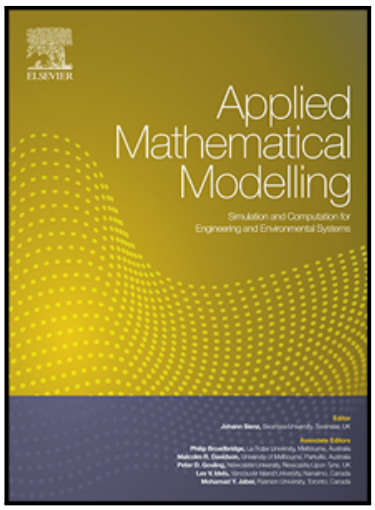


A novel fast overrelaxation updating method for continuous-discontinuous cellular automaton

Fei Yan , Peng-Zhi Pan , Xia-Ting Feng , Shao-Jun Li , Quan Jiang

PII: S0307-904X(18)30427-X
DOI: https://doi.org/10.1016/j.apm.2018.08.025
Reference: APM 12439

To appear in: *Applied Mathematical Modelling*

Received date: 8 May 2018
Revised date: 26 July 2018
Accepted date: 29 August 2018

Please cite this article as: Fei Yan , Peng-Zhi Pan , Xia-Ting Feng , Shao-Jun Li , Quan Jiang , A novel fast overrelaxation updating method for continuous-discontinuous cellular automaton, *Applied Mathematical Modelling* (2018), doi: https://doi.org/10.1016/j.apm.2018.08.025

A novel fast overrelaxation updating method for continuous-discontinuous cellular automaton

Fei Yan*, Peng-Zhi Pan, Xia-Ting Feng, Shao-Jun Li, Quan Jiang

State Key Laboratory of Geomechanics and Geotechnical Engineering, Institute of Rock and Soil Mechanics, Chinese Academy of Sciences, Wuhan 430071, China

*** Corresponding author:** Fei Yan

Address: State Key Laboratory of Geomechanics and Geotechnical Engineering, Institute of Rock and Soil Mechanics, Chinese Academy of Sciences, Xiaohongshan, Wu Chang, Wuhan 430071, China

Tel: 86-27-87198805

Mobile: 86-13477073381

Fax: 86-27-87198413

E-mail: fyan@whrsm.ac.cn

Highlights:

1. A fast successive relaxation updating method for continuous-discontinuous cellular automaton(CDCA) is proposed.
2. A fast CDCA is developed, and increments of displacement and nodal force are enlarged by the accelerating factor.
3. A new discontinuity tracking method which combines cell space cutting and cell neighbor searching is proposed.
4. The optimal value of the accelerating factor is studied, and an adaptive iteration scheme is proposed.

Abstract

Because of its local property, cellular automaton method has been widely applied in different subjects, but the main problem is that the cellular updating is time-consuming. In order to improve its calculation efficiency, a fast successive relaxation updating method is proposed in this paper. Firstly, an accelerating factor ω is defined, and a fast successive relaxation updating theory and its corresponding convergence conditions are developed. In each updating step, the displacement

increment is enlarged ω times as a new increment to replace the old one, similarly, the nodal forces for its neighbors caused by this displacement increment are also enlarged by the same accelerating factor, and do those updating operations until the convergence is achieved. By this method, the convergence rate is greatly improved, by a suitable accelerating factor, 90 to 98 percent of iteration steps are decreased compared to that of the traditional method. Besides, the influence factors for the accelerating factor are studied, and numerical studies show that the suitable accelerating factor is $1.85 < \omega < 1.99$, which is greatly influenced by cell stiffness, and the optimal accelerating factor is $1.96 < \omega < 1.99$. Finally, numerical examples are given to illustrate that the present method is effective and high convergence rate compared to the traditional method.

Keywords

The continuous-discontinuous cellular automaton method; The fast successive relaxation updating method; Updating rule; the accelerating factor; Convergence

1. Introduction

Because of the easy use of simple, local interaction rules to solve complex global behavior, cellular automaton(CA) is widely used for broad areas and a diverse variety of disciplines in recent years. The CA model divides a calculating domain into discrete cells with states and neighbors, and represents the entire system as a set of interacting cells, which evolves in time via simple rules based on cell itself and its neighbor interactions. Based on its locality, uniformity and simple transition rules, CA has provided deep insight into complex processes such as multiphase flows, chemical reaction, drug release, neural activity, social segregation, music composing, solid mechanics and so on. One significant advantage of CA model is that they can be constructed in a modular fashion based on cells, another advantage of CA is that they are intrinsically coarse grained, locality, and easy implementation of the parallel algorithm.

CA was first introduced in the late 1940s by Ulan and Neumann [1]. Since Neumann's early work, CA method has been widely applied for modeling many complex practical

problems, particularly in social sciences, for example, Wolfram [2] demonstrated that some complex natural phenomena can be modeled by CA; Schreckenburg et al. [3] used discrete stochastic models to solve single-lane traffic flow in a ring; later, Green et al. [4] employed CA to represent landscapes, and developed a program of IFNITE based on cellular automaton; Zorzenon and Coutinho [5] applied a CA model to simulate dynamics of HIV infection; Gardner [6] did an artificial life exercises by cellular automata simulation. Another widely used area of CA is physics area, such as, thermal modeling by Honma and Tosaka [7], intrinsic properties analyzing for a carbon nanotube structure by Ryoo et al. [8], electromagnetic phenomena simulating by Simons [9], complex physical systems modeling by Manneville et al. [10] and Chopard [11], reaustenitization of hypoeutectoid steels simulating by Yang et al. [12], vessel morphogenesis analyzing by Markus [13], and comparison studies by Bernsdorf et al. [14].

Besides, CA model is also widely used in biology, urban planning and its related areas in recent years. For example, Lan et al. [15] employed a combination of CA and finite element model to simulate transformation process from austenite to ferrite in a steel, Das et al. [16] employed CA to model stress-dependent grain boundary growth. Ben Youssef and Ben [17] used a three-dimensional CA model for tissue growth, Forestell et al. [18] proposed a simple and useful CA model for the growth of contact inhibited cells, Ivan et al. [19] combined BacMIST and CA algorithm for biofilm growth with Brownian diffusion, Kansal et al. [20] simulated brain tumor growth dynamics using a three-dimensional CA.

As a cellular automaton method, Lattice-Gas Automata (LGA) has been widely applied for fluid dynamics, Navier-Stokes equation, sound wave propagation and so on. For example, Frisch et al. [21] proposed LGA for Navier-Stokes equation, further Chen et al. [22] used FHP based LGA to solve sound wave propagation; later, Eissler et al. [23] simulated flow around chains of cylinders by LGA, and Vogeler and Wolf-Gladrow[24] simulated flow past obstacles in two and three dimensions by Pair interaction LGA; McCarthy[25] used gas cellular automaton for simulation of flow

through arrays of cylinders; Chopard et al. [26] simulated erosion and particles transport in a fluid by LGA, and Chen et al. [27] employed LGA for simulation of flow and heat transfer.

Lattice Boltzmann method(LBM) have been widely used in scientific research and engineering in the past decades, which is originated from LGA. He and Doolen[28] introduced curvilinear coordinates system into LBM and solved flow around a circular cylinder[29]; Krafczyk et al. [30] proposed LBM to model computational fluid dynamics; Masselot and Chopard [31] employed LBM for particle transport and deposition; Przekop et al. [32] analyzed description of the structure of deposited particulate matter in fibrous filters by LBM; Wang et al. [33] used LBM to simulate particle capture process of fibrous filters; Owen et al. [34] combined LBM and immersed moving boundaries for solving fluid-structure interaction; Grucelski and Pozorski [35] and Zhou et al. [36] employed LBM to solve flow past a circular cylinder.

Based on LGA and LBM, the theories and applications of CA to solid mechanics can be seen in many literatures. For example, Eugenio and Rasetti [37] developed a CA model for elasticity, and Olami et al. [38] introduced CA model to represent a quasistatic earthquake model. Then Abdellaoui et al. [39] applied CA for contact problem, and Canyurt et al. [40] employed CA for studies of structural analysis and optimization. Later, Popov and Psakhie [41] proposed a movable CA to model elastoplastic. Khvastunkov and Leggoe [42] combined FEM with CA to analyze failure for ductile alloys. Psakhie et al. [43] developed a combination of finite difference method(FDM) and CA to simulate the behavior of heterogeneous media. Additionally, Slotta et al. [44] studied the convergence properties of CA model for some practical design.

For elastodynamics, Rothman [45] developed a modified CA rules and employed this model to simulate seismic wave, recently, Leamy [46] detailed a physics-based CA model and applied for simulating seismic problem, furthermore, Hopman and Leamy [47] developed a triangular cells CA model, and a comparison studies is done for

simulating elastodynamic response by the present triangular cells and the traditional rectangular cells. 3D effects play a important role in fracture mechanics, then Campagnolo, Lazzarin and Berto[48-50] studied the effects of boundary conditions and cracked components for crack tip stresses. And Li et al. [51-53] proposed a hybrid multi-dimensional model and employed for simulating DSG-PTC-CSP plant.

Based on LBM and solid mechanics, Shen et al. [54] proposed an updating rules based on self organizing of cells, and used this method for elasticity problems, combined this updating rule and elasto-plastic theory, Feng et al. [55] developed a elasto-plastic cellular automaton method (EPCA), and used to simulate the rock crack fracturing under uniaxial compression. According to discontinuous enrichment theory and CA model, Yan and Pan [56] proposed a discontinuous CA model, in which a discontinuous cell model and updating rules are constructed, and employed for simulation of elastic crack fracturing [56], frictional contact problems [57] and cohesive crack propagation [58].

Although some advantages of CA model can be obviously seen in the above literatures, the original schemes of updating operation for all above methods are often time-consuming, and most CPU time of the whole calculating is spent on CA updating, therefore, the updating rules are the most important part for efficient CA model, so Yan et al. [59] developed an adaptive cellular updating scheme for continuous-discontinuous cellular automaton (CDCA), and effectiveness of updating has been improved, but increase amplitude is limited, at most no more than 20 percent of updating steps are decreased, for some large scale practical calculating for engineering, development of much faster updating method for CDCA is imminent, unfortunately, no literatures about fast updating scheme for CA updating can be found up to now.

Aimed at those issues, a fast successive relaxation updating method is proposed to replace the traditional cellular updating scheme in this paper. In this method, an accelerating factor ω is defined, and a fast successive relaxation updating theory is developed. Actually in every updating step, the displacement increment is enlarged

ω times as the new increment, similarly, the nodal forces increment of its neighbors caused by this displacement increment are also enlarged by the same accelerating factor, and do those updating operations until convergence is achieved. By this method, convergence rate is greatly improved, by which it is much smoother and quicker. And using a suitable accelerating factor, at least 90 percent of updating steps are decreased, and at most 98 percent of updating steps are decreased, the efficiency of the present fast updating method is averagely 10 times than that of the traditional CA model. Numerical studies are shown that the suitable value of the accelerating factor is $1.85 < \omega < 1.99$, which is greatly influenced by cell stiffness, and the optimal value of the accelerating factor is $1.96 < \omega < 1.99$. Finally, some numerical examples are given to illustrate that the present method is effective and high convergence rate compared to the traditional cellular automaton method.

2. Theory of continuous-discontinuous cellular automaton method

2. 1 Discontinuity tracking

Employing the advantage of CA model of CDCA, we develop a new method to track the discontinuities, which combines cell space cutting, cell neighbor searching and distance value to crack surface and front. Different from level set method, the discontinuity tracking calculation of the present method is only limited on some finite local cells, and via this method, the calculation effectiveness is greatly improved compared to global searching of level set method.

Employing cell space cutting, firstly, assume that crack tip is located on cell C_1, according to cell neighbor information and crack path location information, the crack path will cut the segment connecting cell node 1 and cell node 2, via cell node 1, cell node 2 and cellular automata model, the next cutting cell C_2 can be easily obtained. According to this rule, until the cutting cell searching for the whole crack path is finished, all cutting cell can be easily obtained, which can be seen in Fig. 1. For all cutting cells, calculating the distances value ϕ_i and φ from each cell node to crack surface and crack front, then via those two distance values, we can precisely track the

discontinuities. According to the whole process of the present method, it can be seen that the tracking process of the present method are only limited on some local cells near cracks, and the data only for those local cells is needed to save.

2. 2 Variational formulation

A linear elastic body under tractions $\mathbf{t}_c$ on crack surfaces Γ_c is considered in this section. According to discontinuous CA model, the variational equation [56-58] can be given as:

$$\begin{aligned} &W^{\text{int}} = W^{ext} + W^{fri} \\ &= \int_\Omega \boldsymbol{\sigma} \cdot \delta\boldsymbol{\varepsilon} d\Omega = \int_\Omega \mathbf{f}_b \delta\mathbf{u} d\Omega + \int_{\Gamma_t} \hat{\mathbf{t}} \delta\mathbf{u} d\Gamma + \int_{\Gamma_c} \mathbf{t}_c (\delta\mathbf{u}^{s+} - \delta\mathbf{u}^{s-}) d\Gamma \end{aligned} \tag{1}$$

in which W^{int}, W^{ext}, W^{fri} are virtual work of internal, external, crack surface friction; $\boldsymbol{\sigma}$, $\boldsymbol{\varepsilon}$ are stress, strain tensor respectively; $\mathbf{f}_b$ is body force; $\mathbf{u}$ is displacement vector; $\hat{\mathbf{t}}$ is traction of boundary; $\mathbf{u}^{s+}$, $\mathbf{u}^{s-}$ are displacement on upper and bottom surface of crack [56-58].

The same as XFEM, applying enriched shape function to describe the discontinuities, we can get the shape function

$$\begin{aligned} \mathbf{u}^h(\mathbf{x}) = &\sum_{j=1}^{n} N_j(\mathbf{x}) \mathbf{d}_j + \sum_{k=1}^{m} N_k(\mathbf{x}) \left(H(\xi) - H(\xi_k) \right) \mathbf{a}_k \\ &k \in P \\ &+ \sum_{i=1}^{t} N_i(\mathbf{x}) \left(\sum_{l=1}^{nf} \left(F_l(\mathbf{x}) - F_l(\mathbf{x}_i) \right) \mathbf{b}_i^l \right) = \tilde{\mathbf{N}} \{ \mathbf{d} \quad \mathbf{a} \quad \mathbf{b} \} \\ &i \in T \end{aligned} \tag{2}$$

where the strong discontinuous enriched function is given as $H(\xi) = \begin{cases} =1 & \xi > 0 \\ =-1 & \xi < 0 \end{cases}$

and the crack tip high gradient stress enriched functions can be written as

$F_l(x) = \left\{ \sqrt{r}\sin(\frac{\theta}{2}), \sqrt{r}\cos(\frac{\theta}{2}), \sqrt{r}\sin(\theta)\sin(\frac{\theta}{2}), \sqrt{r}\sin(\theta)\cos(\frac{\theta}{2}) \right\}$ [56-58], and the approximating function for crack surfaces can be written as

$$\mathbf{w}_c^h(\mathbf{x}) = \mathbf{u}_{S^+}^h - \mathbf{u}_{S^-}^h = 2\sum_{k=1}^{m} N_k(\mathbf{x}) \mathbf{a}_k + 2\sum_{i=1}^{t} N_i(\mathbf{x}) r \mathbf{b}_i^0 = \hat{\mathbf{N}} \{ \mathbf{a} \quad \mathbf{b} \} \tag{3}$$

$k \in P$ $\quad$ $i \in T$

in which $\hat{\mathbf{N}}$ is the shape function related to cohesive or frictional displacement field, then we can get the system linear function [56-58]

$$(\mathbf{k}_u + \mathbf{k}_a + \mathbf{k}_b + \mathbf{k}_c)\mathbf{d} = \mathbf{f}_u + \mathbf{f}_a + \mathbf{f}_b + \mathbf{f}_{t_c} \tag{4}$$

where some symbols and equations are given as follows, the others can be given as [56-58].

$$\mathbf{f}_u = \int_\Omega \mathbf{N}^{\mathbf{T}} \mathbf{f}_b d\Omega + \int_\Gamma \mathbf{N}^{\mathbf{T}} \hat{\mathbf{t}} d\Gamma \tag{5}$$

$$\mathbf{f}_{a,b} = \int_{\Omega^h} \tilde{\mathbf{N}}^{\mathbf{T}} \mathbf{f}_b d\Omega + \int_{\Gamma^h} \tilde{\mathbf{N}}^{\mathbf{T}} \hat{\mathbf{t}} d\Gamma \tag{6}$$

$$\mathbf{f}_{t_c} = \int_S \hat{\mathbf{N}}^T \mathbf{t}_c dS \tag{7}$$

$$\mathbf{k}_c = \int_S \hat{\mathbf{N}}^{\mathbf{T}} \mathbf{G}_c \hat{\mathbf{N}} dS \tag{8}$$

$$\mathbf{k}_{u,a,b} = \int_{\Omega^e} (\mathbf{B}_{u,a,b})^T \mathbf{D} (\mathbf{B}_{u,a,b}) d\Omega \tag{9}$$

in which $\mathbf{G}_c$ is the matrix of crack surface normal and tangential stiffness, Obtain the above matrices according to Eqs. (4) - (9), then use the next section updating rules to solve the solution of the present method.

2.3 Excavation implementation

The same Eq. (2), one can get the enriched shape function for cell which is crossed by excavation boundary

$$\mathbf{u}^h(\mathbf{x}) = \sum_{j=1}^{n} N_j(\mathbf{x})\mathbf{d}_j + \sum_{k=1}^{m} N_k(\mathbf{x})\left(H(\xi) - H(\xi_k)\right)\mathbf{a}_k \tag{10}$$

$k \in P$

And another function is defined to distinguish cells which is located interior or outside of the excavating cavern, which is

$$\begin{cases} V(x) = 1 & \xi \geq 0 \\ V(x) = 0 & \xi < 0 \end{cases} \tag{11}$$

By this method, the discretization of boundary of the cavern is independent of the mesh grid, and the degrees of freedom associated with cells who is entirely inside the cavern are removed from the system of equations, in this case, $V(x) = 0$.

2.4 Cell, neighbor, cell space and its states

It can be seen in Fig. 2, according to discontinuous CA model, cells are composed of continuous cells and discontinuous cells, which can be referred as [56-58]. The cell space includes cell elements E_i^j, cell nodes N_i which belongs to cell elements E_i^j, cell nodes N_i^k of the neighbors of cell nodes N_i. Cell nodes have three types, which are the traditional continuous nodes, the strong discontinuous enriched nodes and the crack tip asymptotic function enriched nodes, and cell elements can be divided into the traditional continuous elements, the fully penetrated elements and the crack tip elements. According to discontinuous CA model [56-58], cell states and relation between cell and its neighbors, which can be seen in Fig. 2, can be referred to references [57-58],

In order to fully describe the states of a cell, some mechanical variables are employed, such as: the displacement vector of cell $\mathbf{u}=\{\mathbf{u},\mathbf{a},\mathbf{b}\}^T$; nodal forces vector $\mathbf{f}=\{\mathbf{f_u},\mathbf{f_a}+\mathbf{f}_{t_c},\mathbf{f_b}+\mathbf{f}_{t_c}\}^T$; stiffness matrices of cell $\mathbf{K}_i=\mathbf{k}_u+\mathbf{k}_a+\mathbf{k}_b+\mathbf{k}_c$; distance values ϕ_i and φ for discontinuities or cracks; cell stress $\boldsymbol{\sigma}$; cell strain ε; relative displacement of crack surfaces $\mathbf{w}_c$, upper surface displacement of crack $\mathbf{u}^{s+}$ and bottom surface displacement of crack $\mathbf{u}^{s-}$; frictional contact stress vector of crack surface $\mathbf{t}_c$; contact state variable of crack surface c_s; the accelerating factor ω and so on, and cell states can also be seen as Fig. 3.

2.5 The traditional updating rules

In this section, the traditional updating rules are introduced, a cell node is considered, the nodal force vector of this cell is , stiffness matrix of this cell is , the nodal freedoms for neighbors of cell are restricted, which is plotted in Fig. 4 [57-58]. The states updating for cell and its neighbors can divided into two steps, 1) displacement increment $\Delta\mathbf{u}_i=\{\Delta\mathbf{u}_i,\Delta\mathbf{a}_i,\Delta\mathbf{b}_i\}^T$ of cell can be obtained from nodal force

increment $\Delta\mathbf{f}_i = \{\Delta\mathbf{f}_i^u, \Delta\mathbf{f}_i^a, \Delta\mathbf{f}_i^b\}^T$ of cell itself, 2) the nodal force increment $\Delta\mathbf{f}_k$ for neighbors is caused by the nodal displacement increment $\Delta\mathbf{u}_i$ of cell N_i, do updating on all cells, until equilibrium is achieved, namely, when $\Delta\mathbf{u}_i \to 0$ and $\Delta\mathbf{f}_i \to 0$ are attached.

The traditional updating steps are written as [57-58]:

(1) Firstly, no frictional contact stress is considered in the first updating step, obtain stiffness matrices of cell , , and nodal force vector for cell , . No contact is considered on the first step, exactly, the crack surfaces are opening in the first updating step, at this time $c_s = 0$.

(2) According to the system function of cellular automaton method for cell node , $\mathbf{K}_i \Delta\mathbf{u}_i = \Delta\mathbf{f}_i$, we can get the increment of degrees of nodal freedom $\Delta\mathbf{u}_i$.

(3) Update nodal force increment $\Delta\mathbf{f}_k$ for neighboring cells via $\Delta\mathbf{u}_i$ from the equation $\Delta\mathbf{f}_k = \mathbf{K}_i \Delta\mathbf{u}_i$, and do this operation for all neighbors of .

(4) According to stress and displacement states of step (2) and (3), obtain contact state c_s for crack surfaces on current step, calculate crack opening displacement $\mathbf{w}_c$, and contact stress increment $\Delta\mathbf{t}_c$.

(5) According to the new values of contact state c_s for crack surfaces and frictional contact stress increment $\Delta\mathbf{t}_c$, update stiffness matrix when contact states of crack surfaces are changed, and renew $\mathbf{f}_i = \{\mathbf{f}_i^u, \mathbf{f}_i^a + \mathbf{f}_{t_c}^a, \mathbf{f}_i^b + \mathbf{f}_{t_c}^b\}^T$ when $\Delta\mathbf{t}_c \neq 0$.

(6) Do step (2) to step (5) on all cells, until $\Delta\mathbf{f}_i < \varepsilon_f$, $\Delta\mathbf{u}_i \to \varepsilon_u$ are achieved, and contact state c_s of crack surfaces, relative displacement of crack surfaces $\mathbf{w}_c$ and frictional contact stress $\Delta\mathbf{t}_c$ are kept the same on the last iteration.

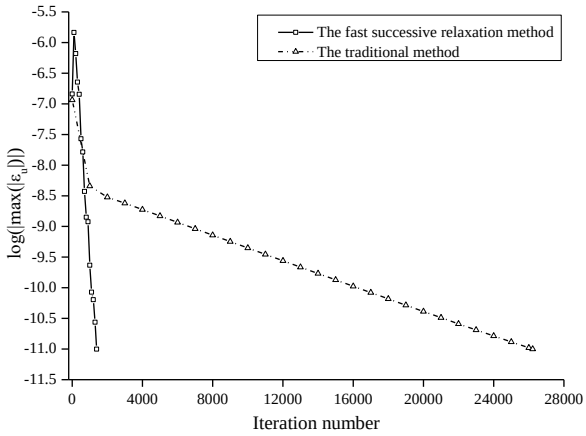


Fig. 8 Comparison for convergence between the present method and the traditional method

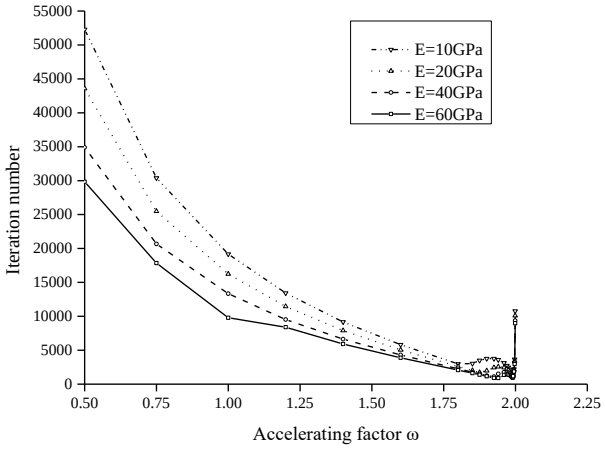


Fig. 9 Iteration numbers for different values of the accelerating factor

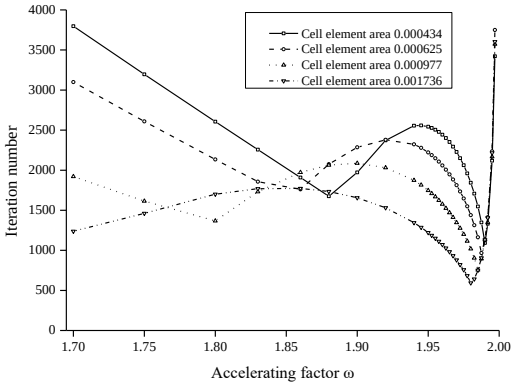
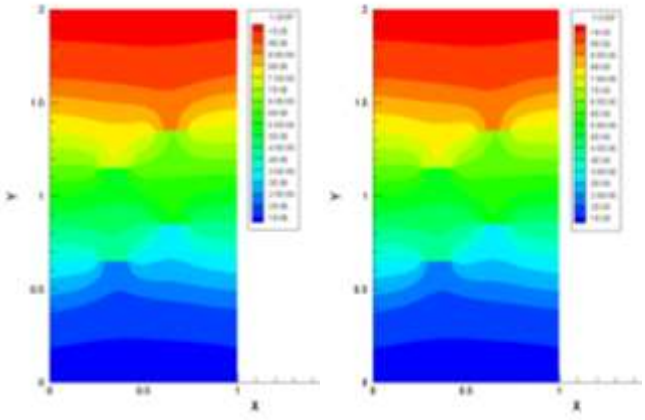


Fig. 10 Iteration number for different areas of cell element and the accelerating factor



a. The present method b. The traditional method

Fig. 11 Comparison for displacement results between the present method and the traditional method

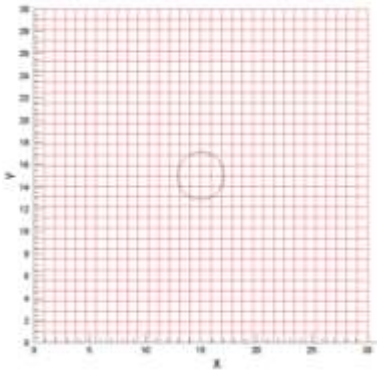


Fig. 12 Model, grid and cavern for rock cavern excavation

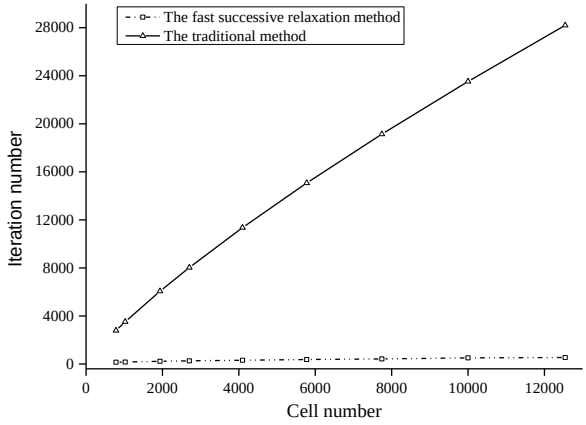


Fig. 13 Comparison for iteration numbers between the present method and the traditional method

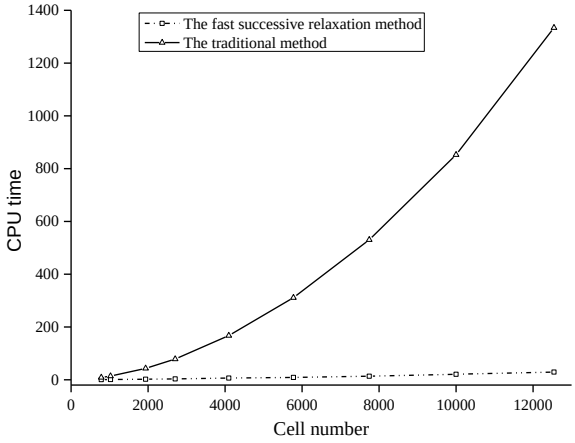


Fig. 14 Comparison for CPU time between the present method and the traditional method

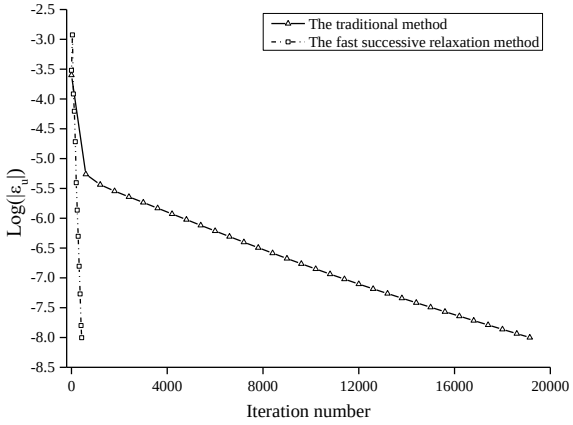


Fig. 15 Comparison for convergence between the present method and the traditional method

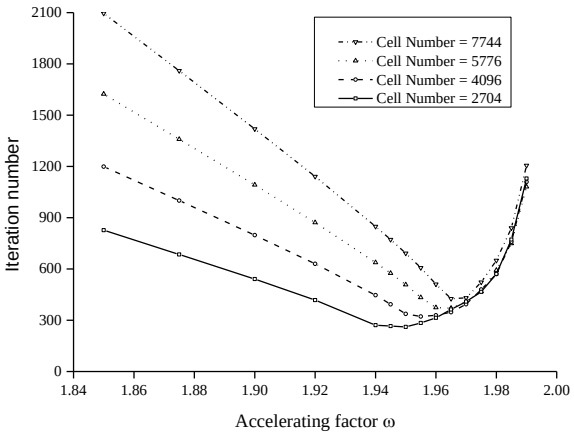
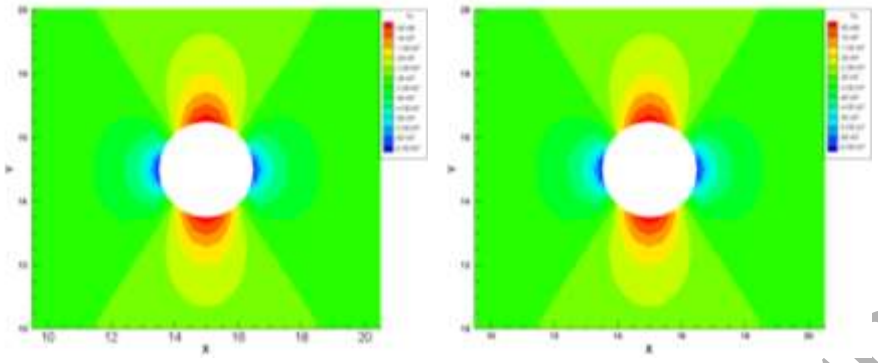


Fig. 16 Iteration numbers for different areas of cell element and the accelerating factor



a. The traditional method b. The present method

Fig. 17 Comparison for displacement results between the present method and the traditional method

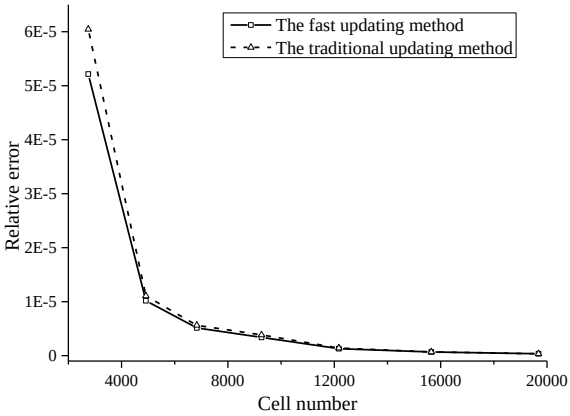


Fig. 18 Comparison for relative error between the present method and the traditional method

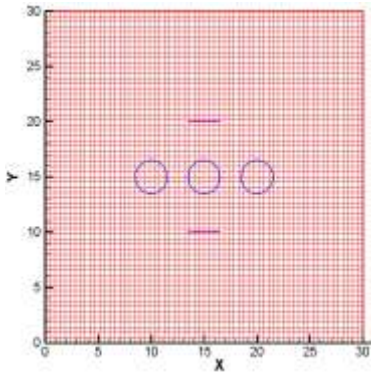


Fig. 19 Model of multiple caverns excavation with cracks

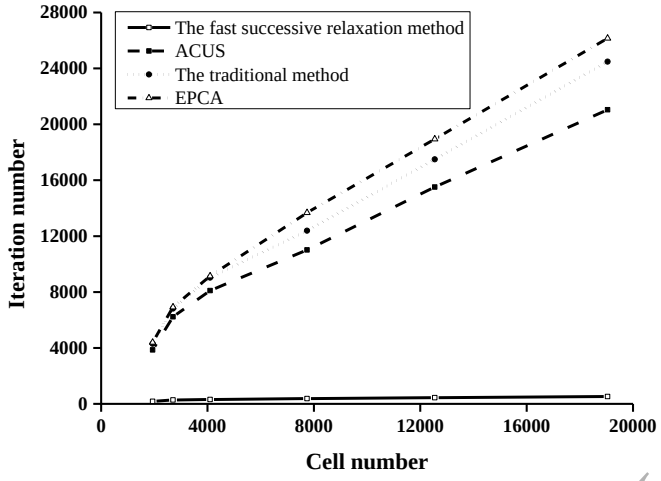


Fig. 20 Comparison of iteration numbers between the present method, the traditional method and EPCA

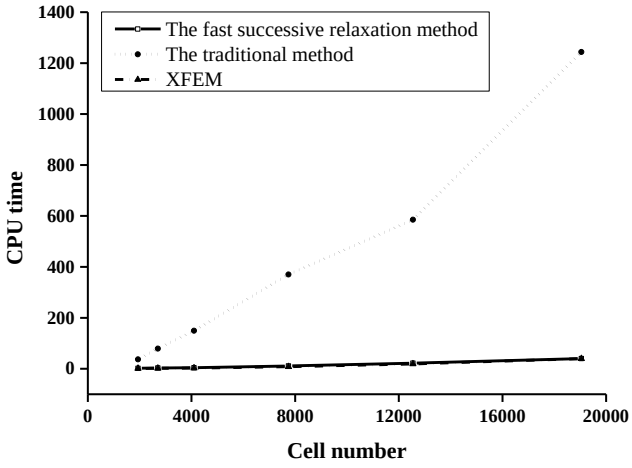


Fig. 21 Comparison of CPU time between the present method, the traditional method and XFEM

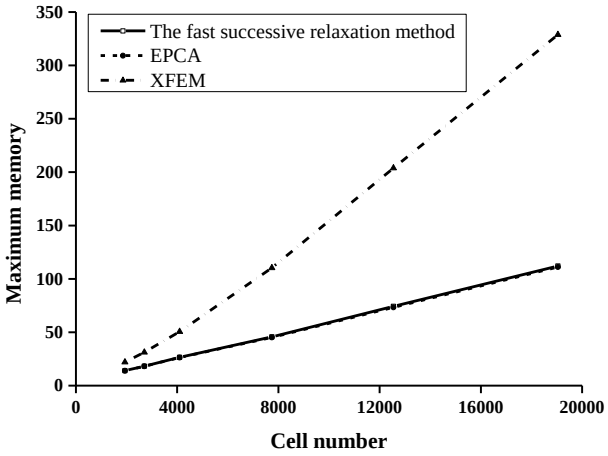


Fig. 22 Comparison of memory requirement between the present method, EPCA and XFEM

Table 1 Iteration numbers for different methods and cell numbers

Cell number	Iteration number		Ratio
	The traditional method N_T	The present method N_F	N_F / N_T
4096	11354	315	0.0277
5776	15073	371	0.0246
7744	19142	426	0.0222
10000	23527	508	0.0216
12544	28203	539	0.0191

Table 2 Displacement results for different methods and cell numbers

Node coordinate	x-direction displacement ($\times 10^{-3}$)				y-direction displacement ($\times 10^{-2}$)			
	The present method	Relative Error (%)	The traditional method	FEM	The present method	Relative Error (%)	The traditional method	FEM
(13.5,15.0)	-6.989	0.03	-6.987	-6.991	-1.842	0.05	-1.841	-1.843
(15.0,13.5)	-9.151	0.02	-9.150	-9.153	-1.340	0.07	-1.341	-1.339
(16.5,15.0)	-11.33	0.08	-11.34	-11.32	-1.842	0.05	-1.841	-1.843
(15.0,16.5)	-9.161	0.02	-9.160	-9.163	-2.347	0.08	-2.346	-2.349
(15.0,30.0)	-9.212	0.02	-9.211	-9.214	-3.702	0.05	-3.702	-3.704
(30.0,15.0)	-18.27	0.05	-18.27	-18.28	-1.833	0.05	-1.832	-1.834

Table 3 Stress results for different methods and cell numbers

Node coordinate	x-direction stress ($\times 10^7$)				y-direction stress ($\times 10^7$)			
	The present method	Relative Error (%)	The traditional method	FEM	The present method	Relative Error (%)	The traditional method	FEM
(13.5,15.0)					-7.050	0.1	-7.050	-7.060
(15.0,13.5)	-3.186	0.2	-3.186	-3.193	-	-	-	-
(16.5,15.0)	-	-	-	-	-7.051	0.1	-7.051	-7.062
(15.0,16.5)	-3.180	0.1	-3.180	-3.185	-	-	-	-
(15.0,30.0)	-2.123	0.04	-2.123	-2.124	-3.000	0.0	-3.000	-3.000
(30.0,15.0)	-2.000	0.0	-2.000	-2.000	-3.039	0.0	-3.039	-3.039