materials

MDPI

Article

Vertex Displacement-Based Discontinuous Deformation Analysis Using Virtual Element Method

Hongming Luo [1], Guanhua Sun [1,2], Lipeng Liu [2] and Wei Jiang [2,3,*]

[1] State Key Laboratory of Geomechanics and Geotechnical Engineering, Institute of Rock and Soil Mechanics, Chinese Academy of Sciences, Wuhan 430071, China; hmluo@whrsm.ac.cn (H.L.); ghsun@whrsm.ac.cn (G.S.)
[2] State Key Laboratory of Simulation and Regulation of Water Cycle in River Basin, China Institute of Water Resources and Hydropower Research, Beijing 100038, China; liulip@iwhr.com
[3] Key Laboratory of Geological Hazards on Three Gorges Reservoir Area, Ministry of Education, China Three Gorges University, Yichang 443002, China
* Correspondence: jiangweilion@163.com

Abstract: To avoid disadvantages caused by rotational degrees of freedom in the original Discontinuous Deformation Analysis (DDA), a new block displacement mode is defined within a time step, where displacements of all the block vertices are taken as the degrees of freedom. An individual virtual element space $V_1(\Omega)$ is defined for a block to illustrate displacement of the block using the Virtual Element Method (VEM). Based on VEM theory, the total potential energy of the block system in DDA is formulated and minimized to obtain the global equilibrium equations. At the end of a time step, the vertex coordinates are updated by adding their incremental displacement to their previous coordinates. In the new method, no explicit expression for the displacement $\boldsymbol{u}$ is required, and all numerical integrations can be easily computed. Four numerical examples originally designed by Shi are analyzed, verifying the effectiveness and precision of the proposed method.

Keywords: DDA; VEM; degrees of freedom; projection

check for updates

Citation: Luo, H.; Sun, G.; Liu, L.; Jiang, W. Vertex Displacement-Based Discontinuous Deformation Analysis Using Virtual Element Method. *Materials* **2021**, *14*, 1252. https://doi.org/10.3390/ma14051252

Academic Editor: Jerzy Smardzewski

Received: 31 December 2020
Accepted: 2 March 2021
Published: 6 March 2021

1. Introduction

Discontinuous Deformation Analysis (DDA) [1], a novel numerical method for analyzing the dynamic mechanical behavior of a block system in cases of large displacement, was verified as an effective tool in solving a variety of discontinuities in rock problems [2,3]. In rock-mass engineering, DDA has been employed to handle a great deal of problems, e.g., landslide process simulations [4,5], slope stability assessments [6,7], blasting effect evaluation [8], crack propagation simulation [9,10], seismic wave propagation analysis [11], and rock burst prediction [12]. Applications of 2D DDA in the modeling of rock-mass dynamics until 2017 were summarized by Ning [13].

Numerous enhancements have been put forward to improve the performance of traditional DDA. To alleviate the sensibility of penalty parameters, contacts between the blocks were remodeled using the Lagrange multiplier method [14], Augmented Lagrangian method [15], Complementary theory [16], and Variational Inequality theory [17]. Apart from modifying kinetic velocity using the dynamic factor [18], the damping effect was used to reflect the energy dissipation by imposing viscous boundary conditions [19] and by adding viscous forces into the equilibrium equation [20]. A couple of strategies, such as adopting a higher-order displacement function [21] and partitioning blocks using finite element mesh [22], were introduced to acquire a more detailed stress field in each block. Moreover, many efforts were made to develop a 3D version of DDA [23–26], considering that 3D DDA is preferred for problems in practical engineering. The biggest bottleneck to establishing a robust 3D version is the lack of an excellent contact theory [27]. Shi [28] proposed the entrance block theory to facilitate contact treatment in 3D DDA. In addition, some efforts, such as GPU-based parallel computation [29] and explicit computation [30], were made to improve the computational efficiency when simulating large-scale problems.

Within a time step, the degrees of freedom ***d*** in the traditional DDA were defined for a block by six independent variables, which consists of two incremental rigid translations, one incremental rigid rotation angle, and three incremental constant strain components. For a point ***x*** in the block, the incremental displacement ***u*** is calculated as the product of ***d*** and the transformation matrix ***T***; then, the global equilibrium equation is derived from this displacement function. The displacement function uses the first-order approximation of sin r_0 = r_0 and cos r_0 = 1 for a small rotation angle r_0, which induces accumulated errors over the time steps and sometimes causes issues. The most noticeable issue is the unreasonable volume expansion when a large rotation occurs. To restrain this issue, some scholars proposed various emendations [31–33], in which the post-adjustment strategy was most popular in the existing DDA codes [34,35]. This method, albeit simple and effective, resulted in a displacement different from the displacement evaluated in the equilibrium equation. Even with a tiny difference, the contact state might be entirely changed for a contact pair. Then, to simulate the continuous–discontinuous deformation, numerical methods for the continuum and non-continuum models were increasingly coupled. DDA was coupled with finite element method (FEM) by some scholars [9,22] for simulation of a rock failure process. In classical FEM, displacements at the element nodes are chosen as the degrees of freedom. This inconsistency in the degrees of freedom between DDA and FEM causes a barrier in developing a compact and efficient code for the coupling method.

In a recent study [36], the displacement of blocks in DDA was reformulated by using Wachspress interpolation to achieve a higher-order stress distribution within blocks. This approach selected displacements at the vertices as the degrees of freedom for a block. In this study, a DDA with such new degrees of freedom is called a vertex displacement-based DDA. This method was improved using the Polygonal Finite Element method (PFEM). Besides Wachspress interpolation [37], a series of interpolations [38–40] were proposed to construct the displacement functions for a polygonal element with more than four nodes in PFEM. By taking displacements at the vertices as the degrees of freedom, these displacement models can avoid demerits caused by the degrees of freedom in an original DDA. However, because the rational function is involved in the shape function, the theoretical derivations and calculations of the stiffness and mass matrix are quite complicated and it is hard to ensure precision of the numerical integration [41,42].

In 2013, the Virtual Element Method (VEM) was proposed to handle the very general polygons, dispensing sophisticated integrations on the element [43–46]. VEM was considered the evolution of the mimetic finite difference method and a generalization of FEM. For flexibility with regard to mesh generation and element shapes, VEM has become a hot topic in numerical methods since it was proposed [47–52]. In this study, a new vertex displacement-based DDA is developed using the strength of VEM. Defining the degrees of freedom by using the incremental displacements ***u*** at the vertices of a block, an individual virtual element space $\boldsymbol{V}_1(\Omega)$ is adopted to describe the displacements of points in the block, and the projector $\Pi_P\boldsymbol{u}$ from $\boldsymbol{V}_1(\Omega)$ on the linear displacement space $\boldsymbol{P}_1(\Omega)$ is deduced. Next, the total potential energy is investigated for the block system. In the potential energy, the bilinear forms of ***u*** are expressed as the summation of the exact solution of $\Pi_P\boldsymbol{u}$ and an approximation of $\boldsymbol{u}$-$\Pi_P\boldsymbol{u}$. The potential energy induced by the contact restraints are derived using the new degrees of freedom. Then, for the block system, the global equilibrium equation is derived based on the principle of minimum potential energy. Finally, the open–close iteration strategy is employed to resolve the global equilibrium equation as the original DDA. The proposed method avoids the issues attributable to first-order approximation for a small rotation angle r_0 in the original DDA and has a higher computational efficiency than vertex displacement-based DDA using displacement functions in PFEM. The validity and effectiveness of the proposed method are verified by several numerical examples.

2. Basic Principles of DDA

As Figure 1 shows, a DDA block system always consists of $\mathrm{n_b}$ discrete blocks with their individual domains and boundaries. The domain Ω^{I} of block I is bounded by $\partial\Omega^{\mathrm{I}}$, which is usually composed of the Dirichlet boundary $\Gamma_{\mathrm{u}}^{\mathrm{I}}$ and the traction boundary $\Gamma_{\mathrm{t}}^{\mathrm{I}}$. The displacement on $\Gamma_{\mathrm{u}}^{\mathrm{I}}$ is prescribed as $\hat{\boldsymbol{u}}^{\mathrm{I}}$, and the surface traction on $\Gamma_{\mathrm{t}}^{\mathrm{I}}$ is denoted by $\boldsymbol{t}^{\mathrm{I}}$. Here, the superscript I is the block index. The deformations and large displacements of a block in DDA are accumulated by the incremental displacements and deformations over time steps. The displacements of blocks are independent from each other, and contact constraints are imposed on the interactions between blocks.

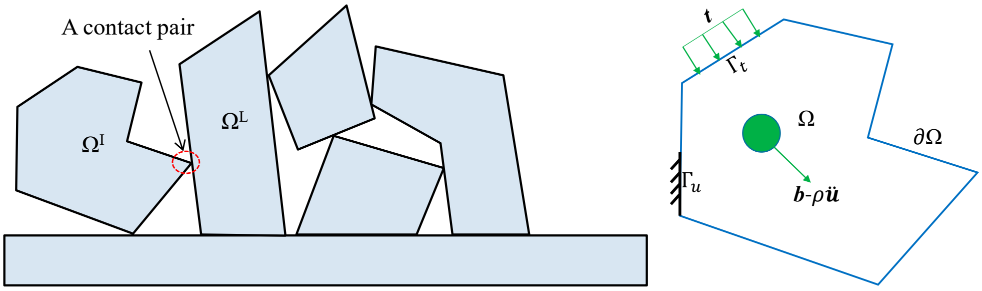

Figure 1. A typical block system and an individual block in Discontinuous Deformation Analysis (DDA).

Within a time step, DDA requires the incremental displacements u = ($\boldsymbol{u}^1$, $\boldsymbol{u}^2$, . . . , $\boldsymbol{u}^{\mathrm{nb}}$) to be the unique minimizer of the total potential energy of the block system:

$$\boldsymbol{u} = \underset{v\in V}{\mathrm{argmin}} J(v),\ J(v) = \frac{1}{2}\alpha(v,v) - f(v) + J'(v) \tag{1}$$

in which $\boldsymbol{V}$ is the incremental displacement space of the block system:

$$V = \boldsymbol{V}^1 \times \boldsymbol{V}^2 \times \ldots \times \boldsymbol{V}^{\mathrm{n}_b},\ \text{with}\ \boldsymbol{V}^I = \left\{ \boldsymbol{v}^I \in \left[H^1\left(\Omega^I\right)\right]^2 : \boldsymbol{v}^I\Big|_{\Gamma_u^I} = \hat{\boldsymbol{u}}^I \right\} \tag{2}$$

where $\boldsymbol{J}'$(v) represents the energy due to the contact constraints. A key feature of DDA is that rigorous contact constraints are used to manage the interactions between blocks. For the contact pair marked in Figure 1, $\boldsymbol{J}'$(v) induced by the contact constraints is determined by $\boldsymbol{u}^{\mathrm{I}}\in\boldsymbol{V}^{\boldsymbol{I}}$ and $\boldsymbol{u}^{\mathrm{L}}\in\boldsymbol{V}^{\boldsymbol{L}}$**.**

The bilinear form α(u, v) and the linear form f(v) are computed for u∈$\boldsymbol{V}$ and v∈$\boldsymbol{V}$ by

$$\alpha(u,v) = \sum_{I=1}^{\mathrm{n}_b} \alpha\left(\boldsymbol{u}^I, \boldsymbol{v}^I\right), f(v) = \sum_{I=1}^{\mathrm{n}_b} f\left(\boldsymbol{v}^I\right) \tag{3}$$

because all blocks are physically isolated in DDA. For ease of description, the superscript indicating blocks is omitted unless otherwise noted. The bilinear form α ($\boldsymbol{u}$, $\boldsymbol{v}$) for $\boldsymbol{u}$∈$\boldsymbol{V}$ and $\boldsymbol{v}$∈$\boldsymbol{V}$ represents the energy due to elastic deformation and the inertial force:

$$\alpha(\boldsymbol{u},\boldsymbol{v}) = \int_\Omega \boldsymbol{\sigma}(\boldsymbol{u}) : \boldsymbol{\varepsilon}(\boldsymbol{v})\,\mathrm{d}\boldsymbol{x} + \int_\Omega \rho\ddot{\boldsymbol{u}} \cdot \boldsymbol{v}\,\mathrm{d}\boldsymbol{x} \tag{4}$$

where $\boldsymbol{\varepsilon}$ is the incremental constant strain decided by $\boldsymbol{u}$:

$$\boldsymbol{\varepsilon}(\boldsymbol{u}) = \frac{1}{2}\left(\nabla\boldsymbol{u} + \nabla^T\boldsymbol{u}\right) \tag{5}$$

in which ∇ is the gradient operator. $\boldsymbol{\sigma}$ is the incremental Cauchy stress related to $\boldsymbol{\varepsilon}$ by the constitutive equation:

$$\boldsymbol{\sigma}(\boldsymbol{u}) = D[\boldsymbol{\varepsilon}(\boldsymbol{u})] \tag{6}$$

where D is a constant elasticity tensor. ρ is the mass per unit area. For dynamic analysis, the inertial force is essential in DDA. New-mark time integration scheme is adopted in the original DDA, and the acceleration is assumed to be constant within a time step. Denoting by $\boldsymbol{V_0}$ the velocity of block I at the origin of a time step and by Δ, the time interval of the time step, the acceleration is computed as

$$\ddot{\boldsymbol{u}} = \frac{2}{\Delta^2}\boldsymbol{u} - \frac{2}{\Delta}\boldsymbol{V}_0 \tag{7}$$

The linear form $f(\boldsymbol{v})$ is defined by

$$f(\boldsymbol{v}) = \int_\Omega \boldsymbol{b}\cdot\boldsymbol{v}\mathrm{d}\boldsymbol{x} - \int_\Omega \sigma_0 : \boldsymbol{\varepsilon}(\boldsymbol{v})\mathrm{d}\boldsymbol{x} + \int_\Omega \boldsymbol{t}\cdot\boldsymbol{v}\mathrm{d}s \tag{8}$$

where $\boldsymbol{b}$ denotes the constant body force and $\boldsymbol{\sigma}_0$ is the constant stress accumulated over the previous time steps.

When the displacement function is determined for a block with the associated degrees of freedom, Equation (1) induces the global equilibrium equations with the following matrix formulation for the block system.

$$\begin{bmatrix} \boldsymbol{K}_{11} & \boldsymbol{K}_{12} & \boldsymbol{K}_{13} & \cdots & \boldsymbol{K}_{1(n_b)} \\ \boldsymbol{K}_{21} & \boldsymbol{K}_{22} & \boldsymbol{K}_{23} & \cdots & \boldsymbol{K}_{2(n_b)} \\ \boldsymbol{K}_{31} & \boldsymbol{K}_{32} & \boldsymbol{K}_{33} & \cdots & \boldsymbol{K}_{3(n_b)} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \boldsymbol{K}_{(n_b)1} & \boldsymbol{K}_{(n_b)2} & \boldsymbol{K}_{(n_b)3} & \cdots & \boldsymbol{K}_{(n_b)(n_b)} \end{bmatrix} \begin{bmatrix} \boldsymbol{d}^1 \\ \boldsymbol{d}^2 \\ \boldsymbol{d}^3 \\ \vdots \\ \boldsymbol{d}^{n_b} \end{bmatrix} = \begin{bmatrix} \boldsymbol{F}^1 \\ \boldsymbol{F}^2 \\ \boldsymbol{F}^3 \\ \vdots \\ \boldsymbol{F}^{n_b} \end{bmatrix} \tag{9}$$

in which $\boldsymbol{d}^I$ is the degrees of freedom concerning block *I*. $\boldsymbol{F}^I$ is the generalized load vector with the same dimension of $\boldsymbol{d}^I$. Denoting the dimension of $\boldsymbol{d}^I$ with dim($\boldsymbol{d}^I$), $\boldsymbol{K}_{II}$ is a dim($\boldsymbol{d}^I$) × dim($\boldsymbol{d}^I$) matrix and $\boldsymbol{K}_{IL(I\neq L)}$ is a dim($\boldsymbol{d}^I$) × dim($\boldsymbol{d}^L$) matrix denoting the contact restraints on blocks *I* and *L*. If no contact pair is provided by blocks *I* and *L* in the current time step, $\boldsymbol{K}_{IL}$ is zero.

3. Demerits Caused by the Original Degrees of Freedom and Previous Attempts to Construct a Vertex Displacement-Based DDA

3.1. Demerits Caused by the Original Degrees of Freedom in DDA

For block Ω, the traditional DDA defines the degrees of freedom $\boldsymbol{d}$ within one time step:

$$\boldsymbol{d} = \begin{pmatrix} u_0 & v_0 & r_0 & \varepsilon_x & \varepsilon_y & \gamma_{xy} \end{pmatrix}^T \tag{10}$$

where u_0 and v_0 represent the increments of the rigid horizontal and vertical translations, respectively; r_0 denotes the increment of the rigid rotation angle around the central point (x_0, y_0) of block Ω; and (ε_x, ε_y, γ_{xy}) denotes the increments of the constant strains. Under small deformation assumption, the increments of the displacement $\boldsymbol{u}$ = $(u, v)^T$ at a point $\boldsymbol{x}$ = (x, y) in block Ω are computed as

$$\boldsymbol{u} = \boldsymbol{T}\boldsymbol{d} \tag{11}$$

and $\boldsymbol{T}$ is the translation matrix as follows:

$$\boldsymbol{T} = \begin{bmatrix} 1 & 0 & -(y-y_o) & (x-x_o) & 0 & 0.5(y-y_o) \\ 0 & 1 & (x-x_o) & 0 & (y-y_o) & 0.5(x-x_o) \end{bmatrix} \tag{12}$$

Equation (11) is a variation of a standard first-order displacement function [1].

The displacement caused by rigid movement and the deformation of the block constitute the displacement of a block under small deformation assumption. Therefore, the displacement function for finite rotation should be

$$\begin{aligned} u &= u_0 + (x-x_0)(\cos r_0 - 1) - (y-y_0)\sin r_0 + (x-x_0)\varepsilon_x + 0.5(y-y_0)\gamma_{xy} \\ v &= v_0 + (x-x_0)\sin r_0 + (y-y_0)(\cos r_0 - 1) + (y-y_0)\varepsilon_y + 0.5(x-x_0)\gamma_{xy} \end{aligned} \tag{13}$$

Comparing Equation (13) with Equation (11), the approximations of cos r_0 = 1 and sin r_0 = r_0 are adopted for the small rotation angle r_0 in Equation (11). The approximation errors accumulated over time steps can lead to false volume expansion when a large rotation occurs. Various modifications were suggested to remedy this defect, in which the post-adjustment strategy is most popular in the existing DDA codes. After vector $\boldsymbol{d}$ is obtained, the post-adjustment strategy employs Equation (13) to calculate the incremental displacement $\boldsymbol{u}$. Although simple and effective in most cases, the resulting displacement must be different from the displacement estimated in the equilibrium equation. Sometimes, for a contact pair, the state adopted in the equilibrium equation may be not coincident with the geometry relationship of the resulted configurations. An example is provided in Section 5.1 to demonstrate this issue.

Besides that, it is noticed that DDA was coupled with a range of numerical methods for continuum models to simulate the continuous–discontinuous deformation, e.g., crack propagation [9,22]. Most popular numerical methods in terms of continuum mechanics select the displacements at the nodes of an element as the degrees of freedom. DDA defines the degrees of freedom using the constant strain, the rigid translation, and the rigid rotation referring to the centroid of a block. The difference in the degrees of freedom causes a barrier in developing a compact code when coupling DDA to other numerical tools.

3.2. Previous Attempts to Construct Vertex Displacement-Based DDA

By taking the increments of the displacements at the vertices as the degrees of freedom, vertex displacement-based DDA can remedy the demerits induced by the original degrees of freedom. Supposing that a block Ω has n vertices, the incremental displacements $\boldsymbol{u}_i$= $(u_i, v_i)^T$ at the vertex $\boldsymbol{x}_i$ constitute the new degrees of freedom:

$$\bar{\boldsymbol{d}} = \begin{pmatrix} u_1 & v_1 & u_2 & v_2 & \ldots & u_n & v_n \end{pmatrix}^T \tag{14}$$

In this way, the penetration value of a contact pair in the equilibrium equations is directly calculated within a time step and the coordinates of the vertices are updated by directly adding the incremental displacements to their previous coordinates, inducing penetrations in the equilibrium equation and allowing the updated configurations to coincide.

By using the new degrees of freedom, the increment $\boldsymbol{u}$ of the displacement at a point $\boldsymbol{x}$ in a block Ω can be computed in a similar method to FEM, that is

$$\boldsymbol{u} = \begin{pmatrix} u \\ v \end{pmatrix} = \begin{pmatrix} N_1 & 0 & N_2 & 0 & \ldots & \ldots & N_n & 0 \\ 0 & N_1 & 0 & N_2 & \ldots & \ldots & 0 & N_n \end{pmatrix} \bar{\boldsymbol{d}} = \boldsymbol{N}\bar{\boldsymbol{d}} \tag{15}$$

where N_i is the shape function in terms of the vertex $\boldsymbol{x}_i$. Because the block in DDA might have more than four vertices, the shape functions developed in PFEM were introduced to construct N_i [36]. In the recent study [36], Wachspress interpolation was selected as the shape function for developing vertex displacement-based DDA.

Sukumar and MalsCh concluded the properties required for N_i to develop an effective PFEM [38]. As shown in Figure 2, a block Ω has n vertices located at $\boldsymbol{x}_i$ and its boundary is composed of n straight edges. The red edges with vertex $\boldsymbol{x}_i$ are denoted by Γ_i. N_i should be satisfied.

(1) Partition of unity, boundedness, and nonnegativity:

$$\sum_{i=1}^{n} N_i(\boldsymbol{x}) = 1, 0 \le N_i(\boldsymbol{x}) \le 1 \tag{16}$$

(2) Kronecker-delta property:

$$N_i(\boldsymbol{x}_j) = \delta_{ij} \tag{17}$$

(3) Linear completeness:

$$\sum_{i=1}^{n} N_i(\boldsymbol{x})\, \boldsymbol{x}_i = \boldsymbol{x} \tag{18}$$

(4) With C^∞ being within block Ω while C^0 is on the boundary, N_i must be piece-wise linear along Γ_i but vanishes at the other edges. This property ensures that the boundary of a straight line is still a straight line after deformation, which benefits contact detection and contact condition imposition.

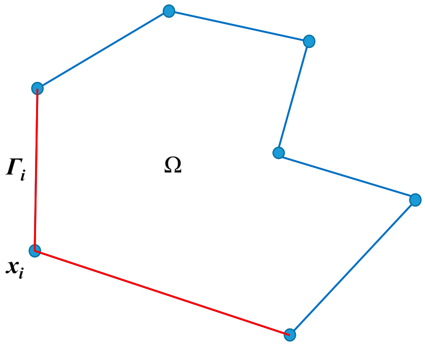

Figure 2. A polygon and its boundary division.

In the last two decades, generalized barycentric coordinates were successfully adopted as the shape function in PFEM. A series of generalized barycentric coordinates, such as Wachspress interpolation [37], maximum entropy interpolation [38], mean value coordinates [39] and Harmonic coordinates [40], were developed, marking remarkable progress in the theory and application of PFEM.

Take Wachspress interpolation for example; the shape function N_i is defined for a point $\boldsymbol{x}$ within a polygon as follows.

$$N_i(\boldsymbol{x}) = \frac{A(\boldsymbol{x}_{i-1}, \boldsymbol{x}_i, \boldsymbol{x}_{i+1}) \left(\prod\limits_{k \neq i, i-1} A(\boldsymbol{x}, \boldsymbol{x}_k, \boldsymbol{x}_{k+1}) \right)}{\sum\limits_{j=1}^{n} A(\boldsymbol{x}_{j-1}, \boldsymbol{x}_j, \boldsymbol{x}_{j+1}) \left(\prod\limits_{k \neq j, j-1} A(\boldsymbol{x}, \boldsymbol{x}_k, \boldsymbol{x}_{k+1}) \right)} \tag{19}$$

where $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_\mathrm{n}$ are the vertices of the polygon arranged counterclockwise and $A(\boldsymbol{x}_\mathrm{i}, \boldsymbol{x}_\mathrm{j}, \boldsymbol{x}_\mathrm{k})$ represents the area of a triangle connecting three points, such as in Figure 3. By observation, the basic principles required for N_i are completely satisfied by Definition (19).

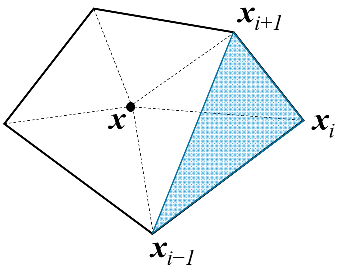

Figure 3. Triangles involved in the calculation of $\mathrm{N_i}$ using Wachspress interpolation.

However, the definitions of N_i in PFEM are the rational functions, which make the derivation and integration process more cumbersome when computing the block matrices. Though some enhancements [38] were made to simplify the formulas of N_i, the rational function is indispensable to definite N_i, which creates difficulties in the integration over a polygon. Up until now, the most popular way to solve this issue was partitioning the polygon into subdomains with certain shapes. Then, the Hammer or Gauss integration scheme was employed to compute the integration on each sub-triangle or sub-rectangle.

The summation of the integrations on subdomains was considered the integration over the original polygon. While the integration error was considerable, some remedies were proposed to address the integration error for PFEM [42]. In conclusion, the additional work in the derivation and programming is very expensive when using the interpolation function N_i in PFEM. When Equation (15) is adopted to govern the block displacement, some advantages of the original DDA can compromised.

4. Vertex Displacement-Based DDA Using VEM

4.1. Virtual Element Spaces

VEM [43–46] provides a new way to construct a vertex displacement-based DDA. Because DDA blocks are physically isolated, an individual virtual element space $\boldsymbol{V}_k(\Omega)$ can be defined for a block as follows:

$$\boldsymbol{V}_{\mathrm{k}}(\Omega) = \left\{ \boldsymbol{v} \in \left[H^1(\Omega) \cap C^0(\partial\Omega) \right]^2 : \Delta\boldsymbol{v} \in [P_{k-2}(\Omega)]^2, \boldsymbol{v}_{|\Gamma} \in [P_k(\Gamma)]^2 \forall\Gamma \in \partial\Omega \right\}, \tag{20}$$

in which Γ denotes a edge of Ω, $k \geq 1$ is a fixed integer index indicating the order of accuracy of the approach, and $P_k(\Omega)$ is the space of polynomials of degree less than or equal to k in Ω. Obviously, $[P_k(\Omega)]^2 \subseteq \boldsymbol{V}_{\mathrm{k}}(\Omega)$. For simplicity, $[P_k(\Omega)]^2$ is denoted as $\boldsymbol{P}_{\mathrm{k}}(\Omega)$.

We take $k = 1$ in this study. There are three main reasons for this choice: (i) The degrees of freedom for $\boldsymbol{V}_1(\Omega)$ only involve the values of $\boldsymbol{v}$ at the vertices of Ω, in accordance with the new degrees of freedom defined in Equation (14). (ii) The incremental displacements are expected to be piece-wise linear at the block boundary, which is satisfied by $\boldsymbol{V}_1(\Omega)$. (iii) In the view that the original DDA adopted a complete first-order displacement function, the degree of accuracy is not threatened by using $\boldsymbol{V}_1(\Omega)$. Therefore, an individual virtual element space $\boldsymbol{V}_1(\Omega)$ is defined for a block, and the associated degrees of freedom are the new degrees of freedom defined in Equation (14). The dimension of the space $\boldsymbol{V}_1(\Omega)$ is 2n, where n denotes the number of vertices for the block.

Provided that there are $\mathrm{n_b}$ blocks in the block system, the total virtual element space can be given as

$$V_1 = \boldsymbol{V}_1^1\left(\Omega^1\right) \times \boldsymbol{V}_1^2\left(\Omega^2\right) \times \ldots \times \boldsymbol{V}_1^{n_b}(\Omega^{n_b}), \tag{21}$$

and the total degrees of freedom comprise the increments of displacements at the vertices of all blocks. As a result, the solution space $\boldsymbol{V}$ in Equation (1) is replaced by $\boldsymbol{V}_1$.

4.2. The Projection Operator $\Pi_P\boldsymbol{u}$: $\boldsymbol{V}_1(\Omega) \rightarrow \boldsymbol{P}_1(\Omega)$

To minimize the total potential energy of the block system, the computation of α $(\boldsymbol{u}, \boldsymbol{v})$ from $\boldsymbol{V}_1(\Omega) \times \boldsymbol{V}_1(\Omega)$ to R defined in Equation (4) is inevitable, which can be accomplished in the framework of VEM theory. Referring to [47], two projectors $\Pi^\nabla\boldsymbol{u}$: $\boldsymbol{V}_1(\Omega) \rightarrow \boldsymbol{P}_1(\Omega)$ and $\Pi^0\boldsymbol{u}$: $\boldsymbol{V}_1(\Omega) \rightarrow \boldsymbol{P}_1(\Omega)$ are respectively defined using the orthogonality conditions:

$$\int_\Omega \nabla\left(\Pi^\nabla\boldsymbol{u}\right) \cdot \nabla\boldsymbol{p}\, \mathrm{dx} = \int_\Omega \nabla\boldsymbol{u} \cdot \nabla\boldsymbol{p}\, \mathrm{dx}\ \forall\boldsymbol{p} \in \boldsymbol{P}_1(\Omega), \tag{22}$$

and

$$\int_\Omega \left(\Pi^0\boldsymbol{u}\right) \cdot \boldsymbol{p}\, \mathrm{dx} = \int_\Omega \boldsymbol{u} \cdot \boldsymbol{p}\, \mathrm{dx}\ \forall\boldsymbol{p} \in \boldsymbol{P}_1(\Omega). \tag{23}$$

where $\Pi^0\boldsymbol{u}$ cannot be obtained from (23) because the zero- and first-order moments of $\boldsymbol{u}$ in Ω are unknown. To solve this issue, an additional property stating that the moments of order 0 and 1 of $\boldsymbol{u}$ and $\Pi^\nabla\boldsymbol{u}$ coincide was added by B. Ahmad [44] to the space $\boldsymbol{V}_1(\Omega)$, resulting in the projection operator of $\Pi^0\boldsymbol{u}$ being the same as the projection operator of $\Pi^\nabla\boldsymbol{u}$. For simplicity, $\Pi_{\mathrm{P}}\boldsymbol{u}$ is adopted to express $\Pi^\nabla\boldsymbol{u}$ and $\Pi^0\boldsymbol{u}$ uniformly. In this study, $\Pi_{\mathrm{P}}\boldsymbol{u}$ is deduced using an approach different from the standard procedure [46].

Beforehand, two notations are defined. For a function $\boldsymbol{w}$, the average of the values it assumes at the vertices of Ω is denoted by $\boldsymbol{w}^*$:

$$\boldsymbol{w}^* = \frac{1}{n}\sum_{i=1}^{n} \boldsymbol{w}(\boldsymbol{x}_i), \tag{24}$$

and the volume average of $\boldsymbol{w}$ over Ω is denoted by $\langle \boldsymbol{w} \rangle$:

$$\langle \boldsymbol{w} \rangle = \frac{1}{S}\int_{\Omega} \boldsymbol{w}\, \mathrm{d}\boldsymbol{x}. \tag{25}$$

where S is the area of Ω.

$\boldsymbol{P}_1(\Omega)$ is the space of linear displacements over Ω, which can be split into two displacement spaces due to rigid motions and constant strain. We denote the space regarding rigid motions by $\boldsymbol{R}(\Omega)$ and the space concerning constant strain by $\boldsymbol{C}(\Omega)$:

$$\boldsymbol{R}(\Omega) = \left\{ \boldsymbol{a} + \boldsymbol{B}_R(\boldsymbol{x} - \boldsymbol{x}^*) : \boldsymbol{a} \in \mathrm{R}^2, \boldsymbol{B}_R \in \mathrm{R}^{2\times 2}, \boldsymbol{B}_R^T = -\boldsymbol{B}_R \right\}, \tag{26}$$

$$\boldsymbol{C}(\Omega) = \left\{ \boldsymbol{B}_C(\boldsymbol{x} - \boldsymbol{x}^*) : \boldsymbol{B}_C \in \mathrm{R}^{2\times 2}, \boldsymbol{B}_C^T = \boldsymbol{B}_C \right\}. \tag{27}$$

Obviously, $\boldsymbol{P}_1(\Omega)$ is the summation of $\boldsymbol{R}(\Omega)$ and $\boldsymbol{C}(\Omega)$. Both $\boldsymbol{R}(\Omega)$ and $\boldsymbol{C}(\Omega)$ are subspaces of $\boldsymbol{V}_1(\Omega)$. Two projection maps, Π_{R}: $\boldsymbol{V}_1(\Omega) \rightarrow \boldsymbol{R}(\Omega)$ and Π_{C}: $\boldsymbol{V}_1(\Omega) \rightarrow \boldsymbol{C}(\Omega)$, are defined to extract the rigid motions and constant strain of $\boldsymbol{u} \in \boldsymbol{V}_1(\Omega)$. Considering that elements of $\boldsymbol{C}(\Omega)$ should contain no rigid motion and that elements of $\boldsymbol{R}(\Omega)$ should contain no constant strain, the following orthogonality conditions are imposed on the two maps:

$$\Pi_R \boldsymbol{c} = 0, \forall \boldsymbol{c} \in \boldsymbol{C}(\Omega), \tag{28}$$

$$\Pi_C \boldsymbol{r} = 0, \forall \boldsymbol{r} \in \boldsymbol{R}(\Omega). \tag{29}$$

Gain et al. [48] proved that projection maps Π_{R} and Π_{C} satisfying the above properties can be given by

$$\Pi_R \boldsymbol{u} = \boldsymbol{u}^* + \langle \boldsymbol{\omega}(\boldsymbol{u}) \rangle (\boldsymbol{x} - \boldsymbol{x}^*) \text{ with } \boldsymbol{\omega}(\boldsymbol{u}) = \frac{1}{2}\left(\nabla \boldsymbol{u} - \nabla^T \boldsymbol{u} \right) \tag{30}$$

and

$$\Pi_C \boldsymbol{u} = \langle \varepsilon(\boldsymbol{u}) \rangle (\boldsymbol{x} - \boldsymbol{x}^*) \tag{31}$$

Due to the orthogonality conditions in Equations (28) and (29), we have

$$\Pi_P \boldsymbol{u} = \Pi_R \boldsymbol{u} + \Pi_C \boldsymbol{u} = \boldsymbol{u}^* + \langle \nabla \boldsymbol{u} \rangle (\boldsymbol{x} - \boldsymbol{x}^*). \tag{32}$$

It is noticed that

$$\int_{\Omega} \nabla (\Pi_P \boldsymbol{u})\, \mathrm{d}\boldsymbol{x} = \int_{\Omega} \langle \nabla \boldsymbol{u} \rangle\, \mathrm{d}\boldsymbol{x} = \int_{\Omega} \nabla \boldsymbol{u}\, \mathrm{d}\boldsymbol{x}, \tag{33}$$

and that $\nabla \boldsymbol{p}$ in Equation (22) contains only constant elements**.** Obviously, the projection map $\Pi_{\mathrm{P}}\boldsymbol{u}$ given by Equation (32) satisfies Equation (22).

Using the integration by parts, the areal integrals in Equation (32) can easily be converted to the boundary integrals, and then, it can be computed precisely because $\boldsymbol{u} = (u, v)$ is linear on each edge of Ω. Denoting the length of edge i connecting vertex $i-1$

and vertex i as l_i, as Figure 4 shows, and the outward unit normal vector on edge i as $\boldsymbol{n}_i = (n_{ix}, n_{iy})^T$, $\langle \nabla \boldsymbol{u} \rangle$ can be computed as

$$\begin{aligned} \langle \nabla \boldsymbol{u} \rangle &= \tfrac{1}{S}\int_{\Omega} \nabla \boldsymbol{u}\, \mathrm{d}\boldsymbol{x} = \tfrac{1}{S}\begin{bmatrix} \int_{\partial\Omega} u n_x\, \mathrm{ds} & \int_{\partial\Omega} u n_y\, \mathrm{ds} \\ \int_{\partial\Omega} v n_x\, \mathrm{ds} & \int_{\partial\Omega} v n_y\, \mathrm{ds} \end{bmatrix} \\ &= \tfrac{1}{S}\begin{bmatrix} \sum\limits_{i=1}^{n} \frac{l_i n_{ix} + l_{i+1} n_{(i+1)x}}{2} u_i & \sum\limits_{i=1}^{n} \frac{l_i n_{iy} + l_{i+1} n_{(i+1)y}}{2} u_i \\ \sum\limits_{i=1}^{n} \frac{l_i n_{ix} + l_{i+1} n_{(i+1)x}}{2} v_i & \sum\limits_{i=1}^{n} \frac{l_i n_{iy} + l_{i+1} n_{(i+1)y}}{2} v_i \end{bmatrix} \end{aligned} \tag{34}$$

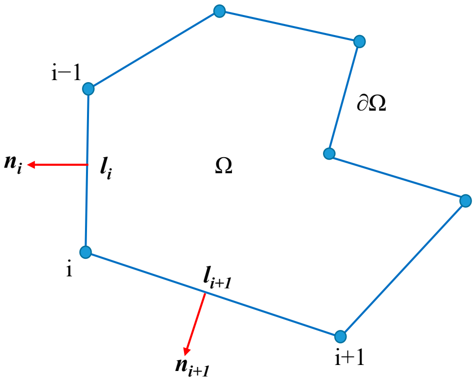

Figure 4. The notation li and $\boldsymbol{n}_i$ in $\boldsymbol{H}$.

By substituting Equation (34) into Equation (32), we have

$$\Pi_P \boldsymbol{u} = \frac{1}{n}\begin{pmatrix} \sum\limits_{i=1}^{n} u_i \\ \sum\limits_{i=1}^{n} v_i \end{pmatrix} + \frac{1}{S}\begin{bmatrix} \sum\limits_{i=1}^{n} \frac{l_i n_{ix} + l_{i+1} n_{(i+1)x}}{2} u_i & \sum\limits_{i=1}^{n} \frac{l_i n_{iy} + l_{i+1} n_{(i+1)y}}{2} u_i \\ \sum\limits_{i=1}^{n} \frac{l_i n_{ix} + l_{i+1} n_{(i+1)x}}{2} v_i & \sum\limits_{i=1}^{n} \frac{l_i n_{iy} + l_{i+1} n_{(i+1)y}}{2} v_i \end{bmatrix} \begin{pmatrix} x - x^* \\ y - y^* \end{pmatrix} \tag{35}$$

For ease of subsequent performance, another matrix expression of $\Pi_P \boldsymbol{u}$ is given as

$$\Pi_P \boldsymbol{u} = \boldsymbol{T}^* \boldsymbol{H} \bar{\boldsymbol{d}}, \tag{36}$$

in which

$$\boldsymbol{T}^* = \begin{bmatrix} 1 & 0 & -(y-y^*) & (x-x^*) & 0 & 0.5(y-y^*) \\ 0 & 1 & (x-x^*) & 0 & (y-y^*) & 0.5(x-x^*) \end{bmatrix} \tag{37}$$

and

$$\boldsymbol{H} = \begin{pmatrix} \frac{1}{n} & 0 & \frac{1}{n} & 0 & \cdots & \frac{1}{n} & 0 \\ 0 & \frac{1}{n} & 0 & \frac{1}{n} & \cdots & 0 & \frac{1}{n} \\ \frac{l_1 n_{1y} + l_2 n_{2y}}{-4S} & \frac{l_1 n_{1x} + l_2 n_{2x}}{4S} & \frac{l_2 n_{2y} + l_3 n_{3y}}{-4S} & \frac{l_2 n_{2x} + l_3 n_{3x}}{4S} & \cdots & \frac{l_n n_{ny} + l_1 n_{1y}}{-4S} & \frac{l_n n_{nx} + l_1 n_{1x}}{4S} \\ \frac{l_1 n_{1x} + l_2 n_{2x}}{2S} & 0 & \frac{l_2 n_{2x} + l_3 n_{3x}}{2S} & 0 & \cdots & \frac{l_n n_{nx} + l_1 n_{1x}}{2S} & 0 \\ 0 & \frac{l_1 n_{1y} + l_2 n_{2y}}{2S} & 0 & \frac{l_2 n_{2y} + l_3 n_{3y}}{2S} & \cdots & 0 & \frac{l_n n_{ny} + l_1 n_{1y}}{2S} \\ \frac{l_1 n_{1y} + l_2 n_{2y}}{2S} & \frac{l_1 n_{1x} + l_2 n_{2x}}{2S} & \frac{l_2 n_{2y} + l_3 n_{3y}}{2S} & \frac{l_2 n_{2x} + l_3 n_{3x}}{2S} & \cdots & \frac{l_n n_{ny} + l_1 n_{1y}}{2S} & \frac{l_n n_{nx} + l_1 n_{1x}}{2S} \end{pmatrix}. \tag{38}$$

Due to the orthogonality conditions in Equations (28) and (29), the constant strain tensor $\boldsymbol{\varepsilon}(\Pi_{\mathrm{P}}\boldsymbol{u})$ can be computed as follows:

$$\boldsymbol{\varepsilon}(\Pi_P\boldsymbol{u}) = \boldsymbol{\varepsilon}(\Pi_C\boldsymbol{u}) = \langle\boldsymbol{\varepsilon}(\boldsymbol{u})\rangle$$

$$= \frac{1}{S}\begin{bmatrix} \sum\limits_{i=1}^{n} \frac{l_i n_{ix}+l_{i+1}n_{(i+1)x}}{2}u_i & \sum\limits_{i=1}^{n}\left(\frac{l_i n_{iy}+l_{i+1}n_{(i+1)y}}{4}u_i + + \frac{l_i n_{ix}+l_{i+1}n_{(i+1)x}}{4}v_i\right) \\ \sum\limits_{i=1}^{n}\left(\frac{l_i n_{iy}+l_{i+1}n_{(i+1)y}}{4}u_i + + \frac{l_i n_{ix}+l_{i+1}n_{(i+1)x}}{4}v_i\right) & \sum\limits_{i=1}^{n} \frac{l_i n_{iy}+l_{i+1}n_{(i+1)y}}{2}v_i \end{bmatrix} \tag{39}$$

and a matrix expression of the constant strain vector $\boldsymbol{\varepsilon}$ = (ε_x, ε_{y}, γ_{xy}) can be given as

$$\boldsymbol{\varepsilon}(\Pi_P\boldsymbol{u}) = \frac{1}{S}\begin{bmatrix} \sum\limits_{i=1}^{n} \frac{l_i n_{ix}+l_{i+1}n_{(i+1)x}}{2}u_i \\ \sum\limits_{i=1}^{n} \frac{l_i n_{iy}+l_{i+1}n_{(i+1)y}}{2}v_i \\ \sum\limits_{i=1}^{n}\left(\frac{l_i n_{iy}+l_{i+1}n_{(i+1)y}}{2}u_i + + \frac{l_i n_{ix}+l_{i+1}n_{(i+1)x}}{2}v_i\right) \end{bmatrix} = \boldsymbol{P}_C\bar{\boldsymbol{d}} \tag{40}$$

in which

$$\boldsymbol{P}_c = \begin{pmatrix} \frac{l_1 n_{1x}+l_2 n_{2x}}{2S} & 0 & \frac{l_2 n_{2x}+l_3 n_{3x}}{2S} & 0 & \cdots & \frac{l_n n_{nx}+l_1 n_{1x}}{2S} & 0 \\ 0 & \frac{l_1 n_{1y}+l_2 n_{2y}}{2S} & 0 & \frac{l_2 n_{2y}+l_3 n_{3y}}{2S} & \cdots & 0 & \frac{l_n n_{ny}+l_1 n_{1y}}{2S} \\ \frac{l_1 n_{1y}+l_2 n_{2y}}{2S} & \frac{l_1 n_{1x}+l_2 n_{2x}}{2S} & \frac{l_2 n_{2y}+l_3 n_{3y}}{2S} & \frac{l_2 n_{2x}+l_3 n_{3x}}{2S} & \cdots & \frac{l_n n_{ny}+l_1 n_{1y}}{2S} & \frac{l_n n_{nx}+l_1 n_{1x}}{2S} \end{pmatrix}. \tag{41}$$

Obviously, $\boldsymbol{P}_{\mathrm{c}}$ is actually composed by the latter three rows in $\boldsymbol{H}$.

An element $\boldsymbol{u}\in\boldsymbol{V}_1(\Omega)$ can be decomposed into $\Pi_{\mathrm{P}}\boldsymbol{u}$ and the residual item $\boldsymbol{u}-\Pi_{\mathrm{P}}\boldsymbol{u}$. Then, a bilinear form from $\boldsymbol{V}_1(\Omega) \times \boldsymbol{V}_1(\Omega)$ to R can be analyzed without the explicit expression of $\boldsymbol{u}$, which will be demonstrated in the next section.

4.3. Computation of $\alpha(\boldsymbol{u}, \boldsymbol{v})$ and $f(\boldsymbol{v})$

For simplicity of expression, α ($\boldsymbol{u}$, $\boldsymbol{v}$) in Equation (4) and $f(\boldsymbol{v})$ in Equation (8) are rewritten as

$$\alpha(\boldsymbol{u},\boldsymbol{v}) = \alpha^E + \alpha^M - \alpha^V, f(\boldsymbol{v}) = f^b + f^t - f^\sigma \tag{42}$$

in which

$$\alpha^E = \int_\Omega \boldsymbol{\varepsilon}^T(\boldsymbol{v})\boldsymbol{\sigma}(\boldsymbol{u})\,\mathrm{d}\boldsymbol{x},\ \alpha^M = \frac{2\rho}{\Delta^2}\int_\Omega \boldsymbol{v}^T\boldsymbol{u}\,\mathrm{d}\boldsymbol{x},\ \alpha^V = \frac{2\rho}{\Delta}\int_\Omega \boldsymbol{v}^T\boldsymbol{V}_0\,\mathrm{d}\boldsymbol{x} \tag{43}$$

and

$$f^b = \int_\Omega \boldsymbol{v}^T\boldsymbol{b}\,\mathrm{d}\boldsymbol{x},\ f^t = \int_{\partial\Omega}\boldsymbol{v}^T\boldsymbol{t}\mathrm{d}s,\ f^\sigma = \int_\Omega \boldsymbol{\varepsilon}^T(\boldsymbol{v})\sigma_0\mathrm{d}\boldsymbol{x} \tag{44}$$

It is noticed that, in this study, $\boldsymbol{V}_0$ is considered an element in $\boldsymbol{V}_1(\Omega)$ to estimate $\boldsymbol{V}_0$ within the block because $\boldsymbol{V}_0$ at the block vertices are directly related to $\boldsymbol{u}$. Consequently, α^V is a bilinear form. $\boldsymbol{V}_0$ can also be evaluated using the rigid body motion and the constant strain associated with $\Pi_{\mathrm{P}}\boldsymbol{u}$. That will induce $\boldsymbol{V}_0$ to become an element in $\boldsymbol{P}_{\boldsymbol{1}}(\Omega)$ and α^V to become a linear form.

The bilinear forms are formulated first. Due to the orthogonality conditions (22) and (23), any bilinear form $\alpha(\boldsymbol{u}, \boldsymbol{v})$ regarding two elements $\boldsymbol{u}$, $\boldsymbol{v}\in\boldsymbol{V}_1(\Omega)$ can be rewritten as

$$\alpha(\boldsymbol{u},\boldsymbol{v}) = a(\Pi_P\boldsymbol{u},\Pi_P\boldsymbol{v}) + a(\boldsymbol{u}-\Pi_P\boldsymbol{u},\boldsymbol{v}-\Pi_P\boldsymbol{v}). \tag{45}$$

On the right side of Equation (45), the first term named the consistency term can be precisely calculated while the second term called the stabilization term represents the contribution of the residual item $\boldsymbol{u}-\Pi_{\mathrm{P}}\boldsymbol{u}$ to the bilinear form. The stabilization term cannot produce an exact solution, so the second term in VEM is usually denoted as $s(\boldsymbol{u}-\Pi_{\mathrm{P}}\boldsymbol{u}$,

$\boldsymbol{v}-\Pi_{\mathrm{P}}\boldsymbol{v}$) to indicate that it is an approximation. The standard scheme of $s(\boldsymbol{u}-\Pi_{\mathrm{P}}\boldsymbol{u}, \boldsymbol{v}-\Pi_{\mathrm{P}}\boldsymbol{v})$ was provided in classical VEM literature [46], but it is quite free in the construction of $s(\boldsymbol{u}-\Pi_{\mathrm{P}}\boldsymbol{u}, \boldsymbol{v}-\Pi_{\mathrm{P}}\boldsymbol{v})$ in practice [48,49]. The stabilization term proposed by Veiga [46] is adopted in this study:

$$s(\boldsymbol{u}-\Pi_P\boldsymbol{u}, \boldsymbol{v}-\Pi_P\boldsymbol{v}) = \sum_{i=1}^{n} \eta[\boldsymbol{u}(\boldsymbol{x}_i) - \Pi_P\boldsymbol{u}(\boldsymbol{x}_i)][\boldsymbol{v}(\boldsymbol{x}_i) - \Pi_P\boldsymbol{v}(\boldsymbol{x}_i)] \tag{46}$$

where η is a positive parameter ensuring the right scaling of the bilinear form assigned to the residual item.

Using Equation (45), the bilinear form α^E is rewritten as follows:

$$\alpha^E(\boldsymbol{u}, \boldsymbol{u}) = \alpha^E(\Pi_P\boldsymbol{u}, \Pi_P\boldsymbol{u}) + s^E(\boldsymbol{u}-\Pi_P\boldsymbol{u}, \boldsymbol{u}-\Pi_P\boldsymbol{u}) \tag{47}$$

Substituting the matrix expression (40) regarding $\boldsymbol{\varepsilon}(\Pi_{\mathrm{P}}\boldsymbol{u})$ into Equation (47), the consistency term is

$$\alpha^E(\Pi_P\boldsymbol{u}, \Pi_P\boldsymbol{u}) = \int_{\Omega} \boldsymbol{\varepsilon}^T(\Pi_P\boldsymbol{u})\boldsymbol{\sigma}(\Pi\boldsymbol{u})\, \mathrm{d}\boldsymbol{x} = \bar{\boldsymbol{d}}^T \left[\boldsymbol{S}\boldsymbol{P}_c^T\boldsymbol{D}\boldsymbol{P}_c \right] \bar{\boldsymbol{d}} \tag{48}$$

in which $\boldsymbol{D}$ is the elastic matrix

The stabilization term in Equation (47) is

$$s^E(\boldsymbol{u}-\Pi_P\boldsymbol{u}, \boldsymbol{u}-\Pi_P\boldsymbol{u}) = \sum_{i=1}^{n} \eta^E[\boldsymbol{u}(\boldsymbol{x}_i) - \Pi_P\boldsymbol{u}(\boldsymbol{x}_i)]^2 = \bar{\boldsymbol{d}}^T \left[\eta^E(\boldsymbol{I}-\boldsymbol{P}_u)^T(\boldsymbol{I}-\boldsymbol{P}_u) \right] \bar{\boldsymbol{d}} \tag{49}$$

where $\boldsymbol{I}$ is a $2n \times 2n$ unit matrix and $\boldsymbol{P}_u$ is a $2n \times 2n$ constant matrix:

$$\boldsymbol{P}_u = \begin{bmatrix} \boldsymbol{T}^*(\boldsymbol{x}_1) \\ \boldsymbol{T}^*(\boldsymbol{x}_2) \\ \vdots \\ \boldsymbol{T}^*(\boldsymbol{x}_n) \end{bmatrix} \boldsymbol{H} \tag{50}$$

In the view that s^{E} is the approximation of α^E, the positive parameters η^E can be decided upon by requiring that s^{E} and α^E are comparable. The consistency term $\alpha^E(\Pi_{\mathrm{P}}\boldsymbol{u}, \Pi_{\mathrm{P}}\boldsymbol{u})$ can be estimated using s^{E} as follows:

$$s^E(\Pi_P\boldsymbol{u}, \Pi_P\boldsymbol{u}) = \bar{\boldsymbol{d}}^T \left[\eta^E \boldsymbol{P}_{\boldsymbol{u}}^T\boldsymbol{P}_{\boldsymbol{u}} \right] \bar{\boldsymbol{d}} \tag{51}$$

The exact solution of $\alpha^E(\Pi_{\mathrm{P}}\boldsymbol{u}, \Pi_{\mathrm{P}}\boldsymbol{u})$ is given in Equation (48). Equating the traces of the two matrices, η^{E} is determined:

$$\eta^E = \frac{trace(\boldsymbol{S}\boldsymbol{P}_c^T\boldsymbol{D}\boldsymbol{P}_c)}{trace(\boldsymbol{P}_{\boldsymbol{u}}^T\boldsymbol{P}_u)} \tag{52}$$

Combining Equations (48) and (49), we have

$$\alpha^E(\boldsymbol{u}, \boldsymbol{u}) = \bar{\boldsymbol{d}}^T \left[\boldsymbol{S}\boldsymbol{P}_c^T\boldsymbol{D}\boldsymbol{P}_c \right] \bar{\boldsymbol{d}} + \bar{\boldsymbol{d}}^T \left[\eta^E(\boldsymbol{I}-\boldsymbol{P}_u)^T(\boldsymbol{I}-\boldsymbol{P}_u) \right] \bar{\boldsymbol{d}} \tag{53}$$

Mimicking what we did for the bilinear form α^E, the bilinear forms α^M and α^V are formulated as

$$\alpha^M(\boldsymbol{u}, \boldsymbol{u}) = \bar{\boldsymbol{d}}^T \left[\frac{2\rho}{\Delta^2} \boldsymbol{H}^T\boldsymbol{P}_m\boldsymbol{H} \right] \bar{\boldsymbol{d}} + \bar{\boldsymbol{d}}^T \left[\eta^M(\boldsymbol{I}-\boldsymbol{P}_u)^T(\boldsymbol{I}-\boldsymbol{P}_u) \right] \bar{\boldsymbol{d}} \tag{54}$$

and

$$\alpha^{V}(\boldsymbol{u},\boldsymbol{V}_0)=\overset{-T}{\boldsymbol{d}}\left[\frac{2\rho}{\Delta}\boldsymbol{H}^{T}\boldsymbol{P}_m\boldsymbol{H}\right]\boldsymbol{V}_0+\overset{-T}{\boldsymbol{d}}\left[\eta^{V}(\boldsymbol{I}-\boldsymbol{P}_{\boldsymbol{u}})^{T}(\boldsymbol{I}-\boldsymbol{P}_{\boldsymbol{u}})\right]\boldsymbol{V}_0, \tag{55}$$

respectively. Here, $\boldsymbol{P}_m$ is a $2n \times 2n$ constant matrix as

$$P_m=\iint\limits_{\Omega}(\boldsymbol{T}^{*})^{T}\boldsymbol{T}^{*}dxdy, \tag{56}$$

$$\eta^{M}=\frac{2\rho}{\Delta^{2}}\frac{trace(\boldsymbol{H}^{T}\boldsymbol{P}_m\boldsymbol{H})}{trace(\boldsymbol{P}_{\boldsymbol{u}}^{T}\boldsymbol{P}_{\boldsymbol{u}})} \tag{57}$$

and

$$\eta^{V}=\frac{2\rho}{\Delta}\frac{trace(\boldsymbol{H}^{T}\boldsymbol{P}_m\boldsymbol{H})}{trace(\boldsymbol{P}_{\boldsymbol{u}}^{T}\boldsymbol{P}_{\boldsymbol{u}})}. \tag{58}$$

The integrations in Equation (56) can be exactly calculated by using the simplex integration. Now, we have successfully formulated all bilinear forms in Equation (43).

Then, the linear forms f^{σ}, f^{b}, and f^{t} in Equation (44) are investigated. Obviously, f^{t} can be computed directly. Because $\boldsymbol{u}$ is linear on each edge of Ω, a definite shape function matrix $\boldsymbol{N}(\boldsymbol{x})$ can be easily determined for a point $\boldsymbol{x}$ on $\partial\Omega$ to compute $\boldsymbol{u}(\boldsymbol{x})$ using Equation (6). Thus, we have

$$f^{t}=\overset{-T}{\boldsymbol{d}}\int_{\partial\Omega}\boldsymbol{N}^{T}(\boldsymbol{x})t\,d\Gamma_t \tag{59}$$

Only the constant stress associated with $\Pi_{\mathrm{P}}\boldsymbol{u}$ can be valued in the first-order VEM, so $\boldsymbol{\sigma}_0$ is a constant stress in this study. Using the integration by parts, the linear form f^{σ} can be processed as

$$f^{\sigma}=\int_{\Omega}\boldsymbol{\varepsilon}^{T}(\boldsymbol{u})\boldsymbol{\sigma}_0\,\mathrm{d}\boldsymbol{x}=\overset{-T}{\boldsymbol{d}}\left(S\boldsymbol{P}_c^{T}\boldsymbol{\sigma}_0\right) \tag{60}$$

Treating the constant body force $\boldsymbol{b}(x,y)$ as an element in $\boldsymbol{P_1}(\Omega)$, we have

$$f^{b}=\int_{\Omega}\boldsymbol{u}^{T}\boldsymbol{b}\,\mathrm{d}\boldsymbol{x}=\int_{\Omega}(\Pi_{P}\boldsymbol{u})^{T}\boldsymbol{b}\,\mathrm{d}\boldsymbol{x}=\overset{-T}{\boldsymbol{d}}\left[\boldsymbol{H}^{T}\int_{\Omega}(\boldsymbol{T}^{*})^{T}\,\mathrm{d}\boldsymbol{x}\right]\boldsymbol{b}=\overset{-T}{\boldsymbol{d}}\left(\boldsymbol{L}^{b}\right)^{T}\boldsymbol{b} \tag{61}$$

in which $\boldsymbol{L}^{b}$ is a $2 \times 2\,n$ matrix as follows:

$$\boldsymbol{L}^{b}=\begin{bmatrix} L_1^{b} & 0 & L_2^{b} & 0 & \cdots & L_n^{b} & 0 \\ 0 & L_1^{b} & 0 & L_2^{b} & \cdots & 0 & L_n^{b} \end{bmatrix} \tag{62}$$

and

$$L_i^{b}=\frac{S}{n}+0.5\begin{pmatrix} l_i n_{ix}+l_{i+1}n_{(i+1)x} & l_i n_{iy}+l_{i+1}n_{(i+1)y} \end{pmatrix}\begin{pmatrix} x_0-x^{*} \\ y_0-y^{*} \end{pmatrix} \tag{63}$$

So far, all bilinear and linear forms in Equations (4) and (8) were formulated. The energy induced by the contact constraints are investigated in next section.

4.4. Computation of $\boldsymbol{J}'(v)$ Due to the Contact Constraints

In the block system, no tension and interpenetration is allowable when the collision of blocks occurs. Three contact states, “open”, “lock”, and “slide”, are defined in DDA. By adopting stiff springs, different contact constraints are imposed according to the state.

A typical contact pair that is composed by a vertex i of Ω_B and an entrance line jk of Ω_A is plotted in Figure 5a. Point o is the closest point from line jk to point i. At the start of a time step, the coordinates of points i, j, k, and o are (x_i, y_i), (x_j, y_j), (x_k, y_k), and (x_o, y_o), respectively. Within this time step, their displacements are (u_i, v_i), (u_j, v_j), (u_k, v_k), and (u_o, v_o), respectively. Then, the normal penetration $d_{\boldsymbol{n}}$ and tangential relative displacement $d_{\boldsymbol{\tau}}$ in Figure 5b can be measured using their new coordinates when the time step terminates.

If $d_{\boldsymbol{n}}$ has a negative value, the contact pair must be "open" and no contact constraint applies, which results in no spring being added into the block system. Otherwise, a normal stiffness spring is added to oppose the penetration, which results in a deformation energy $J^{\boldsymbol{n}}$. Suppose that the two blocks contact each other along a coincident edge. The Coulomb friction law is adopted to determine whether the tangential relative motion is permitted. If the tangential relative motion is inadmissible, the contact pair is "lock" and a stiffness spring along the tangential direction is introduced to resist the tangential relative motion, leading to a deformation energy $J^{\boldsymbol{\tau}}$. Otherwise, the contact pair is "slide" and a pair of friction forces is adopted along the tangential direction, inducing a potential energy J^f. The allowable upper bound of the Coulomb friction law is considered the value of the friction forces. For the block system, $\boldsymbol{J}'(\mathrm{v})$ in Equation (1) can be obtained by collecting $J^{\boldsymbol{n}}$, $J^{\boldsymbol{\tau}}$**,** and J^f of all contact pairs.

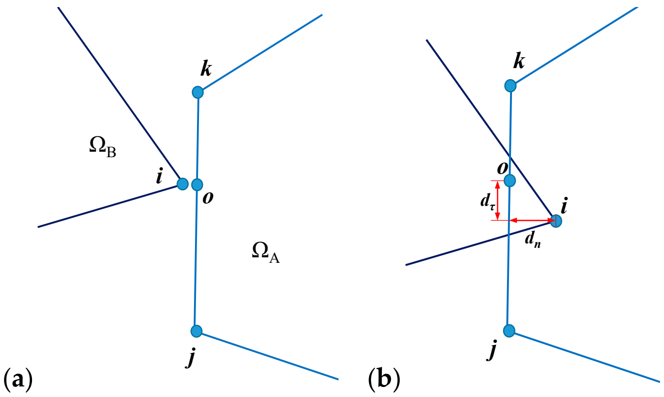

Figure 5. A typical contact pair at (**a**) the beginning and (**b**) the end of the current time step.

In practice, the states of all contact pairs have to be supposed in advance, and then, the "open-close" iteration is executed to guarantee the validity of the states. First, assemble the global equations based on the assumed contact states and obtain an interim displacement solution. Next, check the states of the contact pairs by using the interim displacement solution and alter the previously supposed states. Modify the contact constraints concerning every contact pair. Then, renew and resolve the global equations to obtain an updated interim displacement solution. Finally, stop the iteration when the actual states of all contact pairs agree with the supposed states and acquire the ultimate displacement solution.

Firstly, considering that the displacement should be small in a time step, the normal penetration $d_{\boldsymbol{n}}$ has an approximation (61) by neglecting the second-order infinitesimal value:

$$d_{\boldsymbol{n}} = -\left[\frac{S_0}{l} + \frac{1}{l}\begin{pmatrix} y_j - y_k & x_k - x_j \end{pmatrix}\begin{pmatrix} u_i \\ v_i \end{pmatrix} + \frac{1}{l}\begin{pmatrix} y_k - y_i & x_i - x_k \end{pmatrix}\begin{pmatrix} u_j \\ v_j \end{pmatrix} + \frac{1}{l}\begin{pmatrix} y_i - y_j & x_j - x_i \end{pmatrix}\begin{pmatrix} u_k \\ v_k \end{pmatrix}\right] \tag{64}$$

in which

$$S_0 = \begin{vmatrix} 1 & x_i & y_i \\ 1 & x_j & y_j \\ 1 & x_k & y_k \end{vmatrix}, l = \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2} \tag{65}$$

Given that the stiffness of the normal spring is $\rho_{\boldsymbol{n}}$, the normal spring resisting the normal penetration induces the deformation energy as follows:

$$J^{\boldsymbol{n}} = 0.5\rho_{\boldsymbol{n}} d_{\boldsymbol{n}}^2 \tag{66}$$

Minimization of $J^{\boldsymbol{n}}$ causes three second-order vectors to be added to the global load vector and nine second-order square matrices to be added into the global stiffness matrix.

Then, similar to $d_{\boldsymbol{n}}$, the tangential relative displacement $d_{\boldsymbol{\tau}}$ can be formulated as

$$d_{\boldsymbol{\tau}} = \frac{\overline{S}_0}{l} + \frac{1}{l}\begin{pmatrix} x_k - x_j & y_k - y_j \end{pmatrix}\begin{pmatrix} u_i \\ v_i \end{pmatrix} + \frac{\mu}{l}\begin{pmatrix} x_j - x_k & y_j - y_k \end{pmatrix}\begin{pmatrix} u_j \\ v_j \end{pmatrix} + \frac{1-\mu}{l}\begin{pmatrix} x_j - x_k & y_j - y_k \end{pmatrix}\begin{pmatrix} u_k \\ v_k \end{pmatrix} \tag{67}$$

where

$$\overline{S}_0 = \begin{pmatrix} x_k - x_j & y_k - y_j \end{pmatrix} \begin{pmatrix} x_i - x_0 \\ y_i - y_0 \end{pmatrix}, \mu = \frac{1}{l}\left(\sqrt{(x_0 - x_k)^2 + (y_0 - y_k)^2}\right) \tag{68}$$

Provided that the stiffness of the tangential spring is $\rho_{\boldsymbol{\tau}}$, the tangential spring results in the deformation energy:

$$J^{\tau} = 0.5\rho_{\boldsymbol{\tau}} d_{\boldsymbol{\tau}}^2 \tag{69}$$

Minimization of J^{τ} also leads to three second-order vectors being added to the global load vector and nine second-order square matrices being added into the global stiffness matrix.

Finally, if the contact pair is "slide", a pair pf friction forces is imposed along the tangential direction instead of the tangential spring. The unit direction vector from point j to point k is

$$\boldsymbol{\tau} = \frac{1}{l}\begin{pmatrix} (x_k - x_j) & (y_k - y_j) \end{pmatrix}^T \tag{70}$$

The allowable upper bound of the Coulomb friction law is denoted by f_{up}. For $d_{\boldsymbol{\tau}}$ in Figure 5b, the friction force $f_{up}\boldsymbol{\tau}$ acts on Ω_B at point i and the friction force—$f_{up}\boldsymbol{\tau}$ acts on Ω_A at point o. The potential energy caused by the pair of friction force can be formulated as

$$J^f = -\mu\begin{pmatrix} u_j & v_j \end{pmatrix} f_{up}\boldsymbol{\tau} - (1-\mu)\begin{pmatrix} u_k & v_k \end{pmatrix} f_{up}\boldsymbol{\tau} + \begin{pmatrix} u_i & v_i \end{pmatrix} f_{up}\boldsymbol{\tau} \tag{71}$$

Minimization of J^f causes three second-order vectors to be added to the global load vector. It is noticed that the Coulomb model of friction assumes that the sliding frictional force is proportional to the normal contact force. Provided that the joints between blocks have the mechanical properties of the friction angle φ and the inner cohesion c, f_{up} is equivalent to $|\rho_{\boldsymbol{n}} d_{\boldsymbol{n}}|\tan\varphi + c$. If Equation (64) is adopted to estimate f_{up}, the frictional energy term will contribute not only to the global force vector but also to the stiffness matrix, leading to non-symmetry of the stiffness matrix. To avoid this issue, f_{up} is estimated by using $d_{\boldsymbol{n}}$ obtained in the previous "open–close" iteration. In the view that the variation in $d_{\boldsymbol{n}}$ is small when the "open–close" iteration is about to converge, such a simplification is acceptable.

In the original DDA, the incremental vertex displacements are estimated using Equation (11), while in the proposed method, the incremental vertex displacements are selected as the degrees of freedom. Therefore, the contributions of the contact restraints to the global stiffness matrix and load vector have simpler but more precise expressions in the proposed method than in the original DDA [36].

Minimize the potential energy-induced contact pairs in the block system one by one, and the contribution of $\boldsymbol{J}'$(v) to the global equations are obtained and Equation (9) is fulfilled.

When Equation (9) is fulfilled completely, the global equations are resolved as the traditional DDA. For one time step, the vertex coordinates are updated by adding their incremental displacements to the previous coordinates once the open–close iteration converges. In addition, the constant stress associated with accumulation of the constant strain in $\Pi_P\boldsymbol{u}$ over the foregoing time steps is considered the initial stress for the following time step.

5. Numerical Examples

Some numerical examples are analyzed to test the proposed method in this section. To evaluate the correctness, precision, and efficiency of the new approach, DDA with the post-adjustment strategy and vertex displacements-based DDA based on Wachspress interpolation [36] are adopted to solve these examples. For an objective comparison on the computational efficiency, all computations are conducted on the same laptop with 8 GB of RAM and Intel Core i7-6500U CPU. When conducting contact computation, $\rho_{\boldsymbol{n}}$ was prescribed as 100E and $\rho_{\boldsymbol{\tau}}$ was prescribed as 40E for all methods. Here, E is Young's modulus of the block.

5.1. Rotating Triangular Block Problem

As drawn in Figure 6a, a rotating triangular block problem was designed by the authors to investigate the contact states in the equilibrium equation and in the resulting configurations. There was a triangular block on a rectangular ramp. The block and the ramp had the same material properties: ρ = 2.8 g/cm^3, E = 20 GPa, and Poisson's ratio ν = 0.25. The gravity force of this model was ignored, and a concentrated load P = 10 kN acted at point *5* along the horizontal direction. All vertices of the foundation were fixed as the boundary conditions.

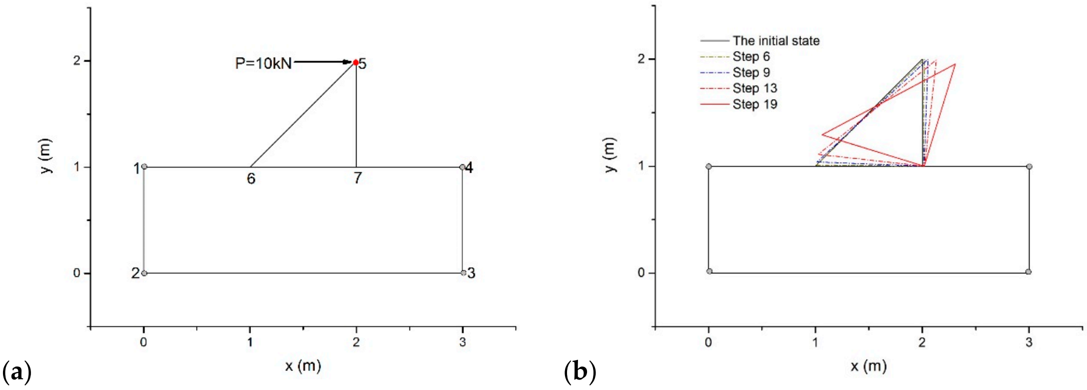

Figure 6. (**a**) A triangular block on the foundation and (**b**) its movement under a horizontal point load.

Firstly, DDA with the post-adjustment strategy was adopted to simulate the dynamic behavior of the model by using the max step displacement ratio δ = 0.1 and Δ = 0.01 s. Figure 6b illustrates the configurations at the end of the 6, 9, 13, and 19 time steps. The triangular block rotated around point *7* for the eccentric moment from P. In the model, edge *14* had no rotation for its two end points that were both fixed. For the contact pair point *7*–edge *14*, we noted the normal penetration value in the equilibrium equations and measured the normal penetration value in the updated configurations. The results are listed in Table 1.

Although the rotational angle r_0 is very tiny in each time step, the difference is sometimes considerable in the two normal penetration values. Moreover, the normal penetration value is negative in the resulting configurations at steps 5, 6, and 7, which violates the actual contact conditions completely.

Then, the example was reanalyzed using the proposed method. For the contact pair point *7*–Edge *14*, the normal penetration values are also given in Table 1. The results indicated that the two normal penetration values are identical when the incremental displacements at the vertices are directly selected as the degrees of freedom.

Finally, vertex displacement-based DDA based on Wachspress interpolation [36] was used to solve this example and produced the same results as the proposed method. In this example, the time consumed by DDA with the post-adjustment strategy was 0.034 s, vertex displacement-based DDA in [36] spent 0.038 s, and the proposed method consumed 0.037 s.

5.2. Sliding Problem

There was a rectangular block on a triangular ramp, as Figure 7a shows. The top surface of the ramp had a slope angle of 45°. The block and ramp were prescribed to have the same mechanical parameters: ρ = 2.5 × 10^3 kg/m^3, E = 35 MPa, and ν = 0.3. The external load only came from the gravitational force with gravity acceleration g = 9.8 m/s^2. No inner cohesion was involved in this model. All vertices of the ramp were fixed as the boundary conditions.

Table 1. The penetrations of the contact pair point 7–edge 14 by DDA with the post-adjustment strategy and the new method.

Time Step	The Original DDA with the Post-Adjustment Strategy		The New Method		The Rotational Angle r_0
	Penetrations in the Equilibrium Equations (m)	Penetrations in the Updated Configurations (m)	Penetrations in the Equilibrium Equations (m)	Penetrations in the Updated Configurations (m)	
1	8.3289×10^{-9}	8.3288×10^{-9}	8.32890×10^{-9}	8.32890×10^{-9}	-0.000001
2	2.5026×10^{-8}	2.5026×10^{-8}	2.50260×10^{-8}	2.50260×10^{-8}	-0.000000
3	7.4918×10^{-8}	7.4918×10^{-8}	7.49180×10^{-8}	7.49180×10^{-8}	-0.000000
4	3.3772×10^{-7}	1.4587×10^{-7}	3.37720×10^{-7}	3.37720×10^{-7}	-0.001073
5	1.0094×10^{-6}	-7.2045×10^{-7}	1.00970×10^{-6}	1.00970×10^{-6}	-0.003218
6	3.0067×10^{-6}	-1.8379×10^{-6}	3.01240×10^{-6}	3.01240×10^{-6}	-0.005375
7	8.9005×10^{-6}	-7.2391×10^{-7}	8.92310×10^{-6}	8.92310×10^{-6}	-0.007554
8	2.6114×10^{-5}	9.8896×10^{-6}	2.62290×10^{-5}	2.62290×10^{-5}	-0.009768
9	7.5032×10^{-5}	5.0108×10^{-5}	7.54400×10^{-5}	7.54400×10^{-5}	-0.012047
10	2.0631×10^{-4}	1.7005×10^{-4}	2.07880×10^{-4}	2.07880×10^{-4}	-0.014444
11	4.9419×10^{-4}	4.4322×10^{-4}	4.97090×10^{-4}	4.97090×10^{-4}	-0.017006
12	7.6165×10^{-4}	6.9448×10^{-4}	7.63320×10^{-4}	7.63320×10^{-4}	-0.019372
13	6.5260×10^{-4}	5.7022×10^{-4}	6.50170×10^{-4}	6.50170×10^{-4}	-0.021275
14	3.2463×10^{-4}	2.2251×10^{-4}	3.22710×10^{-4}	3.22710×10^{-4}	-0.023478
15	2.6022×10^{-4}	1.2772×10^{-4}	2.68970×10^{-4}	2.68970×10^{-4}	-0.026494
16	5.7111×10^{-4}	3.9996×10^{-4}	5.96770×10^{-4}	5.96770×10^{-4}	-0.029810
17	9.6448×10^{-4}	7.5227×10^{-4}	9.91470×10^{-4}	9.91470×10^{-4}	-0.032847
18	9.2970×10^{-4}	6.7950×10^{-4}	9.13440×10^{-4}	9.13440×10^{-4}	-0.035286
19	3.3880×10^{-4}	4.7618×10^{-5}	2.94330×10^{-4}	2.94330×10^{-4}	-0.037664

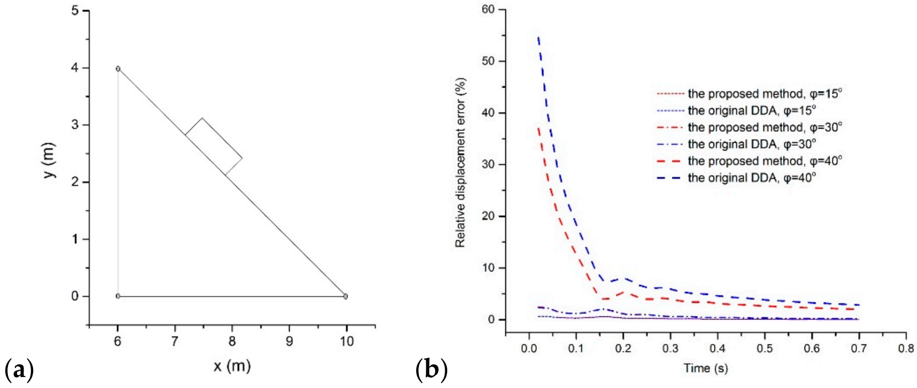

Figure 7. (**a**) The configuration of the sliding problem and (**b**) the relative displacement errors of the proposed method and DDA with the post-adjustment strategy.

The block slides along the ramp if the friction angle of the interface φ is less than 45°, and its displacement is

$$s = 0.5g(\sin 45^o - \tan\varphi \cos 45^o)t^2 \tag{72}$$

in which s denotes the sliding distance (m) of the rectangle and t is the elapsed time (s).

To appraise the accuracy of the new method, three values, 15°, 30°, and 40°, were assigned to the friction angle. Taking Δ = 0.02s and δ = 0.10, 35 time steps were computed. Referring to the analytical solutions of Equation (72), the relative errors of the proposed approach and the original DDA results were measured and are drawn in Figure 7b.

Above all, the relative errors of the two methods both appear to be correlated with the friction angle. Large friction angles always induce larger errors than small angles. The reason behind that is the approximation strategy of the contact force adopted in DDA. For a new contact pair, the proposed method initializes the contact force to 0 as the classical DDA and approximates the exact value according to the resulting interpenetration or inconsistent tangential motion between blocks.

Then, the relative errors of the two methods both rapidly drop off as the time steps increase. In the cases φ = 15° and φ = 30°, the displacement differences in the two methods are quite small. Both methods induced a relative error decreasing from 0.65% to 0.06% when φ = 15° and from 2.4% to 0.2% when φ = 30°, making it difficult to distinguish their relative error curves in Figure 7b. When φ = 40°, the original DDA results in a relative error of 55% at the first time step and a relative error of 2.9% at the end while the new method leads to a relative error of 37% at the first time step and a relative error of 2.0% at the end. Therefore, the proposed method provides higher accuracy results for this problem than the original DDA.

As in the original DDA, the constant stress is estimated using the proposed method because only the constant strain associated with $\Pi_P \boldsymbol{u}$ can be determined within the block. To investigate the difference in stress results of DDA with the post-adjustment strategy and the proposed method, the sliding problem was reanalyzed. Because the stress is unstable in a dynamic analysis, five friction angles, 50°, 55°, 60°, 65°, and 70° were adopted and static analysis with 50 time steps was executed to obtain a stable stress result. Firstly, little difference occurs in the resulting stress values of the two methods. When φ = 50°, the maximal principal stress σ_1 = 0.50 kPa and the minor principal stress σ_2 = −6.91 kPa in the original DDA and σ_1 = 0.63 kPa and σ_2 = −6.66 kPa in the proposed method. When φ = 70°, σ_1 = 2.22 kPa and σ_2 = −5.97 kPa in the original DDA and σ_1 = 2.22 kPa and σ_2 = −5.91 kPa in the proposed method. Then, the minimum principal stress direction appears to increase gently with the friction angle increasing, as shown in Figure 8a, and it stabilizes when the friction angle increases to 65°. The variations in the minimum principal stress direction with the friction angles are plotted for the two methods in Figure 8b. The proposed method has a gentler direction for the minimum principal stress. The stable direction for the minimum principal stress is −76.93° in the original DDA and −76.60° in the proposed method.

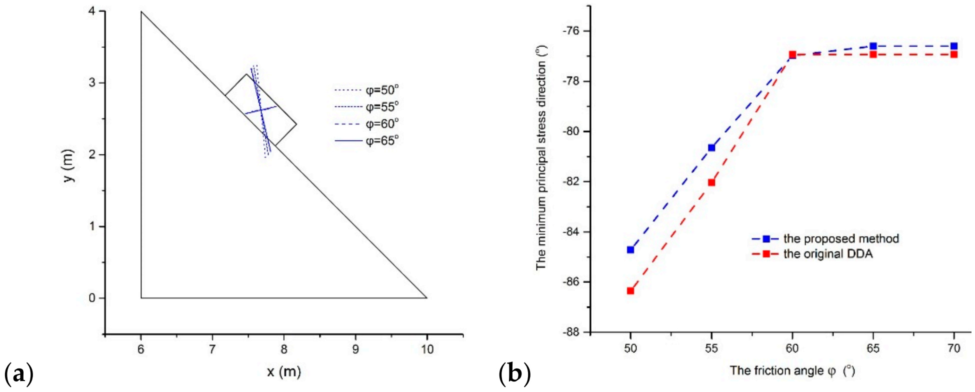

Figure 8. (**a**) The variations in minimum principal stress direction with the friction angles and (**b**) the difference in minimum principal stress direction of the two methods.

For this problem, the calculation times consumed by the three methods appear to have a clear gap. Taking the case φ = 30° as an example, DDA with the post-adjustment strategy took 0.041 s. The time consumed by vertex displacement-based DDA in [36] is 0.108 s. This is because numerical integrations is necessary in computation for a rectangular block when using vertex displacement-based DDA based on Wachspress interpolation. However, only 0.042 s is consumed by vertex displacement-based DDA using VEM.

5.3. Surrounding Rock Problem

The stability assessment of a surrounding rock is an important problem for the design and construction of underground engineering. Figure 9 describes a surrounding rock model composed of 36 rock blocks. Both convex and concave blocks were included in this model. All blocks were assumed to have the same mechanical characteristics: ρ = 2.8 g/cm^3, ν = 0.20, and E = 200 MPa. Only the gravitational force with g = 10 m/s^2 acts on the model. The hoop joints in the surrounding rock divided the model into three layers. The outermost boundaries of this model were fixed as the displacement boundary conditions.

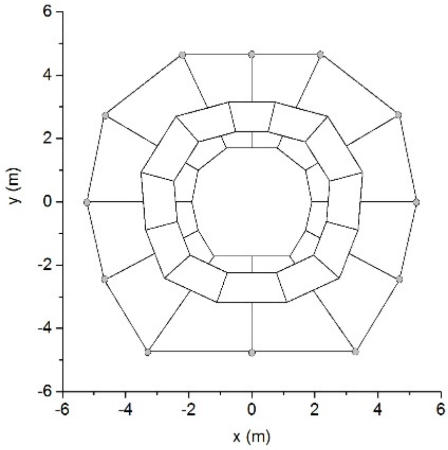

Figure 9. A surrounding rock model.

Using δ = 0.005 and a self-adjusting Δ from the code, the proposed method was employed to conduct a static analysis of the problem. Taking the friction angle φ = 30° for the joints between blocks, Figure 10a illustrates the ultimate configurations and main stresses after 100 time steps. The proposed method arrived at the conclusion that the model is stable. The stress level decreases from the outer layer to the inner layer. At the top and the bottom, the inner layer suffers the largest horizontal compress effect. The top two rocks in the inner layer suffer $\sigma_1 = -0.4$ kPa and $\sigma_2 = -109.7$ kPa, which induces a minimum principal stress direction of 6.7°. The bottom two rocks in the inner layer suffer $\sigma_1 = -15.6$ kPa and $\sigma_2 = -138.6$ kPa, and the direction of the minimum principal stress is 11.4°.

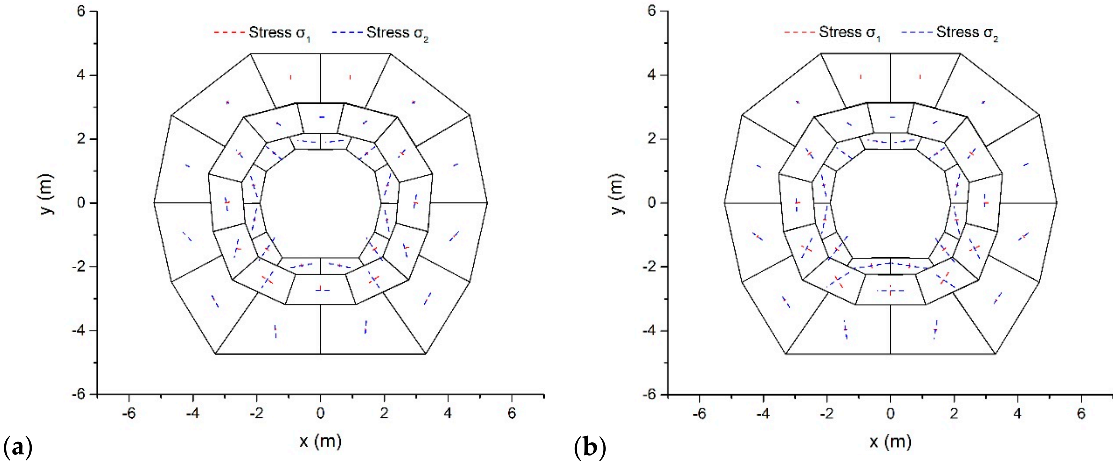

Figure 10. Configurations and main stresses after 100 time steps under two different friction angles: (**a**) φ = 30° and (**b**) φ = 0°.

Considering the mechanical property of the joints may be damaged by the blasting excavation effect, friction was neglected and this model was reanalyzed. Figure 10b plots the results after 100 time steps, which adopts the same tags for stress vectors as Figure 10a. Because of the arch effect, the surrounding rock is also stable and the stress level still decreases from the outer to the inner layers. In contrast, the stress values appear to increase in the case of no friction. The top two rocks in the inner layer suffer $\sigma_1 = -6.2$ kPa and $\sigma_2 = -117.6$ kPa, and the minimum principal stress direction is 9.2°. The bottom two rocks in the inner layer suffer $\sigma_1 = -24.8$ kPa and $\sigma_2 = -227.5$ kPa, which has a minimum principal stress direction of 8.4°. The reason for this is that the stability of the arch only relies on the compression between blocks when the joints friction effect is absent.

The stability of the surrounding rock in this problem was also verified by using DDA with the post-adjustment strategy. Vertex displacement-based DDA based on Wachspress interpolation [36] fails to solve this problem because Wachspress interpolation does not apply for concave blocks. For this problem, more calculation time is consumed by the proposed method than DDA with the post-adjustment strategy, which can be attributed to the fact that the proposed method almost doubles DDA with the post-adjustment strategy in the dimension of the total degrees of freedom. Taking the case with no friction as an example, DDA with the post-adjustment strategy took 0.891 s and the proposed method took 1.117 s.

5.4. Block Wall Failure Problem

As Figure 11 shows, a wall containing 70 blocks was analyzed to test the capacity of the proposed method. The wall had a width of 270 m and a height of 180 m. The mechanical parameters of the blocks were given as $\rho = 2.0 \times 10^3$ kg/m^3, $E = 500$ MPa, and $\nu = 0.25$. The external load only included the gravitational force with $g = 10$ m/s^2. No friction occurred between the blocks. The outer vertices of two base blocks were fixed as the boundary conditions.

Taking $\Delta = 0.012$ s and $\delta = 0.01$, the dynamic behavior of the model was simulated. Figure 12 demonstrates the resulted configurations and main stresses after 7 s.

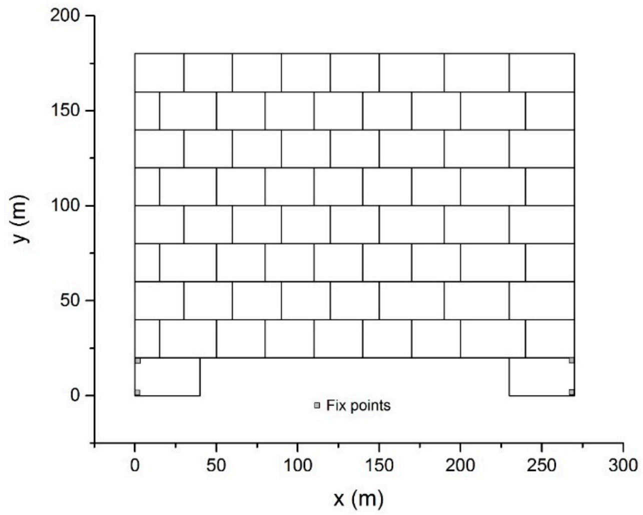

Figure 11. The block wall model.

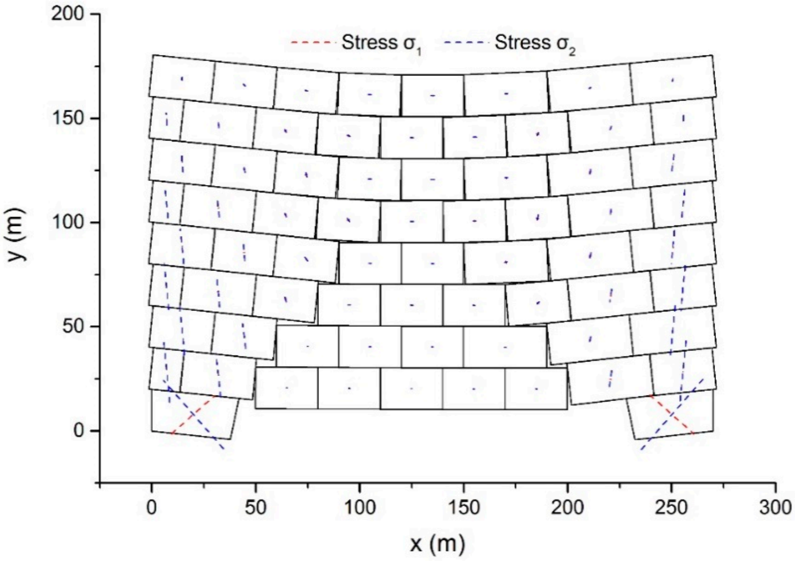

Figure 12. The configurations and main stresses after 7 s using the proposed method.

First, at the bottom, the central blocks descend because of the absence of supports. Next, the middle blocks subside one layer after another. Simultaneously, the movement of the middle blocks leads to rotation and inclination of the left and right blocks. Submitting to the gravitational force of the upper blocks, the bottom two blocks become deformed. After 7 s elapsed, the top boundary of the model appears to be a gentle curve. The stress value of the blocks in the middle is rather small compared to the blocks in other regions, and the two bottom blocks suffer the largest stress. In the bottom right block, the main stresses are σ_1 = 560 kPa and $\sigma_2 = -880$ kPa and the direction of the minimum principal stress is $-48°$.

With Δ and δ unchanged, the dynamic behavior of the example was reanalyzed using DDA with the post-adjustment strategy. The configurations and main stress vectors after 7 s elapsed are shown in Figure 13.

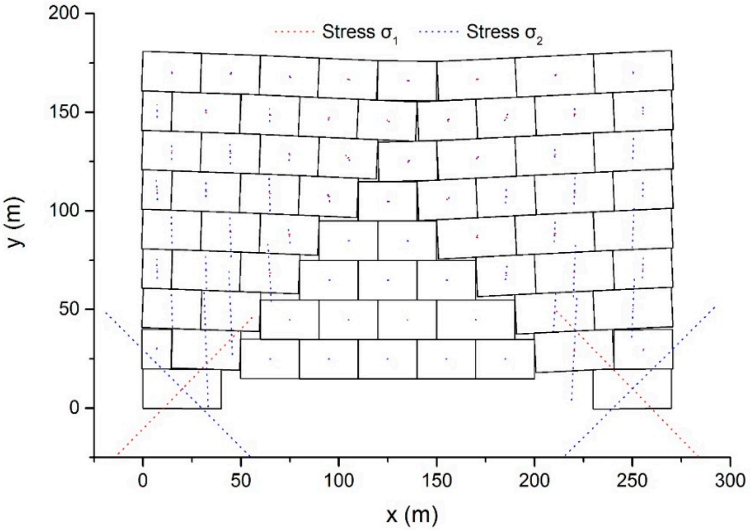

Figure 13. The configurations and main stress vectors after 7 s by DDA with the post-adjustment strategy.

The difference in the results of the proposed method and DDA with the post-adjustment strategy is distinct. The two bottom blocks suffer larger stresses in DDA with the post-adjustment strategy than in the proposed method. In the bottom right block, σ_1 = 2115 kPa and $\sigma_2 = -2205$ kPa, and the minimum principal stress has a direction of $-44°$. Rotation and inclination of the blocks in the left and right regions are smaller compared to the results of the proposed method. The difference in the displacement mode of the two methods has great effects on the difference between the results. The linear displacement function was adopted directly in DDA with the post-adjustment strategy. However, in the case when the vertex number was $n > 3$, the contribution of the higher-order displacement was added to the proposed method. Therefore, the proposed approach provides a block of $n > 3$ vertices with larger deformability than DDA with the post-adjustment strategy. Once the deformation of this block is not restrained by the adjacent blocks, a larger deformation occurs in the proposed method.

For this problem, DDA with the post-adjustment strategy took 1.606 s, the new method took 1.943 s, and vertex displacement-based DDA based on Wachspress interpolation [36] took 4.762 s.

The results of Examples 1 and 2 verified that the proposed method has a higher accuracy in the contact computation than DDA with the post-adjustment strategy. Previous studies to construct vertex displacement-based DDA focused on governing the displacement of a block using the displacement functions in PFEM, e.g., Wachspress interpolation [36]. The issue that Wachspress interpolation does not apply for concave blocks can be remedied by introducing those generalized barycentric coordinates well-defined in the concave domain. If the displacement functions in PFEM are adopted in the construction of vertex displacement-based DDA, the numerical integrations have to be executed for a block with more than three vertices, inducing lower computational efficiency for large-scale problems. Relatively speaking, the proposed method may be a better choice when taking the displacements at the vertices as the degrees of freedom. In the view that the degrees of freedom have a larger dimension in the proposed method than in the original, the new method has a lower computational efficiency than the original DDA. However, a 2% to 20% increase in time consumption is achieved, which is acceptable.

6. Conclusions

In this study, the increments of displacements at block vertices in one time step were selected as the new degrees of freedom for a block. The virtual element spaces were defined for a block and the block system. In the framework of VEM theory, the total potential energy of the DDA block system was estimated and minimized to obtain the global equilibrium equation and load vector. The proposed method can be considered an improvement of DDA by reformulating the block displacement using VEM and can be regarded as an attempt to model the dynamic behavior of the block system using VEM coupled with the contact theory in DDA.

The proposed method has some advantages. On the one hand, the approximation of cos r_0 and sin r_0 in the displacement function of the original DDA is avoided in the proposed method. The degrees of freedom are directly added to the former coordinates in the renewal of the vertex coordinates, which induces a simpler expression regarding the contact restraint impositions. On the other hand, the contact detection and the contact theory can still be adopted as in the original DDA. Meanwhile, most of the numerical integrations are executed along the block boundary when calculating the global stiffness matrix and load vector. Only numerical integrations within the block can be conducted using the simplex integration as in the classical DDA.

Additionally, $\boldsymbol{V}_1(\Omega)$ is taken as the virtual element space for a block in this study. If the stress variability within the block is expected, a higher-order virtual element space instead of $\boldsymbol{V}_1(\Omega)$, e.g., $\boldsymbol{V}_2(\Omega)$ and $\boldsymbol{V}_3(\Omega)$, can be adopted for a block. When using a higher-order virtual element space, the degrees of freedom should be augmented; refer to [46]. Although this study is limited to 2D analysis, the method to improve DDA by reformulating the block displacement using VEM can be mimicked in a 3D case.

Author Contributions: Conceptualization, methodology, H.L. and W.J.; software, G.S.; validation, formal analysis, investigation, resources, data curation, H.L., L.L., and W.J.; writing—original draft preparation, H.L.; writing—review and editing, W.J.; visualization, G.S.; funding acquisition, H.L. and W.J. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the Open Research Fund of State Key Laboratory of Simulation and Regulation of Water Cycle in River Basin (China Institute of Water Resources and Hydropower Research), grant number IWHR-SKL-202020, and by National Natural Science Funds, grant number 52079070 and 42077270.

Data Availability Statement: The data used to support the findings of this study are available from the corresponding author upon request.

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

References

1. Shi, G.H. Discontinuous Deformation Analysis—A New Numerical Model for the Statics and Dynamics of Block Systems. Ph.D. Thesis, Department of Civil Engineering University of California, Berkeley, CA, USA, 1988.
2. MacLaughlin, M.M.; Doolin, D.M. Review of validation of the discontinuous deformation analysis (DDA) method. *Int. J. Numer. Anal. Meth. Geomech.* **2006**, *30*, 271–305. [CrossRef]
3. Jing, L.R. Formulation of discontinuous deformation analysis (DDA)—an implicit discrete element model for block systems. *Eng. Geol.* **1998**, *49*, 371–381. [CrossRef]
4. Yang, Y.T.; Xu, D.D.; Liu, F.; Zheng, H. Modeling the entire progressive failure process of rock slopes using a strength-based criterion. *Comput. Geotech.* **2020**, *126*, 103726. [CrossRef]
5. Chen, K.T.; Wu, J.H. Simulating the failure process of the Xin Mo landslide using discontinuous deformation analysis. *Eng. Geol.* **2018**, *239*, 269–281. [CrossRef]
6. Hatzor, Y.H.; Arzi, A.A.; Zaslavsky, Y.; Shapira, A. Dynamic stability analysis of jointed rock slopes using the DDA method: King Herod's Palace, Masada, Israel. *Int. J. Rock Mech. Min. Sci.* **2004**, *41*, 813–832. [CrossRef]
7. Fu, X.D.; Sheng, Q.; Zhang, Y.H.; Chen, J.; Zhang, S.K.; Zhang, Z.P. Computation of the safety factor for slope stability using discontinuous deformation analysis and the vector sum method. *Comput. Geotech.* **2017**, *92*, 68–76. [CrossRef]
8. Ning, Y.J.; Yang, J.; An, X.M.; Ma, G.W. Modelling rock fracturing and blast-induced rock mass failure via advanced discretization within the discontinuous deformation analysis framework. *Comput. Geotech.* **2011**, *38*, 40–49. [CrossRef]
9. Tang, C.A.; Tang, S.B.; Gong, B.; Bai, H.M. Discontinuous deformation and displacement analysis: From continuous to discontinuous. *Sci. China Tech. Sci.* **2015**, *58*, 1567–1574. [CrossRef]
10. Chen, Y.J.; Zhang, X.; Zhu, W.S.; Wang, W. Modified discontinuous deformation analysis for rock failure: Crack propagation. *Geomech. Eng.* **2018**, *14*, 325–336.
11. Liu, X.Y.; Zhao, Z.Y.; Ma, S.Q. Simulating stress wave with flat-top partition of unity based high-order discontinuous deformation analysis. *Eng. Anal. Bound. Elem.* **2018**, *91*, 110–123. [CrossRef]

12. Chen, G.Q.; He, M.C.; Fan, F.S. Rock burst analysis using DDA numerical simulation. *Int. J. Geomech.* **2018**, *18*, 04018001. [CrossRef]
13. Ning, Y.; Yang, Z.; Wei, B.; Gu, B. Advances in two-dimensional discontinuous deformation analysis for rock-mass dynamics. *Int. J. Geomech.* **2016**, *17*, E6016001. [CrossRef]
14. Cai, Y.; Liang, G.; Shi, G.H. Studying impact problem by LDDA method. In *Discontinuous Deformation Analysis (DDA) and Simulations of Discontinuous Media*; TSI Press: San Antonio, TX, USA, 1996.
15. Bao, H.R.; Zhao, Z.Y.; Tian, Q. On the implementation of augmented Lagrangian method in the two-dimensional discontinuous deformation analysis. *Int. J. Numer. Anal. Meth. Geomech.* **2014**, *38*, 551–571. [CrossRef]
16. Zheng, H.; Jiang, W. Discontinuous deformation analysis based on complementary theory. *Sci. China Ser. E Technol. Sci.* **2009**, *52*, 2547–2554. [CrossRef]
17. Jiang, W.; Zheng, H. Discontinuous deformation analysis based on variational inequality theory. *Int. J. Comput. Methods* **2011**, *8*, 193–208. [CrossRef]
18. Lin, S.; Xie, Z. Performance of DDA time integration. *Sci. China Technol. Sci.* **2015**, *58*, 1558–1566. [CrossRef]
19. Jiao, Y.Y.; Zhang, X.L.; Zhao, J.; Liu, Q.S. Viscous boundary of DDA for modeling stress wave propagation in jointed rock. *Int. J. Rock Mech. Min. Sci.* **2007**, *44*, 1070–1076. [CrossRef]
20. Jiang, Q.; Chen, Y.; Zhou, C.; Yeung, M.-C.R. Kinetic Energy Dissipation and Convergence Criterion of Discontinuous Deformations Analysis (DDA) for Geotechnical Engineering. *Rock Mech. Rock Eng.* **2012**, *46*, 1443–1460. [CrossRef]
21. Beyabanaki, S.A.R.; Jafari, A.; Yeung, M.R. High-order three-dimensional discontinuous deformation analysis (3-D DDA). *Int. J. Numer. Meth. Biomed. Eng.* **2010**, *26*, 1522–1547. [CrossRef]
22. Grayeli, R.; Hatami, K. Implementation of the finite element method in the three-dimensional discontinuous deformation analysis (3D-DDA). *Int. J. Numer. Anal. Meth. Geomech.* **2008**, *32*, 1883–1902. [CrossRef]
23. Wang, J.Q.; Lin, G. Static and dynamic stability analysis using 3D-DDA with incision body scheme. *Earthq. Eng. Eng. Vib.* **2006**, *5*, 273–284. [CrossRef]
24. Liu, J.; Geng, Q.D.; Kong, X.J. A fast common plane identification algorithm for 3D contact problems. *Comput. Geotech.* **2009**, *36*, 41–51. [CrossRef]
25. Ali, R.K.; Ahmad, J.; Wu, J.H. A new algorithm to identify contact patterns between convex blocks for three-dimensional discontinuous deformation analysis. *Comput. Geotech.* **2008**, *35*, 746–759.
26. Zhu, H.; Wu, W.; Chen, J.; Ma, G.; Liu, X.; Zhuang, X. Integration of three dimensional discontinuous deformation analysis (DDA) with binocular photogrammetry for stability analysis of tunnels in blocky rock mass. *Tunn. Undergr. Space Technol.* **2016**, *51*, 30–40. [CrossRef]
27. Jiao, Y.Y.; Huang, G.H.; Zhao, Z.Y.; Zheng, F.; Wang, L. An improved three-dimensional spherical DDA model for simulating rock failure. *Sci. China Tech. Sci.* **2015**, *58*, 1533–1541. [CrossRef]
28. Shi, G.H. Contact theory. *Sci. China Tech. Sci.* **2015**, *58*, 1450–1496. [CrossRef]
29. Song, X.Y.; Huang, D.; Zeng, B. GPU-based parallel computation for discontinuous deformation analysis (DDA) method and its application to modelling earthquake-induced landslide. *Comput. Geotech.* **2017**, *86*, 80–94. [CrossRef]
30. Yang, Y.; Xu, D.; Zheng, H. Explicit Discontinuous Deformation Analysis Method with Lumped Mass Matrix for Highly Discrete Block System. *Int. J. Geomech.* **2018**, *18*, 04018098. [CrossRef]
31. MacLaughlin, M.M.; Sitar, N. Rigid body rotations in DDA. In Proceedings of the First International Forum on Discontinuous Deformation Analysis (DDA) and Simulations of Discontinuous Media, Berkeley, CA, USA, 12–14 June 1996; TSI Press: San Antonio, TX, USA, 1996; pp. 620–635.
32. Cheng, Y.M.; Zhang, Y.H. Rigid body rotation and block internal discretization in DDA analysis. *Int. J. Numer. Anal. Meth. Geomech.* **2000**, *24*, 567–578. [CrossRef]
33. Jiang, W.; Zheng, H. An efficient remedy for the false volume expansion of DDA when simulating large rotation. *Comput. Geotech.* **2015**, *70*, 18–23. [CrossRef]
34. Koo, C.Y.; Chern, J.C. Modification of the DDA method for rigid block problems. *Int. J. Rock Mech. Mining. Sci.* **1998**, *35*, 683–693. [CrossRef]
35. Ke, T.C. The Issue of Rigid-Body Rotation in DDA. In Proceedings of the First International Forum on Discontinuous Deformation Analysis (DDA) and Simulations of Discontinuous Media, Berkeley, CA, USA, 12–14 June 1996; TSI Press: San Antonio, TX, USA, 1996; pp. 318–326.
36. Jiang, W.; Wang, Y.H.; Sun, G.H.; Song, P.C.; Mao, C. Discontinuous Deformation Analysis Based on Wachspress Interpolation Function for Detailed Stress Distribution. *Math. Probl. Eng.* **2018**, *2018*, 4079035. [CrossRef]
37. Dasgupta, G. Interpolants within convex polygon: Wachspress′ shape functions. *J. Aerosp. Eng.* **2003**, *16*, 1–8. [CrossRef]
38. Sukumar, N. Construction of polygonal interpolants: A maximum entropy interpolation. *Int. J. Numer. Meth. Eng.* **2004**, *61*, 2159–2181. [CrossRef]
39. Jiang, W.; Xu, J.J.; Zheng, H.; Wang, Y.H.; Sun, G.H.; Yang, Y.T. Novel displacement function for discontinuous deformation analysis based on mean value coordinates. *Int. J. Numer. Meth. Eng.* **2020**, *121*, 4768–4792. [CrossRef]
40. Joshi, P.; Meyer, M.; DeRose, T.; Green, B.; Sanocki, T. Harmonic coordinates for character articulation. *ACM Trans. Graph.* **2007**, *26*, 1–9. [CrossRef]

41. Talischi, C.; Pereira, A.; Menezes, I.F.M.; Paulino, G.H. Gradient correction for polygonal and polyhedral finite elements. *Int. J. Numer. Methods Eng.* **2015**, *102*, 728–747. [CrossRef]
42. Talischi, C.; Paulino, G.H. Addressing integration error for polygonal finite elements through polynomial projections: A patch test connection. *Math. Models Methods Appl. Sci.* **2014**, *24*, 1701–1727. [CrossRef]
43. Beirão da Veiga, L.; Brezzi, F.; Cangiani, A.; Manzini, G.; Marini, L.D.; Russo, A. Basic principles of virtual element methods. *Math. Models Methods Appl. Sci.* **2013**, *23*, 199–214. [CrossRef]
44. Ahmad, B.; Alsaedi, A.; Brezzi, F.; Marini, L.D.; Russo, A. Equivalent projectors for virtual element methods. *Comput. Math. Appl.* **2013**, *66*, 376–391. [CrossRef]
45. da Veiga LBo Brezzi, F.; Marini, L.D. Virtual Elements for Linear Elasticity Problems. *SIAM J. Numer. Anal.* **2013**, *51*, 794–812. [CrossRef]
46. Beirão da Veiga, L.; Brezzi, F.; Marini, L.D.; Russo, A. The hitchhiker′s guide to the virtual element method. *Math. Models Methods Appl. Sci.* **2014**, *24*, 1541–1573. [CrossRef]
47. Park, K.; Chi, H.; Paulino, G.H. On nonconvex meshes for elastodynamics using virtual element methods with explicit time integration. *Comput. Methods Appl. Mech. Eng.* **2019**, *356*, 669–684. [CrossRef]
48. Gain, A.L.; Talischi, C.; Paulino, G.H. On the Virtual Element Method for three-dimensional linear elasticity problems on arbitrary polyhedral meshes. *Comput. Methods Appl. Mech. Eng.* **2014**, *282*, 132–160. [CrossRef]
49. Wriggers, P.; Rust, W.T.; Reddy, B.D. A virtual element method for contact. *Comput. Mech.* **2016**, *58*, 1039–1050. [CrossRef]
50. Beirão da Veiga, L.; Brezzi, F.; Dassi, F.; Marini, L.D.; Russo, A. Virtual Element approximation of 2D magnetostatic problems. *Comput. Methods Appl. Mech. Eng.* **2017**, *327*, 173–195. [CrossRef]
51. Wriggers, P.; Reddy, B.D.; Rust, W.; Hudobivnik, B. Efficient virtual element formulations for compressible and incompressible finite deformations. *Comput. Mech.* **2017**, *60*, 253–268. [CrossRef]
52. Chi, H.; da Veiga, L.B.; Paulino, G. Some basic formulations of the virtual element method (VEM) for finite deformations. *Comput. Methods Appl. Mech. Eng.* **2017**, *318*, 148–192. [CrossRef]