Available online at www.sciencedirect.com
SCIENCE DIRECT®
Journal of Computational Physics 200 (2004) 399–411
JOURNAL OF COMPUTATIONAL PHYSICS
www.elsevier.com/locate/jcp

A Monte Carlo scheme for tracking filling fronts

Vaughan R. Voller *

Civil Engineering, Saint Anthony Falls Laboratory, National Center for Earth Surface Dynamics, University of Minnesota, Mississippi River at 3rd Avenue St, Minneapolis, MN 55455, USA

Received 2 September 2003; received in revised form 25 March 2004; accepted 13 April 2004
Available online 10 June 2004

Abstract

A Monte Carlo (MC) scheme for tracking filling fronts is developed. The test problem of tracking fluid injected into a thin mold is considered. The MC scheme iteratively redistributes the volume entering the mold among the cells of a spatial discretization. The transition probabilities used in the MC scheme, which determine how the fluid volume is redistributed, are derived from a discrete representation of the governing steady-state pressure equation. Analysis shows that the MC steps are equivalent to an iterative solution of the discrete equations. Further, it is shown that the MC scheme can be reconfigured into the form of a standard Lattice Boltzmann Method (LBM). Results show that the proposed MC scheme is accurate, does not require an explicit field calculation of the fluid pressure field, and, when compared with existing numerical filling algorithms, exhibits computation times over 1000 times faster.

1. Introduction

The need to understand systems that involve multiple scales and deforming domains tax existing computer resources and methods to the limit [1]. This situation has led to the development of new computational approaches that replace traditional discretizations of continuum descriptions (e.g., FEM) with alternative representations. So called "lattice models" are an important class of alternative methods. Broadly speaking, these models develop statistical descriptions of macroscopic processes by the use of simple relationships that propagate "particles" through a grid (lattice). Examples include Lattice Boltzmann Method (LBM) fluid flow models [2–8] and Monte Carlo (MC) heat transfer calculations [9–11]. In the LBM, the evolution of a particle velocity distribution function is determined by solving a discrete Boltzmann equation [2–5]. Appropriate macroscopic properties, that satisfy the Navier–Stokes equations, can then be derived on moment integration of the distribution function [5]. In MC heat transfer calculations

* Tel.: +1-612-625-0764; fax: +1-612-6267-7750.
E-mail address: volle001@umn.edu (V.R. Voller).

0021-9991/$ - see front matter © 2004 Elsevier Inc. All rights reserved.
doi:10.1016/j.jcp.2004.04.009

[9–11], the probabilistic movement of the particles over the lattice is constructed to form a solution sequence for a discrete realization of the continuum heat transfer equation.

The objective of this paper is to develop a lattice method that can be applied in the solution of moving boundary problems, i.e., problems where the transient position of one or more of the domain boundaries is a priori unknown. Perhaps the best-known example of a moving boundary problem is the Stefan melting problem and both LBM [6–8] and MC [9] solutions for tracking the solid–liquid front have been presented in the literature. In this paper, a lattice scheme for tracking the movement of the fluid–air (filling) front during the potential filling of a thin mold [12–14], see Fig. 1, is developed. Under the condition that the specific heat $c \to 0$, this problem is seen as a special case of the one phase Stefan problem. The lattice scheme is developed with the concepts and approaches used in Monte Carlo heat transfer simulations [9–11] and will be referred to as the MC scheme. It is shown, however, that the discrete operations in the MC scheme can be reconfigured in to a form that matches LBM calculations.

The immediate advantage in using the proposed MC scheme is that it does not require an explicit field calculation of the fluid pressure. It is shown, by performance comparison with a well known and used numerical filling algorithm [12], that this feature of the MC scheme dramatically improves efficiency (over 1000 times in some cases) without compromising the accuracy of the front tracking.

2. Governing equations

Consider an initially empty, ventilated, horizontal mold cavity. An incompressible fluid is introduced into the mold at injection points along the mold edge, with the flow rate at a given injection point κ denoted as $Q_\kappa(t)$, see Fig. 1. As time progresses, assuming a no flow condition along the mold edge and conditions that will not lead to fingering, a moving sharp front will form between the entering fluid and escaping air. If the mold is thin a potential flow model is valid, i.e., at a point in the mold,

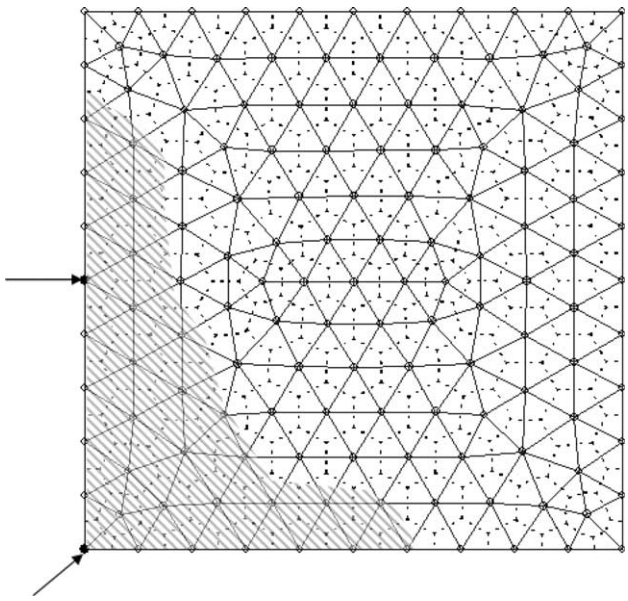

Fig. 1. Plan view schematic of the filling of a thin mold, superimposed with an unstructured mesh.

$$q = -K\nabla p, \tag{1}$$

where q is a volume flux, p is the gage pressure and $K = K(x, y)$ is a fluidity. A statement of volume balance can be obtained by considering an arbitrary control area, A, in the x–y plane, with closed boundary S, i.e.,

$$\frac{\mathrm{d}}{\mathrm{d}t}\int \phi \mathrm{d}A = \oint K\phi\nabla p \cdot \mathbf{n}\,\mathrm{d}S + Q_\kappa, \tag{2}$$

where $\mathbf{n}$ is the outward pointing normal on S, Q_κ will only be non-zero if the point κ is located on the surface S, and ϕ is a microscopic phase marker that takes the value 1 in filled portions of the mold and the value 0 in empty potions. On noting that both $p = 0$ and $\nabla p = 0$ at points where $\phi = 0$, (2) can be rewritten as

$$\frac{\mathrm{d}}{\mathrm{d}t}\int \phi \mathrm{d}A = \oint K\nabla p \cdot \mathbf{n}\,\mathrm{d}S + Q_\kappa. \tag{3}$$

Eq. (3) is essentially an integral form of the well known Volume of Fluid (VOF) equation which has been used as the basis of many successful front tracking codes [15]. In point form (3) can be written as

$$\nabla \cdot (K\nabla p) + Q_\kappa \delta(\mathbf{x} - \mathbf{x}_\kappa) = 0 \tag{4}$$

in the filled domain, with conditions

$$p = 0 \text{ and } K\nabla p \cdot \mathbf{n} = -v_\mathrm{n} \tag{5}$$

on the filling front, where v_n is the front speed in the normal direction.

With the correct definition of K, the system described above can be used to model a variety of filling situations, e.g., (i) the filling of a thin polymer mold [14], (ii) the filling of a porous media [16], and (iii) the filling of an ocean basin with sediment [17]. In all these examples, the problem is to track the interface between the filled and empty regions.

Fig. 1 provides a general numerical grid that can be used to develop a discrete set of equations based on the balance in (3). The grid consists of nodes which are numbered $i = 1, 2, \ldots, n$ and surrounding each node (see dashed lines) a control area Δ_i. Associating each Δ_i with the integration domain in (3) and using suitable approximations for the fluidity and pressure gradients (e.g., standard finite element interpolations) leads to the time implicit system of discrete equations

$$\Delta t\mathbf{A}\mathbf{p} = \mathbf{f} - \mathbf{f}^{\mathrm{old}} - \mathbf{b}. \tag{6}$$

In (6), $\mathbf{p}$ is the vector of nodal pressures, the components $0 \leqslant f_i = \int_{\Delta_i} \phi\,\mathrm{d}A \leqslant \Delta_i$ track the progress of the filling, $b_i = \int_0^{\Delta t} Q_\kappa \delta_{i\kappa}\,\mathrm{d}\alpha$, Δt is the discrete time step, and the superscript "old" refers to value at the previous time step.

An important attribute of the discrete governing Eq. (6), fully discussed by Voller and Chen [13], is that predictions of the fill factor field, $\mathbf{f}$, are independent of the time step used. Briefly, consider the filling of an initially empty mold out to time $t = \sum_{k=1}^{n} \Delta t^k$. By summing (6) over time it is seen that the individual time step solutions of the pressure field, $\mathbf{p}^k$ and the nth time step solution of the fill factor field $\mathbf{f}^n$ satisfy

$$t\mathbf{A}\mathbf{P} = \mathbf{f}^n - \mathbf{B}, \tag{7}$$

where $\mathbf{P} = \frac{1}{t}\sum \Delta t^k \mathbf{p}^k$ and the components $B_i = \int_0^t Q_\kappa \delta_{i\kappa}\,\mathrm{d}\alpha$; an equation identical in form to (6). Hence the nodal fill factor field resulting from a single application of (6) will be identical to the field obtained when multiple time steps (that sum to the given single time step) are used.

3. A numerical solution

A solution of the system (6) requires that

$$p_i = 0 \text{ at any node}, i, \text{where } f_i < \Delta_i, \tag{8}$$

hence the system is non-linear. This feature poses a challenge in seeking an appropriate solution. A standard, experimentally validated, and well-used solution scheme is due to Bruschke and Advani [12]. In a given time step ($t - t + \Delta t^k$) this solution can be implemented in two steps. In step 1, the system

$$\mathbf{A}^* \mathbf{p} = -\mathbf{b} \tag{9}$$

is solved. In the coefficient matrix $\mathbf{A}^*$, diagonal elements a_{ii} are set to a large number ($\sim 10^{20}$) if the current value of $f_i^{\text{old}} < \Delta_i$, the remaining coefficients remain the same as those in the $\mathbf{A}$ of (6). In this way, provided that the current time step is small enough that no nodal volume completes filling, a solution of (9) will produce a pressure field consistent with the non-linear condition in (8). The second step of the solution accounts for the time step constraint by choosing the time step Δt^k such that, for the set of nodes where $f_i^{\text{old}} < \Delta_i$, the calculation

$$\mathbf{f} = \mathbf{f}^{\text{old}} + \Delta t^k \mathbf{A}\mathbf{p} + \mathbf{b}, \tag{10}$$

yields at least one node where $f_i^{\text{old}} < \Delta_i$ and no nodes where $f_i^{\text{old}} > \Delta_i$. In practice

1. To avoid the use of excessively small time steps the search for Δt^k is made over the nodes where

$$f_i^{\text{old}} < f_{\text{fill}} = 0.999\ \Delta_i. \tag{11}$$

2. A Gauss–Seidel iteration scheme is used to solve (10). Convergence of this scheme is declared when successive iterations (n and $n+1$) satisfy

$$\max \left[1 - \frac{p_i^{n+1}}{p_i^n} \right] < \text{tol}. \tag{12}$$

There are alternative schemes for (6), which may be more efficient, e.g., the scheme of Voller and Peng [14] that exploits the time step independence in the equations. The Bruschke and Advani [12] scheme outlined above, however, is robust, widely used, and accurate. Hence, it makes a suitable benchmark to test the accuracy and efficiency of the proposed MC filling scheme.

4. A Monte Carlo scheme

4.1. Basic concepts in a Monte Carlo scheme

A basic MC scheme involves repeated random walks of a particle over the nodes points of a spatial discretization, e.g., the grid and nodes in Fig. 1. In a step, of a given random walk, the probability that a particle, at node i, moves to the neighboring node j is given by

$$\tau_{ij} = \frac{a_{ij}}{a_{ii}}, \tag{13}$$

where the terms on the RHS are coefficients in a discretization of the governing equation of interest, e.g., the elements of the matrix $\mathbf{A}$ in (6) and (7). After repeated random walks, statistics of the combined particle paths can be used to construct an approximate solution of the discrete governing equations [9–11].

The operation of the random walk process can be improved by using the so-called "exodus" method [10,11]. In this approach, the single particle with multiple walks is supplanted by tracking the movement of a set of N particles. At each step the particles residing at a given node point are separated and redistributed among the immediate neighboring nodes according to the ratios given by τ_{ij}. A variation in the Exodus method is to replace the discrete set of N particles with a continuous volume V and then, in each MC step, split and move this volume through the grid via the probabilities τ_{ij}. In this way the field of grid nodal volume fractions at step $k+1$, $\mathbf{v}^{k+1}$, can written in terms of the nodal volume fractions at the previous step

$$\mathbf{v}^{k+1} = \mathbf{T}\mathbf{v}^{k}, \tag{14}$$

where $\mathbf{T}$ is a transition matrix with elements τ_{ij} and zero elements along the diagonal, i.e., $\mathrm{diag}\mathbf{T} = 0$. Note that the replacement of the discrete particles with a continuous volume, analogous to the replacement of the Boolean particle occupation variable in a Lattice Gas Method (LGM) with the continuous particle distribution in a LBM [2,4], eliminates statistical noise in the proposed MC scheme.

4.2. Development of an MC filling scheme

In overview, the basic operation of the proposed MC filling scheme is as follows:

- The scheme is designed to locate the filling front at time t.
- The scheme is initiated by placing volumes $V_\kappa = \int_0^t Q_\kappa \, \mathrm{d}\alpha$ at the nodes associated with injection points. This provides an initial nodal field of volume fractions, $\mathbf{v}^0 = \mathbf{B}$, see (7).
- This volume is moved around the grid according to transition probabilities τ_{ij}, derived, according to (13) from the coefficients in the matrix $\mathbf{A}$ of (6) and (7).
- When a volume fraction arrives at a node that is associated with an under-filled cell, i.e., $f_i < \Delta_i$ a portion of that volume is removed and used to update the fill fraction and thereby continue or complete the filling in that cell.
- The MC steps continue until all the initial volume has been used up in filling the cells of the grid. At this point the filled and partially cells will define the time t position of the filling front.

To determine the validity of the proposed MC scheme it is necessary to show that the above operation approaches a solution of the discrete filling Eq. (7). The sequence of operations in the proposed MC scheme can be formally written as

$$\mathbf{v} = \mathbf{v}^0 + \mathbf{v}^1 + \mathbf{v}^2 + \mathbf{v}^3 + \cdots, \tag{15}$$

where the nodal volumes at step $k+1$ are related to the previous step through

$$\mathbf{v}^{k+1} = \mathbf{T}\mathbf{v}^{k} - \mathbf{e}^{k}, \tag{16}$$

the filled fraction at step (k+1) is

$$\mathbf{f}^{k+1} = \sum_{m=1}^{k} \mathbf{e}^{m}, \tag{17}$$

and the ith element in $\mathbf{e}^k$ is

$$[\mathbf{e}^{k}]_i = \min((\Delta_i - [\mathbf{f}^{k}]_i), [\mathbf{T}\mathbf{v}^{k}]_i), \tag{18}$$

where $[\mathbf{T}\mathbf{v}^k]_i$ is the ith element of the matrix vector product $\mathbf{T}\mathbf{v}^k$. Note use of (18) allows for a smooth transition as a cell completes filling. From (15) and (16), it can be shown that

$$(\mathbf{I} - \mathbf{T})\mathbf{v} = \mathbf{B} - \sum_{m=1}^{k} \mathbf{e}^{m}. \tag{19}$$

If enough steps of the MC are taken (19) approaches

$$(\mathbf{I}-\mathbf{T})\mathbf{v}=\mathbf{B}-\mathbf{f}, \tag{20}$$

and with the construction $\mathbf{v}=t(-\mathrm{diag}\mathbf{A})\mathbf{P}$ (20) becomes

$$t\mathbf{A}\mathbf{P}=\mathbf{f}-\mathbf{B}. \tag{21}$$

Eq. (21) is identical in form to discrete filling equations in (7), further, by (18), the elements of **P** and **f** satisfy the non-linear condition in (8). Hence, by the existence and uniqueness proof presented by Voller and Chen [13], the nodal fill factor field obtained from the proposed MC scheme (17) will match that obtained from a numerical solution of the discrete filling equations in (7).

4.3. The coding of the MC filling scheme

The coding of the MC scheme in (15)–(18) is established as follows:

- The cell variable in the MC is the volume of fluid in a cell and is denoted by F_i.
- The MC is initiated on setting.

$$F_i=\int_0^t Q_\kappa\delta_{i\kappa}\,\mathrm{d}\alpha. \tag{22}$$

- This volume is then iteratively redistributed according to the following rule:

$$\begin{aligned}&\text{if } F_i>\Delta_i \quad (\text{i.e., cell } i \text{ is over filled})\\ &\text{set}\\ &F_j=F_j+\tau_{ij}(F_i-\Delta_i) \quad \forall j\in \text{neigborhood of } i\\ &\text{then reset}\\ &F_i=\Delta_i.\end{aligned} \tag{23}$$

- The progress of the filling of the domain can be tracked by calculating the filled fraction field

$$f_i=\min(\Delta_i,F_i), \tag{24}$$

at the end of each iterative sweep.

- Convergence of the MC iterations can be declared when

$$\left(1-\frac{\sum_n f_i}{\sum_n F_i}\right)<\mathrm{TOL}, \tag{25}$$

where the value TOL is a specified volume error.

- On convergence the position of the fill front can be obtained on defining the nodal field $g_i=f_i/\Delta_i$ and interpolating for the iso-line $g=0.5$.
- The iterations in (23) are similar, in many respects, to a Gauss–Seidel iteration and as such the convergence can be significantly improved by an over relaxation. The MC iterations in (23) are written as

$$\begin{aligned}&\text{if } F_i>\Delta_i\\ &\text{set}\\ &F_j=F_j+\omega\alpha_{ij}(F_i-\Delta_i) \quad \forall j\in \text{neigborhood of } i\\ &\text{then reset}\\ &F_i=F_i-\omega(F_i-\Delta_i),\end{aligned} \tag{26}$$

where $1 < \omega < 2$ is a relaxation factor. Through testing it is found that a good setting for the relaxation is

$$\omega = 1 + 0.99\frac{\sum_n f_i}{\sum_n F_i}, \tag{27}$$

a choice that keeps the relaxation small at early iterations. Use of (26) and (27) will produce rapid and accurate estimates of the fill front location provided that the value of TOL $\leqslant 0.005$ (i.e., the volume balance is satisfied to within 0.5%). If a larger value of TOL is used then it is possible that (26) can satisfy the convergence in (25) while field values of F_i in filled cells are less than Δ_i. While this has little effect on the accuracy of the fill front position prediction, the post process interpolation of the g field can provide erroneous fronts behind the real filling front. Hence, to avoid any problems, in this work the MC specified in (26) and (27) will always be used with a TOL $\leqslant 0.005$.

The key feature in the above coding of the MC filling scheme, which distinguishes it from numerical solution approaches based explicitly on (6), is that there is no pressure field calculation.

4.4. Iterations cast in form of a lattice Boltzmann method

Before the proposed MC scheme is tested it is noted that the iterations in (23)–(25) can be written in the form of a pseudo transient calculation

$$v_k(\mathbf{x} + c_k\Delta t, t + \Delta t) = v_k^0(\mathbf{x}, t), \tag{28}$$

where $v_k(\mathbf{x} + c_k\Delta t, t + \Delta t)$ is interpreted as the volume arriving at the node located at position $\mathbf{x} + c_k\Delta t$ that originated from the node at position $\mathbf{x}$. The nodes are neighboring nodes connected by a single lattice link and the volume of fluid velocity along the link is $c_k = \Delta\mathbf{x}/\Delta t$. After calculation of (28), for all directions, the current volume of fluid at the node at $\mathbf{x}$ follows from:

$$F(\mathbf{x}, t + \Delta t) = f(\mathbf{x}, t) + \sum_{k=1}^{nb} v_k(\mathbf{x}, t + \Delta t), \tag{29}$$

where the sum is over the links to the neighboring nodes. The updated fill fraction at a node is then calculated as

$$f(\mathbf{x}, t + \Delta t) = \min[\Delta(\mathbf{x}), F(\mathbf{x}, t + \Delta t)] \tag{30}$$

and for use at the next pseudo time step the update

$$v_k^0(\mathbf{x}, t + \Delta t) = w_k(\max[0, F(\mathbf{x}, t + \Delta t) - 1]) \tag{31}$$

is made, where if the node at $\mathbf{x}$ is node i and the node at the end of the kth lattice link is j, the weight $w_k \equiv \tau_{ij}$, note, in this way $\sum_{k=1}^{nb} w_k = 1$. The steps, defined by the pseudo time step iteration (28)–(31) continue until the convergence criteria in (25) is satisfied.

On first sight, the solution approach given above may seem cumbersome and the introduction of an alternative notation scheme to define the volume movement on the lattice, awkward. This reconfigured scheme however, has two important attributes:

1. The scheme is suitable for implementation in a parallel architecture.
2. Under the condition of a unit collision frequency, the update in (28) matches the form of the LBM equation using the BGK approximation [4,8].

Hence, although no formal derivation has been made, it is reasonable to conclude that the operations of the proposed MC algorithm will be equivalent to an LBM calculation. One might expect, however, a slower CPU performance due to the lack of over-relaxation and the Jacobi like nature of the iterative scheme.

5. Results

5.1. Test problem

The domain of the test problem, shown in Fig. 2, is a square of area 10000 units^2. Fluid is injected into the domain at the points (0,0) and (0,50). Injection at the point (0,0) is at the constant rate of $Q_1 = 2$ units^2/time. At point (0,50), however, the flow rate increases with time according to $Q_2 = e^{t/100}$ units^2/time. In order to establish an "interesting" fill pattern in the test problem, unless otherwise stated, the nodal fluidities are set as

$$K(x,y) = \max[(x/y),(y/x)]. \tag{32}$$

This choice can be used to simulate "race-tracking" a common feature of RTM molding [13]. This test problem captures all of the significant transient and spatial features of a mold filling problem.

To investigate if the design of the cell will influence the MC performance two alternate cell designs are used, Fig. 2:

1. A Control Volume Finite Difference (CVFD) grid consisting of a 10,000 square control cells each with an area $\Delta = 1$ unit^2. On this grid the coefficients in **A** and subsequently the MC probabilities τ_{ij} are obtained from a standard finite difference discretization of $\nabla \cdot (K \nabla p) = 0$.

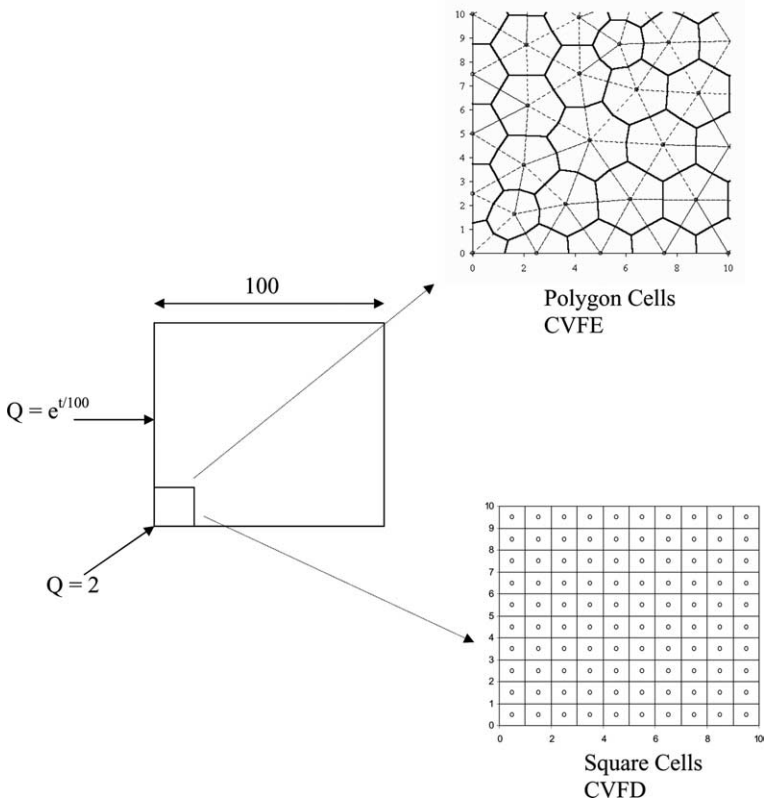

Fig. 2. Plan view of test geometry indicating possible MC cell designs.

2. An unstructured Control Volume Finite Element (CVFE) grid. This grid, generated with the public domain program EasyMesh [18], contains 1932 polygon control cells. The coefficients are obtained from a control volume finite element discretization [19] of $\nabla \cdot (K \nabla p) = 0$.

Note that the grid size in the CVFE grid is about 2.5 times larger than the grid size in the CVFD grid. Hence, in addition to the effect of the grid design, the investigation of the MC performance on the two grids will also provide information on grid size dependence.

5.2. Verification

An initial verification of the MC approach involves the problem where injection is only from the point (0,0) and the fluidity K is set to a constant value. This problem is designed to test the basic performance of the MC. In this problem, at early times, the analytical fill front position will be a 1/4 circle with an area $= t\pi/2$. In particular, at time $t = 3927$ the 1/4 circle will span the corner points (0,100) and (100,0). The MC predicted fill front at time $t = 3927$, see Fig. 3, matches the analytical position. MC predictions for both the CVFD and CVFE cells are shown in Fig. 3. The relatively smooth 1/4 circle front confirms the basic soundness of the proposed MC and also indicates independence from cell design and grid size.

5.3. Performance

Fig. 4 shows the MC (CVFD grid) predicted movement of the fill front when both injection points are operating and the non-constant fluidity in (20) is assumed. Although the MC solution uses the numerical coefficients in its iterations, its obvious advantage over a numerical scheme is that it does not require an explicit calculation of the pressure field. This advantage equates to a significant decrease in computation time. The CPU time, on a 800 MHz Intel Pentium 3 processor, required for the predictions in Fig. 4 are

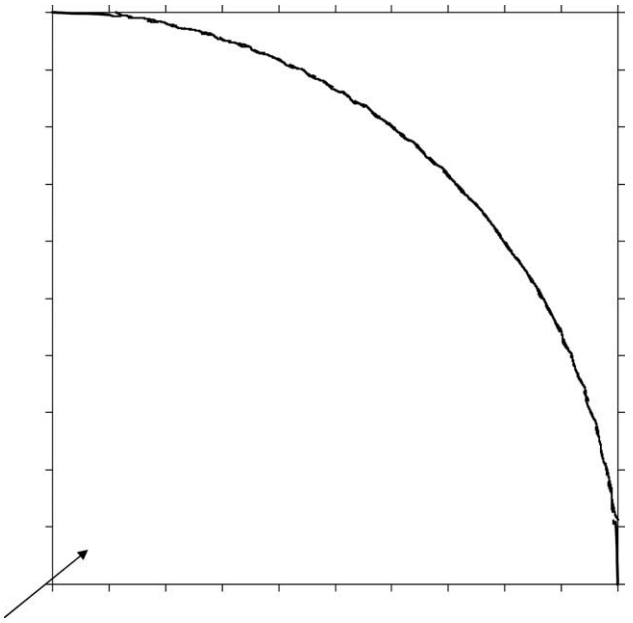

Fig. 3. Filling front position at time $t = 3927$ for constant corner injection and fluidity.

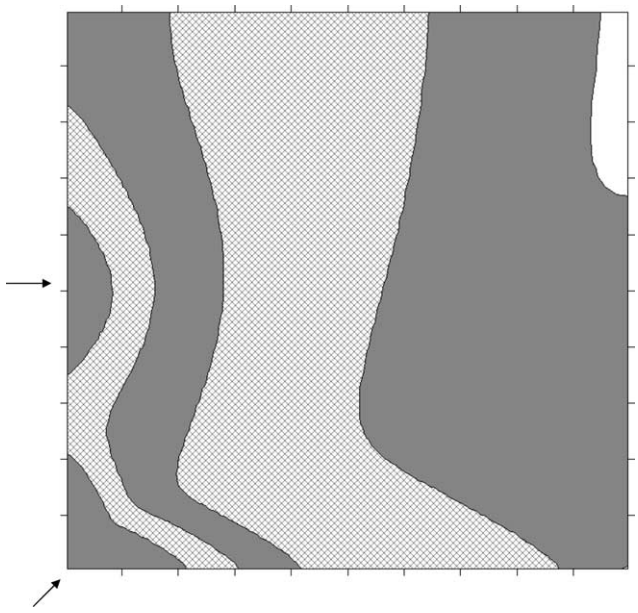

Fig. 4. MC scheme filling front predictions at time $t = 100$, 200, 300, 400 and 450.

Table 1
CPU times (seconds)

Time	Fraction filled	MC	Numerical	MC speed up	Mass error (%)
100	0.037	0.28	367	1311	0.19
200	0.104	0.61	1776	2911	0.325
300	0.251	2.58	4822	1869	0.48
400	0.616	18.34[a]	7857	428	1.5
450	0.98	83.21[a]	–	–	–

[a] Calculated with a mass error setting of 0.5%.

compared with the CPU requirement of the benchmark numerical solution [12] in Table 1. In order to ensure a fair comparison the numerical solution is made with the numerical convergence parameter $\mathrm{tol} = 10^{-5}$ in (12). The resulting volume error is then calculated and if less than 0.5% applied as the MC convergence criterion in (25). If the numerical volume error exceeded 0.5%, however, the more stringent criterion of 0.5% is applied in the MC calculations. The speed-up of the MC scheme, shown in Table 1, is quite extraordinary. At early time steps the speed-up is well over 1000 times. At later times the efficiency decreases but the speed-up over the numerical scheme is still more than two orders of magnitude. Fig. 5 compares the MC and numerical predictions for the fill front position at times $t = 200$ and 400. This result confirms that the speed-up in the MC solution does not come at the expense of accuracy. The small discrepancy in the fronts at $t = 400$ is due to the stricter volume balance tolerance (0.5%) used in the MC solution. Fig. 6 compares the MC front predictions at time $t = 200$ and 400 using both the square and polygon cells. This result confirms that the MC can operate successfully on an unstructured mesh of non-square cells. The result also indicates that the presented results, Figs. 4–6, are independent of grid size.

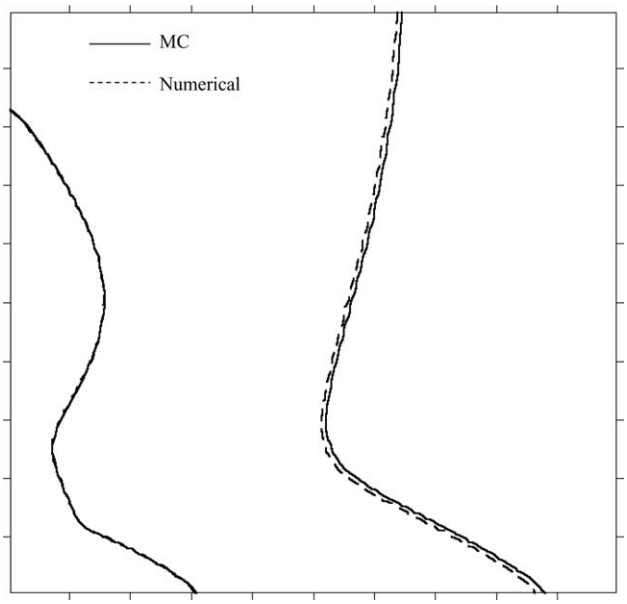

Fig. 5. Comparison of MC and Numerical filling front predictions at time $t = 200$ and 400.

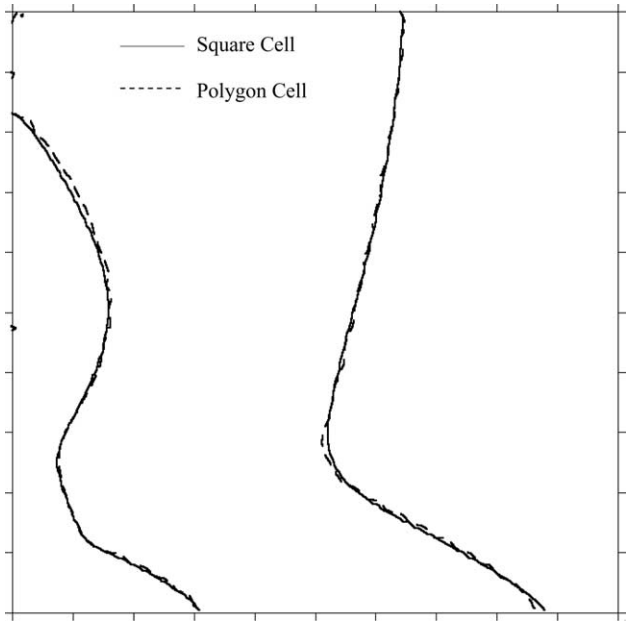

Fig. 6. MC performance with square and polygon cells: front predictions at time $t = 200$ and 400.

Front predictions obtained with the LBM realization are in agreement with those obtained with the proposed MC scheme. As expected, however, see Table 2, the CPU performance of the LBM, while still up to an order of magnitude faster than the standard scheme [12], is slower than the MC scheme. Further work

Table 2
Performance of LBM realization

Time	LBM(CPU)/MC(CPU)
100	76
200	87
300	100
400	101

Calculated with a mass error setting of 0.5%.

is needed to investigate if implementation on a parallel architecture and the use of over-relaxation will improve the LBM performance.

6. Conclusions

In this paper a Monte Carlo scheme for tracking the fluid–air front in the filling of a thin mold cavity has been presented. The MC scheme iteratively redistributes the volume entering the mold. The probabilities in the MC, that determine how the volume is redistributed, are based on an appropriate numerical discretization of $\nabla \cdot (K\nabla p) = 0$. Analysis shows that, with the appropriate constructions, the nodal fields generated by the MC scheme represent an iterative solution of the discrete filling equations in (7). Further, it is shown that the steps in the MC scheme can be reconfigured into the form of a standard Lattice Boltzmann Method (LBM). The performance of the MC scheme has been checked on a test problem that includes all of the significant features found in a practical case. The MC predictions of front position are in close agreement with an alternative, experimentally validated, numerical solution [12]. The advantage of the MC solution, however, is that it does not require an explicit solution of the pressure field. This feature leads to a significant saving in computational effort. When compared with standard numerical approaches the proposed MC scheme can exhibit speedups of over 1000 times without loss of accuracy.

The work here represents an initial attempt at developing MC solutions for moving boundary problem. Further work will be directed at extending the proposed MC scheme (perhaps by formalizing and exploiting its connection to LBM calculations) to deal with more general situations, including, problems with constant pressure gates, Stefan problems, problems with surface tension, problems in geo-morphology, and three-dimensional problems.

Acknowledgements

This work was supported by the STC program of the National Science Foundation via the National Center for Earth-surface Dynamics under the agreement Number EAR- 0120914. The author would like to acknowledge extensive discussions with his colleagues in the NCED working group on Integration Across Scales.

References

[1] V.R. Voller, F. Porte-Agel, Moore's law and numerical modeling, J. Comp. Phys. 179 (2002) 698.
[2] G.R. McNamara, G. Zanetti, Use of the Boltzmann equation to simulate lattice-gas automata, Phys. Rev. Lett. 61 (1988) 2332.
[3] F. Higuera, S. Succi, R. Benzi, Lattice gas dynamics with enhanced collisions, Europhys. Lett. 9 (1989) 345.
[4] S. Chen, G.D. Doolen, Lattice Boltzmann method for fluid flows, Annu. Rev. Fluid Mech. 30 (1998) 329.

[5] S. Chen, D. Martinez, On boundary conditions in lattice Boltzmann methods, Phys. Fluids 8 (1996) 2527.
[6] W. Miller, S. Succi, D. Mansutti, Lattice Boltzmann model for anisotropic liquid–solid phase transition, Phys. Rev. Lett. 86 (2001) 3578.
[7] G. De Fabritiis, A. Mancini, D. Mansutti, S. Succi, Mesoscopic models of liquid/solid phase transitions, Int. J. Mod. Phys. C 9 (1998) 1405.
[8] W.-S. Jiaung, J.-R. Ho, C.-P. Kuo, Lattice Boltzmann method for the heat conduction problem with phase change, Num. Heat Trans. 39 (2001) 167.
[9] A. Haji-Sheikh, E.M. Sparrow, The solution of heat conduction problems by probability methods, J. Heat Trans. 89 (1967) 121.
[10] M.N.O. Sadiku, O. Ajose, Z. Fu, Applying the Exodus method to solve Poisson's equation, IEE Trans. Microw. Theory Tech. 42 (1994) 661.
[11] M. Grigoriu, A Monte Carlo solution of heat conduction and Poisson equations, J. Heat Trans. 122 (2000) 40.
[12] M.V. Bruschke, S.G. Advani, A finite element/control volume approach to model filling in anisotropic porous media, Polym. Compos. 11 (1990) 398.
[13] V.R. Voller, Y.F. Chen, Prediction of filling times of porous cavities, Int. J. Num. Meth. Fluid Flow 23 (1996) 661.
[14] V.R. Voller, S. Peng, An algorithm for analysis of polymer filling of molds, Polym. Eng. Sci. 35 (1995) 1758.
[15] C.W. Hirt, B.D. Nicholls, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201.
[16] V.R. Voller, S. Peng, Y.-F. Chen, Numerical solutions of transient free surface problems in porous media, Int. J. Num. Meth. Eng. 39 (1996) 2889.
[17] J. Swenson, V.R. Voller, C. Paola, G. Parker, J.G. Marr, Fluvio-deltatic sedimentation: a generalized Stefan problem, Eur. J. Appl. Math. 11 (2000) 433.
[18] EasyMesh http://www.dinma.univ.triests.it/~nirftc/research/easymesh Last access June 2003.
[19] S.P. Neuman, P.A. Witherspoon, Analysis of non steady flow with a free surface using the finite-element method, Water Resour. Res. 7 (1971) 611.