Computer animation for fluid simulation of a high viscous fluid melting

S. Lertkulvanich & S. Madarasmi

King Mongkut's University of Technology Thonburi
Bangkok, Thailand

ABSTRACT: This paper presents the modelling and simulation of viscous fluid, which is in the form of free surface flow. The simulation uses a fully implicit Eulerian technique with constant and variable viscosity coefficient with the full term of the viscous stress tensor using conventional 3D Navier-Stokes equations. The method exploits a variation principle which enforces complex boundary conditions on the shear stress at the free surface while the fluid discretisation turns into three positive definite linear equations using Marker-and-Cell to govern the fluid equation in each time step. The example in this paper will demonstrate high viscosity phenomena, such as coiling, folding, etc. Also in this paper are both Dirichlet and Neumann complex boundary condition, as well as the interaction with solid objects and the melting process of solid to low viscous fluid, which is controlled by diffusion heating. The result of the simulation will be used in future work on the simulation of the high temperature environment of a liquid metal flow, such as in Gas Metal Arc Welding: a process that cannot be directly observed.

INTRODUCTION

Melting or fusion is a phenomenon of matter that results in the change in phase of a substance from solid to liquid. Its internal energy is increased resulting in a rise of its temperature to the melting point at which the solid molecular bonds break down and the solid liquefies. On the other hand, if the temperature cools down, it will return to a solid state. When a substance is in a liquid state, it is called a fluid. The word, fluid, technically can be interpreted as a liquid or gas.

A gas fills its container completely, whereas a liquid has its own free surface, the shape of which does not depend on its container. In computer graphics, modelling and simulation of phenomena, such as melting, flowing, dripping, etc. has become popular in many areas. There are examples in films or games that make use of exotic materials, such as wet mud, tar, gas, etc. In industrial areas, simulation of a high temperature environment, such as lava or melting metal, can be a reminder to handle with care and to forecast its real behaviour.

To model the flowing of a fluid-like gas or liquid, there are two common properties viz advection and diffusion. Advection is the property that a fluid can convey objects or convey itself. Diffusion is the property that a fluid expands in all directions related to the shear of the substance. Liquid has a higher shear force than gas. To model a gas, the fluid should have a very low viscosity, whereas to model a liquid, the viscosity should be high. If it is high enough, it will express different behaviour and cannot flow freely like water. Instead, it tries to form a shape, such as a coil or fold over upon itself with intricate details.

These behaviours can be generated by a careful coupling to the air. This phenomenon can be used to model the melting of substance from the solid phase to the fluid phase. Considering the melting of a solid to a fluid, a method will be presented to model the formation of a fluid, with high viscosity or low viscosity, together with its interactions with the environment by controlling the viscosity. Metal arc welding is expected to be modelled in the future.

CONTRIBUTIONS

The aim is to make a simulation and animation of fluids in Gas Metal Arc Welding as the final target of the work. Suitable fluid simulation algorithms to simulate a gas are either the Smoothed Particle Hydrodynamics algorithm (SPH) [1] or Computational Fluid Dynamics (CFD). This work used CFD to simulate a high viscous fluid and fluid melting. The focus was on both high and low fluid viscosities and used a conventional high viscous fluid method [2-4]. Unlike former works, which used extensive simulation, here only the fluid of immediate concern was simulated. At the fluid interface, if the fluid passes into a new space, the values for the empty space are extrapolated from the values at the fluid interface.

RELATED WORKS

Since Stam explored how to make fluid simulation using computer graphics with the finite difference method of the 3D Navier-Stokes equations, the *Stable fluids*, the approach developed in fluid simulation papers has been extended to various kinds of fluid phenomena, such as smoke, free surface flows, fire and even big explosions [5]. Foster and Metaxas [6][7] adapted the Marker-and-Cell (MAC) method of Harlow and Welch [8] using 3D Navier-Stokes equations to deal with liquid flows or free surface flows.

Though quite effective, the simulation required small time steps due to the explicit integration. Later, an implicit viscosity solution method was introduced by Stam using a semi-Lagrangian advection scheme [9]. The Navier-Stokes equations are solved independently for each term. This is now used for fluid simulations but initially smoke was simulated. At that time there were no efficient simulations of free surface flow. Later, Carlson et al introduced a model to handle free surface flow with variable viscosity using geometric mean viscosity [10][11]. The modelled viscosity was controlled by a heat diffusion process. Since then, there has been various work concerned with fluids, such as Rasmussen et al with a simulation having an implicit-explicit scheme [5], and Hong et al, who demonstrated discontinuous jumps in viscosity of two-phase fluids [12]. Some information was added to a model in order to make specific fluid properties such as elastic component by Goktekin et al to model a jelly phenomenon [13]. Batty et al modelled viscous fluid with a stress tensor on diffusion terms but only for a constant viscosity term [2].

GOVERNING EQUATIONS

For animation and some simulation, Navier-Stokes equations were chosen because of their simplicity and good explanation especially for high viscous fluid flows. The momentum Equation (1) has four main terms viz advection, diffusion, pressure gradient and external body force. Advection is the self-movement of fluid, while diffusion is the fluid motion which depends on its viscosity, as shown in the form of the viscous stress tensor. The body force is the external force applied to the fluid. There are two fluid conditions which affect the size of the simulation: compressible and incompressible. The incompressible case is covered in Equation (2) noting the divergence of zero. This means the fluid flow into a cell is equal to the fluid flow out. Equation (3) allows for viscosity, with the viscous stress tensor applied for a variable viscosity fluid.

$$\frac{\partial \vec{u}}{\partial t} = -\vec{u}.\nabla\vec{u} + \frac{1}{\rho}\nabla.\tau - \frac{1}{\rho}\nabla p + \vec{\mathrm{g}} \tag{1}$$

$$\nabla.\vec{u} = 0 \tag{2}$$

$$\tau = \left(\eta(\nabla\vec{u} + \nabla\vec{u}^{T})\right) \tag{3}$$

Where $\vec{u}, \vec{\mathrm{g}}$ and p are the cell properties velocity, gravity (the body force) and pressure. ρ, η and τ are the fluid properties density, dynamic viscosity and fluid viscous stress tensor. Equation (3) for constant dynamic viscosity and an incompressible fluid becomes:

$$\nabla.\tau = \nu\nabla^2\vec{u} \quad \text{where } \nu = \frac{\eta}{\rho} \tag{4}$$

The standard approach called operator splitting is a solution for the velocity $\vec{u}$ for each time step [9]. First, omit the pressure term; next, apply the advection, diffusion and body force in order. The advection is solved by using a semi-Lagrangian method, which traces the fluid to determine its next state.

The diffusion term is described by a Poisson equation using backward timing integration that can be solved using many methods, such as Jacobi, Conjugate Gradient and Multigrid. Mostly used here was the preconditioned Conjugate Gradient method (PCG). Lastly, apply the pressure together with Equation (2), which also forms a Poisson's equation; this can be solved as for the diffusion state. The resultant velocity can be used to update the new fluid position, as shown in Figure 1:

$$\vec{u}^{A} = \vec{u}\left(interpolated\left(-\vec{u}^{n} * \Delta t\right)\right)$$

$$\vec{u}^{B} = \vec{u}^{A} + \Delta t\vec{\mathrm{g}}$$

$$\vec{u}^{D} = Linear\_solve\left(\nabla^2\vec{u}^{B}\right)$$

$$\vec{u}^{n+1} = linear\_solve\ \left(\nabla.\vec{u}^{D} = \nabla^2 p\right)$$

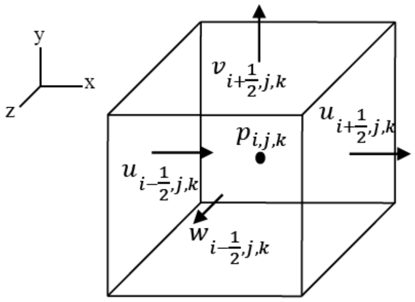
Figure 1: Solving velocity for each time step.

where $\vec{u}^{n}$,$\vec{u}^{n+1}$ and $\vec{u}^{A}$,$\vec{u}^{B}$,$\vec{u}^{D}$are velocity at n^{th} and $(n+1)^{th}$ time step and velocity after advection state, body force state and diffusion state respectively. To reach a solution for variable viscosity with a diffusion term, the Poisson equation is extended to include a viscous stress tensor term as shown in Equation (3).

IMPLEMENTATIONS

Computational fluid dynamic scheme (CFD) was used to simulate a fluid in a three-dimensional simulation space which should provide a good simulation. The space is discretised into a small grid of cell spaces each of which has its own components, i.e. the velocity in each direction and fluid parameters. The simulation method called Marker-and-Cell (MAC) by Harlow and Welch was implemented [8].

The method places the vital components like velocity at the centre of each side of a cell face, while others are placed at the cell centre. The index of cell is at the cell centre and the cell face has the index one half different, depending on the direction of the component, e.g. if the cell has index $i, j,$ and k, the left cell face centre has the index of $i - 1/2, j, k$ and for the bottom face centre is $i, j - 1/2, k$ and so on, as shown in Figure 2. Once the half indices components are used, they are allocated in computer memory, with a shift to a full index. For example, $u_{i-1/2,j,k}$ has the index of $u_{i,j,k}$, $u_{i+1/2,j,k} \Rightarrow u_{i+1,j,k}$ for $v_{i-1/2,j,k} \Rightarrow v_{i,j,k}$ and so on. This means the simulation in each cell requires only that cell's components. This scheme can reduce the velocity jagging with unbiased second-order accuracy [3].



note:

$$\vec{u}_{i,j,k} = \begin{bmatrix} u_{i,j,k} \\ v_{i,j,k} \\ w_{i,j,k} \end{bmatrix}$$

Figure 2: The simulation space grid cell with the position of each component in cell.

The fluid is represented by particles. The cell that has at least one particle is a fluid cell while an empty cell has no particles. Most simulators simulate throughout all the space. To have a fast simulation step, empty cells have zero pressure and velocity. The simulation is only performed on fluid cells. The simulation process for each time step is shown in Figure 3.

Particle collection
Advection process
Diffusion process
Add body force
Calculate Pressure
Update velocity
Moving particles due to cell velocity
Extrapolate the velocity for new empty cell if the particle is in that cell
Update particle position not to exceed the boundary

Figure 3: Simulation process for each time step.

Boundary Condition

Two types of boundary are considered viz a solid boundary and a free surface boundary. The free surface is the surface of a fluid that interfaces to the air. It is straightforward to deal with this surface type. The cell pressure is set to zero due to the viscous stress tensor being zero. For a solid boundary (fluid-solid interface), e.g. a solid wall, there are two types of boundary conditions, i.e. the no-stick condition and the no-slip condition. No-stick condition is used with low viscous or inviscid fluid.

This can be done by setting the velocity component that is normal to solid equal to normal component of the solid velocity $(\vec{u}.\hat{n} = \vec{u}_{solid}\hat{n})$ which, of course, simplifies to $\vec{u}_{\mathbf{solid}} = \mathbf{0}$. For the no-slip condition, the viscous fluid has a low speed of continuous flow since the fluid *sticks* to the solid at the interface while slippage happens in the upper layers. In this case we have [3]:

$$\vec{u}.\hat{n} = \vec{u}_{solid}.\hat{n} \qquad \text{for the upper layer and}$$

$$(\tau\hat{n}) \times \hat{n} = 0 \qquad \text{for solid-fluid interface.}$$

RESULTS

To conduct the test, the focus was on the fluid behaviour with varying viscous coefficient; first, there was a free fall test with low and high viscosity and an impact with a solid boundary with both no-stick and no-slip boundary condition. The results are shown in Figure 4.

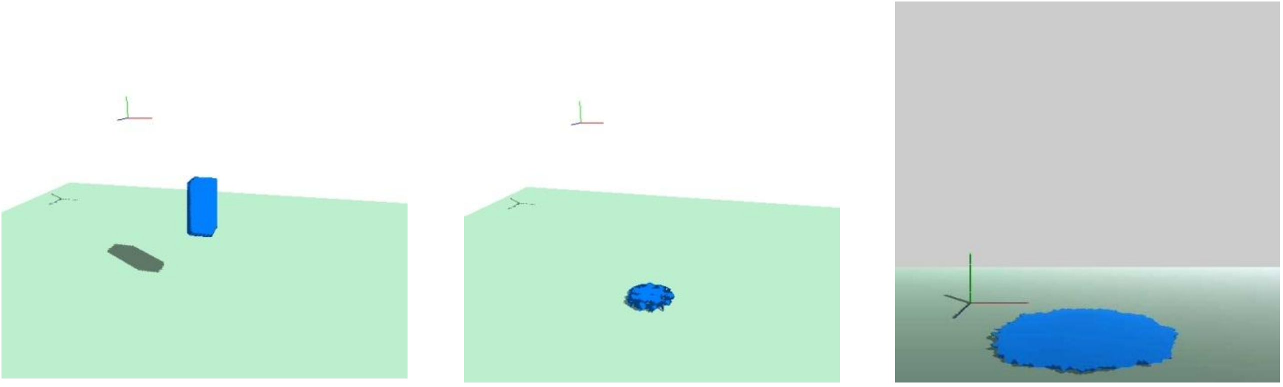


Figure 4: The simulation result with a high and low viscous fluid. The fluid starts with a square cylinder shape (l-r): a), b) and c) are the final results for a high and low viscous fluid.

The simulation was rendered using OpenGL as a graphics tool. Shown above, the shadow is cast from one light source. The algorithm is based on the perspective projection of light to an object to render the shadow on the floor. The dark area is produced by comparing the light to object distance with the object depth field, which is stored in the OpenGL depth buffer (z-buffer). If the light to object distance is greater than object depth value, the shadow is shaded with dark colour. This is implemented using the OpenGL shading language and also the Phong's shading scheme, which is normally used for light shading. The surfaces are constructed using the Marching Cubes algorithm, in which the cell is discretised to a half size of the simulation grid cell [14]. Particles forming the surface are smoothed only with the neighbouring cell instead of with all particles to reduce the surface creation time, which has little effect on the created surface.

CONCLUSIONS AND FUTURE WORK

This paper presents the simulations of viscous fluid, such as water, honey, caramel, etc. The fluid was simulated using both high and low viscous coefficients. The results show fluid behaviour corresponding to the viscous coefficient. The simulation time is satisfactory. Nonetheless, the simulation is determined only to cell level and a cell may be empty. This can be compensated for by reducing the cell size but the simulation time would increase.

For future work, this fluid simulation will be extended to cover different types of fluid interface; the transformation of a substance to a fluid (multi-phase fluid), which can be modelled as a very high viscous fluid. All will be used for the final target, which is the simulation and animation of Gas Metal Arc Welding.

REFERENCES

1. Suriyong, L. and Sanan, S., Interactive physically-based modelling for gaseous object using smoothed particle hydrodynamics. *Proc. Inter. Multi-Conf. of Engineers and Computer Scientists*, Hong Kong, 208-212 (2006).
2. Batty, C. and Bridson, R., Accurate viscous free surfaces for buckling, coiling and rotating liquids. *Proc. ACM/Eurographics Symposium on Computer Animation* (2008).
3. Bridson, R., *Fluid Simulation for Computer Graphics*. A K Peters, Ltd., 107-124 (2009).
4. Williams, B., Fluid Surface Reconstruction from Particles, Degree, Master of Science - MSc. Program, Computer Science, The University of British Columbia (2008).
5. Rassmusen, N., Enright, D., Nguyen, D., Marino, S., Sumner, N., Geiger, W., Hoon, S. and Fedkiw, R., Directable photorealistic liquids. *Proc. 2004 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, 193-202 (2004).
6. Foster, N. and Metaxas, D., Realistic animation of liquids. *J. of Graphical Models Image Processing*, 58, **5**, 471-483 (1996).
7. Foster, N. and Fedkiw, R., Practical animation of liquids. *Proc. 28th Annual Conf. on Computer Graphics and Interactive Techniques*, SIGGRAPH, ACM Press, 23-30 (2001).

8. Harlow, F. and Welch, J., Numerical calculation of time-dependent viscous incompressible flow of fluid with a free surface. *J. of Physics of Fluids*, 8, **12**, 2182-2189 (1965).
9. Stam, J., Stable fluids. *Proc. Computer Graphics*, Annual Conference Series, 121-128 (1999).
10. Carlson, T.M., Peter, J.M., Brooks, V.H. and Turk, G., Melting and flowing. *Proc. ACM SIGGRAPH Symposium on Computer Animation*, San Antonio, Texas, July 21-22, 167-174 (2002).
11. Carlson, M., Mucha, P.J. and Turk, G., Rigid fluid: Animating the interplay between rigid bodies and fluid. *Proc. ACM SIGGRAPH*, 23, **3**, 337-384 (2004).
12. Hong, J.M. and Kim, C.H., Discontinuous fluids. *Proc. of ACM SIGGRAPH 2005*, 24, **3**, 915-920 (2005).
13. Goktekin, G.T., Bargteil, A.W. and O'Brien, J.F, A method for animating viscoelastic fluids. *Proc. Computer Graphics*, SIGGRAPH Annual Conference Series, 23, **3**, 464-468, (2004).
14. Brouke, P., Polygonising a scalar field (1994), 10 October 2010, www.local.wasp.uwa.edu.au/~pbourke/ geometry/polygonise
15. Lorensen, W.E. and Cline, H.E., Marching cubes: a high resolution 3D surface construction algorithm. *Proc. Computer Graphics*, SIGGRAPH, 21, **4**, 163-169 (1987).