Chapter 7
Numerical Solution Method

The peridynamic (PD) equation of motion is an integro-differential equation, which is not usually amenable for analytical solutions. Therefore, its solution is constructed by using numerical techniques for spatial and time integrations. The spatial integration can be performed by using the collocation method of a meshless scheme due to its simplicity. Hence, the domain can be divided into a finite number of subdomains, with integration or collocation (material) points associated with specific volumes (Sect. 7.1). Associated with a particular material point, numerical implementation of spatial integration involves the summation of the volumes of material points within its horizon. However, the volume of each material point may not be embedded in the horizon in its entirety, i.e., the material points located near the surface of the horizon may have truncated volumes. As a result, the volume integration over the horizon may be incorrect if the entire volume of each material point is included in the numerical implementation. Therefore, a volume correction factor is necessary to correct for the extra volume. A volume correction procedure required for such a case is described in Sect. 7.2.

Numerical time integration can be performed by using backward and forward difference explicit integration schemes, although other techniques are also applicable, such as the Adams-Bashforth method, Adams-Moulton method, and Runge–Kutta method. If an explicit integration scheme is adopted, a stability criterion on the value of the incremental time step is necessary to ensure convergence. Details of the time integration scheme and stability criterion are given in Sects. 7.3 and 7.4, respectively.

The PD equation of motion includes the inertial terms; it is not directly applicable to static and quasi-static problems. Hence, a special treatment is required so that the system will converge to a static condition in a short amount of computational time. Although there are different techniques available for this purpose, adaptive dynamic relaxation (ADR) can be utilized (Kilic and Madenci 2010), and it is described in detail in Sect. 7.5.

Another important concern when using a numerical technique is the convergence of the results. It is important to use optimum values of parameters to

E. Madenci and E. Oterkus, *Peridynamic Theory and Its Applications*,
DOI 10.1007/978-1-4614-8465-3_7, © Springer Science+Business Media New York 2014

achieve sufficient accuracy within a suitable amount of computational time. The determination of such PD parameters is described in Sect. 7.6.

As described in Sect. 4.2, the interactions associated with material points close to the free surfaces are truncated, and this causes a reduction of the stiffness of these material points. In other words, these material points do not represent the accurate bulk behavior and require a correction. The correction can be imposed by introducing surface correction factors that can be directly inserted in the equation of motion, as described in Sect. 7.7.

Solution to the PD equation of motion requires initial conditions on displacement and velocity, as well as boundary conditions, as described in Sect. 2.7. Numerical implementations of the initial and boundary conditions are given in Sect. 7.8. If necessary, the introduction of a pre-existing crack is rather straightforward, as explained in Sect. 7.9. Moreover, as a result of extreme loading conditions, such as high velocity boundary conditions, large displacement boundary constraints, impact problems, etc., unexpected damage patterns may occur, especially close to the boundary region. This problem can be overcome by defining "no fail zones" and is also explained in Sect. 7.9. The measure of local damage for crack growth is explained in Sect. 7.10.

Each material point has its own particular family members defined by its horizon. For domains including a large number of material points, it is important to utilize an efficient process to search and establish the family members, and store their information, as presented in Sect. 7.11. Utilization of parallel computing is a crucial process to achieve significant computational efficiency. A brief discussion on parallel computing is given in Sect. 7.12.

The development of a solution algorithm for the PD equation of motion may involve the following steps:

- Specify the input parameters and initialize the matrices.
- Determine a stable time step size for the time integration. If the analysis involves the adaptive dynamic relaxation technique, the time step size is equal to 1.
- Generate the material points.
- Determine the material points inside the horizon of each material point and store them.
- In the case of a pre-existing crack problem, remove the PD interactions that are passing through the crack surfaces.
- Compute the surface correction factor for each material point.
- Apply initial conditions.
- If the analysis involves the adaptive dynamic relaxation technique, construct the stable mass matrix.
- Start time integration.
- Apply boundary conditions.
- Compute the total PD interaction forces acting on each material (collocation) point.
- Terminate the PD interaction if its stretch exceeds the critical stretch.

- If the analysis involves the adaptive dynamic relaxation technique, compute the adaptive dynamic relaxation technique parameters.
- Perform time integration to obtain displacements and velocities.

7.1 Spatial Discretization

In order to solve Eq. 2.22, a collocation method is adopted and the numerical treatment involves the discretization of the domain of interest into subdomains, as shown in Fig. 7.1. The domain can be discretized into subdomains by employing line subdomains for one-dimensional geometries, triangular and quadrilateral subdomains for two-dimensional regions, and hexahedron, tetrahedron, and wedge subdomains for three-dimensional regions, as shown in Fig. 7.2.

After discretizing the domain, the collocation points are placed in the subdomains, as shown in Fig. 7.1. With this meshless discretization scheme, the volume integration in Eq. 2.22 can be approximated as

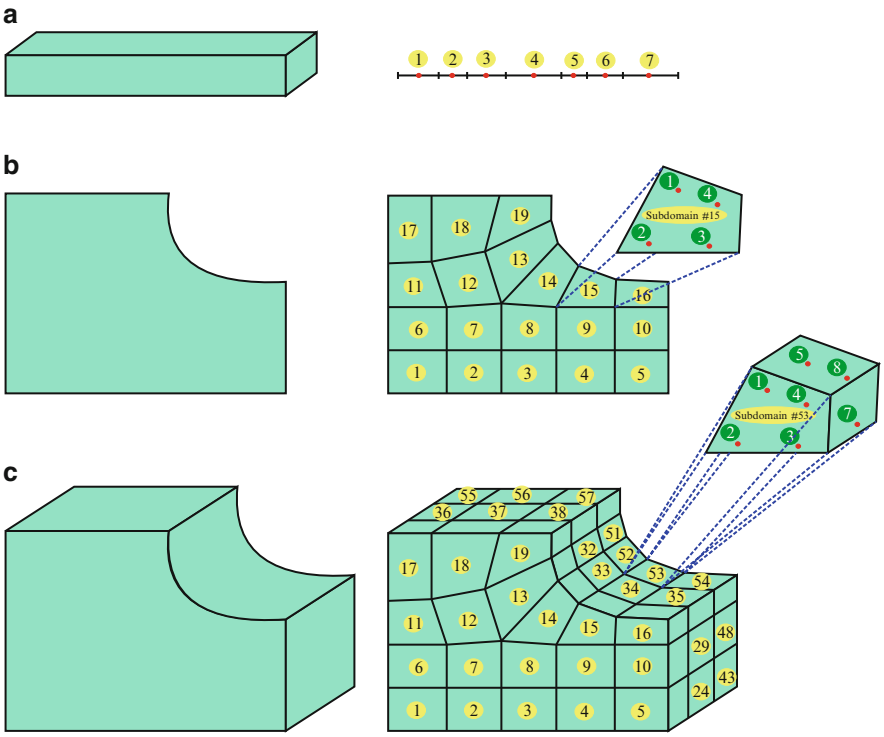


Fig. 7.1 Discretization of the domain of interest for (**a**) one-dimensional, (**b**) two-dimensional, and (**c**) three-dimensional regions

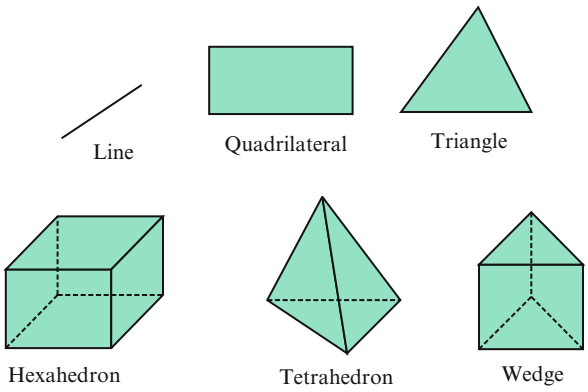


Fig. 7.2 Subdomain shapes for one-, two-, and three-dimensional geometries

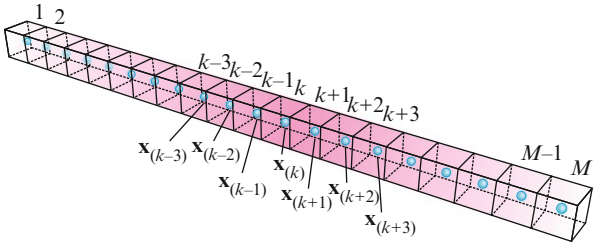


Fig. 7.3 Discretization and material points in a one-dimensional region

$$\rho\left(\mathbf{x}_{(k)}\right)\ddot{\mathbf{u}}\left(\mathbf{x}_{(k)},t\right)=\sum_{e=1}^{N}\sum_{j=1}^{N_e}w_{(j)}\left[\mathbf{t}\left(\mathbf{u}\left(\mathbf{x}_{(j)},t\right)-\mathbf{u}\left(\mathbf{x}_{(k)},t\right),\mathbf{x}_{(j)}-\mathbf{x}_{(k)}\right)\right.$$
$$\left.-\mathbf{t}\left(\mathbf{u}\left(\mathbf{x}_{(k)},t\right)-\mathbf{u}\left(\mathbf{x}_{(j)},t\right),\mathbf{x}_{(k)}-\mathbf{x}_{(j)}\right)\right]V_{(j)}+\mathbf{b}\left(\mathbf{x}_{(k)},t\right), \tag{7.1}$$

where N is the number of subdomains within the horizon and N_e is the number of collocation points in e^{th} subdomain. The position vectors $\mathbf{x}_{(k)}$ and $\mathbf{x}_{(j)}$ represent the locations of the k^{th} and j^{th} collocation (integration) points, respectively. The parameter $w_{(j)}$ is the integration weight of point $\mathbf{x}_{(j)}$. The integration points can be determined as described by Kilic (2008). For a uniform grid of cubic subdomains with one integration point at the center, the weight, $w_{(j)}$, is equal to unity. The volume of the j^{th} cubic subdomain is denoted by $V_{(j)}$

For instance, in the case of a one-dimensional region, the discretization is achieved with M cubic subdomains in which Gaussian integration (collocation) points represent the material points, as shown in Fig. 7.3. Integration points are located at the center of each cubic subdomain with a weight of unity. Note that the truncation error in Eq. 7.1 for this particular case is on the order of $O(\Delta^2)$, where Δ represents spacing between integration (material) points. If a discontinuity is present in the structure, then the error becomes $O(\Delta)$ (Silling and Askari 2005).

7.2 Volume Correction Procedure

Associated with a material point, $\mathbf{x}_{(k)}$, the numerical integration over its horizon is approximated by considering the entire volume of each material point, $\mathbf{x}_{(j)}$, within its horizon. As illustrated in Fig. 7.4, in the case of a uniform spacing of Δ between the material points leading to cubic subdomains $(w_{(j)} = 1)$, and for a horizon of $\delta = 3\Delta$, this numerical approximation leads to summation of the material point volumes within the range of $\xi_{(k)(j)} = |\mathbf{x}_{(j)} - \mathbf{x}_{(k)}| < \delta$. As implemented in the EMU code (Silling 2004), this approximation can be improved by considering the entire volume of the material points within the range of $\xi_{(k)(j)} = |\mathbf{x}_{(j)} - \mathbf{x}_{(k)}| < \delta - r$, in which $r = \Delta/2$, the distance from the surface of the horizon. For the material points that are within the range of $\delta - r < \xi_{(k)(j)} < \delta$, a volume correction factor of $v_{c(j)} = (\delta + r - \xi_{(k)(j)})/2r$ is introduced by using a linear variation between a factor of 1 and ½ depending on the family member's location with respect to the horizon boundary. For the material points that are located outside of this region, the volume correction factor is $v_{c(j)} = 1$.

Thus, the discretized equation of motion, Eq. 7.1, for material point $\mathbf{x}_{(k)}$ including the volume correction can be rewritten as

$$\rho(\mathbf{x}_{(k)})\,\ddot{\mathbf{u}}(\mathbf{x}_{(k)}, t) = \sum_{e=1}^{N} \left[\mathbf{t}\left(\mathbf{u}(\mathbf{x}_{(j)}, t) - \mathbf{u}(\mathbf{x}_{(k)}, t), \mathbf{x}_{(j)} - \mathbf{x}_{(k)}\right) - \mathbf{t}\left(\mathbf{u}(\mathbf{x}_{(k)}, t) - \mathbf{u}(\mathbf{x}_{(j)}, t), \mathbf{x}_{(k)} - \mathbf{x}_{(j)}\right)\right]\left(v_{c(j)} V_{(j)}\right) + \mathbf{b}(\mathbf{x}_{(k)}, t). \tag{7.2}$$

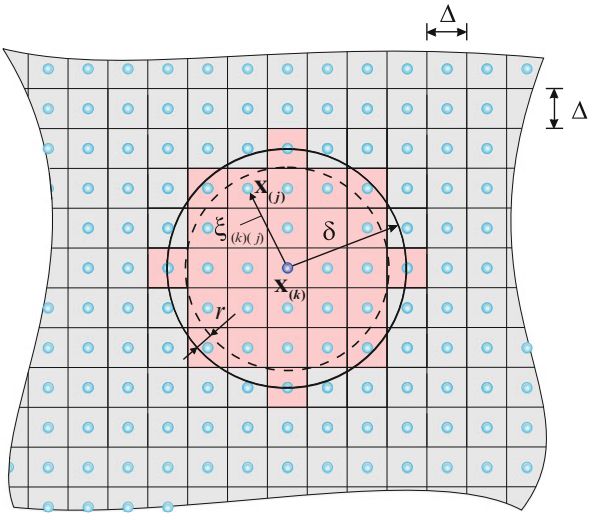


Fig. 7.4 Volume correction for the collocation points inside the horizon

7.3 Time Integration

The time integration of the PD equation of motion in Eq. 7.2 can be performed by using explicit forward and backward difference techniques (Silling 2004). If the solution to Eq. 7.2 at the n^{th} time step of Δt (i.e., $t = n\,\Delta t$) is represented as $\mathbf{u}^n_{(k)} = \mathbf{u}_{(k)}(t = n\Delta t)$, Eq. 7.2 can be rewritten for this time step in the form

$$\rho_{(k)}\ddot{\mathbf{u}}^n_{(k)} = \sum_{j=1}^{N}\left(\mathbf{t}^n_{(k)(j)} - \mathbf{t}^n_{(j)(k)}\right)\left(v_{c(j)}V_{(j)}\right) + \mathbf{b}^n_{(k)}, \tag{7.3}$$

where

$$\mathbf{t}^n_{(k)(j)} = \mathbf{t}^n_{(k)(j)}\left(\mathbf{u}^n_{(j)} - \mathbf{u}^n_{(k)}, \mathbf{x}_{(j)} - \mathbf{x}_{(k)}\right)$$

and

$$\mathbf{t}^n_{(j)(k)} = \mathbf{t}^n_{(j)(k)}\left(\mathbf{u}^n_{(k)} - \mathbf{u}^n_{(j)}, \mathbf{x}_{(k)} - \mathbf{x}_{(j)}\right)$$

represent the force density vectors between the material points located at $\mathbf{x}_{(k)}$ and $\mathbf{x}_{(j)}$. Using Eqs. 4.4 and 4.5, the force density vectors can be explicitly written as

$$\mathbf{t}^n_{(k)(j)} = \frac{\boldsymbol{\xi}_{(k)(j)} + \boldsymbol{\eta}^n_{(k)(j)}}{\left|\boldsymbol{\xi}_{(k)(j)} + \boldsymbol{\eta}^n_{(k)(j)}\right|}\left(2ad\,\delta\frac{\Lambda^n_{(k)(j)}}{\left|\boldsymbol{\xi}_{(k)(j)}\right|}\theta^n_{(k)} + 2b\delta s^n_{(k)(j)}\right) \tag{7.4a}$$

and

$$\mathbf{t}^n_{(j)(k)} = -\frac{\boldsymbol{\xi}_{(k)(j)} + \boldsymbol{\eta}^n_{(k)(j)}}{\left|\boldsymbol{\xi}_{(k)(j)} + \boldsymbol{\eta}^n_{(k)(j)}\right|}\left(2ad\,\delta\frac{\Lambda^n_{(k)(j)}}{\left|\boldsymbol{\xi}_{(k)(j)}\right|}\theta^n_{(j)} + 2b\delta s^n_{(k)(j)}\right), \tag{7.4b}$$

in which the relative position and relative displacement vectors are defined as $\boldsymbol{\xi}_{(k)(j)} = \mathbf{x}_{(j)} - \mathbf{x}_{(k)}$ and $\boldsymbol{\eta}^n_{(k)(j)} = \mathbf{u}^n_{(j)} - \mathbf{u}^n_{(k)}$. Thus, the stretch between material points located at $\mathbf{x}_{(k)}$ and $\mathbf{x}_{(j)}$ at this time step, $s^n_{(k)(j)}$, becomes

$$s^n_{(k)(j)} = \frac{\left|\boldsymbol{\xi}_{(k)(j)} + \boldsymbol{\eta}^n_{(k)(j)}\right| - \left|\boldsymbol{\xi}_{(k)(j)}\right|}{\left|\boldsymbol{\xi}_{(k)(j)}\right|}. \tag{7.5}$$

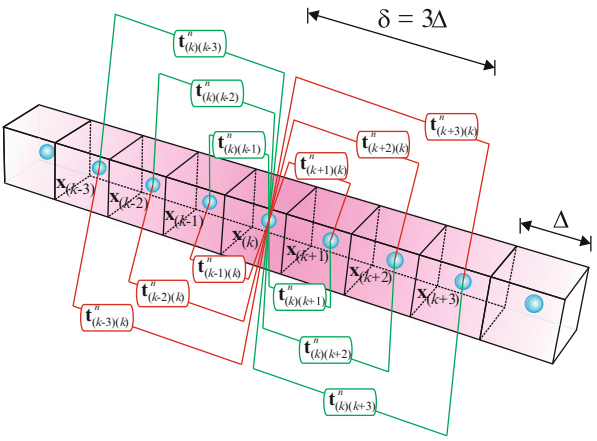


Fig. 7.5 Interaction of material points within the horizon

Furthermore, the dilatation at material points located at $\mathbf{x}_{(k)}$ and $\mathbf{x}_{(j)}$ can be computed from

$$\theta^n_{(k)} = d\delta \sum_{\ell=1}^{N} s^n_{(k)(\ell)} \Lambda^n_{(k)(\ell)} \left(\upsilon_{c(\ell)} V_{(\ell)}\right) \tag{7.6a}$$

and

$$\theta^n_{(j)} = d\delta \sum_{\ell=1}^{N} s^n_{(j)(\ell)} \Lambda^n_{(j)(\ell)} \left(\upsilon_{c(\ell)} V_{(\ell)}\right). \tag{7.6b}$$

As shown in Fig. 7.5, if the material point k interacts with other material points within a horizon of $\delta = 3\Delta$, the peridynamic equation becomes

$$\begin{aligned}
\rho_{(k)} \ddot{\mathbf{u}}^n_{(k)} = & \left(\mathbf{t}^n_{(k)(k+1)} - \mathbf{t}^n_{(k+1)(k)}\right)\left(\upsilon_{c(k+1)} V_{(k+1)}\right) \\
& + \left(\mathbf{t}^n_{(k)(k+2)} - \mathbf{t}^n_{(k+2)(k)}\right)\left(\upsilon_{c(k+2)} V_{(k+2)}\right) \\
& + \left(\mathbf{t}^n_{(k)(k+3)} - \mathbf{t}^n_{(k+3)(k)}\right)\left(\upsilon_{c(k+3)} V_{(k+3)}\right) \\
& + \left(\mathbf{t}^n_{(k)(k-1)} - \mathbf{t}^n_{(k-1)(k)}\right)\left(\upsilon_{c(k-1)} V_{(k-1)}\right) \\
& + \left(\mathbf{t}^n_{(k)(k-2)} - \mathbf{t}^n_{(k-2)(k)}\right)\left(\upsilon_{c(k-2)} V_{(k-2)}\right) \\
& + \left(\mathbf{t}^n_{(k)(k-3)} - \mathbf{t}^n_{(k-3)(k)}\right)\left(\upsilon_{c(k-3)} V_{(k-3)}\right) + \mathbf{b}^n_{(k)}.
\end{aligned} \tag{7.7}$$

After determining the acceleration of a material point at the n^{th} time step from Eq. 7.3, the velocity and displacement at the next time step can be obtained by

employing explicit forward and backward difference techniques in two steps, respectively. The first step determines the velocity at the $(n+1)^{th}$ time step using the known acceleration and the known velocity at the n^{th} time step as

$$\dot{\mathbf{u}}_{(k)}^{n+1} = \ddot{\mathbf{u}}_{(k)}^{n} \Delta t + \dot{\mathbf{u}}_{(k)}^{n}. \tag{7.8}$$

The second step determines the displacement at the $(n+1)^{th}$ time step using the velocity at the $(n+1)^{th}$ time step from Eq. 7.8 and the known displacement at the n^{th} time step as

$$\mathbf{u}_{(k)}^{n+1} = \dot{\mathbf{u}}_{(k)}^{n+1} \Delta t + \mathbf{u}_{(k)}^{n}. \tag{7.9}$$

The same procedure can be applied for other material points as well. For instance, the displacement and velocity of the $(k+1)^{th}$ material point can be obtained as

$$\mathbf{u}_{(k+1)}^{n+1} = \dot{\mathbf{u}}_{(k+1)}^{n+1} \Delta t + \mathbf{u}_{(k+1)}^{n} \tag{7.10a}$$

and

$$\dot{\mathbf{u}}_{(k+1)}^{n+1} = \ddot{\mathbf{u}}_{(k+1)}^{n} \Delta t + \dot{\mathbf{u}}_{(k+1)}^{n}. \tag{7.10b}$$

Note that the numerical error to obtain the displacement value by integrating the computed acceleration value from Eq. 7.3 is on the order of $O(\Delta t^2)$. Hence, the overall numerical error becomes $O(\Delta^2) + O(\Delta t^2)$, including the error from spatial integration (discretization). Furthermore, the overall error is $O(\Delta) + O(\Delta t^2)$, if there is any discontinuity in the structure (Silling and Askari 2005).

7.4 Numerical Stability

Although the explicit time integration scheme is straightforward, it is only conditionally stable. Therefore, a stability condition is necessary to obtain convergent results. A stability condition for the time step size, Δt, is derived based on the approach by Silling and Askari (2005). According to this approach, the standard von Neumann stability analysis can be performed by assuming a displacement variation of

$$u_{(k)}^{n} = \zeta^{n} e^{\left(\kappa k \sqrt{-1}\right)}, \tag{7.11}$$

where κ and ζ are positive real and complex numbers, respectively. The stability analysis requires that $|\zeta| \leq 1$ for all values of κ. Satisfaction of this condition is necessary so the waves do not grow unboundedly over time. By using an explicit central difference formula, Eq. 7.3 results in

$$\rho_{(k)}\left(\frac{u_{(k)}^{n+1} - 2u_{(k)}^{n} + u_{(k)}^{n-1}}{\Delta t^2}\right) = \sum_j \frac{2ad\,\delta\left(\theta_{(k)}^{n} + \theta_{(j)}^{n}\right) + 4b\delta\left(u_{(j)}^{n} - u_{(k)}^{n}\right)}{\left|\boldsymbol{\xi}_{(k)(j)}\right|} v_{c(j)} V_{(j)}, \tag{7.12}$$

where

$$\theta_{(k)}^{n} = d\delta \sum_{\ell} \frac{u_{(\ell)}^{n} - u_{(k)}^{n}}{\left|\boldsymbol{\xi}_{(\ell)(k)}\right|} \left(v_{c(\ell)} V_{(\ell)}\right) \tag{7.13a}$$

and

$$\theta_{(j)}^{n} = d\delta \sum_{\ell} \frac{u_{(\ell)}^{n} - u_{(j)}^{n}}{\left|\boldsymbol{\xi}_{(\ell)(j)}\right|} \left(v_{c(\ell)} V_{(\ell)}\right). \tag{7.13b}$$

Substituting Eqs. 7.11 in 7.12 yields

$$\rho_{(k)}\left(\frac{\zeta^{n+1} - 2\zeta^{n} + \zeta^{n-1}}{\Delta t^2}\right) e^{\left(\kappa k\sqrt{-1}\right)} = \sum_j \left(2ad\,\delta \frac{\left(\theta_{(k)}^{n} + \theta_{(j)}^{n}\right)}{\left|\boldsymbol{\xi}_{(k)(j)}\right|} + 4b\delta \frac{\zeta^{n}\left(e^{\left(\kappa j\sqrt{-1}\right)} - e^{\left(\kappa k\sqrt{-1}\right)}\right)}{\left|\boldsymbol{\xi}_{(k)(j)}\right|}\right) \left(v_{c(j)} V_{(j)}\right), \tag{7.14}$$

where

$$\theta_{(k)}^{n} = d\delta \sum_{\ell} \frac{\zeta^{n} e^{\left(\kappa \ell\sqrt{-1}\right)} - \zeta^{n} e^{\left(\kappa k\sqrt{-1}\right)}}{\left|\boldsymbol{\xi}_{(\ell)(k)}\right|} \left(v_{c(\ell)} V_{(\ell)}\right) \tag{7.15a}$$

and

$$\theta_{(j)}^{n} = d\delta \sum_{\ell} \frac{\zeta^{n} e^{\left(\kappa \ell\sqrt{-1}\right)} - \zeta^{n} e^{\left(\kappa j\sqrt{-1}\right)}}{\left|\boldsymbol{\xi}_{(\ell)(j)}\right|} \left(v_{c(\ell)} V_{(\ell)}\right). \tag{7.15b}$$

Rearranging Eq. 7.14 results in

$$\rho_{(k)}\left(\frac{\zeta^2-2\zeta+1}{\Delta t^2}\right) = \sum_j \left(2ad\,\delta\frac{\left(\bar{\theta}^n_{(k)}+\bar{\theta}^n_{(j)}\right)}{\left|\boldsymbol{\xi}_{(k)(j)}\right|}+4b\delta\frac{\left(e^{\left(\kappa(j-k)\sqrt{-1}\right)}-1\right)}{\left|\boldsymbol{\xi}_{(k)(j)}\right|}\right)\zeta\left(v_{c(j)}V_{(j)}\right), \tag{7.16}$$

where

$$\bar{\theta}^n_{(k)} = d\delta\sum_\ell \frac{\left(e^{\left(\kappa(\ell-k)\sqrt{-1}\right)}-1\right)}{\left|\boldsymbol{\xi}_{(\ell)(k)}\right|}\left(v_{c(\ell)}V_{(\ell)}\right) \tag{7.17a}$$

and

$$\bar{\theta}^n_{(j)} = d\delta\sum_\ell \frac{\left(e^{\left(\kappa(\ell-j)\sqrt{-1}\right)}-1\right)}{\left|\boldsymbol{\xi}_{(\ell)(j)}\right|}\left(v_{c(\ell)}V_{(\ell)}\right). \tag{7.17b}$$

Since exponential terms can be written in terms of sine and cosine functions, and sine is an odd function, Eq. 7.16 can be rewritten as

$$\rho_{(k)}\left(\frac{\zeta^2-2\zeta+1}{\Delta t^2}\right) = \sum_j \left(2ad\,\delta\frac{\left(\bar{\theta}^n_{(k)}+\bar{\theta}^n_{(j)}\right)}{\left|\boldsymbol{\xi}_{(k)(j)}\right|}+4b\delta\frac{(\cos(\kappa(j-k))-1)}{\left|\boldsymbol{\xi}_{(k)(j)}\right|}\right)\zeta\left(v_{c(j)}V_{(j)}\right), \tag{7.18}$$

where

$$\bar{\theta}^n_{(k)} = d\delta\sum_\ell \frac{\cos(\kappa(\ell-k))-1}{\left|\boldsymbol{\xi}_{(\ell)(k)}\right|}\left(v_{c(\ell)}V_{(\ell)}\right) \tag{7.19a}$$

and

$$\bar{\theta}^n_{(j)} = d\delta\sum_\ell \frac{\cos(\kappa(\ell-j))-1}{\left|\boldsymbol{\xi}_{(\ell)(j)}\right|}\left(v_{c(\ell)}V_{(\ell)}\right). \tag{7.19b}$$

By defining

$$M_\kappa = -\frac{1}{2}\sum_j \left(2ad\,\delta \frac{\left(\bar{\theta}^n_{(k)} + \bar{\theta}^n_{(j)}\right)}{\left|\boldsymbol{\xi}_{(k)(j)}\right|} + 4b\delta \frac{(\cos(\kappa(j-k)) - 1)}{\left|\boldsymbol{\xi}_{(k)(j)}\right|} \right) \left(v_{c(j)} V_{(j)}\right), \quad (7.20)$$

Equation 7.18 takes the form

$$\zeta^2 - 2\left(1 - \frac{M_\kappa \Delta t^2}{\rho_{(k)}}\right)\zeta + 1 = 0. \quad (7.21)$$

The solution to the quadratic equation results in

$$\zeta = 1 - \frac{M_\kappa \Delta t^2}{\rho_{(k)}} \pm \sqrt{\left(1 - \frac{M_\kappa \Delta t^2}{\rho_{(k)}}\right)^2 - 1}. \quad (7.22)$$

Enforcing the condition $|\zeta| \leq 1$ yields

$$\Delta t < \sqrt{2\rho_{(k)}/M_\kappa}, \text{ for all } \kappa \text{ values}. \quad (7.23)$$

In order for this condition to be valid for all κ values implies

$$M_\kappa \leq \sum_j \left(2ad\,\delta \frac{\left(d\delta \sum_\ell \left(\frac{1}{\left|\boldsymbol{\xi}_{(\ell)(k)}\right|} + \frac{1}{\left|\boldsymbol{\xi}_{(\ell)(j)}\right|}\right) V_{(\ell)}\right)}{\left|\boldsymbol{\xi}_{(k)(j)}\right|} + \frac{4b\delta}{\left|\boldsymbol{\xi}_{(k)(j)}\right|} \right) \left(v_{c(j)} V_{(j)}\right). \quad (7.24)$$

By using Eqs. 7.23 and 7.24, the stability criterion on the time step size can be expressed as

$$\Delta t < \sqrt{\frac{2\rho_{(k)}}{\sum_j \left(2ad\,\delta \frac{\left(d\delta \sum_\ell \left(\frac{1}{\left|\boldsymbol{\xi}_{(\ell)(k)}\right|} + \frac{1}{\left|\boldsymbol{\xi}_{(\ell)(j)}\right|}\right) V_{(\ell)}\right)}{\left|\boldsymbol{\xi}_{(k)(j)}\right|} + \frac{4b\delta}{\left|\boldsymbol{\xi}_{(k)(j)}\right|} \right) \left(v_{c(j)} V_{(j)}\right)}}. \quad (7.25)$$

The use of a safety factor that has a value of less than 1 is recommended as it makes the analysis more stable in case of some type of nonlinearity in the structure.

It is also worth noting that the stable time step size is dependent on the horizon size rather than the grid size because of the dependency of the PD material parameters on the horizon (Silling and Askari 2005).

7.5 Adaptive Dynamic Relaxation

Although the equation of motion of the peridynamic theory is in dynamic form, it can still be applicable to solve quasi-static or static problems by using a dynamic relaxation technique. As explained by Kilic and Madenci (2010), the dynamic relaxation method is based on the fact that the static solution is the steady-state part of the transient response of the solution. By introducing an artificial damping to the system, the solution is guided to the steady-state solution as fast as possible. However, it is not always possible to determine the most effective damping coefficient. Therefore, the damping coefficient is determined at each time step by using the Adaptive Dynamic Relaxation (ADR) scheme introduced by Underwood (1983).

According to the ADR method, the PD equation of motion is written as a set of ordinary differential equations for all material points in the system by introducing new fictitious inertia and damping terms

$$\mathbf{D}\ddot{\mathbf{U}}(\mathbf{X},t)+c\mathbf{D}\dot{\mathbf{U}}(\mathbf{X},t)=\mathbf{F}(\mathbf{U},\mathbf{U}',\mathbf{X},\mathbf{X}'), \tag{7.26}$$

where $\mathbf{D}$ is the fictitious diagonal density matrix and c is the damping coefficient whose values are determined by Greschgorin's theorem (Underwood 1983) and Rayleigh's quotient, respectively. The vectors $\mathbf{X}$ and $\mathbf{U}$ contain the initial position and displacement of the collocation (material) points, respectively, and they can be expressed as

$$\mathbf{X}^T=\left\{\mathbf{x}_{(1)},\mathbf{x}_{(2)},\ldots,\mathbf{x}_{(M)}\right\} \tag{7.27a}$$

and

$$\mathbf{U}^T=\left\{\mathbf{u}\left(\mathbf{x}_{(1)},t\right),\mathbf{u}\left(\mathbf{x}_{(2)},t\right),\ldots,\mathbf{u}\left(\mathbf{x}_{(M)},t\right)\right\}, \tag{7.27b}$$

where M is the total number of material points in the structure. Finally, the vector $\mathbf{F}$ is composed of PD interaction and body forces and its i^{th} component can be expressed as

$$\mathbf{F}_{(i)}=\sum_{j=1}^{N}\left(\mathbf{t}_{(i)(j)}-\mathbf{t}_{(j)(i)}\right)\left(v_{cj}V_{(j)}\right)+\mathbf{b}_{(i)}. \tag{7.28}$$

By utilizing central-difference explicit integration, displacements and velocities for the next time step can be obtained as

$$\dot{\mathbf{U}}^{n+1/2} = \frac{\left((2 - c^n \Delta t)\dot{\mathbf{U}}^{n-1/2} + 2\Delta t \mathbf{D}^{-1}\mathbf{F}^n\right)}{(2 + c^n \Delta t)} \tag{7.29a}$$

and

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \Delta t\, \dot{\mathbf{U}}^{n+1/2}, \tag{7.29b}$$

where n indicates the n^{th} iteration. Although Eq. 7.29a cannot be used to start the iteration process due to an unknown velocity field at $t^{-1/2}$, it can be assumed that $\mathbf{U}^0 \neq 0$ and $\dot{\mathbf{U}} = 0$. Therefore, the integration can be started by

$$\dot{\mathbf{U}}^{1/2} = \frac{\Delta t\, \mathbf{D}^{-1}\mathbf{F}^0}{2}. \tag{7.30}$$

Note that the only physical term in this algorithm is the force vector, $\mathbf{F}$. The density matrix, $\mathbf{D}$, damping coefficient, c, and time step size , Δt, do not have to be physical quantities. Thus, their values can be chosen to obtain faster convergence.

In dynamic relaxation, a time step size of 1 ($\Delta t = 1$) is a convenient choice. The diagonal elements of the density matrix, $\mathbf{D}$, can be chosen based on Greschgorin's theorem and can be expressed as

$$\lambda_{ii} \geq \frac{1}{4}\Delta t^2 \sum_j \left|K_{ij}\right|, \tag{7.31}$$

in which K_{ij} is the stiffness matrix of the system under consideration. The inequality sign ensures stability of the central-difference explicit integration; the derivation of this stability condition is given by Underwood (1983). Although this approach achieves near-optimal values, these values are coordinate frame dependent because they depend on absolute values of the global stiffness matrix as stated in the context of the finite element method of Lovie and Metzger (1999). Therefore, an alternative way can be followed by choosing the values based on the minimum element dimension to make the frame invariant, as suggested by Sauve and Metzger (1997). This approach seems to reduce overshooting as compared to Greschgorin's theorem.

Hence, the present solutions of the PD equations also utilize a frame-invariant density matrix. The construction of the stiffness matrix requires determination of the derivative of PD interaction forces with respect to the relative displacement vector, $\boldsymbol{\eta}$. Since the PD interaction forces given in Eq. 7.3 are nonlinear functions of $\boldsymbol{\eta}$, it is not always possible to determine its derivative. However, elements of the stiffness matrix can be calculated by using a small displacement assumption as

$$\begin{aligned}\sum_j |K_{ij}| &= \sum_{j=1}^{N} \frac{\partial\left(\mathbf{t}_{(i)(j)} - \mathbf{t}_{(j)(i)}\right)}{\partial\left(\left|\mathbf{u}_{(j)} - \mathbf{u}_{(i)}\right|\right)} \cdot \mathbf{e} \\ &= \sum_{j=1}^{N} \frac{\left|\boldsymbol{\xi}_{(i)(j)} \cdot \mathbf{e}\right|}{\left|\boldsymbol{\xi}_{(i)(j)}\right|} \frac{4\delta}{\left|\boldsymbol{\xi}_{(i)(j)}\right|} \left(\frac{1}{2} \frac{ad^2\delta}{\left|\boldsymbol{\xi}_{(i)(j)}\right|} \left(v_{c(i)} V_{(i)} + v_{c(j)} V_{(j)}\right) + b \right),\end{aligned} \tag{7.32}$$

in which $\mathbf{e}$ is the unit vector along the x-, y-, or z-direction. Note that the summation given in Eq. 7.32 can be employed to determine the elements of the stiffness matrix and it is frame invariant.

As described by Underwood (1983), the damping coefficient can be determined by using the lowest frequency of the system. The lowest frequency can be obtained by utilizing Rayleigh's quotient, which is given as

$$\omega = \sqrt{\frac{\mathbf{U}^T \mathbf{K} \mathbf{U}}{\mathbf{U}^T \mathbf{D} \mathbf{U}}}. \tag{7.33}$$

However, the elements of the density matrix given in Eq. 7.31 may have large numerical values, which make the denominator in Eq. 7.33 numerically difficult to compute. In order to overcome this problem, Eq. 7.26 can be written in a different form at the n^{th} iteration:

$$\ddot{\mathbf{U}}^n(\mathbf{X}, t^n) + c^n \dot{\mathbf{U}}^n(\mathbf{X}, t^n) = \mathbf{D}^{-1} \mathbf{F}^n\left(\mathbf{U}^n, \mathbf{U}'^n, \mathbf{X}, \mathbf{X}'\right). \tag{7.34}$$

The damping coefficient in Eq. 7.34 can be expressed by using Eq. 7.33 as

$$c^n = 2\sqrt{\left((\mathbf{U}^n)^T\, {}^1\mathbf{K}^n \mathbf{U}^n\right) \Big/ \left((\mathbf{U}^n)^T\, \mathbf{U}^n\right)}, \tag{7.35}$$

in which ${}^1\mathbf{K}^n$ is the diagonal "local" stiffness matrix, which is given as

$$ {}^1K_{ii}^n = -\left(F_i^n/\lambda_{ii} - F_i^{n-1}/\lambda_{ii}\right) \Big/ \left(\Delta t\, \dot{u}_i^{n-1/2}\right). \tag{7.36} $$

7.6 Numerical Convergence

The spacing between material points (grid size), Δ, and the horizon size, δ, influence the computational process. It is important to determine the optimum values of these parameters in order to achieve high accuracy with sufficiently small amount of computational time.

As explained in Silling and Askari (2005), the horizon size can be chosen based on the characteristic length dimensions. If dimensions are on the order of the nanoscale, then the horizon may represent the maximum distance of physical interactions between atoms or molecules. Therefore, it is important to specify its actual value for an accurate outcome of the analysis. For macroscale analysis, the horizon does not have a physical correspondence and its value can be chosen based on convenience. To determine the most optimum value of the horizon, a benchmark study of a one-dimensional bar with length L subjected to an initial strain loading of $\partial u_x/\partial x = 0.001\, H(\Delta t - t)$ is considered. The spatial integration is performed by using a very fine grid, so the numerical error due to grid size is minimum. Six different horizon sizes are considered, $\delta = (1, 3, 5, 10, 25, 50)\,\Delta$. For each of these cases, the displacement versus time variation of a collocation point, which is located close to the center of the bar, is monitored and compared against the analytical solution given by Rao (2004). As demonstrated in Fig. 7.6a–f, the highest accuracy is achieved for the horizon sizes of $\delta = \Delta$ and $3\,\Delta$. The discrepancy between analytical and numerical solutions becomes larger when the horizon size increases due the excessive wave dispersion (Silling and Askari 2005).

Furthermore, the computational time increases substantially as the horizon size increases. It is recommended to choose a horizon size of $\delta = 3\,\Delta$ since $\delta = \Delta$ may cause grid dependence on crack propagation and not be able to capture crack branching behavior, as demonstrated in Fig. 7.7 for a square plate with a central crack subjected to a velocity boundary condition of $V_0 = 50\,\mathrm{m/s}$. The model with a horizon size of $\delta = 3\,\Delta$ captures the expected crack branching behavior due to a very high velocity boundary condition, whereas the model with a horizon size of $\delta = \Delta$ can only capture the self-similar crack growth.

As mentioned in Sect. 7.1, the discretization error is on the order of $O(\Delta^2)$. Therefore, it is important to use a sufficient number of grid points to reduce the numerical error and at the same time achieve the desired numerical efficiency. By considering the vibration of a bar, it is possible to visualize the effect of grid size on

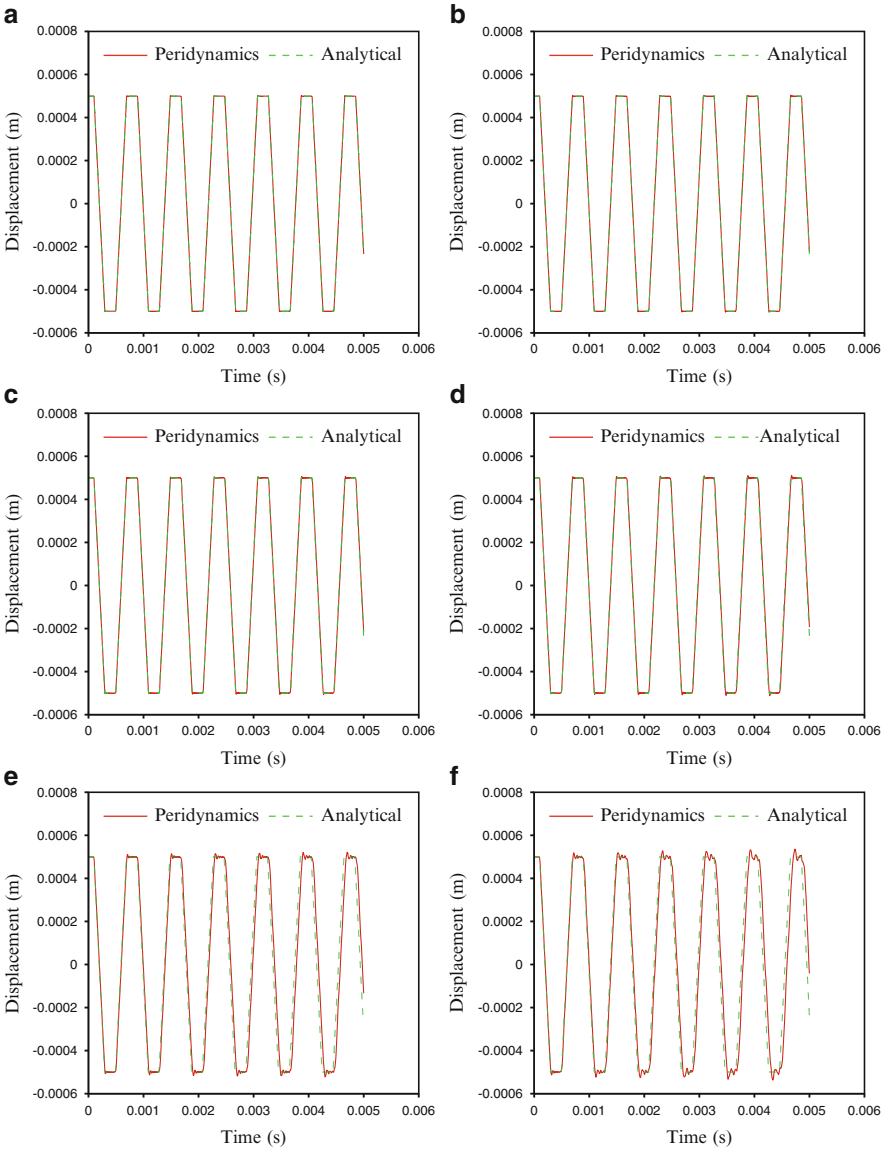


Fig. 7.6 Variation of displacement with time at the center of the bar for horizon size values of (**a**) $\delta = \Delta$, (**b**) $\delta = 3\,\Delta$, (**c**) $\delta = 5\,\Delta$, (**d**) $\delta = 10\,\Delta$, (**e**) $\delta = 25\,\Delta$, and (**f**) $\delta = 50\,\Delta$

the accuracy for four different grid size values, $\Delta = L/10$, $L/100$, $L/1000$, and $L/10000$, as shown in Fig. 7.8a–d. The horizon size is specified as $\delta = 3\Delta$. Sufficient accuracy is obtained at a grid size value of $\Delta = L/1000$. Note that the error in the very coarse grid size case of $\Delta = L/10$ increases as the time progresses.

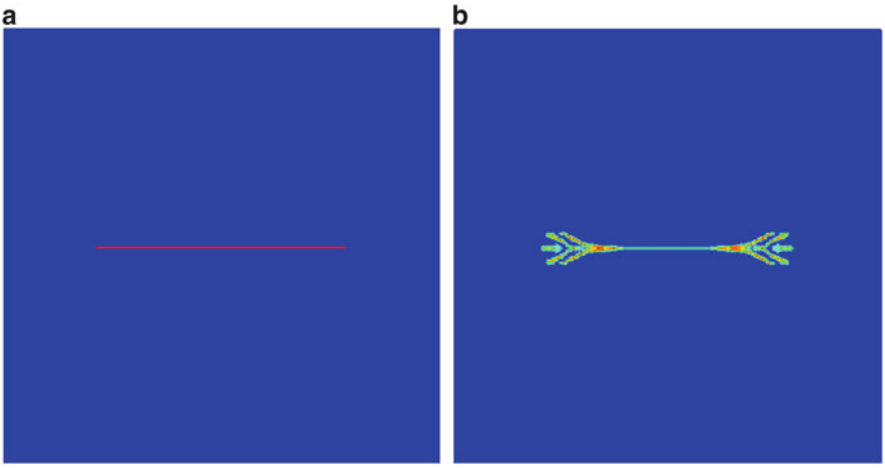


Fig. 7.7 Damage distribution in a square plate with a central crack subjected to a velocity boundary condition of $V_0 = 50$ m/s for horizon values of (**a**) $\delta = \Delta$ and (**b**) $\delta = 3\,\Delta$

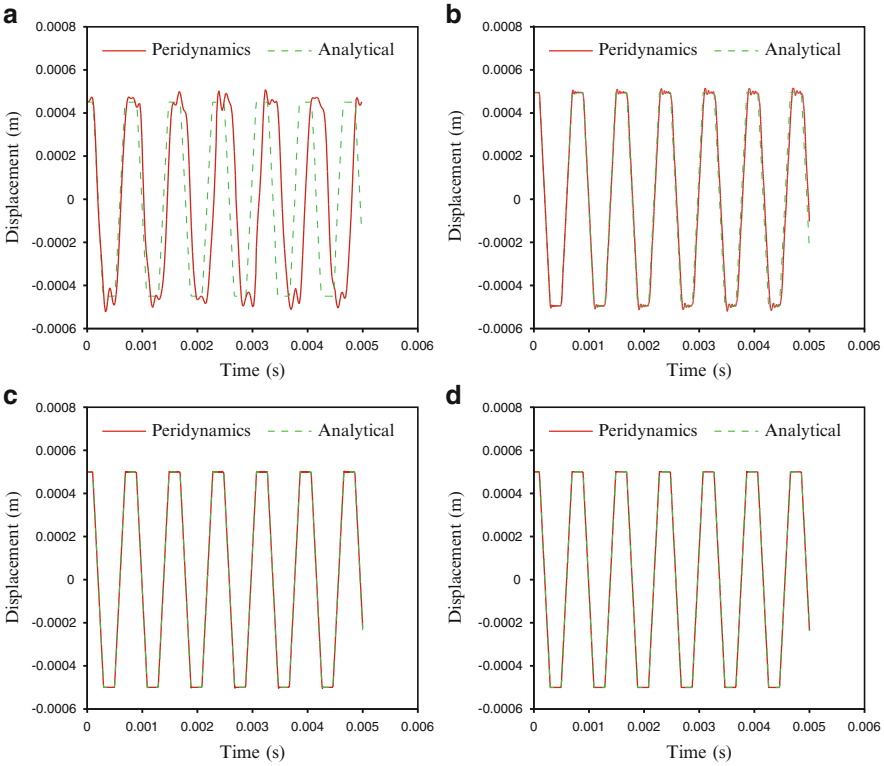


Fig. 7.8 Variation of displacement with time at the center of the bar for grid size values of (**a**) $\Delta = L/10$, (**b**) $\Delta = L/100$, (**c**) $\Delta = L/1000$, and (**d**) $\Delta = L/10000$

7.7 Surface Effects

The lack of interactions due to free surfaces may cause inaccuracies, especially for the material points close to the surfaces. This problem can be largely overcome by introducing surface correction factors. Detailed information about the surface correction factors and their determination procedure are given in Chap. 4. The surface corrections can be directly invoked in the equation of motion, Eq. 7.3, by rewriting it in a slightly different form as

$$\rho_{(k)}\ddot{\mathbf{u}}^n_{(k)} = \sum_{j=1}^{N}\left(\bar{\mathbf{t}}^n_{(k)(j)} - \bar{\mathbf{t}}^n_{(j)(k)}\right)\left(v_{c(j)}V_{(j)}\right) + \mathbf{b}^n_{(k)}, \tag{7.37}$$

where the corrected PD interaction forces can be expressed as

$$\bar{\mathbf{t}}^n_{(k)(j)} = \frac{\boldsymbol{\xi}_{(k)(j)} + \boldsymbol{\eta}^n_{(k)(j)}}{\left|\boldsymbol{\xi}_{(k)(j)} + \boldsymbol{\eta}^n_{(k)(j)}\right|} \times \left(2ad\,\delta G_{(d)(k)(j)}\frac{\Lambda^n_{(k)(j)}}{\left|\boldsymbol{\xi}_{(k)(j)}\right|}\bar{\theta}^n_{(k)} + 2b\delta G_{(b)(k)(j)}s_{(k)(j)}\right) \tag{7.38a}$$

and

$$\bar{\mathbf{t}}^n_{(j)(k)} = -\frac{\boldsymbol{\xi}_{(k)(j)} + \boldsymbol{\eta}^n_{(k)(j)}}{\left|\boldsymbol{\xi}_{(k)(j)} + \boldsymbol{\eta}^n_{(k)(j)}\right|} \times \left(2ad\,\delta G_{(d)(k)(j)}\frac{\Lambda^n_{(k)(j)}}{\left|\boldsymbol{\xi}_{(k)(j)}\right|}\bar{\theta}^n_{(j)} + 2b\delta G_{(b)(k)(j)}s_{(k)(j)}\right). \tag{7.38b}$$

Similarly, the corrected dilatation terms in Eqs. 7.38a, 7.38b are defined as

$$\bar{\theta}^n_{(k)} = d\delta\sum_{\ell=1}^{N} G_{(d)(k)(\ell)}s^n_{(k)(\ell)}\Lambda^n_{(k)(\ell)}\left(v_{c(\ell)}V_{(\ell)}\right) \tag{7.39a}$$

and

$$\bar{\theta}^n_{(j)} = d\delta\sum_{\ell=1}^{N} G_{(d)(j)(\ell)}s^n_{(j)(\ell)}\Lambda^n_{(j)(\ell)}\left(v_{c(\ell)}V_{(\ell)}\right). \tag{7.39b}$$

Note that surface correction factors are consistently required in the time integration process. Therefore, they should be computed prior to the start of time integration. Since the determination of the correction factors requires a test loading condition on the actual structure, it is important to initialize the displacement and velocity values of collocation (material) points before starting the time integration process.

7.8 Application of Initial and Boundary Conditions

The PD equation of motion yields the acceleration of the collocation points. The displacement and velocity of collocation points can be obtained by integrating the acceleration; it requires the initial condition values of these quantities. Therefore, all of the collocation points should be subjected to initial displacement and velocity conditions. Various ways of specifying the initial conditions are explained in detail in Chap. 2. The initial conditions can be specified either in the form of displacement and velocity values on all material points as given in Eqs. 2.23a, b or in terms of displacement and velocity gradients as given in Eqs. 2.25a, b.

As also explained in Chap. 2, the displacement and velocity constraints can be applied in the peridynamic theory by following a different approach than in classical continuum mechanics. The constraint conditions can be imposed to material points inside a fictitious boundary region, $\mathcal{R}_c$, as demonstrated in Fig. 7.9, with a width equivalent to the horizon size, δ. Displacement and velocity constraints can be applied by using Eqs. 2.26 and 2.28, respectively. On the other hand, the external loads can be applied as body loads through a material layer of $\mathcal{R}_\ell$, with a width of Δ, as shown in Fig. 7.9. The magnitude of body force applied to collocation points inside this region can be obtained by using Eqs. 2.34a, b, depending on the nature of the applied loading condition, i.e., distributed pressure or point force.

7.9 Pre-existing Crack and No-Fail Zone

In many practical applications, cracks may initially exist in the structure and be located at various sites of the structure. The PD approach to create these initial cracks is rather straightforward. Any interaction between two material points

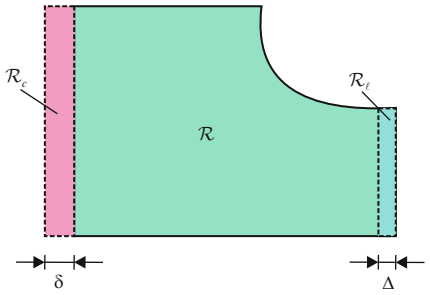


Fig. 7.9 Boundary regions for (**a**) displacement and velocity constraints and (**b**) external loads

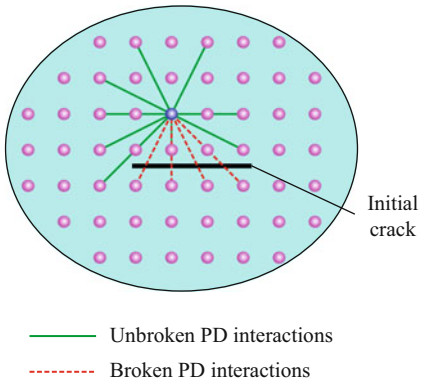


Fig. 7.10 Termination of PD interactions that pass through a crack surface

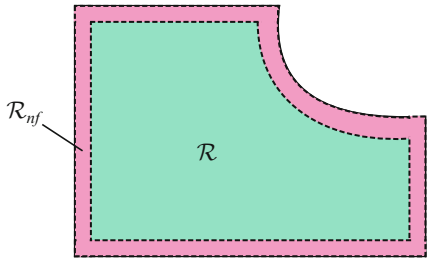


Fig. 7.11 No-fail zones

passing through a crack surface is terminated permanently, as shown in Fig. 7.10. Therefore, an entire set of terminated interactions represents a crack surface. If multiple cracks exist in the structure, the same procedure can be repeated for each crack surface.

For some applications under extreme loading conditions, unexpected failure may occur between collocation points located close to the external boundaries. In such cases, a region with a suitable width can be chosen as a "no fail zone," $\mathcal{R}_{nf}$, as shown in Fig. 7.11. The interactions associated with the collocation points located in this region are not allowed to fail. The thickness of the "no fail zone" should be chosen in such a way that it will have no adverse effect on the overall fracture behavior of the structure.

7.10 Local Damage for Crack Growth

The measure of local damage is dependent on the relationship between the horizon and the material point spacing. For computational efficiency, a horizon is commonly defined by three times the material point spacing, Δ, i.e., $\delta = 3\Delta$.

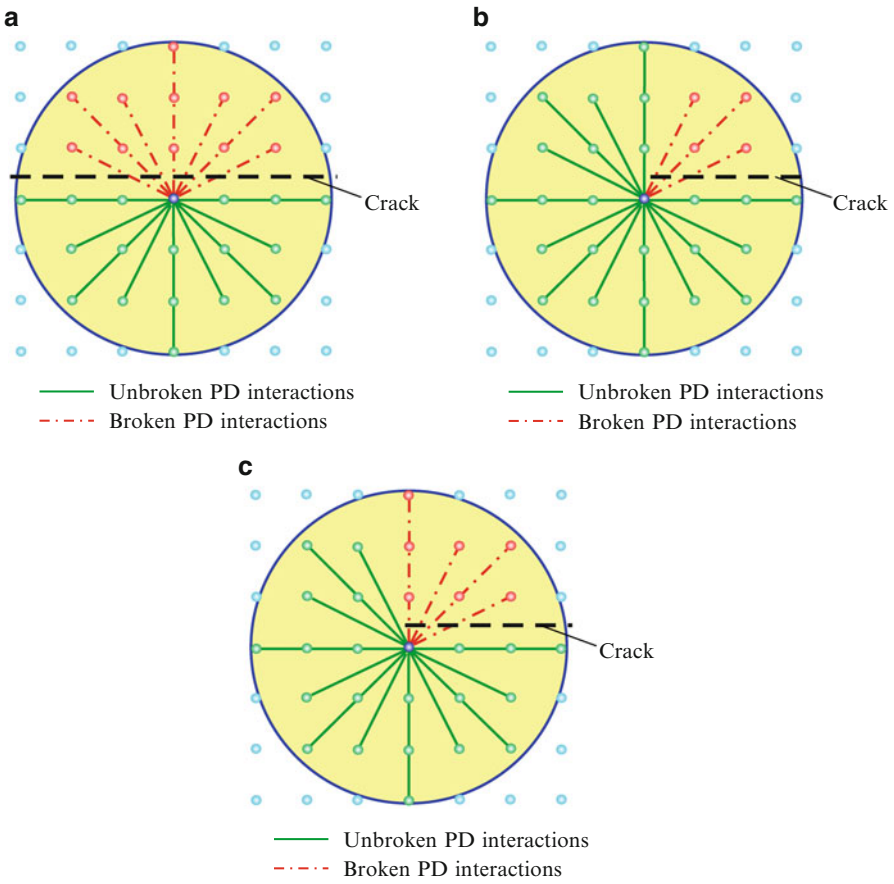


Fig. 7.12 Local damage at a material point (**a**) on the crack plane, (**b**) in front of the crack tip, and (**c**) behind the crack tip

As shown in Fig. 7.12a, the elimination of interaction between the material point $\mathbf{x}_{(j^-)}$ and the others, $\mathbf{x}_{(k^+)}$, above the dashed line representing the crack surface results in a local damage of $\varphi \approx 0.38$ for the material point $\mathbf{x}_{(j^-)}$. Although computationally not feasible, the local damage at material point $\mathbf{x}_{(j^-)}$ approaches one-half as the horizon approaches infinity. If the material point $\mathbf{x}_{(j^-)}$ is located immediately ahead of the dashed line representing the crack surface, as shown in Fig. 7.12b, its interactions are still intact with the material points $\mathbf{x}_{(k^+)}$ above the dashed line and directly aligned with $\mathbf{x}_{(j^-)}$. Thus, the local damage at $\mathbf{x}_{(j^-)}$ is calculated as $\varphi \approx 0.14$. If the material point $\mathbf{x}_{(j^-)}$ is located immediately behind the dashed line representing the crack surface, as shown in Fig. 7.12c, its interactions are no longer intact with the material points $\mathbf{x}_{(k^+)}$ above the dashed line and directly aligned with $\mathbf{x}_{(j^-)}$. Thus, the local damage at $\mathbf{x}_{(j^-)}$ is calculated as $\varphi \approx 0.24$.

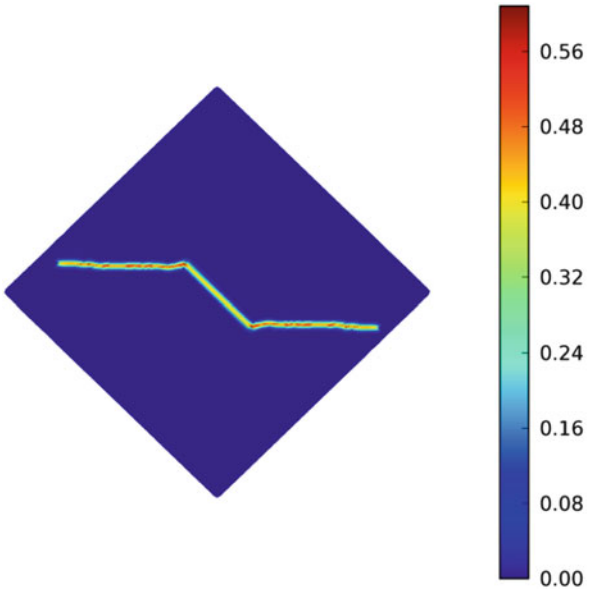


Fig. 7.13 Local damage measure indicating crack path in a PD model

According to these local damage values, a crack path can be established in PD calculations with a horizon size of $\delta = 3\Delta$. However, the local damage does not provide any information to determine the specific broken interactions. Therefore, the local damage values at neighboring points should also be considered in the determination of a crack path.

As the material point is located farther away from a crack surface, its degree of local damage decreases. For example, the local damage values for a material point $\mathbf{x}_{(j^-)}$ located at a distance of 0.5Δ, 1.5Δ, and 2.5Δ from the dashed line (crack surface) are calculated as $\varphi \approx 0.38$, $\varphi \approx 0.16$, and $\varphi \approx 0.02$, respectively. A crack surface results in discernible local damage values only for material points within a distance of 2Δ away from the crack. Therefore, the local damage values can be used to identify crack path and tip with an error of less than 2Δ. Figure 7.13 shows the local damage in a plate with a crack in a peridynamic model. Both the path of the crack and the tip are clearly visible.

7.11 Spatial Partitioning

In the PD theory, the number of interactions is limited by defining a region called the horizon. The horizon makes the computations tractable; otherwise, the number of interactions that needs to be taken into account at each time step is N^2 for N material points inside the body. This is especially very time consuming if the number of material points is large. According to the continuity assumption of the body, a material point must have the same neighbors during the deformation

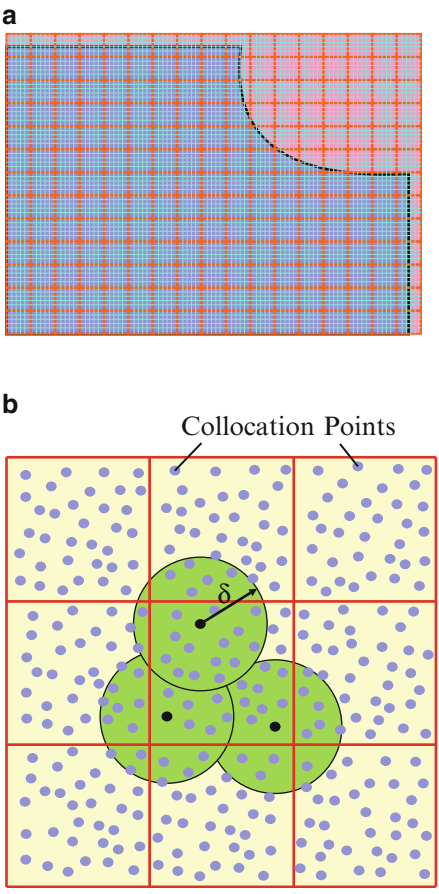


Fig. 7.14 Uniform grid and interaction of collocation points

process. Therefore, it is sufficient to determine the family members of a material point within its horizon only once during the computation process.

While establishing the family members, it may be computationally advantageous to split the domain into equally sized cells, as shown in Fig. 7.14a. The size of the cells should be larger than the horizon size. During the search process of the family members, it is only necessary to examine the collocation points in the neighboring cells, as shown in Fig. 7.14b.

Another important issue is following an efficient process for storing the family members of the collocation points in order to overcome possible memory limitations. For this purpose, two different arrays can be utilized. The first array (see Array #1 in Fig. 7.15) can store all family members of material points sequentially in a single column. The second array (see Array #2 in Fig. 7.15) can be utilized as an indicator for the first array, so that the family members of a particular material point can be easily extracted from the first array. Each element of Array #2 corresponds to the location of the first material point within the family of a

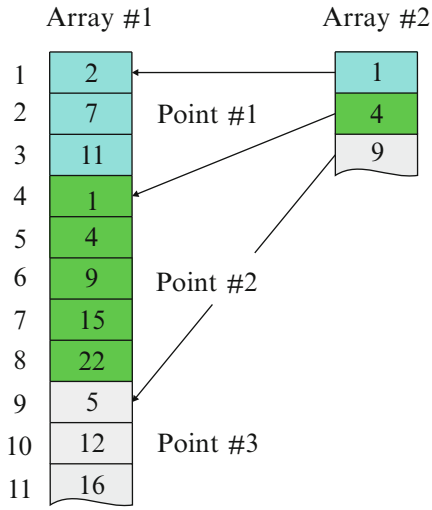


Fig. 7.15 Arrays used to store information on family members

particular point in Array #1. For instance, as shown in Fig. 7.15, the second element of Array #2 (i.e., 4) associated with material point #2 indicates the fourth element of Array #1 as the first material point number within the family of material point #2.

7.12 Utilization of Parallel Computing and Load Balancing

The structure of the PD meshless scheme is very suitable for parallel computing. Therefore, significant time efficiency can be achieved, depending on the number of processors to be utilized. There are various tools available for parallel computing, such as central processing units (CPU) and graphics processing units (GPU). The most important aspect of parallel programming is the load balancing, so that full advantage of the parallel programming can be realized. Efficient load balancing can be obtained by distributing an approximately equal number of collocation points to each processor. Otherwise if a processor finishes its job earlier than others at the end of the time step, then it has to wait for other processors to finish their jobs to proceed to the next time step. The other important issue is to keep the number of PD interactions between collocation points that are assigned to different processors at a minimum level, since the computation of these interactions is carried out by a single processor to avoid a race condition. A race condition occurs when multiple processors try to access the same shared memory.

The computational domain can be divided into subunits and each of these subunits can be assigned to a specific processor by using binary space decomposition, as shown in Fig. 7.16 (Berger and Bokhari 1987). This method can handle variations in the collocation point concentration at different regions of the domain. The decomposition process continues in multiple steps, where each subunit is divided into two new rectangular subunits in every step. Based on the number of

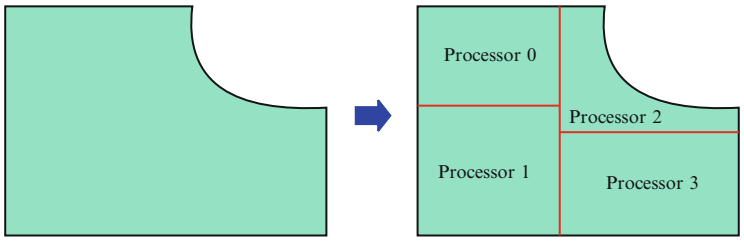
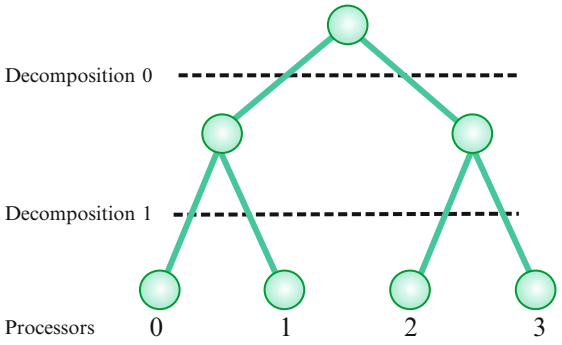


Fig. 7.16 Processor distribution

Fig. 7.17 Tree structure to construct decomposition



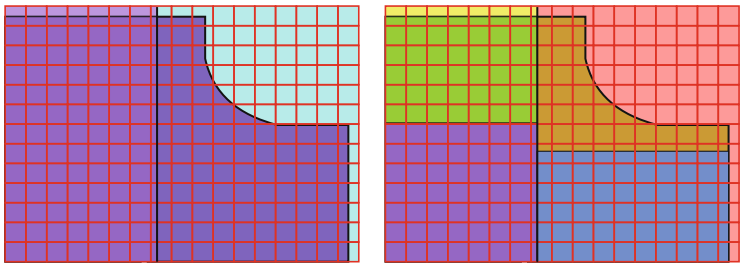


Fig. 7.18 Binary space decomposition

collocation points assigned to each subunit, a workload estimate can be calculated. For instance, if there are p number of available processors, where p is not necessarily an even number, the domain is split into two, each having s_1 and s_2 collocation points. Partitioning is performed such that the ratio of s_1 and s_2 is equal or close to $(p/2)/(p - p/2)$. The partition direction is chosen to be the longest side of the domain in order to reduce the number of interactions between subunits. Then, $p/2$ processors are assigned to the subunit having s_1 collocation points and $p - p/2$ processors are assigned to subunit with s_2 collocation points. Each subunit is then divided into other subunits as long as the assigned number of processors is greater than one. Figures 7.17 and 7.18 demonstrate a tree structure of a two-step binary decomposition by using four processors and the subunits that are assigned to these four processors, respectively.

References

Berger MJ, Bokhari SH (1987) A partition strategy for nonuniform problems on multiprocessors. IEEE Trans Comp C-36:570–580

Kilic B (2008) Peridynamic theory for progressive failure prediction in homogeneous and heterogeneous materials. Dissertation, University Arizona

Kilic B, Madenci E (2010) An adaptive dynamic relaxation method for quasi-static simulations using the peridynamic theory. Theor Appl Fract Mech 53:194–201

Lovie TG, Metzger DR (1999) Lumped mass tensor formulation for dynamic relaxation. In: Hulbert GM (ed) Computer technology, ASME PVP vol 385, Boston, pp 255–260

Rao SS (2004) Mechanical vibrations, 4th edn. Pearson Prentice Hall, Upper Saddle River

Sauve RG, Metzger DR (1997) Advances in dynamic relaxation techniques for nonlinear finite element analysis. J Pres Ves Tech 117:170–176

Silling SA (2004) EMU user's manual, Code Ver. 2.6d. Sandia National Laboratories, Albuquerque

Silling SA, Askari E (2005) A meshfree method based on the peridynamic model of solid mechanics. Comput Struct 83:1526–1535

Underwood P (1983) Dynamic relaxation. Comput Meth Trans Anal 1:245–265