WSPC/Book Trim Size for 9 in x 6 in ws-book9x6

WSPC/Book Trim Size for 9 in x 6 in ws-book9x6

WSPC/Book Trim Size for 9 in x 6 in ws-book9x6

WSPC/Book Trim Size for 9 in x 6 in ws-book9x6

ws-book9x6





Carol Griswold Hoover

Carol has been a faithful source of inspiration and a tireless source of help and good cheer throughout the preparation of this book. Chapter 4 is mainly hers. We two join in dedicating this book to the memories of Robert William Griswold, Edgar Malone Hoover (Junior), and Mary Frances Wolfe Hoover. vi

Dedication and Motivation

Three Thoughts, for students of all ages :

In an emergency it is always a good thing to pause and reflect.

L. Frank Baum's Scarecrow

Happy is he who gets to know the reasons for things.

Virgil

Simplify, simplify, simplify

Thoreau

ws-book9x6

Preface

SPAM, "Smooth Particle Applied Mechanics", is a simple and transparent computational approach to simulating macroscopic nonequilibrium flows. Simple in concept and structure, but broad in scope, it is nicely suited to teaching, to research, and to real applications. SPAM blends the ordinary differential equations of particle mechanics with the partial differential equations and conservation laws of continuum mechanics. It is interesting that continuum mechanics, originally developed in order to avoid the details associated with particles, is itself most easily implemented by particle-based methods, of which SPAM is the simplest.

I expect this book about SPAM to be most useful to researchers who seek to understand and enjoy to learn by doing: writing their own software, dreaming up their own test problems, and challenging the current wisdom. My own experience suggests that careful intercomparisons of results with those of colleagues who write their own independent computer programs is the best way to find and correct errors. I have tried hard to state precisely what is done in the examples given in the book. I intend that others can reproduce these examples, not just qualitatively, but entirely and precisely.

This book is designed to be self-contained, and accessible to both students and researchers. Thoreau's admonition—"Simplify, simplify, simplify"—is specially appropriate to the description and study of smooth particles, for which so many alternative approaches and ideas have been promoted. I have used a "bare bones" notation in writing equations, omitting obvious subscripts and leaving it to the reader to distinguish scalars, vectors, and tensors from the context. Likewise in pursuit of simplicity, I have chosen to omit equation numbers and detailed references to the literature. The recent Liu-Liu book "Smooth Particle Hydrodynamics", published by World Scientific in 2003, can be consulted for a more comprehensive set of references. viii

Preface

Because SPAM has strong links to Newtonian particle mechanics and to statistical mechanics, as well as to the computational simulation of both particulate systems and continua, this book spans a variety of fields. I start out by discussing how fast computers helped SPAM simulations to develop from its roots in particle and continuum mechanics, made possible by computers. This is followed by an analysis of computational methods, along with an assessment of convergence and stability, emphasizing Lyapunov instability, the exponential growth of small perturbations. The details involved in graphics and parallel computing using SPAM bring the subject to the current state of the art. I have included several pedagogical example problems in the text. For clarity and simplicity, these problems treat onedimensional or two-dimensional systems. The reader should have only a little difficulty in extending these ideas to applications in three dimensions.

The work leading me to write this book began nearly fifteen years ago, at the Lawrence Livermore Laboratory in California, where I formulated and defended a modest proposal to study computer simulations of high-strainrate deformation. Tom Weaver, a member of the evaluation committee, asked me whether or not I had considered using "smooth-particle" methods. I had not. One thing led to another: reading the literature, most of it in the form of government sponsored laboratory reports; teaching the technique to students; traveling to visit a sampling of the many experts interested in practical applications of SPAM; collaborations, with Harald Posch and Oyeon Kum, on many applications; and, finally, the writing of this book with the help, inspiration, and support of my wife Carol. Chapter 4, "Computer Programming", in particular, reflects her work. The luxury of retirement, coupled with the generous support of Peter Raboin and Bob Ferencz (Methods Development Group), Rob Sharpe (Center for Computational Engineering), and Son Nguyen (Technical Information Department) in making the research facilities of the Department of Energy's Lawrence Livermore Laboratory available to me, made this book possible.

I thank Lakshmi Narayanan, Lu Jitan, and Anthony Doyle, all of World Scientific Publishing, for stimulating and encouraging the effort needed to write this book. I appreciate the kind suggestions of Paco Uribe (UNAM, Mexico) who read through the manuscript.

Ruby Valley and Great Basin College, Elko County, Nevada, Summer 2006.

v

Contents

Dedication and Motivation

Preface vii Physical Ideas Underlying SPAM 1 1. 1.11 Particles versus Continua 1.23 1.3Newton's Particle Mechanics 4 Eulerian and Lagrangian Continuum Mechanics 91.41.5Computer Simulation of Microscopic Particle Motion . . 141.6Liouville's Theorem; Statistical Mechanics 161.720SPAM [Smooth Particle Applied Mechanics] 241.81.9Example: A Molecular Dynamics Simulation 2631332.Continuum Mechanics 2.133 Summary and Scope of Continuum Mechanics 2.2Evolution Equations for Fluids and Solids 352.3Initial and Boundary Conditions 392.4Constitutive Equations for Equilibrium Fluids 422.5Constitutive Relations for Nonequilibrium Fluids 452.6462.7Constitutive Relations for Elastic Solids 48 2.8Constitutive Relation for Nonequilibrium Plasticity . . . 522.955

х

Contents

	2.10	Example: Heat Conduction in One Dimension	58
	2.11	Example: Sound Propagation in One Dimension	59
	2.12	Example: Rayleigh-Bénard Flow in Two Dimensions	60
	2.13	References	64
3.	Smoot	th Particle Methods	65
	3.1	Summary	65
	3.2	Motivation	66
	3.3	Basic Equations	67
	3.4	Interpolation on an Irregular Grid	68
	3.5	Alternative Averages: $[f_0, f_1, f_2, \dots]$	71
	3.6	Weight Functions	74
	3.7	Continuity Equation from $\nabla \cdot v$ with SPAM	80
	3.8	Evaluating the Spatial Derivatives $\{\nabla \rho, \nabla \cdot P, \nabla \cdot Q\}$.	82
	3.9	SPAM Equation of Motion and Energy Equation	83
	3.10	Rezoning; Does Particle Size Matter?	84
	3.11	Ideal-Gas Isomorphism with SPAM	85
	3.12	Evaluating the Spatial Derivatives $\{\nabla v, \nabla T\}$	87
	3.13	von Neumann-Richtmyer Artificial SPAM Viscosity	89
	3.14	Example: Adiabatic Atmospheric Equilibrium	91
	3.15	Example: Isothermal Atmospheric Equilibrium	94
	3.16	References	97
4.	Comp	uter Programming	99
	4.1	Summary	99
	4.2	FORmula TRANslation languages	100
	4.3	Designing a SPAM program	105
	4.4	Runge-Kutta Integration with Fortran and C	112
	4.5	A Useful Random Number Generator	117
	4.6	Graphic Displays and Analysis	119
	4.7	"Debugging" Tools—Finding Errors	125
	4.8	Parallel Computing	128
	4.9	Mesh Partitioning	131
	4.10	Message Passing Techniques	133
	4.11	Material Interfaces in Parallel Computing	136
		4.11.1 Concentric Annuli Undergoing Rotation	137
		4.11.2 Free Expansion Problem	138
		4.11.3 Crushing of an Elastic-Plastic Sheet	139

Contents			xi
	4.12	4.11.4 Caricature of a Billiard Table	$140\\142$
5.	Initial	and Boundary Conditions, Interpolation	143
	5.1 5.2 5.3 5.4 5.5 5.6 5.7 5.8 5.9	Summary.Initial Coordinates.Mesh Generation for SPAM with Free Boundaries.Implementing Periodic and Mirror Boundaries.Alternative Meshes—Regular Lattices.Elastic Stability of Embedded-Atom Lattices.Invariant Curvature Crystal Stabilization.Example: Heat Transfer in One Dimension with SPAM.Example: Periodic Shear Flow with SPAM.	$143 \\ 144 \\ 147 \\ 150 \\ 156 \\ 157 \\ 162 \\ 164 \\ 167$
	$5.10 \\ 5.11$	Example: Rayleigh-Bénard Flow with SPAM	$171 \\ 175$
6.	Conver 6.1 6.2 6.3 6.4 6.5 6.6 6.7 6.8 6.9 6.10 6.11 6.12 6.13 6.14	argence and Stability Summary	177 178 180 180 181 184 186 188 190 191 192 195 196 199
7.	Lucy a 7.1 7.2 7.3 7.4 7.5	and Embedded-Atom Fluids Summary	 201 201 202 203 205 207

xii

Contents

	7.6	Example: Embedded-Atom Gravitational Relaxation	208
	7.7	Example: Embedded-Atom Model of Falling Water	211
	7.8	Example: Free Expansion of a γ -law Gas $\ldots \ldots \ldots$	213
	7.9	Example: Lucy-Fluid Shockwave Structure	217
	7.10	References	225
8.	SPAM	I: Limitations and Difficulties	227
	8.1	Summary	227
	8.2	Surface Tension	227
	8.3	Tensile Instability	231
	8.4	Monaghan's Motion Equations	233
	8.5	Continuum Mechanics: Stress; Rigid-Body Rotation $\ . \ .$	236
	8.6	Dynamic and Static Constitutive Relations	237
	8.7	Example Deformations with Stress and Strain Rates	241
	8.8	Dynamics with Jaumann's Stress Rotation Rate	244
	8.9	Conservation of Angular Momentum	246
	8.10	Artificial Transport Coefficients	248
	8.11	Residual Stress—Artificial Plasticity in SPAM	249
	8.12	References	251
0	SDAM	I. Sample Applications to Solids	952
9.	SFAM	. Sample Applications to Solids	200
	9.1	Summary	253
	9.2	The Tension Test	254
	9.3	Tension Test via Standard Molecular Dynamics	256
	9.4	Boundary Conditions for Tension	257
	9.5	Initial Conditions for Tension Using SPAM	260
	9.6	Tension Test via SPAM-like Molecular Dynamics	261
	9.7	Tension Test via SPAM	263
	9.8	Failure Algorithms	266
	9.9	Penetration Mechanics	266
	9.10	Penetration via Continuum Mechanics	267
	9.11	Penetration via Standard Molecular Dynamics	270
	9.12	Penetration via SPAM-like Molecular Dynamics	271
	9.13	Penetration via SPAM	272
	9.14	A Research Suggestion	275
	9.15	References	276

xiii

Contents

10.	Summary, Literature, and Outlook			
	10.1	Introduction	277	
	10.2	Current State of the Art	278	
	10.3	Cutting and Machining	279	
	10.4	Structural Response to Waves	280	
	10.5	Dynamics of Sea Ice	281	
	10.6	Astrophysics	283	
	10.7	The Near Future of Parallel Computing	285	
	10.8	An Afterword	286	
	10.9	References	287	
Alphabetical Bibliography				
Index				
Example Problem List				

Chapter 1

Physical Ideas Underlying SPAM

/ Motivation / Particles versus Continua / Newton's Particle Mechanics / Eulerian and Lagrangian Continuum Mechanics / Microscopic Computer Simulation [Molecular Dynamics] / Statistical Mechanics / Macroscopic Particle Simulation / SPAM /

/ 7 Figures /

Example Problem : [Molecular Dynamics—Leapfrog and Runge-Kutta Methods]

1.1 Motivation and Summary

The flow of fluids, the deformation of solids, the design of structures able to withstand large and rapid loads, the analysis of complex failure mechanisms—all of these require numerical simulation techniques. Typical interesting problems of these types are nonlinear, and often "chaotic" as well, meaning that the evolving numerical solutions are exponentially sensitive to small changes in the initial conditions. SPAM is a simple and flexible numerical technique. It has developed so as to address the need for efficient designs of structures responding to rapid loading and for reliable analyses of fluid and solid flows and deformations.

Smooth Particle Applied Mechanics, "SPAM" for short, is a very general approach to the simulation of bulk matter in motion. Its novelty lies in a clever method for smooth interpolation and differentiation within an irregular grid of moving particles. The emergence of SPAM is a natural consequence of the fast high-capacity computers which make it possible to

Physical Ideas Underlying SPAM

attack complex numerical problems. Computers feed on *algorithms*, concrete numerical recipes¹ which can closely approximate the abstract mathematical ideas used to model and to analyze material motion. The last three centuries have seen Newton's analysis of particle motions governed by "action at a distance" abstracted to Euler and Lagrange's analyses of continua composed of unseen mutually interacting particles. The partial differential field equations of continuum mechanics, which include both time and space derivatives, can place heavy demands on digital computers, which are necessarily restricted to a finite number of variables described with finite precision.

Approximating the solution of the partial differential equations describing continuum problems by using *particles*, not atoms but *macroscopic* particles, is a clever idea. Clever ideas in physics often exhibit redundancy, showing up nearly simultaneously in more than one place. In 1977, after about 30 years of computer simulation, physics was ripe for the application of *particle* simulations to *continuum* problems. And it happened then that Gingold, Lucy, and Monaghan discovered the smooth-particle approach to continuum simulations while working at Cambridge University. Their joint discoveries of this new approach resulted from confronting challenging astrophylical problems. Gingold and Monaghan were interested in modeling rotating stars. They found that on the order of one hundred particles provided semiquantitative structural descriptions of such rotating bodies, including both gravitational and magnetic fields.² Lucy was likewise interested in rotating stellar matter, but beyond the stability limit. He was able to explain the production of close binary pairs of stars by following the detailed breakup of rotating masses, likewise represented by about one hundred particles.³ These early calculations demonstrated the ability of smooth particle techniques to deal with longrange gravitational forces. The electric and magnetic field forces found in plasma physics and magnetohydrodynamics can likewise be dealt with, but require special techniques which are unnecessary for the mechanical problems stressed in this book.

For those accustomed to precise analytic solutions the necessarily approximate nature of SPAM solutions might be disconcerting. The errors incurred, usually of the order of a few percent, are quite an acceptable price to be paid in exchange for the ability to solve hard nonlinear problems with a relatively simple and transparent method. The present book

 $\mathbf{2}$

¹Press, Flannery, Teukolsky, and Vetterling, Numerical Recipes (1986).

 $^{^{2}}$ Gingold and Monaghan (1977).

 $^{^{3}}$ Lucy (1977).

Particles versus Continua

is an elaboration and illustration of the refinement of these earlier ideas to applications in fluid and solid mechanics rather than to astrophysics. Numerical continuum problems are considered here in the context of our developing knowledge of particle mechanics, which has recently been enriched by ideas taken from chaos theory and from nonlinear dynamics.⁴ All the numerical methods discussed here will continue to profit from increases in computational speed and capacity.

Here we begin with a summary of classical mechanics, from Newton up until the development of fast computers. This foundation is followed by a quick tour of computational developments, spanning the last half century, which have made it possible to flesh out the earlier theoretical developments with numerical examples.

1.2 Particles versus Continua

Matter can be described by either particle mechanics or continuum mechanics. The two approaches are quite different, both in the underlying assumptions, and in the forms of the results obtained. Particle mechanics, as described in the next Section, and as illustrated in an Example Problem at the end of this Chapter, uses *ordinary* differential equations for r(t) to evolve particle trajectories. The underlying assumption is the functional form of the forcelaw on which the motion depends. No constitutive information (equilibrium equation of state and nonequilibrium transport laws) needs to be specified. All such information follows from the assumed form of the motion equations. Statistical mechanics and kinetic theory provide an understanding of the correspondence between microscopic time averages and macroscopic continuum behavior.⁵ The catch in applying this correspondence is that any desired equation of state and transport information has to be obtained from carefully chosen simulations, with no guarantee that the number-dependence of the results is small or that the state dependence of the constitutive properties is simple. In simulations based on particle mechanics the main problems are (i) making a clever choice for the interparticle forces, (ii) developing appropriate boundary conditions to drive the flow, and (iii) finding useful means for analyzing the results of simulations. The force and boundary choices together with the analyses are useful to the extent that they help to interpret physical reality.

⁴Gleick, *Chaos* (1989).

⁵Hoover, Computational Statistical Mechanics (1991).

Physical Ideas Underlying SPAM

Continuum mechanics begins instead with assumed constitutive relations (nothing whatever needs to be said about particles or forcelaws). These continuum constitutive relations, combined with the conservation laws for mass, momentum, and energy, take the form of *partial* differential equations. These partial differential equations are enough to evolve the state of a continuum, including the fluxes of mass, momentum, and energy within it, from given initial conditions, subject to the imposed boundary conditions. In simulations based on continuum mechanics the main problems are (i) a clever choice of constitutive relations and (ii) a robust algorithm for solving the evolution equations subject to the imposed boundary conditions.

The two approaches, particle and continuum, can agree with one another provided that the continuum constitutive equations correspond to the assumed interparticle forces, and provided that the microscopic fluctuations which are absent in continuum mechanics can be ignored. We will see that the two approaches can also be related to one another through Smooth Particle Applied Mechanics, a method which uses *particles* to solve problems in *continuum* mechanics. Before proceeding with a description of SPAM let us first consider the fundamentals of classical particle mechanics and continuum mechanics.

1.3 Newton's Particle Mechanics

Particle mechanics is built on Isaac Newton's Laws, which linked together earlier intuitive ideas in a new and analytic way. Space and time were primitive notions, with roots predating history. Mass and force are newer concepts, required for the foundation which underlies the structure of scientific understanding. *Mass* is a convenient measure for quantities both rare, the "Troy ounce", and common, the "metric tonne". Navigation and the definition of spatial boundaries require the notion of *length*. Meetings, for business or for pleasure, require *time*.

Newton linked these ideas together by applying his discovery of calculus and differential equations. His resulting Second Law of motion,

$$F = m\ddot{r} = m\dot{v} = ma ,$$

links together the notions of mass, length, and time through the limiting

Newton's Particle Mechanics

time rates of change,

$$\frac{\Delta r}{\Delta t} \rightarrow \dot{r} = v \ ; \ \frac{\Delta v}{\Delta t} \rightarrow \dot{v} = \ddot{r} = a \ ,$$

where the coordinate and velocity changes Δr and Δv occur during the infinitesimally-small time interval Δt . In this book we use superior single and double dots to indicate the first and second time "derivatives" in Newton's calculus. In his Second Law of motion force is F, mass is m, velocity is v, and acceleration is a. The coordinate r is measured in a Cartesian system, with components $\{x, y\}$ in two space dimensions and $\{x, y, z\}$ in three.

Solving Newton's motion equation where F is a constant vector (as in the frictionless motion of a mass point, or a nonspinning cannon ball, in a constant vertical gravitational field g) gives a parabolic trajectory :

$$a(t) = \dot{v}(t) = \ddot{r}(t) = F/m = g ;$$

$$v(t) = \dot{r}(t) = v(0) + gt ;$$

$$r(t) = r(0) + v(0)t + \frac{1}{2}gt^2 .$$

Notice that the two first-order (in time) ordinary differential equations require two initial conditions, { r(0), v(0) }, for a unique solution to result.

Once the system considered becomes even a little more complicated three bodies interacting with mutual inverse-square gravitational forces, for instance—where F_{ij} is the attractive force on mass m_i due to its "action at a distance" with mass m_i :

$$\{ F_{ij} = -Gm_i m_j (r_i - r_j) / |r_i - r_j|^3 \},$$

numerical methods are required to integrate the differential equations of motion.

Even a one-body problem, constrained by simple boundary forces, can have a relatively complicated "chaotic" (Lyapunov-unstable) solution, in which the growth of small errors is exponential in the time. See Sections 6.7 and 6.8 for a detailed approach to the exploration of Lyapunov instability. Analytic methods, based on truncated Taylor's series, are quite hopeless for such problems. Consider, for instance, a single particle with unit mass, placed at the (x, y) origin with $(\dot{x} = \dot{y} = 1)$. Assume that this particle

Physical Ideas Underlying SPAM

moves in the field of four fixed neighbors at $x = \pm \frac{1}{2}, y = \pm \sqrt{\frac{3}{4}}$ where the forces affecting the motion are derived from the short-ranged pair potential,

$$\phi(r < 1) = 100(1 - r^2)^4 \longrightarrow F(r < 1) = 800r(1 - r^2)^3$$
.



The full curve in Figure 1.1 shows the resulting motion. In the brokenline rendering the initial conditions are only slightly different :

$$\dot{x} \rightarrow \dot{x} + 10^{-5}$$
; $\dot{y} \rightarrow \dot{y} - 10^{-5}$

Figure 1.2, based on these same data, shows the *exponential* (Lyapunov unstable) growth of the deviation between these two trajectories.

The most useful numerical approach to solving such mechanics problems is the fourth-order Runge-Kutta integration used to generate the figures. The basic method is transparent as well as "self-starting". That is, given the current values, the algorithm advances the evolving trajectory by a specified time interval dt. This fourth-order method has single-step errors

 $\overline{7}$

Newton's Particle Mechanics

of order $dt^5/5!$ for a timestep dt. This integration technique, developed by Carl Runge from Martin Kutta's root-finding technique, estimates the changes of the dependent variables { r, v } from the known values r(t)and v(t) at time t to the desired values r(t + dt) and v(t + dt) at time t + dt as the weighted averages of four time derivatives. A simpler but lessaccurate second-order Runge-Kutta approach advances the coordinates and velocities through a timestep dt in two distinct stages, rather than four :

$$\{ \tilde{r}(t+dt) = r(t) + v(t)dt ; \tilde{v}(t+dt) = v(t) + a(t)dt \}.$$

$$\{ r(t+dt) = r(t) + \frac{dt}{2} [v(t) + \tilde{v}(t+dt)] ; v(t+dt) = v(t) + \frac{dt}{2} [a(t) + \tilde{a}(t+dt)] \}.$$



("Lyapunov unstable") growth of trajectory perturbations. dt = 0.001.

Physical Ideas Underlying SPAM

The first-guess accelerations { $\tilde{a}(t+dt)$ } are evaluated from the corresponding first-guess set of coordinates { $\tilde{r}(t+dt)$ }. The accuracy of this two-stage integration scheme is "second-order" in the timestep dt. This means that the coordinate and velocity errors incurred are of order dt^3 at each timestep. As a consequence, the error over a sufficiently-short, but fixed, time interval varies as dt^2 .

The better choice (more accurate, but still easy to program), justly popular after its hundred-year history of successes, is the classic four-stage fourth-order Runge-Kutta scheme :

$$\{ r(t+dt) = r(t) + \frac{dt}{6} [v(t) + 2\tilde{v}_2(t+\frac{dt}{2}) + 2\tilde{v}_3(t+\frac{dt}{2}) + \tilde{v}_4(t+dt)] \};$$

$$\{ v(t+dt) = v(t) + \frac{dt}{6} [a(t) + 2\tilde{a}_2(t+\frac{dt}{2}) + 2\tilde{a}_3(t+\frac{dt}{2}) + \tilde{a}_4(t+dt)] \}.$$

The new coordinates and velocities at time t + dt are accurate through terms of order dt^4 . The intermediate coordinates and velocities { \tilde{r}_i, \tilde{v}_i } are given by the time derivatives evaluated in three successive stages, from { $\tilde{r}_2, \tilde{v}_2; \tilde{r}_3, \tilde{v}_3; \tilde{r}_4, \tilde{v}_4$ }, after the initial evaluation of { $\dot{r}_1(t), \dot{v}_1(t)$ } from { r_1, v_1 } = { r(t), v(t) } :

$$\{ \tilde{r}_2(t+\frac{dt}{2}) = r(t) + \frac{dt}{2}v_1(t) \}; \{ \tilde{v}_2(t+\frac{dt}{2}) = v(t) + \frac{dt}{2}a_1(t) \};$$

$$\{ \tilde{r}_3(t+\frac{dt}{2}) = r(t) + \frac{dt}{2}\tilde{v}_2(t+\frac{dt}{2}) \}; \{ \tilde{v}_3(t+\frac{dt}{2}) = v(t) + \frac{dt}{2}\tilde{a}_2(t+\frac{dt}{2}) \};$$

$$\{ \tilde{r}_4(t+dt) = r(t) + dt \tilde{v}_3(t+\frac{dt}{2}) \} ; \{ \tilde{v}_4(t+dt) = v(t) + dt \tilde{a}_3(t+\frac{dt}{2}) \} .$$

The coordinates and velocities at each of the latter three stages follow from the velocities and coordinates computed in the previous stage. The last set of intermediate coordinates and velocities, { \tilde{r}_4, \tilde{v}_4 }, are required for the final fourth-order derivatives.

Finally the derivatives at the four stages,

$$\{\dot{r}_1(t), \dot{v}_1(t); \dot{r}_2(t+\frac{dt}{2}), \dot{v}_2(t+\frac{dt}{2}); \dot{r}_3(t+\frac{dt}{2}), \dot{v}_3(t+\frac{dt}{2}); \dot{r}_4(t+dt), \dot{v}_4(t+dt)\},$$

are averaged, with weights { 1/6, 1/3, 1/3, 1/6 }, to advance { r(t), v(t) } to { r(t + dt), v(t + dt) }. Such multistage Runge-Kutta algorithms are most easily programmed by storing all of the dependent variables ({ r, v } in this case) in a vector (of length 4N for N particles in two-dimensional space and 6N for N particles in three-dimensional space). Their computed time derivatives { \dot{r}, \dot{v} } are stored in a second vector of the same length. For more details see Section 4.4.

Eulerian and Lagrangian Continuum Mechanics

A system composed of Newtonian particles requires a definite number of "degrees of freedom" (independent coordinates) for a complete description of its "configuration" in space. For point masses the number of degrees of freedom is the dimensionality multiplied by the number of particles. For rigid bodies additional *rotational* degrees of freedom (angles) are required, one of them in two dimensions and two in three dimensions. Because computer storage is relatively cheap, the number of degrees of freedom which can be followed numerically depends primarily on computer processor speeds and the number of processors. Machines with *millions* of processors are now on the drawing boards.⁶

Today Runge-Kutta integration makes it possible to follow the motion of *billions* of mutually interacting particles for as many as a *million* timesteps. This is the limit of what can be done today. The timestep *dt* cannot be too large. It is limited by the time required for adjacent particles to exchange information at the speed of sound. If the particles are microscopic models of atoms then the total simulated time is pitifully small, no more than a microsecond of real time, and there is no prospect whatever for reaching times of even one second for macroscopic amounts of matter. The physical extent of such an atomistic model is likewise very limited. It can be no more than a cubic micron. By contrast, the original Gingold-Lucy-Monaghan applications of smooth particles were *astrophysical*, with interstellar particle length scales and correspondingly long times. Today—and even in the forseeable future—it is absolutely necessary to consider a "continuum" description of matter to simulate and discuss interesting and challenging scientific problems on the length and time scales relevant to men.

1.4 Eulerian and Lagrangian Continuum Mechanics

Matter distributed continuously in space, with a mass density ρ depending upon location $\rho(r)$, can require an *infinite* number of degrees of freedom for its description. But so long as the material properties vary smoothly in space it is possible to develop *exact* differential equations governing the properties' time evolution. Consider a one-dimensional flow in which the density $\rho(x,t)$ and velocity $v_x(x,t)$ vary continuously and differentiably in space and time (so that the spatial derivatives $\partial \rho / \partial x$ and $\partial v_x / \partial x$ are also well-defined continuous functions in space and time). Focus on a sufficiently small element of length dx, fixed in space. During a short time

⁶R. Preston, New Yorker Magazine, 11 April 2005.

Physical Ideas Underlying SPAM

interval dt the net change of mass within dx can then be expressed as the difference of the (ρv_x) flows in and out at the element's two endpoints,

$$[(\rho v_x)_{x-(dx/2)} - (\rho v_x)_{x+(dx/2)}]dt$$
.

With the length dx fixed, the change of mass with time can be simplified :

$$\frac{\partial(\rho dx)}{\partial t} \equiv dx \frac{\partial \rho}{\partial t} = (\rho v_x)_{x - (dx/2)} - (\rho v_x)_{x + (dx/2)} .$$

In the limit that both the time interval dt and the element length dx are sufficiently small, this statement of mass conservation gives an *exact* partial differential equation (the continuity equation) for the changing mass density at a fixed location :

$$\frac{\partial \rho}{\partial t} = -\frac{\partial (\rho v_x)}{\partial x} = -\nabla_x \cdot (\rho v_x) \ .$$

In three space dimensions, where the flow velocity is the vector $v = (v_x, v_y, v_z)$, the continuity equation has exactly this same form :

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho v) = -\nabla_x (\rho v)_x - \nabla_y (\rho v)_y - \nabla_z (\rho v)_z .$$

Again the *partial* derivative $\partial \rho / \partial t$ indicates the change of density at a fixed point in space.

An alternative description, using the "comoving" time derivative, the derivative following the motion, $d/dt \equiv \cdots$ (a "superior dot"), can be evaluated from the evolution of two nearby points, embedded in the material and initially separated by dx. For simplicity, consider the evolution of the two neighboring points in one dimension $\{x_0 \pm \frac{dx}{2}\}$ for a small time interval dt and a sufficiently small moving element of length dx:

$$\{x_0 \pm \frac{dx}{2}\} \longrightarrow \{x_0 \pm \frac{dx}{2} + v_0 dt \pm \frac{dx}{2} (\partial v / \partial x)_t dt\}$$

Evidently the initial separation of the points $dx(t_0) \equiv x_{\text{right}} - x_{\text{left}}$ changes whenever there is a velocity gradient. To first order in dt the change is $(\partial v/\partial x)_t dx dt$ during the time dt:

 $\begin{aligned} x_{\text{right}}(t_0) &\to x_{\text{right}}(t_0) + vdt + \frac{dx}{2}(\partial v/\partial x)_t dt ; \\ x_{\text{left}}(t_0) &\to x_{\text{left}}(t_0) + vdt - \frac{dx}{2}(\partial v/\partial x)_t dt ; \\ dx(t_0) &\to dx(t_0)[1 + (\partial v/\partial x)_t dt] = dx(t_0 + dt) ; \end{aligned}$

Eulerian and Lagrangian Continuum Mechanics

$$\frac{[dx(t_0+dt)-dx(t_0)]}{dtdx(t_0)} \to \frac{d\ln dx}{dt} \equiv -\frac{d\ln \rho}{dt} = \frac{\partial v_x}{\partial x}$$

In two or three dimensions the corresponding *comoving* version of the continuity equation is as follows :

$$\frac{d\rho}{dt} = \dot{\rho} = -\rho \nabla \cdot v \longleftrightarrow \frac{d\ln\rho}{dt} = -\nabla \cdot v \; .$$

The continuity equation is a fundamental representation of mass conservation, equating the mass change to differences of mass flow.

The change and flow of momentum within a small moving volume element is calculated in a similar way, except that an additional momentum flux, due to "action at a distance" (from forces) adds the *comoving* momentum flux $P \equiv -\sigma$ to the convective flux ρvv :

$$\frac{\partial(\rho v)}{\partial t} = -\nabla \cdot (P + \rho v v) \equiv \nabla \cdot (\sigma - \rho v v)$$

This "motion equation", which gives the evolution of v and hence—by integration—the evolution of the system's configuration { r }, is a fundamental representation of momentum conservation. Just as in deriving the continuity equation, the equation of motion results from equating an overall change to a difference of corresponding flows. P is the pressure tensor, the negative of the stress tensor σ , $P \equiv -\sigma$. P is a momentum flux. P_{xy} , for instance, the xy component of P, is the comoving flow of x momentum in the y direction.

It is usual to term the evolution equations at a *fixed* point in space the "Eulerian" evolution equations. For the continuity equation and equation of motion the Eulerian forms are :

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho v) \; ; \; \frac{\partial (\rho v)}{\partial t} = -\nabla \cdot (P + \rho v v) \; .$$

The equivalent *comoving* evolution equations are termed "Lagrangian". Lagrangian time derivatives are defined for coordinates which follow the motion. The Lagrangian continuity and motion equations are :

$$\dot{\rho} = -\rho \nabla \cdot v$$
; $\dot{v} = -[\nabla \cdot P]/\rho = [\nabla \cdot \sigma]/\rho$.

Either the Eulerian or the Lagrangian equations can serve as the basis of numerical simulation techniques.

In addition to the flow of mass, as described by the "continuity equation", and the flow of momentum, as described by the "equation of motion", continuum mechanics also accounts for the flow of energy, a third conserved

Physical Ideas Underlying SPAM

quantity. Just as in the case of momentum flux, experience shows that the flow of energy can include both "convective" and "conductive" effects. Convection is bodily transport (as is plainly visible in moving clouds or flames) which takes place at the local fluid velocity v. Conduction is that *additional* transport which still occurs in the absence of fluid motion (or, equivalently, transport occurring in the coordinate frame *comoving* with the fluid). The transmission of heat in a motionless solid rod is the simplest example of comoving energy conduction. Momentum is likewise transmitted through the mechanism of forces (action at a distance) as is described by the pressure tensor. The distinction between free-streaming convective transport and collisional conductive transport is obvious and natural in particle mechanics. The same distinction requires new, and less obvious, phenomenological relations in continuum mechanics. On a macroscopic continuum level the stress tensor $\sigma = -P$ [momentum flow per unit area and time] and heat flux vector Q [energy flow per unit area and time] replace microscopic interparticle forces as the underlying *ad hoc* nonconvective mechanisms for transporting momentum and energy.

From the classical point of view of particle mechanics it is natural to separate the total energy E into two parts, potential and kinetic, $E = \Phi + K$. In the simplest case, a Cartesian-coordinate description of mutually interacting points, the potential energy depends solely on the coordinates and the kinetic energy depends solely on the velocities. For isolated systems the total energy E is fixed. Consider, as a simple example, a single harmonic oscillator, with coordinate q, momentum $p = m\dot{q}$, and force constant κ :

$$E = \phi + K \ ; \ \phi = rac{\kappa q^2}{2} \ ; \ K = rac{p^2}{2m} = rac{m \dot{q}^2}{2} \ .$$

From a given initial condition the state vector (q, p) evolves on the constant-energy ellipse, E(q, p) = E(t = 0). In continuum mechanics the system's total energy density—energy per unit volume, with units $[mass][length]^{2-D}[time]^{-2}$ in D space dimensions—is likewise separated into two parts ,

$$\rho e_{\text{total}} = \rho e + (\rho v^2/2) ,$$

where e is the "internal energy" per unit mass. It is important to recognize that the internal energy of continuum mechanics includes not only the microscopic particle potential energy, but also that part of the particles' kinetic energy which corresponds to macroscopic "heat" energy. The kinetic energy density associated with macroscopic motion, $\frac{\rho v^2}{2}$, or the corre-

Eulerian and Lagrangian Continuum Mechanics

sponding contribution from rotation, makes no contribution to the "internal energy" per unit mass e .

Like the changes of kinetic and potential energy, the macroscopic concepts of heat and work refer to energy changes associated with microscopic motion ("heat") and with macroscopic motion ("work") with the latter energy change resulting from the time-variation of macroscopic coordinate variables. The First Law of Thermodynamics is usually written as a straightforward energy balance, in which the change of total system energy, dE, has two parts, the heat absorbed from the surroundings, dQ, and the work done on those surroundings, dW. If the various changes occur within a short time dt, and can still be separated into thermal and mechanical parts, then the First Law of Thermodynamics can be written as a differential equation :

$$dE = dQ - dW \longleftrightarrow \dot{e} = (\dot{E}/m) = (\dot{Q}/m) - (\dot{W}/m)$$

The latter expression can also be applied locally, where the energy per unit mass e = E/m is a *field* variable, defined at all points within the system. The energy changes $\dot{Q}dt$ and $\dot{W}dt$ due to heat absorbed and work done are both considered in the *comoving* frame which moves with the system's local velocity. The differential form of the First Law separates energy changes (in this comoving frame) into conductive changes ("heat") and energy changes associated with coordinates ("work"). Work can include deformation as well as motion within an external field. This separation of energy changes into heat and work is required for a thermodynamic description.

Mechanics introduces the notion of pressure as the force driving volume changes. Thermodynamics introduces a second notion, temperature, as the driving mechanism for conductive energy changes, the flow of heat. The microscopic analogs of these macroscopic concepts invariably involve averaging, usually over a large number of particles and a long time.⁵ Temperature, for instance, is most simply defined as the mean (microscopic) kinetic energy per Cartesian degree of freedom (when measured in the "comoving" coordinate frame which moves at the material velocity v) :

$$DkT/2 \equiv \langle m(v_{\rm micro} - v)^2/2 \rangle$$

in D space dimensions, where k is Boltzmann's constant. A more general development introduces the ideal-gas thermometer based on this definition, with the thermodynamic temperature T characterizing any system (even

⁵Hoover, Computational Statistical Mechanics (1991).

Physical Ideas Underlying SPAM

a quantum system) with which the (classical) thermometer is in thermal equilibrium.

1.5 Computer Simulation of Microscopic Particle Motion

Molecular dynamics, simulating the evolution of microscopic particle trajectories with computers, has been an active research field for over half a century. The early simulations, at Brookhaven,⁷ Livermore,⁸ and Los Alamos,⁹ were mostly devoted to solving Newton's equations of motion ,

$$\left\{ \begin{array}{l} \ddot{r}=\dot{v}=\frac{F}{m} \end{array} \right\} \ ,$$

where the force F on each particle is the vector sum of contributions from its nearby neighbors. These Newtonian motion equations apply to isolated systems for which the total system energy is a constant of the motion.

Vineyard's studies of radiation damage in solids were an exception to this rule. He introduced special dissipative viscous boundary forces ,

$$\left\{ \begin{array}{c} \frac{F_{\rm viscous}}{m} \equiv -\frac{v}{\tau} \end{array} \right\} \;,$$

designed to reduce the influence of boundaries on interior dynamics. The parameter τ was chosen to optimize the damping of internally-generated sound waves incident on the system boundary.

Much of the early activity in molecular dynamics was devoted to evaluating the thermal and mechanical equations of state for given potential energy functions :

$$\Phi \longrightarrow E(V,T)$$
, $P(V,T)$.

Temperature was evaluated by using the ideal-gas-thermometer definition:

$$\langle p_x^2 = p_y^2 = p_z^2 \rangle \equiv mkT ; p = mv$$
,

for particles of mass m in thermal equilibrium with an ideal gas thermometer maintained at a temperature T. Here k is Boltzmann's constant.

Pressure was evaluated by measuring the mean momentum flux, which separates into kinetic and potential contributions. For a system with

⁷Vineyard, with Gibson, Goland, and Milgram (1960).

⁸Alder and Wainwright (1958).

⁹Fermi, Pasta, and Ulam, as described by Tuck and Menzel (1972).

Computer Simulation of Microscopic Particle Motion

pairwise-additive forces,

$$\Phi = \sum_{i < j} \phi_{ij} \longrightarrow F_i = \sum_j F_{ij}(r_{ij}) \; ; \; r_{ij} \equiv r_i - r_j \; ; \; \{ \; i, j \; \} \; \subset \; V \; ,$$

the instantaneous pressure tensor P, the local average of momentum flux in a comoving volume V, has the form :

$$PV = \sum_{i} \frac{p_{i}p_{i}}{m} + \sum_{i < j} r_{ij}F_{ij} \; ; \; r_{ij} = r_{i} - r_{j} \; ; \; F_{ij} = -\nabla_{i}\phi_{ij} \; .$$

To clarify the meaning of our shorthand notation let us write out the four components { $P_{xx}, P_{xy}, P_{yx}, P_{yy}$ } of the pressure tensor in two space dimensions :

$$P_{xx}V = \sum_{i} (p_x^2/m)_i + \sum_{i < j} (x^2F/r)_{ij} ;$$

$$P_{xy}V = \sum_{i} (p_xp_y/m)_i + \sum_{i < j} (xyF/r)_{ij} = P_{yx}V ;$$

$$P_{yy}V = \sum_{i} (p_y^2/m)_i + \sum_{i < j} (y^2F/r)_{ij} ;$$

with the definitions of the pair quantities :

$$x_{ij} = x_i - x_j$$
; $y_{ij} = y_i - y_j$; $r_{ij} = \sqrt{x_{ij}^2 + y_{ij}^2}$; $F_{ij} = F(r_{ij})$.

The tensor components of P describe the flux (flow per unit area and time) of momentum in space, so that P_{xy} describes the flow of x momentum in the y direction. Rotational stability requires that P be symmetric, $P_{xy} = P_{yx}$, not just for pairwise-additive systems, but in general.⁵

Approximately 25 years were required to complete this phase of the development of molecular dynamics. It is now routine to generate accurate thermodynamic properties for classical systems with given interparticle forces. Work has continued, mostly on more elaborate models for polyatomic molecules, but such activities are hampered by difficulties in forcing quantum systems ("real matter") to be described by a classical model.

During the last 25 years methods based on molecular dynamics have been developed for treating the *non*equilibrium problems more closely related to the subject of this book.¹⁰ On the particle level there is no problem

⁵See page 189 of my *Computational Statistical Mechanics* (1991).

 $^{^{10}}$ Hoover (1997).

ws-book9x6

Physical Ideas Underlying SPAM

in formulating the flows of momentum and energy (the pressure tensor P and the heat flux vector Q). The microscopic expressions for these fluxes have no special nonequilibrium parts. They have exactly the same forms at equilibrium as they do away from equilibrium. The main difficulties in simulating nonequilibrium microscopic systems are (i) formulating proper boundary conditions to drive the flows away from equilibrium and (ii) absorbing the heat these flows generate. Feedback and control are the familiar concepts engineers would use and implement in order to deal with these latter needs.

Shuichi Nosé formalized and implemented the key new idea required for *control* in molecular dynamics: *thermostat forces* designed to enforce a specified *time-averaged* ideal-gas temperature on particular degrees of freedom.¹¹ He and Hans Andersen and I also formulated *barostat forces* designed to enforce a particular average pressure (by making the corresponding volume a function of time). Nosé's ideas were a direct outgrowth of Gibbs' statistical mechanical approach to equilibrium averages, but have the distinct advantage of being equally applicable to nonequilibrium systems. Nosé's ideas make it possible to simulate stationary shear flows and heat flows, giving direct measurements of shear viscosity and thermal conductivity, even far from equilibrium. The consequences of his thermostat forces for the development of statistical mechanics are discussed in the next Section.

1.6 Liouville's Theorem; Statistical Mechanics

Liouville's Theorem provides an interesting link between Nosé's mechanics and the old (1883) Gibbs-Boltzmann equilibrium statistical mechanics.¹² Liouville's Theorem is a fundamental building block for equilibrium statistical mechanics. It describes the flow of the probability density f(q, p, t) in the "phase space" composed of all the particle coordinates and momenta, $\{q, p\}$. The underlying (q, p) flow is governed by the "Hamiltonian" \mathcal{H} :

$$\mathcal{H} = K(p) + \Phi(q) \rightarrow \left\{ \dot{q} = +\frac{\partial \mathcal{H}}{\partial p} ; \dot{p} = -\frac{\partial \mathcal{H}}{\partial q} \right\}$$

The Theorem is a direct consequence of Hamiltonian mechanics. It

 $^{^{11}}$ Nosé (1991).

¹²Hoover, Journal of Chemical Physics (1998).

Liouville's Theorem; Statistical Mechanics

establishes that the time derivative of f , following the motion, vanishes :

$$\left(\frac{\partial f}{\partial t}\right) = -\sum \frac{\partial (f\dot{q})}{\partial q} - \sum \frac{\partial (f\dot{p})}{\partial p} \rightarrow$$
$$\dot{f} = df/dt = \left(\frac{\partial f}{\partial t}\right) + \sum \dot{q} \cdot \nabla_q f + \sum \dot{p} \cdot \nabla_p f =$$
$$-f \sum \left(\frac{\partial \dot{q}}{\partial q} + \frac{\partial \dot{p}}{\partial p}\right) = -f \sum \left(\frac{\partial^2 \mathcal{H}}{\partial q \partial p}\right) + f \sum \left(\frac{\partial^2 \mathcal{H}}{\partial p \partial q}\right) \equiv 0$$

where the sums include all the degrees of freedom required to describe the system's "phase" (q, p). Evidently it is an exact consequence of Hamiltonian mechanics that *equilibrium* systems, for which $\partial f/\partial t$ must vanish also, correspond to phase space distributions in which f(q, p) is exactly the same for all accessible states :

$$\dot{f} = 0 \longrightarrow \left(\frac{\partial f}{\partial t}\right)_{\rm eq} \equiv 0$$

Nosé's *thermostated* systems, described with thermostat forces, can also be discussed from the standpoint of Liouville's Theorem, modified to include the frictional contributions of the nonHamiltonian "Nosé-Hoover" thermostat forces $\{ -\zeta p \}$:

$$F = -\nabla \Phi - \zeta p \rightarrow -f \frac{\partial \dot{p}}{\partial p} = f \zeta \rightarrow$$

$$\dot{f} = 0 \xrightarrow{\zeta} \dot{f}/f = \sum_{\{p\}} \zeta = \dot{S}/k = -\sum_{\{T\}} \dot{Q}/kT$$
.

The "friction coefficient" ζ obeys a feedback equation based on the ratio of the current kinetic energy K to the desired average K_0 , with both represented as sums over all the thermostated degrees of freedom :

$$\dot{\zeta} = \left[\left(\sum \frac{p^2}{m} / \sum kT_0 \right) - 1 \right] / \tau^2 \ .$$

Here τ is a phenomenological relaxation time describing the response of the Nosé-Hoover thermostat with temperature T_0 corresponding to the kinetic energy K_0 . See Reference [5] for more details and many worked-out examples.

ws-book9x6

Physical Ideas Underlying SPAM

With Nosé's friction included, the comoving phase-space density f changes according to \dot{S} , the external entropy production rate due to transfers of heat from the reservoirs at $\{T\}$, at the rates $\{\dot{Q}\}$, to the system. These are the rates at which heat is transferred from Nosé's reservoirs divided by the corresponding reservoir temperatures. In nonequilibrium steady states heat flows from the system to the reservoirs, on average, so that $\langle \dot{Q}/T \rangle$ is negative, and the external entropy increases. The transferred heat increases the states available to the reservoirs while correspondingly decreasing the states available to the nonequilibrium system interacting with those reservoirs.

Entropy had already been associated with phase-space states by Gibbs¹³ and Boltzmann¹⁴ by 1883. Both men recognized the probabilistic tendency of chaotic mechanical systems to explore as many such states as possible and characterized the "approach to equilibrium" in terms of a natural increase in the number of available states. For over a century nonequilibrium systems have been viewed as approaching equilibrium through an increase in the number of available states. Nosé's approach leads instead to a new understanding of nonequilibrium systems in terms of the intrinsic rarity of their states. In Nosé's mechanics the overall increase in available states occurs only within the external heat reservoirs where the nonequilibrium system deposits its generated heat. The nonequilibrium system coupled to these reservoirs has itself fewer and fewer states available as time goes by. Nosé's feedback constraint forces restrict the system dynamics into an ever-smaller fractional-dimensional region of phase space.

Thus external entropy production—gain—corresponds to internal entropy loss, with the system's phase-space density shrinking onto a nonequilibrium "attractor" of ever smaller volume. Because the external entropy production for nonequilibrium stationary states is necessarily positive, the internal phase-space probability density f grows exponentially with time as its phase-space extent shrinks. The nonequilibrium phase-space distribution function $f(q, p, \zeta, t)$ eventually diverges in any steady state so that the corresponding phase volume described by the stationary state actually vanishes. The new dynamical approach developed by using Nosé's mechanics to simulate nonequilibrium steady states shows very directly that such nonequilibrium states are extremely rare in the sense that they occupy a "multifractal" (fractional-dimensional) vanishing volume of phase space.

Figure 1.3 illustrates this nonequilibrium volume and dimensionality

 $^{^{13}}$ Gibbs (1902).

 $^{^{14}}$ Boltzmann, as described by Cercignani (1998).

Liouville's Theorem; Statistical Mechanics

reduction for the simplest possible system, the "Galton Board", in which a single particle is accelerated by gravity while moving within a lattice of fixed scatterers.¹⁵ The ensuing chaotic motion is constrained to occur at constant speed [which is equivalent to a constant kinetic temperature proportional to $(p_x^2 + p_y^2)/mk$]. This constant-speed motion generates a (time-averaged) stationary nonequilibrium state. The (time-averaged) net effect of the motion is to transfer gravitational field energy to a heat reservoir through the medium of the moving particle. This energy transfer, corresponding to a heat loss, substantially reduces the states available to the moving thermostated particle, as is shown in Figure 1.3.



Figure 1.3: Galton Board phase-space states at equilibrium (left) and away from equilibrium (right) illustrating the fractal nature of microscopic nonequilibrium states. The abscissa gives the location of the collisions, $0 < \alpha < \pi$, with π corresponding to the "top" of the scatterer, and 0 corresponding to the bottom. The ordinate $-1 < \sin \beta < +1$ corresponds to the post-collision tangential velocity component.

The Figure shows the available phase-space states, both at and away from equilibrium, for the collisions of a single thermostated hard disk moving through an infinite periodic triangular lattice of fixed disks. In both cases the motion occurs at constant kinetic energy either with or without a vertical external accelerating field. With the field off (equilibrium) all the states in the rectangular phase-space section shown in the Figure are

¹⁵Moran, Hoover, and Bestiale (1987).

Physical Ideas Underlying SPAM

equally likely. With the field on (nonequilibrium steady state) the distribution becomes a zero-volume "fractal" singular distribution. Various fractional dimensionalities have been defined and can be evaluated for relatively simple problems like this one. The "information dimension" of the attractor based on $\langle f \ln f \rangle$ is 1.8, while the "correlation dimension" based on $\langle f^2 \ln f \rangle$, is 1.6. There is a vast literature on this Galton Board problem.

1.7 Simulating Continua with Particles

Continuum mechanics is a "field theory" in which the various field variables $(\rho, v, \sigma, Q, ...) = (\text{density, velocity, stress, heat flux, ...})$ vary continuously in both space and time. Because in principle continua involve an infinite number of degrees of freedom, it is necessary in practical computations somehow to truncate the description to a manageable finite number.

In some cases the field variables can be usefully expressed as truncated Fourier series. Another approach to making the simulation work manageable is to use the "finite-element method". That method visualizes the continuum as partitioned into a set of contiguous space-filling "finite elements", each described by a finite number of degrees of freedom. The field variables within a particular element then vary according to an appropriate "shape function". A linear representation is the simplest. Suppose that the continuum variable C(x) has specified values at the two endpoints of an interval of length dx centered at x. In one dimension the *linear* interpolation of C(x) = a + bx between the two known values at the ends of the interval requires just the two constants (a, b):

$$C_{-} = C(x - \frac{dx}{2}) \text{ and } C_{+} = C(x + \frac{dx}{2}) \longrightarrow a = \frac{C_{-} + C_{+}}{2} - bx \text{ ; } b = \frac{C_{+} - C_{-}}{dx}$$

A similar linear interpolation function can be developed in two dimensions (a + bx + cy) for triangular elements) or in three dimensions (a + bx + cy + dz) for tetrahedral elements). This is a natural approach to follow in a Lagrangian simulation, in which the elements follow the motion of the continuum. If the variation of a continuum property C(x, y) within a two-dimensional element has the "bilinear" shape function, a + bx + cy, it is evident that the shape functions for any two contiguous triangles must agree along their common linear boundary. On the other hand, with this bilinear approach, the gradient of C perpendicular to such a common boundary is typically discontinuous. Continuous spatial derivatives would require at least quadratic shape functions.

Simulating Continua with Particles

"Finite difference methods" are another possibility. These are best suited to Eulerian calculations for which a fixed Cartesian grid made up of line segments of equally-space points, or of squares in two dimensions, or cubes in three, is the simplest choice. Replacing the spatial derivatives $\{ \nabla \cdot v , \nabla \cdot P , \nabla \cdot Q \}$ on the righthandsides of the continuum equations by appropriate differences (based on Taylor's series) then gives rise to a finite set of ordinary differential equations for the time development of the nodal values.

Consider a regular one-dimensional grid of points separated by line segments of length dx, with each line-segment endpoint characterized by its own temperature T. Recall that the phenomenological equation for heat transfer in one dimension is the Fick-Fourier law :

$$\frac{\partial T}{\partial t} = D_T \frac{\partial^2 T}{\partial x^2} \; ,$$

where D_T is the "thermal diffusivity", with units $[length]^2/[time]$. For the discrete fixed-grid-point model of a heat-conducting continuum, this *partial* differential heat-transfer equation is replaced by a finite-difference approximation to the spatial derivative, corresponding to the approximate set of *ordinary* differential equations :

$$\left\{ \dot{T}_{i} = \frac{D_{T}}{dx^{2}} \left[T_{i+1} - 2T_{i} + T_{i-1} \right] \right\} ,$$

for the grid-point temperatures' evolution. The underlying assumption is that the second-difference representation of ∇^2 [with its accompanying error of order $(dx)^2 \nabla^4 T$] is adequate. More details and applications of the finite-difference approach are given in the next Chapter.

When material motion ("flow") is included in the evolution it is usual to contrast Eulerian and Lagrangian techniques. The Eulerian approach uses a fixed grid. This approach has difficulty in tracking the motions of the *boundaries* of the various described materials. The Lagrangian approach makes use of a *comoving* grid (moving with the material velocity) so that material boundaries define, and coincide with, element boundaries.

The Lagrangian finite-element approach, with moving triangles or tetrahedra (or with more elaborate elements, such as shells or hexahedra) has difficulty in dealing with largescale irregular deformations. If the grid becomes badly distorted, with strains of order unity, so that the zones can become tangled, the whole formalism fails. This is because the finite-difference formulæ, which properly describe the derivatives and surface normals needed for flux calculations, are only accurate for *small* deformations. They become

ws-book9x6

Physical Ideas Underlying SPAM

increasingly less accurate for large ones and eventually lead to instabilities. Figure 1.4 shows a Lagrangian mesh, composed of shell elements which deform in response to external shear and compressive forces.¹⁶



A way to avoid the instability problems associated with Eulerian interface tracking and Lagrangian distortion and tangling is to compute derivatives of the continuum field variables on an *irregular* grid, where the grid is itself composed of sufficiently many moving particles. Though the use of an irregular grid to combat distortional instability seems paradoxical, the idea works because the "grid" of particles lacks a memory of its ini-

¹⁶Hoover and Hoover, (2005).

Simulating Continua with Particles

tial configuration, and is consequently self-healing. This particle-grid idea is included in the smooth-particle approach discovered jointly by Gingold, Lucy, and Monaghan. It is described in the next Section. Alternative approaches, which are computationally more cumbersome, consider *particles* moving within a regular Eulerian grid of field points. The "particle-in-cell" method is an example.¹⁷



¹⁷Birdsall and Langdon, Plasma Physics via Computer Simulation (1985).
ws-book9x6

Physical Ideas Underlying SPAM

1.8 SPAM [Smooth Particle Applied Mechanics]

SPAM began in 1976-1977 at Cambridge University's Institute of Astronomy. The discoverers were Robert Gingold, Leon Lucy, and Joseph Monaghan. At that time their research interests centered on complex astrophysical problems which were ill-suited to conventional Eulerian and Lagrangian simulation techniques. Their basic idea was, and is, to express all of the continuum variables on a grid composed of moving particles. If each particle is itself viewed as being distributed over space, with a normalized weight function w, then the mass density ρ at any point r in space can be defined as the sum of the densities $\{mw\}$ of all particles close enough to contribute:



This weight function gives the *simplest* possible smooth-particle recipes, with continuous first and second derivatives, as detailed in Chapter 3.

SPAM [Smooth Particle Applied Mechanics]

$$\rho(r) = \sum m_j w_{rj} ; w_{rj} = w(|r - r_j|).$$

Lucy's weight function, which we use throughout this book to formulate and solve many of our example problems, is illustrated in Figure 1.5.

Continuum properties associated with the particles can likewise be averaged to compute local values :

$$C(r) \equiv \sum m_j w_{rj} C_j / \sum m_j w_{rj} \to \rho(r) C(r) \equiv \sum m_j w_{rj} C_j \ .$$

This intuitive recipe for averages has particularly useful consequences for evaluating gradients, as we will see in detail in Chapter 3. Note particularly that only the weight function $w_{rj} = w(|r - r_j|)$ is affected by the gradient operator ∇_r :

$$abla_r(\rho C)_r \equiv \nabla_r \sum_j m_j w_{rj} C_j = \sum_j m_j C_j \nabla_r w_{rj} \; .$$

There is a significant difference between smooth-particle interpolation and the simpler Eulerian finite-difference and Lagrangian finite-element interpolation schemes. In the smooth-particle case the continuum variables associated with a particular particle, { C_i }, typically *differ* from the field variables evaluated at the location of the particle, { $\langle C(r_i) \rangle$ } because the latter averages include additional contributions from nearby particles. Gradients evaluated *at* the particles' locations { $\langle \nabla C \rangle(r_i)$ } can then be used to evaluate the evolution of particle properties as solutions of ordinary differential equations { \dot{C}_i }. Besides an elegant simplicity, this smoothparticle approach has the additional advantage that fracture and failure can be made to occur naturally and automatically. Even complex astrophysical problems can be treated with SPAM. A relatively recent example is the collision of a comet with the back side of Jupiter.¹⁸

I spell out and illustrate the remaining details of the numerical method in Chapter 3. Because the technique closely resembles molecular dynamics, but with more elaborate accelerations, we lay the groundwork here by sketching the molecular dynamics approach as it applies to a simple 16-particle example.

ws-book9x6

¹⁸Ahrens, Takata, O'Keefe, and Orton (1994).

Physical Ideas Underlying SPAM

1.9 Example: A Molecular Dynamics Simulation

To illustrate the simplest mass-point particle dynamics, "molecular dynamics", consider a simple dynamical simulation. We begin with the 16 particles shown in the Figure 1.6 . The initial configuration is a square lattice. The initial velocities were first chosen at random from the square $(-1 < v_x, v_y < 1)$, and then modified to satisfy two constraints, (i) vanishing center-of-mass velocity :

$$\left(v_i \to v_i - \frac{1}{16}\sum v\right) \longrightarrow \sum v \equiv 0;$$

and (ii), a prescribed initial kinetic energy :

$$\left(v_i \to v_i \sqrt{32/\sum v^2}\right) \longrightarrow \sum v^2 \equiv 32$$
.

The sums here include all 16 particles. The random number generator used to choose their initial velocities appears in Section 4.5.

Two different integration algorithms were used to generate the configurations illustrated in Figure 1.6. The fourth-order Runge-Kutta method used here was already detailed in Section 1.3. The corresponding Fortran and C codes for this integrator are given in Section 4.4. The simpler, but less accurate, "leapfrog" algorithm, also used here, is equivalent to a simple recipe for computing the coordinates at time t + dt from two previously known coordinate values together with the current acceleration at time t. Rearrange the approximate expression for the acceleration,

$$dt^2 a(t) \simeq r(t+dt) - 2r(t) + r(t-dt)$$
,

so as to *define* the new coordinate r(t + dt) in terms of its predecessors at times t and t - dt:

$$r(t+dt) \equiv 2r(t) - r(t-dt) + dt^2 a(t) .$$

A comparison of the Taylor's series expansion (in dt) for the two sides shows that the single-step error in the new coordinate is of order $\frac{1}{12}dt^4\ddot{a}(t)$, so that the algorithm is "third-order", accurate through terms of order $\dot{a}dt^3$.



Example: A Molecular Dynamics Simulation

Figure 1.6: 16 particles before (left top) and after (center top) 24,000 fourth-order Runge-Kutta timesteps of dt = 0.001 each. The pair potential used here is $\phi(r < 1) = 100(1 - r^2)^4$. Velocity reversal, at t = 24, leads to the configuration at the right. The corresponding snapshots in the bottom row correspond to the same times and initial conditions, but to the use of a larger timestep, dt = 0.01. Time histories of the energy components according to the fourth-order Runge-Kutta and third-order Leapfrog algorithms for dt = 0.002 are shown in the following Figure.

With a particle mass of unity, m = 1, the initial kinetic energy of the system is $\sum mv^2/2 = 16$. For simplicity, let all pairs of particles interact with a simple short-ranged pair potential ϕ whenever the separation of the pair is less than unity :

$$\phi(r < 1) = 100(1 - r^2)^4$$
.

In the original configuration the total potential energy vanishes so that the total energy—a constant of the motion—is $E=K+\Phi=16$.

Once the particles start to move, all interacting pairs with r < 1 will experience a repulsive force $F(r < 1) = -\nabla \phi$:

$$F(r < 1) = a(r) = 800r(1 - r^2)^3$$
.

Physical Ideas Underlying SPAM

We eliminate the need for special boundary forces by imposing *periodic* boundary conditions on the 4×4 box in which the motion takes place :

$$x < -2 \rightarrow x = x + 4$$
; $x > +2 \rightarrow x = x - 4$;
 $y < -2 \rightarrow y = y + 4$; $y > +2 \rightarrow y = y - 4$.

In calculating the forces between Particle *i* and Particle *j* these same periodic boundaries require that the smallest (in absolute value) of the three possible separations, $r_{ij} = (x_{ij}, y_{ij})$, in both the *x* and the *y* directions :

$$x_{ij} = x_i - x_j \pm L \text{ or } x_{ij} = x_i - x_j ;$$
$$y_{ij} = y_i - y_j \pm L \text{ or } y_{ij} = y_i - y_j ,$$

be used in calculating the "nearest-image" forces between all interacting pairs of particles. This nearest-image distance is calculated by going through the sequence of steps :

$$x_{ij} < -2 \to x_{ij} = x_{ij} + 4 ; \ x_{ij} > +2 \to x_{ij} = x_{ij} - 4 ;$$
$$y_{ij} < -2 \to y_{ij} = y_{ij} + 4 ; \ y_{ij} > +2 \to y_{ij} = y_{ij} - 4 .$$

With the initial and boundary conditions in place, the ensuing dynamics of this system requires the solution of 64 first-order ordinary differential equations :

$$\{ \dot{r}_i = v_i \}; \{ \dot{v}_i = \sum_j 800r_{ij}(1 - r_{ij}^2)^3 \},\$$

or, equivalently, 32 second-order differential equations :

{
$$\ddot{r}_i = \dot{v}_i = \sum_j 800 r_{ij} (1 - r_{ij}^2)^3$$
 }

where the differential equations for each vector r_i and each vector v_i correspond to pairs of equations for the x and y components of these vectors.

Figure 1.7 shows the time histories of the total, kinetic, and potential energies for two solutions, Runge-Kutta and Leapfrog. In both cases the trajectories were reversed after 12,000 timesteps, with dt = 0.002. In the Runge-Kutta case the energy plot is accurate to five digits at t = 12. It still appears to be perfectly symmetric about the reversal time (24) even though the final configuration, with a total time of 24+24, shows coordinate errors and potential/kinetic energy errors of order 0.1. The apparent symmetry

Example: A Molecular Dynamics Simulation

is illusory. The errors in the Runge-Kutta algorithm are only small, and are *not* time-reversible.



In the Leapfrog case the energy and coordinate errors are much larger, with readily visible discrepancies appearing between simulations with dt values of either 0.001 and 0.002 at t=12, the same time at which the leapfrog trajectories deviate visibly from the more-nearly-accurate Runge-Kutta solutions. The leapfrog trajectory reversibility is still nearly perfect, despite its relatively large trajectory errors. The lack of precise time reversibility for the leapfrog algorithm is due only to computational rounding errors. 19,20

The energy errors from either integration scheme can be estimated directly. The classic fourth-order Runge-Kutta scheme, with a timestep

ws-book9x6

¹⁹Levesque and Verlet (1993).

 $^{^{20}}$ Kum and Hoover (1994).

Physical Ideas Underlying SPAM

dt = 0.002 would ideally incur an error per timestep of the same order as the double-precision computer roundoff error ,

error
$$\simeq (d^5 r/dt^5) dt^5/120 \simeq 10^{-15}$$
.

But the forcelaw discontinuity (in the third derivative), which comes into play at the beginning and end of every collision, gives a somewhat larger coordinate error Δr , of order

$$F \simeq 6400 (\delta r)^3 \longrightarrow \Delta r \simeq \frac{1}{2} 6400 (vdt)^3 dt^2 \simeq 10^{-10} \; .$$

These single-step integration errors, though small, are amplified exponentially fast by the chaotic Lyapunov-unstable many-body dynamics. A comprehensive study of the chaotic dynamics shows that small perturbations grow roughly as $e^{\lambda t}$, where the largest Lyapunov exponent²¹ λ is about 3. Thus a relatively short Runge-Kutta simulation of just 6000 timesteps $(t \rightarrow 12)$ is already long enough that the reversed trajectory visibly fails to reproduce the reversed initial configuration.

It is amusing that a much less accurate simulation, a "bit-reversible" leapfrog algorithm described by Levesque and Verlet,

$$r(t+dt) - 2r(t) + r(t-dt) \equiv [(dt)^2 a(t)]$$
,

where all the coordinates, as well as the combination $(dt)^2 a(t)$ are restricted to be single-precision *integers*, regains the initial condition *perfectly*, without any roundoff error at all, despite the inaccuracy of the corresponding trajectory. We will revist the relatively subtle subject of errors in numerical integration in Chapter 6.

 $^{^{21}}$ Hoover and Posch (1995).

References

1.10 References

- W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes, The Art of Scientific Computing* (Cambridge University Press, London, 1986).
- (2) R. A. Gingold and J. J. Monaghan, "Smoothed Particle Hydrodynamics: Theory and Application to Nonspherical Stars", Monthly Notices of the Royal Astronomical Society 181, 375-389 (1977).
- (3) L. B. Lucy, "A Numerical Approach to the Testing of the Fission Hypothesis", The Astronomical Journal **82**, 1013-1024 (1977).
- (4) J. Gleick, Chaos, Making a New Science (Viking, New York, 1989).
- (5) Wm. G. Hoover, Computational Statistical Mechanics (Elsevier, New York, 1991). [Keisan Toukei Rikigaku, Japanese translation by Koichiro Shida with the supervision of Susumu Kotake (Morikita Shupan, 1999)].
- (6) R. Preston, "Capturing the Unicorn", New Yorker Magazine, 11 April 2005.
- (7) J. B. Gibson, A. N. Goland, M. Milgram, and G. H. Vineyard, "Dynamics of Radiation Damage", Physical Review 120, 1229-1253 (1960).
- (8) B. J. Alder and T. E. Wainwright, "Molecular Dynamics by Electronic Computers", pages 97-131 in *Transport Processes in Statistical Mechanics*, I. Prigogine, Editor (Interscience, New York, 1958).
- (9) The "Fermi-Pasta-Ulam" problem is reviewed and extended in J. L. Tuck and M. T. Menzel, "The Superperiod of the Nonlinear Weighted String Problem", Advances in Mathematics 9, 399-407 (1972).
- (10) Wm. G. Hoover, "Mécanique de Nonéquilibre à la Californienne", Physica A 240, 1-11 (1997).
- (11) S. Nosé, "Constant Temperature Molecular Dynamics Methods", Progress in Theoretical Physics Supplement 103, 1-117 (1991).
- (12) Wm. G. Hoover, "Liouville's Theorems, Gibbs' Entropy, and Multifractal Distributions for Nonequilibrium Steady States", Journal of Chemical Physics **109**, 4164-4170 (1998).
- (13) J. W. Gibbs, *Elementary Principles in Statistical Mechanics* (Oxbow Press, 1991) [first published in 1902].
- (14) C. Cercignani, Ludwig Boltzmann, the Man Who Trusted Atoms (Oxford University Press, 1998).
- (15) B. Moran, W. G. Hoover, and S. Bestiale, "Diffusion in a Periodic Lorentz Gas", Journal of Statistical Physics 48, 709-726 (1987).
- (16) Wm. G. Hoover and C. G. Hoover, "Searching for Auxetics with Dyna3d and ParaDyn", Physica Status Solidi 242b, 585-594 (2005).

Physical Ideas Underlying SPAM

- (17) C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation* (McGraw-Hill, New York, 1985).
- (18) T. J. Ahrens, T. Takata, J. D. O'Keefe, and G. S. Orton, "Impact of Comet Shoemaker-Levy 9 on Jupiter", Geophysical Research Letters 21, 1087-1090 (1994).
- (19) D. Levesque and L. Verlet, "Molecular Dynamics and Time Reversibility", Journal of Statistical Physics 72, 519-537 (1993).
- (20) O. Kum and W. G. Hoover, "Time-Reversible Continuum Mechanics", Journal of Statistical Physics 76, 1075-1081 (1994).
- (21) Wm. G. Hoover and H. A. Posch, "Shear Viscosity via Local Control of Spatiotemporal Chaos in Two-Dimensional Isoenergetic Dense Fluids", Physical Review E 51, 273-279 (1995).

ws-book9x6

Chapter 2

Continuum Mechanics

/ Summary and Scope of Continuum Mechanics / Evolution Equations for Fluids and Solids / Initial and Boundary Conditions / Constitutive Equations for Equilibrium and Nonequilibrium Fluids / Artificial Viscosity and Conductivity / Constitutive Equations for Elastic Solids / Constitutive Equations for Plastic Flow / Algorithm for Plastic Flow /

/ 10 Figures /

Example Problems : [Heat Conduction, Sound Propagation, Rayleigh-Bénard Flow]

2.1 Summary and Scope of Continuum Mechanics

Continuum mechanics, like particle mechanics, quantum mechanics, and other approximate physical descriptions of nature, is a mathematical "model". This model is a set of partial differential equations for the temporal evolution of mass, momentum, and energy flows in a spatial continuum with assumed constitutive relations, initial conditions, and boundary conditions. By generating approximate numerical solutions of these partial differential equations, we seek to approximate flows of "real matter" in space and time. A numerical implementation of continuum mechanics is a useful one to the extent that the agreement is good and that the effort required to program and execute the programs to obtain solutions is not excessive.

Because any physics model is intrinsically incomplete there is always the possibility that an implementation will produce faulty conclusions. There

ws-book9x6

Continuum Mechanics

is always certainty too that a model *can* be applied outside its range of usefulness. Continuum mechanics is at its best in describing relatively simple materials which are homogeneous and isotropic on the scale of observation. Continuum mechanics is *not* able to deal well with the intrinsic complexity of the atomic nanoscale, or even the mesoscopic microscale, as revealed in photomicrographs of ductile fracture surfaces, or zoological specimens, or in systems, such as turbulent flows, in which a wide range of length and time scales must be considered.

Computer algorithms must first of all *discretize* the continuum so as to reduce the number of degrees of freedom from infinity to a manageable number. In most cases this discretization requires a mesh of points on which and within which the field variables are evolved in time. In a well-posed continuum model the differential conservation equations and constitutive relations dictate the flows, with the constitutive relations providing the comoving fluxes of momentum and energy, $P = -\sigma$ and Q respectively, which augment the convective fluxes of mass, momentum, and energy.

Useful algorithms designed to provide approximate solutions to continuum problems give designers and engineers a versatile approach to understanding the response of structures to static and dynamic loads. Ultimately a faithful response must include failure mechanisms—flow, or fracture, or both. Failing structures reveal that continuum mechanics is incomplete. The microscopic flaws governing failure are not to be found in a continuum description. This weakness, once exposed, can be corrected, on a case-by-case basis, by adding phenomenological failure models to the basic conservation and constitutive equations. A large part of modern research in continuum mechanics is devoted to the discovery and refinement of such models.

We discuss here equilibrium fluid and solid constitutive equations, and their extensions to nonequilibrium situations. The constitutive equations for P and Q are required to solve the evolution equations for the density, velocity, and energy. The basic "control-volume" approach of Chapter 1, based on local conservation of mass, momentum, and energy, gives rise to the partial differential evolution equations for ($\dot{\rho}, \dot{v}, \dot{e}$) detailed in the next Section. In addition to the constitutive equations, initial and boundary conditions are also required for a complete formulation of a continuum simulation. A difficult, significant, and active research area is the efficient treatment of material boundaries, particularly those separating materials undergoing relative motion.

We also point out here some of the numerical difficulties which, when

Evolution Equations for Fluids and Solids

ignored, can lead to instabilities. These difficulties require specialized computational approaches to continuum simulations. We describe the artificial viscosity and conductivity used to stabilize flows. We describe also the simplest algorithm for simulating irreversible solid-phase "plastic flow". Three simple example problems, the propagation of heat and sound in one dimension, and the simultaneous convective and conductive flows of mass, momentum, and energy in two dimensions, complete this introductory survey of continuum techniques. Finally, the constitutive equations and the evolution equations described here are written in a form useful for smooth particle simulations. The simulations themselves are illustrated in the examples provided in later Chapters of this book.

2.2 Evolution Equations for Fluids and Solids

The conservation equations describing the flows of mass, momentum, and energy are *exactly the same* for fluids and solids. Once it is assumed that the fluxes (flows per unit area and time) are known the local time rates of change of mass, momentum, and energy follow from the conservation laws. Mass flux is simple, ρv , for any material of density ρ and velocity v. The comoving momentum and heat fluxes P and Q are more complicated, and vary from one material to another as described by the materials' mechanical and thermal "constitutive relations". In this section we consider the simpler problem, computing time rates of change from known fluxes.

To predict the future, the current state of a continuum must be defined everywhere. The simplest reasonable description, based on conservation of mass, momentum, and energy, uses the mass density, the velocity, and energy per unit mass as *dependent* variables. These variables depend upon both place and time. In Chapter 1 we saw that the flows of mass and momentum into a volume element can be used to derive Lagrangian forms of the continuity equation for $\dot{\rho}$ and the "equation of motion" for \dot{v} :

$$\dot{\rho} = -\rho \nabla \cdot v \; ; \; \dot{v} = -(\nabla \cdot P)/\rho \equiv (\nabla \cdot \sigma)/\rho \; .$$

Motion under the influence of a gravitational field g(r) adds an additional acceleration to the equation of motion :

$$\dot{v} = g - (\nabla \cdot P) / \rho \; .$$

In static problems, particularly those involving materials under tension (from which the word "tensor" derives), it is more natural to use the "stress

ws-book9x6

Continuum Mechanics

tensor" $\sigma \equiv -P$ with the corresponding equation of motion :

$$\dot{v} = g + (\nabla \cdot \sigma) / \rho$$
.

Continuum mechanics provides no information on the pressure tensor P beyond assuming those properties necessary for stability. To avoid divergent rotational accelerations it is necessary that P be a symmetric tensor (so that $P_{ij} = P_{ji}$). To avoid divergent volume fluctuations the derivative $dP/d\rho$ must be positive. To avoid divergent shape and temperature fluctuations, the shear modulus and thermal diffusivity must likewise be positive. It is the rôle of the simulator to specify the dependence of the pressure tensor (and the heat flux vector) on past and present values of the material properties ρ , v, and e.

Energy changes described by thermodynamics include the production and exchange of heat as well as the performance of mechanical work. Heat exchange and work together account for the total energy change in any system undergoing both types of processes. This energy-conservation principle has the form given by the First Law of Thermodynamics :

$$d\dot{E} \equiv d\dot{E}_Q - d\dot{E}_W \; ,$$

where the changes in system energy E are classified as heat taken in by the system E_Q or work performed by the system on the surroundings E_W . Here we change the notation of Section 1.4 ($Q \rightarrow E_Q$; $W \rightarrow E_W$) in order to avoid confusion between absorbed heat and the heat flux Q discussed next.

Here we introduce the heat flux vector Q to represent the "comoving" (in the Lagrangian frame of the moving material) flow of energy per unit time and area. With this definition the usual continuum assumptions of continuous time and space derivatives result in the Eulerian form of the energy equation :

$$\frac{\partial [\rho(e+\frac{v^2}{2})]}{\partial t} = -\nabla \cdot [Q + \rho v(e+\frac{v^2}{2})] - \nabla \cdot (v \cdot P) \ .$$

The equation of motion and continuity equation can be used to simplify the energy conservation law to give a comoving Lagrangian energy equation consistent with, and equivalent to, the First Law of Thermodynamics :

$$\rho \dot{e} = \rho \left(\frac{\partial e}{\partial t} + v \cdot \nabla e \right) = \left[-\nabla \cdot Q \right] - \left(\nabla v \right) : P \; .$$

Evolution Equations for Fluids and Solids

The work term $-(\nabla v): P \equiv \sum (-\nabla v)_{ij} P_{ij}$ includes four separate contributions in two dimensions, and nine in three dimensions. Fluid viscosity (Section 2.5) and elastic stress (Section 2.7) are the two simplest constitutive models for the tensor deviations of P from the hydrostatic pressure.

The heat flux Q needs also to be specified by the simulator. In a (hypothetical) material obeying Fourier's Law of heat conduction, $Q = -\kappa \nabla T$, where T is temperature, Q is simply proportional to the temperature gradient. The local temperature is obtained from density and energy through the constitutive equations. In the case of a motionless ideal gas, temperature and energy are both proportional to the mean-squared particle velocity, $T \propto \sum mv^2$. In such a gas the heat flux is given locally by the sum of $mv\frac{v^2}{2}/V$. The sums include all those particles in the (sufficiently small for locality, and sufficiently large to contain many particles) volume V. The heat flux in such a volume would vanish at equilibrium with positive and negative contributions cancelling exactly. For gases, kinetic theory provides an exact recipe linking the conductivity to the interparticle forces.¹

With assumed constitutive relations specifying the fluxes of momentum and energy we can evaluate all the gradients on the righthandsides of the conservation equations for $(\dot{\rho}, \dot{v}, \dot{e})$. This gives us the time evolution of (ρ, v, e) as well as the vector Q and the deviatoric components of the pressure tensor, $P - P_{eq}I$, where I is the unit tensor. Two additional ingredients are required. Evidently *initial conditions* must be specified throughout the system. If, as is usual, the system of interest is also in thermal or mechanical contact with the "outside world" (the "surroundings") it is also necessary to specify *boundary conditions* describing those interactions. Often the boundary itself has an explicit time-dependent motion. The geometric description of a complicated system with a finite-element mesh and the implementation of appropriate boundary conditions are usually the most time-consuming parts of formulating a continuum simulation. They are also the most challenging parts of formulating computer algorithms.

Figure 2.1 shows four stages in the evolution of a "relatively simple" continuum problem, the production and subsequent flow of an $(L \times L)$ square of fluid. We begin with a stress-free rectangular specimen of dense fluid $(L \times \frac{8}{5}L)$ at a density of unity, and add gravity, with the field chosen to give a square $(L \times \frac{8}{5}L) \longrightarrow (L \times L)$. The equilibrated square has a density $\rho = 2$ at its base and the stress-free density $\rho = 1$ at its top. Following equilibration, release of the lateral boundary constraints then

¹S. Chapman and T. G. Cowling, The Mathematical Theory of Nonuniform Gases.

August 16, 2006 17:44

Continuum Mechanics

allows the square specimen to collapse. The solution shown in the Figure was obtained using the Lagrangian continuum code ParaDyn.² During the initial "equilibration phase", three frictionless "stonewall" boundary conditions ,

$$|x| < \frac{L}{2}$$
; $y > 0$ with $P = \rho^3 - \rho^2$ and $g = \frac{5}{2L}$,

(on the left and right and at the bottom) were applied. During equilibration the column shrinks :

$$\left[\begin{array}{c} \ddot{y}_{\mathrm{gravity}} \rightarrow -g \end{array} \right] \longrightarrow \left[\begin{array}{c} (L \times \frac{8}{5}L) \rightarrow (L \times L) \end{array} \right]$$
 .



control (Figure 7.5) were used in the collapse phase shown here though the solution is hardly changed when both these controls are omitted.

²Hoover, DeGroot, and Sherwood, *ParaDyn Manual* (2004).

Initial and Boundary Conditions

During equilibration the height of the column decreases, from $\frac{8}{5}L$ (with g = 0) to L (with $g = \frac{5}{2L}$) in this example. With the adiabatic equation of state, $P = \rho^3 - \rho^2$ the force-balance condition :

$$dP/dy = (dP/d\rho)(d\rho/dy) = (3\rho^2 - 2\rho)d\rho/dy = -\rho g = -\frac{5\rho}{2L}$$
,

has a simple solution for 0 < y/L < 1:

$$\begin{split} \int_{2}^{\rho} (3\rho'-2)d\rho' &= -\frac{5}{2} \int_{0}^{y/L} d\frac{y'}{L} = -\frac{5}{2} (\frac{y}{L}) \ - \\ &\frac{3}{2}\rho^{2} - 2\rho - 2 = -\frac{5}{2} (\frac{y}{L}) \longrightarrow \\ &\rho = \frac{2}{3} + \frac{1}{3} \sqrt{16 - 15(y/L)} \ . \end{split}$$

Notice that the integration of the force-balance condition begins at the bottom of the column (y=0 ; $\rho=2$) .

After equilibration, the left and right conditions at $x = \pm L/2$ are instantaneously removed to allow the subsequent collapse of the column. During the collapse phase the tensile waves propagating in from the newly-free lateral surfaces produce unstable cavitation (the void shown in the Figure). The evolution of the collapse following cavitation is a difficult problem for continuum mechanics and a challenging research area.

2.3 Initial and Boundary Conditions

Properly chosen initial conditions can reduce computational expense. If the desired solution is stationary, convergence can be accelerated by using a coarse-mesh stationary solution as the initial condition for a more elaborate fine-mesh simulation. A mesh for an irregularly-shaped object can be generated by choosing points initially distributed according to any convenient scheme. There is a disadvantage to unnecessarily thin zones. Thus the so-called "Courant condition" for stability requires that the timestep dt be no greater than that corresponding to the thickness of the narrowest element, $dt < dx_{\min}/c$, where c is the sound velocity. A nearest-neighbor algorithm which ignores information coming from farther away during a single timestep is obviously unstable. Accordingly, it is wasteful to use a mesh with unnecessarily thin elements.

Continuum Mechanics

A poor mesh with unnecessarily small elements can be improved by *relaxation*. As an example, consider a set of randomly-placed points or points with randomly-chosen velocities, occupying a portion of a regular lattice within an irregular boundary. Then allow all the mesh points to move according to damped equations of motion :

$$m\ddot{r} = F - \left(\frac{m\dot{r}}{\tau}\right) ,$$

where F is a smooth short-ranged pair potential. Point motions can additionally be constrained by elastic collisions whenever a boundary is reached. Figure 2.2 shows three hundred points, initially chosen motionless at random locations within the space defined by two circular boundaries :

$$x^{2} + y^{2} < 10$$
; $(x - 2)^{2} + (y - 2)^{2} > 4$.



Figure 2.2: Initial and final configurations of 300 particles moving between two circular boundaries in response to damped equations of motion and subject to pairwise-additive repulsive forces from the pair core potential $\phi(r < 1) = (1 - r^2)^4$. 1000 timesteps were used, with $dt = 1/\tau = 0.1$.

The subsequent equilibrating motion according to the specially smooth repulsive pair potential,

$$\phi(r < 1) = (1 - r^2)^4$$
,

was damped with $1/\tau = dt = 0.1$ to provide the fully-relaxed unstructured mesh shown in the Figure.

Initial and Boundary Conditions

Many of the applications of continuum mechanics have been, and are, devoted to astronomical and astrophysical problems in which boundary conditions were not important. But continuum problems involving surfaces penetration, fracture, or heat transfer for example—require algorithmic implementations of realistic boundary conditions. A simple problem, the formation of convective rolls due to a temperature gradient in a gravitational field (the "Rayleigh-Bénard" example problem, treated at this Chapter's end), is prototypical. Temperature and velocity have specified boundary values on a box containing the fluid under investigation. A "good" boundary algorithm is one which minimizes the dependence of the computed results on the number of degrees of freedom used to describe the problem.

Mass, momentum, and energy can enter or exit at system boundaries. Likewise, the location, velocity, and temperature of particular boundaries can be specified. A rigid boundary, even if stationary, can act as a source of momentum, either linear or angular, as well as energy, in the form of heat. The simplest boundary is the free surface of an isolated system subject to no external forces.

Once the boundary conditions are added to the initial conditions and constitutive relations we expect that any well-posed problem has a definite solution. Of course, the existence of a "definite solution" becomes a bit problematic (meaningless in fact) for *chaotic* systems, with "sensitive dependence" on the initial conditions, ($\propto e^{+\lambda t}$). In such a case one can only ask for a "reasonable" solution, as is detailed in Sections 6.7 and 6.8.

Within the system the fluxes can be tracked by applying the constitutive relations to the evolving state variables. The Eulerian fluxes include not only the *comoving* P and Q but also the *convective* contributions proportional to ρv . The total fluxes of mass, momentum, and energy are just those used in deriving the evolution equations from the conservation laws:

$$\{ \rho v , P + \rho v v , Q + \rho v [e + \frac{v^2}{2}] + v \cdot P \}$$

or

$$\{ \ 0 \ , \ P \ , \ Q \ \}$$

for Eulerian or Lagrangian simulations, respectively.

ws-book9x6

 $Continuum \ Mechanics$

2.4 Constitutive Equations for Equilibrium Fluids

The purely mechanical equation of state,

$$P = B_0 \left(\frac{\rho^3}{\rho_0^3} - \frac{\rho^2}{\rho_0^2} \right) \;,$$

is an adequate model for adiabatic flows without thermal effects. The free parameters ρ_0 [the stress-free density] and B_0 [the stress-free bulk modulus] allow the density and sound velocity to be given any desired values. Fluid flows driven by pressure differences and gravity can then be modeled, as in the collapsing-column example of Section 2.2.

Equilibrium thermodynamics describes materials through their thermal and mechanical equations of state. The dependence of the energy on temperature and density is described by the "thermal equation of state", E = NDkT/2, for the simplest example, an N-body ideal gas in D dimensions. The dependence of the pressure on temperature and density, or energy and density, defines the "mechanical equation of state",

$$PV = NkT = (2/D)E ,$$

for the same *D*-dimensional ideal-gas example.

A bare-bones equation of state must describe both a heat capacity dQ/dT > 0 and a compressibility $-(1/V)(\partial V/\partial P) > 0$. Here dQ is the heat required by the temperature change dT and the compressibility describes the response of the system volume when the system performs work dW = PdV on its surroundings. A more complicated constitutive model could incorporate higher derivatives of energy and pressure with respect to temperature and volume, as does van der Waals' closed-form two-parameter equation of state :

$$P = \frac{NkT}{V - Nb} - \frac{N^2a}{V^2} ; E = \frac{DNkT}{2} - \frac{N^2a}{V} .$$

This equation of state was developed by van der Waals so as to describe both the liquid and gas phases with a simple analytic form. Densities near the maximum, $Nb/V \simeq 1$, with temperatures less than the critical temperature, $kT_C = 8a/27b$ correspond to dense-fluid "liquid" states. Low densities correspond to the gas phase. Over most of the density-temperature diagram shown in the Figure 2.3 the homogeneous liquid and gas phases are thermodynamically less stable than is a two-phase mixture. Those parts of

Constitutive Equations for Equilibrium Fluids

the density-temperature plane in which the higher-density liquid and the lower-density gas can stably coexist are shaded in the Figure.



particle. The density and temperature are both given in "reduced units" here, relative to their values at the "critical point". The critical point is the highest-temperature state point at which two fluid phases can still be distinguished : $(V_C/N) = 3b$; $kT_C = 8a/27b$.

The Figure was constructed by minimizing a difference metric, $\Delta P^2 + \Delta (G/N)^2$, based on the pressures and Gibbs' free energies G = E + PV - TS of high/low pairs of density points lying on an isotherm. Isotherm points equally spaced in density were used. The higher (than the pressure minimum) density point and the lower (than the pressure maximum) density point which minimize the metric simultaneously most closely satisfy the

ws-book9x6

$Continuum \ Mechanics$

joint conditions of mechanical and thermodynamic equilibrium :

$$P_{\text{gas}} = P_{\text{liquid}}; \ G_{\text{gas}} = G_{\text{liquid}}.$$

In this specially simple case temperature can be eliminated from the mechanical and thermal equations of state to express pressure directly in terms of energy. Begin with the zero-temperature relations ,

$$E_0 = -\frac{N^2 a}{V} ; P_0 = -\frac{dE_0}{dV} = \frac{E_0}{V}$$

If the "thermal contributions" to the energy and pressure are defined relative to the zero-temperature functions, the resulting equation of state has the specially simple (and useful) "Grüneisen form", with $P_T \propto E_T$:

$$P_T = \frac{2E_T}{D(V - Nb)} \quad \longleftarrow \quad P_T = P - P_0 \ ; \ E_T = E - E_0 \ .$$

A general thermodynamic restriction (one of the Maxwell relations) links the thermal and mechanical equations of state :

$$d(A/T) = -(P/T)dV - (E/T^2)dT \rightarrow$$

$$\frac{\partial^2 (A/T)}{\partial T \partial V} = -\frac{\partial (P/T)_V}{\partial T} = -\frac{\partial (E/T^2)_T}{\partial V} \rightarrow$$
$$\frac{\partial}{\partial T} (P/T)_V = \frac{\partial}{\partial V} (E/T^2)_T \rightarrow$$
$$P - T (\partial P/\partial T)_V = -(\partial E/\partial V)_T .$$

holtz' free energy
$$A - E - TS$$
 where S is the

A is Helmholtz' free energy, A = E - TS, where S is the thermodynamic entropy. Additionally it is necessary that the bulk modulus and heat capacity both be positive,

$$B = -V \frac{dP}{dV} > 0 \; ; \; C = \frac{dE_Q}{dT} > 0 \; .$$

Apart from these consistency and stability conditions the equilibrium equation of state is quite arbitrary. Large-scale computer codes have dozens of equation of state forms available for hundreds of materials.³

³Young, Phase Diagrams of the Elements (1991).

Constitutive Relations for Nonequilibrium Fluids

2.5 Constitutive Relations for Nonequilibrium Fluids

To describe the behavior of materials in motion the equilibrium constitutive equations need to be generalized in order to take dissipative nonequilibrium effects—viscosity, conductivity, plasticity, ...—into account. The simplest nonequilibrium formulations assume linear relations between fluxes and gradients, with the heat flux given by Fourier's heat-flow law and the momentum flux described by Newton's linear viscosity. Fourier's law is $Q = -\kappa \nabla T$. Newton's formulation of shear stress is linear in the corresponding strain rate ,

$$\sigma_{xy} = \eta \left[\left(\frac{\partial v_x}{\partial y} \right) + \left(\frac{\partial v_y}{\partial x} \right) \right].$$

The linear symmetric tensor generalization of viscous stress is the twoparameter constitutive law :

$$\sigma = \sigma_{\rm eq} \mathbf{I} + \lambda \mathbf{I} \nabla \cdot v + \eta [\nabla v + \nabla v^t] ,$$

in *D* spatial dimensions. Here the superior ^t indicates the transpose and I is the "unit tensor", with *D* diagonal elements of unity and all the D(D-1)off-diagonal elements equal to zero. The two viscosity coefficients correspond to the two fundamental modes of deformation fluids can sustain, compression and shear. In any isochoric flow (where $\nabla \cdot v$ vanishes) the extra viscous stress depends only on the *shear* viscosity η .

In a deformation including a volume change, with $\nabla \cdot v \neq 0$, the extra hydrostatic stress associated with the fast expansion or compression rate in two space dimensions,

$$\dot{\epsilon}_{xx} + \dot{\epsilon}_{yy} \equiv \nabla \cdot v = \frac{\dot{V}}{V} \rightarrow \frac{\sigma_{xx} + \sigma_{yy}}{2} - \sigma_{\rm eq} ,$$

is

$$\frac{\sigma_{xx} + \sigma_{yy}}{2} - \sigma_{eq} = \eta \frac{+\dot{V}}{V} + \lambda \frac{+\dot{V}}{V} \equiv (\eta + \lambda)\nabla \cdot v = \eta_v \frac{+\dot{V}}{V}$$

so that the bulk viscosity (in two dimensions) is $\eta_v = \eta + \lambda$. An exactly similar development gives the analog,

$$\frac{\sigma_{xx} + \sigma_{yy} + \sigma_{zz}}{3} - \sigma_{\rm eq} = \eta_v \frac{+\dot{V}}{V} \ ; \ \eta_v = \frac{2}{3}\eta + \lambda \ ,$$

Continuum Mechanics

in three space dimensions. For a "Stokes'-fluid" model lacking bulk viscosity λ must have the special value $-\eta$ in two dimensions (and $-\frac{2}{3}\eta$ in three dimensions).

The transport coefficients (such as the bulk and shear viscosity and thermal diffusivity) in a continuum simulation can all be state dependent. They can also depend on the magnitude of the deviation from equilibrium. If the deviations are sufficiently small, the phenomenological linear laws can be derived from statistical mechanics and the transport coefficients related to equilibrium fluctuations. To go beyond the linear laws is neither straightforward nor particularly fruitful. The *forms* of these same linear laws for momentum and energy transport *can* be exploited to combat numerical instabilities with "artificial" viscosities and conductivities. The instabilities can otherwise prove fatal to numerical simulations. We turn next to these artificial transport coefficients.

2.6 Artificial Viscosity and Conductivity

Because the sound speed $c = \sqrt{(\partial P/\partial \rho)_S}$ is typically an increasing function of density there is a tendency for the faster high-density high-pressure parts of a pressure wave to overtake the slower lower-density lower-pressure regions, thus forming localized shockwaves. A shockwave can be prevented from becoming infinitely steep by viscous and thermal dissipative forces. Ordinary shear and bulk viscosities and thermal conductivity describe dissipation on an atomic scale. The net result is that viscosity and conductivity spread the pressure increase over a relatively tiny distance on the order of an atomistic mean free path.

In any continuum simulation, with element sizes (or computational zone sizes) which are macroscopic in size, the atomistic dissipation described by ordinary microscopic viscosity and heat conductivity is negligibly small. With the much coarser macroscopic spatial resolution of numerical continuum simulations the tendency toward shock formation must be combatted with a much larger completely nonphysical "artificial" viscosity, causing the compressive pressure wave to be spread over a macroscopic distance of at least a few mesh or particle spacings. von Neumann and Richtmyer developed such an "artificial viscosity" so as to smooth the structure of shockwave discontinuities into manageable differentiable forms. The width of the shockwave is artificially increased, using their scheme, from atomic to macroscopic dimensions without affecting the overall conservation laws.

Artificial Viscosity and Conductivity

We will see the need for the artificial transport coefficients in the example problems which follow.

Somewhat similar ideas apply to heat transfer. Because conductivity is typically a strongly increasing function of temperature, the hotter parts of a "heat wave" have a tendency to overtake the colder ones. But because the flow of heat is so slow (diffusive, rather than ballistic) with respect to sound, it is usual to ignore this tendency unless the flow is turbulent.

von Neumann's and Richtmyer's numerical recipe for artificial viscosity⁴ avoids the instabilities associated with large pressure gradients without affecting the equilibrium part of the constitutive relations. Although the spatial scale of pressure waves is increased, the magnitude of the pressure jump is not affected. To understand this idea in more detail consider the steep wave shape generated by a steady compression. Figure 2.4 shows such a steady process.



Figure 2.4: Schematic structure of a compression wave, moving at the shock velocity u_s , which accelerates a fluid to match the driving piston velocity u_p . With artificial viscosity $\eta_{\rm vNR}$, the momentum flux emanating at the piston exceeds the cold fluid flux by $\rho_0 u_s u_p \simeq \eta_{\rm vNR} u_p/dx$.

The piston at the left compresses an initially motionless fluid with density ρ_0 to the greater density ρ . The piston also accelerates the deforming fluid, imparting a final momentum density ρu_p to a fluid volume (per unit area) $u_s t$ during the time t. Evidently the momentum flux P_{xx} in the accelerated fluid is $\rho_0 u_s u_p$. Near the center of a strong shockwave most of this momentum flux is viscous, corresponding to a viscosity coefficient :

$$\eta_{\rm shock} = -P_{xx}/\dot{\epsilon}_{xx} \simeq +P_{xx}/(u_p/dx) =$$

$$\frac{\rho_0 u_s u_p}{u_p/dx} = \rho_0 u_s dx \simeq P_{xx} dx/u_p = \rho_0 c dx ,$$

 $^{^{4}}$ von Neumann and Richtmyer (1950).

ws-book9x6

Continuum Mechanics

(c is the soundspeed) so that the shockwidth dx is proportional to the viscosity.

Consider a practical example. A pressure of 10 kilobars $(10^{10} \text{ ergs/cm}^3)$ in a fluid with a sound velocity c of 10^6 cm/sec and a mass density ρ_0 of 1 gm/cm³ requires a piston velocity u_p of 10^4 cm/sec . Using the viscosity of water, $10^{-2} \text{ gm/(cmsec)}$, gives an estimate for the corresponding shockwidth :

$$dx = \eta \times u_p / P_{xx} = [10^{-2} \text{gm/cmsec}] \times [10^4 \text{cm/sec}] / [10^{10} \text{gm/cmsec}^2]$$
$$\rightarrow dx = 10^{-8} \text{cm} ,$$

a truly microscopic length, too short for any macroscopic simulation to capture.

Microscopic molecular dynamics simulations, with atomistic models of fluids, have shown that this estimate is accurate,⁵ and that shockwidths in condensed matter actually do correspond to the size of an atom. For fluids or solids undergoing compression, von Neumann and Richtmyer's idea was artificially to increase the viscosity in order to increase the shock width to the computational mesh size :

$$\nabla \cdot v < 0 \longrightarrow \eta_{\rm vNR} \simeq \rho_0 c dx$$
.

The *kinematic* viscosity η/ρ is dimensionally a diffusion coefficient (with units [length]²/[time]) giving the diffusive rate of shear momentum. Exactly similar ideas can be applied to the transfer of heat.

2.7 Constitutive Relations for Elastic Solids

Unlike fluids, solids resist shear. Solids can support internal "locked-in" shear stresses on the order of kilobars [1000 atmospheres $\simeq 10^8$ pascals $= 10^8 \text{kg/m(sec)}^2$], even in the absence of external loads, so that the stress tensor within a motionless field-free solid need not be isotropic. Figure 2.5 illustrates an equilibrated elastic solid, composed initially of regular squares of compressed ($\rho/\rho_0 = 1.2$) and expanded ($\rho/\rho_0 = 0.8$) material. In a material which can flow, the small deformations resulting from the locked-in stresses in a macroscopically homogeneous elastic solid are of the order of the ratio of the plastic yield stress to the shear modulus, usually less than a percent.

⁵Holian, Hoover, Moran, and Straub (1987).



Constitutive Relations for Elastic Solids

Generally, small stresses can be described as resulting from the linear strains { $\epsilon_{xx},\epsilon_{xy},\epsilon_{yy}$ } :

$$\epsilon_{xx} = \frac{du_x}{dx} ; \ \epsilon_{xy} = \frac{du_x}{dy} + \frac{du_y}{dx} ; \ \epsilon_{yy} = \frac{du_y}{dy} ;$$

where the displacement vector $u = (u_x, u_y)$ is measured relative to a nearby reference configuration. In the simplest case, with both stress and strain vanishing in the reference configuration, a "linear" solid, obeying Hooke's Law, can be described in terms of two independent elastic constants, one for compression and one for shear. These fundamental constants, the bulk modulus *B* and the shear modulus *G* can alternatively be expressed in terms

ws-book9x6

Continuum Mechanics

of the two Lamé constants (λ and η) or in terms of Young's (longitudinal) modulus E and Poisson's ratio (of transverse contraction to longitudinal extension) ν .

The Lamé-constant constitutive relation for such a linear elastic solid has the same algebraic form as does the stress tensor for a viscous fluid :

$$\sigma = \sigma_{\rm eq} \mathbf{I} + \lambda \mathbf{I} \nabla \cdot \boldsymbol{u} + \eta [\nabla \boldsymbol{u} + \nabla \boldsymbol{u}^t] ,$$

but in the elastic case u is a displacement rather than a velocity and the constitutive properties λ and η are moduli rather than viscosities. There are some interesting parallels between viscous and elastic solutions of the continuum equations which are a consequence of this correspondence.⁶ Pure compression (in two dimensions) with

$$\epsilon_{xx} = \epsilon_{yy} = \frac{+dV}{2V} < 0 \; ,$$

establishes that the two-dimensional Bulk Modulus is equal to the sum of the Lamé constants, $B = -VdP/dV = \lambda + \eta$. In three dimensions $B = \lambda + \frac{2}{3}\eta$. Simple shear, with x displacement proportional to y, $u_x = \epsilon_{xy}y$, shows that the shear modulus, often denoted by G (or by μ), is identical to η in either two or three space dimensions :

$$G = \sigma_{xy} / \epsilon_{xy} = \eta$$
.

The elastic properties can be determined from the prototypical tension test illustrated in Figure 2.6 .

⁶Hoover, Ashurst, and Olness (1974).

Constitutive Relations for Elastic Solids



Figure 2.6: Extension of an elastic bar (height H and length L) by the longitudinal stress σ_{xx} in the absence of transverse stress $\sigma_{yy} = 0$. The [stress/strain] ratio (the small-strain slope of the stress-strain curve) defines Young's modulus E while the negative of the dimensionless ratio of transverse strain to the longitudinal strain defines "Poisson's ratio" ν :

$$E = \sigma_{xx}/\epsilon_{xx} \simeq \frac{d\sigma_{xx}}{d\epsilon_{xx}}$$
; $\nu = -\epsilon_{yy}/\epsilon_{xx}$.

To relate the two Lamé constants to the alternative representation of stress in terms of Young's modulus E and Poisson's ratio ν in two space dimensions first note that a vanishing transverse stress σ_{yy} ,

$$0 = \sigma_{yy} = (\lambda + 2\eta)\epsilon_{yy} + \lambda\epsilon_{xx} ,$$

implies that the ratio ("Poisson's ratio") of the transverse strain to the longitudinal strain is constant :

$$\nu_{\rm 2D} = \frac{-\epsilon_{yy}}{\epsilon_{xx}} = \frac{\lambda}{\lambda + 2\eta} \ .$$

The longitudinal stress σ_{xx} divided by the longitudinal strain ϵ_{xx} is likewise constant, and defines the two-dimensional Young's modulus :

$$\sigma_{xx} = \lambda(\epsilon_{xx} + \epsilon_{yy}) + 2\eta\epsilon_{xx} \longrightarrow E_{2D} \equiv \frac{\sigma_{xx}}{\epsilon_{xx}} = \frac{4\eta(\lambda + \eta)}{\lambda + 2\eta} .$$

Continuum Mechanics

In three space dimensions an exactly similar calculation, with two vanishing transverse stresses, $\sigma_{yy} = \sigma_{zz} = 0$, gives the corresponding results :

$$\nu_{\rm 3D} \equiv \frac{-\epsilon_{yy}}{\epsilon_{xx}} = \frac{-\epsilon_{zz}}{\epsilon_{xx}} = \frac{\lambda}{2\lambda + 2\eta} ; \ E_{\rm 3D} \equiv \frac{\eta(3\lambda + 2\eta)}{\lambda + \eta}$$

The linear-elastic formulation just described is adequate for Lagrangian descriptions of elastic strain because the shapes of Lagrangian elements, when compared to their initial shapes, allow strain to be estimated. If element shapes are not available to give the strains (as in Eulerian or smooth particle descriptions of deformation) it is conventional to integrate the strain *rates* to find the strains. These time derivatives provide the evolution of the strains, as functions of time, from given initial values :

$$\dot{\epsilon}_{xx} = (d/dt) \frac{\partial u_x}{\partial x} = \frac{\partial v_x}{\partial x}$$

$$\dot{\epsilon}_{xy} = (d/dt) \left[\frac{\partial u_x}{\partial y} + \frac{\partial u_y}{\partial x} \right] = \frac{\partial v_x}{\partial y} + \frac{\partial v_y}{\partial x} ;$$

$$\dot{\epsilon}_{yy} = (d/dt) \frac{\partial u_y}{\partial y} = \frac{\partial v_y}{\partial y}$$

With the strains determined the elastic stresses follow. The linear-elastic description of solids is useful until nonlinearities become important. Non-linear elasticity, element rotation, and element failure can each cause the linear description to fail. An alternative approach to elastic strength, which avoids this time integration, is described in Section 5.7.

In most cases material failure occurs before it is necessary to make any nonlinear corrections. Linear behavior must certainly come to an end for sufficiently great tensile or shear stresses. Under extreme tension new surfaces—pores or cracks—will form. Under extreme shear stress the material will flow in such a way that memory of the initial stress-free configuration is lost. We turn next to a phenomenological description of that irreversible flow, "plastic flow".

2.8 Constitutive Relation for Nonequilibrium Plasticity

The simplest irreversible flow model for solids is von Mises' 1913 flow law, according to which the shear stress driving shear flow is not allowed to

Constitutive Relation for Nonequilibrium Plasticity

exceed a maximum value, the "yield strength" Y. A shear stress infinitesimally larger than Y relaxes to the "yield surface" (a surface in stress space, which may depend upon the past history of the dynamics). In the simplest case this yield surface is a surface of constant shear stress Y:

$$(\sigma_{xx} - \sigma_{yy})^2 + 4\sigma_{xy}^2 = Y^2$$

where Y is the "yield strength", the longitudinal plastic flow stress $\pm \sigma_{xx}$ in the presence of a vanishing transverse stress, $\sigma_{yy} = 0$. The yield surface is proportional to the "second invariant" of the stress tensor and corresponds also to an elastic shear energy per unit volume :

$$\rho e_Y = \frac{\eta}{2} [(\epsilon_{xx} - \epsilon_{yy})^2 + \epsilon_{xy}^2] = \frac{Y^2}{8\eta}$$

The trace of the stress tensor is the "first invariant", $\sigma_{xx} + \sigma_{yy}$. These invariants, as the name suggests, are unchanged by a rotation of the coordinate system. Figure 2.7 illustrates the changed values of stress and strain corresponding to a 45-degree rotation of a yielding body with $\sigma_{xx} = \sigma = Y$ and corresponding elastic strains $\epsilon_{xx} = \epsilon_{yy}$. Ideal plastic deformation corresponds to shear taking place at constant volume. In applications with not only expansion and compression, but also shear deformation, computer algorithms designed to separate these effects must be applied. This approach is described in the following Section.



Figure 2.7. The tension-test specifier at the left, subject to the stress $\sigma_{xx} = \sigma = Y$, could alternatively be described in the rotated frame shown at the right, in which the longitudinal stretch and transverse shrinkage correspond to the superposition of tension and constant-volume shear.

Homogeneous linearly-elastic deformations convert a circle (or sphere) into an ellipse (or ellipsoid) in two (or three) dimensions. Evidently the elastic strains, and the stresses which result from them, are related to the changes in length of the two (principal axes) of the deformed circle (or three for a sphere). The *sums* of the corresponding strains, in the directions of

 $Continuum \ Mechanics$

the principal axes, correspond to the changes in area (or volume) :

$$\delta A/A = \epsilon_{11} + \epsilon_{22} ; \ \delta V/V = \epsilon_{11} + \epsilon_{22} + \epsilon_{33} .$$
$$\frac{1}{2}(\sigma_{11} + \sigma_{22}) = (\lambda + \eta)(\epsilon_{11} + \epsilon_{22}) ;$$
$$\frac{1}{3}(\sigma_{11} + \sigma_{22} + \sigma_{33}) = (\lambda + \frac{2}{3}\eta)(\epsilon_{11} + \epsilon_{22} + \epsilon_{33}) .$$

In two space dimensions the stress tensor has only two invariants, the trace and the second invariant $[\propto (\sigma_{xx} - \sigma_{yy})^2 + 4\sigma_{xy}^2]$. For this reason, a complete description of nonhydrostatic deformation requires just one additional variable, the *difference* of the principal strains, the shear strain, $\epsilon_{11} - \epsilon_{22}$, or the corresponding shear stress, $\sigma_{11} - \sigma_{22}$, or the deformational energy density associated with the shear, $\frac{1}{2}(\sigma_{11} - \sigma_{22})(\epsilon_{11} - \epsilon_{22})$. The *two* additional variables required in *three* dimensions can similarly be chosen to emphasize strain, or stress, or energy.

In numerical simulations the orientations of the principal axes must be followed in time. Let us consider the simpler two-dimensional case here. Consider a simple constant-area (shear) deformation with

$$\epsilon_{xy} = \frac{du_x}{dy} = \epsilon \longrightarrow \sigma_{xy} = \eta \epsilon \ ,$$

where the magnitude of the shear strain, ϵ , is infinitesimal. A 45-degree rotation of the coordinate system shows that the equivalent "principal axis" representation of the shear stress, $(\sigma_{11} - \sigma_{22})/2$, can be expressed in the rotated frame :

$$\sigma_{xy} = \sigma_{yx} = \frac{\sigma_{11} - \sigma_{22}}{2} ; \ \epsilon_{xy} = 2\epsilon_{11} = -2\epsilon_{22} = \epsilon_{11} - \epsilon_{22} ;$$
$$\epsilon_{11} = +\frac{\epsilon}{2} ; \ \epsilon_{22} = -\frac{\epsilon}{2} .$$

The work (expressed in units of energy per unit volume) required by the equivalent deformation is the same too :

$$\int \sigma : d\epsilon' = \frac{1}{2}\sigma : \epsilon = \frac{1}{2}[\sigma_{11}\epsilon_{11} + \sigma_{22}\epsilon_{22}] = \frac{1}{2}[\lambda(\epsilon_{11} + \epsilon_{22}) + 2\eta\epsilon_{11}]\epsilon_{11} + \frac{1}{2}[\lambda(\epsilon_{22} + \epsilon_{11}) + 2\eta\epsilon_{22}]\epsilon_{22} = \frac{1}{2}[\lambda(\epsilon_{11} + \epsilon_{22}) + 2\eta\epsilon_{11}]\epsilon_{11} + \frac{1}{2}[\lambda(\epsilon_{22} + \epsilon_{11}) + 2\eta\epsilon_{22}]\epsilon_{22} = \frac{1}{2}[\lambda(\epsilon_{22} + \epsilon_{21}) + 2\eta\epsilon_{22}]\epsilon_{22} = \frac{1}$$

Plasticity Algorithm

$$\eta(\epsilon_{11}^2 + \epsilon_{22}^2) = \eta(\frac{\epsilon^2}{4} + \frac{\epsilon^2}{4}) = \frac{1}{2}\eta\epsilon^2$$
.

2.9 Plasticity Algorithm

The simplest criteria for "plastic flow" are either stress-based or energybased, with irreversible plastic flow occurring whenever the shear stress or shear-strain energy becomes large enough. In two dimensions the stressbased and energy-based criteria are identical. *Plastic flow*, with $\sigma_{xx} = Y$ and $\sigma_{\perp} = \sigma_{yy} = 0$, occurs whenever the inequality

$$4\sigma_{xy}^2 + (\sigma_{xx} - \sigma_{yy})^2 \le Y_{2D}^2$$

is violated. In such a case the material undergoes a permanent deformation or "set". A stress σ_{xx} just exceeding Y_{2D} and applied to the end of a rectangular specimen, along with a vanishing transverse stress, $\sigma_{yy} = 0$, is sufficient to cause plastic yielding.

This physical picture of a limiting maximum shear stress can be implemented in a simple computer algorithm. Whenever the instantaneous yield stress Y_{2D} is exceeded by the invariant shear stress,

$$\sqrt{4\sigma_{xy}^2 + (\sigma_{xx} - \sigma_{yy})^2} \; ,$$

as calculated using the elastic equation of state for one timestep dt, the stress-tensor components are immediately scaled back, by a correction factor f very nearly equal to unity, so that the resulting scaled values lie on (rather than just outside) the "yield surface" :

$$f \equiv [4\sigma_{xy}^2 + (\sigma_{xx} - \sigma_{yy})^2]^{1/2} / Y_{2D} ;$$
$$(\sigma_{xx} - \langle \sigma \rangle) \longrightarrow (\sigma_{xx} - \langle \sigma \rangle) / f ;$$
$$(\sigma_{yy} - \langle \sigma \rangle) \longrightarrow (\sigma_{yy} - \langle \sigma \rangle) / f ;$$
$$\sigma_{xy} \longrightarrow \sigma_{xy} / f ,$$

where $\langle \sigma \rangle = \frac{1}{2}(\sigma_{xx} + \sigma_{yy})$ is the mean stress. The term "yield surface" could be replaced by "yield lines" in two dimensions because the principal-axis stress difference must lie between the two straight *lines*: $|\sigma_{11} - \sigma_{22}| < Y_{2D}$.

ws-book9x6

 $Continuum \ Mechanics$

In three dimensions Tresca's stress-based yield surface,

$$\sigma_{11} - \sigma_{33} = Y \; ,$$

where σ_{11} and σ_{33} are the maximum and minimum principal stresses, has sharp corners. von Mises' alternative energy-based yield surface is smooth. We adopt von Mises' form because it is the simpler of the two models :

$$(\sigma_{11} - \sigma_{22})^2 + (\sigma_{22} - \sigma_{33})^2 + (\sigma_{33} - \sigma_{11})^2 < 2Y_{3D}^2 ,$$

where the principal-stress axes (1, 2, 3) are chosen so that all the off-diagonal stress-tensor components vanish. Figure 2.6 shows two typical snapshots from a three-dimensional tension test based on von Mises' yield criterion.



Figure 2.8: Extension of a tapered bar of elastic-plastic solid using the Lagrangian simulation code Dyna3d with $64\times16\times16$ elements of unit density. $\lambda=\eta=1=40Y$ and the plastic strain at failure is unity. The extension velocity is one tenth the bulk sound speed and the original cross-sectional area varies quadratically with an overall change of ten percent, $\Delta y/L_y=0.9+0.1(2x/L_x)^2$ for ($0<|x|< L_x/2$).

The time-stepping algorithm for the three-dimensional rescaling relations, in an arbitrary Cartesian (x, y, z) coordinate frame, has a form similar to that used in two dimensions, but with the stress-tensor correction factor f including all six independent stress components :

 $f \equiv [4(\sigma_{xy}^2 + \sigma_{yz}^2 + \sigma_{zx}^2) + (\sigma_{xx} - \sigma_{yy})^2 + (\sigma_{yy} - \sigma_{zz})^2 + (\sigma_{zz} - \sigma_{xx})^2]^{1/2} / \sqrt{2} Y_{\rm 3D} \; .$

It is evident that plasticity introduces path dependence and hysteresis into solid flow problems. An initially stress-free and homogeneous square of material, first deformed by a shear exceeding the elastic limit $\sigma_{xy} = Y/2$ and then returned to its initial shape by a second shear deformation :

$$(x,y) \xrightarrow{+Y/2} (x + \epsilon y, y) \xrightarrow{-Y/2} (x, y) ,$$

Plasticity Algorithm

ends up with a nonvanishing residual stress, -Y/2. See Figure 2.9 for the stress-strain path describing a "perfectly plastic" material with a yield strength Y. Generally, the evolution of such path-dependent stresses can only be computed by the time integration of stress *rates*, as is detailed in the solid-phase problems treated in Chapters 8 and 9. Plasticity and failure are particularly hard time-dependent problems. Let us lay a foundation for their study by first considering some simpler stationary-state continuum simulations involving the transport of mass, momentum, and energy.



ws-book9x6

Continuum Mechanics

2.10 Example: Heat Conduction in One Dimension

Pure heat conduction is the simplest nonequilibrium process. Provided that we ignore thermal expansion, there is no dynamics in the evolution, so that the Eulerian and Lagrangian descriptions are exactly the same. Consider a mesh, periodic for convenience, with temperature distribution $\{T(x)\}$. Provided that the mesh is sufficiently fine to describe the gradients by first differences, the heat flux could be calculated from adjacent values of temperature. The best approach is to try out any proposed algorithm and compare the solutions to known results. In the case of a convergent algorithm this "known" result could be obtained by using a sufficiently fine mesh. Begin by computing the heat flux at x from the adjacent temperature values :

$$\left\{ Q(x) = \frac{\kappa}{2dx} [T(x - dx) - T(x + dx)] \right\} .$$

We can then express the time evolution of the temperature in terms of the heat-flux divergence :

$$\{ \dot{T}(x) \propto [Q(x-dx) - Q(x+dx)] \propto T(x-2dx) - 2T(x) + T(x+2dx) \},\$$

or

$$\{ \dot{T}_i \propto T_{i+2} - 2T_i + T_{i-2} \}.$$

Evidently this approach provides no link between the even and oddnumbered nodes. This leads to a relatively benign non-exponential instability (discussed in Section 6.12).

Here we avoid this difficulty by defining, or imagining, the heat flux *between*, rather than *at*, the nodes. Then, an exactly similar development gives the set of ordinary differential equations ,

$$\{ T_i \propto T_{i+1} - 2T_i + T_{i-1} \},\$$

which look more promising (the instability is gone, and the finer resolution reduces the error in \dot{T}_i by a factor of four). Numerical investigation, with periodic boundaries, ten equally-spaced nodes, and a sinusoidal initial condition,

$$T = \sin \frac{2\pi x}{10}$$
; $0 < |x| < 5$,

and using the fourth-order Runge-Kutta method, shows rapid convergence to a constant-temperature profile provided that the timestep is 0.6 or less.

Example: Sound Propagation in One Dimension

A timestep of 0.7 leads rapidly to numerical instability. More detailed investigation would reveal that the critical timestep for convergence varies as the square of the spacing between the nodes and that the error in the decay time varies in the same way.

2.11 Example: Sound Propagation in One Dimension

Consider again a ten-particle periodic mesh of unit-density elements but now with coordinates and velocities { x_i, v_i }, which allow the (Lagrangian) mesh to move and distort under the influence of pressure forces. If the ideal-gas pressure varies as the square of the density, $P = \frac{\rho^2}{2}$, then the differential equations governing the developing mesh could be replaced by the following finite-difference model :

$$\begin{split} \dot{v} &= -\frac{\nabla P}{\rho} \to \left\{ \begin{array}{l} \dot{x}_i = v_i \ ; \ \dot{v}_i = \frac{\rho_{i-1}^2 - \rho_{i+1}^2}{4\rho_i} \end{array} \right\} \ ; \\ &\left\{ \begin{array}{l} \rho_i = \frac{2}{x_{i+1} - x_{i-1}} \end{array} \right\} \ . \end{split}$$

Again we use periodic boundary conditions $\{-5 < x_i < +5\}$ in order to avoid the explicit consideration of boundary effects :

$$\left\{ x_i(t=0) = i - 5.5 ; v_i(t=0) = 0.01 \sin \frac{2\pi x_i}{10} \right\} \text{ for } \left\{ 1 \le i \le 10 \right\}.$$

The fourth-order Runge-Kutta algorithm of Section 4.4 has no problem integrating this set of twenty ordinary differential equations for several sound traversal times provided that the amplitude of the sound wave is not too large—say a few percent variation in density.

A somewhat larger density variation, with $0.01 \rightarrow 0.10$ in the initial conditions, leads to a reproducible instability (that is, the instability looks essentially the same with a large change in dt). Such an instability can be eliminated by the use of artificial viscosity, a viscosity designed to avoid the extreme compression associated with shockwave formation. If the artificial viscosity is used to increase the pressure of the *i*th particle whenever it is being compressed—that is whenever $v_{i-1} > v_{i+1}$ —an even-odd instability results. On the other hand if the pressures of two neighboring particles are *both* increased whenever the two approach each other, the calculation is stable.
60

Continuum Mechanics

2.12 Example: Rayleigh-Bénard Flow in Two Dimensions

Let us finally consider a more complicated example, one which includes simultaneous flows of mass, momentum, and energy in two space dimensions. The Rayleigh-Bénard problem describes a compressible heat conducting fluid subject to hot and cold boundary temperatures in a gravitational field. Here we apply a stable finite-difference algorithm⁷ as described in two of our Physical Review E articles.^{8,9}

The Eulerian equations to be solved are the continuity equation, motion equation, and energy equation discussed in Section 2.2 :

$$\begin{aligned} (\partial \rho / \partial t) &= -\nabla \cdot (\rho v) \ , \\ (\partial v / \partial t) &= -v \cdot \nabla v + (1/\rho) \nabla \cdot \sigma + g \ , \\ (\partial e / \partial t) &= -v \cdot \nabla e + (1/\rho) [\nabla v : \sigma - \nabla \cdot Q] \ . \end{aligned}$$

We simplify the computation by assuming (the assumption is exact for a low-density monatomic gas) that the bulk viscosity vanishes, so that the "second viscosity coefficient" λ , in the more-general constitutive equation:

$$\sigma_{\rm viscous} = \mathrm{I}\lambda\nabla\cdot v + \eta[\nabla v + \nabla v^{\rm t}]$$

has the value $-\eta$ required for the vanishing of the viscous contribution to the "mean stress", $\frac{1}{2}(\sigma_{xx} + \sigma_{yy})$. In this case, the stress tensor σ involves only the shear viscosity coefficient η and the equilibrium equation of state:

$$-\sigma_{eq} = P_{eq} = \rho eI = \rho kTI;$$

$$\sigma_{xx} = -\rho e + \eta [(\partial v_x / \partial x) - (\partial v_y / \partial y)];$$

$$\sigma_{xy} \equiv \sigma_{yx} = \eta [(\partial v_x / \partial y) + (\partial v_y / \partial x)];$$

$$\sigma_{yy} = -\rho e + \eta [(\partial v_y / \partial y) - (\partial v_x / \partial x)].$$

The heat flux vector Q requires the thermal conductivity κ :

$$Q_x = -\kappa (\partial T / \partial x) ; \ Q_y = -\kappa (\partial T / \partial y) .$$

⁷Puhl, Mansour, and Mareschal (1989).

⁸Castillo, Hoover, and Hoover (1997).

⁹Kum, Hoover, and Posch (1995).

Example: Rayleigh-Bénard Flow in Two Dimensions

The variables dependent on space (x, y) and time t are

$$\{ \rho, v_x, v_y, e, \sigma_{xx}, \sigma_{xy}, \sigma_{yy}, Q_x, Q_y \}$$
.

For their evaluation these nine variables require six gradient components :

$$\{ (\partial v_x/\partial x), (\partial v_x/\partial y), (\partial v_y/\partial x), (\partial v_y/\partial y), (\partial T/\partial x), (\partial T/\partial y) \}.$$

To simplify the problem as much as possible we choose the ideal gas law with constant heat capacity as our basic equation of state, together with constant kinematic viscosity and thermal diffusivity. We span the problem domain with a square mesh of cells (dx = dy).

A computation of the flow field can then be based on the following series of three steps for the integration of the four evolution equations for the four variables { ρ , v_x , v_y , e }.

1. $\partial \rho / \partial t$ is evaluated at all the *zone centers* by evaluating v_{\perp} and ρ as two-point averages at the midpoints of each of the four zone faces. These velocities vanish along the top and bottom boundaries of the problem.

2. The four components of the velocity gradient tensor,

$$\{ \frac{\partial v_x}{\partial x}; \frac{\partial v_x}{\partial y}; \frac{\partial v_y}{\partial x}; \frac{\partial v_y}{\partial y} \},\$$

and the temperature gradient components :

$$\{ \partial T / \partial x ; \partial T / \partial y \},\$$

are evaluated as first differences of averaged velocities and temperature at the *midpoints* of the four zone faces. These derivatives, together with the zone centered values of ρ and e provide zone-centered values of the stress tensor and the heat-flux vector.

3. Stress and heat-flux gradients are next evaluated at each node by computing differences of the zone-centered stress and heat-flux components. Mean densities at the nodes are evaluated by averaging the densities at the centers of the four contiguous zones. Densities along the top and bottom boundaries must be specified too. There are many other ways in which the finite-difference spatial derivatives might be formulated. The one chosen here leads to stable difference equations.

62

Continuum Mechanics



continuum simulation on the left]. The temperatures at the top and bottom are respectively 0.5 and 1.5. The overall density is 1.0. The thermal diffusivity and kinematic viscosity are both chosen equal to 0.4.

A Fortran program including fourth-order Runge-Kutta integration of all the continuum equations implementing these ideas requires a few hundred lines of coding. Either periodic boundary conditions or fixed boundary conditions can be implemented along the sides of the problem. Figure 2.10 illustrates the corresponding flow fields for a Rayleigh number of 10,000 and an 80×40 computational grid of 3200 square zones each with area unity. The (dimensionless) Rayleigh number ,

$$R = \frac{gL^4 |d\ln T/dy|}{\nu D_T} ,$$

Example: Rayleigh-Bénard Flow in Two Dimensions

is the ratio of the driving gravitational and thermal forces to the kinematic viscosity ν and thermal diffusivity D_T . For sufficiently large Rayleigh numbers (typically a few thousand) convection results.

In the problem illustrated here, the two-dimensional ideal-gas law, $P = \rho e = \rho kT$, was used, along with equal, and temperature-independent, values of the kinematic viscosity and thermal diffusivity. A useful initial condition, appropriate for the two-roll solution shown in the Figure, is :

$$v_x \propto +\sin(kx)\sin(ky)$$
;

$$v_y \propto +\cos(\frac{3}{2}kx)\cos(\frac{1}{2}ky) ,$$

where the velocity vanishes on all four boundaries and k is chosen so that kx and ky vary from $-\pi$ to $+\pi$. For periodic boundaries on the sides the $\cos(\frac{3}{2}kx)$ can be replaced by $\cos(kx)$. In our simulations we choose

$$g \propto 1/L$$
 : $\nu = D_T \propto L$,

resulting in constant Rayleigh number, constant Reynolds' number, and a stationary flow velocity v(x/L, y/L) as L is increased. The stationary kinetic energy of the flow field becomes proportional to the system size L^2 . In the (80 × 40 = 3200)-zone square-zone example the stationary value of the kinetic energy is 23 with periodic vertical boundary conditions and 9 with static ones.

This Rayleigh-Bénard problem is an instructive one. For given *boundary conditions* (for instance, periodic vertical boundaries with specified temperatures and gravitational field) the solution of the continuum equations of motion is not necessarily unique. Solutions with two, four, or even six rolls can persist for long times provided that proper initial conditions are chosen. Because one can expect that particle simulations are describable by continuum mechanics, for sufficiently many particles, the same lack of uniqueness extends to the time-averaged solutions of the microscopic dynamical equations. See Chapter 6 for additional discussion.

Continuum Mechanics

2.13 References

- S. Chapman and T. G. Cowling, *The Mathematical Theory of Nonuni*form Gases (Cambridge University Press, 1952).
- (2) C. G. Hoover, A. J. DeGroot, and R. J. Sherwood, "ParaDyn: a Parallel Nonlinear Explicit Three-Dimensional Finite-Element Code for Solid and Structural Mechanics", Report UCRL-MA-140943 (Lawrence Livermore National Laboratory, 2004). ParaDyn is the export controlled parallel version of Dyna3d. For a description of the serial code, which is available to everyone, see J. L. Lin, "Dyna3d, A Nonlinear, Explicit, Three-dimensional Finite Element Code for Solid and Structural Mechanics" (Lawrence Livermore National Laboratory, 2004).
- (3) D. A. Young, *Phase Diagrams of the Elements* (University of California, Berkeley, 1991).
- (4) J. von Neumann and R. Richtmyer, "A Method for the Numerical Calculation of Hydrodynamic Shocks", Journal of Applied Physics 21, 232-257 (1950).
- (5) B. L. Holian, W. G. Hoover, B. Moran, and G. K. Straub, "Shockwave Structure via Nonequilibrium Molecular Dynamics and Navier-Stokes Continuum Mechanics", Physical Review A 22, 2798-2808 (1987).
- (6) W. G. Hoover, W. T. Ashurst, and R. J. Olness, "Two-Dimensional Computer Studies of Crystal Stability and Fluid Viscosity", Journal of Chemical Physics 60, 4043-4047 (1974).
- (7) A. Puhl, M. M. Mansour, and M. Mareschal, "Quantitive Comparison of Molecular Dynamics with Hydrodynamics in Rayleigh-Bénard Convection", Physical Review A 40, 1999-2012 (1989).
- (8) V. M. Castillo, Wm. G. Hoover, and C. G. Hoover, "Coexisting Attractors in Compressible Rayleigh-Bénard Flow", Physical Review E 55, 5546-5550 (1997).
- (9) O. Kum, Wm. G. Hoover, and H. A. Posch, "Viscous Conducting Flows with Smooth Particle Applied Mechanics", Physical Review E 52, 4899-4908 (1995).

Chapter 3

Smooth Particle Methods

/ Summary and Motivation / Evolution Equations / Interpolation / Alternative Interpolations / Weight Functions / SPAM Continuity Equation with $\nabla \cdot v$ / Spatial Derivatives ($\nabla \rho$, $\nabla \cdot P$, $\nabla \cdot Q$) / SPAM Equation of Motion and Energy Equation / Rezoning, Effect of Particle Size / Ideal Gas Isomorphism / Alternative Spatial Derivatives (∇v , ∇T) / Artificial Viscosity /

/ 8 Figures /

Example Problems : [Adiabatic Atmosphere, Isothermal Atmosphere]

3.1 Summary

The notion of "smooth particles" allows us to solve continuum problems by following the motion of representative particles. In this approach, just as in the finite-element and finite-difference methods, the partial differential field equations of continuum mechanics become a finite set of ordinary differential equations, here for the evolution of particles' positions and energies. SPAM (Smooth Particle Applied Mechanics) encompasses a set of techniques—each with a corresponding computer algorithm—which are specially useful in simulating complex mechanical deformations of fluids and solids. Smooth particles are not "tracer" particles, or the particles used in "particle-in-cell" methods. Although their mechanics resembles pointparticle mechanics, SPAM particles are also not the usual mass points. Instead the SPAM particles can best be thought of as moving clumps or

distributions of material, characterized by stresses, strains, heat fluxes, temperature and velocity gradients, and internal energies, as well as by the usual coordinates and velocities of ordinary particle mechanics.

In this Chapter we describe the smooth-particle solution method to solve the continuum partial differential equations for { $\dot{\rho}, \ddot{r} = \dot{v}, \dot{e}$ }. This approach results in discrete ordinary differential equations for *particle* properties { $\dot{\rho}_i, \ddot{r}_i = \dot{v}_i, \dot{e}_i$ } with recipes expressing all the local continuum properties as sums of particle contributions. The weight function w(r), introduced in Section 3.4, is the key to implementing SPAM algorithms.

SPAM is unusual in that it can easily treat fluctuations in local variables (such as $\langle v^2 \rangle - \langle v \rangle^2$), not just mean values. SPAM also incorporates all the viscous and conductive dissipative mechanisms which convert work to heat, and can be formulated so as to satisfy *almost* all of the usual conservation laws—mass, linear momentum, and energy—*exactly*. (Angular momentum is still a fly in the ointment, and is discussed separately in Section 8.9).

Some of the algorithmic techniques which SPAM requires have an independent utility in a variety of applications. Smooth-particle interpolation is an example. This interpolation technique can be used to analyze atomistic or molecular simulations by combining local point-particle properties to get average field variables. Here we describe and set out these particle-based tools. The reader can choose for himself those best suited to his particular application.

3.2 Motivation

In the absence of important atomic-scale effects, macroscopic continuum mechanics is the method of choice for simulation. In continuum mechanics the density, velocity, pressure, and other related field variables are assumed to vary *continuously* throughout the space-time region of interest. Undergraduate texts describe the solution of "easy" continuum problems characterized by this smoothness. These easy problems include the flow of heat in response to imposed thermal boundary conditions as well as the linear-elastic response of solids to imposed loads. A variety of grid-based approaches work very well for such problems, as we saw in the example problems of Chapter 2. For more difficult problems in which the structure under study undergoes extreme shape changes and forms new surfaces while resisting interpenetration at interfaces, more sophisticated and flexible techniques are required.

Basic Equations

Smooth Particle Applied Mechanics (SPAM) is such a technique. It provides a versatile approach to many of the difficult problems in computer simulation. The breakup of a cavitating fluid (see Figure 2.1) and the penetration of one solid by another (Figures 9.7 through 9.10) are two such challenging examples. SPAM also provides a simple evaluation method for *all* the continuum variables, as well as the spatial gradients required by the evolution equations, *everywhere*. This *global* evaluation simplifies interpolation, rezoning, and Fourier transformation. Because SPAM is so flexible and easy to program, it should be included in the toolkit of anyone doing simulations. Even so, SPAM is not a panacea. Some of the difficulties involved in its applications are described in Chapter 8.

3.3 Basic Equations

The primary application of SPAM is to the solution of problems in Lagrangian continuum mechanics, where the governing partial differential equations describe the comoving evolution of the density ρ , coordinate r, velocity v, and energy per unit mass e in terms of gradients of the velocity, pressure tensor P, and the heat-flux vector Q:

$$\dot{\rho} = -\rho \nabla \cdot v ;$$

$$\rho \ddot{r} = \rho \dot{v} = -\nabla \cdot P \longleftrightarrow \rho \ddot{x} = \rho \dot{v}_x = -\left(\frac{\partial P_{xx}}{\partial x}\right) - \left(\frac{\partial P_{xy}}{\partial y}\right) ;$$

$$\rho \dot{e} = -P : \nabla v - \nabla \cdot Q .$$

In conventional finite-element simulations, the sum total of these conservation equations for an element (the volume integrals, integrated over that element) give the rates at which the element's mass, comoving momentum, and internal energy change. Momentum change is induced by stress gradients. Internal energy changes as the element does mechanical work on its surroundings and exchanges heat through local variations in temperature. In D space dimensions the double-dot notation $P: \nabla v$ in the energy evolution equation indicates the sum of all D^2 terms of the form $P_{ij}\nabla_i v_j$. The particles of smooth particle applied mechanics (SPAM) resemble finite elements in that SPAM particles have their own individual masses, momenta, and energies. But in SPAM changes in those properties come directly from

the ordinary differential equations for $\{ \dot{r}, \dot{v}, \dot{e} \}$, and do not require a spatial integration step over each element.

The three continuum conservation equations are perfectly general. They apply to gases and liquids as well as to solids. In discussing solids the stress tensor, $\sigma \equiv -P$, is often used. For simplicity we use the pressure tensor P for *all* problems, either fluid or solid. The only requirement for the validity of the governing equations is that the comoving fluxes of momentum and energy, P and Q, are *local* quantities, so that their gradients are well defined. Flows in systems as diverse as phase-separating fluid mixtures, deforming metals, breaking rocks, and colliding astrophysical bodies can all be treated with continuum mechanics and SPAM .

Although the dependent variables { ρ, v, e } are defined throughout space and for a continuously varying time, difficult continuum problems are usually made tractable by "discretizing" space with a grid of points, either moving or stationary, and discretizing time with a closely-spaced set of discrete times. With SPAM there is no need for the spatial grid points to be arranged in a regular manner. Any convenient grid can be used. With these two simplifying discretizations the partial differential field equations of continuum mechanics are replaced by a finite set of ordinary differential equations, with solutions given at a finite set of discrete times. The loworder Runge-Kutta methods described in Chapter 2, and elaborated in Chapter 4, are the simplest and best choices for the time integration.

3.4 Interpolation on an Irregular Grid

Until Gingold, Lucy, and Monaghan conceived of SPAM in 1977, the usual algorithms for solving continuum problems evolved the field variables ρ , v, e on field points making up a regular space-time grid. I learned about such grid-based methods in the 1960s, at the "Rad Lab" in Livermore, from a self-taught expert, Mark Wilkins.¹ Both "comoving" Lagrangian grids, moving with the material, or stationary Eulerian grids, fixed in space, were used. In either case, the spatial derivatives appearing in the continuum equations were approximated as finite differences. Although these gridbased approaches work well for simple problems, they have trouble dealing with the chaotic irregular flows typified by mixing, fracture, and turbulence. The distortion of a comoving mesh, the difficulty of following material interfaces, the prevention of overlaps within a fixed mesh, and the smooth

 $^{^{1}}$ Wilkins (1970).

Interpolation on an Irregular Grid

implementation of boundary conditions all can cause major headaches.

Many techniques have evolved to deal with these problems. SPAM solves the problem of choosing a spatial grid by introducing *particles* whose instantaneous coordinates *define* an interpolation grid for all the field variables. The points locating the SPAM particles can be thought of as the "centers of mass" of the particles. The points themselves make up the interpolation grid used in SPAM. The influence of these particles has a spatial range *h* described by their "weight functions" { w(r < h) }. Typical weight functions are localized, and very smooth, with a *finite* range *h*,

$$w(r > h) = 0$$
; $w'(h) = 0$; $w''(h) = 0$,

but otherwise resemble Gaussian functions centered on the particles. For typical weight functions, see Figure 3.2 in Section 3.6.

It is possible, but not usual, for the weight functions' ranges to vary with time, with location, and with direction. In any case, the grid made up by the spatially extended particles can move and distort continuously, reflecting the particles' motion. This spatial interpolation method is in itself a useful algorithm for "smoothing" data sets which are only known on a discrete and irregular set of points.

These averaging ideas, applied to the *partial* differential equations of continuum mechanics, lead directly to sets of *ordinary* differential equations for the time development { $\dot{r} = v, \dot{v}, \dot{e}$ } of all the particle coordinates, velocities, and energies { r, v, e }. The evolution of the velocity of the *i*th particle involves the contributions of all of its neighbors { j } through the smooth particle equations of motion, which are derived in Section 3.9,

$$\begin{cases} \dot{v}_i = m \sum_j \left[\left(\frac{\sigma}{\rho^2} \right)_i + \left(\frac{\sigma}{\rho^2} \right)_j \right] \cdot \nabla_i w(r_{ij}) = \\ -m \sum_j \left[\left(\frac{P}{\rho^2} \right)_i + \left(\frac{P}{\rho^2} \right)_j \right] \cdot \nabla_i w(r_{ij}) \end{cases}$$

and look very much like the motion equations of molecular dynamics, but involve the individual particle stress or pressure tensors $\{\sigma \equiv -P\}$ in place of the more usual interatomic forces. Note that all the i = j terms vanish. By using this approach one can solve complex continuum problems with a simple particle technique.

Let us illustrate smooth-particle interpolation with the simplest example, the calculation of the smooth-particle mass density at a point "r" in

space. Consider a two-dimensional space including also a set of points $\{r_j\}$ (the SPAM particle coordinates). Here, as usual, we choose the same masses and weight functions for all the particles :

$$\{ m_j = m \} ; \{ w_j(r - r_j) \equiv w(r - r_j) \equiv w_{rj} \}.$$

Each particle's mass density (or probability distribution, or influence function) can be thought of as spread out in space according to the probability density $w(|r - r_j| < h)$, where the range (or extent) h of w characterizes the "size" of the particles.

The sum of all the particle contributions to the mass density ρ at any interpolation point r in space—where r is an arbitrary point, not necessarily the location of a particle—gives the total density (or probability) there. This idea *defines* the smooth-particle density at r :

$$\rho(r) \equiv \rho_r \equiv \sum_j m_j w(r - r_j) ,$$

where the sum is over all particles. Note that each smooth particle j has its own mass, m_j . It is also possible that individual particles can have variable chemical compositions (reactants or products) or physical compositions (ice or water or water vapor), but we exclude these interesting possibilities here.

In all the example problems treated in this book we will only consider the special case in which all the smooth-particle masses have a common value, { $m_j \equiv m$ }. In this Chapter we nevertheless make the possibility of different particular masses explicit for pedagogical reasons, emphasizing that the conservation relations and rezoning procedures can be applied to sets of particles with a selection of masses. Rather than locating the various smooth-particle masses at points, with singular probability densities given by delta functions, the smooth-particle mass distribution is instead smoothed out by superposing the continuously differentiable weight functions { w(r < h) }. Not only w, but also w' and w'', vanish at the limit of the range of the weight function, r = h. The smooth distribution of the mass density which results (with two continuous derivatives everywhere) is the reason that the interpolating particles are referred to as smooth particles.

Very smooth interpolation is the strong point of the smooth particle methods. Offhand, it might appear to be difficult to take a set of discrete function values { $f(r_i)$ }, associated with the irregular set of particle locations { r_i }, and to construct from those an interpolated function with at least two continuous derivatives everywhere. Evidently linear or quadratic

Alternative Averages: $[f_0, f_1, f_2, \dots]$

functions are not enough and a least-squares procedure involving a large grid of points is not an appealing computational task. SPAM solves the very smooth interpolation problem in a very simple and interesting way, by abandoning the idea of fitting the particle values exactly and instead defining local averages in terms of contributions of many contributing particles. An apparent disadvantage (more apparent than real) is the lack of uniqueness of smooth-particle interpolation. We outline here the simplest approach, and consider the lack of uniqueness in the following Section.

At any location r and for any general field variable f defined at all the particle coordinates $\{r_i\}$, we can add all the nearby particle contributions $\{f(r_i)\}$ to obtain an interpolated average value of f(r) in terms of the discrete set of $\{f_i\}$:

$$f(r) = \sum_{i} m_i f_i w(r - r_i) / \sum_{i} m_i w(r - r_i) = \sum_{i} m_i f_i w_{ri} / \rho(r) \longrightarrow$$

$$f(r)\rho(r) \equiv f_r \rho_r \equiv (f\rho)_r \equiv \sum_i f_i m_i w(r-r_i) \equiv \sum_i f_i m_i w_{ri} \; .$$

At any location r, the *interpolated* value of the field variable f(r) is given by a weighted average of contributions $\{f_i\}$ from all those particles i which lie within the range h of the location r. We emphasize that the discrete particle properties $\{f_i\}$, one for each particle, differ from the *continuum* of interpolated averages f(r), which are defined everywhere, not just at the particle locations r_i . In particular, f_i differs from $f(r = r_i)$ whenever, as is usual, there are local fluctuations in f. The variable ρ is the only exception. Within an arbitrary multiplicative factor, density is *always* an average over nearby particles' contributions. We strive to distinguish field variables from particle variables by using the notations $f(r) = f_r$ for field variables and f_i or f_j for particle variables.

3.5 Alternative Averages: $[f_0, f_1, f_2, \dots]$

It might well be thought that averages of field variables have a somewhat arbitrary nature in SPAM. This is quite true. In the last Section we calculated the field-variable average at a point as the ratio of two sums. This average-value definition is only one of many possibilities. In this Section we use the notation f_0 to distinguish this average from the f_1 and f_2 averages

ws-book9x6

Smooth Particle Methods

defined later:

$$f_0(r) \equiv \sum_i f_i m_i w(r - r_i) / \sum_i m_i w(r - r_i) = \sum_i f_i m_i w_{ri} / \rho_r$$
.

The individual particle variables $\{f_i\}$ can be obtained directly from the solutions of the ordinary differential equations for the coordinates, velocities, stresses, and energies or from constitutive relations based on those underlying variables. This $f_0(r)$ average-value definition is the natural one if we regard the individual terms, $\{m_i w_{ri}\}$, as unnormalized probabilities. Then $f_0(r)$ is just the usual statistical expression for an arithmetical average. This definition has an advantage over alternative ones: it exactly reproduces the average value of f(r) when f is constant, with all the $\{f_i\}$ equal.

There are many other plausible alternative averaging methods, some useful and some not. Consider, as examples, these three various possible definitions $\{f_0, f_1, f_2\}$ for an average value at r, f_r , a weighted average of nearby particle variables $\{f_i\}$, including all particles which lie within the range h of the field point r:

$$\rho(r)f_0(r) = (f\rho)_r \equiv \sum_i f_i m_i w_{ri} ; \ w_{ri} \equiv w(r-r_i) ;$$
$$\rho(r) = \rho_r = \sum m_j w_{rj} ;$$

$$f_1(r) \equiv \sum_i (f/\rho)_i m_i w_{ri} \; ; \; (f/\rho)_i \equiv f_i/\rho_i \; ; \; \rho_i \equiv \sum_j m_j w_{ij} \; ; \; w_{ij} \equiv w(r_i - r_j) \; ;$$
$$f_2(r)/\rho(r) \equiv \sum_i (f/\rho^2)_i m_i w_{ri} \; ; \; (\rho^2)_i = \rho_i^2 \; .$$

Any other powers of density, or even functions of density, could be used to define still other smooth-particle averages alternative to f_0 , f_1 , and f_2 .

Evidently the first expression above for f_0 is *exactly correct* if $f = f_0$ is constant, as is the second if $f = f_1 \propto \rho$, as is the third if it happens that $f = f_2 \propto \rho^2$. Proponents of the second scheme view it as a "Monte Carlo" average of f. In order to see that the three schemes really are generally different, consider the simplest possible application of the three averages. Let all of the particle variables $\{f_j\}$ have the common value $f_j \equiv 1$, all the smooth particles have unit mass $m_j = m \equiv 1$, and arrange the particles in a regular periodic lattice at unit density, $\langle \rho \rangle \equiv 1$ (square lattice in two

Alternative Averages: $[f_0, f_1, f_2, \dots]$

dimensions and cubic in three, for instance). With this construction each particle has exactly the same smooth-particle density,

$$\{ \langle \rho \rangle = 1 \neq \rho_i = \rho_j = \sum_j m w_{ij} \} ,$$

for all *i* and *j*. At an *arbitrary* location *r* the three definitions give (where we use ρ_i to indicate a typical individual smooth-particle density computed *at* the *i*th particle site),

$$(f\rho)_r \longrightarrow f_0(r) = \frac{\sum_i f_i w_{ir}}{\rho_r} \equiv 1 ;$$

$$f_r \longrightarrow f_1(r) = \sum_i (f_i w_{ir}/\rho_i) \equiv \frac{\rho_r}{\rho_i} ;$$

$$(f/\rho)_r \longrightarrow f_2(r) = \rho_r \sum_i (f_i w_{ir}/\rho_i^2) \equiv \frac{\rho_r^2}{\rho_i^2} .$$

In practice, the errors incurred by these averages are not very large, at most two percent for h = 2d, where d is the nearest-neighbor separation distance, and about ten times less for h = 3d, using Lucy's weight function, which is discussed in the following Section. For the one-dimensional Lucy weight function with h = 3 the density at each particle is $\rho_i = 65m/64 = 520m/512$ while the interpolated density midway between two particles is $\rho_r = 505m/512$.

In Section 3.7 we will see that the average velocity, $\langle v \rangle$, is best evaluated by using the average f_0 because the resulting continuity equation conserves mass exactly. We will see in Section 3.9 that the average f_2 is best for Pand Q, leading to exact conservation of the momentum and energy. It is interesting, and almost paradoxical, to notice that the substitution $f \equiv g\rho^n$ can be used to convert any of the three averaging methods discussed here to one of the others.

The choice of averaging algorithm is crucial, both for the conservation laws and for reasonable constitutive relations. Because the continuum equations all contain field variable derivatives, such as ∇v , $\nabla \cdot P$, ∇T , ..., the derivative expressions following from the averaging algorithm are crucial too. Consider the gradients generated by the three algorithms:

$$\nabla (f\rho)_r \equiv \sum_j f_j m_j \nabla_r w_{rj} ;$$

$$\nabla f_r \equiv \sum_j (f/\rho)_j m_j \nabla_r w_{rj} ;$$

$$\nabla (f/\rho)_r \equiv \sum_j (f/\rho^2)_j m_j \nabla_r w_{rj} \ .$$

There is no doubt that these three possibilities are different to one another despite the fact that any one of them can be "derived" from any other by making a substitution $f \equiv g\rho^n$. The most *useful* expressions for gradients come from the first and the last of the three possibilities just given. The two resulting expressions for the gradient ∇f are discussed separately, in Sections 3.8 and 3.12. We will see there that the first of the two gradient expressions $\nabla(f\rho)_r$ is ideal for describing the gradients ∇v and ∇T leading to nonequilibrium fluxes of momentum and energy while the last, $\nabla(f/\rho)_r$, is ideal for evaluating $\nabla \cdot P$ and $\nabla \cdot Q$ so as to conserve the overall system's momentum and energy. Before considering the details of those applications let us examine the weight functions underlying smooth-particle simulations in a bit more detail.

3.6 Weight Functions

For reasonable accuracy at a manageable cost the range h of the weight function w(r < h) should be chosen so that the weighted field-variable averages include contributions from about 20 particles. Lucy's and Monaghan's weight functions, described below, are the simplest practical choices. Figure 3.1 compares density results using the two forms. It shows the computed smooth-particle lattice-site densities in regular one-, two-, and threedimensional linear, square, and cubic lattices, as a function of the maximum distance h of the included neighbors.

Lucy introduced a particularly useful and simple choice for w:

$$w(r < h) = \left(\frac{5}{\pi h^2}\right) \left[1 + 3\frac{r}{h}\right] \left[1 - \frac{r}{h}\right]^3 ,$$

where the prefactor here has been chosen to match the normalization condition appropriate to two space dimensions :

$$\int_0^h 2\pi r w(r) dr = 1 \ .$$

75

Weight Functions



Lucy's weight function, a quartic with five coefficients, is arguably the simplest one which satisfies the five minimal conditions of (i) normalization, (ii) w'(0) = 0, (iii) w(h) = 0, (iv) w'(h) = 0, and (v) w''(h) = 0. The normalization of w is chosen to make the representation exact in the hypothetical large-h limiting case of infinitely-many particles, uniformly distributed in space. The one-, two-, and three-dimensional normalizations of Lucy's function are these :

$$\int_0^h 2w_{\rm 1D}(r)dr = 1 \ ; \ \int_0^h 2\pi r w_{\rm 2D}(r)dr = 1 \ ; \ \int_0^h 4\pi r^2 w_{\rm 3D}(r)dr = 1 \ .$$

For Lucy's case it is easy to verify that the prefactors multiplying the basic

quartic function, $[1+3\frac{r}{h}][1-\frac{r}{h}]^3$, must be

$$(5/4h)_{1D}$$
; $(5/\pi h^2)_{2D}$; $(105/16\pi h^3)_{3D}$.

Another form of weight function, cubic rather than quartic, was, and continues to be, recommended by Monaghan and has also been widely used. If one smoothly connects two cubic splines at $r = \frac{h}{2}$, $w(0 < \frac{r}{h} < \frac{1}{2})$ and $w(\frac{1}{2} < \frac{r}{h} < 1)$, and additionally imposes the *six* conditions that w, w', and w'' are continuous at $\frac{r}{h} = \frac{1}{2}$ and vanish at $\frac{r}{h} = 1$, there are *two* additional conditions necessary to fix all *eight* coefficients in the two cubic functions: w must be normalized and w' must vanish as $r \longrightarrow 0$. It is evident that the longer-range part of the weight function $w(\frac{1}{2} < \frac{r}{h} < 1)$ must be proportional to $(1 - \frac{r}{h})^3$. Then it is easy to find the unique cubic function matching w, w', and w'' at $\frac{r}{h} = \frac{1}{2}$. This calculation shows that Monaghan's weight function has the functional form:

$$w_{\text{Monaghan}} \left(0 < \frac{r}{h} < \frac{1}{2} \right) = c \left(\frac{1}{2} - 3\frac{r^2}{h^2} + 3\frac{r^3}{h^3} \right) ;$$
$$w_{\text{Monaghan}} \left(\frac{1}{2} < \frac{r}{h} < 1 \right) = c \left(1 - \frac{r}{h} \right)^3 .$$

Again the normalization constant c depends upon the dimensionality:

$$c_{1\mathrm{D}} = 8/3h \; ; \; c_{2\mathrm{D}} = 80/7\pi h^2 \; ; \; c_{3\mathrm{D}} = 16/\pi h^3 \; .$$

Figure 3.2 shows both Lucy's and Monaghan's weight functions, drawn for the two-dimensional case. There is no special magic in Lucy's function or in Monaghan's. Other choices, such as $w \propto \cos^3(r < \pi/2)$, $(1 - r^2)^3$, or $(1 - r^2)^4$, could equally well be used. Lucy's is only the simplest useful special case among the family of weight functions :

$$\{ w_n \propto \left(1 + n\frac{r}{h}\right) \left(1 - \frac{r}{h}\right)^n \},$$

which vanish at r = h with n - 1 continuous derivatives. We will have a need for w_4 in stabilizing solid lattices. See Section 5.7.

In problems which are inhomogeneous, or anisotropic, it is sometimes convenient to make the smoothing length h depend upon spatial orientation, location, and time. The only essential requirements that any weight function must satisfy are that it "look like" a delta function, with two continuous derivatives everywhere and with both derivatives vanishing at the maximum length, r = h.

77

Weight Functions



Figure 3.2: Lucy's (full line) and Monaghan's (dashed line) weight functions w(r < h) together with their first and second derivatives. The slopes are used to estimate gradients in continuum simulations. The integrals of the two weight functions are identical but the shapes differ slightly. The Figure is drawn using two-dimensional weights with h = 3. The intercepts are $w_{\text{Lucy}}(0) = 35/63\pi$ and $w_{\text{Monaghan}}(0) = 40/63\pi$.

To illustrate the simplest applications of Lucy and Monaghan's weight functions consider again a regular array of points in one dimension with the range h three times the nearest-neighbor spacing. The average densities at each lattice site, according to the two weight functions, are, respectively :

$$\rho_{\text{Lucy}} = (5/12) \left[\frac{3}{27} + \frac{16}{27} + \frac{27}{27} + \frac{16}{27} + \frac{3}{27} \right] = (5/12) \frac{65}{27} = \frac{325}{324} ;$$

$$\rho_{\text{Monaghan}} = (4/9) \left[\frac{2}{27} + \frac{15}{27} + \frac{27}{27} + \frac{15}{27} + \frac{2}{27} \right] = (4/9) \frac{61}{27} = \frac{244}{243} ;$$

For r just inside the range h, Lucy's and Monaghan's weight functions both vanish as the cube of the separation, $w(r) \propto (h-r)^3$. Thus w has two August 16, 2006 17:44

Smooth Particle Methods

continuous derivatives everywhere. Likewise, the first and second spatial derivatives of field-variable sums based on w, such as $\sum_j m_j f_j w(r-r_j) \equiv f(r)\rho(r)$, can have no discontinuities.



angular lattice structures in two dimensions. The curves cross for 1/h values 0.4021, 0.3394, and 0.2857. The choice h = 2.53669 reproduces the actual square-lattice density of unity precisely. The choice h = 2.41266 reproduces the triangular lattice density of unity to ten-digit accuracy.

The density errors, of order one half a percent for h = 3 in one dimension, are no larger in two or three dimensions. Figure 3.3 compares the computed smooth-particle densities (with an overall actual density of unity) for the square-lattice and the triangular-lattice (closest packed, with six nearest neighbors) structures. The abscissa is h. Once h exceeds 2.5 the errors incurred are quite acceptable. The error incurred using Lucy's w(r) at a particle to compute the average density of a completely random

Weight Functions

distribution of particles ,

$$\rho(r_i) = w(0) + \int_0^h (N/V) w(r) 2\pi r dr = w(0) + (N/V) \ ; \ w(0) = 5/\pi h^2 \ ,$$

is not shown. The error $5/\pi h^2$ is, very roughly, a factor of ten larger.

Evidently a choice of h which is "tuned" to a particular structure can stabilize it relative to competing structures. The special choice h = 2.53669 which reproduces the square-lattice density can cause that lattice to form when an initially-random set of points is annealed. See Figure 3.4.

The same type of smooth-particle interpolation method can be usefully applied to the analysis of microscopic molecular dynamics just as it can to continuum problems. This is an alternative to its usual application to continuum problems. In the microscopic case the individual particles have their own kinetic and potential energies. Smooth-particle interpolation of microscopic data can provide a *twice-differentiable, continuous* representation of both these particle energy functions based on the summed contributions from discrete points. In practical applications this interpolation scheme is much more useful than is the completely-formal delta-function expression appropriate to mass points, $\rho \equiv \sum m_j \delta_j$.

The "very smooth" SPAM representation of field variables, with at least two continuous derivatives, is the key to solving the continuum evolution equations for ρ , v, and e. Consider ρ_i , the total smooth-particle mass density computed at the location of particle i. The density ρ_i is given by a special case of the general rule for $\rho(r)$, with $r \longrightarrow r_i$:

$$\rho_i \equiv \rho(r_i) \equiv \sum_j m_j w(r_i - r_j) = \sum_j m_j w_{ij}.$$

Note that this definition, together with the choice of the normalization of w guarantees that the density integral over all of space is identically equal to the total mass. In two dimensions, for instance :

$$\sum_{j} m_{j} \equiv \int \int \rho(x, y) \, dx \, dy \; .$$

80

Smooth Particle Methods



density of unity shown at times 20, 40, 60, and 80. The initial configuration, motionless with the coordinates random, was annealed for a time of 80 with a relaxation time of 10, using the embedded-atom equation of state $P = 100(\rho^3 - \rho^2)$ and the SPAM motion equations of Section 3.9.

3.7 Continuity Equation from $\nabla \cdot v$ with SPAM

The Eulerian and Lagrangian continuity equations,

$$(\partial \rho / \partial t) = -\nabla \cdot (\rho v)$$
 and $\dot{\rho} = -\rho \nabla \cdot v$,

express conservation of mass in continuum mechanics. It is interesting that the usual density *definition* used in SPAM, when applied at the location of

Continuity Equation from $\nabla \cdot v$ with SPAM

an individual particle, particle i:

$$\rho(r) = \rho_r \equiv \sum m_j w(r - r_j) \longrightarrow \rho_i \equiv \sum_j m_j w_{ij} ,$$

must be exactly consistent with the continuity equation for the reason stressed in the last section: mass is automatically conserved by the method. It is instructive to show this consistency in detail. Consider the time-rate-of-change of the density at particle i, due to the motion of that particle and the motions of all the neighboring particles $\{j\}$ with which it interacts.

$$\dot{\rho}_i = \sum m_j \dot{w}_{ij} = \sum m_j (dw/dr)_{ij} [\dot{r}_i \cdot \nabla_i r_{ij} + \dot{r}_j \cdot \nabla_j r_{ji}] = \sum m_j v_{ij} \cdot \nabla_i w_{ij} ,$$

where r_{ij} is the separation and v_{ij} is the relative velocity, $\dot{r}_i - \dot{r}_j$:

$$r_{ij} \equiv r_i - r_j = -r_{ji} ; v_{ij} \equiv v_i - v_j = -v_{ji} .$$

Now consider the divergence of the vector ρv , evaluated at the location of particle i :

$$[\nabla \cdot (\rho v)]_i \equiv (\rho \nabla \cdot v)_i + (v \cdot \nabla \rho)_i .$$

From this we get an expression for $\rho \nabla \cdot v$, which needs to be equal to $-\dot{\rho}$ in order to be consistent with the continuity equation :

$$\rho_i (\nabla \cdot v)_i = [\nabla \cdot (\rho v)]_i - v_i \cdot (\nabla \rho)_i \stackrel{!}{=} -\dot{\rho} .$$

Writing out the smooth-particle forms for the two terms on the righthand side of the $\rho_i(\nabla \cdot v)_i$ expression reproduces the desired equality :

$$(\rho \nabla \cdot v)_i = [\nabla \cdot (\rho v)]_i - v_i \cdot (\nabla \rho)_i =$$

$$\nabla_i \cdot \sum_j m_j w_{ij} v_j - v_i \sum_j m_j \cdot \nabla_i w_{ij} = -\sum_j m_j v_{ij} \cdot \nabla_i w_{ij} \equiv -\dot{\rho}_i \ .$$

Evidently the smooth-particle comoving density change, $\dot{\rho}_i$ for particle *i*, is given by the (smooth-particle version of the) continuum expression :

$$\dot{\rho}_i \equiv -\rho_i (\nabla \cdot v)_i$$
.

This equality makes it possible to *evolve* the density of every particle, beginning with specified initial values (usually chosen equal to the ordinary bulk density) and then solving the N ordinary differential equations for $\{\dot{\rho}_i\}$. This differential-equation approach is often preferable to the summation approach, $\rho_i \equiv \sum_j m_j w_{ij}$ in problems with permanent free surfaces, for

81

ws-book9x6

otherwise particles near a system boundary would have sums considerably smaller than bulk particles. On the other hand, this differential-equation approach can become problematic when fracture and penetration blur or change the distinction between surface and bulk particles. Additionally, the integral of the mass distribution can be quite different to $\sum_j m_j$. The differential equation approach to defining density results in an additional "surface mass" $\simeq \rho h A$, where A is the surface area.

3.8 Evaluating the Spatial Derivatives $\{\nabla \rho, \nabla \cdot P, \nabla \cdot Q\}$

Not only mass, but also linear momentum and energy, can be exactly conserved by SPAM. The desirability of satisfying these conservation laws motivates the usual choices of algorithms for solving the equation of motion and the energy equation in SPAM. Both these equations involve gradients of the pressure tensor P and heat flux vector Q. Let us return to the representation of spatial gradients using SPAM as was discussed briefly in Section 3.5. Starting with the $f_0(r)$ definition,

$$f(r)\rho(r) = f_r \rho_r \equiv \sum_j m_j f_j w(r-r_j) = \sum_j m_j f_j w_{rj} ,$$

implies that the gradients of both expressions are likewise identical :

$$abla[f(r)
ho(r)] \equiv \sum_j m_j f_j
abla w(r-r_j) \; .$$

It is important to see that neither m_j nor f_j is affected by the gradient operator ∇ . Because the $\{m_j\}$ and $\{f_j\}$ are themselves *particle* properties rather than spatial averages, the gradient operator affects only the weight functions $\{w(r - r_j) = w_{rj}\}$ through their explicit dependence on the distances separating the field point r from the particle locations $\{r_j\}$.

It is likewise important to note that choosing $f_0 = 1/\rho$, or $f_1 = 1$, or $f_2 = \rho$, is a particularly *bad* choice because the corresponding gradient recipe :

$$abla(f_0 \rho) \text{ or } \nabla f_1 \text{ or } \nabla(f_2 / \rho) = \nabla(1) =$$

$$\sum (1/\rho_j) m_j \nabla_r w_{rj} \neq 0 ,$$

does *not* generally vanish, as the gradient of any constant must. The smooth particle approach, resting as it does on intuitive definitions and assump-

SPAM Equation of Motion and Energy Equation

tions, requires judgment to avoid pitfalls. We will see that particular choices of f are appropriate to *fluxes* (like heat flux and momentum flux) and that different choices are appropriate for *forces* (like the temperature gradient inducing heat flow and the pressure gradient inducing momentum flow).

3.9 SPAM Equation of Motion and Energy Equation

The choice, $f_0 \equiv (P/\rho^2)$, or $f_1 \equiv (P/\rho)$, or $f_2 \equiv P$, can be used to derive a set of smoothed-particle equations of motion giving *exact* momentum conservation. To see this, we first evaluate the divergence of (P/ρ) in the usual continuum way, following the rules of ordinary calculus :

$$\nabla \cdot (P/\rho) \equiv -(P/\rho^2) \cdot \nabla \rho + (1/\rho) \nabla \cdot P$$
.

We can rearrange this identity to provide a useful form of the continuum motion equation, $\dot{v} = -(1/\rho)\nabla \cdot P$, for the time evolution of the comoving velocity at the point r, v(r):

$$\dot{v} = -(P/\rho^2) \cdot \nabla \rho - \nabla \cdot (P/\rho)$$
.

Using the smooth-particle gradient definition just given, the gradients in the evolution equation, $\nabla \rho$ and $\nabla \cdot (P/\rho)$, can both be replaced by equivalent simple smooth-particle sums :

$$\nabla \rho = \sum_{j} m_{j} \nabla w(r - r_{j}) ;$$
$$\nabla \cdot (P/\rho) = \sum_{j} m_{j} (P/\rho^{2})_{j} \cdot \nabla w(r - r_{j})$$

The density-gradient found here is of course the same one discussed in Section 3.7. If we replace r by r_i in these two gradient definitions, we get a smooth-particle equation of motion for the comoving evolution of Particle i's velocity. To emphasize the antisymmetry of each pair's contributions to the motion equations, we write the equation of motion in terms of the time-rate-of-change of momentum :

$$m_i \ddot{r}_i = m_i \dot{v}_i \equiv m_i \frac{dv_i}{dt} = -\sum_j m_i m_j [(P/\rho^2)_i + (P/\rho^2)_j] \cdot \nabla_i w(r_i - r_j) .$$

Note that $\nabla_i w(r_i - r_j) + \nabla_j w(r_i - r_j)$ is exactly zero. As a consequence, the two contributions made by each $\{i, j\}$ pair of particles to the *total*

linear-momentum-change sum, $\sum m_i \dot{v}_i$, according to the smooth-particle equations of motion, sum to zero. Thus the total system momentum is conserved *exactly*.

It must be admitted that this "derivation" of the smooth-particle motion equation has a suspicious aspect. The velocity v_i of the particle at r_i does not often coincide with the fluid velocity there, $v(r_i)$. It is likewise unlikely that the comoving derivative \dot{v}_i would coincide with $\dot{v}(r_i)$.

Despite this lack of rigor, any convinced reader should be able to construct a very similar argument to confirm that the continuum *energy* equations can likewise be written in a plausible smooth-particle form which conserves total energy *exactly* :

$$m_i \dot{e}_i = -\sum_j m_i m_j [(P/\rho^2)_i + (P/\rho^2)_j] : \frac{1}{2} (v_j - v_i) \nabla_i w (r_i - r_j) - \sum_j m_i m_j [(Q/\rho^2)_i + (Q/\rho^2)_j] \cdot \nabla_i w (r_i - r_j) .$$

There are three steps. First, the heat-flux divergence needs to be written in a symmetric pair-sum form, just as was the divergence of the pressure tensor. Second, the antisymmetry of the derivatives ,

$$\nabla_i w_{ij} + \nabla_j w_{ij} \equiv 0 \; ,$$

guarantees that the summed-up heat-flux contributions to total energy change vanish. Last, it needs also to be established that the mechanical energy changes in the sums $\sum m_i \dot{e}_i$ and $\sum m_i (d/dt)(v_i^2/2) = \sum m_i \dot{v}_i \cdot v_i$ precisely cancel.

3.10 Rezoning; Does Particle Size Matter?

It is important to recognize that the particle size used in SPAM is arbitrary. Any convenient choice will do. Not only can small and large samples be modeled, but also a fixed-size sample can be modeled with particles of different sizes. To illustrate that the SPAM equations are consistent with refining or coarsening the particle mesh consider the effect of replacing each SPAM particle of mass m with two particles of mass m/2. Evidently the density at any point in space remains exactly the same. The density sum

Ideal-Gas Isomorphism with SPAM

at any location r,

$$\rho(r) = \sum_{j} m_j w(r - r_j) \; ,$$

is replaced by a sum with twice as many terms (because two particles now occupy each former particle location), each equal to half its corresponding mass-m predecessor (because each particle mass is half as large).

The smooth-particle equation of motion ,

$$\dot{v}_i = -m \sum_j \left[\left(\frac{P}{\rho^2} \right)_i + \left(\frac{P}{\rho^2} \right)_j \right] \nabla_i w_{ij} ,$$

likewise has *twice* as many terms contributing to the sums, but with each exactly *half* the size of its predecessor. Thus the dynamics of N pairs of $2N \mod m/2$ particles is *identical* to that of N particles of mass m.

Rezoning can be accomplished easily with smooth particles. In the event that more detail is required in a particular region, any SPAM particle can be replaced by two or more smaller ones, choosing the new masses, velocities, and energies so as to reproduce the old mass, momentum, and energy. Similarly, if a region has too many particles, any two of them can be combined to make a more massive and energetic single particle.

The method can be extended in many ways to treat special situations. Chemical reactions can be introduced. Two applications to astrophysics are mentioned in more detail in Chapter 10 . Electromagnetic fields can be included by using tree techniques to evaluate the effect of long-range forces. As is usually the case, the major advances in numerical simulation methods have resulted from the desire to simulate challenging problems.

3.11 Ideal-Gas Isomorphism with SPAM

The motion equation derived in Section 3.9 takes on a particularly simple and thought-provoking form for the ideal-gas polytropic equation of state $P \propto \rho^2$. In this case the smooth particle equations of motion ,

$$m_i \dot{v}_i \equiv m_i \frac{dv_i}{dt} = -\sum_j m_i m_j [(P/\rho^2)_i + (P/\rho^2)_j] \cdot \nabla_i w(r_i - r_j) ,$$

become the familiar Newtonian equations of motion for molecular dynamics:

$$m_i \dot{v}_i \propto -\sum_j \nabla_i w(r_i - r_j) \;,$$

so that the weight function plays the rôle of a Newtonian pair potential. This correspondence is explored in detail in Chapter 7 .

There is a subtle difference: the weight function maximum, $w_{ii} = w(0)$, makes the *largest self-density contribution*, mw(0), to the smooth-particle density ρ_i . In molecular dynamics the total potential energy of the system generally has no self-energy contributions $\phi(r = 0)$. In fact, many of the potentials used in molecular dynamics diverge at r = 0. This apparent difference makes no contribution whatever to the correspondence in the *derivatives* of w and ϕ , so that the trajectories and their time derivatives (coordinates, velocities, and accelerations) can all be made to correspond exactly for the equation of state $P \propto \rho^2$.

Evidently the same correspondence holds true in the event that a shear is imposed on the system, such as a simple shear flow. This nonequilibrium isomorphism suggests an interesting paradox. We expect that an *ideal* gas (such as the polytropic gas with $P \propto \rho^2$) has vanishing transport coefficients. On the other hand Green and Kubo's application of perturbation theory to Gibbs' statistical mechanics establishes that the viscosity and heat conductivity of a gas with a potential function (such as w) are given by time-correlation integrals of stress and heat flux fluctuations. Explicitly, Green and Kubo show that the shear viscosity coefficient η is given by the decay of shear stress fluctuations :

$$\eta kT/V = \int_0^\infty \langle P_{xy}(0) P_{xy}(t) \rangle dt \; .$$

The angular brackets here indicate a large-system equilibrium average, where the mean value of P_{xy} vanishes. Figure 3.5 shows the equilibrium stress autocorrelation function $\langle P_{xy}(0)P_{xy}(t) \rangle$ for 1024 Lucy-potential particles.² It is clear that the corresponding viscosity is nonzero even though the isomorphism just described suggests that the viscosity should vanish. We will analyze the resolution of this paradox by carrying out shear flow simulations for an ideal gas, as an illustrative problem, in Section 5.9.

²Posch, Hoover, and Kum (1995).



Evaluating the Spatial Derivatives $\{\nabla v, \nabla T\}$

3.12 Evaluating the Spatial Derivatives $\{\nabla v, \nabla T\}$

In Section 3.9 we evaluated the divergences of the stress tensor and of the heat flux vector by using the symmetric sums [including $(P/\rho^2)_i + (P/\rho^2)_j$ and $(Q/\rho^2)_i + (Q/\rho^2)_j$] from the f_2 average of Section 3.5. Smooth-particle formulæ for the velocity and temperature gradients, which contribute to the Newtonian pressure tensor and to Fourier's heat flux in viscous heat-conducting fluids, *could* be derived the same way. But it is far better to use the alternative *unsymmetric* sums [including $v_i - v_j$ and $T_i - T_j$] for these gradients, based on the f_0 average of Section 3.5. Both ∇v and ∇T are best evaluated in this same way.

Let us illustrate the approach to the velocity and temperature gradients by considering explicitly the velocity gradient tensor ∇v . Begin with the identity from ordinary calculus :

$$\nabla(v\rho) = \rho \nabla v + v \nabla \rho \; .$$

This provides a useful expression for the velocity gradient tensor :

$$\rho \nabla v = \nabla (v\rho) - v \nabla \rho$$

Now apply the usual f_0 smooth-particle representations of the two gradients on the right, with the result :

$$(\rho \nabla v)_r = \sum_j m_j v_j \nabla_r w(r-r_j) - v_r \sum_j m_j \nabla_r w(r-r_j) \ .$$

If we choose to evaluate the gradients at the location of particle i (that is, with $r \rightarrow r_i$) the two sums on the righthandside can be combined :

$$(\rho \nabla v)_i = \sum_j m_j v_j \nabla_i w(r_i - r_j) - v_i \sum_j m_j \nabla_i w(r_i - r_j) = -\sum_j m_j v_{ij} \nabla_i w_{ij} ,$$

where v_{ij} is the relative velocity, $v_i - v_j$. These expressions for the gradients can then be *symmetrized* by using a mean density, either arithmetic or geometric, for the density appearing on the lefthandside. The symmetrized density ρ_{ij} could be, for instance :

$$\rho_{ij} = \text{ either } \frac{1}{2}(\rho_i + \rho_j) \text{ or } \rho_{ij} = \sqrt{\rho_i \rho_j} .$$

The corresponding expressions for the temperature and velocity gradients at the location of Particle i then *guarantee* that the gradient contributions of each pair of particles are proportional to the corresponding temperature and velocity differences :

$$\nabla T \longrightarrow \sum_{j} m_{ij} [T_j - T_i] \nabla_i w_{ij} / \rho_{ij} ;$$
$$\nabla v \longrightarrow \sum_{j} m_{ij} [v_j - v_i] \nabla_i w_{ij} / \rho_{ij} .$$

The symmetrized mass m_{ij} can likewise be chosen as an arithmetic or geometric mean, with the combination m_{ij}/ρ_{ij} mass-independent, as one should expect on physical grounds.

von Neumann-Richtmyer Artificial SPAM Viscosity

To illustrate that these gradient expressions become exact in the manyparticle uniform-density limit suppose that the x velocity component varies linearly with y (say $v_x = \dot{\epsilon}y$) and compute $\nabla_y v_x$ using integration by parts:

$$\nabla_y v_x \simeq -\int_0^h \int_0^{2\pi} \dot{\epsilon} \frac{y^2}{r} w' r dr d\theta = -\dot{\epsilon} \int_0^h \int_0^{2\pi} r \sin^2(\theta) w' r dr d\theta =$$
$$-\dot{\epsilon} \pi \int_0^h r^2 w' dr = +\dot{\epsilon} \int_0^h 2\pi r w dr \equiv \dot{\epsilon} .$$

This establishes the convergence of the smooth-particle approach in the (unachievable, and hence rather unrealistic) limit of many homogeneouslydistributed particles.

Evidently the formulæ for ∇v and ∇T can be used to describe Newtonian viscosity or Fourier conductivity. In the case of an idealized "Euler fluid", with vanishing viscosity and conductivity, the pressure is purely hydrostatic and is a scalar function of density and energy without any viscous or plastic contributions. The heat flux also vanishes for an Euler fluid. So far we have not restricted the pressure tensor P and the heat flux vector Q in any way. These momentum and energy fluxes can depend upon velocity or temperature gradients, elastic or plastic strains, in addition to the equilibrium thermodynamic dependence on density and energy.

Now that we can evaluate both kinds of gradients, those needed for conservation of momentum and energy ,

$$\nabla \cdot P$$
 and $\nabla \cdot Q$,

and those driving the comoving fluxes of the conserved quantities :

$$\nabla v$$
 and ∇T ,

we have available nearly all of the tools necessary to solve example SPAM problems analogous in difficulty to those example problems considered for ordinary continuum mechanics in Chapter 2. Before turning to those examples we consider one more generally-useful ingredient, the smooth-particle analog of von Neumann and Richtmyer's artificial viscosity.

3.13 von Neumann-Richtmyer Artificial SPAM Viscosity

The tendency of pressure waves to steepen into shockwaves is illustrated, through an example problem, in Section 7.9 . Once the waves become too

steep to be resolved by the computational mesh, instability is the result. To combat this problem von Neumann and Richtmyer suggested using the "artificial viscosity" mentioned in Section 2.6.³ Artificial viscosity is many orders of magnitude larger than the true viscosity. Artificial viscosity is a purely numerical device designed to stabilize numerical methods by spreading shockwaves over several computational zones. In the case of smooth particles this means spreading a shockwave over several particle diameters.

von Neumann's original proposal was to add a viscous contribution to the pressure in any region of the Eulerian or Lagrangian mesh where the flow was contracting, $\nabla \cdot v < 0$, or equivalently, $\dot{\rho} > 0$. Either form,

$$\Delta P_1 = -\rho h c \nabla \cdot v = \dot{\rho} h c \text{ or } \Delta P_2 = +\rho h^2 (\nabla \cdot v)^2 = \dot{\rho}^2 h^2 / \rho ,$$

is positive in a contracting flow, and could be used to increase the shockwidth from the microscopic value to the macroscopic particle "size" h. von Neumann and Richtmyer recommended using the sum of these two artificial viscous pressures, $\Delta P_1 + \Delta P_2$, with the mesh spacing dx replacing the smooth-particle range h. As is usual, there are *many* ways to implement this continuum idea in smooth particle applied mechanics. Let us consider two possibilities.

One can easily evaluate the strict analog of the artificial viscosity, directly from the smooth-particle density change ,

$$\nabla \cdot v \equiv -\dot{\rho}/\rho \; ,$$

evaluated at the location of each particle. Every particle then has an extra hydrostatic force calculated from the viscous contributions { ΔP } :

$$\begin{split} m_i \ddot{x}_i &\longrightarrow m_i \ddot{x}_i - \sum_j m_i m_j (w'x/r)_{ij} [(\Delta P/\rho^2)_i + (\Delta P/\rho^2)_j] ; \\ m_i \ddot{y}_i &\longrightarrow m_i \ddot{y}_i - \sum_j m_i m_j (w'y/r)_{ij} [(\Delta P/\rho^2)_i + (\Delta P/\rho^2)_j] . \end{split}$$

Alternatively, one could equally well consider the contributions made by an ij particle pair to the particle compressions, adding corresponding contributions to the forces acting on the two interacting particles. With this approach ΔP_{ij} for each particle *pair* (as opposed to ΔP_i for each particle), where the relative velocity is negative (so that the particles are getting

 $^{^{3}}$ von Neumann and Richtmyer (1950).

nearer to one another), would be :

$$\Delta P_{ij} = m_{ij}hc(v_{ij} \cdot r_{ij})(w'/r) + (m_{ij}^2h^2/\rho_{ij})(v_{ij} \cdot r_{ij})^2(w'/r)^2 .$$

It might be thought that a tensor version of the artificial viscosity should be used for solids, as opposed to the fluid version considered here. For smooth particles this would suggest an addition to the equation of motion of the form :

$$\dot{v}_i \propto \sum_j (v_j - v_i) \; .$$

But unless the relative velocity $v_j - v_i$ is parallel to $r_j - r_i$ an artificial acceleration of this form cannot properly conserve angular momentum. We return to this important topic in Section 8.9. Let us now turn to applying SPAM techniques to two simple stationary-state problems, adiabatic and isothermal atmospheric equilibria.

3.14 Example: Adiabatic Atmospheric Equilibrium

Consider the simple gas-phase adiabatic equation of state,

$$P = \frac{B_0}{2} \left(\frac{\rho}{\rho_0}\right)^2 \; ,$$

discussed in Section 3.11 . For this equation of state SPAM and molecular dynamics coincide. For simplicity we choose the constants (B_0, ρ_0, m) all equal to unity. Now let the pressure P and the density ρ vary with the vertical coordinate y and consider the mechanical equilibrium of an element dxdy of gas with mass $\rho dxdy$ in the presence of a gravitational field $-g_y$. The equilibrium force-balance equation ,

$$-\frac{dP}{dy}dxdy - \rho g_y dxdy = 0 \longrightarrow$$

$$\frac{dP}{d\rho}\frac{d\rho}{dy} = \frac{2P}{\rho}\frac{d\rho}{dy} = -\rho g_y \longrightarrow \frac{d\rho}{dy} = -g_y ,$$

has the solution :

$$\rho = \rho_{\max} - yg_y = \rho_{\max} - yg_y \left(\frac{\rho_0^2}{B_0}\right) ,$$

91

ws-book9x6

where ρ_{\max} is the density at the "bottom" of the atmosphere, y = 0. The density vanishes at the "top", $y = 1/g_y$. If we start out with an $n \times n = N$ square of field-free unit-density gas, a field of $g_y = 2/n$ should provide an equilibrated square of gas with density 2 on the bottom and density zero at the top: $\rho = 2[1 - (y/n)]$.



Figures 3.6 and 3.7 compare the computed particle positions and the corresponding density profiles reached by systems of 400 and 1600 smooth unit-mass Lucy particles in a periodic box of width \sqrt{N} where the gravitational field strength is $2/n = 2\sqrt{1/N}$. Initial coordinates were chosen randomly within a rectangle, with width \sqrt{N} and height $2\sqrt{N}$, with vanishing initial velocities. The equations of motion were solved for 40,000 timesteps (dt = 0.05), and included both viscous damping and a very smooth and short-ranged repulsive pair core potential :

$$\{ F_{\rm damp} = -\frac{mv}{\tau} \} ; \ \tau \equiv 200 ;$$

$$\Phi_{\rm core} = \sum_{|r_{ij}| < \sigma} 10 \left[1 - \left(\frac{r_{ij}^2}{\sigma^2} \right) \right]^4 \; ; \; \sigma^2 = \frac{4}{5} \; .$$

"Mirror boundary conditions" (described more fully in Sections 5.4 and

Example: Adiabatic Atmospheric Equilibrium

5.10) were applied at the bottom and periodic boundaries were applied at the sides. The core potential was used to discourage overlaps during equilibration. For other examples of its use see Sections 5.3, 7.7, and 8.2.

To compute the one-dimensional density $\rho(y)$ Lucy's $\mathit{one}\text{-dimensional}$ weight function ,

$$w = \frac{5}{4h} [1 - 6r^2 + 8r^3 - 3r^4] \ ; \ r \equiv \frac{|y|}{h} \ h = 3 \ ,$$

was used. The computed density profiles agree well with their theoretical counterparts, an approximately *linear* variation of density with height.



94

Smooth Particle Methods

3.15 Example: Isothermal Atmospheric Equilibrium

Consider once again the gravitational field, $-g_y$, with a mechanical "mirror boundary" at y = 0. We choose a weaker field strength, $g_y = \sqrt{1/N}$, and let the pressure vary *linearly* with density, as it *does* in the case of an *isothermal* ideal gas :

$$\frac{P}{P_0} = \frac{\rho e}{\rho_0 e_0} = \frac{\rho}{\rho_0} \; ; \; e = e_0 = kT_0/m \; ,$$

and as it *would* for an isentropic adiabatic gas with an artificial energy equation :

$$e = \ln \rho \longleftrightarrow P = \rho^2 (\partial e / \partial \rho)_S = \rho$$
.

By considering the continuum dynamics of this artificial adiabatic gas (which mimics the dynamics of a heat-conducting ideal gas with *infinite* heat conductivity) we avoid the need to include shear viscosity and heat conductivity in the smooth-particle equations. The artificial internal energy per unit mass, $e = \ln \rho$, provides a useful numerical check for the evolution algorithm by providing an exact energy conservation law in the absence of any damping, as is detailed below.

For simplicity we choose the constant temperature T_0 , Boltzmann's constant k, and the particle mass m all equal to unity, along with P_0 and ρ_0 . We again use a pair core repulsive potential to avoid overlaps during equilibration :

$$\phi_{core}(r < \sigma) = \epsilon \left[1 - \left(\frac{r^2}{\sigma^2}\right)\right]^4 \; ; \; \epsilon = 10 \; ; \; \sigma = \frac{1}{2} \; .$$

We can reach a stationary { $v_i = 0$ } equilibrium by including also the nonconservative damping force :

$$(F_{\text{damp}}/m)_i \equiv -v_i/\tau ; \tau = 200$$
.

At equilibrium the macroscopic force-balance equation is :

$$\frac{dP}{dy} = \frac{dP}{d\rho}\frac{d\rho}{dy} = d\rho/dy = -\rho g_y \longrightarrow$$
$$\rho = \rho_0 \exp(-yg_y) = \exp(-y/\sqrt{N}) \ .$$

(1

Example: Isothermal Atmospheric Equilibrium

This problem, unlike the adiabatic atmosphere considered in the previous Section, does *not* reduce to a molecular dynamics problem. The isothermal equilibrium requires that the smooth-particle equations of motion be solved:

$$\begin{split} \rho_i &= m \sum_j w_{\text{Lucy}} (r_{ij} < h = 3) \ ; \\ \dot{x}_i &= (v_x)_i \ ; \ \dot{y}_i = (v_y)_i \ ; \\ (\dot{v}_x)_i &= \left(\frac{F_{\text{core}}^x + F_{\text{damp}}^x + F_{\text{mirror}}^x}{m}\right)_i - \sum_j \left[\frac{1}{\rho_i} + \frac{1}{\rho_j}\right] (xw'/r)_{ij} \ ; \\ \dot{v}_y)_i &= \left(\frac{F_{\text{core}}^y + F_{\text{damp}}^y + F_{\text{mirror}}^y}{m}\right)_i - \sum_j \left[\frac{1}{\rho_i} + \frac{1}{\rho_j}\right] (yw'/r)_{ij} - g_y \end{split}$$

In the absence of any explicit damping forces, the total energy,

$$E_{\text{Total}} = E_{\text{Kinetic}} + \sum_{r_{ij} < \sigma} \epsilon \left[1 - \left(\frac{r^2}{\sigma^2} \right) \right]^4 + \sum_i \left[\ln(\rho)_i + \frac{y_i}{\sqrt{N}} \right] + \Phi_{\text{mirror}} ,$$

is a constant of the motion. The "mirror" terms refer to interactions near the mirror boundary y = 0. Any particle with y < h/2 = 1.5 makes a "self contribution" w(2y) to its density. Further, any particle with $y < \sigma/2 = 0.25$ makes a "self contribution" to the total potential energy, $2\phi_{\rm core}(2y)$. Pairs of particles close enough for their images to interact across the mirror must be accounted for too. Given this complexity, the availability of an energy check of the dynamics (in the absence of damping) is welcome.

The only additional simplifying feature in the dynamics is the absence of the energy equations for { e_i } (because the internal energy, per unit mass, is actually constant in this isothermal problem, corresponding to the artificial $e = \ln \rho$ relation). Here we display the particle positions and the smooth-particle density contours for a 1600-particle SPAM simulation. The density field is in quite satisfactory agreement with the theoretical one $\rho = \exp(-y/\sqrt{N})$ as shown in Figure 3.8.
Smooth Particle Methods



Figure 3.8: Particle positions and smooth-particle density contours compared to theoretical contours [$\ln(\rho/\rho_0) = -yg$ for 0 < yg < 2] for 1600 particles. The gravitational field strength for an N-particle system is $\sqrt{\frac{1}{N}}$. This simulation was equilibrated for a time of 5000, using a timestep dt = 0.01 and $\tau = 200$. The kinetic energy was reset to zero at times that were multiples of 500: { t } = { 500, 1000, 1500, ..., 5000 }.

References

The Example Problems considered in this Chapter illustrate the behavior of " γ -law" gases (with $P \propto \rho^{\gamma}$). Adiabatic motions of our earth's atmosphere can be described in this same way by using $\gamma = 1.4$.

3.16 References

- M. L. Wilkins, "Finite Difference Scheme for Calculating Problems in Two Space Dimensions and Time", Journal of Computational Physics 5, 406-414 (1970).
- (2) H. A. Posch, W. G. Hoover, and O. Kum, "Steady State Shear Flows via Nonequilibrium Molecular Dynamics and Smooth Particle Applied Mechanics", Physical Review E 52, 1711-1720 (1995).
- (3) J. von Neumann and R. Richtmyer, "A Method for the Numerical Calculation of Hydrodynamic Shocks", Journal of Applied Physics 21, 232-257 (1950).

August 16, 2006 17:44

WSPC/Book Trim Size for 9 in x 6 in ws-book9x6

98

Smooth Particle Methods

Chapter 4

Computer Programming

/ FORmula TRANslation languages / Designing a SPAM Program / Runge-Kutta Integration with Fortran and C / A Useful Portable Random Number Generator / Graphic Displays and Analysis / Debugging Tools— Finding Errors / Parallel Computing / Partitioning and Boundary Conditions / Message Passing / Decomposition and Contact Algorithms /

/ 14 Figures /

Example Problems :

[Concentric Annuli Undergoing Rotation, Free Expansion Problem, The Crushing of an Elastic-Plastic Sheet, Caricature of a Billiard Table]

4.1 Summary

Most of the time spent in "computer" simulation is actually "human" time devoted to programming, translating the differential equation structure of physics to the finite-difference structure of computation, and finally, to analyzing and understanding the results. Simply implementing doubleprecision arithmetic on an unfamiliar system, or learning to use yet another graphics package for generating postscript plots, are recurring time sinks. This necessary time can be reduced by dividing the programming task into parts and checking the individual parts before they are combined. This division has the fringe benefit that the parts can often be reused in developing other computer programs. Computational checks should first be performed

Computer Programming

on the smallest feasible problems. Debugging tools are very useful for finding errors in larger problems. Graphics (or "visualization") software makes it possible to plot field variables and contours and to produce animations. Parallel computing provides an opportunity to run longer, larger problems.

4.2 FORmula TRANslation languages

Students today have many choices of computer languages to use for implementing numerical algorithms. What are the issues that must be addressed in selecting a *b*est language for their work? To answer this question we briefly consider relevant characteristics of the programming languages, language styles and more importantly, the ease with which mathematics and numerical methods can be translated into usable computer instructions.

A computer language consists of a collection of statement types and a grammar that restricts the ordering and actions implied by the statements. Each statement type is translated into specific computer instructions for assigning memory to variables and arrays, performing arithmetic operations, reading and writing files, and controlling the flow of the statements that are executed as a program runs. Each language has its own statement syntax and associated rules. The language must define variables, along with their type and precision, provide for operations on the variables, and govern the flow of the execution. The following examples illustrate the primary syntactical differences between the Fortran and C language statements. The C language name indicates only that its predecessors were the now defunct "A" and "B" languages. References, 1 and 2 for Fortran; 3 and 4 for C, are listed at the end of the Chapter. Many more can be found on the internet.

Fortran 77 Program

- c Fortran statements begin in column 7.
- c Comment lines are indicated with a c or a C in column 1.
- c A Fortran rule is that a name starting with
- c i through n indicates an integer.
- c An "implicit double precision" statement defines
- c floating point variables with the precision of
- c two computer words. "Single precision"
- c variables have the precision of one computer word.
- c Continuation symbols (such as the & below) are
- c located in column 6.

FORmula TRANslation languages

```
program example
      implicit double precision (a-h,o-z)
      dimension a(4,4),b(4,4)
      do i=1,4
          j = 1
          a(i,j)= i*1.0d00
          b(i,j)= i*1.0d00
          write(6,10)i,j,a(i,j),b(i,j)
          do j=2,4
              a(i,j) = i + 2*j - 2.0d00
              b(i,j) = b(i,j-1) + 1.0d00
              write(6,10)i,j,a(i,j),b(i,j)
10
              format("integers i and j = ",2i3/
              a,b = ", 2f15.7
   &
          end do
      end do
      stop
      end
```

C Program

```
/* This is a comment line. */
/* C has no column syntax rules. */
/* Indentation in C is used to group a block of C */
/* statements together. */
main(void)
{
  /* type the integer variables */
  int i, j ;
  /* type floating point variables */
  double a[4][4], b[4][4];
  for (i=1; i<=4; i++) {</pre>
    j = 1 ;
    a[i-1][j-1] = i ;
    b[i-1][j-1] = i ;
    printf("integers i and j = %i and %i n",i,j);
    printf("a,b = %15.7f %15.7f \n",a[i-1][j-1],b[i-1][j-1]);
    for(j=2; j<=4; j=j+1) {</pre>
```

Computer Programming

```
a[i-1][j-1] = i + 2*j - 2.0 ;
b[i-1][j-1] = b[i-1][j-2] + 1.0 ;
printf("integers i and j = %i and %i \n",i,j) ;
printf("a,b =%15.7f%15.7f \n",a[i-1][j-1],b[i-1][j-1]);
}
}
```

The Fortran and the C Programs produce identical results:

$$a = \begin{pmatrix} 1 & 3 & 5 & 7 \\ 2 & 4 & 6 & 8 \\ 3 & 5 & 7 & 9 \\ 4 & 6 & 8 & 10 \end{pmatrix} ; \ b = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 2 & 3 & 4 & 5 \\ 3 & 4 & 5 & 6 \\ 4 & 5 & 6 & 7 \end{pmatrix} ,$$

but the syntax and rules leading to these results are different. For example, the syntax of a two-dimensional array variable in Fortran is a(i,j) and in C is a[i][j]. One *significant* difference between Fortran and C is that the array locations in C are indexed starting with 0 rather than 1 as in Fortran. This C indexing differs from the usual scientific notation for array subscripting. Notice that the C coding in this example uses a loop index matching that for the Fortran coding so that the calculations match in both programs. However, the array subscripts in C must all be decremented by 1 because of the array indexing rule in C!!

The C main code begins with the arcane statement,

main(void) ,

rather than the "type" statements required in subroutines (see the C example in Section 4.4). The void in the argument list means that no arguments are expected by the main program. All C statements end in a semicolon. There is no such ending syntax in Fortran. In typical implementations the Fortran compiler simply ignores any symbols appearing beyond column 72. C statement "blocks" (the analogs of Fortran "do-loop" interiors) are sets of C statements enclosed by braces, $\{ \ldots \}$.

In C all variables must have a specified "type". Example types are integer (int), character (char), floating point (float), and double precision floating point (double). Unless otherwise indicated, constants in C, such as 0.10, are assumed to be double precision. The specification const indicates

102

}

FORmula TRANslation languages

a constant which cannot be reset during execution. Such a constant can also be typed. For example, const float bongo is a single precision constant called bongo.

In Fortran 77 (the standard was accepted in 1977) there is a very useful implied naming rule: any variables beginning with the letters i through n are integers while variables beginning with the remaining letters are single-word floating point variables. The implicit double precision statement in Fortran 77 is a rule that incorporates the same naming convention for floating point variables but forces the compiler to use two-word "double precision" for all the variables.

Fortran, the language whose name is derived from FORmula TRANslation, was the earliest language designed specially for scientific computing.^{1,2} The simple Fortran 77 example above illustrates the fact that the language was intended for use by scientists. The syntax was defined to make the statements conform to a notation that closely resembles the equations scientists use. The Fortran 77 language has been used extensively for scientific computing. Compilers that translate Fortran statements into hardware instructions have been developed over the past several decades. As a result, modern Fortran compilers are capable of converting numerical algorithms into an *optimal* set of machine instructions. Similarly, compiler technology has evolved to keep pace with new computer architecture designs which automatically carry out simultaneous parallel operations on arrays ("vectorization" and "pipelining"). All modern Fortran compilers automatically produce instructions for vector and pipelining hardware without requiring the programmer to make any syntax changes.

Over the last fifty years additional languages such as Cobol, ADA, Pascal, and C++ have been developed for other purposes, such as operating system development, graphics and interactive screen display (both C and C++), business applications (Cobol and ADA), and research and teaching in computer science (Pascal). Of these, C is the most useful for scientific computing.^{3,4} Fortunately, during the evolution of computer languages, United States government agencies (such as the Department of Energy, the National Science Foundation, and the Department of Defense) have sponsored language committees to define computer language standards. The motivation for the standardization of languages is to provide *portability*

¹McCracken and Salmon, [Fortran] Computing for Engineers & Scientists (1990).

²Chapman, Fortran for Scientists and Engineers (1997).

³Kernighan and Ritchie, The C Programming Language (1988).

⁴Kochan, Programming in C (2004).

Computer Programming

for computer programs. A "portable" program can be run on computers provided by any computer vendor. One of the many issues affecting portability is the precision of floating point numbers. Because error analysis and stability play such an important role in understanding numerical results, computational precision is a significant consideration in designing a program compatible with more than one computer system.

In 2006 there are only two popular languages for scientific computing— Fortran and C. Both these languages are mature. The first C standard was accepted in 1989. By now, each of the two languages has adopted the best features of the other so as to optimize computational efficiency. Input-output is still a shortcoming of C. The Fortran syntax, with complete symmetry between **read** and **write** statements, is far simpler. No doubt a future C-language standard will incorporate improved input-output syntax.

The original C language syntax designed to support operating systems was extended to include arrays. The Fortran standards have added character variables for manipulating strings and structures for grouping data types. Fortran compilers still have the best and simplest syntax for arrays as well as the most efficient compilers. C has the advantage of providing bit and byte (eight binary "bits" are one "byte") manipulation needed for operating system programming. Both languages have matured to include object-oriented programming syntaxes, rules, and methodologies. "Object-oriented" languages are intended to produce modular programs, with self-contained modules that can be checked and debugged independently of the rest of the program. The C version of object-oriented syntax is contained in the C++ language. The object-oriented syntax for Fortran has been included in all Fortran standards starting with Fortran 90/95 .

Is it necessary, or even desirable, to use object-oriented language? For the problems described in this book the answer is "No". Object-oriented programming languages have been developed to accommodate teams of programmers working on large software projects (from fifty thousand to over a million lines of source code). These languages enforce a programming style in order to promote error detection at the compilation stage rather than at execution time. They incorporate additional syntax which enhances the reuse of source coding. The disadvantages of object-oriented languages are (i) the complexity of the rules and syntax and (ii) the extra keystrokes required to specify the "types" (integer, floating point, character) of all the variables in the program. The cost of these disadvantages is increased program development time.

For scientific programming, Fortran is more straightforward and read-

Designing a SPAM program

able, as well as significantly shorter—the relative complexity of C is responsible for the annual "International Obfuscated C Code Contest".⁵ Recent computational-efficiency comparisons of standard Fortran and C programs with their object-oriented analogs show that object-oriented programs are currently about two times slower. Otherwise the speed of a scientific program today depends primarily on processor speed. Storage, and array sizes, are no longer an imporant constraint on computational efficiency.

Our own prototypical use of a programming language is as a tool for developing relatively "short" research programs (less than 2000 lines). Typically we write new Fortran programs to develop results for each project/publication. As described in the next Section, the fourth-order Runge-Kutta integrator and other parts of our programs are reusable for subsequent studies. Therefore it is natural for us to choose a standard notation and style in our illustrative programs. We use Fortran 77 in the examples that follow, but include an illustrative comparison, in Section 4.4, of the Fortran and C versions of a simple dynamic harmonic-oscillator problem so as to give a concrete example of the syntactical differences between the two languages.

4.3 Designing a SPAM program

A research code for smooth particles can be developed relatively easily. The primary ingredients are (i) a main program; (ii) a subroutine to implement the initial conditions, initial; (iii) a Runge-Kutta time integration routine, rk2 or rk4; and (iv) a routine for evaluating the righthandsides of the smooth particle equations, rhs. The use of subroutines can simplify complex repetitious programming tasks. Typically we have the following structure for a smooth particle program:

```
program SPAM
c This is a template for the main program and its subprograms
c in a smooth-particle program called SPAM
    implicit double precision (a-h,o-z)
    parameter (nx=80,ny=40,n=nx*ny,neq=(9*n))
    dimension x(n),y(n),vx(n),vy(n),rho(n)
    dimension e(n),sxx(n),sxy(n),syy(n)
    dimension p(n),tke(n),pot(n)
```

⁵See http://www.ioccc.org for entry rules and recent winners.

Computer Programming

```
dimension yy(neq),yyp(neq)
    . . .
call initial(x,y,vx,vy,rho,sxx,sxy,syy,e)
    . . .
    . . .
call rk4(dt,yy,yyp)
    . . .
    . . .
stop
end
subroutine initial(x,y,vx,vy,rho,sxx,sxy,syy,e)
parameter (nx=80,ny=40,n=nx*ny,neq=(9*n))
    . . .
return
end
subroutine rk4(dt,yy,yyp)
parameter (nx=80,ny=40,n=nx*ny,neq=(9*n))
    . . .
call rhs(dt,yy,yyp)
    . . .
return
end
subroutine rhs(dt,yy,yyp)
parameter (nx=80,ny=40,n=nx*ny,neq=(9*n))
    . . .
call weight(r,w,wp,h)
    . . .
return
end
subroutine weight(r,w,wp,h)
    . . .
return
end
```

The foregoing example illustrates the five key parts of a smooth-particle pro-

Designing a SPAM program

gram: the main program for incrementing the time step loop, the initial subroutine for setting the problem up, the rk4 subroutine for the fourthorder Runge-Kutta time integration, the rhs subroutine for evaluating the righthandsides of the underlying differential equations, and the weight subroutine. This last routine calculates the weight function which underlies all the smooth-particle spatial representations of physical variables.

In this example the **parameter** statement sets integer values for the problem size, **n**, expressed as the product of the number of particles in the x and y directions, **nx** and **ny** respectively, and the number of differential equations to integrate, **neq = 9*n**.

The parameter statement is extremely handy because it provides a fast and convenient way to change problem size variables without the need to edit all of the dimension statements. An alternative is to replace the parameter statements with an "include file" which contains the parameter statements. That is, replace the parameter statements in the above program with the following line, beginning in column 1 :

#include "params.h"

This statement, with the special symbol #, causes the contents of the include "header file" named params.h to be substituted at this point in the source file. The include-file line is processed by a "precompiler", before the code is compiled. The precompiler syntax # is the same for both Fortran and C. In this example the include file contains the parameter statement. The use of subroutines, parameter statements, and include files all save programming time and effort, and simplify debugging.

The computer program must have a method for identifying physical variables with computer storage locations. Physical variables can be scalars, vectors, or higher-rank tensors. Scalars are stored by identifying their value with a name on the lefthandside of an assignment statement. For instance, the Runge-Kutta timestep is stored in the variable dt and the value in the location is 0.01.

dt = 0.01d00

Coordinates, momenta, accelerations, and heat fluxes are all vectors. These can be stored as one-dimensional arrays :

$$yy(4*j - 3) = x(j)$$

Computer Programming

108

yy(4*j - 2) = y(j) yy(4*j - 1) = vx(j) yy(4*j - 0) = vy(j)

or as two-dimensional arrays :

yy(j,1) = x(j) yy(j,2) = y(j) yy(j,3) = vx(j) yy(j,4) = vy(j)

or as three-dimensional arrays :

yy(i,j,1) = x(i,j) yy(i,j,2) = y(i,j) yy(i,j,3) = vx(i,j) yy(i,j,4) = vy(i,j)

Similarly tensors can be treated by indexing an array with a number of indices appropriate to the rank of the tensor. Storage must be allocated for arrays with a **dimension** statement at the beginning of the routine in which the arrays are used. The Fortran statement,

dimension x(n), vx(n,n), yy(n,n,4)

illustrates the array syntax appropriate in one, two, and three dimensions.

All of the above capabilities give us the tools for designing a smoothparticle program. The four steps shown below are followed in the main program, program SPAM, to implement a basic smooth-particle algorithm. The algorithm provides approximate solutions of the differential evolution equations. These typically include particle values of the density, coordinates, velocity, energy, and stress-tensor components. The righthandside evaluation routine, rhs, is often further subdivided into subroutines which calculate the stress and strain rates, pressure, and other details of material behavior. The subroutine weight, not listed here, is called by almost all of the other subprograms.

 Set up problem size parameters and constants: nx, ny, n, neq, dt, h, itmax, iwrite, iplot Designing a SPAM program

```
2. Set up the initial conditions and plots:
      call initial(x,y,vx,vy)
      call pairs(x,y)
      call density(x,y,rho)
      call energy
c particle plots
      call pplot
c contour plots
      call cplot
  3. Execute the timestep loop: do its = 1,itmax
    a) Store values into the Runge-Kutta arrays:
      yy(4*i - 3) = x(i)
      yy(4*i - 2) = y(i)
      yy(4*i - 1) = vx(i)
      yy(4*i - 0) = vy(i)
    b) Integrate the equations, evaluating the righthandsides four times,
      for fourth-order Runge-Kutta, or twice, for a second-order method.
      call rk4(dt,y,yyp) \longrightarrow call rhs(dt,y,yyp)
    c) Reset the variables:
      x(i) = yy(4*i - 3)
      y(i) = yy(4*i - 2)
      vx(i) = yy(4*i - 1)
      vy(i) = yy(4*i - 0)
    d) Calculate additional results specific to each problem.
    e) Write and plot results:
      if((its/iwrite)*iwrite.eq.its) write(10,20)x,y,vx,vy
      if((its/iplot)*iplot.eq.its) call pplot
      if((its/iplot)*iplot.eq.its) call cplot
```

4. If all the do iterations are completed, end the program.

Good programming style enhances the readability of a computer program and is extremely useful in finding the typing, syntactical, and logical mistakes generated when translating the algorithms into a programming language. There are several key points to consider when writing computer programs:

1. Use enough comment statements to help yourself or a colleague find sections of interest in the program. Comment lines can provide helpful clues to the programmer's intent, to the underlying motivation for the algorithm,

110

Computer Programming

and to the identities of the physical concepts appearing in the program.

2. Line up the equal signs in a set of assignment statements.

3. Use blanks to make programming statements more readable, particularly for long assignment statements. Use continuation lines as needed. The largest number of continuation lines allowed is 19 for Fortran 77 and 39 for Fortran 90.

4. Use blank lines to separate sections of the program.

5. Make use of blanks in assignment statements to line up sets of equations calculating the related physical quantities such as the components of a vector or tensor.

6. Use descriptive names for variables and program units. For instance, name the stress tensor components **sxx**, **sxy**, **syy**, and call the initialization routine **initial**.

These programming style ideas are illustrated in the Fortran 77 example below taken from the righthandside evaluation routine in a typical SPAM program.

```
subroutine rhs(dt,yy,yyp)
      implicit double precision (a-h,o-z)
#include params.h
c store the arguments into local arrays
      do i = 1,n
     x(i)
             = yy(i)
      y(i)
             = yy(i+n)
      vx(i) = yy(i+n+n)
      vy(i) = yy(i+n+n+n)
      rho(i) = yy(i+n+n+n+n)
            = yy(i+n+n+n+n)
      e(i)
      sxx(i) = yy(i+n+n+n+n+n)
      sxy(i) = yy(i+n+n+n+n+n+n)
      syy(i) = yy(i+n+n+n+n+n+n+n)
      end do
```

Designing a SPAM program

create initial values for time derivatives and strain rates

```
do i = 1,n
      xdot(i) = vx(i)
      ydot(i) = vy(i)
      vxdot(i) = 0.0d00
      vydot(i) = 0.0d00
      rhodot(i) = 0.0d00
      edot(i) = 0.0d00
      sxxdot(i) = 0.0d00
      sxydot(i) = 0.0d00
      syydot(i) = 0.0d00
      dvxdx(i) = 0.0d00
      dvxdy(i) = 0.0d00
      dvydx(i) = 0.0d00
      dvydy(i) = 0.0d00
      fx(i)
                = 0.0d00
      fy(i)
                = 0.0d00
      vxbar(i) = vx(i)
      vybar(i) = vy(i)
       end do
calculate rhodot, the strain rate tensor and average
c velocities
      do 25 ip = 1, npairs
      i = ni(ip)
      j = nj(ip)
      xij = x(i) - x(j)
      yij = y(i) - y(j)
      vxij = vx(i) - vx(j)
      vyij = vy(i) - vy(j)
      rr = xij*xij + yij*yij
      if(rr.lt.h*h) then
        r = dsqrt(rr)
        call weight(r,w,wp,h)
        rhobar = 0.5d00*(rho(i)+rho(j))
        rrhobar = r*rhobar
        rhodot(i) = rhodot(i) + wp*(vxij*xij+vyij*yij)/r
```

```
Computer Programming
```

```
rhodot(j) = rhodot(j) + wp*(vxij*xij+vyij*yij)/r
choose Monaghan's method of avoiding overlaps
         vxbar(i) = vxbar(i) + w*(vx(j) - vx(i))/rhobar
         vybar(i) = vybar(i) + w*(vy(j) - vy(i))/rhobar
         vxbar(j) = vxbar(j) - w*(vx(j) - vx(i))/rhobar
         vybar(j) = vybar(j) - w*(vy(j) - vy(i))/rhobar
         smxx = -wp*(vxij*xij)/rrhobar
         smxy = -wp*(vxij*yij)/rrhobar
         smyx = -wp*(vyij*xij)/rrhobar
         smyy = -wp*(vyij*yij)/rrhobar
         dvxdx(i) = dvxdx(i) + smxx
         dvxdx(j) = dvxdx(j) + smxx
         dvxdy(i) = dvxdy(i) + smxy
         dvxdy(j) = dvxdy(j) + smxy
         dvydx(i) = dvydx(i) + smyx
         dvydx(j) = dvydx(j) + smyx
         dvydy(i) = dvydy(i) + smyy
         dvydy(j) = dvydy(j) + smyy
       end if
           . . . .
25 continue
```

4.4 Runge-Kutta Integration with Fortran and C

Algorithms for integrating differential equations can be obtained as commercial packages (such as the NAG, for "Numerical Algorithm Group", and IMSL, for 'International Mathematical Subroutine Library", collections as well as the less-costly software packaged with the several books by Press et alii⁶), Such commercial packages incorporate some severe disadvantages for research work, with loss of control and loss of transparency the price of ease of operation. Understanding, speed of execution, ease of operation, and the ability to control the timestep to facilitate error analyses, are all desirable features of a useful integration algorithm.

 $^{^{6}}Numerical Recipes (1986).$

Runge-Kutta Integration with Fortran and C

The best general-purpose integrator is the self-starting fourth-order Runge-Kutta algorithm. In the following short equivalent programs, one in Fortran and one in C, we use the "classic" fourth-order Runge-Kutta integration algorithm to solve the one-dimensional harmonic oscillator problem.⁷ For a sufficiently small Runge-Kutta timestep dt the coordinate q(t)and momentum $p(t) = \dot{q}$ approach the analytic sinusoidal solution of the two coupled ordinary differential equations of motion:

$$\{\ \dot{q}=+p\ ;\ \dot{p}=-q\ \}\rightarrow\{\ q=\cos t\ ;\ p=-\sin t\ \}\ .$$

In this simple linear case it is also possible to express the result of the fourth-order Runge-Kutta algorithm analytically $:^{8}$

$$q(t = ndt) = \left[1 - \frac{dt^6}{72} + \frac{dt^8}{576}\right]^{n/2} \cos(n\lambda dt) ,$$

where the parameter λ can be obtained from the equation :

$$\tan(\lambda dt) = \frac{\left\lfloor dt - \frac{dt^3}{6} \right\rfloor}{\left[1 - \frac{dt^2}{2} + \frac{dt^4}{24}\right]}$$

In problems with time-dependent boundary conditions the righthandsides of differential equations can depend explicitly on time. Accordingly, we include the time (time) in the list of arguments of both rk4 and rhs.

Fortran 77 Calculation of Harmonic Oscillator Dynamics

```
implicit double precision(a-h,o-z)
parameter(neq=2,itmax=400)
dimension yy(neq),yyp(neq)
dt = 0.10d00
time = 0.00d00
yy(1) = 1.0d00
yy(2) = 0.0d00
do it = 1,itmax
call rk4(dt,yy,yyp,time)
q = yy(1)
p = yy(2)
```

 $^{^7{\}rm For}$ an interesting variant representation of this classic method see page 186 of Lorenz' otherwise fine and stimulating 1993 book, *The Essence of Chaos.*

⁸Hoover, Computational Statistical Mechanics (1991), page 15.

```
114
                                              Computer\ Programming
       energy = 0.5d00*(q**2 + p**2)
       write(6,12) time, q, p, energy
       write(8,12) time, q, p, energy
12
       format(4f12.8," [Time, q, p, Energy] ")
       enddo
       stop
       end
       subroutine rk4(dt,yy,yyp,time)
       implicit double precision (a-h,o-z)
       parameter (neq = 2)
       dimension yy(neq),yyp(neq)
       dimension yy1(neq),yy2(neq),yy3(neq),yy4(neq)
       dimension yp1(neq),yp2(neq),yp3(neq),yp4(neq)
       do 1 ieq = 1, neq
       yy1(ieq) = yy(ieq)
1
       call rhs(dt,yy1,yp1,time+0.0d00*dt)
       do 2 ieq = 1, neq
       yy2(ieq) = yy(ieq) + 0.5d00*dt*yp1(ieq)
2
       call rhs(dt,yy2,yp2,time+0.5d00*dt)
       do 3 ieq = 1,neq
3
       yy3(ieq) = yy(ieq) + 0.5d00*dt*yp2(ieq)
       call rhs(dt,yy3,yp3,time+0.5d00*dt)
       do 4 ieq = 1,neq
4
       yy4(ieq) = yy(ieq) + 1.0d00*dt*yp3(ieq)
       call rhs(dt,yy4,yp4,time+1.0d00*dt)
       do 5 ieq = 1,neq
5
       yyp(ieq) = (1.0d00/6.0d00)*
     &(yp1(ieq)+2.0d00*(yp2(ieq)+yp3(ieq))+yp4(ieq))
       do 30 ieq = 1, neq
30
       yy(ieq) = yy(ieq) + dt*yyp(ieq)
       time = time + dt
```

Runge-Kutta Integration with Fortran and C

```
return
end
subroutine rhs(dt,yy,yp,time)
implicit double precision(a-h,o-z)
parameter( neq = 2 )
dimension yy(neq),yp(neq)
yp(1) = +yy(2)
yp(2) = -yy(1)
return
end
```

We have chosen to use statement labels to end the "do loops" in the Runge-Kutta subroutine of this example Fortran program. The alternative "do ... enddo" or "do ... end do" construction appears in the main code.

Next comes the equivalent C program, where we have omitted the extraneous time-dependence in the arguments of **rk4** and **rhs** in order to save space in the already overly long C code. Notice that it is not necessary to indent the statements in C.

C Calculation of Harmonic Oscillator Dynamics

```
#include <stdio.h>
#include <stdib.h>
#include <stdlib.h>
#include <math.h>
#define neq (2)
#define itmax (400)

double rk4( double dt, double yy[neq], double yyp[neq]) ;
double rhs( double yy[neq], double yyp[neq]) ;
FILE *out1 ;
main (void)
{
    int it ;
    double dt, time, q, p, energy ;
    double yy[neq], yyp[neq] ;
```

```
Computer Programming
```

```
out1=fopen("qpe.txt", "w") ;
printf("neq = %d \n",neq) ;
printf("itmax = %d \n",itmax) ;
dt = 0.10;
time = 0.0;
yy[0] = 1.0;
yy[1] = 0.0 ;
for (it = 0 ; it < itmax ; it++)</pre>
{
  rk4(dt,yy,yyp) ;
   q = yy[0];
   p = yy[1];
   time = time + dt ;
   energy = 0.5*(q*q + p*p);
   printf("%15.5f %15.10f Time and Energy n", time, energy);
  printf("%15.5f %15.10f %15.10f Time,q,p \n", time, q, p);
  fprintf(out1, "%12.8f %12.8f %12.8f %12.8f \n",
   time,q,p,energy) ;
}
fclose(out1) ;
}
double rk4( double dt, double yy[neq], double yyp[neq])
{
double yy1[neq],yy2[neq],yy3[neq],yy4[neq] ;
double yp1[neq],yp2[neq],yp3[neq],yp4[neq] ;
int ieq ;
for (ieq = 0 ; ieq < neq ; ieq++)
  yy1[ieq] = yy[ieq] ;
rhs(yy1, yp1) ;
for (ieq = 0 ; ieq < neq ; ieq++)
   yy2[ieq] = yy[ieq] + 0.5*dt*yp1[ieq] ;
rhs(yy2, yp2) ;
```

A Useful Random Number Generator

```
for (ieq = 0 ; ieq < neq ; ieq++)
   yy3[ieq] = yy[ieq] + 0.5*dt*yp2[ieq] ;
rhs(yy3, yp3) ;
for (ieq = 0; ieq < neq; ieq++)
  yy4[ieq] = yy[ieq] + 1.0*dt*yp3[ieq] ;
rhs(yy4, yp4) ;
for (ieq = 0; ieq < neq; ieq++)
   yyp[ieq] =
   (yp1[ieq] + 2.0*(yp2[ieq] + yp3[ieq]) + yp4[ieq])/6.0 ;
for (ieq = 0 ; ieq < neq ; ieq++)
   yy[ieq] = yy[ieq] + dt*yyp[ieq] ;
return 1 ;
}
double rhs( double yy[neq], double yyp[neq])
{
yyp[0] = +yy[1] ;
yyp[1] = -yy[0] ;
return 1 ;
}
```

4.5 A Useful Random Number Generator

Symmetry breaking perturbations are often desirable. Generating them with a simple and portable random-number routine is a real convenience. In the typical case illustrated here each call of the routine generates a new number in the interval from 0 to 1. For colleagues who wish to reproduce each other's work it is necessary that the "random" numbers each uses follow exactly the same sequence. For this purpose we use the following simple function randnum(intx,inty), which returns a new random number, randnum, as well as two corresponding integers intx and inty, each time the function is called :

```
c This is a function to generate a random number
function randnum(intx,inty)
```

Computer Programming

```
implicit double precision (a-h,o-z)
i = 1029*intx + 1731
j = i + 1029*inty + 507*intx - 1731
intx = mod(i,2048)
j = j + (i - intx)/2048
inty = mod(j,2048)
randnum = (intx + 2048*inty)/4194304.0d00
return
end
```

This function generates decimal approximations to rational fractions between 0 and 1. The numerator is a 22-bit integer (equivalent to a 2-digit base-2048 integer) and the denominator is $2^{22} = 2048^2 = 4,194,304$. Each of the possible 2^{22} fractions occurs before the generator repeats so that the cycle length is also 2^{22} , long enough for most purposes. To illustrate how the random number generator works, begin with the (arbitrary) choices intx = inty = 0, giving successively :

The next time randnum is called, with intx = 1731 and inty = 0, the routine calculates :

The next three "random numbers" found in this way are 0.161475420, 0.908619881, and 0.970269918 .

By using *pairs* of such random numbers any other distribution can be simulated. For instance, numbers approximating a Gaussian probability density $\propto \exp(-\frac{r^2}{2})$ with |r| < 5 are generated by accepting the choice,

R1 = 10.0d00*(rand1 - 0.5d00)

whenever the inequality ,

rand2.lt.exp(-R1**2/2.0d00)

is true, where rand1 and rand2 are successive random numbers.

Graphic Displays and Analysis

4.6 Graphic Displays and Analysis

Graphic displays of results from particle simulations are essential for efficiently analysing physical problems and finding errors. There are many software tools for visualizing results from scientific analyses. Some are simple, easy to use, and provide relatively crude quality output easily. The freeware software, gnuplot, is in this class. With some effort, gnuplot can be used to produce publication-quality graphics. A majority of the Figures in this book were generated with gnuplot. Other graphical tools cost money and support good quality output with less work. The commercial products, Mathematica and Matlab, provide the functionality for displaying results in a variety of topologies. The input data for gnuplot, Mathematica, and Matlab are all in the form of text files generated by the user's program.



Sophisticated visualization tools developed for grid-based continuum calculations have been extended to include plotting results from particle simulations. The Griz4 and VisIt visualization tools developed at the

Computer Programming

Lawrence Livermore Laboratory are examples. Griz4 has been used to generate several Figures in this book. See Figures 4.1, 4.2, 4.9, 4.11, 4.13, and 4.14 in this Chapter. Such visualization tools use *binary* file formats so that large three-dimensional databases are read efficiently, without timeconsuming conversion from "asci" to "binary" form. The binary file formats are generated by special-purpose subroutine libraries. The libraries produce a flexible database format to treat the grid objects found in regularly-zoned grids and in finite-element and finite-volume grids. The subroutine libraries are designed to support the element shapes (bricks, shells, rods, ...) that are typical of problems in solid and fluid mechanics. The libraries also include special techniques for visualizing vectors (as arrows) and the components of tensor field variables (as colors, or shades of grey).



An advantage of smooth-particle methods is that all the computed variables can be displayed at particle positions or on any other convenient mesh. The two choices lead to two types of useful graphical displays:

 Results displayed for particles by coloring circles in two dimensions or spheres in three dimensions. See Figure 4.1 for a two-dimensional example.
 Results displayed for a mesh using contour or color plots. See Figure 4.2 for a mesh-based representation of the particle data shown in Figure 4.1.

Graphic Displays and Analysis

Physical results of interest in smooth-particle calculations include density, the kinetic and internal energies, and the components of the coordinates, velocity, acceleration, pressure, and heat flux. See Figure 4.3. The programming for the graphics output should include any additional variables that may be useful in the analysis of a specific problem. For example, in solid mechanics models with material failure it is useful to plot the shear-stress magnitude and the plastic strain.

Variables can be plotted at one instant or at several, as an animation. Animations are essential to visualizing wave propagation and flow. They are a particularly valuable debugging tool.



"self" contributions to the energy Φ discussed in Sections 3.11 and 7.2.

Computer Programming

The Griz4 tool provides many convenient interactive features such as mouse commands to display numerical values at selected particles' locations, rotations, translations, and scaling of three-dimensional objects in a frame, as well as an (x, y) plotting capability. This (x, y) feature makes it possible to plot particle properties as functions of time or to plot two correlated field variables at selected points. See Figure 4.4 for an example.



With most graphics software contour plots require the evaluation of the displayed function f(r) on a regular array of mesh points. Recall (Section 3.5) that a function evaluated at the particle location r_i is typically a sum of weighted contributions from particles j within a distance h of Particle

,

Graphic Displays and Analysis

i. The same idea can be used to evaluate averaged properties at the mesh points { r } required for a contour plot:

$$\left\{ f(r) = \sum_{|r_j - r| < h} f_j w_{rj} / \sum_{|r_j - r| < h} w_{rj} \right\} ; w_{rj} = w(r_{rj}) ; r_{rj} = |r - r_j| .$$



Unless a cell sort has been used, the best method for generating the mesh-point grid values is to loop over particles, summing each particle's

Computer Programming

contributions to all nearby grid points, as is suggested by Figure 4.5.

Boundary conditions can add further complexity to the calculation of the particles' mesh-point contributions. For example, a particle within a distance h of a *mirror* boundary can make *two* additive contributions to nearby grid values. Figures 4.5 and 4.6 illustrate the typical bulk situation. In Figure 4.5 a single particle contributes to many (28 in this case) nearby mesh points. Figure 4.6 illustrates the general case.



Figure 4.6: Within the 1×1 unit cell shown to the left a particle with h = 3 can contribute to field variables at from 26 to 32 nearby mesh points. The seven colors displayed at the right correspond respectively (from top to bottom) to 26, 27, 28, 29, 30, 31, and 32 contributions.

"Debugging" Tools—Finding Errors

4.7 "Debugging" Tools—Finding Errors

"Bugs" are programming errors. The list of underlying causes is endless: misspelling a variable name, omitting a comma, continuing to type past column 72, typing 0 rather than 0 or 1 rather than 1, using inconsistent array lengths, *et cetera*, *et cetera*, *et cetera*.

Compilers detect many bugs, but those which elude this automatic detection become the responsibility of the programmer. Debugging a program is as challenging an aspect of numerical simulation as is the original design and development of the project. Though debugging is a highly individual endeavor, a systematic process will include some of the following ideas:

1. Test any new program against known solutions of simple problems. Both the propagation of sinusoidal density perturbations and the decay of sinusoidal temperature perturbations can be compared with the analytical solutions from the wave and diffusion equations, as in the Example Problems of Chapter 2.

2. Diagnostics, confirming that appropriate conservation laws are satisfied, are valuable checks of algorithms. In the development stages it is useful to print the total momentum and energy at every timestep.

3. If the problem stops with an overflow, lack of initialization, perhaps through misspelling, or division by zero, are the likely culprits. "Runtime" error messages (including the traceback to the offending routine) automatically generated by the system software, are usually the simplest and most reliable clues to follow in such a case. The offending location or variable can often be found by using a commercial debugging tool to generate a "traceback" to the source line in the routine containing the illegal instruction. Examples of debugging tools are DDT, a debugger provided for workstations, Solaris' Developers Workbench software for Sun workstations, and Totalview, a debugging tool for both single-processor and parallel-computer systems. If no clear diagnosis emerges from tracing back to the routine with the error, then the same program can be compiled again (with a -g option included, which generates a symbol table) so that a "symbol table" of variables is available. Then the debugging software can be used to display variables' values during execution.

4. If the problem runs, but the answers are incorrect, generate and analyze graphical output to localize the source(s) of the error(s).

5. Follow up the graphical analysis by using a debugging tool or write statements to isolate the problem further.

Computer Programming

In developing a numerical simulation program, it is best to check each routine as it is developed, using the smallest feasible problem size for a meaningful numerical check. For example, when developing the initialization subroutine, use write or print statements to display on the screen the particle positions or store them in a data file. Turn off any random perturbations from the initial lattice positions to enable checking that the initial positions are correct. Likewise, print or plot the initial velocities. Then scan the arrays to make sure that the numbers appear to be within their expected bounds. If results are written to a data file, then a standard graphics program such as gnuplot can be used to display the particle data or to create contour, surface, or (x, y) plots.

Breaking up algorithms into component subroutines makes the resulting program more readable and saves future time by making much of the programming work reuseable as other programs are developed. Individual subroutines can be checked by calling them from a dummy main program and then writing out data from appropriate places within the subroutine. This approach is essential in debugging subroutines used to interpolate particle data onto a finite-element mesh. It is particularly useful for teams developing large software programs.

Errors which occur at or near boundaries and interfaces are the hardest to find and correct. The first indication that there is such an error typically shows up as a change in the total energy or momentum. These totals are global diagnostics. Next the boundary condition, boundary location, and the specific particles leading to the error(s) must be identified. Such errors nearly always vary with time. One approach to identifying their source is to select one or more particles in a boundary region where physical variables are rapidly varying. Properties for this select group can then be collected for study. For large problems this can be a formidable task, with prohibitively long files of particle data. Interactive graphics and debugging tools are specially helpful in locating the offending particle(s) and inspecting their properties.

Debugging tools are specially useful in diagnosing errors in arrays of particle data. The best, most fully developed, debugging tools are commercial windows-based software packages that can display source code (Fortran or C) lines in one window with current array and scalar values in one or more separate windows. Inspecting arrays in popup windows, using a scroll bar, is a very efficient debugging procedure. See Figure 4.7 for a sample of the Totalview debugging-tool display. Totalview is particularly valuable for

"Debugging" Tools—Finding Errors

debugging parallel programs and is currently the best such tool available. Online Totalview documentation, with many examples, can be found at http://www.etnus.com .

Debugging tools can also be used to stop a running program at preselected lines—"breakpoints"—in the source program. The tool can then display the current values of any desired particle or mesh variables. This method is much faster and simpler to use than recompiling and executing the source code after inserting or removing write or pause statements. Breakpoints can be particularly useful within each of the two or four Runge-Kutta steps, or at the end of a timestep when energies and other averaged results or algorithms are being computed. The debugging tool makes it possible to view partial results as they are computed within the routine **rhs** which evaluates the righthandsides of the equations.



Figure 4.7: The main window from the Totalview debugging tool displays lines in the source code and contains many menu options for stopping the program, for displaying subroutine source lines and values of the arguments in a subroutine call, as well as smaller "window panes" with debugging status displays. Popup windows can be used to display variables (x, vx, time in this example).

Computer Programming

4.8 Parallel Computing

Parallel computers provide opportunities to simulate larger problems and longer timescales, with enhanced detail and complexity, as in some of the following examples :

1. Crash scenarios can be modeled, including detailed descriptions of both occupants and passenger compartments. Typical applications use distributed-memory parallel computers with as many as 32 processors.

2. Turbulence simulations greatly benefit from hundreds and even thousands of processors. The calculations can be extended to include complex theoretical models for turbulence. Likewise a fully nonlinear simulation of turbulence on parallel computers can model extended length and time scales.

3. Material behavior characterized by a statistical description of flaws can be simulated with systems with *billions* of degrees of freedom.

4. The cavitation of the fluid column illustrated in Figure 2.1 of Chapter 2 runs in two hours on 64 processors of a Linux cluster parallel computer using the ParaDyn program. The equivalent run on a slower single-processor computer would require about 256 hours for a single simulation of the fluid column! At present (2006) it takes too long to run a set of problems to study the behavior of such systems without access to a parallel computer. There are applications in many fields that have benefitted from the increased capacity and speeds afforded by parallel computers.

The two most popular parallel computer architectures are: (i) sharedmemory computers, and (ii) distributed-memory computers. A sharedmemory computer consists of many processors sharing a common memory with relatively expensive hardware connecting the processors to the full shared memory. Several companies including Silicon Graphics and IBM have successfully marketed shared-memory systems for small-scale (rather than massively) parallel applications. Generally the number of processors in these computers varies between four and 64, although larger, more-expensive systems have been built. The smaller shared-memory systems with up to 64 processors are gradually being phased out in favor of cheaper, faster distributed-memory cluster computer systems. Cluster computers provide either an interconnected set of single-processor computers or a small-scale parallel computer. Because cluster systems are designed using mass-produced commodity processor and network hardware, university computer science teams can design and build them for their local research groups.

Parallel Computing

The distributed-memory computers include hundreds or thousands of processors, each one connected to all of the other processors through a high speed, (gigabit/second) interconnect network. These computers are referred to as massively-parallel computers or message-passing computers. Each processor has exclusive use of its own memory. The interconnect network is designed for sending data (messages) between processor memories. Data computed in one processor and needed in a second processor must be sent from the first processor to the second processor over the interconnect network.

The largest parallel computers, developed for the National Laboratories and other government agencies in the United States, have been designed by companies including IBM, Silicon Graphics, Intel, Compaq, and Cray Research. These computers are built with thousands of processors and each network connection point services a *node* which includes a small number of shared-memory processors. The primary parallel architecture is distributed-memory with shared-memory nodes typically including as few as four and as many as 16 processors.

Massively parallel computers are evolving very rapidly. Each new generation uses a larger total number of processors, higher speed processors, and new interconnect technologies. The goal of the manufacturers and code developers for large-scale scientific applications is to improve upon today's teraflop computing speeds to reach petaflop speeds (10^{15} floating-point operations per second) soon. Twice a year the massively-parallel computers are evaluated with standard benchmarks to produce lists rating them by speed, total number of processors, and other relevant attributes, such as the category of scientific applications. The results for the fastest computers in the world can be viewed on the website http://www.top500.org.

The labor needed to reprogram algorithms for a message-passing parallel computer can easily be an order of magnitude larger than that required for developing a single-processor algorithm. Despite this increased programming effort, message-passing computers have been the most successful form of parallel computer technology today for problems with more than a million degrees of freedom or for small-scale parallelism on clusters. To use parallel computers *efficiently* (as is described next) the problem domain must be divided into subdomains that are distributed among the processors. Each processor then solves the portion of the main problem on its subdomain in parallel with the partial solutions from other processors, sending and receiving results from other processors as needed. Ideally, the full problem is subdivided (or *partitioned*) in such a way that all pro-

Computer Programming

cessors are kept equally busy. This goal is called *load balancing*. Without load balancing, the computing power of the parallel system is squandered. The remainder of this Chapter outlines the program changes and the new software needed to develop message-passing parallel programs for particle or grid-based continuum mechanics.

The time to process a calculation on a parallel computer includes the processor time as well as the communication time for message passing between processors. Partitioning techniques and message-passing parallel algorithms are designed to split up the computational processing among the processors as evenly as possible while minimizing the time spent on processor-to-processor communication. The speedup s and efficiency η are useful quantitative measurements of parallel algorithm performance. Speedup is the inverse ratio of computer time with N processors to that with a single processor :

$$s = \tau_1 / \tau_N = N\eta \leq N = N\eta_{\text{ideal}}$$
.

The ideal speedup, $s_{\text{ideal}}(\eta = 1) = N$, occurs if the communication time is negligible compared to the calculation time.

Figure 4.8 shows typical performance data for a ground shock problem simulated with the ParaDyn program on four parallel computers several years ago (1996). These simulations follow a shockwave propagating from the upper free-surface portion of the grid to the subterranean tunnel located along the left side of the grid. The coloring of the mesh is by subdomains allocated to each of 128 processors. The density of elements in the mesh is much higher at the top of the model where the shock front is located and the motion is most severe. This same Figure also gives a graphical impression of the usefulness of "cutplanes" as diagnostic tools. Cutplanes are designed to show selected material properties in the interior of a deforming specimen. The apparent "cutplanes" in this Figure are actually symmetry planes and a free surface. In Griz4 cutplanes are defined by specifying an (x, y, z) point together with a vector, such as (1.0, 1.0, 1.0), normal to the cutplane.

Mesh Partitioning



4.9 Mesh Partitioning

The process of subdividing a model system into roughly equal pieces of computational work is referred to as *mesh partitioning* or *domain decomposition*. It is fortunate that our physical systems modeled with SPAM particles that flow through a space, governed by gradients (∇v , $\nabla \cdot P$), can take advantage of partitioning techniques developed for grid-based Eulerian and Lagrangian algorithms.

Continuum mechanics problems of interest, whether simulated with particles or regular Eulerian meshes or Lagrangian finite elements, *always* involve algorithms that must be designed carefully to balance the work among processors (load balancing). Partitioning is a challenge for the topologies
Computer Programming

and interface conditions found in real-world simulations. Graph theory⁹ is a useful conceptual basis for constructing partitions. "Graphs" are simply sets of "vertices", some pairs of which are connected by "lines".

Either a two-or-three-dimensional mesh or a one-or-two-dimensional material interface can be represented as a graph by first associating each element to a *vertex* and then drawing a *line* to any other vertices (elements) that have at least one nodal point in common with the element. This correspondence is illustrated in Figure 4.9 for a portion of a two-dimensional rectangular mesh of quadrilateral elements. Individual elements in such a mesh can interact with up to eight neighbors (26 in three dimensions).



Figure 4.9: A portion of a quadrilateral mesh containing 13 elements defined by 22 nodes is shown at the left. At the right is the corresponding "graph", where the 13 open-circle vertices represent the elements, and the 32 lines represent the interactions between pairs of elements.

The element-to-vertex correspondence shown in Figure 4.9 is the simplest graphical representation of a mesh. A more sophisticated correspondence can represent many contiguous elements by a single vertex, with a *weight* attached to the vertex representing the computational work associated with the elements it represents. In such a graph the lines carry weights also, representing the communication time involved in message passing between the groups of elements. With this representation automated software can be used to find iteratively-optimized partitions of the graph and the corresponding mesh. A partition corresponds to a series of "cuts" separat-

 $^{^{9}}$ Gross and Yellen, (2005).

Message Passing Techniques

ing sets of vertices from each other. The *best* cuts in the graph occur when: 1. The cuts divide the vertices into groups of roughly equal size.

2. The number of lines (edges) cut is minimized;

These cuts are to be made by taking into account both the vertex and line weights, corresponding to minimizing communication between processors with roughly equal amounts of computational work.

Algorithms to find the *best* cuts use an optimization technique that is based on the length-to-area ratios (surface-to-volume ratios in three dimensions) as well as the aspect ratios of the areas (volumes) generated by the partition. Typically, some function of these two variables is optimized and the cuts are only considered acceptable if the number of vertices included in each processor lies within a specified range, { $n_{\min} < n < n_{\max}$ } , where the ranges are given fractions of the average number of elements per processor. This latter condition helps to balance the computational load among processors.

In practice, partitioning algorithms for large meshes are iterative. This is because large problems (with more than a million elements) correspond to graphs that take a prohibitive amount of computer time to partition for a reasonable number of processors (128 or 256 processors). To simplify this problem, a "graph reduction algorithm" can be applied to the graph to reduce its size before finding a solution to the optimization problem. In most practical problems, more than one graph reduction step is applied to achieve a final reduced graph that is tractable. Once a reasonable partition is found on a reduced graph, the reduced graph is expanded back by inverting the steps taken in the reduction process. At each expansion step the boundaries of the subdomains are "smoothed", by exchanging vertices (elements) between adjacent domains.

Graph partioning software, documentation, and references on graph partitioning are available free.¹⁰ The software provides an option to input the mesh directly into the software rather than generating and inputting the corresponding graph.

4.10 Message Passing Techniques

Efficient processor-to-processor communication is vital. In order to calculate the neighbor sums in a SPAM algorithm, particles close to a subdomain

¹⁰see George Karypis' web site: http://www-users.umn.edu/~karypis/ .

Computer Programming

(processor) boundary must have data available from all of their neighbors including neighbor particles in different subdomains (processors).

Most of the time spent communicating data during a parallel SPAM simulation occurs when sending and receiving data needed for particle sums. On parallel computers it is important to limit the search for neighbors within a local region of the domain so that the number of processors participating in the searches is small. A useful way to localize the search for interacting neighbors is to use a grid of cells with particles sorted into the cells. Using cell searches rather than searching the full domain for pairs reduces the computational time for the neighbor search from $O(N^2)$, where N is the total number of particles, to O(nN), where n is the number of particles in the neighboring cells.



Figure 4.10: Properties of particles in the similarly-colored boundary cells are required in these four neighboring processors. These shared particles vary with time and are updated as particles flow between processors.

Message Passing Techniques

Although there are alternative methods for localizing the neighbor searches (such as neighbor lists or a list of interacting pairs), the cell model combines very well with partitioning techniques for parallel computers. The cell size is best chosen slightly larger than the interaction distance, h, so that a 3×3 grid of cells in two-dimensions will enclose all neighbors. Similarly, in three-dimensions nine cubes in each of three planes (one plane of cubes containing the particle, one plane above, and one plane below the plane of cubes containing the particle) will include all of the neighbors within the smoothing length. The cells on the boundary of processors are duplicated in all their neighboring processors. Thus, the neighbor particles in cells at a processor boundary are duplicated in neighboring processors. The data needed to complete neighbor sums are communicated between neighboring processors and available in the boundary cells of each processor. See Figure 4.10. As particles flow through the computational domain, they move into new cells and occupy new subdomains (processors). It is most efficient to choose the cell size such that only that information from a single layer of cells needs to be exchanged for each righthandside evaluation.

The programming which must be added to develop a parallel particle algorithm includes detailed bookkeeping to locate particles within the spatial domain and also in the processor set. Message-passing library calls must be added to communicate data in boundary cells between processors and to calculate global sums over all particles to compute momentum and energy diagnostics.

It is clear that parallel continuum algorithms involving either particles or finite elements are considerably more difficult to program and to debug than their counterparts for single-processor computers. This programming effort has been reduced considerably in the last several years with software specially designed for parallel computers such as the Message Passing Interface (MPI) library,¹¹ libraries with mesh partitioning algorithms, and parallel environment tools for debugging and performance analysis.

¹¹See documentation provided by the MPI forum at http://www.mpi-forum.org/ and the internet tutorials resulting from a Google search for "Message Passing Interface".

136

Computer Programming



4.11 Material Interfaces in Parallel Computing

Partitioning a mesh or partitioning cells containing particles must often be done repeatedly. A single partitioning, at the beginning of a problem, is only sufficient for problems which change very little in their dynamic development. Particularly in treating problems with large and changing surface-to-volume ratios (as in the crushed elastic-plastic sheet shown in Figure 4.13), time-dependent partitioning must be used to maintain the balance of computational work as the dynamics of the material surfaces within the flow unfolds. Let us consider four examples, shown in Figures 4.11, 4.12, 4.13, and 4.14. Each of these examples is a challenging problem, even those shown here as two-dimensional caricature forms. The last three

Material Interfaces in Parallel Computing

of these examples make the need for time-dependent partitioning clear:

- (1) Two concentric annuli undergoing relative rotation.
- (2) Expansion of a compressed gas into an empty container.
- (3) Crushing of an elastic-plastic sheet.
- (4) Colliding balls rolling on a supporting surface.

4.11.1 Concentric Annuli Undergoing Rotation

This problem is a simplified version of the interactions between a nut and a bolt, with the two mechanical parts exerting relatively complicated timevarying forces on one another through a "contact algorithm" as the nut-tobolt connection is tightened. The algorithm prevents substantial overlap of the parts by imposing either Lagrange-multiplier constraints or steep Hooke's-Law repulsive forces. This example illustrates the flexibility of graph partitioning techniques. In many practical problems in solid mechanics the elements on the surfaces (here just those elements adjacent to the bounding circle) make up such a small and regular fraction of the volume represented by the mesh that the interactions of the pairs of surfaces are predictable. Here we know that the only contact between the two parts is the sliding contact between the inside of the nut and the outside of the relatively-rotating bolt.

This predictability of the contact surface is typical of many large mechanical systems made up of subsystems, most of which never come into mechanical contact with each other. In such cases, as in this simple example, it is reasonable to consider *two* independent partitions rather than just one: first, a complete partition of the full mesh description (both annuli in this example) and then, second, a *surface partition* in which all those elements making up the two contact surfaces are jointly allocated to a single processor (or to an appropriate subset of the processors in the general case, where *many* surface pairs are involved). The number of processors dedicated to surface pairs needs to be chosen so that the computational loads on all processors, both surface and bulk, are approximately equal. In practice, by using a Hooke's-Law spring constant appropriate to the bulk modulus of the elements, the computation of surface elements is hardly more time consuming than that of bulk elements. And it is quite fortunate,

Computer Programming

due to the additive nature of the nodal forces, that the surface and bulk calculations can be treated independently of each other. A cell structure can be used to search for pairs of elements (for each pair of surfaces) which might interpenetrate.



4.11.2 Free Expansion Problem

Consider now an example of *time-dependent* partitioning, We consider the fourfold expansion of an ideal gas fluid, detailed in Section 7.8. Initially the gas is confined to a square region making up one fourth of the total area. Figure 4.12 shows the dynamic partition of the particles among 25 processors, arranged checkerboard fashion, as determined using molecular dynamics with 2500 particles.¹² The free expansion problem is an illustration of material flow that would result in a load inbalance for a static partition of the computational mesh. Initially most of the space is empty, while at the end the space is filled relatively homogeneously. Here the particle flow and resulting density gradients clearly could cause a work inbalance if more than four processors were used for this problem. The center processors would lose particles to the processors near the boundary and the

 $^{^{12}}$ Hoover and Posch (1999) used up to 65,536 particles for this problem.

139

Material Interfaces in Parallel Computing

load balance would remain poor until equilibration occurs in the expanded volume. Using a dynamically generated partition in order to redistribute the cells on this mesh divides the work among the processors for an efficient parallel calculation. The partitioning algorithm in this case is designed to minimize the motion of cells into and out of processors. Thus the current graph with its set of cuts is used as the starting point for a new partition.

4.11.3 Crushing of an Elastic-Plastic Sheet



Computer Programming

The dynamics of a thin sheet of aluminum compressed within a shrinking cube bounded by six moving stonewall boundaries is another example of a continuum calculation using finite elements to simulate a system with arbitrary surface motion. This problem is entirely made up of surface "shell" elements. The arbitrary contact of *any* element on the surface with *any* other part of the surface must be treated with an Eulerian grid of cubes for localizing the search for contact. In a parallel calculation, the same grid of cubes is used to load balance the contact calculations. The amount of computational work in a cube depends on both the number of surface elements and the number of nodes in the cube. The amount of communication between two cubes on the grid is likewise weighted by the number of nodes and surface elements that would need to be communicated if a cut were made between the two connected cubes.

4.11.4 Caricature of a Billiard Table

A time-dependent surface partition can be useful whenever the number of surface elements is much larger than the number involved in contact, and in a way which varies with time. A good illustration of time-dependent surface partitioning is the contact of pool balls rolling on a table after the initial racked set of balls is struck by the cue ball. The ball material could be simulated as a deformable elastic material composed of finite elements or as sets of mutually-interacting smooth particles. In principle, all pairs of balls can interact while simultaneously contacting the table surface, and occasionally colliding with the bounding cushions of the table. The points in contact along the surfaces of the balls and on the table and cushions must be determined at each timestep. The partitioning of the table can be treated as a static partition while that for the balls is dynamic, governed by the locations of the balls on the grid of processors which partition the table.

Figure 4.14 illustrates a four-ball caricature of this challenging problem. The balls are of four different sizes and composed of hexahedral elements. For such a problem an Eulerian grid of cells can be used to localize regions on surfaces that are in contact with other surfaces. The grid of cells is used to sort nodes that are connected to surface quadrilaterals. The computational work to solve this material "contact" problem is clearly *not* distributed evenly among the processors using the partition generated for the original mesh with motionless balls. Instead, the surface partitioning is computed dynamically and data needed for computing material surface

141

Material Interfaces in Parallel Computing

interactions is stored (most likely in several processors) along with the surface data. Force calculations and other particle properties from processors with surface interactions are communicated to the set of processors with the connected elements in the bulk. Next these processors update the full configuration (surface and bulk elements and nodes). Finally, the updated surface configurations are communicated back to the processors calculating the surface interactions. Any load balancing that is needed can be done before this final step.



Computer Programming

4.12 References

- D. D. McCracken and W. Salmon, Computing for Engineers & Scientists with Fortran 77 (Wiley, 1990).
- (2) S. J. Chapman, Fortran 90 and 95 for Scientists and Engineers (Mc-Graw Hill, 1997).
- (3) B. Kernighan and D. Ritchie, *The C Programming Language* (Prentice Hall, 1988).
- (4) S. Kochan, *Programming in C* (Sams, Prentice-Hall, Indianapolis, 2004).
- (5) See http://www.ioccc.org for entry rules and recent winners of the International Obfuscated C Code Contest.
- (6) W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, *Numerical Recipes, The Art of Scientific Computing* (Cambridge University Press, London, 1986).
- E. N. Lorenz, *The Essence of Chaos* (University of Washington Press, 1993).
- (8) Wm. G. Hoover, *Computational Statistical Mechanics* (Elsevier, 1991), available free of charge at http://williamhoover.info.
- (9) J. L. Gross and J. Yellen, Graph Theory and its Applications (CRC Press LLC, 2005).
- (10) See George Karypis' website: http://www-users.umn.edu/~karypis/ for mesh-partitioning software.
- (11) See the message passing website: http://www.mpi-forum.org/ for message-passing software.
- (12) Wm. G. Hoover and H. A. Posch, "Entropy Increase in Confined Free Expansions via Molecular Dynamics and Smooth Particle Applied Mechanics", Physical Review E 59, 1770-1776 (1999).

Chapter 5

Initial and Boundary Conditions, Interpolation

/ Initial Conditions / Mesh Generation for SPAM with Free Boundaries / Fixed Boundaries / Free Boundaries / Implementing Periodic and Mirror Boundaries / Alternative Meshes—Regular Lattices / Elastic Stability of Embedded-Atom Lattices / Crystal Stabilization via Invariant Curvature /

/ 12 Figures /

Example Problems :

[Elastic Stability of a Triangular Lattice Fluid, One-Dimensional Heat Transfer, Periodic Shear Flow, Rayleigh-Bénard Flow]

5.1 Summary

All the types of time-dependent simulation algorithms, Eulerian, Lagrangian, and SPAM, have common features. To begin with, all these simulations require initial conditions. At least coordinates and velocities are required; often additional variables such as energy, stress, or elastic and plastic strains must be specified too. To continue, motion equations are required for the variables' evolution, as well as constitutive relations and boundary conditions. Last, a compatible time integration algorithm needs to be developed and implemented.

For simulations which are expected to reach a stationary state the initial conditions are typically guessed on the basis of simple physical principles. Consider the two-roll Rayleigh-Bénard problem of Section 2.12. One way to start out is to choose the simplest nonvanishing Fourier components

144

ws-book9x6

Initial and Boundary Conditions, Interpolation

consistent with the boundary conditions on the flow. We will treat that same flow in this Chapter, as an example problem with SPAM. With gridbased simulation methods the choice of mesh is a relatively straightforward, but tedious chore. A finite-element mesh for an automobile can require months to construct.

SPAM is quite different to the usual finite-element approaches, in that the "volume" (m/ρ) associated with SPAM particles is not a definite localized region within a set of nodes, but must nevertheless be accounted for in the initial conditions, so as to be consistent with the desired material density. In what follows we describe methods for generating and equilibrating meshes suitable for smooth-particle simulations. We consider also the stability of the particle meshes to small displacements.

5.2 Initial Coordinates

SPAM problem descriptions require not only the usual continuum variables (mass density, velocity, internal energy), but also an arrangement of particles roughly consistent with the continuum description. Regular lattices are simple to generate but occasionally their anisotropy (or anisotropicity!) needs to be taken into account. An *irregular* structure can be generated by allowing either an initially-random distribution of particles or a perturbed regular lattice to relax according to a damped molecular dynamics scheme, as was illustrated in Section 2.3.

Such a relaxation technique is specially useful when a nonuniform density is required, as in the presence of centrifugal or gravitational forces. Consider the representation of a fluid in a gravitational field by using a regular lattice of particles. For a simple explicit example we choose the two-dimensional square lattice. For such a lattice to represent a *fluid* it is necessary that the lattice can "flow", with only a viscous inelastic resistance to shear. Smooth particles can make this possible. Figure 5.1 shows the results of two sample smooth-particle dynamics problems for the fluid equation of state :

$$P = B_0 \left[\left(\frac{\rho}{\rho_0} \right)^3 - \left(\frac{\rho}{\rho_0} \right)^2 \right] \longrightarrow \ddot{r}_i \propto \sum_j \left(2 - \frac{\rho_i}{\rho_0} - \frac{\rho_j}{\rho_0} \right) \nabla_i w_{ij} ,$$

with m = 1, $B_0 = 1$, and $\rho_0 = \frac{1}{2}$. This macroscopic smooth-particle dynamics is "isomorphic" (meaning, with identical trajectories) to the microscopic *molecular dynamics* of a system of particles with a many-body

Initial Coordinates

potential Φ which is quadratic in the particle densities { ρ } :

$$\Phi = \sum \phi_i = \frac{mB_0}{2\rho_0} \sum \left[\left(\frac{\rho_i}{\rho_0} \right) - 1 \right]^2 \; ; \; \rho_i \equiv m \sum_j w_{ij} \; \longrightarrow \;$$

$$m_i \ddot{r}_i \equiv -\nabla_i \Phi \propto \sum_j \left(2 - \frac{\rho_i}{\rho_0} - \frac{\rho_j}{\rho_0} \right) \nabla_i w_{ij} \; .$$

Figure 5.1 illustrates the equilibration of 6400 smooth particles of unit mass in a gravitational field, g = 1/40, chosen so as to give twofold maximum compression from the stress-free density. We use the equation of state $P = 8\rho^3 - 4\rho^2$, so that the bulk modulus $B = 24\rho^3 - 8\rho^2$ is unity at the stressfree density $\rho = \rho_0 = \frac{1}{2}$. The original stressfree arrangement of the 6400 particles whose relaxed configuration is shown in the Figure was a rectangle with height 320, width 40, and overall density 0.5.

With a column height H, $[\rho = \frac{1}{2}$ at the top and $\rho = 1$ at the bottom], the mechanical equilibrium relation within the column is :

$$dP/dy = -\rho g = (dP/d\rho)(d\rho/dy) = (24\rho^2 - 8\rho)(d\rho/dy) ,$$

which can be integrated to find the density profile :

$$(6\rho - 1)(2\rho - 1) = 5 - gy \longrightarrow H = 5/g = 200$$
.

No special boundary condition was necessary at the top of the fluid column. With "rigid" but energy-absorbing boundaries the sides and base extracted the normal velocities of any particles contacting them :

$$x < -\sqrt{N/4} \longrightarrow \{ x = -\sqrt{N/4} ; \dot{x} = 0 \} ;$$

$$x > +\sqrt{N/4} \longrightarrow \{ x = +\sqrt{N/4} ; \dot{x} = 0 \};$$

$$y < 0 \longrightarrow \{ y = 0 ; \dot{y} = 0 \}.$$



Initial and Boundary Conditions, Interpolation

Mesh Generation for SPAM with Free Boundaries

The dynamics (25,000 Runge-Kutta timesteps with $dt = \tau = 0.1$) began with random displacements in the range ± 0.25 from the squarelattice coordinates. The density for each particle (which turns out to be $\rho_0 = 1.00293003$ within a perfect square lattice) is computed using Lucy's weight function with a range of h = 3:

$$\rho_i \equiv \sum_j w_{ij} \; ; \; w(r) = \frac{5}{\pi h^2} \left(1 + 3\frac{r}{h} \right) \left(1 - \frac{r}{h} \right)^3 \longrightarrow$$
$$w(r) = \frac{5}{\pi h^2} \left[1 - 6\frac{r^2}{h^2} + 8\frac{r^3}{h^3} - 3\frac{r^4}{h^4} \right] \longrightarrow$$
$$w(r) = \frac{5}{9\pi} \left[1 - \frac{6}{9}r^2 + \frac{8}{27}r^3 - \frac{3}{81}r^4 \right] \; .$$

In the example at the right of the Figure periodic lateral boundaries were used, so as to eliminate surface effects. Surfaces and boundaries are a major fly in the ointment for SPAM. We turn next to the generation of meshes in a way which addresses this difficulty.

5.3 Mesh Generation for SPAM with Free Boundaries

In smooth particle calculations it is desirable that every particle have the same density as does the material it represents. For a regular lattice this can be guaranteed. By adjusting the scale length of the lattice any density greater than mw(0) can be obtained. But any physical problem has boundaries. Smooth particles near such a boundary will start out with a density less than do those in the interior unless special precautions are taken. A linear boundary (or a planar one in three dimensions) corresponds to a loss of nearly half the interior density because about half the neighboring particles are not present.

Consider a one-dimensional example, using the Lucy potential with particles of unit mass, a range of 3 and a nearest-neighbor spacing of unity. In such a linear array of particles the bulk-particle density is :

$$\rho_{\text{bulk}} = w(2) + w(1) + w(0) + w(1) + w(2) =$$

$$(5/12)\left(\frac{3}{27} + \frac{16}{27} + \frac{27}{27} + \frac{16}{27} + \frac{3}{27}\right) = \frac{325}{324} \ .$$

148

ws-book9x6

Initial and Boundary Conditions, Interpolation

On the other hand the density for a surface particle is

$$\rho_{\text{surface}} = w(0) + w(1) + w(2) = (5/12) \left(\frac{27}{27} + \frac{16}{27} + \frac{3}{27}\right) = \frac{230}{324}$$

If the boundary were fixed in space then this lack of density could be offset by defining "ghost" or "mirror" particles to extend the bulk-density environment on the outside of the bounding surface. We use this approach in the Rayleigh-Bénard example problem at the end of this Chapter.

When the boundary is *not* fixed (as in fracture or a breaking wave) but is instead free to move, another approach is required. Consider the use of a *double* particle at the surface (two particles occupying the same lattice site). Then the densities for the two surface particles are :

$$\rho_{\text{surface}} = 2w(0) + w(1) + w(2) = (5/12) \left(\frac{27}{27} + \frac{27}{27} + \frac{16}{27} + \frac{3}{27}\right) = \frac{365}{324} ,$$

considerably closer to the bulk density. This suggests that useful meshes can be generated simply by doubling up the surface particles (or, more cheaply, by doubling the mass of surface particles) and allowing the interior particles to relax from an initially regular or initially random arrangement. See Section 7.5 for a detailed two-dimensional example.

Relaxation to a desired density can be achieved by using a densitydependent force, an additional pair core force to discourage overlaps, and a damping force to induce the relaxation. The simplest density-dependent force follows from the embedded-atom potential function of Section 7.4 :

$$\Phi_{\text{density}} = \frac{mB_0}{2\rho_0} \sum_i \left[\left(\frac{\rho_i}{\rho_0} \right) - 1 \right]^2 \quad --$$

$$\ddot{r}_i = \frac{mB_0}{\rho_0^2} \sum_j \left(2 - \frac{\rho_i}{\rho_0} - \frac{\rho_j}{\rho_0}\right) \nabla_i w_{ij} \ . \label{eq:relation}$$

By itself, this potential tends to make anisotropic stringy structures, as is shown in Figure 5.2 for a 2025-particle example.

149

Mesh Generation for SPAM with Free Boundaries



Figure 5.2: Relaxed structure using the embedded-atom potential in the absence of gravity. The overall density is unity, with reflecting boundaries at ± 22.5 and with h = 3. The leftmost picture shows the 2025-particle configuration resulting from initially random coordinates. The rightmost picture shows the configuration resulting from a 45×45 square lattice with small random displacements (both x and y) chosen from the interval $\pm \frac{1}{4}$. In both cases the particle mass was equal to unity. Undamped motion, for the preliminary time interval 0 < t < 50, was followed by damped motion (with $\tau = 1$) for 50 < t < 100. The Runge-Kutta timestep was 0.1.

The strings can be avoided by using a pair core potential active at short separations :

$$\Phi_{\text{pair}} = \sum_{i < j} \epsilon \left[1 - \left(\frac{r_{ij}^2}{\sigma^2} \right) \right]^4 ; \ |r_{ij}| < \sigma \longrightarrow$$
$$m\ddot{r}_i = -\nabla_i \Phi_{\text{pair}} = \frac{8\epsilon}{\sigma^2} \sum_j r_{ij} \left[1 - \frac{r_{ij}^2}{\sigma^2} \right]^3 .$$

Another effective means for maintaining a reasonable particle-to-particle separation is to introduce hard-disk (hard-sphere in three dimensions) collisions to reverse the line-of-centers velocity $v_{ij\perp}$ for any pairs of particles approaching closer than a specified separation σ :

$$\{ \ |r_{ij}| < \sigma \ ; \ r_{ij} \cdot v_{ij} < 0 \ \} \longrightarrow \left(\ |r_{ij}| \rightarrow \sigma \ ; \ + v_{ij\perp} \rightarrow - v_{ij\perp} \ \right) \ ,$$

with the tangential component of the relative velocity unchanged.

Initial and Boundary Conditions, Interpolation

Combining the hard-particle short-ranged potential with the damping force, $F_{\rm damp}=-mv/\tau$, assures that the structure will relax toward a potential minimum with nearly the right density everywhere and without any very-close nearest-neighbor pairs. Figure 5.3 shows the result obtained with the same 2025-particle system with the pair core potential included.



5.4 Implementing Periodic and Mirror Boundaries

Molecular dynamics provides useful ideas for the formulation of boundary conditions in SPAM. Molecular dynamics is often used as a means for determining constitutive properties (pressure, moduli, heat capacity, ...) for specified interparticle force laws. The constitutive properties obtained in this way are time averages of appropriate functions of the particle coordinates and momenta. To avoid the influence of surface effects *periodic boundaries* are the usual choice for these simulations. A periodic system can be visualized as an infinite array of similar systems, each of area $L_x \times L_y$, arranged checkerboard-fashion in space, with each system identical to the rest apart from a translation of the coordinates, ($n_x L_x, n_y L_y$), where n_x and n_y range over all integral values. Provided that the range of particle interactions is less than L/2 there is still an unambiguous definition of energy (and an equivalent Hamiltonian, from which the usual equations of motion can be derived).

Implementing Periodic and Mirror Boundaries

A topologically more complicated but physically identical model pictures a single system, with its righthandside smoothly attached to its left, and with the top likewise attached to the bottom. Such "periodic" boundary conditions are implemented as in the molecular-dynamics problem at the end of Chapter 1 . Imagine that the x coordinates and y coordinates are restricted to intervals from -L/2 to +L/2. As the motion evolves, any particle crossing a boundary is simply replaced on the opposite side of the system :

$$\begin{aligned} x &< -L/2 \rightarrow x = x + L \ ; \ x > +L/2 \rightarrow x = x - L \ ; \end{aligned}$$

$$\begin{aligned} y &< -L/2 \rightarrow y = y + L \ ; \ y > +L/2 \rightarrow y = y - L \ , \end{aligned}$$

with its velocity unchanged.

In calculating the effect of short-ranged forces between Particle *i* and Particle *j*, periodic boundaries require using the smallest (in absolute value) "nearest-image" value of the three possible separations, $r_{ij} = (x_{ij}, y_{ij})$, in both the *x* and the *y* directions. This nearest-image separation results when the *same* adjustment applied to x_i and y_i for periodic boundary conditions is also applied to x_{ij} and y_{ij} :

$$\begin{aligned} x_{ij} &< -L/2 \to x_{ij} = x_{ij} + L \; ; \; x_{ij} > +L/2 \to x_{ij} = x_{ij} - L \; ; \\ y_{ij} &< -L/2 \to y_{ij} = y_{ij} + L \; ; \; y_{ij} > +L/2 \to y_{ij} = y_{ij} - L \; . \end{aligned}$$

To avoid ambiguity the range of the forces cannot exceed L/2 when periodic boundaries are used. Similar treatments can be carried out for space-filling cells with more complicated shapes (hexagonal, octahedral, dodecahedral) as well as for cells undergoing shear. Straightforward periodic boundary conditions conserve mass, momentum, and energy, but, because there is no real origin, not *angular* momentum. Errors in intensive constitutive properties, such as pressure and temperature, are typically of order (1/N)or $(\ln N)/N$ with periodic boundaries rather than $(1/N)^{1/D}$.

Typical *non*equilibrium problems have real *nonperiodic* boundaries which can be free, moving, or fixed. Because the smooth particle density is a sum of contributions from nearby particles ,

$$\rho(r) = \sum_{j} m_j w(|r - r_j|) ,$$

Initial and Boundary Conditions, Interpolation

it is apparent that calculated particle densities at the "edge" of a material have densities on the order of half the bulk density. Better densities, and better dynamics, can result if the missing nearby particles are replaced by mirror images, as is shown in Figure 5.4 . If, for instance, the velocity parallel to a boundary is required to vanish, this can be ensured by defining the mirror particle's parallel velocity component equal to the negative of that of its twin. Similar rules can be defined to induce or prohibit heat transfer, as in the Rayleigh-Bénard Example Problem of Section 5.10 .



Figure 5.4: The six particles outside the dashed square lie within a distance h/2 of the system boundary. The six interact with "mirror-image" particles shown "outside" the system. The upper-right corner particle interacts with *three* mirrored images of itself. Mirror-image particles, defined by a straight-line boundary, provide a bulk-like environment.

Implementing Periodic and Mirror Boundaries

In molecular dynamics the motion of a mirrored particle follows from the Hamiltonian :

$$\mathcal{H} = \frac{p^2}{2m} + \phi(2|r|) \; ,$$

where r is the line-of-centers separation of the particle with momentum p from the mirror boundary. Straightforward differentiation shows that the forces induced by the mirror boundaries are *twice* the usual ones. Accordingly, to retain energy conservation the potential energy for interactions crossing the mirror surface must be doubled,

$$\mathcal{H} \supset \sum_{\mathrm{mirror}} 2\phi(2|r|) \;.$$

To illustrate these ideas, consider first the example of a single harmonic oscillator, with an equilibrium spacing d so that the oscillator potential energy is $\kappa(r-d)^2/2$. For a *fixed* mirror particle, fixed at x = -d/2, a moving twin oscillates about x = +d/2 at a frequency $\sqrt{\kappa/m}$, where m is the particle mass. If, on the other hand, the mirror image moves too, cooperatively, the force is twice as great (as that computed from $-\nabla\phi$):

$$\mathcal{H} = \frac{1}{2} \left[\frac{p_1^2}{m} + \frac{p_2^2}{m} + \kappa (x_1 - x_2 - d)^2 \right] \longrightarrow$$

$$\dot{x}_1 = \frac{p_1}{m} ;$$

$$\dot{p}_1 = -\kappa(x_1 - x_2 - d) = -2\kappa[x_1 - (d/2)],$$

so that the oscillation frequency exceeds that of the fixed-particle case, $\omega_{\rm mirror}=\sqrt{2}\omega_{\rm fixed}$.

154

Initial and Boundary Conditions, Interpolation



The various types of boundary conditions can usefully be illustrated (see Figure 5.5) and distinguished for a harmonic chain with nearestneighbor interactions. For definiteness, consider a four-particle chain with Particles 1, 2, 3, and 4 confined by a left boundary at x = -2 and a right boundary at x = +2. The force-constant matrix elements $\kappa_{ij} = \nabla_i \nabla_j \mathcal{H}$ (for simplicity we choose the nearest-neighbor Hooke's-Law force constant equal to unity) and their corresponding eigenvalues $\{\lambda\}$ for "free" and for

Implementing Periodic and Mirror Boundaries

"periodic" boundaries are as follows :

$$\kappa_{\rm free} = \begin{pmatrix} +1 & -1 & 0 & 0\\ -1 & +2 & -1 & 0\\ 0 & -1 & +2 & -1\\ 0 & 0 & -1 & +1 \end{pmatrix} \longrightarrow \begin{pmatrix} \lambda(\lambda-2)(\lambda^2 - 4\lambda + 2) = 0\\ \text{or}\\ \{\lambda\} = \{0.000, 0.586, 2.000, 3.414\} \end{pmatrix} ;$$

$$\kappa_{\text{periodic}} = \begin{pmatrix} +2 & -1 & 0 & -1 \\ -1 & +2 & -1 & 0 \\ 0 & -1 & +2 & -1 \\ -1 & 0 & -1 & +2 \end{pmatrix} \longrightarrow \begin{pmatrix} \lambda(\lambda-2)^2(\lambda-4) = 0 \\ \text{or} \\ \{\lambda\} = \{0.000, 2.000, 2.000, 4.000\} \end{pmatrix}$$

The additional force-constant contributions, $\kappa_{14} = \kappa_{41}$, as well as the increases in κ_{11} and κ_{44} , all come from the periodic-image potential :

$$\phi_{14} = \frac{1}{2}(x_1 + 4d - x_4)^2$$
,

which describes the periodic interaction between Particles 1 and 4 and increases the trace of the force-constant matrix from 6 to 8. The matrix trace is equal to the sum of the matrix eigenvalues. Thus this increased trace indicates higher frequencies—the eigenvalues of these matrices are equal to the *squares* of the normal-mode oscillation frequencies consistent with the chosen boundary conditions.

"Rigid" boundary conditions can be implemented by adding two additional particles at $x = \pm 2.5$, each fixed at its end of the chain. "Mirror" boundaries also introduce extra particles, but with their motions *anticorrelated* with the nearest neighbor "inside" the system. The matrices and eigenvalues corresponding to rigid and mirror boundary conditions are :

$$\kappa_{\text{rigid}} = \begin{pmatrix} +2 & -1 & 0 & 0 \\ -1 & +2 & -1 & 0 \\ 0 & -1 & +2 & -1 \\ 0 & 0 & -1 & +2 \end{pmatrix} \longrightarrow \begin{pmatrix} (\lambda^2 - 3\lambda + 1)(\lambda^2 - 5\lambda + 5) = 0 \\ \text{or} \\ \{\lambda\} = \{0.382, 1.382, 2.618, 3.618\} \end{pmatrix} ;$$

$$\kappa_{\text{mirror}} = \begin{pmatrix} +3 & -1 & 0 & 0 \\ -1 & +2 & -1 & 0 \\ 0 & -1 & +2 & -1 \\ 0 & 0 & -1 & +3 \end{pmatrix} \longrightarrow \begin{pmatrix} (\lambda - 2)(\lambda - 4)(\lambda^2 - 4\lambda + 2) = 0 \\ \text{or} \\ \{\lambda\} = \{0.586, 2.000, 3.414, 4.000\} \end{pmatrix} .$$

Initial and Boundary Conditions, Interpolation

Of all of these choices periodic boundaries, with a center-of-mass velocity of zero, are most commonly used. Only in this periodic case are all particles treated alike.

Exactly similar ideas can be applied to boundaries for smooth-particle versions of continuum mechanics. For actual boundaries fixed in space the rigid and mirror choices both work well. In the calculations of \dot{v} and \dot{e} using mirror boundaries, the pair contributions to both time derivatives, proportional to $\nabla_i w_{ij}$, are doubled relative to the contributions which would correspond to fixed image particles. A consistent treatment of the energy including these factors of two results in a perfectly conserved energy.

The SPAM atmospheric-equilibrium problems at the end of Chapter 3 illustrate simple applications of these ideas. Particles close enough to the boundary to interact with their (moving) images undergo twice the usual accelerations. If, at the same time, the internal energies of the image particles are included in the total energy sum, energy overall is nicely conserved.

5.5 Alternative Meshes—Regular Lattices

Any of the wide variety of regular crystal lattices can be used as the basis of a smooth-particle mesh. In two dimensions the simplest choices are the square lattice (four nearest neighbors at a separation $\sqrt{V/N}$) and the "close-packed" triangular lattice (six nearest neighbors at a separation of $\sqrt[4]{4/3}\sqrt{V/N}$). In Section 3.6 we showed that either lattice reproduces the correct mass density within a fraction of a percent provided that the weight function's range h exceeds $\sqrt{6V/N}$.

In three dimensions the variety of lattices is even greater. The simple cubic and body-centered cubic lattices have six and eight nearest neighbors, respectively, while the face-centered cubic and hexagonal closest packed have 12. The accuracy of a particular simulation can be judged by comparing the computed results with those of a second simulation using a different initial grid. It might appear that the criterion of simplicity could be used to favor one lattice over another, but the finding that the density errors are complicated functions of the lattice structure suggests instead that this question has no simple answer.

Two significant properties of the various lattices are (i) their elastic stability and (ii) their isotropy, or lack of it. In atomistic simulations one can ask whether or not a particular lattice is stable to small perturbations. The answer depends upon the forcelaw. Stability for continuum simula-

Elastic Stability of Embedded-Atom Lattices

tions using smooth-particle meshes hinges on the form of the constitutive equations. We elucidate this topic for the simplest case in the following section, the relative stabilities of the square and triangular lattices.

5.6 Elastic Stability of Embedded-Atom Lattices

Using SPAM to represent fluids or isotropic solids could be problematic if the crystal strengths of the underlying lattices prevented flow or introduced substantial anisotropy (or anisotropicity!). It is necessary to quantify crystal strength and anisotropy on a case-by-case basis. To give some guidance we consider three simple examples here, the one-dimensional chain and two two-dimensional lattices. We point out that that either straightforward particle dynamics or lattice dynamics can be applied to analyze these problems. The same ideas can be applied to other structures.

The mechanical stability of a periodic lattice can be assessed either analytically or numerically. The analysis is standard lattice dynamics, as described (for instance) in Max Born and Kerson Huang's 1954 text.¹ Their analysis proceeds by finding the eigenvectors and eigenvalues of the "dynamical matrix"—the matrix composed of the second derivatives $\nabla \nabla \Phi$ of the total potential Φ using periodic boundary conditions. The "periodic" (or "cyclic") boundaries reduce the matrix problem from a large $3N \times 3N$ matrix to N small 3×3 matrices (for three space dimensions, so long as there is only a single particle in the periodic cell used to build up the crystal). This reduction is significant because the computational cost of the large matrix problem varies as the *cube* of the number of particles involved. The corresponding reduced one- and two-dimensional crystals require only the small matrix solutions of 1×1 and 2×2 matrices, respectively. For an irregular arrangement of nodes, or for a regular arrangement lacking periodicity, no such simplification applies.

The simplest lattice structure is the one-dimensional periodic one with unit masses separated from their neighbors by unit distance. We make this example simpler still by choosing the simplest possible potential function relevant to SPAM :

$$\Phi = \sum \phi_i \; ; \; \phi_i = \frac{1}{2} (\rho_i - \rho_0)^2 \; .$$

Here ρ_0 is the calculated particle density at each particle for the equally spaced lattice, with the range of Lucy's weight function equal to 2 (times

¹Born and Huang, Dynamical Theory of Crystal Lattices (reprinted in 1985).

Initial and Boundary Conditions, Interpolation

the nearest-neighbor spacing of unity) so that each particle interacts only with its two nearest neighbors :

$$w_{i,i\pm 1} = \sum \frac{5}{8} \left[1 - \frac{3}{2} x_{i,i\pm 1}^2 + |x_{i,i\pm 1}^3| - \frac{3}{16} x_{i,i\pm 1}^4 \right] .$$

Note the absolute value bars $|\ldots|$ on the cubic term. These serve as a reminder that the weight function is an *even* function of the particle separation, $w(+r) \equiv w(-r)$.

The cyclic boundary condition includes a link between the last particle in the chain and the first—see again Figure 5.5—so that all N particles have two nearest neighbors :

$$w_{1,N} \to w_{N,N+1} = w(x_1 - x_N + N)$$
.

In this perfect periodic lattice the density at each particle is the same :

$$\rho_0 = w(1) + w(0) + w(1) = \frac{5}{8} \left[\frac{5}{16} + 1 + \frac{5}{16} \right] = \frac{65}{64}.$$

The *first* derivatives of the potential all have the form :

$$\nabla_i \Phi = +(\rho_i + \rho_{i+1} - 2\rho_0) \frac{15}{32} - (\rho_i + \rho_{i-1} - 2\rho_0) \frac{15}{32} = (\rho_{i+1} - \rho_{i-1}) \frac{15}{32} ,$$

which follows from the first derivative of the weight function :

$$w'(r=1, h=2) = -\frac{15}{32}$$

The coefficients ($\rho_{i+1}-\rho_{i-1}$) in the first derivatives all vanish in the perfect crystal, where $\rho_i \equiv \rho_0$, so that only the diagonal (i, i) and second-neighbor ($i, i \pm 2$) second derivatives are nonvanishing :

$$\nabla_i \nabla_i \Phi = \frac{15}{32} (\nabla_i w_{i,i+1} - \nabla_i w_{i,i-1}) = \frac{225}{512} ;$$
$$\nabla_{i\pm 1} \nabla_i \Phi = 0 ; \ \nabla_{i\pm 2} \nabla_i \Phi = -\frac{225}{1024} .$$

With periodic boundary conditions the assumed solution for stable λ and- ω (space-and-time) periodic particle displacements is:

$$u_n = \exp(ikn - i\omega t)$$
; $k \equiv 2\pi/\lambda$.

The motion equation and its solution for Particle "0" are typical:

$$\ddot{u}_0 = \frac{225}{1024} [u_{+2} - 2u_0 + u_{-2}] \longrightarrow$$

Elastic Stability of Embedded-Atom Lattices

$$-\omega^2 u_0 = \frac{225}{1024} u_0 [2\cos(2k) - 2] = C u_0 [2\cos(2k) - 2] \longrightarrow$$

$$\omega = \frac{15}{16}\sin(k) = 2\sqrt{C}\sin(k) \ ; \ C = \frac{225}{1024}$$

Although this dispersion relation "looks like" the usual one for stable onedimensional lattice nearest-neighbor dynamics it is actually quite an *un-usual* solution—there is no coupling (in the harmonic approximation) between even and odd-numbered particles. By arranging the odd and even rows and columns of $\nabla \nabla \Phi$ into separated blocks this factorization is obvious. The periodic six-particle case is the simplest generic example :

$$\begin{bmatrix} 2C & 0 & -C & 0 & -C & 0 \\ 0 & 2C & 0 & -C & 0 & -C \\ -C & 0 & 2C & 0 & -C & 0 \\ 0 & -C & 0 & 2C & 0 & -C \\ -C & 0 & -C & 0 & 2C & 0 \\ 0 & -C & 0 & -C & 0 & 2C \end{bmatrix} \longrightarrow \begin{bmatrix} 2C & -C & -C & 0 & 0 & 0 \\ -C & 2C & -C & 0 & 0 & 0 \\ 0 & 0 & 0 & 2C & -C & -C \\ 0 & 0 & 0 & -C & 2C & -C \\ 0 & 0 & 0 & -C & -C & 2C \end{bmatrix}.$$

Dynamical simulations of a periodic chain confirm the theoretical dispersion relation and also show that the (exact anharmonic) motion is stable with a "superperiod" orders of magnitude longer than that just calculated. Thus the one-dimensional periodic chain *is stable* to small perturbations.

Let us turn next to an analysis of the two-dimensional case. This time we begin with a *numerical* approach to stability analysis. Figure 5.6 shows particle trajectories for a square periodic 8×8 system with initial small displacements in the range $-0.05 < (\delta x, \delta y) < +0.05$ and summing to zero. Timesteps of dt = 0.01 or 0.02 lead to essentially identical trajectories with nine-digit conservation of energy. Evidently the square lattice structure with h = 3 is *unstable*. Simulations with $-0.005 < (\delta x, \delta y) < +0.005$, not shown here, confirm this conclusion.

160

Initial and Boundary Conditions, Interpolation



Figure 5.6: Particle trajectories (for times of 200 and 1000) using only the embedded-atom potential ,

 $\Phi = (1/2) \sum_{i} (\rho_i - \rho_0)^2$,

with h=3 with random initial displacements in the range ± 0.05 .

We can, of course, show this same instability directly by analyzing the dynamical matrix for the lattice. All of the nonvanishing second derivatives can easily be calculated numerically, by evaluating finite-difference approximations :

$$\Delta^2 \nabla_i \nabla_i \Phi = +\Phi(x_i - \Delta) - 2\Phi(x_i) + \Phi(x_i + \Delta) ;$$

$$4\Delta^2 \nabla_i \nabla_j \Phi = \frac{-\Phi(x_i - \Delta, x_j + \Delta) + \Phi(x_i + \Delta, x_j + \Delta)}{+\Phi(x_i - \Delta, x_j - \Delta) - \Phi(x_i + \Delta, x_j - \Delta)}$$

Though this notation *appears* purely one-dimensional we intend that the indices i and j range over all 2N coordinates ,

$$1 \leq i, j \leq 2N$$
,

so that *all* the matrix elements, with the various forms $\{\nabla_x^2, \nabla_x \nabla_y, \nabla_y^2\}$, are included here. Just as in the numerical example, consider first the square lattice, at unit density. Offsets (dx, dy) of either 0.001 or 0.0005 relative to the regular-lattice coordinates give eigenvalues of the resulting approximate dynamical matrix with six-figure accuracy. For stability the matrix eigenvalues (apart from the two corresponding to center-of-mass

Elastic Stability of Embedded-Atom Lattices

motion) should all be positive. Vanishing eigenvalues indicate instability. For the square lattice, half the eigenvalues are exactly zero, indicating that the lattice is indeed unstable to shear, and can flow relatively easily in the absence of constraining boundary forces. Easy flow is a desirable property for any fluid-phase model.

Normal mode analysis for a two-dimensional monatomic solid leads to two "branches" of the dispersion relation, $\omega_L(k)$, with N longitudinal modes, and $\omega_T(k)$, with N transverse modes. The transverse modes, with individual particle motions perpendicular to the propagation direction, should be absent in a fluid model, which supports only bulk compressional waves. It is a relatively simple, and still feasible, exercise to diagonalize the $\nabla\nabla\Phi$ matrix for a simple cubic lattice of $6 \times 6 \times 6 = 216$ points. A detailed investigation, with h = 3 so that each particle interacts with 124 neighbors, reveals that 432 of the 648 normal-mode frequencies are precisely zero.

It is an interesting exercise to extend the stability analysis just outlined for the square and cubic lattices to the triangular lattice. In this lattice each particle has six nearest neighbors (at a separation of $r = \frac{4}{3}^{\frac{1}{4}} \simeq 1.07457$), six second neighbors (at 1.86121), six third neighbors (at 2.14914), and twelve fourth neighbors (at 2.84304). This lattice too has a high degree of symmetry and very-low-kinetic-energy molecular dynamics reveals that it too is unstable to deformation for h = 3. It is easy to confirm this conclusion by calculating the second derivatives of the potential energy with respect to the x and y displacements. These then give the eigenvalues, just as in the square-lattice case. The triangular-lattice static lattice density at each particle is 1.00222807 using Lucy's weight function, slightly less than the square-lattice density of 1.00293003 .

The triangular lattice can be stabilized by using a special choice for h which reproduces a density of unity. As we pointed out in Figure 3.3 the choice h = 2.41266 reproduces that density to ten-digit accuracy. The dynamics of a triangular lattice with that choice of h shows that the lattice is mechanically stable, with the particles executing small oscillations about their lattice sites. Evidently the nonlinear stability of lattices is sensitive to the kinetic temperature of the motion ($\propto \langle v^2 \rangle$). To ensure that these velocities are sufficiently "small" for a meaningful stability test the "Einstein frequency" of the lattice should be determined. This is the oscillation frequency of a single particle with all of its neighbors fixed. An initial displacement of a single particle, dx = 0.001, results in oscillations with a period of about $\tau = 28.24$, leading to an estimate of the corresponding

August 16, 2006 17:44

Initial and Boundary Conditions, Interpolation

force constant κ . The results are essentially the same for either the square or triangular lattice :

$$28.24 = \tau = \frac{2\pi}{\omega} \to \omega = \sqrt{\frac{\kappa}{m}} = 0.222 \to \kappa_{\text{triangular}} = \kappa_{\text{square}} = 0.0495 \; .$$

To an accuracy of better than one part in a thousand, these results are unchanged if the initial displacement is doubled, dx = 0.002. Evidently particular lattices can be stabilized to a greater or lesser degree by corresponding choices of the weight function range.

5.7 Invariant Curvature Crystal Stabilization

Consider modeling *solids* with SPAM. Then it is desirable to stabilize regular crystalline structures in order to construct smooth particle meshes with mechanical strength. Constitutive relations which depend simply on density or its gradient cannot do this.² A regular crystalline structure has symmetry. In the simplest case not only does the density gradient at each particle vanish ,

$$\{ (\nabla \rho)_i = 0 \},\$$

but also the second derivative, $\nabla \nabla \rho$, has the spatial symmetry of the lattice. In two space dimensions the rotationally invariant combination of squared second density derivatives,

$$\Phi_{\nabla\nabla\rho} \equiv \frac{1}{2} (\rho_{xx} - \rho_{yy})^2 + 2\rho_{xy}^2 ;$$

$$\rho_{xx} = \frac{\partial^2 \rho}{\partial x^2} ; \ \rho_{yy} = \frac{\partial^2 \rho}{\partial y^2} ; \ \rho_{xy} = \frac{\partial^2 \rho}{\partial x \partial y} .$$

vanishes for regular lattices (such as the square, triangular, and hexagonal lattices). Note the parallel to the invariant shear stress of Section 2.8 . Evidently a shear of these lattices causes changes in one or more of the derivatives so that $\Phi_{\nabla\nabla\rho}$ can provide an elastic resistance to shear.

Consider the triangular lattice at unit number density as an example. For small strains, the energy change $\Delta \Phi$ from either of the shear deformations ,

$$\epsilon_{xx} = -\epsilon_{yy} = \frac{\epsilon}{2} \longrightarrow$$

ws-book9x6

 $^{^{2}}$ Hoover and Hoover (2006).

Invariant Curvature Crystal Stabilization

$$\frac{\Delta \Phi}{V} = \frac{1}{2} \epsilon^2 [C_{11} - C_{12}] = \epsilon^2 G ,$$

or

$$\epsilon_{xy} = \epsilon \longrightarrow$$
$$\frac{\Delta \Phi}{V} = \frac{1}{2} \epsilon^2 C_{44} = \frac{1}{2} \epsilon^2 G ,$$

corresponds to the *same* shear modulus G (because the triangular lattice is "elastically isotropic"). Using the smooth weight function (*three* continuous derivatives, so that the force from $\Phi_{\nabla \nabla \rho}$ is continuous),

$$w_{\text{smooth}}(\tilde{r} = \frac{r}{h} < 1) = \frac{7}{\pi h^2} (1 - \tilde{r})^4 (1 + 4\tilde{r}) ,$$

with a series of small strains $\{\epsilon \to 0\}$, shows that there is a *quadratic* dependence of energy on strain, giving an elastic shear modulus,

$$G = \eta = \frac{\partial^2 \Phi_{\nabla \nabla \rho}}{V \partial \epsilon^2} \; .$$

Because the smooth weight function falls off more rapidly with distance than does Lucy's, our usual choice, h = 3, gies a relatively small shear modulus:

$$w_{\text{smooth}}(\tilde{r} = \frac{r}{3} < 1) = \frac{7}{9\pi}(1 - \tilde{r})^4(1 + 4\tilde{r}) \longrightarrow$$

$$G(h=3) = \eta(h=3) = \frac{\partial^2 \Phi_{\nabla \nabla \rho}}{V \partial \epsilon^2} = 0.0082 .$$

Figure 5.7 shows the variation of the shear modulus with h for the elastically-isotropic triangular lattice. We will make use of this stabilization example in Chapters 8 and 9, where problems involving nonlinear solid deformation are considered.



5.8 Example: Heat Transfer in One Dimension with SPAM

The simplest nonequilibrium problem driven by boundary interactions is the pure conduction problem studied in Section 2.10—steady heat transfer between a hot and a cold reservoir. For equally spaced particles with a constant temperature gradient and a constant thermal diffusivity (we choose both equal to unity here to simplify the illustration), the simplest versions of the corresponding smooth particle equations are :

$$Q = -\nabla T \longrightarrow \left\{ Q_i = +\sum_j [T_i - T_j] \frac{dw_{ij}}{dx_i} \right\} ;$$

164

 $\label{eq:initial and Boundary Conditions, Interpolation$

Example: Heat Transfer in One Dimension with SPAM

$$\dot{T} = -\nabla \cdot Q \longrightarrow \left\{ \dot{T}_i = -\sum_j [Q_i + Q_j] \frac{dw_{ij}}{dx_i} \right\} ,$$

where T is temperature and Q is heat flux. As we emphasized in Chapter 3, the sums and differences on the righthandsides of these relations guarantee two desirable properties of their joint solution: (i) energy is conserved exactly and (ii) the heat flux vanishes when all the particle temperatures are identical.

Provided that the range h of the weight function w(r < h) is sufficiently large, a constant temperature gradient should lead to the heat flux given by Fourier's Law, $Q = -\kappa \nabla T$. We have already arbitrarily chosen the proportionality constants equal to unity. We further choose the particle temperatures to correspond to unit temperature gradient, { $T(x) \equiv x$ }, so that unit thermal diffusivity should give a large-h limiting heat flux of Q = -1. A numerical evaluation of the heat flux using the one-dimensional form of Lucy's weight function :

$$w(\tilde{r} = \frac{r}{h} < 1) = \frac{5}{4h}(1 - \tilde{r})^3(1 + 3\tilde{r}) ,$$

gives the results :

$$Q_{h=2} = -\frac{15}{16} ; Q_{h=3} = -\frac{80}{81} ; Q_{h=4} = -\frac{255}{256} ,$$

for weight-function ranges of 2, 3, and 4. This suggests the exact result :

$$Q(h) = -1 + h^{-4}$$

In fact this surprisingly simple closed form can be derived from the Euler-Maclaurin "sum formula", which relates the sum of $x_{ij}dw_{ij}/dx_i$ over all particles j to the corresponding integral plus an infinite series of Bernoulli-Number corrections. These "BNC" corrections involve the various derivatives of the integrand at the integration endpoints:

$$Q(h) = \sum xw' = \int_{-h}^{+h} xw'dx + BNC = -1 + h^{-4}.$$

The polynomial form of Lucy's weight function guarantees that only a finite number (exactly one in this case) of Bernoulli-number corrections to the integral contribute to the sum of heat fluxes.

This very favorable convergence of the particle sum to the onedimensional continuum flux $Q \longrightarrow -1$ suggests trying the same technique

August 16, 2006 17:44

166

Initial and Boundary Conditions, Interpolation

in two and three space dimensions. Accordingly, we have studied the corresponding two-dimensional problem,³ using the two-dimensional form of Lucy's weight function :

$$w(\tilde{r} = \frac{r}{h} < 1) = \frac{5}{\pi h^2} (1 - \tilde{r})^3 (1 + 3\tilde{r}) .$$

The improvement with increasing h in two dimensions is unfortunately not only less simple, but also much less dramatic than in one. A Rayleigh-Bénard problem treating the details of a two-dimensional heat flow is described at the end of this Chapter.

Detailed investigation of one-dimensional heat flow, with h = 2, reveals that the heat-transfer equation for \dot{T} has the same form as does the equation of motion, for \ddot{x} , considered in the last Section, so that their even-odd instabilities are identical. In both cases the smooth-particle algorithm has no direct effect on a relatively-benign even-odd instability. This instability is most easily understood by analyzing a shortest-wavelength perturbation $\pm dT$ for temperature deviations from the mean which alternate sign in space :

$$T_{\text{odd}} = T + dT$$
; $T_{\text{even}} = T - dT$

In such a case the heat flux at each particle vanishes (by symmetry) so that there is no heat flux divergence to cause the decay of this mode. This lack of decay suggests that a different empirical temperature evolution be used instead :

$$\dot{T}_i = C \sum_j (T_j - T_i) w_{ij} \; .$$

The proportionality constant C needs to be chosen to agree with the decay rate for a small $k=2\pi/\lambda$ (where the wavelength is λ) long-wavelength temperature perturbation.

To evaluate the decay constant C , consider the decay, according to Fourier's law, of a cosine temperature perturbation :

$$\delta T \propto \exp^{-t/\tau} \cos(kx)$$
.

The exact continuum decay time τ is related to the macroscopic thermal diffusivity D_T :

$$\delta \dot{T} = -\delta T/\tau = D_T \nabla^2 \delta T = -D_T k^2 \delta T \longrightarrow \frac{1}{\tau} = D_T k^2 = D_T \left(\frac{2\pi}{\lambda}\right)^2 .$$

ws-book9x6

³Kum, Hoover, and Hoover (2003).

Example: Periodic Shear Flow with SPAM

Consider now the long-wavelength case of the cosine perturbation, keeping terms of order k^2 in the smooth-particle approximation to

$$\dot{T}_0 = C \sum_j (T_j - T_0) w_{ij} :$$

$$\delta T \propto \cos(kx) \simeq 1 - \frac{1}{2} (kx)^2 ,$$

where the origin corresponds to a particle with the maximum possible temperature $T_0=\delta T$. The temperature evolution equation for that particle becomes

$$\dot{T}_0 = -CT_0 \sum_{-h}^{+h} \frac{1}{2} (kx)^2 w(x) = -\frac{T_0}{\tau} ,$$

which agrees with the macroscopic constitutive relation provided that ${\cal C}$ is chosen as follows :

$$C = \frac{D_T}{\sum \frac{x^2 w(x)}{2}} \; .$$

The error incurred by this algorithm can be estimated by evaluating the k^4 contribution to the decay rate. It might be thought that an even *better* algorithm would evaluate the temperature evolution using a model for the *second* derivative of temperature :

$$\dot{T}_i = C \sum_j (T_j - T_i) w_{ij}'' \; .$$

But in the one-dimensional case this second-derivative approach leads to an error (from the k^4w'' neighbor sum) about twice as large. The choice of w rather than w'' is superior. This is because w (0 < x < h) is always positive, while w'' changes sign (at x = h/3).

5.9 Example: Periodic Shear Flow with SPAM

The simplest conceivable boundary condition is periodic, with the system surrounded by periodic replicas of itself. Evidently a static periodic structure arranged in this way has no atypical surface region. The same idea can be applied to a deforming system undergoing simple shear flow by using periodic images in relative motion.⁴ To illustrate, consider the case of

 $^{^{4}}$ As was applied to molecular dynamics in the 70s. See Ashurst and Hoover (1975).
ws-book9x6

Initial and Boundary Conditions, Interpolation

simple shear, with the x velocity varying linearly with the y coordinate, $\dot{x}=\dot{\epsilon}y$, where $\dot{\epsilon}$ is the strain rate ,

$$\dot{\epsilon} = d\dot{x}/dy$$
.

An $L \times L$ periodic system, with portions of eight nearby periodic images, is shown in Figure 5.8. To construct it we have chosen the ideal-gas equation of state $P = \rho^2/2$ with particles of unit mass, so that the smooth-particle equations of motion are identical to those of molecular dynamics with a pairwise-additive potential function :

$$\Phi = \sum_{i < j} \left[\phi_{ij} = w(r_{ij}) \right] \; .$$

The equations of motion of the image particles above and below the central periodic box are exactly the same as those in the central box :

$$\ddot{x}_i = -\sum_j x_{ij} w' / r_{ij} \; ; \; \ddot{y}_i = -\sum_j y_{ij} w' / r_{ij} \; .$$

Because w'(r) vanishes linearly for small r, these accelerations are well behaved as $r \to 0$. The periodic images of the shearing particles differ in their laboratory-frame velocities and coordinates, as follows :

$$\dot{x}_{\text{image}} = \dot{x} \pm \dot{\epsilon}L \quad \longleftarrow \quad y_{\text{image}} = y \pm L;$$

$$x_{\text{image}} = x \pm \dot{\epsilon} L t \quad \longleftarrow \quad y_{\text{image}} = y \pm L \; .$$



Example: Periodic Shear Flow with SPAM

The example shown in the Figure is based on 10^2 particles, and uses the polytropic equation of state considered in Section 3.11. For simplicity we have chosen the density ρ , the strain rate $\dot{\epsilon}$, and the particle mass mall equal to unity. The range of Lucy's weight function is h = 3. The equations of motion are simplest when written in a frame moving with the average flowrate. If we introduce relative velocities $\{v\}$, with

$$v = \dot{r} - \langle \dot{r} \rangle$$
; $\langle \dot{r} \rangle = (\dot{\epsilon}y, 0)$,

then the equations of motion are as follows :

$$\dot{x} = v_x + \dot{\epsilon}y \; ; \; \dot{y} = v_y \; ; \; \dot{v}_i = -m \sum_j [(P/\rho^2)_i + (P/\rho^2)_j] \cdot \nabla_i w_{ij} \; .$$

The nonequilibrium momentum flux $P_{xy} = -\eta \dot{\epsilon}$ for the flow defines an intrinsic shear viscosity η and is also a measure of the rate at which irre-

ws-book9x6

Initial and Boundary Conditions, Interpolation

versible work is done in driving the flow within a volume V:

$$w = \int_0^t \dot{w} dt' = -\dot{\epsilon} \int_0^t V P_{xy} dt' ,$$

for a time t. Figure 5.9 shows the time dependence of the fluid's instantaneous shear stress (left) and internal energy (right) for a smaller strain rate. The strain rate in this example is 0.05. The overall number density is unity, with $N=V=48\times 48=2304$. The shear stress increase, from 0.025 to 0.25 corresponds to a viscosity increase from 0.5 to 5.0 along with an energy-per-particle increase from 0.5 to 1.0.

Shear of an inviscid ideal gas is actually an infinite-Reynolds'-number unstable flow, but here the flow is stabilized by an intrinsic numerical viscosity. The effective Reynolds' number of the simulation, $\rho L_x \dot{\epsilon} L_y / \eta$ is actually of order unity. The intrinsic viscosity is due completely to the finite SPAM-particle size and can alternatively be estimated on the basis of mean free path theory or from the Green-Kubo theory of Section 2.3⁵ Unless the physical viscosity exceeds the intrinsic viscosity the latter dominates, and helps to stabilize, nonequilibrium flows. There is also an intrinsic heat conductivity, which can similarly be measured or estimated.⁶



 $^{^5\}mathrm{Green}$ and Kubo's 1950s results are reviewed by Zwanzig (1965).

 $^{^{6}}$ Hoover and Posch (1996).

Example: Rayleigh-Bénard Flow with SPAM

5.10 Example: Rayleigh-Bénard Flow with SPAM

Lord Rayleigh analyzed thermally-driven convection experiments carried out by Claude Bénard. In the "Rayleigh-Bénard problem" named for them, convective rolls are driven by applying a temperature gradient across an enclosed system in the presence of a gravitational field. Two snapshots of smooth-particle simulations, both using 5000 particles and Lucy's weight function, appear in Figure 5.10 below.



Figure 5.10: Rayleigh-Bénard flow with 5000 smooth particles, initially in a 100 × 50 square-lattice array. The kinematic viscosity and thermal diffusivity are both 0.25. Both the gravitational field strength and the bottom-less-top temperature gradient are 1/H, so that the Rayleigh Number, $H^3g/(\nu\kappa)$, is 40,000. The appearance of flows, using either (i) static (upper) or (ii) dynamic mirror (lower) boundary conditions, as described in the main text, are similar. See Figure 2 of Reference [3].

ws-book9x6

Initial and Boundary Conditions, Interpolation

We considered an accurate Eulerian (fixed-mesh) finite-difference solution of this problem in Section 2.12 . In our earliest investigations of the Rayleigh-Bénard problem using smooth particles we used external mirrorimage particles at the system boundaries, giving each mirror particle the temperature (T_H or T_C) and velocity (v = 0) associated with the nearby boundary. This approach leads to flow fields agreeing with accurate continuum solutions within a few percent when the number of particles used is a few thousand. It is of course essential that the transport coefficients exceed the intrinsic coefficients mentioned in the last Section. This "static-mirror" (with static indicating fixed values of temperature and velocity) is illustrated on the right side of Figure 5.11, and is evidently an improvement over using fixed particles to form a boundary layer. The fixed-particle approach appears on the left side of the Figure.



Figure 5.11: Smooth-particle Rayleigh-Bénard simulations were carried out with the two types of boundary conditions shown here. At the left the boundary particles are given fixed "boundary" values of velocity and temperature, with their position mirroring bulk particles. On the right the mirror-image boundary particles are assigned velocities and temperatures such that the mean values of each bulk-mirror pair are equal to the desired boundary values. See Figure 1 of Reference [3].

Later, our experience with the one-dimensional heat flow problem analyzed in Section 5.5 suggested that we instead use dynamic mirror-image temperatures and velocities which provide the correct temperature and velocity on the boundary :

 $(T, v)_{\text{mirror}} + (T, v)_{\text{interior}} \equiv 2(T, v)_{\text{boundary}}$.

Example: Rayleigh-Bénard Flow with SPAM

These choices for the mirror properties insure that the interpolated temperature and velocity *on* the boundary have their prescribed values. This choice for the temperature also implies that the heat fluxes parallel and perpendicular to the boundary satisfy the two relations :

$$Q_{\text{mirror}}^{\parallel} + Q_{\text{interior}}^{\parallel} \equiv 0 \; ; \; Q_{\text{mirror}}^{\perp} = Q_{\text{interior}}^{\perp}$$

This dynamic mirror approach is illustrated on the right side of Figure 5.11.



Figure 5.12: Time development of the kinetic energy for steady Rayleigh-Bénard flow. The accurate grid-based continuum energy is compared to two smooth-particle simulations using both static and dynamic boundary conditions. The initial conditions for the flow are the same as those used in Chapter 2 :

 $v_x \propto +\sin(kx)\sin(ky)$; $v_y \propto +\cos(\frac{3}{2}kx)\cos(\frac{1}{2}ky)$,

with the temperature varying linearly between the two specified boundary values. See Reference [3].

Numerical implementations of (i) static and (ii) dynamic mirror boundary conditions do lead to significantly different results. But the difference between the two choices, illustrated here for the total kinetic energy of the flow field as it approaches the steady-state value, is small with respect to

ws-book9x6

Initial and Boundary Conditions, Interpolation

the deviation from an accurate continuum simulation based on a converged square grid. Figure 5.12 compares the kinetic energy history for the static and dynamic boundary conditions as well as for the accurate fixed-grid continuum solution taken from Section 2.13 .

We have to conclude that the dramatic improvement in convergence found in one dimension has no simple analog in two (or three) space dimensions. We confirmed this conclusion numerically by studying pure conductive heat flow in two space dimensions for a constant temperature gradient, using both square and triangular lattices of fixed particles, but with dynamic-mirror temperatures. Although convergence to the continuum limit occurs smoothly and stably in either case there is no simple powerlaw dependence of heat flux on the gradient. Oscillations (as a function of h) both above and below the correct continuum result can be observed. We checked also that the specially smooth weight function with *three* vanishing derivatives at h,

$$w_{\text{smooth}} \left(\tilde{r} = \frac{r}{h} < 1 \right)_{2D} = \frac{7}{\pi h^2} \left(1 - \tilde{r} \right)^4 \left(1 + 4\tilde{r} \right) \;,$$

rather than just the two derivatives given by Lucy's form, made no qualitative change to these results $.^3$

 $^{^3\}mathrm{Kum},$ Hoover, and Hoover (2003).

References

5.11 References

- (1) M. Born and K. Huang, *Dynamical Theory of Crystal Lattices* (Clarendon, Oxford, 1985).
- (2) Wm. G. Hoover and C. G. Hoover, "Smooth-Particle Phase Stability with Generalized Density-Dependent Potentials", Physical Review E 73, 016702 (2006).
- (3) O. Kum, W. G. Hoover, and C. G. Hoover, "Smooth Particle Boundary Conditions", Physical Review E 68, 017701 (2003).
- (4) W. T. Ashurst and W. G. Hoover, "Dense-Fluid Shear Viscosity via Nonequilibrium Molecular Dynamics", Physical Review A 11, 658-678 (1975).
- (5) R. W. Zwanzig, "Time Correlation Functions and Transport Coefficients", Annual Review of Physical Chemistry 16, 67-102 (1965).
- (6) Wm. G. Hoover and H. A. Posch, "Numerical Heat Conductivity in Smooth Particle Applied Mechanics", Physical Review E 54, 5142-5145 (1996).

August 16, 2006 17:44

WSPC/Book Trim Size for 9 in x 6 in ws-book9x6

176

Initial and Boundary Conditions, Interpolation

Chapter 6

Convergence and Stability

/ Summary / Existence and Uniqueness / Accuracy and Precision / Time Integration Errors / Short-Time Runge-Kutta Analysis / Long-Time Lyapunov Analysis / Expectations from Molecular Dynamics / Smooth-Particle Spatial Integration Errors / Lattice Instability / Even-Odd Instability /

/ 8 Figures /

Example Problem : [Shear-Flow Convergence]

6.1 Summary

The computational methods we develop in this book are designed to give approximate, but useful, solutions of the partial differential evolution equations of mathematical physics. Our goal is to predict the future by solving these equations. In this Chapter we explore the *quality* of smooth-particle integration algorithms—both the convergence of the solutions to "true" solutions and the stability of the results to small perturbations. The straightforward approach to these questions is to perform series of simulations, comparing results for different numbers of particles, different mesh structures, different boundary treatments, different forms of the weight function, and for different ranges h. Though reliable and educational, and sometimes absolutely necessary, this approach can be time consuming.

Though there is no substitute for the experience gained in solving real, nonlinear problems, one can prepare oneself, and become a better predictor, by first analyzing *simple linear* differential equations. Runge-Kutta

Convergence and Stability

techniques provide straightforward short-time analyses of the linear cases. At long times nonlinearity dominates. Long-time convergence and stability require a discussion of the Lyapunov spectrum

Spatial discretization plays a rôle too. Molecular dynamics provides useful guidance to understanding the influence of irregular meshes. Regular mesh instability can be analyzed by lattice methods. The size dependence found in our example problem, homogeneous shear flow, provides the kind of empirical evidence required to develop good intuition.

6.2 Existence and Uniqueness in Continuum Mechanics

With finite computational resources exact prediction is impossible. Accordingly, the concept of a "true" time evolution is at best a limiting case. The convergence and stability difficulties are already apparent in short-time fewbody simulations of particle mechanics. The governing ordinary differential equations of molecular dynamics are certainly a straightforward recipe for predicting the future from given initial conditions. At short times the precision of these predictions is limited only by the computer wordlength. It is obvious that a short-time solution exists and is unique. This view is a natural outgrowth of our experience with the unique evolving reality of the world around us. It would then seem "reasonable" to imagine the existence of a unique, idealized infinitely-precise solution of the flow equations, extending arbitrarily far into the future.

Lyapunov instability calls the reasonableness of this long-time image into question. The *exponential* growth of perturbations with time is the straightforward result of a linearized analysis. Exponential spreading of trajectories, on the time scale of an atomistic collision, correctly suggests that only a *statistical* idea of a solution can be obtained in molecular dynamics. This letdown is not ordinarily a cause of concern. A "reasonable" initial condition is expected to lead to a "reasonable" (that is, useful) result. But existence and uniqueness of solutions at long times are casualties of Lyapunov instability, even for the ordinary differential equations of molecular dynamics.

The existence-uniqueness situation is more serious still in continuum mechanics. The number of initial conditions contributing to a discretized continuum solution becomes unbounded as the precision increases. There are well-established situations in which continuum mechanics can provide qualitatively *different* solutions from qualitatively *similar* initial conditions.

Existence and Uniqueness in Continuum Mechanics

```
Figure 6.1: Two stationary solutions of the Rayleigh-Bénard problem.
The boundary and constitutive equations are identical for both solutions.
```

These two-roll and four-roll Euler-mesh solutions are fully converged.

One simple example of this lack of uniqueness is the Rayleigh-Bénard problem.¹ For identical constitutive and boundary conditions a particular stationary solution is not necessarily unique. Figure 6.1 shows two quite different solutions. Either one of them—two rolls or four—could result as the long time limit evolves from a nearly-quiescent initial condition. This observation has microscopic consequences too. Because it is an article of faith that molecular dynamics, with sufficiently many particles and not too far from equilibrium, will produce results in accord with continuum mechanics, this example demonstrates that there can be no real uniqueness

¹Castillo, Hoover, and Hoover (1997).

ws-book9x6

Convergence and Stability

in molecular dynamics either. This example points out also that there is a *physical* lack of uniqueness in solving continuum problems, over and above the lack of *algorithmic* uniqueness introduced by the choice of numerical method.

6.3 Accuracy and Precision in Numerical Solutions

The *accuracy* of a numerical method is assessed by comparing its predictions to exact solutions. This presupposes that the problem being solved is wellposed, with an exact solution existing as a limiting case, at least for times which are not too long (remember Lyapunov instability as well as the possible lack of uniqueness!).² The usual computer algorithms replace the differential *flow* equations by a deterministic *map*. Such a map advances an approximate state description, with a finite number of significant figures, over a finite time interval. For such a finite-difference mapping algorithm the limiting "exact" solution would involve making both the time and the space resolutions of the map infinitely "precise". The precision of any calculation simply denotes the inherent numerical error due to rounding and computing with finite computer word lengths. In a typical situation 64 binary bits are used to describe numbers, with 10 bits used to describe the (decimal) exponent and sign. The remaining 54 binary bits correspond to about 16 decimal digits so that the *precision* of a typical 64-bit arithmetic operation is one part in 10^{16} .

6.4 Convergence of Numerical Methods

Convergence is the sine qua non of a numerical method. If the approximate numerical solution fails to improve (that is, fails to get closer to the true solution) as more resources are devoted to it, then the algorithm does not converge and must be abandoned, as an imposter. Typically the numerical algorithms of interest to physicists and engineers (of which the fourth-order Runge-Kutta integration algorithm is the foremost representative) "converge". Making the timestep dt smaller [at the expense of additional integration steps $\propto (t/dt)$ and in response to the perceived need for greater precision] leads one closer to the answer at time t. Once one is "close enough" the approximate solution is adopted, for exact answers are never

²Hoover, Time Reversibility, Computer Simulation, and Chaos (2001).

Runge-Kutta Integration of Linear Problems

possible for the real applications of computational physics.

Numerical approximations to solutions of the partial differential equations of continuum mechanics incorporate errors due to the discrete values of time and space. Let us first consider errors due to the discretization of time. For simplicity we generally restrict ourselves to Runge-Kutta integration schemes. These schemes correspond to truncated Taylor's series. When these algorithms are applied to the four linear problems:

$$\dot{x} = \pm x$$
; $\dot{x} = \pm ix$,

it is relatively easy to assess accuracy, precision, stability, and convergence. The four linear problems just described have the corresponding four analytic solutions :

$$x = e^{\pm t} ; x = e^{\pm it}$$

These four are simple prototypes, or caricatures, of stable $[e^{-t}]$, unstable $[e^{+t}]$, and oscillatory $[e^{\pm it} \equiv \cos(t) \pm i \sin(t)]$ systems. We analyze the errors found in solving these systems numerically in the next Section.

6.5 Runge-Kutta Integration of Linear Problems

We considered two examples of Runge-Kutta integration in Section 1.3, one second-order and the other fourth-order. In order to understand the errors incurred by such integrators, it is essential to check, and easy to verify, that *n*th order Runge-Kutta integrators reproduce the first *n* terms in a Taylor's series in the time. All the Runge-Kutta integrators are developed so as to satisfy this criterion. Beyond the simple "first-order" (with a single-step error of order dt^2) "Euler" scheme ,

$$x(t+dt) = x(t) + dt\dot{x}(t) ,$$

a variety of higher-order schemes can be developed. Let us solve the simplest differential equation $\dot{x} = \pm x$ for a single timestep dt using two plausible second-order methods, the "midpoint" and "endpoints" schemes. The midpoint scheme estimates the change in x by first approximating the derivative at time dt/2:

$$\tilde{x}\left(t+\frac{dt}{2}\right) = x(t) + \frac{dt}{2}\dot{x}(t) = x(t)\left[1\pm\frac{dt}{2}\right]$$
.

Convergence and Stability

Then the final value is computed using the time derivative at $\tilde{x}(t + \frac{dt}{2})$:

$$x(t+dt) = x(t) + dt\dot{x}(\tilde{x}) = x(t) \pm dt \left[x(t) \pm \frac{dt}{2}x(t)\right] = x(t) \left[1 \pm dt + \frac{dt^2}{2}\right]$$

As expected, this second-order Runge-Kutta method reproduces both the first-order and second-order terms in the expansion of the exact single-timestep solution of $\dot{x} = \pm x$:

$$x(t+dt) = x(t)e^{\pm dt} .$$

The endpoints scheme estimates the change in x by *averaging* the initial derivative with an estimate of the final one :

$$\tilde{x}(t+dt) = x(t) + dt\dot{x}(t) ;$$

$$x(t+dt) = x(t) \pm \frac{dt}{2} \left[x(t) + \tilde{x}(t+dt) \right] = x(t) \left[1 \pm dt + \frac{dt^2}{2} \right]$$

The result is exactly the same—both these second-order methods reproduce the second-order terms in Taylor's series for the solution at time t + dt, $x(t)e^{\pm dt}$. For *nonlinear* problems their predictions differ.

Any of the *fourth*-order Runge-Kutta integrators (and there are several), applied to the same problem, provides the analogous fourth-order result :

$$x(t+dt) = x(t) \left[1 \pm dt + \frac{dt^2}{2!} \pm \frac{dt^3}{3!} + \frac{dt^4}{4!} \right] \simeq xe^{\pm dt \mp \frac{dt^5}{5!}}$$

with an error of order $\mp dt^5/5!$.

Evidently the growth or decay rate can be made as accurate as is desired by decreasing dt or by increasing the order of the method. For any fixed dtone can estimate the time at which the summed-up error terms, $\mp \frac{dt^5}{5!}$ for the fourth-order Runge-Kutta methods, have become unacceptably large.

The pair of oscillatory differential equations $\dot{x} = \pm ix$ have the pair of fourth-order (in dt) solutions :

$$x(t+dt) = x(t) \left[1 \pm idt - \frac{dt^2}{2} \mp i\frac{dt^3}{6} + \frac{dt^4}{24} \right] .$$

The second-order solutions reproduce the first three of these five terms. The equivalent Taylor's series for the logarithms of these polynomials provide the first two nonvanishing terms as follows :

$$x(t+dt) = x(t)e^{\pm idt \pm i\frac{dt^3}{3!} + \dots},$$

Runge-Kutta Integration of Linear Problems

in the second-order case, and

$$x(t+dt) = x(t)e^{\pm idt \mp i\frac{dt^3}{5!} + \dots}$$

in the fourth-order case. Evidently $n \equiv \frac{\Delta t}{dt}$ iterations of the algorithms give the corresponding approximations for the second-order and fourth-order solutions :

$$x_2(t + \Delta t) = x(t)e^{\pm i\Delta t [1 + \frac{dt^2}{3!}]};$$

$$x_4(t + \Delta t) = x(t)e^{\pm i\Delta t [1 - \frac{dt^4}{5!}]}.$$

For the oscillatory solutions the second- and fourth-order Runge-Kutta methods produce "phase shifts", $\pm \Delta t dt^2/6$ and $\mp \Delta t dt^4/120$, which can be either positive or negative. These phase-shift errors are more benign than the leading-order growth-rate errors which are consequences of the odd-order Runge-Kutta methods. No doubt this feature is responsible for the relative popularity of the even-order methods.

What lessons do these linear analyses suggest for *nonlinear* problems? We can expect to find phase shifts as well as shifts in the Lyapunov exponents (growth rates) as the order of our integration algorithm changes. These shifts generally vary as power laws of the timestep dt. Given three short-time simulations such a power law can be determined and the error can be estimated and removed, if desired. Linear problems can also suggest which of several algorithms provides least cost for a given accuracy.

Consider the harmonic oscillator. Integrating $\ddot{x} = -x$ over one oscillation period ($0 < t < 2\pi$) using second-order and fourth-order Runge-Kutta integrators should lead to endpoint errors of order one percent when

$$\frac{dt_2^3}{6}\frac{2\pi}{dt_2} = 0.01 \longrightarrow dt_2 = 0.098 \; ;$$

in the second-order case, and

$$\frac{dt_4^5}{120}\frac{2\pi}{dt_4} = 0.01 \longrightarrow dt_4 = 0.66 \ ,$$

in the fourth-order case. Accordingly, even though the fourth-order method requires four force evaluations per step, twice that of the second-order methods, the overall calculation should be roughly three times cheaper for "real" problems, where, as is usually the case, the dominant cost is the evaluation of the righthand side of the differential equations being solved. Though this

183

ws-book9x6

ws-book9x6

Convergence and Stability

simple linear error analysis would suggest using even higher-than-fourthorder methods the additional work is not generally compensated by better performance. A primary reason for this is that the equations of motion seldom have even four continuous time derivatives.

6.6 Stability

The *stability* of a numerical method is important too. Here we leave thermodynamic phase equilibria aside. We instead imagine a simpler idealized mathematical "stability". Even so, "Stability" is a difficult term. This is because Lyapunov *instability* is such a common feature of real systems. Our mathematical models for such systems show that small perturbations can grow large exponentially fast, and on a microscopic timescale governed by interparticle collisions. Both nature and the mathematics used to describe it are unstable in this sense. Only a statistical reasonable prediction of the future can be made. In problems such as evolving phase equilibria, where the nucleation of phases and the location of phase boundaries are the result of unseen fluctuations, it is evident that the details of the simulation results depend on the solution algorithm.

In any event, even in the case of an *unstable* system of equations, like { $\dot{x} = \lambda x$ }, one can determine the accuracy with which the characteristic growth *rates* { λ }, if the time-dependence varies as $e^{\lambda t}$, are reproduced by a numerical algorithm. The growth rates can be expressed in terms of the local Lyapunov spectrum. At any instant of time the number of positive exponents corresponds to the dimensionality of state space in which perturbations grow—the number of negative exponents corresponds to the dimensionality in which they shrink. Even in extremely simple systems—the Lorenz Attractor³ is a good three-dimensional example—the numbers of positive and negative exponents typically vary with time.

In a rough sense, *stability* indicates lack of sensitivity to perturbations. *Instability* indicates sensitivity. These concepts can be made precise by introducing the local (instantaneous) Lyapunov spectrum,^{4,2} which describes the time development of satellite trajectories in the neighborhood of a reference trajectory. "Satellite trajectories", which remain closeby are evidently more stable than those which diverge away.

 $^{^{3}\}mathrm{Hoover},$ Tull (now Hoover!), and Posch (1988).

⁴Hoover, Computational Statistical Mechanics (1991), Section 11.7, page 301, and

² Time Reversibility, Computer Simulation, and Chaos (2001), Section 7.4, page 169.

Stability

In a more systematic detailed approach there can be many meanings of the word "stability". From a computational perspective we would like to indicate how closely a simulation of a system, described by a set of ordinary differential equations, describes the behavior of an exact solution of the same equations. For equations which have a unique long-time solution, such as the exponential decay equation :

$$\dot{x} = -x \longrightarrow x(t \to \infty) \equiv 0$$

we could agree to call "stable" those solution algorithms which do approach x = 0 in the long-time limit. For equations which have a periodic solution,

 $\dot{x} = ix \to \ddot{x} = -x \longrightarrow x(t) = x(0)\cos(t) + v(0)\sin(t) ,$

we could call "stable" an approximate solution which remains within a specified precision ϵ of the exact solution provided that the initial conditions are given within a tolerance δ .

But the usual situation is that the exact form of the long-time solution is either uninteresting or unknown. It might be predictable in principle or it might be completely unpredictable, as is the "solution" for any problem exhibiting Lyapunov instability. We must be prepared to deal with the completely unpredictable (at long times) case. We can sidestep this difficulty by restricting our attention to predictability at a fixed time.

In molecular dynamics it is generally expected that interesting flows are chaotic, exhibiting continuing fluctuations even though the boundary conditions are fixed. This is a less-common situation in continuum mechanics because continuum mechanics ordinarily includes explicit damping. Stationary flows with stationary boundary conditions are the usual continuum textbook fare. Transient flows, as well as nonstationary flows with stationary boundary conditions require a different and more detailed stability criterion. A chaotic Rayleigh-Bénard flow, for instance, could be characterized by time-averaged values of the field quantities and their fluxes. A more sensitive description could involve the rates at which two nearby solutions depart from one another or the rates at which particular correlations decay. There is an interesting pair of studies of such a problem from both the microscopic⁵ and the macroscopic⁶ viewpoints.

In numerical simulations of the partial differential equations of continuum mechanics the stability of the results can depend on the chosen mesh

⁵Watanabe and Kaburaki (1996).

⁶Castillo and Hoover (1998).

ws-book9x6

Convergence and Stability

as well as the details of the difference and time integration schemes. Intrinsic instability can be described by the Lyapunov spectrum, which gives the rates at which perturbations grow or decay. This seems the best approach to a description of numerical methods. The local Lyapunov spectrum (local in time) can be applied to both transient and stationary problems. We outline the method in the next two Sections.

6.7 Lyapunov Instability

Alexander Lyapunov formalized the analysis of nonlinear problems by focusing on their linearization in the neighborhood of a reference trajectory. If we consider the evolution of a small perturbation in the neighborhood of a known solution of the flow equations $\dot{x} = f(x)$ we can write a motion equation describing the time development of the perturbation :

$$\delta = D \cdot \delta \; ; \; \delta \equiv x - x_{\rm ref} \; ,$$

where D (for Dynamical matrix) is the linearization of the flow equations $f(x_{ref})$ at the reference coordinate x_{ref} :

$$D = \partial f / \partial x$$
.

To illustrate the meaning of this notation consider an example which can exhibit Lyapunov instability, a thermostated harmonic oscillator.⁷ For this "Nosé-Hoover oscillator", with coordinate q, momentum p, and friction coefficient ζ , the complete equations of motion (with unit mass, temperature, force constant, and thermostat relaxation time) are three ordinary differential equations :

$$\dot{q} = p \; ; \; \dot{p} = -q - \zeta p \; ; \; \dot{\zeta} = p^2 - 1 \; .$$

In this case the flow velocity f is the vector $(\dot{q}, \dot{p}, \dot{\zeta})$ so that the corresponding dynamical matrix D is the 3×3 matrix :

$$D = \begin{pmatrix} \partial \dot{q}/\partial q & \partial \dot{q}/\partial p & \partial \dot{q}/\partial \zeta \\ \partial \dot{p}/\partial q & \partial \dot{p}/\partial p & \partial \dot{p}/\partial \zeta \\ \partial \dot{\zeta}/\partial q & \partial \dot{\zeta}/\partial p & \partial \dot{\zeta}/\partial \zeta \end{pmatrix} = \begin{pmatrix} 0 + 1 & 0 \\ -1 - \zeta - p \\ 0 & 2p & 0 \end{pmatrix}$$

The standard "eigenvalue-eigenvector" analysis of such a problem begins by assuming the existence of a solution with eigenvalue λ , corresponding

⁷Posch, Hoover, and Vesely (1986).

Lyapunov Instability

to the eigenvector δ :

$$\delta = e^{\lambda t} \to D \cdot \delta \equiv \dot{\delta} = \lambda \delta \; .$$

Incorporating the righthandside $\lambda \delta$ into the matrix D gives the standard form of the eigenvector-eigenvalue problem :

$$(D - \lambda \mathbf{I}) \cdot \delta \equiv 0 \; .$$

To solve it λ has to be chosen so that the determinant of the matrix $(D-\lambda I)$ vanishes. I is the "identity" matrix, with diagonal elements equal to unity and vanishing offdiagonal elements.

An ordinary oscillator, without thermostat, has the 2×2 matrix ,

$$D = \begin{pmatrix} 0 & +1 \\ -1 & 0 \end{pmatrix} \to D - I\lambda = \begin{pmatrix} -\lambda & +1 \\ -1 & -\lambda \end{pmatrix} \to \lambda^2 + 1 = 0 .$$

In this case the two resulting eigenvalues, { λ } = { $\pm i$ }, indicate that a small perturbation (the eigenvector) ,

$$\delta = (\delta_q, \delta_p) \; ,$$

would *oscillate* in time sinusoidally, without any growth at all. In the thermostated case 3×3 case, the eigenvalues include 0, with an eigenvector

$$(\delta_q, \delta_p, \delta_{\zeta}) = \frac{(p, 0, -1)}{\sqrt{p^2 + 1}}$$

and the pair of eigenvalues

$$\lambda = -\frac{\zeta}{2} \pm \frac{\sqrt{\zeta^2 - 4 - 8p^2}}{2}$$

The latter eigenvalues are positive, corresponding to exponential growth, $\delta \propto e^{\lambda t}$, whenever ζ is negative, and indicate decay, possibly oscillatory, whenever the friction coefficient is positive. A detailed investigation of this thermostated oscillator shows that the situation is already quite complicated: for some initial (q, p, ζ) conditions the time development is regular, without Lyapunov instability, while for others, there is a time-averaged positive value for the largest Lyapunov exponent, indicating chaotic behavior and sensitive dependence on the initial conditions.⁷ A thermostat variable, ζ in this case, is not at all necessary for chaos. In the following section we consider a second numerical Lyapunov-unstable example borrowed from ordinary Hamiltonian mechanics.

187

ws-book9x6

⁷Posch, Hoover, and Vesely (1986).

Convergence and Stability

6.8 Stability Analysis for a Chaotic Problem

Consider now the conservative (constant energy) motion of a single twodimensional particle oscillating in the field of its four fixed neighbors, as was illustrated in Section 1.3 . This motion is typically "chaotic", in the sense that the (time-averaged) separation between two (nearby) trajectories grows exponentially in time. Because the forces vanish at r = 1 in this case, with F''' discontinuous there, the equations of motion are officially singular. The largest Lyapunov exponent for such a motion, follows from a time-dependent extension of the eigenvalue-eigenvector approach of the last Section. In this extension the "offset vector" $\delta \equiv x - x_{\rm ref}$ is no longer allowed to stretch or shrink. It is instead constrained to constant length :

$$\dot{\delta} = D(t) \cdot \delta - \lambda_1(t) \delta$$
 and $\dot{\delta} \cdot \delta \equiv 0$.

Here the instantaneous Lagrange multiplier $\lambda_1(t)$, whose long-time average is the largest Lyapunov exponent λ_1 , has the local value required to ensure that the offset-vector separation between two neighboring trajectories remains constant (as well as infinitesimal). For a many-dimensional separation vector δ , the constant-length condition can be expressed in terms of the dynamical matrix D(t):

$$\delta \cdot \dot{\delta} = \delta \cdot D \cdot \delta - \lambda_1(t) \delta \cdot \delta \equiv 0 \longrightarrow$$

$$\lambda_1(t) = \frac{\delta \cdot D \cdot \delta}{\delta \cdot \delta} ; \ \lambda_1 = \langle \ \lambda_1(t) \ \rangle .$$

Figure 6.2 shows two separate trajectories. A Lagrange multiplier, $\lambda_1(t)$, maintains the trajectory separation at a value 10,000 times the initial value chosen in Figure 1.1. Figure 6.3 shows the multiplier as a function of time (on the left). The time-averaged separation rate, $\lambda_1 \equiv \langle \lambda_1(t) \rangle$ is shown too, and appears to be nearing 2.5, in agreement with the visual estimate from Figure 1.2.

Stability Analysis for a Chaotic Problem



Figure 6.2: Two single-particle trajectories for the example considered in Section 1.3 . y(t) is shown as a function of x(t). The interactions with the fixed neighbors are $\phi(r < 1) = (1 - r^2)^4$. The phase-space separation between the trajectories, $\Delta^2 = \Delta p^2 + \Delta r^2$, is held fixed.





ws-book9x6

Convergence and Stability

The instantaneous value corresponding to the largest time-averaged Lyapunov exponent is not simply an instantaneous eigenvalue of the matrix D because the *direction* associated with the (time-averaged) maximum growth rate is itself a function of time. As a consequence the vector δ_1 associated with the exponent λ_1 reflects the *past* history of the trajectory and its growth rate has contributions from *all* of the eigenvalues. After a transient time interval of the order of $1/\langle \lambda_1 \rangle$ the direction of the offset vector becomes independent of its initial choice and provides a unique well-defined $\lambda_1(t)$. Even the instantaneous eigenvalues of D, which do not incorporate this dependence on the past, are typically complex, as in the Nosé-Hoover oscillator problem of Section 6.7, indicating rotation, as well as growth and decay.

The entire Lyapunov spectrum can be determined in this way, and characterizes stability and instability.⁸ A study of the Lyapunov spectrum for different macroscopic algorithms would be educational, not only for determining sensitivity to local parameter changes, but also for discovering which features of the spectrum are physical rather than algorithmic. This area is largely unexplored. Let us now turn from the time dependence of numerical solutions to the effects of spatial variations, beginning with the ordinary differential equations of molecular dynamics.

6.9 Size Dependence: Lessons from Molecular Dynamics

The time integration errors and instabilities just studied would appear to be the whole story for molecular dynamics. But because the particles make up a spatial grid, molecular dynamics can also provide useful ideas for analyzing convergence and stability with smooth particles. The largest reasonable timestep in molecular dynamics insures that a typical vibration time (or collision time) is still spanned by several timesteps. The analog, in continuum mechanics, is the Courant condition, which requires that the timestep be somewhat smaller than the sound traversal time between the closest pair of nodal points.

Thoroughly uniform systems in molecular dynamics are only available through the use of periodic boundary conditions. In such cases (which can represent either fluids or solids) typical errors are of order 1/N or $\ln N/N$. These errors are deviations of time averages from the large-system limiting values with $N \to \infty$. When real surfaces are present (as is always the

⁸Posch and Hoover (1989).

Smooth-Particle Spatial Integration Errors

case in the application of continuum mechanics to structural analysis) then surface effects of order $(1/N)^{1/D}$ ($\sqrt{1/N}$ in two dimensions or $\sqrt[3]{1/N}$ in three) are usual. Thus the main expectation from molecular dynamics would be to find SPAM errors (deviations from fully-converged continuum solutions) of order h/L, where L is the system width. The detailed example problem at the end of this chapter supports this expectation. For situations requiring modest accuracy, errors of order h/L are perfectly acceptable. For greater accuracy, extrapolation is a feasible approach.

6.10 Smooth-Particle Spatial Integration Errors

The Runge-Kutta local time-integration errors of order dt^3 or dt^5 have spatial analogs in smooth-particle simulations. The intrinsic width h of the weight function guarantees at least second-order (in h) spatial integration errors. To estimate these errors suppose that particle values { f_j } follow a Taylor's series description :

$$f_j(x_j) = f(0) + x_j f'(0) + \frac{1}{2} x_j^2 f''(0) + \dots$$

The smooth-particle average of f at x = 0,

$$\langle f(x=0) \rangle \equiv \sum_j w_{0j} f_j / \sum w_{0j} ,$$

has relatively benign first-order contributions, $f'(0)\langle x_j \rangle$. These vanish for a symmetric particle distribution. The second-order contributions, $\frac{1}{2}f''(0)\langle x_j^2 \rangle$, are *nonvanishing*, even for a regular grid. In the limit of many particles, spaced uniformly, the second-order terms can be replaced by an integral :

$$\langle f \rangle_0 - f(0) \longrightarrow \frac{1}{2} f''(0) \int_{-h}^{+h} x^2 w(x) dx .$$

In one dimension, and for Lucy's weight function, the integral is $2h^2/21$ while for Monaghan's it is $h^2/9$. Similar results are obtained in two or three dimensions. When the smooth particles fill space uniformly we can expect to find errors in the field variables { f(r) } of the order of f'' multiplied by the square of the interparticle spacing.

Similar results hold for the first and second spatial derivatives of field

Convergence and Stability

variables :

$$\langle f' \rangle_0 - f'(0) \propto h^2 f'''(0) ; \langle f'' \rangle_0 - f''(0) \propto h^2 f''''(0) .$$

Alternative error estimates for smooth particles could be based on Monte Carlo integration. Such estimates would certainly be appropriate were the particles placed randomly. But because smooth particles tend to distribute themselves homogeneously (particularly if repulsive pair core potentials are used) such Monte Carlo estimates are overly pessimistic.

6.11 Lattice Instability

Typical numerical simulations generate point properties. SPAM seeks to *smooth* these properties by averaging, so as better to describe a continuum. Ideally the arrangement of the points should not matter much. If it does, then the initial grid needs to be chosen with care. In a method like SPAM, in which the points move, there is the potential difficulty that a "good" arrangement of points could become worse with time, perhaps by clumping, or by developing voids. Here we touch on the relative stability of regular lattices of points.

In a one-dimensional problem, other things being equal, an equally-spaced mesh of points seems best and the stability of the lattice to small displacements can be studied as we did in Section 5.6 . In that Section we found that regular lattice arrangements were often unstable at zero pressure unless a special potential chosen to stabilize lattices was used. It would appear that positive pressure would stabilize lattices, with the dynamics governed by an effective repulsive potential $\propto w(r)$. Let us explore lattice stability in the purely repulsive case, $P = \rho^2/2$, first in one dimension, and then in two.

The direct approach is to start with a regular motionless lattice, add small random displacements or velocities (with $\sum \delta = 0$ or $\sum \dot{\delta} = 0$), and examine the dynamical evolution. The results found, for a wide range of h values and for both Lucy's and Monaghan's weight functions, are fully consistent with corresponding lattice-dynamical analyses. Figure 6.4 shows the time evolution of the kinetic energy for a fixed total energy, in response to random displacements in the range $-0.05 < u_x < +0.05$ for h = 2 (above) and h = 5 (below). There is no apparent tendency toward instability.

Lattice Instability



Figure 6.4: Time evolution of the kinetic energy for 24 one-dimensional Lucy particles in a periodic box of length 24 with h = 2 and h = 5 (dashed line, near the bottom). This Lucy dynamics corresponds to solving the SPAM equations of motion with the equation of state $P = \frac{1}{2}\rho^2$. The slower oscillations in the latter case reflect the weaker nature of the restoring forces, which lead to oscillations with $\omega \propto \sqrt{w''} \propto h^{-3/2}$.

In two dimensions the same straightforward dynamical analysis shows that lattice instability is possible for both the square and triangular lattices. The accompanying Figures 6.5 and 6.6 show the evolutions of the squarelattice and triangular-lattice kinetic energies for a time of 100, using the same displacement amplitude, $[-0.05 < u_x, u_y < +0.05]$, as in one dimension. Throughout the evolutions shown in the Figures the total energy is conserved with ten-digit accuracy. The final configurations, also shown in the Figures, have lost the symmetry of the initial regular lattices. This augurs well for the use of smooth particles to model fluids, for which stable lattice structures would imply an unphysical resistance to shear. As we saw in Section 5.6 it is possible to stabilize particular lattices by choosing just

Convergence and Stability

the right weight functions. For solids constitutive relations including $\nabla \rho$ and $\nabla \nabla \rho$ can be used to stabilize regular structures.⁹ See Section 5.7.



Figure 6.5: Time evolution of the total kinetic energy for $12 \times 12 = 144$ Lucy particles in a square periodic box of length 12 with h = 4. The corresponding individual particle trajectories are shown at the right.



Figure 6.6: Time evolution of the total kinetic energy (with corresponding particle trajectories) for $12\times 12=144$ Lucy particles in a periodic triangular-lattice box of length $12\sqrt[4]{4/3}$ and height $12\sqrt[4]{3/4}$ with h=2.

 $^{^{9}}$ Hoover and Hoover (2006).

Even-Odd Instability

6.12 Even-Odd Instability

We encountered two different problems in the Chapter 5 examples exhibiting even-odd instability. A flux-based heat transfer equation, gave the set of coupled first-order ordinary differential equations :

$$\dot{T}_i \propto T_{i+2} - 2T_i + T_{i-2} \; ,$$

while a linear-chain motion equation, led to a related set of second-order equations :

$$\ddot{x}_i \propto x_{i+2} - 2x_i + x_{i-2} \; .$$

Both these examples lacked coupling between adjacent even- and oddnumbered nodes. Such a lack of coupling can be prevented by (i) using shorter-ranged (and more nearly accurate) approximations :

$$T_i \propto T_{i+1} - 2T_i + T_{i-1}$$
; $\ddot{x}_i \propto x_{i+1} - 2x_i + x_{i-1}$

or (ii) by using incorporating artificial terms in the differential equations specially designed to reduce local fluctuations by spreading their influence in space. Artificial viscosity (Section 3.13) and Monaghan's velocity-averaging motion equations (Section 8.4) are two different ways to accomplish this.

By using Runge-Kutta integration for time we avoid the possibility of a "weak" instability involving uncoupled even and odd timesteps. But the space integration can still provide such instability. Most of the equations of mathematical physics can be written as simple differential equations which are "second order" in time, or space, or both, with the diffusion equation and wave equation the most familiar examples :

$$\frac{\partial \rho}{\partial t} = D \nabla^2 \rho \; ; \; \frac{\partial^2 u}{\partial t^2} = c^2 \nabla^2 u \; .$$

Here D is the diffusion coefficient and c is the sound speed. Iterating the finite-difference approximation for the first derivative suggests the simple second-difference scheme for the spatial derivatives :

$$\ddot{u}(x) = \frac{\partial^2 u}{\partial x^2} \simeq \frac{u(x+dx) - 2u(x) + u(x-dx)}{dx^2} \longrightarrow$$
$$u(x) \propto e^{-i\omega t + ikx} .$$

August 16, 2006 17:44

196

Convergence and Stability

The same idea, applied to the time derivatives, gives a simple leapfrog scheme :

$$\ddot{u} \simeq \frac{u(t+dt) - 2u(t) + u(t-dt)}{dt^2} \equiv \frac{u(x+dx) - 2u(x) + u(x-dx)}{dx^2}$$

an alternative to the more-nearly-accurate Runge-Kutta integration.

It is evident that these second-order derivatives can deal with the evenodd instability just considered. For a function which alternates from particle to particle (equally spaced, and in one dimension, for simplicity), the first derivative, second derivative, ... also alternate in just the same way so that only a function including an estimate of the curvature, $\frac{\partial^2 u}{\partial t^2}$ or $\frac{\partial^2 u}{\partial x^2}$, is sufficient to damp such a mode.

Instability in computer algorithms can be physical (or "real"), as in the Lyapunov instability of the thermostated oscillator model, or an artifact stemming from unfortunate choices of timestep, mesh, or algorithm. The Rayleigh-Bénard problem can provide examples of all these features. The relatively benign even-odd instability, a consequence of insufficient coupling between subsets of nodes, can, over long periods of time, cause the divergence of otherwise well-behaved computer algorithms.

Numerical solutions of differential equations are necessarily approximate. Ordinary differential equations are solved by advancing in a series of timesteps, with the size of the timestep governing the accumulating errors. Runge-Kutta algorithms are equivalent to keeping a truncated series representation of the solution. A second-order method generally includes terms of order dt^2 while omitting those of higher order. Accumulation of such errors, t/dt of them for a simulation of duration t, would lead to an overall error proportional to dt were it not for Lyapunov instability. This instability, typical of nonlinear equations, amplifies errors exponentially, $\delta(t) \simeq \delta(0)e^{\lambda t}$. A bifurcation rate of one in thirty timesteps causes a loss of sixteen-digit accuracy in about 1600 timesteps :

$$10^{16} \simeq 2^{1600/30}$$

6.13 Example: Shear-Flow Convergence

The convergence of the smooth-particle method is best demonstrated through example simulations. Static density and heat flux calculations showed good convergence to their continuum limits. Dynamic simulations, such as a constrained periodic shear flow, provide time-varying nonequilibExample: Shear-Flow Convergence

rium examples. Molecular dynamics simulations, using the pair potential

$$\Phi = \sum_{i < j} \phi_{ij} ; \ \phi_{ij}(|r_{ij}| < 1) = 100(1 - r_{ij}^2)^4 ,$$

provide results closely resembling smooth-particle simulations with the Lucy potential. 10 These two functions are compared in Figure 6.7 .



¹⁰Hoover and Posch (1995).



Convergence and Stability

Figure 6.8: The convergence of the steady-state shear viscosity to the limiting large-system value indicates errors of order $\sqrt{1/N}$. The data (for two different strain rates but with the same densities and energies) are for a short-ranged repulsive core potential, $\phi(r < 1) = 100(1 - r^2)^4$ which closely resembles the shape of Lucy's weight function, as indicated in Figure 6.7. Periodic boundaries were used here. See Reference [10].

In carrying out steady-state simulations of this kind it is necessary to use thermostat forces ($\propto -mv$) to extract the dissipated energy provided by the external shear forces. There are several formulations of such forces, such as the Nosé-Hoover forces of Section 6.7, all of which converge to the same large-system limit. These stationary shear simulations are all Lyapunov unstable. Despite this sensitivity of the detailed trajectories to the initial conditions the time-averaged shear stress (which corresponds to an artificial viscosity for the ideal-gas fluid) converges rapidly to a stationary value.

References

6.14 References

- V. M. Castillo, Wm. G. Hoover, and C. G. Hoover, "Coexisting Attractors in Compressible Rayleigh-Bénard Flow", Physical Review E 55, 5546-5550 (1997).
- (2) Wm. G. Hoover, *Time Reversibility, Computer Simulation, and Chaos* (World Scientific, Singapore, 2001).
- (3) W. G. Hoover, C. G. Tull (C. G. *Hoover* as of 10 November 1988!), and H. A. Posch, "Negative Lyapunov Exponents for Dissipative Systems", Physics Letters A 131, 211-215 (1988).
- (4) Wm. G. Hoover, Computational Statistical Mechanics (Elsevier, 1991).
- (5) T. Watanabe and H. Kaburaki, "Increase in Chaotic Motions of Atoms in a Large-Scale Self-Organized Motion", Physical Review E 54, 1504-1509 (1996).
- (6) V. M. Castillo and Wm. G. Hoover, "Entropy Production and Lyapunov Instability at the Onset of Turbulent Convection", Physical Review E 58, 7350-7354 (1998).
- (7) H. A. Posch, W. G. Hoover, and F. J. Vesely, "Canonical Dynamics of the Nosé Oscillator: Stability, Order, and Chaos", Physical Review A 33, 4253-4265 (1986).
- (8) H. A. Posch and W. G. Hoover, "Equilibrium and Nonequilibrium Lyapunov Spectra for Dense Fluids", Physical Review A 39, 2175-2188 (1989).
- (9) Wm. G. Hoover and C. G. Hoover, "Smooth Particle Phase Stability with Generalized Density-Dependent Potentials", Physical Review E 73, 016702 (2006).
- (10) Wm. G. Hoover and H. A. Posch, "Shear Viscosity via Global Control of Spatiotemporal Chaos in Two-Dimensional Isoenergetic Dense Fluids", Physical Review E 51, 273-279 (1995).

August 16, 2006 17:44

WSPC/Book Trim Size for 9 in x 6 in ws-book9x6

200

 $Convergence \ and \ Stability$

Chapter 7

Lucy and Embedded-Atom Fluids

/ Trajectory Isomorphisms / Lucy Fluid / Equilibrium Thermodynamic Properties for the Lucy Fluid / Transport Properties for the Lucy Fluid / Properties of the Embedded-Atom Fluid / Structural Relaxation

/ 9 Figures /

Example Problems :

[Embedded-Atom Column Relaxation, Embedded-Atom Column Collapse, Lucy Fluid Free Expansion, Lucy Fluid Stationary Shockwave]

7.1 Summary

There is often a striking resemblance between smooth-particle trajectories and the trajectories of particles obeying ordinary Newtonian mechanics. This is not a coincidence. In fact there are two significant special cases in which the apparent resemblance is actually a precise identity. One, the Lucy fluid, corresponds to the flow of an ideal γ -law gas, with $P \propto \rho^2$. Sections 5.9 and 6.13 describe the shear flow of this fluid.

The other, an embedded-atom fluid, corresponds to the flow of a dense fluid with stress-free density ρ_0 , where the pressure, $P \propto \rho^3 - \rho_0 \rho^2$, vanishes. The need for an additional surface tension potential for fluids or solids, as well as a potential able to stabilize "solid" phases (the "invariant curvature" potential), was discussed in Section 5.6. Here we consider the constitutive relations for both these special cases from the standpoints of equilibrium statistical mechanics and nonequilibrium kinetic theory. We then illustrate the use of the two models by solving four example problems.

ws-book9x6

Lucy and Embedded-Atom Fluids

7.2 Trajectory Isomorphism for the Lucy Fluid

.

Provided that the pressure varies as ρ^2 , the *macroscopic* smooth-particle equations of motion ,

$$\left\{ \dot{r}_{i} = v_{i} ; \dot{v}_{i} = -m \sum_{j} \left[(P/\rho^{2})_{i} + (P/\rho^{2})_{j} \right] \cdot \nabla_{i} w_{ij} \right\} ,$$

have exactly the same form as do the microscopic pair-potential equations of motion for molecular dynamics :

$$\left\{ \begin{array}{l} \dot{r}_i = v_i \ ; \ \dot{v}_i = -\frac{1}{m} \sum_j \nabla_i \phi_{ij} \end{array} \right\}$$

By choosing the right proportionality constant ($w_{ij} \propto \phi_{ij}$) and the same initial conditions (for { r, v }) it is evident that the smooth-particle and molecular dynamical trajectories are identical ("isomorphic"—having the same form).

It is likewise evident that the pair potential in molecular dynamics corresponds to the weight function for the Lucy fluid. Let us consider the simplest case :

$$\left[\left(\frac{P}{\rho^2}\right)_i + \left(\frac{P}{\rho^2}\right)_j\right] = \frac{1}{m^2} \ ,$$

where all the particles have the same mass m. In this case, with the molecular dynamics and SPAM equations of motion identical, there is also a *near* correspondence between the pairwise-additive molecular potential energy Φ and the summed-up SPAM weight functions :

$$\Phi = \sum_{i < j} \phi_{ij} = \sum_{i < j} w_{ij} = \frac{1}{2} \sum_{i} [(\rho_i/m) - w(0)] .$$

To make the correspondence between the density sum and twice the potential energy perfect, the single-particle contribution to density, mw(0), would have to correspond to a state-independent self energy $\phi_i = \frac{1}{2}w(0)$. Any such additive constant would have no influence on the molecular dynamical motion or on the corresponding constitutive properties (other than redefining the zero of energy).

The macroscopic dynamics of smooth particles, where it is usual that each particle interacts with roughly 20 others, matches *exactly* the micro-

Statistical Thermodynamics for the Lucy Potential

scopic molecular dynamics of a dense fluid governed by the pair potential w. As usual, we will use Lucy's form for w to illustrate :

$$w_{\text{Lucy}}(r < 3) = \frac{5}{9\pi}(1+r)[1-(r/3)^3]$$
.

We use the same special case h = 3 for the two-dimensional free-expansion and shockwave example problems treated at the end of this Chapter.

Molecular dynamics is usually applied to situations in which fluids are only slightly compressed from "normal" density. If the normal density corresponds to a nearest-neighbor spacing of h = 3 then our usual choice $Nm/V = \rho = 1$ corresponds to nine-fold compression, a very dense, but also very soft, fluid. The properties of such a fluid have some intrinsic pedagogical interest, both from the standpoint of equilibrium statistical thermodynamics and from the standpoint of nonequilibrium kinetic theory. Before carrying out our fluid-flow illustrations of the isomorphic trajectories, we consider briefly this statistical background for the constitutive relations.

7.3 Statistical Thermodynamics for the Lucy Potential

Because the Lucy potential is relatively long range (with h = 3 and at an area per particle of unity), as well as weak, the potential energy and pressure are relatively insensitive to the detailed arrangement of the particles. Let us choose particles of unit mass, m = 1. Consider two cases, (i) a square lattice, (with nearest neighbor spacing 1.0), and (ii) a "triangular lattice", (with nearest neighbor spacing $\sqrt[4]{4/3}$, with six nearest neighbors for each particle). Both give the nearly the "right" density, $m \sum w \simeq 1$. With the *w* sums including the self contribution, w(0), the deviations from the actual density, unity, are only a fraction of a percent :

$$\rho = 1 \simeq \sum_{\text{square}} w = 1.00293003 \simeq \sum_{\text{triangular}} w = 1.00222807$$

On the other hand a completely *random* arrangement of N particles—still with an overall density of unity, and ignoring a correction of order 1/N—gives a significantly larger value for the density *at* a particle :

$$\sum_{\text{random}} w = 1 + w(0) = 1.17683883 \; ; \; w(0) = \frac{5}{9\pi} \; ,$$

r
Lucy and Embedded-Atom Fluids

corresponding, from the molecular dynamical viewpoint, to an energy difference, per particle, of (1.177-1.002)/2. The 18% density error is abundant evidence that a random grid is a poor choice for smooth particles. To reduce the random-grid density error to 1% would require h>12, with more than 500 interacting neighbors per particle. Accordingly, a stable low-temperature regular crystalline arrangement, with the lower potential energy, but with a higher heat capacity, 2Nk, should melt to form a disordered phase (heat capacity Nk) at a temperature of about 0.08.

An alternative prediction of the Lucy solid's melting temperature follows from an estimate of the single-particle vibration frequency (the "Einstein frequency") :

$$\begin{split} \Phi &= \Phi_0 + \frac{x^2}{2} \partial^2 \Phi / \partial x^2 \longrightarrow \\ \omega_{\text{Einstein}} &\simeq \sqrt{\sum \phi'' / m} = \sqrt{\sum w''} ; \\ \langle \sum w'' \rangle &\simeq \int_0^3 2\pi r w'' dr = -\int_0^3 2\pi w' dr = 2\pi w(0) = \frac{10}{9} \end{split}$$

Lindemann's Law¹ estimates that melting occurs when the rms vibrational amplitude is about one-tenth the interparticle spacing $d \simeq \sqrt{V/N}$:

$$\frac{\sqrt{\langle x^2 \rangle}}{d} = \sqrt{\frac{kT}{m\omega^2 d^2}} \longrightarrow kT_m \simeq \frac{1}{10} \left[\frac{10}{9} \sqrt{\frac{4}{3}} \right] \,.$$

This prediction is nicely consistent with the energy difference estimated from the regular and random lattice sums above. Evidently we can expect regular solid-like behavior from the Lucy potential whenever the kinetic energy per particle is of order 0.1 or less, and more random fluid-like behavior at higher kinetic energies.

Properties of the solid phase can be estimated accurately by applying lattice dynamics. See Born and Huang's classic book for the details.² Fluid-phase properties can be estimated from the Mayers' virial series, as is explained in their book.³ Such detailed solid and fluid analyses can be bypassed by applying molecular dynamics. A sample molecular dynamics simulation makes up Section 1.9. This brute-force approach is the simplest

¹Hoover and Ross (1971).

²Born and Huang, Dynamical Theory of Crystal Lattices (1985).

³Mayer and Mayer, *Statistical Mechanics* (1977).

Trajectory Isomorphism for the Embedded-Atom Fluid

(as well as most nearly fool proof) method for determining solid and fluid thermodynamic properties. The simulation results for fluids with $kT \ge 0.50$ shown in my 1996 paper with Siegfried Hess⁴ agree quite well with a slight modification of the ideal-gas model :

$$E = E(0) + NkT$$
; $PV = P(0)V + NkT$;
 $P(0) = -dE(0)/dV = E(0)/V$.

Here E(0) and P(0) are the energy and pressure derived from the *pair* contributions alone. That is, the "self-term" w(0) is to be omitted from the expressions for E(0) and P(0) above.

7.4 Trajectory Isomorphism for the Embedded-Atom Fluid

Another simple equation of state, that for a fluid (or a solid) with a stress-free density ρ_0 , gives rise to another trajectory isomorphism linking smooth-particle continuum mechanics to ordinary molecular dynamics. Consider an equation of state which is the difference of two power laws :

$$P = \frac{B_0}{n-m} \left[\left(\frac{\rho}{\rho_0}\right)^n - \left(\frac{\rho}{\rho_0}\right)^m \right] \;,$$

where B_0 is the zero-pressure bulk modulus,

$$B_0 = \rho_0 \left(\frac{\partial P}{\partial \rho}\right)_{\rho_0}$$

For simplicity we choose the particle mass m and the zero-pressure density ρ_0 both equal to unity. Let us further consider the simplest interesting case, already explored in Section 5.6, with n = 3 and m (not the mass) = 2. The corresponding smooth-particle equations of motion are :

$$\{ \dot{r}_i = v_i ; \dot{v}_i = -m \sum_j \left[\left(\frac{P}{\rho^2} \right)_i + \left(\frac{P}{\rho^2} \right)_j \right] \nabla_i w_{ij} = -\sum_j (\rho_i + \rho_j - 2) \nabla_i w_{ij} \}.$$

Exactly this same equation of motion results from an "embedded-atom" microscopic equation of motion. Embedded atoms have long been used

 $^{^{4}}$ Hoover and Hess (1996).

August 16, 2006 17:44

Lucy and Embedded-Atom Fluids

to model the structure and flow of solid and liquid metals.⁵ From this embedded-atom viewpoint the dynamics of an atomic nucleus depends upon the electronic density, with the density a sum of contributions from nearby species, $\rho = \sum w$. The total potential energy Φ determining the motion depends on the density in such a way as to reproduce the stress-free density ρ_0 . The simplest of the power-law equations of state corresponds to Hooke's Law for the potential energy :

$$\Phi = \sum_{i} \phi_i \; ; \; \phi_i = \frac{mB_0}{2\rho_0} \left[\left(\frac{\rho_i}{\rho_0} \right) - 1 \right]^2 \; .$$

 Φ is a "functional", rather than an ordinary function, because the individual densities { ρ_i } themselves depend upon pairs of particle coordinates :

$$\rho_i \equiv \sum_j m_j w_{ij}$$

Chain-rule differentiation of the total potential gives exactly the same forces as does the smooth-particle approach :

$$F_i \equiv -\nabla_i \Phi = -\nabla_i \phi_i - \sum_j \nabla_i \phi_j$$

$$= -\frac{m^2 B_0}{\rho_0^2} \sum_j \left[\left(\frac{\rho_i}{\rho_0} \right) - 1 \right] \nabla_i w_{ij} - \frac{m^2 B_0}{\rho_0^2} \sum_j \left[\left(\frac{\rho_j}{\rho_0} \right) - 1 \right] \nabla_i w_{ij}$$
$$\longrightarrow$$
$$F_i = -\frac{m^2 B_0}{\rho_0^2} \sum_j \left[\frac{\rho_i}{\rho_0} + \frac{\rho_j}{\rho_0} - 2 \right] \nabla_i w_{ij} .$$

This embedded-atom equation of state is a convenient fluid model so long as thermal effects and surface effects can be ignored. The stress-free density is ρ_0 and the bulk modulus at that density is B_0 . In the example problem at the end of this Chapter we test this model's ability to treat the flow of a simple fluid, like water flowing under the influence of gravity. We will see that generalizations of this approach provide useful models for solids too.

⁵Foiles, Baskes, and Daw (1986).

Embedded-Atom Approach to Structural Relaxation



7.5 Embedded-Atom Approach to Structural Relaxation

Figure 7.1: Structures obtained by relaxation using the embedded-atom potential, $\Phi = \frac{1}{2} \sum (\rho_i - 1)^2 \rightarrow P = \rho^3 - \rho^2$, with the range of Lucy's weight function h = 3. Boundary particles are doubled up (and quadrupled at the four corner positions). The 46 × 46 interior particles were initially arranged in a regular square lattice (left) and at random, between (±23, ±23), (right) and then allowed to relax with a relaxation time $\tau = 2$. A pairwise-additive core potential, $\phi = 10(\sigma^2 - r^2)^4$, with $\sigma = 0.2$, was included. The final embedded-atom potential energies of these relaxed structures were less than 10^{-6} per particle, corresponding to typical per-particle density errors of less than one part per thousand.

In simulating macroscopic motions it is desirable that the smoothparticle densities match the density of the material being modeled. For a regular lattice arrangement, with equal densities for every particle, this is no problem, but in more interesting problems, with boundaries, a relaxation process needs to be formulated and executed to achieve an equal-density initial configuration. The embedded-atom approach can be used to do this. If a guessed configuration is relaxed, following the equations of motion appropriate to the desired density ρ_0 , but with a viscous damping force, $-mv/\tau$ added, a good approximation to constant density can be obtained.

Figure 7.1 shows a 2500-particle configuration obtained by relaxing a modified 50×50 portion of a square lattice. If the rows and columns of the original lattice are indexed by (ix, iy) the modifications,

$$ix = 1 \rightarrow x = x + 1$$
; $iy = 1 \rightarrow y = y + 1$;

207

ws-book9x6

208

ws-book9x6

Lucy and Embedded-Atom Fluids

$$ix = 50 \rightarrow x = x - 1$$
; $iy = 50 \rightarrow y = y - 1$.

provide doubled-up particles along the sides of a 48×48 rectangle, and quadrupled particles at the corners, giving a rough approximation to the contributions of the missing particles "outside" the square. See Section 5.3 for the one-dimensional analog.

Equations of motion of the form $\dot{v} = a - v/\tau$, where a is the usual smooth-particle acceleration and the viscous relaxation time τ is 1 or $\frac{1}{2}$, lead to the configuration shown. The Figure also shows an alternative structure in which the coordinates of the interior 46 × 46 particles were chosen randomly within a 46 × 46 area square. In the following Sections we use this structure to simulate the relaxation and flow of falling water in a gravitational field. We will see that generalizations of this approach can provide useful models for solids too.

7.6 Example: Embedded-Atom Gravitational Relaxation

The gravitational-field example problems of Chapter 3 used simple highlycompressible gas-phase equations of state, with the two γ laws, $P \propto \rho^2$ and $P \propto \rho$ corresponding to adiabatic and isothermal equilibria. Here we consider instead the embedded-atom potential, which corresponds to a more-nearly-incompressible fluid with a stress-free density ρ_0 . The constitutive equations for the energy and pressure are as follows :

$$E = \frac{mB_0}{2\rho_0} [(\rho/\rho_0) - 1]^2;$$

$$P = B_0 \left[(\rho/\rho_0)^3 - (\rho/\rho_0)^2 \right] .$$

Let us choose $B_0 = 1$ and $\rho_0 = 1$ along with a gravitational field g (in the y direction) of strength $1/y_{\text{max}} = 1/H$ so that the field energy will increase the density at the bottom of the atmosphere from ρ_0 . The bulk modulus,

$$B = B_0 \left[3(\rho/\rho_0)^3 - 2(\rho/\rho_0)^2 \right]$$

can be made arbitrarily large by adjusting B_0 relative to $gH\rho_0$, though higher sound speeds require smaller timesteps ($dt \simeq h/c$). Solutions of the mechanical equilibrium equations [see Section 2.2 again for the detailed method of solution] predict the following correlation between the Example: Embedded-Atom Gravitational Relaxation

equilibrated column height H and the maximum density $\rho(y=0)$:

$$\frac{gH\rho_0}{B_0} = \langle 8, \frac{5}{2}, \frac{1}{2} \rangle \longrightarrow \frac{\rho(y=0)}{\rho(y=H)} = \{ 3, 2, \frac{4}{3} \} .$$

If we confine the sides of the column (of width $n_x = L = 2x_{\max}$) with reflecting walls and add damping to obtain a relaxed structure ,

$$m\ddot{r}_i = m(0, -g_y) - m\left(\frac{\dot{r}_i}{\tau}\right) - \sum_j \left[\rho_i + \rho_j - 2\right] \nabla w_{ij} ,$$

the system quickly relaxes to a stationary equilibrium. Figure 7.2 illustrates structures obtained with *periodic* boundaries at $x = \pm x_{\text{max}} = \pm L/2$ in the x direction.

In that figure, density contours, from $\rho = 1.1$ near the top to $\rho = 1.9$ near the bottom, are compared for simulations with L = 10, and L = 20, using $8L^2$ smooth particles. The double-headed arrows indicate calculated contour positions using the analytic solution of the force-balance equation.

A pairwise-additive core potential,

$$\phi(r < \sigma) = 10(\sigma^2 - r^2)^4$$
; $\sigma^2 = 0.2$,

was included in the simulations. The relaxation time τ was equal to 2 in these examples. The equilibrated height should be $\frac{5}{8}$ the field-free height, 8L. $P = \rho^3 - \rho^2$. Any reasonable damping time τ , comparable to the sound traversal time, can be used in the equilibration stage. Another equally-effective approach is to monitor the total kinetic energy K. Whenever K(t) reaches a local maximum set all the particle velocities equal to zero.

Evidently static smooth-particle meshes faithful to an inhomogeneous density are relatively easily obtained. In the following Section we consider a challenging time-dependent problem, which unlike the static problem just considered represents a significant challenge to numerical techniques using finite elements or finite differences.



Lucy and Embedded-Atom Fluids

Figure 7.2: Smooth-particle simulations of the embedded-atom atmosphere resulting with $g_y = \frac{1}{2L}$ and equilibrated height H = 5L, using the Lucy weight function with h = 3 and *periodic* x boundaries. $P = \rho^3 - \rho^2$.

211

Example: Embedded-Atom Model of Falling Water

7.7 Example: Embedded-Atom Model of Falling Water

To assess the performance of the embedded-atom equation of state we consider the equilibration and collapse of three rectangular arrays of particles. In these three problems the gravitational field is chosen to result in a square equilibrated system with a density twice the normal stress-free density at the base of the system. The smallest system, $(N = 20 \times 32; H = 20)$ and g = 1/8 = 5/(2H), satisfies the three restrictions :

$$-10 < x < 10$$
; $y > 0$.

The largest system, $(N = n_x \times n_y = 80 \times 128$; H = 80) and g = 1/32 = 5/(2H), satisfies the restrictions :

$$-40 < x < 40$$
; $y > 0$.

Just as before, we use the embedded-atom equation of state for particles with unit mass and unit stressfree density :

$$E/N = \frac{1}{2}(\rho - 1)^2$$
; $P = \rho^2(\rho - 1)$,

so that the equilibrated profile is :

$$\frac{3}{2}\rho^2 - 2\rho - 2 + \frac{5y}{2H} = 0;$$

$$0 < y < H \longleftrightarrow 2 > \rho > 1.$$

The embedded-atom portion of the acceleration of the ith particle is

$$a_i^{\text{EA}} = -\sum_j [\rho_i + \rho_j - 2] \frac{r_{ij}}{|r_{ij}|} w'_{ij}$$

We add a pair core potential to prevent clumping,

$$\phi_{\rm core}(r<\sigma) = \epsilon \left(1-\frac{r^2}{\sigma^2}\right)^4 \ ; \ \epsilon=1 \ ; \ \sigma=0.6 \ ,$$

and then solve the *damped* equations of motion ,

$$\dot{v} = a - \left(\frac{v}{\tau}\right) \;,$$

for all the particles, where a is the usual embedded-atom plus pair core acceleration and the relaxation time τ is $n_x/8 = H/8$, somewhat smaller than the sound traversal time. After a relaxation equilibration period of

Lucy and Embedded-Atom Fluids

about ten sound traversal times, the constraints on the sides are removed and each of the roughly square samples of fluid is allowed to collapse, as is shown in Figure 7.3 .



Simulations carried out in this way show promptly that the free surface of the collapsing fluid is unstable, due to the lack of any surface tension. A simple way to introduce surface tension, detailed in Chapter 8, and adopted in the present simulations, is to add a *surface* potential :

$$\Phi_{
m surface} \propto \sum_i (
abla
ho)_i^2 \; .$$

This approach, with the proportionality constant equal to 0.1 , works very well for this problem.

The three simulations shown in Figure 7.3 correspond very nicely with one another. The snapshot times in the Figure are proportional to \sqrt{N} and the gravitational constant varies as $\sqrt{1/N}$. The times for the largest specimen are { 0, 20, 40 }, where the initial sound velocity $c = \sqrt{3\rho^2 - 2\rho}$ varies from 1, at the top of the column, to $\sqrt{8}$ at the bottom. Unlike the finite-

Example: Free Expansion of a γ -law Gas

element simulation shown in Figure 2.1 , the present SPAM simulations include no explicit failure model.

Figure 7.4 corresponds to the bottom row of Figure 7.3, and shows that part of the fluid for which the pressure is positive. The void in the center of each collapsing column is a tensile region, within which the fluid could separate into separate drops. The shape of this tensile region is well-defined, even in the smallest 640-particle simulation. From this collapsing column problem we conclude that the embedded-atom equation of state, with the core and surface potentials, provides a useful and rapidly-convergent hydrodynamic model for dense fluid flow under the influence of gravity.



with 640, 2560, and 10,240 particles. These three corresponding times match those of the last of the three rows of snapshots shown in Figure 7.3.

7.8 Example: Free Expansion of a γ -law Gas

Consider the special case of an expanding two-dimensional monatomic ideal gas. We imagine an *isentropic* equation of state for the gas. At constant entropy, the pressure of such a gas is proportional to the square of the density, $P \propto \rho^2$. Now consider the dynamics of such a gas, modeled by 2025 smooth Lucy particles with identical masses m = 1. The simple equation-of-state choice converts the smooth-particle equations of motion, the familiar equations of motion :

{
$$\dot{v}_i = -m \sum [(P/\rho^2)_i + (P/\rho^2)_j] \cdot \nabla_i w_{ij}$$
 }

to a simpler form :

$$\{ \dot{v}_i \propto \sum_j \nabla_i w_{ij} \}.$$

These last motion equations are *identical* to those of ordinary molecular dynamics, as we emphasized in Sections 7.2 and 7.4. In this special case

Lucy and Embedded-Atom Fluids

the smooth-particle weight function $w_{ij} \equiv w(r_{ij})$ plays the rôle of a traditional Newtonian pairwise-additive potential function. In the smoothparticle representation of the underlying Euler fluid the pressure tensor Pis a hydrostatic scalar ($\propto \rho^2$) and the heat flux vector Q vanishes.

We now illustrate the smooth-particle method—which looks here exactly like molecular dynamics—by applying it to a challenging problem in two space dimensions, the free expansion of a two-dimensional Euler fluid into a container four times its original size. An interesting feature of the problem is the need to consider boundary conditions and particle interfaces carefully. We will discuss this in detail in Chapter 8.

Because the Euler fluid ideal gas, with the isentropic equation of state $P = \rho e \propto \rho^2$, has neither viscosity (as discussed in Section 3.11) nor heat conductivity, there is an additional puzzling bit of thermodynamics to understand. In the *free* expansion of the gas no energy can be gained or lost (because no external forces interact with the gas) and there is no heat transfer with the surroundings. But according to Gibbs' statistical mechanics, the entropy of the gas must increase to reflect the increase in the "number of states" available to the expanded gas. The number of states varies as the Nth power of the volume ratio, $V_{\rm final}/V_{\rm initial}$, and the gas' entropy change reflects the (logarithm of the) increased number of states: $\Delta S = Nk \ln 4$. We will see that the solution to this puzzle (of a substantial entropy *increase* for an *isentropic* gas) arises naturally in the SPAM simulations. Let us first consider the details of the simulation scheme.^{6,7,8}

The simulation proceeds with the evolution of N = 2025 smooth particles, followed for a few thousand timesteps dt (where the distance traveled by a "fast" particle in a timestep is a small fraction of the range h). For each timestep in a general smooth-particle evolution the last four steps of the following five-step scheme repeat :

- (1) Choose the initial conditions, positions $\{r_i\}$, velocities $\{v_i\}$, and energies $\{e_i\}$ at a density $\rho_0 = 4$.
- (2) Compute ρ_i for each particle.
- (3) Compute the gradients $(\nabla T)_i$ and $(\nabla v)_i$ for each particle. These are typically required for computing the particle values of P_i and Q_i .
- (4) Compute P_i and Q_i at each particle, using the equation of state. For

 $^{^{6}}$ Hoover and Posch (1999).

 $^{^7\}mathrm{Hoover},$ Posch, Castillo, and Hoover (2000).

 $^{^{8}\}mathrm{Hoover}$ and Hoover (2001).

Example: Free Expansion of a γ -law Gas

the ideal-gas example both the viscous pressure stresses and the heat fluxes { Q_i } vanish.

(5) Compute $\dot{v}_i \equiv dv_i/dt$ and $\dot{e}_i \equiv de_i/dt$ —the time derivatives following the particle motion, including the above pressure tensors and heat flux vectors. These derivatives, along with $v_i \equiv \dot{r}_i$, can then be used in the Runge-Kutta integrator. The process then begins again at step 2.

Molecular dynamics requires only three of these steps, with the third and fourth skipped entirely. Step 2, the smooth-particle density sum, is equivalent to the potential energy sum in molecular dynamics.

A conventional fixed-mesh Eulerian grid continuum approach to this problem exhibits catastrophic instabilities⁷ unless artificial damping is added. A conventional Lagrangian continuum approach, with the mesh elements moving with the fluid, can exhibit two kinds of numerical instabilities, the "butterfly" and the "hourglass," as shown in Figure 7.5.⁸ The instabilities can be tamed by introducing specific viscous damping forces to control these modes, but the programming effort is severe. By contrast, the smooth-particle method requires no special precautions. Because the particle grid is continually evolving, with the field variables everywhere smooth, no catastrophic instabilities can occur.



A computer program designed to solve a flow problem requires initial conditions. For the free expansion problem, it is straightforward to choose

⁷Hoover, Posch, Castillo, and Hoover (2000).

 $^{^{8}}$ Hoover and Hoover (2001).

ws-book9x6

Lucy and Embedded-Atom Fluids

the initial particle coordinates (a square grid is fine). By introducing small random offsets—equivalent to small initial random particle velocities—the square symmetry of the grid can be broken. The evolution step—step 5 in the scheme on the last page—requires the pressures, which can be computed from the densities and energies at the locations of the particles, $P_i = \rho_i e_i$.



energy K in the free expansion of 16,384 Lucy particles. The contours corresponding to the average density and to the average kinetic energy separate the black above-average regions from the white below-average ones in the Figure . τ is the sound-traversal time . See Reference [6].

Figure 7.6 is an illustration of the resulting particle motion.⁶ The particles follow the equations of motion of molecular dynamics, with simple cubic-spline forces given by the derivative of Lucy's weight function. In the results shown the range of the weight function was chosen to be equal to six times the initial nearest-neighbor spacing. An underlying set of smooth particles makes it possible to evaluate field variables on a regular grid, not just at the particle locations. This flexibility is particularly advantageous if a rectangular mesh is required, as is the case for some graphics programs and is also the case for computing fast Fourier transforms of the field variables. The density and kinetic energy contours shown in the Figure were computed using the smooth-particle definitions. Black indicates

 $^{^{6}}$ Hoover and Posch (1999).

Example: Lucy-Fluid Shockwave Structure

above-average values and white below-average.

All the results calculated here were computed with *periodic* boundary conditions. These simulations reveal a serious problem, taken up in Chapter 8. Particles *interpenetrate* freely, with no clear interfacial interaction. *Mirror* boundary conditions are another possibility. For mirror boundaries each particle near, but inside the boundary (within the range h) produces an image particle *outside* the boundary, to which boundary values of velocity and energy per unit mass can be attributed. Treating problems in which the location of the boundary is unknown and varies with time (as in the surf near a beach or the formation of a crack or tear) is a challenging research area for smooth particles. Now that parallel processing with thousands of processors is a reality, some of the difficulties in treating boundaries are being alleviated by using the additional computer power to make a more sophisticated description of these interface problems.

Although the expansion example discussed here involves identical particles, with equal masses and the same weight function, sometimes a more general treatment is warranted. It is relatively easy to use a positiondependent particle size, with h depending upon local density. A more sophisticated treatment uses *ellipsoidal* particles, with the principal-axis lengths depending on the density gradient as well as the local density.

7.9 Example: Lucy-Fluid Shockwave Structure

Compression tends to form shockwaves whenever, as is usual, the sound velocity is an increasing function of density :

$$\left(\frac{\partial c^2}{\partial \rho}\right)_S = \left(\frac{\partial^2 P}{\partial \rho^2}\right)_S > 0 \ .$$

The two-dimensional ideal-gas polytropic isentrope and the embedded-atom fluid are two specially useful examples :

$$P = \frac{B_0}{2} \left(\frac{\rho}{\rho_0}\right)^2 ; P = B_0 \left[\left(\frac{\rho}{\rho_0}\right)^3 - \left(\frac{\rho}{\rho_0}\right)^2\right] .$$

When the higher-pressure portions of a moving soundwave overtake the slower lower-pressure components the wave steepens. In the absence of offsetting forces the gradients could become arbitrarily large. In fact, viscous forces broaden a wave, offsetting the steepening tendency of soundwaves

Lucy and Embedded-Atom Fluids

to become shockwaves. In the idealized one-dimensional situation a steady shockwave results when the steepening and broadening tendencies balance.

In this example problem we consider a "one-dimensional" shockwave from the standpoint of the conservation laws. In truth, a shockwave in a two-dimensional system is only one-dimensional on a time-and-spatiallyaveraged basis. In this sense, for one-dimensional waves with *local* values of the density, velocity, stress, heat flux, and energy, we can derive exact relations linking the local values of all these variables to the shock and particle velocities, (u_s, u_p) . The assumptions of locality and stationarity, coupled with conservation laws for mass, momentum, and energy, give a variety of exact relationships linking the flow variables within a shockwave. Analysis and applications of these relationships provide the primary sources of experimental pressure-volume equation-of-state information for materials at high pressures.

We focus here on the nature of the shock transition, the transformation of a cold low-pressure material to a hot high-pressure one. We point out that the viscous (and thermal) dissipation which together prevent real shockwaves from becoming infinitely steep persists in numerical approaches too. In smooth particle methods dissipation arises naturally, though it can, and usually should, be augmented by implementing von Neumann's "artificial viscosity" as is described in Section 2.6.

The basic one-dimensional shockwave geometry is illustrated in Figure 7.7. Three separate points of view, corresponding to three different coordinate systems, are illustrated there. In each of the frames considered hot fluid, at the left of the shockwave, is created by accelerating the cold fluid shown at the right.

Because all three views illustrate exactly the same process the choice of frame is one of æsthetics and convenience. The steady-state frame comoving with the shockwave simplifies most analyses. Conservation of mass, momentum, and energy in that frame can be used to link the final thermodynamic state properties { ρ_1, P_1, e_1 } to the initial state properties { ρ_0, P_0, e_0 } with the help of the shock and particle velocities u_s and u_p :

$$\rho_0 u_s = \rho_1 (u_s - u_p) ;$$

$$P_0 + \rho_0 u_s^2 = P_1 + \rho_1 (u_s - u_p)^2$$
;

$$u_s[P_0 + \rho_0(e_0 + \frac{1}{2}u_s^2)] = (u_s - u_p)[P_1 + \rho_1 e_1 + \frac{\rho_1}{2}(u_s - u_p)^2)].$$



Example: Lucy-Fluid Shockwave Structure



ws-book9x6

Lucy and Embedded-Atom Fluids

The first of these three identities, conservation of mass, equates the mass flows per unit area into and out of the shockwave in the coordinate frame fixed on the shockwave. For a weak shock the particle velocity is small and the shock velocity is the speed of sound. For stronger shocks the shock velocity u_s increases (linearly, to a rough approximation) with u_p . Figure 7.8 shows the variation for an ideal gas.



ticle (or piston) velocity u_p for a two-dimensional monatomic ideal gas. The intercept is the sound speed and the slope *s* there is related to the adiabatic derivative $(\partial B/\partial P)_S = 4s - 1$ where *B* is the adiabatic bulk modulus, $B = \rho(\partial P/\partial \rho)_S$.

The total momentum flux, $P + \rho u^2$ in a reference frame chosen with velocity u, includes a convective part, ρu^2 , and a comoving part, the pressure tensor P. If the wave is *stationary* then the momentum and mass fluxes are necessarily constant throughout the wave, even in the *non*equilibrium portion (where the longitudinal pressure-tensor component P_{xx} in the di-

221

Example: Lucy-Fluid Shockwave Structure

rection of propagation differs from the transverse component P_{yy}). This observation establishes the form of the "Rayleigh line", the *linear* dependence of the longitudinal stress on volume throughout the shockwave : $P_{xx} - P_0 \propto (1/\rho_0) - (1/\rho)$, where the proportionality constant is the square of the (constant) mass flux $M \equiv \rho u = \rho_0 u_s = \rho_1 (u_s - u_p)$. This interesting linear stress-volume relation follows from the constancy of the momentum flux :

$$P + \rho u^2 = P_0 + \rho_0 u_s^2 = P_1 + \rho_1 (u_s - u_p)^2 =$$
$$P_0 + M^2 / \rho_0 = P_1 + M^2 / \rho_1 \longrightarrow$$
$$(P - P_0) \propto (V_0 - V) .$$

Within the shockwave the energy flux contains the comoving heat flux Q as well as the equilibrium components indicated above. Q vanishes in the final (hot) and initial (cold) states. In those equilibrium regions, taking all three conservation equations into account, the "Hugoniot relation" linking the initial and final states follows :

$$e_1 - e_0 = \frac{1}{2}(P_1 + P_0)(v_0 - v_1)$$
,

where the volume per unit mass, v is the inverse of the mass density ρ . Note the close resemblance of this exact nonequilibrium relation to the equilibrium form of the First Law of Thermodynamics, specialized to isentropic $(\dot{S} \equiv 0)$ adiabatic processes:

$$dE = TdS - PdV \longrightarrow \Delta E_S = -P\Delta V_S$$
.

It is perfectly feasible, though tedious, to solve the three conservation relations numerically, as two coupled ordinary differential equations for the velocity u(x) and temperature T(x), provided that the constitutive relations giving the nonequilibrium stress P_{xx} and the nonequilibrium heat flux Q_x are given.⁹ In the case that these are not known in advance the detailed structure of the shockwave can be determined by simulation. In the event that no nonequilibrium effects are involved (as in states sufficiently far from the shockfront) the three equations can be used to eliminate the two velocities, u_p and u_s , with the Hugoniot relation result mentioned above.

For a given equation of state energy can be eliminated from that relation to give an explicit relationship linking the initial and final pressures to the

⁹Holian, Hoover, Moran, and Straub (1987).

Lucy and Embedded-Atom Fluids

degree of compression. For the two-dimensional ideal gas, for example, with E = PV, the Hugoniot relation gives :

$$e_1 - e_0 = P_1 v_1 - P_0 v_0 = \frac{1}{2} (P_1 + P_0) (v_0 - v_1)$$
.

The latter equality gives the hot/cold pressure ratio :

$$\frac{P_1}{P_0} = \frac{(3v_0 - v_1)}{(3v_1 - v_0)} \; .$$

With this relation relating the pressure to the compressed volume: $P_1(v_1)$ the conservation-of-mass and conservation-of-momentum equations can be solved to find the corresponding speeds, $u_s(v_1)$ and $u_p(v_1)$. Evidently the limiting threefold compression (where P_1 diverges) cannot be exceeded.

Consider now a numerical realization of the shockwave process for an ideal gas, using the smooth-particle model with Lucy's potential to represent that gas.¹⁰ The stationary-state boundary conditions can be achieved by matching the boundary mass flows according to the Hugoniot conditions. New slabs of "cold" particles, at an overall density ρ_0 , are introduced into the system (with speed u_s) periodically. The new particles are allowed to accelerate once they reach a distance h beyond the entrance boundary. At a distance h from the opposite (exit) boundary the relative particle motions in slabs of old "hot" particles are set to zero, with the particles leaving the system at the fixed speed $u_s - u_p$.

We consider modeling the ideal gas according to smooth particle applied mechanics. Because viscosity is absent, the equilibrium equation of state ,

$$P = \rho e = \frac{1}{2}\rho^2,$$

simplifies the smooth-particle motion equations ,

$$\left\{ \dot{v}_i = -m \sum_j [(P/\rho^2)_i + (P/\rho^2)_j] \cdot \nabla_i w_{ij} \right\} ,$$

to give the familiar simpler molecular dynamics form :

$$\left\{ \dot{v}_i = -m \sum_j \nabla_i w_{ij} \right\}$$

Again the weight function w(r) plays the rôle of a pair potential $\phi(r)$. As a direct consequence, all of the results which hold for ordinary molecular dynamics hold for the smooth-particle simulation of an ideal-gas shockwave:

⁽i) Strict time-reversibility of the Hamiltonian equations of motion.

¹⁰Kum, Hoover, and Hoover (1997).

Example: Lucy-Fluid Shockwave Structure

(ii) Viscous stresses which can be estimated from equilibrium properties.

(iii) Heat fluxes which can be estimated from equilibrium properties.

The time reversibility requires no special comment except for emphasizing the rôle of the boundaries. In simulating the shockwave a more-orderly lower-entropy fluid is converted into a hotter less-orderly one, which is then discarded. Any attempt to reverse the overall flow *must* fail—the flow *is* irreversible—with the reversed shockwave soon spreading out to form an adiabatic rarefaction fan. Because the equations of motion are exactly the same as those governing Hamiltonian molecular dynamics, the ideal-gas shockwave problem, treated with smooth-particle mechanics, follows the Liouville Theorem (incompressible flow in the phase space, discussed in Section 1.6), despite the macroscopic irreversibility of the shock process. Several detailed simulations have been carried out. Figure 7.9 is a snapshot illustrating the compression of a Lucy-potential fluid with

$$\phi(r) = (5/9\pi)(1+r)(1-\frac{r}{3})^3$$
.

In these simulations a cold Lucy "solid" at a density of unity enters at the left and is compressed to a higher density ($\rho/\rho_0 = 1.50$) and temperature (about 0.85) by the shock process. Evidently the viscosity and heat conductivity associated with the Lucy potential are sufficient to equilibrate the pressure tensor and the heat flux vector within distances of order 40. Green and Kubo's statistical theory of transport coefficients, reviewed by Zwanzig,¹¹ provides an understanding of these results.

If the Lucy representation agreed exactly with the ideal-gas equation of state, $P = \rho e$, we could solve the Hugoniot relation,

$$\Delta E = -\langle P \rangle \Delta V \iff \Delta e = -\langle P \rangle \Delta v ,$$

for the pressure and energy consistent with the compression [$(\rho_0 = 1 \rightarrow \rho_1 = 1.5)$ so that $\Delta v = -1/3$]. The resulting energy change,

$$e_0 = \frac{1}{2} \sum w \simeq 0.5 \rightarrow \Delta e = e_1 - e_0 = e_1 - 0.5$$
,

must be equal to the average pressure ,

$$\langle P \rangle = (P_1 + P_0)/2 = (\rho_0 e_0 + \rho_1 e_1) = (0.5 + 1.5e_1)/2 ,$$

multiplied by the (negative of the) volume change, 1/3:

$$\frac{e_1 - 0.5 = (0.25 + 0.75e_1)/3}{^{11}\text{Zwanzig (1965).}} \longrightarrow (e_1 = 7/9 \rightarrow P_1 = \rho_1 e_1 = 7/6).$$

Lucy and Embedded-Atom Fluids

This simple analysis does agree very well with the measured pressure.¹⁰

A more sophisticated approach can be based on the Grüneisen equation of state mentioned in Section 2.4:

$$P_T = P - P_0 \equiv \gamma E_T / V = \gamma (E - E_0) / V$$
.

where P_T and E_T are the "thermal" contributions to the pressure and energy, over and above their "cold" values,

$$P_0 \equiv -dE_0/dV$$
.

The shockwave data imply that the Grüneisen constant γ is about 1.5. A more detailed analysis of this problem is a good research topic. Both the instabilities of the Lucy lattice structures and the curious flow of heat within the shockwave deserve more investigation. The flow of heat is in the direction opposite to Fourier's heat-flow Law, $Q = -\kappa \nabla T$, within much of the shockwave! See the accompanying Figure 7.9.



Figure 7.9: Kinetic temperature profile in the Lucy shockwave. The longitudinal, transverse, and mean temperatures are all shown. This example has a very interesting feature—because the flow of heat, relative to the comoving frame of the material, is always to the left, from hot to cold, heat moves *against* the direction predicted by Fourier's law in a considerable part of the wave. See Reference [10].

References

7.10 References

- (1) W. G. Hoover and M. Ross, "Statistical Theories of Melting", Contemporary Physics **12**, 339-356 (1971).
- (2) M. Born and K. Huang, Dynamical Theory of Crystal Lattices (Clarendon, Oxford, 1985).
- (3) J. Mayer and M. G. Mayer, *Statistical Mechanics* (Wiley, New York, 1977).
- (4) Wm. G. Hoover and S. Hess, "Equilibrium and Nonequilibrium Thermomechanics for an Effective Pair Potential used in Smooth Particle Applied Mechanics", Physica A 231, 425-438 (1996).
- (5) S. M. Foiles, M. I. Baskes, and M. S. Daw, "Embedded Atom Method Functions for the FCC Metals Cu, Ag, Au, Ni, Pd, Pt, and their Alloys", Physical Review B 33, 7983-7991 (1986).
- (6) Wm. G. Hoover and H. A. Posch, "Entropy Increase in Confined Free Expansions *via* Molecular Dynamics and Smooth-Particle Applied Mechanics", Physical Review E 59, 1770-1776 (1999).
- (7) Wm. G. Hoover, H. A. Posch, V. M. Castillo, and C. G. Hoover, "Computer Simulation of Irreversible Expansions via Molecular Dynamics, Smooth Particle Applied Mechanics, Eulerian, and Lagrangian Continuum Mechanics", Journal of Statistical Physics **100**, 313-326 (2000).
- (8) Wm. G. Hoover and C. G. Hoover, "SPAM-Based Recipes for Continuum Simulations", Computers in Science and Engineering 3, 78-85 (2001).
- (9) B. L. Holian, W. G. Hoover, B. Moran, and G. K. Straub, "Shockwave Structure via Nonequilibrium Molecular Dynamics and Navier-Stokes Continuum Mechanics", Physical Review A 22, 2798-2808 (1987).
- (10) O. Kum, Wm. G. Hoover, and C. G. Hoover, "Temperature Maxima in Stable Two-Dimensional Shockwaves", Physical Review E 56, 462-465 (1997).
- (11) R. W. Zwanzig, "Time Correlation Functions and Transport Coefficients", Annual Review of Physical Chemistry 16, 67-102 (1965).

August 16, 2006 17:44

WSPC/Book Trim Size for 9 in x 6 in ws-book9x6

226

Lucy and Embedded-Atom Fluids

Chapter 8

SPAM: Limitations and Difficulties

/ Surface Tension / Tensile Instability / Monaghan's Motion Equations / Stress and Rigid-Body Rotation in Continuum Mechanics / Dynamic and Static Constitutive Relations / Example Deformations with Stress and Strain Rates / Jaumann's Rigid-Body Stress Rotation Rate / Conservation of Angular Momentum / Artificial Transport Coefficients / Residual Stress—Artificial Plasticity /

/ 9 Figures /

8.1 Summary

The main strengths of SPAM are simplicity, flexibility, and transparency. The main weaknesses are surface and tensile instabilities, smearing and interpenetration of surfaces and interfaces, together with a disquieting lack—angular momentum is not conserved. In this Chapter we point out mechanisms underlying these unpleasant features of SPAM and indicate means which can be used to combat them.

8.2 Surface Tension

The excess energy associated with surfaces (due to the reduced number of attracting neighbors), "surface tension", is responsible for real drops' favoring the minimum-surface spherical shape. Surface tension is totally absent in macroscopic constitutive relations, and has to be added in whenever it is desirable to reproduce physical effects due to surface tension. Capillary waves, with typical wavelengths of order one millimeter and driven by sur-

SPAM: Limitations and Difficulties

face tension, are such an effect. The situation with smooth particles is no different. If the energy is solely a function of density the dynamics is free to choose the maximum-entropy configurations consistent with the energy.

The embedded-atom potential,

$$\Phi_1 = \sum_i \phi_i = \sum_i \frac{mB_0}{2\rho_0} \left[\left(\frac{\rho_i}{\rho_0} \right) - 1 \right]^2 ,$$

has its energy minimum when each particle has the target density ρ_0 . Figure 8.1 shows the configuration which results when 225 unit-mass embeddedatom particles, originally distributed randomly within a rigid 15×15 square, are equilibrated for a time of 1000 using a relaxation time of 10. It is evident that the particles generate a structure with excess *surface*, including internal holes. The overall tendency is to form one-dimensional strings, with the spacing in the strings adjusted so that each particle's density is $\rho_0 = 1$. For the configuration shown here the total potential energy is 10^{-6} with the maximum individual deviation from the target density about 10^{-4} . It is evident that the embedded-atom potential, by itself, has no real surface tension. This problem was briefly discussed in Section 5.3.

To defeat the string-forming tendency, so as to impose a more regular structure, one can first of all introduce forces tending to separate nearby pairs. The pair core potential, Φ_2 , includes any pair of particles with a pair separation less than the core size σ :

$$\Phi_2 = \sum_{r_{ij} < \sigma} \frac{1}{4} [1 - (r_{ij}/\sigma)^2]^4 .$$

This potential vanishes for either regular lattice, square or triangular, at or below the target density imposed by a nearest-neighbor spacing, $d > \sigma$. A 225-particle calculation like the pure embedded-atom one, but with a strong additional pair repulsion, resulted in triangular-lattice fragments.

Evidently one can tailor the relative strengths of the bulk embeddedatom potential and a repulsive pair potential so as to reproduce a desired surface tension. The simplest way to analyze that problem is to avoid the irregularities associated with curvature and corners by relaxing a semiinfinite strip of material. With periodic boundaries in the x direction, and free boundaries in the y direction, the surface is free to relax to its minimum-energy structure. The extra energy which results, relative to a regular structure, can then be ascribed to an effective surface tension.

Surface Tension



computed densities, using Lucy's weight function with h = 3, showed very small deviations from the target density of unity. The lefthand structure corresponds to the pure embedded-atom potential Φ_1 . The righthand structure results from a similar simulation using $\Phi = \Phi_1 + \Phi_2$, with $\sigma^2 = 0.5$. Density deviations up to two percent result from this change.

Another way to introduce an effective surface tension is to include forces, along with an equivalent internal energy, which minimize the *density gradients* which characterize surfaces. Either of the *ad hoc* force choices :

$$F_i \propto -(r_{ij}/|r_{ij}|)w_{ij}$$
 or $F_i \propto +(r_{ij}/|r_{ij}|)w'_{ij}$,

corresponds to attraction, and tends to minimize the surface area.

A more physically motivated contribution to the energy, of the form,

$$\Phi_3 = \frac{1}{2} \sum_i (\nabla \rho)_i^2 \text{ or } \frac{1}{4} \sum_i (\nabla \rho)_i^4 ,$$

where $(\nabla \rho)_i$ is the density gradient *at* Particle *i* :

$$(\nabla \rho)_i \equiv \sum_j m_j \left(\frac{w'}{r}\right)_{ij} r_{ij} \;,$$

leads to interparticle forces involving both w' and w''. For the simpler $\frac{1}{2}(\nabla \rho)^2$ choice, applied to a single pair of one-dimensional particles,

SPAM: Limitations and Difficulties

straightforward differentiation gives the accelerations :

$$\ddot{x}_1 = -\ddot{x}_2 = -\frac{d}{dx_1} [w_{12}^{\prime 2} + w_{21}^{\prime 2}]/2 = -2\frac{x_{12}}{|x_{12}|} w_{12}^{\prime} w_{12}^{\prime \prime} ; \ x_{12} = x_1 - x_2 .$$

The algebra is more complicated for many particles but adds very little time to the simulation. An associated array of vectors, one for each particle, $\{ \nabla \rho(i) \}$, needs to be computed and stored. This can be done within the same (ij) particle-pair loop which generates the particle densities. The two-dimensional equations of motion which follow from the quadratic form of Φ_3 are as follows for the x coordinate of Particle i:

$$\begin{split} \ddot{x}_i &= \sum_j [(\nabla \rho)_j - (\nabla \rho)_i]_x \frac{\partial}{\partial x_i} \left(\frac{xw'}{r}\right)_{ij} + \\ &\sum_j [(\nabla \rho)_j - (\nabla \rho)_i]_y \frac{\partial}{\partial x_i} \left(\frac{yw'}{r}\right)_{ij} \longrightarrow \\ \ddot{x}_i &= \sum_j [(\nabla \rho)_j - (\nabla \rho)_i]_x \left(\frac{+y^2w'}{r^3} + \frac{x^2w''}{r^2}\right)_{ij} + \\ &\sum_j [(\nabla \rho)_j - (\nabla \rho)_i]_y \left(\frac{-xyw'}{r^3} + \frac{xyw''}{r^2}\right)_{ij} \,. \end{split}$$

The \ddot{y} equations of motion can be obtained by permuting x and y.

Figure 8.2 shows three examples of relaxed simulations with a Hamiltonian including both pair-core and density-gradient— $\frac{1}{2}(\nabla \rho)^2$ —potentials:

$$\mathcal{H} = K + \Phi_1 + \Phi_2 + \Phi_3 \; .$$

Initially the particles were motionless, arranged in a regular square grid with unit density. With a timestep of 0.05 and in the absence of any damping the energy for this Hamiltonian is conserved quite well, to an accuracy of one part per million in a single integration step. The structures obtained by viscous relaxation provide reasonable initial conditions for complex simulations involving fracture and flow, as we will see in more detail in the examples worked out in Chapter 9.

231

Tensile Instability



Figure 8.2: Relaxed configurations of 400, 1600, and 6400 embeddedatom particles. Damping, with $\tau = 10$, was applied for a time interval 0 < t < 1000. The summed-up densities, using Lucy's weight function with h = 3 show density deviations along the boundary of several percent and density deviations near the center of order one half percent. The simulations include an additional surface energy proportional to the square of the density gradient, $\Phi_3 = \frac{1}{2} \sum (\nabla \rho)^2$; $\Phi = \Phi_1 + \Phi_2 + \Phi_3$; $\sigma^2 = 0.5$.

A disquieting feature of relaxed smooth-particle structures is their lamellar character. Particularly near free boundaries particles tend to arrange themselves in "strings" ("sheets" in three dimensions). Although the densities of particles in these strings are near the stress-free density the distribution of neighbors is visibly anisotropic. The string-forming tendency of particles near the boundary can be reduced somewhat by a judicious choice of the core size σ in Φ_2 . We illustrate this idea in the tension-test simulations of Chapter 9.

8.3 Tensile Instability

Swegle, Hicks, and Attaway¹ pointed out that SPAM simulations of solids under tension exhibit an exponentially fast instability. If the overall stress state is positive (tensile) and slowly varying in space and time, then the SPAM equations of motion ,

$$\ddot{r}_{i} = \dot{v}_{i} = m \sum_{j} [(\sigma/\rho^{2})_{i} + (\sigma/\rho^{2})_{j}] \cdot \nabla_{i} w_{ij} = -m \sum_{j} [(P/\rho^{2})_{i} + (P/\rho^{2})_{j}] \cdot \nabla_{i} w(r_{ij}) ,$$

¹Journal of Computational Physics (1995).

ws-book9x6

SPAM: Limitations and Difficulties

correspond to those of ordinary molecular dynamics :

$$\dot{r} = v$$
; $m\dot{v} = -\sum \nabla_i \phi(r_{ij})$,

with the smooth-particle weight function w(r) playing the rôle of an attractive molecular dynamics pair potential $\phi(r) \propto -w(r)$. With a positive weight function, along with positive pressure and density, such accelerations correspond to an energetically stable dynamics with smooth repulsive forces. But in regions where the pressure is negative, corresponding to tension, the accelerations instead represent dynamics with purely attractive forces, an unstable situation in which particles can clump together with a binding energy growing quadratically, rather than linearly, in the number of particles.

To illustrate the tensile instability here we consider a regular periodic triangular-lattice structure with the embedded-atom potential function :

$$\Phi = \sum_{i} \frac{1}{2} (\rho_i - \rho_0)^2 ;$$

where ρ_0 is the stress-free density at each lattice site. We put the lattice under tension and begin with a single slowly moving particle, Particle 1 :

$$\epsilon_{xx} = \epsilon_{yy} = 0.01 \; ; \; \dot{x}_1 = 2^n / 10^6 \; ; \; n = 1 \dots 8 \; .$$

Figure 8.3 shows the time development of the total system kinetic energy that results when the density is calculated with Lucy's weight function (h = 3), and where the stress-free density and particle mass are both set equal to unity. After a brief induction time, on the order of the Einstein vibrational period, the kinetic energy grows *exponentially* with time.

The explanation for this instability is simple: the *tensile* interaction of particle pairs corresponds to molecular dynamics with a *negative* purelyattractive pair potential proportional to $-w(r_{ij})$. This instability grows with a characteristic rate of the order of the collision frequency.

Three ideas for avoiding tensile instability are (i) introducing the core potential of Section 8.2, (ii) introducing von Neumann-Richtmyer artificial viscosity (which damps the motion of approaching pairs of particles), and (iii) modifying the conventional relationship between coordinates and velocities :

$$\{ \dot{r} = v \} \longrightarrow \{ \dot{r} \equiv \langle v \rangle \}.$$

Monaghan's Motion Equations

This last idea, suggested by Monaghan,² though completely *ad hoc*, and with some unexplored consequences for the interpretation of the continuity equation, can be implemented in such a way as to conserve linear momentum exactly. Monaghan has also introduced *ad hoc* density-dependent repulsive forces analogous to the pair core potential.³ We discuss the details of his velocity-averaging idea in the following Section.



8.4 Monaghan's Motion Equations

Surfaces and interfaces are specially difficult to treat with smooth particles, because the particle method spreads each particle's influence in space. This makes a faithful representation of sharp boundaries difficult. This interface

²Journal of Computational Physics (1989).

³Journal of Computational Physics (2000).

SPAM: Limitations and Difficulties

problem is particularly noticeable when one material collides with another, as in the free-expansion example problem of Section 7.8 . In continuum mechanics we would expect to see no interpenetration, with the boundary separating the two materials allowing the transport of momentum and energy, but not mass. Two real gases, colliding with each other, would interpenetrate only according to the phenomenological diffusion rate, which is entirely negligible on a macroscopic length scale. Smooth particles do not ordinarily behave in this way. Unless special precautions are taken, two adjacent materials represented by smooth particles *can* interpenetrate, with one passing right *through* the other. To avoid this unrealistic feature of smooth particles Monaghan devised a special equation of motion which we now consider. To begin, consider the individual particle velocities { v_i }.

We emphasized earlier that the particle velocities typically differ from the field velocity v at the same location, due to the averaging process involved in defining field variables. It might seem logical to avoid this interpenetration problem by abandoning the usual relations :

$$\{ \dot{r}_i \equiv v_i \} ,$$

and replacing the particle velocities { v_i } with the corresponding field velocities, evaluated at the particle locations { r_i } :

$$\{ \dot{r}_i \stackrel{?}{\equiv} v(r_i) = \sum_j v_j w_{ij} / \sum_j w_{ij} \}.$$

But a little reflection shows that this approach does not conserve overall momentum. To demonstrate this shortcoming consider three equally-spaced and equally-massive particles in a line, with spacing d between nearest neighbors, and with velocities { -2, +1, +1 } which sum to zero. Let us calculate the field velocities at the particle locations :

$$\dot{x}_1 = \frac{(-2+\alpha)}{(1+\alpha)}$$
; $\dot{x}_2 = \frac{(+1-\alpha)}{(1+2\alpha)}$; $\dot{x}_3 = \frac{(+1+\alpha)}{(1+\alpha)}$

where α is the ratio w(d)/w(0) and w(2d) = 0 so that second-nearest neighbors do not interact. Momentum conservation would require that the summed-up velocities vanish :

$$(1+\alpha)(1+2\alpha)(\dot{x}_1 + \dot{x}_2 + \dot{x}_3) =$$

$$(-2+\alpha)(1+2\alpha) + (1-\alpha)(1+\alpha) + (1+\alpha)(1+2\alpha) = 3\alpha^2 \stackrel{?}{=} 0.$$

Monaghan's Motion Equations

Evidently momentum is only conserved if the three particles are separated beyond the range of the weight function so that they don't interact at all. Thus, replacing { $\dot{r}_i = v_i$ } by { $\dot{r}_i = v(r_i)$ } is a relatively poor idea.

Monaghan³ discovered an only slightly more complicated scheme which does conserve momentum exactly:

$$\{\dot{r}_i = v_i\} \rightarrow$$

$$\{ \dot{r}_i = v_i + \sum_j (m_j v_j - m_i v_i) (w/\rho)_{ij} \}.$$

Either the arithmetic or geometric mean of ρ_i and ρ_j can be used to define ρ_{ij} . The antisymmetric nature of the sum implies that linear momentum is conserved :

$$\sum_{i} \sum_{j} (m_j v_j - m_i v_i) (w/\rho)_{ij} \equiv 0 \longrightarrow$$

$$\sum m_i \dot{r}_i \equiv \sum m_i v_i \; ,$$

as an *exact* consequence of Monaghan's scheme.

An example problem which demonstrates the usefulness of Monaghan's average velocity is the free expansion example problem treated in Section 7.8 . A 2500-particle square-lattice Lucy gas, with m = 1; $P = \rho^2/2$; and -12.5 < x, y < +12.5, has initial density 4 within the central quarter of the stationary square space -25 < x, y < +25. The initial sound speed is $c = \sqrt{dP/d\rho} = 2$. If the freely-expanding gas is allowed to interact with its periodic images in the usual way there is considerable interpenetration. This interpenetration is eliminated with Monaghan's approach. The two approaches are compared in Figure 8.4 after an elapsed time of 10.

³Journal of Computational Physics (1989).

236

ws-book9x6



SPAM: Limitations and Difficulties

occurs in the free expansion problem using the usual smooth-particle equations of motion. The illustration at the right (at exactly the same elapsed time) indicates the reduction in interpenetration which occurs when Monaghan's velocity averaging is used. In both cases particles initially in the first and third quadrants of the problem are shaded more lightly than those initially in the second and fourth quadrants.

Continuum Mechanics: Stress; Rigid-Body Rotation 8.5

As we saw in Chapter 3, the time-rate-of-change of stress has two distinct components, thermodynamic (or "constitutive") and geometric (or "orientational"). Thermodynamic state changes of stress can result from strains or from strain rates, as well as from energy changes. These changes are intrinsic constitutive properties of the material. They differ from orientational effects-when a stressed body is rotated about its center of mass, its stress components relative to a fixed external coordinate system must reflect that rotation.

Consider the simplest such example, a body with constant shear strain ϵ_{xy} in which only the stress-tensor element $\sigma_{xy} \equiv \sigma_{yx}$ is initially nonzero :

$$\sigma = \begin{pmatrix} 0 & \eta \epsilon_{xy} \\ \eta \epsilon_{xy} & 0 \end{pmatrix} \ .$$

Dynamic and Static Constitutive Relations

If this sheared system is rotated as a rigid body, counterclockwise through an angle θ , then the same stress tensor, referred back to the original xycoordinate system, becomes :

$$\sigma(t) = \begin{pmatrix} \cos\theta - \sin\theta\\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} 0 & \eta\epsilon_{xy}\\ \eta\epsilon_{xy} & 0 \end{pmatrix} \begin{pmatrix} \cos\theta & \sin\theta\\ -\sin\theta & \cos\theta \end{pmatrix} = \\ \begin{pmatrix} -\eta\epsilon_{xy}\sin(2\theta) + \eta\epsilon_{xy}\cos(2\theta)\\ +\eta\epsilon_{xy}\cos(2\theta) + \eta\epsilon_{xy}\sin(2\theta) \end{pmatrix},$$

where the last result follows from the double-angle formulæ:

 $2\sin\theta\cos\theta = \sin(2\theta)$; $\cos^2\theta - \sin^2\theta = \cos(2\theta)$.

Notice that stress, a second-rank tensor, is invariant to a rotation of 180 degrees. Whenever rotation and deformation are combined more complicated material behavior results, as is discussed in the following Sections.



8.6 Dynamic and Static Constitutive Relations

Stress and strain in static or nearly-static linear-elastic simulations are easy to calculate. Computer simulations of large and rapid deformations, with

ws-book9x6

SPAM: Limitations and Difficulties

path-dependent irreversible contributions to the stress, are complicated by several types of nonlinearity. When stress depends upon the past history, a dynamic, rather than a static, constitutive relation is required. This need is exacerbated when particle methods are used. Lack of a grid-based definition of deformation means that "strain" is not a simple function of the particle coordinates. For particles, strain can only be calculated as a time integral (of strain rate) from a specified initial condition.

With particles, current stress values have to be computed from the past histories of the particle velocities. Rather than computing stress directly, in terms of instantaneous strains, it is natural instead to consider instantaneous deformation and rotation *rates*. This approach avoids focusing on the connection to the current configuration to that of the initial conditions of the problem. In dynamical simulations we must compute the time integral of the stress rate $\dot{\sigma}$ to find the current stresses σ :

$$\sigma(\rho,\epsilon,e) \longleftarrow \dot{\sigma}(\rho,\dot{\rho},e,\dot{e},\nabla v,\dots) \ .$$

The time-rate-of-change of the stress tensor depends directly on deformation rates derived from ∇v . This explicit dependence of stress on a time integration of the stress rate leads to an important constitutive question (actually a *problem*) associated with stress rotation, which we examine next.

A dynamic computer simulation provides local values of the density, velocity, and energy through the space-time integration of the flow equations for mass, momentum, and energy. Constitutive relations for the stress tensor and the heat flux vector usually require knowing the gradients ∇v and ∇T . The velocity gradient tensor ∇v is directly available. It contains the fundamental description of deformation and rotation.

For particles a simple definition of the hydrodynamic (averaged) velocity gradient at point *i* follows from differentiation of the definition of ρv :

$$\rho(r_i)v(r_i) \equiv m \sum_j v_j w_{ij} \ \longrightarrow$$

$$\rho_i(\nabla v)_i + v_i(\nabla \rho)_i = \rho_i(\nabla v)_i + mv_i \sum_j \nabla_i w_{ij} = m \sum_j v_j \nabla_i w_{ij} - \cdots$$

$$\rho_i(\nabla v)_i = m \sum_j (v_j - v_i) \nabla_i w_{ij} \; .$$

Dynamic and Static Constitutive Relations

For æsthetic reasons it is more usual to use a symmetrized form for $(\nabla v)_i$:

$$(\nabla v)_i = m \sum_j \frac{(v_j - v_i)}{\rho_{ij}} \nabla_i w_{ij} ,$$

where ρ_{ij} is either the arithmetic or geometric mean of ρ_i and ρ_j .

On intuitive grounds it appears "obvious" that the velocity gradient's symmetric and antisymmetric parts, $\frac{1}{2}(\nabla v + \nabla v^t)$ and $\frac{1}{2}(\nabla v - \nabla v^t)$ respectively, describe "deformations" (changes in size and/or shape) as well as "rotations" (changes in orientation). Then, because the stress tensor is necessarily symmetric, it cannot be affected by the "rotational" antisymmetric part of ∇v . The strain-rate tensor responsible for stress is naturally given by the symmetrized velocity-gradient tensor $\frac{1}{2}(\nabla v + \nabla v^t)$. In terms of (constant) Lamé constants the linear-elastic small-strain stress-rate tensor in a fixed Cartesian frame is just the time derivative of the elastic constitutive equations :

$$\sigma_{xx} = (\lambda + 2\eta)\epsilon_{xx} + \lambda\epsilon_{yy} \; ; \; \sigma_{xy} = \eta\epsilon_{xy} \; ; \; \sigma_{yy} = (\lambda + 2\eta)\epsilon_{yy} + \lambda\epsilon_{xx}$$

$$\dot{\sigma}_{xx} = (\lambda + 2\eta)\dot{\epsilon}_{xx} + \lambda\dot{\epsilon}_{yy} \; ; \; \dot{\sigma}_{xy} = \eta\dot{\epsilon}_{xy} \; ; \; \dot{\sigma}_{yy} = (\lambda + 2\eta)\dot{\epsilon}_{yy} + \lambda\dot{\epsilon}_{xx} \; .$$

When there is no rotation in the motion,

$$\frac{1}{2}(\nabla v - \nabla v^t) \equiv 0 ,$$

this description can easily be extended to large strains. In the simplest case, with constant moduli and with $\epsilon_{xy} \equiv 0$, this approach leads directly to "logarithmic strains",

$$\epsilon_{xx} = \ln[L_x(t)/L_x(0)]$$
; $\epsilon_{xy} \equiv 0$ [no rotation]; $\epsilon_{yy} = \ln[L_y(t)/L_y(0)]$.

A consistent treatment of rotation, based on the antisymmetric part of the velocity gradient tensor, introduces difficulties.

To understand these difficulties for deformable bodies, consider first the simpler rigid-body case. Let a stressed rigid body undergo counterclockwise rotation, at the rate ω in the xy plane. We assume that the rotation rate is slow enough that centrifugal effects can be ignored. The rotation rules for a second rank tensor (or the fully equivalent geometric description of
SPAM: Limitations and Difficulties

momentum flux in a rotated frame) give "well-known" results :

$$\dot{\sigma}_{xx} = -2\omega\sigma_{xy} ; \ \dot{\sigma}_{xy} = \omega[\sigma_{xx} - \sigma_{yy}] ; \ \dot{\sigma}_{yy} = +2\omega\sigma_{xy} .$$

As Prager⁴ emphasized, in 1961, these frequently-rediscovered relations for $\{ \dot{\sigma}_{xx}(\omega), \dot{\sigma}_{xy}(\omega), \dot{\sigma}_{yy}(\omega) \}$, the geometric time-rate-of-change of stress, had already been pointed out by Jaumann in 1911, and so are now known as the "Jaumann stress rates".

For a slowly-rotating rigid body there is no ambiguity or approximation in these equations. The Jaumann expressions for *mathematical* stress rotation (as opposed to the *physical* centrifugal effects on stress) describe a stress tensor embedded in the rotating body. But when the body in question is changing *shape* as well as *orientation* due to shear *there is no unambiguous correct description of the rotational effects on stress*. The shear deformation of a deck of cards, parallel to the cards' surfaces, is plainly irrotational, for instance. The gist of the question, "How is stress affected by rotation?", is deciding the extent to which simple shear generates a rotational contribution to the stress tensor's orientation. If rotation is to play a rôle in a dynamics determined by ∇v this rôle can only be expressed in terms of the *antisymmetric* part of the velocity gradient: $\frac{1}{2}(\nabla v - \nabla v^t)$.

Because the "true" extent of stress rotation during shear deformation is a material property depending upon microstructure, not something contained *a priori* in a structureless continuum mechanics, nonlinear rotational solutions of the continuum equations can include an embarassing richness (justly called "academic excess" by Belytschko, Liu, and Moran)⁵ for solids undergoing simultaneous finite deformation and rotations. Any computationally complete "material model", must somehow specify the rotational aspect of stress evolution. Whether or not such an arbitrary model "makes sense" in a particular case can only be resolved by comparison with experimental (or molecular dynamical) data. Any such comparison would promptly reveal that no computational *continuum* model can be fully faithful in its description of the behavior of an experimental or idealized molecular-dynamical material composed of particles.

⁴Introduction to the Mechanics of Continua.

⁵Nonlinear Finite Element Analysis for Continua and Structures (2000).

Example Deformations with Stress and Strain Rates

8.7 Example Deformations with Stress and Strain Rates

The simplest of shear deformations, "simple shear", can be illustrated by the following prototypical deformation field :

$$\dot{x} = \dot{\epsilon}_{xy}y \; ; \; \dot{y} = 0 \; ;$$

 $x \longrightarrow x + \epsilon_{xy}y \; ; \; y \longrightarrow y \; .$

Evidently such a simple shear transforms a circle, $x^2 + y^2 = r^2$, into the corresponding ellipse, $(x - \epsilon y)^2 + y^2 = r^2$. See Figure 8.6. Surprisingly, such a simple shear is also *exactly equivalent* to a rotationless deformation in a rotated $(x, y) \xrightarrow{\text{rot}} (x', y')$ coordinate system :

$$x' \longrightarrow e^{+\epsilon'} x' ; y' \longrightarrow e^{-\epsilon'} y' .$$

To illustrate, consider the special case $\epsilon = \sqrt{4/3}$, which also maps the triangular lattice into itself (as indicated in Figure 8.7, at the end of this Section). The simple shear mapping is the following :

$$x \longrightarrow x + \sqrt{4/3} y \ ; \ y \longrightarrow y \ ,$$

and is illustrated at the top of the Figure. The axes shown embedded in the material each rotate 30 degrees. At the bottom of the Figure an alternative, but equivalent, two-step mapping is indicated. First, the circle is rotated clockwise 30 degrees. Then, longitudinal strains are imposed in the directions (30 degrees and 120 degrees) indicated by the embedded axes:

$$x' \longrightarrow \sqrt{3^{+1}}x' ;$$
$$y' \longrightarrow \sqrt{3^{-1}}y' .$$

Notice that the logarithmic strain ϵ' in this (special) example is $\ln(\sqrt{3})$.

242

SPAM: Limitations and Difficulties



The rotation *rate* during simple shear, based on the local velocity gradient tensor, is constant :

$$\omega = \frac{1}{2} \left[\frac{\partial v_y}{\partial x} - \frac{\partial v_x}{\partial y} \right] = \frac{-\dot{\epsilon}_{xy}}{2} \; .$$

Whether or not a stress rotation actually exists in simple shear is debatable. Consider the simple shear of an initially-square finite element with sides parallel to the x and y axes. Define the angle α made by vectors embedded in the sides (initially $\alpha = \frac{\pi}{2}$). In simple shear the time-rateof-change $\dot{\alpha}$ varies as 1/t for long times t, which would suggest a slowing rate of rotation. This slowing is indeed actual for the material contained in the original square. On the other hand, if we imagine a steady homogeneous plastic deformation, with a constant yield stress, the local rotation rate must *eventually* become everywhere the same. Does the underlying material actually *rotate* in simple shear? If so, does the rotation rate slow down as time goes on? There are no universal answers to these questions.

We consider the simple-shear example in more detail, just to make this point clear. Consider a triangular lattice of particles with nearest neighbors linked by Hooke's-law forces, and undergoing the relatively-large simple shear :

$$u_x = \sqrt{4/3}y \longrightarrow \epsilon_{xy} = \sqrt{4/3}$$
.

For small deformations the Lamé constants for the crystal are simply related

Example Deformations with Stress and Strain Rates

to the nearest-neighbor Hooke's-law force constant κ :

$$\eta = \lambda = \frac{\sqrt{3}\kappa}{4}$$
; $B = \lambda + \eta = 2G = \frac{\sqrt{3}\kappa}{2}$

Figure 8.7 shows both "before" ($\epsilon_{xy}=0$) and "after" ($\epsilon_{xy}=\sqrt{4/3}$) views of the lattice. Initially the nearest-neighbor spacing is unity and the volume per particle is $\sqrt{\frac{3}{4}}=0.866025$. In the event that nearest neighbor particles are connected by unbreakable bonds, with the Hooke's-law potential $\phi=\kappa(r-1)^2/2$, it is apparent that the energy corresponding to the deformation is the linear-elastic value :

$$\frac{\Phi}{N} = \frac{\kappa}{2}(\sqrt{3} - 1)^2 = 0.267949\kappa \; ,$$

corresponding to an energy density

$$E/V = \sqrt{\frac{4}{3}} \frac{\Phi}{N} = 0.309401\kappa$$
 .

Now consider two continuum approximations to this same deformation. If it is viewed as "simple shear", then the energy density after deformation is $\frac{1}{2}\sigma_{xy}\epsilon_{xy}$, with $\epsilon_{xy} = \sqrt{4/3}$, so that

$$E/V = \int \sigma : d\epsilon = \int_0^{\sqrt{4/3}} \sigma_{xy} d\epsilon_{xy} = \eta \int_0^{\sqrt{4/3}} \epsilon_{xy} d\epsilon_{xy} =$$
$$(\eta/2) \frac{4}{3} = \frac{2}{3} \eta = 0.288675 \kappa ,$$

a result which is about seven percent too low.

Alternatively the same deformation could be viewed as due to a stretch in the x direction, with a compensating shrinkage in the y direction :

$$\epsilon_{xx} = \frac{1}{2} \ln 3 = -\epsilon_{yy} \; .$$

According to this picture, the energy density after deformation is

$$\frac{1}{2}(\sigma_{xx} - \sigma_{yy})(\epsilon_{xx} - \epsilon_{yy}) = \frac{1}{2}\frac{\sqrt{3}}{4}(\ln 3)^2\kappa = 0.261312\kappa$$

also too low, and with an error more than twice that of the first continuum estimate. The extent of the difference between the two approaches to the deformation energy suggests that strains of order unity correspond to energy errors of order ten percent.

244

SPAM: Limitations and Difficulties



8.8 Dynamics with Jaumann's Stress Rotation Rate

There is no question that Jaumann's rigid-body expression for (slow) stress rotation at an angular velocity ω is exactly correct :

$$\dot{\sigma}_{xy} = 2\omega \left(\frac{\sigma_{xx} - \sigma_{yy}}{2} \right) \; ; \; \frac{\dot{\sigma}_{xx} - \dot{\sigma}_{yy}}{2} = -2\omega\sigma_{xy} \; .$$

For a deforming body the same approach eventually leads to unphysical results. We saw that simple shears of order unity lead to energy errors of order ten percent in the last Section. Let us analyze the corresponding dynamics here. Consider including "Jaumann-rate" rotation in the description of simple shear. We choose again a linear elastic material with a linear constitutive equation $\dot{\sigma}_{xy} = \eta \dot{\epsilon}_{xy}$. A positive strain rate $\dot{\epsilon} = du_x/dy$ corresponds to a negative rotation rate $\omega = -\dot{\epsilon}/2$. Including the Jaumann-rate rotation in the equations of motion requires the solution of two simultaneous differential equations :

$$\dot{\sigma}_{xy} = \eta \dot{\epsilon} + \omega [\sigma_{xx} - \sigma_{yy}] = \dot{\epsilon} \left[\eta - \frac{\sigma_{xx} - \sigma_{yy}}{2} \right] ;$$

$$\frac{1}{2} [\dot{\sigma}_{xx} - \dot{\sigma}_{yy}] = -2\omega\sigma_{xy} = \dot{\epsilon}\sigma_{xy} .$$

Dunamics with Jaumann's Stress Rotation Rate

The solution of the two coupled differential equations which corresponds to vanishing initial stresses is sinusoidal in time :

$$\sigma_{xy} = \eta \sin(\dot{\epsilon}t) = -\eta \sin(2\omega t) ;$$

$$\frac{1}{2} [\sigma_{xx} - \sigma_{yy}] = \eta [1 - \cos(\dot{\epsilon}t)] = \eta [1 - \cos(2\omega t)]$$

According to linear elasticity, the strain energy per unit volume is

$$\int \sigma : d\epsilon = \frac{1}{2\eta} \left[\sigma_{xy}^2 + \frac{1}{4} (\sigma_{xx} - \sigma_{yy})^2 \right] ,$$

so that at unit shear strain the corresponding energy per unit volume,

$$\frac{\eta}{2}\sin^2(1) + \frac{\eta}{2}[1 - \cos(1)]^2 = 0.459698\eta$$

is similar to, but definitely not the same as, the result from linear elasticity at unit strain, $\frac{1}{2}\eta$.

A perfectly *plastic* material (see Section 2.8) behaves in an analogous way, giving contradictory results for nonlinear strains. In simple shear the plastic shear stress σ_{xy} maintains the constant value Y so that the work of deformation per unit area (without taking rotation into account) is just $Y \epsilon_p$ for a total plastic strain ϵ_p . The deformation of the last Section, with

$$\epsilon_{xy} = \sqrt{\frac{4}{3}} \longrightarrow \frac{E}{V} = 1.154701Y ,$$

could just as easily be described in terms of the principal-axes' strains:

$$\epsilon_{11} = -\epsilon_{22} = \frac{\ln 3}{2} \longrightarrow \frac{E}{V} = 1.098612Y \; .$$

From the physical standpoint it is clear that the state of stress in truly steady shear (assuming proper thermostating to prevent any long-time heating of the workpiece) becomes stationary. On the other hand any attempt to include the rotation in the time development of the stress tensor yields a sinusoidal dependence of stress on time, with the stress tensor varying at twice the frequency of rotation. Evidently the only way to determine whether or not the relatively small effects associated with "stress rotation" improve simulations is to compare simulations with and without rotational contributions. It is fortunate that these troubling effects are small.

SPAM: Limitations and Difficulties

8.9 Conservation of Angular Momentum

Smooth-particle algorithms are typically formulated so as to conserve mass, linear momentum, and energy. These conservative formulations typically do not conserve angular momentum. Conservation of angular momentum is a fundamental principle of continuum mechanics and also applies to particle mechanics when the interactions between particles are central forces. Consider the angular momentum L of a two-dimensional system of particles and its time-rate-of-change \dot{L} , where the forces are assumed to be "central" (parallel to the line linking the interacting particles) :

$$L \equiv m \sum_{i} [(x\dot{y})_{i} - (y\dot{x})_{i}] \longrightarrow \dot{L} \equiv m \sum_{i} [(x\ddot{y})_{i} - (y\ddot{x})_{i}] =$$

$$m \sum_{i} [(x_i(yF/r)_{ij} - y_i(xF/r)_{ij}] = m \sum_{i < j} [(x_{ij}(yF/r)_{ij} - y_{ij}(xF/r)_{ij}] \equiv 0.$$

Evidently the cancellation of the single-particle sums, $+\sum xF_y$ and $-\sum yF_x$, is due to the *central* nature of the forces. Central forces are parallel to the radius vector $r_i - r_j$ with the vector force on Particle *i* due to Particle *j* equal to the scalar, $F = -d\phi/dr$, multiplied by the unit vector with *x* and *y* components in the direction of $r_i - r_j$:

$$\left(\begin{array}{c} \frac{x_i - x_j}{|r_i - r_j|} \ , \ \frac{y_i - y_j}{|r_i - r_j|} \end{array}\right) = \left(\frac{x}{r}\right)_{ij} \ , \ \left(\frac{y}{r}\right)_{ij}$$

In smooth particle mechanics *noncentral* tensor forces are the rule. The interaction between Particles i and j typically makes a nonzero contribution to \dot{L} :

$$\begin{split} \dot{L}_{ij} &= \left(\frac{mw'xy}{r}\right)_{ij} \left[\left(\frac{\sigma_{yy}}{\rho^2}\right)_i + \left(\frac{\sigma_{yy}}{\rho^2}\right)_j - \left(\frac{\sigma_{xx}}{\rho^2}\right)_i - \left(\frac{\sigma_{xx}}{\rho^2}\right)_j \right] + \\ &\left(mw'\left(\frac{x^2 - y^2}{r}\right)\right)_{ij} \left[\left(\frac{\sigma_{xy}}{\rho^2}\right)_i + \left(\frac{\sigma_{xy}}{\rho^2}\right)_j \right] \neq 0 \;. \end{split}$$



Conservation of Angular Momentum

Figure 8.8: 217-atom specimens following a full rotation using (i) SPAM velocities and (ii) Monaghan's average velocities. For this rotation the time dependences of the total angular momenta of the rotating specimens are shown. The decays of angular momentum found are consequences of the *noncentral* forces. These forces respond to the local velocity gradients.

As a consequence, angular momentum for a relatively simple field-free system, such as a rotating elastic hexagon, is not conserved. See Figure 8.8. In relatively simple problems conservation can be artificially imposed, by adding a correctional torque to each ij pair of particles proportional to \dot{L}_{ij} . Because the positive and negative torques do not quite cancel it is necessary

SPAM: Limitations and Difficulties

to scale the corrections to *enforce* cancellation. This approach is successful in conserving angular momentum with SPAM⁶ for problems like that shown in the Figure. For large-scale complex problems, with several independently rotating parts, a local correction scaling would need to be developed to avoid artificial generation and transport of angular momentum.

8.10 Artificial Transport Coefficients

The particulate gridlike nature of SPAM provides *artificial* transport coefficients automatically. The *magnitude* of these effects can be estimated by considering the ordinary fluid transport coefficients described by kinetic theory. In Section 3.11 we pointed out that the gas-phase equation of state $P \propto \rho^2$ provides a smooth-particle dynamics identical to the molecular dynamics of particles interacting with the weight-function potential w(r):

$$\Phi_{\rm gas} = \sum_{i < j} w_{ij} \; .$$

The simple dense-fluid embedded-atom equation of state,

$$P = B_0[(\rho/\rho_0)^3 - (\rho/\rho_0)^2] ,$$

considered in Section 5.2 similarly provides a dynamics identical to the molecular dynamics of particles interacting with a collective many-body potential :

$$\Phi_{\text{liquid}} = \sum_{i} \frac{mB_0}{2\rho_0} \left[(\rho_i / \rho_0) - 1 \right]^2 \; ; \; \rho_i = m \sum_{j} w_{ij} \; .$$

Either dimensional analysis or elementary kinetic theory can be used to establish that the diffusivities (squares of lengths divided by time) describing mass, momentum, and energy transport can be written as products of a typical velocity and a "mean free path". The latter is a mean distance between successive scattering events in the gas, liquid, or solid. Estimates of the transport coefficients inherent in particle representations like SPAM can be based on this idea. In a dense fluid the mean free path is of the same order as the spacing between particles so that the artificial transport occurs with an effective diffusivity of $(V/N)\omega$, where ω can be estimated

⁶Hoover, Hoover, and Merritt (2004).

Residual Stress—Artificial Plasticity in SPAM

from the Einstein frequency. This approach has proved useful for understanding particulate contributions to both the kinematic viscosity and the thermal diffusivity.⁷

8.11 Residual Stress—Artificial Plasticity in SPAM

Just as there are grid-induced viscosities and conductivities in SPAM there are also locked-in stress fluctuations, even in fluids. A real fluid is able to *flow* to relieve shear stress. A fluid or solid represented by SPAM cannot begin to flow until the applied stress exceeds the extra stresses due to the discrete particulate nature. Fortunately these stresses are not "large" and can be controlled. We can estimate them from the individual particle stresses which remain locked into a SPAM structure in the absence of motion. We have seen in Section 3.6 that in the prototypical case of the Lucy weight function, with h = 3, m = 1, and an overall density of unity, that the effective densities of the square and triangular lattices differ by about 0.002 from the overall density and by $5/\pi h^2 = 0.177$ from the random arrangement. We would expect to find that the residual stresses are comparable to the product of the bulk modulus and this lattice-to-lattice volume strain difference of 10^{-3} or 10^{-4} . To illustrate, let us investigate residual stresses for a moderately-large system using the dense-*fluid* equation of state :

$$P = B_0 \left[\left(\frac{\rho}{\rho_0} \right)^3 - \left(\frac{\rho}{\rho_0} \right)^2 \right] \ . \label{eq:posterior}$$

With a modulus and density of unity, the relaxed 1024-particle structure $(t = 100,000; \tau = 0.001)$ shown in Figure 8.9 has typical variations in density of ± 0.00003 . Changing the modulus to 10 leads to much the same distribution of densities, but with the locked-in stresses approximately 10 times greater.

⁷Hoover and Posch (1996).

250

SPAM: Limitations and Difficulties



Figure 8.9: Relaxed periodic structure with the embedded-atom equation of state, using Lucy's weight function with h=3.5, for which the regular triangular and square lattices have nearly the same density. The rms density deviation per particle is approximately 0.00003 in this case.

References

8.12 References

- J. W. Swegle, D. L. Hicks, and S. W. Attaway, "Smooth Particle Hydrodynamics Stability Analysis", Journal of Computational Physics 116, 123-134 (1995).
- (2) J. J. Monaghan, "On the Problem of Penetration in Particle Methods", Journal of Computational Physics 82, 1-15 (1989).
- (3) J. J. Monaghan, "Smooth Particle Hydrodynamics without a Tensile Instability", Journal of Computational Physics **159**, 290-311 (2000).
- (4) W. Prager, *Introduction to the Mechanics of Continua* (Ginn & Company, Boston, 1961 [Dover reprint, 1973]).
- (5) T. Belytschko, W. K. Liu, and B. Moran, Nonlinear Finite Element Analysis for Continua and Structures (Wiley, New York, 2000).
- (6) Wm. G. Hoover, C. G. Hoover, and E. C. Merritt, "Smooth Particle Applied Mechanics: Conservation of Angular Momentum with Tensile Stability and Velocity Averaging", Physical Review E 69, 016702 (2004).
- (7) Wm. G. Hoover and H. A. Posch, "Numerical Heat Conductivity in Smooth Particle Applied Mechanics", Physical Review E 54, 5142-5145 (1996)

August 16, 2006 17:44

WSPC/Book Trim Size for 9 in x 6 in ws-book9x6

252

SPAM: Limitations and Difficulties

Chapter 9

SPAM: Sample Applications to Solids

/ Summary / Tension Test / Tension via Standard Molecular Dynamics / Initial and Boundary Conditions for Tension / Tension Test via SPAM-like Molecular Dynamics / Tension Test via SPAM / Penetration and Failure Algorithms / Penetration via Finite Elements / Penetration via Standard Molecular Dynamics / Penetration via SPAM-like Molecular Dynamics / Penetration via SPAM / Research Suggestion /

/ 10 Figures /

9.1 Summary

Solid-phase simulations, particularly those involving "slow" processes (slow relative to the sound velocity), are particularly challenging problems in view of the instabilities discussed in the preceding Chapter. Here we discuss two problems, one slow and tensile, the other fast and mainly compressive, in order to clarify the distinction and to illustrate the difficulties which can arise. Though these same problems can be solved using standard molecular dynamics, we will see that the atomistic structures required for a realistic yielding response require impractically large specimens. For this reason the problem types treated here are intrinsically *continuum* problems. Nevertheless, in both the problem types explored here—a tension test and the penetration of a plate by a fast-moving ball—we explore both the atomistic and the smooth-particle approaches as well as a hybrid "SPAM-like molecular dynamics". Such solid-phase studies represent a particularly fertile area for development of novel SPAM algorithms and constitutive relations along the lines of the proposals of Chapter 8.

SPAM: Sample Applications to Solids

9.2 The Tension Test

The "tension test", in which a cylinder, or bar, of material is gradually pulled apart, provides the basic stress *versus* strain relations necessary to a continuum description of solid deformation. "Gradually" means that the overall strainrate is very small relative to the inverse sound-traversal time of the specimen ,

$$\dot{\epsilon}_{xx} << c/L_x$$
 .

For small homogeneous strains,

$$\epsilon_{xx} = \frac{\partial u_x}{\partial x} = \frac{\Delta L_x}{L_x} \; ,$$

the longitudinal stretching of a purely elastic material, free of transverse loads, defines Young's Modulus, E:

$$\sigma_{xx} = E\epsilon_{xx}$$
; $\sigma_{yy} = \sigma_{zz} = 0$.

See Figure 9.1 for three snapshots of a tensile test carried out with the conventional Lagrangian finite-element continuum code Dyna3d . This simulation was carried out in two space dimensions by using a "plane-strain" specimen constrained to retain its initial one-zone thickness perpendicular to the xy plane.

In the unconstrained y direction, perpendicular to the applied tensile load, the resulting strain, ϵ_{yy} , is typically (but not necessarily!)¹ negative (compressive). The perpendicular-to-parallel ratio of strains in the tension test defines a second elastic quantity, Poisson's ratio ν :

$$\nu = \frac{-\epsilon_{yy}}{\epsilon_{xx}} \; .$$

The Lamé-constant description of the same tensile experiment appears in Section 2.7 . In the unconstrained three-dimensional case the two transverse strains are equal :

$$\sigma_{yy} = \sigma_{zz} = 0 \longrightarrow \epsilon_{yy} = \epsilon_{zz} \; .$$

The two- and three-dimensional Young's moduli are

$$E_{2D} = \frac{4\eta(\lambda+\eta)}{\lambda+2\eta} ; \ E_{3D} = \frac{\eta(3\lambda+2\eta)}{\lambda+\eta} .$$

¹ "Auxetic and Related Systems", Physica Status Solidi **242b**, March 2005.

The Tension Test



Figure 9.1: Extension of a two-dimensional "plane-strain" tapered elastic-plastic bar using Dyna3d . The motion is driven by a longitudinal displacement of the bar ends, with an overall strain of 0.10 at t=64,000. The small-strain ratio of stress to strain defines Young's longitudinal modulus E=2.5, the limiting slope of the stress-strain plot. For larger "plastic" strains the longitudinal stress defines the yield stress, $\sigma_{xx} \rightarrow Y = 0.025$. In this example the yield stress was chosen equal to E/100 so that plastic flow begins at a strain $\epsilon_{xx}=0.01$. The snapshots correspond to overall strains of $\Delta L/L=0.050,\ 0.075,\ 0.100$. The shading indicates levels of plastic strain. Note the void formation near the center of the specimen.

The basic measured data are the strains ($\epsilon_{xx}, \epsilon_{yy}, \epsilon_{zz}$) as functions of the nonzero longitudinal stress, σ_{xx} . For small strains these data provide

SPAM: Sample Applications to Solids

a two-parameter— (E, ν) , (λ, η) , or (B, G)—elastic description. In either two or in three space dimensions two independent material properties are required to represent both resistance to compression (or dilatation) and to shear. At larger strains the stress-strain data can provide a *plastic* largestrain flow law for the yield stress Y, as well as a measure of the ultimate failure stress of the specimen. The elastic, plastic, and failure data which result from the tension test are the ingredients of a continuum constitutive model for the material being tested.

Though the elastic-plastic tension test is relatively easy to model with a finite-element code like Dyna3d, this same problem is a severe challenge for SPAM. The test involves both quasistatic deformation and tension, while SPAM is at its best in modeling rapid processes involving compression. Nevertheless, the ideas involved in simulating the tension test with SPAM are stimulating. Typically, imagination and innovation are required whenever a new type of problem is attacked. We turn now to the simulation of the tension test with three versions of particle mechanics (i) standard molecular dynamics, (ii) a "SPAM-like" variation of molecular dynamics, and (iii) conventional SPAM.

9.3 Tension Test via Standard Molecular Dynamics

Simulation of the tension test with molecular dynamics requires first of all a forcelaw. Then initial conditions, describing the specimen, and boundary conditions, describing the imposed loads on the specimen, need to be given. We describe these three components of the simulation in turn.

The set of pairwise-additive potential functions :

$$\left\{\phi_{m,n}(r<\sqrt{2}) = \frac{m}{n-m}(2-r^2)^n - \frac{n}{n-m}(2-r^2)^m\right\},\$$

all having a common minimum, $\phi = -1$ at r = 1, and a common cutoff, with $\phi \equiv 0$ for $r > \sqrt{2}$, provides convenient forcelaw choices.² It is possible to vary the stiffness of the potential, through m and n, and the stressfree triangular lattice involves only nearest-neighbor interactions. In the two examples considered in this Chapter we choose (m = 4, n = 8):

$$\phi(r < \sqrt{2}) = (2 - r^2)^8 - 2(2 - r^2)^4$$
.

 $^{^{2}}$ Hoover and Hess (1996).

Boundary Conditions for Tension

Choosing a particle mass of unity completes the definition of the length, time, and distance scales, all based on the common values of the interparticle spacing, binding energy, and particle mass. In the stable cold force-free triangular lattice, with a nearest-neighbor spacing of unity, the binding energy for a bulk particle is -3 and the number density and mass density of the lattice are $\sqrt{4/3}$.

The second derivative at the potential minimum, $\phi''(1) = 4nm$, defines the force constant κ characterizing the vibrational frequency of a diatomic molecule with a single pair bond. This force constant is simply related to the bulk sound velocity of the stress-free static-lattice,

$$c_{m,n} = \sqrt{\frac{\eta + \lambda}{\rho}} = \sqrt{3nm} \; ; \; c_{4,8} = \sqrt{96} = 9.798 \; .$$

As a point of interest, the second derivative of the popular Lennard-Jones potential at its minimum ,

$$\phi_{\rm LJ} = r^{-12} - 2r^{-6} \longrightarrow \phi_{\rm LJ}''(1) = 72$$

coincides with that of potential $\phi_{3,6}$. Unlike the Lennard-Jones potential $\phi_{m,n}$ does not diverge for small r. The potential at r = 0,

$$\phi(0) = \frac{m2^n - n2^m}{n - m} ; \ \phi_{3,6}(0) = 48 ; \ \phi_{4,8}(0) = 224$$

gives an indication of the material strength. With the forcelaw chosen we turn to the problem of developing boundary conditions for tension.

9.4 Boundary Conditions for Tension

It is not at all obvious how best to apply tension to a specimen composed of individual particles. Straightforward attempts to apply isolated forces to the surface particles at the two ends of the specimen lead promptly to failure. A similar failure results if a constant velocity is applied at the ends. An attempt to supply a homogeneous strain field with $v_x \propto x$ inhibits flow in the y direction. A likely solution to this problem, not explored here, is the imposition of *periodic* boundary conditions in the tensile direction.

The particle simulations described here were instead all based on applying a time-and-space-dependent "external force" to all those particles located within two fixed spatial windows in the interior of the specimen. The resulting applied tensile force has extreme values $\pm \delta F$ on two planes

SPAM: Sample Applications to Solids

(lines, in two dimensions) within the specimen, and falls off linearly with separation from these extreme-value planes, as is shown in Figure 9.2 . We also choose these accelerations proportional to the time, which has the effect that the total applied loads increase nearly linearly with time. All particle velocities are initially set equal to zero. In all the particle examples considered here each of the two spatial windows comprises $\frac{3}{8}$ the original length of the specimen and the simulation times are at least ten sound traveral times.



men. The acceleration decreases in magnitude linearly with the separation from the two x values $\pm 3L/16$ where the acceleration has its extreme values $\pm \delta F$. The amplitude of the acceleration is increased linearly with time, $\delta F \propto t/t_{\rm max}$.

We begin with a stress-free portion of a triangular lattice containing $96 \times 24 = 2304$ particles, much like the top illustration of Figure 9.3. A simulation with $\delta F = 0.01$ provides typical results. The calculation was

259

Boundary Conditions for Tension

carried out for a total time of 640, a few dozen sound traversal times. At t = 618, with applied loads of ± 130 corresponding to $\sigma_{xx} \simeq 6$ and a strain $\epsilon_{xx} \simeq 0.10$ near the specimen center, the structure exhibited internal slip failure (second illustration in Figure 9.3). Shortly thereafter the specimen broke into four pieces (lower illustrations in the Figure). The unrealistic nature of this large-strain brittle failure is typical of single-crystal simulations using ordinary molecular dynamics. Before turning to more realistic SPAM-like and SPAM simulations we discuss the construction of the initial conditions for these simulations in the following Section.



260

ws-book9x6

SPAM: Sample Applications to Solids

9.5 Initial Conditions for Tension Using SPAM

A perfectly rectangular tension specimen could break or yield equally-well anywhere along its length L_x . For our SPAM-like and SPAM simulations we use *tapered* specimens—as was also done in the Dyna3d finite-element example—so as to force failure to occur near the middle of the bar. We choose a coordinate system with the bar centered on the origin and we arbitrarily choose its nominal length $L_x = 4L_y$, so that the "aspect ratio" of the workpiece is 4. The extra length gives us needed flexibility in applying the piecewise-linear external forces illustrated in Figure 9.2.

We begin by placing particles randomly within a *tapered* rectangle :

$$-L_x/2 < x_{\text{taper}} < +L_x/2 \; ; \; -y_{\max}(x) < y_{\text{taper}} < +y_{\max}(x) \; ;$$

$$y_{\max}(x) = \left(\frac{L_y}{2}\right) \left[0.93 + 0.10 \sin^2\left(\frac{\pi x_{\text{taper}}}{L_x}\right) \right]$$

In equilibrating the particle positions we use a density-dependent force, a surface force, and a strength force, described in detail in the next Section. Viscous relaxation (starting with a large viscosity, and reducing it at long times) within these boundaries eventually leads to a nearly-static load-free tension specimen. See Figure 9.4. This Figure illustrates an initial condition generated for a tension specimen with $96 \times 24 = 2304$ particles.



elastically during relaxation.

Tension Test via SPAM-like Molecular Dynamics

9.6 Tension Test via SPAM-like Molecular Dynamics

Here we detail the simulation of a tension test for a version of molecular dynamics chosen to closely resemble SPAM. In this case we will add both surface tension and strength to the simple hydrostatic equation of state ,

$$P = B_0 \left[\left(\frac{\rho}{\rho_0} \right)^3 - \left(\frac{\rho}{\rho_0} \right)^2 \right]$$

choosing both the bulk modulus B_0 and the reference density ρ_0 equal to unity.

In modeling the tension-test problem with SPAM-like molecular dynamics one has to choose between two alternative methods for determining the mass density ρ , as mentioned in Section 3.7. We adopt the simpler of the two approaches here, computing the particle densities from the individual pair contributions :

$$\{ \rho_i \equiv m \sum_j w(r_i - r_j) \} .$$

The initial mesh shown in Figure 9.4 uses the very smooth generalization of Lucy's weight function, with a range h = 2.30 that maximizes the shear modulus. See again Figure 5.7 for the h dependence of the shear modulus.

Because the forces corresponding to this density-dependent equation of state,

$$P = \rho^3 - \rho^2 \longleftrightarrow F_i = -\nabla\Phi \; ; \; \Phi = \sum \frac{1}{2}(\rho - 1)^2 \; ,$$

provide no surface tension, we have also added a surface potential,

$$\Phi_{\text{surface}} = \frac{1}{20} \sum_{i} \frac{1}{2} (\nabla \rho)_i^2 ,$$

where the sum includes the squared density gradient evaluated at each particle's location, giving a surface force for Particle i:

$$F_{
m surface} \propto -rac{1}{2}
abla_i \sum_j (
abla
ho)_j^2 \; .$$

Larger coefficients, of order unity, destabilize the tension bar's shape, by overcoming its shear strength, resulting in a surface-dominated relaxed structure which is approximately circular.

262

ws-book9x6

SPAM: Sample Applications to Solids

We have further included the pair "core" potential,

$$\Phi_{\rm core} = 100 \sum_{i < j} [1 - (r_{ij}^2 / \sigma^2)]^4 ; \ r_{ij}^2 < \sigma^2 = 0.64 ,$$

to discourage particles' overlapping. Finally, solid *strength* is provided by using the invariant curvature potential introduced in Chapter 5 :

$$\Phi_{\rm curv} = \frac{1}{2} \sum_{i} \left[\left(\rho_{xx} - \rho_{yy} \right)_{i}^{2} + 4(\rho_{xy})_{i}^{2} \right] ,$$

where the (xx, xy, yy) subscripts indicate derivatives of the density with respect to x and y:

$$\begin{split} \rho_{xx} &= \partial^2 \rho / \partial x^2 ; \\ \rho_{xy} &= \partial^2 \rho / \partial x \partial y ; \\ \rho_{yy} &= \partial^2 \rho / \partial y^2 . \end{split}$$

This curvature potential, with h = 2.30 and the smooth weight function,

$$w_{\text{smooth}}(\tilde{r} = \frac{r}{h = 2.30} < 1) = \frac{7}{5.29\pi} (1 - \tilde{r})^4 (1 + 4\tilde{r}) ,$$

with three continuous derivatives, provides a shear modulus $G = \eta = 0.5$.

After equilibrating the smooth-particle mesh for many sound traversal times, including a damping force $-mv/\tau$, with τ increasing from 1 to 4, we obtained the relaxed structure of Figure 9.4. Applying the tensile loads indicated in Figure 9.2 and with $\tau = 0.1$ we obtain the snapshots from the time history shown in Figure 9.5. The maximum tensile loads for this SPAM-like molecular dynamics were about ± 1 , corresponding to an applied stress $\sigma_{xx} \simeq 0.05$.

Evidently the maximum speed induced by the external forces must be considerably less than the sound speed in order that the external load be applied more-or-less homogeneously throughout the specimen. Faster deformation, with the same form for the external forces, can lead to the formation of *two* necks rather than one. This same effect is present in the finite-element case.

i i

Tension Test via SPAM

Figure 9.5: Three snapshots, equally spaced in time, of a deforming tension test specimen with the initial taper of Figure 9.4. The SPAM-like molecular dynamics was carried out for a total time of 20,000. The total applied load increases nearly linearly with *time*. The piecewise-linear spatial dependence of the final applied load is shown in Figure 9.2.

9.7 Tension Test via SPAM

The main challenge presented by the tension test is modeling the inelastic "plastic" portion of the specimen. There material should flow plastically in forming a "neck" and in generating new surfaces. Any attempt to add additional density to surface particles, by integrating the continuity equation for $\dot{\rho}$ rather than summing weight functions, will lead to inconsistencies once plastic strain develops and creates new surfaces. For simplicity, all three particle simulations carried out in this Chapter, with both molecular dynamics and SPAM, are in two space dimensions. The same methods can be easily extended to three dimensions.

SPAM: Sample Applications to Solids

The tensile simulation with "SPAM-like" molecular dynamics includes the invariant curvature potential in order to provide elastic strength. In SPAM the elastic modulus is instead provided by the constitutive relations. Typical solids fail at strains of order 0.01 to 1.00, causing the specimen to break in half. For any realistic simulation of the tension test not only the elastic strength (as characterized by the shear modulus η), but also the plastic yield stress Y and the tensile failure stress σ_{tensile} need to be specified or otherwise reproduced.



with time. The spatial variation of the load is shown in Figure 9.2 .

Tension Test via SPAM

For the sample SPAM simulation illustrated here, in Figure 9.6 , we have again used the equation of state $P=\rho^3-\rho^2$, with equal values of the two Lamé constants $\lambda=\eta=\frac{1}{2}$, together with an arbitrary yield strength Y=0.01, and a tensile failure stress $\sigma_{\rm tensile}=0.05$. Just as in the molecular dynamics simulation we include also the pair core potential, $\phi=100[1-(r^2/\sigma^2)]^4$ with $\sigma=0.8$, together with viscous damping.

With the form of the density calculation decided (here we use Lucy's weight function with h = 3) and the constitutive model chosen there are still some remaining additional variations on the basic simulation to consider. Monaghan's velocity averaging, von Neumann's viscous drag forces, and Jaumann stress rotation are all possible options. Our use of drag forces, $F_{\rm drag} = -v/\tau$, makes von Neumann's viscosity unnecessary. We include only the Jaumann stress rotation in the simulation shown in Figure 9.6.

Trials combining Monaghan's velocity averaging with viscous drag revealed that the combination can be unstable. Damping relative to the local stream velocity ,

$$F_{\rm drag} = -(v - \langle v \rangle)/\tau ;$$

$$\langle v \rangle_i = v_i + \sum_j (v_j - v_i) w_{ij}/\rho_{ij} ;$$

$$\rho_{ij} = (\rho_i + \rho_j)/2 ,$$

seems not to improve the situation.

Figure 9.6 shows three stages in the plastic deformation of a twodimensional tensile specimen with the chosen external-force amplitude increasing linearly with time. A more sophisticated approach could choose the time variation of the amplitude through a "feedback" equation designed to impose a particular strainrate. A superior alternative to imposing the constraint condition C(t) = 0 is to adjust the force amplitude F_{ext} according to a relaxation equation, $\dot{F}_{\text{ext}} = -C(t)/\tau$. It is interesting to see that in this problem the final "failure" of the tensile specimen, when it comes apart into pieces, is a consequence of the particulate nature of SPAM rather than being a built-in part of the constitutive relation. We consider failure phenomena from the constitutive viewpoint in the following Section.

SPAM: Sample Applications to Solids

9.8 Failure Algorithms

In continuum mechanics dynamic phase equilibria are hard to treat because they involve the creation of new interfaces between coexisting materials. Without additional rules governing the nucleation rate and orientation of these interfaces the problem of locating them is ill-posed. Evidently rules governing nucleation would necessarily incorporate a statistical component. The first liquid region within a homogeneous gradually-heated solid could occur anywhere. "Failure", either ductile or brittle, is similar. Both the locations and the orientations of cracks, voids, or slip lines, any of which can lead to the failure of a solid, need either to be specified or to be imposed by a statistical rule.

Failure criteria can be based on (arbitrary functions of) internal energy, stress, or strain, either instantaneous or integrated over time. Because experimental failure data usually entail large fluctuations and because the microscopic details of failure surfaces vary from material to material it is difficult to conceive of a definitive test to use in developing such a model. Certainly *continuum* mechanics is inherently unable to describe the complex flaw-induced surfaces formed by typical fracture processes.

SPAM offers a way out of this impasse. SPAM contains its own failure mechanism, the separation of particles. Like experimental data, there is no simple way to relate failure in a one-to-one way with analytic failure models. On the other hand the possibility of using different weight functions (perhaps with different weight functions for different particles of the same material), different particle sizes, and different equations of state makes it quite possible to "tune" the particle simulation results to match failure data for particular materials.

9.9 Penetration Mechanics

Penetration is an important failure mode, a challenge for traditional continuum mechanics. The prototypical problem is the penetration of a plate by a fast-moving ball. The materials, their surfaces, and the relative velocity are the initial conditions. The simplest boundary conditions are to ignore all forces other than those due to the interaction of the plate with the projectile. If the impact is sufficiently fast, relative to the sound traversal time of the plate, the lack of support forces is unimportant. To simplify the graphics displays for slower impacts it is convenient to choose the total

Penetration via Continuum Mechanics

momentum equal to zero.

Energy is a useful conceptual basis for analyzing penetration. The internal energy of the impacted plate must increase in the penetration process, both through the creation of new surface (through the mechanisms of plastic flow, melting, or vaporization) and by accelerating the lost material to a velocity comparable to that of the projectile. This simple idea gives an estimate for the ball kinetic energy necessary to penetrate a plate.

Matching the character of the fragments formed during the penetration process is a demanding test of simulations. In the next four Sections we begin by simulating penetration with conventional continuum mechanics and conventional molecular dynamics. This latter approach is entirely free of assumptions (other than the choice of interparticle forces) and produces quasi-experimental data against which the predictions of numerical continuum simulations can be tested. We then consider both SPAM-like and conventional smooth-particle tests for comparison with the continuum and molecular dynamics data.

9.10 Penetration via Continuum Mechanics

In order to correlate continuum simulations with molecular dynamics simulations it is convenient to match the equations of state. We note that the family of (m, n) pair potentials introduced in Section 9.3 corresponds to the set of static cold-lattice mechanical equations of state for the energy and pressure with the energy minimum at $v = v_0$:

$$e_{0} = \frac{3m}{n-m} \left(2 - \frac{v}{v_{0}}\right)^{n} - \frac{3n}{n-m} \left(2 - \frac{v}{v_{0}}\right)^{m};$$
$$P_{0}v_{0} = \frac{3nm}{n-m} \left[\left(2 - \frac{v}{v_{0}}\right)^{n-1} - \left(2 - \frac{v}{v_{0}}\right)^{m-1}\right].$$

The energy e_0 follows from the sixfold coordination number of the triangular lattice, for which the static stress-free volume per particle (corresponding to the energy minimum) is $\sqrt{3/4}$. The pressure P_0 for the cold lattice follows by differentiation: $P_0 = -de_0/dv$. The cold lattice becomes mechanically unstable when dP_0/dV reaches zero :

$$\frac{v}{v_0} = 2 - \left(\frac{m-1}{n-1}\right)^{\frac{1}{n-m}}$$

SPAM: Sample Applications to Solids

To describe the penetration problem with continuum mechanics it is necessary to specify in addition "failure criteria" for the plate material, conditions under which new surfaces can form, allowing the normal stress at the surface to vanish. The simplest failure criteria correspond to a critical tensile stress or to a critical density. If the material has a "yield strength", a shear stress at which the strain first becomes irreversible, corresponding to "plastic" rather than "elastic" behavior then the plastic strain can also be used in the definition of a failure criterion. The plastic strain corresponds to the inelastic work of deformation per unit volume, divided by the yield strength. We use Dyna3d and ParaDyn to model the continuum penetration problem, specifying both the tensile stress and the plastic strain corresponding to failure.

Figure 9.7 displays four sample snapshots from three such finite-element penetration problems. The "ball" is treated as a rigid body of unit density. The plate has the same density, but behaves as an elastic material at stresses below the yield strength Y, which is arbitrarily set equal to one percent of Young's Modulus, Y = E/100 = 0.025. The Lamé constants are equal to unity. "Failure" of the plate elements in the simulations occurs when the tensile stress reaches 0.1 or when the plastic strain reaches 1.0 (bottom three snapshots in the Figure) or 2.0 (top snapshot).

At the lower of the two velocities shown in the Figure, 0.25 , the ball's initial kinetic energy is insufficient to penetrate the plate. The corresponding snapshots in the Figure approximate the final deformed sturcture of the plate. The two snapshots at the higher velocity are at considerably earlier times. In these latter cases there is sufficient energy in the ball to perform the plastic work necessary for complete penetration of the plate.



Penetration via Continuum Mechanics

Figure 9.7: The two snapshots at the top correspond to "low-velocity" failed penetration, but with permanent deformation of the plate. The snapshots at the bottom illustrate two successive stages in the complete penetration and failure of the plate induced by a ball velocity twice as great (0.50). The plane-strain plate bulk modulus is 2. The ball is rigid. No motion is allowed perpendicular to the plane of the problem.

SPAM: Sample Applications to Solids

In the following three Sections we simulate the penetration problem with three methods : molecular dynamics, both conventional and SPAM-like, and with smooth-particle mechanics. The smooth-particle simulation uses the simplest possible failure criterion, a critical tensile stress. Molecular dynamics is the simplest of these approaches, in that it requires no special failure criterion. In molecular dynamics the particles can separate naturally.

9.11 Penetration via Standard Molecular Dynamics

To simulate the penetration of a rectangular plate we choose a "ball" with the same mass density as the plate ($\rho = \sqrt{4/3}$). Within the plate we use the bulk pair potential $\phi_{4,8}$. The ball interacts with plate particles through an extended version of the repulsive pair core potential :

$$\phi_{\rm pp}(r < \sigma) \equiv \phi_{\rm bp}(r - R < \sigma) = 100 \left[1 - (\delta r/\sigma)^2\right]^4$$
;

$$\delta r \equiv r - R < \sigma \; ,$$

where R is the ball radius. This radius is related to the mass of the ball through the definition :

$$M = \pi R^2 \rho = \sqrt{\frac{4}{3}} \pi R^2 \, ,$$

with the ball diameter equal to the plate thickness. Figure 9.8 shows four snapshots of the penetration process according to molecular dynamics.

At relatively low speeds, where the kinetic energy density of the ball is still small relative to the binding energy density of the plate, the ball simply bounces (v = 1 in the Figure). When the ball energy density becomes comparable to that of the plate (v = 2 in the Figure) the plate cracks. At still higher velocities (v = 4 and v = 6 in the Figure) the debris formed during complete penetration becomes increasingly complex. The following two Sections consider this same problem type from the standpoints of SPAM-like molecular dynamics and SPAM.





Figure 9.8: Snapshots, at corresponding times ($t=4\Delta y/v_{\rm ball}$), after the initial ball-plate contact, for four initial velocities : $v_{\rm ball}=\{~1,2,4,8~\}$. The simulation uses standard molecular dynamics, with the pair potential $\phi_{4,8}$ and without any viscous damping. Particle masses in the plate (initially 12 rows of 120 particles each, arranged in the stress-free triangular-lattice structure) are all unity and the Runge-Kutta timestep is 0.0025.

9.12 Penetration via SPAM-like Molecular Dynamics

Using "SPAM-like" molecular dynamics rather than SPAM avoids the need for computing stresses from stress rates, with the associated conceptual problems discussed in Chapter 8. Instead, the curvature potential is used to provide strength. The dynamics proceeds from the total potential :

$$\Phi = \Phi_{\rm EA} + \Phi_{\rm surf} + \Phi_{\rm curv} + \Phi_{\rm core} \ . \label{eq:phi}$$

Here we make the choices :

$$\Phi_{\rm EA} = \frac{1}{2} \sum (\rho_i - \rho_0)^2 \; ; \; \rho_0 = m \sum w_{\rm smooth} (\; r < h = 2.30 \;) \; ;$$

$$\Phi_{\rm surf} = \frac{1}{20} \sum_i \frac{1}{2} (\nabla \rho)_i^2 \; ; \; \Phi_{\rm curv} = \sum \left[\frac{1}{2} (\rho_{xx} - \rho_{yy})^2 + 2\rho_{xy}^2 \right] \; :$$

August 16, 2006 17:44

SPAM: Sample Applications to Solids

$$\Phi_{\rm core} = \sum_{i < j} 100 \left[1 - \left(\frac{r^2}{\sigma^2} \right) \right]^4 \ ; \ \sigma^2 = 0.64 \ ,$$

with the ball-plate interaction computed exactly as before, with standard molecular dynamics.

Figure 9.9 displays snapshots taken with four velocities. Just as before the low-velocity ball bounces, after imparting plastic strain to the work piece, while the high-velocity ball, with an energy density considerably greater than that of the workpiece, produces fragmentation and cratering. A host of interesting simulations can be generated using the basic model outlined here.



9.13 Penetration via SPAM

Just as in the preceding molecular dynamics simulations particles of unit mass are used, with a continuum equation of state related to the (m, n) model described in Section 9.10 with a stress-free density of unity. The

Penetration via SPAM

range of Lucy's weight function,

$$w(r) = \frac{5}{9\pi} (1+r)(1-\frac{r}{3})^3 \longrightarrow \int_0^3 2\pi w(r) dr \equiv 1 ,$$

is as usual 3, three times the nearest-neighbor spacing in a stress-free square lattice structure with unit density. We assume that the solid can be described by constant shear and bulk moduli, G and B respectively, which we choose arbitrarily :

$$G = \eta = \frac{B}{2} = \frac{\lambda + \eta}{2} = \sqrt{24} \; .$$

We consider two different nonlinear failure modes for the material. *Plastic yield* occurs whenever the "shear stress" exceeds the phenomenological "yield strength" Y. A measure of shear stress which is invariant to coordinate rotations is :

$$\sigma_{\rm shear} \equiv [\ \sigma_{xy}^2 + \frac{1}{4}(\sigma_{xx} - \sigma_{yy})^2 \]^{\frac{1}{2}} ,$$

so that plastic flow begins whenever σ_{shear} reaches Y. In the prototypical yield-strength measurement, where the only nonvanishing stress component is σ_{xx} , it should be noticed that yielding first occurs when σ_{xx} is *two times* Y, not just Y. This simplest plastic-flow model is implemented in SPAM by reducing the shear stress to Y after any timestep in which Y is exceeded. First the necessary correction factor is computed :

$$f = Y / \sigma_{\text{shear}}$$
.

Then, the shear stress components are corrected :

$$\sigma_{xy} \longrightarrow f \sigma_{xy} ;$$

$$\sigma_{xx} \longrightarrow \frac{1}{2} (\sigma_{xx} + \sigma_{yy}) + \frac{f}{2} (\sigma_{xx} - \sigma_{yy}) ;$$

$$\sigma_{yy} \longrightarrow \frac{1}{2} (\sigma_{xx} + \sigma_{yy}) - \frac{f}{2} (\sigma_{xx} - \sigma_{yy}) .$$

Tensile failure responds to a critical value of the mean stress rather than to shear stress. Tensile failure occurs whenever the mean stress exceeds the "tensile strength" σ_{tensile} :

$$\frac{1}{2}(\sigma_{xx} + \sigma_{yy}) > \sigma_{\text{tensile}}$$
 .

Such a tensile stress could be relieved in a variety of ways, with or without residual tensile or shear components. The simplest choice, which we adopt



SPAM: Sample Applications to Solids

A Research Suggestion

here, is to set the entire stress tensor equal to zero on failure, with the density returning to the stress-free value of $\sqrt{4/3}$:

$$\sigma_{xx} \longrightarrow 0 ; \ \sigma_{xy} \longrightarrow 0 ; \ \sigma_{yy} \longrightarrow 0 ;$$

$$\rho \longrightarrow \sqrt{4/3}$$
.

Figure 9.10 shows snapshots from a typical simulation with a plate made up of $96 \times 24 = 2304$ particles. A simulation using Monaghan's velocity averaging provides very similar results, but without the obvious symmetrybreaking apparent in the last snapshot.

9.14 A Research Suggestion

The tension test and the ball-plate problem are searching tests for continuum methods. The chosen constitutive relations must include failure models. The boundary conditions (including the applied loads) are crucial, and the evaluation of the results (stresses, plastic strains, damage, fragment characterization) and their size dependence and stability to small changes are all excellent research problems, well-suited to simulations on desktop or large-scale parallel computers.

For about 30 years several advocates of particle simulation techniques, both microscopic and macroscopic, have suggested the possibility of understanding material behavior through intercomparisons of simulations covering the range of microscopic, mesoscopic, and macroscopic scales. So far these investigations have been clouded and frustrated by the perceived need to correlate results with "real data" from experiments with glasses, plastics, metals, and composite materials. A better approach, from the standpoint of understanding, is the intercomparison of simulation techniques for idealized materials in simple situations. I believe that understanding and correlating the problems sketched in this Chapter, first in two dimensions, then in three, and then only later confronting real experimental data, provides the most useful and reliable path to understanding and controlling the behavior of materials in motion.
SPAM: Sample Applications to Solids

9.15 References

- "Auxetic and Related Systems", Physica Status Solidi 242b, March 2005 (K. W. Wojciechowski, A. Alderson, A. Brańka, and K. L. Anderson, editors).
- (2) Wm. G. Hoover, and S. Hess, "Equilibrium and Nonequilibrium Thermomechanics for an Effective Pair Potential used in Smooth Particle Applied Mechanics", Physica A 231, 425-438 (1996).

Chapter 10

Summary, Literature, and Outlook

/ Introduction / Current State of the Art / Cutting and Machining / Ship Stability / Sea Ice Motion / Astrophysics / The Near Future of Parallel Computing / Afterword /

/ 6 Figures /

10.1 Introduction

The complexity of modern computer codes today, together with rapid developments in both software and hardware, practically guarantee the codes' obsolescence tomorrow, when their original developers have gone. The Livermore Laboratory's Dyna3d and ParaDyn codes, for example, are hundreds of thousands of lines long, with complex physical models. The complexity makes the computation of total momentum and energy onerous, so that these important diagnostics are missing in the current software. Because the state-of-the-art software is growing constantly more complicated, assessing the state of the art is very much an attempt to track a moving target.

SPAM has always been an excellent computational technique for following flows of liquids or gases under positive pressure. The treatment of solids or fluids under tension, particularly for long times, requires special techniques. Explicit surface tension, velocity averaging, and density-gradient and invariant-curvature potentials are examples of such techniques mentioned in this book. Solids present an additional difficulty: angular momentum tends to dissipate artificially. Perhaps a technique assigning individual

Summary, Literature, and Outlook

particle angular momenta (so that the "particles" actually spin, to acquire an angular momenta commensurate with their size and rotation rate) to individual particles would be useful.

In much published research (see the examples cited in this Chapter, all of which were found on the internet in 2006) smooth-particle techniques are referred to as "sph" (for Smooth Particle Hydrodynamics). I have consistently avoided this term because it suggests *fluids*, specifically water, in motion, rather than a general approach to solving the partial differential equations of continuum mechanics. SPAM, with its strong links to the internet and the frying pan, seems a pleasant acronym.

10.2 Current State of the Art

There are a variety of interesting ideas which have been developed to solve particular types of smooth-particle problems. Number density, as opposed to mass density, can be taken as fundamental, resulting in a different form for the continuity equation for simulations with different particle masses. The weight functions can be ellipsoidal, rather than spherical, with principal axes which vary in length and in direction as a flow develops.

Smooth weight functions can be abandoned. Weight functions with definite volumes, based on Vornoi polyhedra can be used instead. Figure 10.1 illustrates the two-dimensional case with a representative Voronoi polygon. It is also possible, at considerable computational expense, to introduce an underlying Eulerian interpolation grid and to use interpolated grid values to ensure that linear or quadratic field-variable coordinate dependences are reproduced exactly. Such modifications of the simpler smooth-particle schemes, though complex, are sometimes worthwhile, often interesting, and invariably suggest entirely new research areas in their implementations.

There are many additional topics which can be treated effectively with SPAM. Chemical reactions, particularly shockwaves coupled with explosions, are examples. Magnetohydrodynamics and plasma physics, like astrophysics, require the treatment of longrange forces. This too has been done. Military applications of SPAM abound. There are many interesting and significant penetration simulations involving munitions, tanks, submarines, and aircraft.

In writing this book I make no particular claim to novelty. I have for the most part avoided detailed uncritical references to the voluminous literature in favor of just a few references, mostly to my own work but

Cutting and Machining

including others which I have found to be at least thought-provoking and often useful. Nevertheless, my debt to other writers is substantial. To illustrate only a small portion of it, I describe in what follows a handful of simulations built on stimulating and appealing ideas.



10.3 Cutting and Machining

At a relatively small *mesoscopic* length scale, 10^{-6} to 10^{-2} meters, a variety of metal, plastic, and ceramic failure problems can be solved. Precision diamond turning, with the goal of very flat surfaces, is expensive and timeconsuming. Detailed rheological models of the cutting process can suggest optimum depth and cutting rates as functions of the orientation of the

Summary, Literature, and Outlook

cutting tool relative to the workpiece. Figure 10.2^1 shows two simple twodimensional simulations of the cutting of an aluminum alloy.



Cutting metals or ceramics is a relatively simple problem in molecular dynamics but difficult in continuum mechanics. Cutting requires a model for material failure as well as an algorithm for the disposition of failed material. Such a model can be based on stress, or strain, or energy, or even combinations of these variables. In Heinstein and Segalman's simulations there is a systematic study of chip structure as a function of rake angle, depth of cut, plastic strain at failure, and the number of particles used.

10.4 Structural Response to Waves

The stability of ships' hulls to waves and wave trains is a complex high stakes subject. Smooth-particle models, on a down-to-earth scale of 10 to 100 meters, can easily be brought to bear on this class of problems. In both Heinstein and Segalman's cutting-tool problem and the frigate-stability problem illustrated here, it is convenient to treat the essentially rigid-body part of the problem with finite elements. The cutting tool in Figure 10.2 and the frigate in Figure 10.3 are both described in this way. The flowing materials in the two cases (the aluminum and the water), with their new surface formation and extreme deformation, are much more effectively treated with smooth particles. Thus the stability of ships to

¹Heinstein and Segalman (1997).

Dynamics of Sea Ice

large swells and unusual waves can be successfully attacked by coupling a smooth-particle simulation of water to a finite-element model floating on the water. Cartwright and his coworkers² studied the interaction of waves with realistic ship models using smooth particles. See Figure 10.3 .



Figure 10.3: Cartwright, Xia, Cannon, McGuckin, and Grönenboom's simulation of frigate dynamics in 3-meter seas with a 110-meter wavelength. About 18,000 smooth particles were used. The ship is modeled with finite elements and the sea with smooth particles.

10.5 Dynamics of Sea Ice

Going from wet to a combination of wet and dry water (ice), makes it possible to measure the growth, decay, and movement of large bodies of ice, including its mechanical and thermal responses to the underlying water and the overlying air. By a judicious choice of length scale (100km) it is possible

²Cartwright, Xia, Cannon, McGuckin, and Grönenboom (2006).

Summary, Literature, and Outlook

to model the entire Arctic Ocean with a few thousand smooth particles. Lindsay and Stern's simulation is an interesting hybrid calculation. They use an Eulerian grid which includes a particle-based calculation of strain rates (giving rise to displacements). The model can predict the movement of ice in the Arctic Ocean for periods ranging from days to years. A sample (ten-day) displacement field³ is shown in Figure 10.4.



³Lindsay and Stern (2004).

A strophysics

10.6 Astrophysics

Though my own taste is terrestrial, as in the three example problems just illustrated, rather than astrophysical, we must remember that smoothparticle simulations had their origin at Cambridge University where Gingold, Lucy,⁴ and Monaghan⁵ all worked together to understand interesting features of the night sky. It is no accident that Monaghan's more recent review, 15 years after the discovery of the method, appears in the (1992) Annual Review of Astronomy and Astrophysics. The Astrophysical Journal, Astronomy and Astrophysics, and Astrophysical Letters are all useful sources for such work.

The merging and breakup of stellar objects is fascinating and difficult to simulate, just as is the merging and breakup of liquid drops. These problems are well-suited to simulation with SPAM. Figure 10.5^6 shows six stages in the merger of two neutron stars using relativistic sph with 10^5 particles. It is noteworthy that this work was based on two independent codes, an important, but seldom seen, precaution for complex simulations.

In 1979 Boss and Bodenheimer carried out a prototypical simulation of the gravitational collapse of an initially uniform, but rotating, gas cloud⁷. At a critical density the collapsing cloud heats up, evolves into a binary structure with a bar of material connecting the two components. Simulation of the collapse and division has been carried out by at least *four* sets of workers, again providing the independent checks so necessary in complex computational research work.

Kitsionas and Whitworth⁸ consider dynamic rezoning (by splitting particles as needed) of a gravitational collapse problem. See Figure 10.6 . In their work selected particles are replaced by clusters of 13 smaller particles, using the geometric arrangement of a close-packed lattice. Such a particle-splitting simulation, which begins with 45,000 particles and ends with 140,000, is considerably faster and more efficient than a conventional simulation with 600,000 particles throughout.

⁴Lucy (1977).

 $^{^{5}}$ Gingold and Monaghan (1977).

⁶Faber and Rasio (2000).

⁷Boss and Bodenheimer (1979).

 $^{^8 \}rm Kitsionas$ and Whitworth (2002).



Summary, Literature, and Outlook

The Near Future of Parallel Computing



Figure 10.6: Kitsionas and Whitworth's smooth particle simulations of a self-gravitating collapse model due to Boss and Bodenheimer. The density profile shown at the left, with a logarithmic greyscale, is from a conventional simulation with 600,000 smooth particles. A simulation which began with 45,000 particles and ended up with 140,000 particles through particle splitting provides the nearly identical density profile shown at the right.

10.7 The Near Future of Parallel Computing

A reasonable estimate for the effort to develop a parallel version of a SPAM algorithm from an existing program with a cell structure in place is about one to three months. The experience of the developer with parallel computers and software is the determining factor in such a project.

There is a trend toward using object-oriented syntax, either in Fortran or C, for large programs (greater than 50,000 lines). The disadvantage in using object-oriented syntax for developing research programs is the increased complexity of the syntax and the considerable effort needed to use it. A more profitable way to spend an equivalent amount of research time is to develop parallel programs that make it possible to treat larger problem sizes. The availability and low cost of cluster computers make this an attractive alternative.

Parallel computers currently (2006) are routinely achieving teraflop speeds in many fields: fluid and solid mechanics, electromagnetics, and materials science, using most of the processors on the existing generation

Summary, Literature, and Outlook

of parallel computers. The next target speed threshold is the petaflop computer, which is certainly achievable in the next three to five years!

The parallel software planned for the next generation hardware is being designed in either one of two ways. The existing parallel programs are being expanded to tightly couple more physics modules within a single program. Tight coupling means that the exchange of data from one physical model to another occurs in the same computer program. The other design technique is to couple two highly optimized parallel computer programs together by running the two on subsets of the full processor set used for the simulation. Data on the interfaces between the physical models used in the programs is exchanged between the processor sets assigned to each program. Time step matching becomes an interesting subject for research in either case.

The future remains bright for expanded computational capabilities making larger practical applications possible and opening up new areas for research.

10.8 An Afterword

The few examples cited here, along with the tension and penetration problems elaborated in Chapter 9, should suggest to the reader the broad field of problems to which smooth-particle techiques can readily be applied. The faster and more powerful parallel machines point to a bright future for larger and more richly detailed research projects. I urge the reader to blaze a trail in the vast unexplored territory that is unfolding before us. To the extent that this book saves the reader time and effort in his quest, while stimulating him to formulate, solve, and document interesting smooth-particle problems for others, the book has served its purpose.

287

References

10.9 References

- M. Heinstein and D. Segalman, "Simulation of Orthogonal Cutting with Smooth Particle Hydrodynamics", Sandia Report SAND97-1961 (1997).
- (2) R. Cartwright, J. Xia, S. Cannon, D. McGuckin, and P. Grönenboom, "Motion Prediction of Ships and Yachts by Smoothed Particle Hydrodyamics", Proceedings of the Second High Performance Yacht Design Conference (Auckland, New Zealand, 2006).
- (3) R. W. Lindsay and H. L. Stern, "A New Lagrangian Model of Arctic Sea Ice", Journal of Physical Oceanography 34, 272-283 (2004).
- (4) L. B. Lucy, "A Numerical Approach to the Testing of the Fission Hypothesis", The Astronomical Journal **82**, 1013-1024 (1977).
- (5) R. A. Gingold and J. J. Monaghan, "Smoothed Particle Hydrodynamics: Theory and Application to Nonspherical Stars", Monthly Notices of the Royal Astronomical Society 181, 375-389 (1977).
- (6) J. A. Faber and F. A. Rasio, "Post-Newtonian SPH Calculation of Binary Neutron Star Coalescence: Method and First Results", Physical Review D, 62, 064012 (2000).
- (7) A. P. Boss and L. P. Bodenheimer, "Fragmentation in a Rotating Protostar. A Comparison of Two 3D Computer Codes", Astrophysical Journal 234, 289-295 (1979).
- (8) S. Kitsionas and A. P. Whitworth, "Smoothed Particle Hydrodynamics with Particle Splitting, Applied to Self-Gravitating Collapse", Monthly Noices of the Royal Astronomical Society 330, 129-136 (2002).

August 16, 2006 17:44

WSPC/Book Trim Size for 9 in x 6 in ws-book9x6

288

Summary, Literature, and Outlook

- T. J. Ahrens, T. Takata, J. D. O'Keefe, and G. S. Orton, "Impact of Comet Shoemaker-Levy 9 on Jupiter", Geophysical Research Letters 21, 1087-1090 (1994).
- B. J. Alder and T. E. Wainwright, "Molecular Dynamics by Electronic Computers", pages 97-131 in *Transport Processes in Statistical Mechanics*, I. Prigogine, Editor (Interscience, New York, 1958).
- (3) W. T. Ashurst and W. G. Hoover, "Dense-Fluid Shear Viscosity via Nonequilibrium Molecular Dynamics", Physical Review A 11, 658-678 (1975).
- (4) T. Belytschko, W. K. Liu, and B. Moran, Nonlinear Finite Element Analysis for Continua and Structures (Wiley, New York, 2000).
- (5) C. K. Birdsall and A. B. Langdon, *Plasma Physics via Computer Simulation* (McGraw-Hill, New York, 1985).
- (6) M. Born and K. Huang, Dynamical Theory of Crystal Lattices (Clarendon, Oxford, 1985).
- (7) A. P. Boss and L. P. Bodenheimer, "Fragmentation in a Rotating Protostar. A Comparison of Two 3D Computer Codes", Astrophysical Journal 234, 289-295 (1979).
- (8) R. Cartwright, J. Xia, S. Cannon, D. McGuckin, and P. Grönenboom, "Motion Prediction of Ships and Yachts by Smoothed Particle Hydrodyamics", Proceedings of the Second High Performance Yacht Design Conference (Auckland, New Zealand, 2006).
- (9) V. M. Castillo, Wm. G. Hoover, and C. G. Hoover, "Coexisting Attractors in Compressible Rayleigh-Bénard Flow", Physical Review E 55, 5546-5550 (1997).
- (10) V. M. Castillo and Wm. G. Hoover, "Entropy Production and Lyapunov Instability at the Onset of Turbulent Convection", Physical Review E 58, 7350-7354 (1998).
- (11) C. Cercignani, Ludwig Boltzmann, the Man Who Trusted Atoms (Oxford University Press, 1998).
- (12) S. J. Chapman, Fortran 90 and 95 for Scientists and Engineers (McGraw Hill, 1997).

- (13) S. Chapman and T. G. Cowling, *The Mathematical Theory of Nonuniform Gases* (Cambridge University Press, 1952).
- (14) J. A. Faber and F. A. Rasio, "Post-Newtonian SPH Calculation of Binary Neutron Star Coalescence: Method and First Results", Physical Review D, 62, 064012 (2000).
- (15) S. M. Foiles, M. I. Baskes, and M. S. Daw, "Embedded Atom Method Functions for the FCC Metals Cu, Ag, Au, Ni, Pd, Pt, and their Alloys", Physical Review B 33, 7983-7991 (1986).
- (16) J. W. Gibbs, *Elementary Principles in Statistical Mechanics* (Oxbow Press, 1991) [first published in 1902].
- (17) J. B. Gibson, A. N. Goland, M. Milgram, and G. H. Vineyard, "Dynamics of Radiation Damage", Physical Review **120**, 1229-1253 (1960).
- (18) R. A. Gingold and J. J. Monaghan, "Smoothed Particle Hydrodynamics: Theory and Application to Nonspherical Stars", Monthly Notices of the Royal Astronomical Society 181, 375-389 (1977).
- (19) J. Gleick, Chaos, Making a New Science (Viking, New York, 1989).
- (20) J. L. Gross and J. Yellen, *Graph Theory and its Applications* (CRC Press LLC, Second Edition, 2005).
- (21) M. Heinstein and D. Segalman, "Simulation of Orthogonal Cutting with Smooth Particle Hydrodynamics", Sandia Report SAND97-1961 (1997).
- (22) B. L. Holian, W. G. Hoover, B. Moran, and G. K. Straub, "Shockwave Structure via Nonequilibrium Molecular Dynamics and Navier-Stokes Continuum Mechanics", Physical Review A 22, 2798-2808 (1987).
- (23) C. G. Hoover, A. J. DeGroot, and R. J. Sherwood, "ParaDyn: a Parallel Nonlinear Explicit Three-Dimensional Finite-Element Code for Solid and Structural Mechanics", Report UCRL-MA-140943 (Lawrence Livermore National Laboratory, 2004). ParaDyn is the export controlled parallel version of Dyna3d. For a description of the serial code, which is available to everyone, see J. L. Lin, "Dyna3d, A Nonlinear, Explicit, Three-dimensional Finite Element Code for Solid and Structural Mechanics" (Lawrence Livermore National Laboratory, 2004).
- (24) W. G. Hoover and M. Ross, "Statistical Theories of Melting", Contemporary Physics 12 339-356 (1971).
- (25) W. G. Hoover, W. T. Ashurst, and R. J. Olness, "Two-Dimensional Computer Studies of Crystal Stability and Fluid Viscosity", Journal of Chemical Physics 60, 4043-4047 (1974).
- (26) W. G. Hoover, C. G. Tull (now C. G. Hoover), and H. A. Posch, "Negative Lyapunov Exponents for Dissipative Systems", Physics Letters A 131, 211-215 (1988).
- (27) Wm. G. Hoover and H. A. Posch, "Shear Viscosity via Local Control of Spatiotemporal Chaos in Two-Dimensional Isoenergetic Dense Fluids", Physical Review E 51, 273-279 (1995).
- (28) Wm. G. Hoover and S. Hess, "Equilibrium and Nonequilibrium Thermomechanics for an Effective Pair Potential used in Smooth Particle Applied Mechanics", Physica A 231 425-438 (1996).

- (29) Wm. G. Hoover and H. A. Posch, "Numerical Heat Conductivity in Smooth Particle Applied Mechanics", Physical Review E 54, 5142-5145 (1996).
- (30) Wm. G. Hoover, "Mécanique de Nonéquilibre à la Californienne", Physica A 240, 1-11 (1997).
- (31) Wm. G. Hoover, "Isomorphism Linking Smooth Particles and Embedded Atoms", Physica A 260, 244-254 (1998).
- (32) Wm. G. Hoover, "Liouville's Theorems, Gibbs' Entropy, and Multifractal Distributions for Nonequilibrium Steady States", Journal of Chemical Physics 109, 4164-4170 (1998).
- (33) Wm. G. Hoover, Computational Statistical Mechanics (Elsevier, New York, 1991), available at http://williamhoover.info. [Keisan Toukei Rikigaku, Japanese translation by Koichiro Shida with the supervision of Susumu Kotake (Morikita Shupan, 1999) ISBN: 4627840918].
- (34) Wm. G. Hoover and H. A. Posch, "Entropy Increase in Confined Free Expansions via Molecular Dynamics and Smooth Particle Applied Mechanics", Physical Review E 59, 1770-1776 (1999).
- (35) Wm. G. Hoover, H. A. Posch, V. M. Castillo, and C. G. Hoover, "Computer Simulation of Irreversible Expansions via Molecular Dynamics, Smooth Particle Applied Mechanics, Eulerian, and Lagrangian Continuum Mechanics", Journal of Statistical Physics 100, 313-326 (2000).
- (36) Wm. G. Hoover, *Time Reversibility, Computer Simulation, and Chaos* (World Scientific, Singapore, 2001) [*Jikan no ya, Kompyutah Shimyureishon, Kaosu*, Japanese translation by Koichiro Shida (Morikita Shuppan, 2002) ISBN: 4627153015].
- (37) Wm. G. Hoover, C. G. Hoover, and E. C. Merritt, "Smooth Particle Applied Mechanics: Conservation of Angular Momentum with Tensile Stability and Velocity Averaging", Physical Review E **69**, 016702 (2004).
- (38) Wm. G. Hoover and C. G. Hoover, "Searching for Auxetics with Dyna3d and ParaDyn", Physica Status Solidi 242b, 585-594 (2005).
- (39) Wm. G. Hoover and C. G. Hoover, "Smooth Particle Phase Stability with Generalized Density-Dependent Potentials", Physical Review E 73, 016702 (2006).
- (40) B. Kernighan and D. Ritchie, *The C Programming Language* (Prentice Hall, 1988).
- (41) S. Kitsionas and A. P. Whitworth, "Smoothed Particle Hydrodynamics with Particle Splitting, Applied to Self-Gravitating Collapse", Monthly Noices of the Royal Astronomical Society 330, 129-136 (2002).
- (42) S. Kochan, Programming in C (Sams, Prentice-Hall, Indianapolis, 2004).
- (43) O. Kum and W. G. Hoover, "Time-Reversible Continuum Mechanics", Journal of Statistical Physics 76, 1075-1081 (1994).
- (44) O. Kum, Wm. G. Hoover, and H. A. Posch, "Viscous Conducting Flows with Smooth Particle Applied Mechanics", Physical Review E 52, 4899-4908 (1995).
- (45) O. Kum, Wm. G. Hoover, and C. G. Hoover, "Temperature Maxima in Stable Two-Dimensional Shockwaves". Physical Review E 56, 462-465 (1997).

- (46) O. Kum, W. G. Hoover, and C. G. Hoover, "Smooth Particle Boundary Conditions", Physical Review E 68, 017701 (2003).
- (47) D. Levesque and L. Verlet, "Molecular Dynamics and Time Reversibility", Journal of Statistical Physics 72, 519-537 (1993).
- (48) ParaDyn is an export controlled parallel version of Dyna3d. For a description of the serial code, which is available to everyone, see J. L. Lin, "Dyna3d, A Nonlinear, Explicit, Three-dimensional Finite Element Code for Solid and Structural Mechanics" (LLNL, Livermore, California, February 2004).
- (49) R. W. Lindsay and H. L. Stern, "A New Lagrangian Model of Arctic Sea Ice", Journal of Physical Oceanography 34, 272-283 (2004).
- (50) E. N. Lorenz, "Deterministic Nonperiodic Flow", Journal of Atmospheric Science 20, 130-141 (1963).
- (51) For an interesting variant representation of this classic method see page 186 of Lorenz' otherwise fine book, *The Essence of Chaos* (University of Washington Press, 1993).
- (52) L. B. Lucy, "A Numerical Approach to the Testing of the Fission Hypothesis", The Astronomical Journal 82, 1013-1024 (1977).
- (53) J. Mayer and M. G. Mayer, Statistical Mechanics (Wiley, New York, 1977).
- (54) D. D. McCracken and W. Salmon, Computing for Engineers & Scientists with Fortran 77 (Wiley, 1990).
- (55) J. J. Monaghan, "On the Problem of Penetration in Particle Methods", Journal of Computational Physics 82, 1-15 (1989).
- (56) J. J. Monaghan, "Smooth Particle Hydrodynamics without a Tensile Instability", Journal of Computational Physics 159, 290-311 (2000).
- (57) B. Moran, W. G. Hoover, and S. Bestiale, "Diffusion in a Periodic Lorentz Gas", Journal of Statistical Physics 48, 709-726 (1987).
- (58) S. Nosé, "Constant Temperature Molecular Dynamics Methods", Progress in Theoretical Physics Supplement 103, 1-117 (1991).
- (59) H. A. Posch, W. G. Hoover, and F. J. Vesely, "Canonical Dynamics of the Nosé Oscillator: Stability, Order, and Chaos", Physical Review A 33, 4253-4265 (1986).
- (60) H. A. Posch and W. G. Hoover, "Equilibrium and Nonequilibrium Lyapunov Spectra for Dense Fluids", Physical Review A 39, 2175-2188 (1989).
- (61) H. A. Posch, W. G. Hoover, and O. Kum, "Steady State Shear Flows via Nonequilibrium Molecular Dynamics and Smooth Particle Applied Mechanics", Physical Review E 52, 1711-1720 (1995).
- (62) W. Prager, Introduction to the Mechanics of Continua (Ginn & Company, Boston, 1961 [Dover reprint, 1973]).
- (63) W. H. Press, B. P. Flannery, S. A. Teukolsky, and W. T. Vetterling, Numerical Recipes, The Art of Scientific Computing (Cambridge University Press, London, 1986).
- (64) R. Preston, "Capturing the Unicorn", New Yorker Magazine, 11 April 2005.
- (65) A. Puhl, M. M. Mansour, and M. Mareschal, "Quantitive Comparison of Molecular Dynamics with Hydrodynamics in Rayleigh-Bénard Convection", Physical Review A 40, 1999-2012 (1989).
- (66) J. W. Swegle, D. L. Hicks, and S. W. Attaway, "Smooth Particle Hydrody-

namics Stability Analysis", Journal of Computational Physics **116**, 123-134 (1995).

- (67) The "Fermi-Pasta-Ulam" problem is reviewed and extended in J. L. Tuck and M. T. Menzel, "The Superperiod of the Nonlinear Weighted String Problem", Advances in Mathematics **9**, 399-407 (1972).
- (68) J. von Neumann and R. Richtmyer, "A Method for the Numerical Calculation of Hydrodynamic Shocks", Journal of Applied Physics 21 232-257 (1950).
- (69) T. Watanabe and H. Kaburaki, "Increase in Chaotic Motions of Atoms in a Large-Scale Self-Organized Motion", Physical Review E 54, 1504-1509 (1996).
- (70) M. L. Wilkins, "Finite Difference Scheme for Calculating Problems in Two Space Dimensions and Time", Journal of Computational Physics 5, 406-414 (1970).
- (71) K. W. Wojciechowski, A. Alderson, A. Brańka, and K. L. Anderson, editors, "Auxetic and Related Systems", Physica Status Solidi **242b**, (March, 2005).
- (72) D. A. Young, *Phase Diagrams of the Elements* (University of California, Berkeley, 1991).
- (73) R. W. Zwanzig, "Time Correlation Functions and Transport Coefficients", Annual Review of Physical Chemistry 16, 67-102 (1965).

August 16, 2006 17:44

WSPC/Book Trim Size for 9in x 6in

ws-book9x6

294

 $Alphabetical \ Bibliography$

Index

accuracy versus precision 180 adiabatic atmospheric equilibrium 91 algorithmic convergence 58, 89, 165, 177, 180, 196 algorithmic instability 179 algorithmic stability 178 algorithms 2, 34 aluminum cutting 279 angular momentum loss 246 arrays 107 artificial heat conductivity 46 artificial plasticity 249 artificial transport coefficients 248 artificial viscosity 35, 46, 59, 89 astrophysics 2, 283 auxetic systems 23, 254

billiard table 140 bit reversible algorithm 30 boundary conditions 40, 148, 257 bugs 125 bulk modulus 50 bulk viscosity 45 butterfly instability 215

C computer language 100 C indexing 102 C type statements 102 cell searches 134 chaotic solutions 1, 5, 41, 178 collapsing column 38, 211 comoving equations 11 comoving time derivative 10 computer graphics 119 computer languages 100 conduction 12 conservation laws 35 constitutive equations 34, 42, 45, 48 continuity equation 11, 80 continuum mechanics 4, 20, 33, 67 continuum penetration 267 convection 12 convergence 177, 180 core potential 92, 149, 207, 228 Courant condition 39 critical point 43 curvature potential 262

debugging 125 deformation work 54 degrees of freedom 9 density gradient surface potentials 212, 229 distributed memory 129 dynamical matrix 186 dynamic partitioning 138 Dyna3d 56, 255, 268

eigenvalue - eigenvector analysis 186 elasticity 50 elastic-plastic solid 56 embedded-atom column collapse 211 embedded-atom fluid 205

296

embedded-atom gravitational relaxation 208 embedded-atom isomorphism 205 embedded-atom potential 148, 228 embedded-atom stability 157 energy absorbing boundaries 145 energy equation 36 entropy 18 entropy production 18 equilibrated columns 146, 210 equilibration 38, 40 Euler fluid 214 Eulerian methods 9, 21, 68 Eulerian Rayleigh-Bénard flow 60 even-odd instability 195 evolution equations 11, 35 existence-uniqueness of solutions 178 failure algorithms 266 falling water 211 feedback 17 finite differences 21, 26, 61 finite-element method 20, 22

First Law of Thermodynamics 13, 36, 221 force balance 91, 94 force-constant matrix 154 Fortran 100 Fourier's Law 21 Fourier Law violation 224 fractals 19 free boundary mesh 147 free expansion 138, 213, 236 friction coefficient 17

Galton Board 19 Gibbs' entropy 214 gnuplot 119 graphic displays 119 graph theory 132 gravitational equilibration 144, 208 Green-Kubo viscosity 86 Griz4 120 Grüneisen theory 44, 224

harmonic chain vibrations 154

Index

harmonic oscillator 12, 113 heat 13 heat conduction 58, 94, 164 heat flux vector 36 heat transfer 164 hourglass instability 215 Hugoniot relation 221

ideal-gas isomorphism 85, 202 include file 107 initial conditions 39, 144, 260 initial mesh 39 instability 186, 189 integrators 6, 28, 112 interponetration 235 interpolation 20, 68 intrinsic viscosity 170 invariant curvature stabilization 162 irregular grid methods 68 isothermal atmospheric equilibrium 94, 96

Jaumann stress rates 240, 244

kinematic viscosity 48 kinetic theory 37

Lagrange multiplier 188 Lagrangian mesh 22, 68 Lagrangian methods 9, 21, 68 Lamé constants 50, 254 lattice instability 157, 192 leapfrog algorithm 26, 28, 196 Lindemann melting law 204 linear strains 49 Liouville's Theorem 16 locked-in stress 48, 249 logarithmic strains 239 Lucy fluid 85, 197 Lucy isomorphism 85, 202 Lucy melting 204 Lucy shockwave 217 Lucy thermodynamics 203 Lucy's weight function 24, 74, 77, 197 Lucy viscosity 87, 170

Index

Lyapunov exponent determination 186Lyapunov instability 7, 178, 186 massively parallel computers 129 material interfaces 136 mechanical equation of state 14, 42, 267mesh generation 147mesh instability 192 mesh partitioning 131 mesh point averages 123 message passing 129, 133 microscopic pressure tensor 15 mirror boundaries 92, 124, 150, 152 mirror forces 95 molecular dynamics 14, 26 molecular dynamics penetration 270 molecular dynamics tension test 256 Monaghan motion equation 233 Monaghan velocity averaging 235 Monaghan's weight function 76, 77 neighbor lists 134 Newtonian mechanics 14, 26, 201, 213 Newtonian viscosity 45 nonequilibrium molecular dynamics 15normal mode analysis 161 Nosé-Hoover mechanics 17 Nosé-Hoover stability 186 numerical convergence 165, 174, 190, 198numerical integration 6, 26, 112 numerical integration error 180 numerical stability 184 object-oriented language 104 offset vector 188 pair core potential 92, 94, 149, 197,

211, 228, 270 ParaDyn 38, 128 parallel computing 128, 285 parallel efficiency 130 parameter statement 107

particle mechanics 3, 4 particle methods 65 particle simulations 3 partitioning 131 penetration mechanics 266 periodic boundary conditions 28, 58, 150periodic shear flow 167, 196 phase diagram 43 phase shift 183 plastic flow 53, 55 plasticity algorithm 55 Poisson's ratio 51, 254 portability 103 precision versus accuracy 180 pressure tensor 11, 15, 36, 67 programming 99 programming style 109 random number generator 117 Rayleigh-Bénard flow 60, 171, 179 Rayleigh-Bénard uniqueness 179 Rayleigh line 220 Rayleigh number 62, 171 relaxation 80, 144, 207, 208, 211, 230, 260research suggestion 275 residual stress 48, 57, 249 rezoning 84, 283 rigid body rotation 236 rotational ambiguity 240, 242 Runge-Kutta error 29, 113, 181, 183, 191 Runge-Kutta method 6, 28, 58, 181 Runge-Kutta program 112 sea ice dynamics 281 self density 86 self energy 86, 202 shared memory 128 shear flow convergence 196 shear modulus 50, 164 shear viscosity 45, 60, 86, 170, 198 sheets and strings 231

shell elements 22, 139

ship stability 280

Index

298

shockwave 47, 130, 217 shockwidth 48 size dependence 190 Smooth Particle Methods 65 smooth weight function 163, 262 sound propagation 59 SPAM 1, 24, 65 SPAM artificial viscosity 89 SPAM averages 71 SPAM contour plots 123 SPAM continuity equation 80 SPAM density 70, 78 SPAM density errors 78 SPAM errors 72, 191 SPAM energy conservation 84, 94 SPAM energy equation 83 SPAM equation of motion 69, 83 SPAM gradients 74, 82, 87 SPAM heat transfer 164 SPAM initial conditions 260 SPAM interpolation 25, 70 SPAM penetration 272 SPAM program 105, 214 SPAM Rayleigh Bénard 171 SPAM relaxation 260 SPAM rezoning 84 SPAM shear flow 167 SPAM spatial errors 191 SPAM tension test 263SPAM velocity gradient tensor 238 SPAM weight functions 74 SPAM-like molecular dynamics 261, 271 $\operatorname{SPAM-like}$ penetration 271 speedup 130

stability 6, 36, 157, 177, 184 statistical mechanics 16 Stokes' fluid 46 strain rate 52 stress rotation 240, 244 stress tensor 45, 50, 60, 237 stress tensor invariants 53 strings and sheets 231 subroutines 105, 126 surface tension 201, 212, 227

temperature 14, 17, 224 tensile boundary conditions 257 tensile instability 231 tensile strength 273 tension test 56, 254 thermal equation of state 44 thermodynamics 13, 42 thermostat forces 16 time reversibility 27, 29 Totalview 125, 127 two-dimensional research 275

van der Waals' fluid 42 variable types 102 viscosity 45 von Mises yielding 52, 56 Voronoi polyhedra 278

weight function 24, 74, 77, 197 work 13

yield strength 53, 268 yield surface 55 Young's modulus 51, 255

Example Problem List

Adiabatic Atmospheric Equilibrium 91 Falling Water with Embedded Atoms 211 Free Expansion with Lucy Fluid 213 Gravitational Relaxation with Embedded Atoms 208 Heat Conduction with Continuum Mechanics 58 Heat Transfer with SPAM 164 Isothermal Atmospheric Equilibrium 94 Molecular Dynamics Simulation 26 Periodic Shear Flow with SPAM 167 Shear Flow Convergence 196 Sound Propagation with Continuum Mechanics 59 Rayleigh-Bénard Flow 60 Rayleigh-Bénard Flow with SPAM 171 Shockwave with Lucy Fluid 217

300

Smooth Particle Applied Mechanics



William Graham Hoover was born in Boston, Massachusetts in 1936. He completed his doctoral work under Andrew De Rocco, in Chemical Physics, at the University of Michigan in 1961. Following a year's postdoctoral work with Jacques Poirier, at Duke University, Professor Hoover was hired as a staff physicist by the Radiation Laboratory in Livermore, California (Now the Lawrence Livermore National Laboratory). In 1975 Professor Hoover (now Emeritus Professor) accepted a joint appointment in the Department of Applied Science, College of Engineering, University of California at Davis/Livermore. He is currently a Participating Guest at LLNL in the Methods Development Group, Department of Mechanical Engineering. Professor Hoover has carried out sabbatical work at the Australian National University, The University of Vienna, The Technical University of Berlin, and Keio University (Yokohama). Professor Hoover is a Fellow of the American Physical Society. His books include "Molecular Dynamics" (Springer-Verlag, 1985), "Computational Statistical Mechanics" (Elsevier, 1991), "Time Reversibility, Computer Simulation, and Chaos" (World Scientific, 1999 and 2001), and "SPAM, Smooth Particle Applied Mechanics" (World Scientific, in preparation). His webpage can be found at http://williamhoover.info.