An Introduction to Hydrocode Modeling

Gareth S. Collins

August 2, 2002

1 Introduction

The hydrocode is a computational tool for modeling the behavior of continuous media. In its purest sense, a hydrocode is a computer code for modeling fluid flow at all speeds. It can, however, be adapted to treat material strength and a range of rheological models for material behaviour. In essence, the code considers the effect of external and internal forces on a predefined "mesh" of cells, which represent the system being studied. It assumes that, over a short period of time, these forces are constant and uses them to adjust the geometry of the mesh accordingly. The forces are the recalculated and the process repeats until the required solution is found.

This booklet has been compiled to equip the reader with enough insight to understand the important details and limitations of using a hydrocode; and to introduce the interested reader to a particularly tractable hydrocode: SALES_2. The first part of the report is a general introduction to hydrocode modeling. It discusses the motivation for modeling of this kind, the philosophy of hydrocode modeling and introduces issues relevant to hydrocode modeling in general, such as discretization, accuracy and stability. Accompanying this introduction is the full documentation for SALES_2; a versatile and tractable hydrocode available for download at:

http://www.lpl.arizona.edu/tekton/sales_2.html

Before discussing the details of hydrocode modeling, it is worth recalling the reasons for such tools. Natural processes range in complexity: some processes, such as the behavior of a mass on a spring, may be described completely by a mathematical equation. Other processes, such as the impact of an asteroid on a planet, are complex enough that an analytical solution is not possible. In such cases, observational methods and numerical techniques, using computers, provide the only amenable ways to approach a solution. If the scale of the problem is beyond the range of laboratory or field experiments, or the conditions for the investigation are too extreme, numerical modeling becomes the only way of studying a particular phenomenon.

Hydrocode modeling offers a powerful way to study natural phenomena: from a certain point of view, it can be considered the best instrumented experiment. However, as a tool, hydrocodes are not without their weaknesses and limitations. Care should be taken when choosing a hydrocode, defining the problem and analyzing the results. A broad, but by no means exhaustive, list of things to bear in mind when considering hydrocode modeling is:

- **Is hydrocode modeling required?** Analytical solutions are by far the most powerful descriptions of a process. If a steady-state solution is required (one that does not depend on time), other, simpler, numerical techniques may offer a faster means for deriving a solution. Hydrocode modeling is most applicable to time-dependent, non-linear problems.

- **What is the important physics?** All numerical modeling techniques work on the premise that things should be made as simple as possible (but not any simpler!). In general, the more straightforward the algorithm for solving a problem, the faster the solution will be resolved. It is important, therefore, to strike a balance between considering the physics important to the solution of a problem, and ignoring aspects of the process that are less, or not, important. For example, the physics of impact cratering is dominated by different fundamental principles during different stages of the process. The early part of the impact is controlled mainly by the thermodynamics of shock-compression, whereas the latter stages are controlled almost exclusively by the competition between gravity and the post-shock strength of the target. Thus, to simulate just crater collapse, it is important to develop a rigorous rheological model for the target; but the thermal effects of the collapse process can be justifiably ignored.

- **What is the expected result?** It is also very helpful to have some idea of the expected solution. For example, how long the problem will need to run for, or how large the problem space needs to be.

- **Is the result testable?** Code verification is an often overlooked aspect of hydrocode modeling. After a solution is found, how can its validity be tested? For problems where no analytic solution exists with which to test the model, there is no short answer to this question. The best advice is to try to find a similar, less complicated problem, where an analytic solution does exist. For example, in the case of complex crater collapse calculations, the simple test problem of the collapse of a hemispherical cavity in a fluid half-space may be used. As will be discussed later, it is also important to ensure that the solution is independent of the resolution used for the problem, in both space and time. If a solution changes its character for different time steps, for example, it is probably being affected by numerical artifacts.

- **Is the problem solvable given the hardware available and time constraints?** This problem is, again, difficult to answer a priori. The hardware available will limit the size of the problem space, or the maximum resolution possible. The amount of time available, and the processor speed will also limit the number of time steps available.

- **What kind of hydrocode is appropriate?** There is no such thing as a universal code. Every hydrocode built, has been written with one subset of problems in mind. It is important to know which type of hydrocode is best suited to the problem in hand. The various kinds of hydrocode, plus their strengths and weaknesses will be discussed later.

2 A Brief Overview of Hydrocode Modeling

The essence of hydrocode modeling is summarized in the flow chart in figure 1.

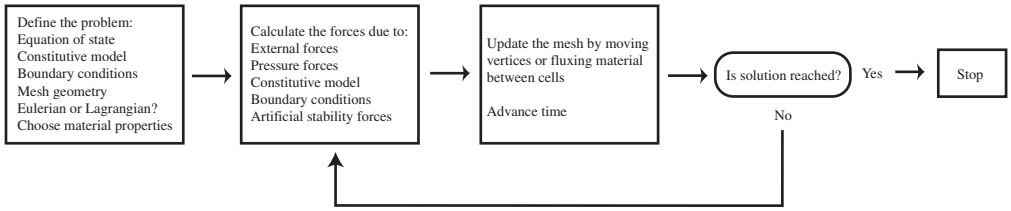

Figure 1: Flow chart summarizing the general scheme of a hydrocode

Hydrocode modeling rests on three pillars, which are used to determine the forces acting on the mesh each time step. These are: the Newtonian laws of motion; the equation of state; and the constitutive model.

The modeling of incompressible, inviscid fluid flow may be described by the Newtonian laws of motion alone, as a set of differential equations established through the principles of conservation of momentum, mass and energy from a macroscopic point of view. These equations are of the form:

$$\text{Conservation of Momentum} \qquad \frac{Dv_i}{Dt} = f_i + \frac{1}{\rho}\frac{\partial \sigma_{ji}}{\partial x_j} \tag{1}$$

$$\text{Conservation of Mass} \qquad \frac{D\rho}{Dt} + \rho\frac{\partial v_i}{\partial x_i} = 0 \tag{2}$$

$$\text{Conservation of Energy} \qquad \frac{DI}{Dt} = -\frac{p}{\rho}\frac{\partial v_i}{\partial x_i} + \frac{1}{\rho}\Pi_{ij}\dot{\epsilon}'_{ij} \tag{3}$$

where ρ is the material density, v_i is the velocity, I is the specific internal energy, σ_{ij} is the stress tensor, which is composed of a hydrostatic part, the pressure p, and a deviatoric part, Π_{ij}. f_i is the external body forces per unit mass, and $\dot{\epsilon}'_{ij}$ is the deviatoric strainrate. The subscripts represent the standard tensorial notation, and summation is implied by repeated indices.

The equation of state relates pressure to the density and internal energy. It thereby accounts for compressibility effects; that is, changes in density and irreversible thermodynamic processes such as shock heating.

$$\text{Equation of State} \qquad p = p(\rho, I) \tag{4}$$

The constitutive model, relates the stress to a combination of strain ϵ_{ij}, strain rate effects $\dot{\epsilon}_{ij}$, internal energy I, and damage D. These describe the effect of deformation (change in shape or strength properties).

$$\text{Constitutive Model} \qquad \sigma_{ij} = g(\epsilon_{ij}, \dot{\epsilon}_{ij}, I, D) \tag{5}$$

Analytical solutions to Equations 1–5 above are only obtainable for circumstances where certain simplifying assumptions may be invoked, reducing the number of variables to be considered (*Anderson*, 1987). In cases of practical interest, where the variables are numerous and the problem is complex, the equations must be solved simultaneously. Numerical techniques, using digital computers, provide the only amenable method to achieve the number of mathematical operations required for the solution.

All hydrocodes utilize some form of the conservation equations; however, the usefulness of the hydrocode depends on the sophistication of the equation of state and constitutive model. A more detailed discussion of these two pillars of hydrocode modeling, including details of their implementation in SALES_2, is discussed in the SALES_2 manual, which accompanies this introduction.

3 The Fundamentals of Hydrocode Modeling

A computer has a finite memory allocation and, therefore, can only represent a continuous media by dividing it into distinct pieces; a process known as discretization. Thus, the approach adopted by a hydrocode is to define a "mesh" to approximate the geometry of interest, which is divided into manageable divisions, known as cells. Scalar quantities within the continuum are assigned to each cell and are constant within the cell. In two dimensions, the cell is usually defined by three or four vertices connected by straight lines to form a triangle or quadrilateral respectively. Vector quantities like velocity are assigned to each vertex. Boundary conditions are also required to control the behavior of the exterior vertices. Mesh generation and boundary conditions are discussed in section 3.1.

The differential equations that describe the kinematic deformation of a continuum can, likewise, only be approximated by operations on a computer. Three fundamental techniques exist for discretizing the differential equations: finite-element schemes, finite-difference schemes and Smooth Particle Hydrodynamic (SPH) techniques. Essentially the three schemes offer different algorithms for solving the same problem; however, each has its benefits and weaknesses, which are discussed in section 3.2.

The conservation equations at the heart of a hydrocode may be written with respect to the material; that is, they describe the motion (displacement, velocity and acceleration) from the reference frame of the continuous media itself. This type of description is known as a Lagrangian (or material) description. The equations may also be written down with respect to a fixed reference frame; a formulation termed an Eulerian (or spatial) description.

A hydrocode may employ either type of formulation to describe the situation of interest. The choice of either mode of description depends on the problem under consideration. The relative merits of these formulations is discussed in section 3.3.

3.1 Mesh generation and boundary conditions

Generating a mesh to represent the geometry of interest, assigning appropriate initial material parameters, and choosing appropriate boundary conditions are the basic inputs for a hydrocode. Certain types of hydrocodes are designed for particular geometries or boundary conditions, again emphasizing the importance of choosing an appropriate hydrocode for a particular problem. 1, 2 and 3D-hydrocodes exist; however, because memory requirements scale with the number of cells, 3D hydrocodes have only recently come into mainstream usage. Frequently, simplifying assumptions are used to reduce the spatial degrees of freedom. For example, assuming a vertical impact allows impact cratering studies to use a 2D cylindrical-coordinate hydrocode in place of a 3D hydrocode. The computing mesh used in SALES_2 consists of a two-dimensional grid of quadrilateral cells for either cylindrical or plane coordinates. Thus, SALES_2 can only solve problems that are inherently two-dimensional, or have a symmetry about an axis.

3.1.1 Mesh Generation

Mesh generation is hydrocode specific (see section 3.2); however, it invariably involves modifying the source code. In SALES_2 the default mesh is a rectangular grid. To modify the form of this mesh, the user must replace the mesh generation algorithm with one of their own. Historically this has been a time-consuming task, particularly for complex problem-specific geometries. However, mesh generation algorithms are now quite sophisticated: a typical approach for a finite-difference hydrocode would be as follows:

1. Generate a rectangular mesh.
2. Redefine, by hand, the vertex positions for the exterior points of the mesh.
3. The interior vertices are then repositioned using an interpolation scheme (for example, *Amsden*, 1973). The interpolation algorithm moves each vertex to the average position of its eight surrounding vertices. The boundary vertices are fixed; they are not moved during the interpolation.
4. After several sweeps over all the vertices the interpolation finishes. The result is a mesh where all cells are (approximately) regular quadrilaterals, for which the finite difference approximations are accurate.

3.1.2 Boundary Conditions

The types of boundary conditions implemented in a hydrocode also vary between specific codes. Common boundary conditions fall into the following categories:

- **Free surface** This is the simplest type of boundary condition, which applies no constraints on the motion of the vertex.
- **Free slip** For a symmetry boundary or a free-slip wall, the normal wall velocities must be kept at zero throughout the calculation. If such a boundary is parallel to the coordinate axes implemented in the hydrocode, this is a simple matter of setting one of the velocity component to zero. If the wall is slanted or curved both velocity components must be adjusted.
- **No slip** For this boundary condition both velocity components are set to zero, regardless of mesh geometry.
- **Specified outflow or inflow** For this type of boundary condition the velocities at the boundary are specified externally. This condition is complicated, however, by the need to set not just velocities but other, cell-centered quantities such as density and internal energy.
- **Continuative outflow or inflow** Similar to the specified flow boundary condition, the typical treatment of such conditions is to set the inflow or outflow velocities, densities, energies, etc. equal to the adjacent cell within the mesh.
- **Forcing** This form of boundary condition applies a stress along or across the boundary. The form of this stress may be constant or time-dependent.

3.2 Discretization

3.2.1 Finite Difference Method

In the finite difference method the spatial derivatives in the differential equations are replaced by difference equations. For example, for some function F the partial derivative $\partial F/\partial x$ becomes $\Delta F/\Delta x$ where the differences are computed at grid points. The first derivative of F at x_n can be represented by a variety of difference formulae:

$$\begin{aligned}
\left.\frac{\partial F}{\partial x}\right|_{x_n} &= \frac{F_{n+1} - F_n}{\Delta x} \\
\left.\frac{\partial F}{\partial x}\right|_{x_n} &= \frac{F_n - F_{n-1}}{\Delta x} \\
\left.\frac{\partial F}{\partial x}\right|_{x_n} &= \frac{F_{n+1} - F_{n-1}}{2(\Delta x)}
\end{aligned} \tag{6}$$

which correspond to forward, backward and central difference equations, respectively. SALES_2 is a finite-difference hydrocode, which uses a central differencing scheme.

The finite-difference method is well-founded and simple to implement. However, it does require (mainly for convenience) that the grid is structured (cells arranged in rows and columns). Consequently, clever coordinate-mapping techniques or adaptive meshing algorithms must be applied in order to solve problems involving complicated geometries. Furthermore, there is no straight-forward way to test the accuracy of a solution, and the scheme is prone to certain types of numerical instability, which require artificial corrections. In general, the accuracy of the solution increases with decreasing cell size; however, limits on the time step mean that small cell sizes imply small time steps, leading to long run times. Accuracy and instability in finite difference algorithms is discussed in more detail later.

3.2.2 Finite Element Method

The finite element method was initially developed on a physical basis for the analysis of problems in structural mechanics; however, it was soon recognized that the method can be applied to a variety of problems (*Bathe and Wilson*, 1976).

Whereas the finite difference method is a pointwise discretization of the problem space, finite element methodology divides the problem space into elements. The elements can be rectilinear or curved and, unlike the finite difference method, need not be arranged in a structured grid. Hence, complicated problem geometries are handled better with a finite element approach.

Interpolation functions (typically a polynomial) are used to represent the variation of a variable over the element. Each element is associated with a set of nodes, whose initial locations are known. The displacement of these nodes are the basic unknowns of the problem. The equations governing the displacements of these nodes are calculated on an element-to-element basis and then combined. A consequence of this fact is that finite element codes may be parallelised as a way to reduce run time.

Once combined, the system of equations relating the forces and displacements at each node is solved by inverting the "stiffness matrix," which represents the constitutive relationship between stress and strain. One advantage of this method is that when the displacements have been derived,

they can be substituted back into the original equations to check for consistency. Any inconsistency is a direct measure for the inaccuracy of the solution and can be corrected for during the simulation.

3.2.3 Smooth Particle Hydrodynamics

Smoothed particle hydrodynamics (SPH) was invented to simulate problems in astrophysics involving fluid masses moving arbitrarily in three dimensions in the absence of boundaries (*Monaghan*, 1988). A typical example is the numerical simulation of the fission of a rapidly rotating star.

SPH involves the motion of a set of points. At any time, the velocity and thermal energy are known at these points. A mass is also assigned to each point and, for this reason, the points are referred to as particles. In order to move the particles correctly during a time step it is necessary to construct forces which an element of fluid would experience. These forces are basically constructed using sophisticated interpolation techniques to determine properties such as density at a given point.

SPH codes offer an attractive alternative to the more well-founded techniques of finite-difference and finite-element, due to the simplicity of the algorithm: most users tend to write their own SPH code. The method is inherently Lagrangian (see below), and therefore, possesses most of the benefits of this formalism; however, SPH does not break down when large displacements are involved, because the particles are not connected.

Although currently in-vogue, and in an ever advancing state of development, SPH codes do suffer from several major short-comings. Currently, there are no robust methods for describing complicated material rheologies such as strength, elasticity, etc. Moreover, by their very nature, SPH codes do not handle certain types of boundary conditions well, further limiting their potential use. Lastly, in problems such as impact calculations where the density varies dramatically (from very dense target rock to low density vapor), SPH suffers because the low density material is represented by too few particles to simulate the problem well.

SPH codes are good for fluid flow problems involving relatively small density differences and primarily inflow or outflow boundary conditions. In particular, they are good for problems involving self-gravity, such as the formation of planets and stars.

3.2.4 Time Integration

Most hydrocodes use explicit time integration to advance the solution in time; that is, functions at the new time step are determined from the known functions at a previous time step. For example, the change in velocity Δu of a certain vertex during one time step dt, due to an external acceleration a, is given as: $\Delta u = adt$. At the next time step the time is advanced, the

old scalar and vertex quantities are replaced by the new ones and the cycle repeats.

3.3 Lagrangian and Eulerian Descriptions

Intuitively, these two alternative descriptions for the behavior of a continuous media may be understood by the analogue of an observer watching the flow of a stream, who decides to add some dye to the water to monitor the flow of particles. If the observer follows the path of the dyed particles in the water, the Lagrangian description applies; if the observer remains at a certain point along the river bank and notes which particles pass with time, the Eulerian description applies.

A hydrocode may employ either type of formulation to describe the situation of interest. To obtain a Lagrangian description the computational mesh is defined to represent the geometry of the problem; points within the mesh (vertices) are attached to the material and move with the material. As time progresses, all the variables of interest are computed for each discrete point in the continuum. Cells defined by adjacent vertices become deformed in shape and size due to the forces acting on them and the constitutive relations between force and deformation. From a mathematical point of view the vertices follow the *particle paths* of the material. Mass, momentum and energy are transported by material flow. Mass within a cell is invariant; changes in density are exclusively due to changes in a cell's volume.

In contrast, the Eulerian description relies on material flowing *through* the mesh. In this case the mesh defines the entire space of interest. As time progresses, the variables of interest are calculated at the fixed points of the grid. Thus, mass, momentum and energy must flow across cell boundaries. The amount of flow between cells is used to compute the new variables within each cell. In this formulation it is the volume of the cell that is invariant and changes in density are due to changes in the mass of a particular cell.

The choice of either mode of description depends on the problem under consideration. To illustrate the differences between Lagrangian and Eulerian material descriptions, as well as the inherent advantages and disadvantages of each, an example inspired by lecture notes from a short course on Impact Modeling by H. J. Melosh is presented. Consider the case of a dinosaur plunging to its demise from a "K/T diving board" Figure 2. Either the Lagrangian or the Eulerian description may be employed to model this situation with a computer.

In the Lagrangian formulation (Figure 3a) the dinosaur is divided into cells, which should be as close as possible to regular quadrilaterals to ensure accuracy from the finite difference approximations used. The cells that make up the dinosaur experience external stresses (gravity) and internal stresses (via the constitutive model) that translate them and deform them. As a consequence, the dinosaur falls, lands and is crushed. The surface of the

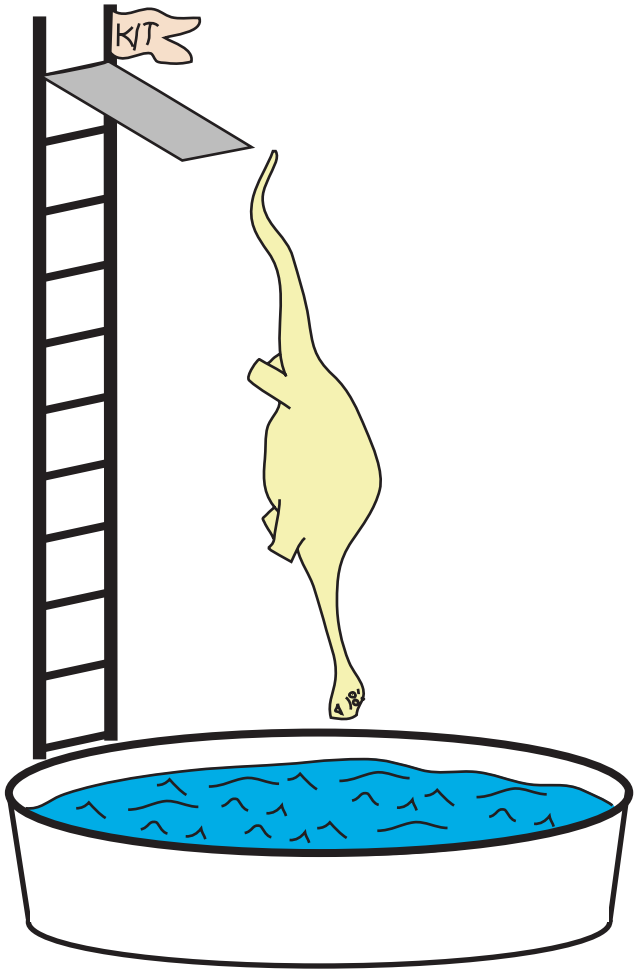

Figure 2: Illustration of the a dinosaur diving from the K/T diving board

dinosaur is defined by the edges of the cells, thus the resolution of the surface is a function of the number and size of the cells used. The more cells the better resolved the free-surface. Furthermore, if the problem involves multiple materials, the boundaries between different materials can be accurately defined by cell boundaries. A significant advantage of the Lagrangian description is its ability to follow the history of the material within any cell. Thus, a material whose properties depend upon its previous history can be modeled; for example, strain hardening or plastic work. Furthermore, it is straightforward to model time-dependent phenomena not included in the solution algorithm implicitly; for example, the temporal decay of acoustic vibrations.

The major weakness with the Lagrangian description comes with extreme deformation. As has just been discussed, the cell quantities of volume, pressure, and so on, are approximated by finite difference relations.

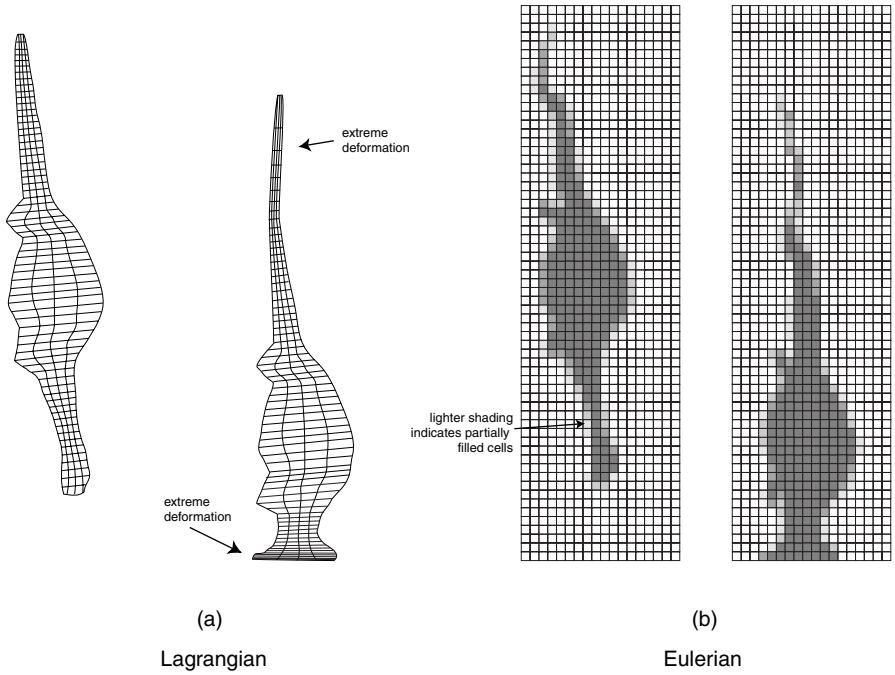

Figure 3: Lagrangian (a) and Eulerian (b) description of the dinosaur diving from the K/T diving board.

These expressions become inaccurate when the form of the cell deviates significantly from a regular quadrilateral. Thus, in cells where deformation is severe (for example in the head of the dinosaur) the calculation will at best produce spurious results and at worst become unstable and stop.

In the Eulerian formulation (Figure 3b) the mesh must define the entire space in which the dinosaur's movement is contained. During the calculation the dinosaur will occupy certain cells completely, other cells partially and other cells not at all. This is illustrated by the shading in Figure 3b; cells containing a greater concentration of dinosaur mass are shaded darker. Consequently, to achieve the same spatial resolution over the dinosaur as the Lagrangian description, the number of cells required is significantly greater. Another disadvantage of the Eulerian description is that as the dinosaur moves through the mesh it will only partially occupy some cells; a problem exists of how to represent cells that are a percentage dinosaur and a percentage space. Hence, unlike the Lagrangian description, where the free surface is precisely defined, the free surface is not tracked exactly in the Eulerian description. In other words, the ability to model the free surface in the Eulerian description is a much more sensitive function of the resolution of the mesh than in the Lagrangian description. Furthermore, some cell quantities like viscosity pose a problem in the Eulerian description. Representing a cell that is partially space and partially dinosaur in terms of mass is relatively

straight forward. However, defining this cell's viscosity is less intuitive.

A particularly tractable hydrocode, SALE, was developed by *Amsden et al.* (1980) to study fluid flow at all speeds. SALE stands for Simplified Arbitrary Lagrangian Eulerian (*Hirt et al.*, 1974); meaning that it is capable of solving fluid-flow problems using either, or a combination, of the two descriptions. It achieves this by solving a Lagrangian set of differential conservation equations and then, if the Eulerian description is desired, remapping the cell and vertex quantities back onto the original grid.

3.4 Accuracy and Stability

It is important to understand the limitations and caveats associated with hydrocode modeling. These issues deal with the accuracy of representing a continuum problem on a computer with a finite memory allocation, and numerical instabilities inherent in the finite difference approximations.

3.4.1 Accuracy

Two types of errors occur which impair the accuracy of a finite difference scheme: round-off errors and truncation errors. Round-off errors are associated with the precision of the computer. These errors are introduced after a repetitive number of calculations in which the computer is constantly rounding the numbers to some significant figure and can be minimized by using high precision. Truncation errors, on the other hand, are a function of the difference scheme used, not the hardware on which the code is run. The essence of this error arises from representing a continuous variable with a discrete number of points, and the magnitude of the errors depends on the mesh intervals in time and space.

The useful accuracy of a given numerical solution may be difficult to determine analytically (*Hirt*, 1968). In any case, it is usually insightful to adopt a "brute force" approach and use a spectrum of computer runs with different meshes and time steps to determine if a calculated effect is physical or simply a numerical artifice. In general, the cell size and time step must be fine enough to resolve all spatial and temporal variations of interest. If a solution exhibits large variations over distances comparable to a cell width, or over times comparable to the time step, it is probably not very reliable.

3.4.2 Resolution

The choice of resolution in space and time is an important issue when using hydrocodes. A simulation should be conducted with a high enough resolution to resolve all the important flow variations in space and time. However, these needs must be balanced by the specifications of available hardware and the time available to run the simulation. The amount of memory required for a given simulation is roughly proportional to the number of cells in the

mesh and the duration of the calculation is dependent on the processor speed and, more importantly, the size of the time step.

The choice of cell size is both hardware and problem specific and is left to the user to decide. A good rule of thumb, however, is to use a time step that is as large as possible given the appropriate restrictions. In many cases, the largest appropriate time step size is limited by the Courant condition on sound signal propagation. This time step dt_{cou} is the minimum time taken for sound to cross a cell:

$$dt_{cou} < \min\left(\frac{dx}{c}, \frac{dy}{c}\right), \tag{7}$$

where dx and dy are the cell dimensions and c is the speed of sound in the material modeled. The minimum implies that every cell in the mesh must be considered to ensure that dt_{cou} satisfies the most restrictive case.

3.4.3 Stability

In addition to concerns over accuracy and resolution, hydrocodes suffer from inherent instabilities arising from the difference schemes used. Numerical methods may give solutions that develop large, high-frequency oscillations in space or time (*Amsden et al.*, 1980); if the physical problem being modeled is known not to exhibit such behaviour, the source may be numerical instability.

Many numerical devices have been developed to mitigate against instability. For example, there are well documented limits placed on the size of the time step to ensure that information cannot travel across a full cell width in a single time step. Three such restrictions relevant to the calculations described in this report are discussed here.

Firstly, all hydrocode calculations should require that the Courant condition (equation 7) on sound signal propagation is not violated. Secondly, the time step should be restricted by the very well-verified condition that fluid cannot be moved more than approximately one cell-width per time step; that is,

$$dt < \min\left(\frac{dx}{|u|}, \frac{dy}{|v|}\right), \tag{8}$$

which is known as the Cauchy, or convective flux limit. The minimum in the above equations implies that every cell in the mesh must be considered to ensure that the dt satisfies the most restrictive case. *Amsden et al.* (1980) recommend that the time step should never exceed one-fifth of the minimum cell transit time, or the minimum sound propagation time.

The final restriction on time step due to numerical stability relates to the stress tensor. When viscous effects are included, the crucial condition to be satisfied in every cell is that

$$dt < \left[\frac{2(\lambda+2\eta)}{\rho}\left(\frac{1}{dx^2}+\frac{1}{dy^2}\right)\right]^{-1} = \frac{\rho dx^2 dy^2}{2(\lambda+2\eta)(dx^2+dy^2)}, \qquad (9)$$

which roughly states that momentum must diffuse less than one cell-width per time step. λ and η correspond to the bulk and shear viscosities respectively.

Another numerical consideration is mesh stability. In a Lagrangian calculation of fluid flow using quadrilateral mesh cells, there are certain mesh deformations that do not result in net pressure changes; that is, there is change in cell shape but no change in cell volume. Such nonphysical motion of vertices is sometimes termed "vertex coasting". To prevent such deformations from slowly degrading the solution, alternate mesh nodes can be coupled with a small artificial restoring force. This "alternate-node coupling" is discussed in detail in the SALE manual.

3.4.4 Treating Shock Waves

The mathematical idealization of a shock wave is an instantaneous jump in pressure, velocity, density and internal energy. The shock is actually broadened by rate-dependent physical mechanisms such as viscosity, damage and pore collapse (*Melosh et al.*, 1992). Nevertheless, rapid changes that occur over shorter distances than the smallest possible cell size are problematic for any numerical treatment using a finite difference representation. A special approach must be adopted, therefore, when modeling the passage of a shock wave with a hydrocode.

One approach for problems involving shock waves, is to apply an artificial viscosity (*von Neumann and Richtmyer*, 1950), which serves to spread the shock over several cell-widths while preserving the character of the jump in material properties. Without such dissipation, spurious velocity oscillations develop behind the shock, which can mask or swamp the true solution (*Anderson*, 1987). A widely cited form for the artificial viscosity term is a pressure addition, q given by:

$$\begin{aligned} q &= A\rho\nabla^2.v \qquad && \nabla.v < 0, \\ q &= 0 && \nabla.v \geq 0, \end{aligned}$$

(*Richtmyer and Morton*, 1967), which states that q is proportional to the square of the velocity divergence $\nabla.v$, when $\nabla.v < 0$; that is, in compression only.

4 Summary: an introduction to SALES_2

This introduction has described, briefly, the fundamental aspects of hydrocode modeling. The important things to take away are:

- Hydrocode modeling is a powerful technique for studying complex natural processes.
- Hydrocode modeling rests on three fundamental pillars, which provide the foundation for the solution algorithm.
- There are different types of hydrocode, which use different methods of discretization and different forms for the governing equations. These methods have advantages and disadvantages; these will determine the suitability of certain hydrocodes to a given problem.
- There is no universal code: the usefulness of a hydrocode relates to the sophistication of the equation of state and the constitutive model implemented.
- All hydrocodes suffer from issues of accuracy and instability. Hence, some form of code verification is essential.

Following this introduction are two documents, which together describe, in detail, the SALES_2 hydrocode. This hydrocode is a relative of an earlier, simpler code SALE. The first document is the original SALE manual. It describes only a very simple equation of state and constitutive model, capable of simulating only Newtonian fluid flow. To adequately model more complex problems SALE has been extensively modified to include sophisticated methods for modeling material response to stress (for example *Melosh et al.*, 1992; *Ivanov et al.*, 1997). SALES_2 retains all the functionality of SALE as well as being capable of modeling elasticity, plasticity, multiple materials and rheologies, and strength degradation via fragmentation. The documentation for SALES_2, which is designed to complement the original SALE manual, is the second document included after this report.

References

Amsden, A. A., Scheme for generating grids, *Journal of Computational Physics*, *11*, 848–859, 1973.

Amsden, A. A., H. M. Ruppel, and C. W. Hirt, Sale: Simplified ALE computer program for fluid flow at all speeds, *Tech. Rep. LA-8095*, Los Alamos National Laboratory, Los Alamos, NM, 1980.

Anderson, C. E., An overview of the theory of hydrocodes, *International Journal of Impact Engineering*, *5*, 33–59, 1987.

Bathe, K.-J., and E. L. Wilson, *Numerical Methods in Finite Element Analysis*, Prentice-Hall, 1976.

Collins, G. S., and H. J. Melosh, Sales_2: A multi-material hydrocode with elastic-plastic strength model and grady-kipp fragmentation algorithm.

Hirt, C. W., Heuristic stability theory for finite difference equations, *J. Comput. Phys.*, *2*, 339, 1968.

Hirt, C. W., A. A. Amsden, and J. L. Cook, An arbitrary lagrangian-eulerian computing method for all flow speeds, *J. Comput. Phys.*, *14*, 227–253, 1974.

Ivanov, B. A., D. Deniem, and G. Neukum, Implementation of dynamic strength models into 2d hydrocodes: Applications for atmospheric breakup and impact cratering., *Int. J. Impact Engineering*, *20*, 411–430, 1997.

Melosh, H. J., E. V. Ryan, and E. Asphaug, Dynamic Fragmentation in Impacts: Hydrocode Simulation of Laboratory Impacts, *JGR*, *97*, 14,735–14,759, 1992.

Monaghan, J. J., An introduction to SPH, *Computer Physics Communications*, *48*, 89–96, 1988.

Richtmyer, R. D., and K. W. Morton, *Difference Methods for Initial-Value Problems*, 2nd ed., Interscience Publishers, New York, 1967.

von Neumann, J., and R. D. Richtmyer, A method for the numerical calculation of hydrodynamic shocks, *J. Applied Phys.*, *21*, 232–237, 1950.