Simulation of Polymer Foaming using the Lattice Boltzmann Method

M. Ataei\textsuperscript{a}, V. Shaayegan\textsuperscript{a}, F. Costa\textsuperscript{b}, S. Han\textsuperscript{c}, C. B. Park\textsuperscript{a}, M. Bussmann\textsuperscript{a,∗}

\textsuperscript{a}Department of Mechanical Engineering, University of Toronto, 5 King’s College Rd, Toronto, ON M5S 3G8, Canada
\textsuperscript{b}Autodesk, Inc., 259-261 Colchester Rd., Kilsyth, VIC. 3137, Australia
\textsuperscript{c}Autodesk, Inc. 2353 North Triphammer Rd., Ithaca, NY 14850, USA

Abstract

This paper presents a 3D lattice Boltzmann model (LBM) for simulation of polymer foaming processes. The model incorporates many of the essential aspects of foaming phenomena: gas diffusion into nucleated bubbles, bubble dynamics and coalescence, surface tension, the stabilizing disjoining pressure between bubbles, and the non-Newtonian behavior of polymers. The model uses a free surface approach to reduce the two-phase (liquid-gas) problem to a single phase liquid problem. The model is validated against the results of the Cell Model of Arefmanesh and Advani, and then results are presented that demonstrate the potential of the LBM model to predict various polymer foaming phenomena.

Keywords: Polymer foaming, Lattice Boltzmann, bubble growth, volume of fluid, numerical algorithms, free surface flow

1. Introduction

Polymer foams play an important role in our everyday life. They are incorporated into a wide variety of products including automobiles, furniture, packaging and insulation. Polymer foams are created by nucleation and growth of gas bubbles in a polymer liquid. In many foam manufacturing processes, a metered amount of gas (e.g. CO\textsubscript{2}) is dissolved into a polymer liquid at high pressure and temperature. A rapid solubility drop (e.g. induced by a pressure drop) gives rise to nucleated gas bubbles, which grow due to the diffusion of gas molecules from the polymer. Ultimately, the bubbles become separated by thin liquid regions (lamellae), forming a cellular structure. Predicting this foaming process is essential for developing desired foam structures with desired properties as, for example, microcellular homogeneous foams tend to improve the mechanical strength per weight of a final structure.

∗Corresponding author
Developing a numerical model for foam evolution is challenging due to its complexity. The numerical model requires coupling of the gas advection-diffusion process with the fluid flow model. This coupling poses a numerical challenge, because the bubble interfaces are arbitrarily-shaped immersed boundaries for the advection-diffusion equation, at which gas concentration must be specified. The interaction between bubble interfaces must also be carefully incorporated, since the final foam structure depends sensitively upon the structure of the lamellae. Simple fluctuations, such as the bursting of a lamella, can significantly change the final cellular structure.

There have been several attempts in the literature to model the foaming process by introducing simplifications. Single bubble models (i.e., “Cell Model”) approximate bubble growth as a 1D problem by neglecting bubble interactions, and approximate a foam as a single isolated bubble in an infinite pool of liquid, or as a periodic array of spherical bubbles (e.g., [2, 4–6, 21, 27]). A comparison of these models can be found in [12]. In a number of 2D models, the transport equations are solved for just a section of a lamella separating neighboring bubbles, assuming that the bubbles are arranged in, for example, hexagonal arrays [13–15]. The arbitrary Lagrangian-Eulerian sharp interface algorithm has also been used to simulate bubble growth in 2D [35].

There is also another class of foaming simulations over much larger length scales, where the interest is not individual bubble growth and interactions, but the goal is to model foam flow into a given mold geometry, i.e. mold-filling. These models omit details of foaming at the bubble scale, but use bubble scale models to extract certain constants and information to incorporate in particle tracers, surrogate models, or population balance equations [16, 21], to approximate the effects of bubble growth on fluid properties such as fluid density and viscosity.

The lattice Boltzmann method (LBM) can be a powerful tool for simulating the growth and interaction of bubbles in a foaming process [11, 22, 23]. Korner et al. [22, 23] used a 2D free surface lattice Boltzmann simulation to study the structure of aluminum foams. They simulated the expansion of gas bubbles in molten aluminum, and studied the influence of viscosity, surface tension, and mold constraints on the final structure of the foam. In contrast to the models mentioned previously, LBM allows for easy implementation of a diffusion boundary condition at bubble-liquid interfaces, which is necessary for calculation of bubble growth rate. The model developed by Korner et al., while powerful, is missing some important physical aspects of the polymer foaming process, such as the non-Newtonian behavior of polymers, the effect of solid skin layers, and the dissolution of bubbles. Although a 3D version of the model has been developed [22], diffusion is not included, and bubble growth is modeled by increasing bubble gas content in proportion to the bubble surface area.

Here, we present a 3D LBM simulation model for polymer foaming, which accounts for the disjoining pressure between bubbles, surface tension, film drainage, and bubble dynamics including bubble growth, deformation, coalescence, bursting, splitting, and dissolution. The shear thinning behavior of polymer liquids is also included. The effect of a solidified skin layer is accounted for via a zero-flux Neumann boundary condition for moving interfaces. The incorporation of the
(a) D2Q9 lattice. \( w_0 = \frac{4}{9}, \ w_1, \ldots, w_4 = \frac{1}{9}, \ w_5, \ldots, w_9 = \frac{1}{36} \).

(b) D3Q19 lattice. \( w_0 = \frac{1}{3}, \ w_1, \ldots, w_6 = \frac{1}{18}, \ w_7, \ldots, w_{18} = \frac{1}{36} \).

Figure 1: 2D and 3D lattices for the Navier-Stokes equation.

zero-flux boundary condition, allows for a direct comparison between the LBM model and the Cell Model \[5\] for the first time. The simulation model has been parallelized by means of the MPI library to allow large-scale simulations.

In the following sections, we present a lattice Boltzmann formulation of polymer foaming, then validate the model by comparison to Cell Model results, and finally demonstrate the capabilities of the model by examining different foaming phenomena.

2. Lattice Boltzmann Method

The lattice Boltzmann method uses a discretized Boltzmann equation with a collision kernel to solve the Navier-Stokes equations for fluid flow. Space is discretized with square (in 2D) or cubic lattices (in 3D) with a finite set of \( N \) discrete lattice velocities \( \vec{e}_i \) (\( i = 0, \ldots, N-1 \)). In 2D and 3D, we use D2Q9 and D3Q19 lattices, respectively. The lattice velocity sets for the D3Q19 lattice are:

\[
\vec{e}_i^T = \begin{cases} 
(0,0,0), \quad i = 0 \\
(\pm 1,0,0),(0,\pm 1,0),(0,0,\pm 1), \quad i = 1, 2, 3, 4, 5, 6 \\
(\pm 1,\pm 1,0),(\pm 1,0,\pm 1),(0,\pm 1,\pm 1), \quad i = 7, \ldots, 18
\end{cases}
\]

Fig. (1) shows the lattices and their respective weights \( w_i \). The spatial resolution of a domain in 3D corresponds to the lattice width \( \delta_l \), such that a box of length \( l_x \), \( l_y \) and \( l_z \) is comprised of \( \frac{l_x}{\delta_l} \), \( \frac{l_y}{\delta_l} \) and \( \frac{l_z}{\delta_l} \) lattices (also referred to as cells in this paper). The particle distribution function (PDF) for each lattice direction at time \( t \) is given by \( f_i(\vec{x},t) \). The lattice Boltzmann equation reads

\[
f_i(\vec{x} + \vec{e}_i \delta_l, t + \delta_t) = f_i(\vec{x},t) + C(\vec{x},t)
\]
where \( \delta_t \) is the time step, and \( \hat{C} \) is a collision operator. Equation (2) is often decomposed into collision and streaming steps:

**Collision**: \( f^c_i(\vec{x}, t) = f_i(\vec{x}, t) + \hat{C}(\vec{x}, t) \) (3)

**Streaming**: \( f_i(\vec{x} + \vec{e}_i \delta_t, t + \delta_t) = f^c_i(\vec{x}, t) \) (4)

where \( f^c_i(\vec{x}, t) \) denotes a post-collision PDF. In this work, the collision operator is replaced by the Bhatnagar-Gross-Krook (BGK) model for isothermal systems with an external force term \( F_i \) [18]:

\[
\hat{C}(\vec{x}, t) = -\frac{\delta_t}{\tau} (f_i(\vec{x}, t) - f^e_i(\vec{x}, t)) - F_i
\]

where \( \tau \) is the time interval between particle collisions.

The equilibrium distribution function \( f^e_i \) is the truncated second-order Taylor expansion of the Maxwell distribution function with respect to flow velocity \( \bar{u} \)

\[
f^e_i(\rho, \bar{u}) = \rho w_i \left( 1 + \frac{\bar{u} \cdot \vec{e}_i}{c_s^2} + \frac{\left( \frac{\bar{u} \cdot \vec{e}_i}{c_s^2} \right)^2}{2c_s^4} - \frac{\bar{u}^2}{2c_s^2} \right)
\] (6)

\( \rho \) is the density and \( c_s \) is the speed of sound, that has a value of \( 1/\sqrt{3} \) for the D2Q9 and D3Q19 lattices. The microscopic quantities \( \rho \) and \( \bar{u} \) are calculated by summation of PDFs:

\[
\rho = \sum_{i=0}^{N} f_i
\]

\[
\bar{u} = \frac{1}{\rho} \sum_{i=0}^{N} f_i \vec{e}_i
\]

and pressure is proportional to the density:

\[
p = \rho c_s^2
\]

The force term \( F_i \) in Eq. 5 accounts for external forces such as gravity \( g \). For an external force generating an acceleration \( \bar{a} \), we have:

\[
F_i = \rho w_i \left( \frac{\vec{e}_i^T \bar{u}}{2c_s^4} - \frac{\bar{u} - \vec{e}_i}{c_s^2} \right) \cdot \bar{a}
\]

The bounce-back scheme [19] is used at wall nodes, meaning that particles colliding with a wall have their momentum reversed, which translates to a no-slip boundary condition. Hereafter, we use \( \delta_t = \delta_t = 1 \) for simplicity.
fluid cells ($\alpha = 1$)

interface cells ($0 < \alpha < 1$)

gas cells ($\alpha = 0$)

Figure 2: Demonstrating the VOF scalar field.

3. Foaming Model

3.1. Free Surface Model

The liquid-gas interface is modelled as a free surface. Since the liquid/gas ratios of densities and viscosities of liquid and gas are often very large, the hydrodynamics of the gas phase can be ignored compared to that of the liquid; i.e. the liquid moves freely with respect to the gas and the only influence of the gas phase is the pressure it exerts on the free surface. Using a volume-of-fluid (VOF) method, one can introduce a dimensionless scalar variable $\alpha$ to track the free surface throughout the computational domain. $\alpha$ is defined as:

$$\alpha = \frac{M}{\rho \delta_x^3}$$  \hspace{1cm} (11)

where $M$ is the liquid mass in a cell. $\alpha$ is equal to one in liquid cells, zero in gas cells, and it has a value between zero and one in all interface cells (see Fig. [2]):

$$\alpha = \begin{cases} 
0, & \forall \vec{x} \in G \\
1, & \forall \vec{x} \in L \\
0 < \alpha < 1 & \forall \vec{x} \in I 
\end{cases}$$  \hspace{1cm} (12)

$G$, $L$, and $I$ represent gas, liquid, and interface cells, respectively.

The value of $\alpha(\vec{x}, t + 1)$ is found by calculating the mass exchange between neighboring cells $\vec{x}$ and $\vec{x} + \vec{e}_i$:

$$\Delta \alpha(\vec{x}, t + 1) = \alpha(\vec{x}, t) + \frac{1}{\rho(\vec{x}, t)} \sum_{i=0}^{N} \Theta(f_i(\vec{x} + \vec{e}_i, t) - f_i(\vec{x}, t))$$  \hspace{1cm} (13)

where $\vec{e}_i = -i$. The parameter $\Theta$ in Eq. (13) weights the mass exchange between two interface cells by their average volume fraction:
\[ \Theta = \begin{cases} 
0, & \vec{x} + \vec{e}_i \in G \\
1, & \vec{x} + \vec{e}_i \in L \\
\frac{1}{2} (\alpha(\vec{x}, t) + \alpha(\vec{x} + \vec{e}_i, t)) & \vec{x} + \vec{e}_i \in I 
\end{cases} \] (14)

Equation (13) conserves mass locally since the mass that leaves one cell is distributed to neighboring cells. An interface cell becomes a liquid cell when \( \alpha(\vec{x}, t) \geq 1 \), and an interface cell becomes an empty cell when \( \alpha(\vec{x}, t) \leq 0 \). Where there is an excess or shortage of mass in a cell (i.e. \( \alpha(\vec{x}, t) > 1 \) or \( \alpha(\vec{x}, t) < 0 \)), the surplus or shortage is uniformly distributed to neighboring interface cells.

3.2. Free Surface Boundary Condition

In the streaming step (i.e. Eq. (4)), the PDFs arriving from gas cells are undefined in interface cells. These undefined PDFs \( f^n_i \) must be reconstructed such that the pressure boundary condition at the free surface is enforced, and so that the velocity of gas cells is equal to the adjacent liquid velocity \( \vec{u}_I \). For this purpose, the unknown distribution functions at the interface are reconstructed as [22]:

\[ f^n_i(\vec{x} - \vec{e}_i, t + 1) = f^{eq}_i(\rho_g, \vec{u}_I) + f^{eq}_i(\rho_g, \vec{u}_I) - f_i(\vec{x} - \vec{e}_i, t + 1) \] (15)

The effect of surface tension is included in the model by modifying the gas density \( \rho_g \) in Eq. (15) [22]:

\[ \rho_g = \frac{p_g - 2 \gamma \kappa(\vec{x})}{c_s^2} \quad \vec{x} \in I \] (16)

\( \gamma \) is the surface tension, and \( \kappa(\vec{x}) \) is the local curvature which is calculated using the gradient of the smoothed \( \alpha \) scalar field as described in Ref. [10]. In Eq. (16), gas pressure \( p_g \) for each individual bubble \( i \) is given by the ideal gas law:

\[ p^i_g = \frac{m^i_g \mathcal{R} T}{V^i} \] (17)

where \( m^i_g \), \( \mathcal{R} \), \( T \), and \( V^i \) are the mass of the bubble, gas constant, temperature, and volume of the bubble.

3.3. Disjoining Pressure

The formation of lamellae between bubbles is numerically unstable using the volume-of-fluid method, because two interface cells belonging to two different bubbles merge together upon contact. The formation of lamellae is made possible by means of a disjoining pressure (\( \Pi \)) that hinders coalescence and stabilizes a foam using a repulsive force [22]. Physically, \( \Pi \) originates from the variation of Gibbs free energy with the distance between two interfaces \( d \) that corresponds to different bubbles [20]. The disjoining pressure is assumed to be active only up to a distance \( d_{max} \) between interfaces, with linear dependence on \( d \):
Figure 3: Calculating the distance between two interfaces belonging to adjacent bubbles. The traversal algorithm marches through the shaded cells.

\[
\Pi = \begin{cases} 
0 & d > d_{\text{max}} \\
 k_{\Pi}(d_{\text{max}} - d) & d < d_{\text{max}}
\end{cases}
\]

where \(k_{\Pi}\) is a constant. The distance \(d\) between two different bubbles \(i\) and \(j\) (see Fig. 3) is evaluated by first moving along the bubble \(i\) interface normal \(\vec{n}\) away from the bubble, until reaching an interface that belongs to bubble \(j\). Then, the interfaces of each bubble are geometrically reconstructed using a piecewise linear interface calculation (PLIC), and \(d\) is calculated as the distance between the reconstructed interfaces (See Fig. (11)).

To detect adjacent bubbles, the fast traversal ray tracing algorithm developed by Amanatides and Woo \(\Pi\) is implemented to march along the normal vector. A full description of the ray tracing algorithm is given in Appendix A. Also, since LBM uses cubic or square lattices, we use the explicit analytical expressions developed by Scardovelli and Zaleski [30] for PLIC calculations, which is significantly faster than using Newton-Raphson iterations.

Similar to the surface tension implementation, the disjoining pressure is added to \(\rho_g\):

\[
\rho_g = \frac{p_g - 2\gamma n(\vec{x}) - \Pi}{c_s^2} 
\]

\(\vec{x} \in I\) (19)

3.4. Advection-Diffusion

The diffusion of dissolved blowing agent within the liquid polymer and into bubbles is governed by an advection-diffusion equation. This equation is solved using another distribution function \(g_i(\vec{x}, t)\) such that the summation of \(g_i\) gives the gas concentration (in terms of mass fraction) at location \(\vec{x}\) at time \(t\):

\[
c(\vec{x}, t) = \sum_i^N g_i(\vec{x}, t) 
\]

The lattice Boltzmann equation for the evolution of \(g_i(\vec{x}, t)\) is given by:
We exploit the linear nature of the advection-diffusion equation to use lattices with fewer velocity vectors. Therefore, the numerical solution of equation (21) is carried out on D2Q5 (in 2D) or D3Q7 (in 3D) lattice topologies (see Figure 4) that coincide with the fluid flow lattices. These lattices require less computational time because of the reduced number of velocities; however, they cannot be used for Eq. (2) due to their lower degree of isotropy \[17\].

\[ \tau_g \] is the relaxation time for the advection-diffusion equation, which relates to the diffusion constant \( D \) of the gas in the liquid through the following equation:

\[ D = c_s^2 \left( \tau_g - \frac{1}{2} \right) \] (22)

where \( c_s^2 \) is equal to 1/2 and 1/4 for the D2Q5 and D3Q7 lattices, respectively.

The equilibrium distribution in the advection-diffusion equation is given by:

\[ g_{eq}^i(\vec{x},t) = w_i c(\vec{x},t) \left( 1 + \vec{e}_i \cdot \vec{u} \right) \] (23)

where \( \vec{u} \) is calculated from Eq. (8).

The gas diffusing into bubbles is calculated using the concentration gradient at the liquid-bubble interface:

\[ \Delta m_g^i = \rho \sum_{\vec{x}_i \in I^i} \left( \sum_{\vec{x} + \vec{e}_i \in F} g_i(\vec{x} + \vec{e}_i, t) - g_i(\vec{x}, t) \right) - c(\vec{x}, t) [\alpha(\vec{x}, t) - \alpha(\vec{x}, t - 1)] \] (24)

where \( I^i \) is the liquid-gas interface of bubble \( i \). The last term in Eq. (24) deducts the pure advection portion of gas transport, which is not diffused into the bubble.
3.5. **Henry’s Law Boundary Condition**

In polymer foams, the concentration of gas at a bubble interface obeys Henry’s law, which expresses a linear relationship between pressure and gas concentration:

\[
c(\vec{x}, t) = k_H p_g \quad \forall \vec{x} \in I
\]  

(25)

This boundary condition is imposed by setting the unknown distribution coming from the gas cells \( g_i^u \) to

\[
g_i^u(\vec{x} - \vec{e}_i, t + 1) = g_i^{eq}(c, \vec{u}_I) + g_i^{eq}(c, \vec{u}_I) - g_i(\vec{x} - \vec{e}_i, t + 1)
\]  

(26)

where \( c \) is given by Eq. (25) [22].

3.6. **Free Surface Neumann Boundary Condition for the Advection-Diffusion Equation**

Foam injection molded products are usually characterized by cellular cores surrounded by solid skin layers, which form when polymer melt contacts the cold surface of a mold. The solid skin layers can hinder the gas diffusion (or gas escape) across the free boundaries of an expanding foam, increasing the gas concentration near the free surface which can cause larger bubbles to form near the skin layers (e.g. see Fig. (5)). Simulating the effect of a skin layer on gas diffusion requires applying a Neumann boundary condition (i.e. zero-flux) to the advection-diffusion equation at free boundaries. The free boundaries are moving, and therefore applying a simple bounce-back boundary condition results in an unphysical concentration of gas at the boundaries, depending on the interface velocity. To solve this problem, we implemented the Markl and Körner Neumann boundary condition model developed for arbitrary moving boundaries [25], which is more physically realistic compared to other models available in the literature (e.g. [8]).

First, the unknown PDFs streaming from the gas phase are approximated by the streamed values from the interface cells to adjacent liquid cells:

\[
g'(\vec{x}, t) = \begin{cases} 
g_i(\vec{x} + \vec{e}_i, t), & \vec{x} - \vec{e}_i \in G \\
g_i(\vec{x}, t), & \text{else.} \end{cases}
\]  

(27)

Next, the flux in the normal direction to the free surface \( q_n \) is computed using the approximated PDFs:

\[
q_n = \vec{n} \cdot \left( \sum_i (\vec{e}_i - (1 - \tau_g)\vec{u}) g_i^t - \tau_g c(\vec{x}, t - 1)\vec{u}(\vec{x}, t - 1) \right)
\]  

(28)

Finally, the flux in the normal direction is inverted and used to correct the approximated PDF to impose the zero-flux condition

\[
g(\vec{x}, t) = \begin{cases} 
g_i(\vec{x}, t) - g_i^t(\vec{x}, t), & \vec{n} \cdot (\vec{e}_i - \vec{u}) < 0 \\
g_i^t(\vec{x}, t), & \text{else.} \end{cases}
\]  

(29)
where $g_i^*$ is calculated from the compensating source term that results in a zero-flux at the free surface (see [25] for details):

$$g_i^*(\vec{x},t) = w_i(\vec{e}_i + (1 - \tau_g)\vec{u}) \cdot (\vec{n} \cdot \mathbf{M}^{-1})$$  \hspace{1cm} (30)$$

$$M = \sum_{i,\vec{n}(\vec{e}_i-\vec{u})<0} w_i(\vec{e}_i\vec{e}_i - (1 - \tau_g)^2\vec{u}\vec{u})$$  \hspace{1cm} (31)$$

3.7. Viscosity

The Carreau-Yasuda model is widely used to describe the shear thinning behavior of polymers. In this model, the apparent viscosity $\mu$ is given by

$$\frac{\mu - \mu_\infty}{\mu_0 - \mu_\infty} = (1 + (\lambda\dot{\gamma})^a)^{\frac{n-1}{a}}$$  \hspace{1cm} (32)$$

where $n$, $a$, and $\lambda$ are empirically-determined material coefficients. At low shear rates ($\dot{\gamma} \ll 1/\lambda$) a Carreau fluid behaves as a Newtonian fluid, and at high shear rates ($\dot{\gamma} \gg 1/\lambda$) as a power-law fluid.

The shear rate tensor for Eq. (32) can be calculated locally from the distribution functions [9]:

$$\dot{\gamma}_{\alpha\beta} = \frac{1}{2\tau_l\epsilon_0} \sum_i \vec{e}_{i,\alpha}\vec{e}_{i,\beta}(f_i - f_i^{eq})$$  \hspace{1cm} (33)$$

where $\alpha$ and $\beta$ are spatial free indices. The magnitude of the shear rate tensor is then calculated as:

$$|\dot{\gamma}| = \sqrt{\sum_{\alpha} \sum_{\beta} \dot{\gamma}_{\alpha\beta}^2}$$  \hspace{1cm} (34)$$

Figure 5: SEM image of an injection molded foam (with mold opening) sample of Polystyrene with 5wt% CO$_2$ [31].
In the collision operator (i.e. Eq. (5)), $\tau$ is related to kinematic viscosity $\nu$ by

$$\nu = c_s^2 \left( \tau - \frac{1}{2} \right)$$  \hspace{1cm} (35)

For a Newtonian fluid, $\tau$ is a constant. For the Carreau-Yasuda model, $\tau$ is treated locally by substituting $\tau$ from Eq. (35) into Eq. (32):

$$\frac{\tau - \tau_\infty}{\tau_o - \tau_\infty} = (1 + (\lambda \dot{\gamma})^a)^{\frac{n-1}{n}}$$  \hspace{1cm} (36)

Eqs. (33) and (36) are both a function of $\tau$, and must be solved implicitly.

4. Implementation

We used the Palabos open source framework which features basic tools for parallelization of LBM simulations using Message Passing Interface (MPI). Palabos uses a data-structure called “multi-block” that is composed of several Cartesian meshes, which combined create the computational domain. These cartesian meshes are distributed to a number of processing units.

Every processor holds information about the bubbles residing on its sub-domain, such as ID, volume, and gas content. The ID of each bubble is unique and global among processors. There are two scenarios that require special communication (data transfer) between processors. The first one relates to the calculation of the disjoining pressure: when a bubble interface is closer than $d_{max}$ from a sub-domain boundary, another bubble could be adjacent to the bubble in another sub-domain. In this case, the communication envelope between the two processors must be expanded as much as $d_{max}$. For instance, if $d_{max} = 3\delta_l$, then the communication envelope must be expanded by three cells between adjacent sub-domains. Second, when a bubble occupies multiple sub-domains, bubble gas content and volume must be shared among all sub-domain processors. Since gas can diffuse into a bubble from different sub-domains, this communication is necessary to calculate the correct value of gas content at each iteration.

The simulations presented here were carried out on the SciNet Niagara supercomputer at the University of Toronto, which is a homogeneous cluster of 60,000 cores (1500 nodes). We evaluated the performance of our model on up to 100 nodes (with 40 processors each). Fig. (6) shows scaling performance of the model for a 3D simulation with 60 million cells and 100 bubbles. The simulation requires about 100GB of RAM. We were able to achieve better than ideal super-linear scaling by increasing the number of nodes. This improvement in performance is not only related to the number of cores, but also to the high memory bandwidth and large cache memory of the CPUs when using more than one node for the simulation.
5. Results and Discussion

5.1. Validation

We begin by comparing results of the 3D LBM model to results of the 1D Cell Model (CM) developed by Arefmanesh and Advani [5], which has been extensively cited in the literature. In what follows, the subscripts 0 and $f$ refer to the initial and final values of a parameter, respectively.

The CM approximates a foam as a homogeneous array of spherical bubbles growing in close proximity to each other. The conservation equations are solved for a representative bubble surrounded by a polymer melt shell as shown in Fig. (7). The radius of the bubble and the radius of the polymer shell are denoted by $R$ and $S$, respectively. $r$ is the radial position ($R \leq r \leq S$) and $c$ is the dissolved gas concentration in weight ratio (wt%). For this system, given the dimensionless parameters in Table 1, an advection-diffusion equation is solved supplemented by a zero-flux boundary condition at the outer shell, and Henry’s law at the bubble-liquid interface, as shown in Fig. (7), to calculate the bubble growth rate.

To perform a simulation equivalent to the CM in LBM, a bubble was placed in the middle of a cubic domain surrounded by a polymer melt shell as shown in Fig. (8). The domain boundaries were set as outflow boundaries. The initial gas content of the shell was set to $c_0$. In the Cell Model, the spherical symmetry of the computational domain allows for use of the Lagrangian frame of reference to set the boundary conditions at $r = R$ and $r = S$. The same is not possible.
Figure 7: Schematic of the Cell Model. The bubble is surrounded by a limited quantity of polymer melt.

\[
\frac{\rho^2}{S_0^2 - R_0^2} = 2.38 \times 10^{-2} \\
\frac{\mu \beta T}{M_{pf}} = 150 \\
\frac{2\gamma}{R_0 p_f} = 3 \times 10^{-3} \\
\frac{4\mu D}{p_f R_0} = 0.15 \\
k_{HF} p_f = 3.33 \times 10^{-6} \\
c_0 = 0.02 \text{ and } 0.01
\]

Table 1: Validation simulation parameters in terms of dimensionless numbers. \( M \) is the molecular weight of the gas and \( \beta \) is the universal gas constant.

In LBM, and so we take advantage of the methodologies introduced in Sections (3.5) and (3.6) to impose Henry’s law and the zero-flux boundary condition.
A comparison of CM and LBM results is shown in Fig. (9) for two different initial gas concentrations ($c_0 = 0.02$ for CM1/LBM1 and $c_0 = 0.01$ for CM2/LBM2). The dimensionless parameters are listed in Table (1). It can be seen that there is a good agreement between the LBM and CM simulations, although the discrepancy between the LBM1 and CM1 results is especially larger at the beginning of bubble growth. In LBM, the size of the grid does not scale with bubble interface area (unlike CM), which can cause overestimation of bubble surface area at the beginning of the simulation, and thus result in a larger bubble growth. Some of the discrepancy can also be attributed to deviation of the bubble from spherical symmetry, due to rapid expansion of the bubble in LBM1 compared to LBM2 that the bubble maintained its spherical symmetry throughout the simulation, apart from small spurious currents that wobbled the bubble/shell interface, as shown in the the simulation snapshots in Fig. (9).
5.2. Foaming

In the following, we parameterize simulations based on lattice units. The conversion between lattice units and SI units is straightforward and can be done by defining a length scale and a reference density for the simulation. Examples of unit conversions for foaming simulations are given in [22, 23]. The system parameters for each simulation are given in the figure captions.

Figure (10) demonstrates a 2D simulation of foaming in a rectangular container using 150 nuclei. The simulation domain consists of 1000 × 1000 cells. Initially, the nuclei are periodically distributed, each with a radius of 3 cells. It can be seen that the Henry’s law boundary condition at the bubble-melt interfaces generates a diffusion profile around each bubble, with lower gas concentration near bubble interfaces. The diffusion flux reduces the available gas in the liquid and grows the bubbles. Once the gas is depleted, the final foam structure is attained.
Figure 10: Volume fraction field (left) and gas concentration field (right) of a 2D simulation of polymer foaming with 150 nuclei distributed periodically. Parameters: $\tau = 0.95$, $\tau_g = 0.53$, $RT = 50$, $c_0 = 0.02$, $k_H = 2 \times 10^{-3}$, $\rho = 1$, $k_H = 10^{-5}$, $g = 0$, $\gamma = 5 \times 10^{-3}$.

5.3. Disjoining Pressure

The effect of the disjoining pressure on foam formation is shown in Fig. (11). As gravity draws liquid to the bottom of the domain, the disjoining pressure
field (shown in Fig. 12) prevents cell borders from further thinning, allowing formation of lamellae between bubbles. Without a disjoining pressure, the foam collapses under the influence of gravity. And while some believe that foaming can only be achieved with the existence of a disjoining pressure 22, 23, Fig. 13 illustrates that foaming may be attainable without disjoining pressure in the absence of gravity, that drains the liquid from between the bubbles. Therefore, for a system with very small bubbles and high viscosity, such as a microcellular foam, due to slow liquid drainage, the foam may solidify into its final structure before collapsing.
Figure 11: The disjoining pressure hinders bubble coalescence and allows development of straight cell walls. Parameters: \( \tau = 0.95, \tau_g = 0.53, RT = 50, c_0 = 0.02, \rho = 1, g = 3 \times 10^{-5}, \gamma = 5 \times 10^{-3}, \text{num. bubbles} = 150, \text{distribution: random.} \)
5.4. Foam Drainage

Under the influence of gravity, liquid gradually leaks out of a foam to the bottom of its container, leaving the top of the foam dry and fragile. This effect is known as foam drainage, and has been extensively studied in the literature for Newtonian and non-Newtonian fluids (e.g. [3, 7, 26, 28, 29, 33]). Foam drainage studies have been largely experimental, and little mathematical and numerical modeling of foam drainage has been done.

As shown in Fig. 14, foam drainage can be observed in the simulation results in the presence of gravity. The liquid accumulates at the bottom of the container, depleting the top of liquid. It can also be seen that after drainage the foam evolves into an array of 2D hexagonal bubbles.
The natural emergence of foam drainage illustrates the capability of the model for foam drainage studies of Newtonian and non-Newtonian systems. For example, in the drainage simulation in Fig. (14), the initial nuclei are distributed uniformly, which minimizes the width of channels that liquid can drain through. Thus, one would expect that a random distribution of bubbles would enhance foam drainage. The simulation result previously shown in Fig. (11) has the same parameters as the one in Fig. (14), but with a random initial bubble distribution. It can be seen that at the final stage of the simulation, a random bubble distribution results in greater drainage of liquid to the bottom of the domain.

Figure 14: The influence of gravity on foam drainage. In both simulations $t = 30000$. Left: Parameters: $\tau = 0.95$, $\tau_g = 0.53$, $RT = 50$, $c_0 = 0.02$, $k_H = 2 \times 10^{-3}$, $\rho = 1$, $k_H = 10^{-5}$, $g = 0$, $\gamma = 5 \times 10^{-3}$, num. bubbles = 150, distribution : uniform. Right: Parameters: Same as (a) except $g = 3 \times 10^{-5}$.

5.5. Non-Newtonian Fluid

To examine the potential of the non-Newtonian model, the drainage simulation of Fig. (14) is reproduced using a large value of $\lambda = 10$, and $\tau_\infty = 0$, to substantially reduce the viscosity due to the shear thinning compared to the viscosity of the system in Fig. (14).

In Fig. (15), the $\tau$ field is shown, which is proportional to the liquid viscosity (see Eq. (35)). Small motions in the thin films between the bubbles generate large strain rates, causing the shear viscosity to be substantially reduced within the cell walls. Experimentally, a shear-thinning viscosity has been shown to enhance foam drainage and bursting of large bubbles at the free surface [26]. The same phenomena can be seen in Fig. (16) which compares foam drainage between Newtonian and non-Newtonian systems.
Figure 15: Reduction of the shear viscosity (proportional to $\tau$) in foam cell walls due to shear thinning. Parameters: $\tau_0 = 0.95$, $\tau_\infty = 0$, $\lambda = 10$, $\tau_g = 0.53$, $RT = 50$, $c_0 = 0.02$, $k_H = 2 \times 10^{-3}$, $\rho = 1$, $k_H = 10^{-5}$, $g = 3 \times 10^{-5}$, $\gamma = 5 \times 10^{-3}$, num. bubbles = 150, distribution: uniform.

Figure 16: Comparison of foam drainage with Newtonian and non-Newtonian viscosity. $t = 30000$ for both systems. Left: Parameters: Same as Fig. (14). Right: Parameters: Same as Figure (15).

5.6. Bubble Dissolution

In foam injection molding (FIM), bubbles nucleate due to a pressure drop at the inlet gate during filling. Due to large shear stresses during the mold-filling process, these gate-nucleated bubbles become undesirably elongated [32]. To promote foam uniformity, in high-pressure FIM, these bubbles are re-dissolved into the polymer by increasing the cavity pressure to above the solubility pressure. The increase in the solubility initiates diffusion of gas from the bubbles to the polymer melt. Uniform bubbles are then nucleated again by a second pressure drop (due to melt shrinkage or mold opening) [34].
The foaming model presented here can also be used to study bubble dissolution. In Fig. (17), a bubble is placed in the middle of a liquid with no initial gas content. The non-zero concentration of gas at the bubble-liquid interface transports gas from the bubble to the liquid, until the bubble is fully dissolved. Finally, the dissolved gas diffuses throughout the liquid.

Figure 17: Dissolution of a bubble in a liquid with no initial gas content. The concentration gradient around the bubble at $t = 0$ is due to the Henry’s law boundary condition at the liquid-bubble interface. Parameters: $\tau = 0.98$, $\tau_g = 0.8$, $RT = 500$, $c_0 = 0$, $k_H = 0$, $\rho = 1$, $k_H = 5 \times 10^{-3}$, $g = 0$, $\gamma = 4.2 \times 10^{-3}$.

5.7. Diffusion Across the Free Interface

Figure (18) shows the effect of a skin layer on foam structure development, which prevents gas from escaping the polymer. It can be seen that accumulated gas results in development of larger bubbles near the free interface in contact with the surroundings (i.e. see Fig. (5)).

Although gas also accumulates near walls, the bubbles there are not as large as the bubbles near free interfaces, as walls prevent bubble expansion.
Figure 18: Comparison of foaming without (top) and with (bottom) a skin layer on the foam surface. A zero-flux boundary condition is applied at the free interface for the bottom simulation. Parameters: \( \tau = 0.95, \tau_g = 0.53, RT = 50, c_0 = 0.01, k_H = 2 \times 10^{-3}, \rho = 1, k_H = 10^{-5}, g = 0, \gamma = 5 \times 10^{-3}, \) num. bubbles = 150, distribution: uniform.

5.8. 3D results

We have developed what we believe is a first 3D foaming model that solves for bubble growth using an advection-diffusion equation and incorporates non-Newtonian viscosity. An example of a 3D simulation is given in Fig. 19. 40 bubble nuclei are randomly distributed in a liquid polymer, each with an initial radius of 3 cells. The domain is made up of 300 \( \times \) 200 \( \times \) 200 cells or 12 million cells in total.
6. Conclusion

We have developed a free surface LBM model that can perform polymer foaming simulations in 2D and 3D. An advection-diffusion equation is coupled to a fluid flow solver to account for dissolved gas diffusion into bubbles, subject to a Henry’s law boundary condition at bubble-polymer interfaces. A volume-of-fluid method is used to track and locate bubble interfaces. The model accounts for disjoining pressure between bubbles, surface tension, film drainage, and bubble dynamics, including bubble growth, deformation, coalescence, bursting and
splitting. To incorporate the disjoining pressure into the model, a fast traversal algorithm is used to calculate the distance between adjacent bubbles. The simulation model is parallelized by means of the MPI library to improve its capability for large scale simulations. Simulation results demonstrate the potential of the model to provide a better understanding of the evolution, structure, and rheology of polymer foams.

In the future, we are planning to incorporate Immerse Boundary Method in the LBM model to enable study of bubble-fiber interaction. We are also working on accelerating the 3D large-scale foaming simulations by orders of magnitude using data-driven deep learning techniques.

Acknowledgements

We thank the Natural Sciences and Engineering Research Council of Canada (NSERC) and Autodesk Inc. for their financial support, and the SciNet supercomputer centre at the University of Toronto for providing computational resources.

References

References

[1] J. Amanatides and A. Woo. A fast voxel traversal algorithm for ray tracing. Dept. of Computer Science, University of Toronto, 1984.

[2] M. Amon and C. Denson. A study of the dynamics of foam growth: Analysis of the growth of closely spaced spherical bubbles. Polymer Engineering & Science, 24(13):1026–1034, 1984.

[3] A. Anazadehsayed, N. Rezaee, and J. Naser. Exterior foam drainage and flow regime switch in the foams. Journal of Colloid and Interface Science, 511:440–446, 2018.

[4] A. Arefmanesh and S. G. Advani. Diffusion-induced growth of gas bubble in a viscoelastic fluid. Rheologica Acta, 30:274, 1991.

[5] A. Arefmanesh and S. G. Advani. Nonisothermal bubble growth in polymeric foams. Polymer Engineering & Science, 35(3):252–260, 1995.

[6] A. Arefmanesh, S. G. Advani, and E. E. Michaelides. A numerical study of bubble growth during low pressure structural foam molding process. Polymer Engineering and Science, 30(20):1330–1337, 1990.

[7] O. Arjmandi-Tash, A. Trybala, F. Mahdi, N. Kovalchuk, and V. Starov. Foams built up by non-Newtonian polymeric solutions: Free drainage. Colloids and Surfaces A: Physicochemical and Engineering Aspects, 521:112–120, 2017.

[8] E. Attar and C. Körner. Lattice Boltzmann model for thermal free surface flows with liquid-solid phase transition. International Journal of Heat and Fluid Flow, 32(1):156–163, 2011.
[9] Z. Chuanhu, C. Songgui, S. Qicheng, and J. Feng. Free-surface simulations of newtonian and non-newtonian fluids with the lattice boltzmann method. Acta Geologica Sinica - English Edition, 90(3):999–1010, 2016.

[10] S. J. Cummins, M. M. Francois, and D. B. Kothe. Estimating curvature from volume fractions. Computers & Structures, 83(6):425–434, 2005.

[11] S. Donath, C. Feichtinger, T. Pohl, J. Götz, and U. Rüde. Localized parallel algorithm for bubble coalescence in free surface lattice-boltzmann method. In H. Sips, D. Epema, and H.-X. Lin, editors, Euro-Par 2009 Parallel Processing, pages 735–746, Berlin, Heidelberg, 2009. Springer Berlin Heidelberg.

[12] R. Elshereef, J. Vlachopoulos, and A. Elkamel. Comparison and analysis of bubble growth and foam formation models. Engineering Computations, 27(3):387–408, 2010.

[13] S. Everitt, O. Harlen, and H. Wilson. Bubble growth in a two-dimensional viscoelastic foam. Journal of Non-Newtonian Fluid Mechanics, 137(1):46–59, 2006.

[14] S. Everitt, O. Harlen, and H. Wilson. Competition and interaction of polydisperse bubbles in polymer foams. Journal of Non-Newtonian Fluid Mechanics, 137(1):60–71, 2006.

[15] S. L. Everitt, O. G. Harlen, and H. J. Wilson. Bubble growth in a two-dimensional viscoelastic foam. Journal of Non-Newtonian Fluid Mechanics, 137(1-3):46–59, 2006.

[16] S. Geier and M. Piesche. Coupled macro and micro-scale modeling of polyurethane foaming processes. The Journal of Computational Multiphase Flows, 6(4):377–390, 2014.

[17] I. Ginzburg. Prediction of the moments in advection-diffusion lattice Boltzmann method. II. Attenuation of the boundary layers via double-A bounce-back flux scheme. Physical Review E, 95:013305, 2017.

[18] X. He and L.-S. Luo. Lattice Boltzmann model for the incompressible Navier-Stokes equation. Journal of Statistical Physics, 88(3):927–944, 1997.

[19] X. He, Q. Zou, L.-S. Luo, and M. Dembo. Analytic solutions of simple flows and analysis of nonslip boundary conditions for the lattice Boltzmann BGK model. Journal of Statistical Physics, 87(1):115–136, 1997.

[20] C. Huber, Y. Su, C. T. Nguyen, A. Parmigiani, H. M. Gonnermann, and J. Dufek. A new bubble dynamics model to study bubble growth, deformation, and coalescence. Journal of Geophysical Research: Solid Earth, 119(1):216–239, 2014.

[21] M. Karimi, H. Droghetti, and D. L. Marchisio. PUFoam : A novel open-source CFD solver for the simulation of polyurethane foams. Computer Physics Communications, 217:138–148, 2017.

[22] C. Körner, M. Thies, T. Hofmann, N. Thürey, and U. Rüde. Lattice Boltzmann model for free surface flow for modeling foaming. Journal of Statistical Physics, 121(1):179–196, 2005.

[23] C. Körner, M. Thies, and R. Singer. Modeling of metal foaming with lattice Boltzmann automata. Advanced Engineering Materials, 4(10):765–769, 2002.
[24] S. N. Leung, C. B. Park, D. Xu, H. Li, and R. G. Fenton. Computer simulation of bubble-growth phenomena in foaming. *Industrial and Engineering Chemistry Research*, 45(23):7823–7831, 2006.

[25] M. Markl and C. Körner. Free surface Neumann boundary condition for the advection-diffusion lattice Boltzmann method. *Journal of Computational Physics*, 301:230–246, 2015.

[26] M. Morey, N. Deshpande, and M. Barigou. Foam destabilization by mechanical and ultrasonic vibrations. *Journal of Colloid and Interface Science*, 219(1):90–98, 1999.

[27] R. D. Patel. Bubble growth in a viscous Newtonian liquid. *Chemical Engineering Science*, 35(11):2352–2356, 1980.

[28] M. Safouane, A. Saint-Jalmes, V. Bergeron, and D. Langevin. Viscosity effects in foam drainage: Newtonian and non-Newtonian foaming fluids. *The European Physical Journal E*, 19(2):195–202, 2006.

[29] A. Saint-Jalmes. Physical chemistry in foam drainage and coarsening. *Soft Matter*, 2:836–849, 2006.

[30] R. Scardovelli and S. Zaleski. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. *Journal of Computational Physics*, 164(1):228–237, 2000.

[31] V. Shaayegan, C. Wang, F. Costa, S. Han, and C. B. Park. Effect of the melt compressibility and the pressure drop rate on the cell-nucleation behavior in foam injection molding with mold opening. *European Polymer Journal*, 92:314–325, 2017.

[32] V. Shaayegan, G. Wang, and C. B. Park. Study of the bubble nucleation and growth mechanisms in high-pressure foam injection molding through in-situ visualization. *European Polymer Journal*, 76:2–13, 2016.

[33] P. Stevenson. Dimensional analysis of foam drainage. *Chemical Engineering Science*, 61(14):4503–4510, 2006.

[34] M. Tromm, V. Shaayegan, C. Wang, H.-P. Heim, and C. B. Park. Investigation of the mold-filling phenomenon in high-pressure foam injection molding and its effects on the cellular structure in expanded foams. *Polymer*, 160:43–52, 2019.

[35] P. Yue, J. J. Feng, C. A. Bertelo, and H. H. Hu. An arbitrary Lagrangian-Eulerian method for simulating bubble growth in polymer foaming. *Journal of Computational Physics*, 226(2):2229–2249, 2007.
Appendix A. Ray Tracing Traversal Algorithm

The ray tracing algorithm for detection of adjacent bubbles is as follows:

For each interface cell, the algorithm marches along the interface normal $\vec{n}$ away from the bubble, until it finds an interface that belongs to another bubble. Referencing the traversal algorithm in 2D described in Algorithm 1 for an interface cell $i$, the line along the negative direction of the interface normal is defined as:

$$\vec{n} = -\vec{n} = \beta \begin{bmatrix} -n_x \\ -n_y \end{bmatrix} \quad \beta \geq 0$$

where $n_x$ and $n_y$ are the components of the normal vector. First, the algorithm calculates the values of $\beta$ required for $\vec{n}$ to cross the vertical ($betaMaxX$) and horizontal ($betaMaxY$) cell boundaries of cell $i$ ($betaMaxX$). The first neighboring cell along $\vec{n}$ is determined by the minimum of $betaMaxX$ and $betaMaxY$. In order to locate the subsequent neighboring cells along $\vec{n}$, the minimum of $betaMaxX$ and $betaMaxY$ is found after incrementing by the value of $\beta$ required to move along $\vec{n}$ for one cell length. Extending the algorithm to 3D is trivial and only requires calculating $betaMaxZ$ in the $z$ direction to calculate the minimum of $betaMaxX$, $betaMaxY$ and $betaMaxZ$. 


Algorithm 1: Traversal algorithm

1. $d_{\text{max}}$: maximum marching distance
2. $nx, ny$: outward normal components
3. $x, y$: cell coordinates
4. $I$: set of interface cells
5. $BC$: set of boundary cells
6. $ID$: bubble IDs
7. foreach $(nx, ny) \in I$ do
   // The value of $\beta$ required to exit the initial vertical and horizontal cell boundaries.
   8. betaMaxX ← abs$(0.5/nx)$;
   9. betaMaxY ← abs$(0.5/ny)$;
   // The value of $\beta$ required to move the length of one vertical and horizontal cell boundary.
   10. betaDeltaX ← $2 \times$ betaMaxX;
   11. betaDeltaY ← $2 \times$ betaMaxY;
   // stepX and stepY Determine direction of the marching algorithm.
   12. stepX ← $nx > 0.0$ ? 1 : $-1$;
   13. stepY ← $ny > 0.0$ ? 1 : $-1$;
   // nextX and nextY are initialized with the starting cell coordinates.
   14. nextX ← $x$;
   15. nextY ← $y$;
6. for $i \leftarrow 1$ to $d_{\text{max}}$ do
5. if betaMaxX < betaMaxY then
   6. betaMaxX += betaDeltaX;
   7. nextX += stepX;
4. else if betaMaxX == betaMaxY then
   8. betaMaxX += betaDeltaX;
   9. betaMaxY += betaDeltaY;
   10. nextX += stepX;
   11. nextY += stepY;
3. else
   12. betaMaxY += betaDeltaY;
   13. nextY += stepY;
2. if $ID(nx, ny) == ID(nextX, nextY)$ then
   14. continue;
1. if $(nextX, nextY) \in BC$ then
   15. break;
0. if $(nextX, nextY) \in I$ then
   16. return $(nextX, nextY)$;