findings outlined in Reference 53, the assumption of constant permeability in the fracture changes slightly the fracture path as compared to the cubic law, and thus, this paper's algorithm will incorporate specific permeability coefficients for fracture domains. On the other hand, employing a partitioning scheme could offer an alternative solution for the case with updating fracture permeabilities, where the fracture domain is segregated from the reservoir domain. Within the reservoir domain, the original critical time step size remains unchanged, whereas for the fracture domain, the critical time step size is dynamically adjusted based on the updated effective permeability. Furthermore, fluid flow within the fracture domain will be iteratively resolved in each solving sequence. Consequently, the stable time increment for a FEM-PD model can be a constant and obtained as:

$$\Delta t_c = \min\left(\Delta t_c^s, \quad \Delta t_c^f\right) \tag{65}$$

MATLAB, an abbreviation of "matrix laboratory", constitutes a proprietary, multi-paradigm programming language and computational environment renowned for its exceptional efficiency in handling vector and matrix operations. The solver sections illustrated in Figure 2 can be effectively realized through the use of vector and matrix operations. Therefore, this paper will implement the FEM-PD software for hydro-mechanical coupled problems within the MATLAB environment, inheriting the framework of the linearized OSB-PD software introduced in Reference 33.

The global matrices $\mathbf{C}_\theta$, $\mathbf{C}_{\underline{e}}$, $\mathbf{K}_{s\underline{e}}$, $\mathbf{K}_{s\theta}$, $\mathbf{K}_{sw}$, $\mathbf{K}_{ws}$ and $\mathbf{K}_{ww}$ play a pivotal role in the matrix-based FEM-PD software. Evidently, these matrices exhibit sparsity and are constructed through the assembly of bond-associated and element matrices. In the software, the system matrices are created and stored using the MATLAB function '*sparse*'. For a 3D discrete FEM-PD model comprising of Nn nodes and N_b bonds, the initialization of essential solver variables within the MATLAB environment is detailed in Appendix A. An illustrative implementation of the solver sections can be found in Appendix B.

In a simulation utilizing explicit iterative solution algorithm, the pre- and post-processing sections will be executed once, while the solver sections will iterate several times. Consequently, enhancing the execution efficiency of the solver sections is pivotal for optimizing the overall computational speed. In MATLAB, the ***Parallel Computing Toolbox*** is available for enabling parallel computations on multi-core processors, GPUs, and computer clusters. This capability can markedly enhance program execution efficiency, especially when tackling computationally and data-intensive tasks. In the solver sections illustrated in Figure 2 and Appendix B, a substantial portion of calculations is vectorized. Therefore,

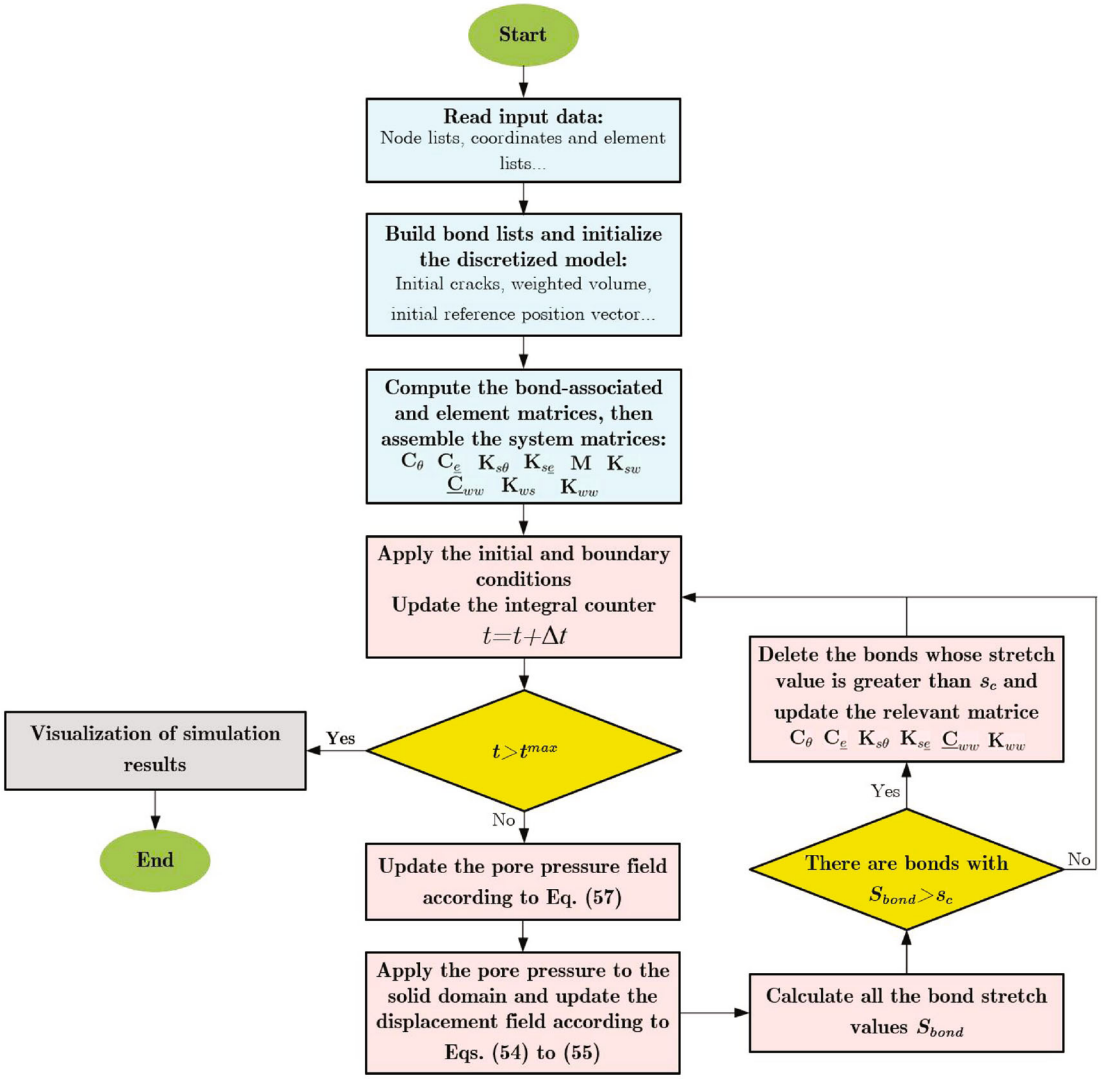


FIGURE 2 Flow chart of the solution algorithm for the FEM-PD coupled model in hydro-mechanical simulations.

the acceleration of the simulation can be easily achieved by transferring the vectors and matrices utilized in the solver to GPUs, a task facilitated by the MATLAB function '*gpuArray*'.

Remark 2. We opted to utilize MATLAB due to several compelling reasons. Firstly, MATLAB boasts comprehensive libraries tailored for numerical computation, facilitating efficient algorithm implementation. Secondly, its user-friendly interface expedites prototyping and rapid development, aligning well with our research objectives. Moreover, MATLAB enjoys widespread acceptance within the academic community, rendering it an ideal choice for research endeavors. While we acknowledge that MATLAB may not offer the same level of performance optimization as lower-level languages, our primary focus in this study was to showcase the effectiveness of our proposed method within a research framework. Additionally, MATLAB's versatility enabled us to swiftly iterate and validate our approach, a crucial aspect given the exploratory nature of our research.

4 | NUMERICAL EXAMPLES

In this section, four examples are presented to demonstrate the effectiveness of the proposed approach and to evaluate the performance of the developed software. All cases have been discretized using uniform grids, with a constant value of the *m*-ratio parameter set to 3.

The in-house software is developed in MATLAB 2023a, and all the simulations are executed on a Desktop computer equipped with a CPU of Intel® Core™ CPU i9-12900K, 3.20 GHz, and 128 GB of RAM. The GPU acceleration is performed on a NVIDIA GeForce RTX 3090 Ti.

4.1 | Example 1: Pore pressure distribution in a single crack

The first example concerns the time-dependent hydraulic pressure distribution within a single crack. This case has previously been addressed in References 2,3,54 to validate their models for simulating the dynamic evolution of pore pressure in fractured saturated porous media. Geometry and boundary conditions are shown in Figure 3.

In the studied specimen, a sudden hydraulic pressure of $P_0 = 9.5$ MPa is applied at the left edge, while the other edges are impermeable. This case represents an 'unsteady' or evolving problem, and the analytical solution is documented in References 2,3. For comparison with the analytical solution, we only focus on the fluid flow portion. The necessary material parameters for the simulation have been listed in Table 1.

The problem is solved using both FEM-PD and PD-only models, with executions performed on both CPU and GPU for comparative analysis. Three cases with different discretization parameters have been considered. The specific spatial and temporal discretization parameters are outlined in Table 2. The total simulation duration is 6.06×10^{-4} s. Figure 4 presents the comparison between the numerical and analytical solutions for pore pressure variations along the crack at the end of the simulations. Corresponding pore pressure distributions are displayed in Figure 5. All FEM-PD solutions exhibit remarkable consistency and closely align with the analytical solution. Conversely, the PD-only solutions exhibit a noticeable difference compared to the analytical solution; however, as Δx decreases, the PD-only solutions gradually converge towards the analytical solution. Figure 6 displays the wall times per thousand iterations for all the conducted cases, confirming that the FEM-PD model outperforms the PD-only model in terms of efficiency. Furthermore, an intriguing observation can be made from Figure 6. In cases with relatively small computational workloads, such as Case 1, the efficiency of execution on CPU may surpass that of GPU. However, as the

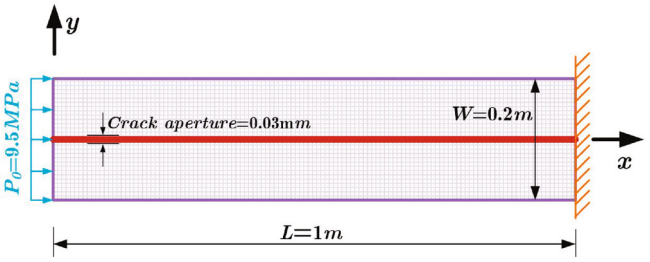


FIGURE 3 Geometry and boundary conditions for a rectangular rock sample with a single crack.

TABLE 1 Mechanical and fluid parameters used in the Example 4.1.

$k_r\ [m^2]$	$k_f\ [m^2]$	α	n	$\mu_w\ [Pa \cdot s]$	$K_w\ [GPa]$
1×10^{-40}	7.5×10^{-11}	1	2×10^{-5}	1×10^{-3}	2.2

TABLE 2 Spatial and temporal discretization parameters used in cases 1, 2 and 3 of the Example 4.1.

Case	$\Delta x\ [mm]$	$\delta\ [mm]$	Number of nodes	$\Delta t\ [s]$	Number of iterations
1	5	15	8442	2×10^{-8}	30,300
2	2.5	7.5	32,882	1×10^{-8}	60,600
3	1.25	3.75	129,762	5×10^{-9}	121,200

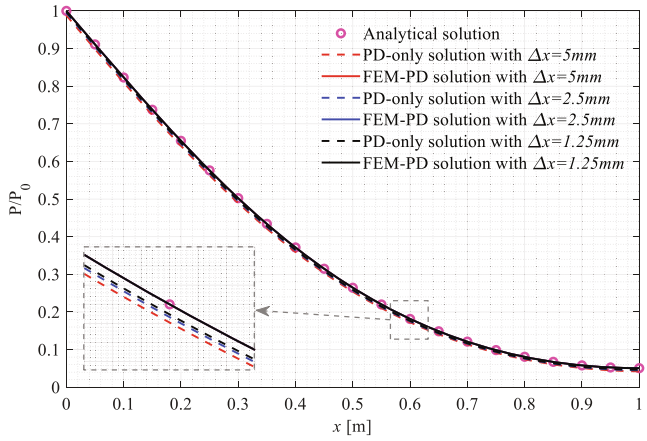


FIGURE 4 Comparison between the analytical and numerical solutions of the pore pressure distribution along the crack.

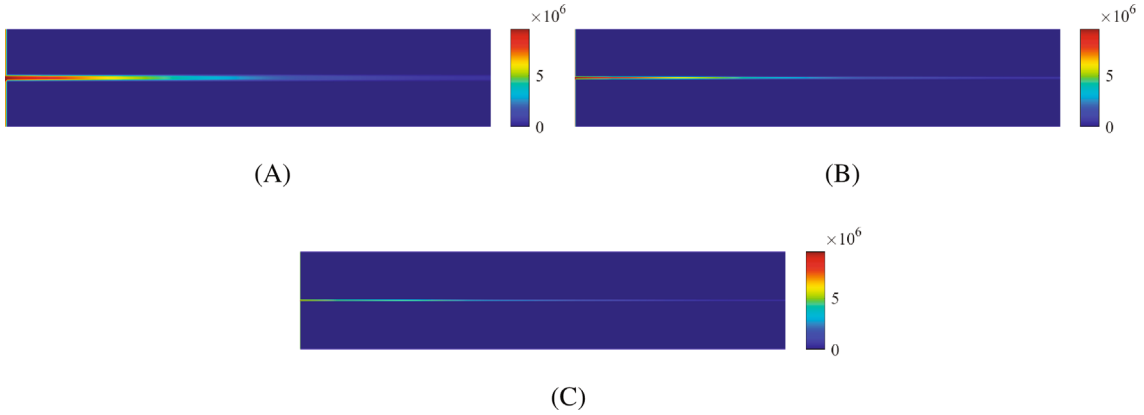


FIGURE 5 Pore pressure distributions in the fractured rock specimen at $t = 6.06 \times 10^{-4} s$ obtained by FEM-PD model in cases (A) 1, (B) 2 and (C) 3. (A) Case 1, (B) Case 2, (C) Case 3.

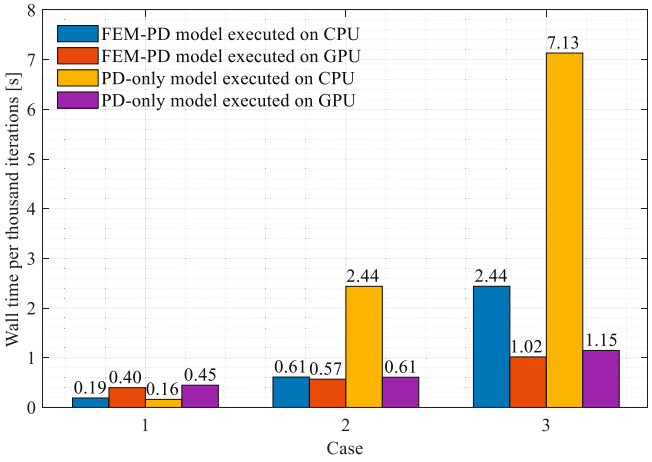


FIGURE 6 Comparison of wall times between the FEM-PD and PD-only models executed on CPU and GPU for the problem of pore pressure distribution in a single crack.

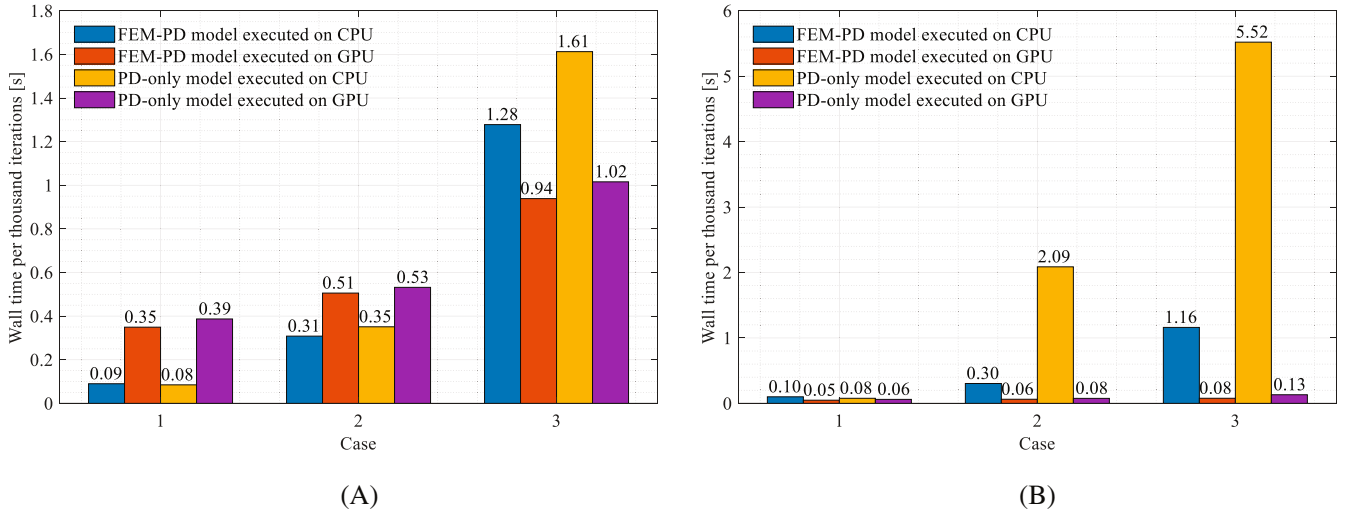


FIGURE 7 Wall times of the matrix and vector operations in the solver sections of the FEM-PD and PD-only models for the problem of pore pressure distribution in a single crack. (A) Wall times of the vector operations in the solver sections, (B) Wall times of the matrix operations in the solver sections.

computational workload gradually increases, the acceleration advantage of GPU over CPU becomes increasingly pronounced. To investigate which type of operation benefits more from GPU acceleration, we have separately recorded the wall times per thousand iterations required for vector operations and matrix operations during the solving process and plotted them in Figure 7A,B. It can be observed that, in comparison to vector operations, GPU acceleration has an extremely significant impact on matrix operations. In addition, Figures 6,7A,B also demonstrate that when running on GPU, the PD-only model can achieve computational efficiency similar to that of the FEM-PD model. However, it is important to note that, compared to the FEM-PD model, the PD-only model is not memory-friendly. When solving the same problem, it incurs a larger computational workload and requires more GPU memory, as described in Figure 8A–F.

4.2 | Example 2: One-dimensional dynamic consolidation problem

The second example pertains to a one-dimensional dynamic consolidation problem of a saturated porous column. This example was previously investigated by Schanz and Cheng[55] involving the transient wave propagation of displacement and pore pressure during the dynamic consolidation process. Analytical solutions for comparison were obtained using Dubner and Abate's method.[56]

Geometry and boundary conditions are shown in Figure 9. The top boundary is characterized by drainage conditions with a pore pressure of $p = 0$ and is subjected to a sudden surface pressure of $P_s = 1Pa$, whereas the remaining boundaries are both impermeable and constrained in their normal direction. The mechanical and fluid parameters employed in the simulations are listed in Table 3. Three cases with varying discretization parameters are conducted for comparative analysis, and their descriptions are provided in Table 4.

Figure 10A,B shows the variations of the vertical displacement on the top edge and the pore pressure at the bottom edge over time obtained by the FEM-PD models. The analytical solutions are also plotted for comparison. The numerical results obtained in cases 2 and 3 exhibit good agreement with the analytical solutions. All numerical results show a trend to converge towards the analytical solution as the mesh size decreases. Furthermore, the wall times per thousand iterations for the three cases executed on CPU and GPU have been recorded and are displayed in Figure 11, describing that, as the computational workload increases, the acceleration effect of GPU compared to CPU becomes increasingly pronounced, achieving a speed-up ratio up to 25 times.

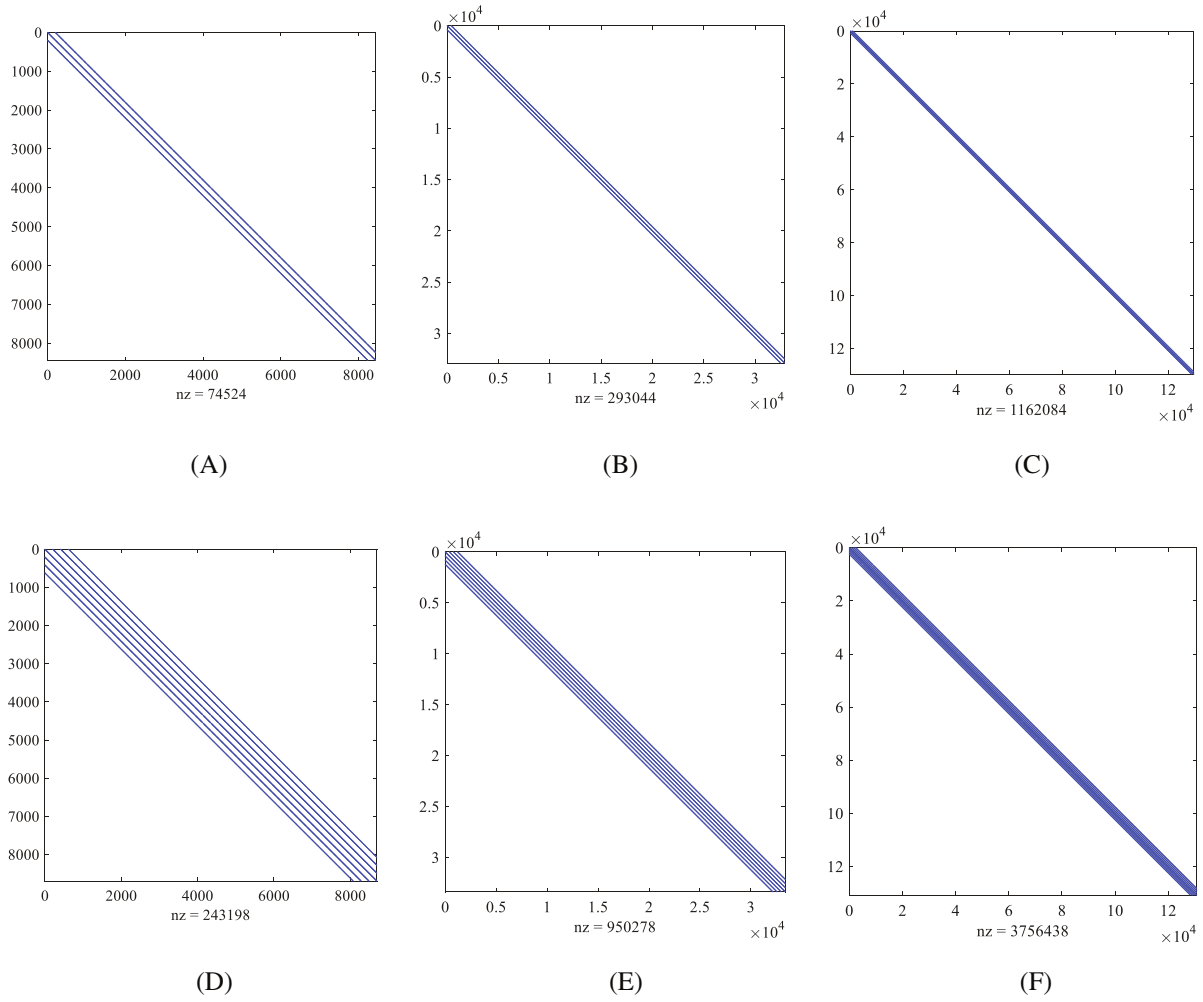


FIGURE 8 The shapes of the global permeability matrices obtained by FEM-PD and PD-only models. (A) FEM-PD model: case 1, (B) FEM-PD model: case 2, (C) FEM-PD model: case 3, (D) PD-only model: case 1, (E) PD-only model: case 2, (F) PD-only model: case 3.

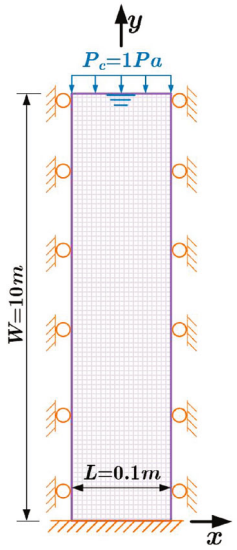


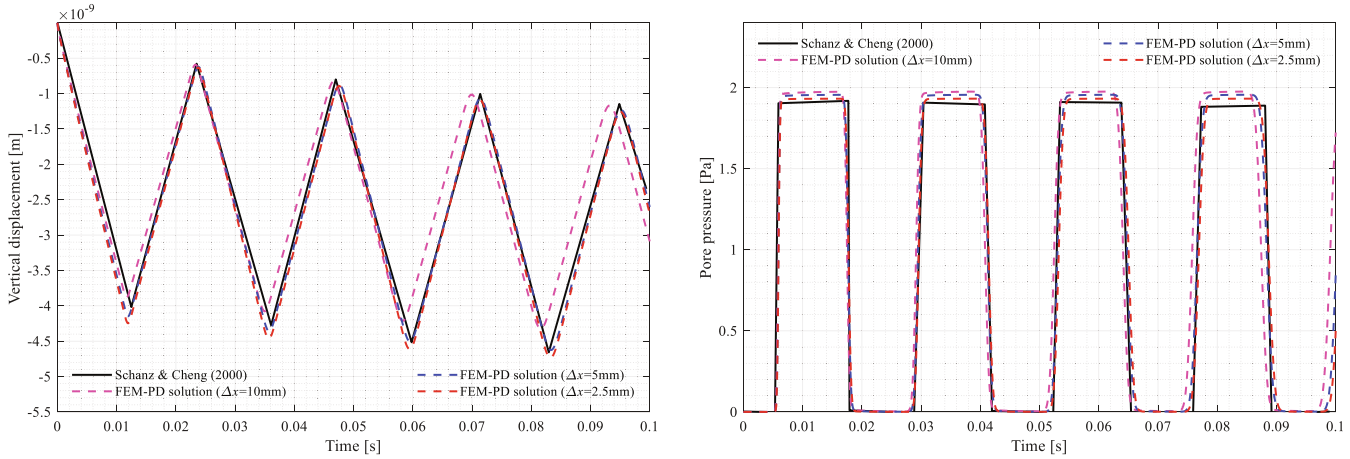
FIGURE 9 Geometry and boundary conditions of the one-dimensional dynamic consolidation problem.

TABLE 3 Mechanical and fluid parameters used in the Example 4.2.

$E\ [GPa]$	ν	$\rho_r\ [kg/m^3]$	$\rho_w\ [kg/m^3]$	α
0.254	0.3	2700	1000	0.981
n	$K_r\ [GPa]$	$K_w\ [GPa]$	$\mu_w\ [Pa \cdot s]$	$k_r\ [m^2]$
0.48	11	3.3	1×10^{-3}	3.55×10^{-12}

TABLE 4 Spatial and temporal discretization parameters used for Example 4.2.

Case	$\Delta x\ [mm]$	$\delta\ [mm]$	Number of nodes	$\Delta t\ [s]$	Number of iterations
1	10	30	11,012	1×10^{-6}	100,000
2	5	15	42,021	5×10^{-7}	200,000
3	2.5	7.5	164,042	2.5×10^{-7}	400,000



(A) (B)

FIGURE 10 Variations of (A) vertical displacement on the top edge and (B) pore pressure on the bottom edge versus time obtained by FEM-PD models in cases 1 ($\Delta x = 10$ mm), 2 ($\Delta x = 5$ mm) and 3 ($\Delta x = 2.5$ mm). (A) Vertical displacement on the top edge, (B) Pore pressure on the bottom edge.

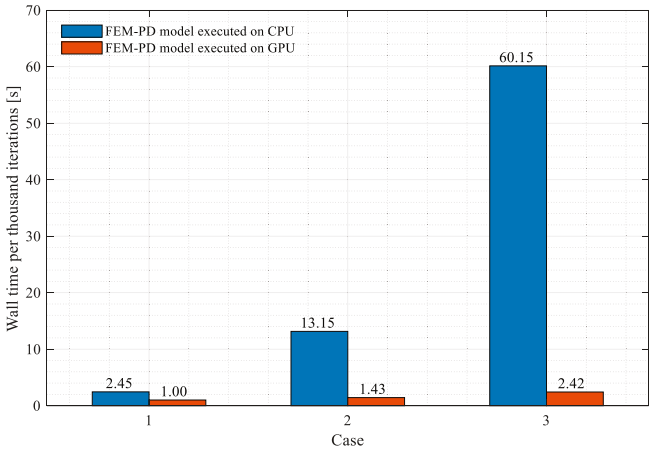


FIGURE 11 Comparison of wall times of the FEM-PD models executed on CPU and GPU for the dynamic consolidation problem.

4.3 | Example 3: Fluid-driven fracture propagation–reproducing the results of the KGD model

The third example is a problem of fluid-driven fracture propagation in a rectangular saturated porous domain, where the loading term is initially applied to the fluid mass balance equation, and its effects are subsequently transmitted to the momentum balance equation through coupling terms. Based on a set of ideal assumptions, Geertsma and De Klerk[57] derived the solution for the problem of fluid-driven fracture propagation under plane strain conditions, which is widely recognized as the KGD HF model. Consistent with the findings reported in Reference 58, a direct comparison between the KGD solution and the results of the developed FEM-PD model is not feasible. This is primarily because the KGD model operates independently of fracture energy, whereas the results from the FEM-PD model are notably influenced by the fracture energy value.[3]

To assess the accuracy of the proposed method for fluid-driven crack propagation, we conducted simulations similar to the case presented in Reference 12. The geometry and boundary conditions are illustrated in Figure 12. The mechanical and fluid parameters used in the simulations are listed in Table 5, with a constant permeability coefficient of $k_f = 2 \times 10^{-11}$ m^2 adopted for the fracture domain. The problem is addressed under plane strain conditions, and the fluid injection initiates from the center of the initial crack with a constant injection rate of $Q_0 = 10^{-4} m^3/s$. A total time duration of 10s is simulated.

Firstly, a series of numerical experiments is conducted to determine the appropriate fracture parameter, where varying values of fracture energy release rate G_c are used. The discretization parameters are taken as, the grid size: $\Delta x = 0.05m$, time step size: $\Delta t = 2.5 \times 10^{-5} s$. Figure 13A,C,E show the numerical results obtained with $G_c = 15, 25, 35$ and 45 J/m^2, respectively. Clearly, at the same injection rate, a smaller value of G_c will result in a faster crack propagation speed, thereby affecting the solutions of pore pressure and crack mouth opening displacement. The numerical results obtained with $G_c = 35J/m^2, 37.5J/m^2$ and $40J/m^2$ are compared in Figure 13B,D,F, demonstrating that $G_c = 37.5J/m^2$ appears to be a suitable choice, as it yields results closest to the analytical solutions. Figure 14A–C present the the change over time of damage levels, y-direction displacement, and pore pressure within the simulation domain in the case of $G_c = 37.5J/m^2$. They illustrate that, with the injection of fluid, the injected fluid flows along the cracks, leading to an increase of fluid pressure in the vicinity of the crack tip, consequently, the solid skeleton undergoes deformation, and the crack propagates. These

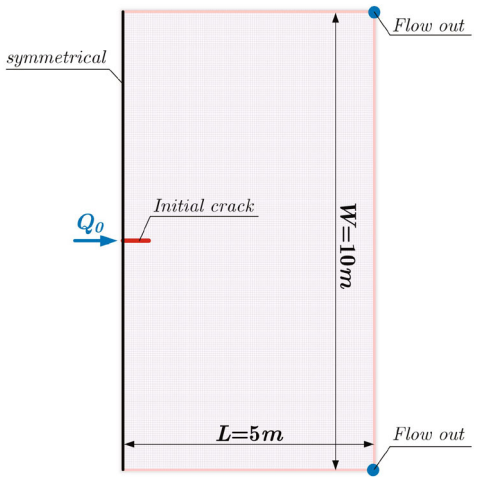
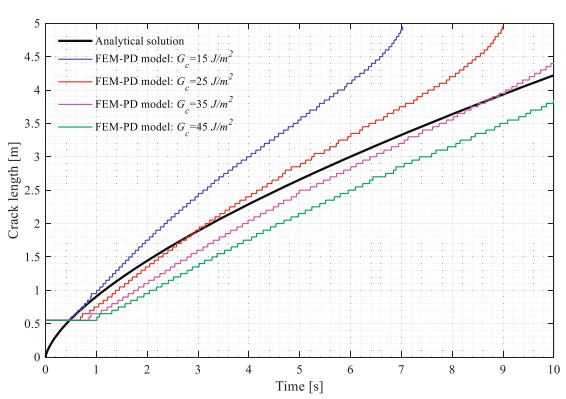


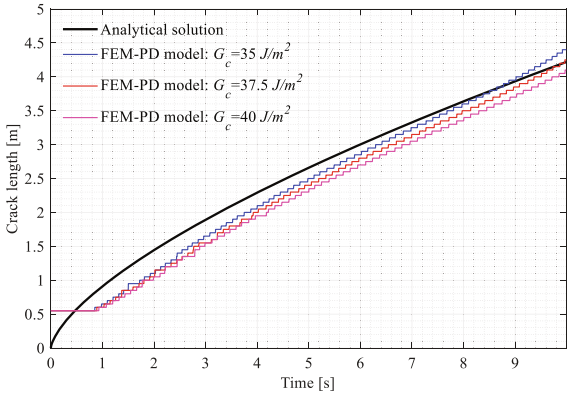
FIGURE 12 Geometry and boundary conditions of the two-dimensional rectangular porous domain.

TABLE 5 Mechanical and fluid parameters used in the Example 4.3.

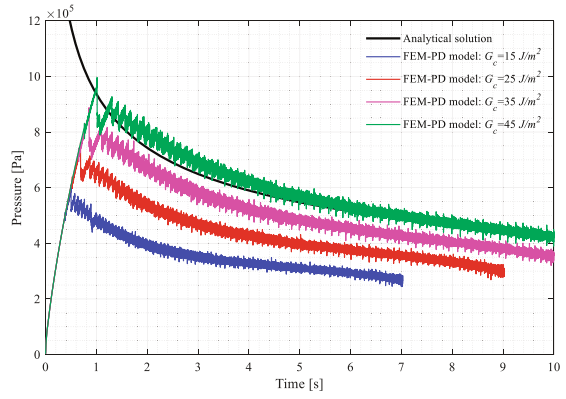
E [GPa]	ν	ρ_r [kg/m^3]	ρ_w [kg/m^3]	α
14.4	0.2	2500	1000	0.7883
n	K_w [GPa]	μ_w [$Pa \cdot s$]	k_r [m^2]	k_f [m^2]
0.19	3	1×10^{-3}	2×10^{-14}	2×10^{-11}



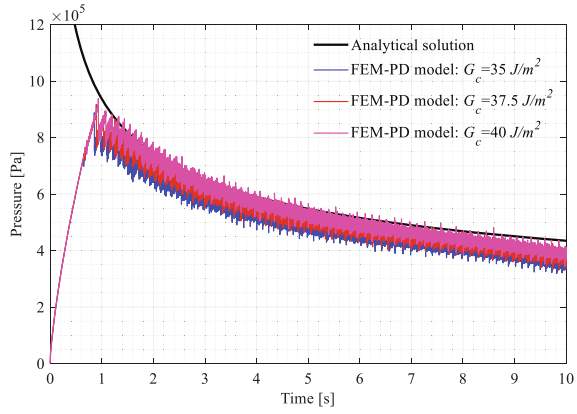
(A) Variations of crack length versus time



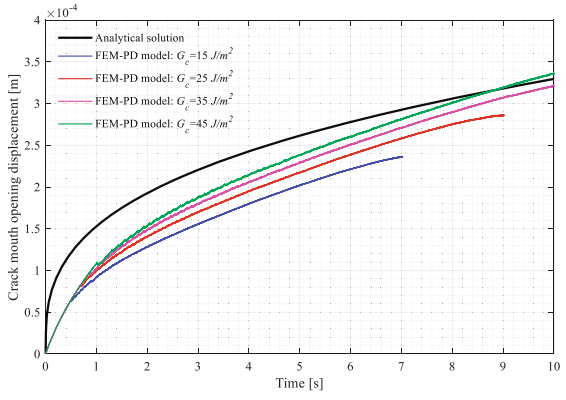
(B) Variations of crack length versus time



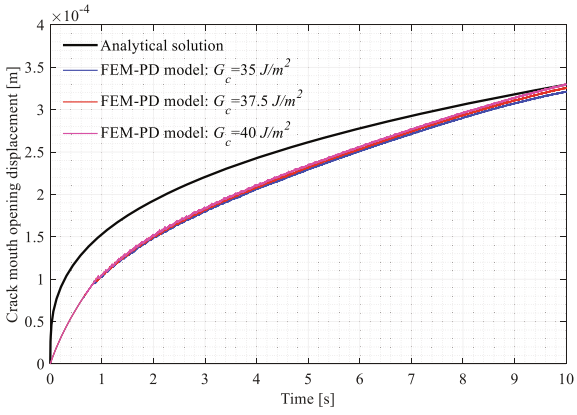
(C)



(D)



(E)



(F)

FIGURE 13 Comparison between the analytical solutions from Reference 57 and numerical results obtained by the FEM-PD model with different fracture energy release rates (G_c = 15, 25, 35, 37.5, 40and45 J/m^2). (A) Variations of crack length versus time, (B) Variations of crack length versus time, (C) Variations of pore pressure at injection point versus time, (D) Variations of pore pressure at injection point versus time, (E) Variations of crack mouth opening displacement versus time, (F) Variations of crack mouth opening displacement versus time.

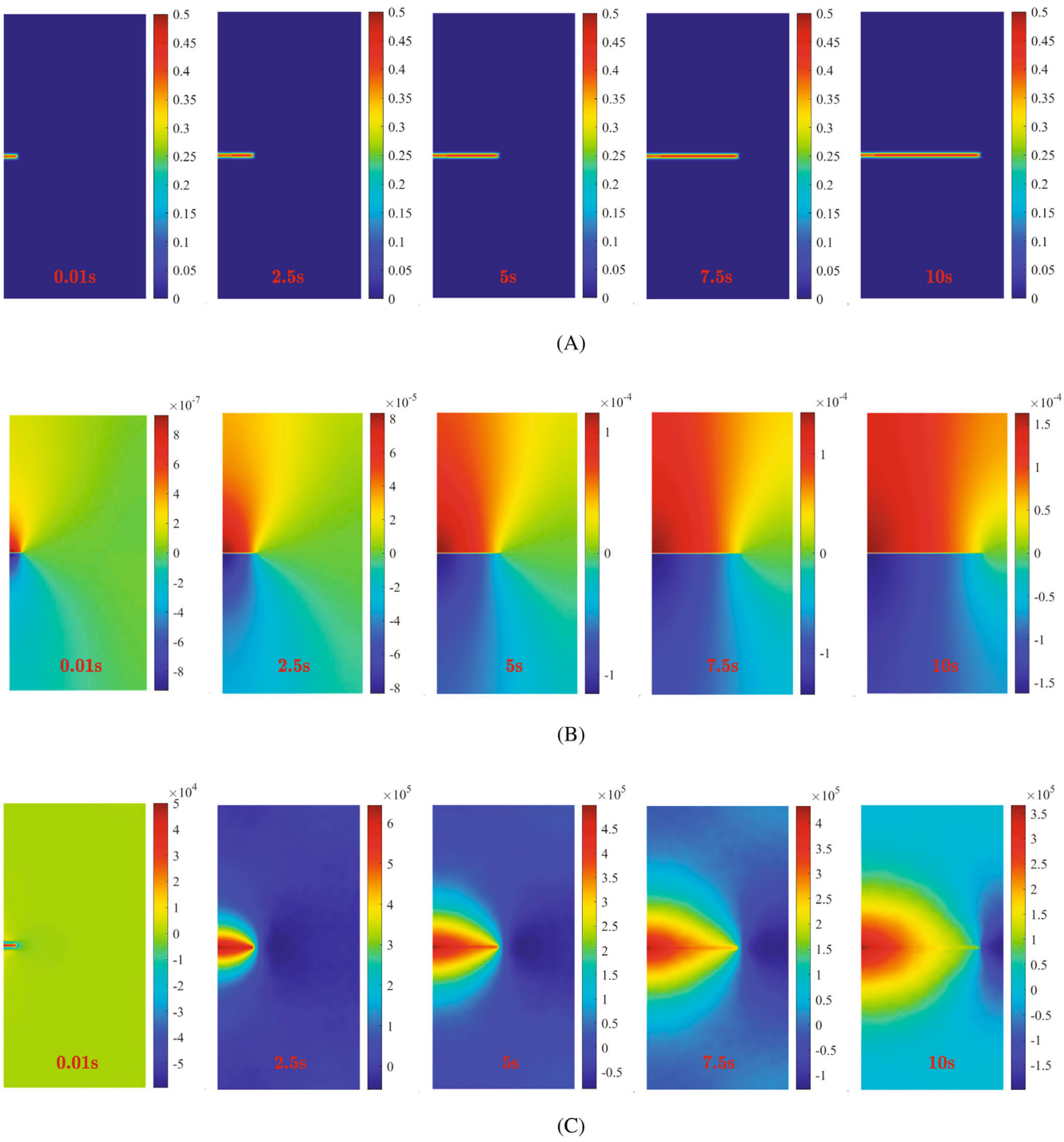


FIGURE 14 Contours of (A) damage levels, (B) y-direction displacement component and (C) pore pressure obtained by the FEM-PD model with $G_c = 37.5$ J/m^2 at several selected time instants. (A) Contours of damage levels, (B) Contours of y-direction displacement component (unit: m), (C) Contours of pore pressure (unit: Pa).

observations suggest that the present method effectively describes the phenomena of spontaneous crack propagation and fluid flow within fracturing saturated porous media in a straightforward and realistic manner.

Subsequently, three cases with different discretization parameters are examined. Detailed information for each case is listed in Table 6. The simulated variations of crack length, pore pressure at injection point and crack mouth opening displacement versus time are plotted in Figure 15A–C. Figure 16A,B present the corresponding contours of damage levels and pore pressure obtained by the FEM-PD model at the end of simulations. These results indicate that, in the case with a smaller value of Δx, a slower crack propagation speed is observed. Consequently, this leads to higher pore pressure within

TABLE 6 Spatial and temporal discretization parameters used for Example 4.3.

Case	Δx [m]	δ [m]	Number of nodes	Δt [s]	Number of iterations
1	0.1	0.3	5304	5×10^{-5}	200,000
2	0.05	0.15	20,604	2.5×10^{-5}	400,000
3	0.025	0.075	81,204	1.25×10^{-5}	800,000

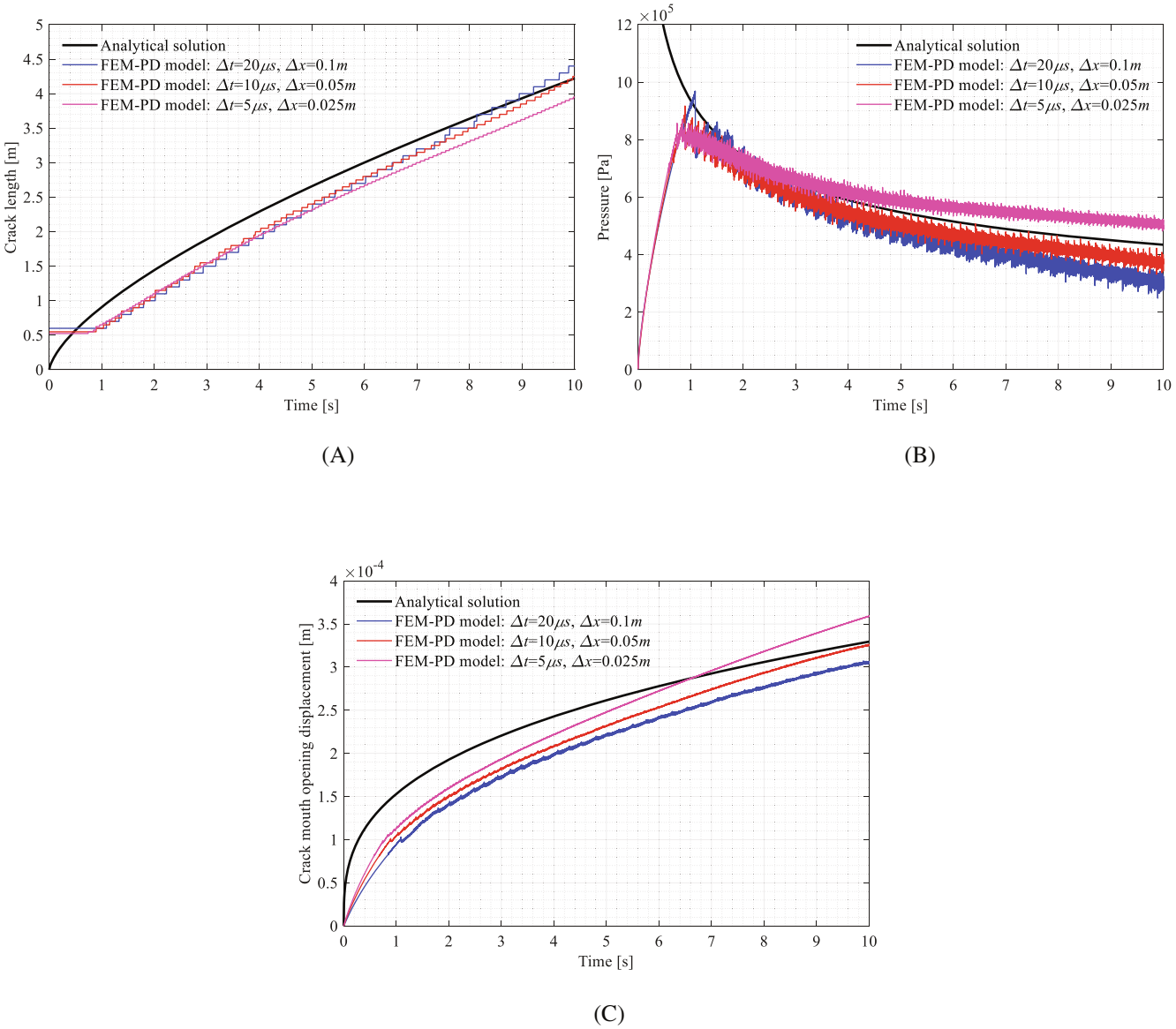


FIGURE 15 Comparison between the analytical solutions from Reference 57 and numerical results obtained by the FEM-PD model with $G_c = 37.5$ J/m^2 and different discretization parameters. (A) Variations of crack length versus time, (B) Variations of pore pressure at injection point versus time, (C) Variations of crack mouth opening displacement versus time.

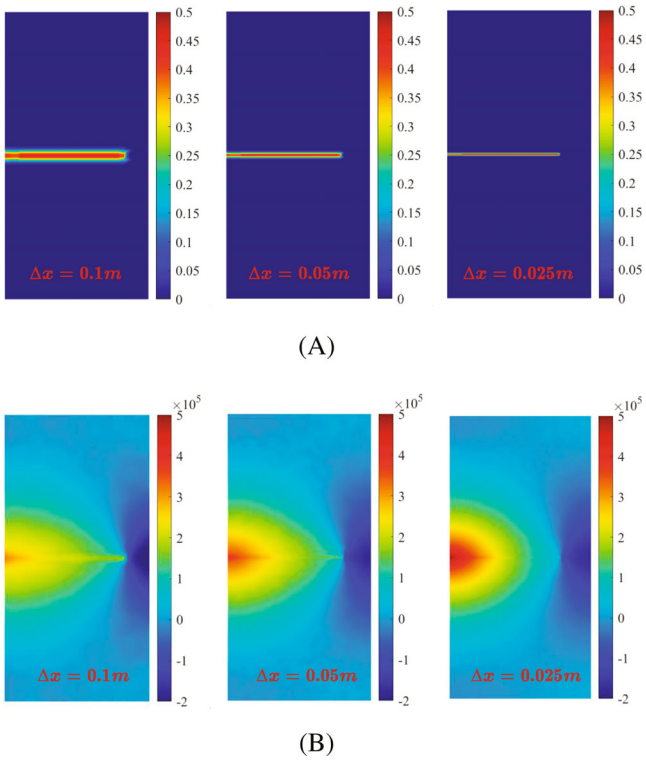


FIGURE 16 Contours of (A) damage levels and (B) pore pressure obtained by the FEM-PD model with $G_c = 37.5$ J/m^2 and different discretization parameters. (A) Contours of damage levels, (B) Contours of pore pressure (unit: Pa).

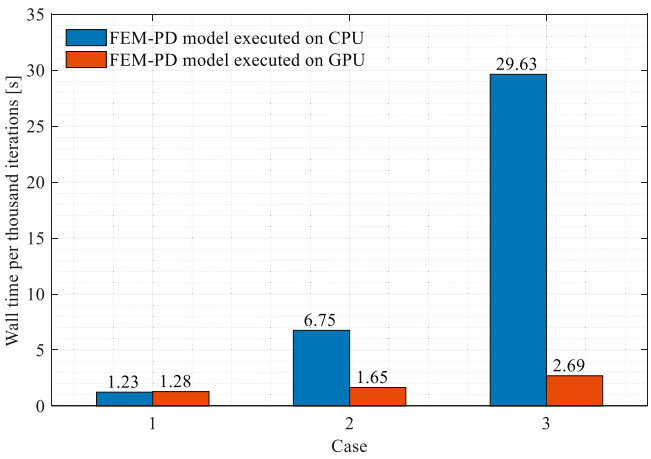


FIGURE 17 Comparison of wall times of the FEM-PD models executed on CPU and GPU for the two-dimensional fluid-driven fracture propagation problem.

the crack domain and a greater crack mouth opening displacement. Overall, it is evident that all cases yield satisfactory numerical results, displaying approximate agreement with the analytical solutions. Figure 17 describes the wall times per thousand iterations of the FEM-PD models executed on CPU and GPU. In line with the findings in the previous subsections, as the computational load increases, the acceleration effect of the GPU comparing to CPU gradually becomes more significant.

Remark 3. The influence functions in the state-based peridynamic model are used to modulate the magnitude of a non-local interaction within the neighbourhood. Some different forms of influence functions have been

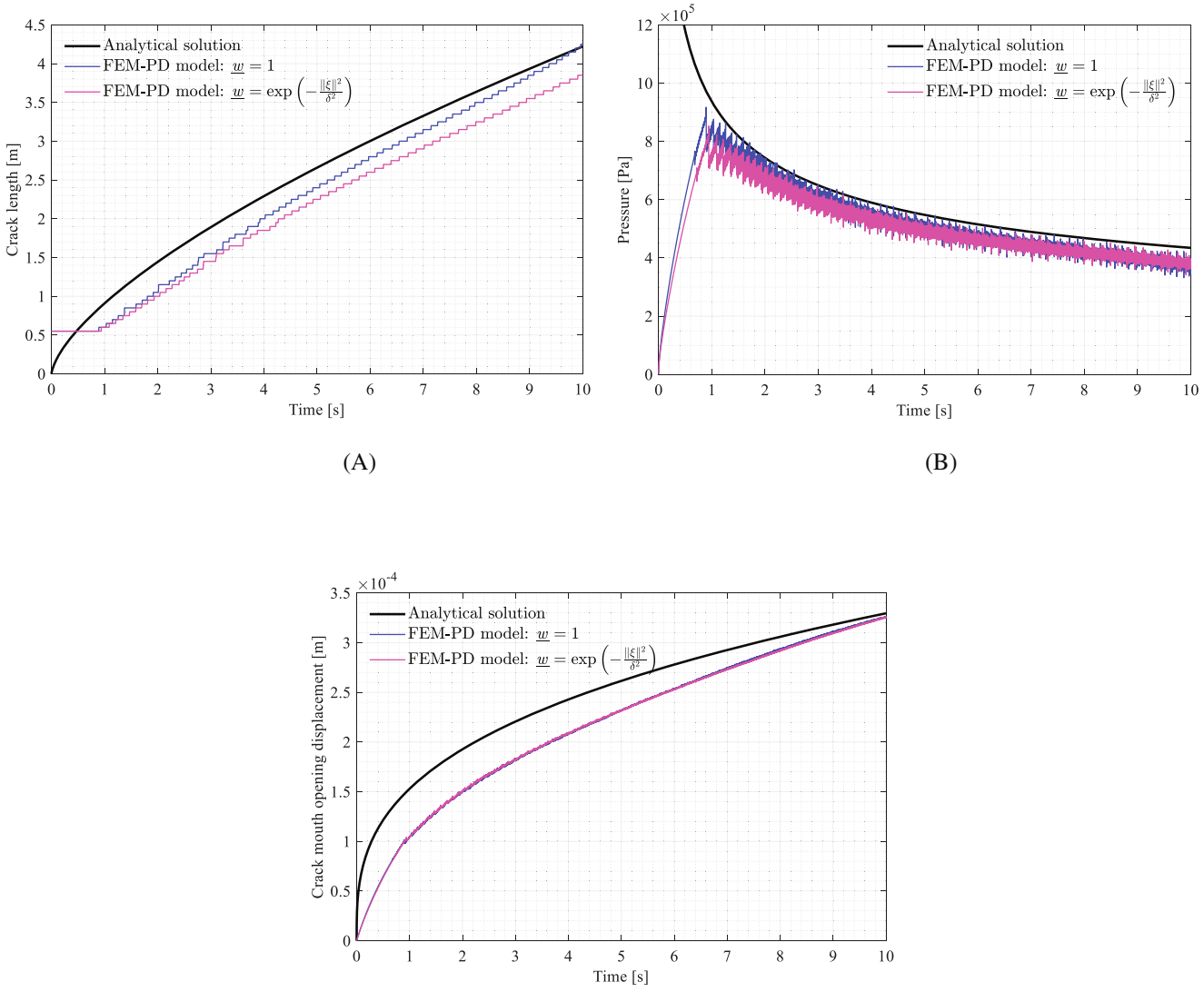


FIGURE 18 Comparison between the analytical solutions from Reference 57 and numerical results obtained by the 2D FEM-PD model with $G_c = 37.5J/m^2$, $\Delta x = 0.05m$, and different influence functions $\left(\underline{w}=1 \text{ and } \underline{w}=\exp\left(-\frac{\|\xi\|^2}{\delta^2}\right)\right)$. (A) Variations of crack length versus time, (B) Variations of pore pressure at injection point versus time, (C) Variations of crack mouth opening displacement versus time.

introduced in References 21,59,60. In the existing literature,[61] the influence function of $\underline{w}=\exp\left(-\frac{\|\xi\|^2}{\delta^2}\right)$ is the most commonly used for the OSB-PD models. Therefore, we simulated the KGD model using $\underline{w}=\exp\left(-\frac{\|\xi\|^2}{\delta^2}\right)$ for the purpose of comparison. In addition, when $\underline{w}=\exp\left(-\frac{\|\xi\|^2}{\delta^2}\right)$ is adopted, the critical stretch value can be derived in the plane strain conditions as:

$$s_c = \sqrt{\frac{G_c}{\left(4\kappa - \frac{28}{9}\mu\right)\Lambda + 8\mu\Lambda'}} \tag{66}$$

where $\Lambda = \frac{813\delta}{1000\pi^2}$ and $\Lambda' = \frac{379\delta}{500\pi}$. The same discretization parameters as those of case 2 are employed, with a fracture energy release rate of $G_c = 37.5J/m^2$. Figure 18A–C depict comparisons between analytical solutions and obtained numerical results, with the results for $\underline{w}=1$ also provided for reference. These results

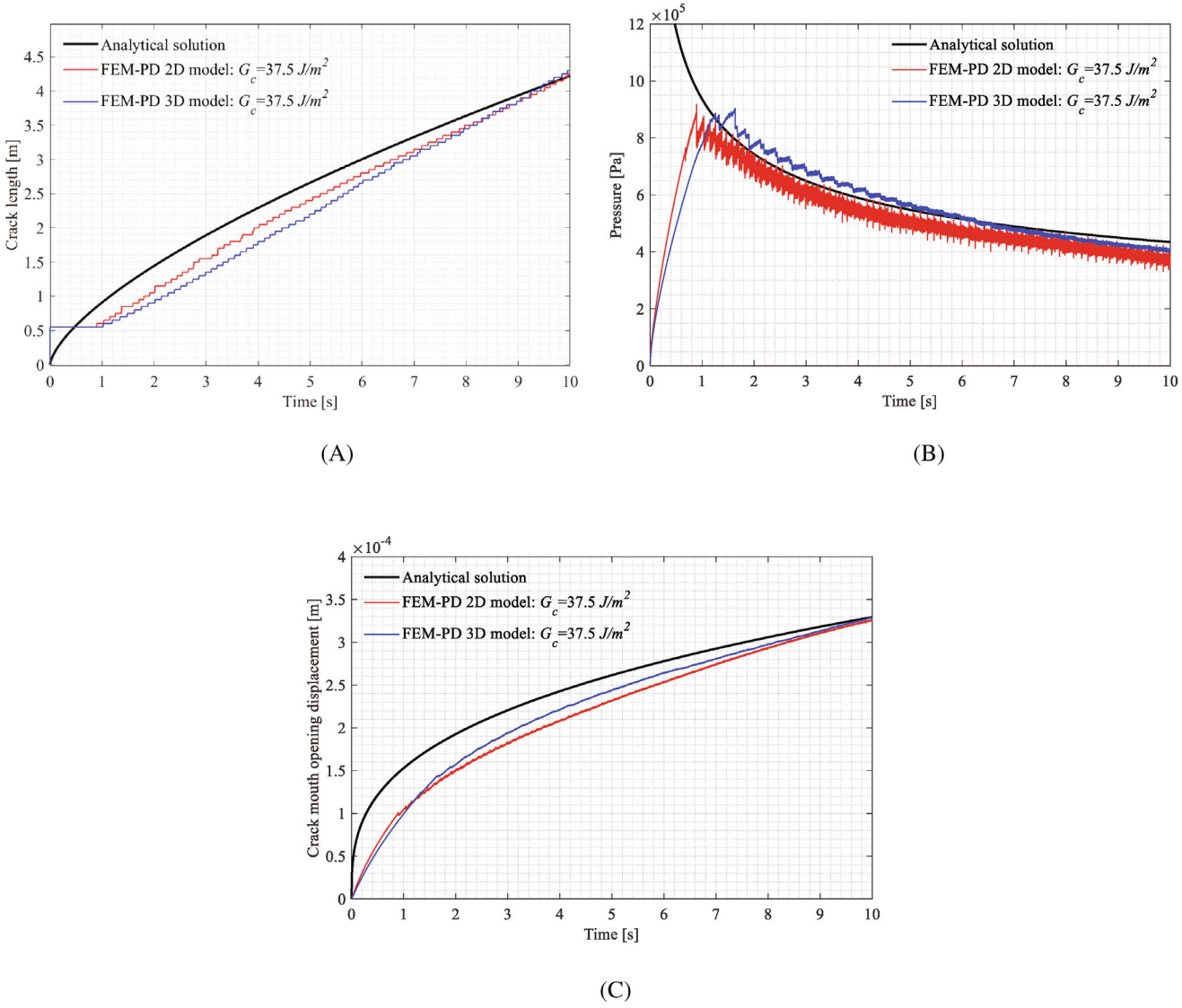


FIGURE 19 Comparison between the analytical solutions from Reference 57 and numerical results obtained by the 2D and 3D FEM-PD models with $G_c = 37.5J/m^2$ and $\Delta x = 0.05m$. (A) Variations of crack length versus time, (B) Variations of pore pressure at injection point versus time, (C) Variations of crack mouth opening displacement versus time.

demonstrate that utilizing different influence functions yields some variations in outcomes. Nonetheless, all numerical results closely approximate the analytical solutions, suggesting that the choice of influence function does not significantly impact result reliability in the studied case. Conversely, as per the findings in Reference 59, the form of the influence function notably influences the dispersion behavior of the PD model. Hence, additional investigation is warranted to explore the impact of the influence function on the FEM-PD model when addressing dynamic problems characterized by pronounced fluctuations.

Finally, simulating the KGD model in three dimensions serves to further validate the accuracy and reliability of the proposed method in this paper. In addition to the parameters described in Table 5, the fracture energy release rate is selected as $G_c = 37.5J/m^2$. The discretization parameters for the PD domain are horizon $\delta = 0.15m$, m-ratio $m = 3$ and consequently grid spacing $\Delta x = \delta/m = 0.05m$. Given the model thickness as $1m$, a 3D discrete model with 453,288 nodes is obtained. A time step size of $\Delta t = 2.5 \times 10^{-5}s$ is adopted for a simulation duration of $10s$. Figure 19A–C

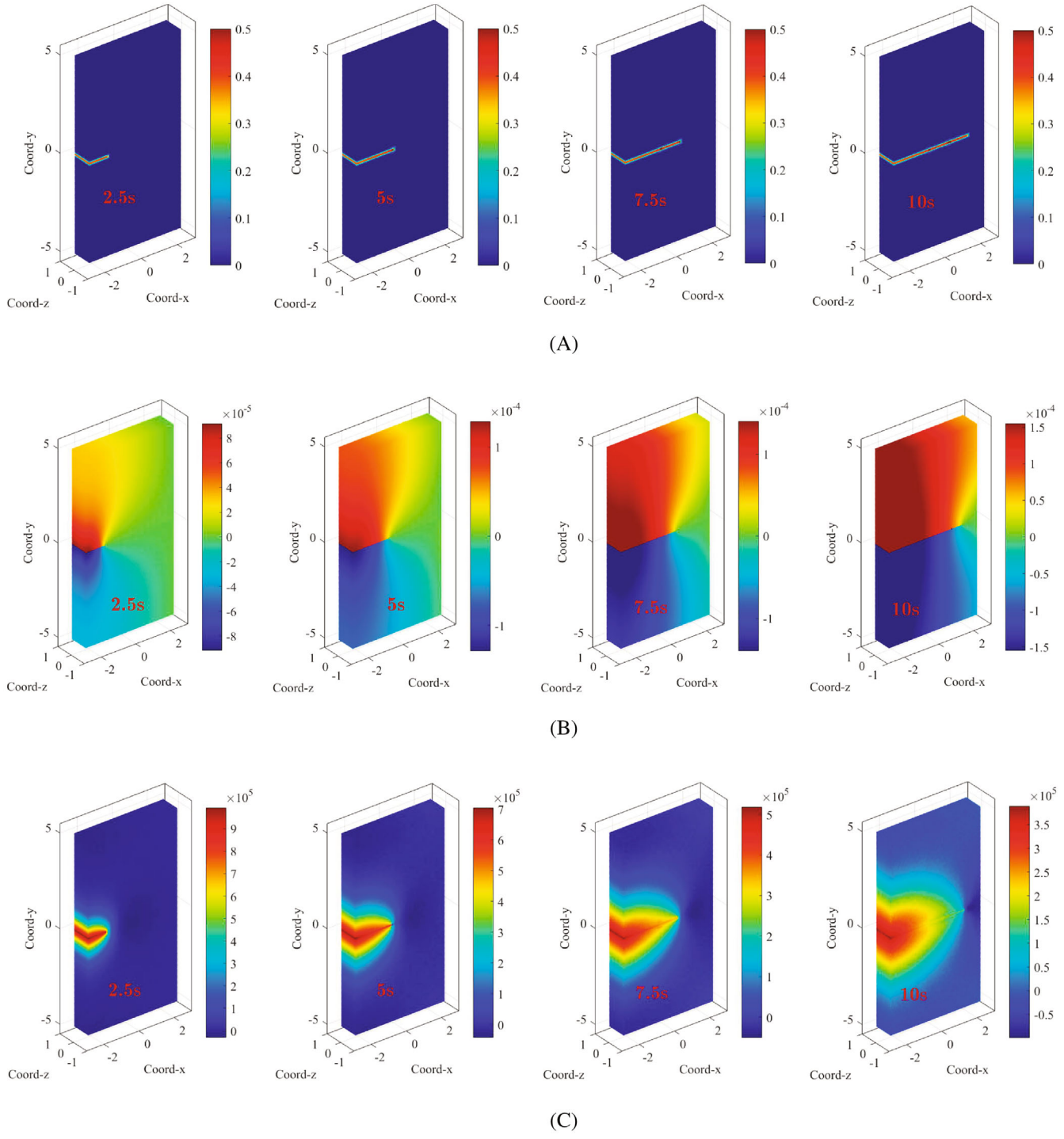


FIGURE 20 Contours of (A) damage levels, (B) y-direction displacement component and (C) pore pressure obtained by the 3D FEM-PD model with $G_c = 37.5J/m^2$ at several selected time instants. (A) Contours of damage levels, (B) Contours of y-direction displacement component (unit: m), Contours of pore pressure (unit: Pa).

present comparisons between analytical solutions and numerical results obtained by the 2D and 3D FEM-PD models. Correspondingly, Figure 20A–C display the contours of damage levels, y-direction displacement components, and pore pressure at selected time instants of 2.5, 5, 7.5, and 10 s. Although some disparity exists between the showcased

numerical results and the analytical solutions, overall comparability remains, underscoring the accuracy of the proposed approach.

4.4 | Example 4: Fluid-driven fracture propagation in a three-dimensional porous domain

The final example involves a fluid-driven fracture propagation problem in 3D, which is addressed to showcase the capability of the proposed approach in simulating 3D HF problems and to further emphasize the efficiency of the developed software.

The geometry and boundary conditions are illustrated in Figure 21. The mechanical and fluid parameters remain consistent with those outlined in Table 5. The fracture energy release rate is taken as $G_c = 100J/m^2$. The discretization parameters are taken as, the grid size: $\Delta x = 0.05m$ and the time step size: $\Delta t = 2 \times 10^{-8}s$, resulting in a discrete model with 551,368 nodes.

Three cases with different fluid injection rates are considered. The fluid injection rates of $Q_0 = 0.05m^3/s$, $0.1m^3/s$ and $0.2m^3/s$ are applied in cases 1, 2 and 3, respectively. In the simulations, the pore pressure values at the injection point are recorded and plotted versus time in Figures 22A, 23A and 24A. Figures 22B,C, 23B,C and 24B,C show crack patterns and pore pressure distributions at several selected time instants after the initiation of hydraulic fracture propagation. These results reveal that, when the injection rate is sufficiently high, the bifurcation of hydraulic crack can be observed, demonstrating non-planar crack propagation behavior in three-dimensional space, which is a typical dynamic phenomenon in fracturing porous media. Moreover, a larger injection rate will cause crack branching behavior to occur earlier and result in wider angles between branching cracks, being consistent with the trends described in References 13,62. Figure 25 describes the comparisons of wall times per thousand iterations of the FEM-PD model executed on CPU and GPU in the simulations of the 3D HF propagation problem. The wall time in case 3 is greater than those in cases 1 and 2. This is primarily due to the generation of a larger number of fractures in case 3, which results in a higher frequency of matrix update operations. Nevertheless, across all the devised cases, the acceleration effect of the GPU remains highly significant, with a speed-up ratio of approximately 30 times compared to the CPU.

Remark 4. In the proposed approach, when utilizing GPUs to accelerate the calculation, all the system matrices are stored in the GPU memory, prioritizing solving efficiency over memory conservation. For instance, in Example 4.4, the discrete model encompasses 551,368 nodes, resulting in over 2 million degrees of freedom. Executing this simulation consumed more than 22 GB of the 24 GB of memory available on the graphics card. Consequently, multi-GPU computing platforms emerge as a viable solution, distributing larger-scale computations across multiple GPUs through matrix partitioning, thus enabling the resolution of even more extensive problems.

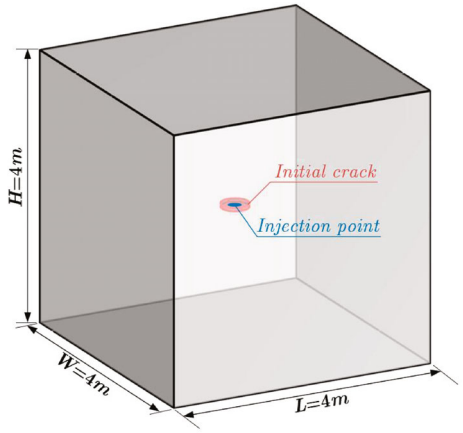
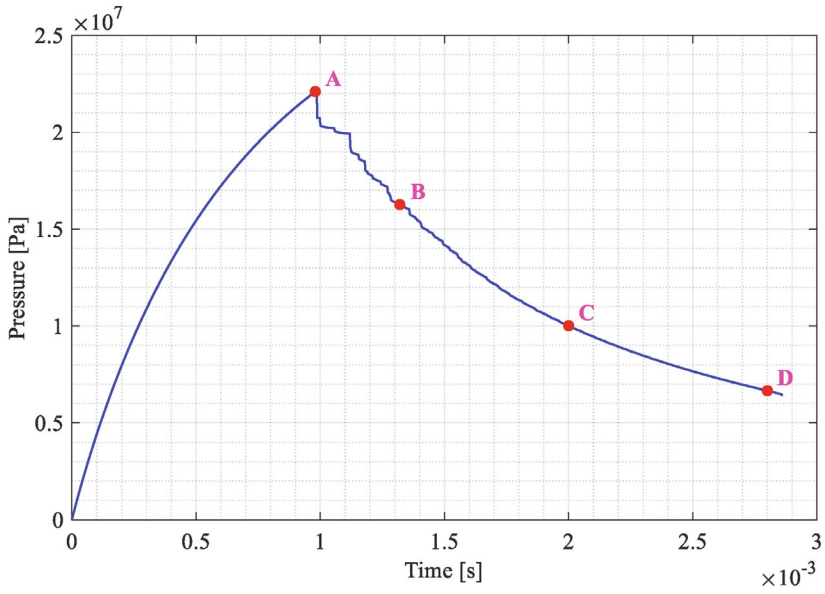
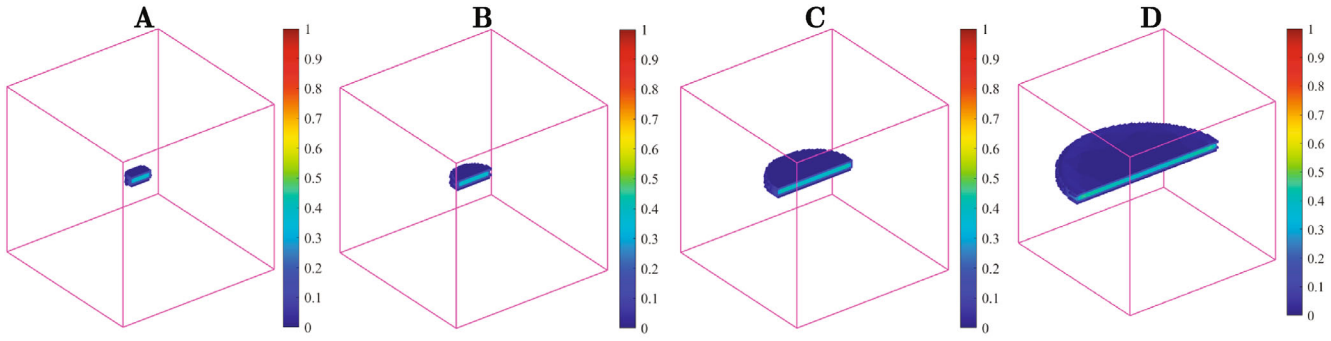


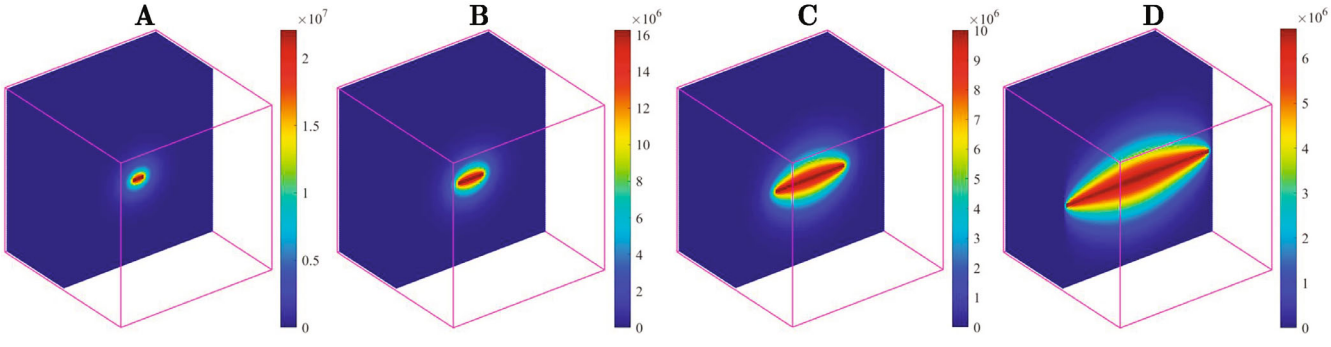
FIGURE 21 Geometry and boundary conditions of the three-dimensional cubical porous domain.



(A)

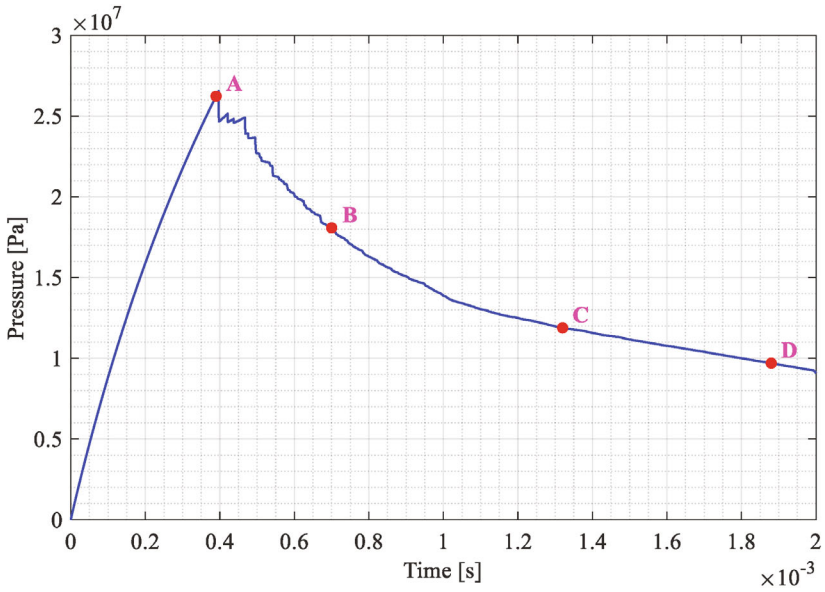


(B)

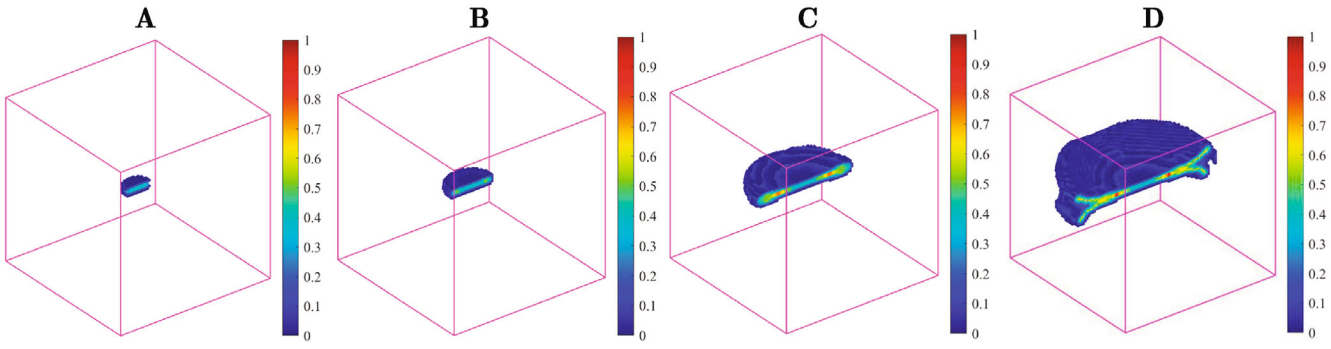


(C)

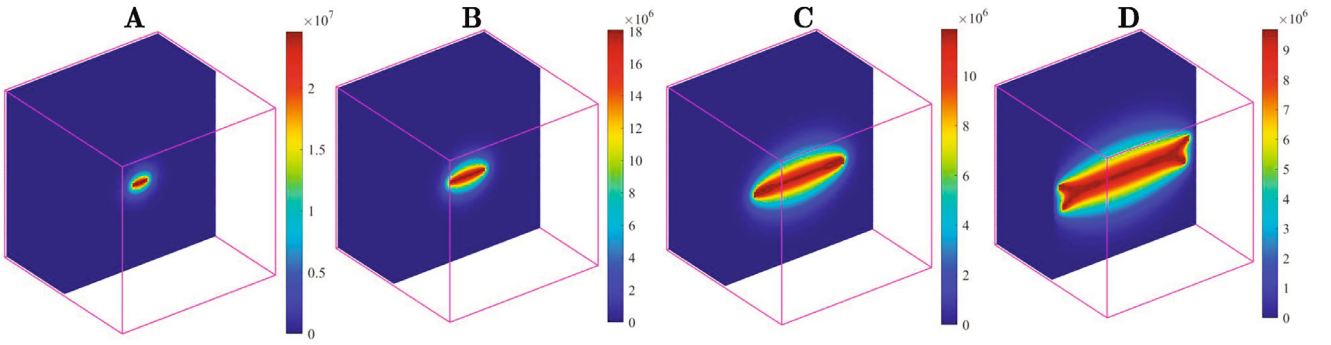
FIGURE 22 Numerical results obtained by the FEM-PD model with an injection rate of $Q_0 = 0.05m^3/s$. (A) Variation of pore pressure versus time at injection point, (B) Damage levels ($\phi > 0$, half model) at the selected time instants, (C) Pore pressure distribution (half model) at the selected time instants.



(A)

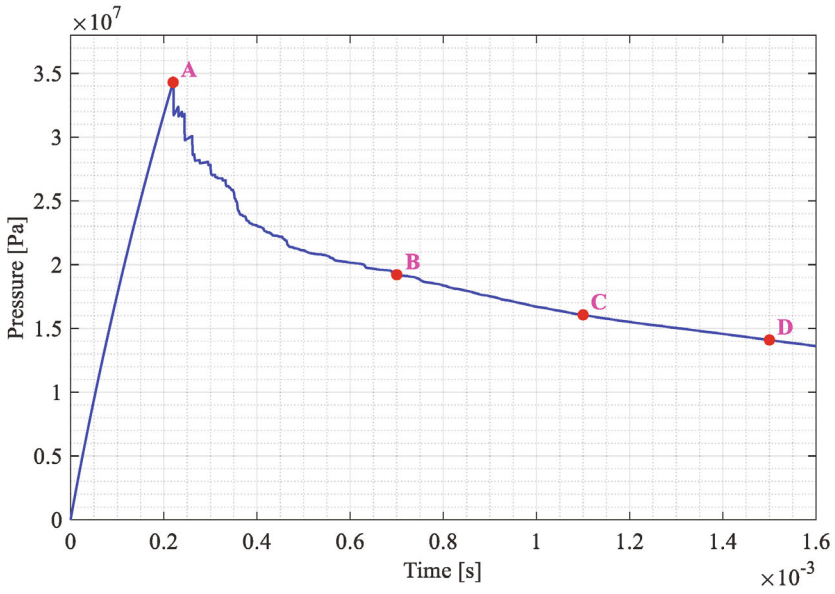


(B)

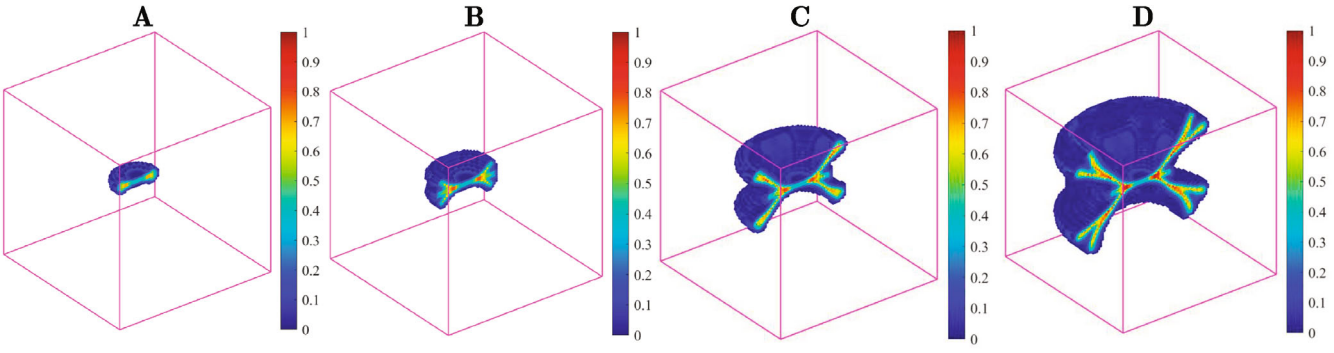


(C)

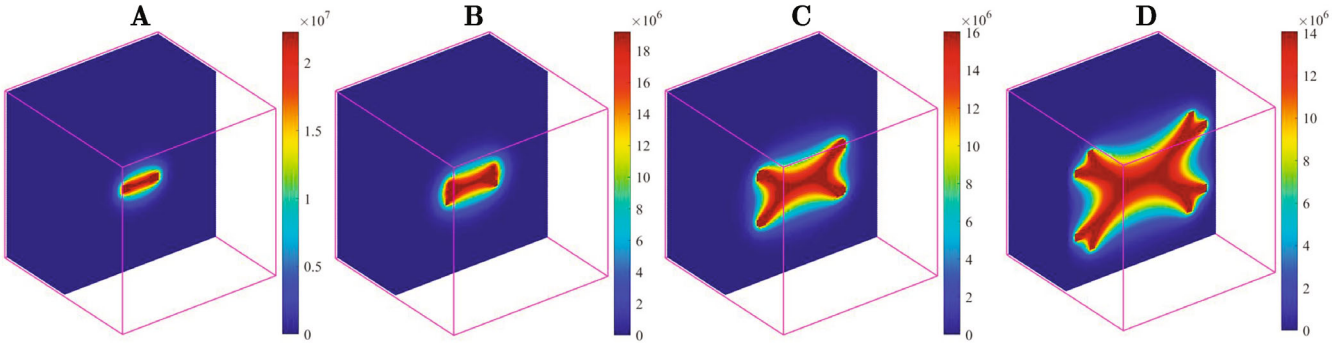
FIGURE 23 Numerical results obtained by the FEM-PD model with an injection rate of $Q_0 = 0.1m^3/s$. (A) Variation of pore pressure versus time at injection point, (B) Damage levels ($\phi > 0$, half model) at the selected time instants, (C) Pore pressure distribution (half model) at the selected time instants.



(A)



(B)



(C)

FIGURE 24 Numerical results obtained by the FEM-PD model with an injection rate of $Q_0 = 0.2m^3/s$. (A) Variation of pore pressure versus time at injection point, (B) Damage levels ($\phi > 0$, half model) at the selected time instants, (C) Pore pressure distribution (half model) at the selected time instants.

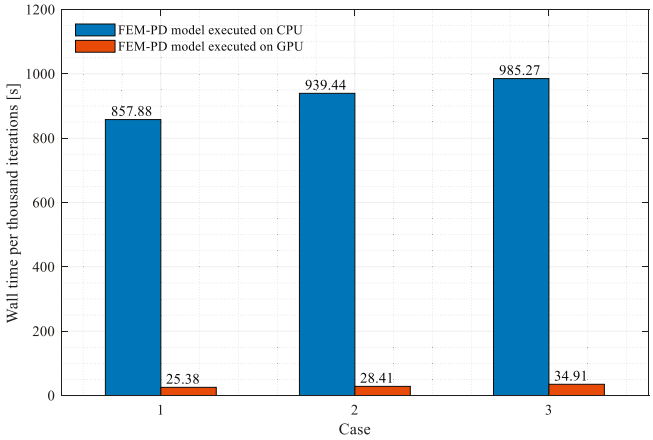


FIGURE 25 Comparison of wall times of the FEM-PD models executed on CPU and GPU for the three-dimensional fluid-driven fracture propagation problem.

5 | CONCLUSIONS

This paper presented a fast explicit solution scheme for the hybrid FEM-PD models designed to simulate the dynamic behavior of saturated porous media under the influence of hydro-mechanical coupling. The matrix-based implementation of this approach was described in detail. The paper also discussed the numerical stability conditions of the adopted explicit method in the undamped system of the FEM-PD model and offered a method for determining the critical time increment. An in-house software was developed in MATLAB, featuring both CPU and GPU versions. Several typical numerical examples were included for validation and to demonstrate the capabilities and effectiveness of the proposed approach.

Firstly, a problem related to pore pressure distribution in a single crack was simulated to validate the capability and effectiveness of the proposed approach in simulating fluid flow in fractured saturated porous media. The accuracy of the proposed method was evidenced by a comparison of FEM-PD solutions with those of PD-only models and analytical solutions. Subsequently, a one-dimensional dynamic consolidation problem was addressed, with the agreement between the numerical results and analytical solutions demonstrating the model's capability to describe the interaction between the fields of deformation and pore pressure in saturated porous media. Finally, fluid-driven hydraulic fracture propagation problems in both 2D and 3D cases were simulated. Various dynamic phenomena, such as stepwise crack advancement, crack branching in a 3D context, and pressure oscillations, were observed, comprehensively validating the effectiveness of the proposed methods in simulating the dynamics of deformation and fracture in saturated porous media under the influence of hydro-mechanical coupling.

Furthermore, the performance of the matrix-based scheme, executed on both CPU and GPU, was also evaluated in terms of wall times per thousand iterations. The findings indicated that, with increasing computational workload, the acceleration effect of the GPU in comparison to the CPU gradually becomes more pronounced.

In summary, the implementation scheme based on the vector and matrix operations in MATLAB offers a significant improvement in efficiency and simplifies the GPU parallel programming of FEM-PD models for hydro-mechanical coupled problems in saturated porous media. This approach minimizes programming complexity, maximizes performance, and opens up possibilities for applying FEM-PD models to solve engineering problems with a substantial number of degrees of freedom. It's worth noting that the proposed matrix-based scheme can also be implemented in other programming languages with efficient libraries for matrix operations, such as Python and Julia, while maintaining high efficiency.[33]

ACKNOWLEDGEMENTS

This research is financially supported by the National Natural Science Foundation of China (Grant no. 42207226); Natural Science Foundation of Sichuan Province (Grant nos. 2023NSFSC0808 and 2022NSFSC1129); State Key Laboratory of Geohazard Prevention and Geoenvironment Protection Independent Research Project (Grant nos. SKLGP2022Z018 and

SKLGP2021Z026). The authors would like to acknowledge the support they received from MIUR, Italy under the research project PRIN-PNRR P2022HLHHB and from University of Padova under the research projects BIRD2023 NR.237212/23 and NR.232492/23. B.A. Schrefler gratefully acknowledges the support of the Technische Universität München–Institute for Advanced Study.

CONFLICT OF INTEREST STATEMENT

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are available from the corresponding author upon reasonable request.

ORCID

Tao Ni https://orcid.org/0000-0002-8454-8246
Bernhard A. Schrefler https://orcid.org/0000-0002-8086-1147

REFERENCES

1. Lee S, Wheeler MF, Wick T. Pressure and fluid-driven fracture propagation in porous media using an adaptive finite element phase field model. *Comput Methods Appl Mech Eng*. 2016;305:111-132.
2. Zhou S, Zhuang X, Rabczuk T. Phase-field modeling of fluid-driven dynamic cracking in porous media. *Comput Methods Appl Mech Eng*. 2019;350:169-198.
3. Ni T, Pesavento F, Zaccariotto M, Galvanetto U, Zhu Q-Z, Schrefler BA. Hybrid fem and peridynamic simulation of hydraulic fracture propagation in saturated porous media. *Comput Methods Appl Mech Eng*. 2020;366:113101.
4. Joseph JB, Ribeiro P, Marcelo V, Guimarães LJ, Junior Chaves CV, Teixeira JDC. Acceleration strategies for tridimensional coupled hydromechanical problems based on cpu and gpu programming in matlab. *An Acad Bras Cienc*. 2022;94:e20211024.
5. Cui X, Wong LNY. A 3d fully thermo–hydro–mechanical coupling model for saturated poroelastic medium. *Comput Methods Appl Mech Eng*. 2022;394:114939.
6. Pandey S, Vishal V, Chaudhuri A. Geothermal reservoir modeling in a coupled thermo-hydro-mechanical-chemical approach: a review. *Earth-Sci Rev*. 2018;185:1157-1169.
7. Ni T, Fan X, Zhang J, Zaccariotto M, Galvanetto U, Schrefler BA. A peridynamic-enhanced finite element method for thermo–hydro–mechanical coupled problems in saturated porous media involving cracks. *Comput Methods Appl Mech Eng*. 2023;417:116376.
8. Chen B, Sun Y, Barboza BR, Barron AR, Li C. Phase-field simulation of hydraulic fracturing with a revised fluid model and hybrid solver. *Eng Fract Mech*. 2020;229:106928.
9. Chen B, Barboza BR, Sun Y, et al. A review of hydraulic fracturing simulation. *Arch Comput Methods Eng*. 2021;29:1-58.
10. Ni T, Pesavento F, Zaccariotto M, Galvanetto U, Schrefler BA. Numerical simulation of forerunning fracture in saturated porous solids with hybrid fem/peridynamic model. *Comput Geotech*. 2021;133:104024.
11. Sun W, Fish J. Coupling of non-ordinary state-based peridynamics and finite element method for fracture propagation in saturated porous media. *Int J Numer Analyt Methods Geomech*. 2021;45(9):1260-1281.
12. Sun Y, Chen B, Edwards MG, Li C. Investigation of hydraulic fracture branching in porous media with a hybrid finite element and peridynamic approach. *Theor Appl Fract Mech*. 2021;116:103133.
13. Sun W, Fish J, Lin P. Numerical simulation of fluid-driven fracturing in orthotropic poroelastic media based on a peridynamics-finite element coupling approach. *Int J Rock Mech min Sci*. 2022;158:105199.
14. Silling SA. Reformulation of elasticity theory for discontinuities and long-range forces. *J Mech Phys Solids*. 2000;48(1):175-209.
15. Silling S, Epton M, Weckner O, Xu J, Askari E. Peridynamic states and constitutive modeling. *J Elasticity*. 2007;88:151-184.
16. Galvanetto U, Mudric T, Shojaei A, Zaccariotto M. An effective way to couple fem meshes and peridynamics grids for the solution of static equilibrium problems. *Mech Res Commun*. 2016;76:41-47.
17. Han F, Lubineau G, Azdoud Y, Askari A. A morphing approach to couple state-based peridynamics with classical continuum mechanics. *Comput Methods Appl Mech Eng*. 2016;301:336-358.
18. Zaccariotto M, Tomasi D, Galvanetto U. An enhanced coupling of pd grids to fe meshes. *Mech Res Commun*. 2017;84:125-135.
19. Zaccariotto M, Mudric T, Tomasi D, Shojaei A, Galvanetto U. Coupling of fem meshes with peridynamic grids. *Comput Methods Appl Mech Eng*. 2018;330:471-497.
20. Fang G, Liu S, Fu M, Wang B, Wu Z, Liang J. A method to couple state-based peridynamics and finite element method for crack propagation problem. *Mech Res Commun*. 2019;95:89-95.
21. Ni T, Zaccariotto M, Zhu Q-Z, Galvanetto U. Coupling of fem and ordinary state-based peridynamics for brittle failure analysis in 3d. *Mech Adv Mater Struct*. 2021;28(9):875-890.
22. Lehoucq RB, Silling SA, Plimpton SJ, Parks ML. Peridynamics with lammps: a user guide. Tech. rep., Citeseer 2008.

23. Silling SA, Askari E. A meshfree method based on the peridynamic model of solid mechanics. *Comput Struct*. 2005;83(17-18): 1526-1535.
24. Sakhavand N. Parallel simulation of reinforced concrete structures using peridynamics. Master's thesis, University of New Mexico 2011.
25. Parks ML, Littlewood DJ, Mitchell JA, Silling SA. Peridigm users' guide v1. 0.0. SAND Report 7800.
26. Lee J, Oh SE, Hong J-W. Parallel programming of a peridynamics code coupled with finite element method. *Int J Fracture*. 2017;203(1-2):99-114.
27. Li X, Ye H, Zhang J. Large-scale simulations of peridynamics on sunway taihulight supercomputer. 49th International Conference on Parallel Processing-ICPP, 1-11. 2020.
28. Diehl P, Jha PK, Kaiser H, Lipton R, Lévesque M. An asynchronous and task-based implementation of peridynamics utilizing hpx–the c++ standard library for parallelism and concurrency. *SN Appl Sci*. 2020;2(12):1-21.
29. Liu W, Hong J-W. Discretized peridynamics for brittle and ductile solids. *Int J Numer Methods Eng*. 2012;89(8):1028-1046.
30. Diehl P. Implementierung eines peridynamik-verfahrens auf gpu. Master's thesis 2012.
31. Zhang G, Bobaru F. Modeling the evolution of fatigue failure with peridynamics, the Romanian journal of technical sciences. *Appl Mech*. 2016;61(1):22-40.
32. Mossaiby F, Shojaei A, Zaccariotto M, Galvanetto U. Opencl implementation of a high performance 3d peridynamic model on graphics accelerators. *Comput Math Appl*. 2017;74(8):1856-1870.
33. Ni T, Zaccariotto M, Zhu Q, Galvanetto U. Matrix-based implementation and gpu acceleration of linearized ordinary state-based peridynamic models in matlab. arXiv preprint arXiv:2309.11273.
34. Silling SA. Linearized theory of peridynamic states. *J Elasticity*. 2010;99:85-111.
35. Le Q, Chan W, Schwartz J. A two-dimensional ordinary, state-based peridynamic model for linearly elastic solids. *Int J Numer Methods Eng*. 2014;98(8):547-561.
36. Sarego G, Le QV, Bobaru F, Zaccariotto M, Galvanetto U. Linearized state-based peridynamics for 2-d problems. *Int J Numer Methods Eng*. 2016;108(10):1174-1197.
37. Van Le Q, Bobaru F. Objectivity of state-based peridynamic models for elasticity. *J Elasticity*. 2018;131(1):1-17.
38. Lewis RW, Schrefler BA. *The Finite Element Method in the Static and Dynamic Deformation and Consolidation of Porous Media, no. BOOK*. John Wiley; 1998.
39. Zhang H, Qiao P. A state-based peridynamic model for quantitative fracture analysis. *Int J Fracture*. 2018;211(1-2):217-235.
40. Ni T, Zhu Q-Z, Zhao L-Y, Li P-F. Peridynamic simulation of fracture in quasi brittle solids using irregular finite element mesh. *Eng Fracture Mech*. 2018;188:320-343.
41. Cao TD, Milanese E, Remij EW, et al. Interaction between crack tip advancement and fluid flow in fracturing saturated porous media. *Mech Res Commun*. 2017;80:24-37.
42. Witherspoon PA, Wang JS, Iwai K, Gale JE. Validity of cubic law for fluid flow in a deformable rock fracture. *Water Resour Res*. 1980;16(6):1016-1024.
43. Zimmerman RW, Bodvarsson GS. Hydraulic conductivity of rock fractures. *Transp Porous Media*. 1996;23:1-30.
44. Ni T, Zaccariotto M, Zhu Q-Z, Galvanetto U. Static solution of crack propagation problems in peridynamics. *Comput Methods Appl Mech Eng*. 2019;346:126-151.
45. Taylor L, Flanagan D. *Pronto 3d: A Three-Dimensional Transient Solid Dynamics Program, Tech. Rep*. Sandia National Labs; 1989.
46. Zhou X, Wang Y, Qian Q. Numerical simulation of crack curving and branching in brittle materials under dynamic loads using the extended non-ordinary state-based peridynamics. *Eur J Mech-A/Solids*. 2016;60:277-299.
47. Hauth M, Etzmuß O, Straßer W. Analysis of numerical methods for the simulation of deformable models. *Vis Comput*. 2003;19: 581-600.
48. Zhang J, Chauhan S. Fast explicit dynamics finite element algorithm for transient heat transfer. *Int J Thermal Sci*. 2019;139:160-175.
49. Rong X, Niu R, Liu G. Stability analysis of smoothed finite element methods with explicit method for transient heat transfer problems. *Int J Comput Methods*. 2020;17(2):1845005.
50. Zienkiewicz OC, Chan A, Pastor M, Schrefler B, Shiomi T. *Computational Geomechanics*. Vol 613. Citeseer; 1999.
51. Khoei A, Vahab M, Hirmand M. An enriched–fem technique for numerical simulation of interacting discontinuities in naturally fractured porous media. *Comput Methods Appl Mech Eng*. 2018;331:197-231.
52. Sun W, Fish J, Guo C. Parallel pd-fem simulation of dynamic fluid-driven fracture branching in saturated porous media. *Eng Fracture Mech*. 2022;274:108782.
53. Simoni L, Secchi S. Cohesive fracture mechanics for a multi-phase porous medium. *Eng Comput*. 2003;20(5/6):675-698.
54. Yang Y, Tang X, Zheng H, Liu Q, Liu Z. Hydraulic fracturing modeling using the enriched numerical manifold method. *App Math Model*. 2018;53:462-486.
55. Schanz M, Cheng A-D. Transient wave propagation in a one-dimensional poroelastic column. *Acta Mech*. 2000;145(1-4):1-18.
56. Dubner H, Abate J. Numerical inversion of laplace transforms by relating them to the finite fourier cosine transform. *J ACM*. 1968;15(1):115-123.
57. Geertsma J, De Klerk F. A rapid method of predicting width and extent of hydraulically induced fractures. *J Petrol Tech*. 1969;21(12):1571-1581.
58. Secchi S, Simoni L, Schrefler BA. Mesh adaptation and transfer schemes for discrete fracture propagation in porous materials. *Int J Numer Anal Methods Geomech*. 2007;31(2):331-345.
59. Seleson P, Parks M. On the role of the influence function in the peridynamic theory. *Int J Multiscale Comput Eng*. 2021;9(6).

60. Gu X, Madenci E, Zhang Q. Revisit of non-ordinary state-based peridynamics. *Eng Fracture Mech*. 2018;190:31-52.
61. Littlewood DJ. *Roadmap for Peridynamic Software Implementation, Tech. Rep*. Sandia National Lab (SNL-NM); 2015.
62. Sun Y, Edwards MG, Chen B, Li C. A state-of-the-art review of crack branching. *Eng Fracture Mech*. 2021;257:108036.

How to cite this article: Ni T, Zhang J, Zaccariotto M, Galvanetto U, Schrefler BA. Matrix-based implementation and GPU acceleration of hybrid FEM and peridynamic model for hydro-mechanical coupled problems. *Int J Numer Methods Eng*. 2024;125(18):e7504. doi: 10.1002/nme.7504

APPENDIX A. INITIALIZATION OF THE VARIABLES REQUIRED FOR THE SOLVER EXECUTED ON CPU

```
%Array of the pore pressure value of each node.
P=zeros(Nn,1);
%Array of the pore pressure value of each node in previous iteration.
Pold=zeros(Nn,1);
%Array of the external source term of each node.
Q=zeros(Nn,1);
%Array of the current source term of each node.
Qc=zeros(Nn,1);
%Array of the displacement components of each node.
U=zeros(3*Nn,1);
%Array of the velocity components of each node.
VEL=zeros(3*Nn,1);
%Array of the velocity components of each node in previous iteration.
VELold=zeros(3*Nn,1);
%Array of the acceleration components of each node.
ACC=zeros(3*Nn,1);
%Array of the peridynamic force components of each node.
PDF=zeros(3*Nn,1);
%Array of the external force components of each node.
EXF=zeros(3*Nn,1);
%Array of the residual force components of each node.
RF=zeros(3*Nn,1);
%Array of the dilatation values of each node.
Theta=zeros(Nn,1);
%Array of the extension scalar state value of each bond.
E=zeros(Nb,1);
```

APPENDIX B. AN EXAMPLE OF IMPLEMENTATION OF THE SOLVER BASED ON VECTOR AND MATRIX OPERATIONS

```
for it=1:IntTimeStep
        %===========Block for applying Boundary Conditions============%
        ...
        %===========Block for updating pore pressure field============%
        Qc=Dt*(Q+Kww*Pold)+Kws*(U-Uold);
        P(PFreeDofs)=Pold(PFreeDofs)+Qc(PFreeDofs)./Cww(PFreeDofs);
        Uold=U;
        %====Block for the computation of Peridynamic force Array====%
        %Calculate the extension scalar values of each bond.
        E=CE*U;
        %Compute the dilatation values of each node.
        Theta=CTH*E;
        %Calculate the peridynamic internal force components of each node.
        PDF=KTH*Theta+KE*E;
        %=======Update the displacement components of each node=======%
        %Calculate the residual force components of each node.
        RF=EXF+PDF+Ksw*P;
```

```
        %Calculate the acceleration components of each node.
        ACC=RF./M;
%Calculate the current velocity components of each node.
        VEL=VELold+ACC*Dt;
        %Calculate the current displacement components of each node.
        U(UFreeDofs)=U(UFreeDofs)+Dt*VEL(UFreeDofs);
        %Store the current velocity components and ppressure value of each node.
        VELold=VEL;
        Pold=P;
        %============Block for applying failure ctriterion============%
        ...
        %=================Block for the result output=================%
        ...
end
```