TITLE: HIGH RESOLUTION FINITE VOLUME PARALLEL SIMULATIONS OF MOULD FILLING AND BINARY ALLOW SOLIDIFICATION ON UNSTRUCTURED 3-D MESHES

AUTHOR(S): A(nand) V. Reddy, T-3
D(ouglas) B. Kothe, T-3
C. Beckermann, Univ. of Iowa
R. C. Ferrell, CPCA, Ltd.
K. L. Lam, ESA-EA

SUBMITTED TO: SP97: The Fourth Decennial International Conference on Solidification Processing, University of Sheffield, UNITED KINGDOM, July 7–10, 1997

By acceptance of this article, the publisher recognizes that the U.S. Government retains a nonexclusive, royalty-free license to publish or reproduce the published form of this contribution, or to allow others to do so, for U.S. Government purposes.

The Los Alamos National Laboratory requests that the publisher identify this article as work performed under the auspices of the U.S. Department of Energy.

DISTRIBUTION OF THIS DOCUMENT IS UNLIMITED

Los Alamos National Laboratory
Los Alamos, New Mexico 87545
DISCLAIMER

This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, make any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the United States Government or any agency thereof.
DISCLAIMER

Portions of this document may be illegible in electronic image products. Images are produced from the best available original document.
High Resolution Finite Volume Parallel Simulations of Mould Filling and Binary Alloy Solidification on Unstructured 3-D Meshes

A.V. Reddy†, D.B. Kothe‡, C.Beckermann§, R.C.Ferrell+ and K.L. Lam#

Abstract
The Los Alamos National Laboratory (LANL) is currently developing a new casting simulation tool (known as Telluride) that employs robust, high-resolution finite volume algorithms for incompressible fluid flow, volume tracking of interfaces, and solidification physics on three-dimensional (3-D) unstructured meshes. Our finite volume algorithms are based on colocated cell-centered schemes that are formally second order in time and space. The flow algorithm is a 3-D extension of recent work on projection method solutions of the Navier-Stokes (NS) equations. Our volume tracking algorithm can accurately track topologically complex interfaces by approximating the interface geometry as piecewise planar. Coupled to our fluid flow algorithm is a comprehensive binary alloy solidification model that incorporates macroscopic descriptions of heat transfer, solute redistribution, and melt convection as well as a microscopic description of segregation. Our finite volume algorithms, which are efficient, parallel, and robust, can yield high-fidelity solutions on a variety of meshes, ranging from those that are structured orthogonal to fully unstructured (finite element). We discuss key computer science issues that have enabled us to efficiently parallelize our unstructured mesh algorithms on both distributed and shared memory computing platforms. These include our functionally object-oriented use of Fortran 90 and new parallel libraries for gather/catter functions (PGSLib) and solutions of linear systems of equations (JTipack90). Examples of our current capabilities are illustrated with simulations of mold filling and solidification of complex 3-D components currently being poured in LANL foundries.

Introduction
Driven by increasing demands on quality and control of microstructure of materials, modeling of casting processes is being increasingly relied upon to predict potential solidification defects and thus improve casting practices, reduce foundry costs, and aid in the design of new and improved materials. Casting simulation tools should be able to capture complicated cast geometries, which often demands the use of unstructured meshes. Most finite volume algorithms, however, are restricted to orthogonal meshes, which force boundaries to be stair-stepped. This introduces errors in the simulation. Accuracy of simulations also depends on the number of grid points used to discretize the physical domain. Unless a simulation can be performed in parallel (where multiple processors are coordinated to perform the work simultaneously) CPU time and memory requirements become unacceptable when desired simulation accuracy and length scale resolution calls for large meshes. This is especially true in 3-D, hence parallelism is imperative in a modern casting simulation tool.

Physical Model
We are currently modeling alloy solidification with a simplified version of the volume averaged two-phase model of Ni and Beckermann [1], where the solid phase is assumed to be stationary. The governing equations are summarized in reference 2, and includes mass, momentum, energy and species conservation equations that are valid in the single-phase liquid and solid regions as well as in the mushy zone. They are supplemented by phase diagram relations, a permeability expression, a back-diffusion model, and others. Simplified energy and species equations are presented in Table 1 to facilitate the discussion of the phase change algorithm below.

---

*Supported by the Department of Energy Accelerated Strategic Computing Initiative Program.
†LANL, Fluid Dynamics Group T-3, Theoretical Division, Los Alamos, NM 87545 (anand@lanl.gov)
‡LANL, Fluid Dynamics Group T-3, Theoretical Division, Los Alamos, NM 87545 (dbk@lanl.gov)
§Dept. Of Mech. Engg., Univ. of Iowa, Iowa City, IA 52245 (becker@icen.uiowa.edu)
#LANL, Engineering Analysis Group ESA-EA, Los Alamos, NM 87545 (klam@lanl.gov)
Numerical Method

The system of conservation equations are integrated over a control (cell) volume to yield their integral (weak) form. Volume integrals are then converted to face integrals, which are approximated as discrete sums over faces. Second-order spatial and temporal accuracy results when face quantities are estimated with monotonicity-preserving time-centered Taylor series expansions. Our overall solution method integrates one time step with a series of fractional substeps, where each substep approximates a selected term of the conservation equations (e.g., advection, viscosity, etc.). This fractional time-step-split method allows our algorithms to be developed in an accurate and modular fashion, and enables easy addition of new physical models.

Incompressible Flow Algorithm

LANL has a long history and made important contributions in the development of computational methods for modeling incompressible interfacial (e.g., free surface) flows. A particularly important contribution occurred in 1965 with the Marker-and-Cell (MAC) method [MAC], which remains (along with its successors) a popular choice to this day. A recent example of a MAC successor is RIPPLE [3], which has become a very popular model for 2-D free surface flows because of its algorithmic and physical model improvements, most notably the continuum surface force (CSF) model for surface tension [4]. Since the inception of the MAC method, numerous advances have taken place in the development of incompressible interfacial flow algorithms over the past three decades. Perhaps most notable is the recent work of Bell, Colella, and coworkers [5,6] in devising high resolution projection method solutions of the Navier-Stokes (NS) equations coupled with modern interface tracking algorithms [7]. This approach has yielded high-fidelity flow solutions that are fully second-order in time and space. The methodology borrows in large part from algorithms devised for high-speed flow, by coupling projection methods with high-order Godunov advection and interface tracking.

Analysis and refinement of these projection methods has resulted in an overall NS solution algorithm that is robust and accurate[8]. We have incorporated material interface variable-density effects (e.g., due to free surfaces) into our flow algorithm with the addition of various modern interface tracking methods [9]. Because of our need to model LANL gravity-pour mold-filling scenarios, we have considered only those interface tracking algorithms capable of following surfaces that undergo gross topology change. In designing interface tracking methods, they must extend to 3-D and have no restrictions on the topological complexity or the number of interfaces that may be represented. After extensively studying many possible approaches [9], our design constraints have led us to PLIC (piecewise-linear interface calculation) volume tracking methods [10,11].

We have extended our PLIC VOF algorithms to 3-D generalized hexahedra grids [7]. In this approach, fluid interfaces are tracked on general nonorthogonal hexahedral meshes and localized over a one cell width for each time step. This is in contrast to high-order continuum advection schemes where discontinuities are typically more diffuse, having minimum transition widths (on the order of the mesh spacing) [9]. We define interface geometry using local fluid volume-fraction data, and assume interfaces to be locally planar within each cell. This results in a globally piecewise planar approximation to the actual interface topology. Knowledge of the interface geometry allows partitioning of total flux volumes into individual fluid flux volumes. Total fluid volume, by construction, is conserved rigorously.

We have extended our projection-based NS solution method to 3-D unstructured grids without needlessly sacrificing robustness, accuracy, or efficiency. Our current approach has borrowed in part from techniques originating in the high-speed-flow aerodynamics community, an example being the least-squares reconstruction schemes devised by Barth [12], which maintain high fidelity on complex unstructured meshes. We have also extended a 3-D unsplit advection technique [13] to unstructured meshes, which avoids operator-splitting and allows consistent use of high-order monotone advection in incompressible flows.

Heat Transfer and Phase Change Algorithm

Phase change in pure materials is characterized by a constant temperature at which the phase change occurs. In such situations the phase change rate is solved from the energy equation. Alloy solidification is more complicated as it involves both energy and species transfer equations.
It can be seen from Table 1 that the species and energy conservation equations are intimately coupled through the phase change rate term. Thus, during the numerical solution of these equations it is essential to have a good estimate of the phase change rate term to achieve speedy convergence. Several methods have been proposed for updating phase change rate during each solution iteration. Schneider and Beckermann [2] propose a method which explicitly solves for the phase change rate by combining the discretized energy and species equations. Voller and Swaminathan [14] propose methods in which the phase change term is linearized as a truncated Taylor series, and old iteration values are then used to estimate the linear term. Here a different approach is taken in which the phase change rate term is estimated using a relation derived using the binary phase diagram relation and the species transfer equations. This method is shown to be efficient and robust.

An iterative scheme for updating the phase change source term in the energy equation (Table 1) can be written as

$$ S = (h_i - h_s) \frac{\rho_s^{m+1} \varepsilon_s^{m+1} - \rho_s^o \varepsilon_s^o}{\delta t} $$

where the superscripts m and o refer to the iteration level and the previous time step respectively. Following Voller, on using a truncated Taylor series expansion for $\rho_s^{m+1} \varepsilon_s^{m+1}$

$$ \rho_s^{m+1} \varepsilon_s^{m+1} = \rho_s^m \varepsilon_s^m + \frac{d(\rho_s \varepsilon_s)}{dT}(T^{m+1} - T^m) $$

When this is substituted in Eq. (1) we get a linear source term in temperature for the energy equation. How ever we need a good estimate for $d(\rho_s \varepsilon_s)/dT$ for speedy convergence of the energy equation. A derivation of a general expression for this term is outlined below.

Discretizing and rearranging the solid species equation (Table 1) one can write

$$ \rho_s \varepsilon_s C_s = \rho_s^o \varepsilon_s^o C_s^o + C_{si} \delta(\rho_s \varepsilon_s) + \frac{S \rho_s D_s}{l_s} \delta t(C_s - C_s^o) $$

similarly discretizing the liquid species equation

$$ \rho_l \varepsilon_l C_l = \rho_l^o \varepsilon_l^o C_l^o + (C_l - C_{si}) \delta(\rho_s \varepsilon_s) + \frac{S \rho_s D_s}{l_s} \delta t(C_l - C_{si}) $$

where the primed quantity refers to quantities after the advection time step has been completed, i.e., the primed quantity is the concentration after the advection source term has been used to modify the old time step concentration (fractional time stepping scheme). (Note that solid is stationary and hence its concentration does not change during the advection step).

Now we can obtain an expression for $d(\rho_s \varepsilon_s)/dT$ by performing the following three steps. First, substitute binary phase diagram relations (for liquid and interfacial solid concentrations) in Eqs. (3) and (4). Second, differentiate the resulting equations with respect to temperature. And finally, eliminate the rate of change of solid concentration with temperature term from the equations resulting from step 2, thus obtaining the required expression.

A few comments about the numerical scheme employed to handle isothermaVeutectic phase change: Voller and Swaminathan[14] employ a large source term procedure to force the temperature of a control volume undergoing solidification to the melting point value. In CG type methods, a large coefficient increases the condition number of the coefficient matrix. This restricts the CG algorithm and the solution does not converge to the desired tolerance. To overcome this difficulty, the neighboring coefficients are set to zero except the diagonal element which is set to one. Then the RHS matrix is set to the melting point temperature. This procedure however makes the matrix non-symmetric. And hence GMRES is employed to solve the set of equations.

**Parallelization Strategy**
Given our use of (almost) fully-implicit finite volume algorithms that frequently invoke indirect addressing (because of the unstructured mesh connectivities), surprisingly high parallel efficiency (> 85%) has been regularly attained for our casting simulations [15-17]. We summarize briefly our parallelization approach here, and refer the interested reader to reference 15 for further details.

Our strategy for parallelization is to explicitly decompose and distribute the global mesh across all processors available to perform work on the problem at hand. This strategy is independent of the whether the memory accessible to the processors is local (distributed memory systems) or global (shared memory systems). We do not choose to rely upon compiler directives (as in High Performance Fortran) or parallelism switches for parallelism, but rather explicitly design the parallelism into our software by separating all communication (indirect addressing) from computation. All indirect addressing functions are performed in gather/scatter procedures that are themselves parallelized by the explicit passing of messages between processors via calls to the MPI library. Platform-specific, explicit parallelism appears only in these procedures (instead of being littered throughout the entire source), which in turn call upon PGSLib [16] to actually perform the MPI-based message passing.

A given time step can require several matrix solutions, so the majority of our solution algorithm is spent in the JTpack90 linear solver library [17]. We currently solve our systems with JTpack 90 in parallel over the entire mesh, rather than invoking a Schwarz decomposition [18]. For orthogonal meshes, we store the matrix and use preconditioned CG to solve the system. For generally nonorthogonal, unstructured meshes, we do not store the matrix and use preconditioned GMRES to solve the system. In all cases, we interface with JTpack90 in matrix-free form, i.e., matrix-vector multiplication is performed with procedures provided by Telluride. This approach avoids having to assimilate and store the matrix, which for a generally unstructured mesh can be intractable.

Results

Filling and solidification results of two simulations are presented below. The coupling between fluid flow and heat transfer is currently being implemented in Telluride and therefore in the examples presented here, filling is done isothermally and after the mold is completely filled the solidification equations are solved assuming a quiescent liquid.

Solidification of an Al-4%Cu Ingot

An Al-4%Cu alloy was solidified in a 0.05x0.1x0.1m box by extracting heat from four points on the x=0 plane by maintaining these points at 600K. Initially the melt is assumed to be superheated by 5K at 921K. Eutectic temperature for Al-Cu is 821K. A mesh of 16x32x32(16384) cubic orthogonal cells was employed to discretize the domain.

Figure 1 shows isosurfaces of 4 different temperatures at 195sec into the simulation. The isosurfaces corresponding to 821K and 916K are the eutectic and liquidus isosurfaces respectively, and hence they demarcate the mushy zone. The isosurfaces close to the heat extraction points are hemispherical as expected.

Filling and Solidification of a Copper Chalice

A more realistic problem (solidification of a copper chalice) is chosen for the second simulation presented here. The copper chalice was cast at a LANL foundry in support of the inertial confinement fusion program. It is essentially a hemispherical shell (two inch diameter) gated at the pole with a cylindrical riser. The riser serves to continuously supply liquid copper to the hemispherical shell during filling solidification (to avoid shrinkage defects). The riser is then cut away and machined after solidification to give the final product (the hemispherical shell).

One quadrant of the chalice is simulated, with the geometric model and computational mesh (6480 unstructured hex elements) being generated with the I-DEAS commercial software package. A filling simulation was performed in which the melt was gravity-poured into the riser. The total filling time was approximately 1.5 seconds. Figure 3 shows the fill at xx seconds.

For the solidification simulation, approximate boundary conditions were estimated from the experimental setup. The initial temperature of the melt is 1543K. A convective heat transfer
coefficient of 25W/m²-K is used for the inner surface of the hemisphere and the bottom of the chalice. For the outer surface a value of 15W/m²-K was used. The top was assumed insulated because of its proximity to a heater. Figures 3 shows the liquid volume fraction field at 425s into the simulation. It can be noticed from Fig.3 that liquid in the hot top solidifies last and thus no shrinkage defects can be expected in the shell. The total solidification time is about 10 minutes.

The above solidification problem with a finer grid was simulated on a parallel machine. Using the Chaco [] decomposition software, the mesh was decomposed into an arbitrary number of submeshes, depending upon the number of processors available to do the problem. Figure 4 shows the decomposed mesh for 8 processors. Impressive parallel efficiencies were achieved, e.g., a speed up of 7 was achieved for a simulation with 8 processors on a 300MHz Digital AlphaServer 8400. For more speed up results refer to 15 and 16.

**Nomenclature**

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ε</td>
<td>volume fraction</td>
</tr>
<tr>
<td>ρ</td>
<td>density (kg/m³)</td>
</tr>
<tr>
<td>C</td>
<td>concentration (wt pct)</td>
</tr>
<tr>
<td>D</td>
<td>mass diffusivity (m²/s)</td>
</tr>
<tr>
<td>h</td>
<td>enthalpy (J/kg)</td>
</tr>
<tr>
<td>k</td>
<td>conductivity (W/m/K)</td>
</tr>
<tr>
<td>l</td>
<td>length scale (m)</td>
</tr>
<tr>
<td>S</td>
<td>interfacial area concentration (m⁻¹)</td>
</tr>
<tr>
<td>T</td>
<td>temperature (K)</td>
</tr>
<tr>
<td>t</td>
<td>time (s)</td>
</tr>
<tr>
<td>v</td>
<td>velocity (m/s)</td>
</tr>
<tr>
<td>i</td>
<td>interfacial/summation variable</td>
</tr>
<tr>
<td>l</td>
<td>liquid</td>
</tr>
<tr>
<td>s</td>
<td>solid</td>
</tr>
<tr>
<td>m, m+1</td>
<td>iteration level</td>
</tr>
<tr>
<td>o</td>
<td>old time step variable</td>
</tr>
</tbody>
</table>

**References**

Table 1: Energy and Species Equations

Mixture energy: \( \frac{\partial}{\partial t} (\varepsilon_s \rho_s h_s + \varepsilon_1 \rho_1 h_1) + \nabla \cdot (\varepsilon_s \rho_s \mathbf{v}_s h_s + \varepsilon_1 \rho_1 \mathbf{v}_1 h_1) = \nabla \cdot \left( \frac{k_s \varepsilon_s}{c_s} \nabla h_s + \frac{k_1 \varepsilon_1}{c_1} \nabla h_1 \right) \)

where, \( h_s = c_{ps} T \) and \( h_1 = c_{p1} T + ((c_{ps} - c_{p1}) T_e + \Delta h) \)

Species: \( \frac{\partial}{\partial t} (\varepsilon_k \rho_k C_k) + \nabla \cdot (\varepsilon_k \rho_k \mathbf{v}_k C_k) = \nabla \cdot (D_k^* \rho_k \varepsilon_k \nabla C_k) + C_{ki} \Gamma_k + \frac{S_s \rho_k D_k^*}{\nu_k} (C_{ki} - C_k) \)
Fig. 1 Temperature isosurfaces

Fig. 2 Chalice solidification (475s)

Fig. 3 Chalice fill (0.5s)

Fig. 4 Exploded view of chalice mesh