

Research Article

<https://doi.org/10.1631/jzus.A2200311>



GPU-accelerated vector-form particle-element method for 3D elastoplastic contact of structures

Wei WANG^{1,2}, Yanfeng ZHENG^{1,3,4✉}, Jingzhe TANG¹, Chao YANG^{1,4}, Yaozhi LUO^{1,4}

¹College of Civil Engineering and Architecture, Zhejiang University, Hangzhou 310058, China

²Center for Balance Architecture, Zhejiang University, Hangzhou 310028, China

³Architectural Design and Research Institute of Zhejiang University Co., Ltd., Hangzhou 310028, China

⁴Key Laboratory of Space Structures of Zhejiang Province, Hangzhou 310058, China

Abstract: A graphics processing unit (GPU)-accelerated vector-form particle-element method, i.e., the finite particle method (FPM), is proposed for 3D elastoplastic contact of structures involving strong nonlinearities and computationally expensive contact calculations. A hexahedral FPM element with reduced integration and anti-hourglass is developed to model structural elastoplastic behaviors. The 3D space containing contact surfaces is decomposed into cubic cells and the contact search is performed between adjacent cells to improve search efficiency. A connected list data structure is used for storing contact particles to facilitate the parallel contact search procedure. The contact constraints are enforced by explicitly applying normal and tangential contact forces to the contact particles. The proposed method is fully accelerated by GPU-based parallel computing. After verification, the performance of the proposed method is compared with the serial finite element code Abaqus/Explicit by testing two large-scale contact examples. The maximum speedup of the proposed method over Abaqus/Explicit is approximately 80 for the overall computation and 340 for contact calculations. Therefore, the proposed method is shown to be effective and efficient.

Key words: Graphics processing unit (GPU); Parallel acceleration; Elastoplastic contact; Contact search; Finite particle method (FPM)

1 Introduction

The numerical modeling of elastoplastic contact of structures is a demanding and challenging task in engineering scenarios (Kang and Kim, 2015; Gao et al., 2022). The strong nonlinearities (i.e., geometrical, material, and contact nonlinearities) involved in elastoplastic contact of structures make it difficult to get accurate solutions (Neto et al., 2017). In addition, the contact calculation processes, especially the contact search and contact constraint enforcement procedures, are usually very time-consuming for large-scale contacts (Wriggers, 2006). Therefore, an effective and efficient method is urgently needed to solve the elastoplastic contact of structures.

The finite element method (FEM) is one of the most widely adopted methods for solving the elastoplastic contact of structures. Hallquist et al. (1985) developed the node-to-segment (NTS) contact algorithm in the FEM context, which is simple and reasonably accurate for simulating elastoplastic contacts with large deformation (Hesch and Betsch, 2011; Biabanaki et al., 2014). Also, many algorithms have been developed to accelerate the time-consuming contact search procedures, such as the bucket sort algorithm (Benson and Hallquist, 1990) and the linear contact search algorithm (Chen et al., 2014). With the rapid development of computer technology, much effort has been put into accelerating contact algorithms using parallel computing techniques (Cai et al., 2015; Cao et al., 2020). However, some procedures of the FEM, such as assembling global stiffness matrices, are inherently difficult to decouple for parallel computing, and lead to undesired computational efficiencies.

In addition to the above numerical methods, a vector-form particle-element method, i.e., the finite

✉ Yanfeng ZHENG, yanfeng39@zju.edu.cn

 Wei WANG, <https://orcid.org/0000-0001-5437-2502>

Yanfeng ZHENG, <https://orcid.org/0000-0002-3278-1935>

Received June 14, 2022; Revision accepted Nov. 25, 2022;
Crosschecked Mar. 29, 2023; Online first May 1, 2023

© Zhejiang University Press 2023

particle method (FPM) (Yu and Luo, 2009; Luo and Yang, 2014; Yang et al., 2014; Zheng et al., 2021), also has the potential to model elastoplastic contact of structures. The FPM assumes that a structure is discretized into elements and particles. The mechanical properties of the structure are represented by particles instead of elements. The motion equations of particles obey Newton's second law, and they are solved using explicit time integration techniques. The rigid-body motion of FPM elements can be effectively separated from the total motion of the elements, making the FPM suitable for modeling large-deformation elastoplastic behaviors of structures. In addition, the contact algorithms used in the FEM, e.g., the NTS algorithm, can be used in the FPM. Furthermore, the FPM is highly parallelizable due to the intrinsically decoupled elemental and particle calculation processes. Therefore, the FPM accelerated by parallel computing techniques is promising as an effective and efficient way of modeling the elastoplastic contact of structures.

There are mainly two kinds of parallel computing strategies: central processing unit (CPU)-based techniques and graphics processing unit (GPU)-based techniques. Specifically, general-purpose computing on GPUs (GPGPU) has been widely used in engineering fields (Hou et al., 2021; Wang SQ et al., 2021) with the development of GPU-oriented application programming interfaces such as NVIDIA compute unified device architecture (CUDA). The GPGPU is more suitable for the parallel acceleration of the FPM than the CPU-based techniques since the elemental and particle calculation procedures can be decoupled and assigned to massive GPU threads. Tang et al. (2020) made the first attempt to develop a universal GPU-based parallel strategy for explicit dynamic analysis using the FPM. Based on this work, the authors proposed a 2D parallel contact algorithm for elastoplastic contacts and achieved a considerable improvement in computational efficiency (Wang W et al., 2021).

In this paper, a GPU-accelerated FPM for 3D elastoplastic contact of structures is proposed. Firstly, a hexahedral element with reduced integration and anti-hourglass is developed to model the 3D elastoplastic behaviors of structures. Subsequently, a 3D parallel contact algorithm is proposed. Specifically, the 3D space containing contact surfaces is decomposed into cubic cells, and the contact search is performed between adjacent cells to improve search efficiency. A

connected list data structure is used for storing contact particles to facilitate the parallel contact search procedure. The contact constraints are enforced by explicitly applying contact forces to contact particles. All the computational procedures are designed to be carried out in GPU. The effectiveness and efficiency of the proposed method are verified in four numerical examples.

2 Hexahedral elements in the FPM

This section presents the fundamentals of the hexahedral elements in the FPM. The motion equations of particles and the solutions to the equations are given. Subsequently, the equations of hexahedral elements are described.

2.1 Motion equations

As illustrated in Fig. 1, a 3D structure is discretized into particles and hexahedral elements. The motion of each particle in the FPM is governed by the following equation:

$$m\ddot{\mathbf{d}} = \mathbf{F}_{\text{ext}} - \mathbf{F}_{\text{int}} - \mathbf{F}_{\text{dmp}} + \mathbf{F}_{\text{c}}, \quad (1)$$

where m and $\mathbf{d}$ denote the mass and displacement of the particle, respectively, $\mathbf{F}_{\text{ext}}$ is the resultant external force, $\mathbf{F}_{\text{dmp}}$ is the damping force, $\mathbf{F}_{\text{c}}$ represents the contact force (Section 3.3), and $\mathbf{F}_{\text{int}}$ denotes the resultant internal force:

$$\mathbf{F}_{\text{int}} = \sum_{i=1}^{n_e} \hat{\mathbf{F}}_{\text{int}}^{e,i}, \quad (2)$$

where n_e is the number of elements containing this particle, i denotes the i th element, and $\hat{\mathbf{F}}_{\text{int}}^{e,i}$ is the elemental internal force, which is derived in Section 2.2. The damping force $\mathbf{F}_{\text{dmp}} = \zeta_d m \dot{\mathbf{d}}$ in Eq. (1) is used to account for the mass damping effect, where ζ_d denotes the mass damping coefficient.

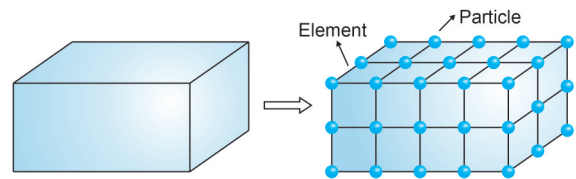


Fig. 1 Discretization of a structure

The central difference method is used to derive an explicit solution to Eq. (1):

$$\mathbf{d}^{t+\Delta t} = \frac{c_1 \Delta t^2}{m} (\mathbf{F}_{\text{ext}}^t - \mathbf{F}_{\text{int}}^t + \mathbf{F}_c^t) + 2c_1 \mathbf{d}^t - c_2 \mathbf{d}^{t-\Delta t}, \quad (3)$$

where t denotes the time, Δt denotes the time step size, $c_1 = (1 + \zeta_d \Delta t/2)^{-1}$, and $c_2 = c_1(1 - \zeta_d \Delta t/2)$.

2.2 Elemental internal forces

2.2.1 Fictitious reverse motion

The movement of a hexahedral element within a time step is shown in Fig. 2. The reference configuration at time t_a is denoted by Ω^a , while the configuration at time $t_b = t_a + \Delta t$ is denoted as Ω^b . The quadrilateral middle surfaces of the element at times t_a and t_b are denoted by M^a and M^b , respectively. The four nodes of the middle surface M^a , denoted as m_1^a, m_2^a, m_3^a , and m_4^a , are midpoints of the edges $(1^a, 5^a)$, $(2^a, 6^a)$, $(3^a, 7^a)$, and $(4^a, 8^a)$, respectively. The total displacement from configuration Ω^a to configuration Ω^b is denoted as $\Delta \mathbf{d}$, which is composed of two parts: rigid-body displacement and deformation. The following procedures are performed to eliminate rigid-body displacement of the element.

At first, the configuration Ω^b is translated to a fictitious configuration Ω' to make sure that particle $1'$ coincides with particle 1^a . The fictitious reverse vector is denoted as $\Delta \mathbf{r}$.

Subsequently, the configuration Ω' is rotated to fictitious configuration Ω'' to ensure that the middle

surface M'' is coplanar with the middle surface M^a . This out-of-plane rotation is along the axis $\mathbf{n}' \times \mathbf{n}^a$, and the rotation angle is

$$\theta_{\text{out}} = \arcsin \frac{|\mathbf{n}' \times \mathbf{n}^a|}{|\mathbf{n}'| \cdot |\mathbf{n}^a|}, \quad (4)$$

where $\mathbf{n}'$ and $\mathbf{n}^a$ denote the normal vectors of the middle surfaces M' and M^a , respectively.

Afterwards, the configuration Ω'' is rotated to the fictitious configuration Ω''' to make sure that the in-plane rigid-body rotation of the middle surface is eliminated. This in-plane rotation is along the axis $\mathbf{n}''$ (the normal vector of the middle surface M''), and the rotation angle is

$$\theta_{\text{in}} = \frac{1}{4} \sum_{i=1}^4 \arcsin \frac{|\mathbf{e}_{ci}'' \times \mathbf{e}_{ci}^a|}{|\mathbf{e}_{ci}''| \cdot |\mathbf{e}_{ci}^a|}, \quad (5)$$

where $\mathbf{e}_{ci}^a$ is the vector pointing from the centroid of the middle surface M^a to the node m_i^a ($i = 1, 2, 3, 4$), and the vector $\mathbf{e}_{ci}''$ is similarly defined.

Finally, the deformation vector $\Delta \mathbf{u}$ from configuration Ω^a to configuration Ω''' is obtained.

2.2.2 Elastoplastic constitutive theory

Once the deformation of a hexahedral element has been obtained, the strain increment of the element $\Delta \boldsymbol{\varepsilon}$ can be calculated by

$$\Delta \boldsymbol{\varepsilon} = \mathbf{B} \Delta \mathbf{u}, \quad (6)$$

where $\mathbf{B}$ reflects the strain-displacement relationship.

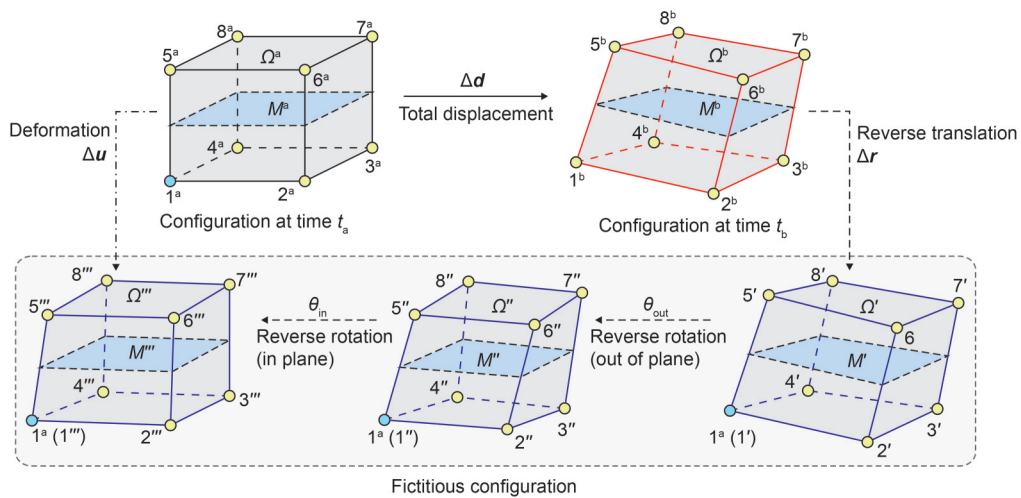


Fig. 2 Obtaining deformation for hexahedral elements

The equations of the stress increment are related to the material of the element. For elastic materials, the stress increment $\Delta\sigma$ can be evaluated as

$$\Delta\sigma = \mathbf{D}^e \Delta\epsilon, \quad (7)$$

where $\mathbf{D}^e$ denotes the elastic stress–strain relation. In addition, the J_2 plasticity model with isotropic hardening is utilized in this study. The yield function is

$$f(\sigma, \kappa) = \sqrt{2J_2} - \kappa \leq 0, \quad (8)$$

where σ is the stress, J_2 denotes the second invariant of the deviatoric stress tensor, and κ represents the plastic internal variable, which can be regarded as a linear function of the cumulative plastic strain. The detailed formulation of the J_2 plasticity model is given in Section S1 of the electronic supplementary materials (ESM). The plastic strain and stress increments in each time step are computed using the radial return algorithm (Simo and Hughes, 1998; Borja, 2013).

2.2.3 Internal forces

The following equation can be obtained using the principle of virtual work:

$$(\delta\Delta\mathbf{u})^T \mathbf{F}_{\text{int}}^e = \int_V (\delta\Delta\epsilon)^T \sigma dV. \quad (9)$$

Substituting Eq. (6) into the right side of Eq. (9), the elemental internal forces $\mathbf{F}_{\text{int}}^e$ can be expressed as the following integral form:

$$\mathbf{F}_{\text{int}}^e = \int_V \mathbf{B}^T \sigma dV. \quad (10)$$

The reduced integration in combination with hourglass control techniques (Flanagan and Belytschko, 1981) is adopted to solve Eq. (10), and the corresponding element is called FPM-H8R. Note that the elemental internal forces need to be corrected by adding anti-hourglass forces:

$$\hat{\mathbf{F}}_{\text{int}}^e = \mathbf{F}_{\text{int}}^e + \mathbf{F}_{\text{HG}}^e, \quad (11)$$

where $\hat{\mathbf{F}}_{\text{int}}^e$ represents the corrected elemental internal forces, and $\mathbf{F}_{\text{HG}}^e$ denotes the elemental anti-hourglass forces.

3 Parallel contact algorithm

The parallel contact algorithm for 3D structures is described in this section. Section 3.1 discusses the discretization of contact surfaces, Section 3.2 gives the parallel contact search procedures, and Section 3.3 describes the enforcement of contact constraints.

3.1 Discretization of contact surfaces

In a typical contact surface pair, as shown in Fig. 3, one contact surface is selected to be the contactor surface and the other to be the target surface. The contact surfaces are discretized using the NTS method. Note that the term “NTS” is referred to as “node-to-face (NTF)” in the following to better describe the 3D contact algorithm in this study. Using the NTF method, the contactor surface is discretized into contactor particles and contactor faces, while the target surface is discretized into target particles and target faces. Note that only quadrilateral contact faces are considered in this study, and they coincide with the faces of hexahedral elements.

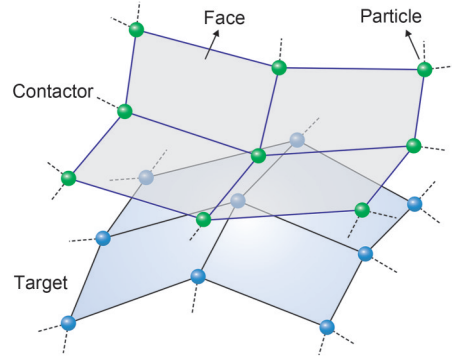


Fig. 3 Contact particles and quadrilateral contact faces

3.2 Parallel contact search

3.2.1 Decomposition of 3D space

As illustrated in Fig. 4, the potential ranges of motion of all contact particles are: $x_{\min} \leq x \leq x_{\max}$, $y_{\min} \leq y \leq y_{\max}$, and $z_{\min} \leq z \leq z_{\max}$, which can be represented by a bounding box. The box is decomposed into a number of identical cubic cells. The cell size L_c is proportional to the average characteristic length of all contact faces.

Each contact particle i_{cp} in the bounding box is located in a cell i_c :

$$i_c = i_{cx} + i_{cy} N_x + i_{cz} N_x N_y, \quad (12)$$

where N_x and N_y denote the cell counts in x and y directions, respectively. The cell index in each direction is expressed as

$$\begin{aligned} i_{cx} &= \text{floor}\left(\frac{x_{cp} - x_{\min}}{L_c}\right), \quad i_{cy} = \text{floor}\left(\frac{y_{cp} - y_{\min}}{L_c}\right), \\ i_{cz} &= \text{floor}\left(\frac{z_{cp} - z_{\min}}{L_c}\right), \end{aligned} \quad (13)$$

where $\text{floor}(x)$ represents the largest integer less than x , and (x_{cp}, y_{cp}, z_{cp}) denotes the coordinate of particle i_{cp} .

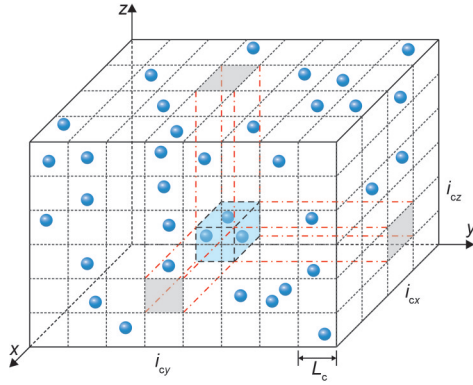


Fig. 4 Decomposition of the 3D space containing contact particles

3.2.2 Connected list data structure

A connected list composed of two arrays is used to store contact particles. This data structure facilitates parallel contact search procedures and also reduces memory requirements in the search procedures. The illustration of the connected list is shown in Fig. 5. The first contact particle in each cell i_c is stored in the array $pHead$, while the rest of the contact particles in this cell are linked one by one in the array $pNext$. A negative value, such as -1 , is used to indicate the termination of the connected list in these two arrays. More details about constructing the connected list data structure can be found in our previous work (Wang W et al., 2021).

3.2.3 Search for the closest target particles

Once the connected list data structure has been constructed, the contact search procedure is carried out by searching for the closest target particles for the contactor particles. In this work, only target particles within cell i_c and the adjacent 26 cells around cell i_c

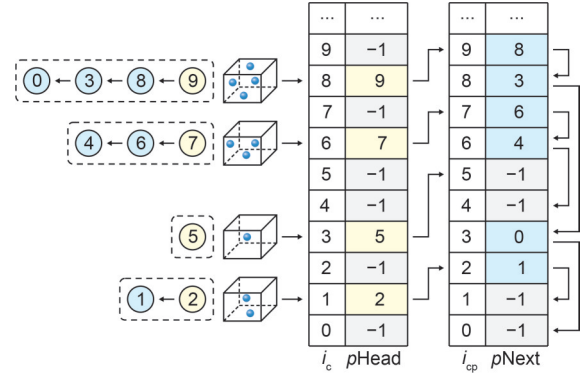


Fig. 5 Schematic illustration of the connected list

will be searched (Zang et al., 2011), which significantly reduces the computational cost of the contact search. The algorithm for the parallel contact search procedures is summarized in Algorithm S1 of the ESM.

3.2.4 Projection onto target faces

After the closest target particles have been found, each contactor particle is then projected onto the target faces. As shown in Fig. 6, the contactor particle P_c is projected onto target face S_i that is connected to the closest target particle P_t , and the projection point is p_i . An isoparametric transformation between face S_i and the quadrilateral isoparametric element is built, and the natural coordinate (ξ_i, η_i) of p_i is obtained. The projection point p_i is inside S_i only if the following condition is met:

$$-1 \leq \xi_i \leq 1, \quad -1 \leq \eta_i \leq 1. \quad (14)$$

The gap vector that points from P_t to P_c is denoted by $\mathbf{g}$, and the normal gap is calculated by

$$g_n = \mathbf{g} \cdot \mathbf{n}_i, \quad (15)$$

where $\mathbf{n}_i$ denotes the normal vector of S_i . In a typical contactor particle-target face pair as depicted in Fig. 6,

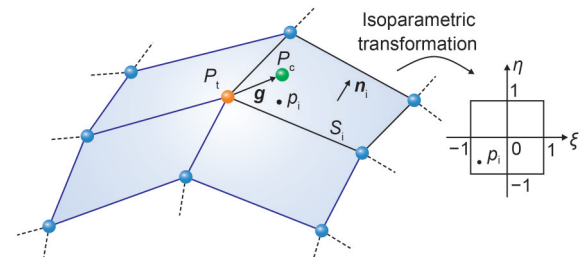


Fig. 6 Projection of contact particles onto target faces

the particle P_c and the face S_i are assumed to be separated if a positive normal gap ($g_n \geq 0$) is obtained, and it is not necessary to compute contact forces in this case. By contrast, contact constraints should be enforced on contact particles that show a negative normal gap ($g_n < 0$).

3.3 Enforcement of contact constraints

The enforcement of contact constraints is performed by directly exerting contact forces on contact particles. The contact forces are computed using the equations of the penalty method.

3.3.1 Non-penetration constraint

The normal contact forces are applied to contact particles to ensure that contactor particles do not penetrate into target faces. Using the penalty method, the normal contact force of particle P_c is obtained by

$$\mathbf{F}_n = \begin{cases} k_{n,i}|g_n| \cdot \mathbf{n}_i, & g_n < 0, \\ 0, & g_n \geq 0, \end{cases} \quad (16)$$

where $k_{n,i}$ is the normal penalty stiffness (Hallquist et al., 1985) of the target face S_i . The expression of $k_{n,i}$ is given in Section S3 of the ESM.

3.3.2 Friction constraint

The tangential contact forces, i.e., friction forces, are exerted on contact particles to account for friction between contact surfaces. The friction force of a contactor particle-target face pair is computed based on the smoothed friction model (Benson and Hallquist, 1990). Given the friction force $\mathbf{F}_s^{t-\Delta t}$ at the previous time step ($t-\Delta t$), and the normal contact force $\mathbf{F}_n^t$, one can compute the friction force at the current time step:

$$\mathbf{F}_s^t = \begin{cases} \mathbf{F}_s^*, & \|\mathbf{F}_s^*\| \leq F_{s,\max}^t, \\ F_{s,\max}^t \frac{\mathbf{F}_s^*}{\|\mathbf{F}_s^*\|}, & \|\mathbf{F}_s^*\| > F_{s,\max}^t, \end{cases} \quad (17)$$

where $F_{s,\max}^t = \mu \|\mathbf{F}_n^t\|$, μ denotes the friction coefficient, and $\mathbf{F}_s^*$ denotes the trial friction force:

$$\mathbf{F}_s^* = \mathbf{F}_s^{t-\Delta t} - k_{t,i}(\mathbf{d}_r^t - \mathbf{d}_r^{t-\Delta t}), \quad (18)$$

where $k_{t,i}$ represents the tangential penalty stiffness, $\mathbf{d}_r^{t-\Delta t}$ and $\mathbf{d}_r^t$ represent the relative displacements of

this contactor particle-target face pair at the previous time step and the current time step, respectively. The expression of $k_{t,i}$ is given in Section S3 of the ESM.

4 Parallel implementation

The proposed method is implemented based on the software developed in our previous studies (Tang et al., 2020; Wang W et al., 2021) using Visual C++ and CUDA toolkit. This section focuses on the parallel implementation of the proposed method.

4.1 Flowchart of parallel FPM solvers

The implementation of the proposed method mainly consists of element, contact, and particle solvers. The flowchart of the parallel FPM solvers is given in Fig. S1 of the ESM. At a certain time step, these three solvers are executed sequentially. The computational procedures in the solvers are performed in GPU threads, and the thread counts are listed in Table S1 of the ESM.

4.2 GPU memory optimization

Among all kinds of GPU memories, i.e., device memories, the global memory has the largest memory space and can be accessed by all GPU threads during their executions. Thus, the global memory is used to store all simulation data in FPM solvers.

The aligned and coalesced memory accesses are required to maximize the global memory bandwidth (Bartezzaghi et al., 2015). To this end, the simulation data in FPM solvers is organized using the storage pattern of SoA (structure of arrays) (Cheng et al., 2014; Lacasta et al., 2014). For example, the device array $pForce$ is a typical array used to store resultant forces of particles, which can be represented by a $N_p \times 3$ matrix, where N_p denotes the number of particles. This array is organized using the SoA memory layout, as shown in Fig. 7. The force component f_x is first stored in the array for each particle one by one, followed by

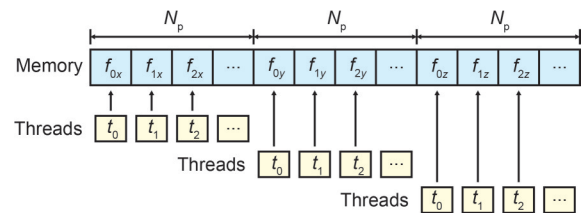


Fig. 7 SoA memory layout

the components f_y and f_z . If there is a memory operation that needs to access the component f_x , then the threads $t_0, t_1, t_2, \dots$ access $f_{0x}, f_{1x}, f_{2x}, \dots$ simultaneously. Because the memory addresses of $f_{0x}, f_{1x}, f_{2x}, \dots$ are adjacent to each other, aligned and coalesced memory accesses are achieved.

5 Numerical examples

This section presents several verification examples and efficiency tests. The hexahedral element FPM-H8R is used in FPM, while the linear brick element with reduced integration C3D8R is used in Abaqus/Explicit. The surface-to-surface contact approach and the penalty contact method are used in Abaqus to model contact.

5.1 Verification examples

Two verification examples are investigated to demonstrate the effectiveness of the proposed method. The first example is an elastic contact case, which is given in Section S5 of the ESM. The second example is an elastoplastic case, which is presented in the following.

The “squeezed plastic tube” example (Seitz et al., 2015) is studied. As shown in Fig. 8, a cylindrical tube structure is squeezed in the middle by two rigid cylindrical dies. The length, inner radius, and outer radius of the tube are 400, 40, and 50 mm, respectively, while those of the dies are 200, 45, and 50 mm, respectively. The tube and the rigid die are discretized into 71680 and 1280 hexahedral elements, respectively.

The two cylindrical dies are subjected to prescribed displacements of $\Delta z=30$ mm towards each other from $t=0$ to $t=1$ s. Young’s modulus and Poisson’s ratio of the tube are 2069 MPa and 0.29, respectively.

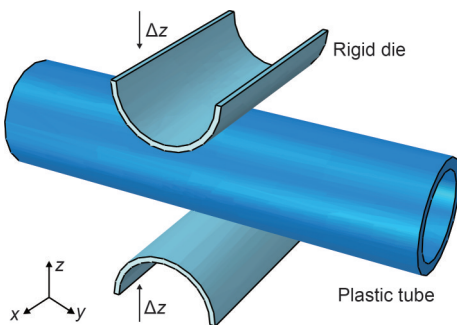


Fig. 8 Squeezed plastic tube: geometry

The plastic properties are given by the initial yield stress of 4.5 MPa and the isotropic linear plastic modulus of 1.0 MPa. The friction coefficient between all contact surfaces is 0.3. The scaling factors s_n and s_t are set to 0.10 and 0.01, respectively. The mass damping coefficient is set to 10.0.

The displacement contour and equivalent plastic strain contour are presented in Figs. 9a and 9b, respectively. It can be found that the plastic regions mainly appear near the middle position of the tube along the axial direction. The time history of total contact forces subjected by the upper die is depicted in Fig. 10. The

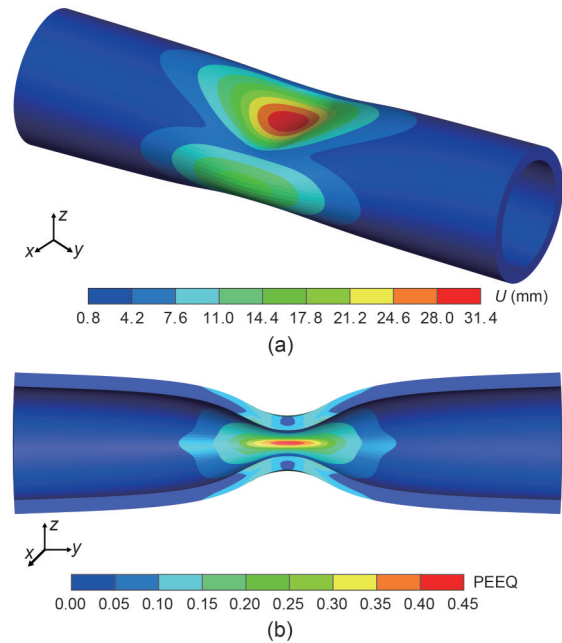


Fig. 9 Contour plots of the tube: (a) displacement U ; (b) equivalent plastic strain PEEQ

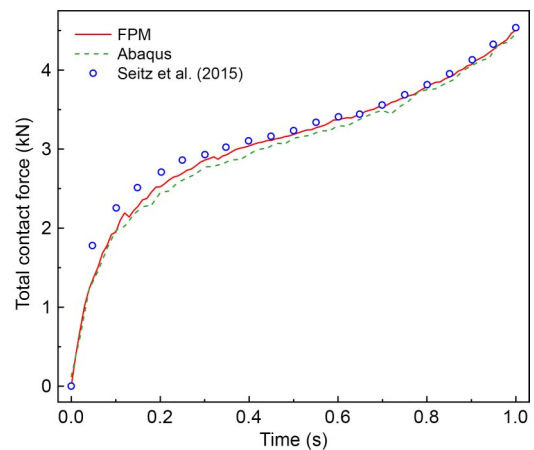


Fig. 10 Squeezed plastic tube: total contact forces

result obtained by FPM is compared with that obtained by Abaqus as well as that in the literature (Seitz et al., 2015) using the semi-smooth Newton method. The above results are in good agreement, indicating the feasibility of the proposed method in modeling elastoplastic frictional contact of structures undergoing large deformation.

5.2 Efficiency tests

The efficiency tests of the proposed method are performed on a computer with double precision. The configuration of the computer is given in Section S6 of the ESM. Two large-scale contact examples are tested. A quasi-static elastic contact example is given in Section S7 of the ESM, and a dynamic elastoplastic contact example is given in the following text. Note that the contact face count and the element count have the same order of magnitude in the above two examples, thus the contact calculation time and the total computational time are comparable. These two examples are solved in three types of solvers, i.e., the parallel FPM solvers running on a GPU, the serial FPM solvers running on a single-core CPU, and Abaqus/Explicit solvers running on the same CPU. It should be pointed out that the parallel FPM solvers and the serial FPM solvers share the same kernel codes. The speedup ratio is introduced to quantify the performance of the parallel implementation (Wang W et al., 2021).

The elastoplastic contact example is adapted from the dynamic elastoplastic contact of a multilayer plate under impact loading (Sharma et al., 2018). As shown in Fig. 11, a cylindrical rod strikes the center of a rectangular multilayer plate. The entire boundary of the plate is constrained. The dimension of the plate is 200 mm×200 mm×5 mm. The radius and height of the rod are 10 and 50 mm, respectively. The initial velocity of the rod is $v_0=5\times10^4$ mm/s.

The elastic material properties of the plate are given by Young's modulus of 70 GPa and Poisson's ratio of 0.3, while the plastic properties are given by the initial yield stress of 285 MPa and the isotropic linear plastic modulus of 900 MPa. Young's modulus and Poisson's ratio of the rod are 210 GPa and 0.3, respectively. The densities of the plate and the rod are 2700 and 7850 kg/m³, respectively. The scaling factor s_n is set to 1.0. The friction between contact surfaces is ignored. Only a quarter of the model is analyzed due to symmetry.

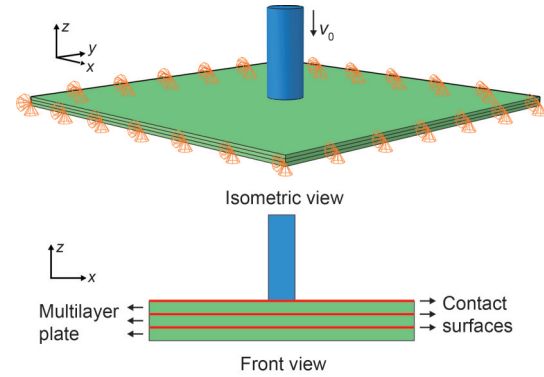


Fig. 11 Multilayer plate under impact loading: geometry

A three-layer plate is first studied to demonstrate the results. Each layer of the quarter plate is discretized into 2500 hexahedral elements, and the quarter rod is discretized into 240 hexahedral elements. An elastic case and an elastoplastic case are studied, in which the material of the plate is assumed to be elastic and elastoplastic, respectively. The time history of the vertical displacement of the center node on the bottom surface of the cylinder is shown in Fig. 12, where the results obtained by FPM and Abaqus are in good agreement. The maximum deformation of the plate in the elastoplastic case is, as expected, greater than that in the elastic case. The deformed configurations of the plate with displacement contours at three instants in the elastoplastic case are presented in Fig. 13. The displacement field is continuous at all contact surfaces of the multilayer plate, indicating the effectiveness of the contact algorithm.

Multilayer plates with five groups of layer counts are studied to test the efficiency of the parallel FPM solvers in elastoplastic contacts. Each layer has a thickness

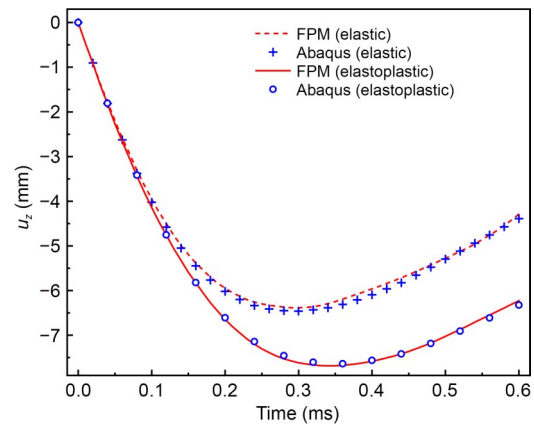


Fig. 12 Vertical displacement u_z of the cylinder

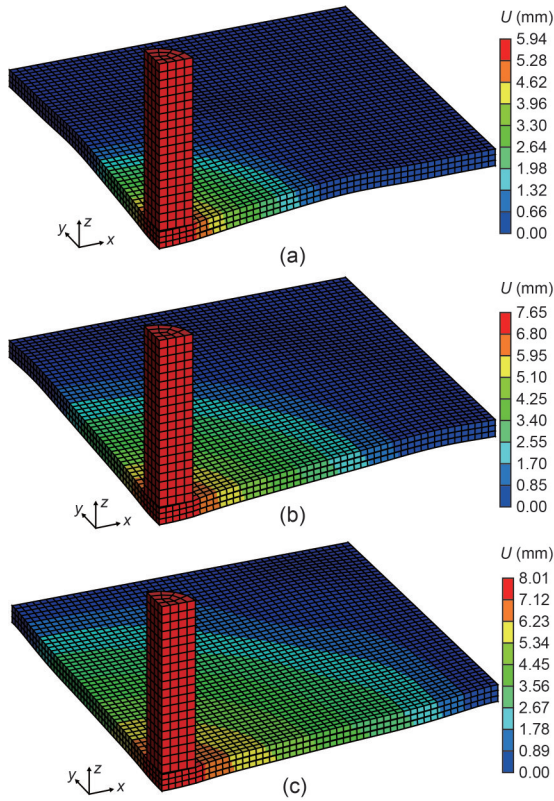


Fig. 13 Contour plots of displacement: (a) $t=0.15$ ms; (b) $t=0.25$ ms; (c) $t=0.35$ ms

of 0.25 mm and is discretized into 62500 ($250 \times 250 \times 1$) hexahedral elements. All cases are solved with a time period of 0.02 ms and a constant time increment of 2×10^{-5} ms. The material of the plate is assumed to be elastoplastic.

The computational times of contact calculation and overall computation are given in Table S4 of the ESM. Specifically, the CPU time consumed by Abaqus contact solver is approximated by Eq. (S5) of the ESM.

The speedups of this example are shown in Fig. 14. In respect of the contact calculation, the parallel FPM is approximately 148 times faster than the serial FPM, while 313 times faster than Abaqus. In respect of the overall computation, the above maximum speedups are 132 and 75, respectively.

Fig. 14 also indicates that the GPU is running at full load when the number of elements reaches approximately 0.8–1.2 million, and the improvement of computational efficiency gradually stabilizes. Similar observations can be found in the literature (Dong et al., 2015). One can achieve higher speedups by using GPUs with larger device memory and more CUDA cores.

The proportions of contact calculation in the overall computational time are approximately 75%, 20%, and 18% in Abaqus, serial FPM, and parallel FPM, respectively, as shown in Fig. 15. Therefore, the

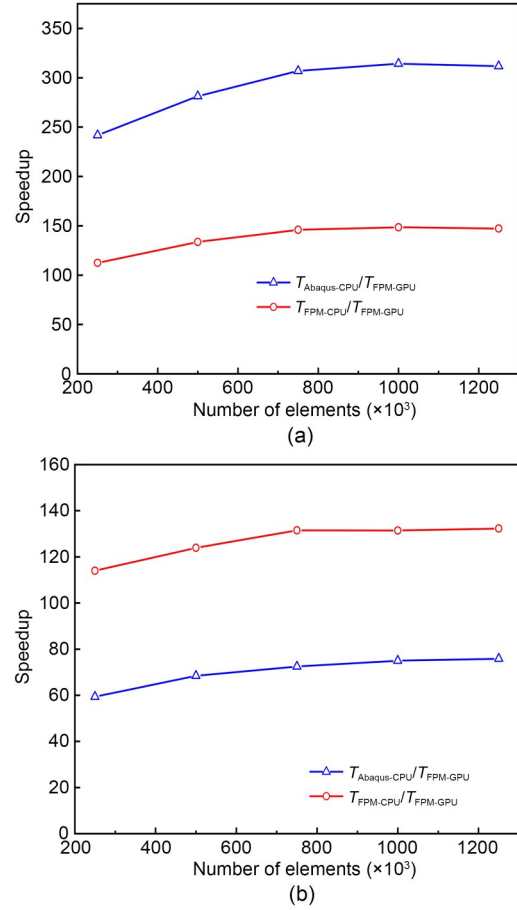


Fig. 14 Multilayer plate under impact loading: (a) speedup of contact calculation; (b) speedup of overall computation

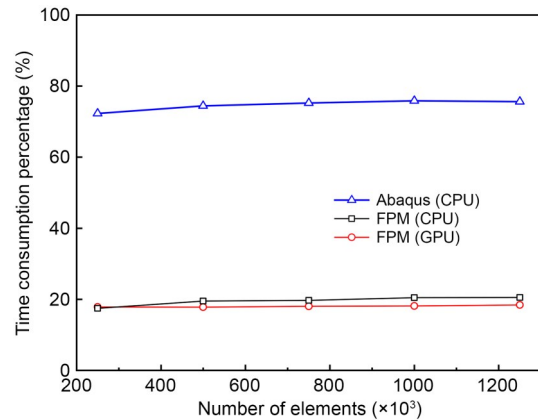


Fig. 15 Multilayer plate under impact loading: proportion of contact calculation time

parallel FPM solvers are proved to be efficient in elastoplastic contacts.

6 Conclusions

A GPU-accelerated FPM is proposed for modeling the 3D elastoplastic contact of structures. A hexahedral element with reduced integration and anti-hourglass is developed to model structural elastoplastic behaviors. In addition, a 3D contact algorithm is proposed. The whole computational procedures of the FPM are implemented to be executed on a GPU. Four illustrative examples are investigated to show the effectiveness and efficiency of the proposed method. Several observations can be drawn:

1. Two verification examples that involve elastic and elastoplastic frictional contacts are investigated. The results obtained by FPM are consistent with the reference results. Thus, the proposed method is effective.
2. The efficiency of the GPU-accelerated FPM is investigated by testing two large-scale contact examples and it is shown that the proposed method is efficient in both elastic and elastoplastic large-scale contacts.
3. The GPU-accelerated FPM is approximately 130–140 times faster than the serial one, and 70–80 times faster than Abaqus. Specifically, the GPU-accelerated FPM contact solver is approximately 110–140 times faster than the serial FPM contact solver, and 310–340 times faster than the Abaqus contact solver.
4. The time proportions of the contact calculation in the overall computation are approximately 18%–19% in the GPU-accelerated FPM, and 75%–80% in Abaqus, indicating that the proposed method dramatically decreases the time consumed in contact calculation processes.

In summary, the proposed method is shown to be efficient, while maintaining computational accuracy, and is promising for modeling large-scale contact in structures. In future work, parallel acceleration using multiple GPUs simultaneously could be studied to further improve the computational efficiency.

Acknowledgments

This work is supported by the National Natural Science Foundation of China (Nos. 51908492, 52008366, and 52238001) and the Zhejiang Provincial Natural Science Foundation of China (Nos. LY21E080022 and LQ21E080019).

Author contributions

Wei WANG conducted the research and wrote the manuscript. Yanfeng ZHENG and Chao YANG helped design the research. Yanfeng ZHENG and Jingzhe TANG revised the manuscript. Yaozhi LUO provided funding and supervision.

Conflict of interest

Wei WANG, Yanfeng ZHENG, Jingzhe TANG, Chao YANG, and Yaozhi LUO declare that they have no conflict of interest.

References

- Bartezzaghi A, Cremonesi M, Parolini N, et al., 2015. An explicit dynamics GPU structural solver for thin shell finite elements. *Computers & Structures*, 154:29-40. <https://doi.org/10.1016/j.compstruc.2015.03.005>
- Benson DJ, Hallquist JO, 1990. A single surface contact algorithm for the post-buckling analysis of shell structures. *Computer Methods in Applied Mechanics and Engineering*, 78(2):141-163. [https://doi.org/10.1016/0045-7825\(90\)90098-7](https://doi.org/10.1016/0045-7825(90)90098-7)
- Biabanaki SOR, Khoei AR, Wriggers P, 2014. Polygonal finite element methods for contact-impact problems on non-conformal meshes. *Computer Methods in Applied Mechanics and Engineering*, 269:198-221. <https://doi.org/10.1016/j.cma.2013.10.025>
- Borja RI, 2013. *Plasticity: Modeling & Computation*. Springer, Berlin, Germany. <https://doi.org/10.1007/978-3-642-38547-6>
- Cai Y, Wang GP, Li GY, et al., 2015. A high performance crashworthiness simulation system based on GPU. *Advances in Engineering Software*, 86:29-38. <https://doi.org/10.1016/j.advengsoft.2015.04.003>
- Cao XG, Cai Y, Cui XY, 2020. A parallel numerical acoustic simulation on a GPU using an edge-based smoothed finite element method. *Advances in Engineering Software*, 148: 102835. <https://doi.org/10.1016/j.advengsoft.2020.102835>
- Chen H, Lei Z, Zang MY, 2014. LC-Grid: a linear global contact search algorithm for finite element analysis. *Computational Mechanics*, 54(5):1285-1301. <https://doi.org/10.1007/s00466-014-1058-5>
- Cheng J, Grossman M, McKercher T, 2014. *Professional CUDA C Programming*. John Wiley & Sons, Indianapolis, USA.
- Dong YK, Wang D, Randolph MF, 2015. A GPU parallel computing strategy for the material point method. *Computers and Geotechnics*, 66:31-38. <https://doi.org/10.1016/j.compgeo.2015.01.009>
- Flanagan DP, Belytschko T, 1981. A uniform strain hexahedron and quadrilateral with orthogonal hourglass control. *International Journal for Numerical Methods in Engineering*, 17(5):679-706. <https://doi.org/10.1002/nme.1620170504>
- Gao B, Wu J, Jia PC, et al., 2022. Experimental and numerical investigation of the polyurea-coated ultra-high-performance

- concrete (UHPC) column under lateral impact loading. *International Journal of Structural Stability and Dynamics*, 22(5):2250037. <https://doi.org/10.1142/S0219455422500377>
- Hallquist JO, Goudreau GL, Benson DJ, 1985. Sliding interfaces with contact-impact in large-scale Lagrangian computations. *Computer Methods in Applied Mechanics and Engineering*, 51(1-3):107-137. [https://doi.org/10.1016/0045-7825\(85\)90030-1](https://doi.org/10.1016/0045-7825(85)90030-1)
- Hesch C, Betsch P, 2011. Transient 3D contact problems-NTS method: mixed methods and conserving integration. *Computational Mechanics*, 48(4):437-449. <https://doi.org/10.1007/s00466-011-0597-2>
- Hou JM, Shi BS, Liang QH, et al., 2021. A graphics processing unit-based robust numerical model for solute transport driven by torrential flow condition. *Journal of Zhejiang University-SCIENCE A (Applied Physics & Engineering)*, 22(10):835-850. <https://doi.org/10.1631/jzus.A2000585>
- Kang H, Kim J, 2015. Progressive collapse of steel moment frames subjected to vehicle impact. *Journal of Performance of Constructed Facilities*, 29(6):04014172. [https://doi.org/10.1061/\(ASCE\)CF.1943-5509.0000665](https://doi.org/10.1061/(ASCE)CF.1943-5509.0000665)
- Lacasta A, Morales-Hernández M, Murillo J, et al., 2014. An optimized GPU implementation of a 2D free surface simulation model on unstructured meshes. *Advances in Engineering Software*, 78:1-15. <https://doi.org/10.1016/j.advengsoft.2014.08.007>
- Luo YZ, Yang C, 2014. A vector-form hybrid particle-element method for modeling and nonlinear shell analysis of thin membranes exhibiting wrinkling. *Journal of Zhejiang University-SCIENCE A (Applied Physics & Engineering)*, 15(5):331-350. <https://doi.org/10.1631/jzus.A1300248>
- Neto DM, Oliveira MC, Menezes LF, 2017. Surface smoothing procedures in computational contact mechanics. *Archives of Computational Methods in Engineering*, 24(1):37-87. <https://doi.org/10.1007/s11831-015-9159-7>
- Seitz A, Popp A, Wall WA, 2015. A semi-smooth newton method for orthotropic plasticity and frictional contact at finite strains. *Computer Methods in Applied Mechanics and Engineering*, 285:228-254. <https://doi.org/10.1016/j.cma.2014.11.003>
- Sharma A, Mishra R, Jain S, et al., 2018. Deformation behavior of single and multi-layered materials under impact loading. *Thin-Walled Structures*, 126:193-204. <https://doi.org/10.1016/j.tws.2017.08.021>
- Simo JC, Hughes TJR, 1998. Computational Inelasticity. Springer, New York, USA.
- Tang JZ, Zheng YF, Yang C, et al., 2020. Parallelized implementation of the finite particle method for explicit dynamics in GPU. *Computer Modeling in Engineering & Sciences*, 122(1):5-31. <https://doi.org/10.32604/cmcs.2020.08104>
- Wang SQ, Zhang Q, Ji SY, 2021. GPU-based parallel algorithm for super-quadric discrete element method and its applications for non-spherical granular flows. *Advances in Engineering Software*, 151:102931. <https://doi.org/10.1016/j.advengsoft.2020.102931>
- Wang W, Zheng YF, Tang JZ, et al., 2021. A GPU-based parallel algorithm for 2D large deformation contact problems using the finite particle method. *Computer Modeling in Engineering & Sciences*, 129(2):595-626. <https://doi.org/10.32604/cmcs.2021.017321>
- Wriggers P, 2006. Computational Contact Mechanics. Springer, Berlin, Germany. <https://doi.org/10.1007/978-3-540-32609-0>
- Yang C, Shen YB, Luo YZ, 2014. An efficient numerical shape analysis for light weight membrane structures. *Journal of Zhejiang University-SCIENCE A (Applied Physics & Engineering)*, 15(4):255-271. <https://doi.org/10.1631/jzus.A1300245>
- Yu Y, Luo YZ, 2009. Finite particle method for kinematically indeterminate bar assemblies. *Journal of Zhejiang University-SCIENCE A*, 10(5):669-676. <https://doi.org/10.1631/jzus.A0820494>
- Zang MY, Gao W, Lei Z, 2011. A contact algorithm for 3D discrete and finite element contact problems based on penalty function method. *Computational Mechanics*, 48(5):541-550. <https://doi.org/10.1007/s00466-011-0606-5>
- Zheng YF, Wan HP, Zhang JY, et al., 2021. Local-coordinate representation for spatial revolute clearance joints based on a vector-form particle-element method. *International Journal of Structural Stability and Dynamics*, 21(7):2150093. <https://doi.org/10.1142/S0219455421500930>

Electronic supplementary materials

Sections S1–S8, Algorithm S1, Figs. S1–S9, Tables S1–S4, Eqs. (S1)–(S5)