Chapter 1

Introduction

Contents

Simulation-based Engineering Science (SBES) [2] is the third pillar of the modern science and engineering, a peer alongside theory and physical experiment [1]. Compared with physical experiment, SBES has the advantages of low cost, safety, and efficiency in solving various kinds of challenging problems. To better simulate those extreme events such as hypervelocity impact, penetration, blast, crack propagation, and multi-phase (solid–liquid–gas) interactions involving failure evolution, yet effectively discretize localized large deformation, the transition among different types of failure modes and fragmentation remains a very difficult task. Based on the way how deformation and motion are described, existing spatial discretization methods can be classified into Lagrangian, Eulerian, and hybrid ones, respectively.

1.1 LAGRANGIAN METHODS

In Lagrangian methods the computational grid is embedded and deformed with the material. Since there is no advection between the grid and material, no advection term appears in the governing equations, which significantly simplifies the solution process. The mass of each material element keeps constant during the solution process, but the element volume varies due to element deformation. Lagrangian methods have the following advantages:

1. They are conceptually more simple and efficient than Eulerian methods. Because there is no advection term that describes the mass flow across element boundaries, the conservation equations for mass, momentum, and energy are simple in form, and can be efficiently solved.

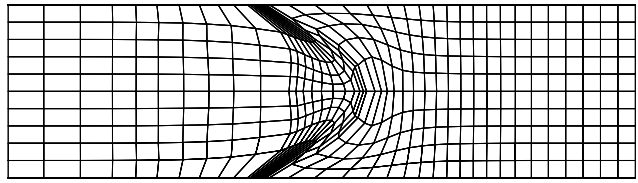

FIGURE 1.1 Lagrangian grid.

2. Element boundaries coincide with the material interfaces during the solution process so that it is easy to impose boundary conditions and to track material interfaces.
3. Since Lagrangian methods track the flow of individual masses, it is easy to implement history-dependent constitutive models.

Fig. 1.1 shows a typical Lagrangian grid which is embedded and deformed with the material. Severe element distortion results in significant errors in numerical solution, and even leads to a negative element volume or area which would cause abnormal termination of the computation. To obtain a stable solution with an explicit time integration scheme, the time step must be smaller than a critical time step which is controlled by the minimum characteristic length of all elements in the grid. Because severe element distortion would significantly decrease the characteristic element length, the time step in a Lagrangian calculation could become smaller and smaller, and finally approach zero, which makes the computation impossible to be completed. To complete a Lagrangian computation for an extreme loading case, a distorted grid must be remeshed and its result must be interpolated to the remeshed grid. The remesh or rezone technique has been successfully used in solving many 1D and 2D problems, but rezoning a complicated 3D material domain is still a challenging task. For a history-dependent material, the history variables are also required to be interpolated from the old grid to the new grid, which may further cause numerical error in stress calculation.

Another way to eliminate the element distortion is to use the erosion technique, which simply deletes the heavily distorted elements. An element is considered to be heavily distorted if its equivalent plastic strain exceeds a user-defined erosion strain value, or the critical time step size is less than a prescribed value. Introducing element erosion can resolve some of the issues related to the severe element distortion, but also introduce new issues. The global system will lose both mass and energy, which can severely affect the simulation outcome. Furthermore, the erosion technique cannot model the formation process of debris cloud and its interaction with other panels in hypervelocity impact simulation.

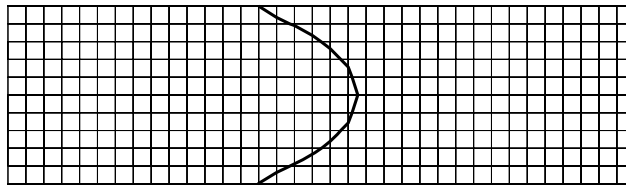

FIGURE 1.2 Eulerian grid.

Many Lagrangian codes have been developed, as shown in the open literature. The HEMP [8] was developed in the early 1960s by Wilkins at the Lawrence Livermore National Laboratory. The HEMP was an explicit Lagrangian finite-difference code that could handle large strains, elastic–plastic flow, wave propagation, and sliding interfaces. The EPIC code [9] was an explicit Lagrangian finite element code developed in the 1970s by Johnson. Both the rezoning and erosion techniques were employed in the EPIC to simulate high velocity impact and blast problems. The PRONTO3D code [10] was a 3D transient solid dynamics code developed at the Sandia National Laboratory for analyzing large deformations of highly nonlinear materials subjected to extremely high strain rates. This code was based on an explicit finite element formulation, and had been coupled with the smoothed particle hydrodynamics (SPH) method through a contact-like algorithm [11]. The DYNA2D and DYNA3D codes were developed in the 1970s at the Lawrence Livermore National Laboratory as explicit Lagrangian finite element codes and were successfully commercialized [12–14].

1.2 EULERIAN METHODS

For problems in which a material domain could become heavily distorted or different materials are mixed, an Eulerian method is more appropriate. In Eulerian methods, the computational grid is fixed in space and does not move with the material such that the material flows through the grid, as shown in Fig. 1.2.

There is no element distortion in Eulerian methods, but the physical variables, such as mass, momentum, and energy, advect between adjacent elements across their interface. The volume of each element keeps constant during the simulation, but its density varies due to the advection of mass. Eulerian methods are suited for modeling large deformations of materials so that most of computational fluid dynamics codes and early hydrocodes for impact and blast simulation employ Eulerian methods.

Eulerian methods only calculate the material quantities advected between elements without explicitly and accurately determining the position of material

interface and free surface so that they are quite awkward in following deforming material interfaces and moving boundaries. Significant efforts have been made to develop interface reconstruction methods.

HELP (Hydrodynamic plus ELastic Plastic) [15], developed by Walsh and Hageman in the 1960s, is a multi-material Eulerian finite difference program for compressible fluid and elastic–plastic flows. To treat the material interface or free surface, massless tracer particles are used, which define the surface position and move across the Eulerian grid. CTH [16] is an Eulerian finite volume code developed at Sandia National Laboratories to model multi-dimensional, multi-material, large deformation, and strong shock wave physics. The CTH code employs a two-step Eulerian solution scheme, a Lagrangian step in which the cells distort to follow the material motion, and a remesh step where the distorted cells are mapped back to the Eulerian mesh. Material interfaces are reconstructed using the Sandia Modified Young's Reconstruction Algorithm. The CTH has adaptive mesh refinement and uses second-order accurate numerical methods to reduce numerical dispersion and dissipation. It is still under development at Sandia National Laboratories [17].

The Zapotec developed at Sandia National Laboratories is a framework that tightly couples the CTH and PRONTO codes [18,19]. In a Zapotec analysis, both CTH and PRONTO are run concurrently. For a given time step, the Zapotec maps the current configuration of a Lagrangian body onto the fixed Eulerian mesh. Any overlapping Lagrangian material is inserted into the Eulerian mesh with the updated mesh data passed back to the CTH. After that the external loading on the Lagrangian material surfaces is determined from the stress state in the Eulerian mesh. These loads are passed back to PRONTO as a set of external nodal forces. After the coupled treatment is completed, both CTH and PRONTO are run independently over the next time step.

1.3 HYBRID METHODS

Both purely Lagrangian and purely Eulerian methods possess different shortcomings and advantages so that it is desirable to find new approaches to take advantage of both methods to better tackle challenging problems. The arbitrary Lagrangian–Eulerian (ALE) method [20] and the particle-in-cell (PIC) method [21,22] are two representatives.

1.3.1 Arbitrary Eulerian–Lagrangian Method and Its Variations

The ALE method was first proposed in the finite difference and finite volume context [23,24], and was subsequently adopted in the finite element context

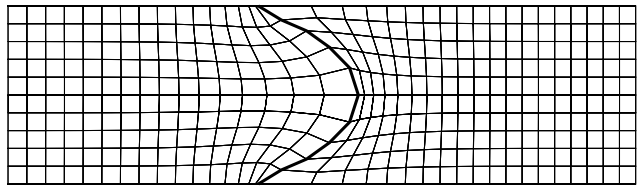

FIGURE 1.3 ALE grid.

[25–27]. The mixed Eulerian–Lagrangian method [24] involves the Eulerian set-up with respect to one dimension and the Lagrangian one to the other dimension which corresponds to the direction of fluid flow. The coupled Eulerian–Lagrangian [23] code employs an Eulerian mesh for the entire region and Lagrangian meshes for the subregions of fluids with nonstationary boundaries approximated by Lagrangian lines.

In the above methods, the computational mesh may be moved with the material in Lagrangian manner, or be held fixed in Eulerian manner, or be moved independently of material deformation to optimize element shapes and to describe the boundaries accurately [20], as shown in Fig. 1.3. Because these methods offer great flexibility in moving the computational mesh, they can handle a much greater distortion of the material than a Lagrangian method, with a higher resolution than that afforded by an Eulerian method. However, the convective terms still pose some problems. Furthermore, designing an efficient and effective mesh-moving algorithm for complicated 3D problems remains a challenging task.

1.3.2 Particle-In-Cell Method and Its Variations

The PIC method was proposed and developed at Los Alamos National Laboratory by Harlow in the late 1950s [21,22,28]. PIC makes use of both Lagrangian and Eulerian descriptions, namely, the fluid is discretized as a set of Lagrangian particles that carry material position, mass, and species information, but the computational mesh is a uniform Eulerian one. A computational cycle is divided into two phases, a Lagrangian phase and an Eulerian (remap or rezone) phase. In the Lagrangian phase, all the variables, including the mesh coordinates and the particle positions, are advanced. In the Eulerian phase, the mesh is mapped back into its original configuration, leaving the particles at their new locations. This process can also be viewed in a time splitting way, namely, the Lagrangian phase updates the quantities by all the processes except for advection, while the Eulerian phase moves the particles and accomplishes all of the advective fluxing [29].

As a variation of the PIC method, the marker-and-cell (MAC) method was developed by Harlow and Welch [30,31] to treat incompressible and free surface flows. In the MAC method, particles are used as markers to define the location of the free-surface, and the Poisson equation for the pressure is solved to treat the fluid incompressibility. The MAC method was the first successful technique for simulating incompressible flows [32].

The original version of PIC is not a fully Lagrangian particle method because only the material position, mass, and species information is carried by the particles, while the remaining quantities are still stored in the computational grid. The transfer of information between the particles and the underlying grid leads to significant numerical diffusion. There are two strategies to reduce the numerical diffusion, namely, second-order accuracy advection scheme [33] and fully Lagrangian particle method. Brackbill et al. developed a fully Lagrangian particle method, FLuid-Implicit-Particle (FLIP) method [34,35], in which each particle carries all of the properties of the fluid, including momentum and energy. FLIP preserves the ability of the original PIC to resolve contact discontinuities, but eliminates the major source of numerical diffusion.

1.3.3 Material Point Method

When working on the penetration problems in the early 1990, Zhen Chen and his former PhD advisor, Buck Schreyer, faced a challenging task to improve the computational fidelity and efficiency of the finite element method (FEM), due to its limitation in the required use of a pin-hole in the mesh design. In a seminar at University of New Mexico, Deborah Sulsky presented the advances of the PIC method, based on her collaborative research on computational fluid dynamics with the scientists at Los Alamos National Laboratory. Since the particle motion in fluid is similar to the penetrator's motion in solid from the viewpoint of hard–soft body interaction, Sulsky's seminar opened Chen's eyes to a new direction of research so that he initiated an interdisciplinary discussion. In collaboration with Sandia National Laboratories, the team of three folks with diversified tastes then started to combine computational fluid dynamics with computational solid dynamics to develop a continuum-based particle method with its first journal paper published in 1994 [5], which was later named as the Material Point Method (MPM). Over the last two decades, many research teams in the world have further developed the MPM and combined the MPM with other numerical methods for multiphase, multiphysics, and multiscale simulations to advance SBES.

The MPM is an extension of the FLIP method from computational fluid dynamics to computational solid dynamics with two key differences. First, the constitutive equations are solved at the particles (material points) rather than the grid cell centers such that the MPM can readily model history-dependent materials. Second, the MPM is formulated in the weak form consistent with the FEM so that the FEM and MPM could be effectively combined together [36–41] for large-scale simulations.

The MPM is a fully Lagrangian particle method which utilizes the advantages of both Eulerian and Lagrangian methods. As compared with Eulerian methods, the numerical dissipation normally associated with a Eulerian method is eliminated, while the complete deformation history of material points are tracked. Compared with Lagrangian methods, mesh distortion and element entanglement are avoided. Therefore, the MPM has demonstrated obvious advantages in tackling those extreme events such as impact, blast, penetration, perforation, machining, fragmentation, and multi-phase interaction involving failure evolution, as demonstrated in Chapter 8.

1.4 MESHFREE METHODS

In addition to the evolution of the MPM, different types of meshfree and particle methods for improved spatial discretization in different problems have also been proposed and developed in the SBES community [42]. Since all these meshfree and particle methods do not use a rigid mesh connectivity compared with the conventional mesh-based methods such as the FEM, they have been applied to many challenging problems of current interests such as impact/contact, localization, crack propagation, penetration, perforation, and fragmentation. Nevertheless, many of the meshfree methods suffer from higher computational costs, and the accuracy of some meshfree methods is still dependent on the node regularities to some extent.

Smoothed particle hydrodynamics (SPH) [43–46] is one of the earliest meshfree Lagrangian particle methods. The SPH was first proposed by Lucy [43] and Gingold and Monaghan [44] in 1977 to solve astrophysical problems in the 3D open space, and has been extensively studied and extended to solid and fluid dynamics problems with large deformations. The SPH and its improved versions have been successfully applied to the hypervelocity impact simulations, and become some of the most popular meshfree methods in this area. Because of their good performance, several commercial softwares, such as AUTODYN, PAM-CRASH, and LS-DYNA, have incorporated the SPH into their solvers. However, the SPH is limited in simulating multiphase interactions involving failure evolution.

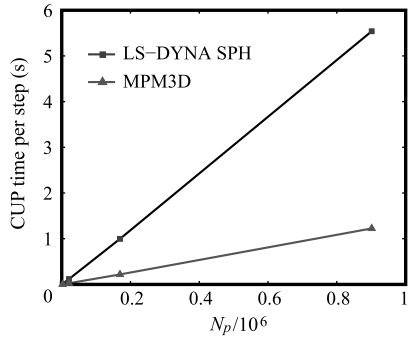

FIGURE 1.4 Efficiency comparison between the MPM and SPH [47].

Ma et al. [47] compared the basic formulation and features of the MPM with SPH from the following aspects: neighbor searching, approximation functions, consistency of shape functions, tensile instability, time integration, boundary conditions, and contact algorithm. A comparative study showed that the MPM possesses many prominent features. The formulation of the MPM is simple and similar to the traditional FEM. The time consuming neighbor searching, which is compulsory in most meshfree methods, is not required in the MPM. The MPM shape functions exactly satisfy the constant and linear consistency. The MPM avoids tensile instability that is annoying in the SPH. The boundary conditions can be applied in the MPM as easily as in the FEM, and the contact algorithm can be efficiently implemented whose cost is linear in the number of material points involved. Because the same regular computational grid can be used in all time steps, the time step keeps constant in the MPM simulations. Numerical studies have showed that the computational efficiency and stability of our MPM3D code are much higher than those of LS-DYNA SPH module.

Fig. 1.4 compares the CPU time per step as used by LS-DYNA SPH module and our MPM3D code in the simulation of the translation motion of a cubic block [48]. It demonstrates that the CPU time per step used by both methods increases linearly with the increase of number of particles, but the rate of increase of the SPH is much higher than that of the MPM.

Ma et al. [47] also investigated the accuracy and efficiency of the SPH and MPM by simulating the impact of a copper cylinder to a rigid wall with an impact velocity of 190 m/s. In the SPH simulations, the constant associated with the smoothing length was set to 1.2 (SPH1) and 1.4 (SPH2), respectively. The value of 1.2 is the default value used in LS-DYNA, and a larger value will increase the computational time but may improve the result with more neighbors for each particle. Fig. 1.5 compares the final configurations of the bar obtained

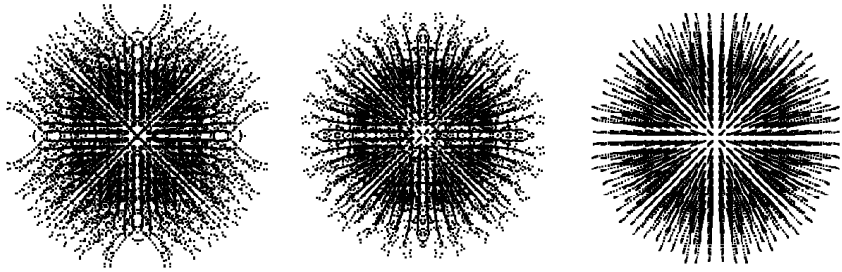

FIGURE 1.5 Final configurations of the Taylor bar impact (top view): (a) SPH1, (b) SPH2, and (c) MPM3D [47].

by the SPH and MPM, which shows that the SPH algorithm suffers from numerical fracture due to tensile instability. Enlarging the smoothing length can alleviate the numerical fracture, but particle clumps may still exist. Furthermore, enlarging the smoothing length increases the time step size, which raises the one-step computational cost significantly.