Computing of Volcanic Plume with Mesh-free Method

Zhixuan Cao

August 30, 2016

1 Abstract

Smoothed particle hydrodynamics(SPH) has several advantages over mesh-based methods in modeling multiple phase free boundary flows. We develop a new volcano plume model employing two phases fluid dynamics governing equations, three-dimensional coordinates, SPH method and message passing interface (MPI) standard parallel computing. $SPH-\varepsilon$ turbulence model which was originally developed for incompressible flow is adopted and extended for compressible flow to capture sub-particle scale turbulent entrainment. Corrected formulism of SPH is used to handle tensile instability and deficiency of particle distribution near the boundaries. The model is first verified by a JPUE simulation where velocity and concentration distribution along axis and acrross cross section are compared with experiments. Then a typical volcano plume is simulated with this model. Top height of the plume and profiles of several local variables are compared with previous simulation results of existing plume models. Analysis on the plume evolution process also shows that pressure at the outer shear margin is higher than center of the plume and surrounding atmosphere.

2 Introduction

The dynamics of explosive eruption clouds has been the central issue of volcanology science for a long time. Primary hazards associated with explosive "grey volcanoes" include pyroclastic density currents (flows and surges), the widespread deposition of airfall tephra, and the threats posed by volcanic ash. In addition, there are also other secondary hazards associated with volcanoes[28]. Simulation of all possible hazzards with one model is difficult due to diversity among these hazzards. Our concerns in current work is the risks that volcanic ash posesed on aircrafts and communities. During volcanic eruptions, volcanic ash transport and dispersion models (VATDs) are used to forecast the location and movement of ash clouds over hours to days. Those VATDs models use input parameters, called "eruption source parameters", such as plume height H, mass eruption rate $\dot{M}$, duration D, and the mass fraction distribution of erupted debris finer than about 4Φ (or $63\mu m$), which can remain in the cloud for many hours or days. Observational data on the value of such parameters are usually unavailable in the first minutes or hours after an eruption is detected. Moreover, observed plume height may change during an eruption, requiring rapid assignment of new parameters. Usually, plumes models are used to provide sources terms for VATDs. This paper reports a new 3D volcano plume model target at providing more accurate source parameters for VATDs.

The basic physics behind the ascent and collapse of volcanic columns has been understood for more than half a century[40, 49, 62]. Several volcano plume models have been developed in the past few decades. Specific aspects of a plume have been studied by one dimensional(1D) plume models which describe the steady state solution of a plume under idealized boundary conditions. The effects of magma type and vent conditions [66, 64], atmospheric conditions[50, 63, 3], external surface water[23], thermal disequilibrium, and particle fallout[65] have been assessed in increasing details through 1D numerical simulations. In these 1D models, the entrainment of air is evaluated based two coefficients: entrainment coefficient for the verical plume and the entrainment coefficient that describes the effect of wind. Different 1D models adopt different entrainment coefficients based on specific formulation or calibration against well-documented cases studies. The feedback from plume to atmosphere is ignored in 1D models. Even though determination of essential parameters such as the entrainment rate are not based on first principles, such simple models nevertheless allow us to investigate the importance of physical mechanisms in a volcanic plume. In addition, these simplified models (eg. [30]) requires little computational sources and can run on standard personal computers or on web sites in very short time. As a result, software ([4, 31, 61] for volcano plume development, [5, 13] for ash dispersion and transportation) based on 1D model are still pretty popular. While these 1D models can generate well-matched results with three dimensional(3D) models for weak plumes, much greater variability is observed for strong plume scenarios, especially for local variables[10]. In addition, the need for greater skills in associated hazards forecasts with 1D models requires 2D and 3D models.

The development of two dimensional(2D) and three dimensional(3D), time-dependent, and multiphase numerical models for volcano plume have provided new explanations for many features of explosive volcanism. One of these 3D models is PDAC [41] which is a non-equilibrium, multiphase, 3D compressible flow model. Conservation equations for each phase is solved separately with finite volume method. And a parallel computing version of PDAC was developed recently [43]. Advanced numerical techniques(second order scheme and semi-explicit time upgrading) was also adopted to improve the accuracy of PDAC[6]. Suzuki[54] developed a three-dimensional time-dependent fluid dynamics model (which is refered as SK-3D) that can reproduce the entrainment process of eruption clouds with relative simpler physics but higher order of accuracy and high spatial resolution, which is applicable to time-dependent phenomena in actual volcanological situations. SK-3D is able to correctly reproduces the quantitative features of turbulent mixing in and around the JPUE(jet or plume which is ejected from a nozzle into a uniform environment) under the ideal experimental conditions. A serials of simulations based on SK-3D was conducted, including determination of the entrainment coefficient of eruption columns as a function of height[52], study the effect of wind feild on entrainment coefficient[53], establishment of the relationship between the observable quantities of the eruption clouds and the eruption conditions at the vent (previously, such relationship is based on 1D models and greatly depends on empirical constants)[51], investigation of the effect of the intensity of turbulence in the umbrella cloud on dispersion and sedimentation of tephra[24]. While the SK-3D foucses on accurately capturing the entainment caused by turbulent mixture with higher resolution and numerical method of higher order, PADC takes the disequilibrium between different phases into acount and hence is a ture multiple phase model, another model ATHAM (Active Tracer High-Resolution Atmospheric Model) [42] foucses more on microphysics within the plume. As pyroclastic flow is not the initial concern of ATHAM, immediately dynamics and thermodynamics equilibrium is assumed in ATHAM (justified for Plinian eruption). Number of governing equations reduced greatly as well as computational cost. The dynamic core of ATHAM solves the compressible Euler equations for momentum, pressure and temperature of the gas particle mixture[42]. The subgrid turbulence closure scheme differentiates between the horizontal and vertical directions [18] to capure turbulent mixing. The cloud microphysics predicts the mass of hydrometeors in liquid and ice phase [17]. An extension was added to simulate particle aggregation in 2006[57, 55]. Additional modules include radiation [26], gas phase chemistry [58] and gas scavenging by hydrometeors[56] were also added. However, the resolution of ATHAM is still pretty coarse. For example, 127×127 grid points on a domain of $200 \times 50km$, with the finest resolution of $100m$ in the center of the model[18]. In a more recent simulation, a domain of $100 \times 100 \times 30km$ is covered with $194 \times 194 \times 13$ grid points.The maximum horizontal and vertical resolution of $50m$ was centred at the volcano, stretching to $1.0km$ vertically and $1.9km$ horizontally at the model boundaries[59]. Besides strengthening their strengthness(ATHAM has been adding more and more microphysics, PDAC was been extended from 2 phase to 3 phases), these models are learning from each other. PDAC is trying to include effect of microphysics into the model while ATHAM wants to extend its ability to modeling pyroclastic flow. Both are using finer and finer resolution.

To summarize, each model has its own foucs based on problem of interest. Different simplification was made based on their scope. Improvements have been making since their birth. Accuracy (depends on comprehensiveness of the model, resolution of discretization, numerical error, and order of accuracy) and simulation time (depends on number of governing equations, resolution, numerical methods and parallel techniques) are always a couple of conflict considerations in 3D plume simulations.

All of these existing 3D models are based on mesh-based method (Eulerian Method). And there is no 3D plume model based on meshfress method (Lagrange Method) yet. Actually, Lagrangian methods have sevaral features that are suitable for volcano plume simulation. Among such Lagrangian methods, Smoothed Particle Hydrodynamics [16, 29], even though relatively newer compared to well designed mesh based methods such as Finite Difference, has shown good agreeemnt with experiments by many applications in fluid dynamic area. And it currently is, by far the most widely used scheme. Specifically, the reason we choose SPH as the numerical method for volcanic plume simulation are as following:

- For mesh based method, either interface tracking (Lagrangian) [**?** **?** **?**] or interface capturing (Eulerian) [**?** **?** **?** **?**] method are used to reconstruct the flow interface of free boundary flow. High computational cost, a tendency to form numerical instabilities and the inability to track complex topological changes are the significant drawbacks of Lagrangian techniques[**?** **?** **?**]. For interface capturing (Eulerian) method, the surrendering of surface detail before the phase transport calculation means that interface reconstruction is required between timesteps to recover the interface information, which need additional numerical effort [**?** **?** **?**]. Being able to adaptively adjust the discretization and automatically construct the interface, SPH does not require additional numerical effort for interface construction and therefor more suitable for volcano plume simulation.

- Advection term in the Navier–Stokes equations does not appear explicitly in discretized formula of SPH (as illustrated in equation(12) - (15)). So advection term will be treated exactly with SPH [33].

- It is easy to include various physics effect (like self gravity, radiative cooling and chemical reaction) into the model. It does not require a major overhaul and re-tooling every time new physics is introduced[35]. This implies easy extention of the model. As discussed in the previous paragraph, existing 3D plume models focus on one or several specific aspects of plume and have been extended to be more comprehensive by introducing more physics. Easy-extensibility of SPH method greatly facilitate possible future extension of current model.
- Another advantage of SPH is that with more than one material, each described by its own set of particles, interface problems are often trivial for SPH but difficult for finite difference schemes. So multiple phase flow can be easily handled by SPH. Adding of new phase to the model also does not require a major overhaul and re-tooling.

As volcanic plume are in nature multiple phase and without pre-defined boundary in the atmosphere, SPH is a suitable numerical method for plume modeling. An additional motivation for using SPH is that our current model is far away from comprehensive and there is a tendency of including new physics (phase) to these existing models. The easy extension feature of SPH perfectly satisfy this demand.

As SPH is still a relative new method in computational fluid dynamics, implementation of SPH for compressible multiphase flow is scarce. Several remedies of classical SPH is adotped and extended in our implementation. SPH is usually adopted in simulating of free surface flow, less attention was paid to pressure outlet boundary conditions. For our model, pressure outlet boundary is required. We impose pressure boundary condition by adding extra layers of static ghost particles. Additional constrain on time step is used to avoid growing up of numerical fluctuations near the pressure boundary. Entrainment due to turbulence is one of critical factors in plume development. There are several turbulence models proposed for SPH method [60, 20]. We adopt a LANS type turbulence model[37], which is computationally cheaper, and extend it for compressible flows. Corrected formulism of SPH is adopted to bypass the well-known tensile instability issues of SPH. Simulation of volcano plume with acceptable accuracy at given time window requires fine resolutions (very high particle counts) that cannot been accomplished without parallel computing. What's more, imposing of some types of boundary conditions (such as realistic wind field, eruption boundary condition) requires dynamically adding and removing of particles during simulation. Effieicent data management scheme and domain adjusting strategies are used to fulfil these requirements. Finally the message passing interface (MPI) standard is implemented to parallelize the computation.

The organization of this paper is as following: The physics model of the plume is first presented in section 3 and leads to a complete mathmetical description of the volcano plume (governing equations and boundary conditions). In section 4 we will introduce the numerical tool, SPH method, briefly. Both the fundemental discretization formulism and techniques that used to handle specific issues involved in plume modelling will be discussed. Validation and verifycation with numerical tests is presented in section 5. In section 6, a discussion on future work is given following a brief summary of this paper.

3 Physical Model

During an explosive eruption, a volcanic jet shoots out from a vent at a speed of several tens to more than 150 meters per second, driven by expanding gas. The jet is initially denser than the surrounding atmosphere and begins to decelerate through negative buoyancy and turbulent interaction with surrounding air. Along jet margins these processes generate cauliflower-like vortices that entrain air and heat it, reducing the bulk density of the entire jet, in many cases, to less than that of the surrounding atmosphere. Once it becomes buoyant, such a jet develops into a plinian or subplinian plume, rising kilometers to tens of kilometers until its heat is diluted enough that buoyancy is lost. Jets that lose their momentum before becoming buoyant collapse back onto the Earth's surface and transform into pyroclastic flows, surges and ignimbrites.

We adopt the model developed by Suzuki[54], which was designed to describe the injection of a mixture of solid pyroclasts and volcanic gas from a circular vent above a flat surface of the earth in a stationary atmosphere. Another critical element, the wind field is not considered in this model yet. Introducing of wind field will need imposing of pressure boundary condition in a dynamic way. Other factors, such as aggregation and phase changes of volcanic gas, will also come to play critical roles under particular eruption conditions. Capturing of these processes need a more comprehensive model. At present stage, we are more interested in the most basic (and the most critical) aspects of volcano plume and therefor devote our effort to this relative simple model. It is worthwhile to mention here that because SPH is adopted, adding of these physics into our model would require much less work in programming compared to mesh based methods.

3.1 Governing equations[54]

Based on Navier-Stokes equations, the following assumptions are made:

- Molecular viscosity and physics heat conduct is neglected since turbulent exchange coefficients are dominant.
- Erupted material consist of solid with different size and mixture of gas (various constituent) is assumed to be well mixed and behave like a single phase fluid (phase 2). Air (also a mixture of different gas) is assumed to be another phase (phase 1).
- Immediate thermodynamics equilibrium is assumed so that no separate energy equation is needed for each phase. As a result, there is only one energy equation for both phases (heat exchange term between different phases will not show up under this assumption).
- Immediate dynamics equilibrium is assumed so that no separate momentum equation is needed for each phase. As a result, there is only one vector moment equation for both phases (drag force term will not show up with this assumption).
- Because of above assumptions, all other micro-physics process (like phase change of H_2O, aggregation, decomposition, absorption of gas on the surface of solid, solution of gas into the liquid,) and chemical process will not be considered in this model.

The governing equations for this model is:

$$\begin{aligned}
\frac{\partial\rho}{\partial t}+\nabla\cdot(\rho\mathbf{v}) &= 0 \quad (1)\\
\frac{\partial\rho\xi}{\partial t}+\nabla\cdot(\rho\xi\mathbf{v}) &= 0 \quad (2)\\
\frac{\partial\rho\mathbf{v}}{\partial t}+\nabla\cdot(\rho\mathbf{v}\mathbf{v}+p\mathbf{I}) &= \rho\mathbf{g} \quad (3)\\
\frac{\partial\rho E}{\partial t}+\nabla\cdot[(\rho E+p)\mathbf{v}] &= \rho\mathbf{g}\cdot\mathbf{v} \quad (4)
\end{aligned}$$

ξ is the mass fraction of ejected material. $E = e + K$ is total energy which is summation of kinetic energy K and internal energy e. An additional equation is required to close the system. In this model, the equation for closing the system is the EOS for ideal gas:

$$p = (\gamma_m - 1)\rho e \quad (5)$$

Where

$$\begin{aligned}
\gamma_m &= R_m/C_{vm} + 1 \quad (6)\\
R_m &= n_g R_g + n_a R_a \quad (7)\\
C_{vm} &= n_s C_{vs} + n_g C_{vg} + n_a C_{va} \quad (8)\\
n_a &= 1 - \xi \quad (9)\\
n_g &= \xi n_{g0} \quad (10)\\
n_s &= \xi - n_g \quad (11)
\end{aligned}$$

Where, C_v is specific heat with constant volume, n is mass fraction, R is gas constant. The subscript m represents mixture of ejected material and air, s represents solid portion in ejected material, g represents gas portion in the ejected material and a represents air. By deducting Kinetic energy from energy equation, subtracting mass conservation from momentum equation, combining transient term and advection term into material derivative term, the governing equation is put into the final formulism, in which advection term does not show up explicitly.

$$\begin{aligned}
\frac{D\rho}{Dt}+\rho\nabla\cdot\mathbf{v} &= 0 \quad (12)\\
\frac{D\rho\xi}{Dt}+\rho\xi\nabla\cdot\mathbf{v} &= 0 \quad (13)\\
\frac{D\mathbf{v}}{Dt}+\frac{\nabla P}{\rho} &= \mathbf{g} \quad (14)\\
\frac{De}{Dt}+\frac{P\nabla\cdot\mathbf{v}}{\rho} &= 0 \quad (15)
\end{aligned}$$

3.2 Boundary conditions

In current model the initial domain is a box. The boundaries are categorized into eruption vent (a circle area at the center of the bottom of the box), wall boundary (box bottom), pressure boundary (other faces of the box).

3.2.1 Eruption vent

At the vent, temperature of erupted material T, eruption velocity $\mathbf{v}$, the mass fraction of gas in erupted material n_{g0} and mass discharge rate $\dot{M}$ is given. The pressure of erupted material p is assumed to be the same as ambient pressure for pressure-balanced eruption. The radius of vent is determined from ρ, $\dot{M}$ and $\mathbf{v}$. To write them in terms of primitive variables.

$$\rho = const = p/(R_m T) \tag{16}$$

$$\xi = const = 1 \tag{17}$$

$$\mathbf{v} = const = \{u, v, w\}^T \tag{18}$$

$$\frac{\partial e}{\partial n} = \dot{M}e/A \tag{19}$$

Where A is the area of the vent.

3.2.2 Non-slip wall boundary

Velocity is zero for non-slip wall boundary. If assume the boundary to be adiabatic, heat flux should be zero on the bounday. The flux of mass should also be zero. As a result, internal energy flux, which consists of heat flux and energy flux carried by mass flux, will vanish on the wall boundary. So the wall boundary condition in terms of primitive variables are:

$$\frac{\partial \rho}{\partial n} = const = 0 \tag{20}$$

$$\frac{\partial \xi}{\partial n} = const = 0 \tag{21}$$

$$\mathbf{v} = const = \{0, 0, 0\}^T \tag{22}$$

$$\frac{\partial e}{\partial n} = 0 \tag{23}$$

3.2.3 Open outlet pressure boundary condition

The pressure of the surrounding atmosphere is given. Except for the pressure, boundary condtions for density, velocity, and energy should depends on the solution. As we ignore the viscosity, the shear stress is gone and norm stress (whose magnitude euquals to pressure) will balance with ambient pressure.

$$p = p_a(h) \tag{24}$$

4 SPH Method

SPH is a meshfree Lagragian method. In SPH, the domain is discretized as a set of particles or discretization points and the position of each particle (discretization points) is updated at every time step. Approximation of all field variables (velocity, density and pressure, ect.) is obtained by interpolation based on discretization points. The physical laws (such as conservation laws of mass, momentum and energy) written in the form of partial differential equations need to be transformed into the Lagrangian particle formalism of SPH. Using a kernel function that provides the weighted estimation of the field variables at any point, the integral equations are evaluated as sums over neighbor particles. Thus, physical properties (density, velocity, internal energy, ect.) associated to the particle are updated based on its neighbors. The kernel function has a compact support determined by smoothing length of each particle. There are several review papers [36, 33, 46, 47, 34], giving a pretty comprehensive view over SPH, we will only refer here to the representation of the constitutive equations in SPH notation.

4.1 Fundametal principle

There are several routines for discretizing governing equations (partitial differential equations or ordernary differential equations) with SPH. There are slight difference between discretized formulism obtained by different routines. We present here one of them following JJ Monaghan[36, 33]. The start point of approximating a function with SPH is one translation property of Dirac function.

$$A(x) = \int_{-\infty}^{\infty} A(\mathbf{x}\prime)\delta(\mathbf{x}\prime - \mathbf{x})d\mathbf{x}\prime \tag{25}$$

The Dirac function in equation (25) can be approximated by a weighting funtion $w(\mathbf{x}-\mathbf{x}\prime,h)$ (or $w(\mathbf{x}\prime-\mathbf{x},h)$) which will tend to a Dirac function when $h \rightarrow 0$:

$$\lim_{h\rightarrow 0} w(\mathbf{x}\prime-\mathbf{x},h) = \delta(\mathbf{x}\prime-\mathbf{x}) \tag{26}$$

The function $A(\mathbf{x})$ then can be approximated by:

$$A(\mathbf{x}) \approx < A(\mathbf{x}) > = \int_{\Omega} A(\mathbf{x}\prime)w(\mathbf{x}-\mathbf{x}\prime,h)d\mathbf{x}\prime + O(h^2) \tag{27}$$

Where h is smoothing length, determining the interaction distance. According to equation (27), the order of accuracy of SPH method is second order. However, in practice, second order can not be achieved because there is no guarantee on the symmetricity of particle distribution in real simulation [46]. Recall that $d\mathbf{x}\prime = \frac{dm\prime}{\rho\prime}$, the integration equation (27) can be approximated by summation and lead to an approximation of the function at a particle (idiscretization point) a:

$$A(\mathbf{x}) \approx \sum_b m_b \frac{A_b}{\rho_b} w(\mathbf{x}-\mathbf{x}_b,h) \tag{28}$$

Where the summation is over all the particles b within the region of compact support of the weighting function. Gradient terms may be straightforwardly calculated by taking the derivative of (28), giving

$$\begin{aligned} \nabla A(\mathbf{x}) &= \frac{\partial}{\partial \mathbf{x}} \int_{\Omega} A(\mathbf{x}\prime)w(\mathbf{x}-\mathbf{x}\prime,h)d\mathbf{x}\prime + O(h^2) \\ &\approx \sum_b m_b \frac{A_b}{\rho_b} \nabla w(\mathbf{x}-\mathbf{x}_b,h) \end{aligned} \tag{29}$$

For vector quantities the expressions are similar, simply replacing A with $\mathbf{A}$ in (28), giving

$$\mathbf{A}(\mathbf{x}) \approx \sum_b m_b \frac{\mathbf{A}_b}{\rho_b} w(\mathbf{x}-\mathbf{x}_b,h) \tag{30}$$

$$\nabla \cdot \mathbf{A}(\mathbf{x}) \approx \sum_b m_b \frac{\mathbf{A}_b}{\rho_b} \cdot \nabla w(\mathbf{x}-\mathbf{x}_b,h) \tag{31}$$

$$\nabla \times \mathbf{A}(\mathbf{x}) \approx \sum_b m_b \frac{\mathbf{A}_b}{\rho_b} \times \nabla w(\mathbf{x}-\mathbf{x}_b,h) \tag{32}$$

$$\nabla^j \mathbf{A}^i(\mathbf{x}) \approx \sum_b m_b \frac{\mathbf{A}_b^i}{\rho_b} \nabla^j w(\mathbf{x}-\mathbf{x}_b,h) \tag{33}$$

4.2 Weighting function

As described in previous section, the weighting function (or kernel function) is used to replace Dirac function and should satisfy equation (26). So it can be viewed as approximation form of Dirac function and satisfy normalization condition:

$$\int w(\mathbf{x}-\mathbf{x}\prime,h)d\mathbf{x}\prime = 1 \tag{34}$$

Besides normalization, weighting function of particle a has to be symmetric with respect to a to ensure that neighbor particles locate at the same distance away will contribute equally to SPH summation (equation 35). Weighting function also need to satisfy conditions such as positivity and compact support. In addition, kernel function must be monotonically decreasing with the distance between particles.

$$w(\mathbf{x}-\mathbf{x}\prime,h) = w(\mathbf{x}\prime-\mathbf{x},h) \tag{35}$$

There is a wide variety of possible weighting functions, such as spline functions (with different orders) and Gaussian function. Generally, the accuracy increases with the order of the polynomials of the kernel function, but the computational time also increases as number of interactions increase. We are adopting a truncated Gaussian function as the weighting function in our simulation.

$$w(\mathbf{x}-\mathbf{x}_b) = \begin{cases} \frac{1}{(h\sqrt{\pi})^d} exp[-(\frac{\mathbf{x}-\mathbf{x}\prime}{h})^2] & |\mathbf{x}-\mathbf{x}\prime| \leq 3h \\ 0 & Otherwise \end{cases} \tag{36}$$

Where d is number of dimensions. The derivative of weighting function:

$$\nabla w(\mathbf{x}-\mathbf{x}_b) = \begin{cases} -2(\frac{\mathbf{x}-\mathbf{x}\prime}{h^2})\frac{1}{(h\sqrt{\pi})^d}exp[-(\frac{\mathbf{x}-\mathbf{x}\prime}{h})^2] & |\mathbf{x}-\mathbf{x}\prime| \leq 3h \\ 0 & Otherwise \end{cases} \tag{37}$$

4.3 Discretization of governing equations

The basic interpolants given in equation (28 - 33) provide a general way of obtaining SPH expressions of governing equations. The problem is that using these expressions 'as is' in general leads to quite poor gradient estimates. Various tricks are used to conserve linear and angular momentum and thermal energy [36]. We only refer here one of these possible discretizations of compressible Euler equations with SPH:

$$<\rho_a>=\sum m_b w_{ab}(h_a) \tag{38}$$

$$<\frac{d\mathbf{v}_a}{dt}>=\sum_b m_b(\frac{p_b}{\rho_b^2}+\frac{p_a}{\rho_a^2}+\Pi_{ab})\nabla_a w_{ab}(h)+\mathbf{g} \tag{39}$$

$$<\frac{de_a}{dt}>=0.5\sum_b m_b\mathbf{v}_{ab}(\frac{p_b}{\rho_b^2}+\frac{p_a}{\rho_a^2}+\Pi_{ab})\cdot\nabla_a w_{ab}(h) \tag{40}$$

Where, $\mathbf{v}_{ab}=\mathbf{v}_a-\mathbf{v}_b$. Π is artificial viscosity term, will be discussed in section 4.4. However, these discretized formulations will not guarantee conservation of entropy. We can obtain conservation of entropy by giving up conservation of mass or thermal energy adopting alternative discretized formulations [36]. As a Lagrangian method, particle position will also been update at every time step.

$$\frac{d\mathbf{x}_a}{dt}=\mathbf{v}_a \tag{41}$$

Adding new physics and new phases into the model is trivial in terms of discretization. For example, add of new source (or sink) into equation (38), add a drag force into equation (39) and a heat exchange term into equation (40) will lead to the new discretization form:

$$<\rho_a>=\sum m_b w_{ab}(h_a)+\dot{\rho}(\mathbf{x},t) \tag{42}$$

$$<\frac{d\mathbf{v}_a}{dt}>=\sum_b m_b(\frac{p_b}{\rho_b^2}+\frac{p_a}{\rho_a^2}+\Pi_{ab})\nabla_a w_{ab}(h)+\mathbf{g}+D\sum_b m_b\frac{\mathbf{v}_b-\mathbf{v}_a}{\rho_b} \tag{43}$$

$$<\frac{de_a}{dt}>=0.5\sum_b m_b\mathbf{v}_{ab}(\frac{p_b}{\rho_b^2}+\frac{p_a}{\rho_a^2}+\Pi_{ab})\cdot\nabla_a w_{ab}(h)+\sum_b\frac{m_b}{\rho_b}(\kappa_a+\kappa_b)\frac{(T_a-T_b)}{\mathbf{r}_a-\mathbf{r}_b}w_{ab}(h) \tag{44}$$

Where the source term $\dot{\rho}$ can be a "sink" of erupted vapor due to phase change which is considered in ATHAM[42]. D is drag force coefficient. κ is heat conduction coefficient. Other physics can be added easily in the same way. Adding of these new terms will lead to modification at several lines in the source code. The drag force term should show up only when dynamics dis-equilibrium between different phases is considered. In that case, each phase will need one set of Navier-Strokes type of governing equations. Adding of new phases into SPH code will only need add several new lines for the new phase besides adding interaction terms.

4.4 Artificial viscosity

In classic SPH, shock waves are handled by introducing artificial viscosity to smear out discontinuities. Π, in discretized momentum equation and energy equation (39) (40), represents artificial viscosity term which is essentially a second derivative term. As in the case of first derivatives, second derivatives can be estimated by differentiating a SPH interpolant twice. However, such formulism has two disadvantages: First, it is very sensitive to particle disorder. Second, the second derivative of the kernel can change sign and will lead to unphysical representation (viscous dissipation will decrease the entropy). One of the most commonly used form of artificial viscosity is:

$$\Pi_{ab}=-\frac{\nu}{\bar{\rho}_{ab}}\frac{\mathbf{v}_{ab}\cdot\mathbf{x}_{ab}}{\mathbf{x}_{ab}^2+(\eta h)^2} \tag{45}$$

Absolute(kinematic) viscous coefficient ν is defined as:

$$\nu=\alpha\bar{h}_{ab}\bar{c}_{ab} \tag{46}$$

Where

$$\bar{c}_{ab} = \frac{c_a + c_b}{2} \tag{47}$$

$$\bar{\rho}_{ab} = \frac{\rho_a + \rho_b}{2} \tag{48}$$

$$\mathbf{v}_{ab} = \mathbf{v}_a - \mathbf{v}_b \tag{49}$$

$$\mathbf{x}_{ab} = \mathbf{x}_a - \mathbf{x}_b \tag{50}$$

The artificial viscosity term Π_{ab} is a Galilean invariant and vanishes for rigid rotation. It will produce a repulsive force between two particles when they are approaching each other and an attractive force when they are receding from each other.

The SPH viscosity can be related to a continuum viscosity by converting the summation to integrals [33]. It has been shown that the shear viscosity coefficient $\eta = \frac{\rho\alpha hc}{8}$ and the bulk viscosity coefficient $\zeta = \frac{5\eta}{3}$ for two dimensional and $\eta = \frac{\rho\alpha hc}{10}$, $\zeta = \frac{5\eta}{3}$ for three dimensional. An extra term was added to ν considering aspects of the dissipative term in shock solutions based on Riemann solvers and lead to:

$$\Pi_{ab} = \begin{cases} \dfrac{-\alpha\mu_{ab}\bar{c}_{ab} + \beta\mu_{ab}^2}{\bar{\rho}_{ab}} & \mathbf{v}_{ab} \cdot \mathbf{x}_{ab} < 0 \\ 0 & \mathbf{v}_{ab} \cdot \mathbf{x}_{ab} > 0 \end{cases} \tag{51}$$

Where

$$\mu_{ab} = \frac{h\mathbf{v}_{ab} \cdot \mathbf{x}_{ab}}{\mathbf{x}_{ab}^2 + (\eta h)^2} \tag{52}$$

α and β are free parameters that must be adjusted for each case. $\alpha = 1$ and $\beta = 2$ are recommended by Monaghan for best results. In our simulation, these two parameters are calibrated to $\alpha = 0.3$ and $\beta = 0.6$. η is usually taken as 0.1 for preventing singularities when $\mathbf{x}_{ab} = 0$.

4.5 Timestep

The physical properties (velocity, density, position and pressure) change every time step due to particle interactions. We have to consider the Courant condition, which is in spirit similar to the Courant condition for the grid-based Lagrangian methods, to determine the amount of the time step Δt.

$$\Delta t = CFL \min_a \left\{ \frac{[\frac{m_a}{\rho_a}]^{\frac{1}{d}}}{c_a} \right\} \tag{53}$$

Where c_a is sound speed of at particle a, d is number of dimensions. First order Euler time integration, with $CFL = 0.2$ has been used in three dimensional calculations.

4.6 Tensile instability and corrected derivatives

The classical SPH method suffered from tensile instability and boundary deficiency. To address these difficulties, Chen [8] proposed a corrective SPH method. For 1D case, by employing a Taylor expansion for $A(x)$ about $\mathbf{x}_a$, multiplying both side by kernel function and then do intergration over the domain, gives

$$\int_\Omega A(x)w(x - x_a, h)dx = A_a \int_\Omega w(x - x_a, h)dx + A_{xa} \int_\Omega (x - x_a)w(x - x_a, h)dx + ... \tag{54}$$

Ignore derivative terms and write the intergral in particle approximation form:

$$A_a = \frac{\sum_b m_b \dfrac{A_b}{\rho_b} w(x - x_b, h)}{\sum_b m_b \dfrac{1}{\rho_b} w(x - x_b, h)} \tag{55}$$

Equation 55 implies that $\sum_b m_b \dfrac{1}{\rho_b} w(x - x_b, h) = 1$, which can be viewd as the approximation form of equation 34. The first order derivative term can be obtained in a similar way and lead to:

$$\nabla A_a = \frac{\sum_b m_b \dfrac{A_b}{\rho_b} \nabla_a w(x - x_b, h)}{\sum_b m_b \dfrac{x_b - x_a}{\rho_b} \nabla_a w(x - x_b, h)} \tag{56}$$

For higher dimesions problem, expression for function approximation are exactly the same as equation (55) even though the derivation is different. The first order derivative can be solved explicitly or numerically from:

$$M_{\alpha\beta a} A_{\alpha a} = F_{\beta a} \tag{57}$$

where

$$M_{\alpha\beta a} = \sum_b m_b \frac{\mathbf{x}_b^\alpha - \mathbf{x}_a^\alpha}{\rho_b} \nabla_a^\beta w(\mathbf{x}_a - \mathbf{x}_b, h) \tag{58}$$

$$A_{\alpha a} = \frac{\partial A}{\partial x^\alpha}(\mathbf{x}_a) \tag{59}$$

$$F_{\beta a} = \sum_b m_b \frac{A_b - A_a}{\rho_b} \nabla_a^\beta w(\mathbf{x}_a - \mathbf{x}_b, h) \tag{60}$$

4.7 multiple phase SPH

Numerical simulation of multiphase flows is usually difficult due to the existence of interface between two fluids. And the interface is usually moving or deformable. This present severe challenges to conventional Eulerian grid-based numerical methods. As we discussed in introduction section, special algorithms to treat and track the interface between different phases is required for mesh-based method to deal with mulphase problems. What's worse, the engulfment process during plume development makes geometry of interface very complicated. SPH is able to capture interface without additional numerical effort, which makes it suitable for volcano plume simulation. Take the governing equation adopted in our model as an example. Different phases of fluid (air and erupted material) will be represented by two different sets of SPH particles. Based on assumption we made in section 3, only density need to be updated respectively for each phase. The updating of density is exactly the same as equation (38) in spirit. Particles of phase one will not be counted while evaluate density of phase two and vice verse. Updating of density is based on the following discretized equations.

$$< \rho_\alpha^a >= \frac{\sum m_b w_{\alpha b}(h_\alpha)}{\sum \frac{m_b}{\rho_b} w_{\alpha b}(h_\alpha) + \sum \frac{m_j}{\rho_j} w_{\alpha j}(h_\alpha)} \tag{61}$$

$$< \rho_\alpha^{sg} >= \frac{\sum_j m_j w_{\alpha j}(h_\alpha)}{\sum \frac{m_b}{\rho_b} w_{\alpha b}(h_\alpha) + \sum \frac{m_j}{\rho_j} w_{\alpha j}(h_\alpha)} \tag{62}$$

Where Where the subscript a and b represents air particles (phase 1) while i and j represents erupted material. β and α represents either erupted material or air. ρ_a^a is density of phase 1 (air). ρ_i^{sg} is density of phase 2 (erupted material). $\rho = \rho^a + \rho^{sg}$ is density of mixture of phase 1 and phase 2. In the areas far away from interface, updating of density is exactly the same as that for single phase flow. For example, in the right side and left side portition (or blue) of figure 1, where there are only air particles, so equation (62) will lead to zero and total density will be:

$$\rho_\alpha = \rho_\alpha^a = \frac{\sum m_b w_{\alpha b}(h_\alpha)}{\sum \frac{m_b}{\rho_b} w_{\alpha b}} \tag{63}$$

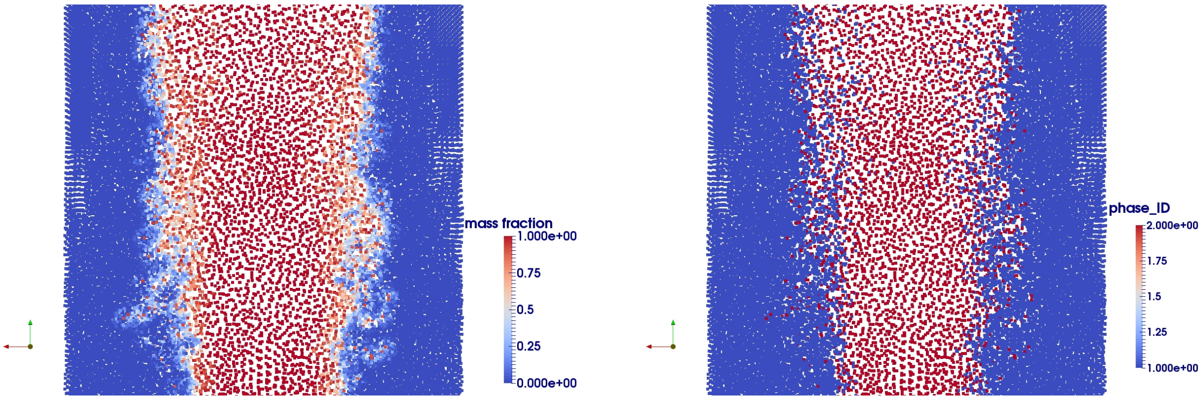
Which is a special case of equation (55). For these areas occupied by only particles of phase two and far away from interface, very similarly, the equation for density update will become:

$$\rho_\alpha = \rho_\alpha^{sg} = \frac{\sum m_j w_{\alpha j}(h_\alpha)}{\sum \frac{m_j}{\rho_j} w_{\alpha j}} \tag{64}$$

4.8 Turbulence modelling with SPH

Typically, turbulence fluctuations in a volcanic column occur on very different length scales, ranging from millimeters up to hundreds of meters. The entrainment of air and heat tranfer are dominated by such turbulent fluctuations as turbulent exchange coefficient would be several magnitudes larger than molecular viscosity/ heat conduction coefficient. One way to include enough turbulence in the model is to use SPH-SPS (SPH sub-particle-scale) method. Another way is to use extremely fine resolution (The direct numerical simulation method) at the expense of much higher computational cost that results from both increase in number of particles and decrease in time steps constrained by CFL condition. Here we choose the SPH-SPS method to capture momentum and energy exchange caused by smaller scale turbulence.

Even though free surface flow are in nature turbulent, detailed research in turbulence using the SPH method



(a) Mass fraction

(b) Phase of particles

Figure 1: In the left figure, the blue particles represent air particles the red ones represent erupted material. The right figure shows corresponding mass fraction. Mass fraction are evaluated based on equation (61) and equation (61) without any additional numerical tricky.

is rather scarce. Among the first papers on turbulence are the implementations of the α turbulence model [19] by Monaghan [32] and the Reynolds averaged Navier–Stokes (RANS) approach by Violeau and Issa [60]. A first wall-bounded large eddy simulation (LES) was performed by Issa in 2004 [20]. An improved version of $SPH-\alpha$ method is proposed in 2011 by Monaghan [37], which we refer to as $SPH-\varepsilon$ method. We will adopt $SPH-\varepsilon$ method in simulation of volcano plume. As most of these papers constrain their focus on incompressible or weakly compressible fluid. To simulate turbulence in plume, we need to extend existing incompressible turbulent modelling methods to compressible flows.

$SPH-\alpha$ model [32] is a powerful extension of the XSPH algorithm which reduces disorder at short length scales and retains the constants of the motion. But in that paper [32], energy conservation equation is discretized in a way that is exactly the same as usual SPH. There is no special treatment on energy conservation equation. And the recent improved version [37] of $SPH-\alpha$ method focus on incompressible flow only. Actually, there are several additional terms in the averaged compressible Navier-Stokes equations besides the Reynolds stress term [48]. The averaged momentum equation for compressible flow are in the same form as that for incompressible flow, all of other additional terms except for turbulent stress term are in the energy equation). However, most turbulence modeling focuses on the Reynolds stress terms which are either solved directly or defined via a constitutive relation. Less attention is typically given to the other terms that need to be modeled. Most commonly, a Reynolds analogy is used to model the turbulent heat flux. Simulations of heat transfer or other scalar transfer in turbulent flow simply involve adding transport terms for thermal energy or species concentration, at the expense of greater storage and longer computing times but without other difficulties [7]. We adpot this strategy in our simulation of turbulent heat transfer with $SPH-\varepsilon$. PDAC [41] also adopted a Reynolds analogy in its turbulence modelling. The additional terms associated with molecular diffusion and turbulent transport in the energy equation are either modeled in different ways or neglected sometimes[48]. We neglect these terms in our simulation.

4.8.1 $SPH-\varepsilon$ turbulence model

In $SPH-\varepsilon$ method [37], a turbulence model is constructed within the framework of SPH in such a way that general principles such as conservation of energy, momentum and circulation are satisfied using the ideas associated with the Lagrangian averaged Navier–Stokes equations(LANS). The basic idea of $SPH-\varepsilon$ is that determine a smoothed velocity $\widehat{\mathbf{v}}$ by a linear operation on the unsmoothed velocity $\mathbf{v}$. The SPH particles move with this smoothed velocity and the average motion of the fluid is determined by equation(65):

$$\frac{d\mathbf{x}_a}{dt} = \widehat{\mathbf{v}}_a \tag{65}$$

Lagrange's equations contain extra velocity dependent term which represents the stresses induced by the smoothing. Once the form of the smoothing is chosen these stresses are determined. The typical LANS model uses a smoothed velocity $\widehat{\mathbf{v}}$ defined in terms of the unsmoothed velocity $\mathbf{v}$ by:

$$\widehat{\mathbf{v}}(\mathbf{x}) = \int \mathbf{v}(\mathbf{x}\prime)G(|\mathbf{x}\prime - \mathbf{x}|, l)d\mathbf{x}\prime \tag{66}$$

Where G satisfies:

$$\int G(|\mathbf{x}\prime - \mathbf{x}|, l) d\mathbf{x}\prime = 1 \tag{67}$$

and is a member of a sequence of functions which tends to the δ function in the limit where $l \rightarrow 0$. A typical example is Gaussian. The length scale l determines the characteristic width of the kernel and the distance over which the velocity is smoothed.
It is common practice in $LANS - \alpha$ to use a differential equation for the smoothing rather than the integral form and finally reach to system of equations that need to be solved implicitly. In $SPH - \varepsilon$ method, a XSPH smoothing is adopted which will conserve linear and angular momentum. This furtherly avoid solving of system of equations and make the method simple to implement and cheap for computation. The discretized form of momentum equation is obtained through lengthy derivation based on Lagrange's equations. Derivation details and other discussions, such as conservation of momentum and energy, satisfactory of circulation theory, energy/momentum spectrum analysis based on Founrier transfer are detailed in Monaghan's paper[37]. Here we just have a brief summary on key steps.
The smooth that Monaghan adopted is:

$$\widehat{\mathbf{v}}(\mathbf{x}) = \mathbf{v}(\mathbf{x}) + \epsilon \int (\mathbf{v}(\mathbf{x}\prime) - \mathbf{v}(\mathbf{x})) G(|\mathbf{x}\prime - \mathbf{x}|, l) d\mathbf{x}\prime \tag{68}$$

As function G has the same feature as kernel function w, SPH approximation of the intergration will lead to:

$$\widehat{\mathbf{v}}(\mathbf{x}) = \mathbf{v}(\mathbf{x}) + \epsilon \sum_b m_b \frac{(\mathbf{v}_b - \mathbf{v})}{\rho_b} G(|\mathbf{x}_b - \mathbf{x}|, l) \tag{69}$$

By making the replacement:

$$\frac{G(|\mathbf{x}_b - \mathbf{x}_a|, l_a)}{\rho_b} \rightarrow \frac{K_{ab}}{M} \tag{70}$$

Where $K_{ab} = l^d G_{ab}$, $M = \rho_0 l^d$ in which d is the dimension and ρ_0 is initial density. $SPH - \varepsilon$ turbulence model is obtained after longthy derivation:

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b [m_b (\frac{p_b}{\rho_b^2} + \frac{p_a}{\rho_a^2}) \nabla_a w_{ab}(h_a)] + \sum_b m_b \frac{\varepsilon}{2} \frac{\mathbf{v}_{ab} \cdot \mathbf{v}_{ab}}{M} \nabla_a K_{ab} \tag{71}$$

Instead of using equation(71) which is from Monaghan's paper, we make inverse replacement of equation(70) and notice that if l is uniform and keep constant:

$$\nabla K_{ab} = \nabla(l^d G_{ab}) = l^d \nabla G_{ab} \tag{72}$$

So the discretized momentum equation with $SPH - \varepsilon$ turbulence model in our simulation will be:

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b [m_b (\frac{p_b}{\rho_b^2} + \frac{p_a}{\rho_a^2}) \nabla_a w_{ab}(h_a)] + \sum_b m_b \Phi_{ab} \nabla_a G_{ab}(l_a) \tag{73}$$

where

$$\Phi_{ab} = \frac{\varepsilon}{2} \frac{\mathbf{v}_{ab} \cdot \mathbf{v}_{ab}}{\rho_b} \tag{74}$$

and take $\varepsilon = 0.8$ as recommended by JJ Monaghan.

4.8.2 Turbulence heat transfer

We adopt Reynolds analogy to get heat transfer coefficient due to turbulence. Prandtl number is defined as:

$$Pr = \frac{C_p \mu}{\kappa} \tag{75}$$

Where, μ is dynamic viscosity, κ is thermal conductivity. And μ can be written in term of abslolute viscosity (kinematic viscosity) as:

$$\mu = \rho \nu \tag{76}$$

Then

$$\kappa = \frac{C_p \mu}{Pr} \tag{77}$$

Typical value of Pr_t for air is $0.7 \sim 0.9$. We take $Pr_t = 0.85$ for gases as recommended by Kays, William M [21] from summarizing of experimental results. JJ Monaghan [33] summarized the simulation of viscosity and heat conduction in his review on SPH. We will refer to his summary in our following discussion. The additional term in discretized momentum equation, equation (73), is the turbulent shear stress term. Recall that viscosity term can be discretized with SPH as equation (45). Comparing of this term to physical viscous term for compressible gas shows that it has both bulk viscosity and shear viscosity, where shear viscosity coefficient is[33]:

$$\nu_t = S\nu \tag{78}$$

with

$$S = \begin{cases} \dfrac{1}{6} & for \quad d = 3 \\ \\ \dfrac{5}{24} & for \quad d = 2 \end{cases} \tag{79}$$

If we can reformulate the "turbulence stress" term in a form which is similar to molecular shear viscous term. then the turbulent viscosity coefficient can be inferred from that form. Reformulate the "turbulence stress" term:

$$\sum_b \frac{\varepsilon}{2}\frac{m_b}{\rho_b}\mathbf{v}_{ab}\cdot\mathbf{v}_{ab}\nabla_a G_{ab}(l_a) = \sum_b \frac{\varepsilon}{2S} m_b \frac{\mathbf{v}_{ab}}{\rho_b}\frac{S\mathbf{v}_{ab}\cdot\mathbf{x}_{ab}}{x_{ab}^2}\frac{x_{ab}^2}{\mathbf{x}_{ab}}\nabla_a G_{ab}(l_a) \tag{80}$$

Then the turbulent viscosity coefficient will be

$$\nu_t = \frac{\varepsilon}{2S}\frac{\mathbf{v}_{ab}\cdot\mathbf{x}_{ab}}{\rho_b} \tag{81}$$

Please notice that turbulent viscosity term has opposite sigh with molecular viscosity term in discretized momentum equation and there is a minus sign in the expression of Π_{ab}, they will cancelled out. However, the above equation is correct only for 1D. For 2D or 3D, it is not easy to get the explicit expression of viscosity.
We adopt an alternative way: obtaining a value for each pair of particles instead of persist on getting an anlytical expression. Choosing smoothing function as the same as SPH kernel and the smoothing scale l as the same as smoothing length h, the ratio between turbulent shear stress and physical shear stress is:

$$\begin{aligned} \Upsilon_{ab} &= \frac{\dfrac{\varepsilon}{2}\dfrac{\mathbf{v}_{ab}\cdot\mathbf{v}_{ab}}{\rho_b}}{\dfrac{S\nu}{\rho_{ab}}\frac{\mathbf{v}_{ab}\cdot\mathbf{x}_{ab}}{x_{ab}^2+\eta^2 h_{ab}^2}} \\ &= \frac{\varepsilon(x_{ab}^2+\eta^2 h_{ab}^2)}{2S\nu}\frac{\mathbf{v}_{ab}\cdot\mathbf{v}_{ab}}{\mathbf{v}_{ab}\cdot\mathbf{x}_{ab}} \end{aligned} \tag{82}$$

Υ_{ab} is essentially equivalent to ratio between turbulent viscous effect attributed to particle b on particle a and molecular viscous effect attributed to particle b on particle a. Turbulent viscosity can be easily obtained by:

$$\begin{aligned} \nu_{t,ab} &= \nu\Upsilon_{ab} \\ &= \frac{\varepsilon(x_{ab}^2+\eta^2 h_{ab}^2)}{2S}\frac{\mathbf{v}_{ab}\cdot\mathbf{v}_{ab}}{\mathbf{v}_{ab}\cdot\mathbf{x}_{ab}} \end{aligned} \tag{83}$$

The corresponding turbulent thermal conductivity should be

$$\kappa_{t,ab} = \frac{\varepsilon\overline{C_{p,ab}}\overline{\rho_{ab}}(x_{ab}^2+\eta^2 h_{ab}^2)\mathbf{v}_{ab}\cdot\mathbf{v}_{ab}}{2SPr_t\mathbf{v}_{ab}\cdot\mathbf{x}_{ab}} \tag{84}$$

The term that used to prevent singularity now can be removed.

$$\kappa_{t,ab} = \frac{\varepsilon\overline{C_{p,ab}}\overline{\rho_{ab}}x_{ab}^2\mathbf{v}_{ab}\cdot\mathbf{v}_{ab}}{2SPr_t\mathbf{v}_{ab}\cdot\mathbf{x}_{ab}} \tag{85}$$

We also need to prevent singularity, so:

$$\kappa_{t,ab} = \begin{cases} 0 & if \quad \mathbf{v}_{ab} = 0 \quad or \quad \mathbf{x}_{ab} = 0 \\ \dfrac{\varepsilon\overline{C_{p,ab}}\overline{\rho_{ab}}x_{ab}^2\mathbf{v}_{ab}\cdot\mathbf{v}_{ab}}{2SPr_t\mathbf{v}_{ab}\cdot\mathbf{x}_{ab}} & otherwise \end{cases} \tag{86}$$

The heat conduction equation without source term is:

$$C_p \frac{dT}{dt} = \frac{1}{\rho}\nabla(\kappa \nabla T) \tag{87}$$

Second spatial derivative can be approximated with SPH by following JJ Monaghan [33].

$$C_p \frac{dT}{dt} = \sum_b \frac{m_b}{\rho_a \rho_b}(\kappa_a + \kappa_b)(T_a - T_b)F_{ab}(h_a) \tag{88}$$

Where

$$F_{ab}\mathbf{x}_{ab} = \nabla_a w_{ab} \tag{89}$$

And $F_{ab} \leq 0$ which will guarantee that heat flux will flow from hot to cold. Plug the turbulent thermal conductivity into the heat conduction equation:

$$\begin{aligned} C_p \frac{dT}{dt} &= \sum_b \frac{m_b}{\rho_a \rho_b}(\kappa_a + \kappa_b)(T_a - T_b)F_{ab}(h_a) \\ &= 2\sum_b \frac{m_b}{\rho_a \rho_b} \frac{\overline{C_{p,ab}} S \varepsilon x_{ab}^2 \mathbf{v}_{ab} \cdot \mathbf{v}_{ab}}{2Pr_t \mathbf{v}_{ab} \cdot \mathbf{x}_{ab}}(T_a - T_b)F_{ab}(h_a) \end{aligned} \tag{90}$$

Notice that the number 2 at the front of the equation comes from intergration approximation of second derivative [9]. By further simplification, we get:

$$C_p \frac{dT}{dt} = \frac{S\varepsilon}{Pr_t} \sum_b \frac{m_b}{\rho_a \rho_b} \frac{\varepsilon \overline{C_{p,ab}}\,\overline{\rho_{ab}} x_{ab}^2 \mathbf{v}_{ab} \cdot \mathbf{v}_{ab}}{2SPr_t \mathbf{v}_{ab} \cdot \mathbf{x}_{ab}}(T_a - T_b)F_{ab}(h_a) \tag{91}$$

4.8.3 Discretized governing Equation with $SPH - \varepsilon$ turbulence model

Plug discretized turbulent stress term and turbulent heat transfer term into the momentum and energy equation, we got new discretized governing equation:

$$< \frac{d\mathbf{v}_\alpha}{dt} >= -\sum_b [m_b(\frac{p_b}{\rho_b^2} + \frac{p_\alpha}{\rho_\alpha^2} + \Pi_{\alpha b} - \Phi_{\alpha b}) \nabla_\alpha w_{\alpha b}(h_\alpha)] - \sum_j [m_j(\frac{p_j}{\rho_j^2} + \frac{p_\alpha}{\rho_\alpha^2} + \Pi_{\alpha j} - \Phi_{\alpha j}) \nabla_\alpha w_{\alpha j}(h_\alpha)] + \mathbf{g} \tag{92}$$

With

$$\Phi_{\alpha\beta} = \frac{\varepsilon}{2} \frac{\mathbf{v}_{\alpha\beta} \cdot \mathbf{v}_{\alpha\beta}}{\rho_\beta} \tag{93}$$

$$\begin{aligned} < \frac{de_\alpha}{dt} > = 0.5 \sum_b [m_b \widehat{\mathbf{v}_{\alpha b}}(\frac{p_b}{\rho_b^2} + \frac{p_\alpha}{\rho_\alpha^2} + \Pi_{\alpha b} - \Phi_{\alpha b}) \nabla_\alpha w_{\alpha b}(h_\alpha] + \sum_b \frac{m_b}{\rho_\alpha \rho_b} \kappa_{t,\alpha b}(T_\alpha - T_b)F_{\alpha b}(h_\alpha) \\ + 0.5 \sum_j [m_j \widehat{\mathbf{v}_{\alpha b}}(\frac{p_j}{\rho_j^2} + \frac{p_\alpha}{\rho_\alpha^2} + \Pi_{\alpha j} - \Phi_{\alpha j}) \nabla_\alpha w_{\alpha j}(h_\alpha)] + \sum_j \frac{m_j}{\rho_\alpha \rho_j} \kappa_{t,\alpha j}(T_\alpha - T_j)F_{\alpha j}(h_\alpha) \end{aligned} \tag{94}$$

with

$$\kappa_{t,\alpha\beta} = \frac{\varepsilon \overline{C_{p,ab}}\,\overline{\rho_{ab}} x_{ab}^2 \mathbf{v}_{ab} \cdot \mathbf{v}_{ab}}{2SPr_t \mathbf{v}_{ab} \cdot \mathbf{x}_{ab}} \tag{95}$$

As the particle-scale movement of flow is based on smoothed velocity, the velocity in the energy equation should also be smoothed. The filtering process will be done accroding to equation (69). Position of particles will be updated according to equation (65). Smoothed velocity is also used while computing artificial viscosity Π_{ab}.

4.9 Boundary conditions in SPH

4.9.1 Wall boundary condtion

Traditionally either ghost particles that mirror SPH particles across the boundary [14], or boundary forces [38] have been used to impose the wall boundary conditions. One disadvantage of the later one is that the boundary forces tend to corrupt the solution in the neighborhood of such boundary forces. In addition, the natural way of imposing eruption boundary condition is using eruption ghost particles. To impose boundary conditions in a consistent way we adopt a modified version of ghost particle method [25] for wall boundary conditions. Stationary ghosts particles are deployed the same way as deployment of real particles. Instead of enforcing the symmetry particle by particle, symmetric field across the boundary is explicitly enforced. Ghost positions are reflected into the domain and physical properties are calculated at these reflected positions by SPH interpolations. It should be noted that ghost particles are not used for these interpolations. We assign a velocity to each wall ghost particles that is the negative of the interpolated velocity of its corresponding reflection inside the fluid domain. These wall ghost particles will serve as neighbors in momentum and energy update. Implementation details about this method can be found in [25].

4.9.2 Eruption boundary condition

The natural way of imposing boundary condition is using erupt ghost particles which moves with eruption velocity and bears the same temperature of erupt material. A parabolic velocity profile which represents a fully developed Hagen–Poiseuille flow is used to determine the particle velocity. The correctness of eruption boundary condtion for density and mass fraction eruption can be guaranteed by choosing mass of erupted ghost particles given distribution of these particles. The internal energy associated with these particles are set to values so that equation 19 is satisfied. The density, momentum, and internal energy of these eruption ghost particles will not be updated before they move above ground. As soon as they enter the atmosphere, the ghost particle will be shift to real particle and its physics properties and position will be updated based on discretized governing equations. New ghost particle will be added at the bottom of the eruption vent as these exising ghost particles move upwards.

4.9.3 Pressure boundary condition

An other boundary in this simulation is pressure outlet boundary. For flow in a straight channel, it is possible to treat the exit the same as the entry with a prescribed velocity profile. For flow with more complex channel, an exit far downstream of the flow disturbance is also feasible. However, the natural boundary (equation 24) condition is more suitable for plume simulation where the outlet is open atmosphere. The way we impose pressure boundary conditon is simple: adding several layers of pressure ghost particles souroundning the real atmosphere particles. Physics properties except for velocity (pressure, density, temperature) are determined based on the elevation of particles. Velocity is set to zero for static atmosphere. The physics properties for pressure ghost particles will not be updated while these for real particles will be updated at every time step. As position of all pressure ghost particles are keep constant, we are essentially imposing a static pressure boundary condition. Real particles that moves out the the pressure boundary will be removed.

As simulation progress, change in position and physical properties of real particles near pressure boundary boundary might corrupt pressure boundary condition that established initially. This shortcoming is relieved by choosing a larger computational domain so that these might-be-corrupted boundary will be far away from turbulent mixing area. In addition, to avoid enlarging fluctuation, we add another constrain on time step:

$$\Delta t \leq CFL_p \frac{h}{v} \tag{96}$$

Where CFL_p is a safty coefficient which has similar functionalality as the normal CFL number. Too small CFL_p will slow down simulation while too large CFL_p will lose its ability of mitigate numerical fluctuation near the boundary. The proper CFL_p is determined by series of simulation tests.

4.10 Parallellism and performance

Among existing CPU parallel SPH schemes, most of them focus on neighbors searching algorithm and dynamic load balancing. (eg.[14, 11]). Less attention has been paid to developing of more flexible data management schemes for more complicated problems. Motivated by techniques developed for mesh based methods, we present a complete framwork for parallelizing SPH program with MPI standard model allowing flexible and efficient data access in this paper.

Any implementation of SPH code requires efficient searching and updating of neighbors during simulation. Of the many choices possible for this we adopted a background grid which was proposed by Monaghan and Lattanzio[39] and is quite popular in parallel SPH. The background grid is also used for domain decomposition in SPH. We refer to the elements of background grid, namely squares for two dimension and cubics for three dimension, as buckets. As for the actual storage of data representing the physical quantities associated to each particle, different strategies have been adopted in existing implementations of SPH. In both SPHysics and DualSPHysics[11], the physical quantities of each particle (position, velocity, density...) are stored in

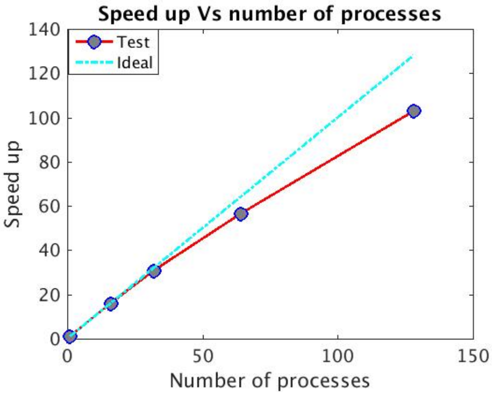


Figure 2: Strong scalability test results

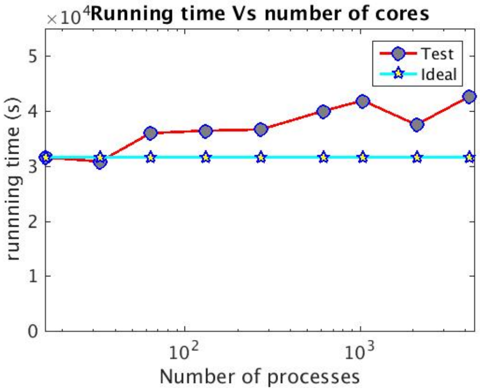


Figure 3: Weak scalability test results

arrays, and the particles (and the arrays with particle data) are reordered following the order of the cells. This has two advantages: 1) access pattern is more regular and more efficient, 2) it is ease to identify the particles that belong to a cell by using a range since the first particle of each cell is known. But adding, deleting and especially accessing of particles are cumbersome. Ferrari[14] adopted linked lists using pointers so that particles can be deleted or added during the simulation. Storage problems caused by fix-size arrays are thereby also eliminated. We define C++ classes which contain all data of particle and bucket. As for the management of data, we adopt hash tables to store pointers to particles and buckets, which give us not only flexibility of deleting and adding element, but also quicker access compared with linked list. Instead of using the ”nature manner” to number particles, we adopt SFC based index to give each particle and background bucket an unique identifier – a strategy known to preserve some locality at minimal cost. The SFC based numbering strategy is further extended to include time step information so that particles added at the same position but different time will have different identifiers. As for domain decomposition, even though more complicated graph-based partition tools[2] might get higher quality decomposition, they requires much more effort in programming and computation. So we adopt an easy-programming scheme based on SFC [45].

Experiments have been carried out on the computational cluster of Center for Computational Research (CCR) at Buffalo. E5645 CPUs running at 2.40GHz clock rate with 4GB memory per core on a Q-Logic Infiniband is used in these tests. Each node comprise of two sockets with 6 of these cores. Memory and level 3 cache are shared on each node. The initial domain is $[-4.8km, 4.8km] \times [-4.8km, 4.8km] \times [0km, 6km]$. Almost linear speed up is observed in figure 2. The weak scalability test is conducted with the same initial domain and various smoothing lengths. Each simulation is run for 400 steps. The average number of real particles of each process is kept constant at 25900. As shown in figure 3, simulation times increase around 1/3 when number of cores increase from 16 to 4208. For the test problem in this paper, the volcanic plume will finally reach to a region of $[-30km\ 30km] \times [-30km\ 30km] \times [1.5km\ 40km]$ after around 400 seconds of eruption. When numerical simulation goes up to 90 seconds, the plume is still within a region of $[-10km\ 10km] \times [-10km\ 10km] \times [0km\ 25km]$. This implies that adjusting of domain can avoid computing large number of uninfluenced air particles, especially for the beginning stage of simulation. Figure (4) shows that simulation time of the test problem is greatly reduced when we adopt the domain adjusting strategy in our code.

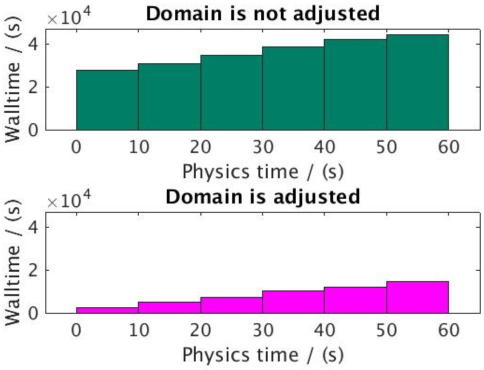


Figure 4: The effect of domain adjusting on simulation time. The figure on the top shows simultion time without domain adjusting, the figure on the bottom shows simultion time with domain adjusting. Different bins represent simulation time up to specific physics time indicated by x axis.

5 Verification and validation

We will now present a series of numerical simulations that were used to verify and validate the model. The SPH plume model is first verified by a JPUE simulation. Velocity and mass fraction distribution both along the central axis and cross transverse are compared with experimental results. The parten of ambient particles entrainment is also clearly shown. Then a simulation of representative strong volcano plume is conducted. Both global variables and local variable are comparable with simulation results from exisitng 1D and 3D plume models. Analysis on evolution of plume shows detailed air entrainment mechanism and the structure of plume are consistent with simulation results of existing 3D models.

5.1 Simulation of JPUE

JPUE can be considered as a simplified volcanic plume. While the effect of stratified atmosphere and the effect of expansion due to high temperature in volcanic plume are not represented, JPUE reproduces the entrainment due to turbulent mixing which is one of the key elements in volcanic plume development. In this section, we will verify that our code and the turbulence model we adopted is able to reproduce feature of turbulent entrainment. There exist consistently good experimental data [44, 27, 12] that describe the JPUE flow field giving an insight into detailes of JPUE, such as transverse velocity and concetration profile. As many of these experiments were conducted with liquid, we replace the original equation of state (equation(5)) with a weakly compressible Tait equation of state [1] to avoid solving the Poisson equation.

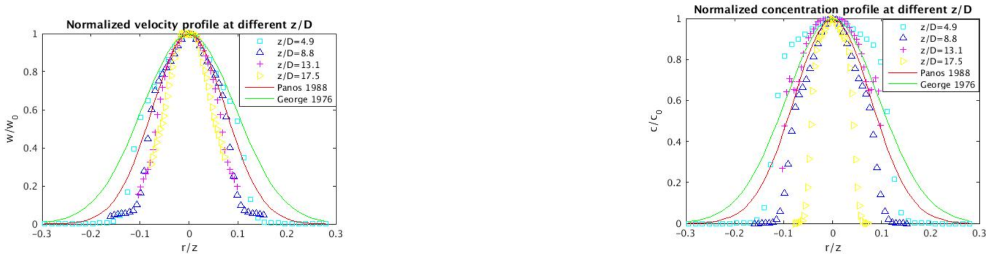
$$p = B[(\frac{\rho}{\rho_0})^{\gamma} - 1] \tag{97}$$

with $\gamma = 7$ and B is evaluated by:

$$B = \frac{\rho_0 c^2}{\gamma} \tag{98}$$

Where c is speed of sound in the liquid. Energy equation is actually decoupled from momentum and mass conservation equation by using this EOS. In addition, the ”atmosphere” is assumed to be uniform and gravity is set to be zero in JPUE simulation. We set the temperature and density of ejected material from the vent the same as its surrounding ambient. This will further simplify the scenario for the convenience of studying turbulent mixing. In this section, an axisymmetric JPUE which ejects from a round vent is simulated. The concetration and velocity distribution in both axial and radial direction were compared against various available experimental results.

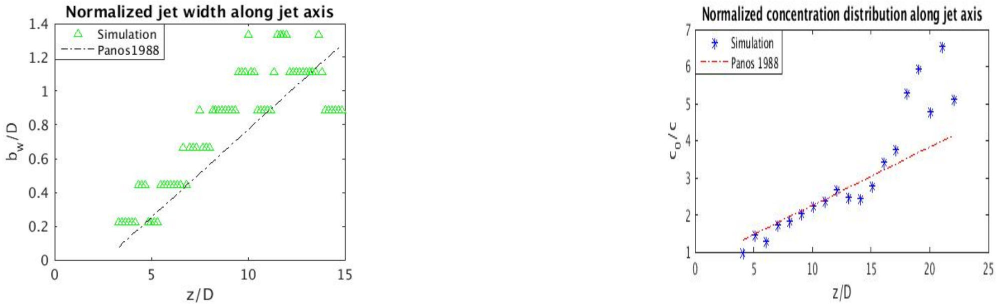
One overall feature of JPUE is ”self-similarity” which means that the evolution of the JPUE is determined soley by the local scale of length and velocity which theoretically account for the fact that the rate of entrainment at the edge of JPUE is proportional to a characteristic velociy at each height. As a results, physical and numerical experiments with different setups are comparable on a non-dimensional basis. A three dimensional JPUE simulation is coducted and results are compared with experiments [44, 15] for verification purpose in figure 5 and figure 6. Experimental data of concentration and velocity distribution across the cross-section was fit into a Gaussian profile: $\frac{\varphi}{\varphi_c} = exp[-coef(\frac{r}{z})^2)]$ (solid line) even though there is no priori reason. Panos also fit concentration distribution and jet width based on velocity along central line into a straight line



(a) Velocity

(b) Concetration

Figure 5: Dimensionless concetration and velocity distribution across the cross-section.



(a) Velocity

(b) Concetration

Figure 6: Dimensionless concetration and velocity distribution along centraline.

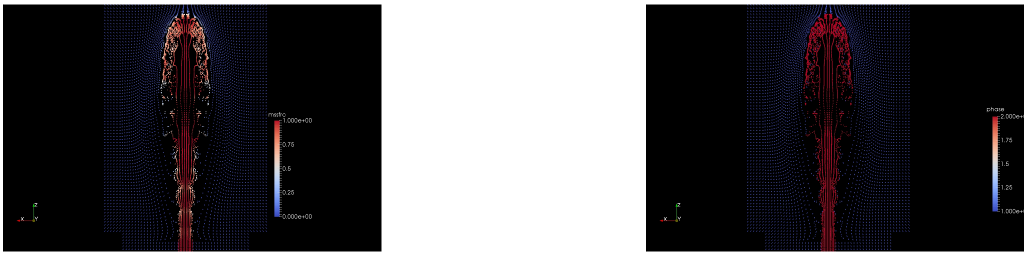
: $\frac{\varphi_0}{\varphi_c} = slope(\frac{z}{D} + intercept)$ (dash line). Where φ is either velocity or concetration, the subscript c represents the centraline, 0 represents the cross-sectionally averaged exit value. r is the distance from the centraline on any cross-section. z is the axial distance from the origin of the jet transverse section underconsideration. D is the diameter of vent. The coefficient $coef$ for concetration is 80 and 50 respectively according to Panos and George. $coef$ for velocity is 90 and 55 respectively according to Panos and George. $slope$ for jet width based on velocity is 0.104 and for concetration is 0.157. $intercept$ for jet width based on velocity is 2.58 while that for the concetration is 4.35.

Although both velocity and concetration are found to be well matched with experimental results, a small disparity in both velocity and concetration are observed near the boundary of the jet. There are several factors that would attribute to such disparity. Reynolds number are not reported in many experiments assuming a high enough Reynolds number. In addition, some details of the of the experiments, such as exit velocity profile, viscosity of the experimental liquid, are not reported, neither. These factors prevent us from repeating these experiments exactly with simulation. However, the features of JPUE is correctly reproduced with our code.

We also investigated the mixing due to turbulence in JPUE simulation by checking the mixture of two phases: the ejected liquid and ambient liquid. What need to be noticed is that the mass of particle of phase one (blue) is 8 time of that of phase 2. It is a trade off between resolution and computational expense to give particle of two phases different particles mass. Both advanced SPH techniques like adaptive particles size and better data management and access strategies need to be exploited to address this issue. It is shown in figure 7 that the ejected material and ambient fluids are mixed through eddies at the outer shear region. And the inner dense core dispersed gradually due to erosion of the outer shear region. Hence, our confidence in the numerical correctness of this model is greatly reinforced.

5.2 Simulation on Plume

The physics of volcano plume is more complicated than JPUE in several aspects. Besides turbulent entrainment of ambient fluids, development of volcano plume also involves heating up of entrained air and then expanding in a stratified atmosphere. We will aplly our model to simulate plume development and in purpose of further validation. A strong eruption column without wind is tested in this section. Input parameters, material properties and atmosphere are choosen to be the same as one case in an inter-comparison study on eruptive column models [10]. The strong plume scenario was based on climactic phase of the Pinatubo eruption(Philippines, 15 June 1991). Material properties are selected based on properties of Pinatubo and Shinmoe-dake eruption.The plume evolution process are first investigated. Then both global variable and local variables are compared



(a) mass fraction of erupted material

(b) entrainment of air particles

Figure 7: First figure shows the mass fraction of erupted material.The second figure shows particle distribution at the moment corresponding to first figure. Ambient particles (blue) are gradually entrained and mixed with erupted particles (red) as jet flows upward.

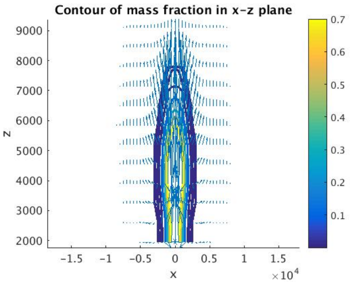


Figure 8: Contour of mass fraction and velocity at 15 seconds, column rises up quickly due to intial momentum.

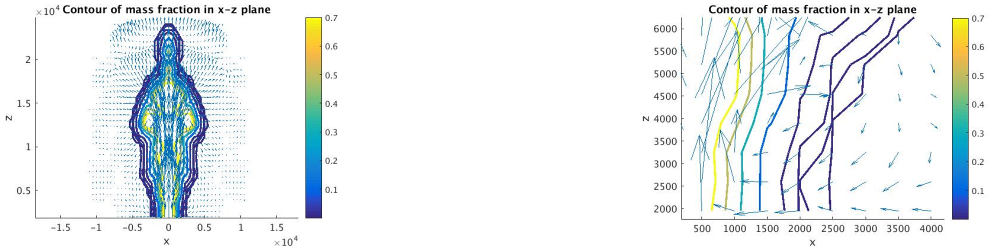
with existing models.

5.2.1 Evolution of plume

Figure 8 12 shows the dynamic evolution process of volcano plume. The contour level of mass fraction are 0.7, 0.5, 0.3, 0.1, 1×10^{-1}, 1×10^{-2}, 1×10^{-3}, 1×10^{-4} and 1×10^{-5}. The plume keeps rising up and entraining air into the plume untill it reaches its top height(around t=200 seconds). The mixtures of erupted material and air falls back to neutral height and starts spreading radially after reaching the top height. Detailed velocity quiver plots show eddies at the edge of column entrainning air into the plume. It also shows in these figures that the plume near the vent maintains a concentric structure with an inner dense core surrounded by an outer shear region which is consistent with observations and previous simulation.

5.2.2 Global and local variables

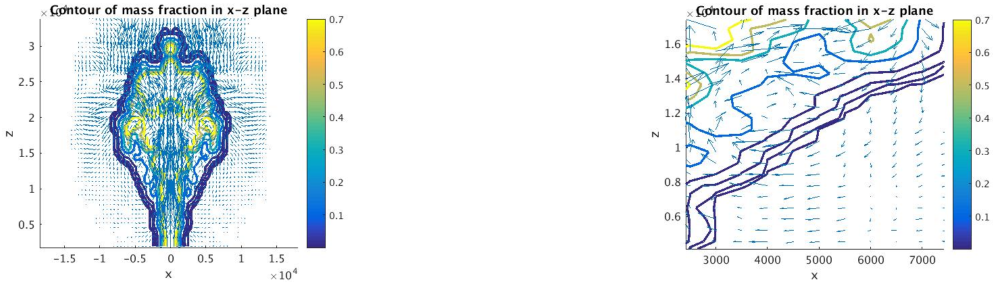
One of the key global quantities of great interest is the altitude to which the plume rises. The top height predicted by our model is around 40km which agrees well with other plume models. For example, the height predicted by PDAC is 42500m, by SK-3D is 39920m, by ATHAM is 33392m. As for local variables, the temperature, the mass fraction of entrained air, the gas mass fraction, the mass fraction of solid, the radius of



(a) Overall view

(b) Detailed view

Figure 9: Contour of mass fraction and velocity at 50 seconds, column keeps rising up and at the same time entraining air into the plume at the outer shear region. The entrainment of air is shown in a zoomed view in which the velocity quiver clearly shows entrainment of air into the plume at the margin of plume.



(a) Overall view

(b) Detailed view

Figure 10: Contour of mass fraction and velocity at 100 seconds, it keeps rising up and entraining air. The entrainment of air is clearly shown in dzoomed view

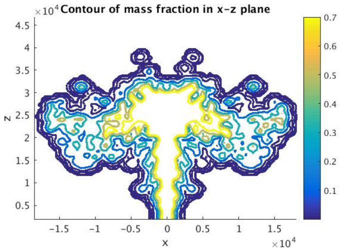


Figure 11: Contour of mass fraction at 200 seconds, plume reaches its top height

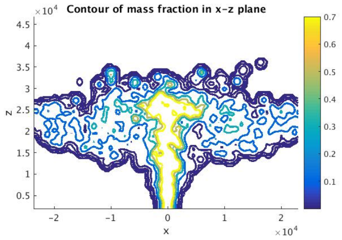


Figure 12: Contour of mass fraction at 300 seconds, the cloud is spreading

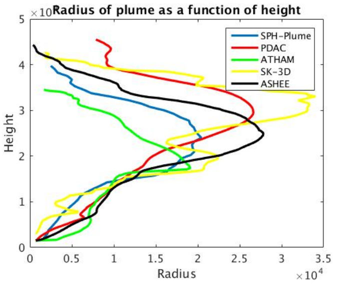


Figure 13: Radius of the strong plume without wind after reaching its top height

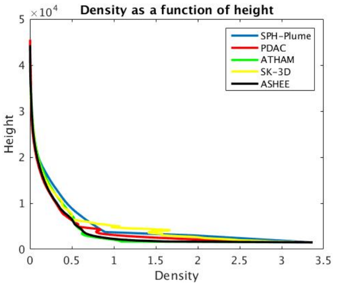


Figure 14: Density of the strong plume without wind after reaching its top height

the plume as a function of height are compared with existing 1D and 3D models in figure 15-17. To get rid of significant fluctuations in time and space we did time averaging and spatial integration of the 3D flow fields by following Cerminara[**?**] As particles distribute in the space in a disordered manner. We first project simulation results (on disordered particles) onto a pre-defined grid before doing time average and spatial intergration. The interpolation method in projection is the basic SPH kernel based interpolation.

The profile of plume radius, density, velocity, mass fraction and temperature matches with simulation results of exisitng 3D models in general sense. The basic phenomena in volcano plume development is correctly captured by our model: the plume first rises up to its top height and then falls down to the netural buoyancy height and starts expanding.

As the height increase, the amount of entrained air will increase. At the netural height, where the umbrella expanding, the entrainment decrease due to lack of air surrounding the column. The profile of mass fraction of gas, solid and air are consistent with these results predicted by exisitng 3D plume models in general. More and more cool air will be entrained into the plume and mix with hot erupted material as the plume rises up. Heat exchange between air and erupted material will also cool the plume down. So the temperature of the plume decreases as the height increase as shown in figure 17. To summarize, analysis on JPUE simulation verified the correctness of the code. The comparison with existing 3D models further verified the correctness of our code. The dynamic evolution of the plume agrees with mechanism analysis of plume development, which convinced us that the physics model that takes major factors in plume development into accout.

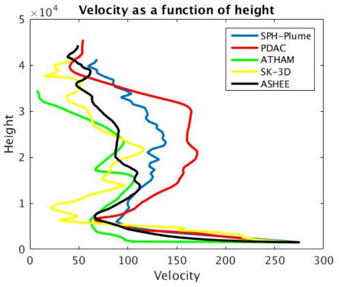
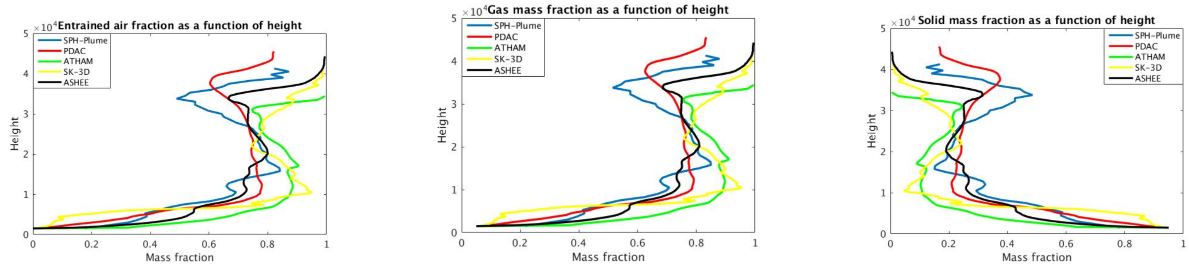


Figure 15: Velocity of the strong plume without wind after reaching its top height



(a) The mass fraction of air entrained into the plume

(b) The mass fraction of gas (air+vapor) in the plume

(c) The mass fraction of solid in the plume

Figure 16: The mass fraction of entrained air, gas, and solid as a function of height

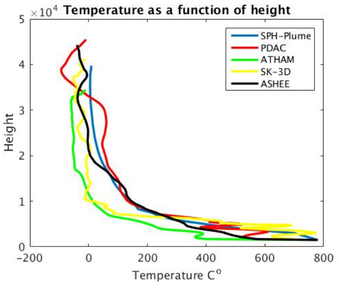


Figure 17: Temperature as a function of height

6 Conclusion and discussion

A new plume model was developed based on SPH method. Advanced numerical techniques in SPH were exploited and talored for this model. High performance computing was used to mitigate confliction between accuracy (depends on comprehensiveness of the model, resolution, order of accuracy of numerical methods, shceme for time upgrading ...) and simulation time (depends on, comprehensiveness, resolution, order of accuracy of numerical methods, shceme for time upgrading ... and computational techniques). The correctness of the code and model was verified and validated by a series of test simulation.
Currently existing 3D models focus on certain aspect of the volcano plume (PDAC on pyroclastic flow, ATHAM on microphysics, and SK-3D on entrainment with higher accuracy and higer oder of accuracy) and hence, naturely, different assumptions were made in these models. However, these different aspects of volcano plume are not independent and actually are coupled. In addtion, there is no absolute boundary to determine which kind of hazard is dominant in certain eruption. So it is necessary to simulate all associated hazards in one model. Actually, effort has already been put on developing more comprehensive plume models. For example, a large-particle module (LPM) was added to ATHAM to track the paths of rocky particles (pyroclasts or tephra) within the plume and predicts where these particles fall[22]. We were motivated by such a tendency of plume modeling to choose SPH as our numerical tool. As mentioned in the previous section, SPH method has very good extension features and adding of new physics requires tiny modification on the code compared with mesh-based methods. The last but not the least, the dramatisc development of computational power makes establishing of a comprehensvie model feasible. Even though right now, current computational capacity does not allow us to have too comprehensive model, the easy-extension feature of SPH makes it feasible to keep adding new physics into the model when necessary. What presented in this paper is actually our initial effort and results towards developing a first priciple based plume model with comprehensive physics, adopting proper numerical tools and power of high performance computing. More advanced numerical techniques, such as adaptive particle size, Godnuv-SPH, semi-explicit time advancing scheme and better data management strategies and algorithms are on our list to exploit.

References

[1] Markus Becker and Matthias Teschner. Weakly compressible sph for free surface flows. In *Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 209–217. Eurographics Association, 2007.

[2] Rupak Biswas and Leonid Oliker. Experiments with repartitioning and load balancing adaptive meshes. In *Grid Generation and Adaptive Algorithms*, pages 89–111. Springer, 1999.

[3] M Bursik. Effect of wind on the rise height of volcanic plumes. *Geophys. Res. Lett*, 28(18):3621–3624, 2001.

[4] Marcus I Bursik. bent: A model of plumes in crossflow, Nov 2010. URL `https://vhub.org/resources/267`.

[5] Marcus I Bursik, Alejandro Uriel Carbonara, and Sean Michael Zawicki. puffin, Jan 2013. URL `https://vhub.org/resources/114`.

[6] S Carcano, L Bonaventura, T Esposti Ongaro, and A Neri. A semi-implicit, second-order-accurate numerical model for multiphase underexpanded volcanic jets. *Geoscientific Model Development*, 6(6):1905–1924, 2013.

[7] Tuncer Cebeci. *Analysis of turbulent flows with computer programs*. Butterworth-Heinemann, 2013.

[8] JK Chen, JE Beraun, and CJ Jih. An improvement for tensile instability in smoothed particle hydrodynamics. *Computational Mechanics*, 23(4):279–287, 1999.

[9] Paul W Cleary and Joseph J Monaghan. Conduction modelling using smoothed particle hydrodynamics. *Journal of Computational Physics*, 148(1):227–264, 1999.

[10] A Costa, YJ Suzuki, M Cerminara, BJ Devenish, T Esposti Ongaro, M Herzog, AR Van Eaton, LC Denby, M Bursik, M de'Michieli Vitturi, et al. Results of the eruptive column model inter-comparison study. *Journal of Volcanology and Geothermal Research*, 2016.

[11] AJC Crespo, JM Domínguez, BD Rogers, M Gómez-Gesteira, S Longshaw, R Canelas, R Vacondio, A Barreiro, and O García-Feal. Dualsphysics: Open-source parallel cfd solver based on smoothed particle hydrodynamics (sph). *Computer Physics Communications*, 187:204–216, 2015.

[12] Paul E Dimotakis, Richard C Miake-Lye, and Dimitris A Papantoniou. Structure and dynamics of round turbulent jets. *Physics of Fluids*, 26(11):3185–3192, 1983.

[13] R.R. Draxler and G.D. Rolph. Hysplit (hybrid single-particle lagrangian integrated trajectory) model access via noaa arl ready website (). noaa air resources laboratory, silver spring, 2015.

[14] Angela Ferrari, Michael Dumbser, Eleuterio F Toro, and Aronne Armanini. A new 3d parallel sph scheme for free surface flows. *Computers & Fluids*, 38(6):1203–1217, 2009.

[15] William K George, Ronald L Alpert, and Francesco Tamanini. Turbulence measurements in an axisymmetric buoyant plume. *International Journal of Heat and Mass Transfer*, 20(11):1145–1154, 1977.

[16] Robert A Gingold and Joseph J Monaghan. Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly notices of the royal astronomical society*, 181(3):375–389, 1977.

[17] Michael Herzog, Hans-F Graf, Christiane Textor, and Josef M Oberhuber. The effect of phase changes of water on the development of volcanic plumes. *Journal of Volcanology and Geothermal Research*, 87(1): 55–74, 1998.

[18] Michael Herzog, Josef M Oberhuber, and Hans-F Graf. A prognostic turbulence scheme for the nonhydrostatic plume model atham. *Journal of the atmospheric sciences*, 60(22):2783–2796, 2003.

[19] Darryl D Holm. Fluctuation effects on 3d lagrangian mean and eulerian mean fluid motion. *Physica D: Nonlinear Phenomena*, 133(1):215–269, 1999.

[20] Reza Issa. Numerical assessment of the smoothed particle hydrodynamics gridless method for incompressible flows and its extension to turbulent flows. 2005.

[21] William M Kays. Turbulent prandtl number—where are we? *Journal of Heat Transfer*, 116(2):284–295, 1994.

[22] Shannon Kobs. *Modeling particle motion and near-vent deposition in explosive volcanic eruptions*. State University of New York at Buffalo, 2009.

[23] Takehiro Koyaguchi and Andrew W Woods. On the formation of eruption columns following explosive mixing of magma and surface-water. *Journal of Geophysical Research: Solid Earth (1978–2012)*, 101(B3): 5561–5574, 1996.

[24] Takehiro Koyaguchi, Kiyokatsu Ochiai, and Yujiro J Suzuki. The effect of intensity of turbulence in umbrella cloud on tephra dispersion during explosive volcanic eruptions: Experimental and numerical approaches. *Journal of Volcanology and Geothermal Research*, 186(1):68–78, 2009.

[25] Dinesh Kumar, Abani K Patra, E Bruce Pitman, and H Chi. Parallel godunov smoothed particle hydrodynamics (sph) with improved treatment of boundary conditions and an application to granular flows. *Computer Physics Communications*, 184(10):2277–2286, 2013.

[26] Bärbel Langmann, Michael Herzog, and Hans-F Graf. Radiative forcing of sulfate aerosols as determined by a regional circulation-chemistry transport model. *Journal of Aerosol Science*, 28:S389–S390, 1997.

[27] EJ List. Turbulent jets and plumes. *Annual review of fluid mechanics*, 14(1):189–212, 1982.

[28] John P Lockwood and Richard W Hazlett. *Volcanoes: global perspectives*. John Wiley & Sons, 2013.

[29] Leon B Lucy. A numerical approach to the testing of the fission hypothesis. *The astronomical journal*, 82: 1013–1024, 1977.

[30] Larry G Mastin. A user-friendly one-dimensional model for wet volcanic plumes. *Geochemistry, Geophysics, Geosystems*, 8(3), 2007.

[31] Larry G Mastin. Plumeria 2.3.1, Dec 2011. URL `https://vhub.org/resources/1194`.

[32] JJ Monaghan. Sph compressible turbulence. *Monthly Notices of the Royal Astronomical Society*, 335(3): 843–852, 2002.

[33] JJ Monaghan. Smoothed particle hydrodynamics. *Reports on progress in physics*, 68(8):1703, 2005.

[34] JJ Monaghan. Smoothed particle hydrodynamics and its diverse applications. *Annual Review of Fluid Mechanics*, 44:323–346, 2012.

[35] JJ Monaghan and A Kocharyan. Sph simulation of multi-phase flow. *Computer Physics Communications*, 87(1):225–235, 1995.

[36] Joe J Monaghan. Smoothed particle hydrodynamics. *Annual review of astronomy and astrophysics*, 30: 543–574, 1992.

[37] Joe J Monaghan. A turbulence model for smoothed particle hydrodynamics. *European Journal of Mechanics-B/Fluids*, 30(4):360–370, 2011.

[38] Joe J Monaghan and Jules B Kajtar. Sph particle boundary forces for arbitrary boundaries. *Computer Physics Communications*, 180(10):1811–1820, 2009.

[39] Joseph J Monaghan and John C Lattanzio. A refined particle method for astrophysical problems. *Astronomy and astrophysics*, 149:135–143, 1985.

[40] BR Morton, Geoffrey Taylor, and JS Turner. Turbulent gravitational convection from maintained and instantaneous sources. In *Proceedings of the Royal Society of London A: Mathematical, Physical and Engineering Sciences*, volume 234, pages 1–23. The Royal Society, 1956.

[41] Augusto Neri, Tomaso Esposti Ongaro, Giovanni Macedonio, and Dimitri Gidaspow. Multiparticle simulation of collapsing volcanic columns and pyroclastic flow. *Journal of Geophysical Research: Solid Earth (1978–2012)*, 108(B4), 2003.

[42] Josef M Oberhuber, Michael Herzog, Hans-F Graf, and Karsten Schwanke. Volcanic plume simulation on large scales. *Journal of Volcanology and Geothermal Research*, 87(1):29–53, 1998.

[43] T Esposti Ongaro, Carlo Cavazzoni, Giovanni Erbacci, Augusto Neri, and Maria-Vittoria Salvetti. A parallel multiphase flow code for the 3d simulation of explosive volcanic eruptions. *Parallel Computing*, 33 (7):541–560, 2007.

[44] Panos N Papanicolaou and E John List. Investigations of round vertical turbulent buoyant jets. *Journal of Fluid Mechanics*, 195:341–391, 1988.

[45] AK Patra and DW Kim. Efficient mesh partitioning for adaptive hp finite element meshes. In *In International Conference on Domain Decomposition Methods*. Citeseer, 1999.

[46] Daniel J Price. Smoothed particle hydrodynamics and magnetohydrodynamics. *Journal of Computational Physics*, 231(3):759–794, 2012.

[47] Stephan Rosswog. Astrophysical smooth particle hydrodynamics. *New Astronomy Reviews*, 53(4):78–104, 2009.

[48] Christopher Rumsey. Implementing turbulence models into the compressible rans equations, Agu 2014. URL `http://turbmodels.larc.nasa.gov/implementrans.html`.

[49] Mark Settle. Volcanic eruption clouds and the thermal power output of explosive eruptions. *Journal of Volcanology and Geothermal Research*, 3(3):309–324, 1978.

[50] Robert Stephen John Sparks, MI Bursik, SN Carey, Jennifer Gilbert, LS Glaze, H Sigurdsson, and AW Woods. *Volcanic plumes*. Wiley, 1997.

[51] YJ Suzuki and T Koyaguchi. A three-dimensional numerical simulation of spreading umbrella clouds. *Journal of Geophysical Research: Solid Earth (1978–2012)*, 114(B3), 2009.

[52] YJ Suzuki and T Koyaguchi. Numerical determination of the efficiency of entrainment in volcanic eruption columns. *Geophysical Research Letters*, 37(5), 2010.

[53] Yujiro J Suzuki and Takehiro Koyaguchi. 3d numerical simulation of volcanic eruption clouds during the 2011 shinmoe-dake eruptions. *Earth, Planets and Space*, 65(6):581–589, 2013.

[54] Yujiro J Suzuki, Takehiro Koyaguchi, Masaki Ogawa, and Izumi Hachisu. A numerical study of turbulent mixing in eruption clouds using a three-dimensional fluid dynamics model. *Journal of Geophysical Research: Solid Earth (1978–2012)*, 110(B8), 2005.

[55] C Textor, Hans-F Graf, Michael Herzog, Josef M Oberhuber, William I Rose, and GGJ Ernst. Volcanic particle aggregation in explosive eruption columns. part ii: Numerical experiments. *Journal of Volcanology and Geothermal Research*, 150(4):378–394, 2006.

[56] Christiane Textor, Hans-F Graf, Michael Herzog, and JM Oberhuber. Injection of gases into the stratosphere by explosive volcanic eruptions. *Journal of Geophysical Research: Atmospheres (1984–2012)*, 108 (D19), 2003.

[57] Christiane Textor, Hans-F Graf, Michael Herzog, Josef M Oberhuber, William I Rose, and Gerald GJ Ernst. Volcanic particle aggregation in explosive eruption columns. part i: Parameterization of the microphysics of hydrometeors and ash. *Journal of Volcanology and Geothermal Research*, 150(4):359–377, 2006.

[58] J Trentmann, MO Andreae, H-F Graf, PV Hobbs, RD Ottmar, and T Trautmann. Simulation of a biomass-burning plume: Comparison of model results with observations. *Journal of Geophysical Research: Atmospheres (1984–2012)*, 107(D2):AAC–5, 2002.

[59] Alexa R Van Eaton, Larry G Mastin, Michael Herzog, Hans F Schwaiger, David J Schneider, Kristi L Wallace, and Amanda B Clarke. Hail formation triggers rapid ash aggregation in volcanic plumes. *Nature communications*, 6, 2015.

[60] Damien Violeau and Reza Issa. Numerical modelling of complex turbulent free-surface flows with the sph method: an overview. *International Journal for Numerical Methods in Fluids*, 53:277–304, 2007.

[61] Mattia de' Michieli Vitturi. Plumemom, Mar 2015. URL `https://vhub.org/resources/3541`.

[62] L Wilson, RSJ Sparks, TC Huang, and ND Watkins. The control of volcanic column heights by eruption energetics and dynamics. *Journal of Geophysical Research: Solid Earth (1978–2012)*, 83(B4):1829–1836, 1978.

[63] Andrew W Woods. Moist convection and the injection of volcanic ash into the atmosphere. *Journal of Geophysical Research: Solid Earth (1978–2012)*, 98(B10):17627–17636, 1993.

[64] Andrew W Woods and Sally M Bower. The decompression of volcanic jets in a crater during explosive volcanic eruptions. *Earth and Planetary Science Letters*, 131(3):189–205, 1995.

[65] Andrew W Woods and Marcus I Bursik. Particle fallout, thermal disequilibrium and volcanic plumes. *Bulletin of Volcanology*, 53(7):559–570, 1991.

[66] AW Woods. The fluid dynamics and thermodynamics of eruption columns. *Bulletin of Volcanology*, 50(3): 169–193, 1988.