Advances in Applied Mathematics and Mechanics
Adv. Appl. Math. Mech., Vol. **16**, No. 2, pp. 305-330

DOI: 10.4208/aamm.OA-2023-0059
April 2024

A Fast Implementation of the Linear Bond-Based Peridynamic Beam Model

Hao Tian[1], Xianchu Yang[1], Chenguang Liu[1] and Guilin Liu[2,*]

[1] *School of Mathematical Sciences, Ocean University of China, Qingdao, Shandong 266100, China*
[2] *College of Engineering, Ocean University of China, Qingdao, Shandong 266100, China*

Received 22 February 2023; Accepted (in revised version) 26 April 2023

Abstract. While the theory of peridynamics (PD) holds significant potential in engineering, its application is often limited by the significant computational costs by the nonlocality of PD. This research is based on a three-dimensional(3D) complex Timoshenko beam structure with six degrees of freedom. We propose a fast meshfree method based on the linear bond-based PD model of the stiffness matrix structure by ingeniously using the matrix decomposition strategy to maintain the Teoplitz structure of the stiffness matrix. This method significantly reduces the amount of calculation and storage without losing accuracy, reduces the amount of calculation from $\mathcal{O}(N^2)$ to $\mathcal{O}(N\log N)$, and decreases the storage capacity from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. We validate the effectiveness of our approach through numerical examples, particularly in multi-beam structures. We demonstrate that our method realizes algorithm acceleration in numerical simulations of multi-beam structures subjected to static concentrated loads.

AMS subject classifications: 65R20, 65Y10

Key words: Bond-based peridynamics, beam model, fast methods, Teoplitz matrix.

1 Introduction

Classical continuum mechanics relies on partial differential equations, making it challenging to describe models with discontinuities. To address this issue, peridynamics was proposed by Silling [1] as a nonlocal model based on an integral equation. This model effectively solves the problems of crack propagation and multiple crack propagation [2]. Consequently, peridynamics has attracted extensive research in modeling methods, digital technology, and applications. Three constitutive relations have been defined for PD [3]: bond-based PD, ordinary state-based PD, and nonordinary state-based

*Corresponding author.
Email: liuguilin73@ouc.edu.cn (G. Liu)

PD. While the bond-based PD limits Poisson's ratio to a fixed value, this restriction is relieved in the state-based model. Currently, PD is widely used to simulate damage prediction and crack propagation in various materials, including elastic and plastic ones [4–7], as well as nonlinear elastic and composite materials [8–11].

Analyzing complex engineering structures using three-dimensional PD models can be computationally expensive. To mitigate this, structure-simplified models such as beams, plates, and shells are often used to reduce the computational cost [12, 13]. Silling et al. first introduced a one-dimensional bar model with axial force [14]. For the Euler-Bernoulli beam, both ordinary and non-ordinary state-based PD beam models have been proposed [15, 16]. More recently, a three-dimensional Euler-Bernoulli PD beam model was proposed by Liu et al. [17]. For the Timoshenko beam, a PD beam model that describes axial, bending, and torsional deformations has been introduced [18]. Other models, such as those proposed by Diyaroglu et al. that introduce shear deformations using a bond-based PD beam model [19], and a 6 DOFs bond-based PD beam model that includes axial, bending, shear, and torsional deformation [20], have also been proposed. Moreover, the mixed formulation method has been developed to alleviate shear locking in PD beam models [21]. Yang et al. extended the PD beam and plate models to finite element frames [22], proposing high-order beam and functionally graded Timoshenko beam formulations [23,24]. Recently, PD beam and shell models based on the micro-beam bond were proposed using the via interpolation method [25,26].

Numerous numerical methods, such as meshfree, finite difference, finite element, and collocation methods, have been developed to solve the above PD models [27–33]. The asymptotic compatibility scheme, which preserves the limit behavior of the zero horizons of the non-local operator into the limit behavior of the local differential operator, provides consistency between the local and non-local models [33,34]. However, the high computational cost of PD poses limitations, particularly for multidimensional situations, even in the case of structure-simplified PD models. To overcome this issue, several efforts have been made, such as coupling methods that use PD only in the area around the crack while using classical mechanics [35,36] and other geometric analyses [37,38] in other areas. Additionally, fast methods based on the convolution structure of the PD model have been proposed [39,40]. To reduce the computational expense, a super-fast peridynamic model that reduces the number of inner loop operations has been introduced [41].

Recently, researchers have turned to the use of stiffness matrix structures as a way of reducing computational costs without sacrificing accuracy, a trend popularized by faster methods that decrease calculation times from $\mathcal{O}(N^2)$ to $\mathcal{O}(N\log N)$. In 2010, a fast method [42] on the basis of the stiffness matrix Toeplitz structure was proposed to solve the one-dimensional static linear bond-based peridynamics. Then, according to the two-dimensional nonlocal diffusion model [43], a fast collocation method based on the TBT matrix structure was given, which can be considered an approximate scalar-valued model. And a fast collocation method of two-dimensional static linear bond-based peridynamics with volume boundary conditions was found [44]. We used an equivalent but more effective method to evaluate it. A fast method [45] was also proposed to solve

the nonlocal diffusion model with variable coefficients, and a discontinuous method was proposed to solve the linear bond-based PD model [46], which has discontinuous solutions. A fast algorithm [47, 48] with preprocessing was proposed in 2020, accelerating the iterative method's convergence. However, despite these advancements, no similar method has been proposed for the PD beam model. To address this gap, this research presents a fast algorithm that utilized the Toeplitz structure of the stiffness matrix of the Timoshenko bond-based PD beam model as its basis, this method contributes significantly to the efforts to optimize computational efficiency, it represents a valuable addition to the growing body of knowledge aimed at sustaining the productivity of computation-based scientific studies.

In this paper, we discusses a newly established fast meshfree method for the PD Timoshenko model, which exploits the quasi-Toeplitz structure of the stiffness matrix. The focus is on decomposing the stiffness matrix to minimize calculation and storage while maintaining accuracy. This approach reduces the amount of calculation from $\mathcal{O}(N^2)$ to $\mathcal{O}(N\log N)$ and storage from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. The new method accelerates numerical simulations of multi-beam structures under static concentrated loads.

The following articles are organized as follows. Section 2 provides a review of the motion equation of the Timoshenko bond-based PD beam model and its discretization in space using the meshfree method. Section 3 demonstrates the Adaptive Dynamic Relaxation (ADR) method for time discretization in static problems. In Section 4, the stiffness matrix's matrix structure in the Timoshenko PD beam model is analyzed, and an acceleration process is presented based on this structure. Section 5 includes numerical examples that demonstrate the accuracy and acceleration effect of the fast method.

2 Linear bond-base peridynamic beam model and its discretization

Peridynamics is considered a new formula using the continuum theory of integral equations and a continuous version of molecular dynamics, which was first introduced by Silling [1], we can also be denoted it as

$$\boldsymbol{\rho}(\mathbf{x})\ddot{\mathbf{u}}(\mathbf{x},t)=\int_{H_x}\left(\mathbf{t}\left(\mathbf{u}'-\mathbf{u},\mathbf{x}'-\mathbf{x},t\right)-\mathbf{t}'\left(\mathbf{u}-\mathbf{u}',\mathbf{x}-\mathbf{x}',t\right)\right)dH_x+\mathbf{b}(\mathbf{x},t), \tag{2.1}$$

where $\mathbf{u}$ and $\mathbf{x}$ denote the displacement vector field, H_x represents the neighborhood, $\boldsymbol{\rho}$, $\mathbf{t}$, and $\mathbf{b}$ denote the mass density, the pairwise force function, and the body force vector. In discrete form, this formula can also be given as

$$\boldsymbol{\rho}_k\ddot{\mathbf{u}}_k=\sum_{j=1}^{M_k}\left(\mathbf{t}_{k,j}\left(\mathbf{u}_j-\mathbf{u}_k,\mathbf{x}_j-\mathbf{x}_k,t\right)-\mathbf{t}_{j,k}\left(\mathbf{u}_k-\mathbf{u}_j,\mathbf{x}_k-\mathbf{x}_j,t\right)\right)V_j+\mathbf{b}_k, \tag{2.2}$$

where material point $\mathbf{x}_j$ represents the family member of material point $\mathbf{x}_k$. V_j is the volume of material point $\mathbf{x}_j$. The amount of family members of $\mathbf{x}_k$ is recorded as M_k, $\mathbf{t}_{kj}$

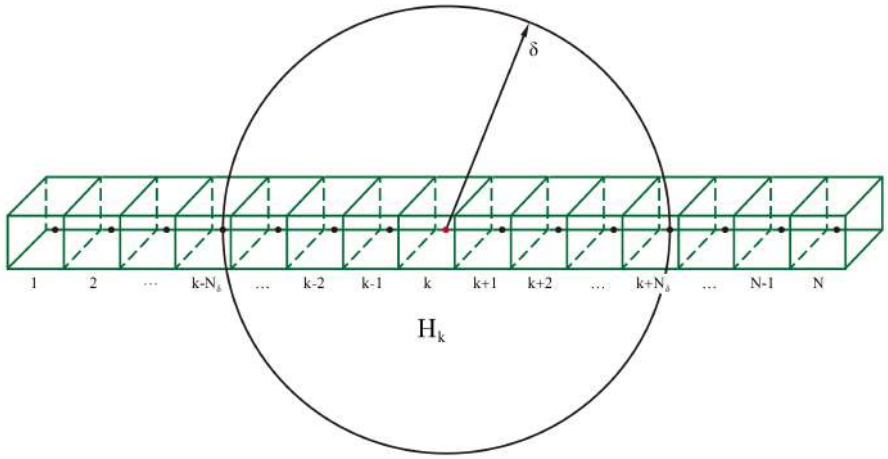

Figure 1: Material points and horizon size for the PD beam model.

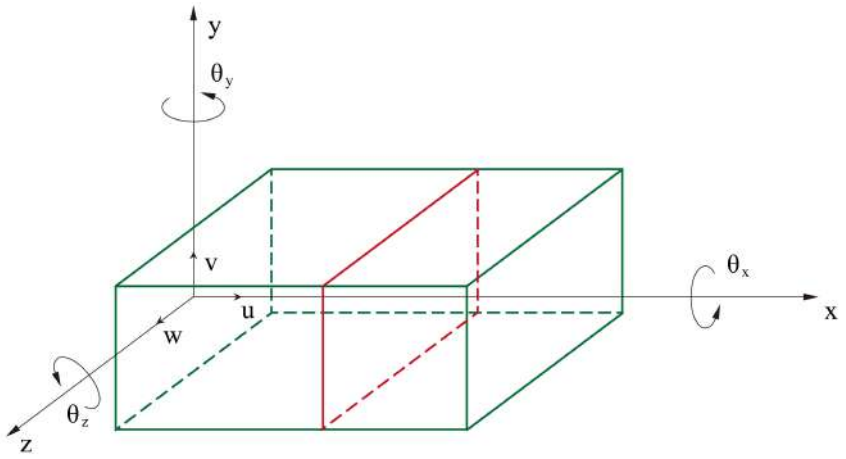

Figure 2: Beam model with 6 DOFs.

means the force density that material point $\mathbf{x}_k$ exerts on point $\mathbf{x}_j$. As shown in Fig. 1, H_k is a neighborhood named a horizon, $\delta = N_\delta \Delta x$ represents the maximum distance of interaction between two material points, which is named after horizon size, where Δx represents the distance between two adjacent points.

In this study, each beam is considered a Timoshenko beam structure, which has six degrees of freedom [50]; As shown in Fig. 2, the beam has three displacements (u,v,w) and three cross-sectional rotations $(\theta^x,\theta^y,\theta^z)$.

As explained by [20], for a beam, the strain energy density (SED) could be seperated

into four components, including axial, bending, shear, and torsional as

$$W_{(k)}^{PD}=W_{\text{axial }(k)}^{PD}+W_{\text{bending }(k)}^{PD}+W_{\text{shear }(k)}^{PD}+W_{\text{torsional }(k)}^{PD}. \tag{2.3}$$

The axial component of SED was brought in

$$W_{axial(k)}^{PD}=\frac{1}{2}\sum_{j=1}^{M_k}\frac{1}{2}C_{ax}\left(\frac{u_j-u_k}{\xi}\right)^2\xi V_j, \tag{2.4}$$

where $\xi=|\mathbf{x}_j-\mathbf{x}_k|$ represents the distance between $\mathbf{x}_k$ and $\mathbf{x}_j$, $C_{ax}=2E/A\delta^2$ represents the bond constant. E is the elastic modulus, A denotes the sectional area of the beam.

The bending component of SED was brought in

$$W_{\text{bending }(k)}^{PD}=\frac{1}{2}\sum_{j=1}^{M_k}\frac{1}{2}\left[C_{by}\left(\frac{\theta_j^y-\theta_k^y}{\xi}\right)^2+C_{bz}\left(\frac{\theta_j^z-\theta_k^z}{\xi}\right)^2\right]\xi V_j, \tag{2.5}$$

where $C_{bx}=2EI_{yy}/A^2\delta^2$ and $C_{by}=2EI_{zz}/A^2\delta^2$ represent the bending bond constant, I_{yy} and I_{zz} represent the area moment of inertia as

$$I_{yy}=\int_A z^2dA,\quad I_{zz}=\int_A y^2dA. \tag{2.6}$$

The shear component of SED was brought in

$$\begin{aligned}W_{\text{shear }(k)}^{PD}=\frac{1}{2}\sum_{j=1}^{M_k}\frac{1}{2}C_s\Bigg[&\left(\frac{u_j^y-u_k^y}{\xi}\operatorname{sgn}(x_j-x_k)-\frac{\theta_j^z+\theta_k^z}{2}\right)^2\\&+\left(\frac{u_j^z-u_k^z}{\xi}\operatorname{sgn}(x_j-x_k)+\frac{\theta_j^y+\theta_k^y}{2}\right)^2\Bigg]\xi V_j\end{aligned} \tag{2.7}$$

with

$$\operatorname{sgn}(x_j-x_k)=\frac{x_j-x_k}{\xi}, \tag{2.8}$$

where $C_s=2K_sG/A\delta^2$ represents the shear bond constant, G denotes the shear modulus, $K_s=1.5$ represents the shear correction factor.

The torsional component of SED was brought in

$$W_{\text{torsional }(k)}^{PD}=\frac{1}{2}\sum_{j=1}^{M_k}\frac{1}{2}C_t\left(\frac{\theta_j^x-\theta_k^x}{\xi}\right)^2\xi V_j, \tag{2.9}$$

where $C_t=2K_tG/A^2\delta^2$ represents the torsional bond constant, K_t represents the torsional constant as

$$K_t=\int_A\left[z^2+y^2\right]dA. \tag{2.10}$$

With the help of the Euler-Lagrange equation, the motion equation of the beam could be obtained [49]. We can write it as

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{d}_i}\right)-\frac{\partial L}{\partial d_i}=0, \tag{2.11}$$

where d_i denotes the degree of freedom with $d=\{u\ v\ w\ \theta^x\ \theta^y\ \theta^z\}$, $\dot{d}_i$ denotes the time derivative of d_i and L is the Lagrangian.

The Lagrangian function of the beam could be represented as $L=T-U$, where T and U mean the total kinetic energy and the total potential energy, respectively, given as

$$T=\frac{\rho}{2}\sum_{k=1}^{M_k}\left((\dot{u}_k)^2+(\dot{v}_k)^2+(\dot{w}_k)^2+\frac{I_{xx}}{A}\left(\dot{\theta}_k^x\right)^2+\frac{I_{yy}}{A}\left(\dot{\theta}_k^x\right)^2+\frac{I_{zz}}{A}\left(\dot{\theta}_k^x\right)^2\right)V_k, \tag{2.12a}$$

$$U=\sum_{k=1}^{M_k}\left(W_{(k)}^{PD}-b_k^x u_k^x-b_k^y u_k^y-b_k^z u_k^z-m_k^x\theta_k^x-m_k^y\theta_k^y-m_k^z\theta_k^z\right)V_k, \tag{2.12b}$$

where b_k^x, b_k^y, b_k^z denote the applied body forces, and m_k^x, m_k^y, m_k^z denote the moment for per unit volume at material point $\mathbf{x}_k$. The equation can be written as

$$\rho\ddot{u}_k=C_{ax}\sum_{j=1}^{M_k}\left(\frac{u_j-u_k}{\xi}\right)V_j+b_k^x, \tag{2.13a}$$

$$\rho\ddot{v}_k=C_s\sum_{j=1}^{M_k}\left(\frac{v_j-v_k}{\xi}-\frac{\theta_j^z+\theta_k^z}{2}\operatorname{sgn}\left(x_j-x_k\right)\right)V_j+b_k^y, \tag{2.13b}$$

$$\rho\ddot{w}_k=C_s\sum_{j=1}^{M_k}\left(\frac{w_j-w_k}{\xi}+\frac{\theta_j^y+\theta_k^y}{2}\operatorname{sgn}\left(x_j-x_k\right)\right)V_j+b_k^z, \tag{2.13c}$$

$$\frac{\rho I_{xx}}{A}\ddot{\theta}_k^x=C_t\sum_{j=1}^{M_k}\left(\frac{\theta_j^x-\theta_k^x}{\xi}\right)V_j+m_k^x, \tag{2.13d}$$

$$\frac{\rho I_{yy}}{A}\ddot{\theta}_k^y=C_{by}\sum_{j=1}^{M_k}\left(\frac{\theta_j^y-\theta_k^y}{\xi}\right)V_j-\frac{1}{2}C_s\sum_{j=1}^{N}\left(\frac{w_j-w_k}{\xi}\operatorname{sgn}\left(x_j-x_k\right)+\frac{\theta_j^y+\theta_k^y}{2}\right)\xi V_j+m_k^y, \tag{2.13e}$$

$$\frac{\rho I_{zz}}{A}\ddot{\theta}_k^z=C_{bz}\sum_{j=1}^{M_k}\left(\frac{\theta_j^z-\theta_k^z}{\xi}\right)V_j+\frac{1}{2}C_s\sum_{j=1}^{N}\left(\frac{v_j-v_k}{\xi}\operatorname{sgn}\left(x_j-x_k\right)-\frac{\theta_j^z+\theta_k^z}{2}\right)\xi V_j+m_k^z. \tag{2.13f}$$

We can also write Eq. (2.13) in the vector form as

$$\boldsymbol{\rho}_k\ddot{\boldsymbol{d}}_k=\sum_{j=1}^{M_k}\mathbf{f}_{k,j}V_j+\mathbf{b}_k, \tag{2.14}$$

where $\ddot{d}$ represents the second time derivative of d

$$\boldsymbol{\rho}_k=\begin{bmatrix}\rho & 0 & 0 & 0 & 0 & 0\\ 0 & \rho & 0 & 0 & 0 & 0\\ 0 & 0 & \rho & 0 & 0 & 0\\ 0 & 0 & 0 & \dfrac{\rho I_{xx}}{A} & 0 & 0\\ 0 & 0 & 0 & 0 & \dfrac{\rho I_{yy}}{A} & 0\\ 0 & 0 & 0 & 0 & 0 & \dfrac{\rho I_{zz}}{A}\end{bmatrix},\quad \ddot{d}_k=\begin{bmatrix}\ddot{u}_k\\ \ddot{v}_k\\ \ddot{w}_k\\ \ddot{\theta}_k^x\\ \ddot{\theta}_k^y\\ \ddot{\theta}_k^z\end{bmatrix},\quad \mathbf{b}_k=\begin{bmatrix}b_k^x\\ b_k^y\\ b_k^z\\ m_k^x\\ m_k^y\\ m_k^z\end{bmatrix}, \tag{2.15}$$

and

$$\mathbf{f}_{k,j}=\begin{bmatrix}f_{k,j}^u & f_{k,j}^v & f_{k,j}^w & f_{k,j}^{\theta_x} & f_{k,j}^{\theta_y} & f_{k,j}^{\theta_z}\end{bmatrix}^T \tag{2.16}$$

with

$$f_{k,j}^u=C_{ax}\left(\frac{u_j-u_k}{\xi}\right)V_j,\quad f_{k,j}^v=C_s\left(\frac{v_j-v_k}{\xi}-\frac{\theta_j^z+\theta_k^z}{2}\operatorname{sgn}\left(x_{(j)}-x_{(k)}\right)\right)V_j, \tag{2.17a}$$

$$f_{k,j}^w=C_s\left(\frac{w_j-w_k}{\xi}+\frac{\theta_j^y+\theta_k^y}{2}\operatorname{sgn}\left(x_{(j)}-x_{(k)}\right)\right)V_j,\quad f_{k,j}^{\theta_x}=C_t\left(\frac{\theta_j^x-\theta_k^x}{\xi}\right)V_j, \tag{2.17b}$$

$$f_{k,j}^{\theta_y}=C_{by}\left(\frac{\theta_j^y-\theta_k^y}{\xi}\right)V_j-\frac{1}{2}C_s\left(\frac{w_j-w_k}{\xi}\operatorname{sgn}\left(x_{(j)}-x_{(k)}\right)+\frac{\theta_j^y+\theta_k^y}{2}\right)\xi V_j, \tag{2.17c}$$

$$f_{k,j}^{\theta_z}=C_{bz}\left(\frac{\theta_j^z-\theta_k^z}{\xi}\right)V_j+\frac{1}{2}C_s\left(\frac{v_j-v_k}{\xi}\operatorname{sgn}\left(x_{(j)}-x_{(k)}\right)-\frac{\theta_j^z+\theta_k^z}{2}\right)\xi V_j. \tag{2.17d}$$

Eq. (2.17) of each beam could be expressed by the following matrix form

$$\boldsymbol{\rho}\ddot{d}=\mathbf{A}d+\mathbf{b},\quad \mathbf{f}=\mathbf{A}d, \tag{2.18}$$

with $\boldsymbol{\rho}$, d, $\mathbf{b}$ could be expressed by the following block matrix forms

$$\boldsymbol{\rho}=\begin{bmatrix}\boldsymbol{\rho}_1 & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{0} & \boldsymbol{\rho}_2 & \ddots & \vdots\\ \vdots & \ddots & \ddots & \mathbf{0}\\ \mathbf{0} & \cdots & \mathbf{0} & \boldsymbol{\rho}_n\end{bmatrix},\quad d=\begin{bmatrix}d_1\\ d_2\\ \vdots\\ d_n\end{bmatrix},\quad \mathbf{b}=\begin{bmatrix}\mathbf{b}_1\\ \mathbf{b}_2\\ \vdots\\ \mathbf{b}_n\end{bmatrix}. \tag{2.19}$$

3 Temporal discretization of static peridynamic beam model

For the static problems studied in this paper, time discretization can be realized by an explicit scheme using the ADR method, which is mainly designed for explicitly solving

the static load problem of the model. We use t^n to express the time at timestep n, Δt is a constant time step size, and the displacement vector d at time t^n is written as $d^n = d(\mathbf{x}, t^n)$. The mass matrix $\mathbf{m}$ can be substituted by a stable mass vector $\mathbf{M}$, and we add a damping matrix $\mathbf{C}$ into the equation of motion given in Eq. (2.18) as

$$\mathbf{M}\ddot{d}^n + \mathbf{C}\dot{d}^n = \mathbf{A}d^n + \mathbf{b}^n, \tag{3.1}$$

where $\mathbf{M}$ is the fictitious diagonal density matrix [20] as

$$\mathbf{M} = \begin{bmatrix} \mathbf{\Lambda} & 0 & \cdots & 0 \\ 0 & \mathbf{\Lambda} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \mathbf{\Lambda} \end{bmatrix}, \quad \mathbf{\Lambda} = \begin{bmatrix} \Lambda_u & 0 & 0 & 0 & 0 & 0 \\ 0 & \Lambda_v & 0 & 0 & 0 & 0 \\ 0 & 0 & \Lambda_w & 0 & 0 & 0 \\ 0 & 0 & 0 & \Lambda_{\theta_x} & 0 & 0 \\ 0 & 0 & 0 & 0 & \Lambda_{\theta_y} & 0 \\ 0 & 0 & 0 & 0 & 0 & \Lambda_{\theta_z} \end{bmatrix}, \tag{3.2}$$

with

$$\Lambda_u = \Lambda_v = \Lambda_w = \max\left(\frac{1}{4}\frac{\Delta t^2 A C_{ax}\delta}{\Delta x}, \frac{1}{4}\frac{\Delta t^2 A C_s \delta}{\Delta x}\right), \tag{3.3a}$$

$$\Lambda_{\theta_x} = \Lambda_{\theta_y} = \Lambda_{\theta_z} = \max\left(\frac{1}{4}\frac{\Delta t^2 A C_t \delta}{\Delta x}, \frac{1}{4}\frac{\Delta t^2 A C_{by}\delta}{\Delta x}, \frac{1}{4}\frac{\Delta t^2 A C_{bz}\delta}{\Delta x}\right). \tag{3.3b}$$

The damping matrix $\mathbf{C}$ is defined as $\mathbf{C} = C_n\mathbf{M}$, where C_n represents the damping coefficient. To facilitate reaching a steady-state regime quickly, artificial damping has been introduced, and the damping coefficient is adaptively changed at each time step to ensure its optimal effectiveness.

$$C_n = 2\sqrt{\frac{\mathbf{d^n}^T\mathbf{K^n}\mathbf{d^n}}{\mathbf{d^n}^T\mathbf{d^n}}} \tag{3.4}$$

with $\mathbf{K^n}$ denote the diagonal "local" stiffness matrix [51]. The displacement can be calculated by Algorithm 3.1.

At this time, since $\mathbf{M}$ is a diagonal matrix, the calculation amount in Algorithm 3.1 is determined by $\mathbf{A}d$.

4 A fast meshfree method

When using the ADR method to solve static problems, the main calculation is to solve $\mathbf{f} = \mathbf{A}d$, a traditional matrix-vector multiplication $\mathbf{A}d$ has a computational complexity of $\mathcal{O}(N^2)$ and a memory requirement of $\mathcal{O}(N^2)$, with the increasing of material points, the computational complexity, and the memory requirement will proliferate, so we developed a fast meshfree method to reduce the computational complexity from $\mathcal{O}(N^2)$ to

Algorithm 3.1 Adaptive dynamic relaxation method.

Require: Stiffness matrix $\mathbf{A}$; Initial displacement d^0; Loading $\mathbf{b}$; Time step $\Delta t=1$; Fictitious diagonal density matrix $\mathbf{M}$.

while not converged **do**

Calculation parameter C_n;

if $n=1$ **then**

$$\dot{d}^{1/2}=\frac{\Delta t}{\mathbf{M}}\frac{\mathbf{A}d^0+\mathbf{b}^0}{2}$$

else if $n>0$ **then**

$$\dot{d}^{n+1/2}=\frac{(2-\Delta tC_n)\dot{d}^{n-1/2}+2\Delta t(\mathbf{A}d^n+\mathbf{b}^n)/\mathbf{M}}{2+\Delta tC_n}$$

end if

end while

$$d^{n+1}=d^n+\Delta t\dot{d}^{n+1/2}$$

$\mathcal{O}(N\log N)$ and reduce the memory requirement from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. This fast meshfree method uses fast Fourier transforms (FFT) to solve $\mathbf{A}d$, which $\mathbf{A}$ must be a Toeplitz matrix. Therefore, we need to make the matrix meet the conditions through some matrix transformations.

4.1 A fast meshfree method of PD for single beam

We begin this section by considering a linear bond-based peridynamic single beam model $\Omega=\Omega_s\cup\Omega_c=[x_l,x_s]$, where Ω is the spatial domain and the material points are evenly distributed. As shown in Fig. 3, we divide Ω_s into $\Omega_{in}\cup\Omega_e$ to investigate the properties of different material points.

- $\Omega_c=\{\mathbf{x}_i:d_i(t)=\mathbf{h}_i(t)\}$;
- $\Omega_{in}=\{\mathbf{x}_i:[x_i-\delta,x_i+\delta]\cap\Omega=[x_i-\delta,x_i+\delta]\}$;
- $\Omega_e=\{\mathbf{x}_i:[x_i-\delta,x_i+\delta]\cap\Omega\neq[x_i-\delta,x_i+\delta]\}$.

Here, Ω_c is volume constrained boundary with the prescribed displacement data $\mathbf{h}_i(t)$. Ω_{in} is the internal area, where the material points have $2N_\delta$ family members. The domain Ω_e includes the material points still in Ω_s, but the number of family members owned by material points is less than $2N_\delta$. N_{in}, N_e, N_c respectively represent the number of material points in areas Ω_{in}, Ω_e, Ω_c, $N_{in}+N_e=N$.

4.1.1 The stiffness matrix form of PD beam model

The matrix $\overline{\mathbf{A}}$ is the stiffness matrix form of the PD beam model, defined as

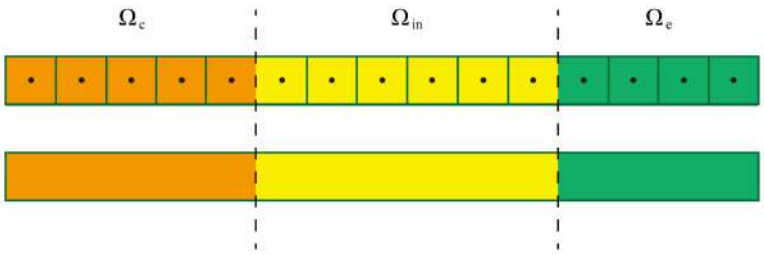

Figure 3: Different domains of Ω. The yellow part is the sub-domain Ω_{in}. The orange part is the sub-domain Ω_c. The green part is the sub-domain Ω_e.

$$\overline{\mathbf{A}}=\begin{bmatrix} \overline{\mathbf{A}}_{1,1} & \overline{\mathbf{A}}_{1,2} & \cdots & \overline{\mathbf{A}}_{1,N+N_c} \\ \overline{\mathbf{A}}_{2,1} & \overline{\mathbf{A}}_{2,2} & \cdots & \overline{\mathbf{A}}_{2,N+N_c} \\ \vdots & \vdots & \ddots & \vdots \\ \overline{\mathbf{A}}_{N+N_c,1} & \overline{\mathbf{A}}_{N+N_c,2} & \cdots & \overline{\mathbf{A}}_{N+N_c,N+N_c} \end{bmatrix}, \quad \overline{\mathbf{d}}=\begin{bmatrix} \overline{\mathbf{d}}_1 \\ \overline{\mathbf{d}}_2 \\ \vdots \\ \overline{\mathbf{d}}_{N+N_c} \end{bmatrix}. \tag{4.1}$$

$\overline{\mathbf{A}}_{i,j}$ represents the action of the material point $\mathbf{x}_j$ on material point $\mathbf{x}_i$, when we calculate $\overline{\mathbf{f}}=\overline{\mathbf{A}}\overline{\mathbf{d}}$, due to $\overline{\mathbf{d}}_i(t)=\mathbf{h}_i(t)$, $\mathbf{x}_i\in\Omega_c$, therefore, the calculation of $\overline{\mathbf{d}}_i$ is meaningless and not considered in the matrix. To ensure that the matrix is square and the calculation of $\overline{\mathbf{A}}_{j,i}\overline{\mathbf{d}}_i$ is usually less than $\mathcal{O}(N\log N)$, we can move it to the other side of the equation. So we just need to calculate $\mathbf{f}=\mathbf{A}d$ where

$$\mathbf{A}=\begin{bmatrix} \mathbf{A}_{1,1} & \mathbf{A}_{1,2} & \cdots & \mathbf{A}_{1,N} \\ \mathbf{A}_{2,1} & \mathbf{A}_{2,2} & \cdots & \mathbf{A}_{2,N} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{A}_{N,1} & \mathbf{A}_{N,2} & \cdots & \mathbf{A}_{N,N} \end{bmatrix}, \quad d=\begin{bmatrix} d_1 \\ d_2 \\ \vdots \\ d_N \end{bmatrix}. \tag{4.2}$$

All material points belong to the area Ω_s. Due to the existence of horizon size δ, $\mathbf{A}_{i,j}=0$, $|j-i|>N_\delta$, $\mathbf{A}$ becomes a banded matrix, which is implied by

$$\mathbf{A}=\begin{bmatrix} \mathbf{A}_{1,1} & \cdots & \mathbf{A}_{1,N_\delta+1} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{A}_{N_\delta+1,1} & \ddots & \mathbf{A}_{N_\delta+1,N_\delta+1} & \ddots & \ddots & \mathbf{0} & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \mathbf{A}_{N_\delta+2,N_\delta+2} & \ddots & \ddots & \mathbf{0} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ddots & \mathbf{0} & \ddots & \ddots & \ddots & \ddots & \ddots & \mathbf{0} \\ \mathbf{0} & \ddots & \ddots & \mathbf{0} & \ddots & \ddots & \mathbf{A}_{N-N_\delta,N-N_\delta} & \ddots & \mathbf{A}_{N-N_\delta,N} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{A}_{N,N-N_\delta} & \cdots & \mathbf{A}_{N,N} \end{bmatrix} \tag{4.3}$$

with

$$\mathbf{A}_{ij}=\begin{bmatrix} A_{i,j}^{uu} & A_{i,j}^{uv} & A_{i,j}^{uw} & A_{i,j}^{u\theta_x} & A_{i,j}^{u\theta_y} & A_{i,j}^{u\theta_z} \\ A_{i,j}^{vu} & A_{i,j}^{vv} & A_{i,j}^{vw} & A_{i,j}^{v\theta_x} & A_{i,j}^{v\theta_y} & A_{i,j}^{v\theta_z} \\ A_{i,j}^{wu} & A_{i,j}^{wv} & A_{i,j}^{ww} & A_{i,j}^{w\theta_x} & A_{i,j}^{w\theta_y} & A_{i,j}^{w\theta_z} \\ A_{i,j}^{\theta_x u} & A_{i,j}^{\theta_x v} & A_{i,j}^{\theta_x w} & A_{i,j}^{\theta_x\theta_x} & A_{i,j}^{\theta_x\theta_y} & A_{i,j}^{\theta_x\theta_z} \\ A_{i,j}^{\theta_y u} & A_{i,j}^{\theta_y v} & A_{i,j}^{\theta_y w} & A_{i,j}^{\theta_y\theta_x} & A_{i,j}^{\theta_y\theta_y} & A_{i,j}^{\theta_y\theta_z} \\ A_{i,j}^{\theta_z u} & A_{i,j}^{\theta_z v} & A_{i,j}^{\theta_z w} & A_{i,j}^{\theta_z\theta_x} & A_{i,j}^{\theta_z\theta_y} & A_{i,j}^{\theta_z\theta_z} \end{bmatrix}$$

$$=\begin{bmatrix} \frac{C_{ax}}{\xi_{ij}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \frac{C_s}{\xi_{ij}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\frac{C_s}{2}\mathrm{sgn} \\ \mathbf{0} & \mathbf{0} & \frac{C_s}{\xi_{ij}} & \mathbf{0} & \frac{C_s}{2}\mathrm{sgn} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & \frac{C_t}{\xi_{ij}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & -\frac{C_s}{2}\mathrm{sgn} & \mathbf{0} & \frac{C_{by}}{\xi_{ij}}-\frac{C_s}{4}\xi_{ij} & \mathbf{0} \\ \mathbf{0} & \frac{C_s}{2}\mathrm{sgn} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \frac{C_{bz}}{\xi_{ij}}-\frac{C_s}{4}\xi_{ij} \end{bmatrix}, \tag{4.4a}$$

$$\mathbf{A}_{ii}=\begin{bmatrix} A_{i,i}^{uu} & A_{i,i}^{uv} & A_{i,i}^{uw} & A_{i,i}^{u\theta_x} & A_{i,i}^{u\theta_y} & A_{i,i}^{u\theta_z} \\ A_{i,i}^{vu} & A_{i,i}^{vv} & A_{i,i}^{vw} & A_{i,i}^{v\theta_x} & A_{i,i}^{v\theta_y} & A_{i,i}^{v\theta_z} \\ A_{i,i}^{wu} & A_{i,i}^{wv} & A_{i,i}^{ww} & A_{i,i}^{w\theta_x} & A_{i,i}^{w\theta_y} & A_{i,i}^{w\theta_z} \\ A_{i,i}^{\theta_x u} & A_{i,i}^{\theta_x v} & A_{i,i}^{\theta_x w} & A_{i,i}^{\theta_x\theta_x} & A_{i,i}^{\theta_x\theta_y} & A_{i,i}^{\theta_x\theta_z} \\ A_{i,i}^{\theta_y u} & A_{i,i}^{\theta_y v} & A_{i,i}^{\theta_y w} & A_{i,i}^{\theta_y\theta_x} & A_{i,i}^{\theta_y\theta_y} & A_{i,i}^{\theta_y\theta_z} \\ A_{i,i}^{\theta_z u} & A_{i,i}^{\theta_z v} & A_{i,i}^{\theta_z w} & A_{i,i}^{\theta_z\theta_x} & A_{i,i}^{\theta_z\theta_y} & A_{i,i}^{\theta_z\theta_z} \end{bmatrix}$$

$$=\sum_{j=1}^{M_i}\begin{bmatrix} -\frac{C_{ax}}{\xi_{ij}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & -\frac{C_s}{\xi_{ij}} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\frac{C_s}{2}\mathrm{sgn} \\ \mathbf{0} & \mathbf{0} & -\frac{C_s}{\xi_{ij}} & \mathbf{0} & \frac{C_s}{2}\mathrm{sgn} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & -\frac{C_t}{\xi_{ij}} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{0} & \frac{C_s}{2}\mathrm{sgn} & \mathbf{0} & -\frac{C_{by}}{\xi_{ij}}-\frac{C_s}{4}\xi_{ij} & \mathbf{0} \\ \mathbf{0} & -\frac{C_s}{2}\mathrm{sgn} & \mathbf{0} & \mathbf{0} & \mathbf{0} & -\frac{C_{bz}}{\xi_{ij}}-\frac{C_s}{4}\xi_{ij} \end{bmatrix}, \tag{4.4b}$$

where $i\neq j$, $\mathrm{sgn}=\mathrm{sgn}\left(x_j-x_i\right)$, M_i represents the number of family members at material point $\mathbf{x}_i$.

Due to the stiffness matrix, $\mathbf{A}$ is not a Toeplitz matrix. To use the fast meshfree method, decomposing $\mathbf{A}$ into ten new matrices $\mathbf{A}^{uu}$, $\mathbf{A}^{vv}$, $\mathbf{A}^{ww}$, $\mathbf{A}^{\theta_x\theta_x}$, $\mathbf{A}^{\theta_y\theta_y}$, $\mathbf{A}^{\theta_z\theta_z}$, $\mathbf{A}^{v\theta_z}$, $\mathbf{A}^{w\theta_y}$, $\mathbf{A}^{\theta_y w}$, $\mathbf{A}^{\theta_z v}$, we have

$$\mathbf{A}^{uu}=\begin{bmatrix} A_{1,1}^{uu} & A_{1,2}^{uu} & \cdots & A_{1,N}^{uu} \\ A_{2,1}^{uu} & A_{2,2}^{uu} & \cdots & A_{2,N}^{uu} \\ \vdots & \vdots & \ddots & \vdots \\ A_{N,1}^{uu} & A_{N,2}^{uu} & \cdots & A_{N,N}^{uu} \end{bmatrix}. \tag{4.5}$$

Similarly, we can get $\mathbf{A}^{vv}$, $\mathbf{A}^{ww}$, $\mathbf{A}^{\theta_x\theta_x}$, $\mathbf{A}^{\theta_y\theta_y}$, $\mathbf{A}^{\theta_z\theta_z}$, $\mathbf{A}^{v\theta_z}$, $\mathbf{A}^{w\theta_y}$, $\mathbf{A}^{\theta_y w}$, $\mathbf{A}^{\theta_z v}$. Decompose d into six vectors $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$, $\boldsymbol{\theta_x}$, $\boldsymbol{\theta_y}$, $\boldsymbol{\theta_z}$, we observe that

$$\mathbf{u}=\begin{bmatrix} u_1 & u_2 & \cdots & u_n \end{bmatrix}^T. \tag{4.6}$$

Similarly, we can get $\mathbf{v}$, $\mathbf{w}$, $\boldsymbol{\theta_x}$, $\boldsymbol{\theta_y}$, $\boldsymbol{\theta_z}$, and through the above decomposition, we can decompose $\mathbf{f}=\mathbf{A}d$ as follows

$$\mathbf{f}^u=\mathbf{A}^{uu}\mathbf{u}, \qquad \mathbf{f}^v=\mathbf{A}^{vv}\mathbf{v}+\mathbf{A}^{v\theta_z}\boldsymbol{\theta_z}, \qquad \mathbf{f}^w=\mathbf{A}^{ww}\mathbf{w}+\mathbf{A}^{w\theta_y}\boldsymbol{\theta_y}, \tag{4.7a}$$

$$\mathbf{f}^{\theta_x}=\mathbf{A}^{\theta_x\theta_x}\boldsymbol{\theta_x}, \qquad \mathbf{f}^{\theta_y}=\mathbf{A}^{\theta_y w}\mathbf{w}+\mathbf{A}^{\theta_y\theta_y}\boldsymbol{\theta_y}, \qquad \mathbf{f}^{\theta_z}=\mathbf{A}^{\theta_z v}\mathbf{v}+\mathbf{A}^{\theta_z\theta_z}\boldsymbol{\theta_z}, \tag{4.7b}$$

with $\mathbf{f}$ can be merged by $\mathbf{f}^u$, $\mathbf{f}^v$, $\mathbf{f}^w$, $\mathbf{f}^{\theta_x}$, $\mathbf{f}^{\theta_y}$, $\mathbf{f}^{\theta_z}$.

Let's take $\mathbf{A}^{uu}$ as an example to analyze the structure of ten decomposed submatrices. Due to the limitation of δ, each material point in the beam model interacts with the surrounding points at most $\mathbf{A}^{uu}$ can be expressed as the following banded matrix

$$\mathbf{A}^{uu}=\begin{bmatrix} -\sum\limits_j \dfrac{C_{ax}}{\xi_{1,j}} & \cdots & \dfrac{C_{ax}}{\xi_{1,1+N_\delta}} & \mathbf{0} & \cdots & \cdots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \dfrac{C_{ax}}{\xi_{1+N_\delta,1}} & \ddots & -\sum\limits_j \dfrac{C_{ax}}{\xi_{1+N_\delta,j}} & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & -\sum\limits_j \dfrac{C_{ax}}{\xi_{N-1-N_\delta,j}} & \ddots & \dfrac{C_{ax}}{\xi_{N-N_\delta,N}} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \cdots & \cdots & \cdots & \dfrac{C_{ax}}{\xi_{N,N-N_\delta}} & \cdots & -\sum\limits_j \dfrac{C_{ax}}{\xi_{N,j}} \end{bmatrix}, \tag{4.8}$$

where the block bandwidth is $2N_\delta+1$.

Furthermore, due to the material points being evenly distributed, it's easy to see that $A_{i,j}^{uu}$ does not depend on the position of $\mathbf{x}_i$ or $\mathbf{x}_j$ but on the distance x_i-x_j. However, for the values on the main diagonal of the matrix, the influence area of $\mathbf{x}_i\in\Omega_{in}$ is complete,

$M_i = 2N_\delta$, but the family members of $\mathbf{x}_i \in \Omega_e$ are incomplete, so we only need to repair $A^{uu}_{i,i}$, $x_i \in \Omega_e$ on the main diagonal of the matrix, and then get a Toeplitz matrix $\mathbf{B}^{uu}$ and a patch matrix $\mathbf{C}^{uu}$, given by

$$\mathbf{C}^{uu} = \mathbf{B}^{uu} - \mathbf{A}^{uu}. \tag{4.9}$$

Similarly, we can get the Toeplitz matrices structure matrices $\mathbf{B}^{vv}$, $\mathbf{B}^{ww}$, $\mathbf{B}^{\theta_x\theta_x}$, $\mathbf{B}^{\theta_y\theta_y}$, $\mathbf{B}^{\theta_z\theta_z}$, $\mathbf{B}^{v\theta_z}$, $\mathbf{B}^{w\theta_y}$, $\mathbf{B}^{\theta_y w}$, $\mathbf{B}^{\theta_z v}$, and the patch matrices $\mathbf{C}^{vv}$, $\mathbf{C}^{ww}$, $\mathbf{C}^{\theta_x\theta_x}$, $\mathbf{C}^{\theta_y\theta_y}$, $\mathbf{C}^{\theta_z\theta_z}$, $\mathbf{C}^{v\theta_z}$, $\mathbf{C}^{w\theta_y}$, $\mathbf{C}^{\theta_y w}$, $\mathbf{C}^{\theta_z v}$.

4.1.2 Acceleration of form $\mathbf{f} = \mathbf{A}d$

Let's take $\mathbf{B}^{uu}\mathbf{u}$ as an example, due to $\mathbf{B}^{uu}$ is Toeplitz matrix, so matrix $\mathbf{B}^{uu}_N$ can be embedded into a circulant matrix $\overline{\mathbf{B}}^{uu}_{2N}$.

$$\overline{\mathbf{B}}^{uu} = \begin{pmatrix} \mathbf{B}^{uu} & \tilde{\mathbf{B}}^{uu} \\ \tilde{\mathbf{B}}^{uu} & \mathbf{B}^{uu} \end{pmatrix}, \tag{4.10}$$

where the matrix $\tilde{\mathbf{B}}^{uu}_N$ is defined by

$$\tilde{\mathbf{B}}^{uu}_N = \begin{pmatrix} \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} & B^{uu}_{N-N_\delta+1,N-1} & \cdots & B^{uu}_{N,N-1} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} & \ddots & \mathbf{0} & \ddots & \ddots & B^{uu}_{N,N-N_\delta} \\ \mathbf{0} & \ddots & \mathbf{0} & \mathbf{0} & \ddots & \ddots & \mathbf{0} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ \mathbf{0} & \ddots & \mathbf{0} & \ddots & \ddots & \ddots & \ddots & \ddots & \mathbf{0} \\ B^{uu}_{1,1+N_\delta} & \ddots & \ddots & \mathbf{0} & \ddots & \ddots & \mathbf{0} & \ddots & \mathbf{0} \\ \vdots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \ddots & \vdots \\ B^{uu}_{1,2} & \cdots & B^{uu}_{N_\delta,2} & \mathbf{0} & \cdots & \mathbf{0} & \mathbf{0} & \cdots & \mathbf{0} \end{pmatrix}. \tag{4.11}$$

Let $\mathbf{g}^{uu}_{2N}$ be the first column vector of $\overline{\mathbf{B}}^{uu}$, as

$$\mathbf{g}^{uu} = \begin{bmatrix} B^{uu}_{1,1} & \cdots & B^{uu}_{1+N_\delta,1} & \mathbf{0} & \cdots & \mathbf{0} & B^{uu}_{1,1+N_\delta} & \cdots & B^{uu}_{1,2} \end{bmatrix}^T. \tag{4.12}$$

This reduces the storage and evaluation of the matrix $\mathbf{B}^{uu}$ from $\mathcal{O}(N^2)$ to $\mathcal{O}(2N_\delta+1) = \mathcal{O}(N)$. And then, we need to extend the vectors $\mathbf{u}$ to $\overline{\mathbf{u}}_{2N}$, as

$$\overline{\mathbf{u}} = \begin{bmatrix} \mathbf{u} & \mathbf{0} & \cdots & \mathbf{0} \end{bmatrix}^T. \tag{4.13}$$

$\hat{\mathbf{p}}^{uu}$ can be obtainbed by FFT [42] in Algorithm 4.1, and then

$$\mathbf{p}^{uu} = \mathbf{A}^{uu}\mathbf{u} = \mathbf{B}^{uu}\mathbf{u} - \mathbf{C}^{uu}\mathbf{u} = \hat{\mathbf{p}}^{uu} - \mathbf{C}^{uu}\mathbf{u} \tag{4.14}$$

Algorithm 4.1 Fast matrix-vector multiplcation.

Require: Extended matrix $\mathbf{g}^{uu}$; Extended vector $\overline{\mathbf{u}}$
$\hat{\mathbf{g}}^{uu} = FFT(\mathbf{g}^{uu})$
$\hat{\mathbf{u}} = FFT(\overline{\mathbf{u}})$
$\mathbf{w}^{uu} = \hat{\mathbf{g}}^{uu} \bullet \hat{\mathbf{u}}$
$\hat{\mathbf{w}}^{uu} = IFFT(\mathbf{w}^{uu})$
while $i = 1:N$ **do**
$\hat{\mathbf{p}}^{uu}(i) = \hat{\mathbf{w}}^{uu}(i)$
end while

with $\mathbf{C}^{uu}$ is only non-zero on the main diagonal of the matrix, the computation of $\mathbf{C}^{uu}\mathbf{u}$ will not exceed $\mathcal{O}(N\log N)$, for which, the computation of Eq. (4.14) is only affected by $\mathbf{B}^{uu}\mathbf{u}$.

Similarly, we can get $\mathbf{p}^{vv}$, $\mathbf{p}^{ww}$, $\mathbf{p}^{\theta_x\theta_x}$, $\mathbf{p}^{\theta_y\theta_y}$, $\mathbf{p}^{\theta_z\theta_z}$, $\mathbf{p}^{v\theta_z}$, $\mathbf{p}^{w\theta_y}$, $\mathbf{p}^{\theta_y w}$, $\mathbf{p}^{\theta_z v}$.

$$\mathbf{f}^{u} = \mathbf{p}^{uu}, \qquad \mathbf{f}^{v} = \mathbf{p}^{vv} + \mathbf{p}^{v\theta_z}, \qquad \mathbf{f}^{w} = \mathbf{p}^{ww} + \mathbf{p}^{w\theta_y}, \tag{4.15a}$$

$$\mathbf{f}^{\theta_x} = \mathbf{p}^{\theta_x\theta_x}, \qquad \mathbf{f}^{\theta_y} = \mathbf{p}^{\theta_y\theta_y} + \mathbf{p}^{\theta_y w}, \qquad \mathbf{f}^{\theta_z} = \mathbf{p}^{\theta_z\theta_z} + \mathbf{p}^{\theta_z v}. \tag{4.15b}$$

And $\mathbf{f}$ can be merged by $\mathbf{f}^{u}$, $\mathbf{f}^{v}$, $\mathbf{f}^{w}$, $\mathbf{f}^{\theta_x}$, $\mathbf{f}^{\theta_y}$, $\mathbf{f}^{\theta_z}$. Apply the above process to Algorithm 3.1, we can calculate d^{n+1}.

The numerical procedure is shown in Fig. 4. By the above method, the computational cost of $\mathbf{f} = \mathbf{A}d$ can be implemented in $\mathcal{O}(N^2)$ to $\mathcal{O}(2N\log 2N) = \mathcal{O}(N\log N)$ operations and the storage memory of $\mathbf{A}$ can be decreased from $\mathcal{O}(N^2)$ to $\mathcal{O}(2N) = \mathcal{O}(N)$ via the fast Fourier transform (FFT).

4.2 A fast meshfree method of PD for multi-beam

4.2.1 Transformation of coordinate systems and equations of motion

Eq. (2.13) are deduced in the local coordinate system. Nevertheless, in practical problems, multi-beam structures must be considered in the global coordinate system. Therefore, we need to convert the equation of motion for each beam into the same global coordinate system.

In the global coordinate system, the centreline of each straight beam has unit vectors $\overrightarrow{n_x}$, $\overrightarrow{n_y}$, $\overrightarrow{n_z}$ to know the direction of this beam, the conversion of global and local coordinates can be completed by the coordinate transformation matrix $\mathbf{H}$, which is equivalent to

$$\mathbf{x} = \mathbf{H}\mathbf{x}' \tag{4.16}$$

with

$$\mathbf{H} = \begin{bmatrix} \vec{n}_x & \vec{n}_y & \vec{n}_z \end{bmatrix}^T, \tag{4.17}$$

where $\mathbf{x}$ represents the local coordinates and $\mathbf{x}'$ represents the global coordinates.

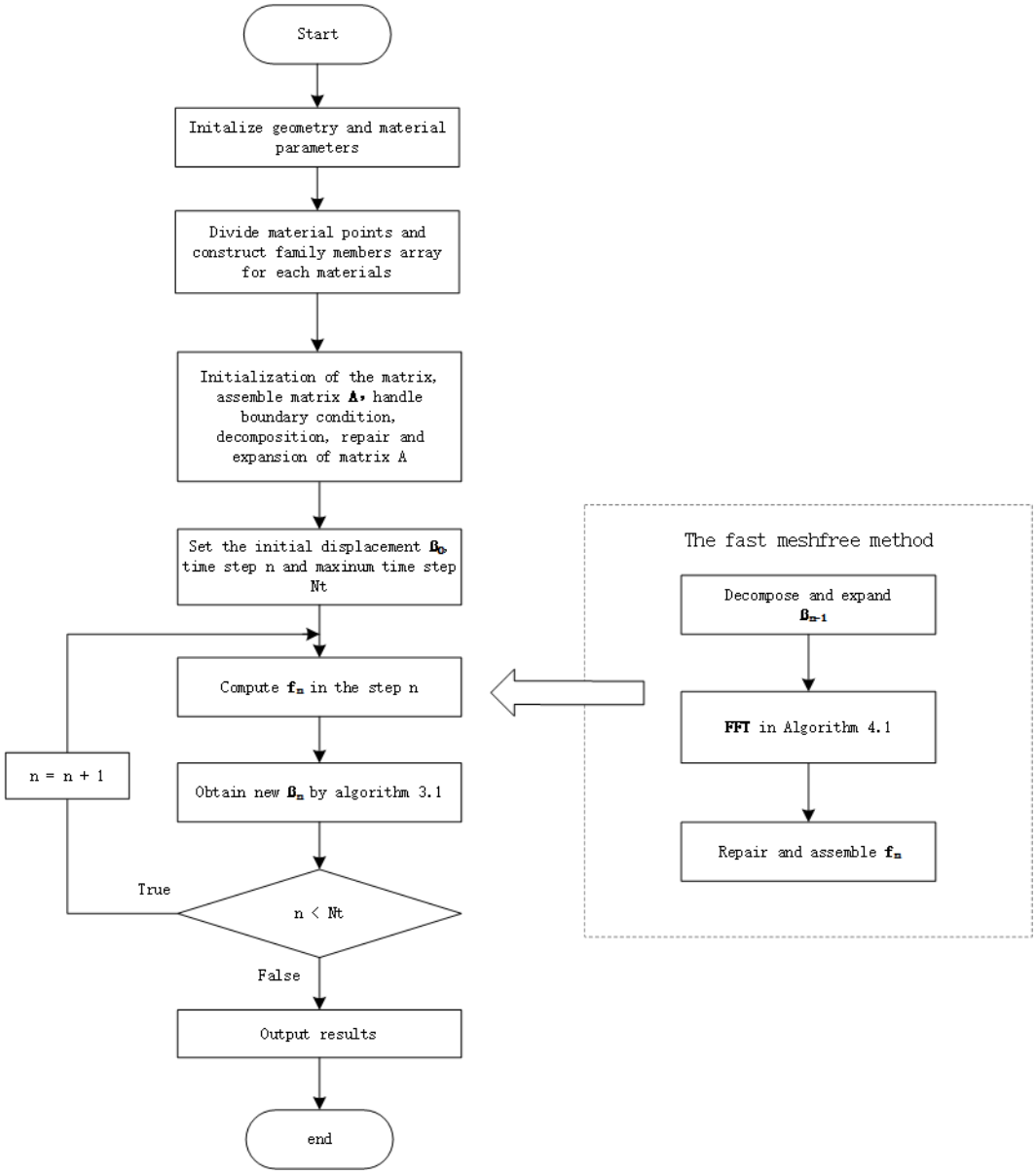

Figure 4: The fast meshfree method of single beam.

The conversion of global and local displacement vectors can also be completed by displacement transformation matrix $\mathbf{T}_{6N}$, which is equivalent to

$$d = \mathbf{T}d' \tag{4.18}$$

with

$$\mathbf{T}=\begin{bmatrix}\mathbf{K} & \mathbf{0} & \cdots & \mathbf{0}\\ \mathbf{0} & \mathbf{K} & \ddots & \vdots\\ \vdots & \ddots & \ddots & \mathbf{0}\\ \mathbf{0} & \cdots & \mathbf{0} & \mathbf{K}\end{bmatrix},\quad \mathbf{K}=\begin{bmatrix}\mathbf{H} & \mathbf{0}\\ \mathbf{0} & \mathbf{H}\end{bmatrix}, \tag{4.19}$$

where d denotes the local coordinates and d' denotes the global coordinates.

Multiply the equation by $\mathbf{T}^T$ to the left, the Eq. (2.18) can be written as

$$\mathbf{T}^T\boldsymbol{\rho}\mathbf{T}\mathbf{T}^T\ddot{d}=\mathbf{T}^T\mathbf{A}\mathbf{T}\mathbf{T}^Td+\mathbf{T}^T\mathbf{b} \tag{4.20}$$

with

$$\boldsymbol{\rho}'=\mathbf{T}^T\boldsymbol{\rho}\mathbf{T},\quad \mathbf{A}'=\mathbf{T}^T\mathbf{A}\mathbf{T},\quad d'=\mathbf{T}^Td,\quad \mathbf{b}'=\mathbf{T}^T\mathbf{b}, \tag{4.21}$$

and

$$\mathbf{f}'=\mathbf{A}'d'=\mathbf{T}^T\mathbf{A}\mathbf{T}\mathbf{T}^Td=\mathbf{T}^T\mathbf{f}. \tag{4.22}$$

In the global coordinate system, invoking Eq. (4.21), the equation of motion given in Eq. (4.20) can be converted as

$$\boldsymbol{\rho}'\ddot{d}'=\mathbf{A}'d'+\mathbf{b}'. \tag{4.23}$$

4.2.2 Handling of joint points

As shown in Fig. 5, beams A and B are joined at one point. First, we discretize each beam into material points. And then, for the point at the connection, delete unnecessary point, and keep only one material point as $A^{(i,j)}$, this point denotes the joint point for two beams.

The joint point $A^{(i,j)}$ has family members in two beams. Different beams may have different geometrical and material properties, so if $A^{(i,j)}$ interacts with a material point in beam A, the parameters of $A^{(i,j)}$ are same as beam A. Else the parameters of $A^{(i,j)}$ are same as beam B.

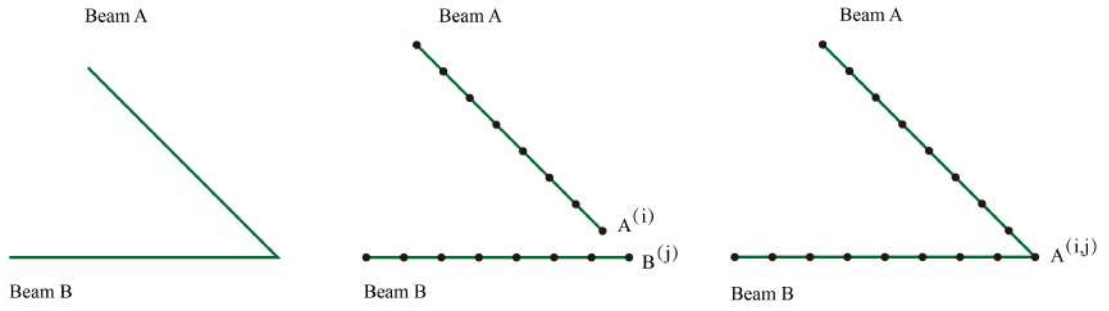

Figure 5: The joint point of two beams.

4.2.3 Implementation of the fast meshfree method for multi-beam

First, we can get the initial whole displacement d_0' in global coordinate system, we can split d_0' to get the displacement $d_0'^{(i)}$ of each beam in global coordinate system. And then, we can get the displacement $d_0^{(i)}$ of each beam in local coordinate system by

$$d_0^{(i)} = \mathbf{T}^{(i)} d_0'^{(i)}, \tag{4.24}$$

where $\mathbf{T}^{(i)}$ stands for the displacement transformation matrix of the ith beam. According to the method in Section 4.1, we can get the internal force for each beam, record the internal force of the ith beam in the local coordinate system as $\mathbf{f}^{(i)}$, and the internal force in a global coordinate system can be determined by

$$\mathbf{f}'^{(i)} = \mathbf{T}^{(i)^T} \mathbf{f}^{(i)}. \tag{4.25}$$

And we can assemble the whole internal force $\mathbf{F}'$ of multiple beams by $\mathbf{f}'^{(i)}$ in the global coordinate system. Applying the above process to Algorithm 3.1, we can calculate the whole displacement d' of multiple beams in the global coordinate system.

Repeat the above process, and the fast meshfree method of peridynamic multiple beams can be realized.

5 Numerical results

This section presents numerical examples that demonstrate the capabilities of the proposed method. The computational cost of static problems is compared between the traditional meshfree method and the fast meshfree method. The ADR method is used to solve these static problems. We implement these methods in Matlab and run all experiments on a workstation with Intel Xeon Gold 6240(2.6GHz) logical processors and 2048G installed memory. All the experiments are done using double precision. The beams mentioned in the following experiments are made of steel material. The material parameters are Young's modulus $E = 2\times10^{11}\mathrm{N/m^2}$, the shear modulus $G = 1\times10^{11}\mathrm{N/m^2}$, and the mass density $\rho = 7850\mathrm{kg/m^2}$.

5.1 Straight beam structure subjected to a constant concentrated load

In this experiment, a concentrated load is subjected to straight beam structure. As shown in Fig. 6(left), the length of the beam is $L = 1$ m, and $A = 0.1\times0.1$ m^2 is the area of the square cross-section. One end of the straight beam is fixed in the wall, and the other end is applied with concentrated loading $F_{z'} = 5\times10^3$ N.

In the peridynamic model, each beam is uniformly divided into 101, 201, 401, 801, 1601, 3201, 6401, and 12801 material points. To implement the fixed end of the beam, we add N_δ fictitious points at the end where the horizon size $\delta = 0.03$m. As shown in

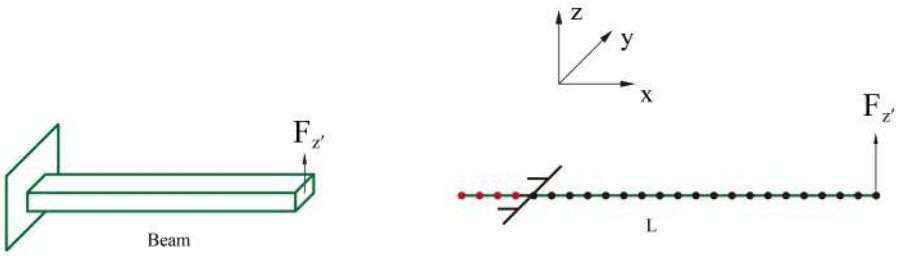

Figure 6: Straight beam subjected to concentrated load (left) geometry, (right) PD discretization.

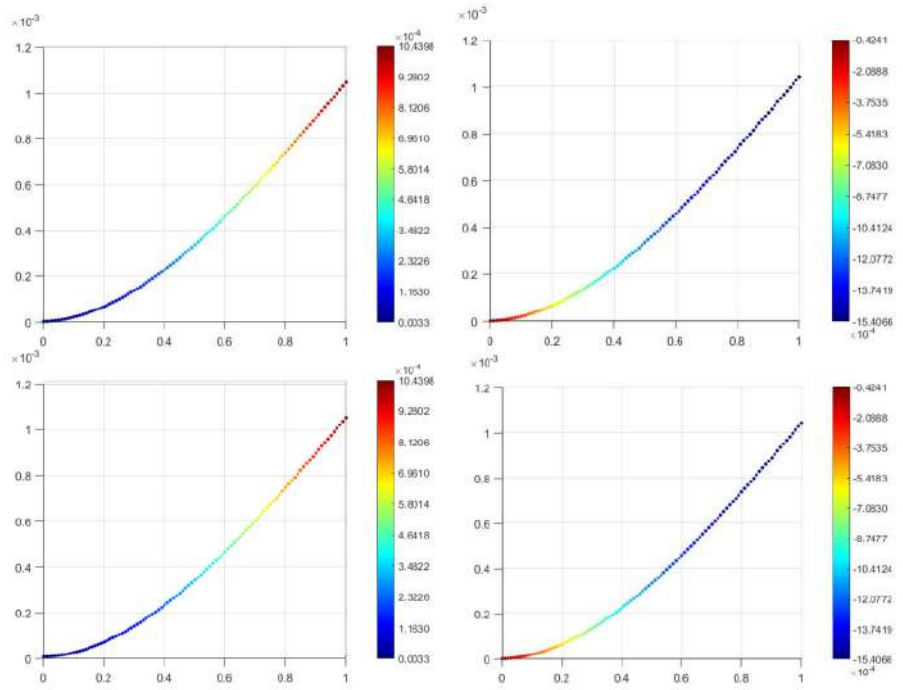

Figure 7: Variation of $\mathbf{w}$,$\boldsymbol{\theta}_y$ in straight beam structure, (above) the meshfree method, (below) the fast meshfree method.

Fig. 6(right), the black points are real material points, and the red are fictitious material points.

In terms of PD prediction results, due to the small deformation assumption, only $\mathbf{w}$ and $\boldsymbol{\theta}_y$ are two nonzero DOFs of the beam. Variations of the displacement $\mathbf{w}$ and the rotation $\boldsymbol{\theta}_y$ along a straight beam are observed in Fig. 7. The two figures above show the results of the meshfree method, and the two figures below show the results of the fast meshfree method. We can see that the results of the fast meshfree method agree well with the traditional meshfree method.

Table 1: Computational time of the meshfree method and fast meshfree method in straight beam structure.

Material points of beam		401	801	1601	3201	6401	12801
Time steps		22000	30000	40000	57000	80000	110000
Run time	Meshfree method	12s	1m1s	6m26s	40m54s	4h1m	23h11m
	Fast meshfree method	32s	1m19s	3m27s	10m20s	31m53s	1h31m

In terms of computational cost, Table 1 displays the computational time of the meshfree method and the fast meshfree method. Because the time is too short when there are few material points, the data is recorded when the number of material points gets greater than 401. To study the performance of the fast meshfree method in more complex cases, we fix the size of δ, increase the number of material points and give a corresponding time comparison.

As illustrated in Table 1, with the initial number of material points, the fast meshfree method has no obvious computational advantage over the meshfree method and requires longer computing time. This is caused by the large proportion of matrix assembly time. With the increase of material points, the computing advantages of the fast meshfree method are more and more obvious. In the ADR method, the number of time steps to achieve stability increases with the amount of material points. Excluding the influence of iteration steps, the growth of computing time is consistent with the comparison between $\mathcal{O}(N^2)$ and $\mathcal{O}(N\log N)$ in the analysis. As shown in Table 1, when the quantity of material points increases from N to $2N$, the computational cost of the meshfree method will increase by $2^2=4$, while the computational cost of the fast meshfree method will increase by $2\log 2N/\log N$, for each time step. The final increasing rate should be the mutiplication of the increasing rate each time step and the total number of time steps.

5.2 2D frame subjected to a constant concentrated load

In this example, we study a 2D frame subjected to a concentrated load. As shown in Fig. 8(left), the steel frame consists of two beams. Both beams have the same length

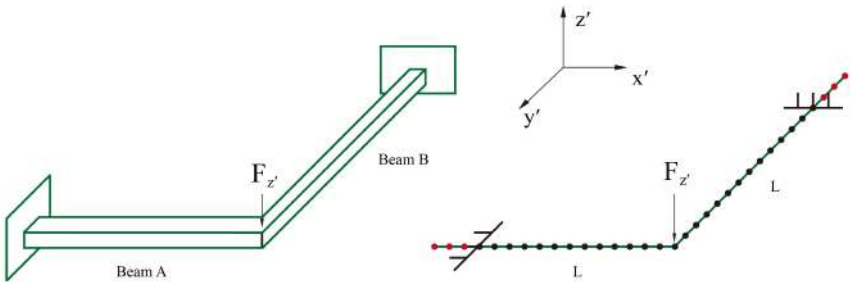

Figure 8: 2D frame subjected to concentrated load (left) geometry, (right) PD discretization.

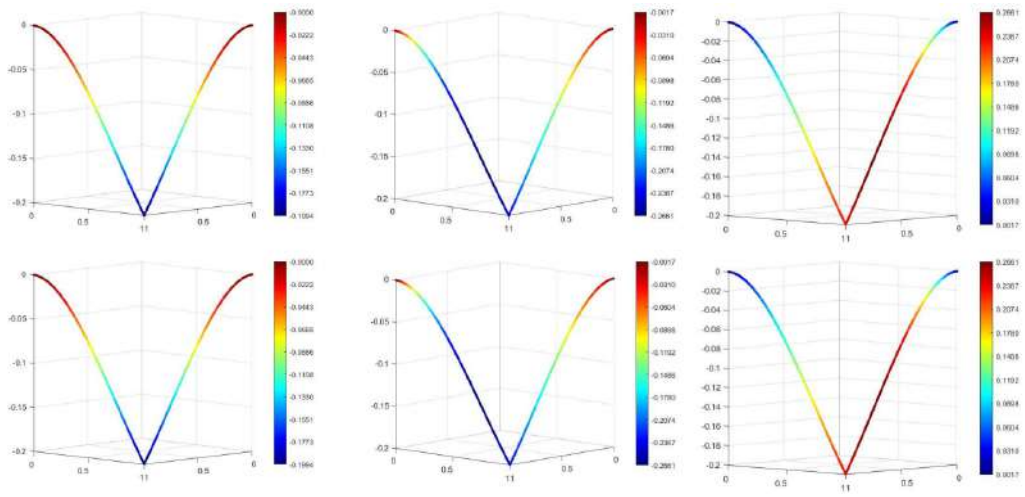

Figure 9: Variation of $\mathbf{w}$, $\boldsymbol{\theta}_x$, $\boldsymbol{\theta}_y$ in the 2D frame, (above) the meshfree method, (below) the fast meshfree method.

($L=1$m) and square cross-section ($A=0.1\times0.1$m^2). The frame is clamped at both ends and applies concentrated force $F_{z'}=-3\times10^6$N at the connection of two beams $(L,L,0)$.

From the perspective of the peridynamic model, each beam is uniformly divided into 201, 401, 801, 1601, 3201, and 6401 material points. To impose the fixed ends of the frame, we add N_δ fictitious points at both ends. As shown in Fig. 8(right), the black points are real material points, and the red points are fictitious. Here, horizon size $\delta=0.015$m.

As shown in Fig. 9, by the small deformation assumption, the beam has three non-zero DOFs $\mathbf{w}$, $\boldsymbol{\theta}_x$ and $\boldsymbol{\theta}_y$. The figures above show the displacement and rotation change of the meshfree method in the global coordinate system, and the figures below show the displacement and rotation change of the fast meshfree method. From Fig. 9, we can see that the results of the two methods are consistent. For the purpose of verifying the performance of the fast meshfree method in more complex cases, we fix δ and increase the number of material points. Table 2 shows the corresponding computational time comparison of the two methods.

As we can see from Table 2, as the number of the material points goes up, the number of time steps for the ADR method to achieve stability also increases correspondingly.

Table 2: Computational time of the meshfree method and fast meshfree method in 2D frame.

Material points of each beam		201	401	801	1601	3201	6401
Time steps		15000	20000	30000	40000	55000	80000
Run time	Meshfree method	28s	3m13s	19m36s	1h59m	12h29m	4d1h
	Fast meshfree method	29s	1m3s	2m46s	7m2s	19m45s	53m1s

At first, the computational time of the meshfree and fast meshfree methods are almost the same. However, with the densification of the material points, the difference in the computational efficiency between the two methods gradually appears. As the conclusion in the analysis, the computational time of the meshfree method increases by $\mathcal{O}(N^2)$, while the computational time of the fast meshfree method increases by $\mathcal{O}(N\log N)$. Finally, there is a huge time gap between several days and less than one hour.

5.3 Multi-beam structure subjected to a constant concentrated load

In this section, an example of a multi-beam structure subjected to concentrated load is solved by applying the present method. As shown in Fig. 10(right), the multi-beam structure consists of twelve beams. $L_1 = L_2 = 1$m, $L_3 = L_4 = 0.8$m, $H = 0.8$m, both beams have the same square cross-section $A = 0.1 \times 0.1\text{m}^2$. The multi-beam structure subjected to concentrated loading $F_{x'} = 5 \times 10^3$ N at joint point 8.

In this example, all the beams are divided with the same number of material points. Similarly, we add N_δ fictitious points at the end of beams 1, 2, 3, 4, 9, 10, 11, and 12 to implement the fixed ends of the structure, along negative z direction. For beams 1, 2, 3 and 4, horizon size $\delta = 0.0244$m. For beams 5, 6, 7 and 8, horizon size $\delta = 0.0241$m, and horizon size $\delta = 0.0363$m for beams 9, 10, 11 and 12.

A joint point belongs to several beams at the same time. In each beam, the volume of the joint point is different, so it is difficult to calculate the force density. Therefore, for Eq. (2.18) of each beam, we multiply the volume of the material points on both sides of the equation so that we can convert the force density into a force and solve the above problem.

Figs. 11 and 12 represent the displacement $\mathbf{u}$, $\mathbf{v}$, $\mathbf{w}$ and rotation $\boldsymbol{\theta}_x$, $\boldsymbol{\theta}_y$, $\boldsymbol{\theta}_z$ variations along each beam. The above three figures show the results of the meshfree method, and the below three figures are the results of the fast meshfree method. For 6 DOFs, we can get consistent results with the meshfree method and fast meshfree method.

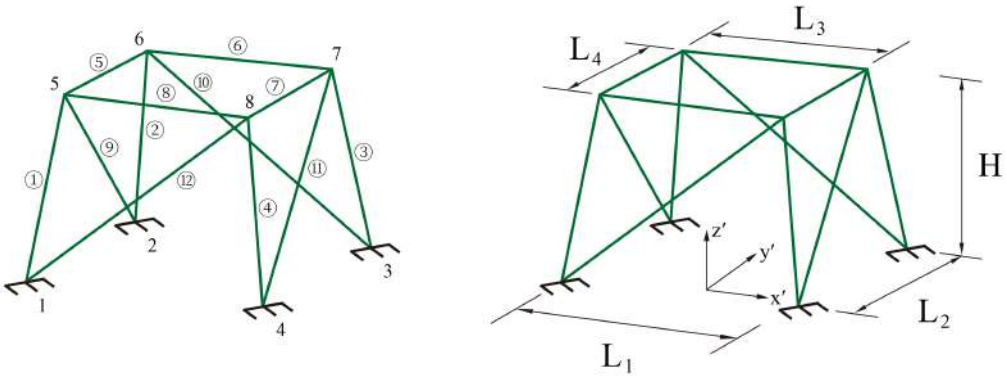

Figure 10: Multi-beam structure subjected to concentrated load (left) geometry, (right) PD discretization.

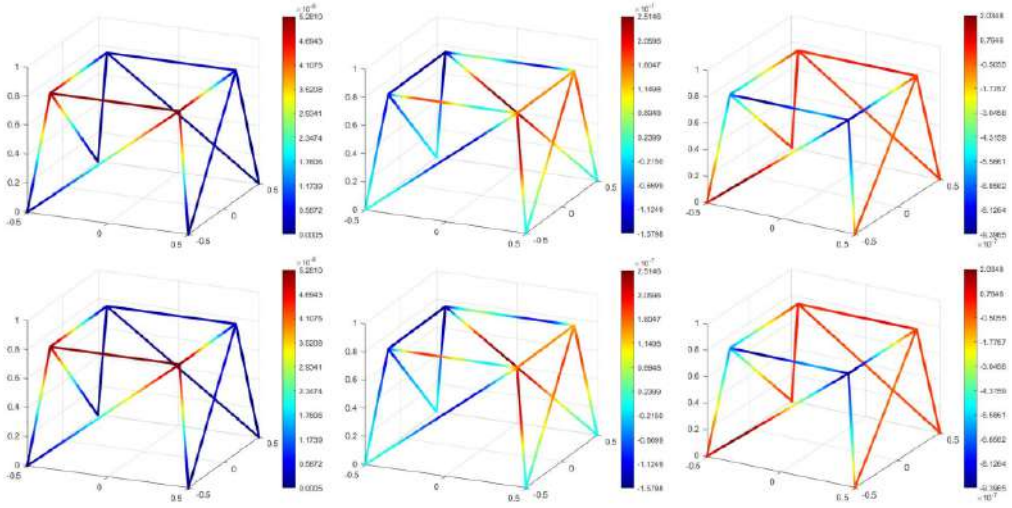

Figure 11: Variation of displacement **u**, **v**, **w** in multi-beam structure, (above) the meshfree method, (below) the fast meshfree method.

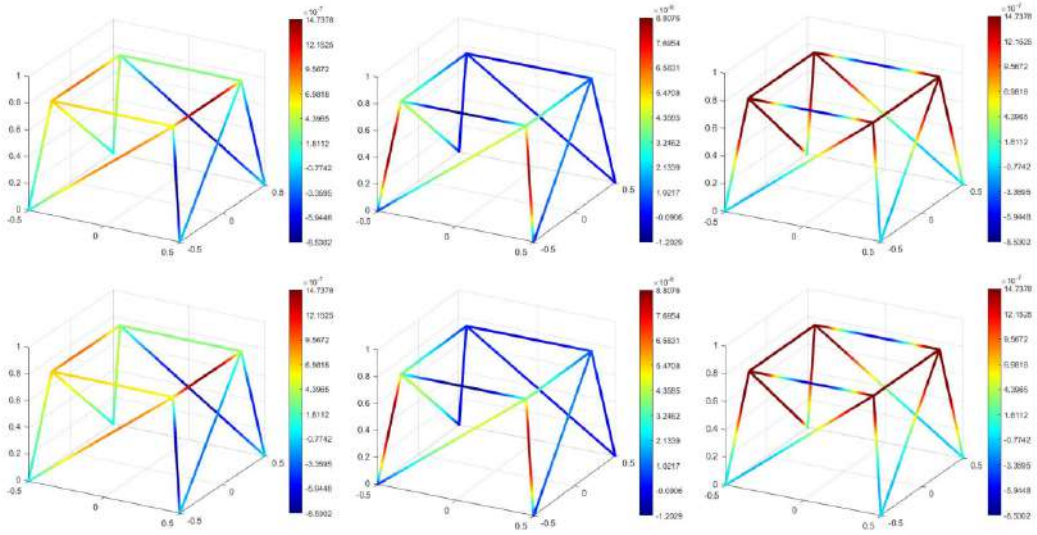

Figure 12: Variation of θ_x, θ_y, θ_z in multi-beam structure, (above) the meshfree method, (below) the fast meshfree method.

Table 3 shows the computational time of the meshfree method and the fast meshfree method. We fix the size of δ and divide each beam into 101, 201, 401, 801, 1601, 3201, and 6401 material points and give a corresponding time comparison. Only the computational time within ten days is recorded in the table. Similarly, the computational time is affected by time steps, and the growth of computational time is consistent with the comparison

Table 3: Computational time of the meshfree method and fast meshfree method in multi-beam structure.

Material points of each beam		101	201	401	801	1601	3201	6401
Time steps		5000	9000	18000	34000	68000	128000	250000
Run time	Meshfree method	8s	59s	6m13s	47m49s	6h24m	2d2h	-
	Fast meshfree method	17s	1m23s	5m	17m20s	1h1m	3h59m	14h35m

between $\mathcal{O}(N^2)$ and $\mathcal{O}(N\log N)$ in our analysis. We can see that when the number of material points reaches a certain number, the computational cost of the meshfree method can not be accepted. Therefore, the fast meshfree method becomes critical.

6 Conclusions

The main contribution of this study is the development of a fast meshfree method based on the Timoshenko beam model of the linear bond-based peridynamic. For a single beam structure, the stiffness matrix is decomposed according to boundary conditions and refined to meet the Toeplitz structure, reducing storage requirements from $\mathcal{O}(N^2)$ to $\mathcal{O}(N)$. The fast Fourier transform is used for PD integral calculation, reducing computational costs from $\mathcal{O}(N^2)$ to $\mathcal{O}(N\log N)$ compared to the traditional meshfree method. For multi-beam structures, vector splitting and coordinate system transformation are used to transform them into a combination of single beam problems.

This study proposes a fast meshfree method that considers the matrix structure and can be applied to the Timoshenko beam model of peridynamics. The method enables numerical simulations of complex beam structures, and several numerical examples are presented, including the straight beam structure, the 2D straight beam frame, and the multi-beam structure with a concentrated load. Computational efficiency comparisons between the fast meshfree method and the meshfree method demonstrate a significant reduction in calculation time, from several days to less than an hour. The results are consistent with theory, and ongoing efforts aim to apply the fast meshfree method to peridynamic plate and shell models and complex offshore platform structures.

Acknowledgements

This work was supported by the Fundamental Research Funds for the Central Universities (Nos. 202042008 and 202264006) and the National Natural Science Foundation of China (No. 52071306). The work was carried out at the Marine Big Data Center of the Institute for Advanced Ocean Study of the Ocean University of China.

References

[1] S. A. SILLING, *Reformulation of elasticity theory for discontinuities and long-range forces,* J. Mech. Phys. Solids, 48(1) (2000), pp. 175–209.

[2] A. JAVILI, R. MORASATA, E. OTERKUS AND S. OTERKUS, *Peridynamics review,* Math. Mech. Solids, 24 (2019), pp. 3714–3739.

[3] S. A. SILLING, M. EPTON, O. WECKNER, J. XU AND E. ASKARI, *Peridynamic states and constitutive modeling,* J. Elasticity, 88(2) (2007), pp. 151–184.

[4] M. NOWAK, K. MULEWSKA, A. AZAROV, L. KURPASKA AND A. USTRZYCK, *A peridynamic elasto-plastic damage model for ion-irradiated materials,* Int. J. Mech. Sci., 237 (2023), 107806.

[5] T. ZHANG AND X. ZHOU, *A modified axisymmetric ordinary state-based peridynamics with shear deformation for elastic and fracture problems in brittle solids,* Euro. J. Mech. Solids, 77 (2019), 103810.

[6] E. MADENCI AND S. OTERKUS, *Ordinary state-based peridynamics for plastic deformation according to von Mises yield criteria with isotropic hardening,* J. Mech. Phys. Solid, 86 (2016), pp. 192–219.

[7] E. MADENCI AND S. OTERKUS, *Ordinary state-based peridynamics for thermoviscoelastic deformation,* Eng. Fract. Mech., 175 (2017), pp. 31–45.

[8] C. T. NGUYEN AND S. OTERKUS, *Ordinary state-based peridynamic model for geometrically nonlinear analysis,* Eng. Fract. Mech., 224(1) (2019), 106750.

[9] Y. HUANG, S. OTERKUS, H. HOU, E. OTERKUS, Z. WEI AND S. ZHANG, *Peridynamic model for visco-hyperelastic material deformation in different strain rates,* Continuum Mech. Thermodyn, 34 (2022), pp. 977–1011.

[10] G. ONGARO, R. BERTANI, U. GALVANETTO, A. PONTEFISSO AND M. ZACCARIOTTO, *A multiscale peridynamic framework for modelling mechanical properties of polymer-based nanocomposite,* Eng. Fract. Mech., 274 (2022), 108751.

[11] J. M. ZHAN, X. H. YAO AND F. HAN, *An approach of peridynamic modeling associated with molecular dynamics for fracture simulation of particle reinforced metal matrix composites,* Compos. Struct., 250 (2020), 112613.

[12] A. CHAIKITTIRATANA AND N. WATTANASAKULPONG, *Dynamic loadings induced vibration of third order shear deformable FG-CNTRC beams: Gram-Schmidt-Ritz method,* Adv. Appl. Math. Mech., 14 (2022), pp. 816–841.

[13] S. PORNPEERAKEAT AND S. KATAWAETHWARAG, *Chebyshev-Lagrange multipliers technique for vibration analysis of functionally graded material beams using various beam theories,* Adv. Appl. Math. Mech., 14 (2022), pp. 871–892.

[14] S. A. SILLING, M. ZIMMERMANN AND R. ABEYARATNE, *Deformation of a peridynamic bar,* J. Elasticity, 73(1) (2003), pp. 173–190.

[15] J. OGRADY AND J. T. FOSTER, *Peridynamic beams: a non-ordinary, state-based model,* Int. J. Solid Struct., 51(18) (2014), pp. 3177–3183.

[16] C. DIYAROGLU, E. OTERKUS AND S. OTERKUS, *An Euler-bernoulli Beam Formulation in an Ordinary State-based Peridynamic Framework,* Math. Mech. Solids, 24(2) (2019), 1081286517728424.

[17] S. LIU, G. FANG, J. LIANG, M. FU, B. WANG AND X. YAN, *Study of three-dimensional Euler-Bernoulli beam structures using element-based peridynamic model,* Euro. J. Mech.-A/Solids, 86 (2021), 104186.

[18] Z. YANG, S. OTERKUS AND E. OTERKUS, *Peridynamic formulation for Timoshenko beam,* Procedia Structural Integrity, 28 (4) (2020), pp. 464–471.

[19] C. DIYAROGLU, E. OTERKUS, S. OTERKUS AND E. MADENCI, *Peridynamics for bending of beams and plates with transverse shear deformation,* Int. J. Solid Struct., 69-70 (2015), pp. 152–168.
[20] C. T. NGUYEN AND S. OTERKUS, *Peridynamics formulation for beam structures to predict damage in offshore structures,* Ocean Eng., 173 (2019), pp. 244–267.
[21] R. BAI, H. NACEUR, G. LIANG, J. ZHAO, J. YI, X. LI, S. YUAN, H. PU AND J. LUO, *Alleviation of shear locking in the Peridynamic Timoshenko beam model using the developed mixed formulation method,* Comput. Particle Mech., 10 (2023), pp. 627–643.
[22] Z. YANG, E. OTERKUS, C. T. NGUYEN AND S. OTERKUS, *Implementation of peridynamic beam and plate formulations in finite element framework,* Continuum Mech. Thermodyn., 31 (2019), pp. 301–315.
[23] Z. YANG, E. OTERKUS AND S. OTERKUS, *Peridynamic higher-order beam formulation,* J. Peridyn. Nonlocal Model., 3(1) (2021), pp. 67–83.
[24] Z. YANG, E. OTERKUS AND S. OTERKUS, *Analysis of functionally graded Timoshenko beams by using peridynamics,* J. Peridyn. Nonlocal Model, 3(2) (2021), pp. 148–166.
[25] G. SHEN, Y. XIA, P. HU AND G. ZHENG, *Construction of peridynamic beam and shell models on the basis of the micro-beam bond obtained via interpolation method,* Euro. J. Mech.-A/Solids, 86 (2021), 104174.
[26] G. SHEN, Y. XIA, W. LI, G. ZHENG AND P. HU, *Modeling of peridynamic beams and shells with transverse shear effect via interpolation method,* Comput. Methods Appl. Mech. Eng., 378 (2021), 113716.
[27] T. BODE1, C. WEISSENFELS AND P. WRIGGERS, *Peridynamic Galerkin method: an attractive alternative to finite elements,* Comput. Mech., 70 (2022), pp. 723–743.
[28] R. CAO, M. CHEN, Y. QI, J. SHI AND X. YIN, *Analysis of (shifted) piecewise quadratic polynomial collocation for nonlocal diffusion model,* Appl. Numer. Math., 185 (2022).
[29] S. SILLING AND E. ASKARI, *A meshfree method based on the peridynamic model of solid mechanics,* Comput. Struct., 83 (2005), pp. 1526–1535.
[30] J. LU AND Y. NIE, *A reduced-order fast reproducing kernel collocation method for nonlocal models with inhomogeneous volume constraints,* Comput. Math. Appl., 121 (2022), pp. 52–61.
[31] J. LU, M. YANG AND Y. NIE, *Convergence analysis of Jacobi spectral collocation methods for weakly singular nonlocal diffusion equations with volume constraints,* Appl. Math. Comput., 431 (2022), 127345.
[32] S. ZHANG AND Y. NIE, *Localized Chebyshev and MLS collocation methods for solving 2D steady state nonlocal diffusion and peridynamic equations,* Math. Comput. Simul., 206 (2023), pp. 264–285.
[33] X. TIAN AND Q. DU, *Analysis and comparison of different approximations to nonlocal diffusion and linear peridynamic equations,* SIAM J. Numer. Anal., 51 (2013), pp. 3458–3482.
[34] Q. DU AND J. YANG, *Asymptotically compatible Fourier spectral approximations of nonlocal allen-cahn equation,* SIAM J. Numer. Anal., 54(3) (2016), pp. 1899–1919.
[35] J. ZHANG, F. HANB, Z. YANG AND J. CUI, *Coupling of an atomistic model and bond-based peridynamic model using an extended Arlequin framework,* Comput. Methods Appl. Mech. Eng., 403 (2023), 115663.
[36] G. ZHENG, L. LI, F. HAN, Y. XIA, G. SHEN AND P. HU, *Coupled peridynamic model for geometrically nonlinear deformation and fracture analysis of slender beam structures,* Int. J. Numer. Methods Eng., 123 (2022), 12.
[37] Y. XIA, X. MENG, G. ZHENG AND G. SHEN, *The Coupling Approach of Isogeometric Analysis and Peridynamics for Plane Problem with Non-Uniform Control Net,* J. Peridyn. Nonlocal Model.,

4 (4) (2022), pp. 475–500.
[38] Y. Xia, H. Wang, G. Zheng, G. Shen and P. Hu, *Discontinuous Galerkin isogeometric analysis with peridynamic model for crack simulation of shell structure,* Comput. Methods Appl. Mech. Eng., 398 (2022), 115193.
[39] S. Jafarzadeh, F. Mousavi, A. Larios and F. Bobaru, *A general and fast convolution-based method for peridynamics: Applications to elasticity and brittle fracture,* Comput. Methods Appl. Mech. Eng., 392 (2022), 114666.
[40] S. Jafarzadeh, L. Wang, A. Larios and F. Bobaru, *A fast convolution-based method for peridynamic transient diffusion in arbitrary domains,* Comput. Methods Appl. Mech. Eng., 375 (2021), 113633.
[41] D. A. Abdoh, B. B. Yin, *Venkatesh R. Kodur and K. M. Liew, Computationally efficient and effective peridynamic model for cracks and fractures in homogeneous and heterogeneous materials,* Comput. Methods Appl. Mech. Eng., 399 (2022), 115318.
[42] H. Wang and H. Tian, *A fast Galerkin method with efficient matrix assembly and storage for a peridynamic model,* J. Comput. Phys., 231 (2012), pp. 7730–7738.
[43] H. Wang and H. Tian, *A fast and faithful collocation method with efficient matrix assembly for a two-dimensional nonlocal diffusion model,* Comput. Methods Appl. Mech. Eng., 273 (2014), pp. 19–36.
[44] X. Zhang and H. Wang, *A fast collocation method for a static bond-based linear peridynamic model,* Comput. Methods Appl. Mech. Eng., 311 (2016), pp. 280–303.
[45] C. Wang and H. Wang, *A fast collocation method for a variable-coefficient nonlocal diffusion model,* J. Comput. Phys., 330 (2017), pp. 114–126.
[46] H. Liu, A. Cheng and H. Wang, *A fast discontinuous Galerkin method for a bond-based linear peridynamic model discretized on a locally refined composite mesh,* J. Sci. Comput., 76 (2018), pp. 913–942.
[47] X. Zhang, X. Li, A. Cheng and H. Wang, *A preconditioned fast collocation method for a linear bond-based peridynamic model,* Adv. Difference Equations, 2020 (2020), 244.
[48] X. Zhang, A. Cheng and H. Wang, *A Preconditioned Fast Collocation Method for a Linear Nonlocal Diffusion Model in Convex Domains,* IEEE Access, 8 (2020), pp. 182366–18237510.
[49] E. Madenci and E. Oterkus, Peridynamic Theory and Its Applications, Springer, New York, 17 (2014).
[50] S. P. Timoshenko and J. N. Goodier, Theory of Elasticity, McGraw-Hill, London New York, 3 (1970), pp. 403–408.
[51] B. Kilic and E. Madenci, *An adaptive dynamic relaxation method for quasi-static simulations using the peridynamic theory,* Theor. Appl. Fract. Mech., 53(3) (2010), pp. 194–204.