The University of Sheffield

Department of Mechanical and Process Engineering

RERVM COGNOSCERE CAVSAS

Computer Modelling of Solidification of Pure Metals and Alloys

By

Michael Rudolf Barkhudarov

A thesis submitted in fulfilment of the requirements for the Degree of Doctor of Philosophy at the Department of Mechanical and Process Engineering, University of Sheffield.

June, 1995

Computer Modelling of Solidification of Pure Metals and Alloys

Summary

Two numerical models have been developed to describe the volumetric changes during solidification in pure metals and alloys and to predict shrinkage defects in the castings of general three-dimensional configuration. The first model is based on the full system of the Continuity, Navier-Stokes and Enthalpy Equations. Volumetric changes are described by introducing a source term in the Continuity Equation which is a function of the rate of local phase transformation. The model is capable of simulating both volumetric shrinkage and expansion.

The second simplified shrinkage model involves the solution of only the Enthalpy Equation. Simplifying assumptions that the feeding flow is governed only by gravity and solidification rate and that phase transformation proceeds only from liquid to solid allowed the fluid flow equations to be excluded from consideration.

The numerical implementation of both models is based on an existing proprietary general purpose CFD code, FLOW-3D, which already contains a numerical algorithm for incompressible fluid flow with heat transfer and phase transformation. An important part of the code is the Volume Of Fluid (VOF) algorithm for tracking multiple free surfaces. The VOF function is employed in both shrinkage models to describe shrinkage cavity formation.

Several modifications to FLOW-3D have been made to improve the accuracy and efficiency of the metal/mould heat transfer and solidification algorithms.

As part of the development of the upwind differencing advection algorithm used in the simulations, the Leith's method is incorporated into the public domain two-dimensional SOLA code. It is shown that the resulting scheme is unconditionally stable despite being explicit.

M. R. Barkhudarov

Contents

Chapter 1

Introduction

1.1 Casting Technology

Casting technologies differ greatly in casting and die materials used, filling methods, cooling rates and final product properties. *Ingots* and *billets* are castings of simple shape, usually suitable for mechanical working, *e.g.* forging, rolling, extrusion and drawing. In *investment* casting (or *lost wax*, or *precision* casting) a pattern of wax or other fusible material is die cast and a plaster of Paris mould is then made round it. When the plaster of Paris hardens the wax is burnt away, leaving a clean cavity for the metal casting with precise dimensions and good surfaces. These and many others processes share the same physical phenomena, that is: fluid flow of the liquid metal, heat flow from the hot metal to the cold mould and finally liquid to solid phase transformation of the metal in the mould [7].

Each of these processes and their interactions depend not only on the thermophysical properties of the materials, but also on the starting and boundary conditions, *i.e.* the initial metal and mould temperatures, pouring method and mould geometry. All these factors together are called *casting design.*

A castings design is aimed at producing the part with the desired properties, at the lowest cost, using available materials. One of the objectives is to produce uniform, equi-axed, fairly fine crystals in large castings, for this structure gives metal with the best strength and other mechanical properties [6,7,8].

Apart from grain structure, several other factors determine the quality of a cast metal. In all casting processes care must be taken to avoid splashing the metal on the sides of the mould, otherwise splashed solid films form and then become coated with oxide which prevents them from bonding with the rest of the casting (*cold shuts*). Oxide films on the liquid surface may also become folded into the metal unless the pouring is done with care. The oxide film on aluminium is particularly troublesome, partly for this reason and partly because, being a continuous and mechanically strong film, it prevents the metal from filling sharp corners and narrow channels in the mould [2]. These effects can be predicted given an appropriate numerical model for fluid flow and free surface.

Other main factors are *segregation* and *porosity* [2,6,7,8]. Porosity in a casting is largely defined by *gas evolution* and *shrinkage* in the solidifying metal. Predicting macro-porosity due to shrinkage is the aim of the mathematical and numerical modelling presented in this work.

The shrinkage which occurs on solidification is a source of many practical difficulties in the casting of metals. The change of the volume for most metals varies between 2 to 6 per cent (a few, such as bismuth and gallium, expand on freezing). Shrinkage is equally important on both macroscopic and microscopic scales, the latter appearing between dendrites and grains, and may lead to an unsound casting, full of fine-scale porosity, unless the interdendritic liquid channels can all be continuously fed with liquid from a central supply to make up for the volume difference when they freeze. For a sound casting the metal should freeze progressively from the casting to the feeder with no trapping of liquid in pockets totally enclosed by solid; and the zone in which freezing takes place, between the fully solid and fully liquid regions, should be narrow so that the feeding liquid is not required to flow along long and narrow interdendritic channels. The casting must of course be provided with a feeder head to maintain a pool of liquid to feed the solidifying metal below as it flexes and shrinks.

1.2 Casting Modelling in Modern Industry

Until recently casting industry relied mostly on experience, intuition and a kind of esoteric knowledge passed on from generation to generation. But over the last 10 to 15 years there has been an increase in interaction between foundries and computer software developing research groups. Due to commercial pressure from other metal forming techniques and the necessity of having high quality products, the foundry designer is looking for up-to-date analytical tools to improve the casting techniques. The need for a more sophisticated approach to the design of the manufacturing process is also dictated by the increased range of new alloys and products manufactured using casting technology for which no previous experience is available.

It is widely accepted that the use of a computer model which simulates the physical phenomena of a casting process could save time and resources during the design stage. It helps to reduce the risk of misruns, defect formation *etc.* Due to the breathtaking achievements of the computer industry and equally wonderful developments in the computational fluid dynamics there are a number of commercially available software packages capable of reliable, robust and sufficiently fast simulations of a remarkable range of fluid flows: viscous and inviscid, compressible and incompressible fluid flows with transient free surfaces and shock waves; non-Newtonian fluids, stress analysis, heat transfer are also included into some numerical models. More and more computer codes are produced specifically to describe solidification and they are successfully used on a regular basis.

However, by no means the task of modelling the casting process fully, *e.g.* microstructure and defect formation, can be considered completed. Accurate simulation of filling stage is still considered a difficult task and in many cases skipped, assuming an instantaneous filling. Accuracy of cooling and solidification solution requires an adequate description of the mould geometry, metal/mould interface and a knowledge of the material properties. Finally, the phase transformation and defect formation often involves microscale phenomena and pose a problem of coupling the corresponding micromodels to the standard macromodels used in computer simulations.

The existing computer tools differ in capabilities, accuracy, applicability, and none of them models the process in full. There is still an increasing demand for a universal and reliable analytical tool which would both increase the productivity of the industry and help to create an insight into physical phenomena occurring during the solidification process. Here is an incomplete list of what a foundryman would expect from a computer model to do according to Piwonka [1]:

- predict how the casting will fill with liquid metal;
- establish the temperature distribution in the mould and the liquid metal at the start of solidification;
- describe how this temperature distribution in the liquid changes as a result of thermal and solutal convection;
- describe the solidification of the casting;
- show the local chemical composition as a result of segregation;

- predict the grain size and shape;
- predict the distribution of gas content within the casting;
- predict the secondary dendrite spacing;
- describe the distortion of the casting as it freezes;
- predict the distribution of residual stresses within the casting;
- map minimum values of mechanical properties which the casting is expected to have;
- estimate the production cost of the casting.

The objective of the present work is to model physical phenomena in

- mould filling with viscous fluid including free surface and solidification effects;
- solidification with volumetric shrinkage due to the phase transformation.

Attention is being specifically paid to volumetric shrinkage since it is one of the main causes of macro-defects in castings and there is still no adequate computer model which would reliably predict these defects. Fluid flow is inlcuded into consideration to investigate its influence on the defect formation.

The proposed mathematical model will be able to predict the formation of casting defects due to shrinkage of the solidifying material involving the formation of *macro*-cavities. The work is based on a proprietary general purpose software *FLOW-3D*[1]. The code is capable of modelling the first four items listed immediately above though some improvements in the heat transfer and solidification models have been made as a result of the present work.

Numerical modelling of mould filling and solidification can be divided into four major parts:

- fluid flow, involving discretisation and simultaneous solution of the continuity and Navier-Stokes equations;
- free surface tracking which concerns mainly the filling stage when there are multiple and essentially transient free surfaces;

[1] The code is a property of *FLOW SCIENCE Inc.*, Los Alamos, U.S.A.

- heat conduction and heat transfer during and after filling; here the major computational effort goes into resolving the metal/mould interface, given that almost all the heat from the metal is lost to the mould. Besides, if the metal and mould thermal properties differ greatly, then steep temperature gradients are expected near the interface which also need to be accurately accounted for;
- solidification during and after filling; here attention is paid to the way the latent heat is accounted for.

Fluid flow modelling is far more difficult than heat transfer calculation since fluid motion requires a solution of three momentum equations (in three dimensions) and a continuity equations which are all coupled, even though in terms of the real time filling takes only a small fraction of the whole casting process. Nevetherless it is widely accepted that filling affects the subsequent solidification and possible defect formation because [2]

- heat transfer during mould filling affects the temperature distribution in the casting after the filling. This is especially true for thin wall castings where a great deal of the superheat may be removed during filling [2]. Without going through the mould filling process, it is impossible to ascertain correct initial conditions for subsequent solidification analysis;
- fluid flow modelling of the liquid melt plays a critical role in optimising the design of a casting system which consists of a mould, sprue, runner and ingates, in order to eliminate mould erosion and porosity problems. It is important to know how the filling proceeds, *i.e.* whether the metal splashes or any premature freezing occurs. Cold shuts and misruns can only be avoided by knowing the detailed flow field and heat transfer characteristics of the liquid melt during the mould filling process. Large castings, though, in which filling takes only a small fraction of the whole time, may be less sensitive to the initial fluid flow.
- fluid flow and heat transfer have a profound effects on grain size, macro- and micro-porosity and segregation of alloying elements of a casting. Fluid flow occurs during solidification after the mould is full due to residual circulation, thermal convection and feeding.
- foreign inclusions, oxide films, introduced during filling stage, may also be a cause of a poor quality casting.

The shrinkage formation model developed in this work is based on full Navier-Stokes and continuity equations coupled with a solidification model. Pressure and velocities are dependent on the volumetric changes occurring during the phase transformation. The model would allow one to simulate the effects of residual circulation and/or thermal convection on the solidification and to predict velocities in the feeding flow. The use of the full fluid flow equations makes the model most general, *e.g.* volumetric *expansion* due to remelting can also be easily handled. The numerical implementation of the model is based on the VOF method which uses the fluid fraction function to represent free surfaces and voids.

1.3 Thesis Overview

A brief overview of the numerical methods for solving Navier-Stokes equations with free moving boundaries, heat transfer and phase transformation is given in Chapter 2.

The outline of the general numerical method, including equation discretisation, free surface representation and the solution procedure, is given in Chapter 3. Those are the basis of FLOW-3D and form the framework in which the numerical modelling is carried out. The latter includes modifications of the metal/mould heat transfer algorithm which improved the accuracy of the calculations and enhancements to the solidification model, presented in Chapter 4.

Chapter 5 contains mathematical and numerical equations for two shrinkage models: a full model (M1) which is based on Navier-Stokes equations, and a simplified model (M2) which employs the energy conservation equation and does not describe fluid flow during solidification.

Simulations results are presented in Chapter 6 and are compared with the results of experimental castings available from literature. A detailed discussion of the results and numerical problems encountered during calculations is given in Chapter 7.

Finally, Chapter 8 contains conclusions and recommendations for further work on mathematical and numerical modelling.

Chapter 2

Overview of the Mathematical and Numerical Modelling

2.1 Governing Equations

We shall be dealing with the flow, heat transfer and phase change of gases and liquids (melts) commonly termed as fluids. The Eulerian description of the evolution of fluid at each point in time and space of the physical space will be used rather than the Lagrangian method where the evolution of individual physical fluid volumes is described. The main difference between the two methods is that an Eulerian coordinate system is fixed in the physical space while a Lagrangian coordinate system is attached to the fluid particles and moves and deforms with the fluid [3].

2.1.1 Conservation Equations

The Eulerian hydrodynamical equations describing a combined fluid and heat flow can be written in the following general differential form[1], *e.g.* [3,4]:

Continuity Equation

$$\frac{\partial \rho}{\partial t} = -\nabla \cdot (\rho \mathrm{v}) \qquad (2.1)$$

[1] That is assuming that all functions in these equations are continuous and have continuous first derivatives. In a more general case of a flow with discontinuities the integral form of the conservation equations will have to be used [3].

Momentum Equation - Newton's Second Law

$$\frac{\partial \mathbf{v}}{\partial t} = -(\mathbf{v}\cdot\nabla)\mathbf{v} + \frac{1}{\rho}\nabla\cdot\hat{\mathbf{P}} + \frac{1}{\rho}\mathbf{F} \tag{2.2}$$

conservation of energy

$$\frac{\partial E}{\partial t} = -(\mathbf{v}\cdot\nabla)E + \frac{1}{\rho}(\hat{\mathbf{P}}\cdot\nabla)\cdot\mathbf{v} + \frac{dq^e}{dt} \tag{2.3}$$

where

ρ is the fluid density,

$\mathbf{v}$ – the fluid velocity,

$\hat{\mathbf{P}}$ – the stress tensor,

E – the fluid internal energy per unit mass and

dq^e/dt – the external heat flux.

Cartesian coordinate system will be used throughout the thesis.

The Continuity Equation, Eq. (2.1), states that a change of the density in a fluid element is equal to the total flux of the fluid advected (or convected) into the element and it is also a statement of the conservation of mass.

Momentum Equation, Eq. (2.2), states that the fluid momentum in a fluid element changes due to the advection of the momentum into the element, action of external *surface* forces $\hat{\mathbf{P}}$, and of external *body* forces $\mathbf{F}$.

A fluid is called *viscous* if the stress tensor $\hat{\mathbf{P}}$ can be written as

$$P_{ij} = -p\delta_{ij} + \tau_{ij}, \qquad i,j = 1,2,3 \tag{2.4}$$

where δ_{ij} is the Kronecker delta and

$$p = p(\rho, T)$$

$$\tau_{ij} = \tau_{ij}(e_{kl}, T)$$

where

$$e_{ij} = \frac{1}{2}\left(\frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j}\right)$$

is the shear rate tensor; p the normal stress, or pressure, and $\hat{\tau}$ the shear stress tensor. In general, p and $\hat{\tau}$ may depend on some other physical parameters, *e.g.* temperature and phase concentrations, but we are not considering these complications here.

The dependence of $\hat{\tau}$ on $\hat{\mathrm{e}}$ defines the viscous model of the fluid. If the values of e_{ij} are small then τ_{ij} can be expanded by Taylor series about $e_{ij} = 0$ and retaining only linear terms. In this case the shear stress is a linear function of the shear rate

$$\tau_{ij} = B^{ijkl} e_{kl}$$

and coefficients B^{ijkl} are dependent on temperature and other physical parameters. All gases and simple liquids, molten metals and slags obey the linear viscosity model which are termed Newtonian fluids.

Assuming further that the fluid is isotropic (that is, its properties at a point are independent of the geometrical direction), components of tensor $\hat{\mathbf{B}}$ are reduced to only two independent parameters and the final form of the stress tensor $\hat{\mathbf{P}}$ is

$$P_{ij} = \delta_{ij}(-p + \lambda\, div\mathrm{v}) + \mu\left(\frac{\partial v_j}{\partial x_i} + \frac{\partial v_i}{\partial x_j}\right) \tag{2.5}$$

where λ and μ are constant and positive coefficients, μ is the fluid dynamic viscosity coefficient and $\eta = \lambda + \frac{2}{3}\mu$ is the second viscosity coefficient. Momentum Eq. (2.2) for constant λ and μ can be written as

$$\frac{\partial \mathrm{v}}{\partial t} = -\mathrm{v}\cdot\nabla\mathrm{v} - \frac{1}{\rho}\nabla p + \frac{(\lambda+\mu)}{\rho}\nabla(\nabla\cdot\mathrm{v}) + \nu\Delta\mathrm{v} + \frac{1}{\rho}\mathbf{F} \tag{2.6}$$

with $\nu = \mu/\rho$ being the fluid kinematic viscosity. Eq. (2.6) is called the Navier-Stokes equation.

Eqs. (2.1) and (2.2) may be simplified further by assuming the incompressibility of the fluid, *i.e.*

$$\rho = \rho_0 = const \tag{2.7}$$

Then $\nabla\cdot\mathrm{v} = 0$ in Eqs. (2.5) and (2.6).

The density of the fluid does change due to thermal contraction/expansion (linear contraction) which results in thermal, or natural, convection in the fluid in the presence of body forces. The thermally induced flow can be an important factor, for example, in redistributing heat, and can be taken into account by the use of the Boussinesq approximation:

$$\rho = \rho_0 + \delta\rho, \qquad \frac{\delta\rho}{\rho_0} \ll 1$$

Then $\rho = \rho_0$ in all the terms of Eq. (2.6) except for the pressure gradient term, where variable density is retained. A simple linear dependence of $\delta\rho$ on temperature can be used to complete the model:

$$\delta\rho = -\beta\,\rho_0\,(T - T_0)$$

where T_0 is the reference temperature and

$$\beta = \frac{1}{V}\left(\frac{\partial V}{\partial T}\right)_p$$

is a constant positive expansion coefficient.

The Energy Conservation Eq. (2.3) states that a change of the internal energy E in a fluid element is the result of the advection of E into and out of the element, work due to the internal surface forces, $-\frac{1}{\rho}(\hat{\mathbf{P}} \cdot \nabla) \cdot \mathbf{v}$, and a flux of the external heat dq^e/dt. Using the Continuity Equation and Eq. (2.4) we can write

$$\frac{1}{\rho}(\hat{\mathbf{P}} \cdot \nabla) \cdot \mathbf{v} = -p\frac{d1/\rho}{dt} + \frac{1}{\rho}\tau_{ij}e_{ij} \tag{2.8}$$

The second term on the righthand side of Eq. (2.8) constitutes the heating of the fluid due to the viscous friction.

We will assume that viscous heating is negligibly small compared with the advection and the external heat flux terms in Eq. (2.3). This is true for all fluids for the range of flows considered in this work. Viscous friction may become important in high pressure die-casting flows [5].

External heat flux can be written in terms of a vector quantity using simple geometrical considerations:

$$\frac{dq^e}{dt} = -\frac{1}{\rho}div\,\mathbf{q} \tag{2.9}$$

According to Fourier's law, which is derived similarly to the linear viscous fluid model [3], the heat flux is expressed as a linear function of the the temperature gradient in the fluid:

$$\mathbf{q} = -k\nabla T \tag{2.10}$$

where k is the fluid thermal conductivity; it is positive and it may be a function of temperature and other physical parameters.

Eq. (2.3) can be rewritten now as

$$\frac{\partial E}{\partial t} = -(\mathrm{v}\cdot\nabla)E - p\frac{(d1/\rho)}{dt} + \frac{1}{\rho}\nabla\cdot(k\nabla T) \tag{2.11}$$

The main assumption concerning thermodynamical processes during solidification is that the system is always in the state of thermodynamical equilibrium. This assumption allows us to use the laws of the equilibrium thermodynamics and operate with thermodynamical functions of state, which in turn are functions of equilibrium parameters such as temperature, pressure and density.

Strictly speaking a system can reach the equilibrium state only if in does not interact with other systems and is given sufficient time to allow the system to be in the equilibrium at each intermediate point of the process. In reality the interaction processes, such as metal-mould heat flow, occur at finite speeds. In that case we can define a *local equilibrium* so that each infinitesimally small volume of the system is at equilibrium with its own temperature and pressure, though the system as a whole is not. This assumption is valid if heat fluxes across these volumes are not too large. According to Fourier's law, Eq. (2.10), the fluxes are small if temperature gradients are small[2].

Large velocities during filling stages can lead to the development of turbulence in the metal flow. Reynolds number in such casting processes as gravity and pressure die castings can reach values of 10^5 implying that turbulence levels may be very high. Turbulence in the metal affects velocity distribution in the flow, filling time, free surface motion and heat transfer [7].

One of the frequently used approaches to the mathematical modelling of turbulence is a semi-empirical analysis of turbulent mean quantities, in which only the gross properties of a turbulent flow, such as mean velocity and temperature profiles, are described [134]. Interest in turbulent fluctuations is mainly confined to relating certain turblent-shear correlations, such as $\overline{u'v'}$, to the properties of the mean flow. It is not in the scope of this work to cover the derivation of the turbulence models. Comprehensive descriptions of those can be found in references [134,10]. We will only mention here two of the most widely used 'mean' turbulence models: $k-\epsilon$ and RNG $k-\epsilon$ models. These models reduce turbulence variables to two independent mean quantities: average kinetic energy of turbulent fluctuations, k, and its dissipation, ϵ

$$k = \frac{1}{2}\overline{(u'^2 + v'^2 + w'^2)}$$

[2] It should be noted here that in many industrial situations solidification proceeds so fast that even a local equilibrium cannot be assumed.

$$\epsilon = \nu \overline{\left(\frac{\partial u_i'}{\partial x_j} + \frac{\partial u_j'}{\partial x_i}\right) \frac{\partial u_j'}{\partial x_i}}$$

where ' denotes turbulent fluctuation quantities. A transport equation is derived for each of k and ϵ and turbulence is taken into accout in Navier-Stokes equations by using an effective viscosity coefficient ν_{eff}

$$\nu_{eff} = \nu_0 + \nu_T$$

where ν_0 is the molecular value of the kinematic viscosity and ν_T the turbulent viscosity

$$\nu_T = C_\nu \frac{k^2}{\epsilon}$$

In the $k - \epsilon$ model typically $C_\nu = 0.09$, as derived from experimental studies, while in the RNG $k - \epsilon$ model C_ν is a function of local shear rates. In the latter case the model is more accurate and suitable for metal flow conditions in complicated runner and mould systems of typical industrial castings.

2.1.2 Solidification Modelling

When modelling solidification it is useful to introduce, in addition to the internal energy E, two other functions of state, the specific enthalpy H

$$H = E + \frac{p}{\rho}, \qquad dH = TdS + \frac{dp}{\rho} \tag{2.12}$$

and the Gibbs free energy G

$$G = H - TS, \qquad dG = -SdT + \frac{dp}{\rho} \tag{2.13}$$

where S is the specific entropy and $dq^e = TdS$ [3,6].

Liquid to solid phase transformation, or solidification, is a sharp change of state from the disordered atomic arrangement of the liquid to the ordered arrangement of the crystal, at a single temperature (or melting point). The effect of pressure on solidification can usually be neglected (except at high pressures). Solids and liquids are only slightly compressed by pressures of order 1 atm. and their enthalpy is then hardly affected. It is readily seen then from Eq. (2.13) that the Gibbs free energy is not affected by equilibrium solidification. If dm is an amount of the solidified liquid in an element of fluid, then for this element

$$dG_{dm} = dH_{dm} - TdS_{dm} = 0$$

or

$$dH_{dm} = TdS_{dm} \tag{2.14}$$

where dH_{dm} is the heat given to or taken from the surroundings of the element, called latent heat. Eq. (2.14) shows that though phase transformation occurs isothermally, it is accompanied by heat flow leading to a change in the enthalpy. The latent heat of solidification L per unit mass is

$$L = \frac{dH_{dm}}{dm} \tag{2.15}$$

The latent heat of melting can be thought of as the energy required to pull the atoms apart to the more openly packed structure of the liquid.

It follows from Eqs. (2.11) and (2.12) that the fluid specific heat at a constant pressure, C_p, is needed to evaluate the enthalpy:

$$H = \int_{T_0}^{T} C_p dT \tag{2.16}$$

while for the internal energy the specific heat at a constant volume, C_V, is required:

$$E = \int_{T_0}^{T} C_V dT$$

Obviously, for liquids and solids, it is much easier to measure C_p than C_V, and the advantage of using the enthalpy instead of the internal energy is apparent. Taking into account the latent heat we have:

$$H = \int_{T_0}^{T} C_p dT + \Delta H, \qquad \Delta H = (1 - f_s) L \tag{2.17}$$

where f_s is the mass fraction of solid per unit mass of fluid.

The fraction of solid function f_s has to be defined so that Eq. (2.17) can be solved for temperature. If solidification is assumed to be in equilibrium, then for pure metals it will proceed at a constant temperature T_m and fraction of solid is uniquely defined by the enthalpy. For alloys the equilibrium occurs over a range of temperatures. The lever rule for binary alloys then relates the initial alloy composition, c_0, and the current values of compositions of the liquid, c_l, and solid, c_s, to f_s [6,7]:

$$f_s = \frac{c_0 - c_l}{c_s - c_l} \tag{2.18}$$

If the phase diagram of the alloy is known then compositions c_l and c_s can be found from the diagram given a value of the current temperature.

There are several factors, however, that may complicate this procedure.

For the phase transformation to proceed in the direction of freezing, dG has to be positive to create a potential for the process [6,8]:

$$dG = dG_{V,sl} + dW > 0 \tag{2.19}$$

According to Eq. (2.13), to achieve a positive change of the Gibbs free energy, $dG > 0$, at a constant pressure, $dp = 0$, the liquid has to be cooled below its melting point, $dT < 0$ (since the entropy is positive, $S > 0$). $dG_{V,sl}$ is the volumetric free energy which is the difference in the free energies of the solid and liquid phases of volume V. $dG_{V,sl} < 0$ for $T < T_m$, due to the difference in the entropies of the two phases. $dW > 0$ is the free energy of the atoms at the liquid/solid interface or transitional region and can be associated with the work that has to be done to make a nucleus. The actual value of dW depends on the properties of the metal atoms as well as the interface curvature.

When a new phase is being formed in the system, this phase often appears first as small nuclei in the old phase, which then grow by the addition of more material from the old phase. The surface energy dominates the nucleation process because of the small size of the nuclei and its high surface curvature. Experimental studies show that in a pure metal free from inclusions and away from the walls a substantial undercooling is required to promote solidification – approximately $0.2T_m$, *e.g.* 295°K for pure iron [6].

Once an undercooled droplet is nucleated, freezing is then very rapid. This differs from the casting of metals under practical conditions in a foundry where nucleation occurs at temperatures within 10°K of the ideal freezing point due to the presence of foreign inclusions and mould walls, which serve as nucleation centres. The rate of growth of the solid on the nucleus in this case is much smaller, as it is controlled by the rate of removal of the latent heat released at the solid-liquid interface.

Recently a number of 'macro-micro' models have been proposed to predict the microstructural evolution of a casting [11]-[16]. Comparisons of the predicted grain structures, using a rather sophisticated probabilistic model, with the experimental results for dendritic and eutectic alloys, given by Gandin *et al* [17] are very impressive. However, the main difficulty in applying this model in a general case is the requirement of a very fine spatial resolution: the maximum computational cell size for 2-D and 3-D simulations was 10 μm which is unacceptable for a 3-D industrial scale fluid flow and solidification modeling. Besides, modelling of the solidification kinetics requires additional input parameters, *e.g.* the relationship between the undercooling ΔT and the number of grains $n_{\Delta T}$ in different parts of the casting. These are either defined from some assumptions and/or obtained empirically. The latter makes kinetics models case-dependent and complicates its validation. Besides, it is difficult to take into account the influence of the metal flow on the grain evolution, the importance of which has been shown by Ortega [18].

In alloy systems solidification speed may be comparable to the rate of solute diffusion and advection. In this case the distribution of the solute in both solid and liquid phase will be non-uniform and this will affect the local temperature of solidification. To model these phenomena the solute transport equation will need to be solved for each phase [19]

$$\frac{\partial c}{\partial t} + (\nabla \cdot \mathrm{v})\, c = \nabla \cdot (D \nabla c) \qquad (2.20)$$

where c is the solute concentration and D the diffusion coefficient. Given the values of the composition and temperature the value of the solid fraction can be found from the alloy phase diagram [7].

2.1.3 Boundary Conditions

The set of hydrodynamical equations for an incompressible Newtonian fluid, using Eqs. (2.1), (2.6), (2.7), (2.11) and (2.12), is

$$div\, \mathrm{v} = 0 \qquad (2.21)$$

$$\frac{\partial \mathrm{v}}{\partial t} = -\mathrm{v} \cdot \nabla \mathrm{v} - \frac{1}{\rho_0} \nabla p + \nu \Delta \mathrm{v} + \frac{1}{\rho_0} \mathbf{F} \qquad (2.22)$$

$$\frac{\partial H}{\partial t} = -(\mathrm{v} \cdot \nabla) H + \frac{1}{\rho_0} \nabla \cdot (k \nabla T) \qquad (2.23)$$

Elliptic Eq. (2.21), parabolic Eqs. (2.22), (2.23) and Eq. (2.17) are a system of non-linear, coupled second order partial differential equations written in terms of the fluid velocity, pressure and temperature as the unknown variables. Only one boundary condition for the velocity and for the temperature at each fixed boundary will be sufficient to define the solution of the these equations; the conditions can be either of Dirichlet type (a specified velocity/temperature value) or of Neumann type (a specified value of the velocity gradient/heat flux normal to the boundary) or a linear combination of both [20,21]. Velocity, pressure and temperature are then defined uniquely away from the boundaries given the appropriate initial conditions.

Boundary conditions specific to the viscous fluid model are kinematic no-slip (Dirichlet) conditions at the walls

$$\mathrm{v} = 0 \qquad (2.24)$$

In the presence of a free surface, which is defined as a boundary between two fluids with very different densities (for example, for water and air $\rho_w/\rho_a \approx 1000$), the lighter fluid (gas) can be substituted by a constant pressure boundary condition at the free surface. The exclusion of the gas phase from consideration poses an additional problem of setting the boundary condition at the generally transient free surface.

The kinematic boundary condition at the free surface is

$$\frac{\partial l}{\partial t} + (\mathrm{v} \cdot \nabla) l = 0 \tag{2.25}$$

which states that an element of the free surface l moves with the fluid.

Dynamic boundary conditions at the free surface are:

- pressure in the fluid p_{fs} results from the sum of the gas pressure, surface tension force and viscous stresses and can be written as

$$p_{fs} = p_g + \sigma\zeta + 2\mu\frac{\partial v_n}{\partial n} \tag{2.26}$$

where

σ is the surface tension coefficient,

ζ is the surface curvature ($\zeta > 0$ when the fluid surface is convex),

$\mathbf{n}$ is the surface unit normal pointing in the fluid-gas direction and

$v_n = \mathbf{v} \cdot \mathbf{n}$;

Zero flux of the tangential momentum is assumed through the free surface. For tangential viscous stress this gives

$$\hat{\tau}_\tau = 0 \tag{2.27}$$

where the subscript τ indicates the projection onto the plane tangent to the free surface. This assumption is well justified for casting problems due to generally large Reynolds numbers in the liquid metal flow. An example of a case when tangent stresses are important is the interaction between ocean surface waters and atmospheric winds.

Heat flow boundary conditions at the walls and at the free surface can be specified in terms of the heat flux $\mathbf{q}_{bc}$ per unit area. The same expression can be used in both cases, that is:

$$\mathbf{q}_{bc} = h\,(T_f - T_b) \cdot \mathbf{n} \tag{2.28}$$

where

T_f is the fluid temperature at the interface,

T_b is the boundary temperature which is either the interfacial mould temperature or the ambient temperature of the air,

h is the heat transfer coefficient which characterises the thermal resistance of the interface and

$\mathbf{n}$ is the interface unit normal.

The value of h varies with temperature and geometrical properties of the interface, such as the wall surface finish and the size of an air gap between the metal and the mould. At the free surface h is defined by the convection conditions in the gas, such as turbulence intensity, whether it is natural or forced and the free surface orientation. Radiation heat transfer coefficient is strongly temperature dependent [22] and is influenced by the presence of other radiating surfaces. Ludley and Szekely showed that for a molten steel held in a ladle the heat loss from the unprotected surface was of the order of 20 to 30% of that lost to the ladle walls by conduction [23].

There are few cases for which analytical expressions for h are available [22] and normally experimentally defined values are used [24].

Values of thermo-physical properties of metals and mould materials, *e.g.* specific heat and thermal conductivity, are required to solve Eqs. (2.21)-(2.23). These are usually temperature dependent and must be obtained by experiments. Besides, for metals all properties change sharply during the phase transformation [6]. Often it is sufficient to take this change into account by assuming the properties of each individual phase to be constant.

2.2 Overview of the Numerical Methods

Numerical methods have to be used to solve Eqs. (2.21)-(2.23) since these are non-linear, transient, coupled second-order equations. There are four widely adopted basic approaches to obtain a numerical solution of these equations: the finite difference, finite element, the finite volume and the boundary element method [25].

A. Finite Difference Method.

The finite difference method (FDM) is based on the properties of Taylor expansions and on the straight-forward application of the definition of derivatives. It is the oldest of the methods applied to obtain numerical solutions of differential equations, and the first application is considered to have been developed by Euler in 1768. The idea of finite difference methods is quite simple, since it corresponds to an estimation of a derivative by the ratio of two differences according to the definition of the derivative.

For a function $u(x)$ the derivative at a point x can be approximated, for example, by forward differencing:

$$\frac{\partial u}{\partial x} = \frac{u(x + \Delta x) - u(x)}{\Delta x} + O(\Delta x)$$

indicating that the truncation error $O(x)$ goes to zero as the first power of Δx.

There is a variety of finite difference approximations for the derivatives of functions [26,27]. The idea behind obtaining higher-order approximations as well as approximating higher-order derivatives lies in Taylor series expansion of the unknown function around values at grid points.

Most finite difference models used in solidification simulation have employed regular spaced rectangular meshes, which are not particularly suitable for representing complex shaped castings, although curved surfaces can be approximated by using a fine node spacing and step approach [28] (see also FAVOR method in Section 3.3.2).

For curvilinear meshes the discretization of the equations can be performed after a transformation from the physical space (x, y, z) to a Cartesian, computational space (ξ, η, ζ) [29,30]. This method of *body fitted coordinates* combines the flexibility of the finite element domain discretisation and the simplicity of the finite difference equation approximations [31,32].

Current development of the FDM is in multi-block gridding and general unctructured grid approaches which are aimed at overcoming the traditional inefficiency of the method in resolving arbitrarily shaped boundaries [50,121].

B. Finite Volume Method.

This is the technique in which the integral formulation of the conservation laws are discretized directly in the physical space. The finite volume method (FVM) takes advantage of an arbitrary mesh, where large number of options are open for the definition of the control volumes around which the conservation laws are expressed.

The method uses the conservation equations in their general integral form. For a scalar quantity U, with volume sources Q, and a flux vector $\mathbf{F}$, which contains only convective contributions, the integral equation for a discrete volume Ω with boundary S is given by

$$\frac{\partial}{\partial t}\int_{\Omega} U\,dt + \int_{S} \mathbf{F}\cdot d\mathbf{S} = \int_{\Omega} Q\,d\Omega \tag{2.29}$$

The essential significance of this formulation lies in the presence of the surface integral and the fact that the time variation of U inside the volume only depends on the surface values of the fluxes. The advantage, especially in the absence of source terms, is that the fluxes are calculated only on two-dimensional surfaces. Eq. (2.29) is replaced by the discrete form

$$\frac{\partial}{\partial t}(U_J\Omega_J) + \sum_{sides}(\mathbf{F}\cdot\mathbf{S}) = Q_J\Omega_J \tag{2.30}$$

index J indicates that the quantity is averaged over the volume Ω_J.

The following constrains on the choice of Ω_J volumes for a conservative control volume method have to be satisfied:

- their sum should cover the whole domain Ω;
- adjacent Ω_J may overlap if each internal surface Γ_J is common to two volumes;
- fluxes along a cell surface have to be computed by formulas independent of the cell in which they are considered.

The last requirement ensures that the conservative property is satisfied.

C. Finite Element Method.

The finite element method (FEM) originated from the field of structural analysis as a result of research carried out mainly between 1940 and 1960. The concept of 'elements' can be traced back to the techniques used in stress calculations, whereby a structure was subdivided into small substructures of various shapes and re-assembled after each 'element' had been analysed. The development of this technique and its formal elaboration led to the introduction of what is now called the finite element method by Turner *et al* [33] in a 1956 paper dealing with the properties of a triangular element in plane stress problems.

After its successful applications to structural mechanics it soon appeared that the method could also be used to solve continuous field problems [34] including solidification [35].

The main advantage of the finite element method is the ease with which complex geometries can be handled and the wall boundary conditions can be set more accurately. The flexibility regarding node size and shape means that curved surfaces can be accurately meshed.

However, the process of meshing is still an art, requiring an understanding of the problem and of the finite element method itself. In an attempt to simplify the meshing process interest has recently focussed on automeshing, combining Computer Aided Design (CAD) packages with finite element programs [36]. Lewis *et al* [37] have demonstrated how adaptive meshing can be used advantageously during the actual solidification calculation. A fine node spacing is employed only in those regions of the casting which are in the mushy zone.

Apart from setting the complex task of generating the best mesh, finite element method has other disadvantages. Generally, it requires more computer time for calculations than the FDM. Besides, any physical problem written in finite element formulation loses its clarity, the physical nature of the initial equations becomes obscured by the mathematical symbolism. Despite this and supported by successful developments in computer technology, the finite element method is widely used at present, both in commercial and scientific programming [38]-[40]. It has been successfully applied to model casting process in 2-D [35,41] and 3-D [42,43] and commercial 3-D FEM packages, such as *ProCAST* [44,45], have also been marketed.

D. Boundary Element Method.

An alternative to the more familiar finite element, finite difference and control volume approaches is the boundary element method (BEM). With this method, the differential equations describing the flow are recast into integral representations so that the solution away from the boundaries is written in terms of the boundary values. The solution of the flow problem is then based on the numerical quadrature of the integrals. It has been shown that this method can be successfully applied to phase transformation and free surface problems with moving boundaries, as well as to general viscous flows [25,39,46,47]. Detailed description of the boundary element method can be found in these references.

2.3 Discretisation of the Equations of Motion

The finite volume approach in structured grids will be used in this work to solve the fluid motion and energy Eqs. (2.21)-(2.23). The choice is dictated by the use of a commercial CFD code FLOW-3D which employs this method (see Chapter 3).

Velocity components and pressure are dependent variables (primitive variables) in the Continuity and Navier-Stokes equations. A specific feature of these equations is that they do not contain the term $\partial p/\partial t$, in other words pressure is an implicit variable in an incompressible flow [35]. At each time step, velocity components and pressure in the mesh cells have to be found simultaneously (due to the incompressibility of the fluid) by solving the discretised numerical equations.

The usual way of obtaining a numerical solution is to use a predictor-corrector scheme [25,27]. Some other methods of solving the incompressible flow equations, for example the *artificial compressibility* method which allows a conservation-like equation for pressure to be written, are also given by Roache [27]. Let us assume a first order forward-differencing for the time derivatives with the time step Δt, as is commonly used irrespective of the space-discretisation method. At time t^{n+1} new, first-guess, velocities are predicted using Navier-Stokes equations (in two dimensions for simplicity):

$$u_0^{n+1} = u^n + \Delta t \cdot (-D_a - D_p^n + D_F + D_\mu)_x \tag{2.31}$$

$$v_0^{n+1} = v^n + \Delta t \cdot (-D_a - D_p^n + D_F + D_\mu)_y \tag{2.32}$$

where D_a, D_p, D_F and D_μ are the discretised advection, pressure, body forces and viscous terms, respectively. Pressure in Eqs. (2.31), (2.32) is taken at the old time level t_n. Newly obtained velocity will not, in general, satisfy the discretised continuity equation. Index '0' means that u_0^{n+1} and v_0^{n+1} are used as first guess velocities in the subsequent pressure correction procedure. In the Marker-And-Cell (MAC) method [48] the incompressibility constraint is imposed by adjusting the cell pressures. For example, if the divergence of a cell is negative, corresponding to a net flow of mass into the cell, the cell pressure is increased to eliminate the inflow. Because there is one pressure variable for each cell, the divergence for each cell can be driven to zero in this way. The pressure and velocity adjustment must be done iteratively as updating velocities in one cell will upset the mass balance in its neighbours. Iterations proceed until the velocity divergence in each cell is below some predetermined parameter ϵ (see Section 2.3.2).

2.3.1 Advection and Diffusion Terms Approximations

The MAC method is distinguished by the form of the primitive-variable equations used, the differencing scheme, the cell structure, and the use of "marker particles".

It is advantageous, when using the primitive variables, to employ a staggered grid, in which pressure (and all other scalar variables) is located at the cell centres and velocities are located at the cell faces [48] (Fig. 2.1*a*). The differencing scheme for the momentum equations employed in the MAC method is the forward-time, centred-spaced method. For a cell (i,j) in a uniformly spaced two-dimensional grid at time level t^{n+1}

$$\begin{aligned} \frac{\partial u}{\partial t}\Big|_{i,j}^{n+1} &= \frac{u_{i,j}^{n+1} - u_{i,j}^{n}}{\Delta t} \\ D_{p,x,i,j}^{n} &= \frac{p_{i+1,j}^{n} - p_{i,j}^{n}}{\Delta x} \end{aligned}$$

and

$$D_{\mu,x,i,j}^{n} = \nu\left[\frac{u_{i+1,j}^{n} - 2u_{i,j}^{n} + u_{i-1,j}^{n}}{\Delta x^2} + \frac{u_{i,j+1}^{n} - 2u_{i,j}^{n} + u_{i,j-1}^{n}}{\Delta y^2}\right]$$

The evaluation of

$$D_{a,x,i,j}^{n} = \left(\frac{\partial u^2}{\partial x} + \frac{\partial vu}{\partial y}\right)\Big|_{i,j}^{n} = \frac{u_R^2 - u_L^2}{\Delta x} + \frac{(vu)_T - (vu)_B}{\Delta y}$$

where u_R and u_L are given by Eq. (2.35) below and

$$\begin{aligned} (uv)_T &= \frac{1}{4}\left(v_{i,j} + v_{i,j+1}\right)\left(u_{i,j} + u_{i,j+1}\right) \\ (uv)_B &= \frac{1}{4}\left(v_{i,j-1} + v_{i,j}\right)\left(u_{i,j-1} + u_{i,j}\right) \end{aligned}$$

$D_{p,x,i,j}$, $D_{\mu,x,i,j}$ and $D_{a,x,i,j}$ are discretised x-direction components of the pressure gradient, shear stress and advection, respectively, at node (i,j).

Numerical stability analysis shows that the MAC method is unconditionally unstable for inviscid flows and requires a non-zero viscosity to maintain stability [27]. Moreover, central differencing for the advection and viscous terms introduces a limitation to the cell Reynolds number which for a one-dimensional flow is

$$Re_c = \frac{u\Delta x}{\nu}, \qquad Re_c \leq 2 \tag{2.33}$$

The control volume, upwind differencing approach to solving Eqs. (2.17), (2.21)-(2.23) is widely used [5,49,50,51]. The upwind differencing scheme was first used by

Gentry *et al* [52] and Hirt *et al* developed it for incompressible flows and named the method SOLA (SOLution Algorithm, see Section 2.3.2) [53].

Fig. 2.1c shows a computational cell and its control volume (CV) for calculating the x-direction momentum fluxes. Assuming for simplicity a uniform mesh spacing Δx, at position (i,j) we have:

$$D_{a,x,i,j} = \frac{u_R u_{i,j} - u_L u_{i-1,j}}{\Delta x} + \frac{v_T u_{i,j} - v_B u_{i,j-1}}{\Delta y}, \quad if \quad u_R, u_L, v_T, v_B > 0 \tag{2.34}$$

where

$$u_R = \frac{1}{2}(u_{i,j} + u_{i+1,j}), \quad u_L = \frac{1}{2}(u_{i,j} + u_{i-1,j}) \tag{2.35}$$

and

$$v_T = \frac{1}{2}(v_{i,j} + v_{i-1,j}), \quad v_B = \frac{1}{2}(v_{i,j-1} + v_{i-1,j-1}) \tag{2.36}$$

are the estimates by averaging of the velocities at the control volume boundaries. Viscous terms are discretised by central differencing, in the same way as in MAC method.

The upwind differencing for the advection terms removes the cell Reynolds number restriction imposed by Eq. (2.33) for viscous flows and is conditionally stable for inviscid fluids [27]. The essence of one-sided differencing scheme given by Eqs. (2.34)-(2.36) is to “advect” the value of a variable (velocity) taken upstream from the considered node. Sometimes the method is also called the *donor cell* method. If for example u_R in Eq. (2.34) is negative then $u_{i+1,j}$, instead of $u_{i,j,}$, is used as the quantity advected through the right boundary of the control volume.

Advantages of the upwind method in this formulation are [27]

- it possesses the *transportive* property which means that the effect of a perturbation is only advected in the direction of the velocity. All methods which use centred-space derivatives for the advection terms do not possess this property and filters have to be designed to avoid spurious solutions.
- the method is *conservative*, *i.e.* it preserves the fluid momentum.
- compared with the central differencing method, the upwind method is not stability limited by cell Reynolds number. Vanka [54] used the advantages of both schemes by applying the upwind method only in the cells which had $Re_c \geq 2$ and the central differencing for all other cells, thus obtaining a “hybrid” method.

Roache [27] mentions a surprising agreement of the upwind "donor cell" method with some second-order calculations of the driven cavity problem. In fact the term "accuracy" of a numerical method is rather arbitrarily defined. In most cases it is understood in the formal Taylor series expansion sense. It is possible to represent spatial derivatives "more accurately" in this sense but not to retain for example the conservative property.

A disadvantage of the upwind method is that it introduces an artificial viscosity due to second order truncation errors introduced by the first order differencing for the advection terms.

$$\nu_x = \frac{1}{2}u\Delta x(1 - c_x)$$

and

$$\nu_y = \frac{1}{2}v\Delta y(1 - c_y)$$

where $c_x = \Delta t\, u/\Delta x$, $c_y = \Delta t\, v/\Delta y$ are the cell Courant numbers. The artificial viscosity effects would be minimised if c_x and c_y are close to unity which is hardly achievable in practical two- or three-dimensional flows.

This disadvantage of the first order upwind differencing makes it inapplicable for the advection of variables with sharp discontinuities in the flow region, such as free surfaces. Section 2.5 describes two numerical methods specially developed to preserve the discontinuities.

It has been shown that the "donor cell" upwind method demonstrate good comparisons with physical experiments at high Reynolds numbers [27]. In calculation of driven cavity flow, Torrance *et al* [55] have shown that the upwind differencing method applied to the conservative equations is considerably more accurate than second-order differencing applied to the non-conservative equations.

If advection and/or viscous terms on the left-hand side of Eqs. (2.31) and (2.32) are estimated at time t_{n+1}, *i.e.* implicitly, then these equations become coupled and have to be solved at all nodes simultaneously to obtain the first-guess velocities u_0^{n+1} and v_0^{n+1}. This is achieved by *inner* iterations on the velocities. The inner iterations usually take much less iterations to converge then the outer, *i.e.* pressure iterations.

Roache [27] suggests that ideally the advection terms should be approximated explicitly and the viscous terms implicitly. This would reflect the physical nature of these processes, that is the finite velocity of advection and the infinite velocity of diffusion[3].

[3] The infinite diffusion velocity means that if there is a local perturbation in a uniform distribution of a variable, then it will instantaneously affect *all* areas due to the diffusion.

In the present work all terms on the right-hand sides of Eqs. (2.31), (2.32), except for the pressure gradients, are approximated explicitly, that is at time t_n. In this case the solution matrix for u_0^{n+1} and v_0^{n+1} is diagonal and the first-guess velocities are easily calculated at each node. The explicitness introduces some limitations on the time-step size Δt which are discussed in Section 2.10.

As has already been mentioned, the upwind differencing method has first order formal accuracy. It is well known that higher order methods, both in space and time, pose a problem of resolving the boundary conditions without losing accuracy [27]. Often only first order equations can be used at the boundaries. The first order truncation errors propagate from the boundaries into the computational domain, depriving the solution of its higher-order accuracy. One of the specific features of casting is that the ratio of the internal mould walls area, at which the boundary conditions are set for both the heat and fluid flows, to the volume of the open-to-flow domain is large. This fact reduces the advantages of employing higher-order accurate discretisation methods for the heat and fluid flow equations and justifies the use of the first order numerical method.

Among all other numerical methods, the Leith's approximation scheme for the advection terms is worth of mention since it is fully explicit but is unconditionally stable [27]. It is also called the *semi – Lagrangian* method [56,57]. The details and an application of the method are given in Appendix B.

2.3.2 Iterative Solution Methods

Due to the incompressibility of the fluid pressure in the Navier-Stokes equations and velocity components in the Continuity Equation have to be approximated implicitily. This leads to the following set of algebraic equations (in two dimensions)

$$u_{i,j}^{n+1} = \hat{u}_{i,j}^{n} - \frac{1}{\rho_0}\frac{p_{i+1,j}^{n+1} - p_{i,j}^{n+1}}{\Delta x} \tag{2.37}$$

$$v_{i,j}^{n+1} = \hat{v}_{i,j}^{n} - \frac{1}{\rho_0}\frac{p_{i,j+1}^{n+1} - p_{i,j}^{n+1}}{\Delta y} \tag{2.38}$$

where $\hat{u}_{i,j}^{n}$ and $\hat{v}_{i,j}^{n}$ include terms approximated at time level n. Pressure gradient terms in Eqs. (2.37) and (2.38) are approximated using the cell-centred pressures as they lie exactly on the control volume boundaries (see Fig. 2.1*c*). If we choose a computational cell (i, j) of a staggered grid (see Fig. 2.1*b*) as the control volume for the Continuity Equation, then its discretised form at time t^{n+1} is

$$L_{i,j} = \frac{1}{\Delta x}(u_{i,j}^{n+1} - u_{i-1,j}^{n+1}) + \frac{1}{\Delta y}(v_{i,j}^{n+1} - v_{i,j-1}^{n+1}) = 0 \tag{2.39}$$

Eqs. (2.37)-(2.39) constitute a system of coupled linear equations of size $N \times M$ and an iteration method is usually emloyed to solve it since a direct matrix inversion is generally numerically inefficient.

The iteration procedure to bring the velocity divergence to zero in every cell and update pressures is a major part of the computational effort in obtaining numerical solution. Pressure iterations may take as much as 90% of the total CPU-time [5] and much attention is paid to its optimisation.

If L^l is the value of the divergence after l iterations, then [33]

$$\Delta p_{l+1} = -\frac{L^l}{2\frac{\Delta t}{\rho_0}\left(\frac{1}{\Delta x^2} + \frac{1}{\Delta y^2}\right)} \tag{2.40}$$

$$p_{l+1} = p_l + \Delta p_{l+1} \tag{2.41}$$

$$u_{i,j}^{l+1} = u_{i,j}^{l} + \frac{\Delta t}{\rho_0 \Delta x}\Delta p_{l+1} \tag{2.42}$$

$$u_{i-1,j}^{l+1} = u_{i-1,j}^{l} - \frac{\Delta t}{\rho_0 \Delta x}\Delta p_{l+1} \tag{2.43}$$

$$v_{i,j}^{l+1} = v_{i,j}^{l} + \frac{\Delta t}{\rho_0 \Delta y}\Delta p_{l+1} \tag{2.44}$$

$$v_{i,j-1}^{l+1} = v_{i,j-1}^{l} - \frac{\Delta t}{\rho_0 \Delta y}\Delta p_{l+1} \tag{2.45}$$

For each cell encountered, the divergence $L_{i,j}$ is computed using the most current velocity values available. The iterations continue until $\mid L \mid < \epsilon$. This technique is called Gauss-Seidel (GS) iteration method. The technique which uses only previous iteration values to compute the new values through the whole iteration cycle is called Jacobi method.

Lipinski *et al* [5] argue that one should use Jacobi type pressure iterations to assure symmetric results. When $L_{i,j}^l$ is calculated in a control volume using GS method and the mesh is swept in x-direction from $i = 1$ to $i = imax$, then $u_{i-1,j}$ has already been iterated in $i-1, j$ cell while $u_{i,j}$ has not. This introduces an asymmetry in symmetric flows. GS method *is* symmetric at the level of convergence, but when the results on the way to convergence are allowed to be non-symmetric problems can occur due to truncation errors eventually leading to a non-symmetric final solution.

The effect of this asymmetry could be reduced by alternating the direction of the mesh sweeping at different time steps.

However, no such problems have been encountered using GS method in the present work. This method has also the advantage of smaller computer memory required for storage, though Jacobi method would be easier to vectorise and/or parallelise.

The two iteration methods described above represent the point-by-point (PP) iteration procedure[4]. The main drawback of these iteration schemes is that the lowest wavelength error components damp most slowly. Thus, regardless of the initial error distribution these components will dominate for large iteration number l [27]. Frankel [58] showed that, for the most resistant error components, asymptotically l GS iterations are worth $2l$ Jacobi iterations. In other words the latter converges at half the speed of GS method [59].

The are a few variations of the above procedures aiming at achieving faster convergence.

In the *line – by – line* (LL) method a line of cells is chosen. It is assumed that pressures along the neighbouring lines are known from their "latest" values. The correction values for p's along the chosen line are calculated by directly solving the tridiagonal simultaneous linear equations using available efficient solvers.

In the *block – by – block* (BB) iteration method, developed by Wang [49], four cells, 2×2, are chosen as a group. Assuming again that p's in the surrounding cells are known from their "latest" values, the equations for the four pressures in the chosen cells are solved simultaneously by Cramer's rule or direct matrix inversion. In cells which cannot be organized into groups a point-by-point method may be used.

Convergence of the iteration may be accelerated by multiplying Δp in Eq. (2.41) by an over-relaxation factor ω [51,59]. Combined with the GS method it produces the *successive over – relaxation* (SOR) method. To preserve the convergence the over-relaxation factor has to satisfy the condition

$$0 < \omega < 2$$

The optimum value of ω which gives the fastest convergence lies in the interval between 1.0 and 2.0 and depends on the coefficient matrix of the linear system. For the incompressible fluid problems a value of ω between 1.7 and 1.8 is often optimum.

[4] The Jacobi method is also known as Richardson's method or method of ***simultaneous displacements***, and the Gauss-Seidel algorithm is also known as Liebman's method or ***successive replacements*** method.

It can be shown that the convergence acceleration due to over-relaxation is similar to choosing a maximum time step size in explicit methods for solving transient equations [27].

The PP, LL and BB methods were used to solve a simple two-dimensional problem in a square domain covered by $N \times N$ square mesh cells [60]. On one of the boundaries of the domain pressure was fixed at a constant value while the pressure inside the domain was given an arbitrary initial value. Then one of the three iteration methods were used to obtain the solution, which is obviously a uniform pressure equal to the boundary value.

The following facts were found in the test calculation.

- The BB method takes the shortest computing time to obtain the solution; and the LL method needs the longest computing time among the three methods.
- The number of iterations required are almost the same for LL and BB methods. The iteration number needed for the PP method was always larger, and more so for smaller values of ω. The reason for this is that the information diffuses slowly relative to the other techniques because the pressure value of a node is influenced only by its neighbours.
- Different values of ω were tested. For each iteration method there is an optimum value of ω with which minimum number of iterations is needed. This value is about 1.6 to 1.7.
- The optimum value of ω increases as the number of cells N increases. However, it cannot exceed 2.0.

In a flow problem which involves transient free surfaces and moving obstacles, use of the BB and LL methods becomes difficult because the coefficient arrays at different locations are not the same. Additional computing time is required to determine these coefficients. Therefore, the PP method appears to be the most efficient for these problems.

A widely-used variation of the line-by-line method for solving the linear algebraic equations is alternating-direction-implicit (ADI) scheme [61]. This method converts the line-by-line routine into a direct solver of two-dimensional linear problems through splitting the time step in halves (for 2-D, or in three equal parts for 3-D) and judicious alternation of the direction in which the triagonal matrix algorithm is employed. If the ADI method is used in the y-direction, the iteration consists of a sweep through all i cells solving for the pressures and velocities in the j-index direction.

High rates of convergence can be obtained with *preconditioning* techniques, particularly when coupled to *conjugate gradient* methods [62]. However, conjugate gradients methods require substantial computer memory storage [25,27].

In SOR methods, the number of iterations required for convergence, l_{max}, increases with N, the total cell number. For ADI methods applied to square regions, l_{max} is almost independent of N, so that for large enough N, ADI methods are preferable. In the numerical experiments of Birkoff *et al* [63] ADI methods were nearly four times faster than the optimum SOR method in a 40 × 40 grid. But for non-rectangular regions the ADI methods are not certainly known to be faster, and the SOR methods are easier to program.

Another comparison of the PP, LL and ADI methods was carried out by Chuan *et al* [64]. They considered the Stephan problem.

The finite volume, fully implicit method was used to obtain the system of algebraic equations that represent transient 3-D heat conduction with phase change, modelled by the enthalpy method. The material used was water, freezing at $T_m = 0^oC$, though an artificial "mushy zone" was used for the reasons discussed in Section 2.8.

The computational domain was a cube with three adjacent cold walls and the other three being adiabatic. The resulting equations were solved by fully implicit algorithms in one, two and three dimensions. The iteration involves the unknown temperatures and the position of the solidification front at the new time step.

The conclusions of the tests can be summarised as follows:

- For this type of non-linear problem the LL and PP solvers perform equally well in terms of CPU efficiency and accuracy. Only for the three-dimensional case does the ADI method substantially outperform the other two procedures.
- The relation between CPU time and time step size is non-linear. The use of too large a time step affects accuracy while providing little CPU savings.
- Solution accuracy is strongly dependent on the value of ζ, the temperature range given to the mushy zone.

Chuan *et al* also concluded that the enthalpy method in combination with the finite volume method produces an effective model to simulate freezing.

The simplicity and flexibility of the point-by-point SOR method in combination with the finite volume upwind differencing scheme produced an efficient solver for incompressible transient fluid flows called SOLA [53]. Its extension to include free surfaces, SOLA-VOF [51] (see Sections 2.4 and 2.5), is now widely used in mould filling simulations [49], [65]-[67].

Vanka [54] applied the *block – implicit multigrid* solution algorithm for incompressible flows. The multigrid technique is based on the idea that each frequency range of error must be iterated on an optimised grid. For a given grid the error component wavelengths comparable with the mesh size are smoothed out most efficiently. The multigrid technique cycles between coarse and finer grids until all the frequency components are appropriately smoothed. For a predefined set of grids, the solution is initiated on the coarsest grid M. GS iterations are performed on grid M until it is fully converged. The converged solution is then interpolated to the next, finer grid and is iterated on until the required convergence level. When the finest level is solved to the desired accuracy, then the whole solution cycle is terminated. In this formulation, when iterations start on a given grid the long wavelength error components have been already smoothed on the coarser grids.

The multigrid technique has been shown to be one of the most efficient solvers for linear and nonlinear equations [25,68] though the procedure of choosing the grid hierarchy and the interpolation of the variables between them requires some delicate programming [69]. It is unclear, though, whether this method will speed up the smoothing of the error components with wavelengths larger than the scale of the computational domain, *e.g.* uniform fluid pressure adjustments.

2.4 Free Surface Representation

Tracking transient free surfaces is an important part of a mould filling simulation, both in the physical and numerical aspects. A brief survey of the previous work on free surfaces is outlined below.

1. *Height Function Method* [53]. A simple means of representing a free surface is to define its height (in the z-direction) from a reference plane ($x - y$ plane) as a function of position in the reference plane and time, *i.e.* $h = f(x, y, t)$. This method is efficient, but limited to simple geometries and single-value surfaces.

2. *Line Segment Method* [70]. A generalisation of the one- or two-dimensional height function method uses chains of short line segments, or points connected

by line segments. This method removes the limitation of single-value surfaces, but the extension to three dimensions is extremely difficult.

3. *Marker Particles Method* [48]. This is also called Marker And Cell method (MAC). In this method marker particles are spread over all fluid-occupied regions, with each particle specified to move with the fluid velocity at its location. The MAC offers the advantage of eliminating all logic problems associated with intersecting surfaces and can readily be extended to three dimensions. However, it requires a significant increase in computer storage and running time.

   A number of variations of the MAC method exists, *e.g.* Simplified MAC method (SMAC) and Leading Marker Method (LMM). These variations aim at saving the computer storage and cutting the computer running time. Lin and Hwang [71], for example, store and compute only particles in the surface cells and those interior cells that are next to the surface. In this formulation the method cannot describe the opening of new voids in the fluid. An algorithm has to be designed for introducing new particles if the number of surface cells and their neighbours increases sharply during filling. Both are more likely to occur in a three-dimensional flow.

4. *Volume of Fluid Method* [51]. The Volume Of Fluid (VOF) method uses a continuous function to describe fluid configuration. It evolved from the original MAC method. The $F(x, y, z, t)$ function has the unity value at any point occupied by fluid and zero otherwise; it preserves its discontinuous nature to track a free surface boundary. When averaged over the cells of a computing mesh, the average values of F is equal to the fractional volume of the cell occupied by fluid. Cells with F value between zero and unity contain a free surface. The VOF method requires much less computer storage space and running time, and has none of the logic problems of handling intersecting surfaces in three dimensions.

   The flexibility of the VOF method has enabled its use in combination with a variety of numerical algorithms [5,49,65,67,72,74,75,76], including those based on the finite element approach [35,43].

5. *Variable Density or Two Phase Method* This method models free surface by introducing a density discontinuity: regions free of fluid are assumed to be filled with air [50]. In this case the fluid equations are solved for both fluid and air. The advantage of this method over the VOF method is that the boundary conditions at the free surface do not have to be set explicitly. It

allows problems with two or more fluids of different densities in contact with each other to be modelled.

However, for most of the casting problems air can be modelled as a region with constant pressure without the need to resolve the velocity field in it. This assumption can be made because of the large density differences between metal and air, and velocities of air are small compared with the speed of sound in the air.

Therefore solving the full density equations for metal and air may be a waste of computer resources. Besides, a higher order numerical scheme is required for the solution of this equation to preserve sharp interfaces between the fluids (see Section 2.5).

One way to accelerate the computation is to consider a fictitious air, called "numerical air" whose properties are such that the linear system issued from the discretised problem is solved rapidly. Careful attention has to be paid to interfacial conditions between the fictitious air and the fluid. The basic idea of the method is to solve a flow model in the air which gives very shallow pressure gradients. Consequently, the constant pressure condition at the free surface will be almost satisfied. This procedure is used in a commercial CFD package SIMULOR developed to model mould filling [77,78]. A shortcoming of the method is a relative imprecision of the exact location of the free surface if the mesh is coarse.

6. *Lagrangian Method.* This method appeared only recently and is still being developed (not to be confused with the Lagrangian method of tracking discontinuities developed by Hirt *et al* [79]). It is based on the representation of the flowing or deforming medium as a collection of mutually interacting parcels of material [80]. Individual parcels move as solid bodies subjected to the boundary conditions. The main problem in this method is to define the forces acting between the parcels as well as energy exchange and interaction with walls. These definitions depend on the number (or size) of the parcels used and at present are chosen posteriori making the resulting motion look more realistic. However, if the problem of arbitrariness is overcome, the method offers flexibility and control in introducing properties of the materials.

The VOF method is employed in FLOW-3D and is used in the present work for tracking free surfaces. The transport equation solved for F is

$$\frac{\partial F}{\partial t} + (\nabla \cdot \mathrm{v}) F = 0 \tag{2.46}$$

Velocities are obtained from the momentum and continuity equations prior to advecting the fraction of fluid function.

Once the fluid configuration is defined, pressure boundary conditions, $p_{i,j}$, have to be set at the centres of the cells containing the free surface. In the original SOLA-VOF code this pressure is obtained by linear interpolation (or extrapolation, depending on whether the surface cell centre is covered with fluid on not) between the surface pressure p_s and the pressure $p_{i,j-1}$ in the full fluid cell neighbour in the direction normal to the free surface as shown in Fig. 2.2*a*:

$$p_{i,j} = (1-\eta)\, p_{i,j-1} + \eta\, p_s \tag{2.47}$$

where $\eta = \Delta y/d$ and d is the distance from the centre of cell $(i, j-1)$ to the free surface. As pressure $p_{i,j-1}$ takes part in the iteration procedure, $p_{i,j}$ also changes from iteration to iteration according to Eq. (2.47). The following relation must be satisfied to ensure convergence [81]

$$\eta < 2 \tag{2.48}$$

which is always true in uniform meshes [60]. But it may not be the case in a non-uniformly spaced mesh. For $\eta > 2$, if $p_{i,j-1}$ changes by $\Delta p_{i,j-1}$ during an iteration, then $p_{i,j}$ will change by a greater quantity $(1-\eta)\Delta p_{i,j-1}$. During the next iteration $p_{i,j-1}$ will try to compensate for the change in the boundary pressure causing a swing of $p_{i,j}$ in the other direction but with an even larger amplitude, and no convergence will be reached.

To avoid such problems (and to speed up the calculation) Anzai and Niyama [82] used p_s as the cell-centred value of the pressure in the surface cell, *i.e.* put $p_{i,j} = p_s$. The results of their calculations for a die mould filling show little difference from the original method. This is not surprising given that the velocity scale was 10 m/s and the shape of the free surface is defined by the bulk liquid flow rather than the surface gravity waves. The Froude number $N_{Fr} = u^2/gd$ for the mesh size $d = 5\ mm$ is around 50. This method, however, is not acceptable for low Froude numbers characteristic for shrinkage feeding flow, for example, ($N_{Fr} \approx 0.005$ for velocities of the order of 1 mm/s) because the shape of the free surface in this case is mainly defined by the differences in pressures in the surface cells.

Another method of calculating surface cell pressures is to assume hydrostatic pressure distribution in the surface cell between the free surface and the cell centre [83]. If G_n is the component of the body forces (*e.g.* gravity) normal to the free surface then, using Fig. 2.2*b*

$$p_{i,j} = p_s + \rho_0\, G_n\, h \tag{2.49}$$

where h is the distance between the free surface and the cell centre. It is important that h is negative when the cell centre is outside the fluid. This procedure ensures continuous and smooth change of pressure in a cell as the free surface passes through it. Eq. (2.49) is a good approximation in a low Froude number flow and works equally well in fast flows with the velocity magnitude of the order of $1m/s$ [84]. In this formulation pressures in the surface cells are fixed during the iteration procedure.

A shortcoming of using the body forces to define the surface cell pressures is clear in a situation when body forces are absent but a non-zero pressure gradient exists in the fluid due to a difference in pressures applied at the fluid boundaries [85]. In that case, according to Eq. (2.49), the surface pressure is assigned to the cell centre without any adjustment, ignoring the net pressure gradient in the fluid.

By setting surface cell pressures using Eq. (2.49) surface gravity waves are generated, and care must be taken to avoid possible numerical instability arising from their propagation (see Section 2.10).

2.5 Free Surface Advection

In Section 2.3.1 the upwind (donor-cell) differencing methods was described to calculate the advection the fluid momentum. Unfortunately, it is only first order accurate and is highly diffusive and thus unsuitable for the free surface advection. A direct application of this method would result in smearing of the interface, leading to an non-physical result.

Usmani *et al* [35] used a continuous, rather than a step-like, fluid fraction function to avoid the numerical difficulties in advecting the step function. In their approach the free surface is associated with the value of $F(t,x,y) = 0.0$, instead of the range of values between 0.0 and 1.0 as used in the VOF method. $F(t,x,y) < 0.0$ indicates the fluid region and $F(t,x,y) > 0.0$ indicates the voids. Furthermore, at every time step function F is smoothed in a layer on both sides of the free surface to obtain a linear profile of F in the direction normal to the free surface. This serves two purposes: first, a linear distribution of F reduces the numerical diffusion as it is proportional to the second spatial derivative of F; second, the smoothing helps to avoid developing undesirable steep gradients of the F-function.

However, this approach is only suitable for tracking an already existing free surface. Opening new void is beyond its capabilities, in other words the full cells are always assumed to stay full unless the free surface passes though them. The original VOF method allows one not only to track existing free surfaces, but also to model the opening of new voids and free surfaces. The latter is crucial in solidification shrinkage modelling.

In the VOF method the *donor – acceptor* flux approximation is used to advect free surface. The essential idea is to use information about F downstream as well as upstream of a flux boundary to establish an approximate interface shape and then to use this shape in computing the flux. In each computational cell containing free surface, the free surface is represented by a plane parallel to one of the coordinate planes which depends on the distribution of F in the neighbouring cells.

When solving scalar advection Eq. (2.46) in a staggered grid, control volumes are chosen to coincide with the cells so that the convective fluxes can be easily calculated at the cell faces.

Consider the amount of F to be fluxed in the x-direction through a cell face during a time step Δt. The flux of volume crossing this face per unit cross-sectional area is $\mathcal{L} = u\Delta t$, where u is the normal velocity at the cell face. The sign of u determines the donor and acceptor cells, *i.e.* cells loosing and gaining volume, respectively. The amount of F fluxed across the cell face is defined as δF times the cell face area

$$\delta F = min\{F_{AD} \cdot \mid \mathcal{L} \mid + F_C \, , \, F_D \, \Delta x_D\} \tag{2.50}$$

and

$$F_C = max\{(F_{DM} - F_{AD}) \cdot \mid \mathcal{L} \mid - (F_{DM} - F_D \Delta x_D \, , \, 0.0\}$$

Here single subscripts denote the acceptor (A) and donor (D) cells. The double subscript, (AD) refers to either A or D, depending on the orientation of the interface relative to the direction of flow as explained below. F_{DM} is the maximum of F_D and the F value in the cell upstream of the donor cell.

The *min* feature in Eq. (2.50) prevents the fluxing of more F from the donor cell than it has to give, while the *max* feature accounts for an additional F flux, F_C, if the amount of void $(1.0 - F)$ to be fluxed exceeds the void amount available. Fig. 2.3 provides an explanation of Eq. (2.50). The donor and acceptor cells are defined in Fig. 2.3*a* for fluxing across a vertical cell face. When $AD = D$, the flux is a donor cell value,

$$\delta F = F_D \cdot \mathcal{L}$$

in which the value in the donor cell is used to define that portion of the cell face area exposed to fluid (Fig. 2.3*b*). Satisfying Courant stability criterion (Section 2.10) guarantees that it is not possible to empty the donor cell in this case.

When $AD = A$, the value of F in the acceptor cell is used to define that portion of the cell face area which F is flowing through. In Fig. 2.3*c*, all the fluid in the donor cell is fluxed because everything lying between the dashed line and the cell boundary moves into the acceptor cell. This is an example of the *min* test in Eq. (2.50). In Fig. 2.3*d* more fluid than the amount $F_A \cdot \mathcal{L}$ must be fluxed, so this is an example of the *max* test.

Whether the acceptor or donor cell is used depends on the mean surface orientation. The acceptor cell is used when the surface is advected in a direction normal to itself, otherwise, the donor cell value is used. The reason for testing on surface orientation is that an incorrect steepening of surface waves will occur if the acceptor cell is always used to compute the fluxes. In effect, the acceptor method is numerically unstable because it introduces a negative diffusion of F. Instabilities do not grow to unbounded values, however, because of the *min* and *max* tests used in the flux definition. On the other hand, when the surface is advected normal to itself, a steepening that maintains the step-function character of F is exactly what is required.

Once the flux has been computed by the above method, it is multiplied by the flux boundary area to get the amount of the fluid to be subtracted from the donor cell and added to the acceptor cell. In this way the fluid volume defined by F is conserved. When the advection process is repeated for all cell boundaries in the mesh, the resulting F values correspond to the time-advanced values satisfying Eq. (2.46) and still define sharp interfaces.

Youngs [86] developed a 2-D VOF advection model in which the reconstructed free surface is allowed to have a non-zero slope in a cell, rather than forcing it to be parallel to a coordinate plane. The Youngs' method is presumably more accurate, but excessive calculations may be required to define the orientation of the free surface in every cell when the method is applied in three dimensions.

An example of a second-order in space and time differencing scheme specially developed to overcome numerical diffusion is the van Leer method [87]. This method preserves the monotonicity of the advected variable thus preventing the spurious oscillations that often accompany higher-order methods.

Referring to Fig. 2.3*e*, the flux through the right-hand face of the cell can be computed as [50]

$$dF_e = -\frac{u_e \Delta t}{\Delta x}(F_e - F_w)$$

for positive u_e. The van Leer approach determines the face values in terms of local gradients of F:

$$F_e = F_p + \frac{\Delta x}{2}\left(\frac{\partial F}{\partial x}\right)_p (1 - u_e\frac{\Delta t}{\Delta x}) \qquad for \qquad u_e > 0$$

$$F_e = F_p - \frac{\Delta x}{2}\left(\frac{\partial F}{\partial x}\right)_E (1 + u_e\frac{\Delta t}{\Delta x}) \qquad for \qquad u_e < 0$$

The gradient $(\partial F/\partial x)_p$ is computed as

$$\left(\frac{\partial F}{\partial x}\right)_p = \frac{2\,sgn(\delta_e)}{\Delta x} min\{\mid \delta_e \mid, \frac{\mid \delta_e \mid + \mid \delta_w \mid}{2}, \mid \delta_w \mid\}$$

where

$$\delta_e = F_E - F_p, \qquad \delta_w = F_p - F_w$$

and

$$sgn(\delta_e) = \begin{cases} +1 & \text{if } \delta_e > 0 \\ -1 & \text{if } \delta_e < 0 \end{cases}$$

To ensure monotonicity, if $\delta_e \times \delta_w < 0$ then $(\partial F/\partial x)_p = 0$ and it reverts to upwind conditions.

The van Leer method has been proven particularly successful in preserving sharp discontinuities and was applied to advecting free surface in mould filling simulations [50,88].

The use of the VOF method with either the donor-acceptor or the van Leer advection schemes limits the time step size so that the free surface cannot be moved by more than one cell in one cycle. This restriction can be inconvenient, especially if it is significantly more severe than other time step limitations, *e.g.* viscous or heat transfer time step stability limits.

Swaminathan and Voller [72,73] suggested an "enthalpy type" model for advecting free surfaces. The analogy is used with the enthalpy method of describing phase transformations, according to which a computational cell undergoing the phase change will remain fixed at the melting/freezing temperature until the associated latent heat has been extracted/supplied (Section 2.8). In this situation a cell changing phase is unable to transport heat to cells which have not yet changed phase.

In the 'enthalpy type' free surface advection model a new variable, G is introduced such that

$$0 \leq F < 1 \quad \text{when} \quad G = 0$$

$$0 < G \leq 1 \quad \text{when} \quad F = 1$$

where F is the ordinary VOF function. The transport equation is then used in the form

$$\frac{\partial F}{\partial t} + (\nabla \cdot \mathrm{v}) G = 0 \tag{2.51}$$

In the context of the analogy with the phase change problem, F takes the role of the enthalpy and G of the temperature. Eq. (2.51) is equivalent to an isothermal phase change problem in which the specific heat is zero, *i.e.* the 'enthalpy' consists only of the latent heat.

The chosen relationship between F and G ensures that in a numerical solution, a control volume that is filling can only *receive* fluid from its neighbours; it cannot pass on any fluid to a neighbouring cell until it is completely full.

Eq. (2.51) is discretised in a fully implicit, first order form and solved in each control volume simultaneously. The simultaneous solution procedure is significantly simplified due to the relationship between F and G.

The advantages of the Enthalpy-VOF (EVOF) method are:

1. no numerical diffusion and smearing of the free surface is present due to the introduction of variable G;
2. due to the implicitness of the numerical method there is no restriction on the size of the time step.

The disadvantages are:

1. due to the chosen $F-G$ relationship, once a computational cell is full, it cannot be emptied again. Consequently, problems in which regions of the mould cavity fill and then empty (*e.g.* fillings with sloshing or *shrinkage feeding* flow) are outside the scope of the EVOF method;
2. to take full advantage of the large time step size advection terms in the momentum equations must also be approximated in an implicit way. This usually would require an interation method to solve the discretised equations thus introducing additional calculations in the solution procedure.
3. if the time step is so large that the free surface is advanced by several cells, a problem arises of setting velocities and pressures in the newly filled cells

which are required as the initial values for the pressure correction iteration procedure. As the solution for the Laplace equation for pressure, derived from the continuity equation (2.21), is defined by the boundary conditions [27], the first-guess pressure values are unimportant for the converged solution, if correct boundary conditions are given (though the number of iterations necessary to achieve the convergence *is* affected by the initial value). However, it is unclear how to calculate the velocities.

4. the condition for a computational cell to be full before passing the fluid to an empty neighbour is a good approximation if the free surface is advected in the direction normal to the free surface. In a case when the fluid flow is parallel to the free surface this approximation will lead to an incorrect position of the free surface.

At present the VOF approach, in conjunction with the donor-acceptor or van Leer advection method, is the most powerful and general tool to describe transient free surfaces.

2.6 Discretisation of the Energy Equation

Eq. (2.23) is to be solved numerically for the fluid enthalpy. A numerical analogue of it for a control volume V, chosen to coincide with a cell, is

$$H_V^{n+1} = H_V^n + \Delta t (\Delta H_{ad}^n + Q_T) \tag{2.52}$$

Here a first order forward differencing is used for the temporal derivative. ΔH_{ad}^n is the advective flux of the fluid enthalpy across the cell faces, estimated at time level n, *i.e.* explicitly. It is closely related to the advection of the fraction of fluid function, F (Section 2.5). Once the amount δF of the VOF function that is advected through a cell face per unit area has been calculated, the enthalpy advection is estimated by

$$\delta H = H_D^n \, \delta F \tag{2.53}$$

where H_D is the enthalpy of the upstream (donor) cell per unit fluid volume. The total flux of the enthalpy into the cell is then equal to the sum of the fluxes through each cell face times the corresponding face areas:

$$\Delta H_{ad}^n = \sum_{faces} A_k \delta H_k^n$$

In that way the fluid enthalpy is conserved as the fluid moves through the mesh.

The discretised Fourier conduction terms in Eq. (2.52) are represented by Q_T which is the sum of fluxes at control volume faces. In two-dimensions, a uniform grid and constant heat conduction coefficient k, the central differencing for the Laplacian term in Eq. (2.23) at cell (i,j) gives

$$Q_T = k\left(\frac{T_{i+1,j} - 2T_{i,j} + T_{i-1,j}}{\Delta x^2} + \frac{T_{i,j+1} - 2T_{i,j} + T_{i,j-1}}{\Delta y^2}\right) \tag{2.54}$$

Eq. (2.54) gives a second order accurate approximation for thermal diffusion. The advantage of this approximation is that it only involves the immediate cell neighbours, in other words, it uses the same expression for all internal fluid or mould cells. For the boundary cells, either the temperature or the heat flux must be specified as the boundary condition.

A fully explicit formulation for the conductive heat fluxes, *i.e.* with all temperatures in Eq. (2.54) evaluated at time level t_n, is employed and a time step stability limit is introduced. For a one-dimensional flow it is (see also Section 2.10)

$$\Delta t < \frac{\Delta x^2}{2\,\alpha}$$

where $\alpha = k/\rho C$ is the diffusion coefficient.

A fully implicit approximation of the conduction term would imply all temperature on right-hand side of Eq. (2.54) to be taken at the new time level t_{n+1}. This will remove the time step restriction but the system of the discretised equations becomes coupled and iterations are required at each time step to obtain the solution [52]. Note that at each iteration Eq. (2.17) will have to be solved for the temperature. The latter can be complicated by a non-linear dependence of the fraction of solid function f_s on the temperature (Section 2.8) and the whole solution procedure may prove to be ineffective. To avoid this complication, the energy equation can be written solely in terms of temperature and the phase transformation taken into account using the "effective specific heat" [5] (see also Section 2.8).

A semi-implicit method was suggested by DuFort and Frankel [89]. The essence of the approach is to use only the central node temperature in expression (2.54), $T_{i,j}$, at the new time level t_{n+1}; all the other temperatures are taken at time t_n. In this case there is no stability limit for the time step and the solution matrix for the temperatures is purely diagonal. The cost of employing this method can be demonstrated on a one-dimensional heat conduction problem without phase change and fluid flow. Eq. (2.52) then can be written for a node i as

$$T_i^{n+1} = T_i^n + \Delta t\,\alpha\frac{T_{i+1}^n - 2T_i^{n+1} + T_{i-1}^n}{\Delta x^2} \tag{2.55}$$

or

$$T_i^{n+1} = T_i^n + \Delta t' \alpha \frac{T_{i+1}^n - 2T_i^n + T_{i-1}^n}{\Delta x^2}$$

where

$$\Delta t' = \frac{\Delta t}{1 + \frac{2\alpha\Delta t}{\Delta x^2}}$$

and

$$\lim_{\Delta t \to \infty} \Delta t' = \frac{\Delta x^2}{2\alpha}$$

Thus Eq. (2.55) is equivalent to a fully explicit expression, with an erroneously small time step. It is erroneous because the user may think that the result of n time steps applies to the time $n\Delta t$, when it actually should apply to the time $n\Delta t'$. The errors result from the absence of energy conservation property in this method since temperatures at different time levels are used to evaluate the heat flux in the right-hand side of Eq. (2.55).

A fully explicit first order numerical approximation to the thermal diffusion term (2.54) will be used in the present work due to its simplicity with regard to the solution matrix inversion and to the development of the shrinkage model.

2.7 Modelling the Metal/Mould Interfacial Heat Transfer

It is not always important to use a higher-order approximation for the thermal diffusion term because temperature gradients in the casting are small. More effort is required to obtain an accurate approximation of the heat flux across the metal/mould interface given by

$$dq = h \cdot dA \cdot (T_{metal} - T_{mould}) \qquad (2.56)$$

where dA is an element of the interface, temperatures T refer to the metal and mould interfacial temperatures and the reciprocal of h is proportional to the thermal resistance of the interface. The problem of calculating dq is closely related to the description of the casting geometry.

The importance of the interfacial heat transfer strongly influences the choice of the numerical technique to solve Eqs. (2.21)-(2.23). Often the solution accuracy for fluid flow is sacrificed in favour of the heat transfer.

The use of unstructured meshes in the finite element approach allows one to solve successfully both problems of the geometry and heat transfer modelling. The latter is achieved by placing the mesh nodes on the interface (boundary fitted mesh) and using temperatures, the metal and the mould, at each of those nodes so that the temperature difference in Eq. (2.56) can be accurately estimated [90]. Besides, a formation of an air gap can be described if the nodes are fixed to the solid material as it moves off the mould wall [50]. The effect of the gap size is then taken into account by adjusting the heat transfer coefficient h. An experimental procedure for measuring the gap size and the heat transfer coefficient has been given by Hou and Pehlke [24]. Evans *et al* presented an experimental techniques of measuring the interfacial heat flux [91].

The use of structured mesh in the finite volume approach introduces the following problem. Generally the metal/mould interface is represented by a stepwise surface because each computational cell can be either completely open to flow or completely blocked by the mould, and the approximate interface passes along the cell faces. A finer mesh is not a solution because (a) there is always a limit to how fine a mesh one can use and (b) refining the mesh does not make a better approximation for the interface area. Figs. 2.4a, b show a finite difference approximation to a straight line by two meshes, a coarse and a fine. Although the finer mesh representation seems to be closer to the straight line the length of the discretised line in each case is the same and depends only on the line inclination. As a consequence, finite difference cell nodes do not lie on the interface, and the cell-centred temperatures cannot represent the interfacial temperatures.

Another way of describing solid walls is called Fractional Area and VOlume Ratio (FAVOR) technique [88,123]. In this method a cell volume and faces are allowed to be partially blocked (Fig. 2.4c) Four additional arrays are required then for each cell (three in two dimensions): the fractional open cell volume and fractional open areas for the right, back and top cell faces as shown in Figure 2.4d for a two-dimensional cell. In this case an interface containing cell has two temperatures: metal and mould.

FAVOR method allows much smaller numbers of cells to be used to represent curved surfaces and gives better wall boundary condition approximations not only for the heat transfer, but for the viscous stresses as well. The price paid for the improved accuracy are the additional storage requirement and possibly a more severe time step stability limit (Section 2.10 and Chapter 3)

It is shown in Section 4.1 how, using the FAVOR technique, temperature nodes can be efficiently introduced *at the interface* on both sides without any significant increase in the computation. The actual interfacial area in each cell can be estimated at the preprocessing stage and used as dA in Eq. (2.56) [92].

Structured finite difference and finite volume meshes are fixed in space and cannot be used to model efficiently the movement of the solid material and the gap formation due to thermal stresses. An attempt to model the gap formation was made by Huang *et al* [94]. Their model is based on the assumption that the magnitude of an interfacial gap size primarily depends on the thermal contraction of the casting as it solidifies. For a point P at the metal/mould interface, its movement relative to the geometrical centre of the casting and the movement of the casting as a whole due to gravity is estimated. As a result the point can move away from the interface creating a gap of size Δl. The heat transfer coefficient is adjusted according to the formula

$$h = \frac{k_g(T)}{B + C\Delta l}$$

where $k_g(T)$ is the gas thermal conductivity at temperature T, and B and C are fitted from experimental data. This model is simple and efficient and has been implemented in two-dimensions.

There have been numerous attempts to simplify and speed up the solution algorithm for the energy equation. Dantzig and collaborators [95,96] developed a *boundary curvature method* based on the fact that, generally, information on mould temperature is not required. Hence, the mould can be excluded completely from the calculation given that the correct heat fluxes at the metal/mould interface are known. The latter is calculated by correlating the actual interface shape with simple geometries for which an analytical solution exists and applying this solution to calculate the heat fluxes and temperatures in the mould without discretising the mould domain and solving the energy equation in it.

The boundary curvature method is especially attractive for modelling sand castings, because large differences in metal and sand thermal diffusivities require substantial computational effort to accurately calculate heat flux in the sand. A much finer mesh would have to be used in the sand than in the casting itself.

However, the boundary curvature method is limited by its lack of generality in describing complex geometries and by lack of physical transparency in the formulation of the method. The method has been implemented in SPIDER, a finite element code [97].

An analytical solution can be used to describe the penetration of heat into the mould for a short time after the contact between the metal and mould wall, when the thermal gradients in the sand are very steep and cannot be resolved by the finite volume mesh [93]. Stoehr and Wang [66] used an analytical expression for the interfacial heat flux per unit area when modelling thin wall sand casting:

$$q = \sqrt{\frac{k\rho C}{\pi t}}\,(T_{metal} - T_{sand})$$

in which k, ρ and C refer to the relevant properties of the sand and time is measured from the arrival of metal to that point. This is an attractive alternative to mesh refinement at the interface. However, care must be taken in using the analytical functions since physical conditions at the interface do not generally correspond to those assumed by the analytical solutuion. Besides, a special algorithm has to be derived to make a smooth transition from the analytical solution to the numerical one at later times of simulation.

Two-domain methods have been also developed in which the casting and mould are meshed independently and the two solutions of Eq. (2.52) are found in each of the domains separately. The interface boundary conditions are then used to match the metal and mould temperatures. This technique allows one to take advantage of the fact that the energy equation for the mould does not contain advection and phase transformation terms:

$$\frac{\partial T}{\partial t} = \frac{1}{\rho C}\nabla(k\nabla T) \tag{2.57}$$

Different time step sizes can be used in the two domains and linear temporal interpolation is then employed to bring the solutions together [98].

A similar approach is used by Chen and Tsai [99]. At a given time step Eq. (2.52) is solved together with the interfacial condition, Eq. (2.56), to obtain *only* the casting temperature distribution. The mould temperature is taken from the previous time step. Then in the next time step Eq. (2.57) is solved with the known temperature from the previous time step solution for metal. At each time step fully implicit formulation is used for the corresponding temperatures, and the solution is found iteratively. The method allowed optimum time steps and iteration parameters to be used for each material.

The general shortcoming of the two-domain method with separate time stepping is the absence of the conservation property: the net energy change in the mould may not be equal to the loss of the energy in the metal. It can be shown that for the Chen and Tsai method this results in a similar effect to that of the DuFort-Frankel method described earlier in this Section. Neglecting advection, phase transformation and assuming for simplicity a lump temperature model for both metal and mould, the Chen-Tsai model becomes

$$\frac{T_1^{n+1} - T_1^{n-1}}{2\,\Delta t} = -\alpha_1\,(T_1^{n+1} - T_2^{n}) \tag{2.58}$$

$$\frac{T_2^{n+2} - T_2^n}{2\,\Delta t} = \alpha_2\,(T_1^{n+1} - T_2^{n+2}) \tag{2.59}$$

The same time step size was chosen for domains 1 (metal) and 2 (mould) with $\alpha_1 = h/\rho_1 C_1$ and $\alpha_2 = h/\rho_2 C_2$. The metal and mould temperatures are calculated at times t_{n+1} and t_{n+2} respectively. The loss of the conservation property arises from two factors. Firstly, the heat fluxes out of the metal and into the mould are evaluated at different times. Secondly, the two temperatures used to evaluate these fluxes in both equations are taken from different times. If T_1^{n+1} on the right-hand side of Eq. (2.58) is substituted by an estimate for the metal temperature at time t_n

$$T_1^{n'} = \frac{1}{2}(T_1^{n-1} + T_1^{n+1}) \tag{2.60}$$

then the heat flux and time t_n will be estimated in a more conserving manner. Eqs. (2.58), (2.59) can be rewritten after some algebra as:

$$\frac{T_1^{n+1} - T_1^{n-1}}{2\,\Delta t'} = -\alpha_1\,(T_1^{n'} - T_2^n) \tag{2.61}$$

$$\frac{T_2^{n+2} - T_2^n}{2\,\Delta t''} = \alpha_2\,(T_1^{n+1} - T_2^{n+1'}) \tag{2.62}$$

where

$$T_2^{n+1'} = \frac{1}{2}(T_2^{n+2} + T_2^n)$$

and

$$\Delta t' = \frac{\Delta t}{1 + \alpha_1 \Delta t}, \qquad \Delta t'' = \frac{\Delta t}{1 + \alpha_2 \Delta t}$$

Similarly to the DuFort-Frankel method, the actual time steps, $\Delta t'$ and $\Delta t''$, at which the solutions for the metal and mould advance are smaller than Δt. Furthermore, the times, to which those solutions actually correspond, are not the same for metal and mould and the difference grows as the calculation proceeds because $\Delta t' \neq \Delta t''$ due to the differences in thermal diffusivities. The error in the Chen-Tsai method (as well as the DuFort-Frankel) is a truncation error , *i.e.* it is proportional to a power of the time step size.

A two-domain method is also used in the present work, but energy equations for the metal and mould are coupled at the same time level and solved simultaneously. The latter is simplified by the use of a fully explicit formulation of the numerical analogue equations.

2.8 Modelling the Latent Heat Release

There are several widely used numerical methods to take into account the latent heat in the energy equation.

Enthalpy Method. The enthalpy method of accounting for the latent heat release uses the Energy Equation in the form of Eq. (2.23). Once a numerical solution for H at a certain time level is found, using one of the methods described in Sections 2.6 and 2.7, the current released latent heat and metal temperature are found from Eq. (2.17).

The advantage of the enthalpy method (Eq. (2.23)) is that it is not possible for a node to drop through the freezing range without the latent heat being accounted for, since it is incorporated at every stage of the computation. In other words the enthalpy method is energy conserving.

However, this model has a disadvantage when solidification occurs at a constant temperature T_m (pure metals or eutectic alloys), or within a narrow temperature range. If the solidification front is passing through a cell, then the temperature at the node is kept equal to T_m. In other words, the temperature of the solidification front is assigned to the whole cell while it contains the front. This generates numerical 'heat waves' which distort the temperature field (Fig. 2.5*b*). Amplitudes of the waves are higher for larger temperature gradients in the casting while a wider mushy zone and smaller latent heat would decrease this effect.

An artificial freezing range is usually introduced to prevent the cooling curves from exhibiting false plateaus, However, this will cause a diffusion of the solid/liquid interface. In general, the smaller the artificial mushy zone, the better the solution accuracy [99].

Another enthalpy-based method, called *enthalpy diffusion model*, has been suggested by Mundin and Fortes [104]. Here the energy equation is written in the form

$$\frac{\partial H}{\partial t} = \nabla \cdot (\alpha \nabla H) \tag{2.63}$$

where α varies from $k/\rho C$ to zero (in the moving front). Eq. (2.67) gives an energy conserving numerical solution. The temperature oscillations, typical in the enthalpy method, are absent but a stretching of the phase changing region is introduced since enthalpy is continuous across the liquid/solid interface.

Numerical results become increasingly sensitive to the choice of a solidification model for alloys with large freezing range $T_l - T_s$ [105]. As was mentioned in Section 2.1.2 it is not among the objectives of this work to include solidification kinetics. We will only note that the enthalpy method assumes the enthalpy of a cell to be a *unique* function of the cell temperature. This would introduce a difficulty in modelling undercooling, characteristic to the kinetics models, as it implies at least three enthalpy values for one temperature when the cell is, first, undercooled, then the temperature recovers due to solidification and, finally, decreases as the solidification progresses.

Several authors used experimentally measured evolution of the metal enthalpy versus temperature to incorporate the latent heat release [100,106]. If a metal sample, V, is cooled at a sufficiently slow rate to assume it to be at a uniform temperature T, then the change of the total metal enthropy is

$$\frac{dH}{dt} = V\rho\left(C\frac{dT}{dt} - L\frac{df_s}{dt}\right) = \frac{h}{A}(T_\infty - T)$$

where h, A, and T_∞ are the metal/mould heat transfer coefficient, the interface area and the ambient temperature respectively. Since

$$\frac{df_s}{dt} = \frac{df_s}{dT}\frac{dT}{dt}$$

the total latent heat released to a certain time can be expressed as a function of the temperature and cooling rate

$$L\frac{df_s}{dt} = C\left(1 + \Theta\frac{T - T_\infty}{dT/dt}\right), \qquad \Theta = \frac{hA}{\rho VC}$$

and all that has to be done is to accurately measure $T(t)$.

The experimental method is perhaps the most accurate and flexible way to describe the latent heat release, especially for multi-component alloys. Most solidification modelling packages have an input option to include a table of enthalpy-versus-temperature values.

Modified Specific Heat Method. This method involves artificially raising the specific heat within the freezing range to account for the release of latent heat. The simplest method employs a constant higher value of specific heat within the solidification range [66,93]:

$$C_{eff} = C + \frac{L}{T_l - T_s}, \qquad T_s \leq T \leq T_l \tag{2.64}$$

where T_l and T_s are liquidus and solidus temperatures, respectively. This is only suitable for narrow freezing range alloys. A more accurate expression is given by [100]

$$C_{eff} = C - L \cdot \frac{df_s}{dT}$$

where df_s/dT can be defined by using an appropriate solidification model.

If the liquidus and solidus lines are assumed to be straight, together with a complete mixing of the solute in both liquid and solid phases, then the lever rule applies [7] (see also Eq. (2.18) in Section 2.1.2):

$$f_s = \frac{1}{1-K}\left(\frac{T_l - T}{T_m - T}\right), \qquad T_s \leq T \leq T_l \tag{2.65}$$

where K is the partition coefficient and T_m is the melting point of the solvent metal.

If the conditions of the solidification are such that there is no diffusion of the solute in solid and complete diffusion in the liquid phase, then the fraction of solid is determined by Scheil equation:

$$f_s = 1 - \left(\frac{T_m - T}{T_m - T_l}\right)^{\frac{1}{K-1}}, \qquad T_e < T \leq T_l$$
$$f_s = 1, \qquad T \leq T_e \tag{2.66}$$

where T_e is the eutectic temperature. This is an acceptable approximation if the time scale of the diffusion in the solid is much larger than that of any other involved macroscopic process.

A problem with the modified specific heat method is that it is possible for a node to 'jump' over the freezing range in a single time step, thereby missing out the latent heat evolution. This is illustrated in Fig. 2.5*a*, where the effective heat capacity is calculated using Eq. (2.63). A post-iterative check is necessary to ensure that the latent heat has been accounted for. If this not the case then the nodal temperature must be readjusted accordingly [101].

It may be difficult to justify the application of these models in the form of Eqs. (2.64) and (2.65) in numerical modelling. For example, Scheil equation, Eq. (2.65), assumes an infinite diffusivity of the solute in the liquid, no diffusion in the solid phase of the material and that the *total* amount of the solute in the casting, M_c, is constant

$$M_c = const \tag{2.67}$$

It implies that the liquid phase in the casting always has a uniform concentration, and the fraction of solid function in Eq. (2.65) is the net mass solid to liquid ratio in the casting.

In the control volume approach Eq. (2.65) is applied to every control volume, with f_s being the fraction of solid *in the control volume.* This effectively means that Eq. (2.66), instead of being reasonably applied to the whole casting volume, is used in each control volume, thus, becoming much stronger and, generally, inaccurate and making the numerical solution mesh-dependent. In reality the amount of solute in each control volume may change due to fluid advection and diffusion.

Similar problems can be seen in the application of the lever rule. Despite that, Eqs. (2.64) and (2.65) and their variations have produced useful predictions.

Temperature Recovery Method. In the case of pure metals the latent heat is divided by the specific heat to give a virtual temperature, ΔT_V [102]. This may be regarded as the temperature change over which an amount of specific heat equal to the latent heat would be evolved. Using one of the above solidification models the variation of enthalpy within the solidification range can be obtained. The modified specific heat capacity at a particular temperature can then be determined from the slope of the enthalpy versus temperature curve.

The liberation of latent heat is accounted for by holding the temperature of a node at the freezing point until a number of excess degrees equal to the virtual temperature have been accumulated after which the temperature is allowed to fall.

In alloy solidification the latent heat can be accounted for in a stepped manner according to the percentage of solid formed at different temperatures in the freezing range [103]. The disadvantage of the temperature recovery method is that it requires that additional post-iterative calculations be made.

In the present work the enthalpy method will be employed to model solidification, using Eqs. (2.17) and (2.23).

2.9 Solidification Shrinkage Modelling

Porosity in castings can be either shrinkage induced or gas induced.

Shrinkage induced porosity results when there is an inadequate supply of liquid metal to counter the volumetric shrinkage on solidification exhibited by most metals.

Gas induced porosity may result from the evolution of gas during solidification by either one of the two mechanisms:

- by a decrease in the solubility of dissolved elemental gas upon solidification (*e.g.* H_2 in aluminium);

- by the formation of a compound gas during solidification (*e.g.* CO in steel).

A further possibility is that air bubbles may be introduced into the liquid metal at the filling stage. If these bubbles are unable to escape then gas porosity will result [2].

Usually in castings the gas content is kept as low as possible by using either inert gas degassing techniques, vacuum treatment or by the addition of an element which form a solid compound with the dissolved gas. Exceptions to this are steels in which CO is allowed to form to compensate for solidification shrinkage. A similar technique is sometimes used in die casting when dispersed porosity is considered less harmful than a concentrated shrinkage cavity.

Macroporosity, or concentrated cavities, appear in castings in two forms: primary and secondary porosity. Primary cavities are located at the top of a casting, usually in the feeders, and result from lowering of the liquid metal surface due to the gravity driven feeding flow.

If during the solidification of a pure metal part of the casting freezes over, leaving a substantial pocket of liquid metal isolated from the feeding top, then gross internal, or secondary porosity results. In the case of alloys solidifying with a dendritic front, it is not necessary for the metal to freeze over to 100% solid to stop the feeding flow. This is because when the fraction of solid in the mushy zone is high, the remaining interdentritic liquid regions do not make a continuous passage connecting the feeder and the rest of the liquid casting.

Microporosity generally occurs randomly between the dendrite arms of alloys solidifying in a pasty manner. Uram *et al* [107] *et al* observed that the amount of microporosity between dendrites in a columnar region is considerably less than in the equiaxed region. The reason for this is that the interdendritic liquid regions in the columnar region are orientated along the crystal axis and they are more likely to connect the shrinking material in the mushy zone with the bulk liquid metal.

Campbell [2] presented a schematic representation of the feeding mechanisms active at different stages during an alloy solidification as shown in Fig. 2.6. Liquid feeding can occur at low pressure gradients and will be dominant during the early stages of solidification. Mass feeding will continue until solid particles impinge on each other. This mode of feeding is most effective when the size of the equiaxed crystals is small. When mass feeding ceases, liquid flow through the interdendritic channels continues until the pressure drop increases to such an extent that feeding is no longer possible, after which microporosity formation will result.

In the formation of the internal porosity the contributions of dissolved gas and the pressure drop across the mushy zone due to incompressibility of metal are additive. Assuming no barrier to pore nucleation the condition for the formation of a pore of radius r is [7]

$$p \leq p_g - \frac{2\sigma}{r}$$

where p is the local metal pressure and p_g is the equilibrium gas pressure.

At high gas contents pores can form early during solidification and remain spherical in shape. Gas bubbles forming at later stages will become trapped in the mushy region, their shape distorted to the shape of the interdendritic space. The total void volume is the sum of shrinkage and the evolved gas volume.

The present research is primarily concerned with *shrinkage* porosity, though a uniformly distributed gas porosity could be taken into account by decreasing accordingly the solid phase density (Chapter 7). The exclusion of detailed gas microporosity modelling is mainly due to the complexity of the physical phenomena. Experimental results indicate that pore sizes in alloys varies markedly with cooling rate, gas content and, to a lesser extent, with grain refining [108].

The first attempts at predicting the occurrence of *shrinkage* cavities in castings using numerical techniques were based on the fact that unsoundness will result in a region of a casting in which a pocket of liquid metal is isolated from the feeder by solid metal, or by a semi-solid region which has a fraction of solid greater than some critical value [109]. Whilst this method gives an indication of where porosity will form by using a numerical model of solidification to locate thermal centres within a casting, the actual size and shape of shrinkage cavities are not predicted. This method can in fact give misleading results since the last liquid to solidify will not normally correspond to the centre of the shrinkage cavity.

Imafuku and Chijiiwa [110,111] developed a method based on mass conservation which predicted the shape of gross shrinkage cavities and extent of the microporosity which might surround them. The total volume contraction is calculated for each time step by a summation of the volume contraction within each individual element due to solidification shrinkage. This results from a change in fraction solid between t_{n+1} and the previous time, t_n. The distributions of temperature and solid fraction are determined using a 2-D finite element analysis. Interdendritic flow is assumed to cease at $f_{s,cr} = 0.67$ fraction solid. If any liquid is trapped by the mushy zone at $f_s > f_{s,cr}$, then a new cavity forms at the top of the trapped region. When all of the casting is at a fraction solid greater than 0.67, the remaining semi-solid region is considered to be a likely area of microporosity. The actual distribution of the porosity is calculated using the Solid Fraction Gradient Method, which assumes that pores are generated by the solidification shrinkage of a large quantity of molten metal remaining at the time of the $f_{s,cr}$ isosurface disappearance.

The main assumptions of this model are:

- molten metal flows downwards due to the gravity, *i.e.* effects of pressure and viscosity are neglected;

- the flow speed is much greater than that of the solidification front; hence feeding can be assumed to occur instantaneously;

- the volume of shrinkage cavity is equal to the volumetric contraction by solidification;

- the melt has full fluidity for $f_s < f_{s,cr}$.

Whilst this method is not able to predict the formation of dispersed porosity in regions of low temperature gradients it can predict the actual shape and position of gross shrinkage cavities. This is not the case with methods which ignore any consideration of fluid flow and simply attempt to locate thermal centres within the casting.

The shortcomings of the shrinkage prediction method proposed by Imafuku and Chijiiwa are

- no heat flux due to the metal flow is calculated; this can be a combination of the residual circulation from the filling stage, the feeding flow and thermal convection. The latter may be increasingly important for low conductivity metals, such as steels.

- the use of the Solid Fraction Gradient Method requires a high accuracy numerical heat conduction/transfer model, and may be sensitive to the method of calculating latent heat.

- it is not clear from the description of the method [111] whether the computational elements are allowed to be partially filled during the calculation and how is it taken into account in terms of the heat flow.

A similar method was used by Nagasaka *et al* [112]. Here the authors use the solid fraction gradient, ∇f_s, as the key parameter that controls the driving force for feeding. The movement of liquid was considered only in the direction of the gradient, *i.e.* normal to the solidification front. In addition a critical value of ∇f_s is applied as a criterion for the feeding. At each time step all possible feeding paths (consisting of cells with $f_s < f_{s,cr}$) to a solidifying cell are determined and the solid fraction gradient is calculated along each of them. The cell then can only be fed if at least in one feeding path ∇f_s does not fall below the critical value. The total shrinkage volume generates a macro-cavity in the elements with the minimum pressure head thus taking into account gravity.

This method also requires a high degree of numerical accuracy in evaluating the fraction of solid gradients. Furthermore, the determination of all possible feeding paths for each solidifying cell may be inefficient in 3-D problems.

The agreement with the experiments obtained by Imafuku and Chijiiwa and Nagasaka *et al*, is good, although the accuracy is limited by the size of the individual elements. The choice of the critical solid fraction value is rather arbitrary. A value of $f_{s,cr} = 0.9$ has been suggested by Davies [113] and Spittle and Brown [114].

The method of Niyama *et al* [115] is based on the experimental observation that centreline porosity results if the temperature gradient, G, along the centreline is below a certain value. It was confirmed experimentally that this technique can successfully predict centreline shrinkage in 100 mm diameter steel casting. A temperature gradient of $0.2\,^oC/mm$ was determined as the critical temperature gradient required to avoid centreline shrinkage, but its value was shown to vary with the section thickness.

The same authors have found that if the criterion $G/\sqrt{R}$ is used, where R is the cooling rate, then the same critical value of $0.8\,^oC^{1/2}s^{1/2}mm^{-1}$ can be applied to all castings of similar compositions regardless of the section size. Although this criterion is essentially empirical, the authors showed that it has a theoretical basis [116]. However, it is unclear how to apply this criterion to general casting configurations since $G/\sqrt{R}$ is a *dimensional* parameter.

The $G/\sqrt{R}$ parameter is frequently used in commercial programs simulating solidification and some successful results are given by Hansen *et al* [28] and Huang *et al* [117].

Jesko and Zajac proposed the Velocity of the Solidus Line (VTS) as the parameter indicating the likelyhood of microporosity formation in a casting [118]. Microporosity appears when two or more crystallisation fronts impinge on each other. The local VTS then increases dramatically and a critical value, VTS_{cr}, can be introduced such that, if $\mathrm{VTS}>\mathrm{VTS}_{cr}$ at a point, then this point is a likely location of microporosity.

This method was applied to predict the centreline porosity in a cylindrical steel casting, for which VTS$_{cr}$=3 mm/s was chosen.

The authors claimed that their method is more general than the $G/\sqrt{R}$ method because the value of VTS$_{cr}$ does not depend on the general cooling rate conditions and casting geometry. However, VTS$_{cr}$ depends on the freezing range on a particular alloy and ideally it should be determined experimentally. Furthermore, the VTS method cannot be used to predict macroshrinkage.

An advantage of criterion methods is their simplicity and ease of use. They can be easily added to an appropriate computer code.

The main disadvantage of the criterion methods is that they require experimental verification of one or more of the critical parameters, making these models geometry and materials dependent. Criterion methods also do not model the dynamic interaction between developing macro- and micro-porosity regions with the rest of the casting. An internal cavity would affect the fluid and heat flow in the casting.

A more fundamental approach to predicting microporosity formation in castings solidifying in predominantly equiaxed manner was described by Kubo and Pehlke [119]. The technique involved a 2-D computation of heat flow coupled with a calculation of flow velocities and pressure drop. For each volume element the following mass balance is applied:

$$\left(1 - \frac{\rho_s}{\rho_l}\right)\frac{\partial f_l}{\partial t} = \frac{\partial f_V}{\partial t} - \nabla \cdot (f_l \cdot \mathrm{v}) \tag{2.68}$$

where ρ_s and ρ_l are the solid and liquid phase densities, respectively, f_l is the *volumetric* liquid fraction and f_V is the fraction of the porosity volume. Eq. (2.68) states that shrinkage during solidification is compensated by interdentritic fluid flow and the growth of porosity.

The liquid fraction f_l is found from a finite volume solution of the macroscopic energy equation. The velocity is found from Darcy's law

$$\mathrm{v} = -\frac{1}{K\mu f_l}(\nabla p - \rho \mathrm{g}) \tag{2.69}$$

where

$$\frac{1}{K} = \frac{f_l^3 d^2}{180.0\,(1 - f_l)^2}$$

is the permiability coefficient and d is the dendrite cell size. Gas pressure in the porosity region is calculated using equation

$$p = p_g - \frac{2\sigma}{r} \tag{2.70}$$

The diameter of the porosity forming first is assumed to be the same as the root diameter of the dendrite cells.

The conservation equation for gas content, Q, in this case hydrogen in an aluminium alloy, is

$$Q_0 = (1 - f_l)Q_s + f_l Q_l + \alpha_Q \frac{p_g f_V}{T} \qquad (2.71)$$

where Q_0 is the initial gas content and Q_s, Q_l are the gas contents in the solid and liquid phases, respectively. The last term in Eq. (2.71) describes the amount of the gas in pores.

When the volume element is in the mushy zone, the amount of porosity is calculated from metal pressure p using Eqs. (2.68) and (2.69). The gas pressure is then calculated from Eq. (2.70) and the new amount of porosity from Eq. (2.71).

In this model feeding flow velocities are estimated at each time step, though the advection terms in the momentum and energy equations are neglected. The flow is assumed to be dominated by the friction and gravity forces. However, this model is more suitable for predictions of micro-porosity due to gas evolution than for macroshrinkage modelling.

Stoehr and Wang [66] included a volumetric source term in the continuity equation in their coupled 2-D heat transfer and fluid flow model. If the solid/liquid mixture density in each control volume is estimated as the volume average[5]

$$\rho = f_s \rho_s + (1 - f_s)\rho_l \qquad (2.73)$$

then the Continuity Equation, Eq. (2.21), can be rewritten, for $0 < f_s < 1$, as

$$div\,\mathrm{v} = \frac{\rho_l - \rho_s}{\rho}\left(\frac{\partial f_s}{\partial t} + \mathrm{v}\cdot\nabla f_s\right) \qquad (2.74)$$

For $f_s = 0$ and $f_s = 1$ the source term is zero.

The source term in Eq. (2.74), evaluated from the solution of the heat transfer equation, would generate the flow necessary to accommodate the volumetric change and can describe both shrinkage and expansion (*e.g.* due to remelting) of the metal.

[5] f_s in Eq. (2.73) is the *volume* fraction of solid. If the mass fraction of solid was used, as everywhere else in the text, then Eq. (2.73) would transform to

$$\rho = \frac{\rho_s \rho_l}{f_s \rho_l + (1 - f_s)\rho_s} \qquad (2.72)$$

However, the authors do not present any numerical technique to find the solution of the coupled continuity and momentum equations, nor do they claim to have used the model to simulate the formation of shrinkage cavities.

Evans *et al* [120] developed a shrinkage model based on heat flow only for a 2D axisymmetric cylindrical geometry. They employed a value of the critical solid fraction $f_{s,cr} = 0.8$ which was higher than that used by Imafuku and Chijiiwa. Micro-porosity is estimated after the minimum solid fraction in the casting reaches the critical value. This value of $f_{s,cr}$ is used to estimate feeding in the vertical direction. In the horizontal direction it is assumed that feeding occurs until the metal is fully solid, *i.e.* effectively $f_{s,cr} = 1.0$ for the horizontal flow. Cells are not allowed to be partially filled. Instead, the liquid level is reduced by a full cell layer every time when sufficient volumetric shrinkage has been accumulated. This requires a fine mesh to reduce the resulting truncation error.

Another attempt to include solidification shrinkage effect was made by Fryer *et al* [121]. Their approach consists of solving the coupled enthalpy and stress equations. Stresses are assumed to be linear functions of the strain, so that in two-dimensions

$$\begin{aligned}
\sigma_{xx} &= \kappa\left(\frac{\partial \epsilon_x}{\partial x} + \mu_\sigma \frac{\partial \epsilon_y}{\partial y} - (1+\mu_\sigma)\,\beta\,\Delta T\right) \\
\sigma_{yy} &= \kappa\left(\frac{\partial \epsilon_y}{\partial y} + \mu_\sigma \frac{\partial \epsilon_x}{\partial x} - (1+\mu_\sigma)\,\beta\,\Delta T\right) \\
\sigma_{xy} &= \kappa\left(\frac{\partial \epsilon_x}{\partial y} + \frac{\partial \epsilon_y}{\partial x}\right)
\end{aligned}$$

where ϵ_x, ϵ_y are the displacements, $\kappa = E/(1-\mu_\sigma^2)$, μ_σ is the Poissons ratio, E the Youngs modulus and β the coefficient of linear thermal expansion.

The enthalpy equation is solved in the whole domain while the stress equations are only solved in the solid phase. The boundary conditions for the latter are such that the solid metal surface, including the liquid/solid interface, moves along the direction of the interface normal and towards the solid phase due to thermal contraction of the latter.

The authors use control volume approach with unstructured mesh in which the shape of the individual control volumes is initially rectangular, but then it changes to follow deformation of the material. The solution proceeds as follows:

1. solve enthalpy equation;
2. estimate ΔT in each control volume and solve the stress equations;

3. using the newly obtained displacements estimate the size of the air gap and the movement and deformation of each control volume;
4. calculate porosity within the liquid material by estimating the stretching of the liquid control volumes due to the movement of the solid phase. The stress boundary conditions at the liquid/solid interface ensure that the solid phase pulls the liquid phase apart, forcing it to create voids. The void regions are evenly distributed in the liquid phase at each time step, therefore, gravity is not taken into account.

For the metal/mould heat transfer calculation coincident nodes are used as described in Section 2.7.

The assumption inherent in step 4 above may be unrealistic. Cavities inside a casting appear because of the volumetric changes during the phase transformation occurring at the liquid/solid interface, and not because of the linear contraction of the solid phase upon cooling. Fig. 2.7 shows the predicted cavity in a 2-D square casting. In fact, given only thermal solid contraction, there should not be an *internal* cavity. The cavity in Fig. 2.7 results from the way the boundary conditions are set at the liquid/solid interface. The actual movement of the interface due to the thermal contraction of the solid phase depends on its position relative to the geometrical centre of the casting and the mould walls but it may not always move towards the solid phase along the normal to the boundary.

A development of another model for predicting macroshrinkage, which is based on full hydrodynamical equations, is described in Chapter 5 of this work.

2.10 Numerical Stability Considerations

There are several restrictions on time-step size must be observed to avoid numerical instabilities.

- for the explicit upwind differencing scheme, described in Section 2.3.1, the time step size has to satisfy the Courant-Friedrichs-Lewy (CFL) condition

$$\Delta t < min_{i,j,k}\left(\frac{\Delta x_i}{u_{i,j,k}}, \frac{\Delta y_j}{v_{i,j,k}}, \frac{\Delta z_k}{w_{i,j,k}}\right) \qquad (2.75)$$

  where u, v and w are velocity components. Physically this condition means that fluid must not flow across more than one computational cell per time step.

- if diffusion terms are approximated explicitly, then the time step is limited further:

$$\Delta t < \frac{1}{2 \cdot max_{i,j,k}\left[\alpha_{i,j,k} \cdot \left(\frac{1}{\Delta x_i^2} + \frac{1}{\Delta y_j^2} + \frac{1}{\Delta z_k^2}\right)\right]} \qquad (2.76)$$

  where $\alpha = \mu/\rho$ for the viscous terms in the momentum equation (Section 2.3.1) and $\alpha = k/\rho C$ for the heat diffusion terms in the energy equation (Section 2.6)[6].

  The restriction physically means that no quantity should diffuse more than one cell in one time step.

- free surfaces introduce another type of stability condition associated with the propagation of surface waves and the way the free surface cell pressure is determined in the calculation (Section 2.4). If a body force G is applied to the fluid in a direction normal to the free surface, there may be surface waves with speeds of order of $\sqrt{G\,h}$, where h is the depth of fluid or length of the wave. The value of h is defined by the pressure interpolation procedure in the surface cell, given by Eq. (2.49), and is equal to the cell size in the direction normal to the free surface.

  The actual condition is that surface waves should not propagate more than one cell in one time step. For example, if z is the normal direction to the free surface, then

$$\Delta t < min_{i,j}\left(\frac{\Delta x_i, \Delta y_j}{\sqrt{\Delta z_k\, h}}\right) \qquad (2.77)$$

  Similar limits must be imposed in the x and y directions for each cell containing a free surface.

The range of the stable time step sizes is given by the condition, Eqs. (2.75)-(2.77), with the smallest right-hand side.

These time step size stability limits result from using the von Neumann stability analysis of the linearised numerical analogue equations [27]. In this method the influence of boundaries is excluded and the solution behaviour is investigated by analysing the evolution of a single Fourier component of the numerical solution.

[6] In fact,

$$\alpha_{i,j,k} = \frac{k}{\rho\frac{dH}{dT}}$$

is a more correct expression. As $dH/dT = C + L\,df_s/dT$, the time step restriction becomes less restrictive for partially solid cells and, therefore, the latent heat has a stabilising role in an explicit numerical algorithm.

The mathematical foundations for the analysis of convergence and stability of numerical schemes are well-developed only for linear systems. The results from the linear theory are used as guidelines to nonlinear problems, the justification depending on numerical experiments. In practice, the stability limits given by right-hand sides of Eqs. (2.75)-(2.77) are decreased further by multiplying them by so called *safety factors* which usually do not exceed 0.5.

2.11 Continuation Section

The main interest of the present work is the development of a hydrodynamic model to simulate coupled fluid flow and shrinkage during solidification. Shrinkage defects are among the main causes of poor quality castings and a computer model capable of accurate predictions of theses defects.

Most of the existing numerical shrinkage models ignore fluid flow, including residual circulation, thermal convection and shrinkage induced flow. Those models which do include fluid flow have not been used to produce a comprehensive analysis of the influence of the fluid flow on the shrinkage defect formation.

The shrinkage model based on full fluid flow equations and which is described in Chapter 5 includes numerical models for:

- fluid flow;

- free surface;

- heat conduction and metal/mould heat transfer;

- latent heat release.

The work is based on a commercially available numerical model of fluid and heat flow with free surface motion described in Chapter 3. The metal/mould interfacial heat transfer and solidification models are enhanced to improve the accuracy and include physical models as described in Chapter 4.

An additional shrinkage model will be developed which is based only on heat flow equations thus ingnoring the fluid flow aspects of the solidification and shrinkage process. This model is used for comparisons of its simulation results with the results of the full shrinkage model. It will be shown that in some situations the results of the two models are close inducating that fluid flow may be not important in these cases.

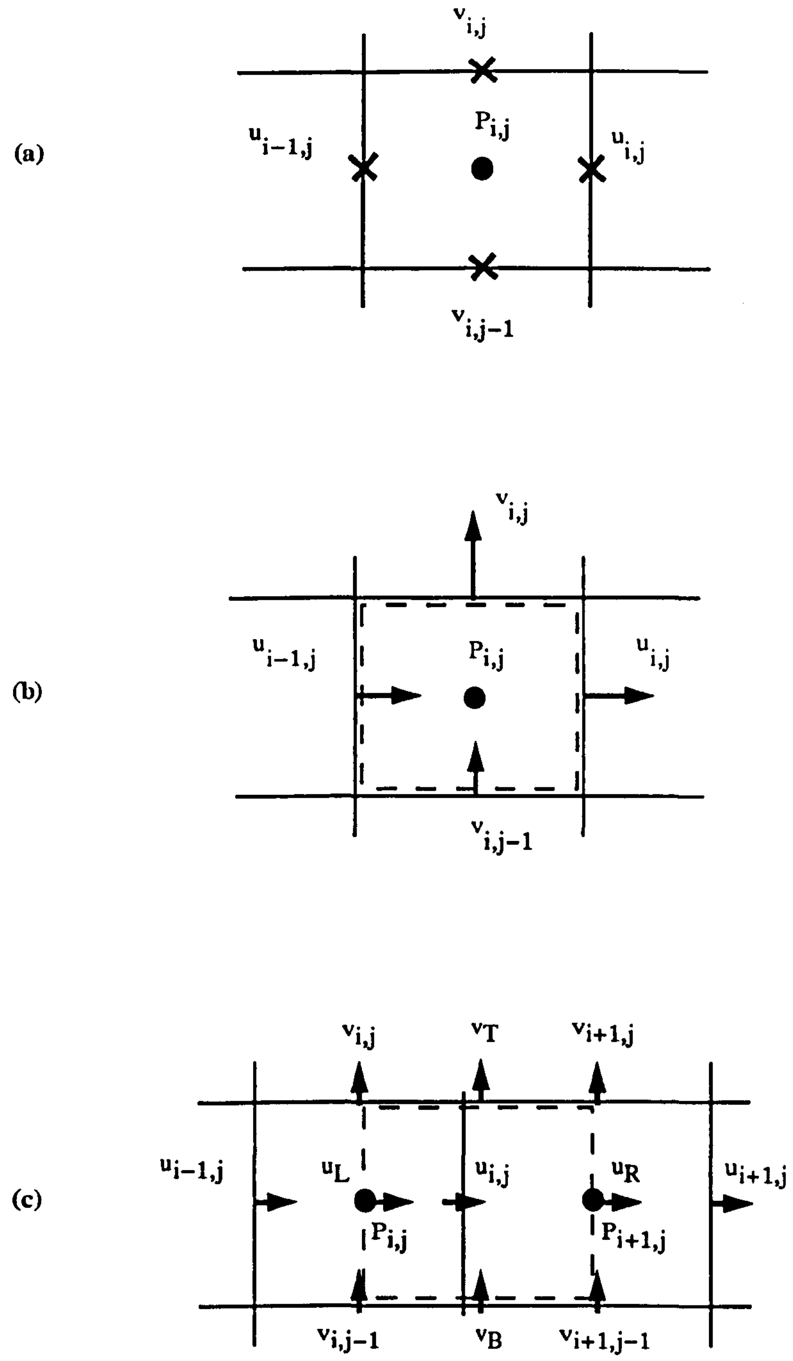


Figure 2.1. (a) the staggered mesh arrangement for the primitive variables; (b) the control volume for the continuity equation discretisation; (c) the control volume for the *x*–component of the momentum equation discretisation in the upwind differencing method.

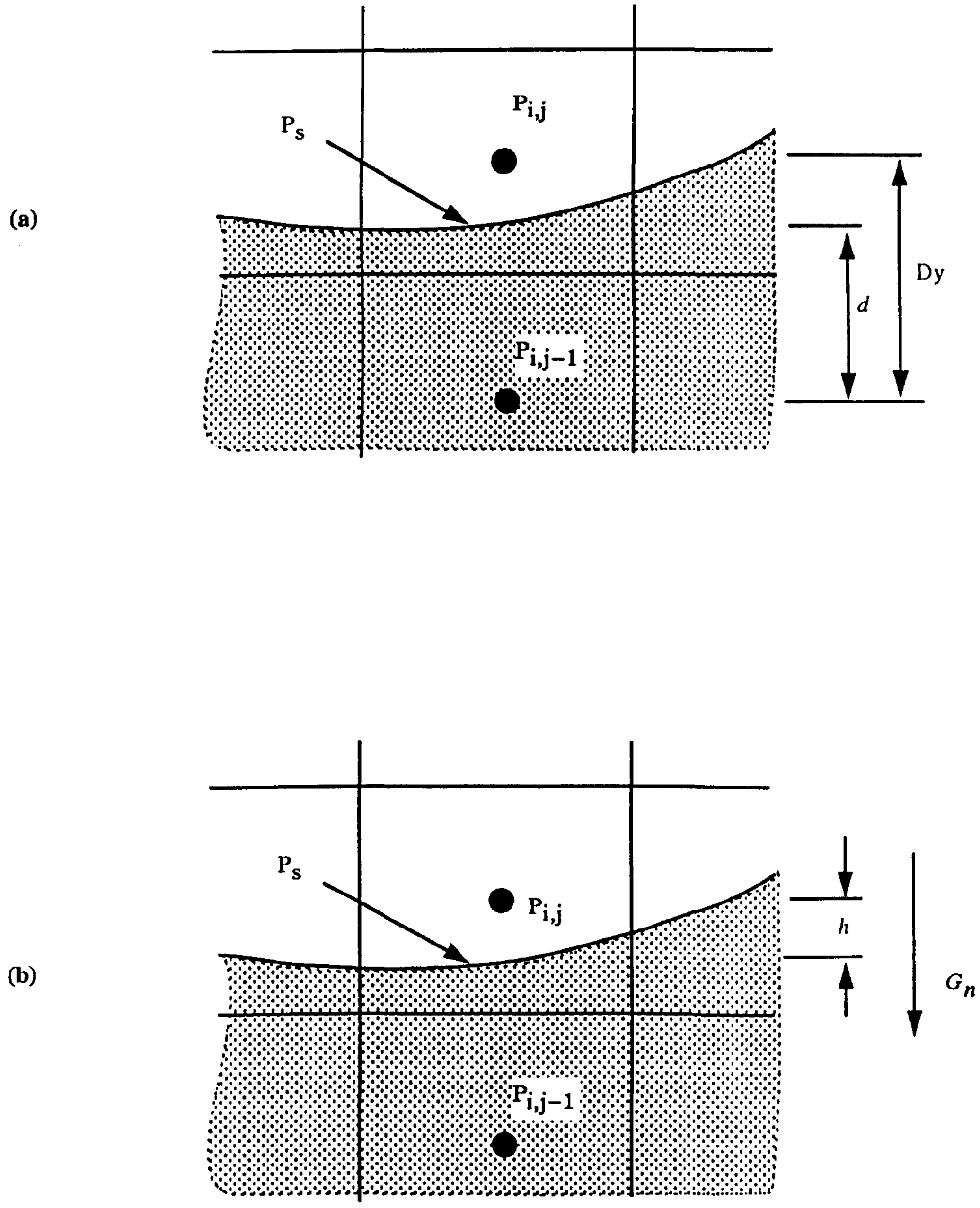


Figure 2.2. Setting the pressure in surface cell (i,j): (a) using linear interpolation between $P_{i,j}$ and P_s; (b) assuming hydrostatic distribution between the free surface and the cell centre due to the body forces G_n; h is negative if the cell centre is outside the fluid.

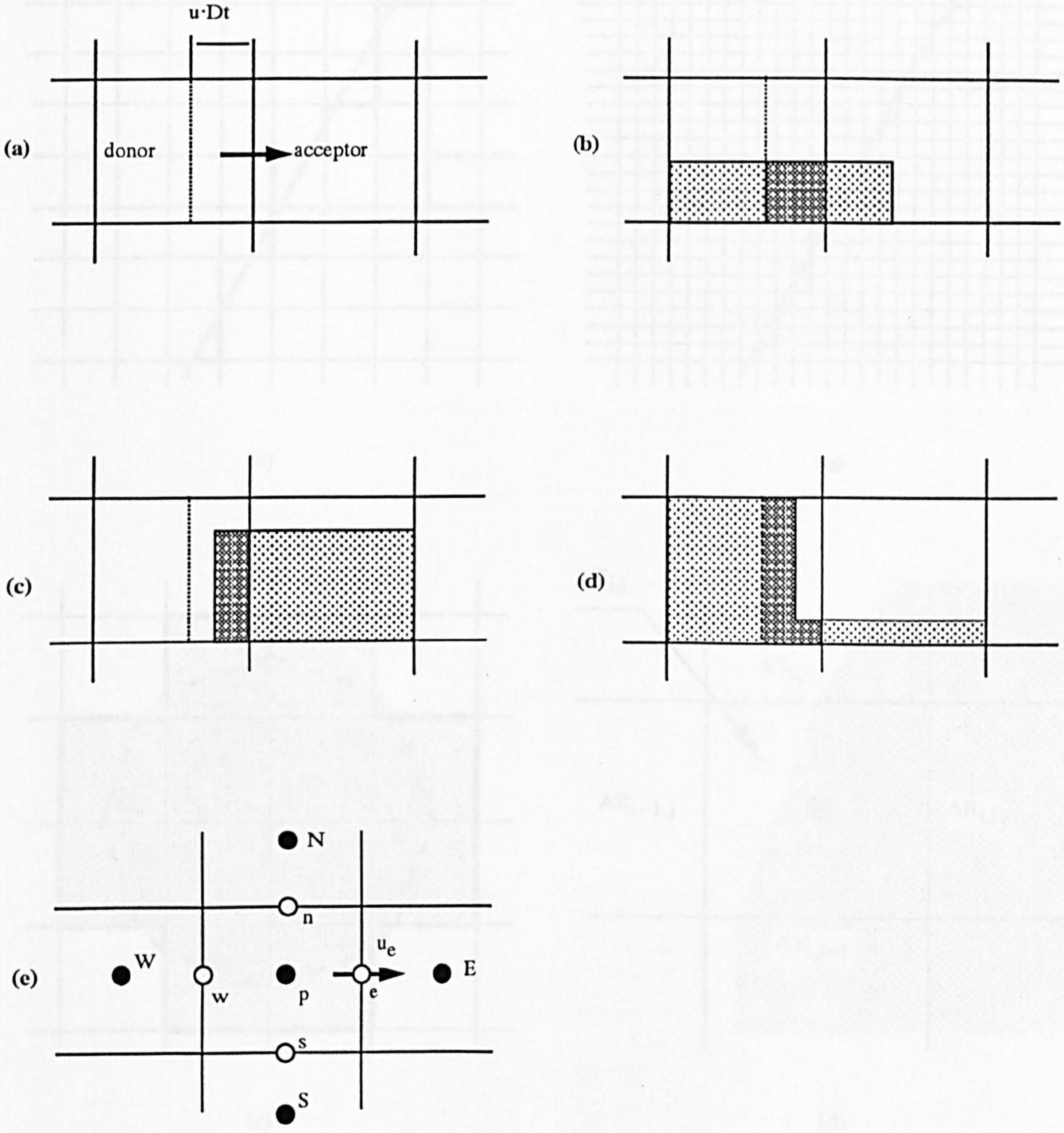


Figure 2.3. Examples of free surface shapes used in the advection of *F*. The donor-acceptor arrangement is shown in (a) where the dotted line indicates the left boundary of the total volume being advected. The dark shaded regions shown in (b-d) are the actual fluxed amounts of *F*. Cell notation for the van Leer method is given in (e).

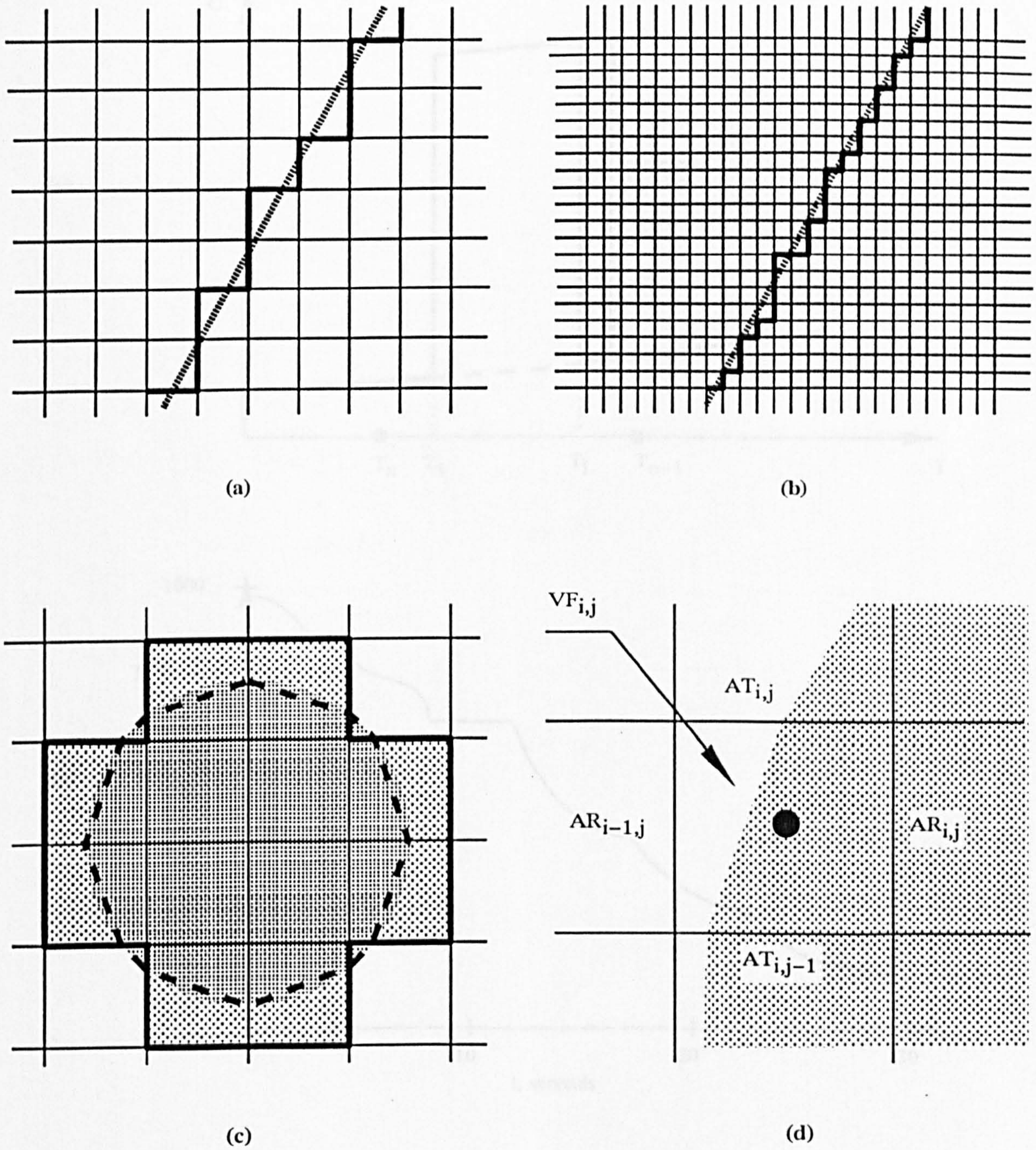


Figure 2.4. Line and surface disctretisation techniques: (a) and (b) show the conventional finite difference technique transforming a line (dotted line) into a stepwise line (heavy solid line); in both cases the discretised line length is the same. (c) The discretisation of a circle using the conventional technique (light shaded area) and the FAVOR method (dashed line). (d) Variables introduced by the FAVOR method; in this case $VF_{i.j}=0.35$, $AR_{i-1,j}=1.0$, $AR_{i,j}=0.0$, $AT_{i,j-1}=0.2$ and $AT_{i.j}=0.5$.

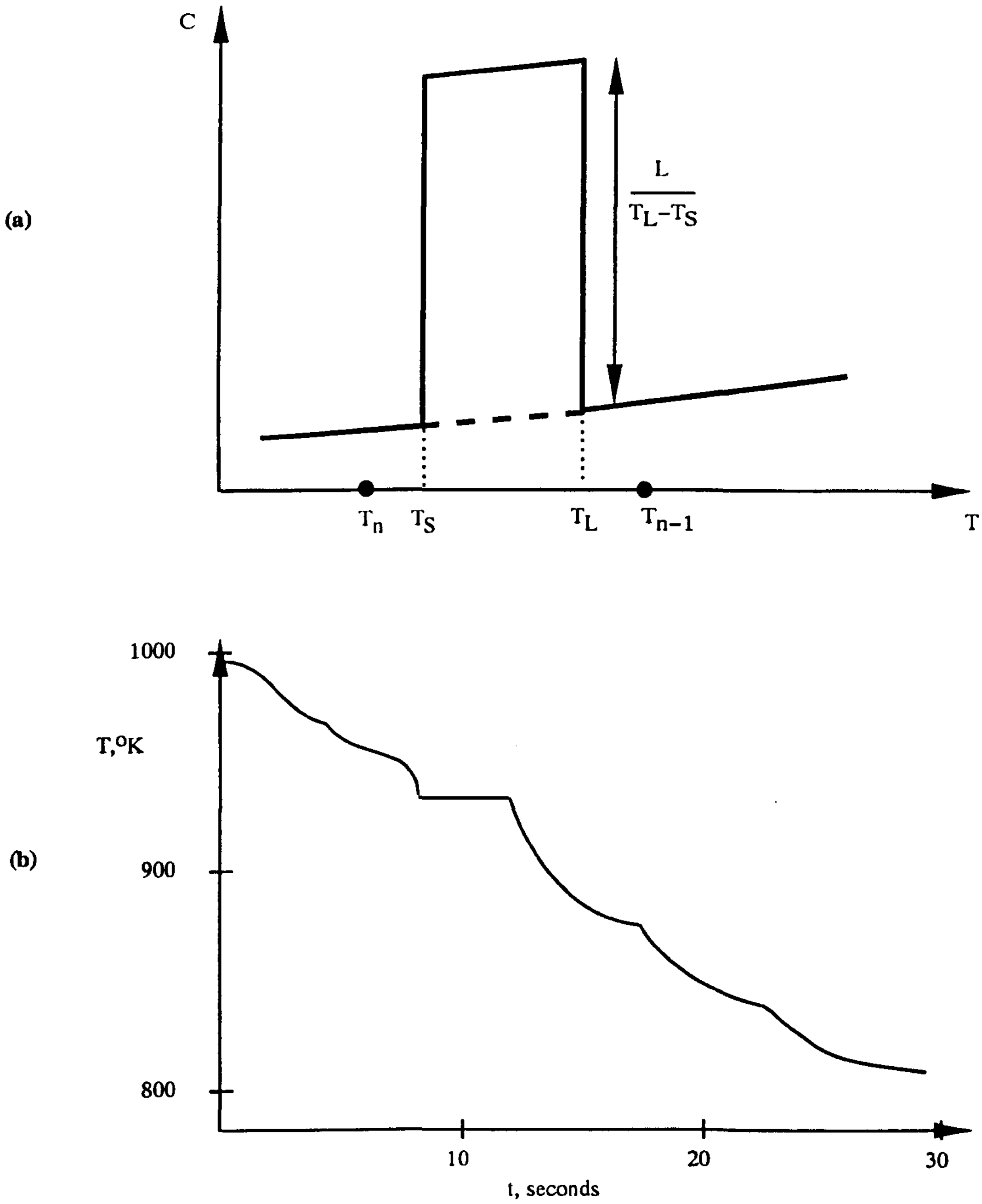


Figure 2.5. (a) The modified specific heat method (equation (72)). If the cell temperature 'jumps' from T_{n-1} to T_n in one time step, then the latent heat is not accounted for. (b) The 'heat wave' effect due to the use of the enthalpy method is shown on a simulated cell cooling curve. The distance between peaks corresponds to the solidification front passing through a cell. The solidification temperature is 933°K.

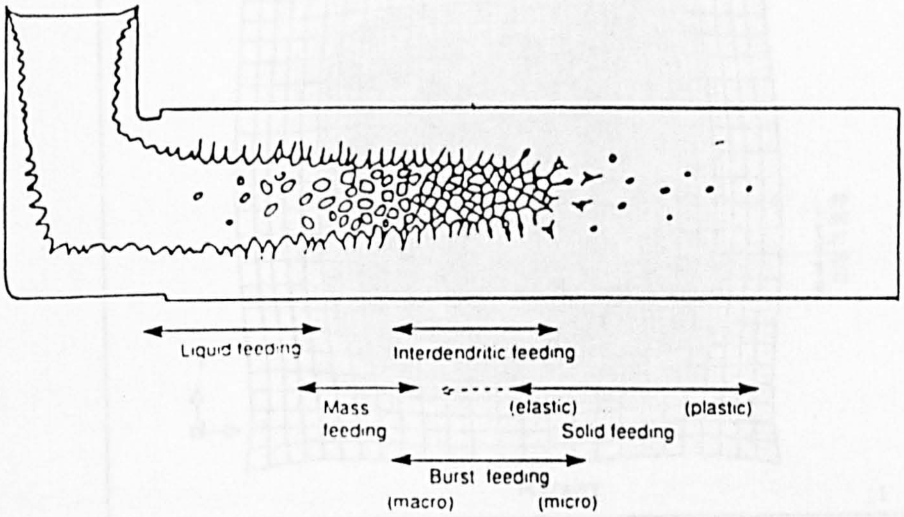


Figure 2.6. Classification of solidification shrinkage feeding flows [2].

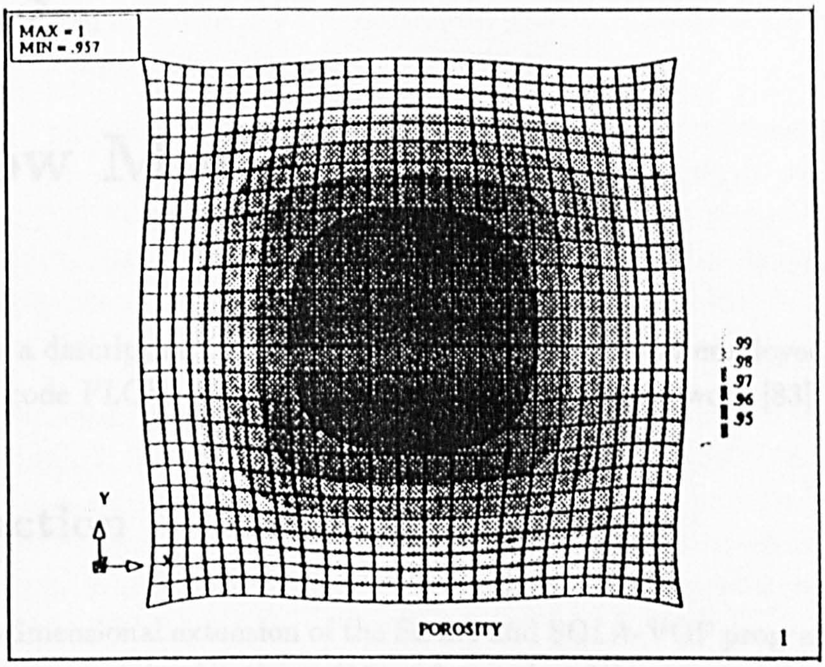


Figure 2.7. Predicted porosity and deformation in a 2-D square block using stress model [120].

Chapter 3

Fluid Flow Model

This chapter contains a description of the parts of the fluid flow model employed in the proprietary CFD code FLOW-3D that are relevant to the present work [83].

3.1 Introduction

The model is a three-dimensional extension of the SOLA and SOLA-VOF programs. Various input routines are contained in the code to aid users in setting up a mesh and in generating geometrically complicated boundary and initial conditions. Similarly, extensive graphics capabilities are provided to assist the user in understanding the results of calculations. All input variables, mesh, obstacles and graphical requests are specified by the user in the input file PREPIN.INP, an example of which is shown in Fig. 3.1.

Extensive tests have been performed on the isothermal fluid flow model. The tests were supported by experimental simulations using water. It was found that FLOW-3D was capable of modelling successfully fast moving flows typical for many casting configurations [84,122].

3.2 Governing Equations

The set of the hydrodynamical equations constituting the mathematical model for casting simulation consists of continuity Eq. (2.21) for an incompressible fluid, Navier-Stokes Eq. (2.22) for linear viscous fluid and the energy equation written in terms of enthalpy to model heat flow and phase transformation, Eqs. (2.17) and (2.23).

These equations will be solved with the following assumtions:

- thermal convection is excluded from consideration to simplify the analysis of the predicted shrinkage induced flow (Chapters 6 and 7), though there is a capability of modelling thermal convection in the code;
- undercooling and nucleation effects are also neglected assuming that solidification is solely controlled by the removal of the latent heat. In other words, $dG = 0$ (Eq. (2.19)) during phase transformation and liquid at the solidification front is always at the ideal melting point. This assumption simplifies greatly the mathematical modelling of solidification and is acceptable for simulations of macro-shrinkage effects in both pure metals and alloys.
- Eq. (2.20) for the solute evolution in a melt is not used in the present work. Instead, simple models like lever rule or Scheil equation (Eqs. (2.64), (2.65)) will be used to determine fraction of solid function. This assumption is partially justified since the choice of latent heat release model has little effect on the metal temperature and freezing time for alloys with narrow freezing range, *e.g.* low carbon steels [105].

3.2.1 Boundary Conditions

Eqs. (2.17), (2.21)-(2.23) are solved with boundary conditions given by Eqs. (2.24) and (2.25). In many cases of fluid flow during casting surface tension and viscous forces in Eq. (2.26) can be neglected. This is due to high Weber numbers during the filling stage (the ratio of the dynamic pressure to the surface tension force)

$$We = \frac{\rho R v^2}{\sigma} \gg 1$$

where R is the characteristic free surface curvature radius, and practically zero free surface curvature at later stages of solidification, *i.e.* high Bond numbers

$$Bo = \frac{g\rho R}{\sigma} \gg 1$$

which is the ratio of the gravitational force to the surface tension force[1].

[1] In aluninium based alloys σ may be large because of the formation of an oxide film on the metal surface [2].

Viscous stresses at the free surface are small compared to the fluid inertia, *i.e.* Reynolds number is large

$$Re = \frac{Lv\rho}{\mu} \gg 1,$$

and to the gravitational force, *i.e.* Galileo number is large

$$Ga = \frac{g\rho^2 L}{\mu^2} \gg 1$$

where L is the characteristic length scale of the free surface.

After neglecting the viscous and surface tension forces Eq. (2.26) for the fluid pressure at the free surface reduces to

$$p = p_g$$

A more rigorous statement than Eq. (2.27) is used to express the tangential viscous stresses at the free surface:

$$\frac{\partial v_n}{\partial l} = 0$$

$$\frac{\partial \mathbf{v}_\tau}{\partial n} = 0$$

where $\partial/\partial l$ is a gradient along an arbitrary direction l in the plane tangent to the free surface. This assumption substantially simplifies the setting of the viscous boundary conditions in the numerical model.

Further simplification of the mathematical model is made by assuming that the free surface is an adiabatic boundary, *i.e.* neglecting radiation and convective heat losses to the air,

$$\mathbf{q}_{fs} = 0$$

This may be a valid assumption for the following reasons:

a) the wall heat transfer coefficient is usually much larger than that of the free surface[2];

b) the total area of the free surface is usually much smaller than that of the metal mould interface;

c) often free surfaces only exist during the filling stage; if there is a free surface after filling but the air is confined by the mould walls then the air is heated up quickly, reducing the subsequent heat transfer by radiation and convection.

[2] Radiation may be important for high temperature metals and alloys, such as steels. Mould erosion has even been observed due to the radiation.

Eq. (2.28) is used for the heat flux per unit area at the metal/mould interface

$$q = h\left(T_{metal} - T_{mould}\right)$$

In many practical cases a constant value for h is a good approximation.

3.3 Numerical Model

The following computational methods are used to solve numerically Eqs. (3.1)-(3.3) in three dimensions:

1. Finite volume approach with structured staggered meshes consisting of rectangular non-uniform cells; the conservation equations are easily discretised using this method and the physical conservation laws are maintained in their numerical analogues. The physical transparency of the method and the resulting discretised equations simplifies the task of understanding and developing the model. FAVOR technique is used to represent cells partially blocked by the mould material [123].

2. all temporal derivatives are approximated by the first order accurate backward differencing; all spatial derivatives, except for the pressure gradients, are estimated at the 'old' time level, *i.e.* explicitly, introducing time step size limits to maintain stability of the numerical solution.

3. momentum advection terms are approximated using the upwind (donor cell) differencing method which, being of first order formal accuracy[3], possesses transportive and conservative properties though introduces numerical diffusion;

4. viscous and thermal diffusion terms are approximated by a central differencing, which first-order accurate in non-uniform grids;

5. as pressure gradients are estimated implicitly, discretised momentum and continuity Eqs. are solved simultaneously using Gauss-Seidel successive-over-relaxation (SOR) method. Though perhaps not the fastest, SOR iteration method offers great flexibility and sufficient efficiency in cases of complicated geometry and fluid configuration. The method also proves to be adequate for the numerical implementation of the shrinkage model, described in Chapter 5;

[3] Higher-order numerical discretisation methods for the advection terms in the momentum equations, *e.g.* based on central differencing or on the van Leer method [87], are also available in the code [83]

6. free surface is represented by the VOF method, which proved to be one of the most accurate technique to describe transient fluid interfaces; donor-acceptor advection scheme was chosen to solve the conservation equation for the fluid fraction function, Eq. (2.46), so that the sharpness of the free surface is preserved; all other scalar quantities, such as enthalpy, that are discontinuous across free surface are advected in the same way.

7. the enthalpy solidification model is completed either by lever rule, Eq. (2.64), or Scheil equation, Eq. (2.65). For pure materials a solidifying cell is kept at the melting point until it is fully solid;

8. metal/mould interfacial heat flux is estimated using a constant value of the heat transfer coefficient and using collapsed (zero volume) control volumes on both sides of the interface to represent the interfacial temperatures (Section 4.1).

3.3.1 General Outline of the Solution Method

Eqs. (2.21)-(2.23) are solved numerically using finite-volume approximations. The flow region is subdivided into a mesh of fixed rectangular cells of width Δx_i, depth Δy_j and height Δz_k.

Fluid velocities and pressures are located at staggered mesh locations as shown in Fig. 3.2:

- u-velocities and fractional areas A_x at the centres of cell faces normal to the x direction;
- v-velocities and fractional area A_y at the centres of cell faces normal to the y direction;
- w-velocities and fractional areas A_z at the centres of cell faces normal to the z direction;
- Scalar quantities, such as pressure, fluid fraction, fractional volume, density, enthalpy and temperature, are at cell centres.

To construct discrete numerical approximations to the governing equations, control volumes are defined surrounding each dependent variable location. For each control volume, surface fluxes, surface stresses and body forces can be computed in terms of surrounding variable values.

Most terms in the equations are evaluated using the current time-level values of the local variables. This produces a simple and efficient explicit computational scheme, though it requires a restrictive time step size to maintain computationally stable and accurate solution.

Pressures and velocities are coupled implicitly by using time-advanced pressures in the momentum equations and time-advanced velocities in the continuity equation and solved iteratively.

The procedure for advancing a solution through one time increment, Δt, consists of three steps:

1. Explicit approximations of the momentum equations by upwind differencing, described in Section 2.3.1, are used to compute the first guess for new time-level velocities using the initial conditions or previous time-level values for all velocities, pressure and other forces.
2. To satisfy the continuity equation the pressures are iteratively adjusted by Gauss-Seidel SOR or an optional ADI method, described in Section 2.3.2, in each cell; the velocity changes induced by each pressure change are added to the velocities computed in step (1).
3. Finally, when there is a free surface, it must be updated using Eq. (3.5) and the donor-acceptor advection method, described in Section 2.5, to give new fluid configuration. Energy values must be updated to reflect advective and diffusive processes.

Repetition of these steps will advance a solution through any desired time interval. At each step, of course, appropriate boundary conditions must be imposed at all mesh, obstacle and free-boundary surfaces.

3.3.2 Equation Discretisation

Curved obstacles, wall boundaries, or other geometric features are embedded in the mesh by defining the fractional face areas and fractional volumes of the cells that are open to flow. The fractional areas and volumes are incorporated into flow equations as shown below in Eqs. (3.1)-(3.4).

Flow equations can now be written in the following form using Cartesian coordinates:

the continuity rquation

$$\frac{\partial}{\partial x}(u\,A_x) + \frac{\partial}{\partial y}(v\,A_y) + \frac{\partial}{\partial z}(w\,A_z) = 0 \qquad (3.1)$$

momentum equations (with constant viscosity for simplicity)

$$\frac{\partial u}{\partial t} + \frac{1}{V_F}\left[u\,A_x\frac{\partial u}{\partial x} + v\,A_y\frac{\partial u}{\partial y} + w\,A_z\frac{\partial u}{\partial z}\right] = \frac{1}{\rho_0}\frac{\partial p}{\partial x} + G_x - D_x$$
$$+\nu\left\{2\frac{\partial}{\partial x}\left[A_x\frac{\partial u}{\partial x}\right] + \frac{\partial}{\partial y}\left[A_y\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right] + \frac{\partial}{\partial z}\left[A_z\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)\right]\right\} \qquad (3.2)$$

$$\frac{\partial v}{\partial t} + \frac{1}{V_F}\left[u\,A_x\frac{\partial v}{\partial x} + v\,A_y\frac{\partial v}{\partial y} + w\,A_z\frac{\partial v}{\partial z}\right] = \frac{1}{\rho_0}\frac{\partial p}{\partial y} + G_y - D_y$$
$$+\nu\left\{2\frac{\partial}{\partial y}\left[A_y\frac{\partial v}{\partial y}\right] + \frac{\partial}{\partial x}\left[A_x\left(\frac{\partial v}{\partial x} + \frac{\partial u}{\partial y}\right)\right] + \frac{\partial}{\partial z}\left[A_z\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)\right]\right\} \qquad (3.3)$$

$$\frac{\partial w}{\partial t} + \frac{1}{V_F}\left[u\,A_x\frac{\partial w}{\partial x} + v\,A_y\frac{\partial w}{\partial y} + w\,A_z\frac{\partial w}{\partial z}\right] = \frac{1}{\rho_0}\frac{\partial p}{\partial z} + G_z - D_z$$
$$+\nu\left\{2\frac{\partial}{\partial z}\left[A_z\frac{\partial w}{\partial z}\right] + \frac{\partial}{\partial y}\left[A_y\left(\frac{\partial v}{\partial z} + \frac{\partial w}{\partial y}\right)\right] + \frac{\partial}{\partial x}\left[A_x\left(\frac{\partial u}{\partial z} + \frac{\partial w}{\partial x}\right)\right]\right\} \qquad (3.4)$$

where (G_x, G_y, G_z) denotes body forces and $\mathbf{D} = (D_x, D_y, D_z)$ is the drag force employed to model flow in the porous media and is written in a general form

$$\mathbf{D} = K \cdot \mathrm{v}, \quad K \geq 0 \qquad (3.5)$$

Addition of a drag force proportional to the first power of velocity has many useful applications. It allows the drag force to be approximated in the numerical equations implicitly so that no additional restriction on the time step size is imposed.

The enthalpy equation is written as

$$V_F\frac{\partial H}{\partial t} + \frac{\partial}{\partial x}(uA_xH) + \frac{\partial}{\partial y}(vA_yH) + \frac{\partial}{\partial z}(wA_zH)$$
$$= \frac{1}{\rho_0}\left[\frac{\partial}{\partial x}\left(kA_x\frac{\partial T}{\partial x}\right) + \frac{\partial}{\partial y}\left(kA_y\frac{\partial T}{\partial y}\right) + \frac{\partial}{\partial z}\left(kA_z\frac{\partial T}{\partial z}\right)\right] \qquad (3.6)$$

and, finally, the equation for the volume of fluid fraction function F

$$V_F\frac{\partial F}{\partial t} + \frac{\partial}{\partial x}(uA_xF) + \frac{\partial}{\partial y}(vA_yF) + \frac{\partial}{\partial z}(wA_zF) = 0 \qquad (3.7)$$

For a one-fluid flow F represents the *open* volume fraction occupied by the fluid. For a cell with dimensions dx, dy, dz and total volume

$$V_0 = dx\,dy\,dz$$

the open volume is

$$V_{open} = V_F \cdot V_0$$

and the cell fluid volume is

$$V_{fluid} = F \cdot V_{open} = F \cdot V_F \cdot dx\,dy\,dz$$

Thus, fluid of constant density exists where $F = 1$, and void regions correspond to locations where $F = 0$. "Voids" are regions without fluid mass that have a uniform pressure assigned to them. Physically, they represent regions filled with a vapor of gas whose density is insignificant with respect to the fluid density.

When Eqs. (3.1)-(3.7) are applied to a partially blocked rectangular control volume cell, then V_F, A_x, A_y and A_z represent the fractional open-to-flow volume and face areas of the cell, respectively. New terms have to be introduced on the right-hand side of momentum Eqs. (3.2)-(3.4), ws_x, ws_y, ws_z, associated with the wall shear stress. If these terms are omitted then there is no wall shear stress because the remaining terms contain the fractional flow areas (A_x, A_y, A_z) which vanish at walls.

The energy equation for the solid obstacle (mould) contain the complements of V_F and area fractions:

$$(1 - V_F)\,\rho_m C_m \frac{\partial T_m}{\partial t} = \frac{\partial}{\partial x}\left(k_m(1 - A_x)\frac{\partial T_m}{\partial x}\right) + \frac{\partial}{\partial y}\left(k_m(1 - A_y)\frac{\partial T_m}{\partial y}\right) + \frac{\partial}{\partial z}\left(k_m(1 - A_z)\frac{\partial T_m}{\partial z}\right) \tag{3.8}$$

where subscript m indicates the mould parameters.

The use of the FAVOR method in numerical modelling influences the time step size stability limit described in Section 2.10. For example, the CFL criterion (Eq. (2.75)) transfers into

$$\Delta t < V_F \cdot min_{i,j,k}\left(\frac{\Delta x_i}{A_{x_{i,j,k}} u_{i,j,k}}, \frac{\Delta y_j}{A_{y_{i,j,k}} v_{i,j,k}}, \frac{\Delta z_k}{A_{z_{i,j,k}} w_{i,j,k}}\right) \tag{3.9}$$

It can be seen from Eq. (3.14) that cells with small fractional volume and large face areas open to the flow can introduce very small time step size limits. Normally, however, the small V_F/A ratio is balanced by small velocities at such faces.

Fractional volume and areas in each cell are estimated at the preprocessing stage and then used during the calculation. Within each cell containing metal/mould interface the mould surface is assumed to be flat. FAVOR method allows the program to calculate viscous shear stresses and metal/mould heat transfer more accurately than if a stepwise approach to geometry description was used, since the fluid flow direction at the walls, the wall location and area in each cell are estimated with better accuracy.

A generic form for the finite-difference approximation of momentum Eq. (3.2), for example, is

$$\begin{aligned} u_{i,j,k}^{n+1} &= u_{i,j,k}^{n} + \Delta t_{n+1} \cdot (- \frac{p_{i+1,j,k}^{n+1} - p_{i,j,k}^{n+1}}{\rho \Delta x_{i+\frac{1}{2}}} \\ &+ G_x - FUX - FUY - FUZ + VISX - ws_x) \end{aligned} \tag{3.10}$$

where, $\Delta x_{i+\frac{1}{2}} = (\Delta x_i + \Delta x_{i+1})/2$.

The advective, viscous and body force terms have an obvious meaning, *e.g.* FUX means the advective flux of u in the x direction; $VISX$ is the x-component viscous force; G_x includes gravitational and other body forces and ws_x is the viscous wall force in the x direction.

The advective and viscous terms are all evaluated using old-time level (n) values for velocities (wall shear stresses are implicitly evaluated as described below in this Section). Old-time pressure values p^n are used to get a first guess for the new velocities.

Specific approximations chosen for the various acceleration terms in Eq. (3.2) are relatively unimportant and have been described in Section 2.3. In FLOW-3D a modified donor-cell approximation has been developed that retains its accuracy in a variable mesh and reduces to a conservative difference expression when the mesh is uniform [51,123]. This method approximates advective fluxes in the nonconservative form $u \cdot \nabla u$. Then the general form of this approximation is

$$\begin{aligned} FUX &= \frac{1}{2 \cdot V_{FC}} \left[(u_R - \mid u_R \mid) \cdot \frac{u_{i+1,j,k} - u_{i,j,k}}{\Delta x_{i+1}} \right. \\ &+ \left. (u_L + \mid u_L \mid) \cdot \frac{u_{i,j,k} - u_{i-1,j,k}}{\Delta x_i} \right] \end{aligned} \tag{3.11}$$

where

$$\begin{aligned} u_R &= 0.5 \cdot (u_{i+1,j,k} A_{FR,i+1,j,k} + u_{i,j,k} A_{FR,i,j,k}) \\ u_L &= 0.5 \cdot (u_{i,j,k} A_{FR,i,j,k} + u_{i-1,j,k} A_{FR,i-1,j,k}) \end{aligned}$$

and

$$V_{FC} = \frac{\Delta x_i V_{F,i,j,k} + \Delta x_{i+1} V_{F,i+1,j,k}}{\Delta x_i + \Delta x_{i+1}}$$

All other acceleration terms in the momentum equations are approximated by standard central differences. The difference techniques for viscous diffusion-processes are fairly straightforward. First order differences are applied to velocity components to obtain local shear rates, which are then multiplied by averaged cell-faced area fractions. The results are then differenced to approximate the net viscous stress. All quantities are evaluated explicitly in these calculations.

The approach for the wall stress in the w-velocity equation, for example, is as follows. Wall shears influencing w can arise from wall areas located on x and y cell faces surrounding w. For any one of these faces, if the fractional area A is less than unity, the remaining area fraction $(1 - A)$ is considered to be a wall on which a stress is generated. On an x face to the right of w, for instance, the force due to wall shear, ws_z, is

$$ws_z = \frac{\partial}{\partial x}\left(\mu \frac{\partial w}{\partial x}\right) \approx -\frac{2\mu(1 - A_x)(w - w_0)}{A_z \Delta x^2}$$

where A_z and Δx are evaluated in the cell in which w is located. A_x is an average for the cells between which w is located. The velocity w_0 is zero at any interior obstacle boundary, but at a mesh boundary it is equal to the z-direction tangential velocity of the boundary.

Similar stresses are evaluated at each of the four surrounding cell walls and their sum is taken as the total stress. Wall stresses are included in an implicit way to avoid possible numerical instabilities arising in cells with large wall areas and small flow volumes. Although this makes the momentum equation for w implicit, it is trivial to solve since it is linear in w and is not coupled to equations for other cells.

3.3.3 Pressure Solution Algorithm: Incompressible SOR Method

Momentum equations can be rewritten in the following form

$$u^{n+1}_{i,j,k} = \tilde{u}^n_{i,j,k} + \Delta t_{n+1} \cdot \left(-\frac{p^{n+1}_{i+1,j,k} - p^{n+1}_{i,j,k}}{\rho \Delta x_{i+\frac{1}{2}}} - K^n_{i+\frac{1}{2},j,k} u^{n+1}_{i,j,k} \right) \quad (3.12)$$

$$v^{n+1}_{i,j,k} = \tilde{v}^n_{i,j,k} + \Delta t_{n+1} \cdot \left(-\frac{p^{n+1}_{i,j+1,k} - p^{n+1}_{i,j,k}}{\rho \Delta y_{j+\frac{1}{2}}} - K^n_{i,j+\frac{1}{2},k} v^{n+1}_{i,j,k} \right) \quad (3.13)$$

$$w_{i,j,k}^{n+1} = \tilde{w}_{i,j,k}^{n} + \Delta t_{n+1} \cdot \left(-\frac{p_{i,j,k+1}^{n+1} - p_{i,j,k}^{n+1}}{\rho \Delta z_{k+\frac{1}{2}}} - K_{i,j,k+\frac{1}{2}}^{n} w_{i,j,k}^{n+1} \right) \tag{3.14}$$

where, *e.g.*

$$K_{i+\frac{1}{2},j,k}^{n} = \frac{2\, K_{i,j,k}^{n} \cdot K_{i+1,j,k}^{n}}{K_{i,j,k}^{n} + K_{i+1,j,k}^{n}}$$

and $\tilde{u}$, $\tilde{v}$ and $\tilde{w}$ include all explicitly evaluated terms and the wall shear stresses.

For incompressible flows the continuity Eq. (3.1) can be interpreted as an elliptic condition on the cell pressures and velocities. Velocities computed from Eqs. (3.12)-(3.14) must satisfy the following discretised approximation of continuity Eq. (3.1)

$$\frac{u_{i,j,k}^{n+1} A_{FR,i,j,k} - u_{i-1,j,k}^{n+1} A_{FR,i-1,j,k}}{\Delta x_i} + \frac{v_{i,j,k}^{n+1} A_{FB,i,j,k} - v_{i,j-1,k}^{n+1} A_{FB,i,j-1,k}}{\Delta y_j} + \frac{w_{i,j,k}^{n+1} A_{FT,i,j,k} - w_{i,j,k-1}^{n+1} A_{FT,i,j,k-1}}{\Delta z_k} = 0 \tag{3.15}$$

Eqs. (3.12)-(3.15) must be solved simultaneously for all $(n+1)$-level parameters, *i.e.* velocities and pressures. This is done using a successive over-relaxation method. The computational mesh is swept cell by cell starting with the first non-boundary cell. Calculations are only performed in cells that contain fluid and have no empty neighbours. The pressure change needed to make the velocities in cell (i,j,k) satisfy Eq. (3.15) is

$$\Delta p_{i,j,k} = -\frac{L_{i,j,k}}{\partial L_{i,j,k} / \partial p_{i,j,k}} \tag{3.16}$$

where L is the left-hand side of Eq. (3.15) and

$$\begin{aligned}\frac{\partial L_{i,j,k}}{\partial p_{i,j,k}} = \frac{\Delta t}{\rho_0} \cdot \Bigg[& \frac{1}{\Delta x_i} \left(\frac{A_{FR,i,j,k}}{\Delta x_{i+\frac{1}{2}}(1 + K_{i+\frac{1}{2},j,k}^{n} \Delta t)} + \frac{A_{FR,i-1,j,k}}{\Delta x_{i-\frac{1}{2}}(1 + K_{i-\frac{1}{2},j,k}^{n} \Delta t)} \right) \\ + & \frac{1}{\Delta y_j} \left(\frac{A_{FB,i,j,k}}{\Delta y_{j+\frac{1}{2}}(1 + K_{i,j+\frac{1}{2},k}^{n} \Delta t)} + \frac{A_{FB,i,j-1,k}}{\Delta y_{j-\frac{1}{2}}(1 + K_{i,j-\frac{1}{2},k}^{n} \Delta t)} \right) \\ + & \frac{1}{\Delta z_k} \left(\frac{A_{FT,i,j,k}}{\Delta z_{k+\frac{1}{2}}(1 + K_{i,j,k+\frac{1}{2}}^{n} \Delta t)} + \frac{A_{FT,i,j,k-1}}{\Delta z_{k-\frac{1}{2}}(1 + K_{i,j,k-\frac{1}{2}}^{n} \Delta t)} \right) \Bigg] \end{aligned} \tag{3.17}$$

(see also Eqs. (2.39)-(2.45)). Eq. (3.16) is simply a Newton type of relaxation process that will produce the value of p needed to make $L = 0$. Then for u-velocities, for example,

$$u_{i,j,k}^{n+1} \longrightarrow u_{i,j,k}^{n+1} + \frac{\Delta p_{i,j,k}}{\rho_0 \, \Delta x_{i+\frac{1}{2},j,k}} \cdot \frac{\Delta t^{n+1}}{1 + \Delta t^{n+1} K_{i+\frac{1}{2},j,k}^{n}} \tag{3.18}$$

$$u_{i-1,j,k} \longrightarrow u_{i-1,j,k} - \frac{\Delta p_{i,j,k}}{\rho_0 \Delta x_{i-\frac{1}{2},j,k}} \cdot \frac{\Delta t^{n+1}}{1 + \Delta t^{n+1} K^n_{i-\frac{1}{2},j,k}} \qquad (3.19)$$

In each cell the velocity values used in evaluating L and in Eqs. (3.18), (3.19) are most current values available during the iteration process. To start the iteration process, the new estimated velocities from Eqs. (3.2)-(3.14) are used with the pressures remaining from the previous time step. Velocities located at zero area faces are not adjusted.

In cells containing a free surface, that is, a cell containing fluid, but with one or more empty neighbours, a different procedure is used (Section 2.4). The boundary condition wanted in these cells is that the pressure is a specified value, p_s, at the surface. The surface pressure is set equal to the neighbouring void region pressure, P_g,

$$p_{s_{i,j,k}} = P_g(n) \qquad (3.20)$$

where n is the index of the adjacent void region. This pressure is then imposed on the solution by extrapolating it to the pressure, $p_{i,j,k}$, at the centre of the surface cell assuming a hydrostatic distribution within the cell (Eq. (2.49)). The hydrostatic variation depends on the net body force in the direction normal to the surface. This surface pressure is not changed during the pressure iteration, but treated as a fixed boundary value[4].

A complete iteration consists of adjusting pressures and velocities in all full cells according to Eqs. (3.16), (3.18) and (3.19). Convergence of the iteration is achieved when all cells have $L_{i,j,k}$ values whose magnitudes are below some small number, $\epsilon \cdot V_{F,i,j,k}$. Typically, ϵ is of order 10^{-3} s^{-1}, although it can vary with the specific problem being solved.

Convergence of the iteration is accelerated by multiplying Δp in Eq. (3.16) by an over-relaxation factor, ω. A value of ω equal to 1.7 or 1.8 is often optimum.

3.3.4 Fluid Configuration

The donor-acceptor advection method (Section 2.5) with the newly obtained velocities is employed to find the new time level fluid configuration by solving Eq. (3.7). Other scalar quantities are advected using the same method as described in Section 2.6, *i.e.* if δF is the quantity of fluid advected in x-direction from cell (i, j, k) to

[4] The fact that surface cell pressures do not take part in the iteration process effectively means that surface pressure is evaluated *explicitly*. This may lead to a numerical instability if a void pressure varies with the void volume as $p\,V^\gamma = const$, where γ is a constant parameter [85]

cell $(i+1, j, k)$ over the time Δt per unit cross-sectional area, then the amount of enthalpy advected through the same cell face is

$$\delta(\rho H) = (\rho H)_{i,j,k} \cdot \delta F \cdot A_{FR,i,j,k} \, \Delta y_j \, \Delta z_k \tag{3.21}$$

The new F values occasionally have values slightly less than zero or slightly greater than unity. Therefore, after the scalar advection calculations have been completed, a pass is made through the mesh to reset values of F less than zero back to zero and values of F greater than one back to one. Accumulated changes in fluid volume introduced by these adjustments during a calculation are recorded and printed out. The total fluid volume is also printed. Volume errors after hundreds of cycles are usually a fraction of a percent of the total fluid volume.

For application of free surface boundary conditions and advection of the F function, it is necessary to assign an approximate normal direction to the surface. The free surface in each surface cell is approximated by a plane normal to one of the coordinate directions so that the normal points in the direction of an empty neighbour cell. If the surface cell has more than one empty neighbour, then the direction selected is the one having the largest F value in the opposite neighbour cell.

Once the inward normal direction has been determined, the surface *location* is defined by a flat surface normal to this coordinate direction that extends to the proper height.

3.3.5 Heat Conduction and Heat Transfer

A two-domain method is also used in the present work, but energy Eqs. for the metal and mould are coupled at the same time level and solved simultaneously. The latter is simplified by the use of a fully explicit formulation of the numerical analogue equations. For a control volume P (which in this case conincides with a mesh cell) the heat flux δq per unit area at each face is calculated as

$$\delta q = k_a \frac{T_P - T_{adj}}{\Delta x}$$

where T_{adj} is the temperature of the adjacent control volume across the face, k_a is the harmonic average of thermal conductivities of the two cells

$$k_a = \frac{2 k_P k_{adj}}{k_P + k_{adj}} \tag{3.22}$$

and Δx is the distance between the two cell centres. The use of the harmonic average, rather than the arithmetic one, is generally a more accurate procedure, more obvious so in a case when the values of k_P and k_{adj} differ substantially. In the present work thermal conductivity varies between: solid and liquid phases in metal, and different materials in the mould.

If a cell face includes a metal/mould interface than the heat flux is estimated by using Eq. (2.56). The same flux δq (times the face area) is used for both cells sharing the face, thus, ensuring the conservation property. The total heat flux into the cell is equal to the sum of all the conductive and interfacial fluxes multiplied by corresponding areas.

Thermal conduction and heat transfer terms appear in both the fluid and obstacle energy Eqs. (3.6) and (3.8). These terms are treated in analogus ways. The numerical analogue of Eq. (3.6), with forward time differencing and first order spatial approximations, is

$$\begin{aligned} V_{F,i,j,k}\,\rho_0\,\frac{F^{n+1}_{i,j,k}\,H^{n+1}_{i,j,k} - F^{n+1}_{i,j,k}\,H^{n}_{i,j,k}}{\Delta t^{n+1}} &= X^n_{i,j,k} \\ + \sum_{faces} \frac{A \cdot k}{\Delta}\,(T^n_{i,j,k} - T^n_{adjacent}) &+ h_{i,j,k}\,(WA)_{i,j,k}\,(T^n_{i,j,k} - TW^n_{i,j,k}) \end{aligned} \quad (3.23)$$

where X represents advection terms, T and TW are cell-centred fluid and obstacle temperatures respectively, h heat transfer coefficient, WA an interfacial area, A a cell face area, k an averaged heat transfer coefficient (Eq. (3.23)) and Δ an appropriate spatial increment.

The same procedure is employed when solving the solid energy Eq. (3.8) wall temperatures.

```
2-D Cylinder: Shrinkage Modelling.
 $xput
  remark='computational parameters',
    icolor=1,  autot=2, epsi=0.0001,  twfin=1100.0,
    delt=1,    itb=1,   pltdt=100,    prtdt=10000.0,
    ifenrg=2,  ihtc=2,  ipdis=1,
    cyl=1.0,   ishr=1,
  remark='physical properties',
     gz=-980.0,   pcav=-100000.0,
     rhof=2.54,   rhofs=2.71,
  remark='thermal properties',
     tl1=933,       ts1=933, clht1=3.97+09,
     cv1=0.896+07,  cvs1=1.17e+07,
     thc1=1.0e+07,  thcs1=2.25e+07,
  remark='boundary conditions',
    wl=1, wr=2, wf=1, wbk=1, wb=2, wt=2,
    tbcd=293.0, tbct(1,2)=293.0, tbct(1,5)=293.0, tbct(1,6)=293.0,
 $end
 $mesh
   nxcelt=12,
     px(1)=0.0, nxcell(1)=7,
     px(2)=4.9, sizex(2)=0.7,
     px(3)=12.0,
   nycelt=1,
     py(1)=0.0,
     py(2)=1.0,
   nzcelt=25,
     pz(1)=6.0,
     pz(2)=13.6, sizez(2)=0.7, nzcell(2)=20,
     pz(3)=30.0,
 $end
 $obs
   nobs=2,
     iofo(1,1)=1,
       cc(1)=-1.0, rah(1)=4.9, zh(1)=13.6,
     iofo(1,2)=2,
       cc(2)=-1.0, ral(2)=4.9,
      kobs(1)=60000,      kobs(2)=60000,
      rcobs(1)=1.7e+07,   rcobs(2)=1.7e+07,
      hobs1(1)=1.5e+06,   hobs1(2)=1.5e+06,
 $end
 $fl
   flht=28.6,
 $end
 $bf
 $end
 $temp
      tempi=1023,
 $end
 $motn
 $end
 $grafic
     ncplts=3,
     yc1(1)=0.5, yc2(1)=0.5, kontyp(1)=6,
     yc1(2)=0.5, yc2(2)=0.5, kontyp(2)=5,
     yc1(3)=0.5, yc2(3)=0.5, kontyp(3)=5, ictyp(3)=1,
     nvplts=1,
      yv1(1)=0.5, yv2(1)=0.5,
     xloc(1)=0.001, yloc(1)=0.001, zloc(1)=21.8,
 $end
 $parts
 $end
```

Problem header

Numerical and physical parameters and properties, specification of the numerical model required for the simulation.

Mesh specifications

Obstacle (mould) geometry and physical properties.

Initial fluid configuration

Baffle specifications

Initial temperature distribution

Non-inertial forces

Graphical requests

Marker particales specifications

Figure 3.1. An example of FLOW-3D input file PREPIN.INP

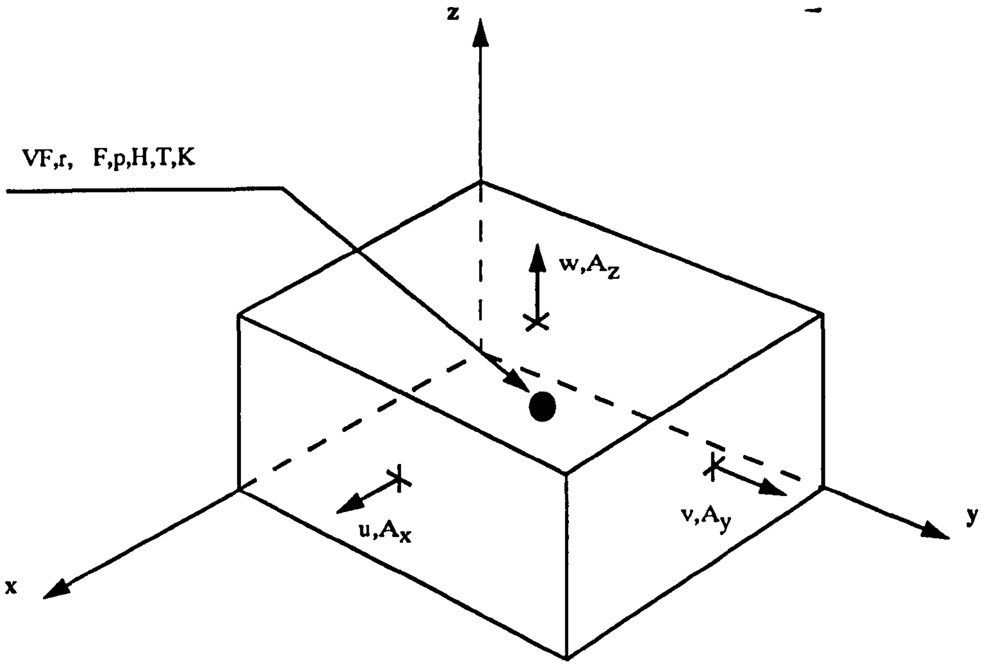


Figure 3.2. Location of variables in a FLOW-3D mesh cell

Chapter 4

Heat Transfer and Solidification

This chapter contains modifications of FLOW-3D introduced to improve the accuracy of its metal/mould heat transfer and solidification models.

4.1 Heat Transfer Algorithm

Care must be taken to resolve areas where the flow variables have steep gradients. Discontinuity surfaces and internal boundaries represent such areas. VOF method, for example, allows to resolve free surface boundary *inside* a computational mesh cell and set the boundary conditions in the cell according to the amount of fluid in it and the surface orientation. Another example is the use of the FAVOR method to describe the obstacle surfaces inside the mesh cells. The FAVOR method requires much less cells to represent a sphere, for example, than if a stepwise approach was employed.

Further use of the FAVOR method has been made to improve the accuracy of the metal/mould heat transfer calculations, taking into account that cell-centred nodes do not, in general, lie on the interface, Assume that temperature profile between the metal/mould interface and the nearby cell node is linear on both sides of the interface. In this case if dx_1 is the distance from a fluid interfacial cell node to the interface and dx_2 is the distance from a corresponding mould interfacial cell node to the interface, then the effective heat transfer coefficient for those nodes is [92]

$$h_{eff} = \frac{1}{\frac{dx_1}{k_1} + \frac{1}{h} + \frac{dx_2}{k_2}}$$

where k_1 and k_2 are the metal and mould thermal conductivities respectively and

$1/h$ is the interfacial thermal resistance. Then the heat flux across the interface can be calculated using the metal and mould cell temperatures T_1 and T_2, located at the nodes, as

$$q = h_{eff}\,(T_1 - T_2)$$

This procedure gives a higher-order accurate approximation to the interfacial boundary conditions than if h was used to estimated the flux. As will be shown below in this Section, this approach is equivalent to introducing two additional, zero-volume cells on each side of the interface. The idea has been first introduced and successfully applied in conjunction with finite element method [90,5].

Consider two neighbouring cells, one of which is completely open to the flow ($V_F = 1$) and the other completely blocked ($V_F = 0$), so that their common face represent the metal/mould interface. According to the control volume approach, T_1 is the average fluid temperature in cell 1 and T_2 is the average mould temperature in cell 2 (Fig. 4.1a). Both temperatures are located at the cell centres. The x-axis is chosen to be normal to the interface positioned at $x = 0$.

In the basic method the flux per unit area between the two cells is calculated as

$$Q_1 = h\,(T_1 - T_2) \tag{4.1}$$

The actual interfacial boundary condition, given by Eq. (2.56), involves metal and mould *surface* temperatures, T_{s1} and T_{s2} respectively

$$Q_0 = h\,(T_{s1} - T_{s2}) \tag{4.2}$$

with $1/h$ representing the interface thermal resistance. For the case shown in Fig. 4.1a, T_{s1} is located at the right face centre of cell 1 and T_{s2} is located at the left face centre of cell 2. To simplify the analysis, suppose that

$$T_1 = T_{s1} \tag{4.3}$$

To estimate the accuracy of Eq. (4.1), we will expand T_2 in Taylor series about $x = 0$:

$$T_2 = T_{s2} + \frac{1}{2}\frac{\partial T}{\partial x}_{|x=0}\Delta x + \frac{1}{4}\frac{\partial^2 T}{\partial x^2}_{|x=0}\Delta x^2 + O(\Delta x^3) \tag{4.4}$$

where Δx is the size of cell 2. Substituting T_2 from Eq. (4.4) into Eq. (4.1) and taking into account Eqs. (4.2) and (4.3) yields

$$\begin{aligned} Q_1 &= h\cdot\left(T_{s1} - T_{s2} - \frac{1}{2}\frac{\partial T}{\partial x}_{|x=0}\Delta x + O(\Delta x^2)\right) \\ &= Q_0 - h\,\frac{1}{2}\frac{\partial T}{\partial x}_{|x=0}\Delta x + O(\Delta x^2) \\ &= Q_0\left(1 + \frac{1}{2}h\frac{\Delta x}{k}\right) + O(\Delta x^2) \end{aligned} \tag{4.5}$$

where k is the mould conductivity and

$$Q_0 = -k \frac{\partial T}{\partial x}_{|x=0} \tag{4.6}$$

It is clear from Eq. (4.5) that boundary condition approximation given by Eq. (4.1) is of the first order formal accuracy with respect to the cell size Δx. For low thermal conductivity mould materials, such as sands, the first order term on the right-hand side of Eq. (4.5) may appear to be larger than the zeroth order term[1]. The heat flux Q_1 is *overestimated* compared to Q_0 because the zeroth and first order terms have the same sign.

The basic, first order heat transfer method represents metal/mould interface by a row of cells with a non-zero thickness. Therefore, the interface has a thermal capacity proportional to the layer thickness and the mould specific heat. Since interfacial cell sizes and mould geometry vary across the computational domain, the value of Q_1 becomes strongly mesh dependent.

The idea of the modified algorithm is to represent the interface by a zero-thickness surface with zero thermal capacity. To realise this idea, a new control volume of thickness δx (cell 2') is introduced *inside* cell 2, along the interface, as shown in Fig. 4.1*b*, and temperature $T_{2'}$ is assigned to the new cell. The heat balance between the interface, cell 2' and cell 2 is

$$\delta x \, \rho C \frac{dT_{2'}}{dt} = k \frac{T_2 - T_{2'}}{\frac{1}{2}\Delta x} - h(T_{2'} - T_{s1}) \tag{4.7}$$

$$(\Delta x - \delta x) \rho C \frac{dT_2}{dt} = -k \frac{T_2 - T_{2'}}{\frac{1}{2}\Delta x} \tag{4.8}$$

where thermal conduction heat transfer occurs between cells 2 and 2'. If the thickness of cell 2' is put to zero, $\delta x \rightarrow 0$, then $T_{2'}$ will be located exactly at the interface and Eqs. (4.7), (4.8) can be rewritten as

$$0 = k \frac{T_2 - T_{2'}}{\frac{1}{2}\Delta x} - h(T_{2'} - T_{s1}) \tag{4.9}$$

$$\Delta x \, \rho C \frac{dT_2}{dt} = -k \frac{T_2 - T_{2'}}{\frac{1}{2}\Delta x} \tag{4.10}$$

The interfacial temperature $T_{2'}$ can be expressed as a function of T_{s1} and T_2 using Eq. (4.9)

$$T_{2'} = \frac{h T_{s1} + \frac{k}{\frac{1}{2}\Delta x} T_2}{h + \frac{k}{\frac{1}{2}\Delta x}} \tag{4.11}$$

[1] For silica sand the typical values are $h = 1.5 \times 10^3 \; Watt \, m^{-2} K^{-1}$, $k = 0.6 \; Watt \, m^{-1} K^{-1}$ and $\Delta x = 5 \; mm$, and the first order term on the right-hand side of Eq. (4.5) is over six times larger than the zeroth order term.

and Eq. (4.10) transforms into

$$\Delta x \, \rho \, C \frac{dT_2}{dt} = h_{eff} \, (T_{s1} - T_2) = Q_2 \tag{4.12}$$

where

$$h_{eff} = \frac{h \, \frac{k}{\frac{1}{2}\Delta x}}{h + \frac{k}{\frac{1}{2}\Delta x}} \tag{4.13}$$

is the effective heat transfer coefficient for the cell-centred temperature T_2. The expression for h_{eff} takes into account the resistance of the interface and of the mould material lying between the interface and the location of T_2.

It follows from Eqs. (4.4), (4.6), (4.12) and (4.13) that

$$\begin{aligned} Q_2 &= h_{eff} \cdot (T_{s1} - T_2) \\ &= h_{eff} \cdot \left(T_{s1} - T_{s2} - \frac{1}{2} \frac{\partial T}{\partial x}_{|x=0} \Delta x \; - \; \frac{1}{4} \frac{\partial^2 T}{\partial x^2}_{|x=0} \Delta x^2 \; - \; O(\Delta x^3) \right) \\ &= Q_0 - \frac{1}{4} h_{eff} \cdot \frac{\partial^2 T}{\partial x^2}_{|x=0} \Delta x^2 + O(\Delta x^3) = Q_0 + O(\Delta x^2) \end{aligned} \tag{4.14}$$

Eq. (4.14) shows that Q_2 is a second order accurate approximation to Q_0. Besides, the second order term is proportional to $\frac{\partial^2 T}{\partial x^2}$ which is normally much smaller than the first order derivative $\frac{\partial T}{\partial x}$ in Eq. (4.5).

Furthermore, for the error function solution for a one-dimensional heat flow into a semi-infinite media, $\frac{\partial^2 T}{\partial x^2}_{|x=0} = 0$ and the accuracy of Q_2 increases to a third order. In the example above this analytical solution exists if $T_{s1} = const$ and $h \rightarrow \infty$ or

$$h_{eff} \rightarrow \frac{k}{\frac{1}{2}\Delta x}$$

The modified method is more accurate because it resolves the temperature profile in the interfacial cell by a linear function, instead of the uniform temperature used in the basic method. The main assumptions in the modified method are:

- the interface has no heat capacity, *i.e.* the heat fluxes on both sides of the interface are equal;
- the temperature profile between the interface and the centre of the cell is linear as a consequence of the first order approximation of the heat conduction terms.

The analysis of this Section suggests that the gain in accuracy of the modified heat transfer method over the basic one is larger for larger values of $\frac{1}{2}\Delta x h/k$.

As a simple test for the modification, a problem of transient one-dimensional heat flux into a semi-infinite sand mould has been chosen. The boundary of the sand is at $x = 0$ and the boundary condition is

$$h \cdot [T_0 - T(0,t)] = -k \frac{\partial T}{\partial x}_{|x=0} \tag{4.15}$$

where h and k are constant parameters and T_0 is a constant ambient temperature. Initially the sand is at a uniform temperature T_i. The analytical solution, $T(x,t)$ is given by [22]

$$\frac{T(x,t) - T_i}{T_0 - T_i} = 1 - erf(X) - \left[exp\left(\frac{hx}{k} + \frac{h^2\alpha t}{k^2}\right)\right] \cdot \left[1 - erf\left(X + \frac{h\sqrt{\alpha t}}{k}\right)\right] \tag{4.16}$$

with $X = \frac{x}{2\sqrt{\alpha t}}$ and $\alpha = \frac{k}{\rho C}$.

For the numerical simulation the length of the sand is 200 *mm*, which is a good approximation of a semi-infinity during the first hour of simulation for the mould properties listed in Table 1.

First, a coarse uniform mesh is used consisting of $N = 10$ cells with $\Delta x = 20$ *mm*, as shown in Fig. 4.2*a*. The mesh is chosen in such a way that the left face of the first cell in sand coincides with the metal/mould interface. The temperature at the centre of this cell, that is at $x = 10$ *mm*, is recorded during the first hour of simulation and the results of the first order and modified methods are compared to the analytical solution in Fig. 4.3*a*. Fig. 4.3*b* shows the predicted interfacial heat fluxes, Q_1 and Q_2, normalised by the analytical solution and Fig. 4.3*c* shows the total predicted energy fluxed through the interface, that is $\int_0^t Q_1\,dt$ and $\int_0^t Q_2\,dt$, also normalised by the analytical solution.

It is clear from Fig. 4.3 that the modified heat transfer algorithm describes more accurately both temperature and energy evolution in the mould. The total energy flux, predicted by the modified method, is within 20% of the analytical solution soon after 500 *s*, while the basic procedure gives an error of more than 30% even at t=3600 *s*.

The second computer simulation is carried out with a non-uniform mesh with N=10 cells and the interfacial cell size $\Delta x = 2$ *mm*, the rest of the cell sizes gradually increasing towards the right domain boundary as shown in Fig. 4.2*b*. Temperatures at two positions are recorded: at $x = 10$ *mm*, as before, and at $x = 1$ *mm* which is at the centre of the interfacial cell.

The same time step size, $\Delta t = 5\ s$, is used throughout the simulations, which is sufficiently small to ensure numerical stability for the two meshes.

Fig. 4.4*a* shows the comparisons of the temperatures histories given by the two algorithms and the analytical solution and Figs. 4.4*b*, *c* show the comparisons of the interfacial heat flux and total fluxed energy respectively.

The use of the non-uniform mesh improves the accuracy dramatically. As expected, the modified method shows less mesh dependence. In a general case, this would make the heat extraction along the metal/mould interface more uniform.

In both Figs. 4.3*c* and 4.4*c* lines representing the total fluxed energy predicted by each of the two methods lie on opposite sides of the exact solution: the first order algorithm overestimates the flux and the modified one underestimates it.

If the assumption of Eq. (4.3) is dropped then the same treatment is applied to cell 1 in Fig. 4.1*b* to introduce the metal surface temperature $T_{1'}$. The result is (see also Eqs. (2.61), (2.62))

$$Q_2 = h_{eff} \cdot (T_1 - T_2) \tag{4.17}$$

where

$$h_{eff} = \frac{\frac{1}{4} h \frac{k_1}{\Delta x_1} \frac{k_2}{\Delta x_2}}{\frac{1}{2} h \frac{k_1}{\Delta x_1} + \frac{1}{4} \frac{k_1}{\Delta x_1} \frac{k_2}{\Delta x_2} + \frac{1}{2} h \frac{k_2}{\Delta x_2}} \tag{4.18}$$

and indexes 1 and 2 refer to cells 1 and 2 respectively and $1/h_{eff}$ is the effective thermal resistance between locations of temperatures T_1 and T_2, consisting of the interface resistance $1/h$ and resistances of materials on both sides of the interface, $\Delta x_1/2k_1$ and $\Delta x_2/2k_2$.

Consider now a situation where the metal mould interface is inside a cell, *i.e.* the interfacial cell is partially blocked. Let $V_{F,1}$ be the fractional open volume in it and T_0 and T_1 the cell-centred metal and mould temperatures respectively. Suppose that the interface is normal to the x-axis and cell 2 is the next fully blocked cell neighbour along the axis, with temperature T_2 (Fig. 4.1*c*).

Assuming first that T_0 represents the metal surface temperature, the energy balance between the two cells is

$$(1 - V_{F,1})\,\rho C \frac{dT_1}{dt} = h\,(T_0 - T_1) + \frac{k}{\Delta x} \cdot (T_2 - T_1) \tag{4.19}$$

$$\rho C \frac{dT_2}{dt} = -\frac{k}{\Delta x} \cdot (T_2 - T_1) \tag{4.20}$$

In the basic heat transfer algorithm

$$\Delta x = \frac{\Delta x_1 + \Delta x_2}{2}$$

is the distance between centres of cells 1 and 2 irrespective of the value of $V_{F,1}$. Therefore, only the thermal resistance between these two points, $r = \Delta x/k$, is taken into account in Eqs. (4.19), (4.20). The resistance between the interface and the centre of cell 2, estimated by assuming linear temperature distribution between the nodes, is

$$r_a = \frac{(1 - V_{F,1})\,\Delta x_1 + \frac{1}{2}\Delta x_2}{k} \tag{4.21}$$

Compared to r_a, the basic method overestimates the resistance if $V_{F,1} > 0.5$, and underestimates if $V_{F,1} < 0.5$.

In the modified method a surface mould temperature, $T_{1'}$, is introduced at the interface in a similar way as in the case of $V_F = 0$. Then Eqs. (4.19), (4.20) are transformed into

$$\begin{aligned} 0 &= \frac{k}{\Delta x_{1'}} \cdot (T_1 - T_{1'}) - h\,(T_{1'} - T_0) && (4.22) \\ (1 - V_{F,1})\rho C \frac{dT_1}{dt} &= -\frac{k}{\Delta x_{1'}} \cdot (T_1 - T_{1'}) + \frac{k}{\Delta x_{2'}} \cdot (T_2 - T_1) && (4.23) \\ \rho C \frac{dT_2}{dt} &= -\frac{k}{\Delta x_{2'}} \cdot (T_2 - T_1) && (4.24) \end{aligned}$$

Eq. (4.22) constitutes the heat flux balance at the interface and

$$\Delta x_{1'} = \frac{(1 - V_{F,1})\,\Delta x_1}{2}$$

$$\Delta x_{2'} = \frac{(1 - V_{F,1})\,\Delta x_1 + \Delta x_2}{2}$$

in other words, T_1 in Eqs. (4.22)-(4.24) is effectively located not at the cell 1 centre but at the centre $x_{1'}$ of the blocked volume, $(1 - V_{F,1})V_1$, where V_1 is the total volume of cell 1.

It can be seen from Eqs. (4.22)-(4.24) that the total thermal resistance between the interface and cell 2 centre is equal to r_a.

Substituting $T_{1'}$ from Eq. (4.22) into Eq. (4.23), gives

$$(1 - V_{F,1})\rho C \frac{dT_1}{dt} = h_{eff} \cdot (T_0 - T_1) + \frac{k}{\Delta x_{2'}} \cdot (T_2 - T_1) \tag{4.25}$$

with

$$h_{eff} = \frac{h \frac{k}{\Delta x_{1'}}}{h + \frac{k}{\Delta x_{1'}}}$$

Applying a similar procedure on the metal side of the interface, the heat flux between $x_{1'}$ and $x_{1''}$, the centre of the open volume $V_{F,1} \cdot V_1$ of cell 1 with temperature T_0, is expressed as

$$Q = h_{eff}(T_0 - T_1) \tag{4.26}$$

where

$$h_{eff} = \frac{h \frac{k_{me}}{\Delta x_{1''}} \frac{k_{mo}}{\Delta x_{1'}}}{h \frac{k_{me}}{\Delta x_{1''}} + \frac{k_{me}}{\Delta x_{1''}} \frac{k_{mo}}{\Delta x_{1'}} + h \frac{k_{mo}}{\Delta x_{1'}}} \tag{4.27}$$

k_{me}, k_{mo} being the metal and mould thermal conductivities respectively, and

$$\Delta x_{1''} = \frac{V_{F,1} \, \Delta x_1}{2} = 0.5 - \Delta x_{1'}$$

A simple one-dimensional metal/mould heat transfer problem is used to demonstrate the differences between the basic and the modified methods results. The metal is kept at a constant uniform temperature and the interfacial heat flux is compared with the analytical solution given by Eq. (4.16). The uniform mesh with $\Delta x = 10$ *mm* is fixed, while the position of the interface is varied within one cell, covering the range of V_F values between zero and unity. The mould properties are given in Table 1.

Numerical solutions for the interfacial heat flux and total energy, fluxed through the interface, at $t = 30$ *s* from the beginning of the simulation are shown in Figs. 4.5*a* and 4.5*b*, respectively. Each numerical result is normalised by the corresponding analytical solution. Compared to the first order method, the modified method results are more accurate and less dependent on the value of V_F in the interfacial cell, though it is obvious that the used mesh resolution is not sufficient to give a satisfactory accuracy for all values of V_F. It is interesting that their appear to be an optimum value of V_F for each method, which may be dependent on the material and interface properties.

In a general case, when the interface in a cell is arbitrarily orientated, $x_{1'}$ and $x_{1''}$ are associated with the geometrical centres of the blocked and open volumes of the interfacial cell. $\Delta x_{1'}$ and $\Delta x_{1''}$ are approximated by

$$\Delta x_{1'} = (1 - V_{F,1}) \, \Delta x_e \tag{4.28}$$

$$\Delta x_{1''} = V_{F,1} \, \Delta x_e \tag{4.29}$$

where

$$\Delta x_e = min(\frac{V_1}{A_1}, \Delta x_n) \tag{4.30}$$

Δx_n is the size of the cell along the coordinate axis in the direction nearest to the direction of the interface normal and A_1 is the area of the interface in the cell.

4.2 Solidification Model

4.2.1 Alloy Solidification

The integral on the right-hand side of the expression for the enthalpy given by Eq. (3.4) can be written as

$$\int_0^T CdT = \int_0^{T_s} CdT + \int_{T_s}^{T_l} CdT + \int_{T_l}^{T} CdT \tag{4.31}$$

If constant values of C_s and C_l are assumed then the first and third integrals on the right-hand side of Eq. (4.31) are easily evaluated. The second integral contains specific heat of the solid/liquid mixture and is variable across the freezing range if $C_s \neq C_l$

$$C = f_s C_s + (1 - f_s) C_l \tag{4.32}$$

and

$$\int_{T_s}^{T_l} CdT = - \int_0^1 [f_s C_s + (1 - f_s) C_l] \frac{dT}{df_s} df_s \tag{4.33}$$

A simplification is made in the calculation of the integral by assuming that

$$\frac{dT}{df_s} = T_s - T_l = const$$

Thus

$$\int_0^1 [f_s C_s + (1 - f_s) C_l] \frac{dT}{df_s} df_s \approx \frac{1}{2}(C_s + C_l)(T_l - T_s) \tag{4.34}$$

And for $T_s < T < T_l$

$$\begin{aligned}\int_{f_s}^1 [f_s C_s \ &+ \ (1 - f_s) C_l] \frac{dT}{df_s} df_s \approx \frac{1}{2} [(1 + f_s) C_s \\ &+ \ (1 - f_s) C_l] (T - T_s)\end{aligned} \tag{4.35}$$

where f_s corresponds to temperature T.

The effect of the contribution to the enthalpy given by Eq. (4.34) or (4.35) is that during solidification both latent heat *and* specific heat have to be removed as the temperature decreases between T_l and T_s, unlike solidification at a constant temperature. This generally leads to larger solidification times for alloys than for pure materials, given the same latent heat. The effect is more apparent for larger specific heat and freezing range.

The simplest way to model alloy solidification is to assuming linear variation of solid fraction with temperature in the freezing range

$$f_s = \frac{T_l - T}{T_l - T_s}, \qquad T_s < T < T_l \tag{4.36}$$

This model can be enhanced by adding optional lever rule and Scheil Eqs. (2.64), (2.65) (Section 2.8). It can be seen from Eq. (4.35) that, within the freezing range, enthalpy becomes a nonlinear function of the temperature, even for the linear latent heat release,

$$\begin{aligned} H(T) &= C_s T_s + \frac{1}{2}\left[(1 + f_s(T))C_s + (1 - f_s(T))C_l\right](T - T_s) \\ &+ (1 - f_s(T))\,L \end{aligned} \tag{4.37}$$

where $f_s(T)$ is defined by one of the solidification models. For the Scheil's model Eq. (4.37) transforms into

$$\begin{aligned} H(T) &= C_s T_e + \frac{1}{2}\left[(f_{s,e} + f_s(T))C_s + (2 - f_{s,e} - f_s(T))C_l\right](T - T_e) \\ &+ (1 - f_s(T))\,L \end{aligned} \tag{4.38}$$

where $f_{s,e} = f_s(T_e)$.

At the end of every time cycle Eq. (4.37) or (4.38) must be solved in each cell to find the solid fraction and temperature from a known value of the enthalpy. In the original model (Eq. (3.9)), $H(T)$ is a linear function and its resolution for temperature is trivial.

Fig. 4.6 shows the variation of the fraction of solid and enthalpy for an Al-4.5%Cu alloy (Table 3) for each of the solidification models, given by Eqs. (2.64), (2.65), (4.36), (4.37) and (4.38).

In the modified solidification algorithm Eq. (4.37) is solved by the Newton iteration method in which at iteration $n + 1$

$$T_{n+1} = T_n - \frac{f(T_n)}{df(T_n)/dT}$$

where $f(T_n)$ is the difference between the left- and right-hand sides of Eq. (4.37). The starting temperature is taken from the previous time step. Iterations stop when

$$| T_{n+1} - T_n |< \epsilon, \qquad \epsilon = \frac{T_l - T_s}{10000}$$

It usually takes one to five iterations to satisfy the convergence criteria since $H(T)$ is a smooth function and $dH/dT > 0$ for all temperatures. Generally, the number of iterations depends on the cooling rate and the time step size.

When $T = T_e$ (for the Scheil model), Eq. (4.38) becomes a linear function of f_s

$$H(f_s) = C_s T_e + (1 - f_s) L$$

and no iteration is necessary to obtain the solution.

4.2.2 Fraction of Solid

The solid fraction function, f_s, is not explicitly calculated in the original code nor is f_s used as an output variable for graphical and numerical analysis at the end of the simulation. However, spatial and temporal evolution of the fraction of solid function could provide important information about the progress of solidification.

A modification of the code has been made to include solid fraction as an additional variable. It is either a function of temperature for solidification models defined by Eqs. (2.64), (2.65) or (4.36) or is an independent variable for pure metal and eutectic solidification processes. Solid fraction, stored in an additional array, serves as an input for other parts of the program, such as drag function calculation (Section 4.3) and shrinkage simulation (Chapter 5).

A modification has also been made to include solid fraction as an output quantity. The modification is based on the existing routine for storing and displaying numerical data, so that all display options for other variables are applicable to the solid fraction. Those include:

- history plots for the evolution of the solid fraction with time at specified locations in the casting.
- 2-D spatial colour and contour line plots. This can be combined with velocity vectors and marker particle plots.
- 3-D contour surface plots highlighting the spatial distribution of a particular value of f_s.

- both 2-D and 3-D spatial plots can be stored with predefined time increments for subsequent animation.

Additional dashed single contour line plot of $f_s = 0.5$ is provided to visualise the approximate position of the solidification front, especially useful for narrow freezing range alloys. This is done similarly to the existing FLOW-3D method of visualising the position of the free surface.

Fig. 4.7 shows an example of a 2-D colour temperature plot with flow velocities and solidification front. More examples of the use of the solid fraction can be seen in other sections.

4.3 Solidification Drag Model

When metal solidifies, its position in space no longer changes, *i.e.* the velocity of the solid phase is zero (or equal to a constant non-zero value in the case of continuous casting process, for example). One of the ways to account numerically for the corresponding change of fluid momentum is to increase the solid/liquid mixture viscosity coefficient μ_{eff} in a solidifying cell so that for a completely solid cell $\mu_{eff} \rightarrow \infty$ [124]

$$\mu_{eff} = \mu \left(1 - \frac{f_s}{f_{s,p}}\right)^{-2.5\, f_{s,p}} \tag{4.39}$$

where $f_{s,p}$ is the solid fraction at close packing of the solid (equals to 0.62 for solid spherical particles), similar to the critical solid fraction $f_{s,cr}$ at which any flow seizes according to the shrinkage models described in Section 2.9.

As a result of the "effective viscosity" model, given by Eq. (4.39), the solid material will adhere to the mould, including moving walls. The latter occurs during continuous, twin-roll and melt-spinning casting processes.

Viscous terms in the discretised momentum equations will have to be included *implicitly* since the explicit formulation will cause severe time step size limitations as μ_{eff} increases. The implicitness will make momentum Eqs. (3.7)-(3.9) coupled with each other, requiring an efficient solution for the first guess velocities.

In the present model the cell momentum change due to solidification is accounted for by using the drag force concept. The solid phase material is assumed to be at rest or moving at a uniform speed.

Here we will only consider the drag due to solidification. The main role of the drag force is to reduce the fluid velocity to zero when it solidifies. Assuming that solid material is at rest with respect to the computational mesh, the solidification/melting process is approximated by using a drag coefficient, $K(T)$, that is a function of temperature or fraction of solid. The drag should be effectively infinite when material is in the solid phase. At intermediate states, consisting of a mush, the drag should also assume an intermediate value. The behaviour of the drag coefficient for $0 < f_s < 1$ is very much the choice of the user since it is difficult to validate any particular drag model. The only restriction is that K must be non-negative.

For a Darcy-type solidification drag, the flow in the mushy zone is thought of as a flow of the liquid fraction through a fixed matrix structure created by the dendrites. Coefficient K is then dependent on the volumetric solid fraction [125] (Eq. (2.70))

$$K \sim \frac{f_{sv}^2}{(1 - f_{sv})^3} \tag{4.40}$$

Variations of Eq. (4.40) include introducing a critical solid fraction, so that the drag is equal to zero for $f_s < f_{s,cr}$, before the solid phase forms a rigid structure. Such regions are often called "slurry zones", where solid material floats freely in the metal [2].

Huang *et al* suggested two expressions for the drag coefficient: one for a columnar dendritic region, K_c, and the other, K_e, for an equiaxed [117]. The drag is dependent on the specific surface area available for flow in the dendritic mushy region

$$K_c = \frac{8\pi\mu}{\rho_l u_s \beta d_p sin\theta} \tag{4.41}$$

$$K_e = \frac{216\mu L}{\rho_l u_s \beta d_0^2 f_l} \tag{4.42}$$

where u_s is the solidus velocity, $\beta = 1 - \rho_l/\rho_s$ volumetric shrinkage coefficient, d_p and d_0 the primary dendritic spacing and the equiaxed grain diameter, respectively, θ the equivalent vertex angle of the primary dendrites and L denotes the length of the mushy zone.

Another equation for the drag force coefficient is presented in this Section. The derivation is based on the simplifying assumption that there is *no* frictional interaction between solid and liquid phases. This assumption is more appropriate in the situations when the liquid/solid interface can be represented with a surface rather than a mushy zone.

Consider a cell full of fluid (Fig. 4.8*a*). At time t, m_s and m_l are masses of the solid and liquid phases in the cell respectively

$$m = m_s + m_l \tag{4.43}$$

and

$$f_s = \frac{m_s}{m}$$

Since the solidified fluid velocity is zero, the total momentum of the cell is equal to the liquid phase momentum

$$M = m_l u_l = (1 - f_s) m \, u_l \tag{4.44}$$

Suppose that there are no body and pressure forces acting on the cell and that the liquid phase can flow freely, *i.e.* neglect viscous friction between the two phases. In that case during solidification, assuming also that volumetric shrinkage effects are negligible,

$$u_l = const, \qquad m = const \tag{4.45}$$

The cell momentum change over the time dt due to the loss of mass by the liquid phase to the solid phase, using Eqs. (4.44), (4.45), is

$$dM = u_l \, dm_l = -m \, u_l \, df_s \tag{4.46}$$

For the solid/liquid mixture velocity in the cell, which appears in the discretised equations,

$$u_m = \frac{M}{m} = (1 - f_s) \, u_l \tag{4.47}$$

and it follows from Eq. (4.45) that

$$dM = m \, du_m \tag{4.48}$$

Now equalising the right-hand sides of Eqs. (4.46) and (4.48), divided by dt, and taking into account Eq. (4.47) yields

$$\frac{du_m}{dt} = -\frac{u_m}{1 - f_s} \frac{df_s}{dt} \tag{4.49}$$

the solution of which is

$$u_m = u_0 (1 - f_s) \tag{4.50}$$

where u_0 is the liquid velocity for $f_s = 0$. It is easily seen that $u_m = 0$ when $f_s = 1$.

Comparing the right-hand side of Eq. (4.49) with the definition of the drag force, Eq. (3.10), gives an expression for the drag force coefficient

$$K = \frac{1}{1-f_s}\frac{df_s}{dt} \tag{4.51}$$

In a more general case, when m and u_l are not constant because of the net inflow or outflow of fluid in the cell and the action of other forces, the drag coefficient becomes

$$K = \frac{1}{m_l}\frac{dm_s}{dt} \tag{4.52}$$

The right-hand side of Eq. (4.52) needs to be defined for $f_s = 1$ as both the numerator and denominator vanish at that point. By putting

$$K = \infty \quad \text{for } f_s = 1$$

the condition of an infinite drag in the solid phase is satisfied.

The drag is zero in a partially solidified cell if the solidification front in it does not move. If part of the cell melts, *i.e.* $dm_s < 0$ then $K = 0$ as the cell momentum does not change because the newly created liquid has zero velocity immediately after melting. Thus, condition $K \geq 0$ is also satisfied.

A 1-D simulation of pure aluminium solidification has been carried out to test the drag model. The geometrical setup is shown in Fig. 4.8*b*. The metal is flowing at a constant velocity u_0 over a horizontal mould plate in the absence of pressure, body and viscous forces. The results are shown in Fig. 4.9. As the metal in a cell starts to solidify, the drag force is applied and the cell horizontal velocity changes in accordance with Eq. (4.50), as shown in Figs. 4.9*b* – *d* for three consecutively solidifying cells.

The drag force given by Eq. (4.52) does not represent any physical force acting on the fluid. It is just a consequence of using a solid/liquid mixture velocity, defined by Eq. (4.47), rather than modelling the flow of each phase separately.

Eq. (4.52) is applied here to both mushy and plane front solidification. However, frictional drag in the mushy zone may dominate the flow and the developed drag model will not be sufficiently accurate.

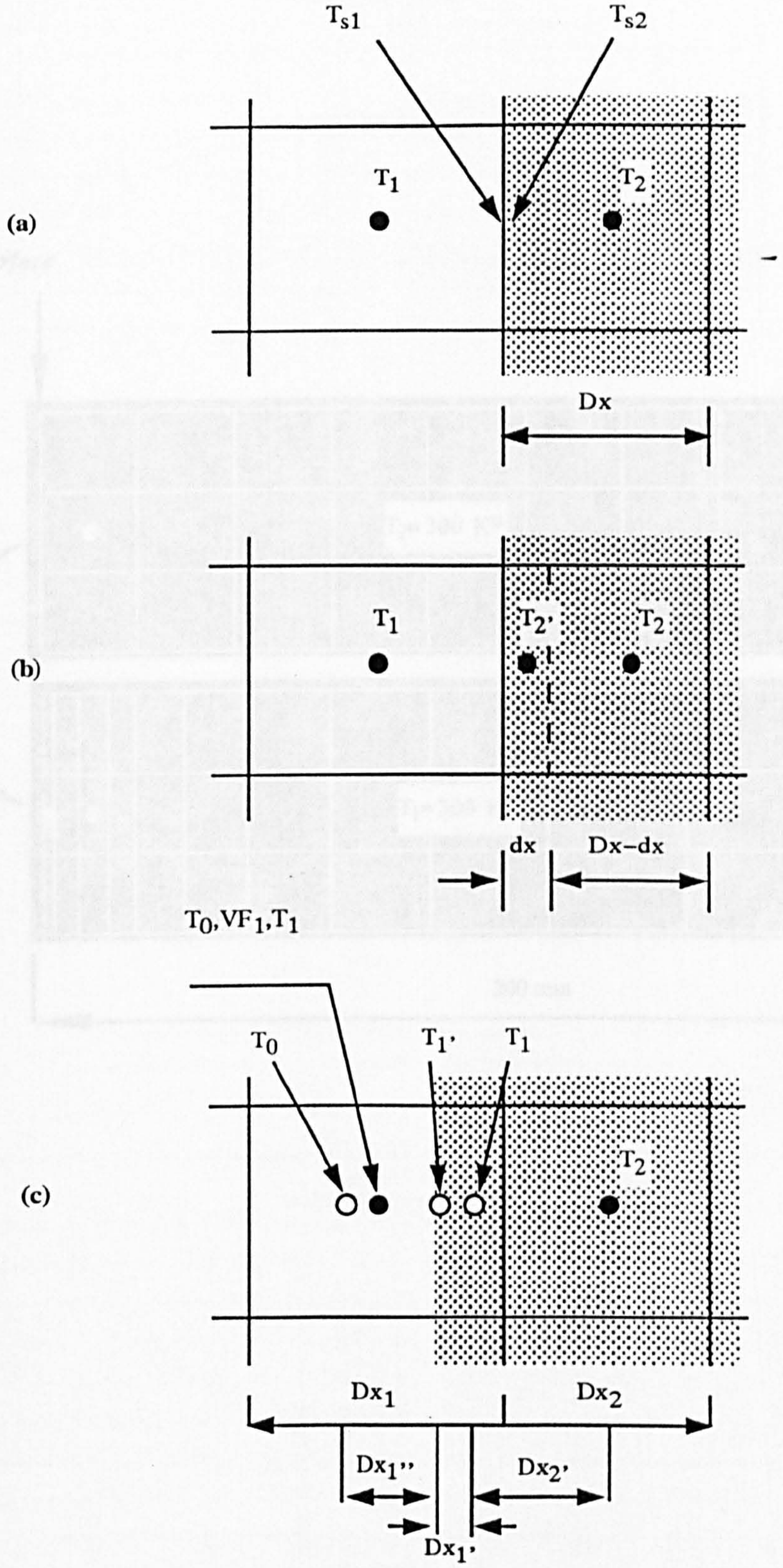


Figure 4.1. (a) Cell-cented emperatures T_1 and T_2 are used in the basic heat transfer algorithm instead of T_{s1} and T_{s2}. (b) Temperature $T_{2'}$ approaches T_{s2} as δx reduces zero. (c) In the interfacial cell ● is the node used in the basic method and ○ are the nodes used in the modified algorithm.

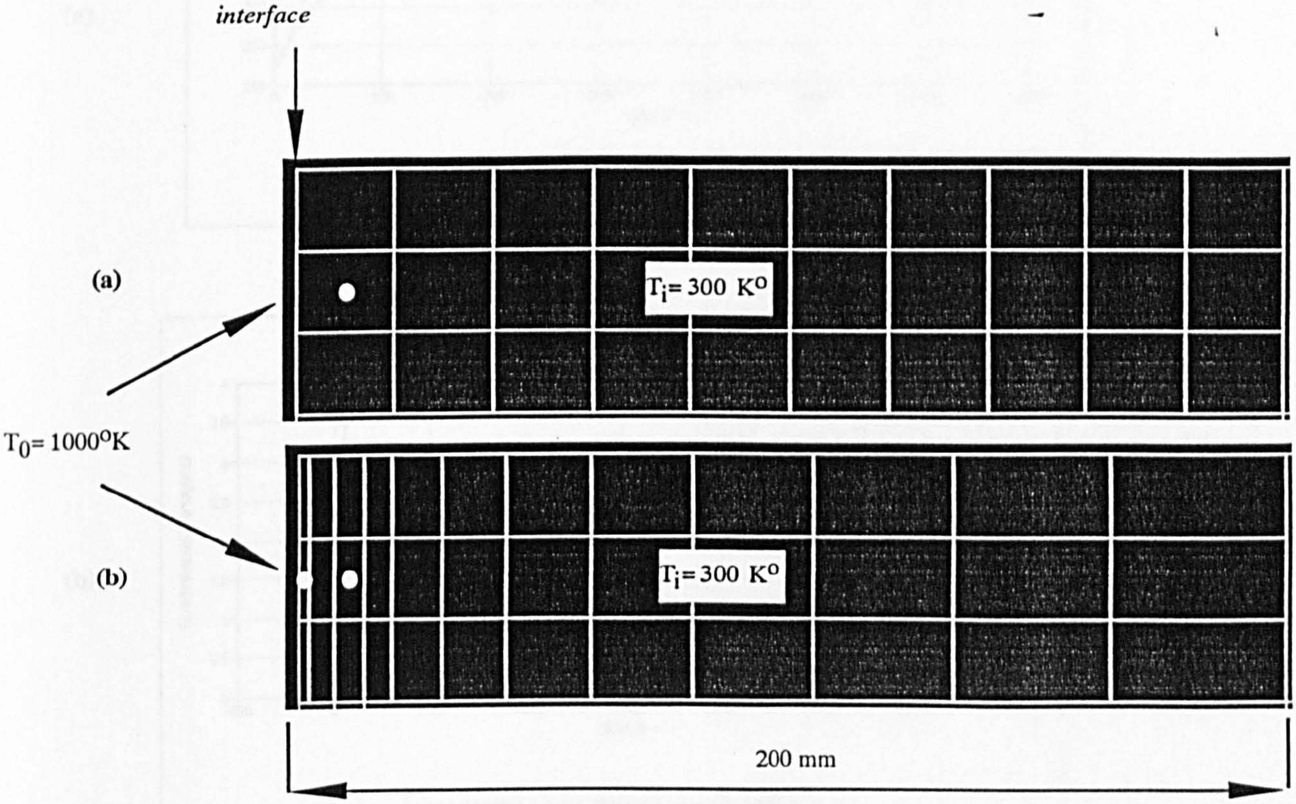


Figure 4.2. Uniform mesh (a), with Δx=*20 mm*, and non-uniform mesh (b), with the smallest cell size *2 mm*, used to simulate the one-dimensional heat flow used as a test for the modified heat transfer algorithm. White dots show where the temperatures werecompared.

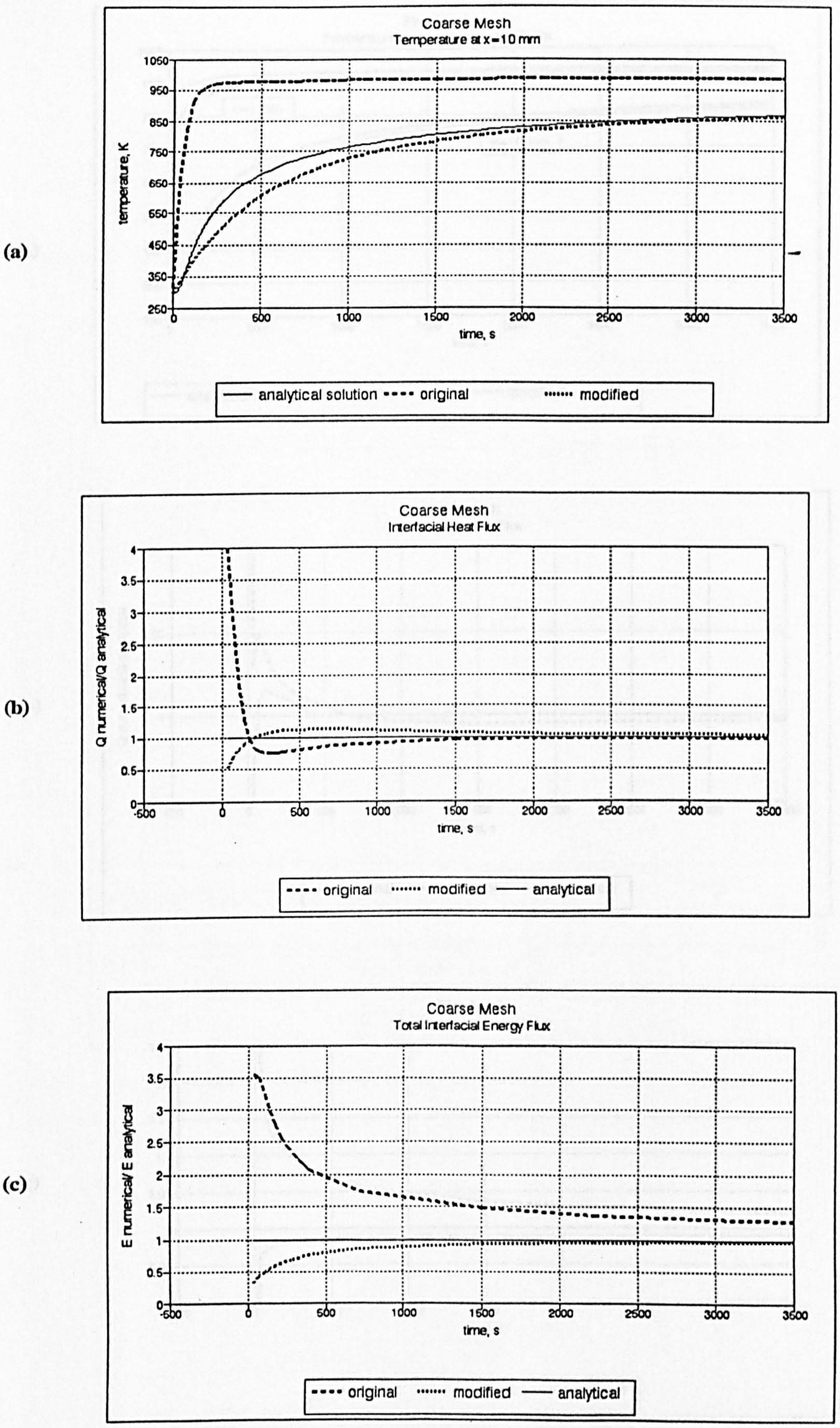


Figure 4.3. Comparisons of the original and modified heat transfer algorithm results for coarse mesh shown in figure 4.2,(a): (a) temperature at x=10 mm from the interface, (b) interfacial heat flux and (c) total energy fluxed through the interface.

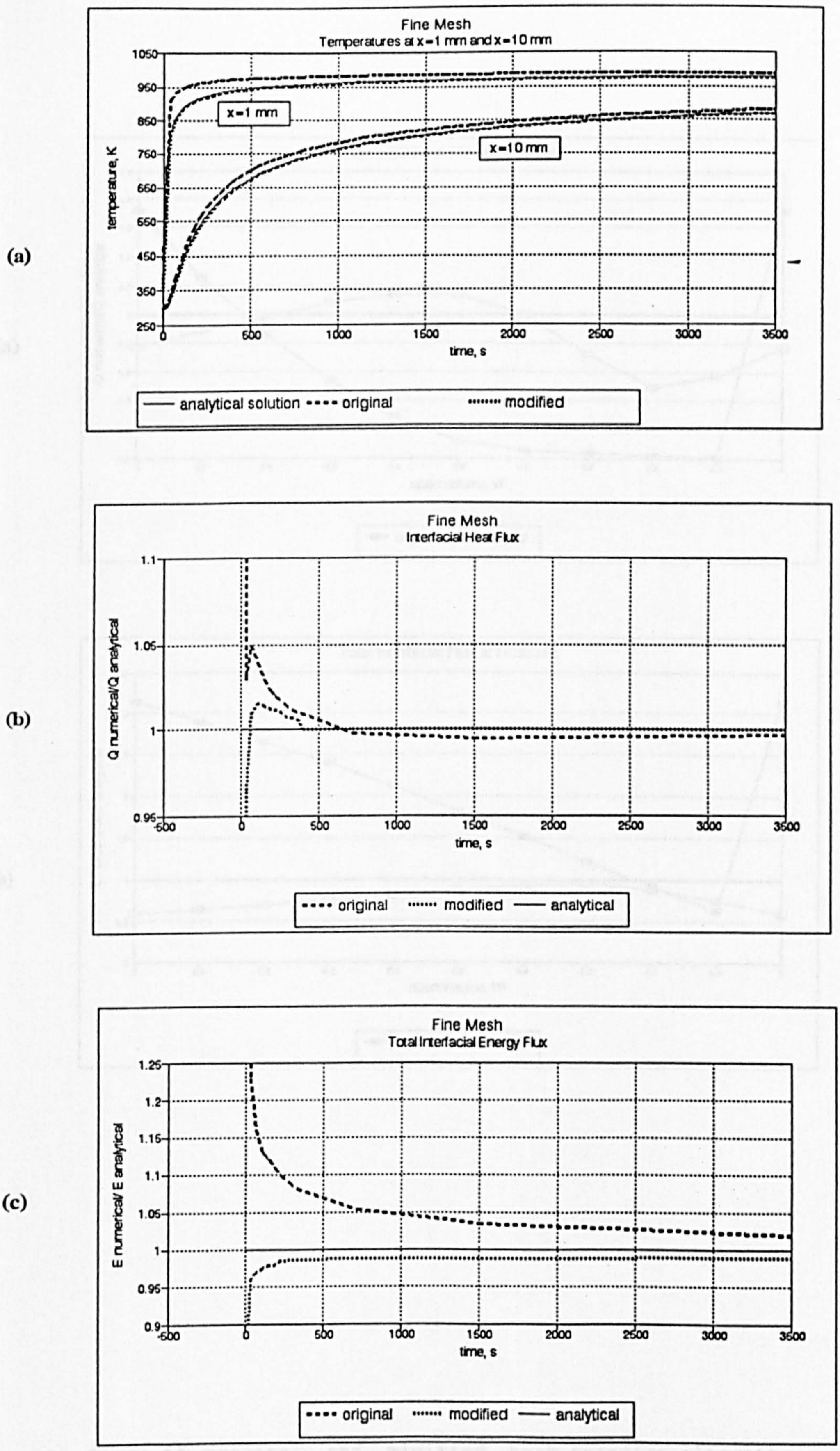


Figure 4.4. Same as in figure 4.3 but for fine mesh (figure 4.2,(b)). Tmperatures are compared at two locations: 1 mm and 10 mm.

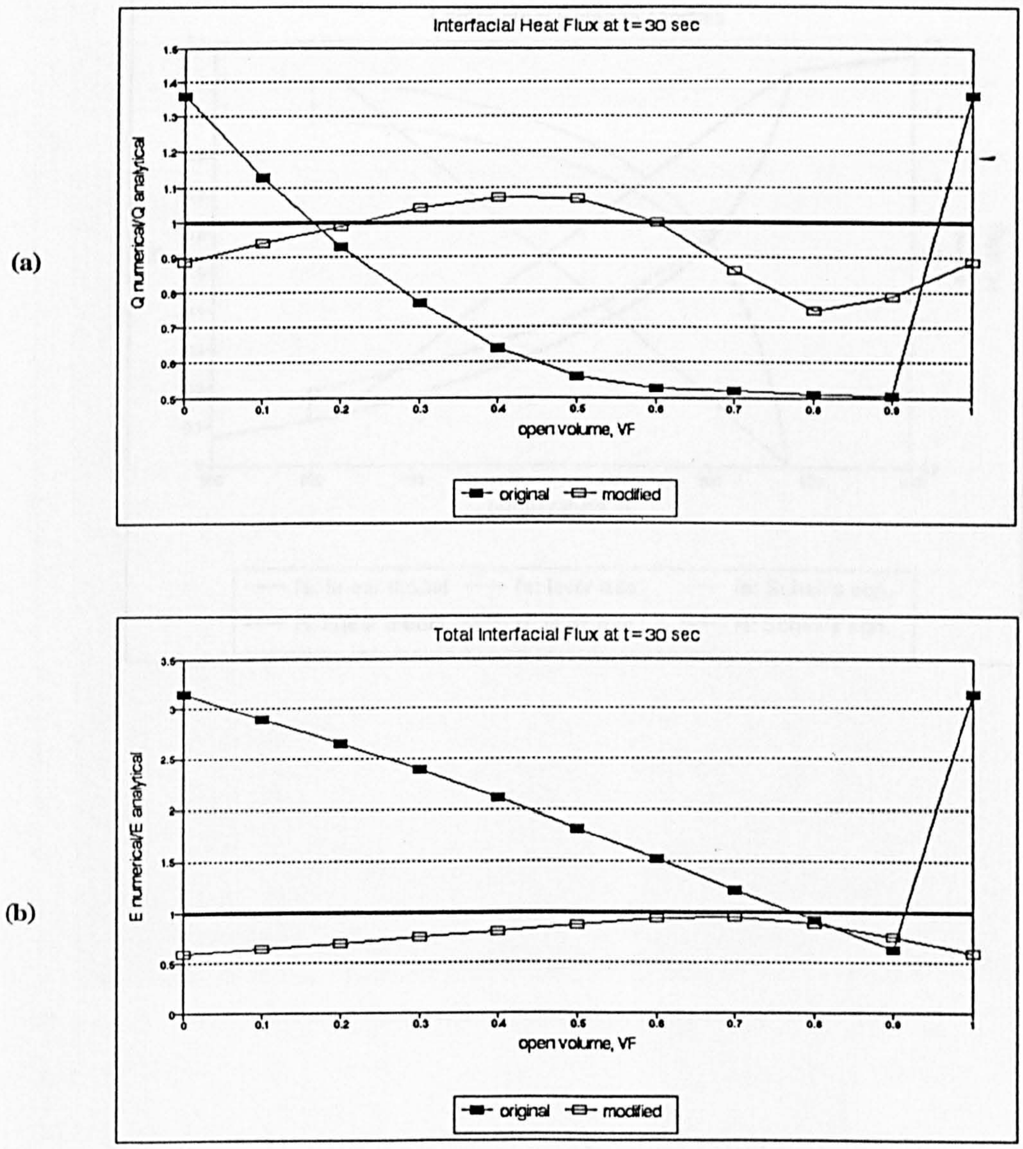


Figure 4.5. Original and midified heat transfer algorithm result dependences on the value of V_F in the interfacial cell for coarse mesh showm in figure 4.2,(a) at t=30 s: (a) interfacial heat flux, (b) total energy fluxed through the interface.

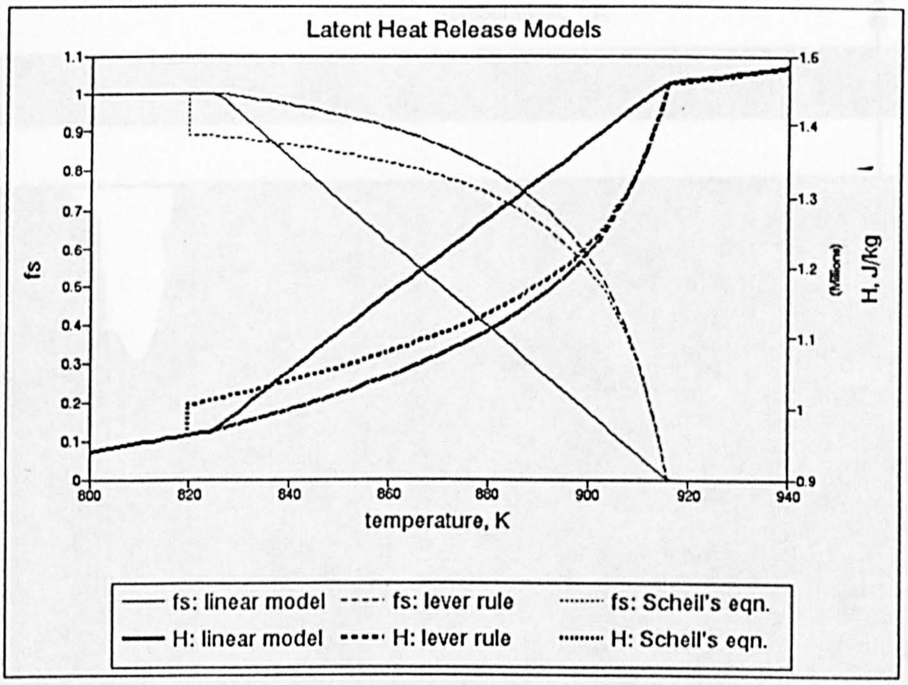


Figure 4.6. Fraction of solid (fs, thin lines) and enthalpy (H, heavy lines) as functions of temperature for linear (solid), lever rule (dashed) and Scheil's (dotted) latent heat release modes.

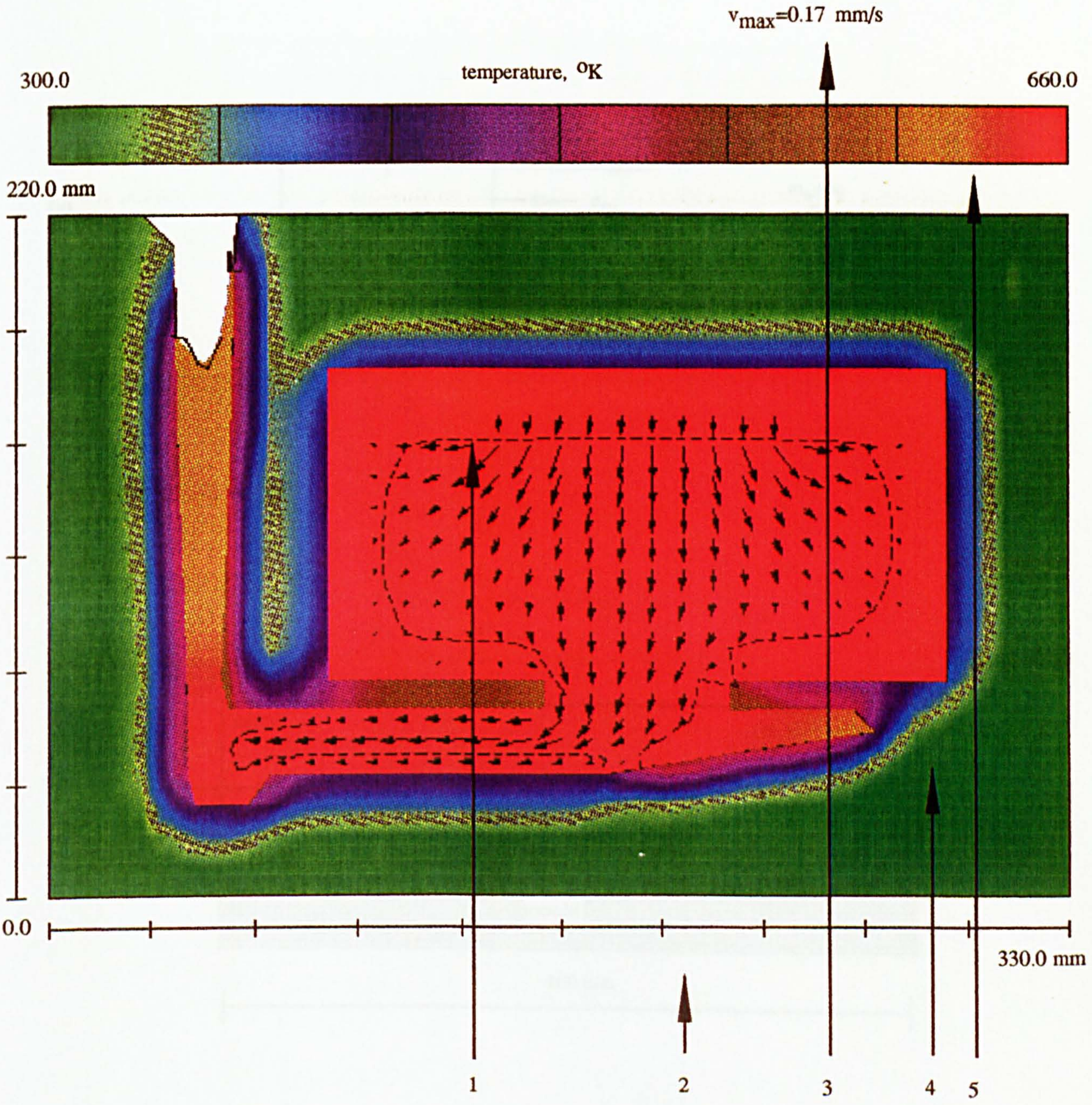


Figure 4.7. A typical 2-D graphical output from a casting solidification and shrinkage simulation: 1. casting temperature contours with the position of the solidification front (dashed line) and velocity vectors; 2. coordinate axis with casting dimensions in cm; 3. the maximum velocity on the plot; 4. mould temperature contours; 5. the temperature scale.

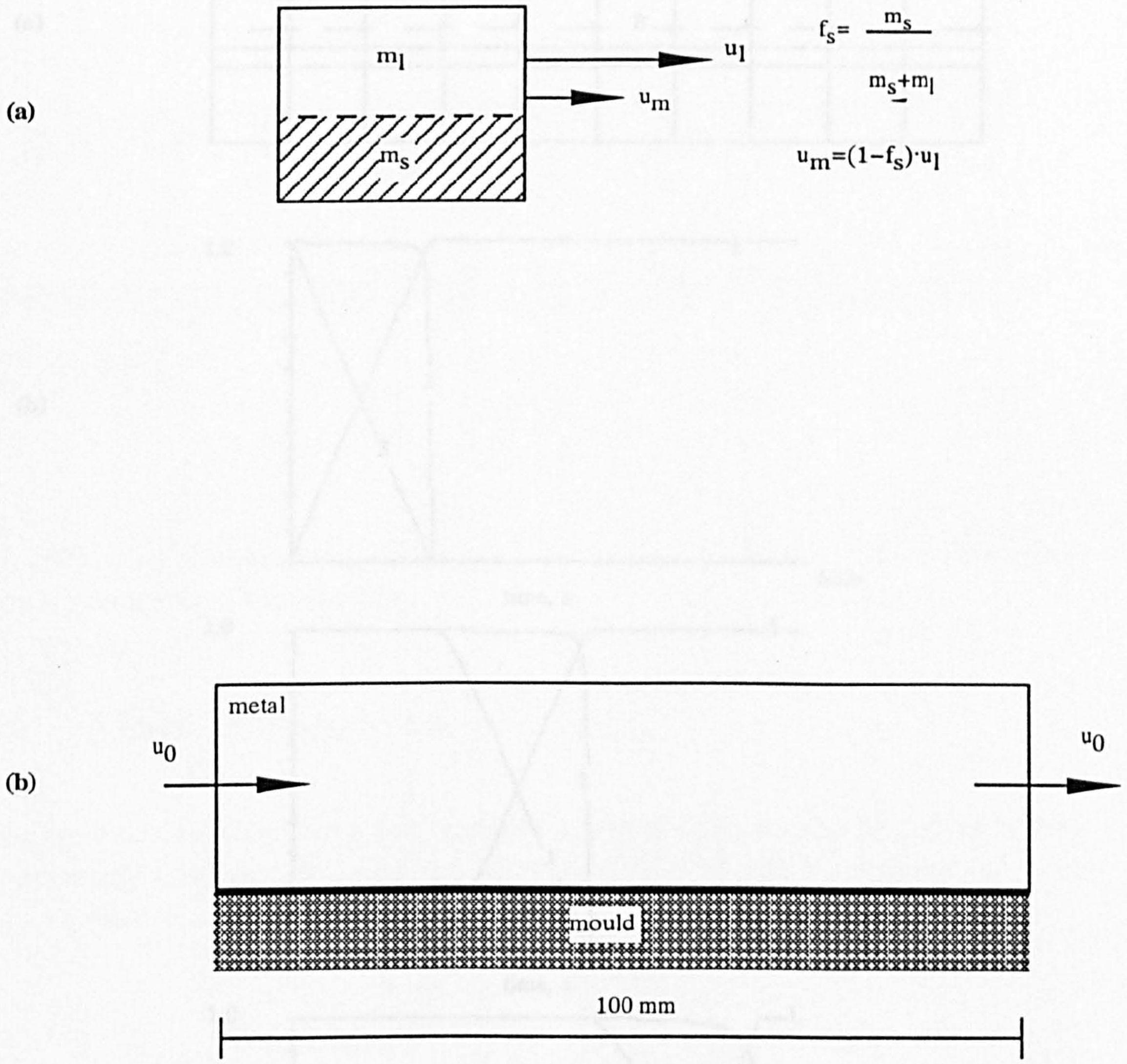


Figure 4.8. (a) Liquid velocity u_l and solid-liquid mixture velocity u_m in a partially solid cell. (b) One-dimensional test simulation for the new drag force model.

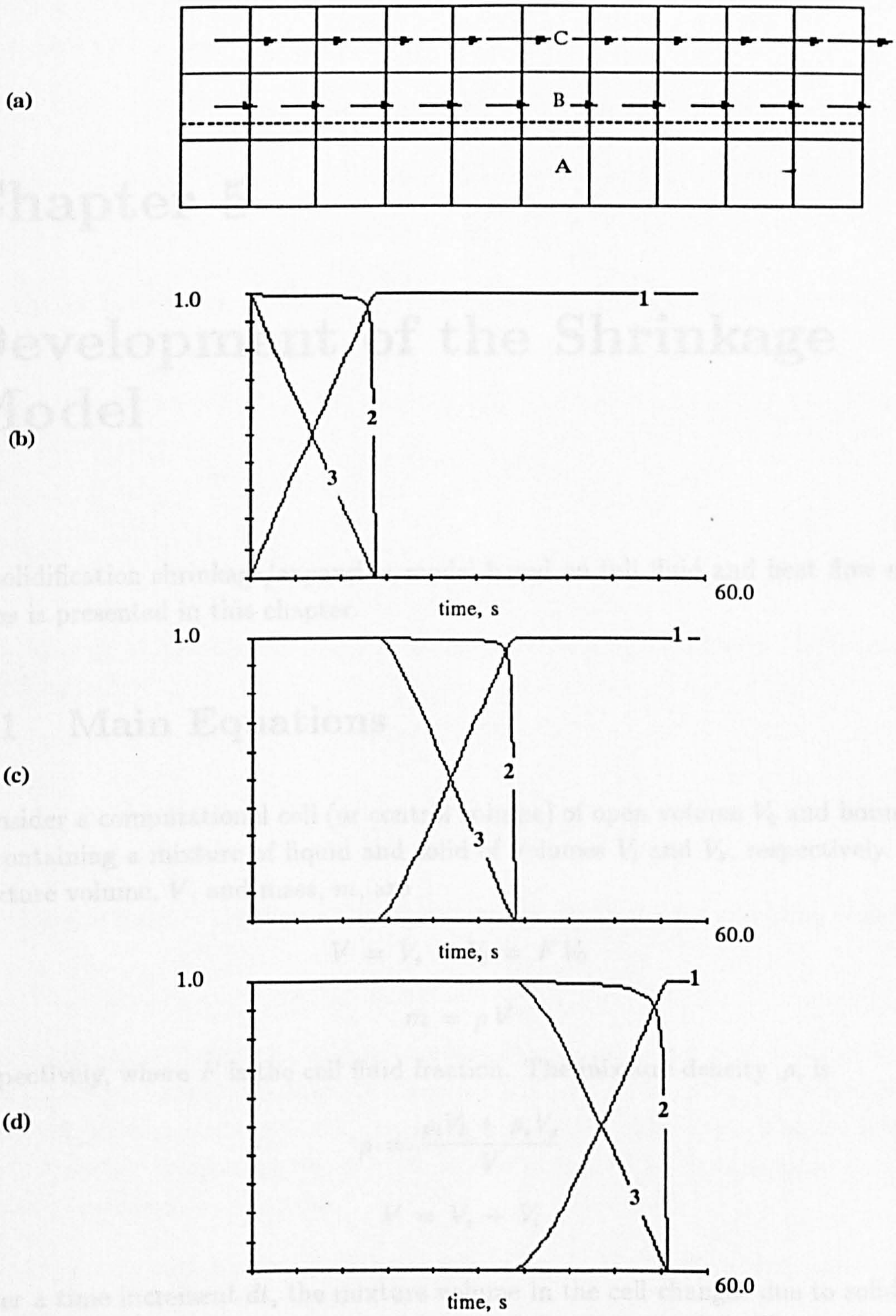


Figure 4.9. 1-D heat and fluid flow simulation result for the flow shown in figure 4.8,(b): (a) velocities and solidification front (dashed line) at t=23.0 s; (b)-(d) 1 - fraction of solid, 2 - $\frac{1}{1+K\Delta t}$, K is the drag coefficient, 3 - horizontal velocity for cells A, B, C, respectively.

Chapter 5

Development of the Shrinkage Model

A solidification shrinkage/expansion model based on full fluid and heat flow equations is presented in this chapter.

5.1 Main Equations

Consider a computational cell (or control volume) of open volume V_0 and boundary Σ containing a mixture of liquid and solid of volumes V_l and V_s, respectively. The mixture volume, V, and mass, m, are

$$V = V_s + V_l = F\,V_0 \tag{5.1}$$

$$m = \rho\,V \tag{5.2}$$

respectively, where F is the cell fluid fraction. The mixture density, ρ, is

$$\rho = \frac{\rho_l V_l + \rho_s V_s}{V} \tag{5.3}$$

$$V = V_s + V_l \tag{5.4}$$

Over a time increment dt, the mixture volume in the cell changes due to solidification, dV^*, and a flux of the liquid phase through cell faces, dV_l'. The total change in the mixture volume, dV, is

$$dV = dV_s + dV_l = dV^* + dV_l' \tag{5.5}$$

Volumetric changes due to phase transformation occur at constant cell fluid mass, m, so that (using Eqs. (5.1)-(5.3))

$$dV^* = \left(1 - \frac{\rho_s}{\rho_l}\right) dV_s = S\,V_0\,dt \tag{5.6}$$

where

$$S = \left(1 - \frac{\rho_s}{\rho_l}\right) \frac{dV_s}{dt}\frac{1}{V_0} \tag{5.7}$$

constitutes the change of volume per unit time per unit cell volume due to phase transformation.

The corresponding change of the mixture density, from Eqs. (5.1) and (5.3), is

$$d\rho = (\rho_s - \rho_l)\frac{V_l dV_s - V_s dV_l}{V^2} = d\rho^* + d\rho' \tag{5.8}$$

where

$$d\rho^* = \rho(\frac{\rho_s}{\rho_l} - 1)\frac{dV_s}{V} = -\frac{S\rho}{F}\,dt \tag{5.9}$$

is the change of density due to phase transformation only. Similarly, the fraction of fluid function is

$$dF = \frac{dV}{V_0} = dF^* + dF' \tag{5.10}$$

and

$$dF^* = \left(1 - \frac{\rho_s}{\rho_l}\right)\frac{dV_s}{V_0} = S\,dt \tag{5.11}$$

The integral form of the continuity equation, Eq. (2.1), for the cell open volume V_0 is

$$\rho_l \cdot \int_\Sigma v_n\,d\sigma = -\frac{\partial}{\partial t}\int_V \rho\,d\tau \tag{5.12}$$

The liquid phase density is used in the surface integral because it is assumed that only pure liquid phase can flow. At those parts of the cell boundary that are blocked by the solidified material, velocity is zero and no contribution is made to the integral. The volumetric integral on the right-hand side of Eq. (5.11) is calculated over fluid volume V, since $V_0 - V$ is void.

Replacing the density on the right-hand side of Eq. (5.11) by the average cell-centred value, given by Eq. (5.3), yields

$$\rho_l \cdot \int_\Sigma v_n\,d\sigma = -V_0 \cdot \frac{\partial}{\partial t}(F \cdot \rho) \tag{5.13}$$

Eq. (3.6) for the fluid fraction function must be rewritten to take into consideration Eqs. (5.9) and (5.10)

$$\frac{\partial F}{\partial t} + \frac{1}{V_0} \cdot \int_\Sigma v_n F d\sigma = S \tag{5.14}$$

We will neglect the variation of the mixture density in the momentum-equation, Eq. (2.22), so that only the drag force in it accounts for solidification.

The integral form of enthalpy Eq. (2.23) is

$$\frac{\partial}{\partial t} \int_V \rho H \, d\tau = -\rho_l \cdot \int_\Sigma H_l v_n \, d\sigma + q_\Sigma \tag{5.15}$$

where q_Σ denotes heat conduction and transfer fluxes through the cell boundary. H_l is the enthalpy of the liquid phase even if two phases are present along the boundary since only the liquid phase flows. After replacing ρ and H by the cell-centred average quantities, Eq. (5.14) can be rewritten as

$$V_0 \cdot \frac{\partial}{\partial t}(\rho F H) = -\rho_l \cdot \int_\Sigma H_l v_n \, d\sigma + q_\Sigma \tag{5.16}$$

Eqs. (5.6), (5.7), (5.12), (5.13), and (5.15), together with the momentum equations, represent the mathematical model of the solidification process with volumetric changes.

5.2 Numerical Technique

5.2.1 Source Term Calculation

dV_s is defined by solving the fluid and heat flow equations. Therefore the calculation of its source term and the solution of these equations are coupled. An implicit representation of the source S, defined by Eq. (5.6), would require a simultaneous solution of the fluid flow and enthalpy equations, making the numerical algorithm complicated and inefficient.

The procedure is greatly simplified if S is represented explicitly.

At the beginning of $n + 1$ time cycle the value of dV_s is estimated in each cell using values of fluid fraction and solid fraction functions known at times t_{n-1} and t_n

$$dV_s^n = V_0 \cdot (f_{sv}^n F^n - f_{sv}^{n-1} F^{n-1}) \tag{5.17}$$

where

$$f_{sv} = \frac{\rho_l \cdot f_s}{\rho_l f_s + \rho_s(1 - f_s)}$$

is the cell volumetric fraction of solid.

The source term then is

$$S^n = \left(1 - \frac{\rho_s}{\rho_l}\right) \cdot \frac{dV_s^n}{\Delta t^{n+1}} \frac{1}{V_0} \tag{5.18}$$

The use of Δt^{n+1}, rather than Δt^n, ensures that the subtracted volume at cycle $n+1$ exactly corresponds to the volumetric changes given by Eq. (5.16).

The corresponding change of the cell fluid fraction is given by Eq. (5.10). After all the advection contributions to the cell fluid content are calculated (as described in Section 2.5), giving an intermediate value $\tilde{F}_{i,j,k}^{n+1}$, the final adjustment is made to include the volumetric change

$$F_{i,j,k}^{n+1} = \tilde{F}_{i,j,k}^{n+1} + S_{i,j,k}^n \cdot \Delta t^{n+1} \tag{5.19}$$

The left-hand side of Eq. (5.15) can be represented in the following form

$$\frac{\partial}{\partial t}(\rho F H) = \rho \frac{\partial F H}{\partial t} + F H \frac{\partial \rho}{\partial t} \tag{5.20}$$

Combining Eqs. (5.15) and (5.19) yields

$$\rho_l \cdot \frac{\partial F H}{\partial t} = \frac{\rho_l}{\rho V_0}\left(-\rho_l \int_\Sigma H_l v_n \, d\sigma + q_\Sigma\right) - \frac{1}{\rho}\frac{\partial \rho}{\partial t} \rho_l F H \tag{5.21}$$

The enthalpy equation is used in this form in the numerical calculations. It effectively implies that $\rho_l FH$, instead of ρFH, is used as the cell total heat content. This simplifies the process of specifying initial and boundary conditions though an additional source term appears on the right-hand side.

A discretised form of Eq. (5.20) is

$$u^{n+1} = u^n + \Delta t^{n+1} \cdot (FX^n + R^n u^l) \tag{5.22}$$

where $u = \rho_l FH$, FX^n denotes advection and heat fluxes terms, and $R^n = \frac{1}{\rho}\frac{\partial \rho}{\partial t}$ estimated at $t = t^{n+1}$. Term Ru is similar to the drag force term, given by Eq. (3.10), except that R can be of any sign.

A simpler version of Eq. (5.22) can be considered to find l,

$$u^{n+1} = u^n + \Delta t^{n+1} \cdot R^n u^l \tag{5.23}$$

If $R^n < 0$ then stability analysis shows that $l = n + 1$ to ensure unconditional numerical stability. Eq. (5.22) is easily resolved for u^{n+1} since the source term is linear. The explicit formulation, *i.e.* $l = n$, is stable only if $\Delta t^{n+1} < 1/R^n$.

If $R^n > 0$ then both explicit and implicit methods give exponentially growing solutions since the analytical solution is an exponent. The explicit method, however, is preferable because it always gives zero overshoot, while the implicit method gives zero overshoot only if $\Delta t^{n+1} < -1/R^n$.

The drag coefficient at $n + 1$ time level is calculated from Eq. (4.55) in which

$$m_l = \rho_l F^n (1 - f^n_{sv}) V_0$$

$$m_s = \rho_s dV^n_s$$

Combining expressions for m_l and m_s with Eq. (5.17), the drag coefficient is given by

$$K^{n+1} = \frac{S^n}{F^n(1 - f^n_{sv})\left(\frac{\rho_l}{\rho_s} - 1\right)} \tag{5.24}$$

5.2.2 Modification of the Solution Algorithm

In an internal cell the liquid/solid mixture volume is constant due to continuity. Since liquid phase is incompressible, the change of mixture volume volume dV^*, Eq. (5.5), must be equal to the net amount of fluid fluxed through the cell boundary in time dt

$$\int_\Sigma v_n \, d\sigma = S\, V_0 \tag{5.25}$$

The numerical implementation of Eq. (5.24) requires the modification of Eq. (3.21) as follows

$$\begin{aligned} L_{i,j,k} &= \frac{u^{n+1}_{i,j,k} A_{FR,i,j,k} - u^{n+1}_{i-1,j,k} A_{FR,i-1,j,k}}{\Delta x_i} \\ &+ \frac{v^{n+1}_{i,j,k} A_{FB,i,j,k} - v^{n+1}_{i,j-1,k} A_{FB,i,j-1,k}}{\Delta y_j} \\ &+ \frac{w^{n+1}_{i,j,k} A_{FT,i,j,k} - w^{n+1}_{i,j,k-1} A_{FT,i,j,k-1}}{\Delta z_k} - S^n_{i,j,k} = 0 \end{aligned} \tag{5.26}$$

while $\partial L_{i,j,k}/\partial p_{i,j,k}$, in Eq. (3.17) remains unchanged since the source term S^n is formulated explicitly and does not depend on p^{n+1}.

For a solidifying cell $S < 0$ if $\rho_s > \rho_l$, leading to a negative value of the cell velocity divergence, *div* v. The latter means that there is a net flux of liquid metal *into* the cell. Similarly, if $S > 0$, the velocity divergence is positive and the remaining liquid is pushed out of the cell.

Special care must be taken if a cell (or a cluster of adjacent cells) containing both phases are surrounded by solidified cells. In this case velocities at cell faces are equal to zero and if $S \neq 0$, then there is no solution for the cell pressure if conditions of incompressibility and continuity are not dropped.

If $S > 0$ then the remaining liquid must flow out of the cell. In practice pressure in such confined region will increase causing solid phase deformations and, perhaps, even partial remelting. Since these effects are not described by the present model, such situations are resolved by setting to zero the net volume increase in the confined region, *i.e.* $S = 0$, so that numerical solution does exist. This is equivalent to assuming that the liquid becomes compressible to accommodate the increase in volume.

5.2.3 Opening Internal Voids: $S < 0$

The solution is more complicated if the source term, S, in a confined cell is negative. In this case

$$L_{i,j,k} = -S_{i,j,k} > 0$$

and, according to Eq. (3.22), pressure will decrease indefinitely.

In practice this will cause dissolved gasses to evolve into bubbles and to open internal voids which, after complete solidification, remain as shrinkage defects. A critical pressure, p_{cr}, at which bubbles start to appear is employed in cavitation models [3].

The same approach is employed in the present numerical model. The value of the critical pressure is unimportant so long as it is much smaller than the pressures in the casting to avoid ambiguity on where a cavity should be created. This is because cell pressure serves as the only flag to open a cavity. Since the actual value of p_{cr} is arbitrary and internal cavity volumes at every time step are defined only by the amount of the volumetric shrinkage, the gas evolution process is not described by the model.

When

$$p_{i,j,k} \leq p_{cr} \tag{5.27}$$

then the cell changes from an 'incompressible and continuous' cell, for which convergence criterion is

$$\mid L_{i,j,k} \mid < \epsilon \cdot V_{F,i,j,k}$$

and pressure is defined by iterations, to an 'internal cavity' cell, for which

$$p_{i,j,k} = p_{cr}$$

and the value of $L_{i,j,k}$ is defined by iterations. The condition of fluid continuity in the cell is dropped and the cell then serves as a feeding basin for other solidifying cells.

This algorithm as based on the cavitation model in the original version of FLOW-3D. In the latter, input parameter PCAV defines the critical pressure.

However, there are differences.

In the FLOw-3D cavitation model, cells in which $p < PCAV$ are defined at each time step *before* the iteration begins, so that during iteration their pressure is driven to $PCAV$ and a net outflow of fluid from these cells is allowed to open a cavitation void. The 'cavitation' status of the cells is not changed during iteration. The rate of the cavity growth is a free parameter and is defined by the user.

In the present shrinkage model the status of a cell *can* be changed by the algorithm during iterations. Suppose after l iterations pressure in a cell is greater than p_{cr}:

$$p^{l}_{i,j,k} > p_{cr}$$

and

$$L^{l}_{i,j,k} > 0$$

Then it follows from Eqs. (3.24) and (3.25) that

$$\Delta p^{l+1}_{i,j,k} < 0$$

If

$$p^{l+1}_{i,j,k} = p^{l}_{i,j,k} + \Delta p^{l+1}_{i,j,k} < p_{cr}$$

then the following adjustment is made

$$\Delta p^{l+1}_{i,j,k} = p_{cr} - p^{l}_{i,j,k} \qquad (5.28)$$

so that

$$p^{l+1}_{i,j,k} = p_{cr}$$

Velocity adjustments for the cell (i, j, k) (Eqs. (3.24) and (3.25) and corresponding expressions for the other components) are made using Δp given in Eq. (5.27). For the next iteration cell (i, j, k) is marked as an 'internal cavity' cell.

In the subsequent iterations, $l' > l + 1$, the magnitude of $L_{i,j,k}^{l'}$ is not required to be below the value of the convergence criterion ϵ if $p_{i,j,k}^{l'}$ does not increase. As soon as the value of $p_{i,j,k}^{l'}$ exceeds p_{cr}, the 'incompressible and continuous' status is returned to the cell, even if a cavity has already been created in it.

This procedure serves two purposes:

- Situations are avoided where iterations are carried out on equations which do not have a solution. The latter would happen if pressures and velocities are iterated in a confined region and the continuity condition is imposed on every cell in the region[1].

  Allowing cells with $p \leq p_{cr}$ to have a non-zero velocity divergence ensures the existence of a solution.

- More than one cell can appear to have pressure below p_{cr}. This allows the openning of internal voids in several cells simultaneously, *i.e.* in one time step.

In the shrinkage model the critical pressure is substantially lower than casting (atmospheric) pressure. If the confined liquid region consists of several cells then a uniform pressure decrease is required until the lowest cell pressure in the region reaches the value of p_{cr}. This uniform pressure adjustment represents an error component of an infinite wavelength which, as mentioned in Section 2.3.2, is the slowest to converge.

To avoid convergence difficulties in such situations, an additional subroutine was incorporated into the code. At the beginning of every time step it finds all confined liquid regions. A liquid region is considered confined if its boundary consists of mould cells and/or fluid cells that cannot feed and no internal cavity has yet been opened in it. A cell cannot feed if the drag force coefficient in it is infinitely large (which may be the case even if the cell still contains liquid as described in Section 5.2.4). Then a uniform pressure adjustment is made in each of the identified regions before the iterations start, so that the minimum pressure in it is exactly equal to p_{cr}. A uniform change of pressure in a confined liquid region does not influence the

[1] A predefined maximum number of iterations per time cycle, by default equal to 1000, is used in FLOW-3D to stop excessive iteration.

solution for velocities in the region since only pressure gradient is present in the incompressible fluid equations.

If the net source term S in a confined region is positive than the situation is resolved as described in Section 5.2.2. If S is negative than it will take only one iteration for the minimum pressure in the region to fall below the value of p_{cr} and the corresponding cell (and possibly some of its neighbours) will be the location of the opening of a new internal void.

As the solidification proceeds internal cavities grow. Eventually some cells may turn into surface cells or become completely empty. Pressure in the empty cells is set to be equal to p_{cr} and treated similarly as all other void regions, and pressures in the surface cells are set in the usual way, that is using Eq. (3.26).

5.2.4 Feeding Criterion

In many situations feeding criterion, such as critical solid fraction $f_{s,cr}$, remains a useful albeit arbitrary simplification of shrinkage modelling.

Ideally, the model should adjust fluid pressures and velocities according to the drag force by solving the flow equations. For example, if liquid has to flow through a high drag area to feed a solidifying region then the pressure in the region should decrease as the drag force increases. Eventually the pressure reaches the critical value and a cavity starts to open in the solidifying region, though some feeding may still occur despite the high drag.

In practice, iteration convergence slows in the presence of significant drag force, especially if the high drag region is located at the entrance into a large volume of liquid which needs to be fed, as shown in Fig. 5.1. Pressure in such a volume must be lowered to pull the fluid from outside through the high drag region. Convergence rate will decrease since this pressure adjustment introduces a long wavelength error component.

To avoid solving flow equations in the presence of a high drag force, a critical value of solid fraction is used as a cut-off parameter for flow.

For the drag force coefficient, given by Eq. (5.23), the value of the term $K' = \frac{1}{1+K\Delta t}$ in Eqs. (3.25)-(3.27) lies usually above 0.9 and only for $f_s \rightarrow 1$ does this term plays a substantial role (depending, of course, on the size of the time step as well).

Therefore, for pure materials, for which Eq. (5.23) has been derived, the drag force poses little difficulty in achieving convergence. In accordance with this, it is assumed that for pure metal a cell can feed and be fed if $f_s < 1$. Flow through a cell seizes, *i.e.* $K' = 0$, when the cell becomes fully solid. In other words, for pure metals $f_{s,cr} = 1$.

For alloy systems flow in a mushy zone experiences much larger resistance. Instead of using a high value of the drag coefficient in such locations, a critical solid fraction $f_{s,cr} < 1$ is used so that flow into a cell ceases when $f_s \geq f_{s,cr}$. Eq. (5.23) is still used for solid fraction below $f_{s,cr}$. This treatment of alloys is an idealisation of the flow in mushy zones, when for low f_s values the melt represents a mixture of liquid and freely floating solid crystals. As solidification progresses crystals starts forming a rigid structure and the flow resistance increases steeply over a small range of f_s values.

The main limitations of the feeding criterion approach are:

- in alloys flow resistance can be anisotropic. Feeding is easier along the columnar crystals than across, and a single value of $f_{s,cr}$ is not sufficient to resolve such situation. In that sense the critical solid fraction gradient [112], ∇f_s, (Section 2.9) or anisotropic drag coefficient [117] (Eqs. (4.41), (4.42)) offer some flexibility;
- the value of $f_{s,cr}$ varies for different materials, casting geometries and solidification conditions. The right choice of $f_{s,cr}$ is almost an art.

The present shrinkage model has been developed to predict macro-defects in castings, since micro-porosity simulation requires modelling of gas evolution, interdendritic flow, nucleation and dendrite growth. The value of $f_{s,cr}$ divides the total shrinkage volume between micro- and macro-porosity. Here the critical fraction of solid method is employed to predict *macro-cavities*, unlike most other shrinkage models. Micro-porosity will inevitably be present as a consequence of using $f_{s,cr} < 1$ for alloy systems, but its spatial distribution is not claimed to be accurate and will not be compared with experimental data[2].

The value of the critical solid fraction is included in the list of input parameters specified in PREPIN.INP file as FRSLCR. However, for pure materials it is always equal to one.

[2] If the average micro-porosity in the solidified casting is known *a priori*, *e.g.* from the liquid metal gas content, then the solid phase density ρ_s can be reduced correspondingly to account for that. Modelling of micro-porosity can then be avoided by setting $f_{s,cr} = 1.0$.

5.3 Simplified Shrinkage Model

The shrinkage model described above is capable of modelling liquid metal volumetric shrinkage/expansion due to phase transformation, including the induced flows and cavity formation, occurring in both phase change directions. The model is applicable to most general 3-D mould and casting configurations.

The cost of that is the necessity to solve fluid flow equations not only during the filling stage, but also during solidification making calculations CPU intensive. Furthermore, the time step size is smaller than in a no-flow situation. Usually the most severe restriction comes from the free surface gravity waves stability criterion, Eq. (2.77) of Section 2.10 [3]. Since the solidification stage is normally much longer than the filling stage, a significant increase in the CPU time is obvious.

A simplification of the full shrinkage model (hereafter referred to as model M1) can be found by considering the following [110,126]:

- In most castings the phase transformation proceeds in one direction: from liquid to solid. This means that the volumetric shrinkage also proceeds in one direction.
- Often solidification occurs in such conditions that the liquid metal free surface is horizontal and flat all the time.
- In most situations gravity feeding can certainly be assumed to occur instantaneously, *i.e* shrinkage induced forces are much smaller than gravity forces.

With these assumptions, another shrinkage model (M2), which does not solve the fluid flow equations, has been developed. In brief, at each time step

1. Isolated liquid regions are identified in the casting. This time an isolated region is defined as one bounded by solidified cells or mould, *and* free surface.
2. The total volumetric change due to phase change over Δt is estimated in each of the regions and removed from the top of it. The 'top' of a region is defined by the direction of the gravity.

[3] The CFL criterion (Eq. (2.75)) is not a problem since normally flow velocities are small during solidification.

5.3.1 Numerical Implementation of the Simplified Model

It is essential that gravity is normal to one of the coordinate planes and the free surface is aligned to that plane.

At time step $n+1$ the volumetric change in each cell $dV_{i,j,k}^{n}$, occurred during the previous time step Δt^{n}, is estimated using Eq. (5.16).

Eq. (5.13) is not solved in every cell directly. For an isolated liquid region R_l of volume V_l^n the total change of volume is

$$dV_l^n = - \sum_{(i,j,k)\in R_l} dV_{i,j,k}^{n} \geq 0$$

and

$$V_l^{n+1} = V_l^n - dV_l^n$$

The volume subtraction is executed in the following manner (assuming gravity is pointing in the negative z direction).

1. the 'top' layer of cells $k = k^*$ is defined so that $k^* + 1$ represents either void or mould or solid metal;
2. the total available liquid volume, $V_{l_k^*}$, is estimated for k^* cells in R_l

   $$V_{l_{k^*}} = \sum_{(i,j,k^*)\in R_l} (1 - f_{sv,i,j,k}^{n}) \cdot F_{i,j,k}^{n}$$

   where $f_{sv,i,j,k}^{n}$ is the volumetric solid fraction in cell (i,j,k) at time t^n.
3. if $V_{l_{k^*}} < dV_l^n$ then for $(i,j,k^*) \in R_l$

   $$F_{i,j,k^*}^{n+1} = f_{sv,i,j,k^*}^{n}$$

   and

   $$dV_l^n = dV_l^n - V_{l_{k^*}}$$
   $$k^* = k^* - 1$$

   and the procedure is repeated from step 2;
4. For $V_{l_{k^*}} \geq dV_l^n$. Since gravity is normal to k^* coordinate plane the liquid cells in this plane should have the same value of the fluid fraction, F^{n+1}. However, the fluid mixture level in the cells can only be reduced at the most by their liquid phase content. Therefore,

(a) an estimate to F^{n+1} is made assuming that the mixture level is reduced in every cell by the same amount

$$F'^{n+1} = \frac{V_{F_{k^*}} - dV_l^n}{V_{k^*}}$$

where initially $V_{F_{k^*}}$ is the total *fluid* volume of k^* cells of tĥe region and V_{k^*} is the total open volume of the k^* cells.

(b) if in cell (i, j, k^*) $F'^{n+1} < f_{sv,i,j,k^*} \cdot F^n_{i,j,k^*}$
then

$$\begin{aligned} F^{n+1}_{i,j,k^*} &= f_{sv,i,j,k^*} \cdot F^n_{i,j,k^*} \\ V_{F_{k^*}} &= V_{F_{k^*}} - F^n_{i,j,k^*} \\ dV_l^n &= dV_l^n - \left(F^n_{i,j,k^*} - F^{n+1}_{i,j,k^*}\right) \\ V_{k^*} &= V_{k^*} - V_{i,j,k^*} \end{aligned}$$

where V_{i,j,k^*} is the open volume of cell (i, j, k^*). The procedure is then repeated from step (a) with cell (i, j, k^*) excluded from the consideration.

A schematic representation of the algorithm is shown in Fig. 5.2.

Enthalpy Eq. (5.20) is solved in a reduced form

$$\rho_l \cdot \frac{\partial F\, H}{\partial t} = \frac{\rho_l}{\rho\, V_0} \cdot q_\Sigma \tag{5.29}$$

because the source term in Eq. (5.20) is balanced by the surplus of enthalpy delivered by the net fluid flow into the cell. As the flow term is excluded from consideration in the M2 model, so must be the source term.

The drag force coefficient calculation is also simplified because it is only necessary to know when the drag becomes infinite to identify isolated liquid regions

$$K = \begin{cases} 0 & \text{if } f_s < f_{s,cr} \\ \infty & \text{if } f_s \geq f_{s,cr} \end{cases} \tag{5.30}$$

The advantage of the M2 model is:

- CPU time is saved as momentum and continuity equations are not solved and no iteration is needed to obtain velocities and pressures. Enthalpy equation is also simplified. The time step size, controlled by heat conduction/transfer, is usually considerably larger that when the full system of equations is solved.

The limitations of the M2 model are:

- Free surface must be plane and normal to the gravity; its orientation must not change during solidification.
- The mesh must be chosen in such a way that the free surface is parallel to one of the coordinate planes. In other words, body forces direction have to be constant and parallel to one of the mesh directions.
- Volumetric expansion, *e.g.* due to remelting, cannot be modelled.
- No flow phenomena is possible. Feeding is only taken into account by lowering the liquid metal free surface. Fluid velocities are not explicitly estimated. Thus, effects of thermal convection are neglected. Processes where advection is important, such as continuous and centrifugal casting, are out of the scope of the simplified model.

The simplified shrinkage model is similar to the one developed by J.Evans [120] (Section 2.9) with two major differences: first, the M2 model is applicable to general 3-D casting configurations (with the limitations stated in the preceding paragraph), while Evans' model was designed to model only centre line porosity in cylindrical ingots. Secondly, cells here are allowed to be partially filled, while in Evans' model they can only be either full or empty: a cell is not emptied until enough shrinkage volume has accumulated to empty it completely. Furthermore the same value of the critical fraction of solid is used in all directions in the M2 model.

The choice between M1 and M2 models is made by the user through the input parameter ISHR: ISHR=1 initiates model M1 and a value of ISHR above 1 initiates model M2.

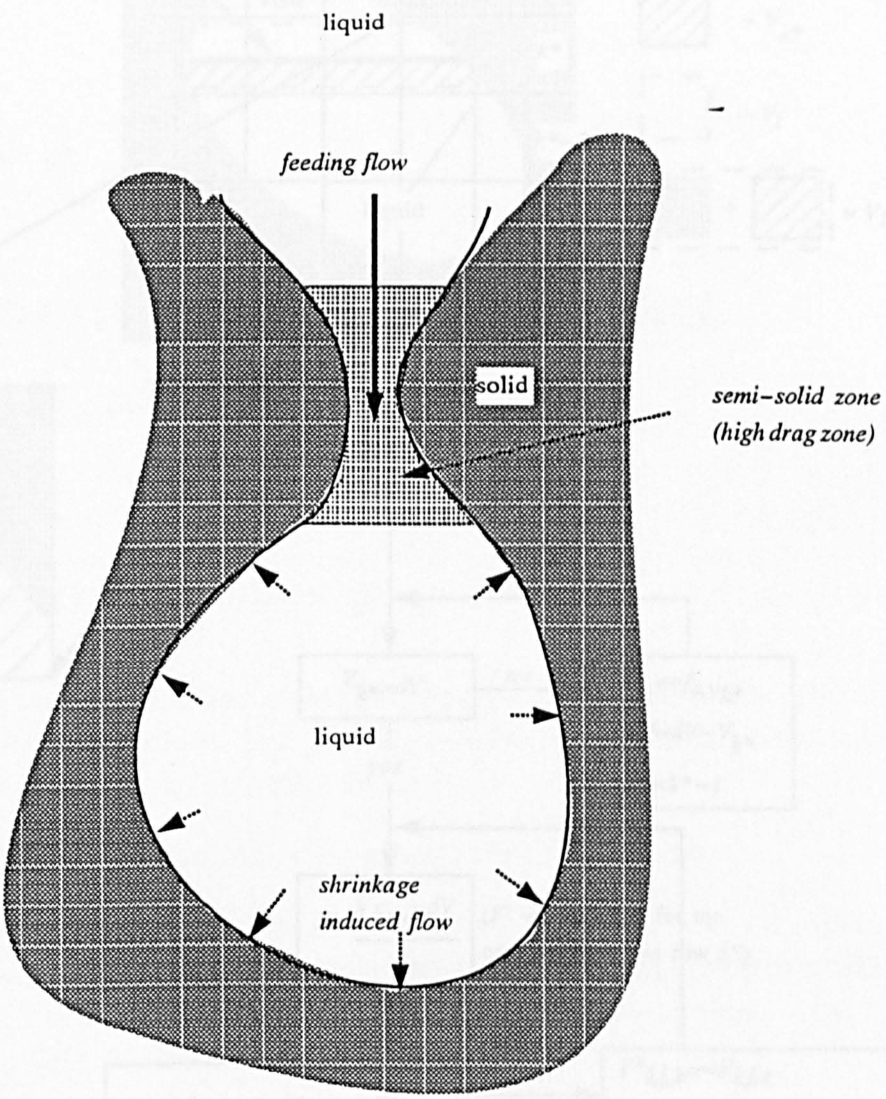


Figure 5.1. Feeding of the shrinkage in a large liquid volume occurring through a high drag zone can slow down convergence in the region.

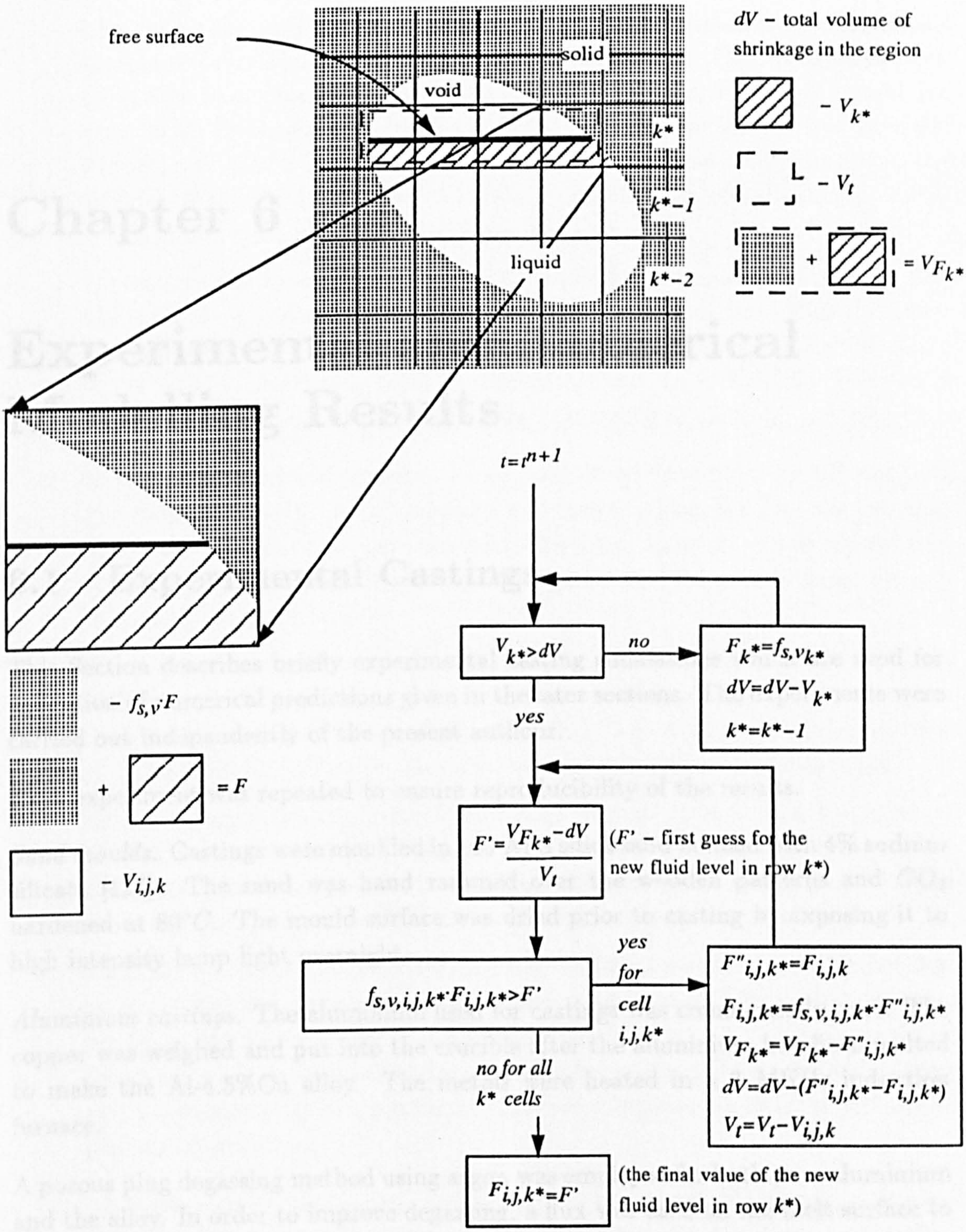

free surface
solid
void
k^*
k^*-1
k^*-2
liquid
dV – total volume of shrinkage in the region
– V_{k^*}
– V_t
+ = $V_{F_{k^*}}$
– $f_{s,v} \cdot F$
+ = F
– $V_{i,j,k}$
$t = t^{n+1}$
$V_{k^*} > dV$
no
$F_{k^*} = f_{s,v_{k^*}}$
$dV = dV - V_{k^*}$
$k^* = k^* - 1$
yes
$F' = \frac{V_{F_{k^*}} - dV}{V_t}$
(F' – first guess for the new fluid level in row k^*)
$f_{s,v,i,j,k^*} \cdot F_{i,j,k^*} > F'$
yes for cell i,j,k^*
$F''_{i,j,k^*} = F_{i,j,k}$
$F_{i,j,k^*} = f_{s,v,i,j,k^*} \cdot F''_{i,j,k^*}$
$V_{F_{k^*}} = V_{F_{k^*}} - F''_{i,j,k^*}$
$dV = dV - (F''_{i,j,k^*} - F_{i,j,k^*})$
$V_t = V_t - V_{i,j,k}$
no for all k^* cells
$F_{i,j,k^*} = F'$
(the final value of the new fluid level in row k^*)

Figure 5.2. Schematic diagram for the simplified shrinkage model (M2) algorithm for section 5.3.1.

Chapter 6

Experimental and Numerical Modelling Results

6.1 Experimental Castings

This Section describes briefly experimental casting simulations which are used for validation of numerical predictions given in the later sections. The experiments were carried out independently of the present authour.

Each experiment was repeated to ensure reproducibility of the results.

Sand moulds. Castings were moulded in 100 AFS silica sand bonded with 4% sodium silicate [127]. The sand was hand rammed over the wooden patterns and CO_2 hardened at 80^oC. The mould surface was dried prior to casting by exposing it to high intensity lamp light overnight.

Aluminium castings. The aluminium used for castings was commercially pure. The copper was weighed and put into the crucible after the aluminium had been melted to make the Al-4.5%Cu alloy. The metals were heated in a 3 MKHz induction furnace.

A porous plug degassing method using argon was employed for both pure aluminium and the alloy. In order to improve degassing, a flux was used on the melt surface to avoid air entrapment during degassing.

1. *Aluminium-sand cylindrical casting (E1).* In this experiment pure aluminium was cast into a 100 mm diameter and 150 mm length sand mould [128]. Fig.

6.1a shows the overall geometry and dimensions of the casting. The metal was poured directly into the mould at 90^0K superheat. The filling stage took a fraction of a second so that initial conditions in the metal and mould are close to an instantaneous filling situation. Initially the mould was at room temperature of 293^oK. Immediately after the filling had been completed the open top of the metal was covered with an insulating board preheated to 770^oK to minimise heat losses from the metal free surface.

Thermocouples were placed both in the mould and metal, as shown in Fig. 6.1b, and temperature readings were recorded for half an hour after the pouring. According to the thermocouple located at the centre of the casting (th/c No. 1 in Fig. 6.31), the total solidification time, t_s, is 754.0 s. Figure 6.3 shows the sectioned casting with a shrinkage cavity at the top.

2. *Chill mould cylindrical castings.* These castings were produced by J.Evans [126]. The mould geometry and dimensions are shown in Fig. 6.4 The top cylindrical mould was made of grey cast iron. This was fixed on top of a pure iron cylindrical mould of the larger diameter, positioned on a sand base, using a 18 mm thick steel plate.

   One casting was made in pure iron (E2) and a second in 0.6%C, 1.1%Mn, 0.2%Si steel (E3). Since any gas evolution would have affected the shape of the shrinkage cavities, the castings were fully killed. This required the addition of 0.15%Si and 0.05%Al to the pure iron and 0.05%Al to the alloy steel.

   Both castings were top poured, the pure iron at 1843^oK and the alloy at 1873^oK corresponding to 34^oK and 64^oK degrees of superheat, respectively. The top of the casting was then insulated using Kaowool board. Solidification times are not known exactly since no thermocouple data is available.

   Figs. 6.5a, b show the sectioned castings. The pure metal casting has two major centre-line shrinkage cavities: one at the top and the other, internal cavity, at the location where the mould diameter changes. The alloy casting shows a similar pattern but the two cavities are connected by a narrow passage and the lower cavity extends farther down the casting axis than in the case of the pure iron casting.

3. *Aluminium-sand boot-shaped casting (E4).* This essentially three-dimensional casting was produced by J.Ortega [18]. Fig. 6.6 shows the shape and dimensions of the mould. The metal was poured into the mould through the feeder top with a small superheat of 10^0K and filled the mould in 4.7 seconds. According to the thermocouple readings t_s = 490.0 s. The top of the casting was not insulated so that heat was lost to atmosphere during solidification.

Fig. 6.7 shows the centre plane section of the solidified casting with a well defined cavity in the feeder.

4. *T-shaped sand casting.* These simple shaped casting experiments allowed the influence of changes in gating and feeding on the size and location of shrinkage cavities to be assessed [129]. In particular, the position and size of cavities at the T-junction were observed. Casting dimensions and overview are shown in Figs. 6.8 and 6.9, respectively. The top of the feeder is open to atmosphere.

   The runner system had been designed to ensure control of the filling conditions of the casting. A reservoir at the top of the sprue was first filled with the melt, keeping the sprue entrance closed with a stopper (Fig. 6.8*b*). The stopper was removed as soon as the melt temperature, measured by a single thermocouple, reached a specified value.

   The filling time was varied by changing the diameter of the choke, d_{ch}, at the bottom of the sprue. Three values were used: $d_{ch} = 8\ mm$, $15\ mm$, and $22\ mm$.

   Feeding conditions at the T-junction were varied by changing the vertical to horizontal section thickness ratio, $R_{a/b} = a/b$ (Fig. 6.8*a*), *viz.* $R_{a/b} = 0.6$, 1.0, and 1.67.

   Two values of pouring temperature, $T = 963^oK$ and $T = 1013^oK$, were used corresponding to 30^oK and 80^oK of superheat, respectively.

   (a) *Pure aluminium (E5-E16).* Figs. 6.10 and 6.11 show the centre plane section of the casting for pure aluminium with typical shrinkage cavities. The table in Fig. 6.12 summarises the results of this set of experiments.

   When $R_{a/b} = 1.0$ (E7-E12) a small cavity appears in the horizontal section of the T-junction. The cavity is either oval in shape and positioned at the junction (Fig. 6.11*b*) or flat and wide and positioned in the horizontal section (Fig. 6.11*a*). Figs. 6.13*a*, *b* show the X-ray photographs of these cavities. The bottom and top surfaces of the horizontal section underneath and above the cavity are slightly concave. This could reduce the size and position of the cavity.

   The cavity at the T-junction for $R_{a/b} = 1.67$ (E13, E14) consists of two separate parts (Fig. 6.10*c*). A likely reason for that is the aspiration of air at one of sharp corners of the junction due to a low pressure in the metal. The same could have happened in the casting shown in Fig. 6.11*b* (castings E10-E12 in Fig. 12), where the cavity has a round bubble-like shape with smooth surface.

Two variations of the mould design were made in an attempt to eliminate the defect. First, a 10 *mm* wide chill was inset into the mould (E15) and a 10 *mm* thick insulating pad was used in the second case (E16, Fig. 6.14), both leading to the disappearance of the shrinkage cavity as shown in Fig. 6.15.

(b) *Al-4.5%Cu alloy (E17-E21).* When an Al-4.5% was used instead of pure metal no shrinkage occurred in the T-junction for $R_{a/b} = 1.0$ (E18-E19) but castings were subjected to some microshrinkage. The amount of the latter depends on the hydrogen content of the melt. A small cavity though is present at the junction in the case of $R_{a/b} = 1.67$ (E20-E21).

Fig. 6.16 shows centre plane section of the alloy castings and other experimental results for the alloy are summarised in the table in Fig. 6.12.

Typical primary cavity shapes in the feeder for pure aluminium and alloy castings are shown in Fig. 6.17.

6.2 Full Shrinkage Model (M1)

This Section presents validation results of the full shrinkage model against two simple test of the model first and then the experiments listed in Section 6.1. All computer simulations are summarised in Appendix A while Fig. 6.19 gives the computational efficiency data for some of the calculations, together with predicted solidification times.

Constant values of material thermophysical properties are used for both for metal and mould, though solid and liquid phase properties can differ:

- silica sand, Table 1 [7]

- pure aluminium, Table 2 [7]

- Al-4.5%Cu alloy, Table 3 [130]

- pure iron, Table 4 [126]

- 0.6%C, 1.1%Mn, 0.2% alloy steel, Table 5 [126]

6.2.1 2-D Test Casting Simulations

The two simple simulations presented in this Section assumed instantaneous filling with uniform initial temperature distribution at the melting point of the metal.

A. Casting with a narrow feeding passage (S1.1, Appednix A). In this two-dimensional pure aluminium-chill casting simulation the mould has a narrow passage through which the left-hand part of the casting (section A) is fed at the early stage of solidification by pressure head in section B, as shown in Fig. 6.20*a*. This feeding pattern prevails until the passage is blocked by fully solid metal ($f_{s,cr} = 1.0$ for pure materials). The direction of the feeding in section A then changes to the opposite leading to a cavity appearing at its top (Fig. 6.20*b*). The critical pressure, p_{cr}, at which internal cavities would appear in the metal, was set at $-10^4\ N/m^2$. The ambient gauge pressure is zero. As was said in Section 5.2.3, p_{cr} can have any value below the gauge pressure.

History plots for the pressure and vertical velocity in a cell located at the centre of section A are shown in Fig. 6.21*a*. First, pressure decreases continuously due to the decrease of the pressure head in section B. Then its value drops to $p_{cr} = -10^4\ N/m^2$ with a simultaneous change of the vertical velocity direction when the narrow passage solidifies.

As expected, predicted feeding flow velocities are higher at the beginning of solidification, when the cooling rate is larger, and the average is $0.8 - 1.0\ mm/s$. At the end of the process they decrease to around $0.2 - 0.4\ mm/s$.

Fig. 6.22 shows the resulting porosity distribution in the solidified casting and Figs. 6.21*b* – *d* show the iteration count for $\epsilon = 0.001\ s^{-1}$, the fluid volume, V_f, and the mean fluid kinetic energy, E_m, evolutions. The iteration number increases temporarily when the passages narrows at $t \approx 12.0\ s$ and feeding becomes more difficult.

The volumetric shrinkage is 6.3%, defined by the values of the liquid and solid phase densities: 2540 kg/m^3 and 2710 kg/m^3, respectively. The theoretical shrinkage volume, ΔV_{th}, is defined by equation

$$\Delta V_{th} = V_l(1 - \frac{\rho_l}{\rho_s}) \tag{6.1}$$

The initial liquid metal volume, V_l, is 8,700 mm^3, so that ΔV_{th} is 545.76 mm^3. The predicted value is $\Delta V_p = 548.80\ mm^3$ giving an error of 0.56% most of which, $\Delta V_c = 3.0\ mm^3$ or 0.55%, is a consequence of the non-zero convergence criterion and can be reduced by decreasing ϵ[1]. Therefore, it can be concluded that the full shrinkage model predicted the shrinkage volume practically exactly.

The mesh is uniform in both directions and the time step size is $\Delta t = 1.6 \times 10^{-2}$ s throughout the calculation as dictated by the free surface stability criterion (Eq. (2.77)). The total simulation time is 50.0 s while solidification finishes in 44.0 s. The total CPU time[2] for this problem is $t_{CPU} = 128$ s with the average iteration count per time cycle of $N_{it} = 15$.

B. Square Casting (S1.2). A simulation of a square, symmetrical casting has become almost a classical test for shrinkage modelling [121]. The model must be able to predict the vertical shift in the position of the shrinkage cavity due to the gravity.

A pure aluminium casting, 50×50 mm^2, is surrounded by the sand mould, 90×90 mm^2, meshed by a uniform grid of 9×9 cells. Initially heat transfer and conduction fluxes are symmetrical in all four directions (Fig. 6.23*b*).

Fig. 6.23 shows the final cavity shape and the developed bottom-top asymmetry of the metal and mould temperatures due to the shrinkage induced gap formation at the top of the casting. As expected, the cavity is not located at the position of the last-liquid-to-solidify, *i.e.* at the casting centre. This result is an improvement from that of Fryer *et al* [121] who showed the cavity to be at the centre of the casting.

Predicted feeding flow velocities average around $0.1 - 0.2$ mm/s. Due to the stricter convergence criterion of $\epsilon = 10^{-6}$ s, ΔV_c constitutes only about $10^{-5}\%$ of the total metal volume and ΔV_p corresponds to the theoretical value with the 0.002% accuracy.

6.2.2 Aluminium-Sand Cylindrical Casting

A series of simulations of casting E1 described in Section 6.1 has been carried out to test the mesh sensitivity of the shrinkage model. The filling stage is excluded from the simulation since the casting was filled very quickly. Instead, uniform initial temperature distribution is assumed at $90^o K$ superheat.

The properties of the insulating material, obtained from the supplier, are given in

[1] Since $\epsilon > 0$ then each cell of volume ΔV has a net inflow or outflow of the order of $\epsilon \cdot \Delta V$ after the convergence criterion is satisfied. The total volumetric error is recordered at each time step as an output variable ΔV_c (see also Section 7.2.4).

[2] All simulations were made using a Silicon Graphics Indigo 4000 computer.

Table 6. Initially the insulator was at 770^oK.

Fig. 6.24 shows the computational domain for simulation S2.1 (see Appendix A). Only half of the casting was enmeshed due to axisymmetry. The meshes were chosen to be uniform in both x- and z-directions with $\Delta x = \Delta z = \Delta$ to ensure equal resolution at both horizontal and vertical metal/mould interfaces. The following meshes were used:

(S2.1) 6×15, $\Delta = 16.7\ mm$;

(S2.2) 12×30, $\Delta = 8.3\ mm$;

(S2.3) 20×50, $\Delta = 5\ mm$;

(S2.4) 36×90, $\Delta = 2.8\ mm$;

all with the same convergence criterion $\epsilon = 10^{-4}\ s^{-1}$.

The time step size in each case was defined either by the free surface or by the conduction stability limit, depending on whether the are liquid $surface$ cells or not, and in average equals to $\Delta t_{S2.1} = 1.0\ s$, $\Delta t_{S2.2} = 0.15\ s$, $\Delta t_{S2.3} = 0.04\ s$ and $\Delta t_{S2.4} = 0.17\ s$.

The total CPU time, the average CPU time per time cycle and per iteration, together with the time step size, are given in Fig. 6.25*a* as functions of the total number of cells N. The iteration number does not vary substantially between the four cases, averaging to 3-5 iterations per time step increasing to around 10 iterations only for the finest mesh (S2.4). Simulations were continued until the last liquid metal solidified. Since t_s is different in all four cases, t_{CPU} in each case corresponds to a somewhat different physical time (see Fig. 6.25*b*).

Predicted cavity shapes and temperature distributions at the end of solidification for all four cases are given in Fig. 6.26 (compared with Fig. 6.3). Fig. 6.25*b* shows the predicted depth of the shrinkage cavity, h_{sh}, and t_s in per cent of the experimental results as a function of N. For the finest mesh (simulation S2.4) the cavity depth is underestimated by 14% and the solidification time is 27% above the experimental value of 754 s.

Fig. 6.25*c* shows ΔV_c and the error of ΔV_p in relation to the theoretical value of the shrinkage volume ΔV_{th} given by Eq. (6.1). The errors are small, less than 0.3%, and the accuracy of the shrinkage volume prediction is mainly defined by the convergence errors ΔV_c. Both ΔV_c and ΔV_p show little dependence on mesh resolution as does the average feeding flow velocity in the bulk of the liquid metal also plotted in Fig.

6.25*c*. ($\approx 0.02\ mm/s$).

The progress of solidification for simulation S2.2 is given in Fig. 6.27. Figs. 6.28*a*, *b* shows the evolution of the horizontal and vertical velocity components in a cell at the casting geometrical centre. The absolute values of both components grow as the solidification front approaches the cell, reaching a maximum of abouT 0.05 mm/s. The share of the horizontal feeding in the total flow in the cell also increases, although the vertical velocity is by far the dominant one. Figs. 6.29$a - d$ show the evolution of the time step size, iteration count, the total fluid volume and the mean kinetic energy.

Another simulation (S2.5) was carried out with a non-uniform mesh, $N = 19 \times 33$ cells, with finer resolution, $\Delta x = \Delta z = 2\ mm$, at metal/mould interfaces as shown in Fig. 6.30*a*. The resulting cavity shape is shown in Fig. 6.30*b* and other output results are presented in Fig. 6.25$a - c$. It can be seen that the non-uniform mesh makes the calculation more accurate though reducing the maximum time step size. Fig. 6.31 shows the comparison of predicted temperature histories with the experimental measurements at locations shown in Fig. 6.1*b*.

6.2.3 Chill Mould Cylindrical Casting

This Section includes modelling of casting E2 and E3 described in Section 6.1. The pouring stage is omitted since the experimental filling times are short, besides, fast cooling rates largely define the temperature distribution in the casting.

The enmeshed computation domain is shown in Fig. 6.32*a* with a 10×53 mesh covering only half of the casting due to the symmetry. The chill mould is assumed to be at constant uniform temperature of $298^{o}K$ throughout the simulation. The top of the casting is fully insulated. Despite the availability of experimentally defined time-dependent metal/mould heat transfer coefficients and interfacial heat fluxes [126], computer simulations were simplified by assuming in each case a constant heat transfer coefficient obtained from the measurements. This assumption was made because only constant heat transfer coefficeints could be used in the standard input to FLOW-3D.

The volumetric shrinkage is 3.85% for both the pure metal and alloy castings, defined by the densities of the liquid and solid metal (see Tables 4 and 5).

6.2.3.1 Pure Iron

This simulation is referred to as simulation S3.1 (see Appendix A). The value of the heat transfer coefficient, $h = 2370.0\ W/m^2K$ corresponds to the maximum heat transfer coefficient for a bare chill [126]. Fig. 6.32$b - e$ shows predicted shrinkage cavities, temperature distribution (degrees Celsius) and solidification front position at $t = 50\ s$, $t = 100\ s$, $t = 150\ s$ and at $t = 200\ s$. Solidification took $t_s = 177\ s$. Comparison with the shrinkage defects given in Fig. 6.5a shows that the predicted shape and position of shrinkage cavities are very close to the experimental result.

Fig. 6.33 shows the plot of the maximum horizontal and vertical velocities in the feeding flow. The flow velocity increases as the vertical liquid channel in the upper part of the casting narrows, before it is completely solid. At its maximum it reaches 25 mm/s which is two orders of magnitude larger than velocity in the bulk of the liquid metal in the lower part.

To investigate the influence of the metal/mould interface properties on the cavity formation simulations with the following modifications were carried out:

- S3.2: the heat transfer coefficient reduced by a factor of ten uniformly along the metal/chill interface, that is $h = 237.0\ W/m^2K$ corresponding to the minimum value for a coated chill [126]. The result is given in Fig. 6.34a;
- S3.3: the heat transfer coefficient reduced by a factor of ten only for the lower, wider part of the chill mould (Fig. 6.34b).
- S3.4: the heat transfer coefficient reduced by a factor of ten only for the upper, narrow part of the chill mould (Fig. 6.34c);

Solidification times in these cases are 457 s, 208 s and 448 s, respectively. Cavity shapes predicted in simulations S3.1 and S3.2 are similar although in the latter the metal does not freeze over in the narrow part of the mould. This suggests that solidification in the lower part of the mould in S3.2 becomes somewhat faster than in the upper part at later stages of the process, so that liquid metal in the vertical channel is drained downwards before it has time to solidify.

There is a small centre line porosity region in casting S3.4 and the primary shrinkage cavity is much shallower than that in casting S3.1 while casting S3.3 has a mush larger secondary cavity. The *variability* of the heat transfer coefficient along the interface appears to play a larger role in the final cavity shape than the actual value of h.

6.2.3.2 Alloy Steel

The alloy solidification and shrinkage model was used here to simulate the steel casting E3 shown in Fig. 6.5*b*. The lever rule was employed to describe the latent heat release and two values of the critical fraction of solid, $f_{s,cr}$ = 0.67 (S3.5) and $f_{s,cr}$ = 1.0 (S3.6), were tested.

Fig. 6.35 shows the distribution of the fraction of solid function at intermediate stages of solidification for $f_{s,cr}$ = 1.0 and the final predicted shape of shrinkage cavities is shown in Fig. 6.36. Note that for $f_{s,cr}$ = 0.67 part of the shrinkage volume goes into the uniform 1% microporosity in the final casting (the total volumetric shrinkage is the same for both simulations). Comparison of the result with the experimental casting E3 given in Fig. 6.5*b* shows clearly that $f_{s,cr}$ = 1.0 gives a better approximation. This may be explained by the relatively narrow mushy zone in the casting during solidification (Fig. 6.35) hence the alloy flow behaviour is close to that of a pure metal. For the same reason the shape of the shrinkage cavity in this case appears to depend little on the latent heat release model. Fig. 6.36c shows predicted cavities for the linear mode for $f_{s,cr}$ = 1.0 (simulation S3.7).

The solidification times are (S3.5) t_s = 190.1 *s*, (S3.6) t_s = 192.5 *s*, and (S3.7) t_s = 193.6 *s*. The average t_s is 8.5% larger than that for the pure metal simulation S3.1 since the alloy solidus temperature is lower than the melting point of the pure metal.

There is a slight increase in CPU time in the calculations for alloy modelling because temperature in every cell is found iteratively from the value of the enthalpy (see Section 4.2.1). The number of iterations required is normally 2 or 3 leading to insignificant increase in computational effort: in average 10% per time step.

6.2.4 Aluminium-Sand Boot-Shaped Casting

Casting E4 of Section 6.1 has been simulated in three dimensions (simulation S4.1, Appendix A). As before, the pouring stage was substituted here by the instantaneous filling assumption at the uniform 10^oK superheat. The top of the casting is open to atmosphere and insulated[3]. Figure 6.37 shows the enmeshed part of the mould. As before, geometrical symmetry was used to simulate the solidification in only half of the casting. The total number of cells is 9361, of which 4,645 are open to the flow, *i.e.* $V_F < 1$.

[3] Free surface is an adiabatic boundary in all simulations in the present work.

The predicted solidification time is 556.1 s which is 13.5% larger than the experimental value. Fig. 6.38 shows a sequence of 2-D centre plane sections of the casting and four 3-D views are given in Fig. 6.39. The predicted shape of the primary cavity in the feeder is in good agreement with the experimental result (Fig. 6.7). There is a small secondary cavity in the casting itself, indicating that the feeder is too small. The experimental casting, however, is sound.

Feeding velocities in the liquid bulk do not exceed 0.4 mm/s. The evolution of the metal volume, mean kinetic energy, iteration count and time step size are given in Fig. 6.40 and other calculation results for this simulation are presented in the table in Fig. 6.19.

6.2.5 T-Shaped Sand Casting: Pure Aluminium

In this Section simulation results of filling, solidification and shrinkage of castings E5-16 (Fig. 6.12) are presented. The alloy modelling is described in Section 6.3.4. Fig. 6.41 shows the mould and the casting, included into the enmeshed domain. Partially blocked cells are clearly visible in the feeder and sprue. A total of 46,816 cells is used here, of which 14,551 are in the casting and the runner system. As is mentioned in Section 6.2.2, it is important to have the same mesh size along the metal/mould interface. This is easily implemented at plane interfaces, but it is impossible to have the same mesh resolution along the cylindrical feeder interface (and the conical sprue) in Cartesian coordinates. The open volume V_F varies in cells at curved interfaces; hence the accuracy of the heat transfer flux calculations in these cells also varies according to Fig. 4.5.

Only the bottom part of the pouring basin was included. A pressure boundary condition, linearly decreasing with time to approximate the change of pressure head in the basin, and a constant pouring temperature, obtained from the experiment, was set. The filling was assumed complete when metal in the feeder was at the level of the sprue entrance, as was done during the experiments. At that time the boundary pressure was balanced by the pressure head in the feeder and practically no liquid entered the casting.

The filling time is controlled by the size of the choke, d_{ch}, at the bottom of the sprue in the same way as was done in the experimental procedure. The following are the results for vertical to horizontal section ratio, $R_{a/b} = 1.0$;

1. $d_{ch} = 22\ mm:\ t_{fill} = 6.0\ s$;

2. $d_{ch} = 15\ mm:\ t_{fill} = 7.02\ s$;

3. $d_{ch} = 8\ mm:\ t_{fill} = 22.5\ s$.

Predicted and measured filling times for pure aluminium and $d_{ch} = 8.0\ mm$ and $d_{ch} = 15.0\ mm$ are compared in Fig. 6.42. Errors vary between -7.6% to 9.8% of the experimental times.

A. $R_{a/b}$=1.0.

Figure 6.43 shows a 3-D view of the filling process for $d_{ch} = 15.0\ mm$. Sloshing and free surface break up are clearly visible in the horizontal section of the casting. Figure 6.44 gives a 2-D view of the filling with metal and mould temperature contours for the 80^oK superheat case (simulation S5.1). The evolution of the mean kinetic energy and total metal volume in the casting during filling are given in Fig. 6.45. The former serves as an indicator of the quality of filling. In the present case the kinetic energy has two peaks: one at the time when the metal hits the bottom of the sprue and the other, smaller, when the metal falls down the lower section of the T-junction.

The effect of the filling time on the temperature distribution in the casting at the end of the filling is given in Fig. 6.46 where results for $d_{ch} = 15.0\ mm$ and $d_{ch} = 8.0\ mm$ with 80^oK of the initial superheat are compared. At the end of the latter, slower filling, the T-junction has already started to solidify since the minimum temperature is 6^oK below the melting point. For $d_{ch} = 15.0\ mm$ there is still at least 25^oK superheat left in the metal after the pouring stopped. Thermocouple readings are also indicated for comparison for $d_{ch} = 15.0\ mm$. The average deviation of the predicted temperatures from the experimental results is 9.2% of the maximum temperature difference in the casting (60^oK). The total solidification time appears to be virtually constant for all filling rates, including the instantaneous filling and equal to 475 s (Fig. 6.19).

Solidification progress, feeding and shrinkage formation of simulation S5.2 are shown in Fig. 6.47. The initial metal superheat was 30^oK. A small liquid region became isolated at the T-junction leading to a shrinkage cavity there.

Figure 6.48 shows that the size of the cavity in the T-junction is practically independent of the superheat.

Figure 6.49. shows a 3-D view on the predicted shrinkage cavities for the three filling times and for an instantaneous filling, all with 80^oK superheat (simulations S5.3, S5.1, S5.4 and S5.9, respectively). The shape of cavities in the junction is thin in the vertical direction and wide in the horizontal plane, similar to the experimental results (Figs. 6.10*b* and 6.11*a* and castings E7-E9 in Fig. 6.12). The cavities lie mostly in the horizontal section slightly protruding into the vertical section. That the cavity size increases as the filling time decreases. There is no cavity when the filling time is 22.5 s ($d_{ch} = 8.0\ mm$, Fig. 6.49*a*), and the cavity size is maximum when the filling is excluded form the simulations (Fig. 6.49*d*).

The general trend is that the faster the filling the larger is the cavity in the T-junction. This may be explained by the fact that a slower filling results in a larger temperature gradient in the direction of feeder–junction at the end of the filling, thus promoting a directional solidification from the junction towards the feeder.

Comparisons of the thermocouple readings at three locations in the metal with the predicted cooling curves for 500 s are shown in Fig. 6.50 for casting E8 and simulation S5.1. The temperatures are shown in degrees Celcius. Thermocouples were placed at the T-junction, at the bottom of the feeder and at the bottom of the sprue. The simulation (broken lines) correctly predicted that

- At the end of the filling stage, at $t = 7\ s$, metal at T-junction has the lowest temperature and metal in the sprue is the hottest.
- Metal at the bottom of the sprue freezes in around 100 s after the beginning of the pouring.
- Metal in the T-junction and at the bottom of the feeder freeze practically simultaneously arnoud 250 seconds after the beginning of the filling.

At the end of the filling stage, the measured temperatures are 726^oC, 736^oC and 744^oC, and the predicted ones are 726^oC, 735^oC and 745^oC, respectively. At $t = 500\ s$, the largest error in the predicted temperatures, 80^o, is at the bottom of the sprure, where the overall temperature variation was 350^o. This may be explained by the lack of resolution in the sprue. The smallest error, 10^oC, is at the bottom of the feeder where the temperature varied by 150^oCin 500 s. The error and the variation in temperature at the third location are 30^oC and 250^o, respectively.

The filling stage was simulated with $\epsilon = 0.005\ s^{-1}$. At the end of the filling calculations were stopped and restarted from the same time with $\epsilon = 0.00025\ s^{-1}$. A smaller ϵ was required during solidification because velocities are significantly smaller at this stage than during filling. The distribution of the variables at the end of the filling are used as the initial conditions for the restart calculations. The table in Fig. 6.19 contains information on the mesh sizes, run times, time step sizes and dreezing times for this calculation.

B. $R_{a/b}$=0.6 and $R_{a/b}$=1.67

The influence of the geometry on the shrinkage in the junction is simulated for the fixed choke size d_{ch} = 15.0 mm with the two values of the pouring temperatures used in Section 6.8.5A. As in S5.1 to S5.3, each simulation here included filling, heat transfer, solidification and the use of the full shrinkage model M1.

For $R_{a/b} = 0.6$ the castings appears to be sound irrespective of the pouring temperature, in agreement with the experiment (Figs. 6.10a and 6.51a, b). The pouring temperature also does not significantly affect the predicted defects in the T-junction for $R_{a/b}$ = 1.67, where a large cavity is predicted in the vertical section of the T-junction, as shown in Fig. 6.51$c - f$. This is in agreement with experimental result shown in Fig. 6.10c. The lower part of the predicted cavity did not show the experimental separate shrinkage defect in the T-junction. The difference may be due to air aspiration during solidification in the experiment, which was not accounted for in the simulation.

6.3 Simplified Shrinkage Model (M2)

In this Section the simplified shrinkage model M2 is tested against the experimental and M1 model results. Further modelling is carried out of the Al-4.5%Cu alloy T-shaped casting. In each case the geometry, mesh and initial conditions are kept the same as those in the corresponding full model calculations of Section 6.2. The only parameter needed to be changed in the input files was ISHR, which specifies the choice of a model (Section 5.3.1).

6.3.1 Aluminium-Sand Cylindrical Casting

The four meshes described in Section 6.2.2 are employed here to investigate accuracy of predictions and calculation speed of the simplified model. Simulations S2.11, S2.12, S2.13 and S2.14, correspond to simulations S2.1, S2.2, S2.3 and S2.4, respectively. Figure 6.53 shows the final predicted cavities for the four cases and these are compared with those shown in Fig. 6.26.

The accuracy of the predicted final values of h_{sh} are presented in Fig. 6.25*b* in per cent of the experimental result. The % errors of depth predictions are almost twice as large as those given by the full shrinkage model. This is despite the fact that ΔV_p is practically exactly equal to the theoretical value. The volumetric error ΔV_c is zero in the M2 model since there are no volume errors associated with fluid flow an free surface advection.

The overall cooling rate and metal volume evolutions for simulations S2.12 with the M2 model and S2.2 with the M1 model are compared in Fig. 6.54. In both cases the differences between the two model predictions are negligible compared with the total variation of the parameters.

The total CPU time and the CPU time per time cycle for simulations S2.1-S2.4, using the M1 model, and for simulations S2.11-S2.14, using the M2 model, are plotted in Fig. 6.55. Due to the fast convergence of the M1 model solution for this casting, the CPU time per time step for the M2 model constitutes only 30 (simluation S2.11) to 80% (simulation S2.14) of that of the M1 model. Additional gain in speed for the M2 model is due to larger time step size, so that, for example, for the S2.14 simulation the total CPU time is more than 10 times smaller than that for the S2.4 simulation.

6.3.2 Chill Mould Cylindrical Casting

Simulations of castings, described in Section 6.2.3, were repeated using the M2 shrinkage model.

A. Pure Iron. Simulation S3.1 was repeated with the simplified shrinkage model (S3.11). Figure 6.56 shows the progress of solidification with temperature contours and solidification front position for pure iron solidification. The final shapes of the cavities predicted by M1 (Fig. 6.32*e*) and M2 models are similar. The solidification times predicted by the two models differ by 2.5 *s* which constitutes 1.4% of the average time.

The time step size was $\Delta t = 1.36$ *s*, which is 186 times larger than that for the M2 model simulation. The total CPU time is only 13.4 *s*, that is 730 times smaller than that for the M1 model. The predicted change of volume ΔV_p constitutes 99.997% of ΔV_{th}, achieving a better accuracy than that of the M1 model.

B. Alloy Steel. Simulations S3.5 ($f_{s,cr}$ = 0.67) and S3.6 ($f_{s,cr}$ = 1.0) were also repeated using the M2 model (S3.15 and S3.16, respectively). Figure 6.57*a*, *b* shows results of the shrinkage cavity predictions which are close to those shown in Fig. 6.36*a*, *b* of the M1 model. The total solidification times for cases S3.5 and S3.15 are 190.1 *s* and 189.0 *s*, respectively, differing by less than 0.6%.

The metal volume evolution for the pure iron and the alloy ($f_{s,cr} = 1.\theta$), calculated by the M2 model, are given in Fig. 6.58. It takes 11 *s* (6.2 %) longer for the alloy to solidify because pure metal freezes at a higher temperature.

The M2 model alloy simulations took 270 times less CPU time than those by the M1 model (Fig. 6.19).

Unlike simulations S2.11-14 for the aluminium-sand cylindrical casting, The M2 model shrinkage predictions for the chill mould iron and steel castings are similar to the results of the M1 model. This indicates that fluid flow and energy advection were less important in the chill mould casting solidification than it were in the aluninium-sand mould one. The reason for that could the fast solidification of metal in the chill mould, more than 6 times faster than the solidification in the sand mould.

6.3.3 Aluminium-Sand Boot-Shaped Casting

The simplified shrinkage model was employed to repeat simulation S4.1 of Section 6.2.4 (simulation S4.11). Figure 6.59 shows the progress of the solidification and shrinkage formation in the centre plane. Following a tendency of the simplified model to give shallower cavities for solidification in sand moulds than the full model, the shrinkage cavity in Fig. 6.59*f* does not penetrate as far into the casting as does the one in Fig. 6.38*f*. The difference between the predictions of the two model in this case is in favour of the M2 model since its result is closer to the experiment (Fig. 6.7).

The cooling rate and volume evolution are given in Fig. 6.60 in comparison to the M1 model results. The simplified shrinkage model gives the exact shrinkage volume while the full model underestimates it by 6.7% of the total shrinkage volume due to convergence errors.

The M2 model took 7 times less CPU time to complete the calculations and the average time step was two times larger than those in the M1 model simulation.

6.3.4 T-Shaped Casting: $d_{ch} = 15.0\ mm$

Shrinkage simulations of the T-shaped casting using the M2 model were initiated by restarting the program from the moment the filling stage ended, *i.e.* the filling stage and the starting conditions are exactly the same for both shrinkage models. However, all velocities were set to zero when the simplified model was switched on.

Simulations were made only for the pouring temperature of $T = 1013^{o}K$ ($80^{o}K$ superheat) since the full model (Section 6.2.5) and the experimental results showed little variation of the shrinkage in the T-junction with the initial superheat. The choke size was also fixed at $d_{ch} = 15.0\ mm$.

6.3.4.1 Pure Aluminium: $R_{a/b} = 1.0$

Simulation S5.1 was repeated using the M2 model (S5.11). Figs. 6.61 and 6.62 show a 2-D and a 3-D view, respectively, on the progress of solidification. The 3-D plots clearly show the shape of the liquid region and the feeding path to the T-junction. At $t = 300\ s$ the path thinned, and froze over at a later time, resulting in a shrinkage cavity in the junction. The final shape and position of shrinkage cavities shown, in Fig. 6.61*d*, are close to those given in Fig. 6.48*a*. Solidification time predicted by the M2 model is 510.0 s which is 7.4% larger than that predicted by the M1 model (475 s). The results of the two models are similar, although the M1 model gives a slightly smaller cavity in the T-junction and a deeper cavity in the feeder, the latter being closer to the experimental result.

If an instantaneous filling is assumed then the cavity in the junction is larger, positioned higher and further away from the feeder suggesting that the horizontal section of the junction freezes earlier (Fig. 6.63). This result is predicted by both models (simulations S5.10 and S5.12).

When the filling stage, which is the same for both models, is included, the gain in speed for the simplified model is by the factor of 4.3. If only the solidification stage is considered, then the increase in speed is by a factor of 6.1.

Given that the results of the two models differ little, the M2 model can be used to carry out efficiently a number simulations with varied physical and/or numerical parameters. In this particular case the following four variation fo the mould design were made in an attempt to eliminate the cavity in the T-junction:

1. S5.13: a chill block placed along the vertical section of the T-junction (Fig.

6.64*a*).

2. S5.14: an insulating pad, $10 \times 50 \times 100\ mm^3$, placed underneath the horizontal section close to the junction (Fig. 6.64*b*).
3. S5.15: an insulating pad, $10 \times 100 \times 100\ mm^3$, placed underneath the horizontal section (Fig. 6.64*c*)).
4. S5.16: an insulating pad, $10 \times 50 \times 100\ mm^3$, placed underneath the horizontal section away from the junction (Fig. 6.64*d*).

The cavity in the junction disappeared in cases 1 and 4, in agreement with the experimental result (Figs. 6.65 and 6.15). No experimental data is available for comparison for the other two cases. In case 2 the cavity became even larger since solidification in the junction progressed slower and a larger liquid volume was trapped there when the horizontal junction froze to the right of the insulator. In case 3 the cavity, though very small in size, moved away from the junction but remained in the horizontal section.

6.3.4.2 Al-4.5%Cu Alloy

In this Section simulation results of solidification and shrinkage of castings E17-21 are presented. In particular, the influence of the critical fraction of solid value $f_{s,cr}$ on the predicted defects is investigated. The filling stage was included in the calculations in the same manner as that for pure aluminium simulations. Predicted filling times for the pure metal and the alloy are similar as shown in Fig. 6.19. The difference between the two suggests that metals partially solidify and remelt *during filling*. Figure 6.66*a* shows the distribution of temperature at the end of the alloy filling for $R_{a/b} = 1.0$. The lever rule was used for all simulations below.

A. $R_{a/b} = 1.0$. For $R_{a/b} = 1.0$ there was no macroshrinkage in the T-junction for $f_{s,cr} = 0.67$ (S5.16) and for $f_{s,cr} = 1.0$ (S5.17), as shown in the experment (Figs. 6.66*b*, *c* and 6.16*a*). Figures 6.66*b*, *c* show distributed microporosity of less than 2% exists for the lower value of $f_{s,cr}$ and zero microporosity is predicted for $f_{s,cr} = 1.0$.

The predicted total solidification time for alloy is 920 *s* which is substantially longer than for the pure aluminium, 473 *s*. The experimental time is 870 *s*. Figure 6.67 shows calculated and measured cooling curves in the metal at the T-junction and in the middle of the horizontal section for $R_{a/b} = 1.0$ (simulation S5.16). Predicted cooling curves are similar to the experimental ones. The maximum deviations between the computed and measured temperatures are at the end of the simulation, $t = 1000\ s$, and are $35°C$ in the T-junction and $15°C$ in the horizontal section.

$B.\ R_{a/b} = 1.67$. Four values of $f_{s,cr}$=0.67, 0.8, 0.9 and 1.0, were tested in the simulation of the casting with $R_{a/b} = 1.67$. Figure 6.68 shows predicted porosity distributions. A macro-cavity in the T-junction can be distinguished only for the first two cases. It is interesting to note that the position of the cavity is the same in both cases, though the volume is larger for $f_{s,cr} = 0.67$ that for $f_{s,cr} = 0.8$. Experimental results show a cavity in the junction located lower than predicted in these two case (Fig. 6.16*b*).

The degree of the distributed microporosity varies in the four cases, increasing for smaller values of $f_{s,cr}$, never exceeding experimentally measured porosities of 1 to 2.5%, if $f_{s,cr} \geq 0.67$. For $f_{s,cr} = 1.0$ the casting is completely sound which is never the case in the experimental castings due to the presence of dissolved gasses in the melt. Predicted solidification times vary with the value of $f_{s,cr}$: from $t_s = 840\ s$ for $f_{s,cr} = 0.67$ to $t_s = 800.0\ s$ for $f_{s,cr} = 1.0$. The experimental solidification time is 770 s.

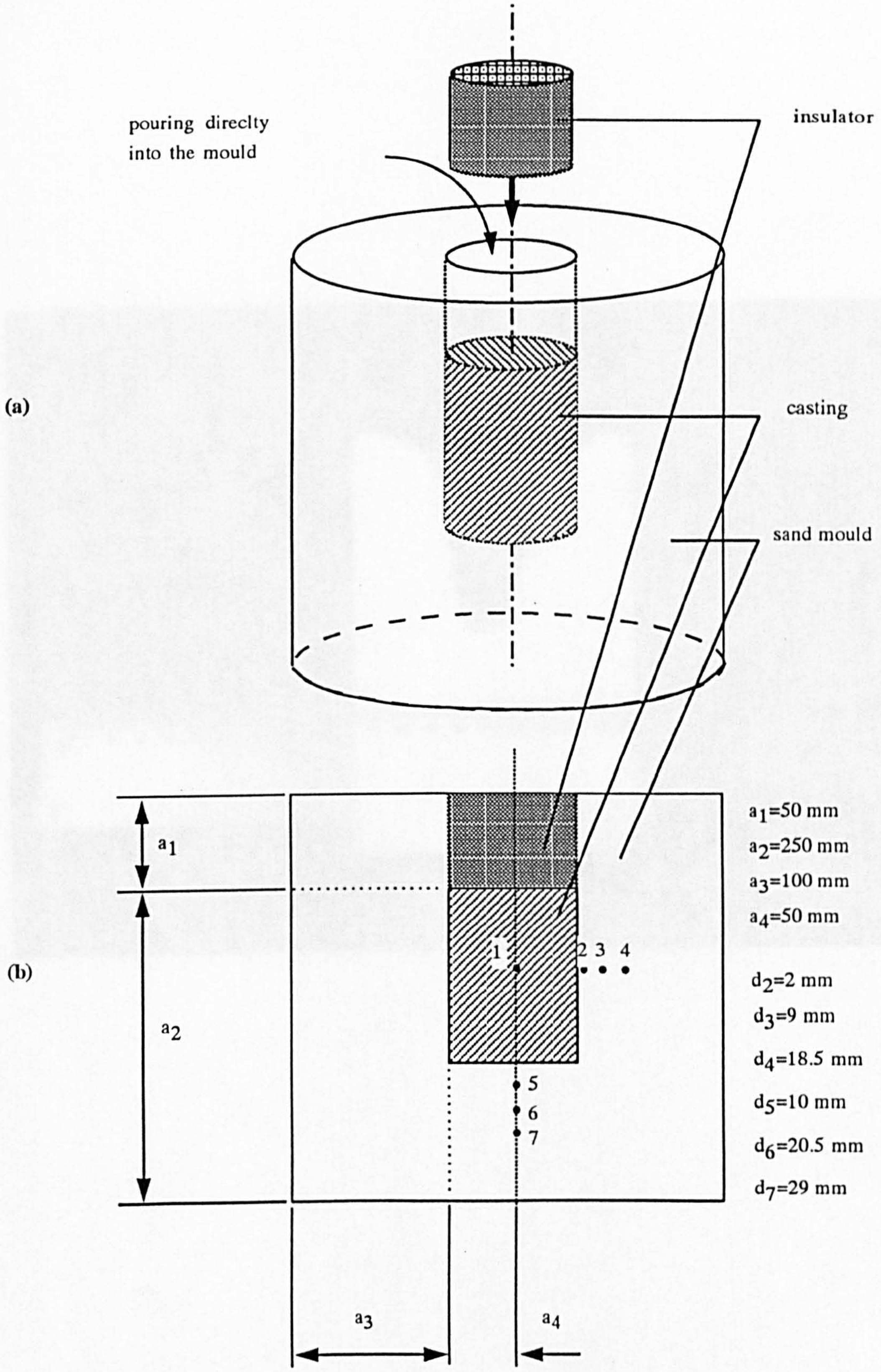


Figure 6.1. (a) The overall geometry of casting E1 with an insulator flux fitted after the pouring; (b) the dimensions of the mould a_i and distances d_i from the thermocouple positions to the nearest interface.

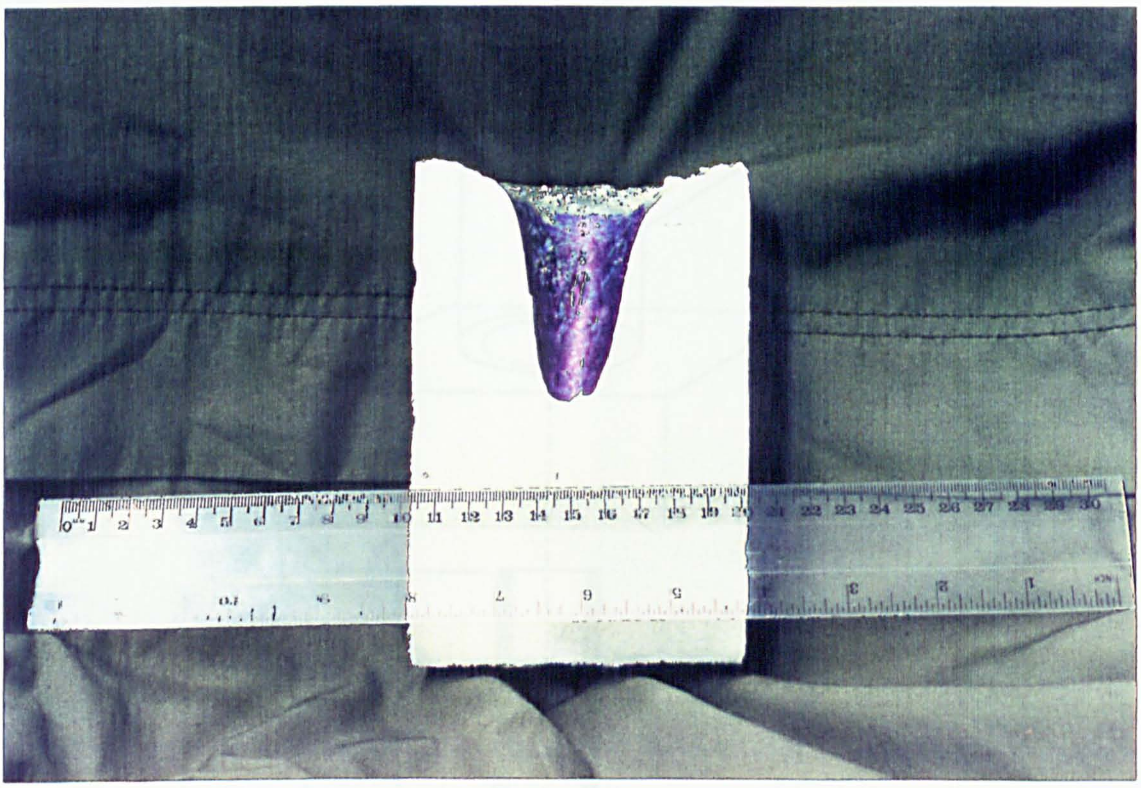


Figure 6.3. Primary shrinkage cavity in aluminium-sand casting E8.

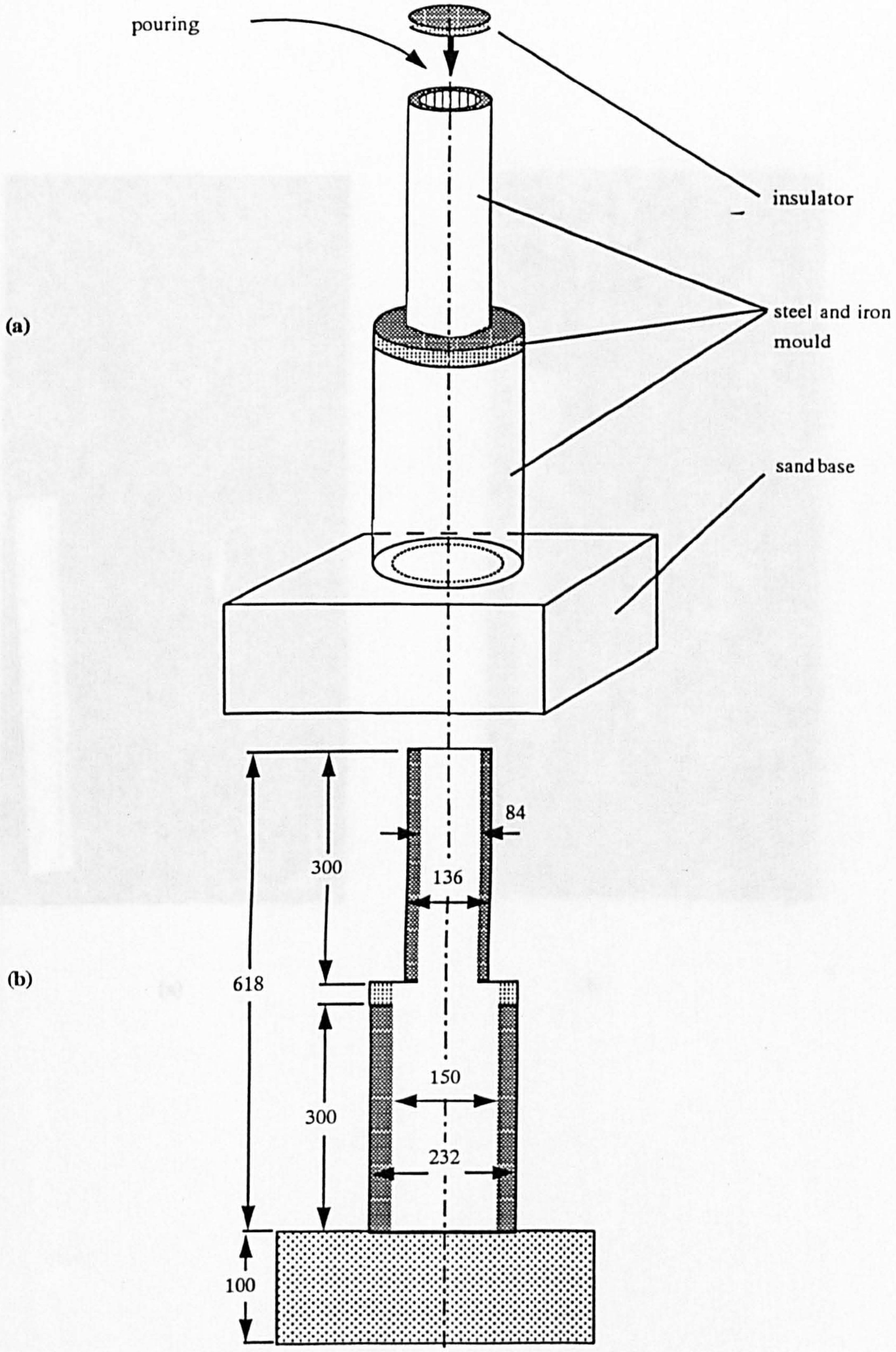
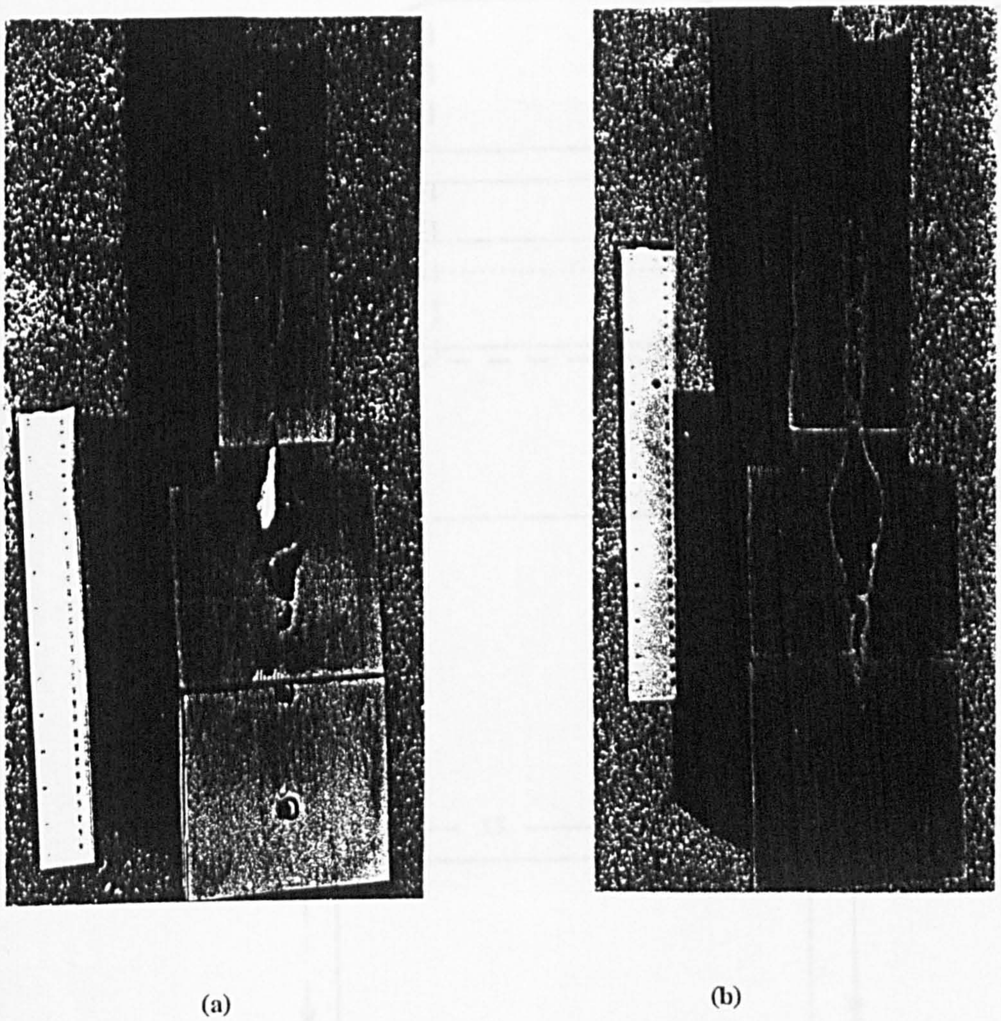


Figure 6.4. Overal geometry (a) and dimensions (b) of the chill mould cylindrical casting in *mm* (castings E2 and E3).



(a)

(b)

Figure 6.5. Shrinkage cavities in (a) iron-chill casting E2 and (b) 0.6%C, 1.1%Mn, 0.2%Si steel casting E3.

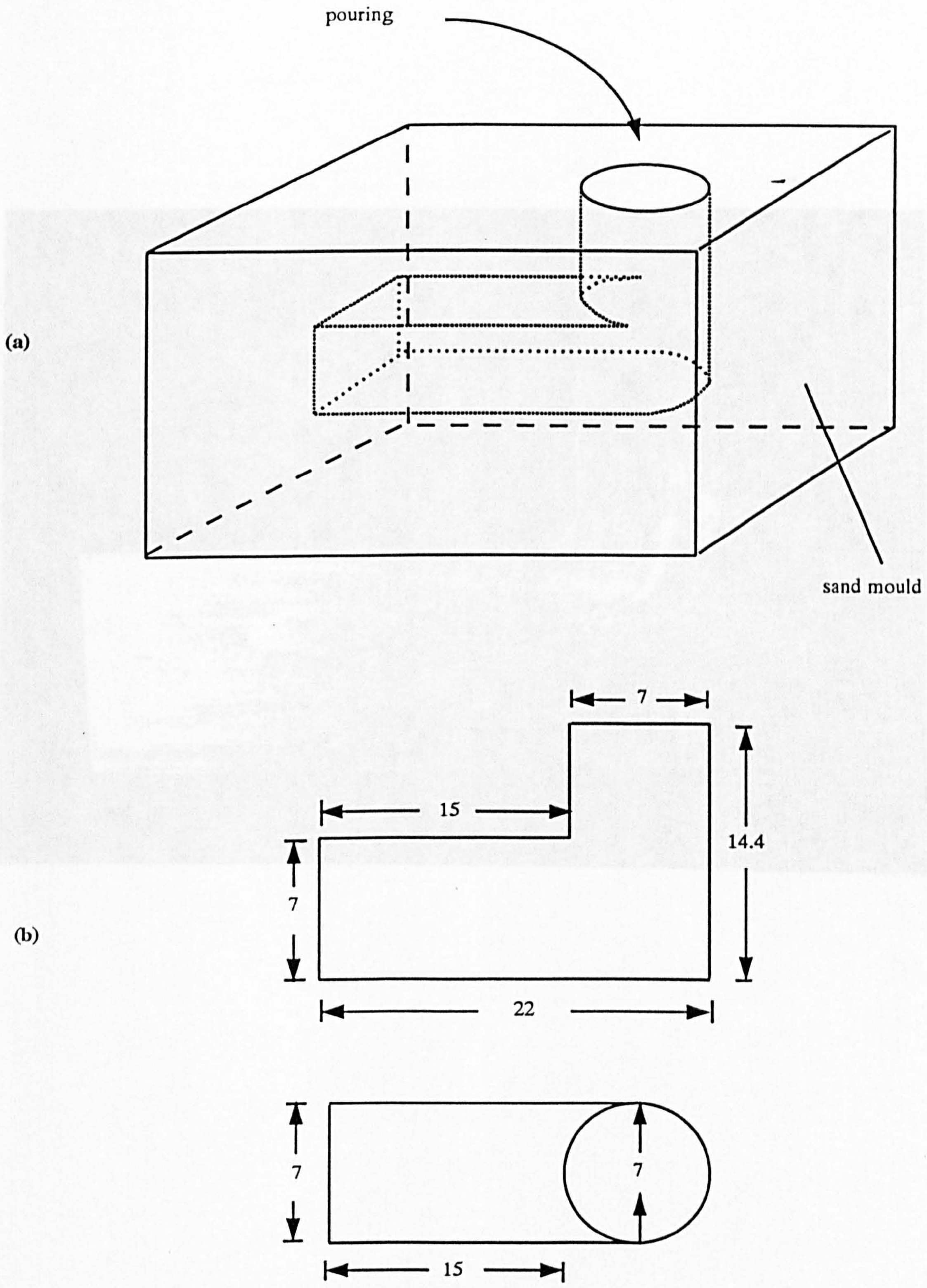


Figure 6.6. The overall geometry (a) and dimensions in *mm* (b) of the boot-shaped casting (E4).

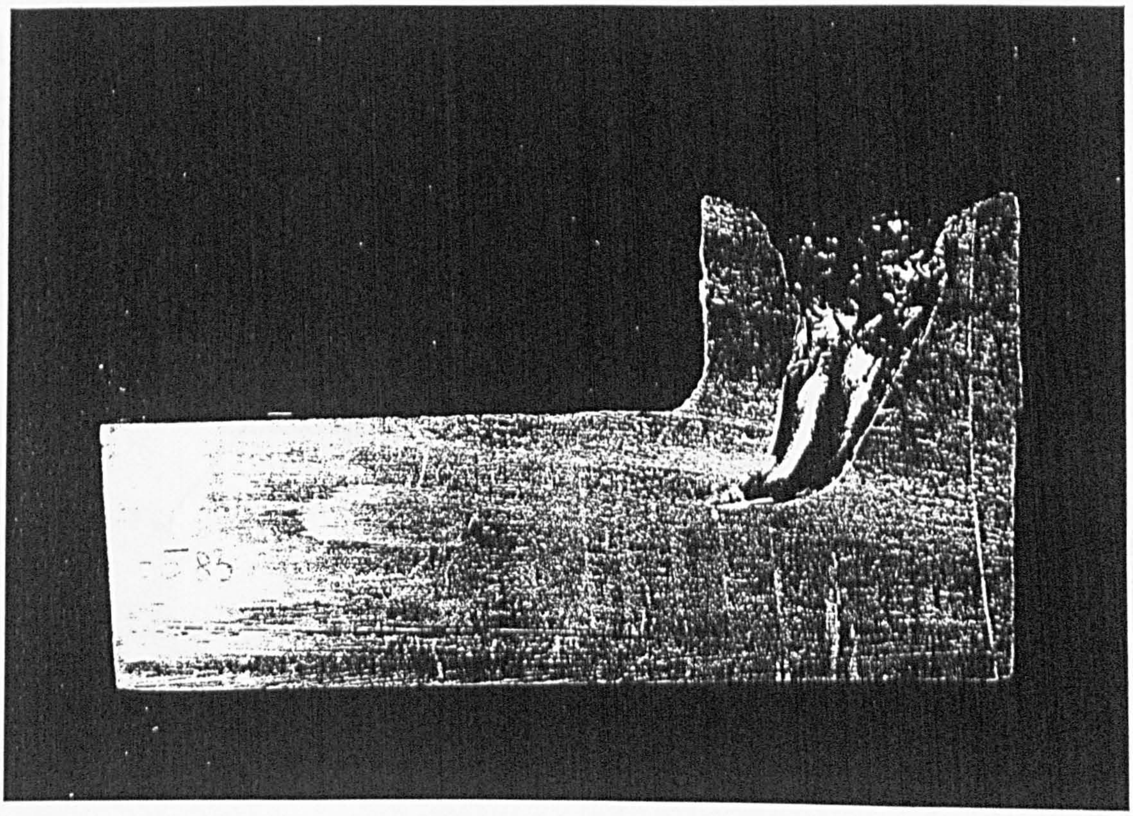
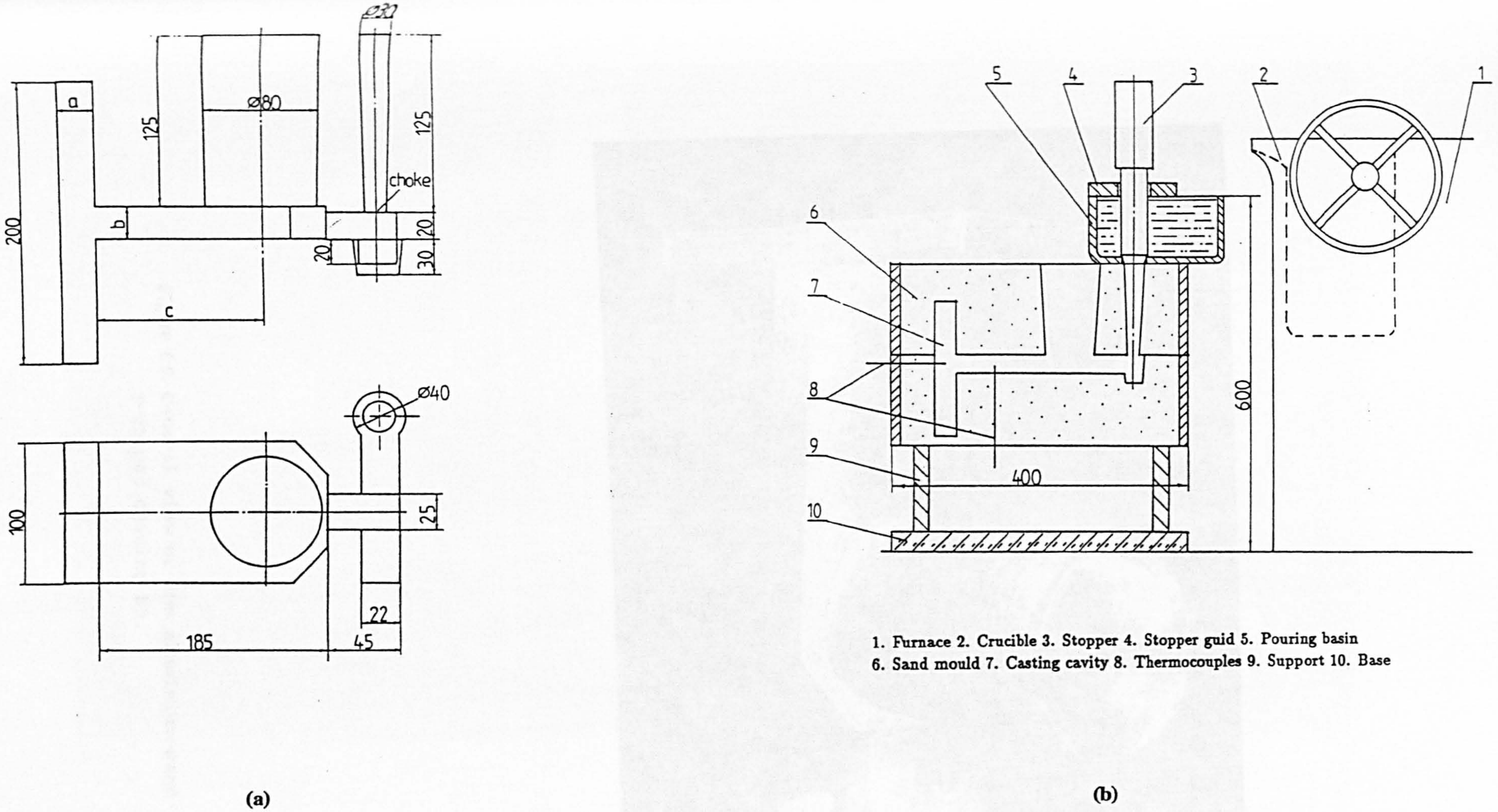


Figure 6.7. Shrinkage cavity in aluminium-sand casting E4.



1. Furnace 2. Crucible 3. Stopper 4. Stopper guid 5. Pouring basin
6. Sand mould 7. Casting cavity 8. Thermocouples 9. Support 10. Base

(a)

(b)

Figure 6.8. (a) Geometry and dimensions and (b) pouring method for the T-shaped casting.

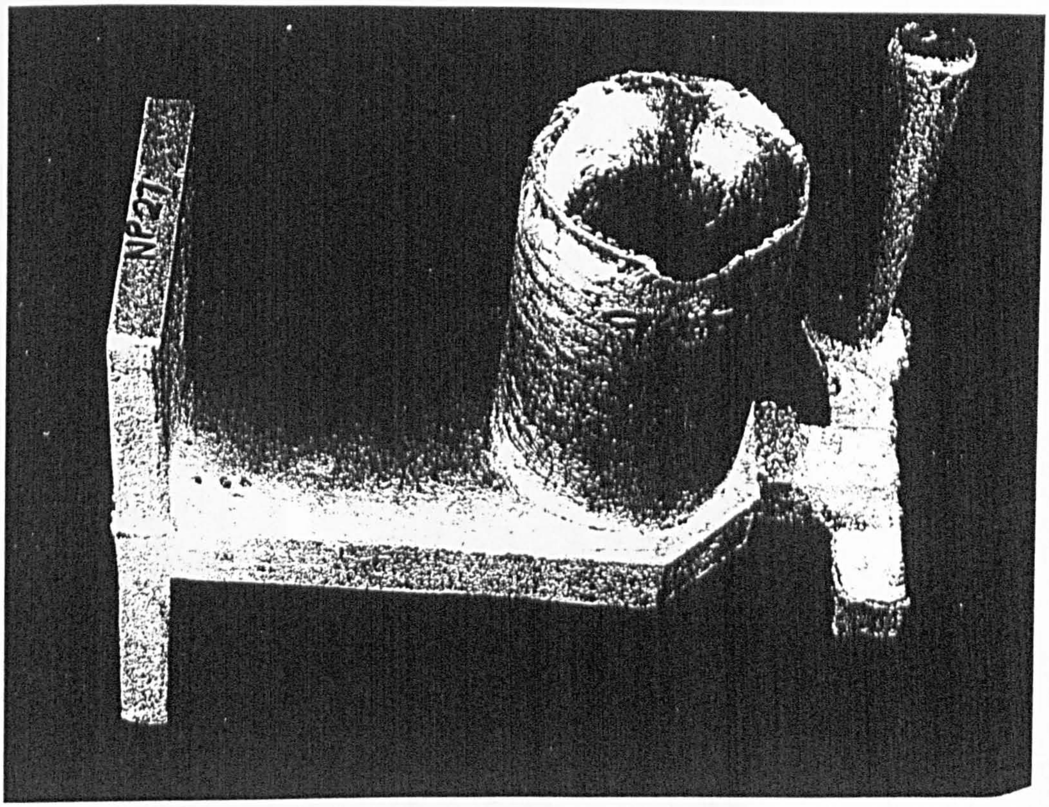


Figure 6.9. General view of the aluminium-sand T-shaped casting E5.

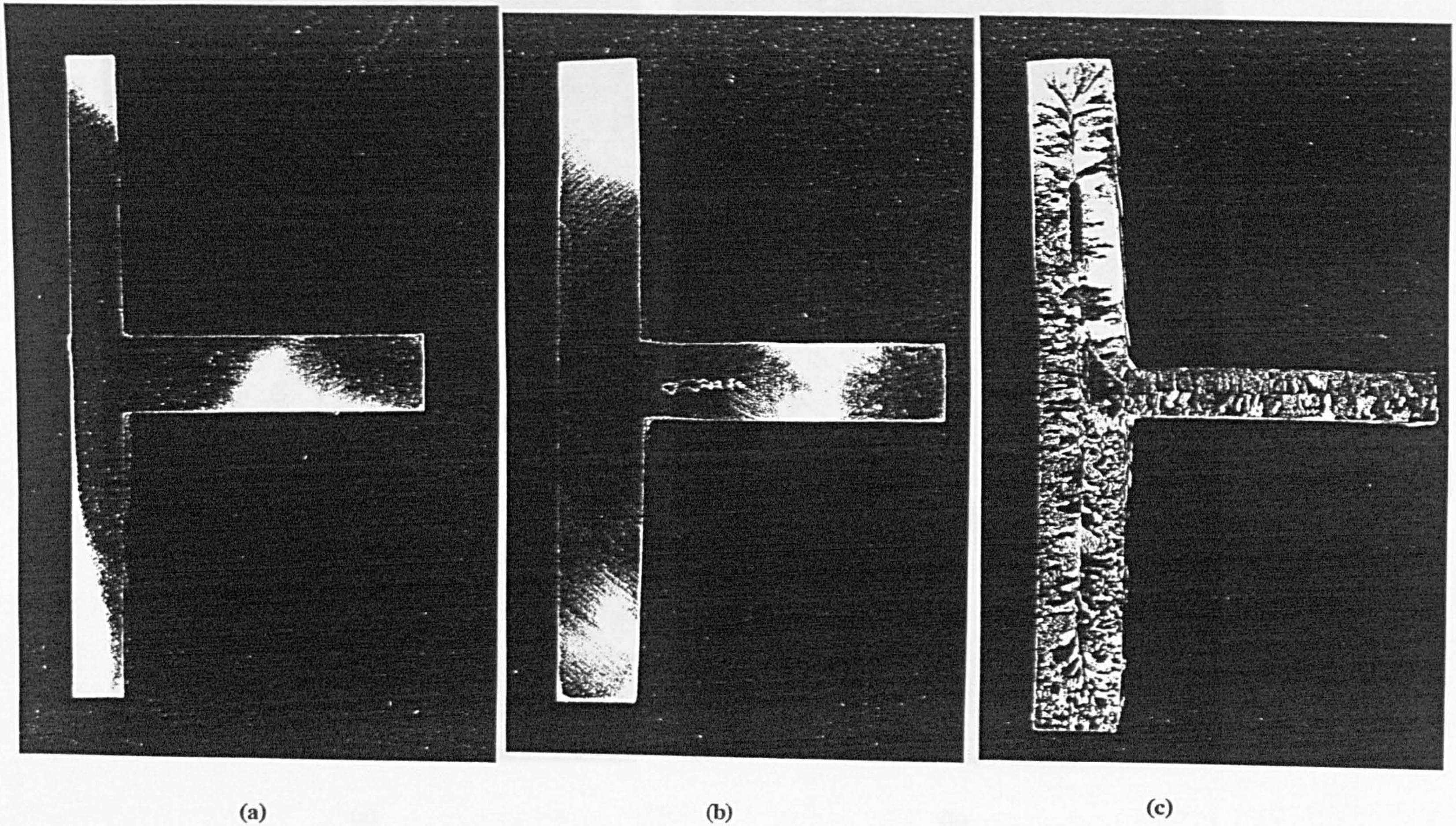


Figure 6.10. Centre-plane sections of aluminium castings (a) E5 (R=0.6), (b) E8 (R=1.0) and (c) E14 (R=1.67).

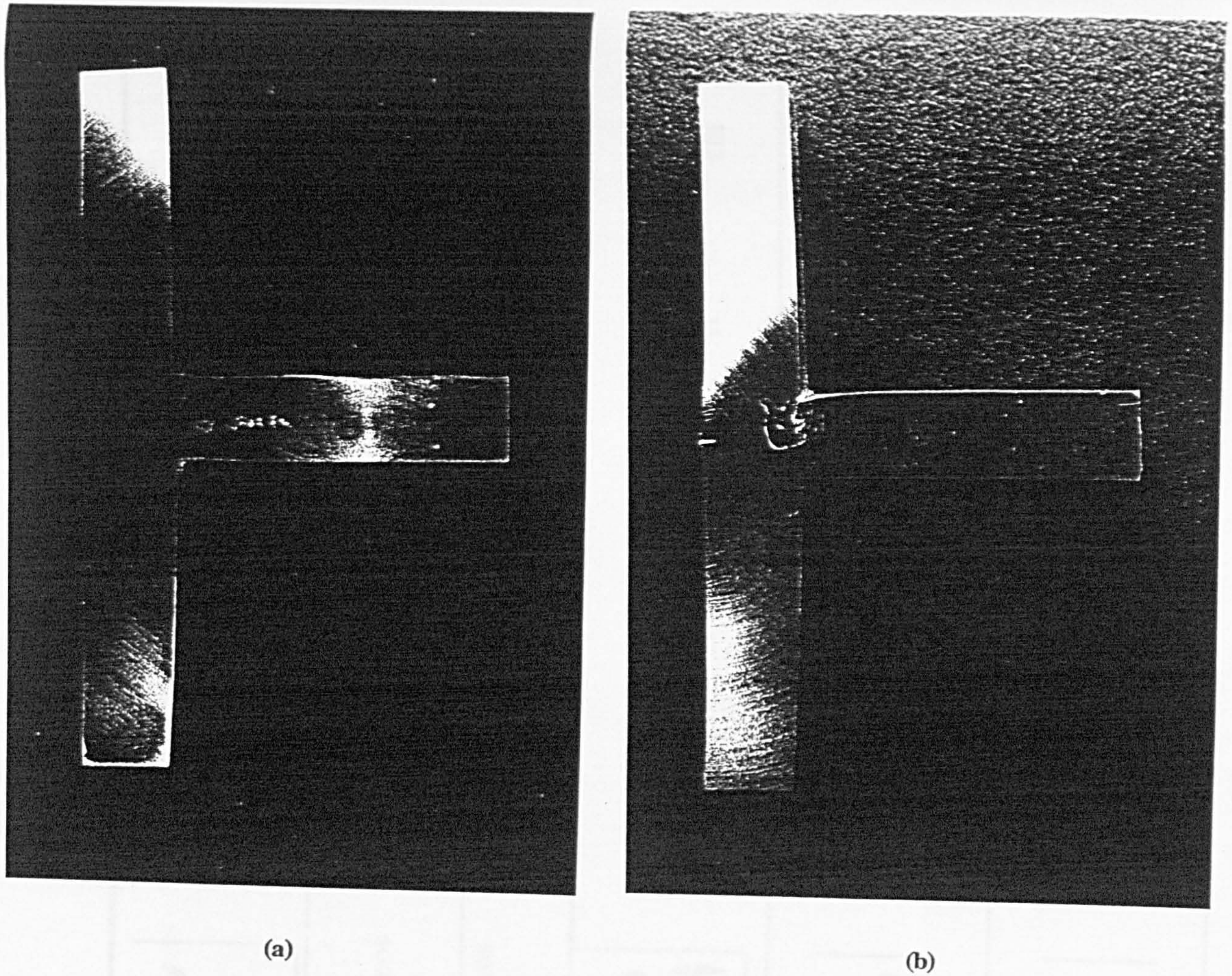


Figure 6.11. Variation of the cavity shape and position in the T-junction for castings with R=1.0: (a) E8 and (b) E12.

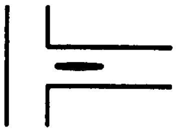
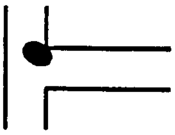
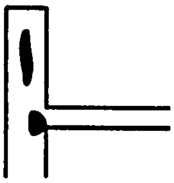
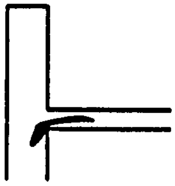
Experiment No.	Choke size d_{ch}, (mm)	Initial superheat ^{o}K	Geometry R	Filling time, s	Macro–shrinkage in the T–junction
pure aluminium					
E5	15	963	0.6	5.8	sound
E6	15	1013	0.6	6.0	
E7	8	1013	1.0	20.5	
E8	15	1013	1.0	6.8	
E9	8	963	1.0	21.0	
E10	15	963	1.0	6.8	
E11	22	1013	1.0	5.0	
E12	15	1013	1.0	6.8	
E13	15	963	1.67	7.2	
E14	15	1013	1.67	7.0	
Al–4.5%Cu alloy					
E17	15	963	0.6	5.8	sound
E18	15	963	1.0	6.9	sound
E19	15	1013	1.0	6.8	
E20	15	963	1.67	7.1	
E21	15	1013	1.67	7.0	

Figure 6.12. Summary of the T-shaped casting experiments in all-sand mould.

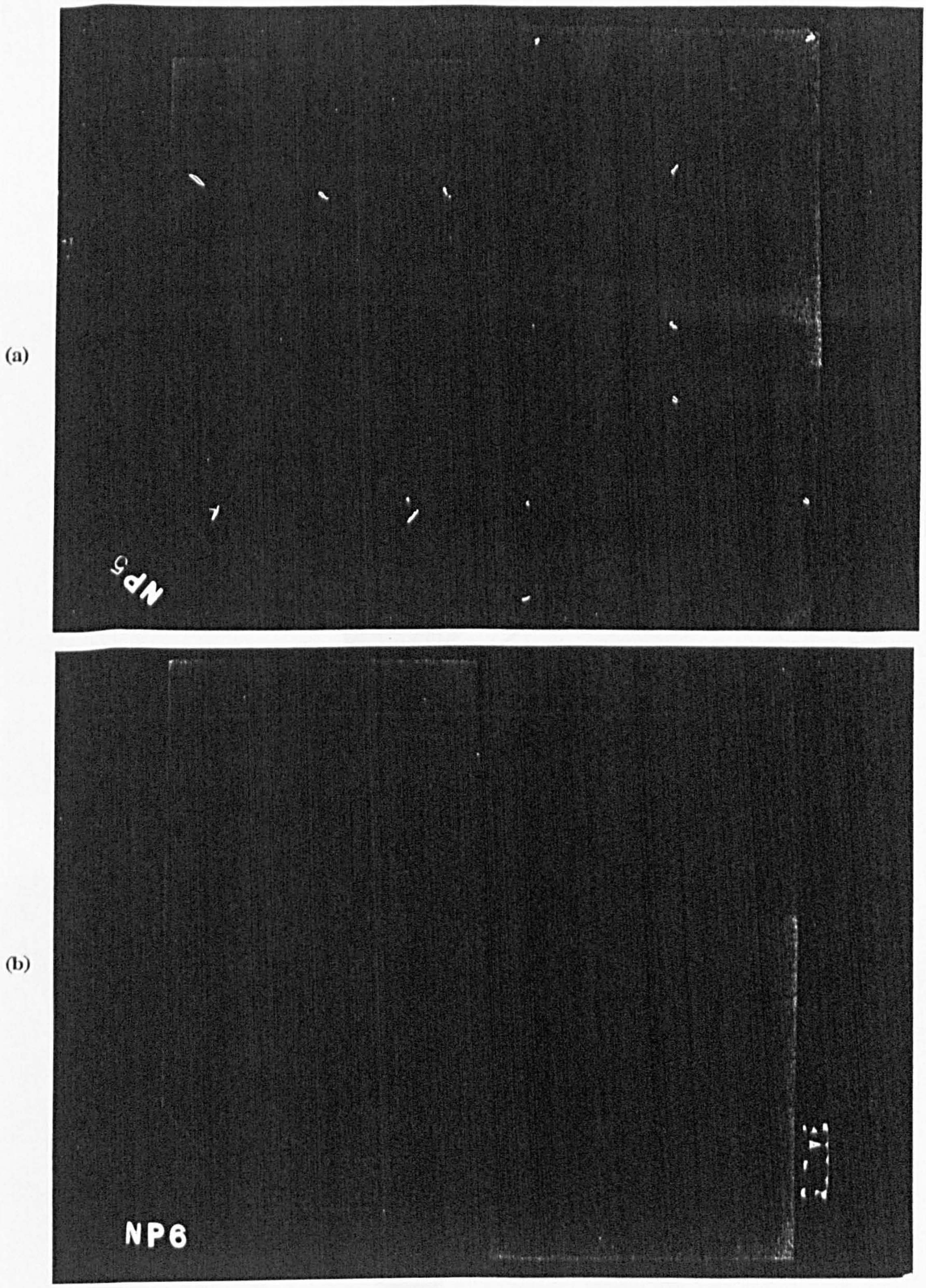


Figure 6.13. X-ray photographs of the vertical and horizontal sections of castings (a) E8 and (b) E12 (see figure 6.11).

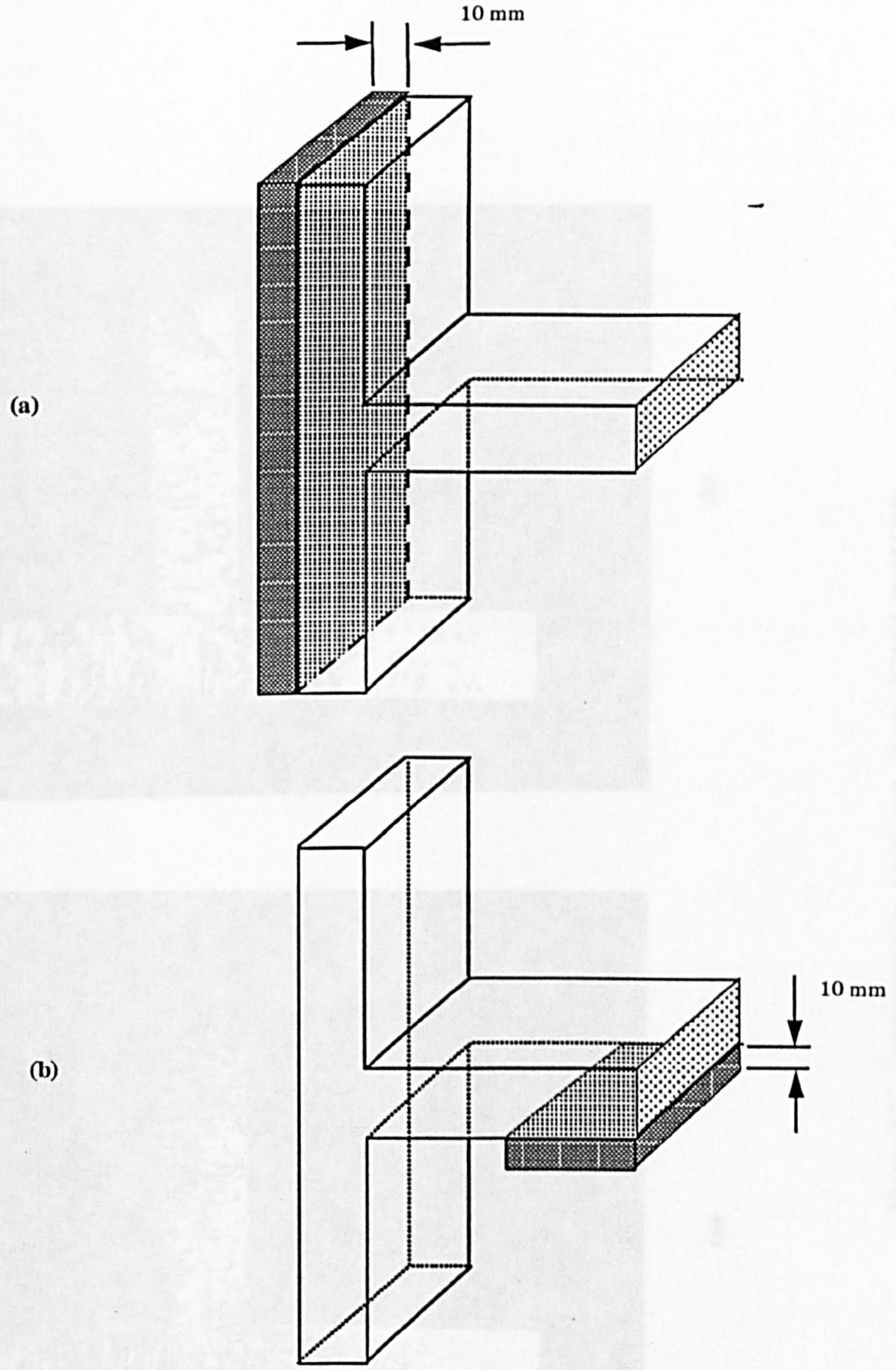
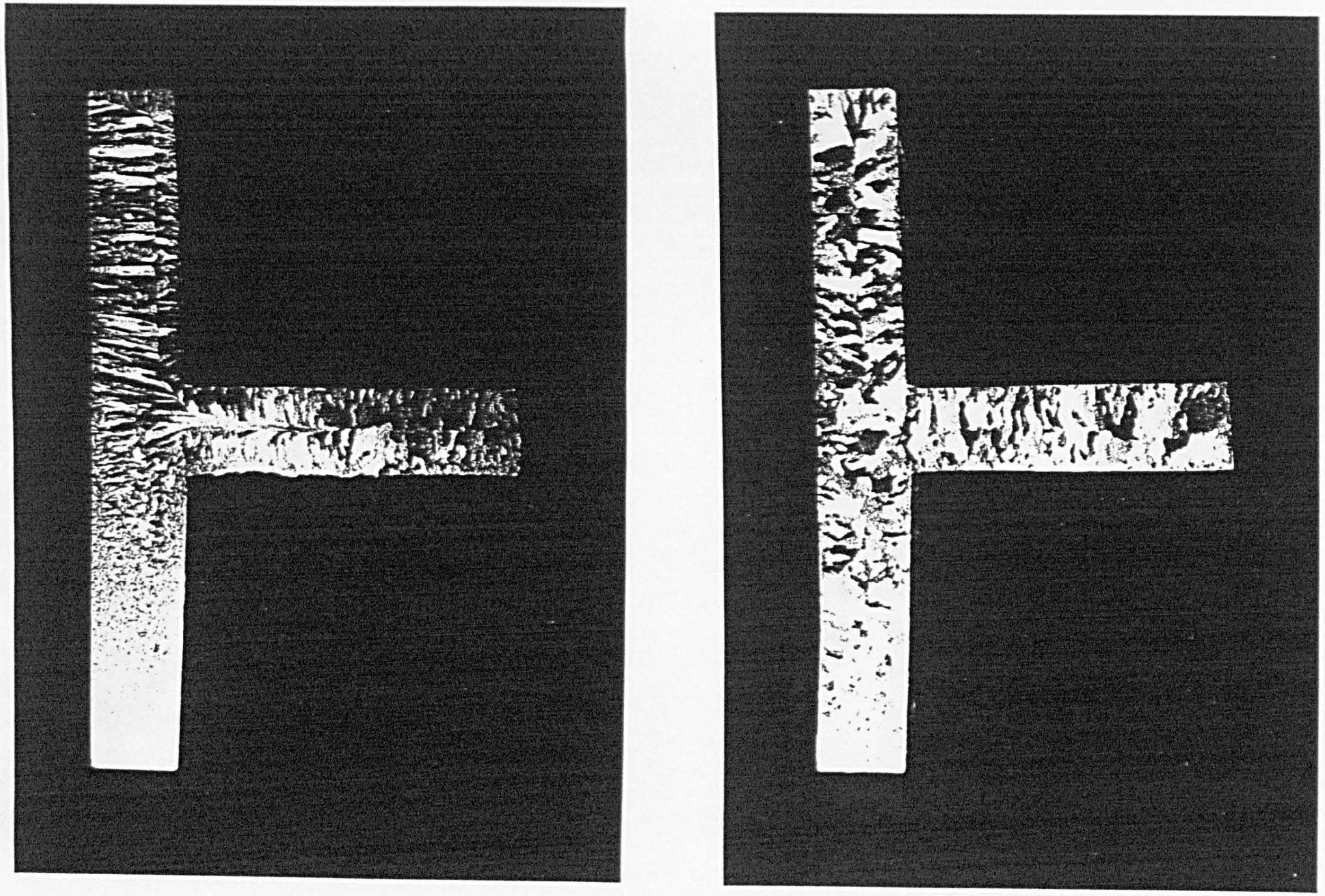


Figure 6.14. The chill block (a) and the insulating pad (b) experimental setup for the T-shaped casting.



(a)

(b)

Figure 6.15. Centre-plane sections of T-shaped castings (a) with the chill and (b) with the insulating pad (R=1.0).

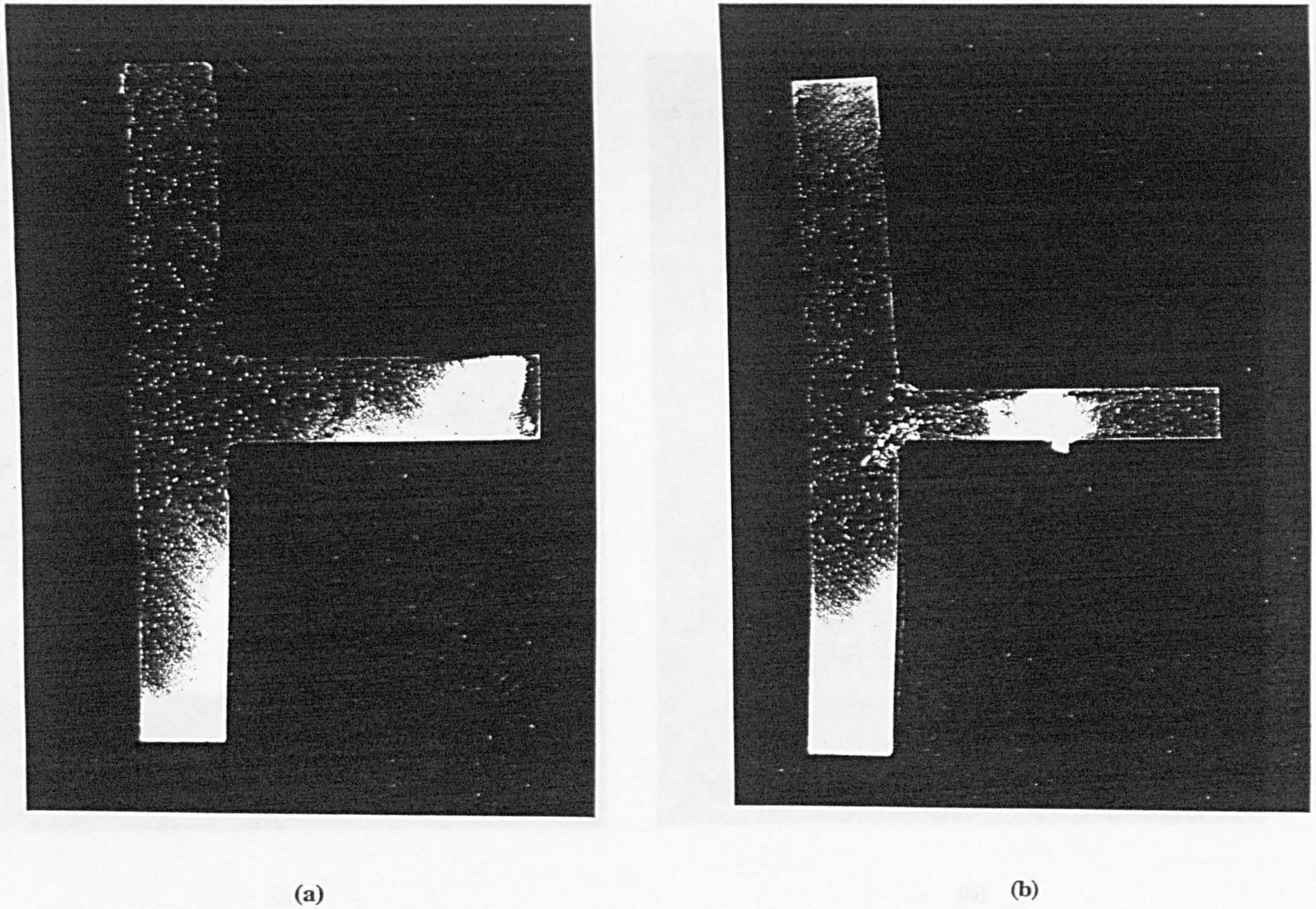


Figure 6.16. Centre-plane sections of Al-4.5%Cu castings (a) E19 (R=1.0) and (b) E21 (R=1.67).

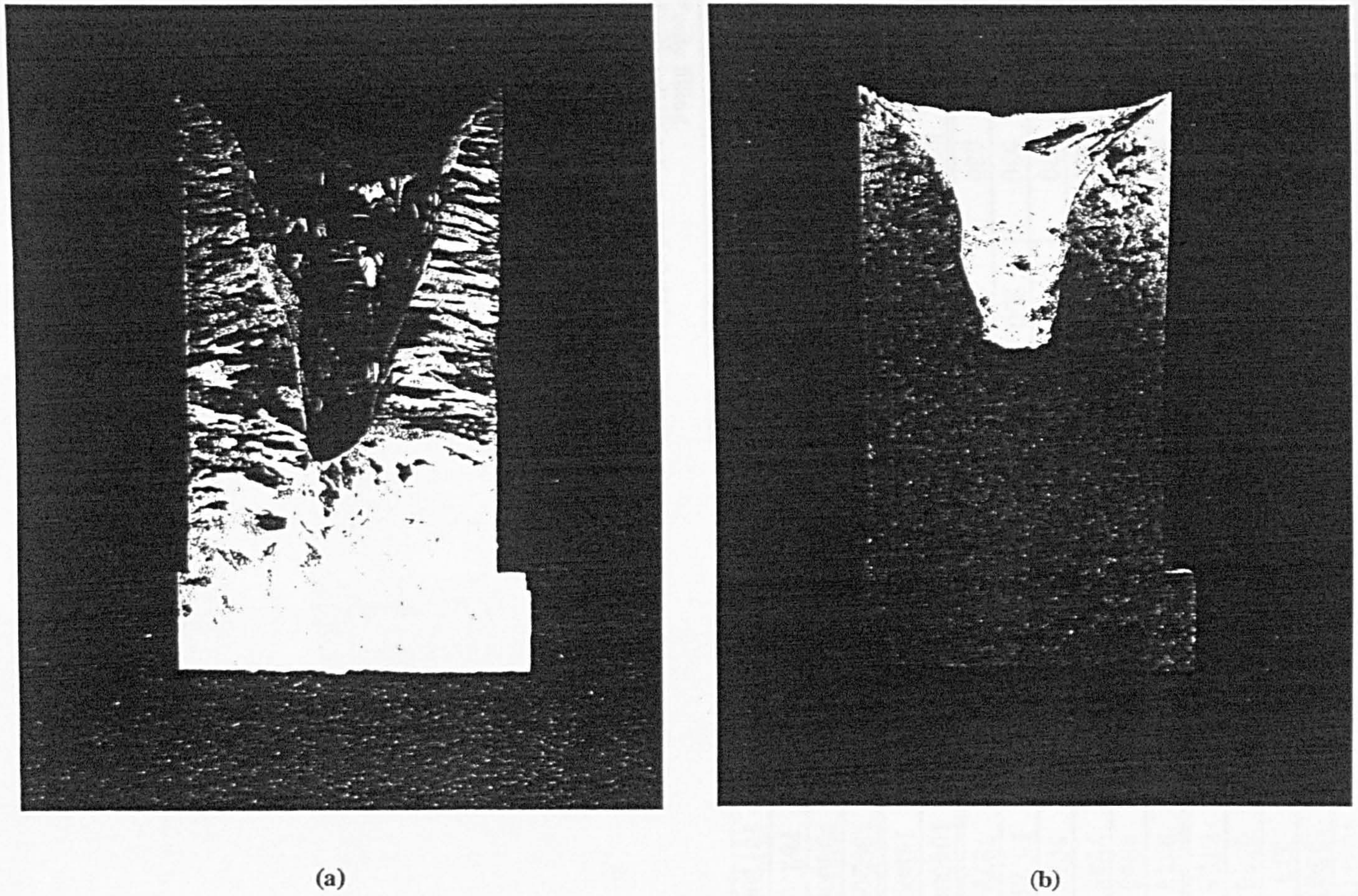


Figure 6.17. Typical primary cavities in the feeders of T-shaped castings: (a) aluminium and (b) Al-4.5%Cu.

Sim. No.	number of mesh cells	problem time, s	ϵ s^{-1}	average iter. no	number of time steps	total CPU time, s	CPU per time step, s	freezing time, s
S1.1	150	50	0.001	20	1820	135	0.074	43.9
S1.2	81	22	10^{-6}	10	1063	36.25	0.034	21.3
S2.1	90	1200	0.0001	3	1402	42.5	0.03	1135
S2.2	360	1200	0.0001	4	8966	808.8	0.09	1013
S2.3	1000	1200	0.0001	4	28545	7557	0.265	975
S2.4	3240	1200	0.0001	15	82503	103710	1.26	957.4
S2.5	627	1200	0.0001	3	89905	15800	0.176	986
S2.12	360	1200	-	-	6930	313	0.045	1011
S2.24	360	1200	0.01	4	63714	6856	0.11	1050
S2.34	360	1200	0.01	5	71901	6725	0.093	1033
S3.1	530	200	0.001	45	19454	9813	0.5	177
S3.5	530	200	0.001	50	10755	4900	0.46	191.2
S3.6	530	200	0.001	40	14810	7785	0.53	193.5
S3.7	530	200	0.001	40	9386	5013	0.53	194.6
S3.11	530	200	-	-	200	18.23	0.09	177
S3.21	530	200	0.001	35	21850	7972	0.36	173.2
S4.1	6300	600	0.0001	75	36315	104000	2.86	559
S4.11	6300	600	-	-	25800	14800	0.57	561
S5.1[a]	36244	7.5	0.005	40	7877	139200	17.7	-
S5.1[b]	36244	500	0.00025	120	43820	793000	18.1	473
S5.11[b]	36244	500	-	-	20835	78270	3.76	478.1
S5.17[b]	36244	1000	-	-	35801	137200	3.83	920

[a] only filling

[b] only solidification and shrinkage

Figure 6.19. Summary of calculation efficiency information for a selection of simulations.

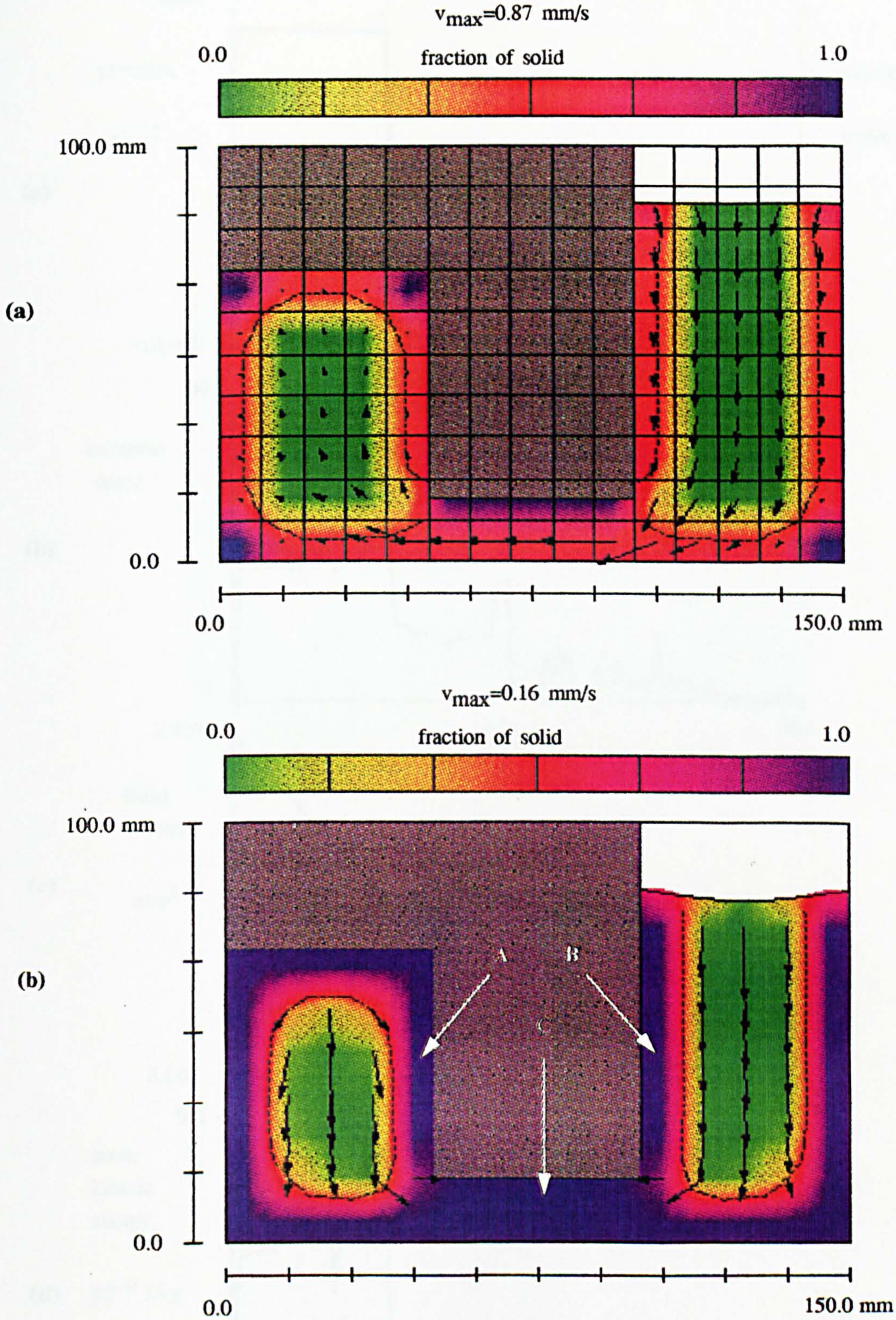


Figure 6.20. 2-D simulation S1.1. (a) Feeding occurs from section B to A (mesh is also shown); (b) sections A and B a fed independently when the narrow passage C solidifies. Colours denote solid fraction distribution.

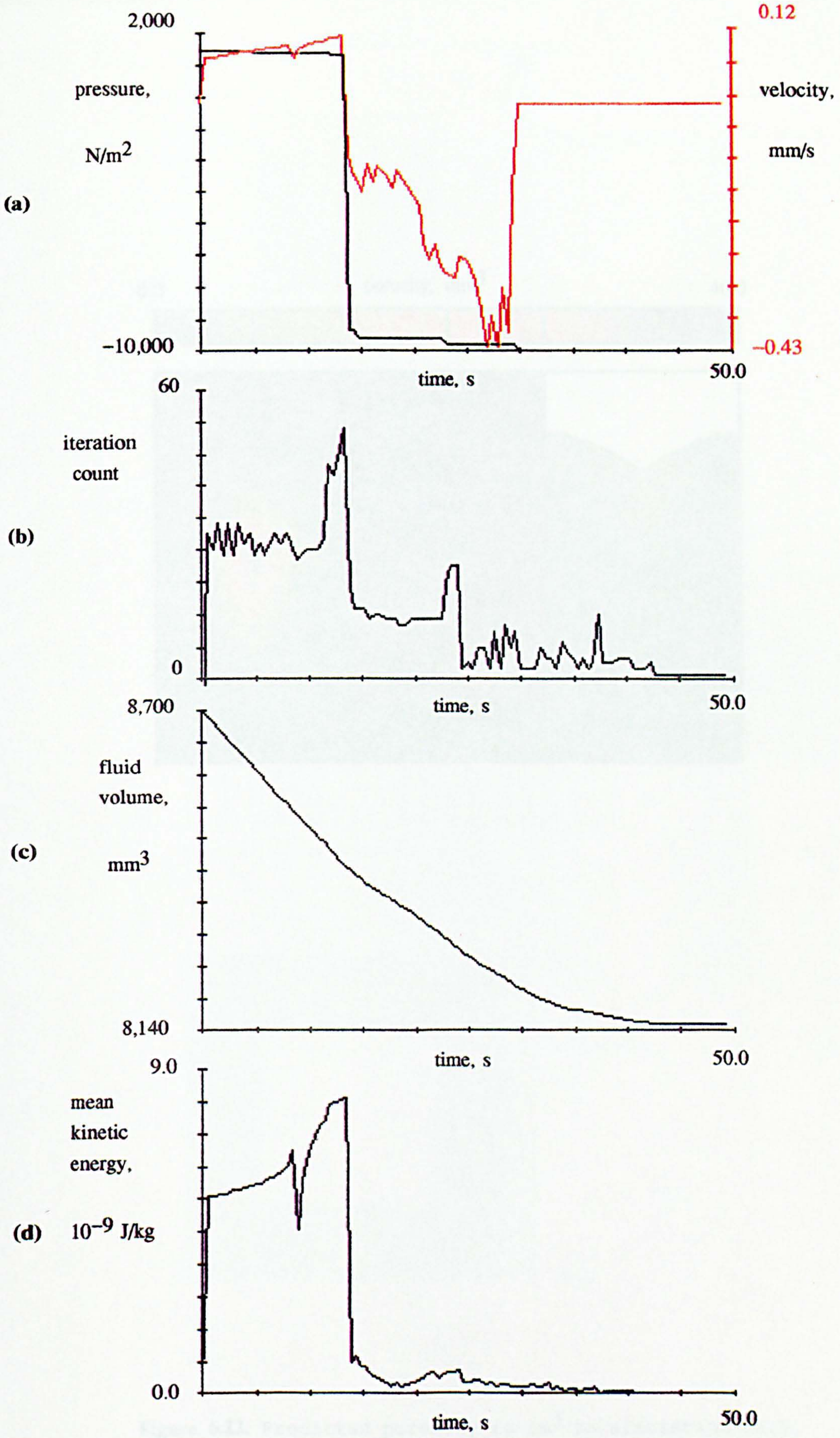


Figure 6.21. (a) Pressure and vertical velocity (red line) histories in the middle of section A in figure 6.20; (b) iteration count, (c) metal volume and (d) mean kinetic energy evolutions for simulation S1.1.

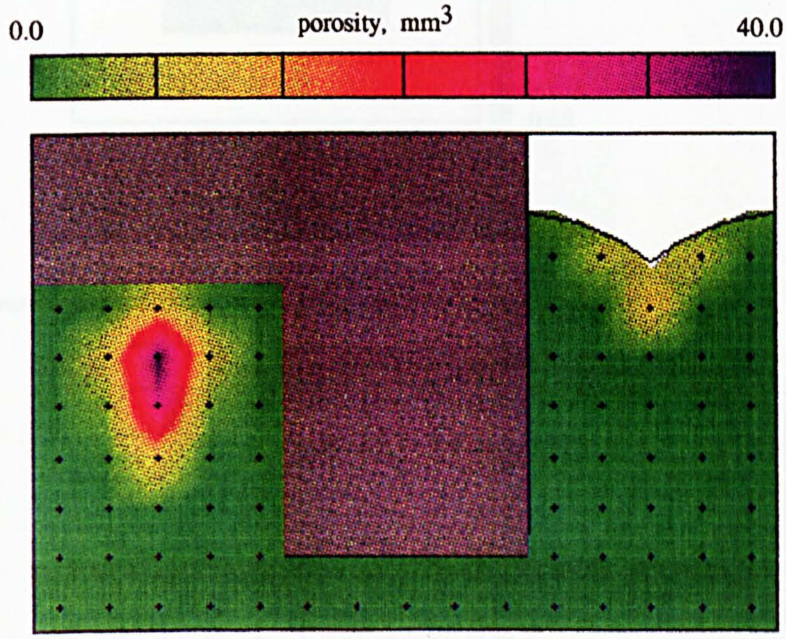


Figure 6.22. Predicted porosity in cm^3 in simulation S1.1.

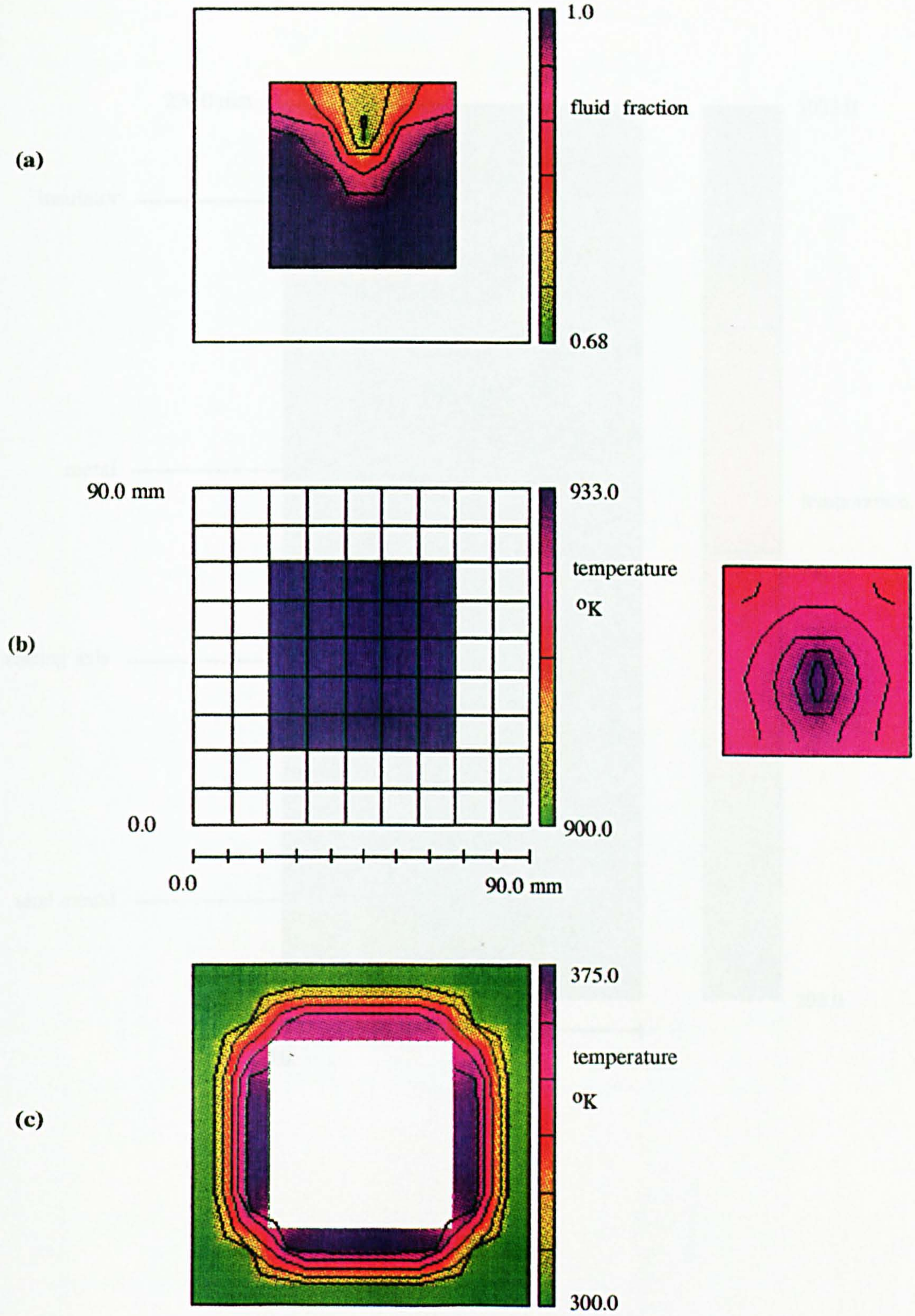


Figure 6.23. Simulation S1.2. (a) Predicted porosity; (b) Temperature contours at the beginning and at the end of solidification; (c) Mould temperatures at the end of solidification.

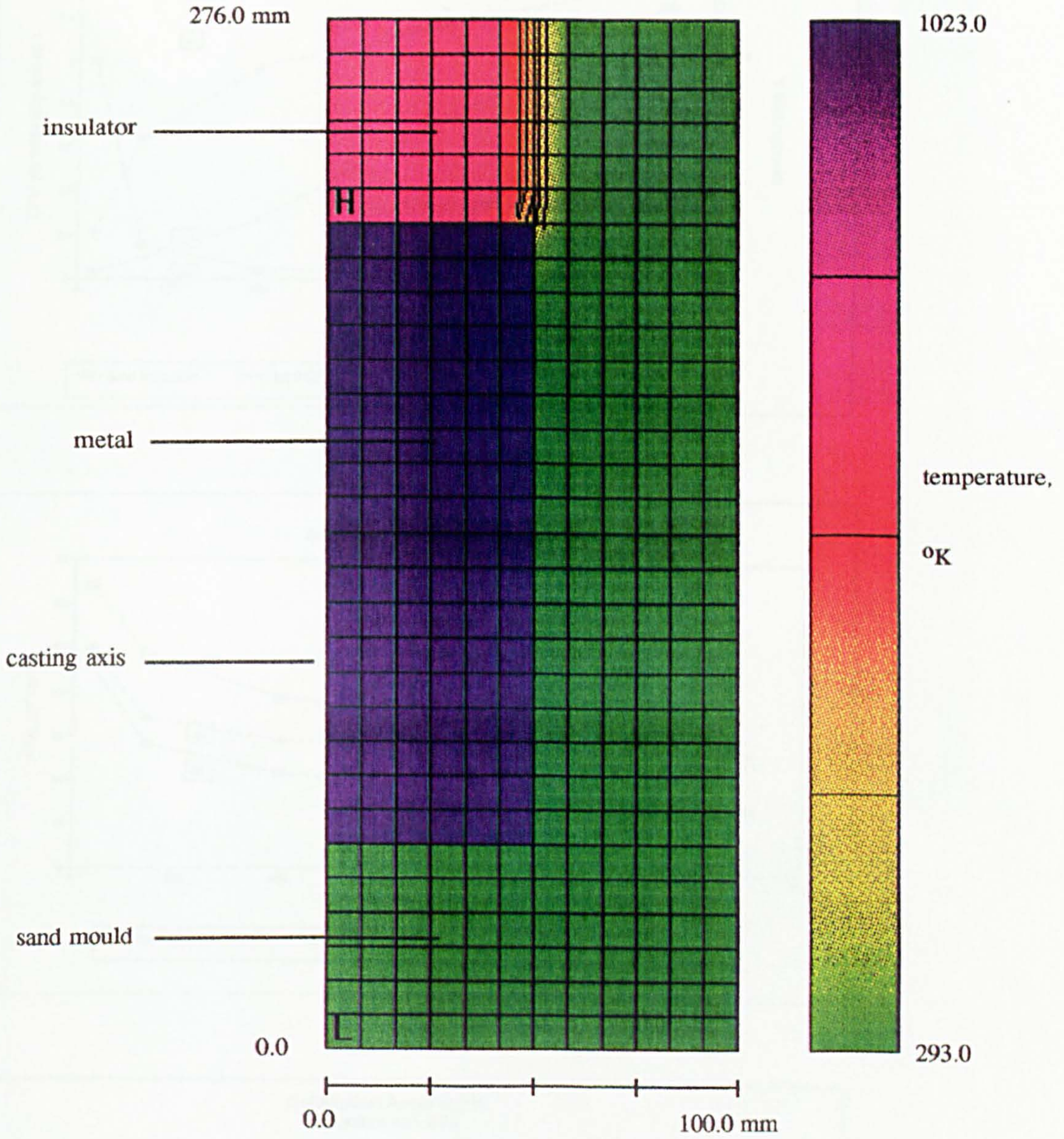


Figure 6.24. The initial conditions and mesh for simulation S2.2.

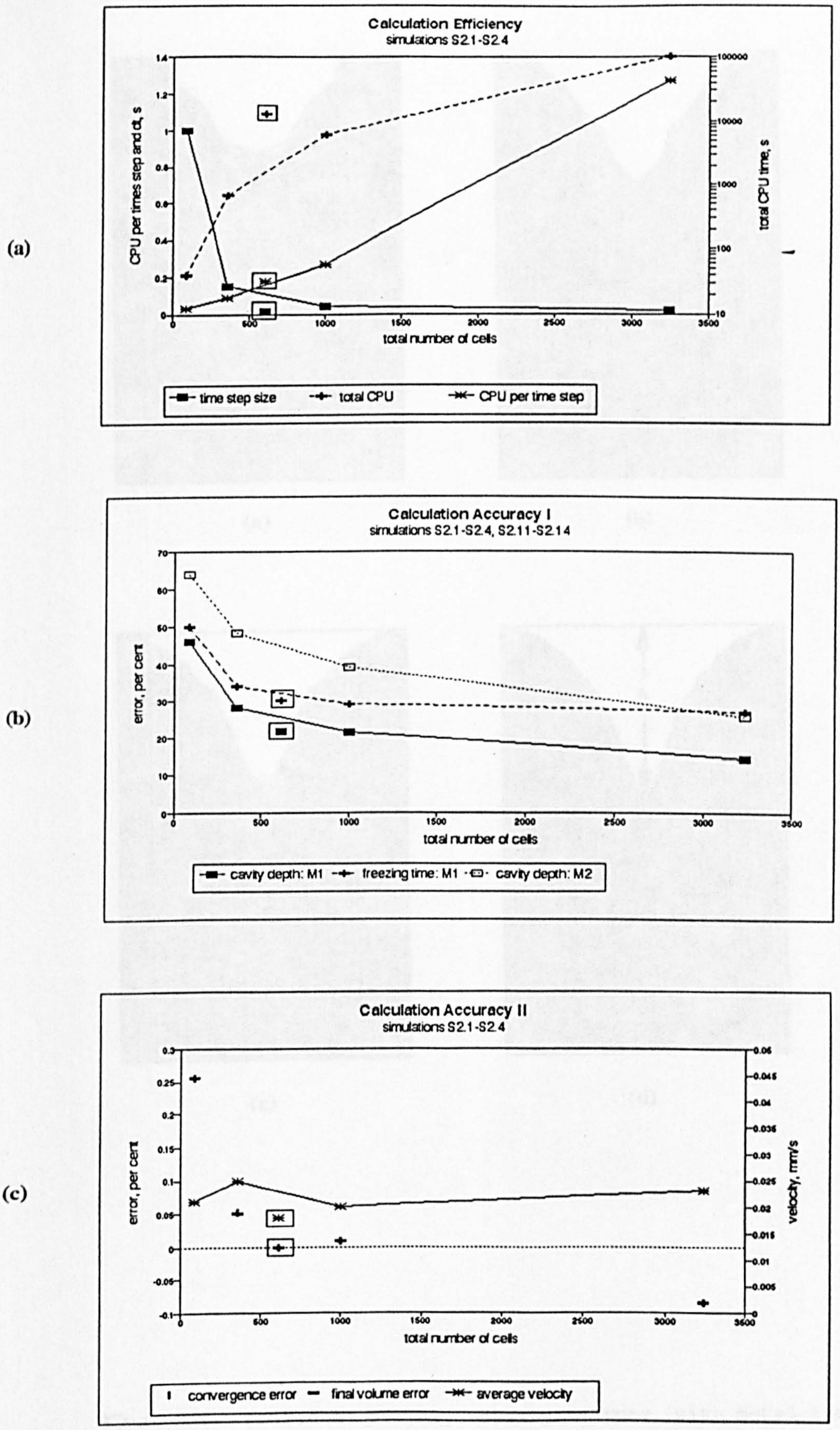


Figure 6.25. Influence of mesh resolution on shrinkage simulation results: (a) time step size, total CPU time and CPU time per time step for simulations S2.1-S2.4; (b) cavity depths predicted by the full shrinkage model (S2.1-S2.4) and the simplified model (s2.1-S2.4) and predicted freezing times (s2.1-S2.4); (c) volumetric error due to convergence and total volumetric error in per cent of the shrinkage volume and average feeding flow velocity (Simulations S2.1-S2.4). Framed markers shows results of simulation S2.5.

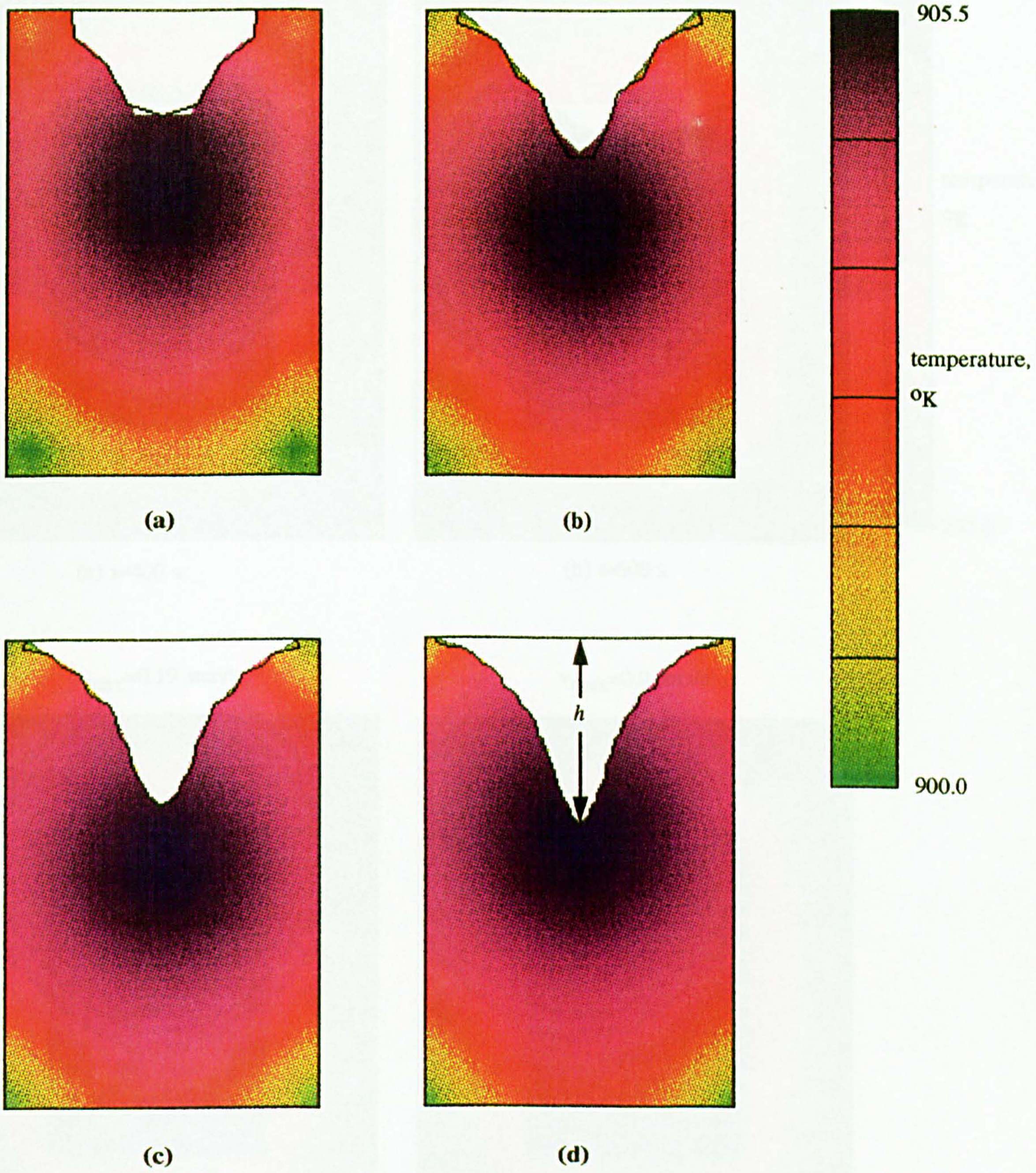
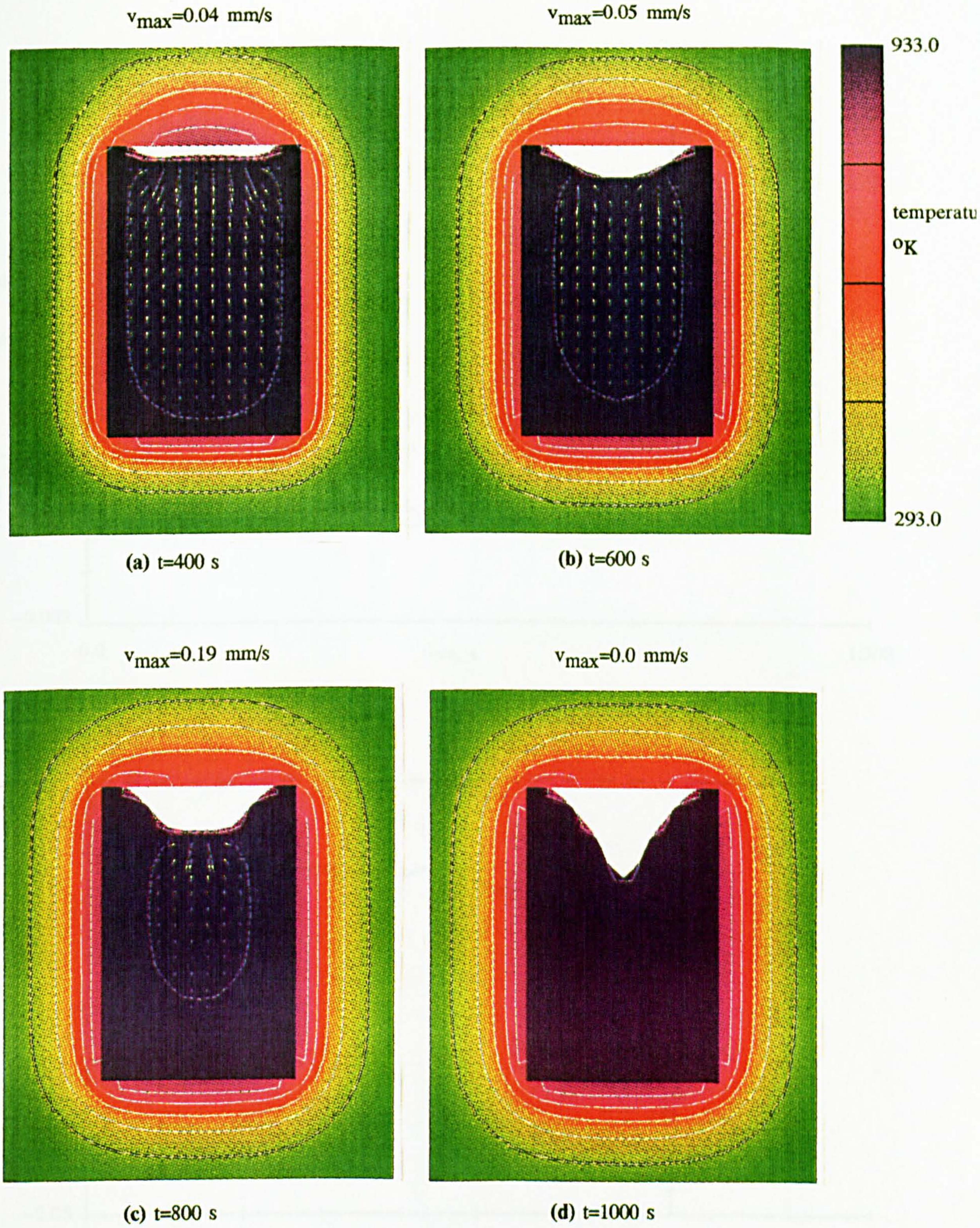


Figure 6.26. Predicted primary cavity shapes with metal temperature distributions for simulations S2.1-4: (a) N=90; (b) N=360; (c) N=1000 and (d) N=3240 (N - the total number of cells).



(a) t=400 s

(b) t=600 s

(c) t=800 s

(d) t=1000 s

Figure 6.27. Simulation S2.2: the progress of solidification and shrinkage. Colours denote temperature distribution in metal and mould.

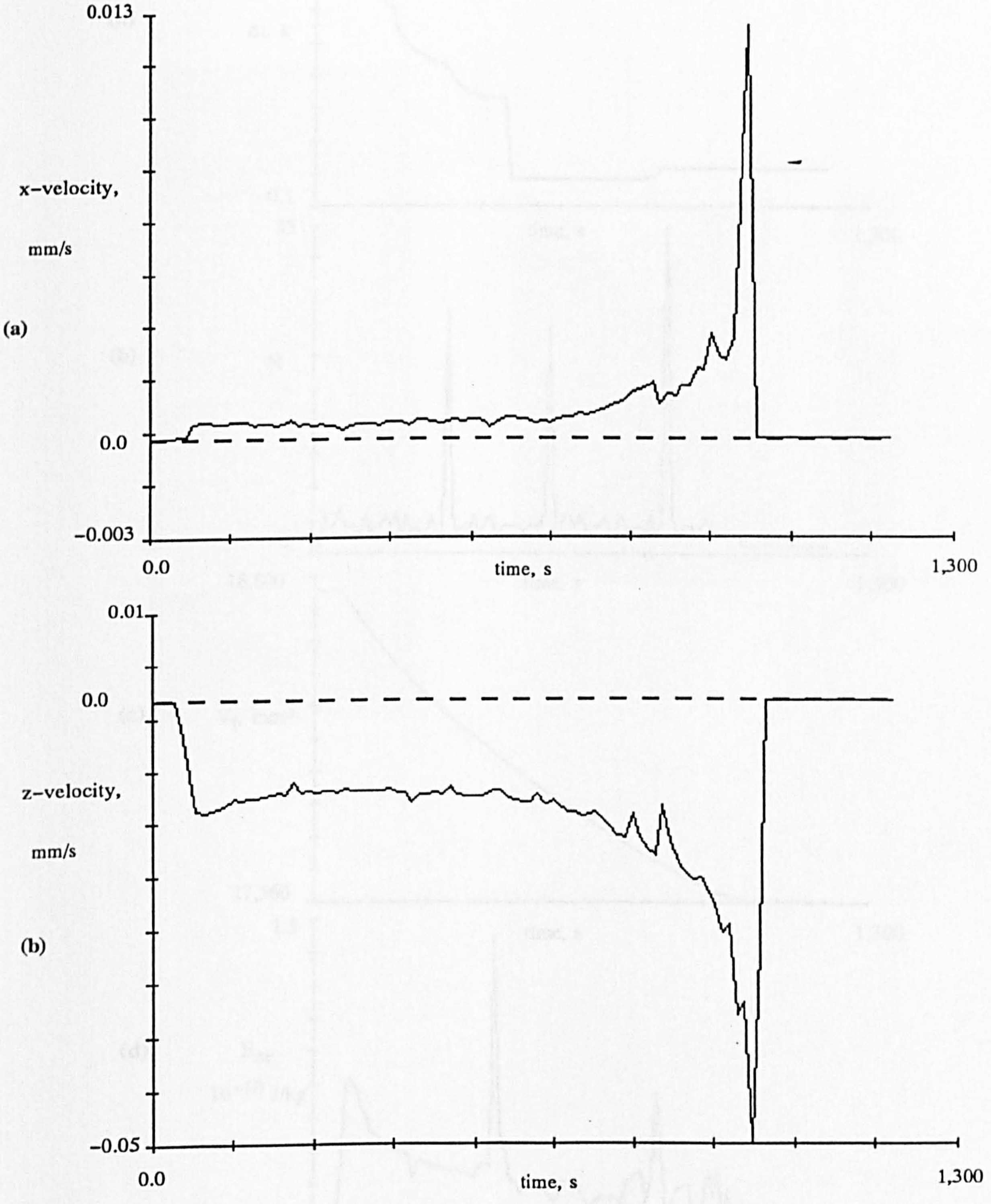


Figure 6.28. Simulation S2.2: horizontal (a) and vertical (b) velocities in the centre of the casting.

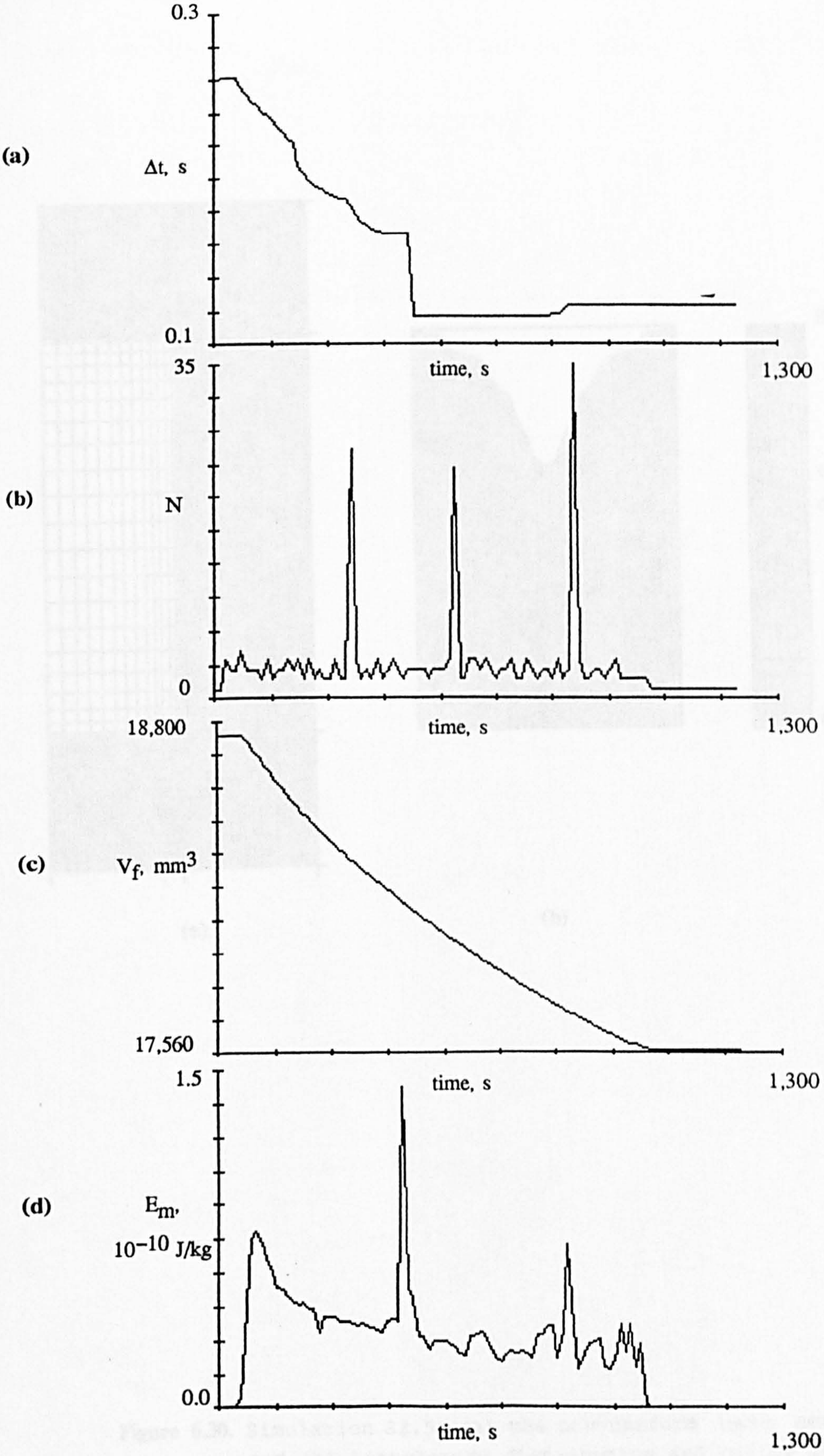


Figure 6.29. Simulation S2.2: (a) time step size, (b) iteration count, (c) metal volume and (c) metal mean kinetic energy.

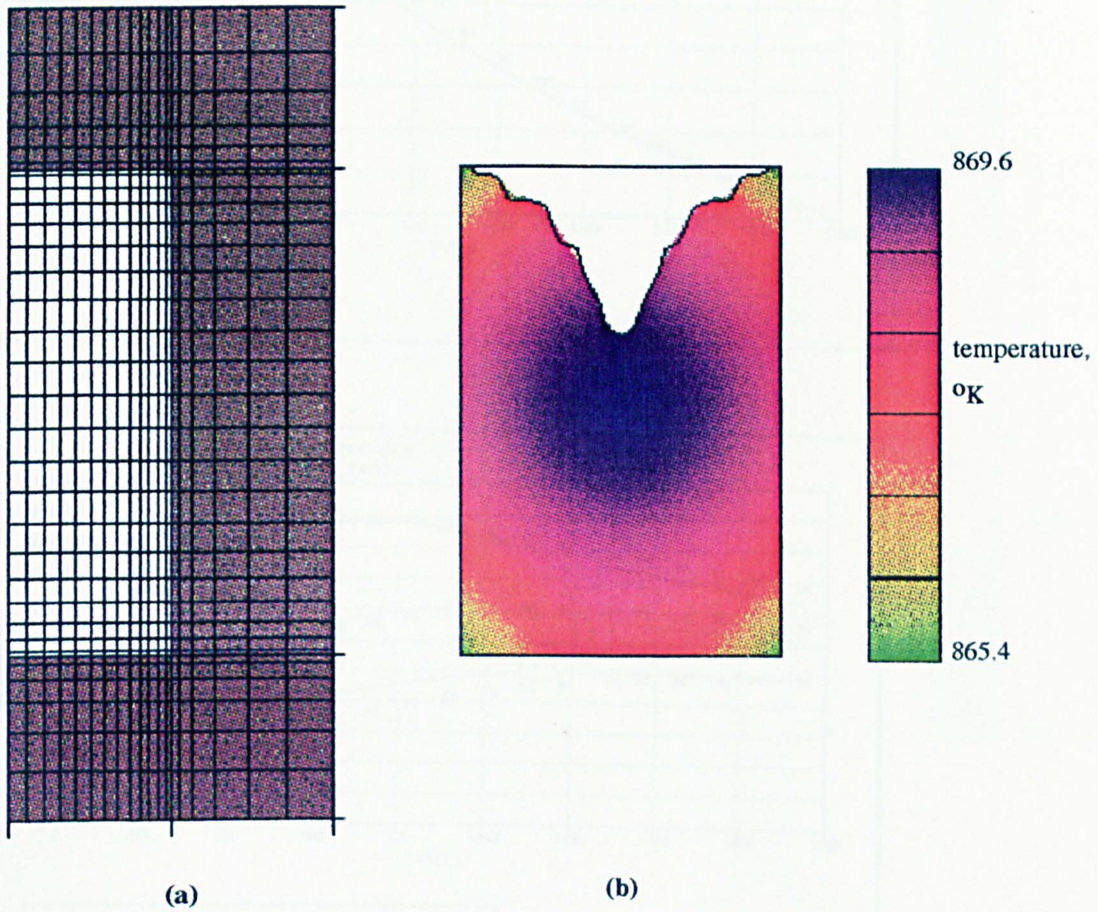


Figure 6.30. Simulation S2.5: (a) the non-uniform mesh setup and (b) temperature distribution and cavity shape at t=1,200.0 s.

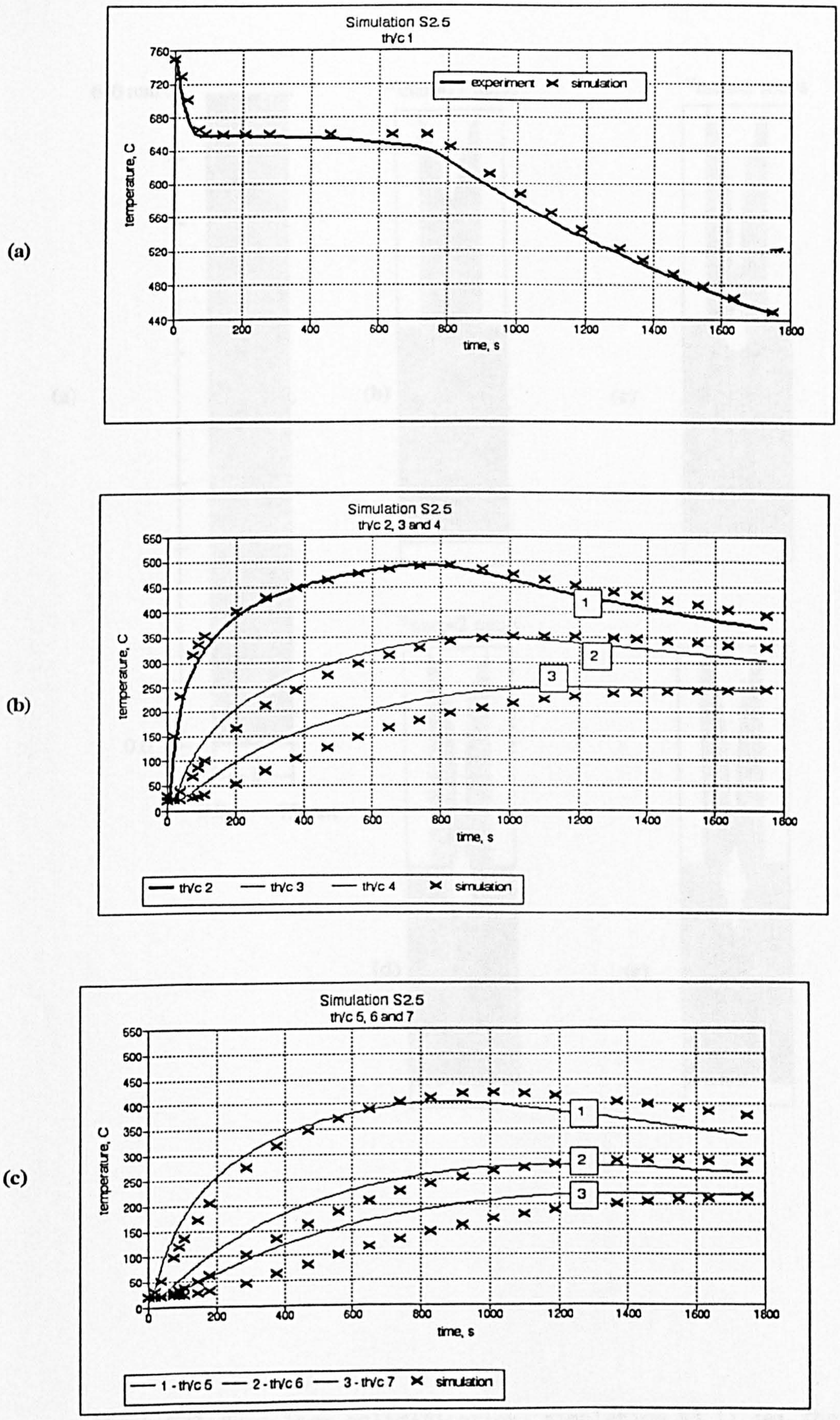


Figure 6.31. Temperatures histories predicted in simulation S5.1 (crosses) and measured (solid line) in (a) casting centre, (b) three location in the mould wall and (c) three locations at the mould bottom as shown in figure 6.1,(b).

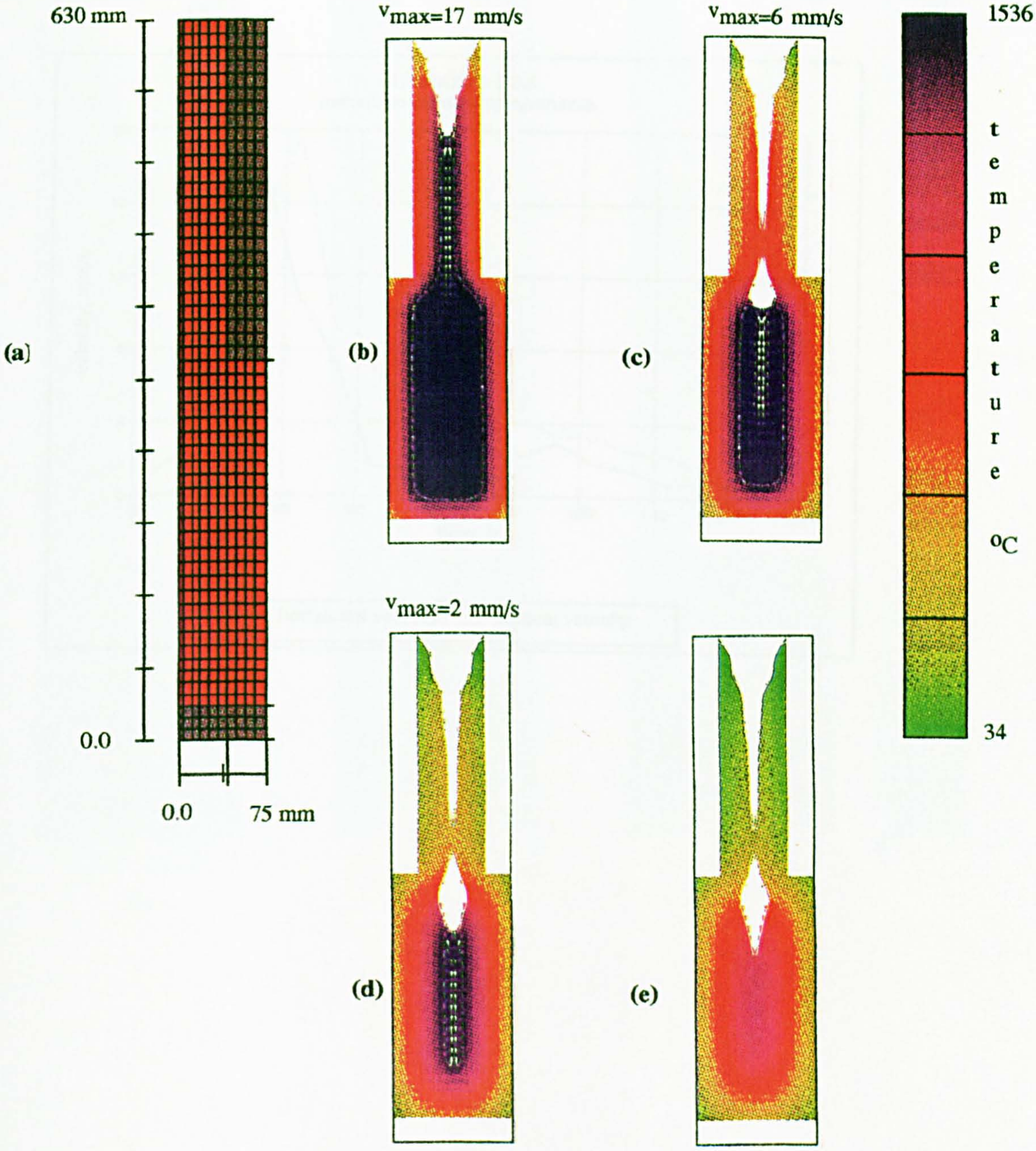


Figure 6.32. Pure iron solidification simulation S3.1: (a) The enmeshed domain. Temperature distributions and cavity development at (b) t=50 s, (c) t=100 s, (d) t=150 s and (e) t=200 s.

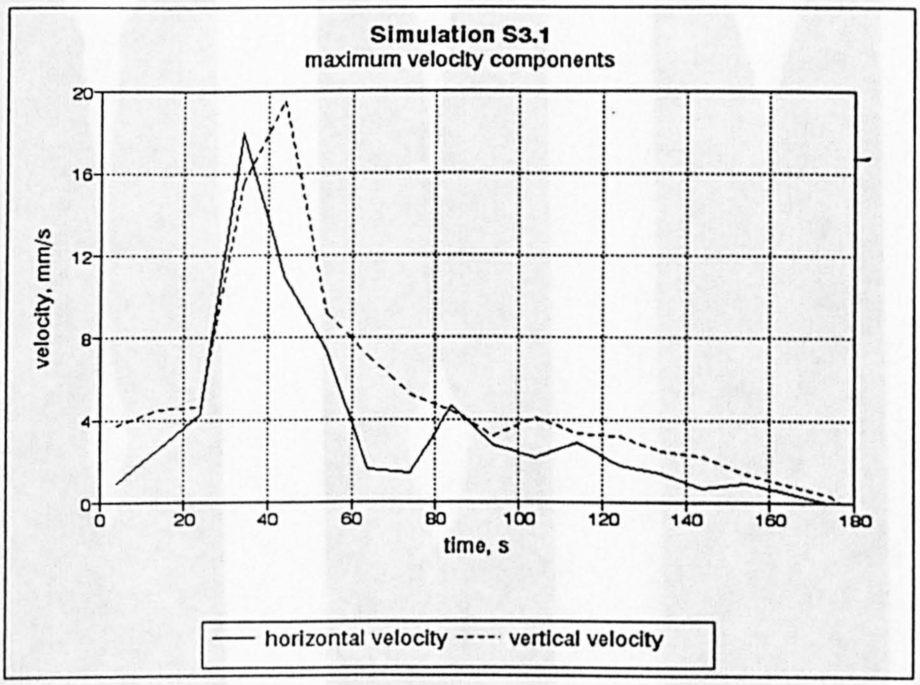


Figure 6.33. Iron-chill cylindrical casting simulation S3.1: maximum vertical (dashed line) and horizontal (solid line) velocities during solidification.

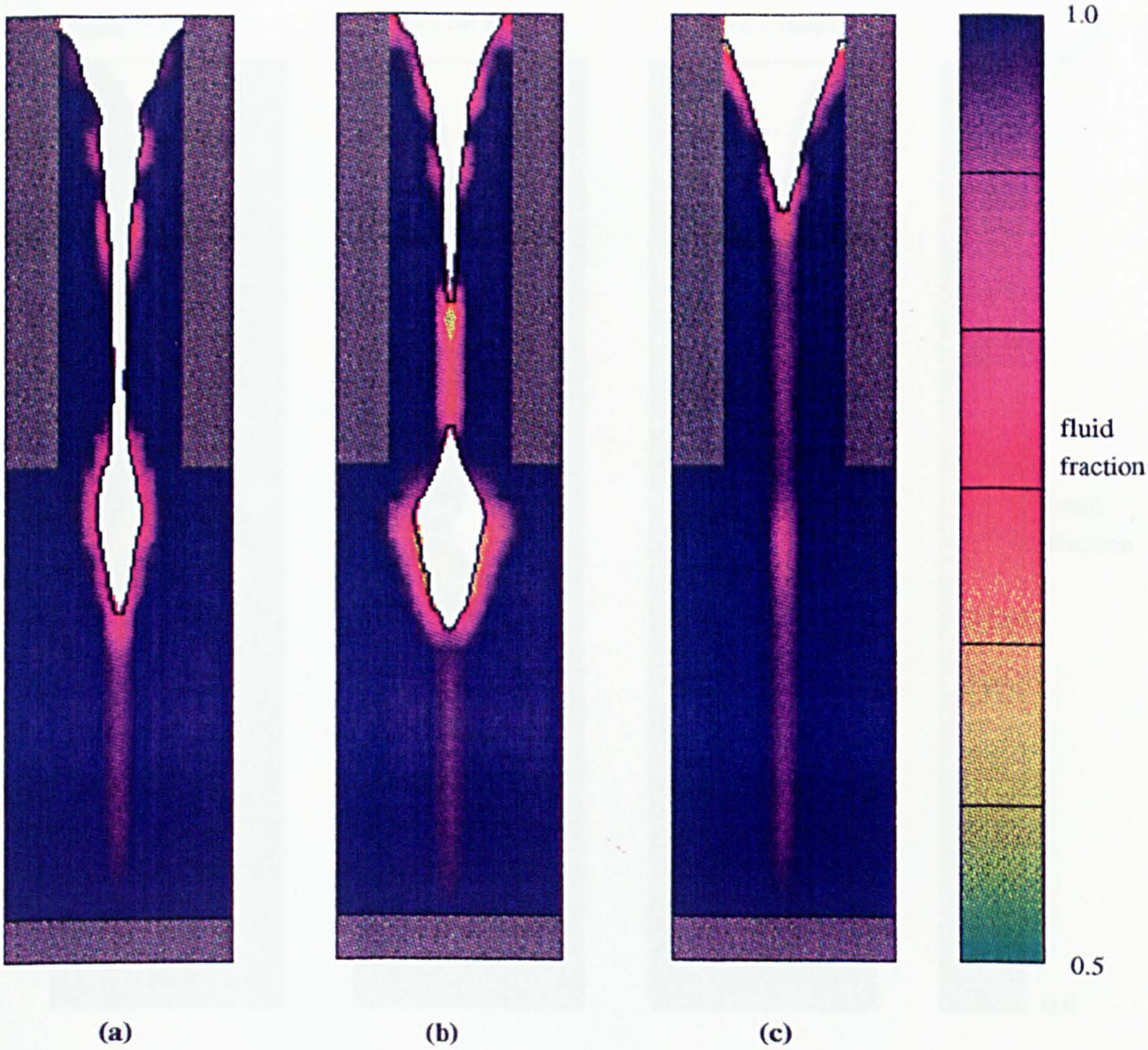


Figure 6.34. Fluid function distribution for pure iron simulations S3.2-4: (a) uniformly smaller heat transfer coefficient, (b) smaller h.t.c only in the lower half of the casting and (c) smaller h.t.c only in the upper half of the casting.

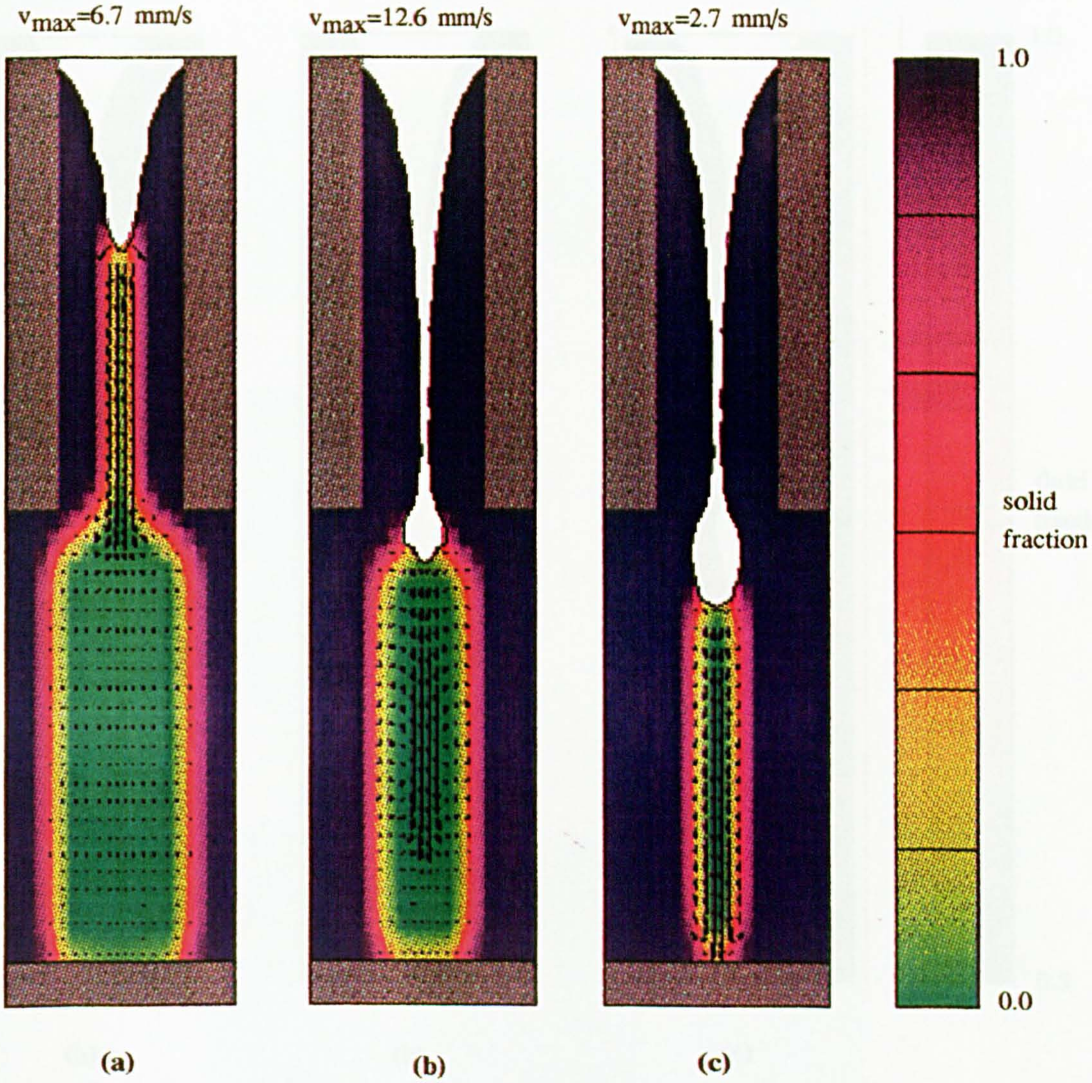


Figure 6.35. Simulation S3.6: fraction of solid function disctibution for alloy steel solidification at (a) t=50 s, (b) t=100 s and (c) t=150 s.

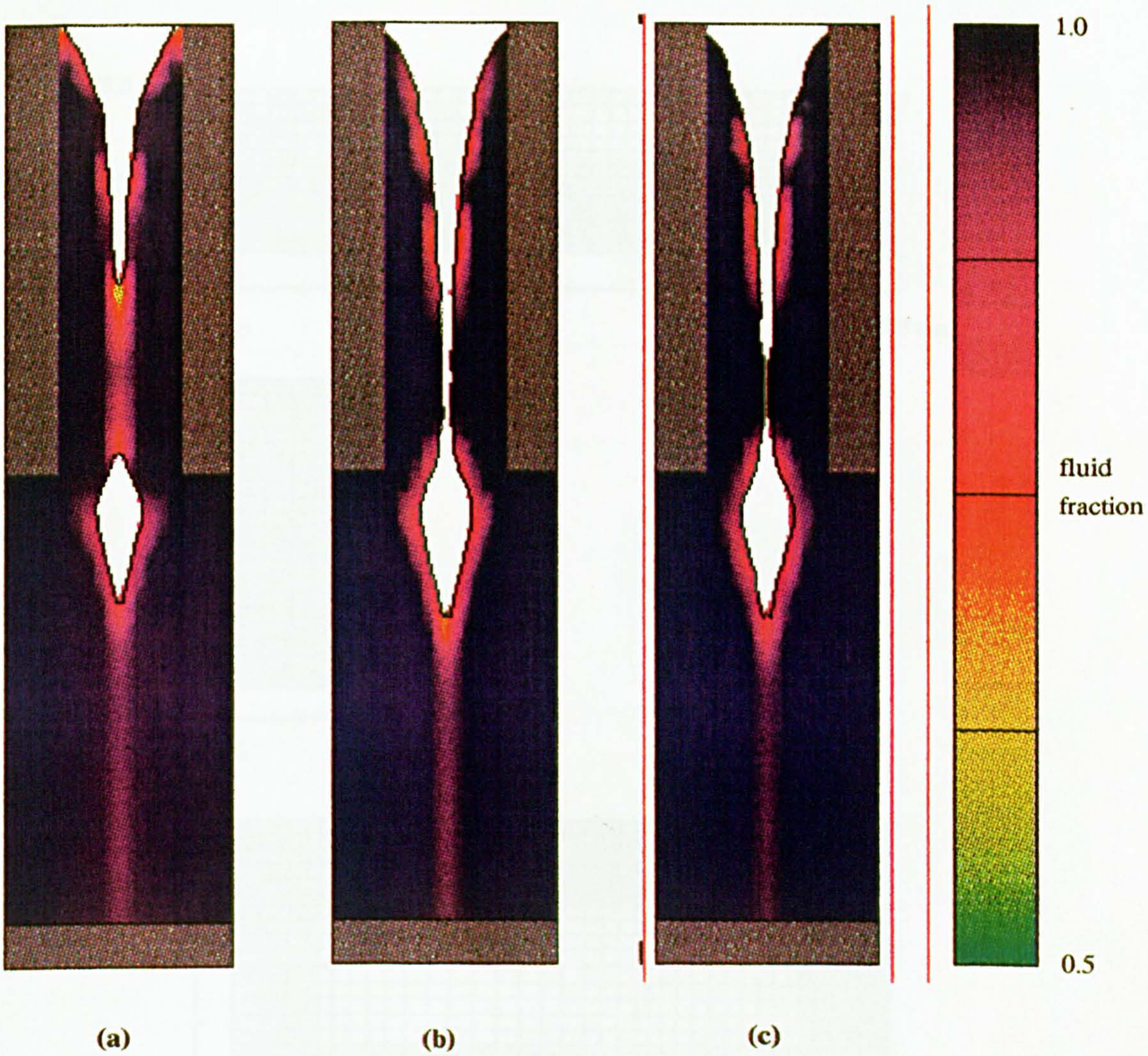


Figure 6.36. Predicted cavity shapes and porosity for alloy steel simulations S3.5-7: (a) $fs_{cr}=0.67$, (b) $fs_{cr}=1.0$ (lever rule) and (c) $fs_{cr}=1.0$ (linear latent heat release). There is a 1% microporosity in case (a).

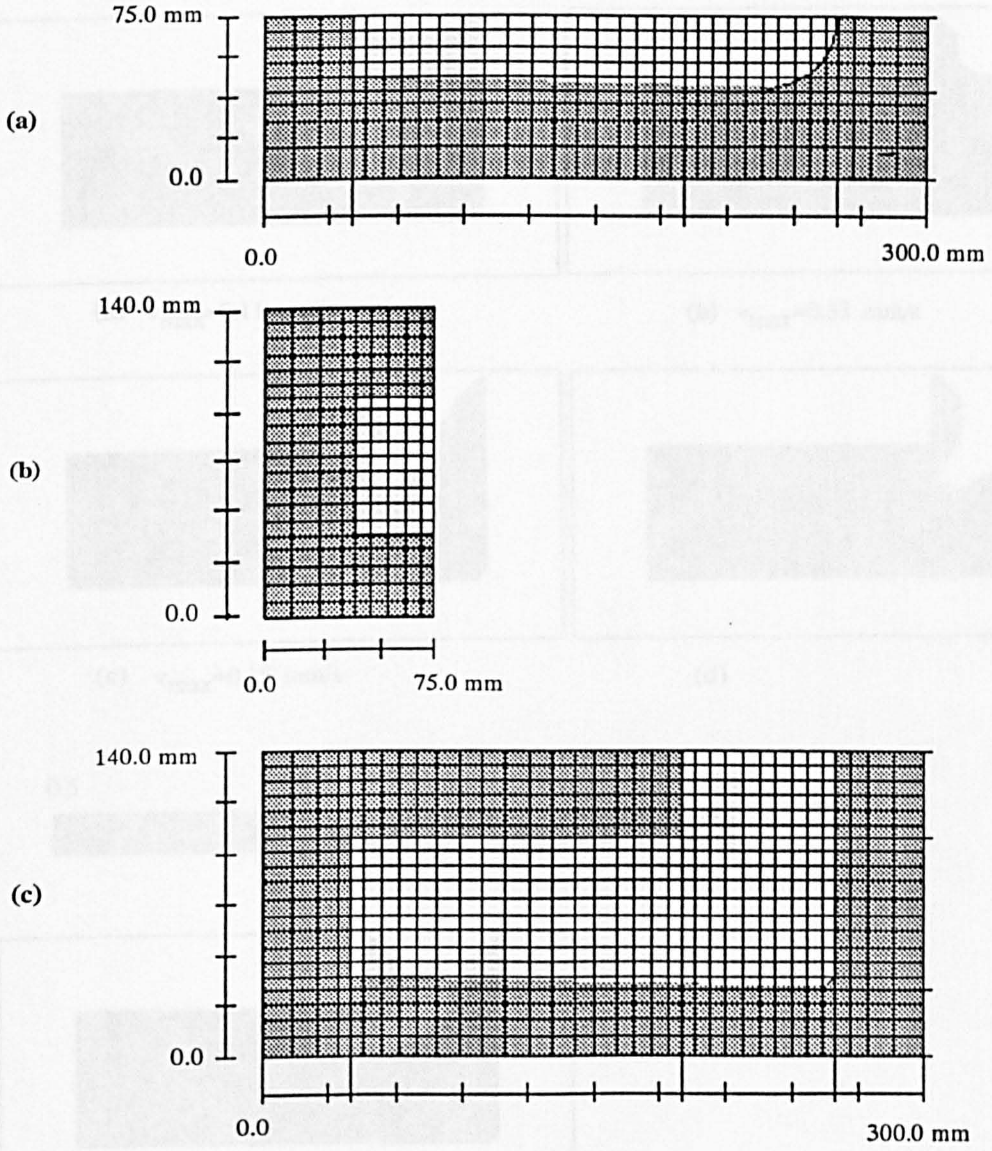
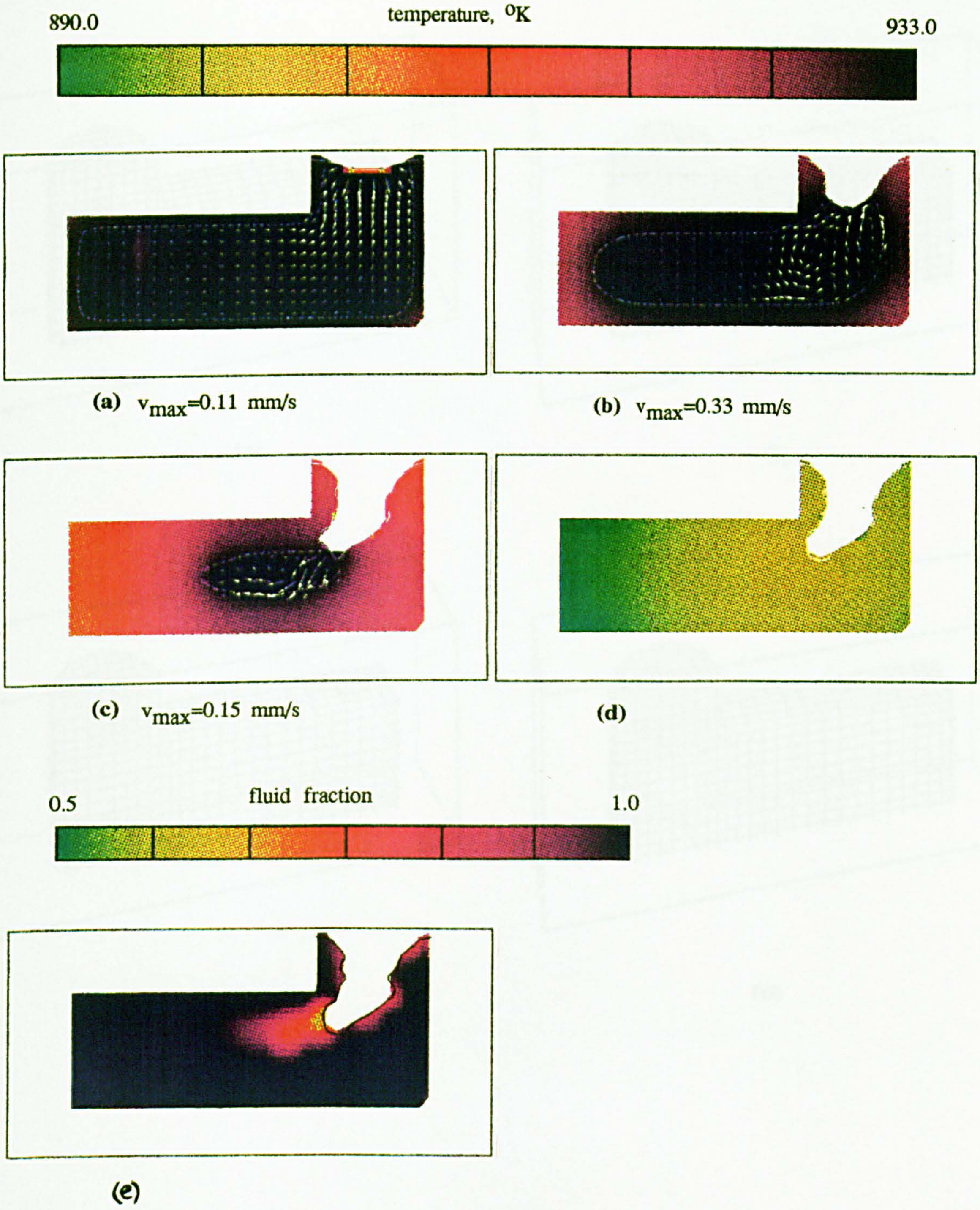


Figure 6.37. The enmeshed domain for boot-shaped casting simulation S4.1: (a) x-y plane, (b) y-z plane and (c) x-z plane.



(a) v_{max}=0.11 mm/s

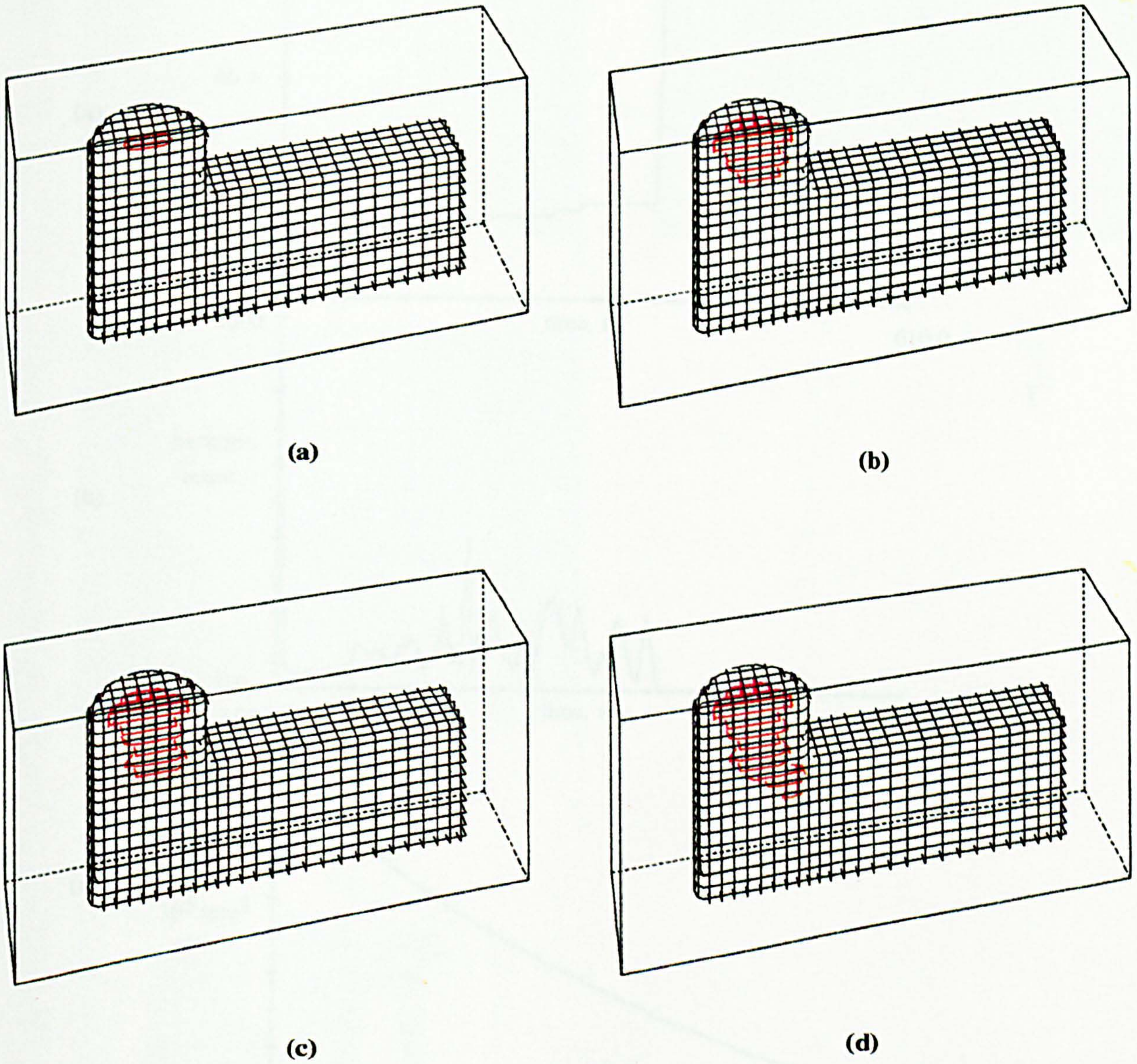
(b) v_{max}=0.33 mm/s

(c) v_{max}=0.15 mm/s

(d)

(e)

Figure 6.38. Boot-shaped casting simulation S4.1. Vecolities and temperature contours: (a) t=100.0 s, (b) t=300.0 s, (c) t=500.0 s and (d) t=600.0 s; (e) the distribution of the fluid fraction function in the solidified casting.



(a)

(b)

(c)

(d)

Figure 6.39. Simulation S4.1: 3-D cavity shape at (a) t=100.0 s, (b) t=300.0 s, (c) t=400.0 s and (d) t=600.0 s.

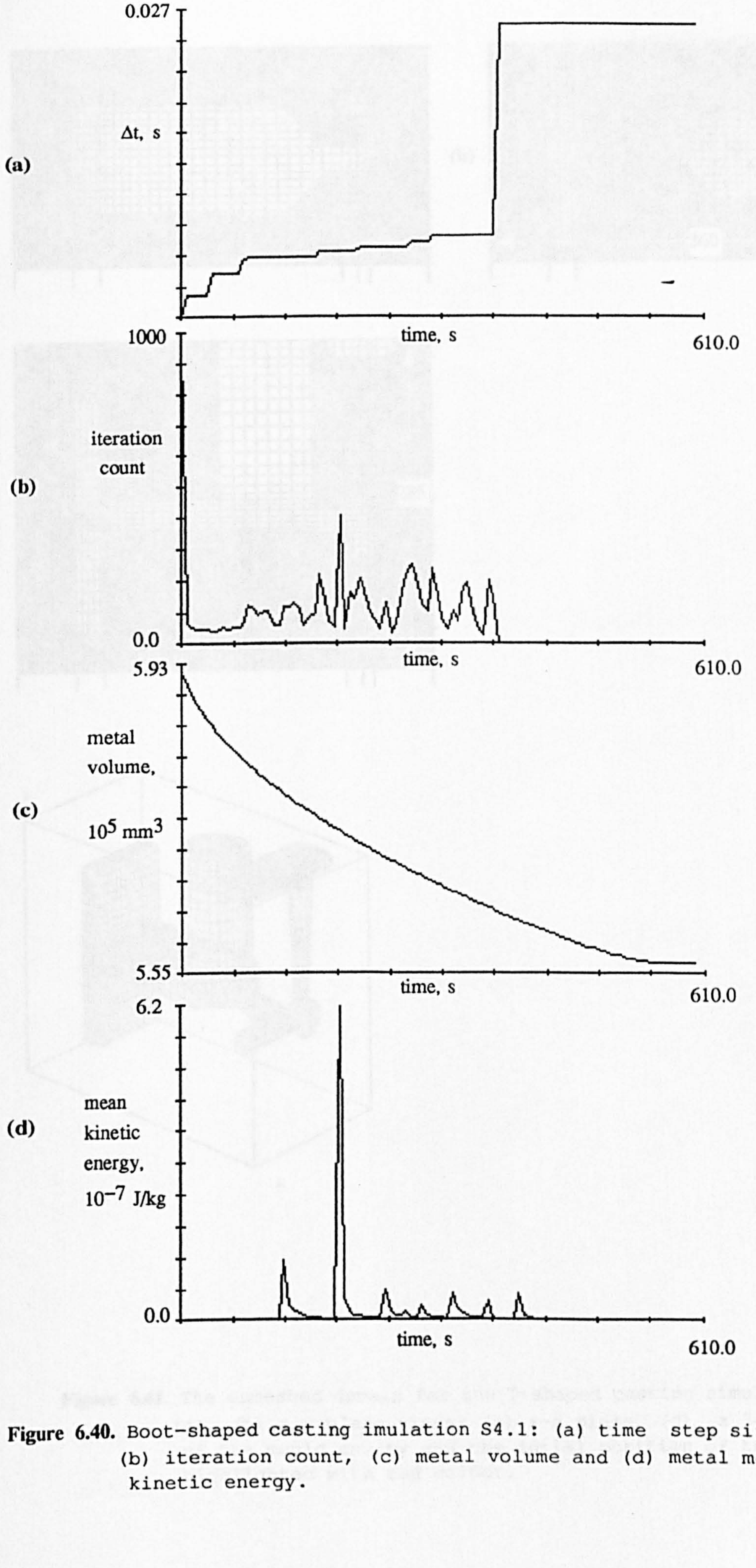


Figure 6.40. Boot-shaped casting imulation S4.1: (a) time step size, (b) iteration count, (c) metal volume and (d) metal mean kinetic energy.

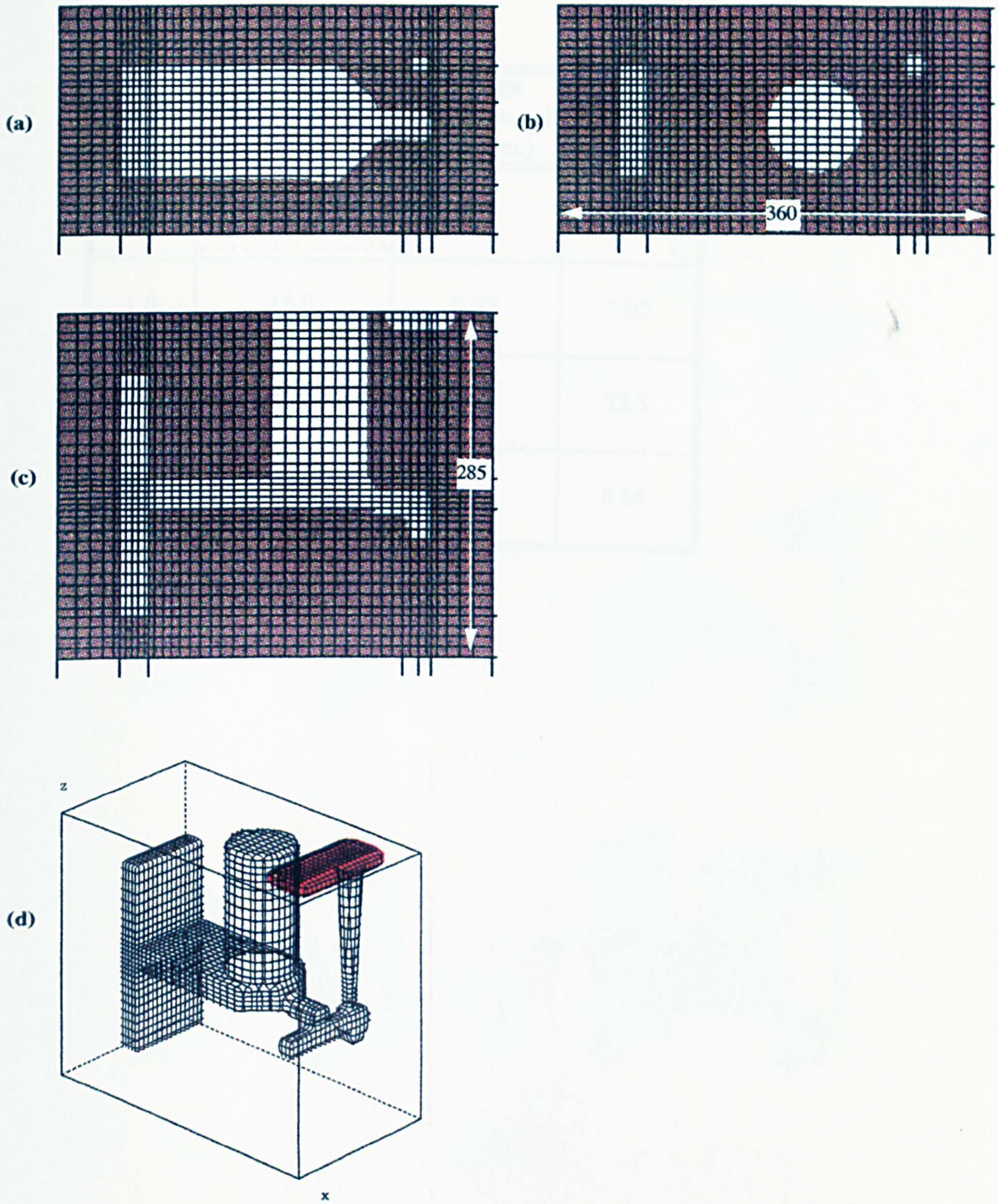


Figure 6.41. The enmeshed domain for the T-shaped casting simulation: (a), (b) x-y plane views; (c) x-z plane, (d) a 3-D view of the mould cavity and the intial position of the metal highlighted with red colour.

R=a/b	choke diameter (mm)	average experimental time (sec)	simulated time (sec)
0.6	15.0	5.92	6.5
1.0	15.0	6.82	7.02
1.0	8.0	20.75	22.5
1.7	15.0	7.1	6.56

Figure 6.42. Comparison of predicted and measured filling times for T-shaped casting.

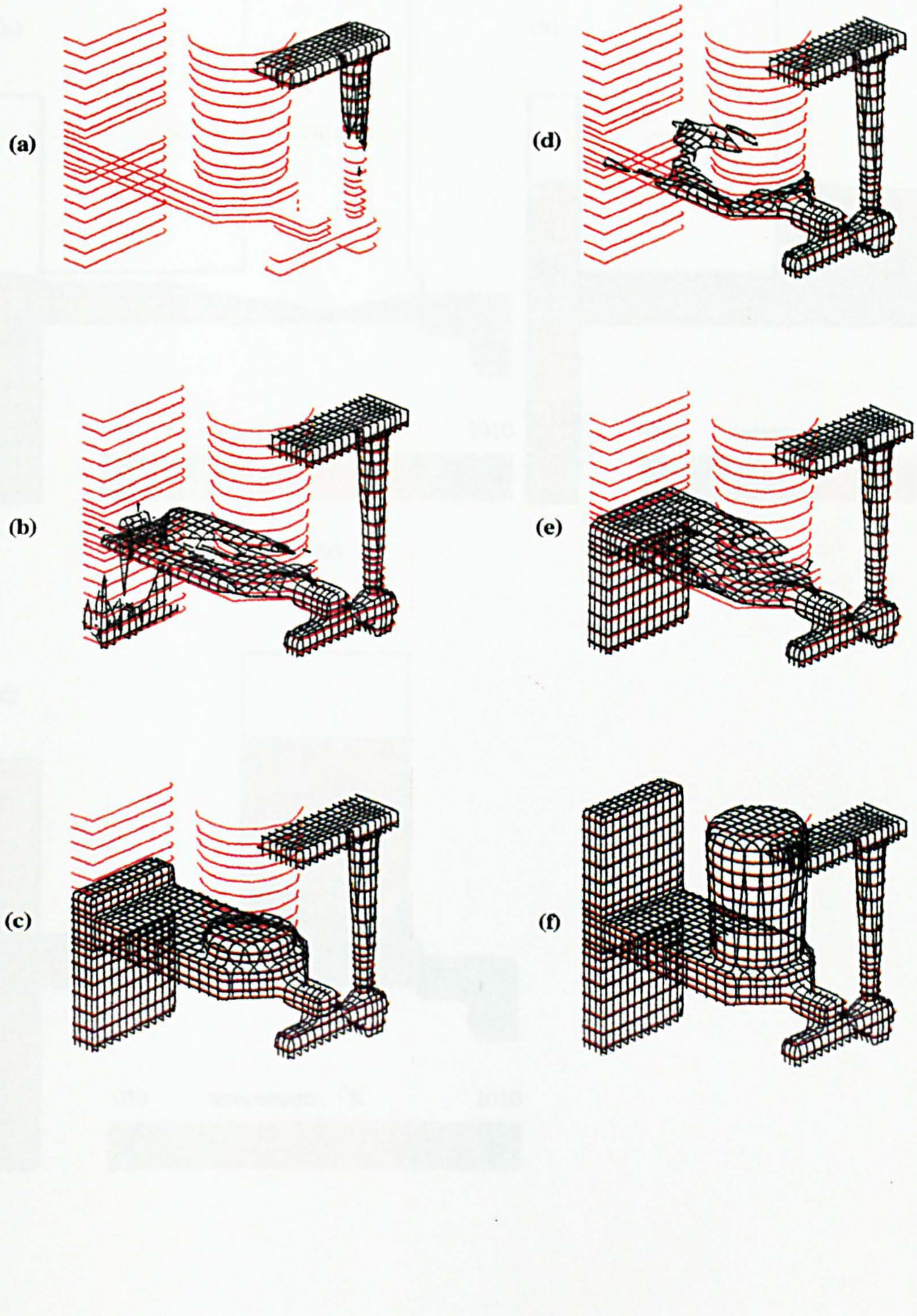


Figure 6.43. Simulation S5.1: filling sequence (the mould surface is outlined in red): (a) t=0.01 s, (b) t=0.7 s, (c) t=1.0 s, (d) t=2.4 s, (e) t=3.9 s and (f) t=7.0s.

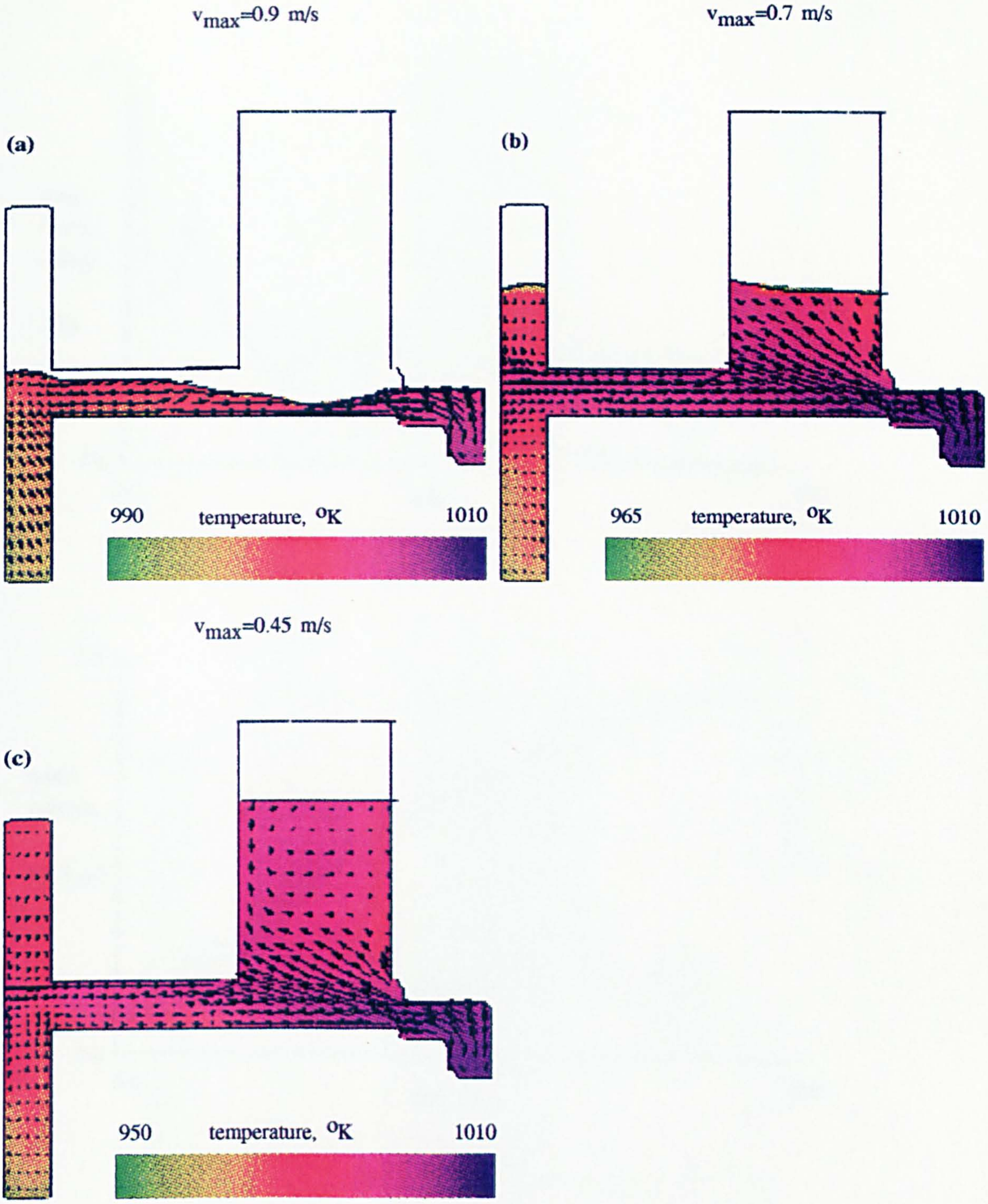


Figure 6.44. Simulation S5.1: Temperature distributions and velocities during filling at (a) t=2.0 s, (b) t=4.0 s and (c) t=6.0 s.

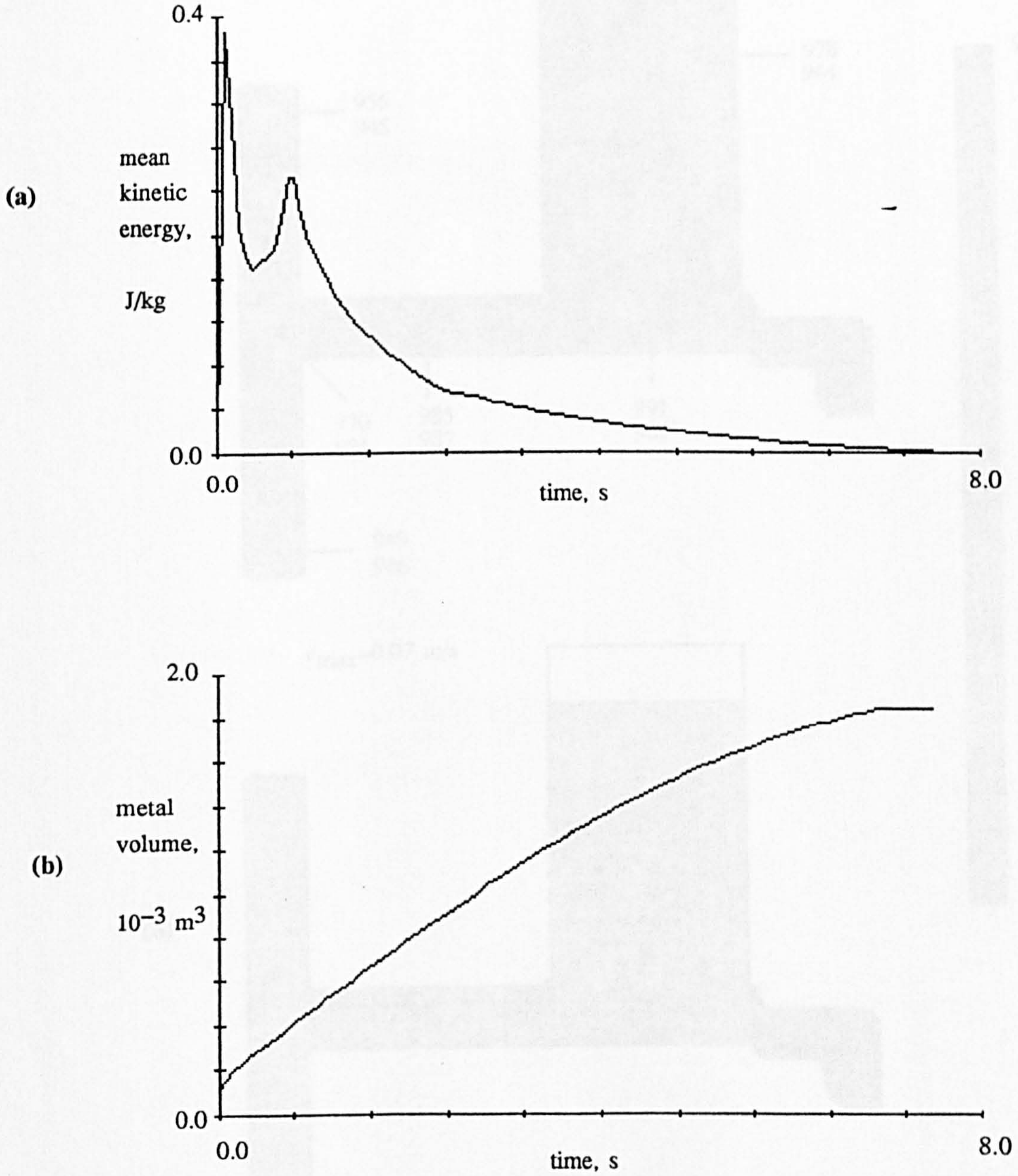


Figure 6.45. Simulation S5.1: (a) mean metal kinetic energy and (b) metal volume evolution during filling.

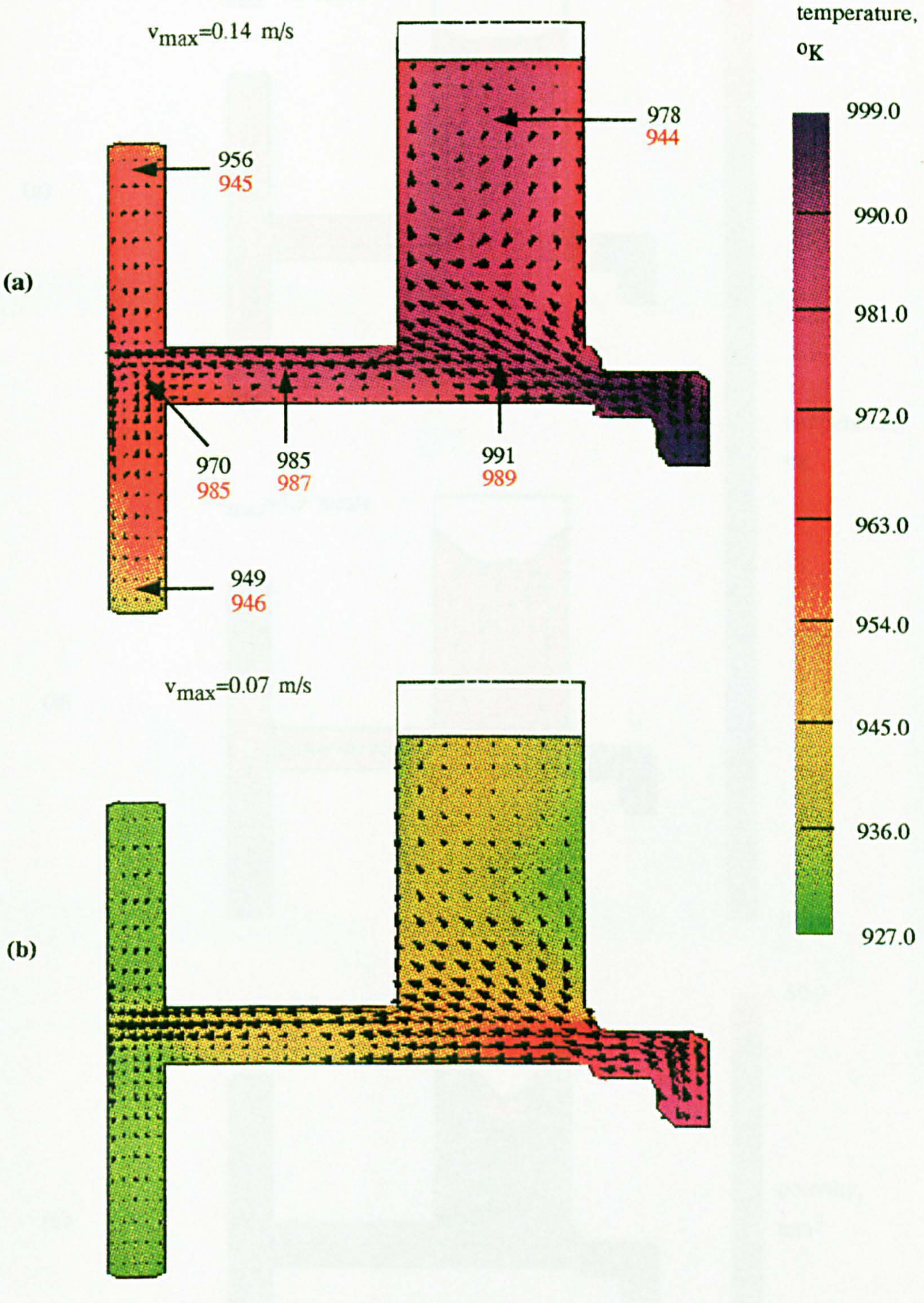


Figure 6.46. Temperature distribution and velocities at the end of the filling for (a) 15 mm choke, t=7.5 s (simulation S5.1) and (b) 8 mm choke at t=22.0 s (simulation S5.3). The initial superheat in each case is 80^{0}K. Maximum velocities are shown in cm/s. Arrows show the thermocouple positions with the measured (red) and predicted (black) temperatures.

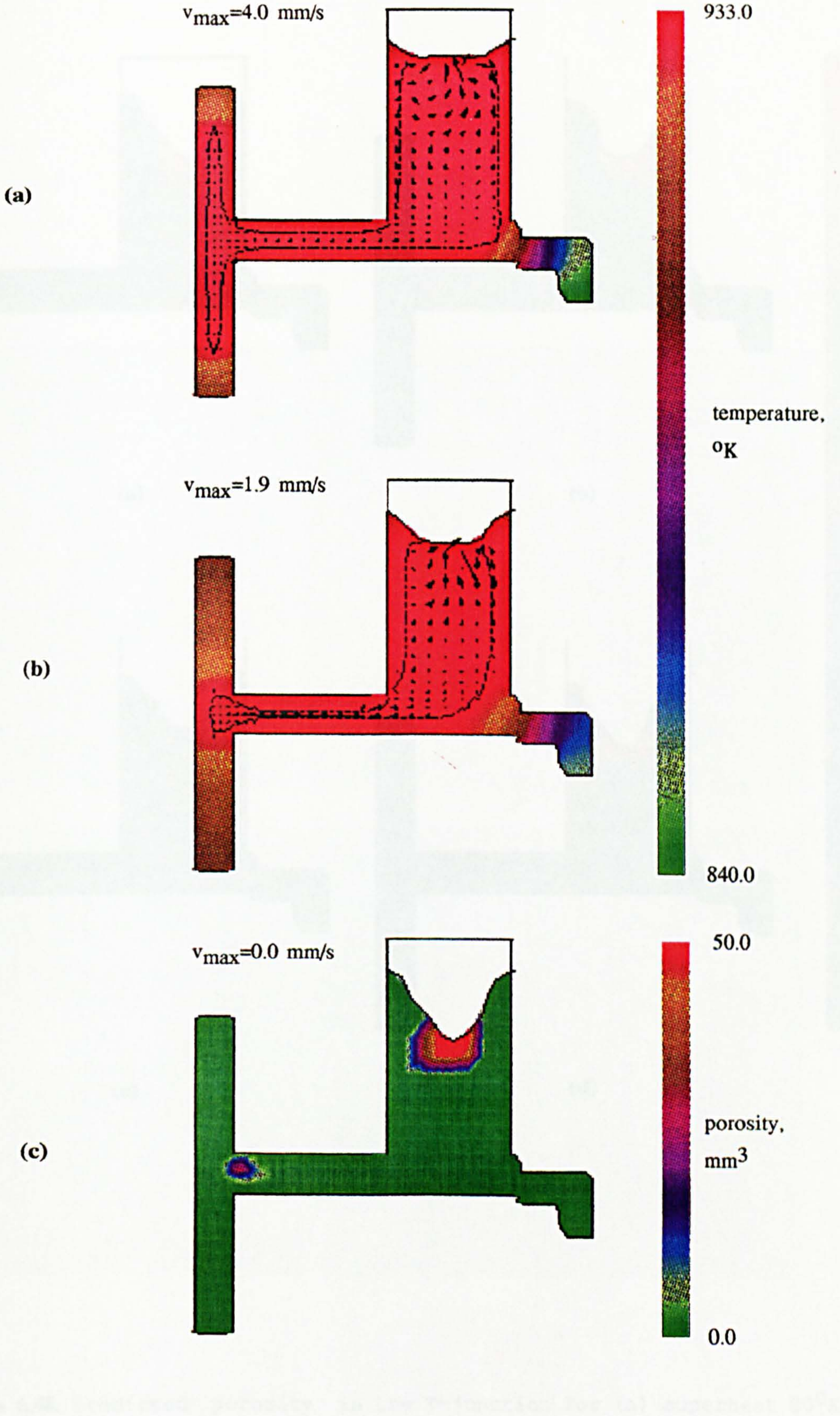


Figure 6.47. Simulation S5.2: Temperature distribution, velocities and solidification front position at (a) t=120.0 s and (b) t=180.0 s; (c) Porosity distribution in the solidified casting.

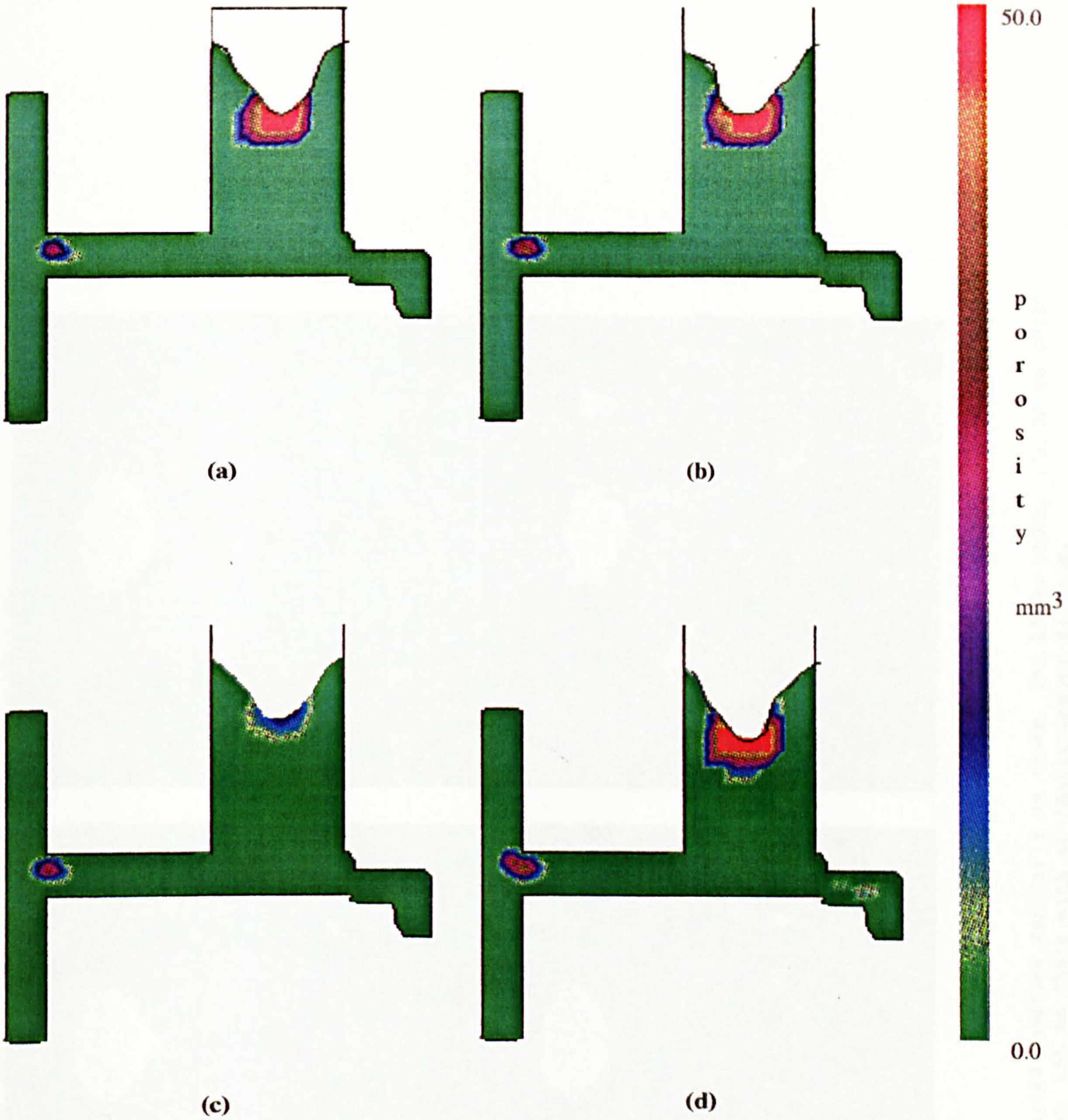


Figure 6.48. Predicted porosity in the T-junction for (a) superheat 80^{0}K, filling time 7.0 s; (b) superheat 80^{0}K, instantaneous filling; (c) superheat 30^{0}K, filling time 7.0 s; (d) superheat 30^{0}K, instantaneous filling.

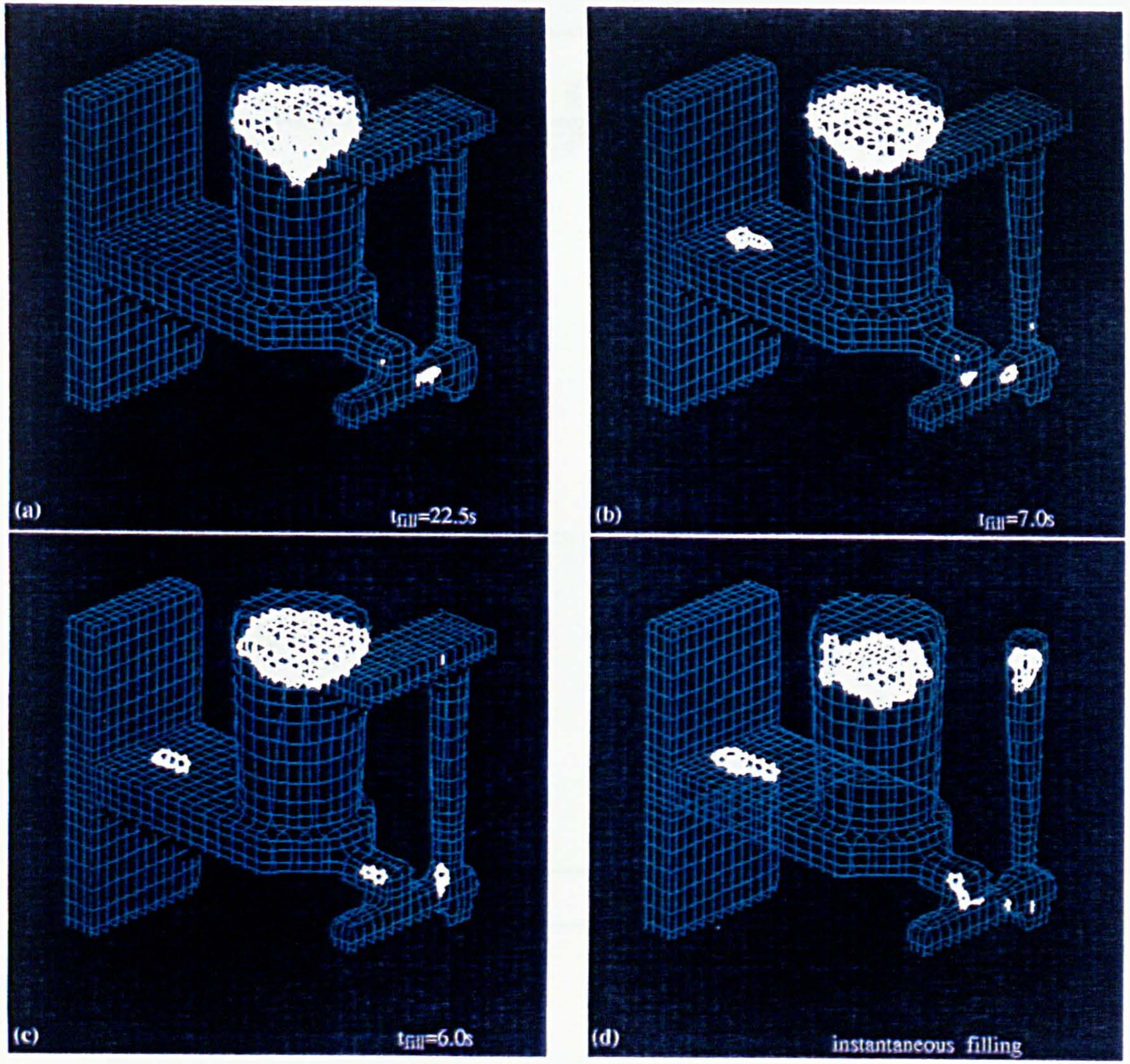


Figure 6.49. Predicted cavities for (a) 8 mm choke, (b) 15 mm choke, (c) 22 mm choke and (d) 155 mm choke with an instantaneous filling.

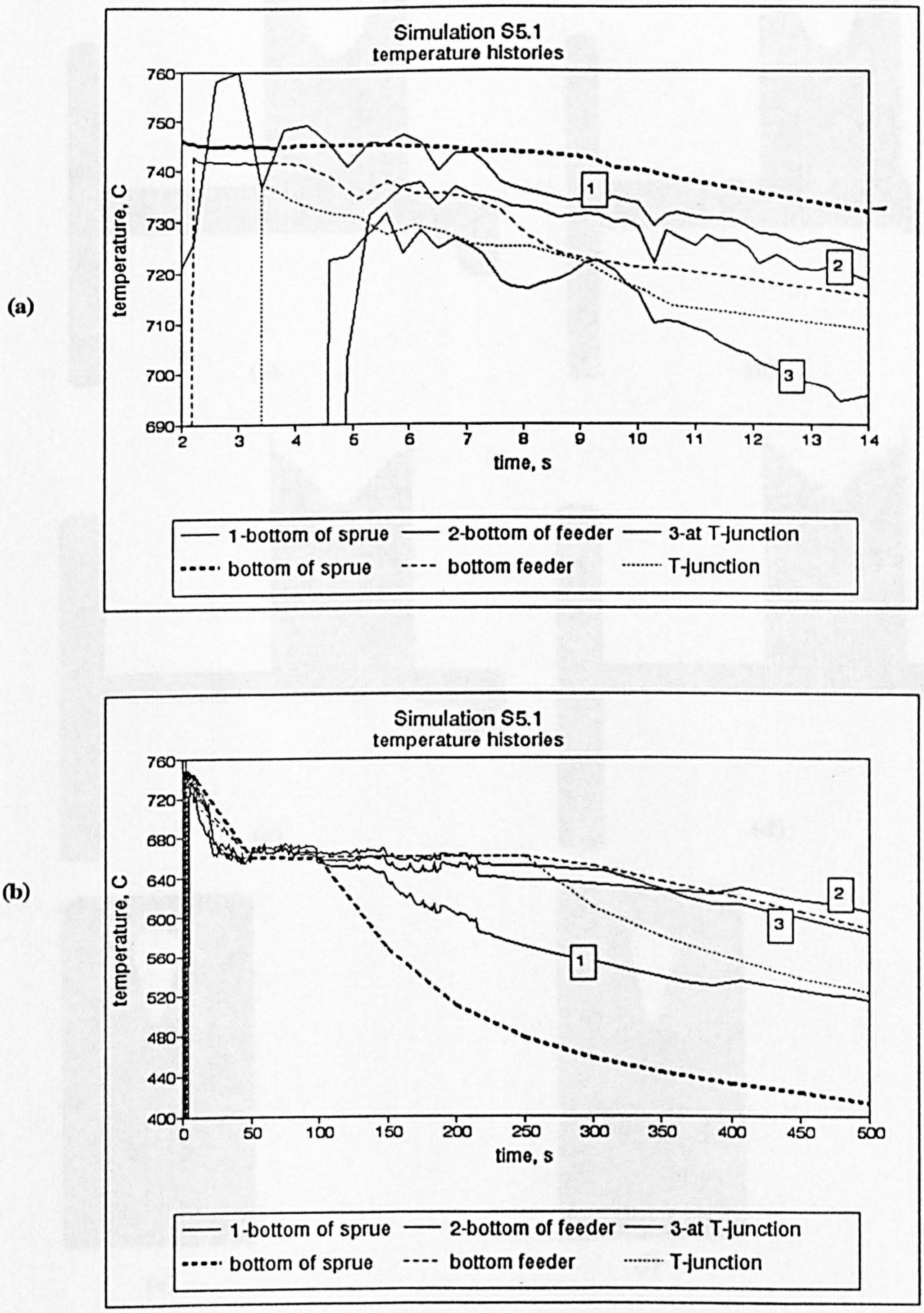


Figure 6.50. Comparisons of metal cooling curves at three locations in the T-shaped casting for simulation S5.1: (A) during filling and (b) during solidification.

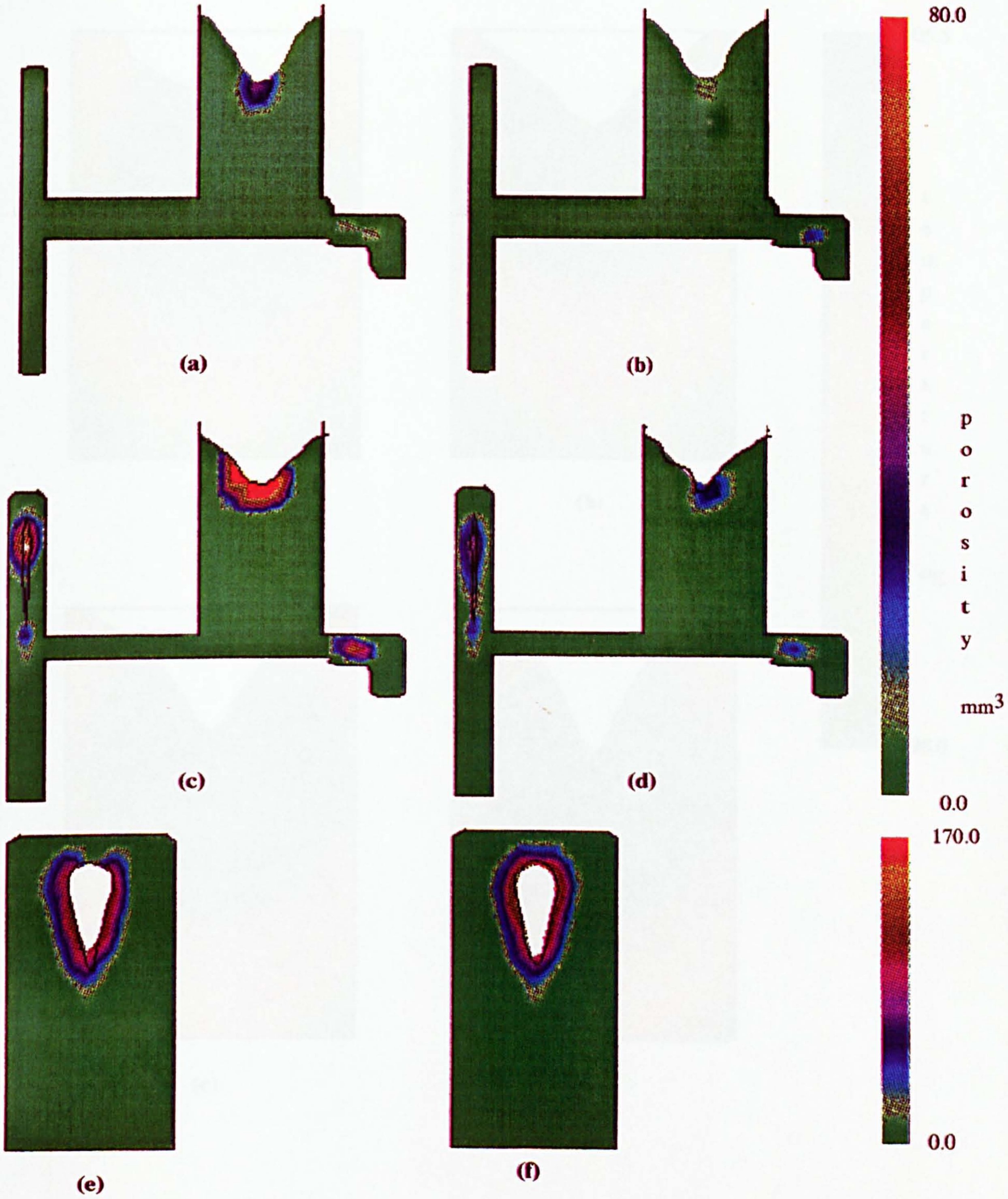


Figure 6.51. Simulations S5.4-7. Porosity distribution: (a) R=0.6, superheat 30^0K, (b) R=0.6, superheat 80^0K, (c),(e) R=1.67, superheat 30^0K and (d),(f) R=1.67, superheat 80^0K.

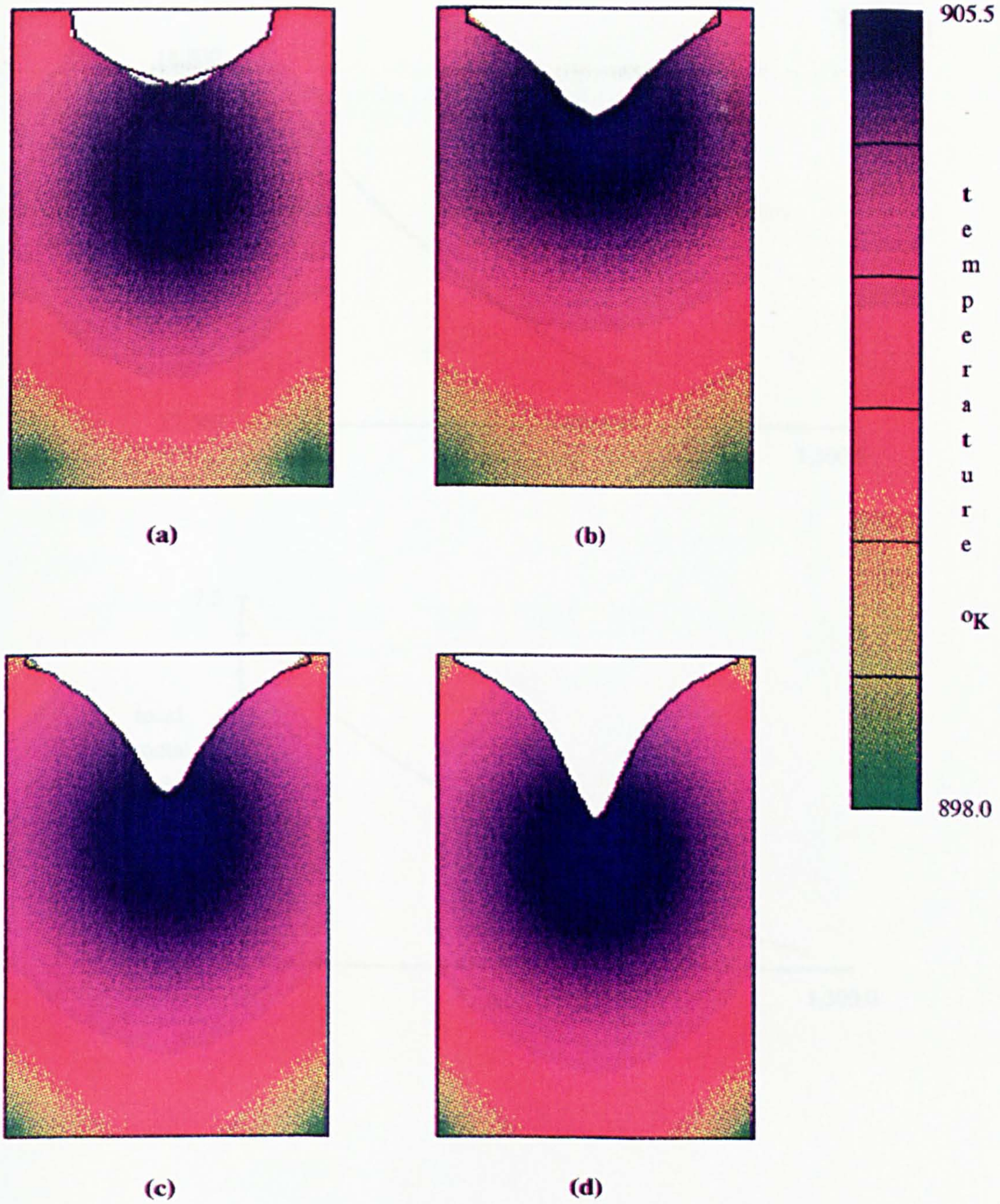


Figure 6.53. Predicted primary cavity shapes with metal temperature distributions using the M2 shrinkage model for simulations S2.11-14: (a) N=90; (b) N=360; (c) N=1000 and (d) N=3240 (N - the total number of cells). See figure 6.26.

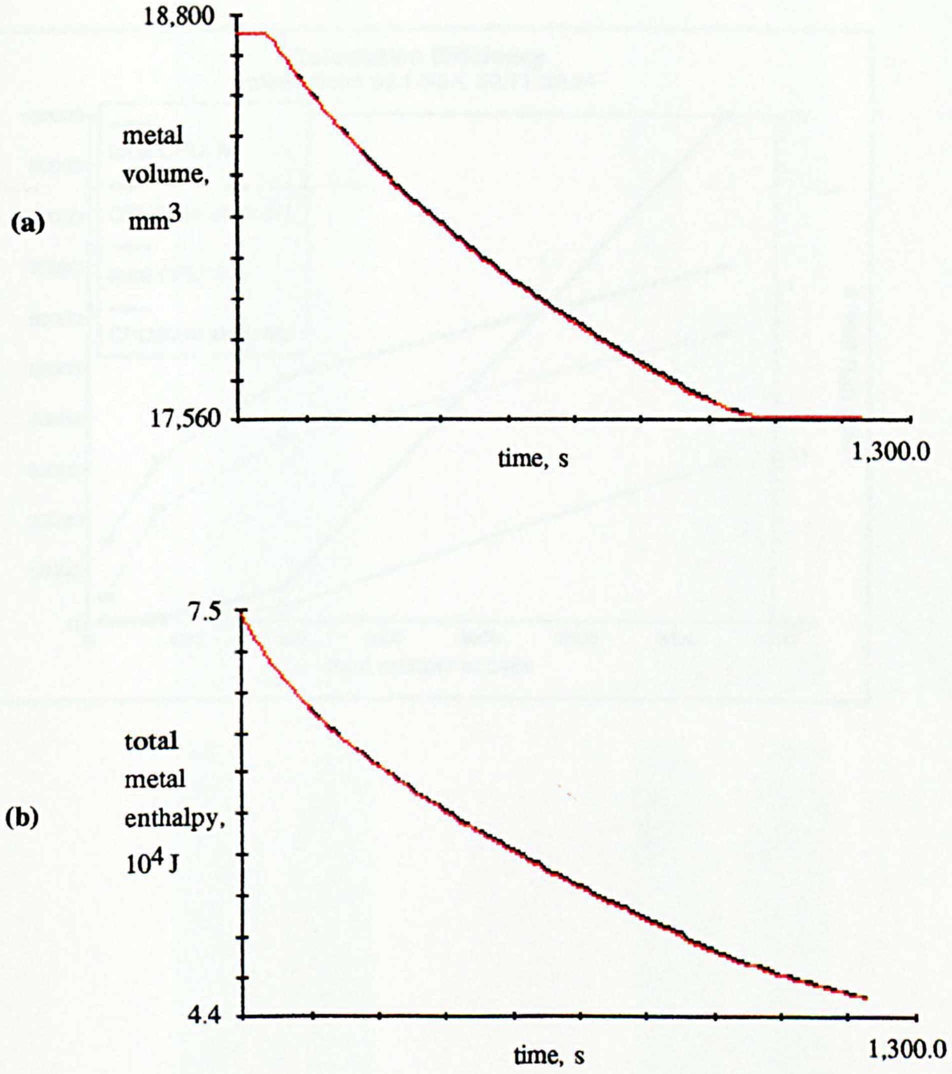


Figure 6.54. (a) Metal volume and (b) total enthalpy evolutions for cylindrical casting simulations S2.2 (full model, black) and S2.12 (simplified model, red).

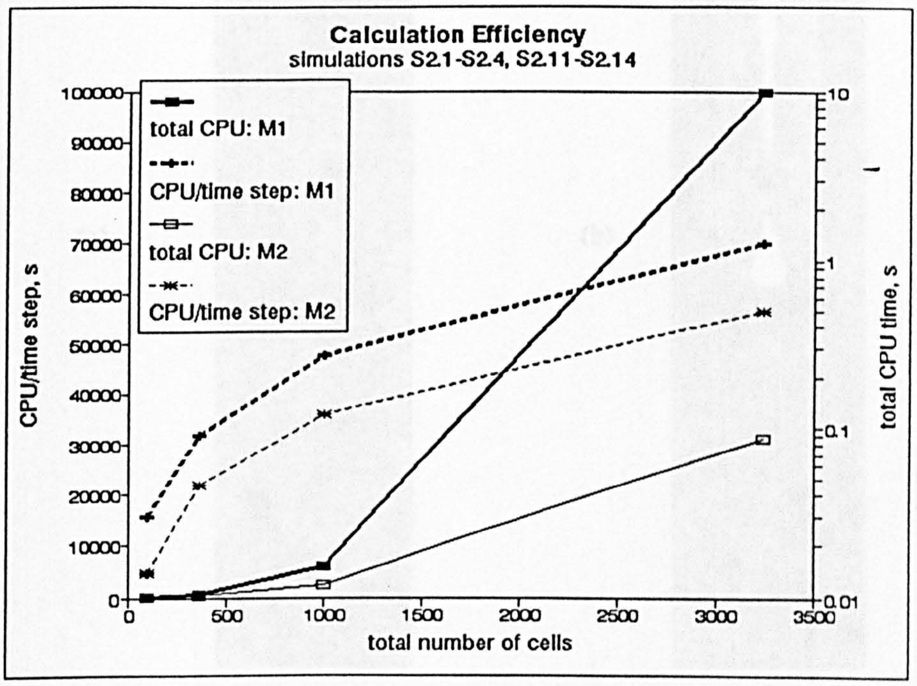


Figure 6.55. Total CPU time and CPU time per time step as functions of the uniform mesh cell number for the full shrinkage model M1 (simulations S2.1-S2.4) and for the simplified shrinkage model M2 (simulations S2.11-S2.14).

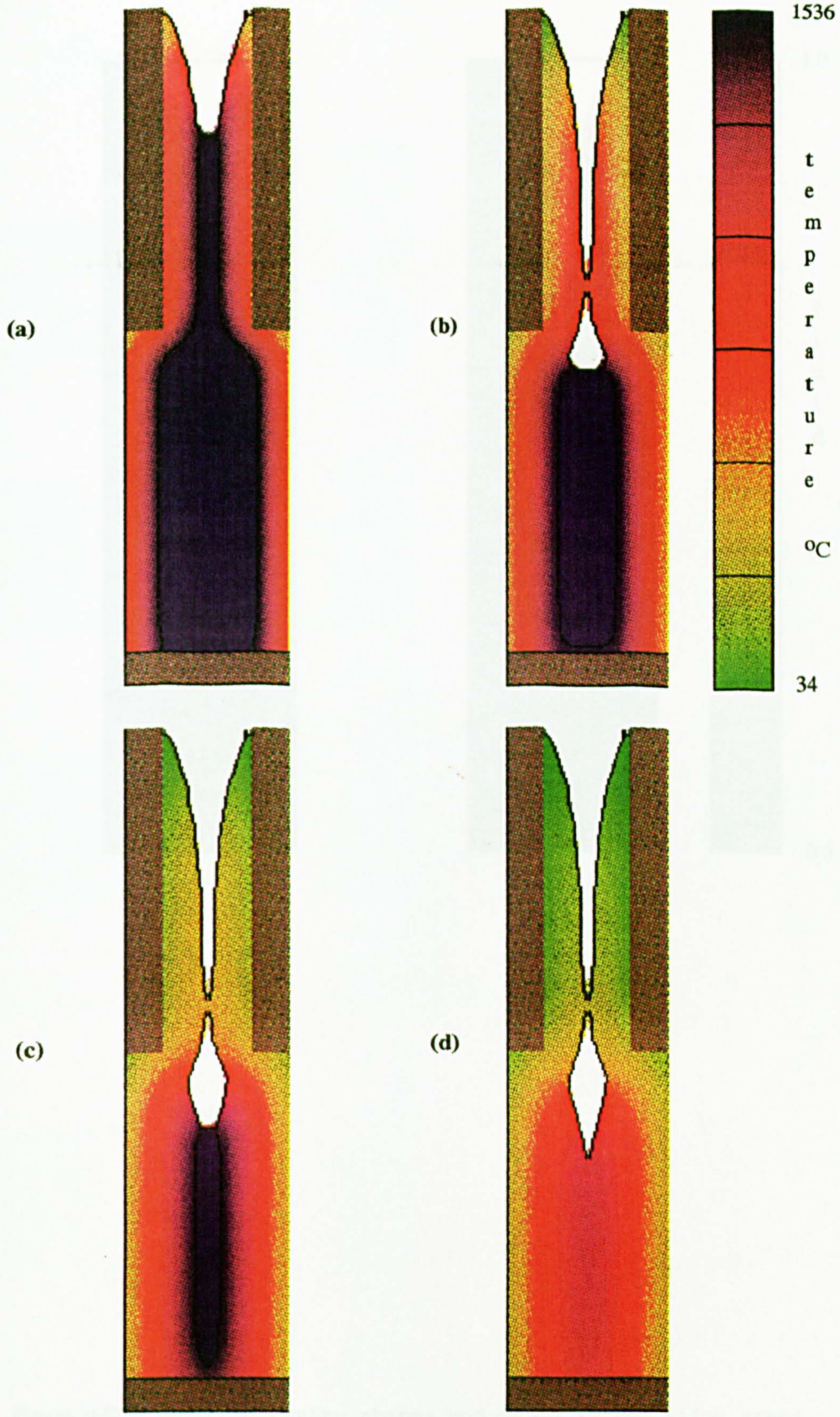


Figure 6.56. Pure iron solidification simulation S3.11 using the M2 model. Temperature distributions and cavity development at (a) t=50 s, (b) t=100 s, (c) t=150 s and (d) t=200 s.

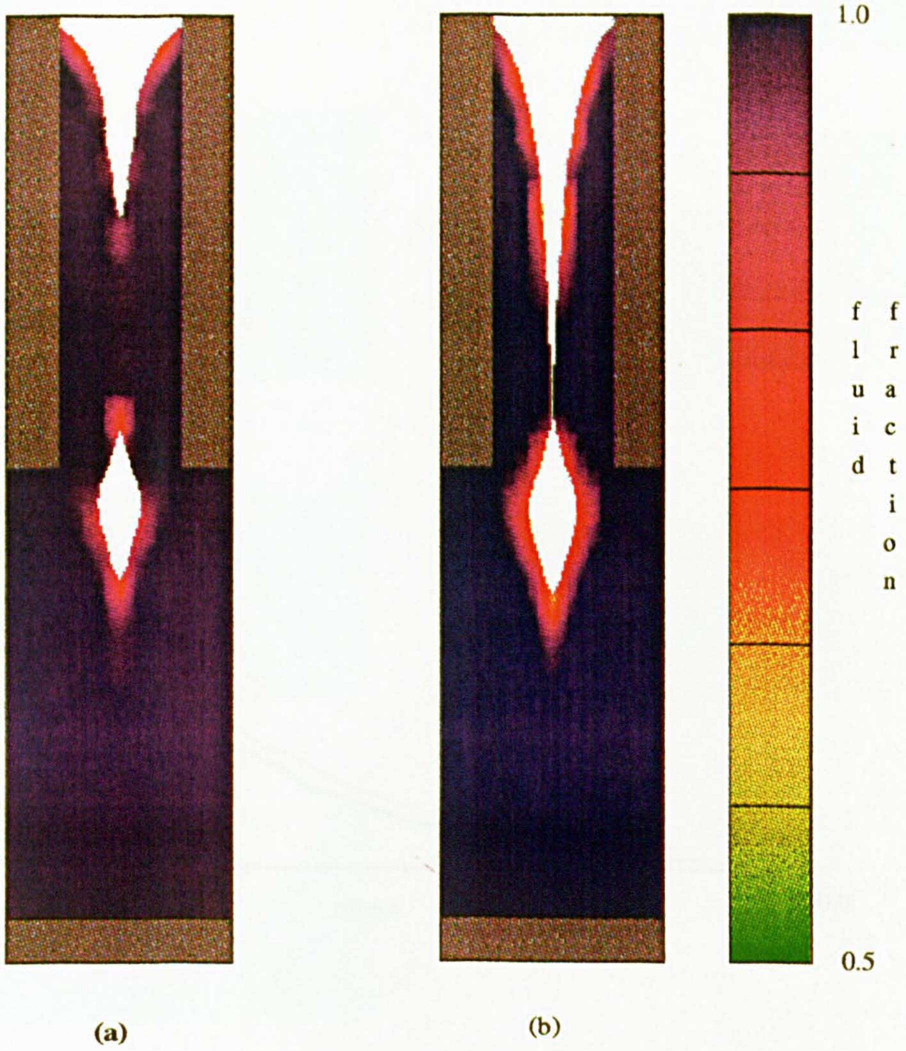


Figure 6.57. Predicted cavity shapes and porosity for alloy steel simulations S3.15 and S3.16 using the M2 model: (a) fs_{cr}=0.67, (b) fs_{cr}=1.0. There is a 1% microporosity in case (a).

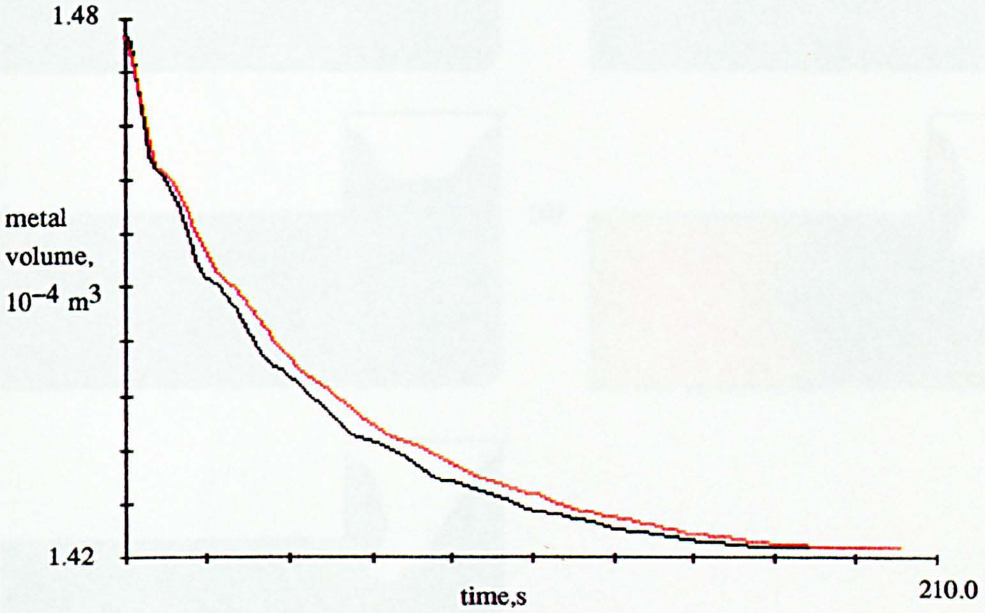


Figure 6.58. Metal volume evolution: pure iron (black) and alloy steel (red) simulated by the simplified model.

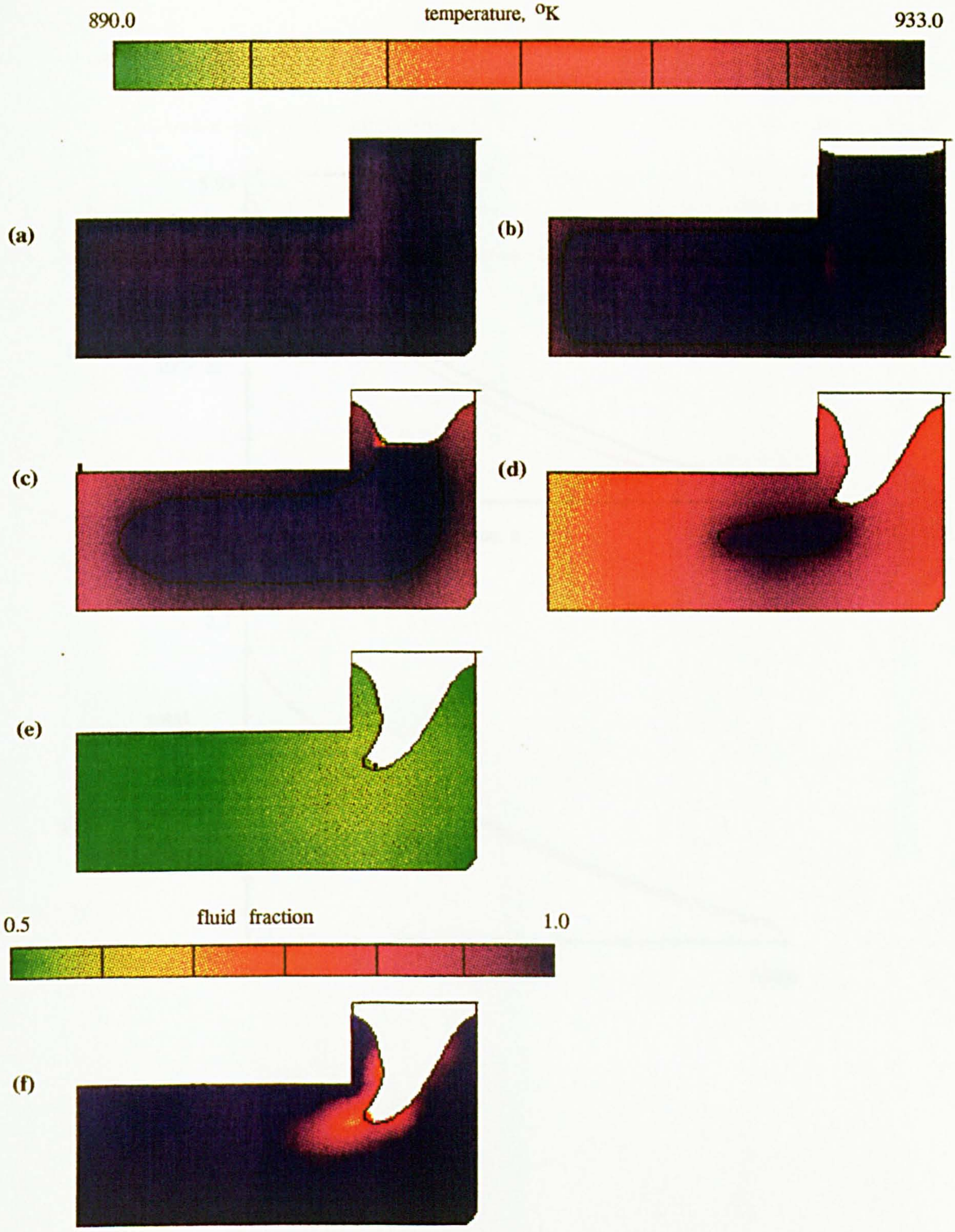


Figure 6.59. Boot-shaped casting simulation S4.11 using the M2 model. Temperature contours: (a) t=0.0 s, (b) t=100.0 s, (c) t=300.0 s, (d) t=500.0 s and (e) t=600.0 s; (f) the distribution of the fluid fraction function in the solidified casting.

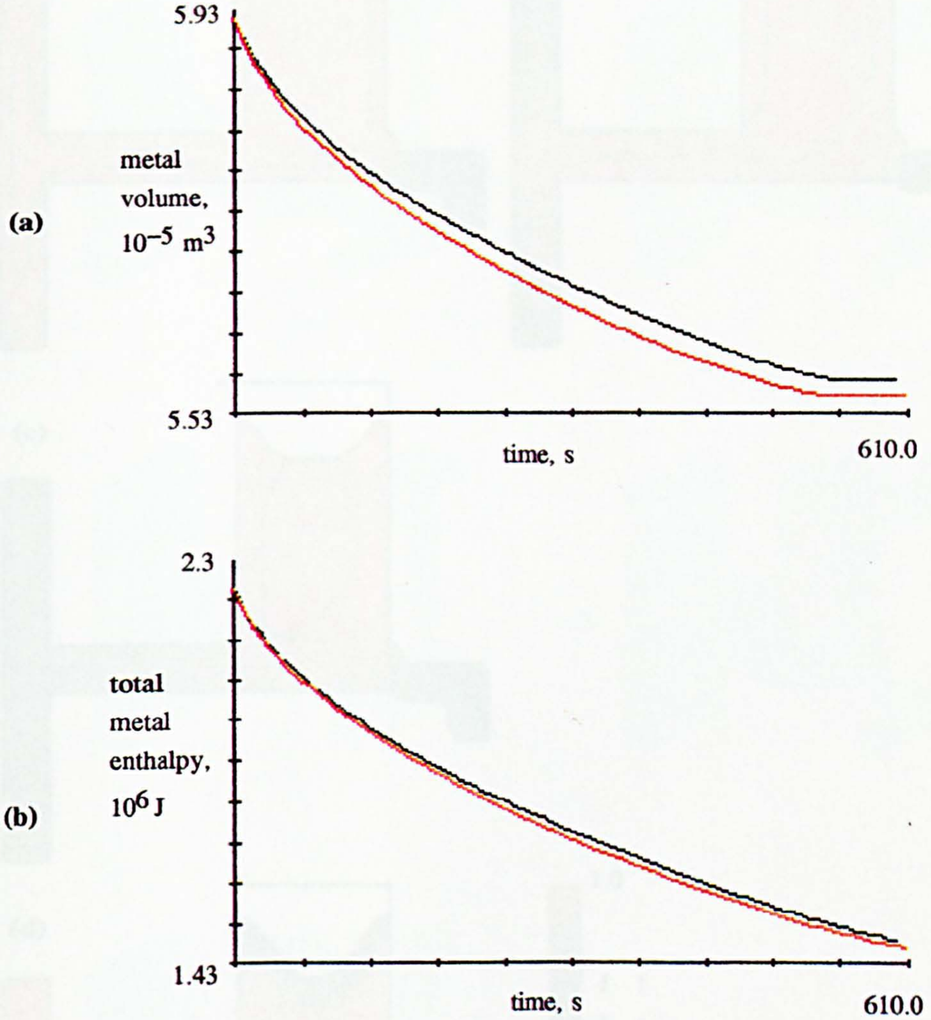


Figure 6.60. (a) Metal volume and (b) total enthalpy evolutions for boot-shaped casting simulations S4.1 (full model, black) and S4.11 (simplified model, red).

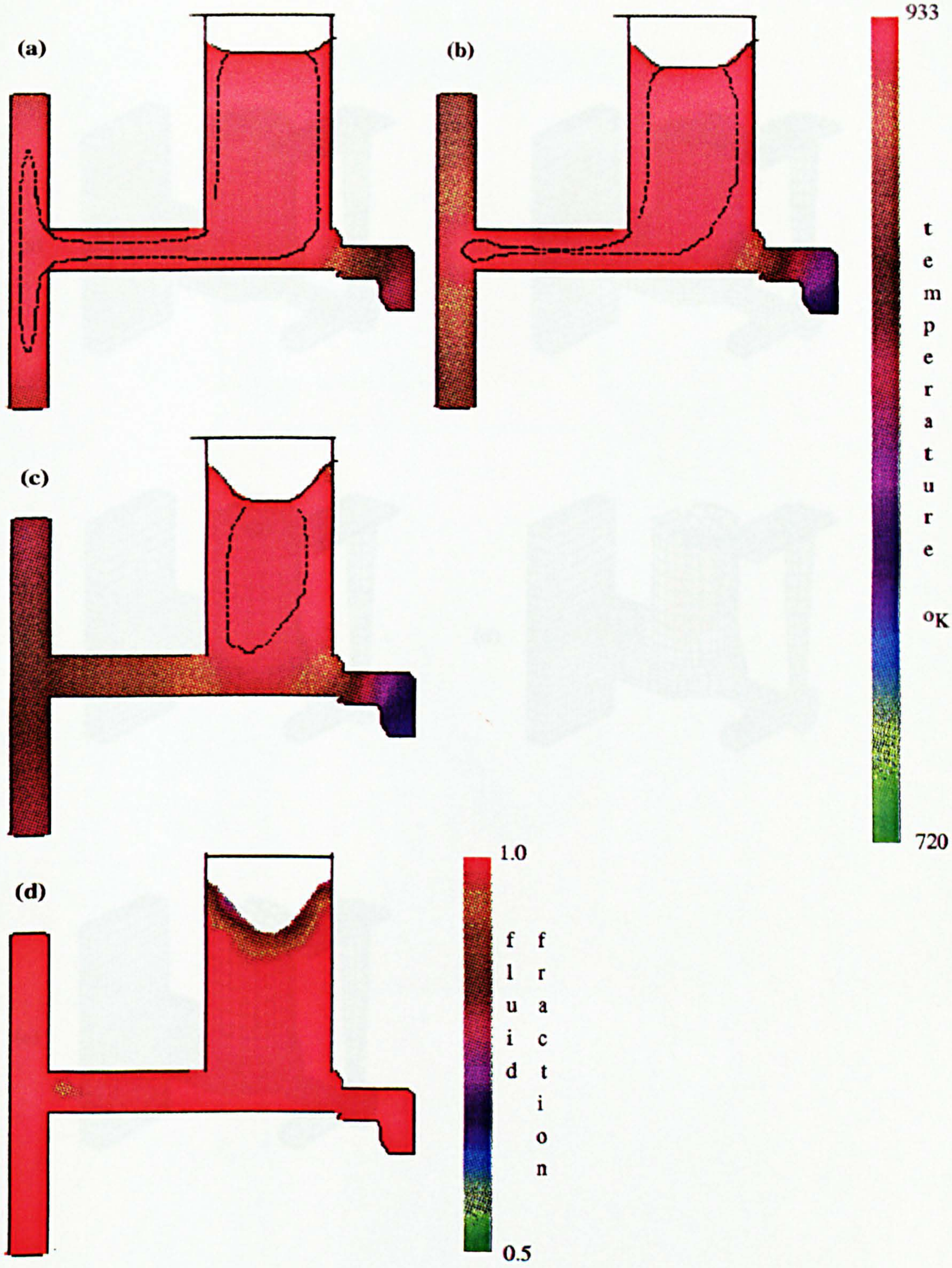


Figure 6.61. Simulation S5.11: temperature contours and solidification front position at (a) t=200 s, (b) t=300 s and (c) t=400 s. (d) fluid fraction contours in the solidified casting.

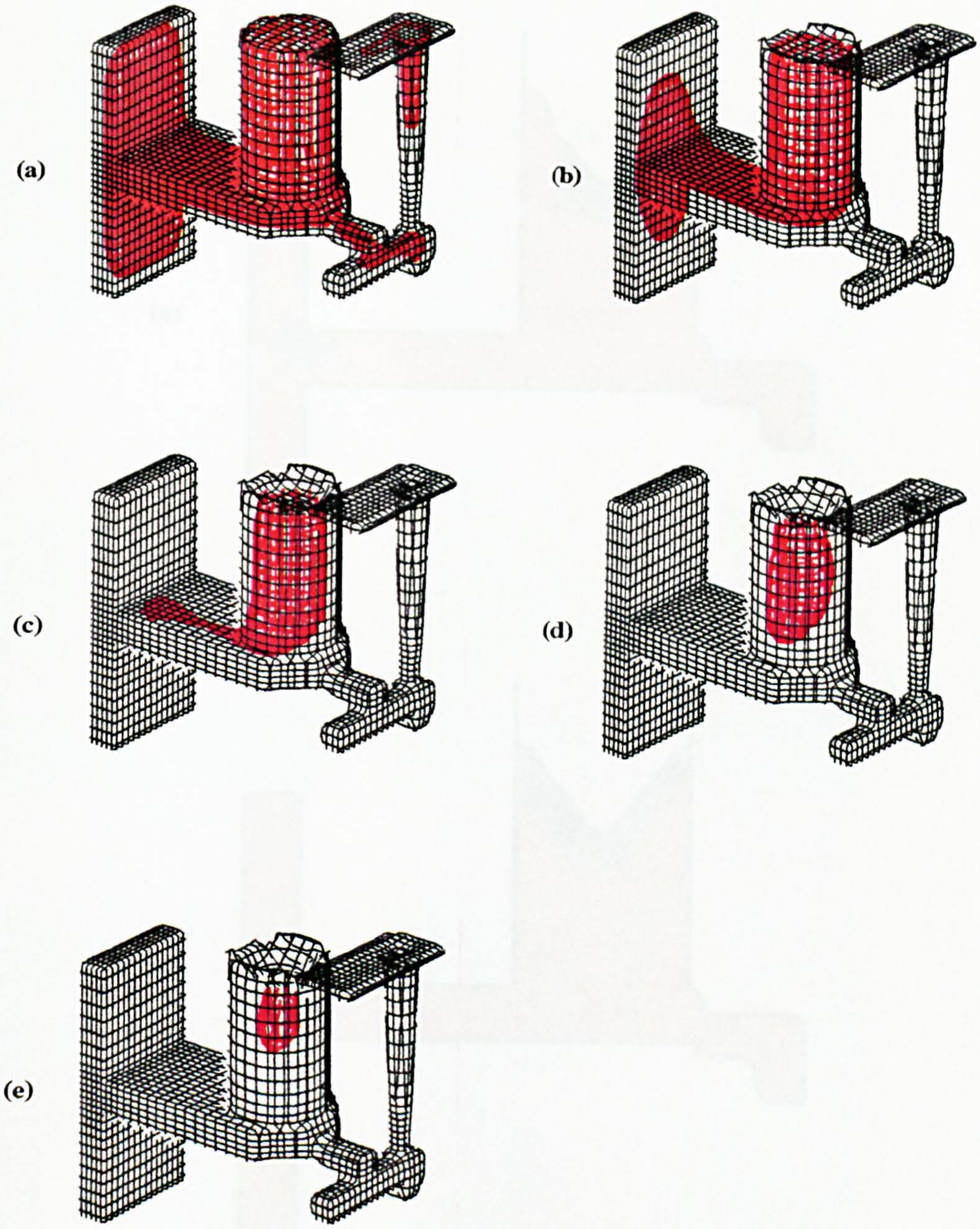


Figure 6.62. Simulation S5.11: 3-D plots of the overall fluid (black) and liquid (red) regions. (a) t=100 s, (b) t=200 s, (c) t=300 s, (d) t=400 s and (e) t=500 s.

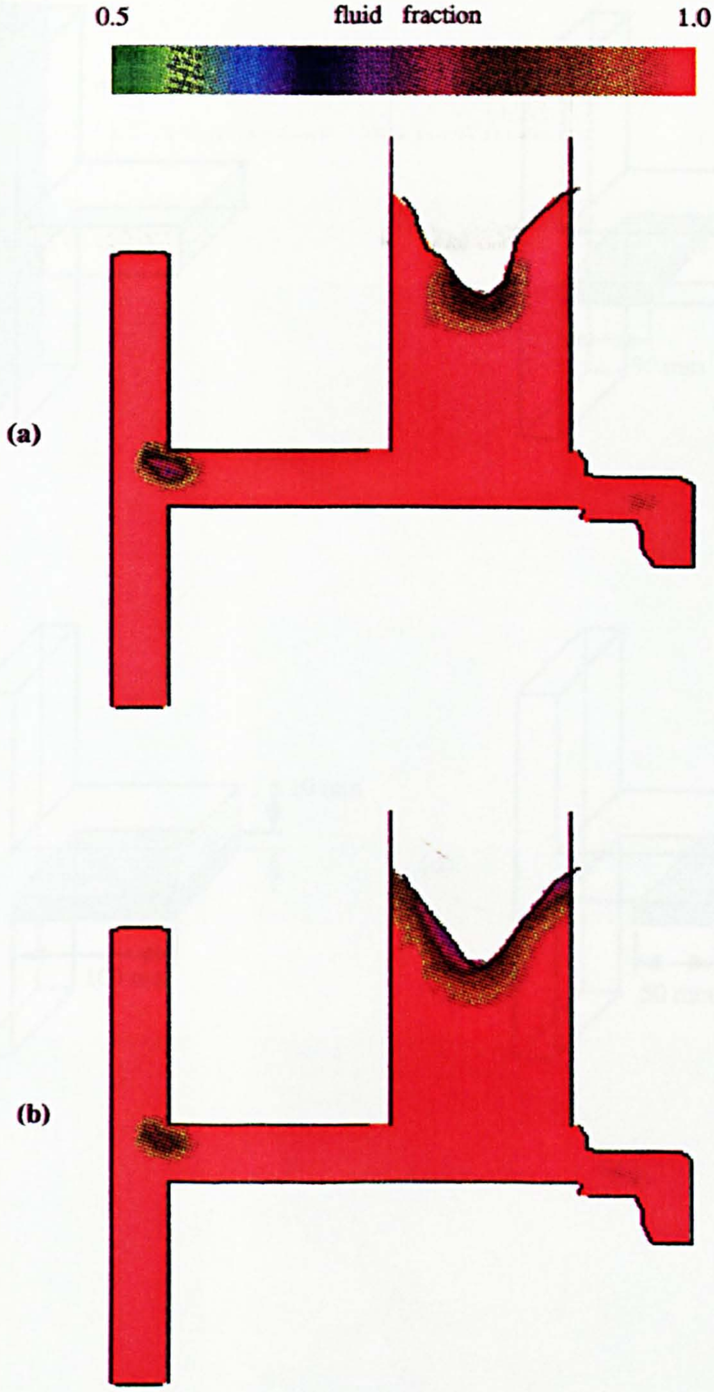


Figure 6.63. Fluid fraction distribution for an instantaneous filling with 30^{O}K superheat: (a) M1 model (S5.10) and (b) M2 model (s5.12).

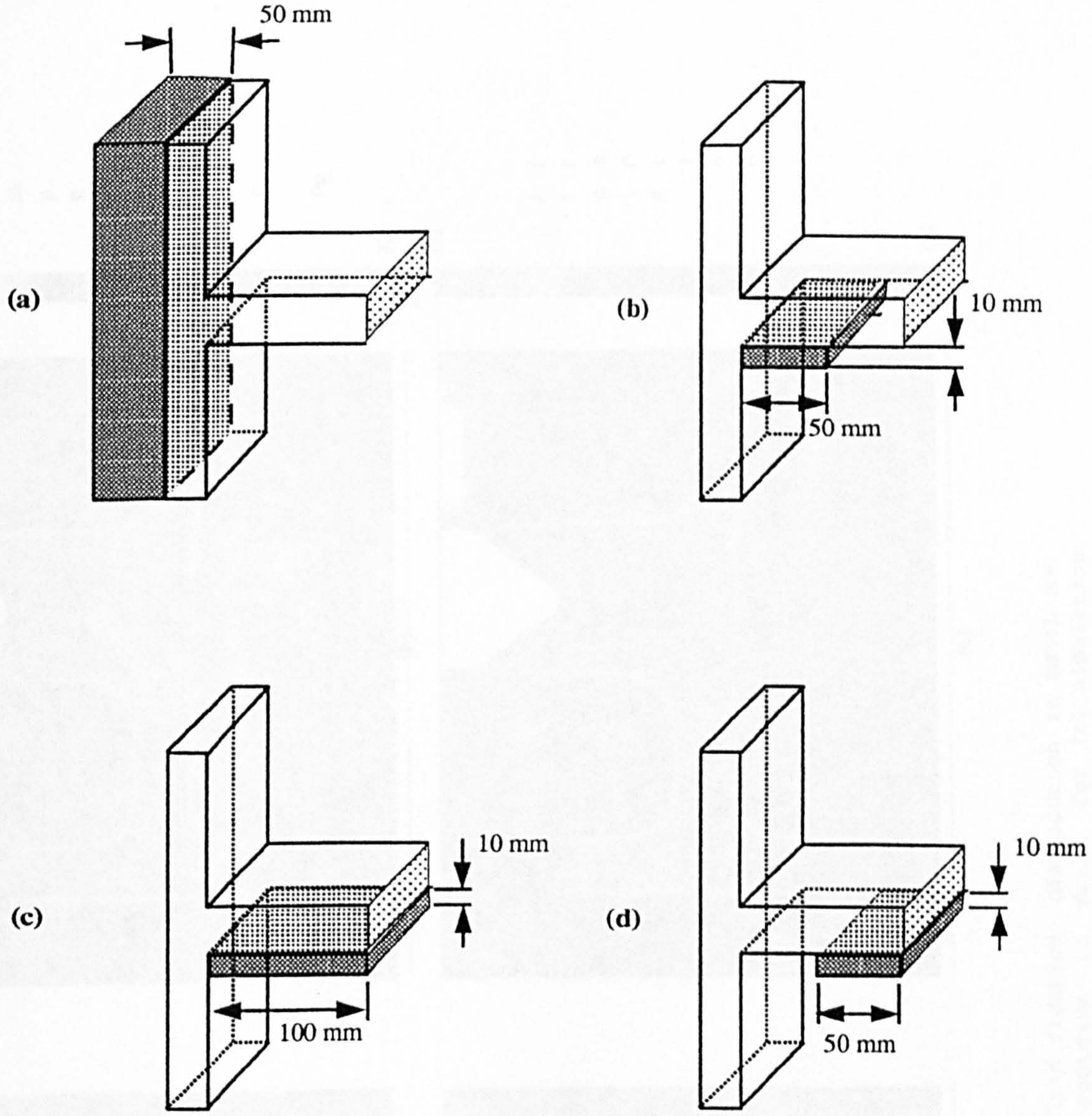


Figure 6.64. (a) The chill block in simulation S5.12, (b) insulator i1 in simulation S5.13, (c) insulator i2 in simulation S5.14 and (d) insulator i3 in simulation S5.15 for the T-shaped casting.

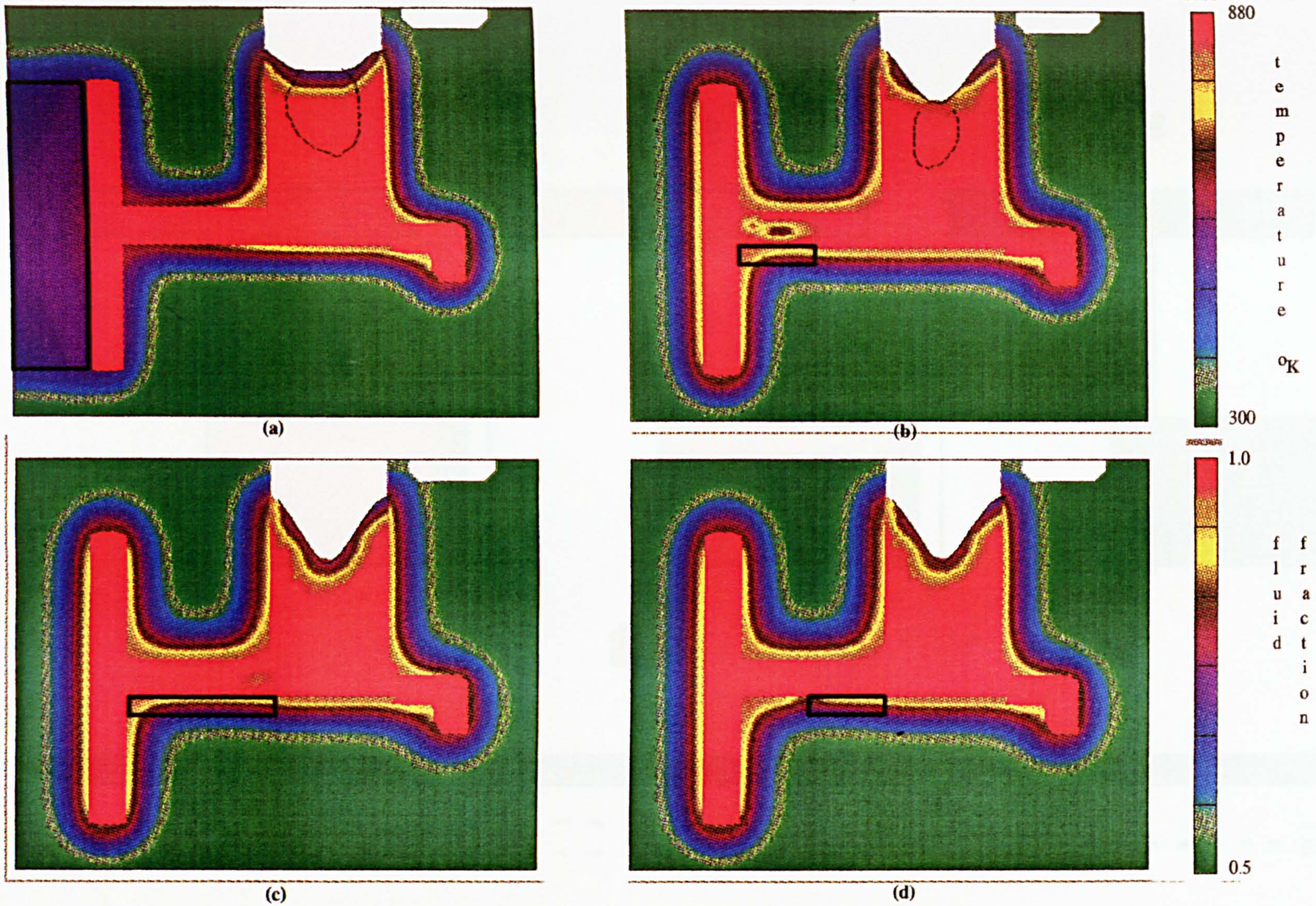


Figure 6.65. Predicted fluid fraction distribution in metal and temperature contours in mould for (a) simulation S5.12 with a chill, (b)-(d) simulations S5.13-S5.15, respectively, with insulators (see figure 6.64).

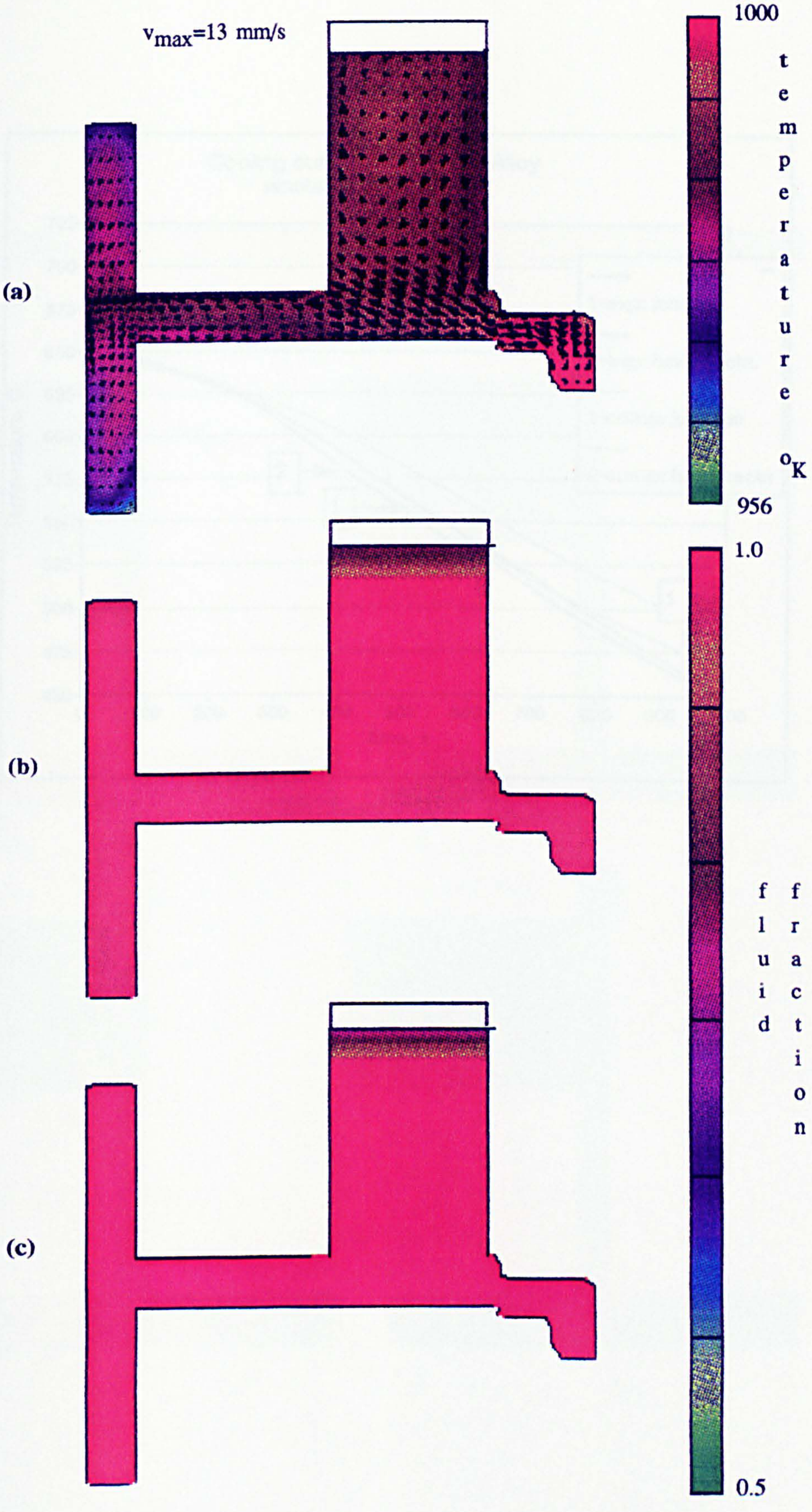


Figure 6.66. Al-4.5%Cu alloy, R=1.0. (a) Temperature distribution at the end of filling, t=7.5 s; fluid fraction distribution in the solidified casting at t=1000 s: (b) $f_{s,cr}$=0.67 and (c) $f_{s,cr}$=1.0.

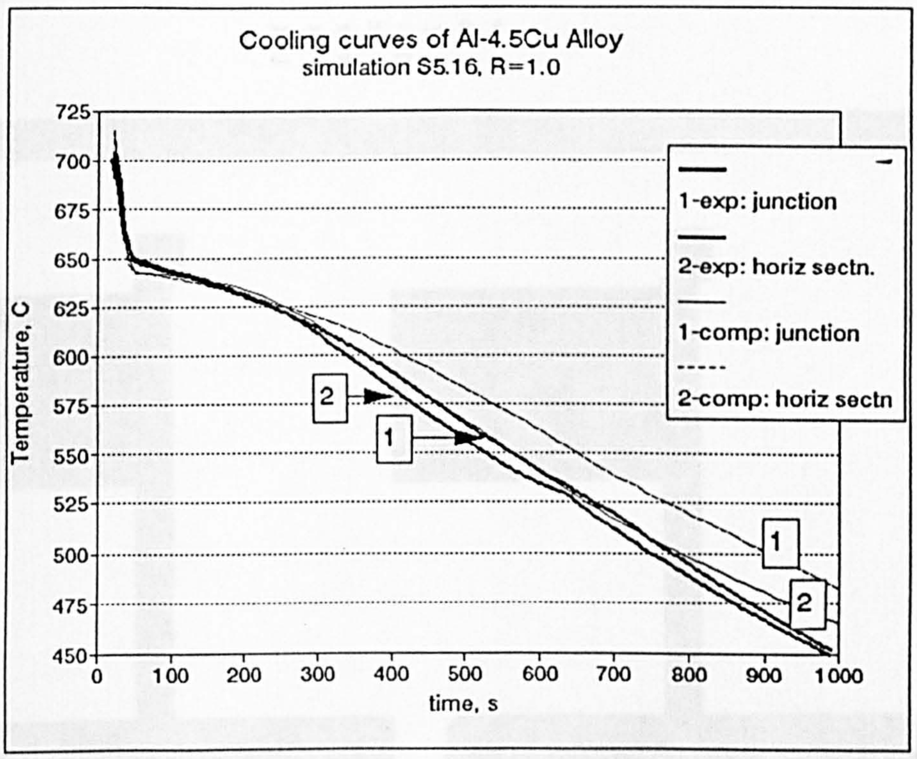


Figure 6.67. Cooling curves at two locations in the Al-4.5%Cu T-shaped casting (simulation S5.16 , R=1.0): predicted (thin lines) and measured (heavy solid lines).

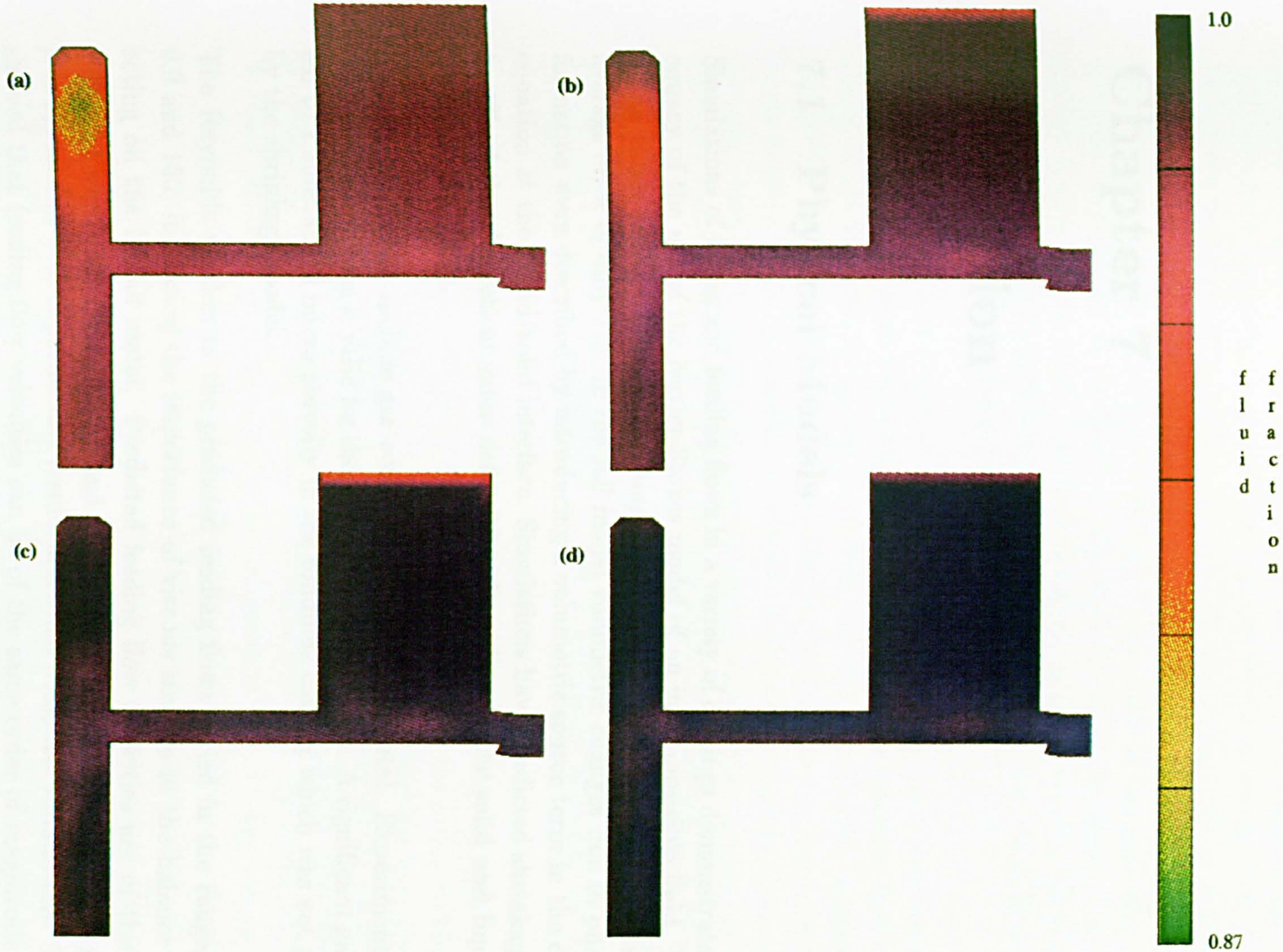


Figure 6.68. Fluid fraction distribution at the end of solidification for Al-4.5%Cu alloy, R=1.67, at t=1000 s: (a) $f_{s,cr}=0.67$, (b) $f_{s,cr}=0.8$, (c) $f_{s,cr}=0.9$ and (d) $f_{s,cr}=1.0$.

Chapter 7

Discussion

7.1 Physical Models

Simulations of filling and feeding flows in a variety of castings demonstrated the adequacy of the use of the Navier-Stokes model of an incompressible fluid. Predicted filling times for the T-shaped castings matched the experimental results with an average error of only 7%. In the full model volumetric changes due to phase transformation were described by introducing a volumetric source term in the continuity equation at the liquid/solid interface. Simulations have predicted shrinkage volume to 1% of the theoretical value defined by the values of the solid and liquid phase densities, Eq. (6.1).

The model did not include gas evolution in the liquid metal. Experiments showed that this assumption is valid for thoroughly degassed melts. A significant gas content led to a substantial micro-porosity in the solidified casting which was not predicted by the shrinkage model.

The Reynolds number in the predicted feeding flows varied in the range between 0.0 and 100, indicating the importance of viscous stresses in the balance of forces acting on the luquid metal. Predicted feeding flow velocities are of the order of $0.01 - 0.1\ mm/s$ for sand castings and $0.1 - 1.0\ mm/s$ for chill mould castings. This is in agreement with experimental results and other computer simulations. Tsai [133] showed that feeding flow velocities can be of the same order of magnitude as those occurring during metal thermal and solutal convection and predicted velocities of around $0.1 - 0.7\ mm/s$ for a 1% Cr steel. Maples and Poirirer simulated solutal convection in a Al-4.5%Cu alloy with velocities of up to $0.03\ mm/s$ [132].

Natural convection effects were not included in simulations to simplify the analysis. Thermal and solutal convection in feeders would increase heat exchange in the vertical direction resulting, in general, in a deeper primary cavity. The depth of the latter was largely underestimated in simulations of the aluminum-sand cylindrical casting and the T-shaped casting where a large feeder was present. Generally, the primary cavity depth in a feeder is defined by the ratio of the heat flux q_h through the sides and the top of the feeder to the heat flux q_v at the feeder's bottom. The smaller the ratio q_h/q_v is the smaller is the depth of the cavity. In the extreme case of $q_h = 0$ (a fully insulated feeder) the surface of the solidified metal in the feeder is flat and aligned with the horiztontal plane.

The error in predicted depth of the primary cavity was more significant in alloy T-shaped casting simulations (Figs. 6.17*b* and 6.67). This indicates that the largest source of the inaccuracy in alloy modelling was the solidification model. For the Al-4.5%Cu alloy the solute is heavier than the solvent so that liquid with smaller solute content would accumulate at the top of the feeder leading to an increase in the local values of T_l and T_s thus accelerating the solidification process at the feeder top. The latter would increase the effective value of q_h leading to a deeper cavity. The employed alloy solidification model did not include segregation and diffusion of the alloy components. Instead, a simplified approach was taken by using the lever rule. Further errors were introduced by the numerical implementation of the lever rule (section 2.8).

Free surface was treated as an adiabatic boundary neglecting radiative and convective heat losses into atmosphere. To study the effect of these losses on the primary cavity depth a test simulation was carried out for the cylindrical casting shown in Fig 6.3 in which the insulator was replaced by sand. (simulation S2.45 with the non-uniform mesh shown in Fig. 6.30*a*). This modification was introduced to approximate an additional heat loss due to radiation since the latter could not be modelled directly. The final cavity shape is shown in Fig. 7.1 and its depth differs from the experimental result by only 5% compared to the 21% error when the insulator was used (Fig. 6.30*b*). This result indicates that an inclusion of a non-adiabatic boundary condition and free surface may imporove the accuracy of the shrinkage model.

In all simulations material properties of metal solid and liquid phases and those of mould were assumed to be temperature independent. This assumption might be a source of additional errors in temperature predictions shown Figs. 6.31, 6.50 and 6.67. It is well known that thermal properties of sand are especially sensitive to temperature variations [134]. However, small differences between predicted and measured temperatures in Fig. 6.31, less than 10% in average, indicate that these errors are insignificant.

Metal/mould heat transfer coefficient in sand castings was defined assuming good contact between the two materials. This worked well while the metal was liquid as can be seen in Fig. 6.50 where temperatures at three locations for pure metal T-shaped casting are shown. After metal solidified in the T-junction and in the sprue, the difference between predicted and measured temperatures increase substantially. The measured cooling rates are smaller than the predicted suggesting that heat transfer coefficients at these locations decreased as metal solidified, perhaps, due to the formation of air gaps between the solidified metal and the mould. Predicted temperature at the bottom of the feeder stays close to the measured one until the end of the simulation. This indicates that no gap formed at this location, possibly due to the shear weight of the feeder. In the case of the alloy, differences in the measured and predicted temperatures are smaller and do not change abruptly after the metal solidified (Fig. 6.67). As in the case of the pure metal, it can be seen that predicted cooling rates at these locations are higher than the measures ones indicating that smaller air gaps, if any, have formed in the alloy casting, perhaps, due to the semisolid nature of the mushy zone during solidification.

Observations of the experimental T-shaped castings showed that the shape and volume of the internal cavity in the T-junction might be changed due to deformation of the solid metal (Fig. 6.11*b*). The deformation occurs because of the low pressure in the trapped liquid metal as it solidifies and shrinks. This deformation led to tearing of the solid metal and penetration of air into the metal through the permiable mould resulting in a cavity with charateristically smooth edges (Fig. 6.13*b*). These effects could not be predicted by the developed models since neither solid phase deformation nor air flow through the mould walls were included in the modelling. However, in most of the experimental castings these effects were small or were not present at all.

7.2 Numerical Modelling

The main part of the numerical description of liquid metal flow is the VOF method of tracking free surfaces. It is also employed in the shrinkage models to describe shrinkage defects in solidifying castings. Comparisons with the experimental resutls show that the VOF method is an efficient and accurate method for modelling both filling and shrinkage cavity formation.

Upwind differencing method for the approximation of the advection terms in Navier-

Storkes and energy equations, central differencing for the approximation of viscous terms and the SOR iteration technique for the solution of the coupled pressure-velocity equations provided sufficiently accuarate solutions for the liquid metal flow. This can be seen from the comparison of the predicted and measured filling times for the T-shaped castings (Fig. 6.42). The simplicity of the numerical implementation of the SOR iteration method allowed the source terms, associated with the volumteric shrinkage, to be easily incorporated into the solution algorithm. In general, the development of the shrinkage models on the basis of an existing general purpose CFD code proved to be highly efficient.

7.2.1 Accuracy and Efficiency of Calculations

The accuracy of the predictions of shrinkage volume were defined mainly by the value of the convergence criterion ϵ. However, in the case of the sand mould the depth of the primary cavity was always underestimated when compared with experimental results.

As was mentioned in Section 7.1, the primary cavity depth in a feeder is proportional to the ratio of the heat fluxes q_h and q_v. Several additional factors may have led to an underestimation of q_h/q_v in the simulations:

1. The interfacial heat flux at the vertical sides of the feeder may be underestimated due to the lack of mesh resolution. It was shown in Section 4.1 that the flux *is* dependent on mesh size and the related value of the open volume V_F in the interfacial cells (Figs. 4.3-4.5). That might apply to the T-shaped casting since the resolution in the casting is better than that in the feeder, as can be seen in Fig. 6.41. However, this dependence was removed in the cylindrical sand-mould casting simulations S2.1-S2.4 by employing uniform meshes. In this case all interfacial cells have the same size, therefore the heat transfer coefficient, Eq. (4.13), and truncation errors in the metal/mould heat flux calculation, Eq.(4.14), are the same in these cells. Here, even with the finest mesh (S2.4) the cavity depth was still 14% smaller than the experimental result (Figs. 6.25 and 6.26).

2. Finite mesh resolution introduces error in representing the free surface of the solidifying cell adjacent to the wall. This error affects the shape of the feeder cavity. Figure 7.2 shows that in solidifying surface cell, the fluid level drops lower than it would if there was a finer resolution in the horizontal direction to resolve the curvature of the free surface in the interfacial cell (shaded area).

The discrepancy is greater for pure metals since the liquid flow seizes only when $f_s = 1$ in the cell. If the free surface in the first cell freezes at δh_1 *lower* than the correct position, then the metal volume $\delta h_1 A_1$ is equally distributed between the rest of the liquid surface cells, where A_1 is the cell area in the horizontal plane. As a result the liquid level in these cells is *higher* than the correct value by

$$\delta h = \delta h_1 \frac{A_1}{A_0}$$

where A_0 is the total area of the remaining liquid surface cells in the horizontal plane. Further error is introduced when the second cell solidifies and so on. Since the total cavity volume is predicted accurately, the final cell to solidify, which actually defines the cavity depth, accumulates all the errors. The error becomes larger for cylindrical feeders where, as the freezing front moves towards the feeder axis, A_0 decreases as r decreases, therefore, δh grows as $1/r$, where r is the distance from the axis.

The mesh related factors can be decreased by refining the mesh and using a uniformly spaced mesh.

3. An error could result from assuming conduction heat transfer between the centre of liquid interfacial cells and the interface (Eqs. (4.9), (4.10) and (4.22)-(4.24)). Hoadley *et al* [131] showed that convective heat transfer resulting from liquid flow may be significant. The ratio of the heat transfer by convection to conduction is described by Rayleigh Number [4]

$$Ra = \frac{\rho^2 L^3 C \beta \Delta T}{k}$$

where L is the characteristic length, β the coefficient of thermal volume expansion, and ΔT the temperature difference across L. For a typical cell dimension of $L = 5\ mm$, $\Delta T \approx 100^o K$ and for pure aluminium $N_{Ra} < 0.1$. This is comparable with the error, 14%, in the shrinkage cavity depth prediction. An inclusion of the convective heat transfer would increase the cavity depth since it occurs mainly at the vertical interfaces (because feeding and thermal convection occur mainly in the vertical direction), thus, increasing the ratio q_h/q_v.

The position and size of the secondary internal cavities were predicted with good accuracy, especially for pure metals. These are less dependent on the accuracy of free surface description but an accurate calculation of the interfacial heat fluxes is also crucial.

An important feature of the secondary internal cavities in the T-junction is their dependence on the filling time as shown in Fig. 6.49. The slowest filling time of 22.5 s ensures a *sound* casting. This result supports the general idea that the final casting quality depends on the filling parameters. However, this dependence was not observed in the experiments where variation in the size of the secondary cavity for the same filling rate was comparable, or even larger, to the predicted variation for different filling rates.

Despite improvements made by modifying the original heat transfer algorithm of FLOW-3D (see Section 4.1), the mesh dependence of the interfacial heat flux is a significant shortcoming of the present model. The use of a uniformly spaced mesh is not the ultimate solution since the results also depend on the value of V_F in the interfacial cells and the variation of the latter is inevitable for curved mould surfaces. The use of very fine meshes is restricted by hardware limitations. At present the total cell number of 50,000 appears to be the limit.

A solution to the mesh dependence problem could be the use of analytical functions to describe the heat flux at the interfaces [66,93,95,96]. However, this remains restricted by the small number of available analytical solutions.

The deficiencies of the heat transfer algorithm are clearly shown in the comparisons of the temperature histories (Figs. 6.31 and 6.50). The solidification time for the aluminium-sand cylindrical casting is about 26% above the experimental value of 754 s for the finest mesh (simulation S2.4). Additional errors are introduced by the use of constant metal and mould thermal properties and constant interface thermal resistances. However, these may not be crucial for accurate shrinkage defect predictions as has been shown for the chill mould casting (simulations S3.1-S3.4 in Fig. 6.34). These show that it is not the absolute value of the heat flux, but its variation along the interface that largely defines the position and size of the cavities in the solidified casting. The latter also indicates the necessity of modelling the formation of a gap between the mould and the metal. A gap of 0.05 mm can lead to an increase in the interfacial thermal resistance by up to two orders of magnitudes as shown by Hou and Pehlke [24]. These authors also demonstrated that temperature prediction may be improved if an experimentally measured time-dependent heat transfer coefficient is used in the calculations.

A problem for alloy solidification modelling was posed by the use of the critical fraction of solid as the feeding criterion. It was shown in simulations S5.18-S5.21 in Fig. 6.67 that the size and even the occurrence of the internal macro-porosity in the T-junction was sensitive to the value of $f_{s,cr}$. These simulations were expected to be less accurate since the Al-4.5%Cu alloy has a wide freezing range and the whole casting was in a mushy state during most of the solidification time. This condition magnified the deficiencies of applying the model to alloy solidification. In addition, the use of the lever and Scheil's model in the computational cells in the form of Eqs. (2.64) and (2.65) was not consistent with the assumptions of the two solidification models.

One of the main difficulties in porosity prediction in an alloy is the occurrence of micro-porosity. The shrinkage models developed by the present author were not designed to simulate micro-porosity. However, micro-porosity is an inevitable consequence if $f_{s,cr} < 1.0$. The value of $f_{s,cr}$ defines the distribution of the shrinkage volume between the macro- and micro-porosity. A way around is to reduce the density of the solid phase by assuming uniform micro-porosity distribution and set $f_{s,cr} = 1.0$ which implies that all the shrinkage volume goes now into macro-porosity. If the expected porosity is $r\%$, then the effective solid phase density is

$$\rho_{s,eff} = \rho_s \cdot (1 - r/100)$$

This requires *a priori* knowledge of micro-porosity, and it has to be uniformly distributed.

A more accurate description of feeding in the mushy zone is the use of the critical value of the solid fraction gradient, $\nabla f_{s,cr}$ [112]. It takes into account that feeding is easier along the direction of the dendrite growth than across it. However, the value of $\nabla f_{s,cr}$ may vary between alloys, mould geometries and even cooling rates. Clearly, a more sophisticated, detailed model of alloy solidification and feeding is required.

As was shown in Section 7.1, predicted velocity magnitudes in the feeding a flow are consistent with existing numerical and experimental data. The magnitude of the average feeding flow velocities are defined mainly by the solidification time and the total volumetric shrinkage (or the ratio ρ_l/ρ_s). Since the latter is fixed for a particular metal, the accuracy of the velocity predictions depends on the accuracy of the solidification rate prediction. The mould geometry also plays a part in defining the velocities. For the chill mould casting S3.1, for example, velocities went up to 10 mm/s for a short period during solidification when feeding occurred through a narrow passage in the metal (Fig. 6.33).

7.2.2 Drag Force

The drag force in the form of Eqs. (3.10) and (4.51) or (4.52) does not describe a real physical force but arises from the fact that numerical equations are applied to cells which contain a solid-liquid mixture and the solid phase does not flow. Strictly speaking this approach is valid only for pure metal solidification since for alloys the solid phase crystals can float at low solid fraction values, and furthermore, the frictional drag force in the mushy zone is also important. The latter is not taken into account by the model to avoid convergence problems in the high drag regions.

The use of the drag force in calculating flow losses due to solidification is not satisfactory also for other reasons:

- The drag coefficient, Eq. (4.51), does not ensure that the solid phase remains frozen in space. For example, $K = 0$ when $df_s/dt = 0$ even if $f_s \neq 0$.

  The enthalpy advection algorithm is closely related to maintaining a zero solid phase velocity (see Section 7.2.2). In the M1 model only the liquid enthalpy content is advected. No account is taken, however, of the amount of liquid phase left in a cell. Thus, more liquid enthalpy can be advected out of a cell than it may provide[1].

- Pressure, gravity and viscous forces in the momentum Eqs. (3.7)-(3.9) are applied to the full cell volume while they should affect only the liquid phase and their influence on the cell momentum should be reduced accordingly. The effect of the drag force is thus diminished, especially when the f_s value is close to unity.

- There may be situations where the drag coefficient in nearly solid cells becomes very high leading to excessive iterating (see Section 7.2.3).

- Finally, even for pure metals there may be significant flow losses due to shear srtesses at liquid/solid interface in the regions of high values of the solid fraction. The expression for the drag force could be modified to include the viscous friction effects but it is not clear how to do it efficiently in a general way in

[1] A test could be introduced for the amount of enthalpy to be advected through a cell face, δH_{adv}, in *surface* cells:

$$\delta H_{adv} = min(u \cdot A \cdot H_l \cdot \Delta t,\ H_l \cdot V_l)$$

where u is the velocity, A the face area, H_l the liquid phase enthalpy and V_l the volume of the liquid left in the cell. It is inappropriate though for a full, internal cell due to the fluid continuity and incompressibility conditions. These imply that $\delta H_{adv} = u \cdot A \cdot H_l \cdot \Delta t$ *must* be advected between neighbouring full cells.

three dimensions, even if the solidification front is assumed to be a plane interface in each cell.

The situation is even more complicated for alloy solidification where the solidification front cannot generally be represented by a surface.

A way to remove the deficiencies of the employed drag force formulation may be found in extending the FAVOR method to take into account cell blockages by solid phase in a manner similarly to that for the mould. This can be accomplished by the following transformation of the fractional open volume and face areas:

$$V_{F,m} = (1 - F\,f_{sv})\,V_F, \qquad A_{ab,m} = A_{ab}\,\frac{2\,(1 - F_a\,f_{sv,a})(1 - F_b\,f_{sv,b})}{2 - F_a\,f_{sv,a} - F_b\,f_{sv,b}} \tag{7.1}$$

where A_{ab} is the open area fraction at the common face of neighbouring cells a and b, f_{sv} is the volumetric solid fraction function and index m means a modified variable. Similarly, a 'modified' fluid fraction F_m can be introduced

$$F_m = F\,(1 - f_{sv}) \tag{7.2}$$

to describe the remaining liquid in the cell which can flow. The modified variables $V_{F,m}$, $A_{ab,m}$ and F_m have the same meaning as the original variables but relate to the liquid phase only while the solid phase is treated as a part of the solid structure of the mould.

Eqs. (7.1) and (7.2) should be applied to advection terms. The original variables must still be used for heat conduction and transfer, and the enthalpy still refers to the full, liquid+solid, fluid volume in a cell. In other words, a formal implementation of this method to describe the effect of the solidification on the fluid flow would require a combined use of the original and modified fluid, volume and area fractions.

This method has the following advantages:

- There is no need for the atrificial drag force in the form of Eq. (4.52) because cell velocity in the modified method is not the mixture velocity u_m but the actual liquid phase velocity u_l (Fig. 4.8*a*). Therefore, possible convergence problems in the high drag regions are removed. The momentum equations become more consistent since all the terms refer to the liquid phase only.

- Since velocities and enthalpy advection terms are calculated with respect to the amount of the liquid phase left in the cells, the solid phase is ensured to be frozen in space.

- For pure material solidification, when the solid/liquid interface can be represented by a surface, stresses between the two phases can be calculated by the same algorithm as for the shear stresses between the liquid and the mould walls. Therefore, there is no need for a frictional drag force. For alloys an additional drag force, accounting for flow losses in the mushy zone, should be developed and introduced to momentum equations.

There will be implications for the time step size limitation since the latter depends on the values of V_F and A (Eq. (3.14)). However, the time step can only decrease by half, as follows from Eq. (7.1):

$$\frac{V_{F,a,m}}{A_{ab,m}} = \frac{2 - f_{sv,a} \cdot F_a - f_{sv,b} \cdot F_b}{2\,(1 - f_{sv,b})} \cdot \frac{V_{F,a}}{A_{ab}} \geq \frac{1}{2} \cdot \frac{V_{F,a}}{A_{ab}}$$

if in cells a and b: $0 \leq f_{sv} < 1$ and $0 \leq F \leq 1$.

A disadvantage of Eqs. (7.1) and (7.2) is the assumption that the solid phase it attached to the mould walls and cannot move.

7.2.3 Enthalpy Method and Enthalpy Advection

One of the undesirable consequences of using the enthalpy method, as shown in Fig. 2.5*b*, is the temperature fluctuations on both sides of the solidification front for isothermal phase transformation. This in turn causes fluctuations in the solidification rate and, since the latter defines the shrinkage rate and the feeding flow, these two have discontinuities, as shown in Fig. 7.4 in the metal volume and mean kinetic energy histories for simulation S3.1.

The fluctuations arise from the fact that the melting temperature T_m in a solidifying cell is assigned to the cell centre instead of the solidification front in the cell, thus distorting temperature gradients in the vicinity of the cell. This error may be minimised by employing a linear interpolation (not *extra*polation) of temperatures between the solidification front and the centre of a neighbouring cell to obtain the cell-centred value of the temperature in the cell that contains the front (Fig. 7.3). This is akin to setting pressures in the surface cells using Eq. (2.49). However, it may require substantial numerical effort since the orientation of the solidification front in the cell must be defined in three dimensions.

The use of such interpolation will alter the unique correspondence of the cell-centred temperature to the cell enthalpy, thus complicating the procedure of defining the cell-centred temperature from the value of the enthalpy. This difficulty could be overcome by treating the two phases as separate fluids so that two temperatures and enthalpies can be used in one cell. Substantial programming effort and computer memory would then be required to introduce this modification as one cell could then contain four materials: liquid and solid phases, mould and void.

The advection of only the liquid phase is ensured by using the enthalpy equation in the form of Eq. (5.20). An alternative form of this equation is

$$\rho_l \cdot \frac{\partial F H}{\partial t} = \frac{\rho_l}{\rho V_0}\left(-\rho_l \int_{\Sigma} H_m v_n \, d\sigma + q_{\Sigma}\right) - \frac{1}{\rho}\frac{\partial \rho}{\partial t} \rho_l F H \qquad (7.3)$$

in which the mixture cell-centred enthalpy H_m is present in the advection terms rather than the liquid phase enthalpy H_l. Eq. (7.3) was used to simulate the pure iron casting (S3.21) with the result shown in Fig. 7.5. It can be seen that the solid phase is 'stripped' by the metal flow from the mould walls and carried into the bulk of the casting which is clearly is a non-physical result.

7.2.4 Convergence

Several convergence 'failures' have occurred cylindrical casting simulations using the M1 model, S3.1-S3.4, as shown on a diagnostics print-out in Fig. 7.6[2]. A general increase in iterations per cycle began at $t \approx 30$ s (lines > 25). Most of the iteration failures occurred when liquid in the upper half of the casting froze to leave a narrow passage through which the rest of the liquid was fed (Fig. 6.32). Factors leading to an increase in iteration can be summarised as follows:

1. Velocities in the liquid cells along the passage are two orders of magnitude larger than those in the liquid bulk due to the incompressibility of the liquid ($\sim$ 10 mm/s compared to $\sim$ 0.1 mm/s, the difference magnified by the cylindrical geometry). Therefore, the value of ϵ may appear to be too small for these cells to achieve reasonable convergence rate (see also Section 7.2.4).

2. A liquid cell may change from 'internal' cell to 'surface' cell accompanied by an abrupt change of pressure in it. Consider a partially solid cell A in Fig. 7.7a. The neighbour cell B is fully liquid and cell C is completely solid. As solidification proceeds the fluid level in cell B goes down, so does its pressure.

[2] As mentioned in the footnote in Section 5.2.3, 'iteration failures' are diagnosed when the iteration number in a cycle reaches a predefined maximum. These failures mean that the convergence is slow, not that the solution does not converge.

By the time cell B is nearly empty at time step n pressure in it is, according to Eq. (2.49),

$$p_B^n \approx p_0 - 0.5\,\rho_l\,G_z \cdot \Delta z \tag{7.4}$$

where p_0 is the void pressure, G_z gravity in the z-direction and Δz the cell size in the vertical direction. Pressure in cell A is found iteratively even if there is little fluid left in it since cell A does not have empty neighbours and is treated as an incompressible internal cell. Its pressure then should not exceed p_B^n because cell B feeds cell A:

$$p_A^n \leq p_B^n \tag{7.5}$$

Suppose cell B is emptied during the next time cycle $n+1$ (Fig. 7.7*b*). Then cell A becomes a surface cell with *vertically* orientated free surface since cell B is the only empty neighbour (see Section 3.3.4) and

$$p_A^{n+1} = p_B^{n+1} = p_0$$

because $G_x = 0$. According to Eqs. (7.4) and (7.5), pressure in cell A increased in one time step by

$$\Delta p_A = p_A^{n+1} - p_A^n \approx 0.5\,\rho_l G_z \cdot \Delta z \tag{7.6}$$

creating a pressure pulse which may result in velocities of up to 1 m/s at cell faces A-D, D-E and E-B. The latter leads to fluid entering cell B again at time step $n+2$ and the situation repeats. Often it results in pressure iteration failures at time steps $n+1$ and $n+3$ which can be seen in the print-out in Fig. 7.6 (lines 19-20, 35-36 and 38-39).

Such flow fluctuations sometimes affect the shape of primary shrinkage cavities by introducing ridges along its surface. This can be seen in Figs. 6.30 and 6.32, as well as in Fig. 6.39 for the boot-shaped casting (simulation S4.1). The mean kinetic energy in Fig. 6.40*d* has several sharp steep peaks. Each peak corresponds to a velocity increase in a surface cell due to the pressure change given by Eq. (7.6). The maximum mean kinetic energy is equal to 6.2×10^{-7} J/kg, which is three orders of magnitude larger than the average value. Substantial number of interations was therefore required for convergence, with the average around 100.

The increase of pressure in cell A would not result in the increase of velocities only if cell A was completely solid by the time cell B was empty. A simulation (S4.21) was carried out to compare with simulation S4.1. The only difference was the use of Eq. (7.3) in the former and Eq. (5.20) in the latter. The results of the simulation S4.21 are shown in Fig. 7.8. The mean kinetic energy plot has fewer spikes and the maximum value is 8.5×10^{-8} J/kg which almost ten times smaller than that in the simulation S4.1. Convergence rate also improved with the average iteration number of around 60. A possible explanation to the difference in the results is that when Eq. (7.3) was used then the metal in cell A in Fig. 7.8 was more likely to be fully solid by the time cell B was empty. In this case the drag force in cell A was effectively infinite and no velocity fluctuations occurred.

3. A similar effect on the fluid flow is produced when an 'internal cavity' cell, which contains both luquid and void and has a fixed pressure, $p = p_{cr}$, becomes fully empty (cell A in Fig. 7.9) during time step $n+1$. According to Section 5.2.3, when cell A turns from an 'internal cavity' cell into an empty cell its pressure does not change

$$p_A^n = p_A^{n+1} = p_{cr}$$

At $t = t^n$ pressure in cell B is

$$p_B^n \approx p_{cr} + \rho_l \cdot G_z \cdot \Delta z$$

At $t = t^{n+1}$ cell B becomes a surface cell and its pressure is calculated as

$$p_B^{n+1} = p_{cr} + 0.5 \cdot \rho_l \cdot G_z \cdot F_B^{n+1} \cdot \Delta z$$

If $F_B^{n+1} \approx 1.0$, which is usually the case, then p_B decreases by a value close to $0.5\rho_l G_z \Delta z$ in one time step. This generates a sudden upward flow which can put some fluid back into cell A. The latter will cause a reverse change of pressure in cell B. These fluctuations can proceed for several time steps requiring additional iteration effort. This can be seen in Fig. 7.6 (lines 42-52).

Numerical problems described in items 3 and 4 are the main causes of slow convergence rate during shrinkage modelling using the M1 model.

7.2.5 Convergence Criterion ϵ

To test the influence of the convergence criterion value, ϵ, on the accuracy of the solution and to define its optimum value, a series of simulations was carried out using the full shrinkage model for the cylindrical casting S2.2 in which the insulator is removed so that a free surface is present throughout the simulation (S2.21-S2.26)[3].

[3] As was described in Section 2.3.2, at each time step pressure iterations are considered converged if the modulus of the velocity divergence, $divv$, is smaller than ϵ in every mesh cell.

The value of ϵ was varied between 0.0001 s^{-1} and 1.0 s^{-1}. Fig. 7.10 shows the volumetric error and the average iteration number as functions of ϵ. The optimum value of the convergence criterion in these simulations appears to be close to 0.001 s^{-1} when the volumetric error is less than 0.23% of the total volume and the average iteration number is only 7. Fig. 7.11 shows predicted distributions of the fluid fraction function. A lack of convergence for large values of ϵ resulted in a 'porosity' region along the casting axis. A reasonable result, both in terms of the volumetric error (< 0.7%) and the free surface shape, wass given with the value of ϵ up to 0.01 s^{-1}.

For comparison, the results of the same simulations but with no shrinkage, *i.e.* $\rho_l = \rho_s$, are shown in Fig. 7.12 (simulations S2.31-S2.36). In contrast to the shrinkage simulations, the volumetric error here is above 65% for $\epsilon = 0.1$ s^{-1}, resulting in a virtually complete emptying of the mould. It can be seen from Fig. 7.10 that in the absence of the shrinkage induced flow a substantially smaller value of the convergence criterion is required to obtain a solution which accurate in terms of the conservation of the fluid volume.

For an internal cell of volume ΔV the volumetric error after the convergence has been reached is

$$\Delta V_{err} = div\,\mathbf{v} \cdot \Delta V \leq \epsilon \cdot \Delta V \tag{7.7}$$

Generally, the sign of $div\mathbf{v}$ varies from cell to cell reducing the net volumetric error in the domain.

For transient flow problems, *e.g.* mould filling, an approximate rule of thumb for choosing the value of ϵ is [83]

$$\epsilon = 0.0002 \cdot \frac{U}{L} \tag{7.8}$$

where U is the magnitude of the velocity variation across the domain and L is the length scale on which this variation occurs. Generally, the value of ϵ also depends on the time step size and geometrical features of the flow. The magnitude of the velocity error, δu_{err}, due to iteration in a cell of size Δx can be estimated as

$$\delta u_{err} \leq \Delta x \cdot \epsilon = 0.0002 \cdot \frac{\Delta x}{L} U \tag{7.9}$$

or, since $\Delta x/L$ is approximately equal to the inverse of the cell number N across the domain,

$$\delta u_{err} \approx 0.0002 \cdot \frac{U}{N} \tag{7.10}$$

If for example $N = 10$ than the velocity error at each time step is within 0.002% of the overall velocity variation in the domain.

If Eq. (7.8) is applied to the cylindrical casting simulation S2.21, where $U \approx 0.05\ mm/s$ and $L = 150\ mm$ (the initial depth of the melt), then $\epsilon = 7.0 \cdot 10^{-8}\ s^{-1}$. This value is much too small for an efficient calculation as can be seen from Fig. 7.10: for $\epsilon < 0.01\ s^{-1}$ the gain in accuracy in terms of the volumetric error, as compared to the results for $\epsilon = 0.1\ s^{-1}$, is negligible while the average iteration number increases steeply. The reason for this is that it is not the velocity variation with time that defines the total iteration. Feeding velocities are mainly defined by the solidification rate and the latter varies very slowly. Pressures in the casting, however, do vary substantially as the free surface moves downwards to reduce the effective pressure head. For example, for casting S2.45 the pressure head changed by $h = 50\ mm$ (h is the predicted depth of the shrinkage cavity) during solidification and the corresponding change in pressure, Δp, is

$$\Delta p = \rho_l h G \approx 1,250\ kg/m\,s^2$$

An *effective* velocity U_{eff} can be defined as

$$U_{eff} = \sqrt{\frac{\Delta p}{\rho_l}} = 0.7\ m/s \qquad (7.11)$$

Substituting U_{eff} into Eq. (7.8) gives $\epsilon \approx 0.0009\ s^{-1}$ which is more in line with the results of the ϵ-test in Figs. 7.10 and 7.11.

Summarising Eqs. (7.8)-(7.11) it may be concluded that for pressure dominated flows, that is when

$$\frac{2\Delta p}{\rho U^2} \gg 1$$

where Δp is the characteristic pressure change, the convergence criterion can be estimated as

$$\epsilon = 0.0002\frac{1}{L} \cdot \sqrt{\frac{\Delta p}{\rho_l}} = 0.0002\frac{1}{L} \cdot \sqrt{hG} \qquad (7.12)$$

where h is the change of the free surface position due to shrinkage.

For simulation S3.1 with $h = 0.25\ m$ and $L = 0.6\ m$, Eq. (7.12) gives $\epsilon = 0.00053\ s^{-1}$. This is close to the actual value used in the simulation, $\epsilon = 0.001\ s^{-1}$. Eq. (7.12) also explains the increase in the iteration count when pressure fluctuations occurred at the free surface in this simulation, as described in Section 7.2.3. For example, the instantaneous pressure change given by Eq. (7.6) with $\Delta z = 12\ mm$ corresponds to $U_{eff} \approx 0.25\ m/s$ requiring the value of $\epsilon = 0.0042\ s^{-1}$ which is 4 times larger than the employed value.

7.2.6 Time Step Size Limitations

Whilst the heat transfer and the free surface time step size limitations dominate the solidification stage, the CFL criterion, Eq. (3.14) controls Δt during the filling stage. Figs. 7.14*a*, *b* show the variation of the time step size during the 7.0 and 22.5 seconds of filling for the T-shaped casting (S5.1 and S5.3), respectively. The average velocities during the fillings were 0.6 m/s and 0.2 m/s (the maximum velocity was at the choke at 2 m/s in both cases), respectively, while the minimum cell size is 5 mm.

The time step restriction during filling is a significant factor: for the T-shaped casting simulation S5.1 the filling time was almost 60 times smaller than the solidification time (7.5 s and 475 s) but the CPU time for the filling stage simulation was only 5.2 times smaller than that for the solidification and shrinkage (Fig. 6.19).

The shrinkage model was not used during filling since there was no significant solidification at this stage. When a shrinkage model was employed the increase in the CPU time per time step due to the additional calculations was less than 5%.

7.3 M1 and M2 Shrinkage Models

The M1 model is fundamentally more correct than the simplified M2 model as the latter does not account for fluid flow. The full shrinkage model is also more general since fluid flow and free surface shape are calculated dynamically without any *apriori* assumptions. Besides, the mathematical and numerical formulation of the M1 model make possible its further development to include, for example, gas evolution.

However, differences in shrinkage cavity predictions of the two models in the casting cases that were considered in the present work are small. The similarity of the resutls may be explained, in the first place, by the validity of the assumptions made in the M2 model listed in Section 5.3, that is that the formation of shrinkage cavities is mainly governed by gravity. Secondly, redistribution of heat in the melt was mainly due to metal/mould heat transfer and conduction while advective heat transfer played a minor role. Therefore neglecting the latter by the M2 model did not introduce significant errors in shrinkage cavity predictions, cooling rates and solidification times. Moreover, cavities predicted by the two models for the chill mould casting are almost identical which may be explained by the fast cooling rate in the chill mould so that the relative importance of the advection of heat was even smaller than that in the sand castings.

The effect of the absence of the advective heat transfer can be seen in a shallower primary cavity depth predicted by the simplified model in the sand castings simulations (*e. g.* Figs. 6.26 and 6.53).

The differences are greater in the numerical efficiency of the two models. The gain in speed of the M2 model is first of all due to the use of a larger time step size which is defined by the heat transfer process (Eq. (2.76)). For the Ml model the time step is defined by the free surface stability limit (Eq. (2.77)) during most of the simulation time. The ratio of the typical time steps sizes used in M2 and Ml models varies from case to case depending on the mesh and properties of the metal and mould. The maximum ratio was 540 for the chill casting simulations. Fig. 7.13 shows a comparison of time step sizes used by the two models for simulations S5.5 and S5.22.

Further speed up of M2 model calculations is achieved becasue only the energy equation is solved in the model, while the Ml model includes the solution of the full system of the continuity, momemtum and energy equations. In the Ml model it is often difficult to predict how much of the numerical effort will be required to reach convergence. It is, therefore, extremely advantageous that fluid flow solution is excluded from the M2 model, especially when it does not significantly affect accuracy.

Finally, since no free surface advection is involved in the M2 model and the liquid metal free surface is always assumed to be horizontal, the predicted free surface shape is smoother than those given by the full model as was shown, for example, in the boot-shaped casting simulations S4.1 and S4.11 (Figs. 6.38 and 6.59).

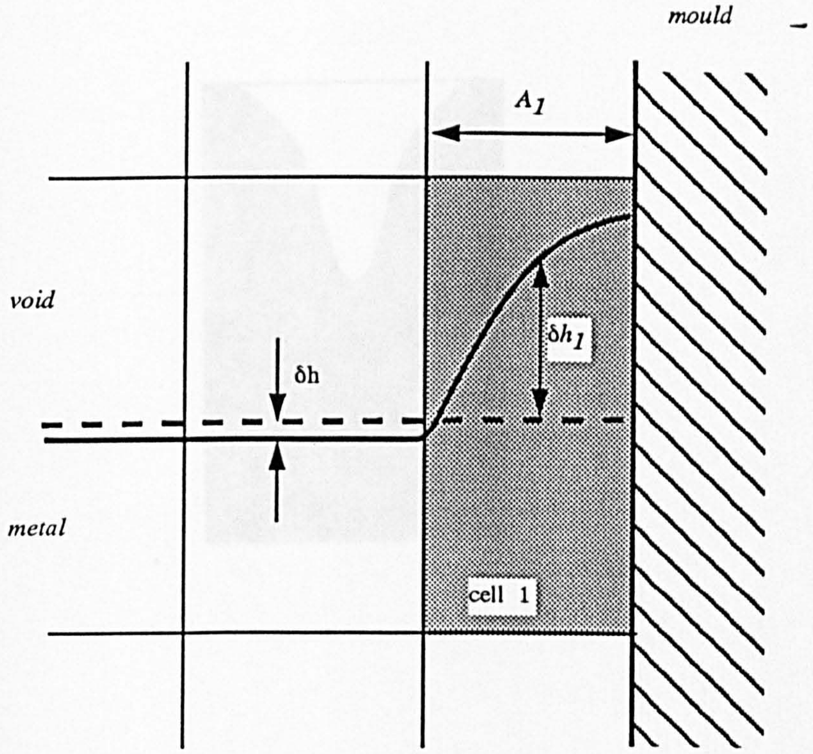


Figure 7.1. When cell 1 (shaded with grey) solidifies the calculated fluid level in it (dashed line) is lower than the actual level (solid line) by δh_1 due to truncation errors in representing free surface.

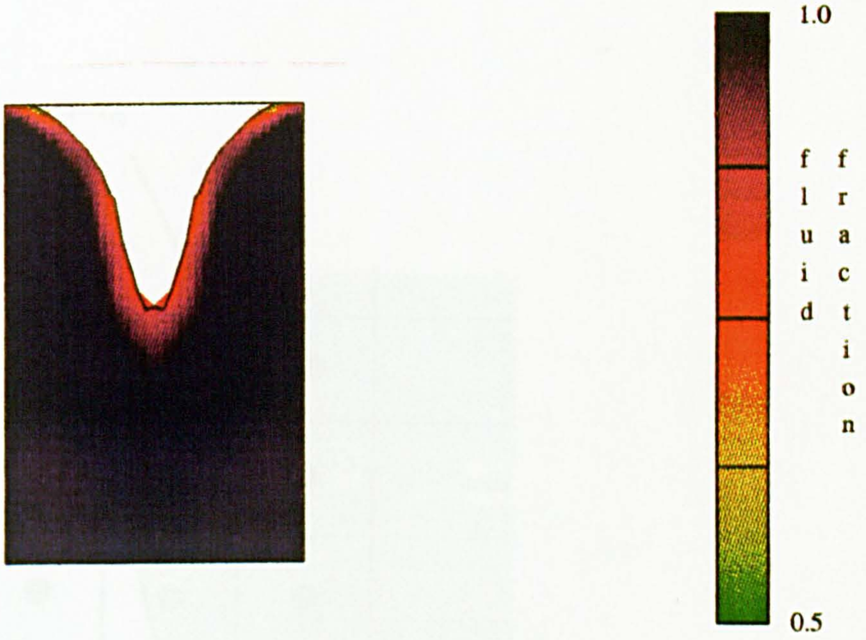


Figure 7.2. Shrinkage cavity in all-sand mould for aluminium (simulation S2.45).

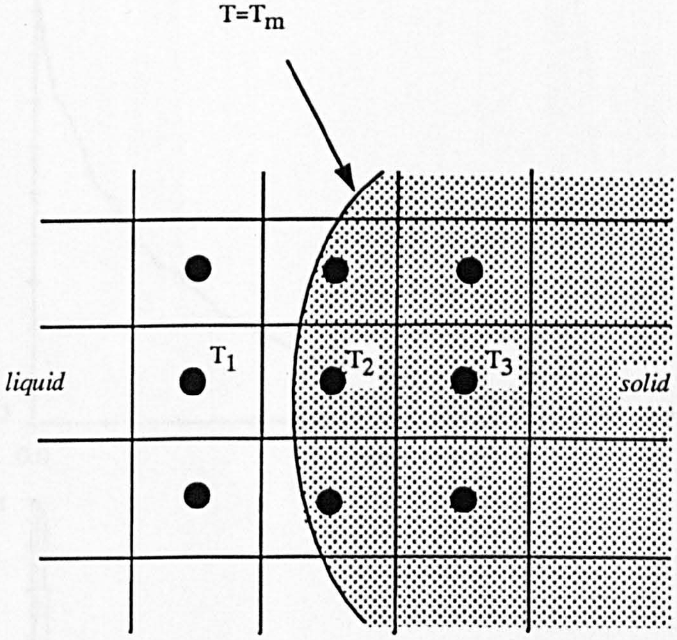


Figure 7.3. Temperature T_2 can be found by linear interpolation between the solidification front and cell with temperature T_3. In standard enthalpy methods $T_2=T_m$.

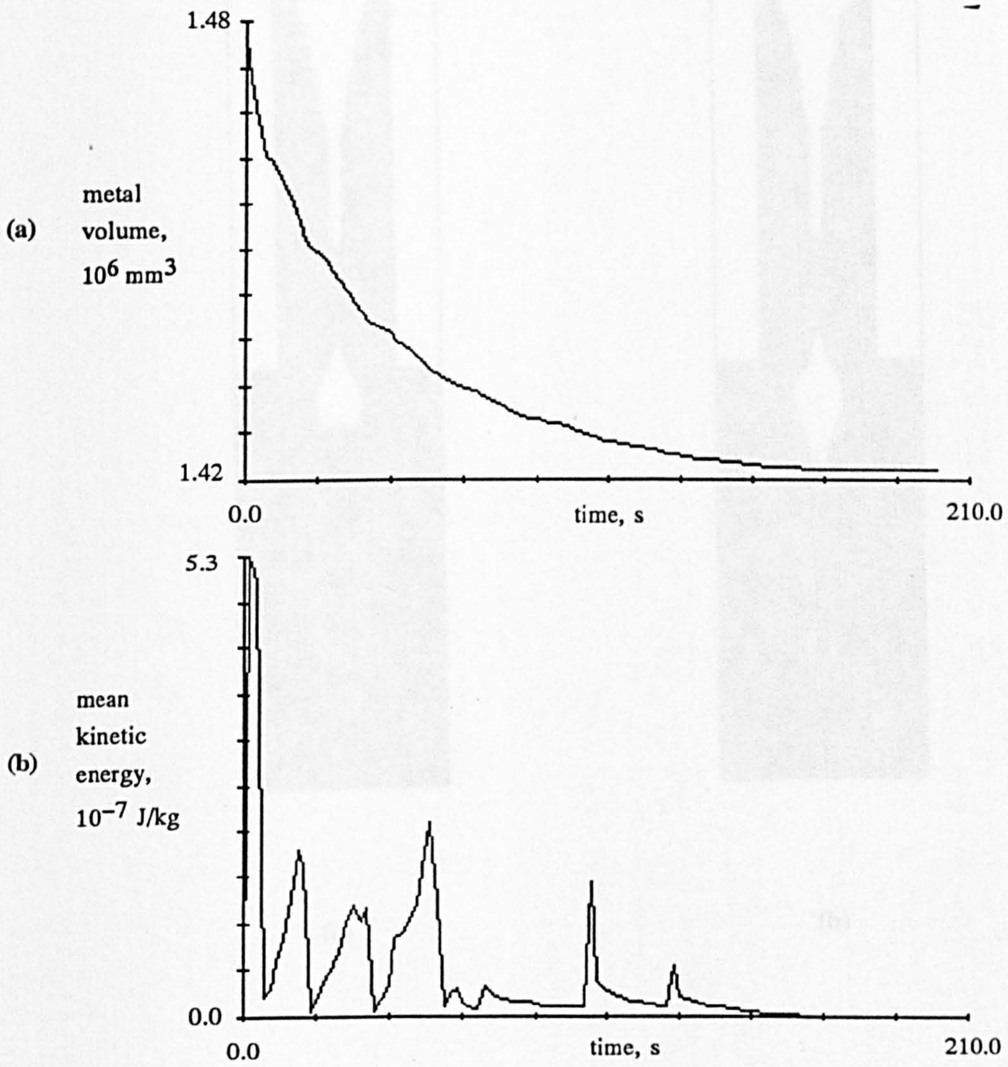


Figure 7.4. (a) Metal volume and (b) mean kinetic energy evolutions for simulation S3.1. The curves are not smooth due to the use of the enthalpy method.

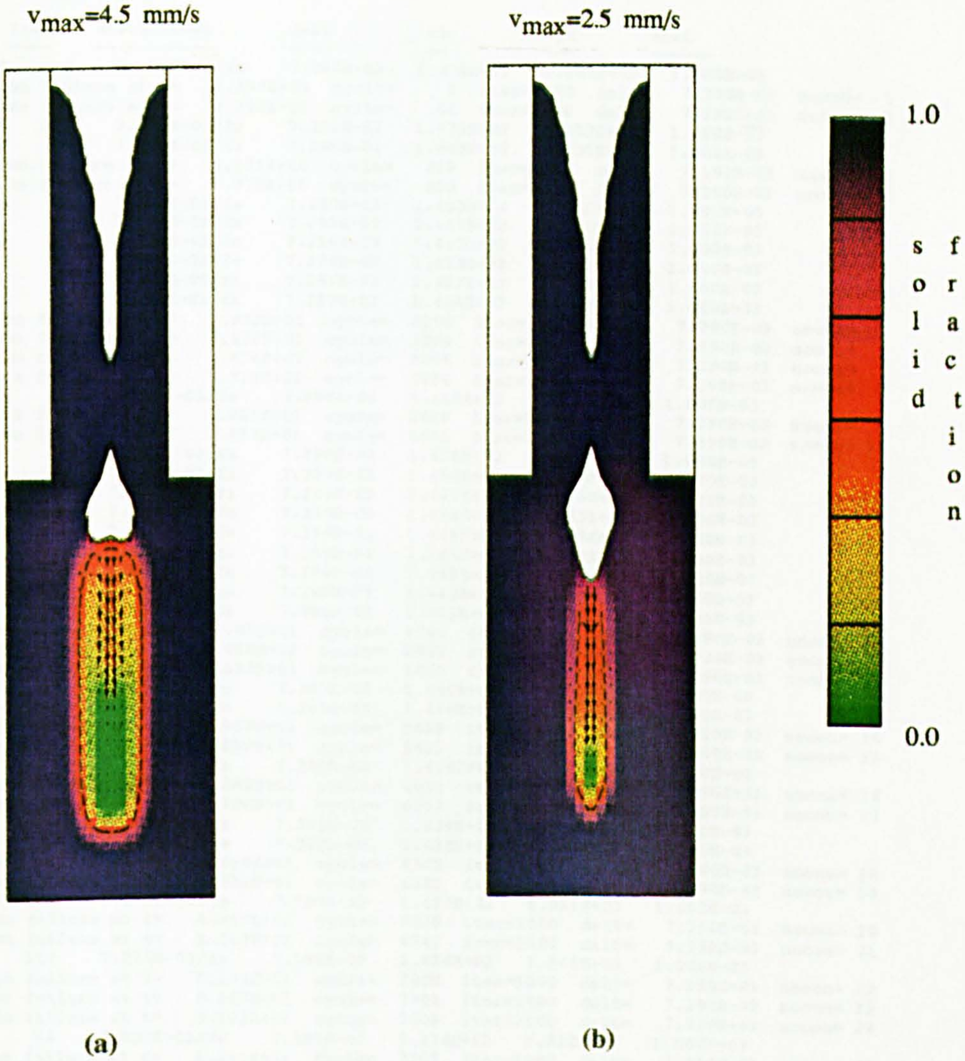


Figure 7.5. Simulation S3.21 using equation (7.3): (a) t=120.0 s and (b) t=150.0 s. It can be seen that solid phase is washed into the bulk of the liquid metal by the feeding flow.

```
     t          cycle  iter    dtstbl/code      delt        vl          cpu        epsi
    ----        -----  ----    -----------      ----        --          ---        -----
1   0.000E+00      0      0    7.290E-03/fs    7.290E-03   1.478E+02   1.280E+00   1.000E-03
2   pressure iteration failure at t=  7.290E-03  cycle=    1  iter=1000  delt=   7.290E-03  nocon=  1
3   pressure iteration failure at t=  3.208E-01  cycle=   44  iter=1000  delt=   7.290E-03  nocon=  2
4   2.005E+00    275    224    7.290E-03/fs    7.290E-03   1.473E+02   2.802E+02   1.000E-03
5   4.010E+00    550     36    7.290E-03/fs    7.290E-03   1.468E+02   4.721E+02   1.000E-03
6   pressure iteration failure at t=  5.971E+00  cycle=  819  iter=1000  delt=   7.290E-03  nocon=  3
7   pressure iteration failure at t=  5.978E+00  cycle=  820  iter=1000  delt=   7.290E-03  nocon=  4
8   6.014E+00    825     12    7.290E-03/fs    7.290E-03   1.463E+02   7.616E+02   1.000E-03
9   8.019E+00   1100     22    7.290E-03/fs    7.290E-03   1.462E+02   8.612E+02   1.000E-03
10  1.002E+01   1375     28    7.290E-03/fs    7.290E-03   1.461E+02   9.551E+02   1.000E-03
11  1.203E+01   1650     30    7.290E-03/fs    7.290E-03   1.459E+02   1.062E+03   1.000E-03
12  1.403E+01   1925     91    7.290E-03/fs    7.290E-03   1.457E+02   1.243E+03   1.000E-03
13  1.604E+01   2200     42    7.290E-03/fs    7.290E-03   1.454E+02   1.460E+03   1.000E-03
14  pressure iteration failure at t=  1.672E+01  cycle= 2294  iter=1000  delt=   7.290E-03  nocon=  5
15  pressure iteration failure at t=  1.673E+01  cycle= 2295  iter=1000  delt=   7.290E-03  nocon=  6
16  pressure iteration failure at t=  1.674E+01  cycle= 2296  iter=1000  delt=   7.290E-03  nocon=  7
17  pressure iteration failure at t=  1.716E+01  cycle= 2354  iter=1000  delt=   7.290E-03  nocon=  8
18  1.804E+01   2475     32    7.290E-03/fs    7.290E-03   1.452E+02   2.073E+03   1.000E-03
19  pressure iteration failure at t=  1.931E+01  cycle= 2649  iter=1000  delt=   7.290E-03  nocon=  9
20  pressure iteration failure at t=  1.933E+01  cycle= 2651  iter=1000  delt=   7.290E-03  nocon= 10
21  2.005E+01   2750     18    7.290E-03/fs    7.290E-03   1.450E+02   2.201E+03   1.000E-03
22  2.205E+01   3025     18    7.290E-03/fs    7.290E-03   1.450E+02   2.270E+03   1.000E-03
23  2.406E+01   3300     28    7.290E-03/fs    7.290E-03   1.449E+02   2.352E+03   1.000E-03
24  2.606E+01   3575     22    7.290E-03/fs    7.290E-03   1.448E+02   2.463E+03   1.000E-03
25  2.807E+01   3850     19    7.290E-03/fs    7.290E-03   1.446E+02   2.589E+03   1.000E-03
26  3.007E+01   4125    268    7.290E-03/fs    7.290E-03   1.445E+02   2.803E+03   1.000E-03
27  3.208E+01   4404     52    7.290E-03/fs    7.290E-03   1.443E+02   3.009E+03   1.000E-03
28  3.408E+01   4679     53    7.290E-03/fs    7.290E-03   1.442E+02   3.138E+03   1.000E-03
29  3.609E+01   4954    371    7.290E-03/fs    7.290E-03   1.441E+02   3.375E+03   1.000E-03
30  pressure iteration failure at t=  3.687E+01  cycle= 5061  iter=1000  delt=   7.290E-03  nocon= 11
31  pressure iteration failure at t=  3.688E+01  cycle= 5062  iter=1000  delt=   7.290E-03  nocon= 12
32  pressure iteration failure at t=  3.688E+01  cycle= 5063  iter=1000  delt=   7.290E-03  nocon= 13
33  3.809E+01   5229     34    7.290E-03/fs    7.290E-03   1.440E+02   3.650E+03   1.000E-03
34  4.010E+01   5504    266    7.290E-03/fs    7.290E-03   1.440E+02   3.771E+03   1.000E-03
35  pressure iteration failure at t=  4.089E+01  cycle= 5613  iter=1000  delt=   7.290E-03  nocon= 14
36  pressure iteration failure at t=  4.091E+01  cycle= 5615  iter=1000  delt=   7.290E-03  nocon= 15
37  4.210E+01   5779    233    7.290E-03/fs    7.290E-03   1.439E+02   4.025E+03   1.000E-03
38  pressure iteration failure at t=  4.382E+01  cycle= 6015  iter=1000  delt=   7.290E-03  nocon= 16
39  pressure iteration failure at t=  4.384E+01  cycle= 6017  iter=1000  delt=   7.290E-03  nocon= 17
40  4.411E+01   6054    521    7.290E-03/fs    7.290E-03   1.439E+02   4.333E+03   1.000E-03
41  4.611E+01   6329     48    7.290E-03/fs    7.290E-03   1.438E+02   4.604E+03   1.000E-03
42  pressure iteration failure at t=  4.650E+01  cycle= 6382  iter=1000  delt=   7.290E-03  nocon= 18
43  pressure iteration failure at t=  4.652E+01  cycle= 6385  iter=1000  delt=   7.290E-03  nocon= 19
44  4.812E+01   6604     34    7.290E-03/fs    7.290E-03   1.437E+02   5.021E+03   1.000E-03
45  pressure iteration failure at t=  4.910E+01  cycle= 6739  iter=1000  delt=   7.290E-03  nocon= 20
46  pressure iteration failure at t=  4.912E+01  cycle= 6742  iter=1000  delt=   7.290E-03  nocon= 21
47  5.012E+01   6879    511    7.290E-03/fs    7.290E-03   1.436E+02   5.368E+03   1.000E-03
48  pressure iteration failure at t=  5.100E+01  cycle= 7000  iter=1000  delt=   7.290E-03  nocon= 22
49  pressure iteration failure at t=  5.102E+01  cycle= 7002  iter=1000  delt=   7.290E-03  nocon= 23
50  pressure iteration failure at t=  5.103E+01  cycle= 7003  iter=1000  delt=   7.290E-03  nocon= 24
51  5.213E+01   7154     54    7.290E-03/fs    7.290E-03   1.436E+02   5.812E+03   1.000E-03
52  pressure iteration failure at t=  5.251E+01  cycle= 7207  iter=1000  delt=   7.290E-03  nocon= 25
53  5.454E+01   7448      2    4.916E-01/cx    3.151E-02   1.435E+02   4.955E+02   1.000E-03   54
54  5.460E+01   7450      3    7.290E-03/fs    7.290E-03   1.435E+02   4.959E+02   1.000E-03
55  5.512E+01   7481      9    7.290E-03/fs    7.290E-03   1.434E+02   6.401E+02   1.000E-03

....
```

Figure 7.6. Calculation diagnostics for simulation S3.1. Convergence failures are highlighted with 'pressure iteration ...' messages.

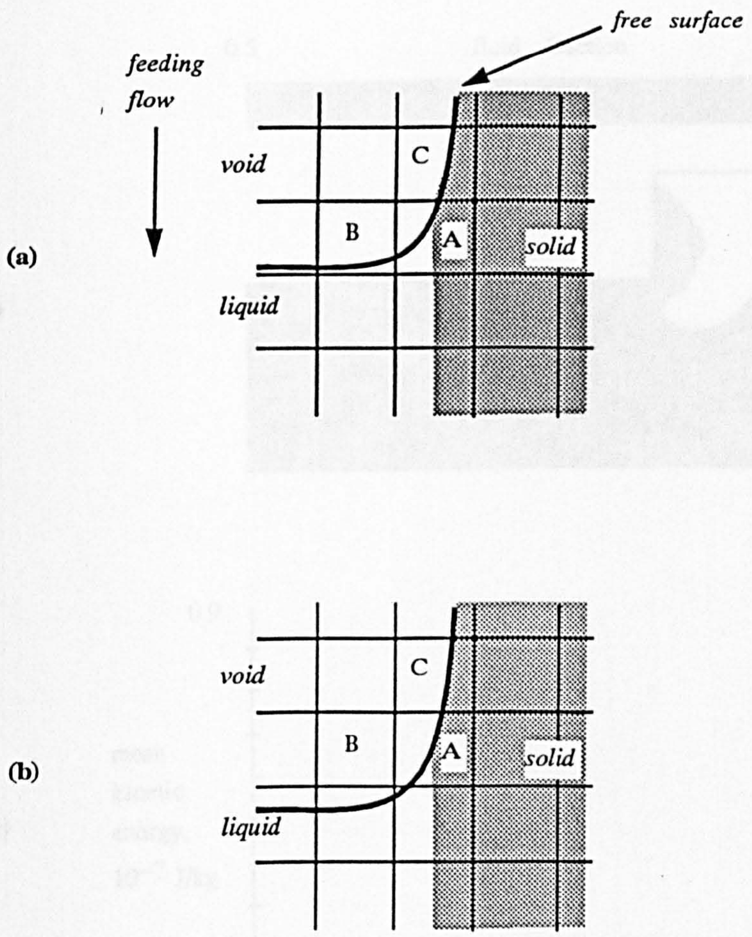


Figure 7.7. The status of partially solid call A changes from ‘internal cell’ at $t=t_n$ (a) to ‘surface cell’ at $t=t_{n+1}$ (b). As a result, pressure p_A changes abruptly (section 7.2.3).

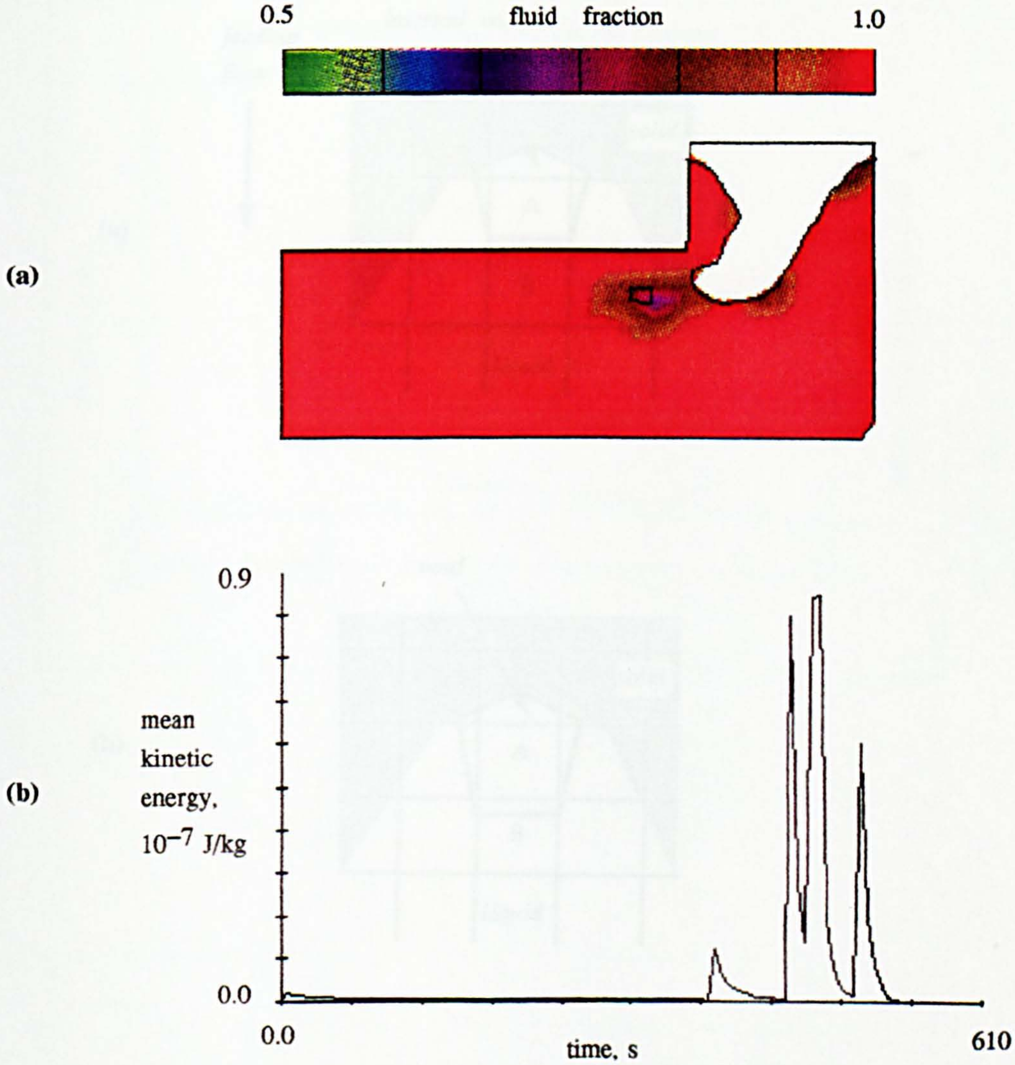


Figure 7.8. Simulation S4.21 using equation (7.3): (a) fluid fraction distribution in the solidified casting and (b) mean kinetic energy evolution.

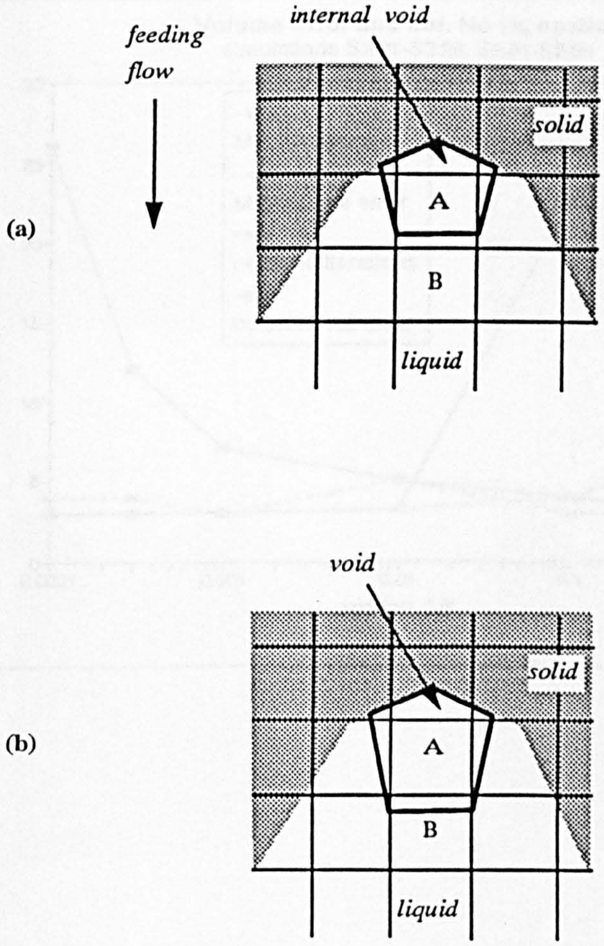


Figure 7.9. The status of liquid call A changes from 'internal cavity cell' at $t=t_n$ (a) to 'empty cell' at $t=t_{n+1}$ (b). As a result, pressure p_A changes abruptly (section 7.2.3).

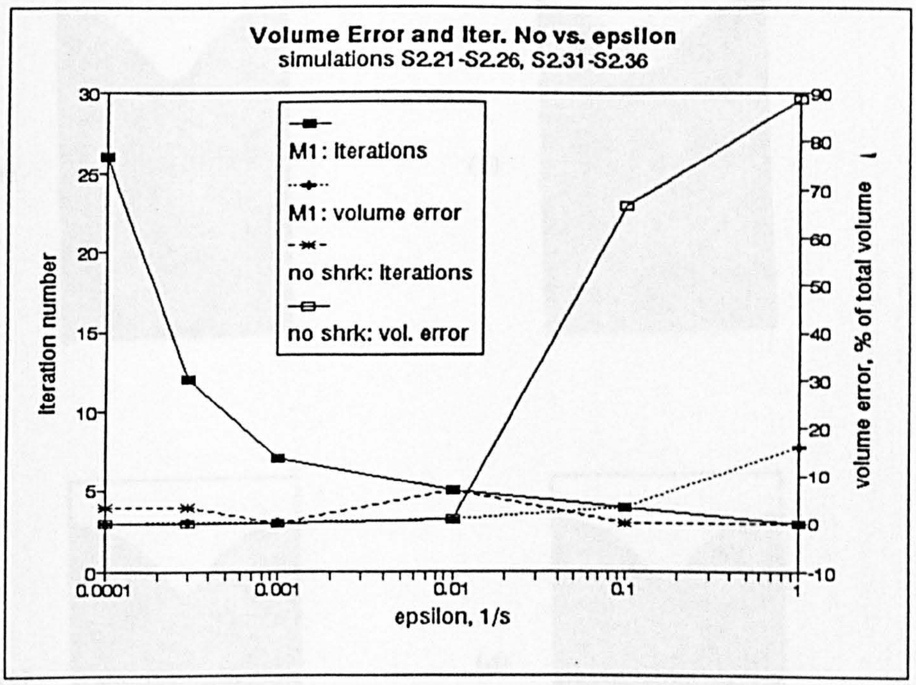


Figure 7.10. Average iteration number and volumetric error due to finite convergence residuals as functions of the convergence criterion ε for the 2-D cylindrical sand mould casting using the full shrinkage model M1 (simulations S2.21-S.26) and with no shrinkage at all (simulations S2.31-S2.36).

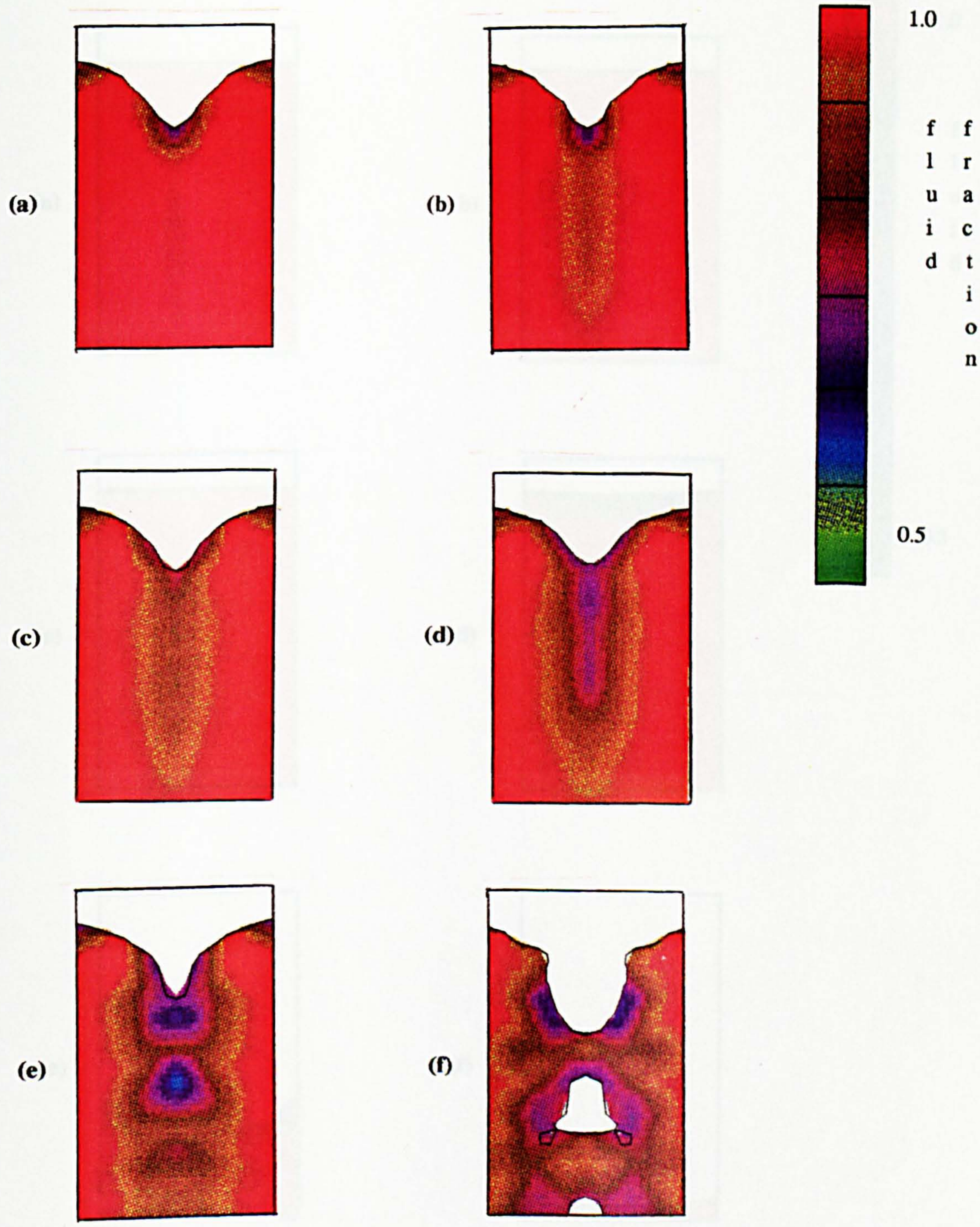


Figure 7.11. Fluid fraction distribution in solidified castings using shrinkage model M1 (simulations S2.21-S2.26): (a) ε=0.0001 s^{-1}, (b) ε=0.0003 s^{-1}, (c) ε=0.001 s^{-1}, (d) ε=0.01 s^{-1}, (e) ε=0.1 s^{-1} and (f) ε=1.0 s^{-1}.

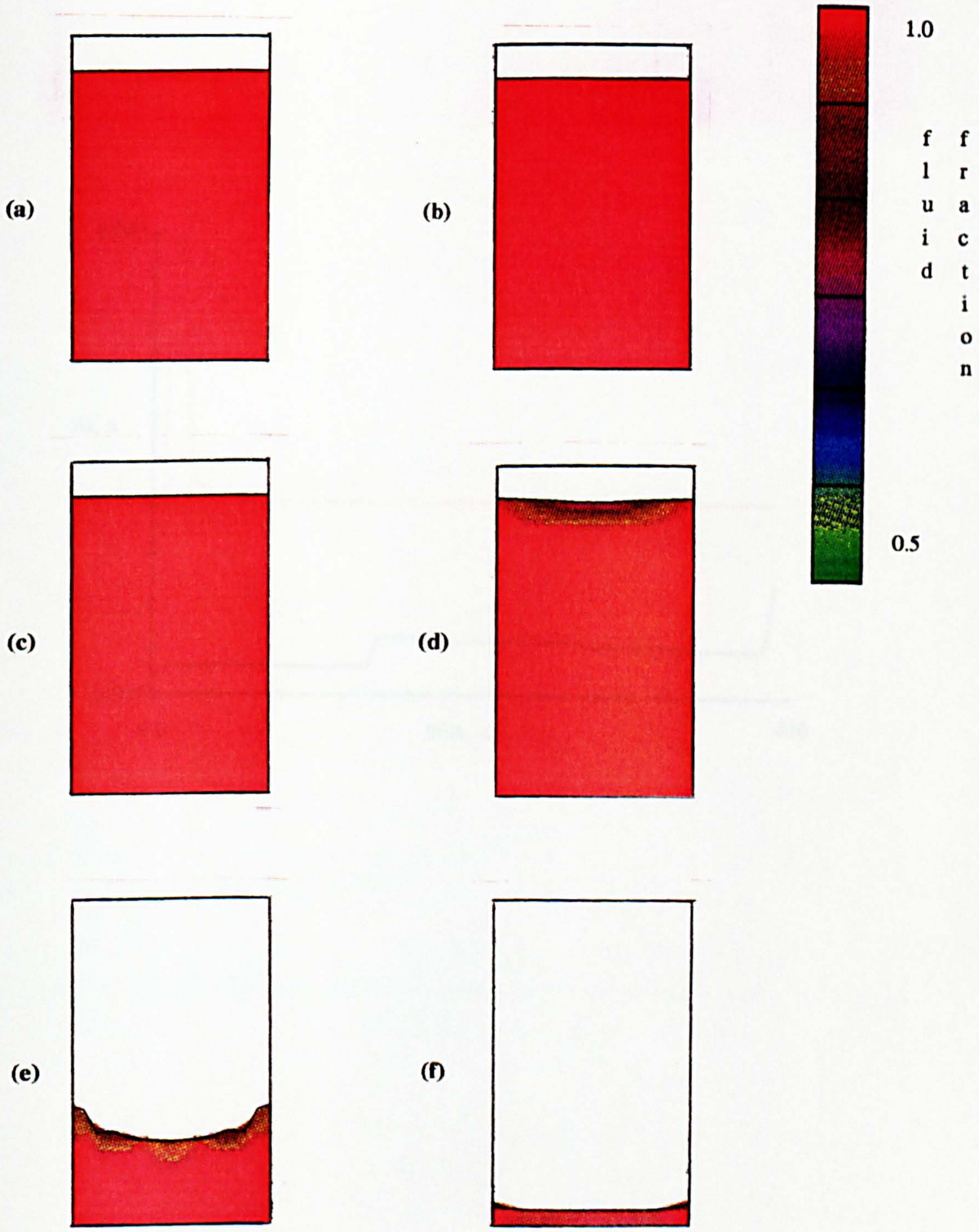


Figure 7.12. Fluid fraction distribution in solidified castings with no shrinkage in the metal (simulations S2.31-S2.36): (a) ε=0.0001 s^{-1}, (b) ε=0.0003 s^{-1}, (c) ε=0.001 s^{-1}, (d) ε=0.01 s^{-1}, (e) ε=0.1 s^{-1} and (f) ε=1.0 s^{-1}.

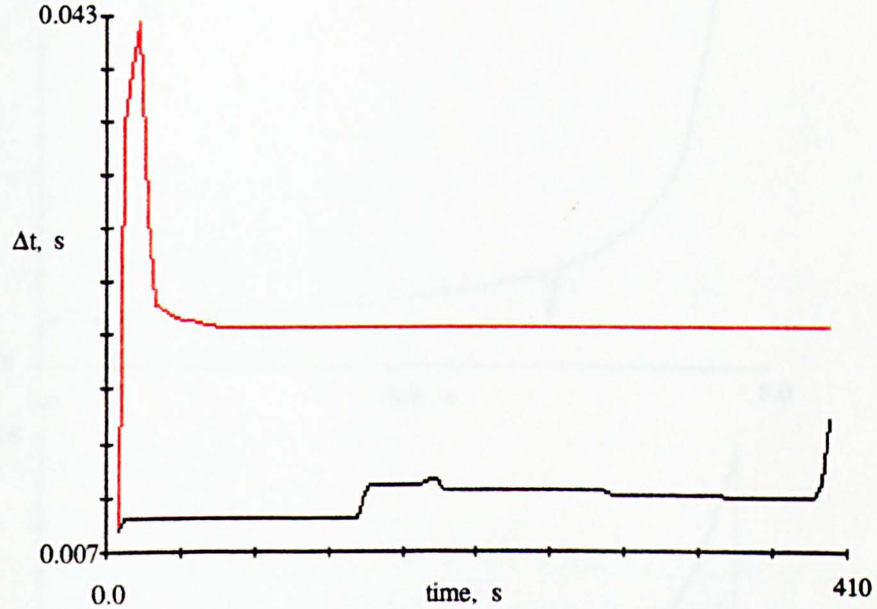


Figure 7.13. Comparison of the time step sizes used in full model simulation S5.5 (black line) and in simplified model simulation S5.22 (red line).

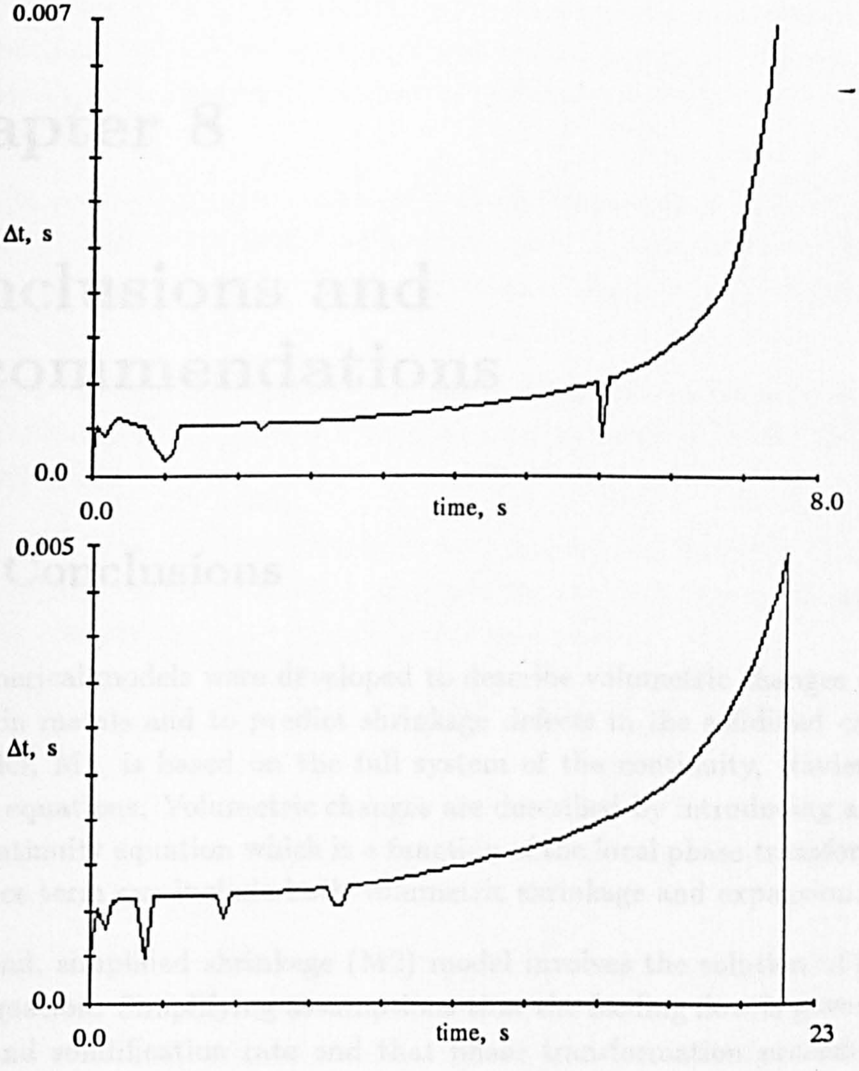


Figure 7.14. Time step size evolutions during filling for (a) 15 mm choke (simulation S5.1) and (b) 8 mm choke (simulation S5.3).

Chapter 8

Conclusions and Recommendations

8.1 Conclusions

Two numerical models were developed to describe volumetric changes during solidification in metals and to predict shrinkage defects in the solidified castings. The first model, Ml, is based on the full system of the continuity, Navier-Stokes and enthalpy equations. Volumetric changes are described by introducing a source term in the continuity equation which is a function of the local phase transformation rate. The source term can include both volumetric shrinkage and expansion.

The second, simplified shrinkage (M2) model involves the solution of only the enthalpy equation. Simplifying assumptions that the feeding flow is governed only by gravity and solidification rate and that phase transformation procedes only from liquid to solid allowed the fluid flow equations to be excluded from consideration.

The numerical implementation of both models in three-dimensions was based on the existing commercial, general purpose CFD code FLOW-3D which already contained a numerical algorithm for incompressible fluid flow with heat transfer and phase tranformation. An important part of the code is the Volume Of Fluid (VOF) algorithm for tracking free surface interfaces. The VOF function was employed in both developed models to describe shrinkage cavity formation. Additions to FLOW-3D to incorporate the shrinkage models include:

- Inclusion of the source term in the continuity equation and the equation for the VOF function, F.

- Modification of the SOR pressure-velocity iteration scheme to allow internal cavities to open and close.
- Development of a new algorithm for the simplified shrinkage model which describes free surface evolution and cavity formation without solving fluid flow and F equations.

In addition to these changes to the code which were required to incorporate the new shrinkage models, a few modifications were also made to improve the accuracy of the metal/mould heat trasnfer and solidification algorithms. These include:

- Linear interpolation of metal and mould temperatures between the metal/mould interface and cell nodes to obtain a better approximation for the heat trasnfer coefficient.
- Development of a new expression for the drag coefficient for the drag force in momentum equations which accounts for the fact that a computational cell may contain a mixture of solid and liquid phases.
- Addition of the lever rule and Scheil equation to describe solidification in alloys.

The critical value of the solid fraction was used as the feeding criterion in both models.

As a result of those modifications and developments, a number of new input parameters were introduced to the code. These input variables specify the shrinkage model, the value of the critical solid fraction and the solidification model. The solid fraction was also added as an output variable for both spatial and history plotting. All additional input and output, associated with these developments, is part of the standard input and output interface of the FLOW-3D copy installed at The University of Sheffield and could be distributed and used together with the standard code on a commercial basis.

The developed shrinkage models were applied to a variety of castings of different configurations, involving pure metals (aluminium, iron), alloys (aluminium-copper, steel), sand and chill moulds. The results were compared with experimental data. The influence of the filling stage on shrinkage defects in the final casting was investigated.

As part of the development of the upwind differencing advection algorithm used in the simulations, the Leith's method was incorporated into the standard two-dimensional SOLA code. It is shown that the resulting scheme is unconditionally stable despite being explicit.

The following is a summary of the results and conclusions.

1. In general, the results of the full shrinkage model are only marginally closer to the experimental ones than those of the simplified model. The simplified model underestimates the depth of the primary cavities even more than the full model and predicts an earlier separation of internal liquid region from the feeder. The latter led to larger secondary cavities in the T-shaped casting. These differences are explained by the fact that the simplified model does not describe advective heat transfer in the feeding flow. Nevertheless, in the case of the pure iron and steel cylinder casting the two models gave practically identical cavity predictions close to the experimental results.

2. Predictions of the size and position of secondary cavities are close to the experimental results, especially for pure metals.

3. Pure metal simulations for the T-shaped casting demonstrated the importance of modelling of the mould filling stage. It is shown that the size and occurrence of these shrinkage defects *are* dependent on the filling rate. For the T-shaped casting it was possible to eliminate the shrinkage in the T-junction by increasing the filling time by a factor of three.

4. The largest discrepancies between numerical predictions and experimental results were for Al-4.5%Cu alloy castings.

5. Despite an accuracy of 1% for shrinkage volume predictions, the depth of primary cavities in feeders tends to be underestimated. This is due to truncation errors in free surface representation (Section 7.1) and inadequate physical modelling heat transfer processes, such as radiative and convective heat exchange, which leads to smaller heat losses from the top of the feeder.

6. Cooling rates and total solidification times simulated by the two models are very similar in most cases (less than 1% difference). The small difference arises from a more accurate conservation of the total fluid volume by the M2 model since it does not have errors associated with finite iteration residuals and advection of the fluid fraction function, F.

7. Both shrinkage models can be applied after the filling stage simulation has been completed. The simplified shrinkage model requires less storage and computer time, therefore, it can be used efficiently in cases where the metal flow *during* solidification is not important and the cooling process is dominated by conduction within the metal and heat transfer at metal/mould interface. On average the simplified model runs 10 to 20 times faster than the full model.

8. The SOR iteration method, together with the VOF technique for free surface description, are flexibile and accurate for application for shrinkage modelling. However, when a large volume of liquid free surface is resolved only by one or two cells, which is a likely situation during shrinkage modelling, iteration problems arise if the free surface boundary pressure changes abruptly due to a change in the slope. Such pressure pulses magnify long wavelength residuals which decrease slowly in SOR iterations.

9. The time step size in the M1 shrinkage model is controlled by the free surface stability limit during most of the simulation time, which is a consequence of assuming hydrostatic pressure distribution in surface cells. In the simplified model the time step size is limited by the heat transfer stability criterion. This is usually several times larger than the free surface time step limit. The largest deficiency of the full shrinkage model remains its computational cost due to time step limitations and additional effort to achieve iteration convergence.

10. The shape of the cavities in solidified castings is mesh dependent though the shrinkage volume is not. The mesh dependence arises from truncation errors in free surface representation and temperature gradient calculations. However, these errors decrease as the mesh is refined.

11. A value of unity for the critical fraction of solid $f_{s,cr}$, used as a feeding criterion, gives good results for pure metals. For alloys a value above 0.8 appears to be most reasonable. For fast solidification and a narrow freezing range alloy, $f_{s,cr} = 1.0$ also gives best results.

12. Approximation of the temperature profile in the interfacial cells between the cell centre and the interface by a linear function gives a second order accuracy with respect to the cell size Δx in the direction normal to the interface. Assuming a constant temperature profile between these points reduces the accuracy to the first order, besides, the coefficient $h\partial T/\partial x|_{x=0}$ of the first order term is large for low diffusivity materials such as sand, especially at the start of the simulation. In the latter case the first order term can be substantially larger than the zeroth order term.

13. The second order heat transfer algorithm, in conjunction with the FAVOR method for representing mould walls, substantially reduces mesh dependence of the interfacial heat flux, compared with the first order method.

14. The solidification drag which arises from the fact that momentum equations are applied to the solid/liquid mixture, proves to be an adequate method for describing momentum changes due to phase transformation. In particular, fully solid computational cells are marked with an infinitely large value of the drag that is used to define the boundaries of liquid regions in the casting.

15. Since velocities in a feeding flow are usually small, between 0.1 to 10 mm/s, but pressure in the liquid metal bulk changes significantly as the free surface level drops, the feeding flow is pressure dominated, that is

$$\frac{2\Delta p}{\rho U^2} \gg 1$$

In that case the value of the convergence criterion ϵ required for an efficient calculation is several orders of magnitude higher than that estimated from the velocity scale alone. An appropriate formula to estimate ϵ was worked out to complement the existing one for velocity (or inertia) dominated flows. Feeding flow stabilises numerical solution in the sense that it allows one to use a much larger value of ϵ than in the absence of shrinkage.

16. It has been shown that the Leith's method for approximating advection terms in momentum equations can be used in the standard SOLA algorithm, instead of the upwind differencing. The advantage of the Leith's method, also called 'semi-Lagrangian' method, is in its unconditional numerical stability, so that the CFL restriction for the time step size is removed. However, preliminary results show no gain in the efficientcy of the calculations since an increase in the time steps size causes an increase in iteration. This indicates that large time steps reduce the accuracy of the first-guess velocity approximation. An application of the method in a colocated grid would improve its efficiency.

8.2 Recommendations for Future Work

8.2.1 Physical Modelling

1. The solidification model may be further developed to include an accurate description of macrosegregation in alloy solidification. On this basis a model

of flow in the mushy zone and feeding, more sophisticated than the critical fraction of solid criterion, should be formulated. Lack of accuracy in alloy solidification remains the largest problem in the physical model used in the present work.

2. The model should be extended to include radiative and convective heat transfer at the free surface.

3. The effect of thermal convection on shrinkage defect formation could be investigated using the full shrinkage model.

4. The full shrinkage model can be used as the basis for developing more sophisticated models to include gas evolution and the formation of microporosity.

8.2.2 Numerical Modelling

1. Further steps should be maid in improving the efficiency of the velocity-pressure solution algorithm.

2. Since the full shrinkage model requires substantial computational power it will be advantageous to establish more rigorously conditions when modelling of flow during solidification is required. Alternatively the simplified model can be used to obtain results for a variety of casting designs and then the full model can be employed to verify the final design.

3. The procedure of setting pressure boundary condition at free surfaces should be made more accurate for flows with low Froude numbers to avoid convergence problems during shrinkage simulations due to large sudden changes in surface cell pressures.

4. Numerical experiments could be carried out to establish the optimum values of numerical parameters, such as the convergence criterion ϵ and the overrelaxation factor ω. Ultimately an automatic procedure could be worked out to define these parameters in each case.

5. The truncation accuracy of the numerical equation for the metal/mould interface heat transfer should be improved to minimise mesh dependence.

6. The simplified model does not employ the upwind differencing nor the donor-acceptor advection scheme for the free surface which constitute the core of the

FLOW-3D. Therefore the shrinkage model could separated be from FLOW-3D to create a smaller software package which can be used as a solidification modelling tool on smaller computers.

7. Further work should be carried out to investigate ways of applying the Leith's method to momentum, enthalpy and VOF equations to increase the efficiency of mould filling simulations in which the CFL stability criterion poses a severe restriction on the time step size when the upwind and donor-acceptor advection schemes are used.

Bibliography

[1] T.S.Piwonka, "Integrating Computer Modeling with the Needs of Modern Foundries", in "Modeling of Casting, Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 639-647, 1993.

[2] J.Campbell, "Castings", Butterworth-Heinemann, Oxford, 1991.

[3] L.I.Sedov, "Mekhanika Sploshnoi Sredy" (Continuum Mechanics), Nauka, Moscow, 1976.

[4] J.Szekely, "Fluid Flow Phenomena in Metals Processing", Academic Press, London, 1979.

[5] D.M.Lipinski, W.Schaefer and E.Flender, "Numerical Modeling of the Filling Sequence and Solidification of Castings", in "Modeling of Casting, Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 389-396, 1993.

[6] A. Cottrell, "An Introduction to Metallurgy", Edward Arnold Publishers, London, 1975.

[7] M.C.Flemings, "Solidification Processing", McGraw-Hill Book Company, New-York, 1974.

[8] J.W.Christian, "The Theory of Transformations in Metals and Alloys", Pergamon Press, Oxford, 1965.

[9] F.M.White, "Viscous Fluid Flow", McGraw-Hill Book Company, New York, 1974.

[10] J.O.Hinze, "Turbulence", McGraw-Hill Book Company, New York, 1975.

[11] S.Sundarraj, V.R.Voller, "A Discussion on Results of Recent Microsegregation Models", in "EPD Congress 1992", eds. J.P.Hager, TMS Publication, pp 461-479, 1991.

[12] G.Upadhya, C.M.Wang and A.J.Paul, "Solidification Modeling: Coupling of Solidification Kinetics in a Comprehensive Model with Fluid Flow and Heat Transfer for Cast Iron", in "EPD Congress 1992", eds. J.P.Hager, TMS Publication, pp 773-787, 1991.

[13] Ph.Thevoz, M.Rappaz, "Modeling of Equiaxed Microstructures in Solidification Processes: An Overview of Solute Diffusion Models", in "Modeling of Casting Welding and Advanced Solidification Processes V", eds. M.Rappaz, M.R.*Özgü* and K.W.Mahin, TMS Publication, pp 387-394, 1991.

[14] D.E.Loper, "Dynamical Processes in Slurries and Mushes", in "Nature and Properties of Semi-Solid Materials", ads. J.A.Sekhar and J.A.Dantzig, TMS Publication, pp 91-104, 1992.

[15] W.J.Boettinger, A.A.Wheeler, B.T.Murray, G.B.McFadden and Kobayashi, "Calculation of Alloy Solidification Morphologies Using the Phase-Field Method" in "Modeling of Casting, Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 79-86, 1993.

[16] S.G.R.Brown and J.A.Spittle, "A 2-D Implicit Finite Difference Model to Simulate the Columnar to Equiaxed Zone Transition", in "Modeling of Casting Welding and Advanced Solidification Processes V", eds. M.Rappaz, M.R.*Özgü* and K.W.Mahin, TMS Publication, pp 395-402, 1991.

[17] Ch.-A.Gandin, Ch.Charbon and M.Rappaz, "Probabilistic Modeling of Grain Formation in Solidification Processes", in "Modeling of Casting, Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 21-28, 1993.

[18] J.Ortega, "Influence of Convection on Macrostructure of Aluminium and Al-Cu Alloys", Ph.D. Thesis, The University of Sheffield, 1994.

[19] C.Beckermann, "Modeling of Transport Phenomena in Mushy Zones", in "Modeling of Casting, Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 181-192, 1993.

[20] A.Sommerfield, "Partial Differential Equations in Physics", Academic Press, New-York, 1964.

[21] P.M.Morse and H.Feshbach, "Methods of Theoretical Physics", Part I, McGraw-Hill Book Company, New-York, 1953.

[22] J.P.Holman, "Heat Transfer", McGraw-Hill Book Company, New-York, 1976.

[23] J.H.Ludley, J.Szekely, J. Iron Steel Inst., **204**, p. 12, 1966.

[24] T.X.Hou, R.D.Pehlke, "Determination of Mold-Metal Interfacial Heat Transfer and Simulation of Solidification of an Aluminium-13%Silicon Casting", AFS Transactions, **91**, pp 129-136, 1988.

[25] C.Hirsch, "Numerical Computation of Internal and External Flows. Volume 1: Fundamentals of Numerical Discretisation", Jonh Wiley& Sons, Chichester, 1988.

[26] M.L.James, G.M.Smith and J.C.Wolford, "Applied Numerical Methods for Digital Computation with FORTRAN", International Textbook Company, Scranton, 1970.

[27] P.J.Roache, "Computational Fluid Dynamics", Hermosa Publishers, Albuquerque, 1985.

[28] P.N.Hansen, G.C.Hartman and J.C.Sturm, "Elimination of Shrinkage Defects Through Use of Computer Simulation", AFS Transactions, **146**, pp 477-483, 1991.

[29] R.Ananth and W.N.Gill, "Self-consistent Theory of Dendritic Growth with Convection", J.Crystal Growth, **108**, pp 173-189, 1991.

[30] Ge-Cheng Zha, Dao-Zhi Liu and Tie-You Ma, "An Efficient Upwind/Relaxation Algorithm for the Euler and Navier-Stokes Equations", Int. J. Num. Meth. Fluids, **9**, pp 517-529, 1989.

[31] J.Häuser, C.Taylor, eds., "Numerical Grid Generation in Computational Fluid Dynamics", Pineridge Press, Swansea, 1986.

[32] A.J.Baker, "Finite Element Computational Fluid Dynamics", Hemisphere Publishing Corporation, London, 1983.

[33] M.J.Turner, R.W.Clough, H.C.Martin, L.P.Topp, "Stiffness and Deflection Analysis of Complex Structures", J. Aeron. Soc., **23**, pp 805-822, 1956.

[34] O.C.Zienkiewicz, Y.K.Cheung, "Finite Elements in the Solution of Field Problems", The Engineer, pp 507-510, 1965.

[35] A.S.Usmani, J.T.Cross, R.W.Lewis, "Finite Element Modelling of Mould Filling in Arbitrary Shaped Moulds", in "EPD Congress 1992", eds. J.P.Hager, TMS Publication, pp 579-594, 1991.

[36] J.R.Howell, ?, in "Modelling and Control of Casting and Welding Processes, Vol.3", eds. S.Kou and R.Mehrabian, AIME, p. 377, 1986.

[37] R.W.Lewis, H.C.Huang and A.S.Usmani, "Remeshing Approaches to Finite Element Simulation of Casting Problems" in "Heat Transfer and Phase Change Problems", Eurotherm Seminar 6, Extended Abstracts, Deyt, Holland, pp 37-39, 1988.

[38] G.Bonacina, G.Comini, "On the Solution of the Non-linear Heat Conduction Equations by Numerical Methods", Int. J. Mass Transfer, **16**, pp 581-589, 1983.

[39] T.X.Hou, R.D.Pehlke and J.O.Wilkes, "Computer Simulation of Casting Solidification Using a Combination of the Finite Element and Boundary Element Methods", in "Modeling of Casting Welding and Advanced Solidification Processes V", eds. M.Rappaz, M.R.*Özgü* and K.W.Mahin, TMS Publication, pp 15-23, 1991.

[40] N.Kondo, N.Tosaka and T.Nishimura, "Computation of Incompressible Viscous Flows by the Third-order Upwind Finite Element Method", Int. J. Num. Meth. Fluids, **15**, pp 1013-1024, 1992.

[41] Y.F.Zhang, W.K.Liu, "Casting Filling Simulations of Thin-Walled Cavities with Solidification", in "Modeling of Casting, Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 413-420, 1993.

[42] D.M.White, M.T.Samonds, "Finite Element Free Surface Modeling", in "Modeling of Casting, Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 357-364, 1993.

[43] G.P.Backer, "Finite Element Free Surface Flow Analysis: A New Tool for Foundry Engineers", in "Modeling of Casting, Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 405-412, 1993.

[44] A.M.Hines, J.S.Tu, "Modeling of Fluid Flow and Heat Transfer in the Solidification of Superalloy Investment Castings", in "Modeling of Casting, Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 461-468, 1993.

[45] A.Shapiro, W.Stein and P.Rabain, "Casting Process Modeling Using ProCAST and CAST2D", in "Modeling of Casting, Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 493-500, 1993.

[46] "Developments in Boundary Element Methods-2", eds. P.K.Banerjee, R.P.Shaw, Applied Science Publishers, London, 1982.

[47] J.Clark, "Free and Moving Boundary Problems", Clarendon Press, Oxford, 1984.

[48] F.H.Harlow, J.E.Welch, "The MAC Method: A Computing Technique for Solving Viscous, Incompressible, Transient Fluid Flow Problems Involving Free Surfaces", Physics of Fluids, **8**, 12, pp 2182-2189, 1965.

[49] R.A.Stoehr, C.Wang, "Advances in Fluid Flow, Heat Transfer and Solidification Modeling and Applications to Actual Foundry Problems", in "Modeling of Casting Welding and Advanced Solidification Processes V", eds. M.Rappaz, M.R.*Özgü* and K.W.Mahin, TMS Publication, pp 725-732, 1991.

[50] K.S.Chan, K.Pericleous, M.Cross, "Numerical Simulation of Flows Encountered During Mold-Filling", Appl. Math. Modelling, **15**, pp 624-631, 1991.

[51] C.W.Hirt, B.D.Nichols, "Volume of Fluid (VOF) Method for the Dynamics of Free Boundaries", J.Comput. Phys., **39**, pp 201-225, 1981.

[52] R.A.Gentry, R.E.Martin and B.J.Daly, "An Eulerian Differencing Method for Unsteady Compressible Flow Problems", J. Comput. Phys., **1**, pp 87-118, 1966.

[53] C.W.Hirt, B.D.Nichols and N.C.Romero, "SOLA-A Numerical Solution Algorithm for Transient Fluid Flows", Los Alamos Scientific Laboratory report LA-5852, 1975.

[54] S.P.Vanka, "Block-Implicit Multigrid Solution of Navier-Stokes Equations in Primitive Variables", J. Comput. Phys., **65**, pp 138-158, 1986.

[55] K.Torrance, R.Davis, K.Eike, P.Gill, D.Gutman, A.Hsui, S.Lyons and H.Zien, "Cavity Flows Driven by Buyoancy and Shear", J. Fluid Mechanics, **51**, part 2, pp 221-231, 1972.

[56] A.Robert, "A Semi-Lagrangian and Semi-Implicit Numerical Integration Scheme for Primitive Meteorological Equations", J.Meteorological Soc. Japan, **60**, pp 86-103, 1982.

[57] J.R.Bates, "Analysis and Implementation of the Semi-Lagrangian Technique", Proc. 1983 ECMWF Seminar on Numerical Methods for Weather Prediction, **2**, pp 135-150, 1984.

[58] S.P.Frankel, "Convergence Rates of Iterative Treatments of Partial Differential Equations", Math. Tables and Other Aids to Computation", **4**, pp 65-75, 1950.

[59] J.M.Ortega, W.C.Rheinboldt, "Iterative Solution of Nonlinear Equations in Several Variables", Academic Press, New-York, 1970.

[60] C.Wang, "Computer Modeling of Fluid Flow and Heat Transfer in Metal Castings", Ph.D. Thesis, The University of Pittsburgh, Pittsburgh, 15261, 1990.

[61] J.Douglas, J.E.Gunn, "A General Formulation of Alternating Direction Method – Part I: Parabolic and Hyperbolic Systems", Numerische Mathematik, **6**, pp 428-453, 1964.

[62] P.Hageman, D.M.Young, "Applied Iterative Methods", Academic Press, New York, 1981.

[63] G.Birkoff, R.S.Varga and D.Young, "Alternating Direction Implicit Methods", Advances in Computers, **3**, ed. F.L.Alt, Academic Press, New-York, pp 190-274, 1962.

[64] C.H.Chuan, W.C.Schreiber and C.L.Huang, "A Comparison of ADI, Line-by-Line and Gauss-Seidel in Conjunction with the Finite Volume Method for Solving 3-D Conduction/Freezing Problems", ?, 1989.

[65] H.-J. Liu, W.-S.Hwang, "Combined Fluid Flow and Heat Transfer Analysis for the filling of Castings", AFS Transactions, **144**, pp 447-458, 1988.

[66] R.A.Stoehr, C.Wang, "Coupled Heat Transfer and Fluid Flow in the Filling of Castings", AFS Transactions, **176**, pp 733-740, 1988.

[67] H.J.Lin, H.L.Tsai, "Numerical Simulation of of an Integrated filling-Solidification Casting System", in "Modeling of Casting, Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 381-388, 1993.

[68] A.Brandt, J.E.Dendy, Jr. and H.Ruppel, "The Multigrid Method for Semi-Implicit Hydrodynamics Codes", J. Comput. Phys., **34**, pp 348-370, 1980.

[69] A.Brandt, "Guide to Multigrid Development" in "Multigrid Methods, Lecture Notes in Mathematics", Vol. 960, Springer Verlag, New York, 1982.

[70] B.D.Nichols, C.W.Hirt, "Improved Free Surface Boundary Conditions for Numerical Incompressible Flow Simulations", J. Comput. Phys., **8**, pp 434-448, 1971.

[71] H.-J. Liu, W.-S.Hwang, "Three-Dimensional Fluid Flow Simulation for Mold Filling", AFS Transactions **171**, pp 855-862, 1989.

[72] C.R.Swaminathan, V.R.Voller, "An 'Enthalpy Type' Formulation for the Numerical Modeling of Mold Filling", in "Modeling of Casting, Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 365-372, 1993.

[73] C.R.Swaminathan, V.R.Voller, "A Time-Implicit Filling Algorithm", Appl. Math. Modelling, **18**, pp 101-108, 1994.

[74] J.R.Hartin, M.L.Tims, C.W.Wang and E.Meyer, "Solidification Modeling of Centrifugally Cast Titanium Aluminides", in "EPD Congress 1992", eds. J.P.Hager, TMS Publication, pp 899-914, 1991.

[75] G.Trapaga, J.Szekely, "Mathematical Modeling of the Isothermal Impingement of Liquid Droplets in Spraying Processes", Metallurgical Transactions, **22B**, pp 901-914, 1991.

[76] C.W.Chen, C.C.Yang and W.S.Hwang, "Modeling of Puddle Formation in PFC Process", in "Modeling of Casting, Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 443-450, 1993.

[77] G.K.Reziak, P.Gilotte and R.Hamar, "Modeling of Fluid Flow During Mold Filling", in "Modeling of Casting, Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 435-442, 1993.

[78] A.Bourg, A.Latrobe, P.Large, P.Laty and C.Rigant, "SIMULOR and Fast Solutions for Mold Filling Simulations", Bristol, September 23, 1991.

[79] C.W.Hirt, J.L.Cook and T.D.Butler, "A Lagrangian Method for Calculating the Dynamics of an Incompressible Fluid with Free Surface", J. Comput. Phys., **5**, No. 1, pp 103-124, 1970.

[80] E. Gutierrez-Miravete, "Computer Simulation of Mold Filling Processes", in "EPD Congress 1992", eds. J.P.Hager, TMS Publication, pp 481-487, 1991.

[81] J.B.Bugg, R.D.Rowe, "Modelling the Initial Motion of Large Cylindrical and Spherical Bubbles", Int. J. Num. Meth. Fluids, **13**, pp 109-129, 1991.

[82] K.Anzai, E.N.Niyama, "Simplified VOF and Adaptive Pressure Iteration Methods for Mold Filling Simulation", in "Modeling of Casting, Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 469-476, 1993.

[83] FLOW-3D Manual, FLOW SCIENCE Inc., Los Alamos, NM, 1991.

[84] M.Barkhudarov, H.You, J.Beech, S.B.Chin and D.H.Kirkwood, "Validation and Development of FLOW-3D for Casting", in "EPD Congress 1992", eds. J.P.Hager, TMS Publication, pp 595-603, 1991.

[85] M.R.Barkhudarov, S.B.Chin, "Stability of a Numerical Algorithm for Gas Bubble Modelling", Int. J. Num. Meth. Fluids, **19**, pp 415-437, 1994.

[86] D.L.Young, "Time-Dependent Multi-Material Flow with Large Fluid Distortion", Numerical Methods for Fluid Dynamics, ed. K.W.Morton and M.J.Baines, Academic Press, New-York, pp 272-285, 1982.

[87] B.van Leer, "Towards the Ultimate Conservative Difference Scheme, IV: A New Approach to Numerical Convection", J. Comput. Phys., **23**, pp 263-275, 1977.

[88] D.B.Kothe, R.C.Mjolsness, "RIPPLE: A New Model for Incompressible Flows with Free Surfaces", AIAA Journal, **30**, No. 11, pp 2694-2700, 1992.

[89] E.C.DuFort, S.P.Frankel, "Stability Conditions in the Numerical Treatment of Parabolic Differential Equations", Math. Tables and Other Aids to Computation, **7**, pp 135-152, 1953.

[90] R.O.Stafford, A.B.Rice and D.F.Pinella, "Investment Casting Process Design - Part 2: Solidification Simulation", Proc. ASME Intl. Computers in Engng. Conf., Aug. 9-13, pp 437-445, 1987.

[91] J.F.Evans, J.Beech and D.H.Kirkwood, "The Determination of Metal/Mold Interfacial Heat Transfer Coefficients and the Prediction of Gross Shrinkage Cavities in Chill Mold Castings", in "Modeling of Casting Welding and Advanced Solidification Processes V", eds. M.Rappaz, M.R.*Özgü* and K.W.Mahin, TMS Publication, pp 531-538, 1991.

[92] L.E.Smiley, "Use of a Personal Computer to Predict Casting Heat Flow and Solidification", AFS Transactions, **67**, pp 689-696, 1988.

[93] M.Lipinski, W.Schäfer and S.Andersen, "Modeling of Combined Heat and Fluid Flow for Determination of Filling Sequence for Real Complex Shaped Castings", in "Modeling of Casting Welding and Advanced Solidification Processes V", eds. M.Rappaz, M.R.*Özgü* and K.W.Mahin, TMS Publication, pp 771-776, 1991.

[94] H.Huang, J.L.Hill and J.T.Berry, "A Free Thermal Contraction Method for Modelling the Heat-Transfer Coefficient at the Casting/Mould Interface", Cast Metals, **5**, No. 4, pp 212-216, 1993.

[95] J.A.Dantzig, S.C.Lu, "Modeling of Heat Flow in Sand Castings: Part I. The Boundary Curvature Method", Metallurgical Transactions B, **16B**, pp 195-202, 1985.

[96] J.A.Dantzig, J.W.Wiese, "Modeling of Heat Flow in Sand Castings: Part II. Applications of the Boundary Curvature Method", Metallurgical Transactions B, **16B**, pp 203-209, 1985.

[97] J.W.Wiese, J.A.Dantzig, "SPIDER and the Boundary Curvature Method: Simulating the Solidification of Foundry Castings", Appl. Math. Modelling, **12**, pp 213-220, 1988.

[98] T.X.Hou, R.D.Pehlke and J.O.Wilkes, "Solidification Modeling: Reduction of Computing Time in the Mold Region", AFS Transactions, **101**, pp 325-332, 1991.

[99] J.H.Chen and H.L.Tsai, "An Efficient and Accurate Numerical Algorithm for Multi-Dimensional Modeling of Casting Solidification, Part I: Control Volume Method", AFS Transactions, pp 539-546, 1990.

[100] C.H.Su, H.L.Tsai, "A Direct Method to Include Latent Heat Effect for Modelling Casting Solidification", AFS Transactions, **82**, pp 781-789, 1991.

[101] P.R.Sahm, P.N.Hansen, "Numerical Simulation and Modelling of Casting and Solidification Processes for Foundry and Casthouse", CIATF, 1984.

[102] I. Ohnaka, T.Fukusako, "Calculation of Solidification of Casting by a Matrix Method", ISIJ Transactions, **17**, pp 410-418, 1977.

[103] M.I.Thambau, V.Panchanathan, ?, AFS Transactions, **88**, p.167, 1980.

[104] M.J.Mundin, M.Fortes, "An Accurate Finite Element Method of Solution of Phase Change Problems, Based on Enthalpy Diffusion", ?.

[105] J.H.Chen and H.L.Tsai, "Comparison of Different Modes of Latent Heat Release for Modeling Casting Solidification", AFS Transactions, pp 539-546, 1990.

[106] K.G.Upadhya, D.M.Stefanescu, K.Lieu and D.P.Yeager, "Computer-Aided Cooling Curve Analysis: Principles and Applications in Metal", AFS Transactions, **47**, pp 61-66, 1989.

[107] S.Z.Uram ?, ?, AFS Transactions, **68**, p. 347, 1960.

[108] Q.T.Fang, D.A.Granger, "Porosity Formation in Modified and Unmodified A356 Alloy Castings", AFS Transactions, **209**, pp 989-1000, 1989.

[109] J.G.Henzel, J.Keverian, "The Theory and Application of a Digital Computer in Predicting Solidification Patterns", Journal of Metals, **17**, No. 5, pp 561-568, 1965.

[110] I.Imafuku, K.Chijiiwa, "Application and Consideration of the Shrinkage Cavity Prediction Method", AFS Transactions, **9**, pp 463-474, 1983.

[111] I.Imafuku, K.Chijiiwa, "A Mathematical Model for Shrinkage Cavity Prediction in Steel Castings", AFS Transactions, **10**, pp 527-540, 1983.

[112] Y.Nagasaka, S. Kiguchi and M.Nachi, "Three-Dimensional Computer Simulation of Casting Processes", AFS Transactions, **117**, pp 553-564, 1989.

[113] V. de L.Davies, ?, AFS Cast Metals Res. J., **11**, p. 33, 1975.

[114] J.A.Spittle, S.G.R.Brown, "Computer Simulation of the Effects of Alloy Variables on the Grain Structures of Castings", Acta Metallurgica, **37**, pp 1803-1810, 1989.

[115] E.Niyama, T.Uchida, M.Morikawa and S.Saito, "Predicting Shrinkage in Large Steel Castings from Temperature Gradient Calculation", AFS Int. Cast Metals Res. J., **6**, pp 16-22, 1981.

[116] E.Niyama, T.Uchida, M.Morikawa and S.Saito, "A Method of Shrinkage Prediction and Its Application to Steel Casting Practice", AFS Int. Cast Metals Res. J., **9**, pp 52-63, 1982.

[117] H.Huang, V.K.Suri, N.El-Kaddah and J.T.Berry, "The Effect of Interdendritic Feeding on Microporosity Formation", in "Modeling of Casting,-Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 219-226, 1993.

[118] V.Jesko, J.Zajac, "A New Method for Shrinkage Prediction in Casting Walls", AFS Transactions, **48**, pp 681-684, 1991.

[119] K.Kubo, R.D.Pehlke, ?, AFS Transactions, **94**, p. 753, 1986.

[120] J.F.Evans, J.Beech and D.H.Kirkwood, "The Prediction of Shrinkage Cavities in Cast Steel Rolls", Cast Metals, **5**, No. 3, pp 130-141, 1992.

[121] Y.D.Fryer, C.Fryer, M.Cross and P.Chow, "Predicting Macro Porosity in Shape Castings Using an Integrated Control Volume Unstructured Mech (CV-UM) Framework", in "Modeling of Casting, Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 143-151, 1993.

[122] M.Barkhudarov, H.You, J.Ortega, J.Beech, S.B.Chin, D.H.Kirkwood, "Experimental Validation and Development of FLOW-3D for Casting Problems", in "Modeling of Casting, Welding and Advanced Solidification Processes VI", eds. T.S.Piwonka, V.Voller and L. Katgerman, TMS Publication, pp 421-434, 1993.

[123] C.W.Hirt, J.M.Sicilian, "A Porosity Technique for the Definition of the Obstacles in Rectangular Cell Meshes", Proc. Fourth Int. Conference Ship. Hydro., National Academy of Science, Washington, DC, September, pp 212-235, 1985.

[124] M.Ishii and T.C.Chawla, "Local Drag Laws in Dispersed Two-Phase Flow", Argonne Nat. Lab. report NUREG/CR-1230, 1979.

[125] V.R.Voller, "An Overview of the Modelling of Heat and Fluid Flow in Solidification Systems", in "Modeling of Casting Welding and Advanced Solidification Processes V", eds. M.Rappaz, M.R.*Özgü* and K.W.Mahin, TMS Publication, pp 661-674, 1991.

[126] J.F.Evans, "Central Unsoundness in Cast Steel Rolls", Ph.D. Thesis, The University of Sheffield, 1990.

[127] H.You, "Computer and Physical Modelling of Heat and Fluid Flow in Relation to the Solidification of Alloys", Ph.D. Thesis, The University of Sheffield, in preparation.

[128] CASM ACME Review Report No.3, The University of Sheffield, 1993.

[129] J.Beech, S.B.Chin, D.H.Kirkwood, M.Barkhudarov, H.You, R.I.Kangley, J.Ortega, "Computer Modelling of Flow and Solidification of Alloys", ACME Review Reports, The University of Sheffield, 1993-94.

[130] A.L.Kearney, "Properties of Cast Aluminium Alloys", in Metals Handbook, tenth edition, Vol.2, pp 152-154, 1990.

[131] A.F.A.Hoadley, D.M.Scott and T.J.Smith, "The Thermal Boundary Layer in a Convecting Melt", in "Heat Transfer and Phase Change Problems", Eurotherm Seminar 6, Extended Abstracts, Deyt, Holland, pp 10-12, 1988.

[132] A.L.Maples, D.R.Poirier, "Convection in the Two-Phase Zone of Solidifying Alloys", Metallurgical Trans. B, **15B**, pp 163-172, 1984.

[133] Hai-Lung Tsai, "Studies on Shrinkage Effect During Casting Solidification", in "Modeling of Casting Welding and Advanced Solidification Processes V", eds. M.Rappaz, M.R.*Özgü* and K.W.Mahin, TMS Publication, pp 749-754, 1991.

[134] R.D.Pehlke, A.Jeyarajan and H.Wada, "Summary of Thermal Properties for Casting Alloys and Mold Materials", Report No. NSF/MEA-82028, University of Michigan, December 1982.

[135] A.I.Vazhnik, "A method for Forecasting Time-Averaged Fields of Meteorological Elements on the Basis of a Barotropic Model of the Atmosphere", Meteorologia y Gidrologia, No. 9, pp 30-42, 1988.

[136] A.Robert, T.L.Yee and H.Ritchie, "A Semi-Lagrangian and Semi-Implicit Numerical Integration Scheme for Multilevel Atmospheric Models", Monthly Weather Review, **113**, pp 388-394, 1985.

Appendix A

Summary of Computer Simulations

Appendix B

Semi-Lagrangian (Leith's) Method for SOLA

The application of the Leith's method to the discretisation of the advection terms in the momentum equation in two dimensions in the absence of free surfaces is described in this Appendix. This advection scheme is explicit and unconditionally stable. It posseses the same properties as the upwind differencing scheme, that is it is transportive and conservative. It can be shown that compared to the upwind differencing scheme the Leith's method introduces less numerical diffusion. This because first order truncation errors in the discretisation of the advection terms in the Leith's method do not contain terms proportional to (for x-component)

$$\frac{\partial^2 u}{\partial x \partial y}, \; \frac{\partial^2 u}{\partial x \partial z} \; and,, \; \frac{\partial^2 u}{\partial z \partial y}.$$

The conservation equation for the x-component of the velocity $u(t, \mathrm{x})$ is

$$\frac{\partial u}{\partial t} + (\mathrm{v} \cdot \nabla) u = S \tag{B.1}$$

where $S(t, \mathrm{x})$ comprises pressure, body force and viscous terms, may be written using the *substantive* derivative rather than the partial:

$$\frac{du}{dt} = S \tag{B.2}$$

Integration along the trajectory $\mathrm{x}(t)$ between t and t_0 gives

$$u(t, \mathrm{x}(t)) = u(t_0, \mathrm{x}(t_0)) + \int_{t_0}^{t} S(\tau, \mathrm{x}(\tau)) \, d\tau \tag{B.3}$$

where τ is the integration parameter. The trajectory $\mathbf{x}(t)$ is defined by ordinary differential equations which, in two dimensions, are

$$\frac{dx}{dt} = u, \qquad \frac{dy}{dt} = v \tag{B.4}$$

Equation (B.3) is solved at each node with $t_0 = t^n$ and $t = t^{n+1}$ and $\mathbf{x}(t^{n+1})$ defined as

$$x_{n+1} = x_{i,j}, \qquad y_{n+1} = y_{i,j}$$

Equation (B.4) is solved numerically using the n-level velocities to obtain the position of the $\mathbf{x}(t^n)$ point. In general, x_n and y_n will not fall on a computational node. A linear interpolation from four (eight in 3-D) nearest nodes is used to obtain the value of $u(t^n, x_n, y_n)$. If the point (x_n, y_n) lies *between* these nodes (in other words *interpolation*, and not *extrapolation*, is used) then the *unconditional* numerical stability of the method with respect to advection is ensured [135].

Roache [27] mistakenly concludes that the Leith's method is only conditionally stable with the same time step restriction as the explicit upwind differencing method. The mistake arises from the implicit assumption by Roache that $u(t^n, x_n, y_n)$ is always interpolated from the same node points. In that case, of course, the point (x_n, y_n) has to fall between these nodes to maintain stability and that is the origin of the time step restriction.

The integral of the function S can be estimated independently of the advection terms. Perhaps the simplest approximation is to use the value of S at the node (i,j) at time t^n, *i.e.* use the explicit approach:

$$\int_{t_0}^{t} S(\tau, \mathbf{x}(\tau))\, d\tau \approx \Delta t\, S(t^n, x_{i,j}, y_{i,j}) = \Delta t \cdot S^n_{i,j} \tag{B.5}$$

The representation of the source term given in Eq. (B.5) is equivalent to an explicit formulation and is first order accurate in time. It may introduce a stability time step limit, *e.g.* that given by Eq. (2.76) if S includes viscous stresses.

The semi-Lagrangian method in the form of Eqs. (B.2-B.5) is an efficient algorithm because it combines the advantage of the unlimited time-step size for advection and the ease of the matrix inversion characteristic of standard explicit advection methods.

If linear interpolation is used for evaluating $u(t^n, x_n, y_n)$, then a numerical diffusion and a *phase* error are introduced [27]. The latter means that perturbations of different wavelength propagate at different speeds in the numerical solution even for a uniform velocity field. This is of course of no concern if a steady state solution is sought. The diffusion and phase error effects can be reduced or even removed completely in two ways. One is to obtain a more accurate solution x_n and y_n of the trajectory Eq. (B.4) by splitting Δt into several smaller time steps. Solving equations at each of the intermediate step will involve also an estimation of the velocities at each intermediate trajectory point by interpolation. The other way is to use a higher order interpolation to estimate $u(t^n, x_n, y_n)$. Normally it would involve more nodes around (x_n, y_n) but a linear interpolation will still be necessary at the boundaries.

Apart from unconditional stability for the advection terms, the method also possesses the conservative and transportive properties [27]. It is interesting no note that, unlike most other higher-order schemes, the Leith's method preserves the transportive property even in its higher-order accuracy formulation.

The Leith's method has been well developed and extensively applied in the atmospheric circulation modelling [135,136]. The absence of solid boundaries in the horizontal direction removes the difficulty of resolving the trajectories near walls as well as allows one to use a higher-order accuracy version of the method throughout the computational domain.

It appears possible to use this method successfully for an incompressible fluid flow in the presence of walls. Here the method is incorporated into SOLA program [53] and applied to a two-dimensional flow problem. In a staggered grid (Fig. 2.1*a*) the calculation of the first guess for u-velocity in cell (i,j) at $t = t^{n+1}$ proceeds as follows:

1. coordinates x_{a_0} and y_{a_0} are chosen to be at the nodal point where velocity $u_{i,j}$ is located (point a_0 in Fig. B.1*a*) so that

$$u_{a_0} = u_{i,j}$$

2. the y-direction velocity component v_{a_0} at point a is estimated by linear interpolation which in a uniform grid is

$$v_{a_0} = \frac{1}{4}\left(v_{i,j-1} + v_{i,j} + v_{i+1,j-1} + v_{i+1,j}\right)$$

3. Equation (B.4) are solved by dividing the time step into m equal parts. For each step l with $\Delta t_m = \Delta t/m$, coordinates x_{a_l} and y_{a_l} of an intermediate point a_l are found by

$$x_{a_l} = x_{a_{l-1}} - u_{a_{l-1}} \cdot \Delta t_m, \qquad y_{a_l} = y_{a_{l-1}} - v_{a_{l-1}} \cdot \Delta t_m \qquad \text{(B.6)}$$

velocities u_{a_l} and v_{a_l} at each point a_l are found by linear interpolation from the neighbouring nodes.

4. velocity u_{a_m} at the final point a_m is found by linear interpolation from the four neighbouring nodes;

5. the new time level velocity at point a_0 is estimated as

$$u_{i,j}^{n+1} = u_{a_m} + \Delta t \cdot S_{i,j}^{n} \tag{B.7}$$

6. a similar procedure is carried out for the v-component of the velocity.

If an intermediate point $a_{l'}$ appears to be in a cell blocked by a wall (for simplicity partially blocked cells will not be considered), then the time step Δt_m is divided by half and the solution is restarted from the previous point $a_{l'-1}$. If the situation repeats, then Δt_m is reduced further by a factor of four and so on. The fact that this procedure will not carry on infinitely can be demonstrated on the following one-dimensional example.

Suppose that the space $x \leq 0$ is blocked by an obstacle and there is a computational cell covering a segment $0 < x < 1$. One face of the cell ($x = 0$) is blocked by the wall and the velocity component u_0 on the opposite face ($x = 1$) points in the direction away from the wall, $u_0 > 0$ (Fig. B.1*b*). The velocity between the two faces is defined by linear interpolation

$$u = x \cdot u_0$$

In this case the solution of equation $dx/dt = u$ is

$$x = x_0 \cdot e^{u_0 t}$$

where x_0 is a constant of integration. If at $t = t_1$ $x_1 = 1$ then at $t = t_2 < t_1$

$$x_2 = e^{-u_0(t_1 - t_2)} > 0$$

This equation means that whatever the integration time $t_1 - t_2$ is, the analytical solution for x_2 is always positive and *outside* the obstacle (though it can be infinitely close to it if $u_0 > 0$). In terms of the numerical solution it means that if the time period between t_2 and t_1 is divided into sufficiently small sub-time steps, then the numerical result should be close to the analytical one, that is, also outside the wall.

The modified SOLA program is given in appendix C. The Leith's method was used here to simulate the inviscid flow in a 'water-cooled reactor' described by Hirt *et al* [53]. Figure B.2 shows the geometry and boundary conditions setup for the problem. The velocity at the entrance was constant and set equal to $u = -0.01\ m/s$. A 20×30 mesh, uniform in each direction, was used with $\Delta x = 5\ mm$ and $\Delta y = 6.5\ mm$. With the upwind differencing scheme the time step size was chosen to be $\Delta t = 0.05\ s$ to maintain numerical stability. Figure B.3 shows the result for $\Delta t = 0.07\ s$ where an instability occurred at $t = 1.7\ s$ of simulation due to a violation of the CFL criterion. The total simulation time was $8.0\ s$. The convergence criterion was $\epsilon = 0.005\ s^{-1}$ in all simulations. Figure B.4 shows velocity plots at four different times produced by the unmodified SOLA. The total CPU time was $t_{cpu} = 21.0\ s$ with the average iteration count equal $N_{it} = 21$. To simplify comparisons, the CPU times and average iteration counts were normalised to those of the unmodified SOLA simulation. Therefore, for the latter $t_{cpu} = 1$ and $N_{it} = 1$.

For the Leith's method four time step sizes were used: $\Delta t_1 = 0.05\ s$, $\Delta t_2 = 0.1\ s$, $\Delta t_3 = 0.2\ s$ and $\Delta t_4 = 0.5\ s$. Figure B.5 displays velocity plots for $\Delta t = 0.05\ s$ and $m = 1$ at the same times as in Fig. B.4 showing close agreement with the upwind method predictions. In this case $t_{cpu} = 1.11$ and $N_{it} = 1.17$.

Figure B.6 shows velocity plots at $t = 6\ s$ for $\Delta t = 0.05\ s$ and $m = 1$, $m = 2$, $m = 5$ and $m = 10$. The difference between these predictions is small. The dependence of N_{it} on parameter m is shown in Fig. B.7 for $\Delta t = 0.05\ s$ and $\Delta t = 0.5\ s$. Though an increase in iteration count can be seen for intermediate values of m, convergence improves as m grows. The latter is because a more accurate solution of Eqs. (B.4) is obtained if m is large, giving a better estimate for the first-guess velocities.

Figure B.8 shows velocity plots at $t = 6.0\ s$ for the four time step sizes with $m = 1$. It is clear that the solution was stable in all cases though its accuracy deteriorated as the time step size increased. This also lead to slower convergence as shown in Figure B.9 where the dependence of average iteration count on the time step size is shown. Fig. B.10 gives predicted flows for $\Delta t = 0.5\ s$ and $m = 1$, $m = 2$, $m = 10$ and $m = 20$. As in Fig. B.6, little improvement can be seen with the increase of m. This can be explained by the crude approximation for the pressure gradient terms, given by Eq. (B.5). The larger the time step size the less accurate is the assumption that the work of the pressure forces along the fluid particle trajectory can be approximated by the work of the pressure forces at only one point of the trajectory, made in Eq. (B.5).

Finally, Fig. B.11 summarises CPU times of all simulations in this Appendix as functions of the time step size and the value of m. It can be seen that none of the simulations employing the Leith's method was more efficient than the original method.

The potential of the Leith's method can be realised more efficiently if the integral on the right-hand side of Eq. (B.3) is approximated more accurately. This will give a better estimate for the first-guess velocities, therefore, accelerating convergence an improving the overall accuracy of the solution.

It is unclear as to the best means of applying this method to transient free surface flow. In the SOLA-VOF algorithm velocities in the void are not known, therefore Eq. (B.4) cannot be solved in empty cells which are being filled with fluid during the current time step. A possible means is to use the variable density method (Section 2.4) in which velocities are estimated both in the fluid and in the air. This will require a higher order version of the Leith's method to avoid a diffusion of the fluid/air interface.

The use of a staggered grid for the Leith's method reduces the efficiency of the algorithm because in every computational cell velocity components are defined at different locations. Therefore Eq. (B.4) has to be solved separately for each velocity component. The starting and finishing points of the solution of Eq. (B.4) are different for each component and interpolation coefficients are also different. In a colocated grid all three velocity components are located at the cell centre. In that case Eq. (B.4) would have to be solved only once with the cell centre as the starting point. The interpolation coefficients would also be computed just once for the finishing point of the solution of Eq. (B.4).

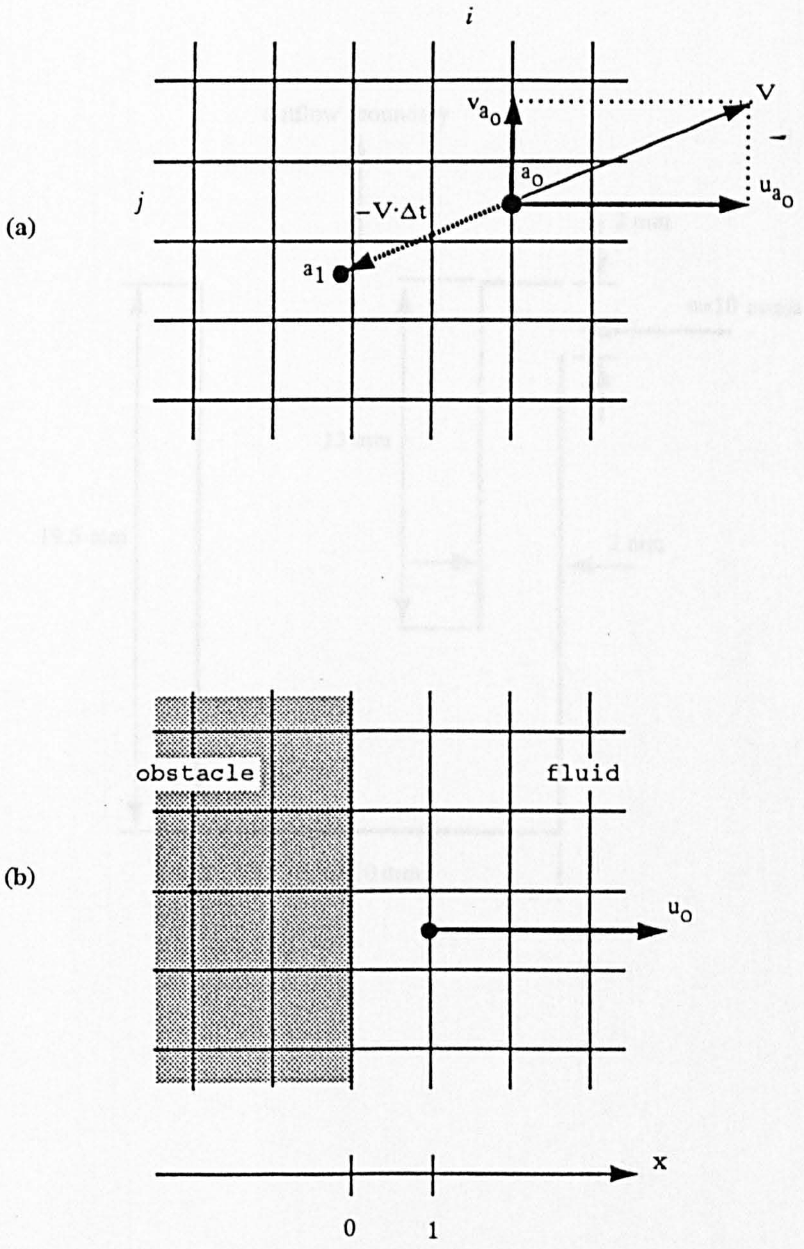


Figure B1 (a) The application of the Leith's method to cell *(i,j)* with *m=1*. (b) The use of the Leith's method at walls

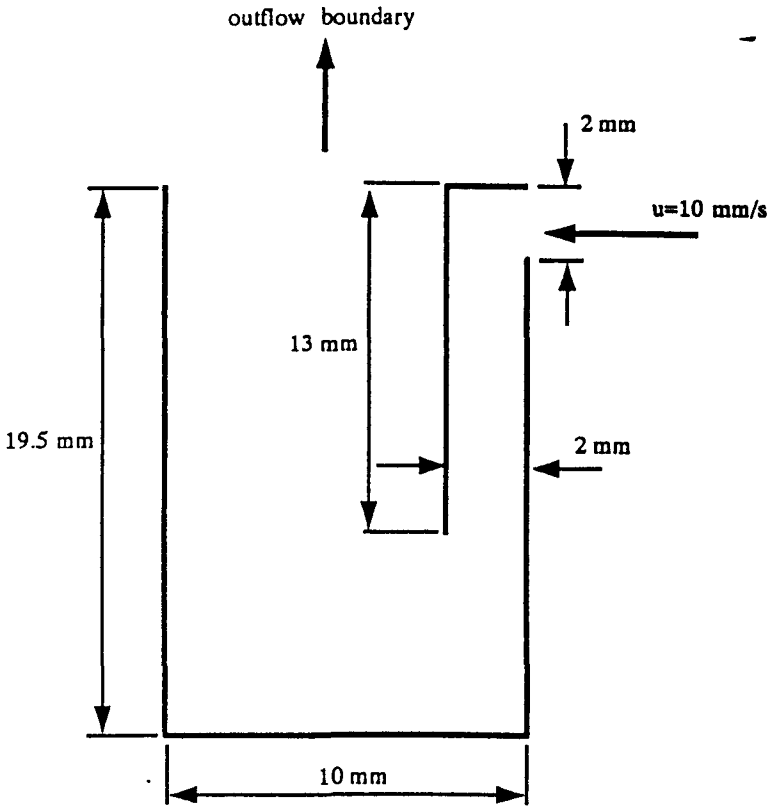
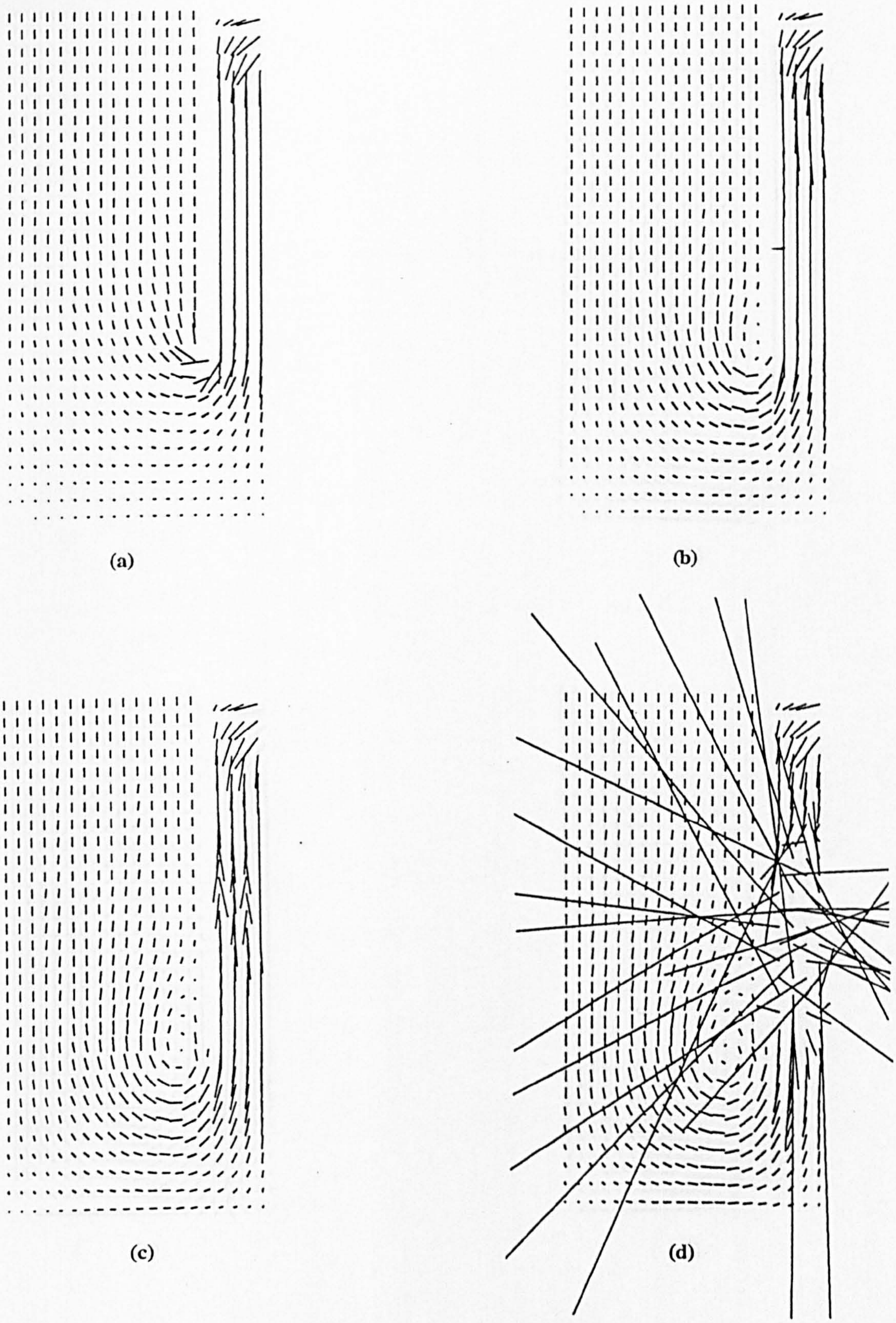


Figure B2 Geometry and boundary conditions for the 2-D test flow for the Leith's method.



(a)

(b)

(c)

(d)

Figure B3 Velocity vectors calculated using the upwind differencing method: Δt=0.07 s. (a) t=0.42 s, (b) t=0.84 s, (c) t=1.26 s and (d) t=1.68 s. The onset of the instability due to too large the time step is clearly vesible.

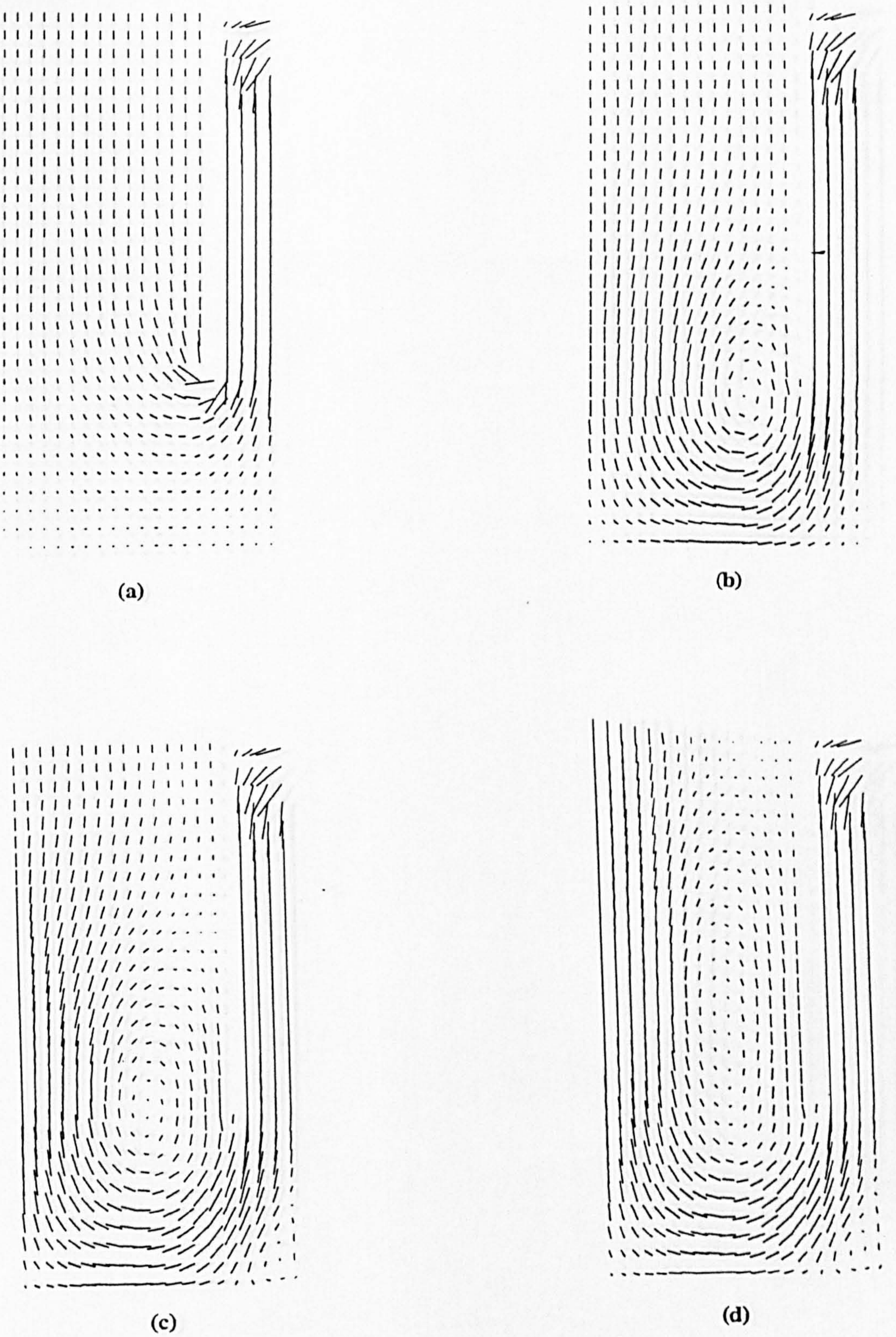
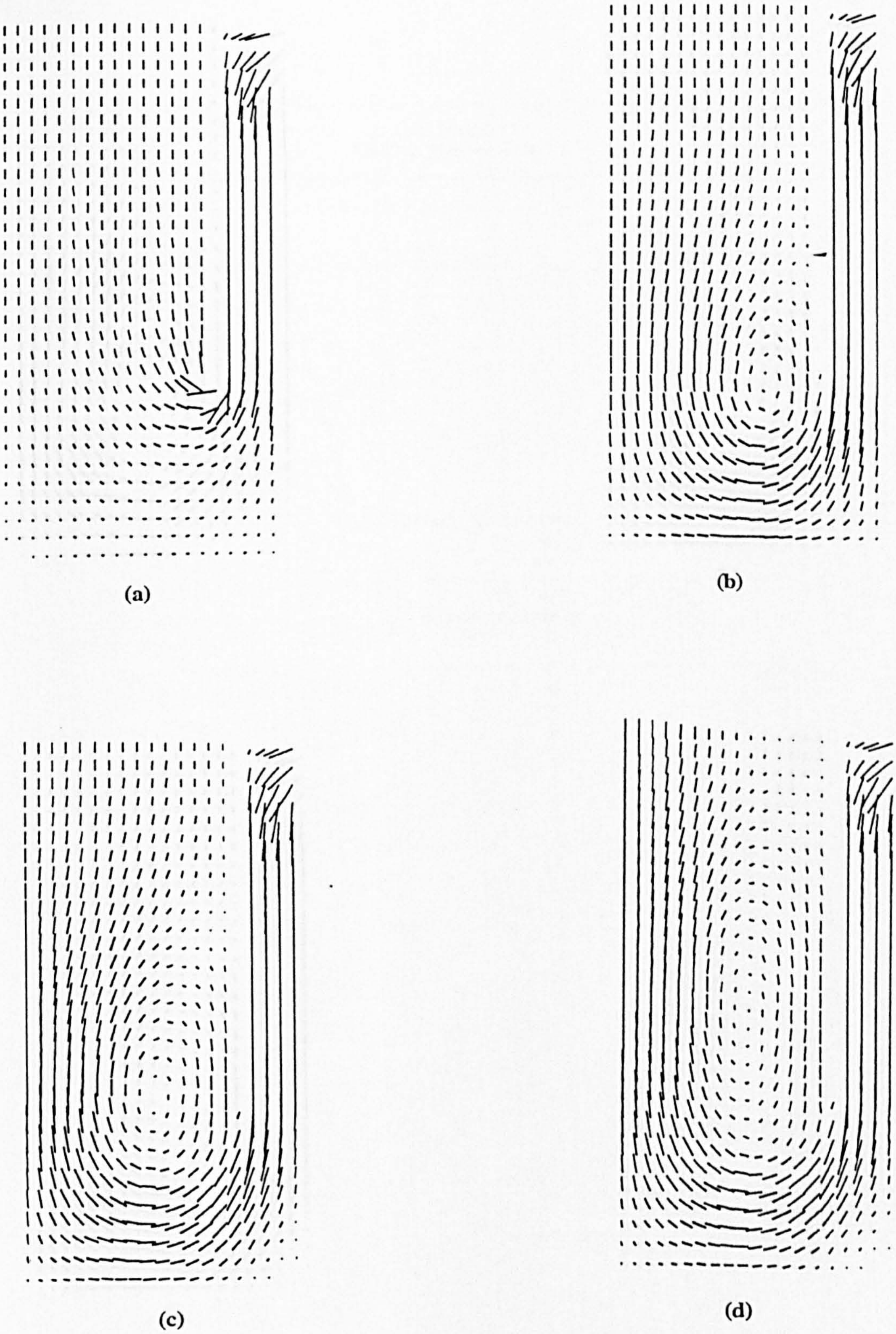


Figure B4 Velocity vectors calculated using the upwind differencing method: Δt=0.05 s.(a) t=2.0 s,(b) t=4.0 s, (c) t=6.0 s and (d) t=8.0 s.



(a)

(b)

(c)

(d)

Figure B5 Velocity vectors calculated using the Leith's (semi-Lagrangian) method: Δt=0.05 s, m=1. (a) t=2.0 s,(b) t=4.0 s, (c) t=6.0 s and (d) t=8.0 s.

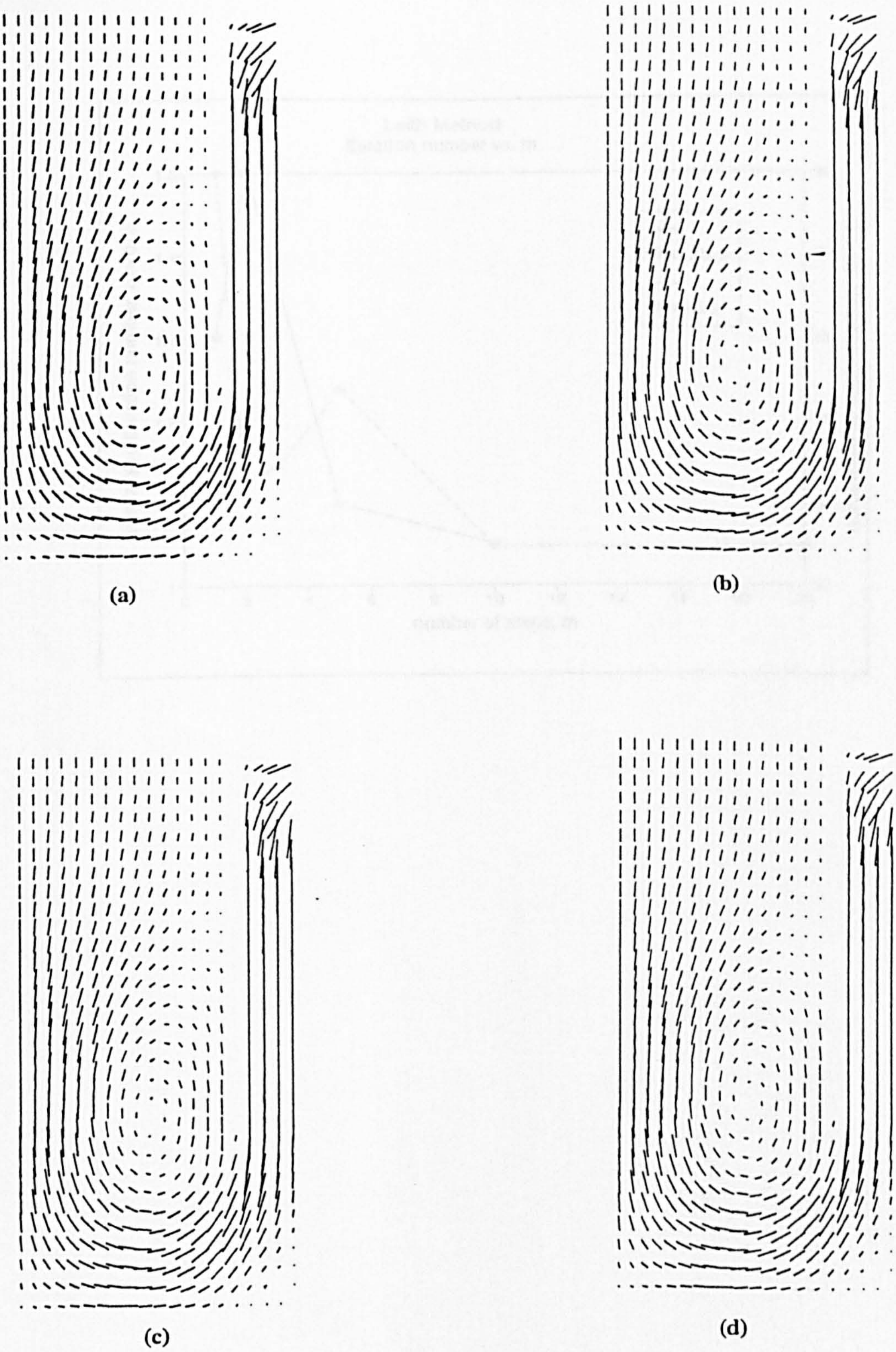


Figure B6 Velocity vectors calculated using the Leith's method at t=6 s: Δt=0.05 s, (a) m=1,(b) m=2, (c) m=5 and (d) m=10.

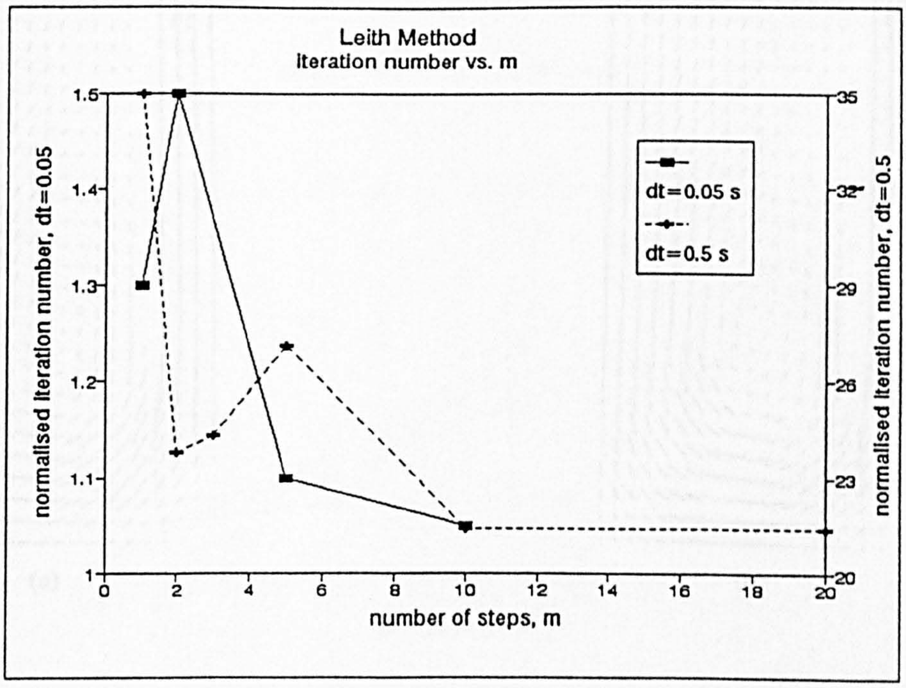


Figure B7 Average iteration number in the Leith's method as function of *m* for Δt=0.05 s (left *y*-axis) and Δt=0.5 s (right *y*-axis)

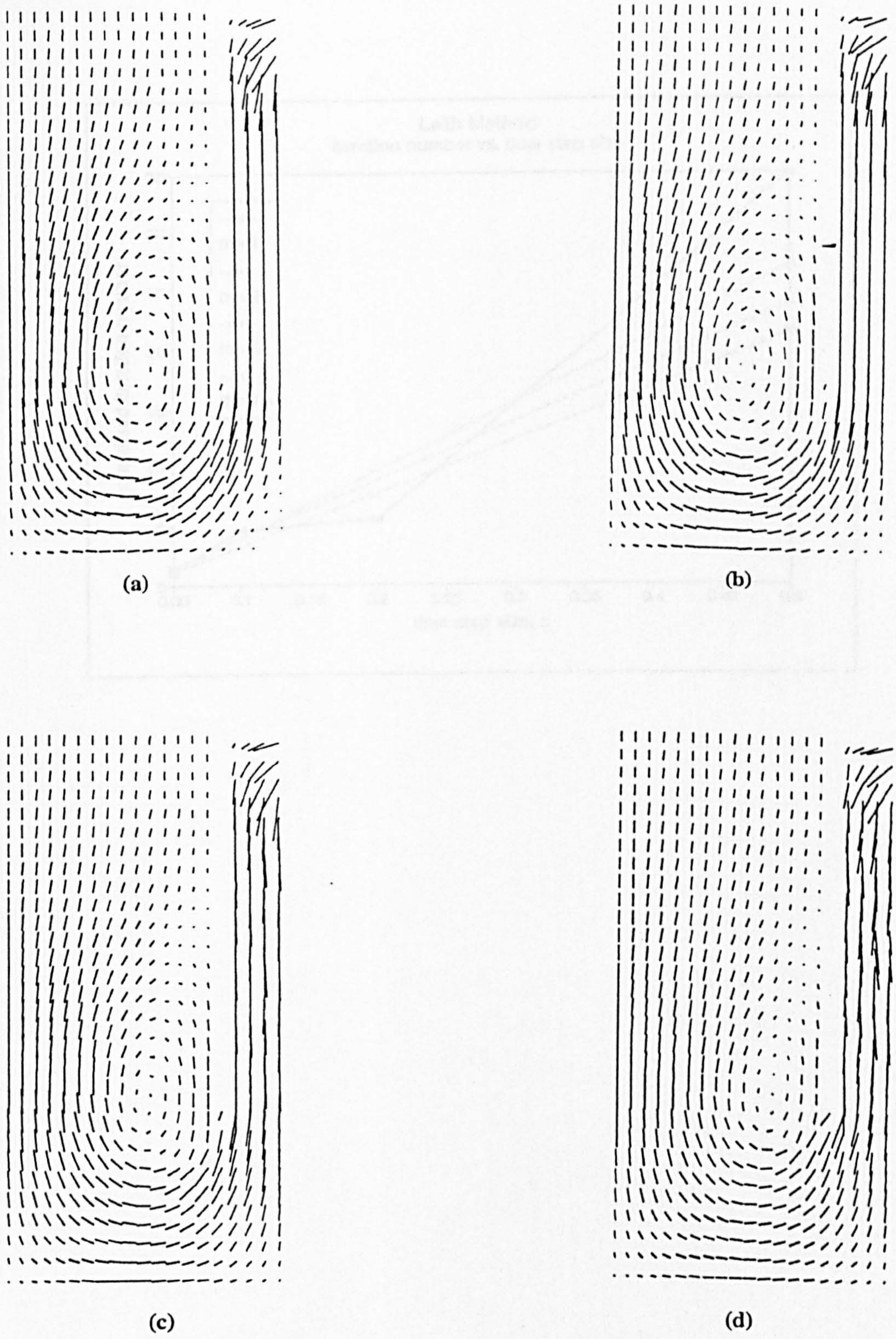


Figure B8 Velocity vectors calculated using the Leith's method at t=6 s: m=1, (a) Δt=0.05 s,(b) Δt=0.1 s, (c) Δt=0.2 s and (d) Δt=0.5 s.

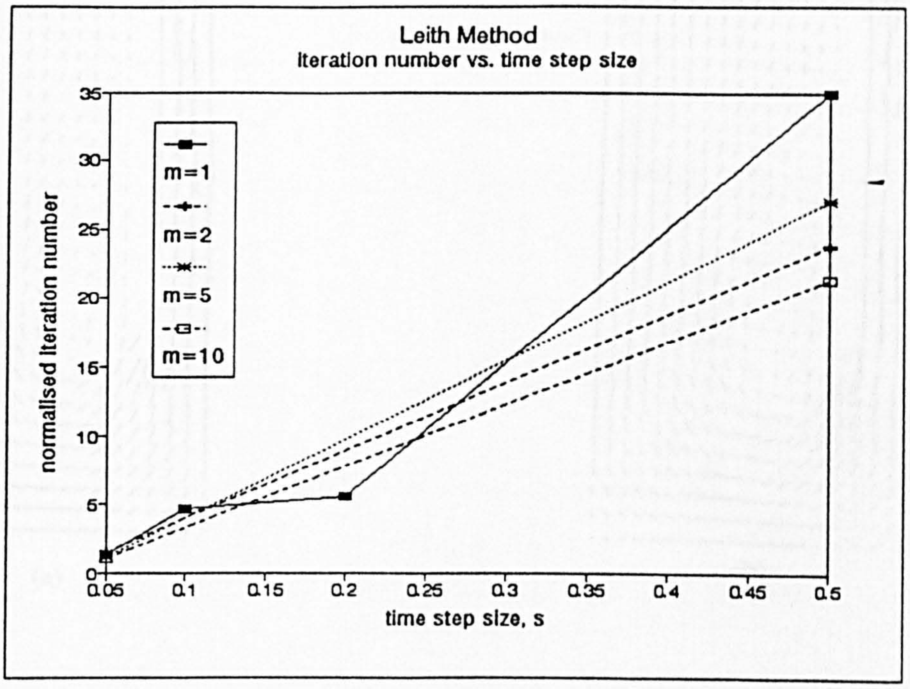


Figure B9. Average iteration number as function of the time step size for the Leith's method with a range of m values.

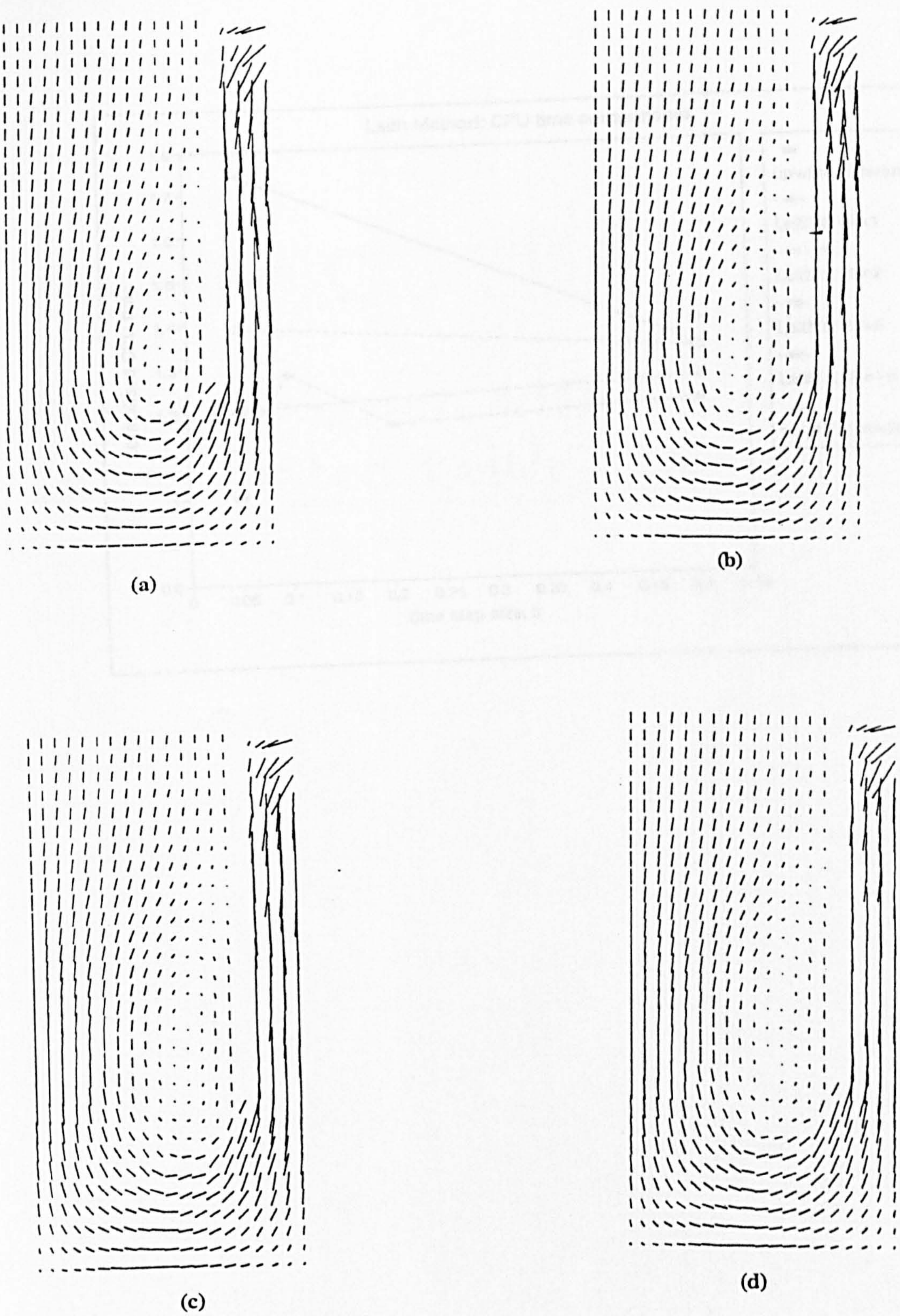


Figure B10 Velocity vectors calculated using the Leith's method at t=6 s: Δt=0.5 s, (a) m=1, (b) m=2, (c) m=10 and (d) m=20.

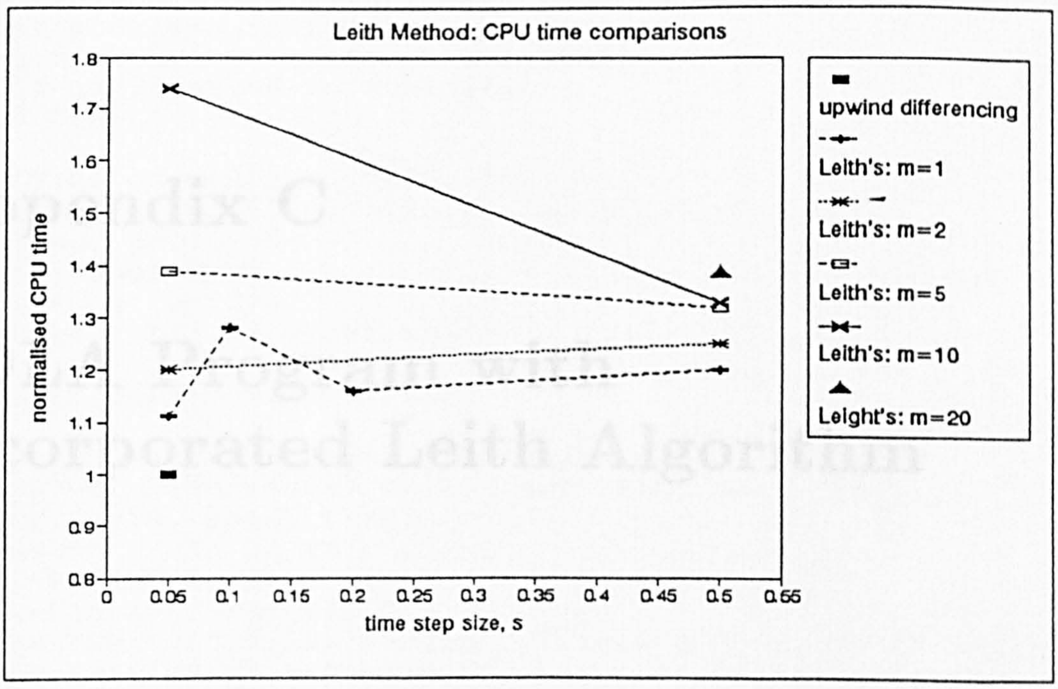


Figure B11 Total CPU time as function of the time step size for the original SOLA upwind differencing method (filled square) and the Leith's (or semi-Lagrangian) method for a range of *m* values.

Appendix C

SOLA Program with Incorporated Leith Algorithm

```
       program sola
c
c Modified SOLA program with linear Leith's advection
c term approximation instead of the original
c upwind-differencing scheme. The Leith's
c method is implemented in subroutines
c EQTN and GINTER
c
        dimension u(36,36),v(36,36),
     * un(36,36),vn(36,36),p(36,36),xput(24)
      real nu, pn(36,36)
      real*8 cpu1,cpu2
      real*4 tarray(2)
      integer cycle,wl,wr,wt,wb
c
      integer nf(36,36)
c
        common /dat/ nf, neqn, imax, jmax
        common /mesh/ delt, delx, rdx, xmin, xmax, dely,
     *  rdy, ymin, ymax, rrxy
        common /vel/ un,vn
      call etime(tarray)
      cpu1=tarray(1)+tarray(2)
        open(unit=10,file="semiin",status="old",form="formatted")
        open(unit=9,file="semiout",status="old",form="formatted")
c
c     read and print initial input data
c
        xmin=0.0
        xmax=1.0
        ymin=0.0
        ymax=1.0
      read(10,26) num
      read(10,25) (xput(i),i=1,num)
      ibar=xput(1)
      jbar = xput(2)
      delx = xput(3)
      dely = xput(4)
      delt = xput(5)
      nu = xput(6)
      cyl = xput(7)
      epsi = xput(8)
      dzro = xput(9)
      gx = xput(10)
      gy = xput(11)
      ui = xput(12)
      vi = xput(13)
      velmx = xput(14)
      twfin = xput(15)
      cwprt = xput(16)
      cwplt = xput(17)
      omg = xput(18)
      alpha = xput(19)
      wl = xput(20)
      wr = xput(21)
      wt = xput(22)
      wb = xput(23)
      neqn=xput(24)
      write(9,50)(xput(i),i=1,num)
  25 format(4(6x,e12.5))
```

```
   26 format(6x,i2)
   27 format(1h ,18x,10a8,1x,a10,2(1x,a8))
   35 format(1h1)
   44 format(6x,7hcycle= ,i5,8x,4htd= ,1pe12.5,8x,
     1     4ht2= ,e12.5,9x,6hiter= ,i5)
   45 format(10a8)
   71  format(1x,i4,e10.3,i4,/,100(3e10.3))
   72  format(1x,i4,5x,3(6x,1pe10.3))
   46 format(80x,3ht= ,1pe10.3,4x,7hcycle= ,i4)
   47 format(6x,1hi,7x,1hj,12x,1hu,17x,1hv,18x,1hp)
   48 format(4x,i3,5x,i3,3(6x,1pe12.5))
   49 format(6x,6hiter= ,i5,10x,6htime= ,1pe12.5,10x,7hcycle= ,i4)
 50   format(1x,e12.5)
c   50 format(1h ,5x,6hibar= ,1pe12.5/6x,6hjbar= ,e12.5/
c     16x,6hdelx= ,e12.5/6x,6hdely= ,e12.5/6x,6hdelt= ,
c     2e12.5/8x,4hnu= ,e12.5/7x,5hcyl= ,e12.5/6x,6hepsi= ,
c     3e12.5/6x,6hdzro= ,e12.5/8x,4hgx= ,e12.5/8x,4hgy= ,
c     4e12.5/8x,4hui= ,e12.5/8x,4hvi= ,
c     5e12.5/5x,7hvelmx= ,e12.5/5x,7htwfin= ,e12.5/5x,
c     67hcwprt= ,e12.5/5x,7hcwplt= ,e12.5/7x,5homg= ,
c     7e12.5/5x,7halpha= ,e12.5/8x,4hwl= ,e12.5/8x,
c     84hwr= ,e12.5/8x,4hwt= ,e12.5/8x,4hwb= ,e12.5)
cc
c     compute constant terms and initialize necessary variables
c
      imax=ibar+2
      jmax=jbar+2
      im1=imax-1
      jm1=jmax-1
      rdx=1.0/delx
      rdy=1.0/dely
      rrxy=rdx*rdy
        delx2=delx**2
        dely2=dely**2
        rdx2=rdx**2
        rdy2=rdy**2
        dxdy=delx*dely
        dtdx=delt*rdx
        dtdy=delt*rdy
      jm2=jmax-2
      im2=imax-2
c
c * * NF is used as a flag for blocked and open cells
c * * NF=1 - open cell, NF=0 - blocked cells
c
        do 10 i=2,im1
        do 10 j=2,jm1
 10     nf(i,j)=1
c
      t=0.
      iter=0
      cycle=0
      twprt=0.
      twplt=0.
      beta= omg/(2.*delt*(rdx**2+rdy**2))
c
c     special input data
c
c
c     set initial velocity field into u and v arrays
```

```
c
      do 560 i=2,im1
      do 560 j=2,jm1
      u(i,j)= ui
      v(i,j)= vi
  560 continue
      assign 5000 to kret
      go to 2000
c
c     start cycle
c
 1000 continue
      iter=0
      flg=1.
      assign 3000 to kret
c
c     compute temporary u and v
c********************************************************************
c * * temporary u and v are computed in EQTN * *
c*********************************************************************
      do 1111 i = 2,im1
       xxu=xmin+delx*(i-1)
        xxv=xxu-0.5*delx
      do 1111 j = 2,jm1
       yyv=ymin+dely*(j-1)
        yyu=yyv-0.5*dely
      if(nf(i,j).eq.0.or.nf(i+1,j).eq.0) goto 1112
        call eqtn(xxu,yyu,uint,vtest)
      visx= nu*((un(i+1,j)-2.*un(i,j)+un(i-1,j))*rdx2+
     1          (un(i,j+1)-2.*un(i,j)+un(i,j-1))*rdy2
     2   +cyl*((un(i+1,j)-un(i-1,j))/(2.*delx2*float(i-1))
     3   -un(i,j)/(delx2*float(i-1)**2)))
      u(i,j)= uint+delt*((p(i,j)-p(i+1,j))*rdx + gx+visx)
 1112 continue
      if(nf(i,j).eq.0.or.nf(i,j+1).eq.0) goto 1111
        call eqtn(xxv,yyv,utest,vint)
      visy= nu*((vn(i+1,j)-2.*vn(i,j)+vn(i-1,j))*rdx2+
     1          (vn(i,j+1)-2.*vn(i,j)+vn(i,j-1))*rdy2
     2    +cyl*(vn(i+1,j)-vn(i-1,j))/(2.*delx2*(float(i)-1.5)))
      v(i,j)= vint+delt*((p(i,j)-p(i,j+1))*rdy + gy+visy)
 1111 continue
c**********************************************************************
c**********************************************************************
c
c     set boundary conditions

 2000 continue
      do 2200 j=1,jmax
      go to(2020,2040,2060,2080),wl
 2020 u(1,j)=0.0
      v(1,j)=v(2,j)
c
      nf(1,j)=0
c
      go to 2100
 2040 u(1,j)=0.0
      v(1,j)=-v(2,j)
c
      nf(1,j)=0
c
```

```
      go to 2100
 2060 if(iter.gt.0 .and. flg.gt.0.5)go to 2100
      u(1,j)=u(2,j)
      v(1,j)=v(2,j)
c
        nf(1,j)=1
c
      go to 2100
 2080 u(1,j)=u(im2,j)
      v(1,j)=v(im2,j)
c
        nf(1,j)=1
c
 2100 go to (2120,2140,2160,2180),wr
 2120 u(im1,j)=0.0
      v(imax,j)=v(im1,j)
c
      nf(imax,j)=0
c
      go to 2200
 2140 u(im1,j)=0.0
      v(imax,j)=-v(im1,j)
c
      nf(imax,j)=0
c
      go to 2200
 2160 if(iter.gt.0 .and. flg.gt.0.5) go to 2200
      u(im1,j)= u(im2,j)*(float(im2-1)/float(im2)*cyl+(1.0-cyl))
      v(imax,j)=v(im1,j)
c
        nf(imax,j)=1
c
      go to 2200
 2180 u(im1,j)=u(2,j)
      v(im1,j)=v(2,j)
      p(im1,j)=p(2,j)
      v(imax,j)=v(3,j)
c
        nf(imax,j)=1
c
 2200 continue
      do 2500 i=1,imax
      go to (2320,2340,2360,2380),wt
 2320 v(i,jm1)=0.0
      u(i,jmax)=u(i,jm1)
c
      nf(i,jmax)=0
c
      go to 2400
 2340 v(i,jm1)=0.0
      u(i,jmax)=-u(i,jm1)
c
      nf(i,jmax)=0
c
      go to 2400
 2360 if(iter.gt.0 .and. flg.gt.0.5) go to 2400
      v(i,jm1)=v(i,jm2)
      u(i,jmax)=u(i,jm1)
c
        nf(i,jmax)=1
```

```
c
      go to 2400
 2380 v(i,jm1)=v(i,2)
      u(i,jm1)=u(i,2)
      p(i,jm1)=p(i,2)
      u(i,jmax)=u(i,3)
c
        nf(i,jmax)=1
c
 2400 go to (2420,2440,2460,2480),wb
 2420 v(i,1)=0.0
      u(i,1)=u(i,2)
c
      nf(i,1)=0
c
      go to 2500
 2440 v(i,1)=0.0
      u(i,1)=-u(i,2)
c
      nf(i,1)=0
c
      go to 2500
 2460 if(iter.gt.0 .and. flg.gt.0.5) go to 2500
      v(i,1)=v(i,2)
      u(i,1)=u(i,2)
c
        nf(i,1)=1
c
      go to 2500
 2480 v(i,1)=v(i,jm2)
      u(i,1)=u(i,jm2)
c
        nf(i,1)=1
c
 2500 continue

c     special boundary conditions
c
           do 2801 j=12,jm1
c
            nf(17,j)=0.0
c
            u(16,j)=0.0
            u(17,j)=0.0
            v(17,j)=0.0
 2801   continue
            v(17,11)=0.0
            u(21,29)=-1.0
            u(21,30)=-1.0
            u(21,31)=-1.0
           do 2800 i=18,im1
            v(i,jm1)=0.0
 2800   continue
           do 2802 j=2,28
            u(21,j)=0.0
c
            nf(22,j)=0.0
c
 2802   continue
c
```

```
      go to kret(3000,5000)
 3000 continue
c
c     has convergence been reached
c
      if(flg.eq.0.)go to 4000
      iter=iter+1
      if(iter.lt.5000) go to 3050
c     if(iter.lt.500) go to 3050
c     if(cycle.lt.10) go to 4000
      if(cycle.lt.100) go to 4000
      t= 1.e+10
      go to 5000
 3050 flg=0.0
c
c     compute updated cell pressure and velocities
c
      do 3500 j=2,jm1
      do 3500 i=2,im1
c
      if(nf(i,j).eq.0) goto 3500
c
      d=rdx*(u(i,j)-u(i-1,j))+rdy*(v(i,j)-v(i,j-1))+cyl*(u(i,j)
     1+u(i-1,j))/(2.*delx*(float(i)-1.5))
      if(abs(d/dzro).ge.epsi)flg=1.0
      delp= -beta*d
      p(i,j)=p(i,j)+delp
      u(i,j)=u(i,j)+dtdx*delp
      u(i-1,j)=u(i-1,j)-dtdx*delp
      v(i,j)=v(i,j)+dtdy*delp
      v(i,j-1)=v(i,j-1)-dtdy*delp
 3500 continue
      go to 2000
 4000 continue
c
c     print and plot
c
 5000 continue
      if(t.gt.0.)go to 5030
       print 50,(xput(i),i=1,num)
 5030 continue
      if(cycle.le.0) go to 5100
      if(t+1.e-6 .lt. twplt) go to 5600
      twplt=twplt+cwplt*delt
 5100 continue
        print 49, iter,t,cycle
         print 46,t,cycle
c
c   list velocity and pressure fields
c
 5600 continue
      if(cycle.le.0) go to 5800
      if(t+1.e-6.lt.twprt) go to 6000
      twprt=twprt+cwprt*delt
 5800 continue
      do 5900 i= 2,im1
      do 5900 j=2,jm1
      write(9,48) i,j,u(i,j),v(i,j),p(i,j)
 5900 continue
c
```

```
c     set the advance time velocities u and v into the un and vn arrays
c
 6000 continue
      do 6100 i=1,imax
      do 6100 j=1,jmax
      un(i,j)=u(i,j)
      vn(i,j)=v(i,j)
      pn(i,j)=p(i,j)
 6100 continue
c
c     advance time t=t+delt
c
      t=t+delt
      if(t.gt.twfin)  go to 6500
      cycle=cycle+1
      go to 1000
 6500 continue
      call etime(tarray)
      cpu2=tarray(1)+tarray(2)
      cpu2=cpu2-cpu1
      print*, 'total CPU time: ',cpu2
      call exit(1)
      end
c ********************************************************************
c
       subroutine eqtn(x,y,uint,vint)
c
c   solve trajectory equations dx/dt=-u and dy/dt/=-v
c   and get interpolated velocities at the
c   newly computed (x,y) location
c
        real u(36,36), v(36,36)
c
        integer nf(36,36)
c
        common /vel/ u,v
        common /mesh/ dt,dx,rdx,xmin,xmax,dy,rdy,ymin,ymax,dd
        common /dat/ nf, neqn, imax, jmax
c
c  interpolate velocities to (x,y)
c
        call ginter(x,y,uii,vii)
c
c  NEQN - the number of sub-time steps to solve
c  the trajectory equations
c
        ddt=dt/neqn
        n=neqn
         x0=x
         y0=y
        u0=uii
        v0=vii
c
        do 10 k=1,n
          x1=x0-u0*ddt
          y1=y0-v0*ddt
         i1=(x0-xmin)*rdx+2.0
         j1=(y0-ymin)*rdy+2.0
                if(i1.le.0) i1=1
                if(i1.gt.imax) i1=imax
                if(j1.le.0) j1=1
                if(j1.gt.jmax) j1=jmax
          ddt1=ddt
          m=1
          key=0
          if(nf(i1,j1).eq.1) goto 40
c
c  subdivide the time step further if
c  the trajectory hits a blocked cell
c
  30   continue
          x2=x0
          y2=y0
          u2=u0
          v2=v0
          ddt1=ddt1*0.5
          m=2*m
          do 20 l=1,m
          x1=x2-u2*ddt1
          y1=y2-v2*ddt1
         i1=(x2-xmin)*rdx+2.0
         j1=(y2-ymin)*rdy+2.0
                if(i1.le.0) i1=1
                if(i1.gt.imax) i1=imax
                if(j1.le.0) j1=1
                if(j1.gt.jmax) j1=jmax
          if(nf(i1,j1).eq.1) goto 50
          key=key+1
          if(key.gt.20) then
          uint=u2
          vint=v2
          goto 100
          endif
          goto 30
 50     continue
        if(l.eq.m) goto 40
        x2=x1
        y2=y1
        call ginter(x2,y2,u2,v2)
 20     continue
 40     continue
         x0=x1
         y0=y1
c
c  interpolate velocities to (x0,y0)
c
        call ginter(x0,y0,u0,v0)
 10     continue
        uint=u0
        vint=v0
 100    continue
        return
        end
c**************************************************
c
        subroutine ginter(x,y,uint,vint)
c
c  interpolated velocities to the point (x,y)
c
        real u(36,36), v(36,36)
```

```
      integer nf(36,36)
      common /vel/ u,v
      common /mesh/ dt,dx,rdx,xmin,xmax,dy,rdy,ymin,ymax,dd
      common /dat/ nf, ne, imax, jmax
c
c  define the indices of the cells which will be
c  used for interpolation
c
      i1=(x-xmin)*rdx
      if(i1.ge.1) goto 11
         i1=1
 11   continue
      if(i1.le.imax) goto 12
         i1=imax
 12   continue
      x1=x-(i1-1.0)*dx
      if(i2.ge.1) goto 13
         i2=1
 13   continue
      if(i2.le.imax) goto 14
         i2=imax
 14   continue
      x2=(i2-1.0)*dx-x
      j1=(y-ymin)*rdy
      if(j1.ge.1) goto 15
         j1=1
 15   continue
      if(j1.le.jmax) goto 16
         j1=jmax
 16   continue
      j2=j1+1
      y1=y-(j1-1.0)*dy
      if(j2.ge.1) goto 17
         j2=1
 17   continue
      if(j2.le.jmax) goto 18
         j2=jmax
 18   continue
      y2=(j2-1.0)*dy-y
c
      i0=2*x1*rdx
      j0=2*y1*rdy
      i3=i1+2*i0
      j3=j1+2*j0
c
c  linearly interpolate velocities
c
      c=abs(-y1+0.5*dy)
      d=abs(-y1+dy*(2*j0-0.5))
      uint=(d*(u(i1,j2)*x2+u(i2,j2)*x1)+
     *  c*(u(i1,j3)*x2+u(i2,j3)*x1))*dd
      c=abs(-x1+dx*0.5)
      d=abs(-x1+dx*(2*i0-0.5))
      vint=(d*(v(i2,j1)*y2+v(i2,j2)*y1)+
     *  c*(v(i3,j1)*y2+v(i3,j2)*y1))*dd
      return
      end
```