Dear Editor,

Please find below the detailed answers to the referee comments including the resulting changes made to our revised manuscript, as already publicly uploaded as final author comment to the discussion of our manuscript. We additionally appended the latex-diff of the manuscript published by GMDD and the revised version.

Best regards,
Abani Patra (also on behalf of all co-authors)

Authors' reply to the comments by M. de Michieli Vitturi (Referee #1) of the manuscript gmd-2017-119

Plume-SPH 1.0: A three-dimensional, dusty-gas volcanic plume model based on smoothed particle hydrodynamics

by Zhixuan Cao et al.

The comments by the reviewer are recited in italics, followed by our reply in upright font. Revision are highlighted with blue. Equation, section, and figure numbers that in our response are corresponding to these in the original version of GMD discussion paper if not specified.

General Comments

This paper presents an initial effort towards developing a plume model with comprehensive physics based on the smoothed particles hydrodynamics method. To my knowledge, this is the first numerical model of volcanic plumes adopting this technique and this is a good addition to the existing models. While the model, the derivation of the discretized equations and the computational techniques are satisfactory for a reader expert in SPH, a few more details would help in the comprehension other readers. The scientific content of the paper could be improved by a better and extended description of the applications. Overall, the manuscript represent a substantial contribution to modelling science within the scope of Geoscientific Model Development and it is suitable for publication after some improvements and corrections are made. These are described below.

We would like to thank the reviewer for carefully reading our manuscript and giving positive review and constructive remarks. We have revised the manuscript as shown in the supplemental PDF file, and we hope that we have dealt with all suggestions in an adequate manner. The revised manuscript is also attached.

On behalf of all co-authors.

The following are our responses to reviewer's specific comments. Modifications are made accordingly in the manuscript.

Specific comments
Abstract

Line 8. What does dynamic and thermodynamic equilibrium with surrounding air mean? No relative velocity? Same temperature? I think it is an equilibrium between volcanic gas and particles.

Yes, dynamic equilibrium means same velocity while thermodynamic equilibrium means same temperature. We treat erupted material (includes volcanic gas and particles) as one phase, so assumption of dynamic equilibrium and thermodynamic equilibrium are made for erupted material. Such assumption is made implicitly when we assume that erupted material is well mixed and behaves like a single phase fluid. What we intended to say by "dynamic and thermodynamic equilibrium between air and erupted material" is actually "dynamic equilibrium and thermodynamic equilibrium between erupted material and air that is entrained into the plume". That is to say, we assume that surrounding air and erupted material will reach an equilibrium status (same velocity and temperature) as soon as they get mixed.

We realize that the original statement is misleading and modified it as "dynamic equilibrium and thermodynamic equilibrium between erupted material and air that is entrained into the plume". We thank the reviewer for pointing this out.

Line 23. The model is not really compared with the top height of the Pinatubo eruption, because atmospheric conditions are changed from real ones (no wind in imposed, weak vs strong in Costa et al 2016); in addition, MER is fixed. So, it is more a comparison with results from other 3D models, for an eruption with the same MER estimated for Pinatubo 1991.

Thank you for your clarification on this. We revised our original statement "The model is verified by comparing velocity and concentration distribution along the central axis and on the transverse cross with experimental results of JPUE (jet or plume that is ejected from a nozzle into a uniform environment) and the top height of the Pinatubo eruption of 15 June 1991." to "The code is first verified by 1D shock tube tests, then by comparing velocity and concentration distribution along the central axis and on the transverse cross with experimental results of JPUE (jet or plume that is ejected from a nozzle into a uniform environment). Profiles of

several integrated variables are compared with those calculated by existing 3D plume models for an eruption with the same MER (mass eruption rate) estimated for the Pinatubo eruption of June 15 1991. Our results are consistent with existing 3D plume models."

Section 1.3

Page 4, lines 23-25. It is not clear to me why interface tracking or interface capturing are mentioned here for mesh based method. I think all the existing 3D Eulerian models (based on mesh based method) do not model the interface between the plume and the atmosphere. While I understand that this represent a big advantage for other applications of SPH, for example for dam-break problems, because it allows to solve only for the flow region and not for the surrounding, for a volcanic plume mixing between plume and air is important and I don't really see a clear interface between them. Please clarify and expand the point.

We first thank the reviewer for asking this question. This question pushed us to do more investigation on "interface construction" in mesh based methods. It is critical to capture or track the interface during simulation for immiscible flow since the interfaces are the boundaries separating different fluids. This is the major motivation for using interface tracking and interface capturing method in multi-phase flows. However, this is not the case for volcanic plume modeling as "fluids" in our plume model are miscible.

However, the literature reveals that clear transient interfaces do exist in miscible flow. There are experiments [11] showing a sharp interface between air and plume. Crimaldi and Koseff [7] present another experiment with higher resolution that shows an even more detailed and more complicated interface. We postulate that it is through the interface (or boundary) that mixing happens. Analogously, Jacobson et al. [9] studied detailed mixing mechanism at a flat interface.

However, quantifying these mixing processes in real implementation is challenging because of the scale disparity between the large-scale fluid motion and the diffusion processes on interface that ultimately lead to mixing. The energy transfer between these scales occurs through turbulent motion, created either by fluid instabilities or by breaking internal waves. For shearing flow, like volcanic plume, the fluid instabilities dominated the mixing process. Ideally, one would like to be able to include the effects of mixing on the large scale dynamics without resolving the detailed interface structure and dynamics of turbulence to reduce computational cost. Such a strategy was also used in all other mesh-based 3D plume models (though different turbulence closures were adopted by different models).

More importantly, thanks to this reexamination of our approach we now have a deeper understanding of why we need a turbulence model in plume modeling. Our original motivation for using a turbulence model was simply based on the following logic: The Reynolds number is large enough hence the flow is turbulent, we can not resolve all turbulent exchange at all different scales with a coarse resolution, so we need turbulence model to resolve sub-particle scale turbulence. Now we can see the necessity of including a turbulence model to resolve mixing which happens at a much smaller scale.

By resolving sub-particle scale (or sub-mesh scale) turbulent mixing with turbulence closure, relative coarse resolution is used in all 3D plume models (both particle based model and mesh-based models). Such solution is a trade-off between computational accuracy and computational cost. And in this case, interface might not be clearly observed in simulation due to coarse resolution.

Interface construction will become necessary and important when we attempt to include the effects of mixing by resolving the detailed interface structure and dynamics of turbulence. This option is not attractive at current stage of plume modeling as turbulence closure can give an acceptable accuracy with affordable computational cost.

Remedies of standard SPH is needed if detailed mixing processes need to be resolved, for example [5, 16]. Remedies, such as adding of surface tension [2], are needed for tracking interface while simulating immiscible flow using SPH. Even though, as a Lagrangian method, interface constructing is explicit through capturing of the locations of the particles and much simpler than Eulerian Methods.

Modification in the manuscripts: 1) Since the exact location of interface is less important for plume modeling when turbulence model is adopted, we changed this paragraph to: "Interface tracking is explicit in SPH through capturing of the locations of the particles. Less numerical effort is required for interface construction when we attempt to include the effects of mixing by resolving the detailed interface structure and dynamics of turbulence." and put it as the last item among SPH features. 2) Added a short paragraph (Basically the paragraph above of this response) at the beginning of section 3.8 explaining the motivation for using a turbulence model. 3) Changed the title of section 3.7 to "Mass fraction update". We re-organized this section and focused it more on mass fraction updating. We make a brief remark on interface construction using SPH and mesh-based methods at the end of this section.

Section 3. SPH method

Sometime equation are referenced before they have been introduced. See for example lines 23 and 24 at page 10 where there is a reference to Eqs. 34 and 35, which are in introduced in the next subsection. Again, at page 10,

sentence at lines 25-26 should be referred to the discretised version of the equations, while discretisation is presented later in the section. Page 11, line 1. It is written region of compact support before having stated the properties of the weighting function. Maybe add an equation that define this property: $w(x - xb, h) = 0$ if $|x - xb| > kh$ for some k, i.e. the support is proportional to the smoothing length. A suggestion. . . Probably the issues raised above could be solved moving the content of section 3.2 to section 3.1, after equation 26.

As suggested by the reviewer, we reorganized the order of equations and merged section 3.1 and section 3.2 into one section. We also moved the section for "Artificial viscosity" (section 3.4 in original version) before the section on "governing equation discretization" (section 3.3 in original version).

Section 3.3

Page 12, line 22. The sink term has been added to eq. 42, representing the discretized form of mixture equation. If the term is associated with a phase change, it would be better to add it in the equation for a phase and not for the mixture with density ρ.

First of all we would like to clarify that the purpose of showing Eq. (42)   (44) is to show that only slight modifications is needed in discretized governing equations when new physics is considered. In addition, we do not have separate equation for mass conservation of volcanic gas (due to the assumption that erupted material behaves like a single phase).

We completely agree with the reviewer that terms for sink associated with phase change should be in the equation of the corresponding phase. For future extensions of current model, such as the example mentioned by the reviewer, considering phase change of volcanic gas, such sink term should be added to mass conservation equation of volcanic gas. To make the article more readable, we avoid discussing new physics models, which are easily understandable for readers working on plume modeling but might cause confusion for readers without such background.

Page 12, lines 24-25. The drag force term should show up only when dynamics disequilibrium between different phases is considered. In this case is the summation in the last term of eq. 43 only extended to particles of a different phase?

The purpose of showing Eq. (43) is to show that only slight modifications is needed in discretized governing equations when new physics (drag force term) is considered. The drag force term is not considered in current model, as we treat erupted material as one phase and assume immediate dynamic equilibrium between entrained air and erupted material. So dynamics disequilibrium between different phases (for example, solid particles of different size and volcanic gas) are currently not considered in the model.

If such disequilibrium needs to be considered in future extension of current model, drag force effect should be considered. The form used to represent drag force effects will depend on new assumptions and the new physics model. For example, if the physics model is a four phases model, one phase for atmosphere air, one phase for volcanic gas, one phase for large size solid particles and another for small size ash (maybe immediate equilibrium assumption between air and gas should be made). Drag effects between gas (air) and particles should be considered. As for the small size ash, we can assume they behave like single phase fluid (possibly not a reasonable assumption) then there should not drag force effect within the ash phase. Otherwise. drag effect within fine ash phase need to be considered.

We changed the original sentence "The drag force term should show up only when dynamics disequilibrium between different phases is considered" to "The drag force term should show up when dynamics disequilibrium between different phases is considered. By deleting the word "only", we changed original necessary and sufficient statement to a sufficient statement.

Section 3.4

Page 13, line 5. Please define particle disorder.
We replaced "particle disorder" with "irregular distribution of particles"

Section 3.5

Page 14, line 5. how sound speed of a particle is defined for the gas-solid mixture? Please add an equation
It is calculated by $c = (\gamma_m * \frac{p}{\rho})^{0.5}$ – we added an equation in our manuscript.

Page 14, line 5. What is the order of magnitude of the time step for the tests presented, with a fully explicit scheme?
It depends on many factors: the value of smoothing length, particle mass, CFL number et al.. Time step also changes during simulation. For the simulation of Pinatubo eruption in this paper ($CFL = 0.2$ was used, smoothing length is $170m$), The time step is around 0.01s.

Section 3.6

Page 14, line 8. The classical SPH method was known to suffer from tensile instability and boundary deficiency. Please describe there problems with some example, in order to help the readers not expert in SPH.

We added explanations and references on tensile instability: "Tests of the standard SPH method indicate an instability in the tensile regime, while the calculations are stable in compression. A simple test calculation exhibiting the instability involves a body which is subject to an uniform initial stress, either compressive or tensile. If the initial stress is tensile, a very small velocity perturbation on a single particle can lead to particles clumping together, forming large voids and seriously corrupting density distribution. But if the initial stress is compressive, the small velocity perturbation on a single particle can not lead to any changes in particle distribution. See paper by Swegle et al. [14] for more details."

Page 14, line 14. Equation (55) implies that .... = 1 Why? It is not equation 55 which implies the term is equal to 1, but the properties of the kernel. Furthermore, this is true for the integral (not discretised) formulation, and only for points far away from the boundary (see Chen et al., 1999).

The sentence is misleading. Sorry for that. Here is the logic: The new approximation of function A, Eq. (55), should be consistent with original formulation, Eq. (28). Compare Eq. (55) and Eq. (28), if the denominator is 1, then these two equations will be the same. so it implies that the denominator in Eq. (55) should be close to 1. We completely agree with you that the normalization condition originally comes from property of kernel function. By the statement "implies that ... = 1" we were actually trying to show a consistent connection between this new approximation formulation and properties of the kernel function. In other words, in the derivation based on Taylor series expansion, we never take the property of the kernel into account but finally reach to an equation that is consistent with a property of kernel function.

In addition, the summation (Eq. (28)) is usually a good approximation of integration in the area far away from boundaries. Hence the denominator in Eq. (55) is close to 1. it has ignorable effect on evaluating of A_a. That is to say, Eq. (55) and Eq. (28) are the same for particles far away from boundaries. But for particles close to boundaries, the denominator will definitely not be close to 1. For these cases, the non-unit denominator helps normalize the summation and overcomes boundary deficiency.

We changed the original sentence "Equation (55) implies that $\sum_b m_b \rho_b w(x - x_b, h) = 1$, which can be viewed as the approximation form of Eq. (34)." to "Notice that the denominator in Eq. (55) is actually summation approximation of Eq. (34). That is to say, Eq. (55) and Eq. (28) are the same for particles far away from boundaries as the denominator in Eq. (55) becomes 1 in that case."

Section 3.7

Page 14, line 21-22. Numerical simulation of multiphase flows is usually difficult due to the existence of complex evolving interfaces between phases. This is true when the different phases are immiscible. But in the application investigated, the volcanic plume, phases are not immiscible and mixing is very important. So, there is no need to track the interface between phases as in Eulerian grid-based numerical methods. Conversely, standard formulations of SPH cannot resolve fluid mixing and instabilities at flow boundaries. It should be discussed in the paper if, and how, this is has been addressed in the model presented. Please take a look at: J. I. Read, T. Hayfield, O. Agertz; Resolving mixing in smoothed particle hydrodynamics, Monthly Notices of the Royal Astronomical Society, Volume 405, Issue 3, 1 July 2010, Pages 15131530

As we have mentioned in our response to reviewer's previous question regarding interface capturing/tracking in SPH, interface construction will become necessary and important when we attempt to include the effects of mixing by resolving the detailed interface structure and dynamics of turbulence. This option is not attractive at current stage of plume modeling as turbulence closure can give an acceptable accuracy with affordable computational cost.

The following is our response to reviewer's concern on how did we handle the issue of classical SPH: "Classical SPH has problems correctly integrating fluid instabilities and mixing at boundaries". We will call it "mixing challenge" in later paragraph for short.

We were not aware of Read's paper previously. But we adopted several techniques (corrected formulation for tensile instability issue, smaller artificial viscosity coefficients, turbulence model with heat transfer considered) that probably have similar effect as the methods proposed in Read's paper. In Read's paper, one of his remedies of SPH is based on fixing the clumping instability which is actually tensile instability. We just adopted a different method (Chen's method) to handle this issue.

In addition, before we integrated the turbulence model, we got a "mushroom like" plume and almost no mixing happened (exactly due to the "mixing challenge"). The simulation results became much more realistic after adding the turbulence model. So the turbulence model helped us to resolve sub-particle scale mixing. As has been mentioned, it is a common practice in CFD to adopt turbulence closure to resolve instabilities (which dominate the mixing process for shearing flow).

Agertz et al. [1] did a Kelvin-Helmholtz (KH) test showing that using smaller artificial viscosity coefficient

can help get more mixing (Fig. 15 and discussion in Section 6.1 in his paper). In our model we also use a much smaller artificial viscosity (See the last paragraph in section 3.4, we use $\alpha = 0.3$, while traditionally α is taken as 1, our original motivation was to avoid excessive artificial viscosity).

Price [12] believes that the mixing issue is due to the fact that entropy is discontinuous at the boundaries while density is continuous. He found that adding thermal conductivity at boundaries can improve mixing in SPH as thermal conductivity can smooth the entropy. The thermal conductivity (heat transfer) due to turbulence is considered in our model, what's more, the thermal conductivity coefficient is much larger at the interfaces due to large shearing effect at the interface. This can definitely help to mitigate the "mixing challenge" if Price is correct. While, we were not motivated by Price's paper to adding a turbulent heat transfer term into our model – we were motivatd by the need to have a turbulent heat transfer term for compressible flow in which energy conservation equation is coupled with momentum conservation and mass conservation equation. It appears we were lucky and "adopted" Price's method. Wadsley et al. [15] and Ritchie and Thomas [13] made similar arguments as Price. Borgani et al. [3] believes that the "mixing challenge" of SPH is due to its poor ability of capturing contact discontinuity. They show that using GSPH can avoid appearance of spurious pressure force and help to to follow the Kelvin instability. Since people still have different opinions on the sources of "mixing issue", it would be interesting to see the connections between these different opinions. But such work is obviously out of our scope in this paper. These remedies of traditional SPH we adopted and extended in our model helped us to overcome or at least relieve the "mixing challenge".

At the end, we would thank reviewer for asking this critical question and pointing out other possible ways for remedying traditional SPH.

Section 3.8.1

Page 16, line 10. For a reader like me who is more familiar with Eulerian and mesh based formulation, it would be good to state at the beginning that, differently from RANS (Reynolds Averaged Navier-Stokes) equations, which are time-averaged equations of motion, here the Lagrangian average is in space and not in time.

Thank you for your suggestion for improving readability of the article. We totally agree with you and have made the revision as you suggested.

Section 3.8.2

Page 20, line 8. Is this equation the definition of F_{ab} or a property? If this is the definition, where is the argument of the function F?

It is the definition of F_{ab}. By the way, $F_{ab}(h)$ is short for $F(\mathbf{x}_a - \mathbf{x}_b, h)$, we added clarification in our revised manuscript.

Section 3.9.1 Wall boundary condition

Please discuss the limitation of this approach when a complex topography is considered. In particular, how could you deal with convex geometries?

In our current model, the ground is assumed to be a flat ground. Since the topography is pretty simple, we did not experience any trouble. It has been shown in some other papers that the approach for imposing wall boundary conditions also works as well for more complex topography [10]. One potential issue for complicated topography might be deployment of wall ghost particles. For flat boundary, we simply deploy several layers of ghost particles with equal interval and each ghost particle has the same mass. It has not been investigated yet how different deployment of wall ghost particle will influence the accuracy of imposed boundary conditions. A more basic question is how many degree of freedom should we allow for ghost particle deployment: only enough to allow changing of particle position or enough to allow using different particle mass size?

For non-collapsed plume, the flat ground assumption made in our current model should be good enough. For other phenomena, such as PDCs, a more realistic topography is required. As we mentioned in the manuscripts, current plume models are still not comprehensive enough. Each model has its own problem of interest and assumptions were made accordingly. Fortunately, we have been seeing efforts on making these models to be more general. As the first trial of using SPH in plume modeling, we prefer to narrow down to the problem of interest and focus on more fundamental aspects.

We appreciate such discussion proposed by the reviewer which are constructive for future model improvement. We added the following sentence "In our current model, the ground is assumed to be flat. For more complicated topography, it has been shown in other work [10] that this method works as well. We do have concern regarding potential limitation of this method of deployment of ghost particles for more complicated boundaries in three dimensions. Fortunately, current model does not involve complicated wall boundary."

Section 3.9.3

Keeping the pressure constant at atmospheric boundary can represent a problem when particles exit from the domain and choosing a larger computational domain cannot be the solution for long simulations. A better way to implement the boundary conditions at the exit would be to impose a condition on the total pressure ($p_\rho - 0.5\rho u^2$, where $p_{rho} = p - \rho g h$). This allow to have pressure changes at the boundaries associated with outflow.

We have two questions regarding your suggestion: 1) When we assign pressure value to these static pressure ghost particles, the decrease of pressure due to increase of height (h) has already been taken into account. That is to say, pressure of our pressure ghost particles is consistent with meteorological data when establishing the initial condition. Should we still deduce $\rho g h$ from pressure? 2) Should the total pressure equals to a pressure **plus** kinetic energy, or minus kinetic energy? Similar questions arise for definition of p_ρ. Should $\rho g h$ be added to or deduced from pressure? Because total pressure in other fields, such as thermal dynamic analysis of flowing process in engines is defined as pressure plus kinetic energy.

In our current model, pressure ghost particles are assumed to be stationary. It does not make a difference whether we use pressure or total pressure since velocity of pressure ghost particles are zero. The next important improvement of our current model under planning is taking the effect of wind field into account. In that case, the pressure ghost particles will not be static any more. The idea of using total pressure can be naturally implemented for that case. Thank you very much for pointing out a potential way for improving pressure outlet boundary condition.

Section 4. Verification and validation

deals with the numerical resolution of the equations of the model, not with the agreement between the model and reality. It checks no code errors have been introduced in the code and one way to do it is a comparison with analytical solutions. I think it is better to simply rename the section "Applications or Results

Since JPUE is not the phenomena that we aim to model. Simulation of JPUE should not be validation (validation is to make sure simulations are consistent with reality for the application of interest). Meanwhile, we compared our results against experimental results. We might call such comparison (simulation against measurements) as verification in a more general sense (?).

We added simulation results of shock tube tests which were compared against analytic solutions in a separate subsection. By adding such "pure" verification tests, we keep the title of this section unchanged.

Section 4.1

Page 25, line 24. Please look also at the results from this paper: Ezzamel, Adam, Pietro Salizzoni, and Gary R. Hunt. "Dynamical variability of axisymmetric buoyant plumes." Journal of Fluid Mechanics 765 (2015): 576-611.

Thank you for pointing out a source for accessing more recent experimental results. We have added the paper to the references. Ezzamel et al. [8] did not provide coefficients of their fit equations. So we did not add plots corresponding to Ezzamel's results in our figures.

Page 26. Please put figure 6 and 7 together as to subplots Page 26, line 12. The expression for the Gaussian profile should be written as an equation (on a single line and numbered) and the terms should be defined after it is introduced, before writing the expression for the straight line. Page 27. Please put figure 8 and 9 together as to subplots Page 28, line 1. Please write the straight line expression as an equation on a single line.

We revised manuscripts as your suggestions for Fig. 6 - Fig. 9

Page 28, lines 8-9. results, a small disparity in both velocity and concentration are observed near the boundary of the jet. Looking at figure 10, it seems that near the boundary of the jet there are regions without particles, and this could be the reason of the differences with experimental results. The emptying of particles in region with steep density gradients is described in section 3.4 of this paper: Ritchie, Benedict W., and Peter A. Thomas. "Multiphase smoothed-particle hydrodynamics." Monthly Notices of the Royal Astronomical Society 323.3 (2001): 743-756. Look for example at their figure 9.

Thank you very much for pointing this out. The low particle density (or the emptying of particles as what you call it) near the vent should be at least one reason of the disparity. We added an explanation of the disparity based on that paper.

Ritchie's paper proposed two remedies of standard SPH: 1) Keep a density-weighted quantity constant. This might prevent the smoothing length changing in a proper way. One potential issue with such method is that there is no guarantee of enough number of neighbor particles when cavity forms in the fluid field. Probably his method could help us cure the disparity but might also introduce other problems. 2) The second remedy he proposed is assuming a constant pressure within the kernel (not sure whether such assumption can reflect real physics or not). The density is then updated based on equation of state. One good property of density updating equation in standard SPH adopted in our model is that the total mass is strictly conserved. All other alternative methods for density updating, for example, density updating based on mass conservation

PDE, are not able to guarantee strict conservation of total mass. I did not see any investigation in that paper about how well does his method conserve mass.

Quite a lot variations of SPH has been proposed. Each has advantages and disadvantages. Making a proper trade-off is one of the challenging things for implementing SPH in simulations of complicated phenomena. And this is actually where we spent a lot of effort. Our choices were made either based on our preference (for example, we prefer strict conservation of mass, momentum and energy) or relevance of test simulations shown in these papers.

We thank the reviewer for proposing such discussion.

Page 28. End of section. The paper presents also details about the numerical implementation and the parallelisation of the code, so it would be interesting to have more info for this test about computational cost, number of particles, core/cpu used

The performance benchmark results are reported in 3.10. More Details has been published in another paper [4]. We added the reference in section 3.10.

The performance benchmarking should be done in non-shared mode of the compute nodes, while computational nodes are usually shared among users in regular use. The difference between exclusive testing and no-exclusive testing might be large. That's why we put performance testing in a separate section and did not mention any thing, like total simulation time, in this section.

Section 4.2.1

Page 29, lines 7-8. From Costa et al. 2016 "For the erupted particles, only two size classes were considered, representing coarse ash (Φc) and fine ash (Φf), each comprising 50 wt. & of the erupted particles " Here I cannot find any info about particles ...

We treat all erupted material including particles of different size and volcanic gas as a single phase. So our model is not able to make use of such detailed information. In our model, only the mass fraction of total solid particles is used, and dose not treat particles of different size separately. That's why we did not mention anything about detailed particle size distribution in the table for input parameters. Details about assumptions on which our model relies can be found in section 2. These assumptions are also briefly summarized in abstract. To avoid confusing readers, we added emphasis on which portion of Costa's data are used in our simulation.

Page 29, line 13. In Fig. 1B it is plotted the meteo profile where it is also shown the presence of wind. Please clarify.

We use a static pressure boundary condition (pressure ghost particles have zero velocity) and do not consider the effect of wind field. Even though wind field data and specific humidity are also provided in Fig. 1B in the paper by Costa et al. [6], we only used density, temperature and pressure and did not use wind velocity and specific humidity in our simulation. We added clarifications on this in the revised manuscripts.

Page 30. Why there are no figures showing a 2d section of the plume, as done for the previous test? It would be interesting to see a vertical section of some variables (for example instantaneous value of mixture velocity modulus) and velocity streamlines, as shown in Cerminara et al.

Visualization of the simulated plume is added, see Fig. 10 in the revised manuscript

Page 30, line 1. The averaging technique of Cerminara et al. is defined for grid-based eulerian models. It would be interesting to know the details of the implementation of the technique for the SPH code (perhaps in an appendix).

Page 30, lines 1-4. As particles distribute in a disordered manner in the space in SPH simulation results. We first project simulation results (on disordered particles) onto a pre-defined grid before doing time average and spatial integration. The project method is the basic SPH kernel based interpolation. These lines are confused, please check.

For these two questions above. We added an appendix describing the posting process in detail. Then we replaced the sentence "As particles distribute in a disordered manner in the space in SPH simulation results. We first project simulation results (on disordered particles) onto a pre-defined grid before doing time average and spatial integration. The projection method is the basic SPH kernel based interpolation" by a new sentence "As particles distribute irregularly in the space in SPH simulation results. We need to project simulation results (on irregular particles) onto a pre-defined grid before doing time average and spatial integration. See appendix A for more details of post processing."

Page 30, line 11. Eq. 95 is referenced here, so it should be written immediately after this sentence.

Yes, we adjusted the position of this equation and other two equations.

Page 32, end of section. In this section it would be good to have more details on the simulation (number of particles, computational cost, number of cores/cpu), and also to make a comparison with the other models regarding the time needed for a simulation.

Performance benchmarking of our solver is reported in section 3.10 separately. More details have been published in a separate paper. We added an reference to the paper. As the simulation of Pinatubo reported here was done on computational cluster sharing computational nodes with other users, the computational performance given by a statement like "spend XXX hours on XXX cpus" might be misleading.

To give an intuitive impression of the "computational performance" of our solver: it took around 15 days on 240 cores for the Pinatubo simulation. The smoothing length is 170m, $CLF = 0.2$, the total duration of simulation is 550 s after eruption. The size of computational domain is $[-40800, 40800] \times [-40800, 40800] \times [1500, 55000]$;

A comparison between all existing 3D models in terms of computational cost would be interesting. considering many factors could influence the total simulation time, such as resolution, size of the computational domain, duration of simulation, type of hardware on which the simulation is done, number of CPUs, etc., more careful benchmark testing are required for doing good comparison.

Technical corrections

See pdf attached. Please also note the supplement to this comment: https://www.geosci-model-dev-discuss.net/gmd-2017-119/gmd-2017-119-RC1-supplement.pdf

Other revisions related to languages and typos have been made according to reviewer's suggestions in the supplement PDF that he attached.

The only one suggestion that we did not follow is his suggestion for replacing "heat" with "positive buoyancy". on page 6 line 24. The reason for using "heat" is: Since the erupted material are hot, they keep heating up entrained air until its heat exhausts. And due to such heating up process, the plume can keep expanding and reducing its bulk density and keep a positive buoyancy. So we prefer to use "heat" here. We are definitely open and happy to discuss with reviewers about which word is more proper.

We thanks him again for his careful reading and many constructive comments.

Major adjustments of the manuscripts

Here is a summary on major adjustments made in the revised manuscript:

- In section 3, moved contents in section 3.2 right behind Eq. (26) and merged section 3.1 and 3.2 into one section. Moved section 3.4 before 3.3. Changed the tile of section 3.7 and adjusted the content in that section.

- Added a new subsection in section 4 showing some 1D shock tube tests against analytical results.

- In section 4, added 4 figures showing mass fraction and velocity of simulated volcanic plume.

- Added an Appendix on post processing of SPH simulation results.

References

[1] Oscar Agertz, Ben Moore, Joachim Stadel, Doug Potter, Francesco Miniati, Justin Read, Lucio Mayer, Artur Gawryszczak, Andrey Kravtsov, Åke Nordlund, et al. Fundamental differences between sph and grid methods. *Mon. Not. R. Astron. Soc.*, 380(3):963–978, 2007.

[2] H Elekaei Behjati, M Navvab Kashani, and MJ Biggs. Modelling of immiscible liquid-liquid systems by smoothed particle hydrodynamics. *Journal of Colloid and Interface Science*, 2017.

[3] S Borgani, G Murante, R Brunino, and S-H Cha. Hydrodynamic simulations with the godunov sph. In *Advances in Computational Astrophysics: Methods, Tools, and Outcome*, volume 453, page 259, 2012.

[4] Zhixuan Cao, Abani Patra, and Matthew Jones. Data management and volcano plume simulation with parallel sph method and dynamic halo domains. *Procedia Comput. Sci.,*, 108:786–795, 2017.

[5] Seung-Hoon Cha, Shu-Ichiro Inutsuka, and Sergei Nayakshin. Kelvin–helmholtz instabilities with godunov smoothed particle hydrodynamics. *Mon. Not. R. Astron. Soc.*, 403(3):1165–1174, 2010.

[6] A Costa, YJ Suzuki, M Cerminara, BJ Devenish, Thmaso Esposti Ongaro, M Herzog, AR Van Eaton, LC Denby, M Bursik, Mattia de' Michieli Vitturi, et al. Results of the eruptive column model intercomparison study. *J. Volcanol Geoth. Res.*, 2016.

[7] JP Crimaldi and JR Koseff. High-resolution measurements of the spatial and temporal scalar structure of a turbulent plume. *Experiments in Fluids*, 31(1):90–102, 2001.

[8] Adam Ezzamel, Pietro Salizzoni, and Gary R Hunt. Dynamical variability of axisymmetric buoyant plumes. *J. Fluid Mech.*, 765:576–611, 2015.

[9] Tivon Jacobson, Paul A Milewski, and Esteban G Tabak. Mixing closures for conservation laws in stratified flows. *Studies in Applied Mathematics*, 121(1):89–116, 2008.

[10] Dinesh Kumar, Abani K Patra, E Bruce Pitman, and H Chi. Parallel godunov smoothed particle hydro-dynamics (sph) with improved treatment of boundary conditions and an application to granular flows. *Comput. Phys. Commun.*, 184(10):2277–2286, 2013.

[11] Dimitris Papantoniou and E John List. Large-scale structure in the far field of byoyant jets. *J. Fluid Mech.*, 209:151–190, 1989.

[12] Daniel J Price. Modelling discontinuities and kelvin–helmholtz instabilities in sph. *J. Comput. Phys.*, 227 (24):10040–10057, 2008.

[13] Benedict W Ritchie and Peter A Thomas. Multiphase smoothed-particle hydrodynamics. *Mon. Not. R. Astron. Soc.*, 323(3):743–756, 2001.

[14] JW Swegle, DL Hicks, and SW Attaway. Smoothed particle hydrodynamics stability analysis. *J. Comput. Phys.*, 116(1):123–134, 1995.

[15] JW Wadsley, G Veeravalli, and HMP Couchman. On the treatment of entropy mixing in numerical cosmology. *Mon. Not. R. Astron. Soc.*, 387(1):427–438, 2008.

[16] GX Zhu, L Zou, Z Chen, AM Wang, and MB Liu. An improved sph model for multiphase flows with large density ratios. *International Journal for Numerical Methods in Fluids*.

Authors' reply to the comments by Anonymous Referee (Referee #2) of the manuscript gmd-2017-119

Plume-SPH 1.0: A three-dimensional, dusty-gas volcanic plume model based on smoothed particle hydrodynamics

by Zhixuan Cao et al.

The comments by the reviewer are recited in italics, followed by our reply in upright font. Revision are highlighted with blue. Equation, section, and figure numbers in our response are corresponding to these in the original version of GMD discussion paper.

General Comments

After reading the paper, I think the proposed method is highly detailed and the type of discretization properly described.

We would like to thank the reviewer for carefully reading our work and giving constructive comments. We have revised the manuscript as shown in the supplemental PDF file, and we hope that we have dealt with all suggestions in an adequate manner. The revised manuscript is also attached.

On behalf of all co-authors.

The following are our responses to reviewer's specific comments. Modifications are made accordingly in the revised manuscript.

Specific comments

1. Please add the recent papers, which I consider, are related to your research:

(a) Costa, A., Suzuki, Y., & Koyaguchi, T. (2018). Understanding the plume dynamics of explosive super-eruptions. Nature communications, 9(1), 654.

(b) Terray, L., Gauthier, P. J., Salerno, G., Caltabiano, T., Spina, A. L., Sellitto, P., & Briole, P. (2018). A New Degassing Model to Infer Magma Dynamics from Radioactive Disequilibria in Volcanic Plumes. Geosciences, 8(1), 27.

Thanks for pointing out some of the more recent work on volcanic plume modeling. Work by Costa et al. [2] motivates the development of more comprehensive 3D computational models like those in this paper. Reading this paper has inspired new application ideas for Plume-SPH. Citation of this paper is added in "Introduction" section to emphasize the advantage of developing more comprehensive plume model based on SPH.

We have considered coupling a volcanic plume model with a magma reservoir model. It is critical for any plume simulation to use more accurate eruption conditions, which might be obtained from a magma reservoir model. Thank you for pointing out the new degassing model to infer magma dynamics in volcanic plumes [3]. Comments regarding coupling volcanic plume model with magma reservoir model are added in conclusion section of this paper along with proper citation.

It would be nice to provide a theoretical bound for the computational effort of your method for a single simulation step and the numerical-grid resolution. You can make use of, for instance, the number of long-computations (e.g., matrix- vector products). Then, please provide a theoretical bound for such value when computations are performed across different processors.

Our interpretation of the reviewer's request for "theoretical bound for computational effort" is the operational complexity of the algorithms used. Apologize if we have misunderstood .

Without accounting for the detailed algorithm, the complexity for raw SPH method is $O(N * N)$ operations. Where, N is the total number of particles (SPH particles are essentially equivalent to discretization points in mesh-based methods). By adopting a background mesh and a compact-supported kernel function, the complexity will be reduced to $O(MN + mN)$. Where m is the average of number of particles within the compact support of the kernel function, M is number of particles among which neighbor searches are carried out. Please note that

number of particles within the compact support of the kernel function is not constant in our case. Basic analysis on time complexity is added in section "Parallellism and Performance".

When computations are performed across different processors, theoretically speaking, the total computational time will be $(s + \frac{p}{n} + \bar{p})$, where $(s + p)$(time needed for$\sim O(MN + mN)$ operations) is total sequential computational time with p representing computation that is parallelizable and s represents computation that can not be parallelized, n is total number of processors and $\bar{p}$ is extra computation and communication introduced by parallelization, for example, unnecessary duplication of work, communication overhead, time to split and combine.

It requires complete analysis of the detailed algorithms and instrumented data gathering to obtain more accurate theoretical and achieved bounds. We agree that such time complexity analysis would be very helpful for future performance optimization. However, considering HPC is not the major focus of this paper, we would like to reserve this topic for future research. We thank the reviewer again for proposing this idea.

It is not clear for me how the parallelization is performed, for instance, for a given time, do you split the domain across different processors? in such case, what constraints must be satisfied at each local domain in order to guarantee a consistent numerical solution of your equations?. Other possibility is to speed-up matrix computations, is this your case? or both?

The parallelization is not covered with enough details in this paper as it has been addressed in a separate paper [1]. The parallelization is achieved only by splitting the computational domain (Fig. 2(c) in GMD discussion paper shows a typical domain decomposition). No matrix computation is involved in the SPH scheme, so there is no parallelization regarding solving the matrix in our case. For any subdomain, information from its neighboring subdomains is required when updating physical quantities. To guarantee consistency, data is synchronized after each updates of physical quantities in the shared (or "halo") regions.

To address your questions and clean up confusion, we made a major revision in section 3.10 (is section 3.9 in the revised manuscript). In addition, portion of our response to reviewer's second comments are also added in section 3.10.

We thank reviewer again for the careful reading and constructive comments.

Major adjustments of the manuscripts

Here is a summary on major adjustments made in the revised manuscript: Major revision is made in "Parallelism and performance" section (section 3.10 in original manuscript and 3.9 in the revised manuscript).

References

[1] Zhixuan Cao, Abani Patra, and Matthew Jones. Data management and volcano plume simulation with parallel sph method and dynamic halo domains. *Procedia Comput. Sci.,*, 108:786–795, 2017.

[2] Antonio Costa, Yujiro Suzuki, and Takehiro Koyaguchi. Understanding the plume dynamics of explosive super-eruptions. *Nature communications*, 9(1):654, 2018.

[3] Luca Terray, Pierre-J Gauthier, Giuseppe Salerno, Tommaso Caltabiano, Alessandro La Spina, Pasquale Sellitto, and Pierre Briole. A new degassing model to infer magma dynamics from radioactive disequilibria in volcanic plumes. *Geosciences*, 8(1):27, 2018.

Plume-SPH 1.0: A three-dimensional, dusty-gas volcanic plume model based on smoothed particle hydrodynamics

Zhixuan Cao[1], Abani Patra[1], Marcus Bursik[2], E. Bruce Pitman[3], and Matthew Jones[4]

[1]Department of Mechanical and Aerospace Engineering and Computational Data Science and Engineering, University at Buffalo, SUNY, New York, USA
[2]Department of Geology, University at Buffalo, SUNY, New York, USA
[3]Department of Material Design and Innovation, University at Buffalo, SUNY, New York, USA
[4]Center for Computational Research, University at Buffalo, SUNY, New York, USA

Correspondence: Abani Patra (abani@buffalo.edu)

Abstract.

Plume-SPH provides the ~~the~~ first particle based simulation of volcanic plumes. SPH (smoothed particle hydrodynamics) has several advantages over currently used mesh based methods in modeling of multiphase free boundary flows like volcanic plumes. This tool will provide more accurate eruption source terms to users of VATDs (Volcanic ash transport and dispersion

5 models) greatly improving volcanic ash forecasts. The accuracy of these terms is crucial for forecasts from VATDs and the 3D SPH model presented here will provide better numerical accuracy. As an initial effort to exploit the feasibility and advantages of SPH in volcanic plume modeling, we adopt a relatively simple physics model (3D dusty-gas dynamic model assuming well mixed eruption ~~materiel and dynamic~~ material, dynamic equilibrium and thermodynamic equilibrium between ~~air and~~ erupted material and air that entrained into the plume, and minimal effect of winds) targeted at capturing the salient features of a

10 volcanic plume. The documented open source code is easily obtained and extended to incorporate other models of physics of interest to the large community of researchers investigating multiphase free boundary flows of volcanic or other origins.

The Plume-SPH code also incorporates several newly developed techniques in SPH needed to address numerical challenges in simulating multiphase compressible turbulent flow. The code should thus be also of general interest to the much larger community of researchers using and developing SPH based tools. In particular, the $SPH - \varepsilon$ turbulence model is to capture

15 mixing at unresolved scales~~, heat~~. Heat exchange due to turbulence is calculated by a Reynolds analogy and a corrected SPH is used to handle tensile instability and deficiency of particle distribution near the boundaries. We also developed methodology to impose velocity inlet and pressure outlet boundary conditions, both of which are scarce in traditional implementations of SPH.

The core solver of our model is parallelized with MPI (message passing interface) obtaining good weak and strong scalability

20 using novel techniques for data management using a SFCs (space-filling curves) and object creation time based indexing and hash table based storage scheme. These techniques are of interest to researchers engaged in developing particle in cell type methods. The ~~model is verified by~~ code is first verified by 1D shock tube tests, then by comparing velocity and concentration distribution along the central axis and on the transverse cross with experimental results of JPUE (jet or plume that is ejected from a nozzle into a uniform environment)~~and the top height of the~~. Profiles of several integrated variables are compared

with those calculated by existing 3D plume models for an eruption with the same MER (mass eruption rate) estimated for the Pinatubo eruption of ~~15 June~~ June 15 1991. Our results are consistent with ~~both observations and~~ existing 3D plume models. ~~Profiles of several integrated variables are compared with those calculated in existing 3D plume models, and further verify our model.~~ Analysis of the plume evolution process ~~illustrates~~ demonstrates that this model is able to reproduce the physics of plume development.

1 Introduction

1.1 Volcanic ash hazards

Primary hazards associated with explosive volcanic eruptions include pyroclastic density currents (flows and surges), the widespread deposition of airfall tephra, and the threats to aviation posed by volcanic ash in the atmosphere. Simulation of all possible hazards with one model is difficult due to the fact that different length scales dominate different hazards. Our focus here is the hazard that volcanic ash poses to aircraft.

During volcanic eruptions, VATDs (volcanic ash transport and dispersion models) are used to forecast the location and movement of ash clouds at timescales that range from hours to days. VATDs use eruption source parameters, such as plume height, mass eruption rate, duration, and the mass fraction distribution of erupted ~~debris~~ particles finer than about 4Φ (or ~~63μ~~62.5μm), which can remain in the cloud for many hours or days. Observational data for such parameters are usually unavailable in the first minutes or hours after an eruption is detected. Moreover, these input parameters are subject to change during an eruption, requiring rapid re-assignment of new parameters. Usually, plume models are used to provide these source terms for VATDs and the forecast accuracy is critically dependent on these models. This paper reports on a new 3D (three dimensional) volcanic plume model designed to exploit the advantages of mesh-free methods for 3D modeling of such plumes that involve multiphase free boundary flows.

1.2 Existing plume models

Several 1D (one dimensional) volcanic plume models have been developed in the past few decades, ranging from the most basic 1D model (Woods, 1988) which only accounts for mass conservation to more recently developed 1D models (Bursik, 2001; Mastin, 2007; Degruyter and Bonadonna, 2012; Woodhouse et al., 2013; Devenish, 2013; de'Michieli Vitturi et al., 2015; Folch et al., 2016; Pouget et al., 2016) which tend to account for more comprehensive physics effects. For example, FPLUME-1.0 (Folch et al., 2016) accounts for wind effect, entrainment of moisture, water phase change, particle fallout and re-entrainment and even wet aggregation of ash. However, in these 1D models, the entrainment of air is evaluated based on two coefficients: entrainment coefficient ~~for the vertical plume and the entrainment coefficient that describes the effect of wind~~due to turbulence in the rising buoyant jet and the crosswind field. Different 1D models adopt different entrainment coefficients based on specific formulation or calibration against well-documented case studies. The feedback from plume to atmosphere is usually ignored in 1D models. Even though determination of essential parameters such as the entrainment is 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 require little computational resource and can run on standard personal computers or on web sites in very short time. As a result, 1D software for volcanic plume development (such as Bursik, 2010; Mastin, 2011; de' Michieli Vitturi, 2015), combined with VATDs (such as Bursik et al., 2013; Draxler and Rolph, 2015) are widely used in research and practice. While these 1D models can generate well-matched results with 3D (three dimensional) models for weak plumes, much greater variability is observed for strong plume scenarios, especially for local variables (Costa et al., 2016). In addition, there is need for greater ~~skills~~ skill in hazards forecasts especially where the plume model is used to generate source conditions for complex 2D (two dimensional) and 3D VATD models.

The development of 2D and 3D, time-dependent, and multiphase numerical models for volcanic plumes has provided new explanations for many features of explosive volcanism. For example, a recent study based on 3D fluid dynamical model (Costa et al., 2018) shows that simple extrapolations of integral models for Plinian columns to those of super-eruption plumes are not valid and their dynamics diverge from current ideas of how volcanic plumes operate. One of the earliest of these is the 3D model PDAC (Pyroclastic Dispersion Analysis Code) (Neri et al., 2003) which is a non-equilibrium, multiphase, 3D compressible flow model. Conservation equations for each phase are solved separately with the finite volume method. A parallel computing version of PDAC was also developed (Esposti Ongaro et al., 2007). Advanced numerical techniques, such as a second order scheme and semi-explicit time stepping, was also adopted afterwards to improve the accuracy of PDAC (Carcano et al., 2013).

Another 3D model, SK-3D (Suzuki et al., 2005) is a 3D time-dependent fluid dynamics model that attempts to reproduce the entrainment process of eruption clouds with relatively simple physics but with high order numerical accuracy and high spatial resolution. A series of simulations based on SK-3D was reported, including establishment of the relationship between the observable quantities of the eruption clouds and the eruption conditions at the vent (Suzuki and Koyaguchi, 2009), investigation of the effect of the intensity of turbulence in the umbrella cloud on dispersion and sedimentation of tephra (Koyaguchi et al., 2009), determination of the entrainment coefficients of eruption columns as a function of height (Suzuki and Koyaguchi, 2010) and investigation of the effect of wind field on entrainment coefficient (Suzuki and Koyaguchi, 2013).

While SK-3D focuses on accurately capturing the entrainment caused by turbulent mixture with higher resolution and numerical method of higher order, PDAC takes the disequilibrium between different phases into account and hence is a true multiphase model. Another 3D model, ATHAM (Active Tracer High-Resolution Atmospheric Model) (Oberhuber et al., 1998) focuses more on microphysics within the plume. As pyroclastic flow is not the initial concern of ATHAM, dynamic and thermodynamical equilibrium is assumed in ATHAM. The dynamic core of ATHAM solves the compressible Euler equations for momentum, pressure and temperature of the gas particle mixture (Oberhuber et al., 1998). The subgrid-scale turbulence closure scheme that differentiates between the horizontal and vertical directions (Herzog et al., 2003) ~~captures~~ is adopted to capture turbulent mixing. The cloud microphysics predicts the mass of hydrometeors in liquid and ice phase (Herzog et al., 1998). Additional modules, including gas phase chemistry (Trentmann et al., 2002) and gas scavenging by hydrometeors (Textor et al., 2003) were added lately. A further extension was made to include particle aggregation (Textor et al., 2006b, a). However, the resolution of ATHAM is still coarse compared with SK-3D and PDAC.

Besides adding to their special strengths (ATHAM has been adding more and more microphysics, PDAC was extended to consider more phases), these models are also adding core strengths. PDAC development has begun to include the effect of microphysics into the model while ATHAM development has extended its ability to modeling pyroclastic flow. Both are using finer and finer resolution.

5 Recently, a first order, nonequilibrium compressible 3D model, ASHEE (Cerminara et al., 2016a), was introduced based on three dimensional N-phases Eulerian transportation equations, which are a full set of mass, momentum and energy transport equations for a mixture of gas and dispersed particles. ASHEE is valid for low concentration and low Stokes number region and much faster than N-phases Eulerian model. The model is based on the open source numerical solver OpenFOAM (Weller et al., 1998), adapting its unstructured finite volume solver.

10 To summarize, each 3D model has its own focus based on the problem of interest and modeling/numerical choices made. Accuracy of simulation (depending on comprehensiveness of the model, resolution of discretization, numerical error, and order of accuracy) and simulation time (depending on number of governing equations, resolution, numerical methods and parallel techniques) are always ~~a pair of~~ conflicting considerations in 3D plume simulations.

1.3 Features of SPH

15 To the best of our knowledge, all of the existing 3D plume models use mesh based Eulerian methods, and there are no 3D plume models based on mesh free Lagrangian methods. Lagrangian methods have several features that are suitable for volcanic plume simulation that we outline below. Among such Lagrangian methods, smoothed particle hydrodynamics (Gingold and Monaghan, 1977; Lucy, 1977) based simulations have shown good agreement with experiments for many applications in fluid dynamics. ~~And it currently~~ Currently it is, by far, the most widely used mesh-free scheme. Our implementations of SPH in 20 volcanology follows on earlier efforts (Bursik et al., 2003; Hérault et al., 2010; Haddad et al., 2016). Specifically, we choose SPH as the numerical method for volcanic plume simulation to enable:

- better investigation of mixing phenomena;

- accurate modeling of the development of ZFE (Zone of Flow Establishment), ZEF (Zone of Established Flow) investigation and relation to column collapse and the questions relating to the development of entrainment;

25 - easy inclusion of particles of different sizes (phases) and investigation of detailed mechanics of sedimentation and drag force interaction in lower plume.

These are enabled by the following features of SPH:

- ~~For mesh based method, either interface tracking (Lagrangian) (Harlow et al., 1965; Wrobel and Brebbia, 1991; Cheng and Armfield~~ ~~interface capturing (Eulerian) (Hirt and Nichols, 1981; Youngs, 1982; Gerlach et al., 2006; Gopala and van Wachem, 2008) methods~~ 30 ~~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 tracking techniques~~ ~~(Hirt and Nichols, 1981; Unverdi and Tryggvason, 1992; Anderson et al., 1998) . For interface capturing (Eulerian) method,~~

~~the surrendering of surface detail before the phase transport calculation means that interface reconstruction is required between time steps to recover the interface information, which needs additional numerical effort (Hirt and Nichols, 1981; Youngs, 198~~ Advection term in the Navier-Stokes equations does not appear explicitly in discretized formula of SPH (as illustrated in Eq. (12) to Eq. (15)).

- It is easy to include various physics effects (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 (Monaghan and Kocharyan, 1995). This implies that accounting for more physics is easier for SPH model.

10 - With more than one ~~material~~phase, each described by its own set of particles, interface problems between phases are often trivial for SPH but difficult for mesh based schemes. So multiphase flow can be easily handled by SPH. Adding of new phases to the model also does not require a major overhaul and re-tooling. As will be shown in later paragraphs, adding of new phases only leads to adding of several lines into the source code for new phases and additional interaction terms between existing phases and newly added phases.

15 - Interface tracking is explicit in SPH through capturing of the locations of the particles. Less numerical effort is required for interface construction when we attempt to include the effects of mixing by resolving the detailed interface structure and dynamics of turbulence.

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 accounting for more physics or more phases. Easy extensibility and capability 20 of handling ~~multiples phase flow without~~ multiple phase flow with less additional numerical effort greatly facilitate future extension of SPH models. As volcanic plumes are in nature multiphase and without pre-defined boundary in the atmosphere, SPH is a suitable numerical method for plume modeling. The core physics, such as entrainment of air and thermal expansion, are essential for all plume modeling while some other physics, such as water condensation and aggregation etc., are important in specific scenarios. As an initial effort on exploiting advantages of SPH in volcanic plume modeling, we focus on capturing 25 basic features in plume development using a ~~robust numerically~~ numerically robust and computationally efficient framework with support for scalable parallel computing.

Open source availability and the relatively easy extensibility of SPH will facilitate development of a more comprehensive community driven model.

1.4 Our contributions

30 Even though SPH has been known for several decades, implementations of SPH for compressible multiphase turbulent flows are few. Colagrossi and Landrini (2003) proposed a multiphase SPH model for numerical simulation of air entrapment in violent fluid-structure interactions. Hu and Adams (2007) and Adami et al. (2010) proposed weakly compressible and incompressible

multiphase SPH solvers in their papers. Monaghan and Rafiee (2013) also proposed new formulations for weakly compressible fluid with speed of sound sufficiently large to guarantee that the relative density variations are typically 1%. Chen et al. (2015) recently presented a new SPH model for weakly compressible multiphase flows with complex interfaces and large density differences. All of them focus on incompressible or weakly compressible flow.

5   Several issues endemic to classical SPH, like tensile instabilities, compressible turbulence modeling and turbulent heat exchange, are fixed in our implementation. The most popular applications of SPH (and their original motivating application) has been in the simulation of free surface flow, such as breaking-waves and floods. Less attention was paid to velocity inlet and pressure outlet boundary conditions which are required in plume modeling.

- We develop methodology to impose pressure boundary conditions by adding extra layers of static ghost particles. Addi-
10   tional constraints on the time step is are used to avoid the growth of numerical fluctuations near the pressure boundary. We impose a velocity inlet boundary condition by placing several layers of ghost particles moving with eruption velocity.

- Turbulence model is crucial for reproducing the entrainment of air. There are several turbulence models proposed for SPH method (Issa, 2005; Violeau and Issa, 2007). We adopt a LANS (Lagrangian Averaged Navier-Stokes) turbulence model (Monaghan, 2011), which was originally proposed for incompressible flow, and is extended here for compressible
15   flow accounting for turbulent heat exchange.

- Corrected formulation of SPH (Chen et al., 1999) is adapted adopted to bypass the well-known tensile instability issues of classical SPH.

- Simulation of volcanic plumes with acceptable accuracy requires fine resolution (very high particle counts) that cannot be accomplished without parallel computing using large process counts. The core solver of our model is parallelized by
20   distributed memory MPI (message passing interface standard) parallelism. In addition, a dynamic load balancing strategy is also developed.

- Imposition of some types of boundary conditions (such as eruption boundary condition) requires dynamically adding and removing of particles during simulation. To address this issue, we adopt an efficient data management scheme based on time dependent SFC (space filling curve) induced indexing and hash table. The computational cost is further reduced
25   by adjusting simulation domain adaptively.

The physical model of the plume is first presented in section 2 and leads to a complete mathematical description of the volcanic plume (governing equations and boundary conditions). In section 3, we briefly introduce the numerical tool – the SPH method. Both the fundamental discretization formulation and techniques that are used to handle specific issues involved in plume modeling are discussed. Verification and validation with numerical tests are presented in section 4. In section 5, a
30   discussion on future work is given following a brief summary.

2 Physical model

2.1 Description of the model

During an explosive eruption, a volcanic jet erupts out from a vent with 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. Cauliflower-like vortices are generated along jet margins, within which process, air is entrained and heated up, 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 up to several kilometers to tens of kilometers until its heat is exhausted. Jets that lose their momentum before becoming buoyant collapse back onto the ground and transform into pyroclastic flows, surges and ignimbrites. During the process of plume rising up, relatively larger particles might separate from main stream of the plume, falling down onto the ground and ~~probably~~ possibly be re-entrained into the plume at a lower height (Ernst et al., 1996). Within this process, erupted vapor condenses to liquid (droplet) and even further to ice. Latent heat released from phase change of erupted vapor further heats up entrained air and further dilutes the bulk density. The entrained vapor might also experience a similar process and impact plume development. Particle aggregation processes (Carey and Sigurdsson, 1982; Taddeucci et al., 2011), either due to presence of liquid water, resulting from particle collision or driven by electrostatic forces might occur inside plume and thereby affect the sedimentation.

All in all, the process of plume development is essentially a multiphase turbulent mixing process coupled with heat transfer and other microphysical and chemical reactions.

As an initial effort on exploiting the feasibility and advantage of SPH in plume modeling, our model is designed to describe an injection of well mixed solid and volcanic gas from a circular vent above a flat surface into a stratified stationary atmosphere following SK-3D (Suzuki et al., 2005). In this model, molecular viscosity and heat conduction is neglected since turbulent exchange coefficients are dominant. Erupted material consisting of solid with different size and mixture of gases is assumed to be well mixed and behave like a single phase fluid (phase 2) which is valid for eruptions with fine particles and ash. Air (also a mixture of different gases) is assumed to be another phase (phase 1). Thermodynamic 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 does not show up under this assumption). Dynamic equilibrium is assumed so that no separate momentum equation is needed for each phase. As a result, there is only one vector momentum equation for both phases (drag force term does not show up with this assumption).

Because of the above assumptions, all other microphysical processes (such as the phase changes of H_2O aggregation, disaggregation, absorption of gas on the surface of solids, solution of gas into a liquid) and chemical processes are not considered in this model. These ignored microphysics factors would play critical roles under particular eruption conditions. Capturing of these processes needs a more comprehensive model. One critical element in plume development, the effect of wind, is also not yet considered in this model. Introducing wind effects in our model requires dynamic pressure boundary conditions, which requires more numerical effort and algorithm design. To summarize, our model is not valid for eruptions where wind effect

plays a significant role in its development, usually refered as a weak plume. Our model also lacks the ability in modeling plumes with large particles or eruptions in which microphysics plays non-ignorable roles, such as ~~a~~ an eruption of *El Chichón* volcano on April 4th 1982 (Sigurdsson et al., 1984; Folch et al., 2016). We are focussed here on developing the SPH based methodoloy in the context of the more basic (and more critical) aspects of volcanic plume and therefore devote our effort to

5 this relative simpler model. It is worthwhile to mention here that because SPH is adopted as our numerical method, adding of these physics into our model would require much less work in terms of programming compared to mesh based methods. Since our plan is an open source distribution of the tool we believe some of these enhancements will rapidly ensue with community participation.

2.2 Governing equations

10 Based on above assumptions, the governing equations of our model are given as (which is the same as the governing equations of SK-3D (Suzuki et al., 2005)):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{v}) = 0 \tag{1}$$

$$\frac{\partial \rho \xi}{\partial t} + \nabla \cdot (\rho \xi \mathbf{v}) = 0 \tag{2}$$

$$\frac{\partial \rho \mathbf{v}}{\partial t} + \nabla \cdot (\rho \mathbf{v} \mathbf{v} + p \mathbf{I}) = \rho \mathbf{g} \tag{3}$$

15 $$\frac{\partial \rho E}{\partial t} + \nabla \cdot [(\rho E + p) \mathbf{v}] = \rho \mathbf{g} \cdot \mathbf{v} \tag{4}$$

~~Where~~ where ρ is the density, $\mathbf{v}$ is the velocity, ξ is the mass fraction of ejected material, $\mathbf{g}$ is the gravitational acceleration, $\mathbf{I}$ is a unit tensor. $E = e + K$ is the total energy which is a 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 following EOS (equation of state).

20 $$p = (\gamma_m - 1) \rho e \tag{5}$$

~~Where~~ where

$$\gamma_m = R_m / C_{vm} + 1 \tag{6}$$

$$R_m = \xi_g R_g + \xi_a R_a \tag{7}$$

$$C_{vm} = \xi_s C_{vs} + \xi_g C_{vg} + \xi_a C_{va} \tag{8}$$

$$\xi_a = 1 - \xi \tag{9}$$

$$\xi_g = \xi \cdot \xi_{g0} \tag{10}$$

$$\xi_s = \xi - \xi_g \tag{11}$$

5 ~~Where~~where, C_v is the specific heat with constant volume, R is the gas constant. ξ ~~with subscript~~ is the mass fraction of~~corresponding constituent~~erupted material. The subscript m represents mixture of ejected material and air, s represents solid portion in the ejected material, g represents gas portion in the ejected material~~and~~, a represents air, 0 represents phyiscal properties of erupted material. ξ_{g0} is the mass fraction of vapor in the erupted material

In mesh based methods, governing equations in Eulerian form, Eq. (1) to Eq. (4), are directly discretized. For SPH, governing
10 equations in Lagrange form are needed. By deducting kinetic energy from energy equation, subtracting mass conservation from momentum equation, combining transient term and advection term into material derivative term ~~,~~(For any function A, material derivative is defined as $\frac{DA}{Dt} = \frac{\partial A}{\partial t} + \mathbf{v} \cdot \nabla A$), the governing equations are put into the final form, in which advection term does not appear explicitly.

$$\frac{D\rho}{Dt} + \rho \nabla \cdot \mathbf{v} = 0 \tag{12}$$

$$\frac{D\rho\xi}{Dt} + \rho\xi \nabla \cdot \mathbf{v} = 0 \tag{13}$$

$$\frac{D\mathbf{v}}{Dt} + \frac{\nabla P}{\rho} = \mathbf{g} \tag{14}$$

$$\frac{De}{Dt} + \frac{P\nabla \cdot \mathbf{v}}{\rho} = 0 \tag{15}$$

2.3 Boundary conditions

In the current model the initial domain is a box. The boundaries are categorized as the velocity inlet (a circular area at the
20 center of the bottom of the box), wall boundary (box bottom) ~~,~~and pressure outlet (other faces of the box).

2.3.1 Velocity inlet

At the vent, temperature of erupted material T, eruption velocity $\mathbf{v}$, the mass fraction of vapor in erupted material ξ_{g0} and mass discharge rate $\dot{M}$ are 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 $\mathbf{v}$. Equation (16) to (19) gives velocity ~~outlet~~inlet boundary

condition wrote in terms of primitive variables.

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

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

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

$$\frac{\partial e}{\partial n} = \dot{M} e_0 / \left(\pi r^2\right) \tag{19}$$

~~Where~~ where r is the radius of the vent, n is the direction normal to the boundary, e_0 is specific internal energy of erupted material which can be calculated based on temperature and specific heat of erupted material.

2.3.2 Non-slip wall boundary

Velocity is zero for non-slip wall boundary. If we assume the boundary to be adiabatic, heat flux should be zero on the boundary. 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, vanishes on the wall boundary. Equation (20) to (23) gives no-slip wall boundary condition written in terms of primitive variables.

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

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

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

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

2.3.3 Open outlet pressure boundary condition

The pressure of the surrounding atmosphere is given. Except for the pressure, boundary values for density, velocity, and energy on the outlet should depend on the solution. As we ignore the viscosity, the shear stress is ignored and normal stress (whose magnitude equals to pressure) balances the ambient pressure.

$$p = p_a\left(z\right) \tag{24}$$

3 SPH method

SPH is a mesh-free Lagragian method. In SPH, the domain is discretized by a set of particles or discretization points and the position of each particle is updated at every time step based on the motion computed. 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 PDEs (partial differential equations) or ODEs (ordinary 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 estimated as sums over particles in a compact subdomain defined by the support of the kernel function associated with the discretization points. Thus, field variables associated to the particle are updated based on its neighbors. Each kernel function has a compact support determined by smoothing length of each particle. There are several review papers by Monaghan (1992, 2005); Rosswog (2009); Price (2012); Monaghan (2012), giving a pretty comprehensive view over SPH. We only refer here to the representation of the constitutive equations in SPH and put more focus on specific numerical techniques for plume modeling.

3.1 Fundamental principles

There are several procedures for discretizing governing equations (PDEs or ODEs) with SPH. We present here one of them following Monaghan (1992, 2005, 2012). The starting point of approximating a function with SPH is the translation property of the Dirac function $\delta(\mathbf{x})$, for an arbitary function $A(\mathbf{x})$, the following equation holds.

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

The Dirac function δ in Eq. (25) can be approximated by a weighting function $w(\mathbf{x} - \mathbf{x}\prime, h)$ (or $w(\mathbf{x}\prime - \mathbf{x}, h)$) which tends to a Dirac function when the smoothing length $h \to 0$:

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

An The weighting function, as an approximate form of the Dirac function, should satisfy the normalization condition:

$$\int w(\mathbf{x} - \mathbf{x}\prime, h) \, d\mathbf{x}\prime = 1 \tag{27}$$

Besides normalization, the weighting function of particle a has to be symmetric with respect to a to ensure that neighbor particles located at the same distance away from a contribute equally to SPH summation equation, see Eq. (28).

$$w(\mathbf{x} - \mathbf{x}\prime, h) = w(\mathbf{x}\prime - \mathbf{x}, h) \tag{28}$$

The weighting function also needs to satisfy conditions such as positivity and compact support. In addition, the kernel function must be monotonically decreasing with the distance between particles.

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

$$w(\mathbf{x} - \mathbf{x}\prime) = \begin{cases} \dfrac{1}{(h\sqrt{\pi})^d} exp\left[-\left(\dfrac{\mathbf{x} - \mathbf{x}\prime}{h}\right)^2\right] & |\mathbf{x} - \mathbf{x}\prime| \leq 3h \\ 0 & \text{Otherwise} \end{cases} \tag{29}$$

where d is number of dimensions. The derivative of the weighting function is:

$$\nabla w\left(\mathbf{x}-\mathbf{x}\prime\right) = \begin{cases} -2\left(\dfrac{\mathbf{x}-\mathbf{x}\prime}{h}\right)\dfrac{1}{(h\sqrt{\pi})^d} exp\left[-\left(\dfrac{\mathbf{x}-\mathbf{x}\prime}{h}\right)^2\right] & |\mathbf{x}-\mathbf{x}\prime| \leq 3h \\ \\ 0 & \text{Otherwise} \end{cases} \tag{30}$$

By replacing δ function in Eq. (26) with the kernel function w, an arbitrary function $A(\mathbf{x})$ can then be approximated by:

$$A(\mathbf{x}) \approx\, <A(\mathbf{x})> =\int_\Omega A\left(\mathbf{x}\prime\right)w\left(\mathbf{x}-\mathbf{x}\prime,h\right)d\mathbf{x}\prime + O\left(h^2\right) \tag{31}$$

5 ~~Where h is the smoothing length, determining the interaction distance.~~ As the weighting function is symmetric (Eq. (28)) and satisfies the normalization condition (Eq. (27)), odd error terms in Eq. (31) vanishe leading to a second order approximation. However, in practice, second order of accuracy can not be achieved because there is no guarantee on the symmetry of particle distribution in real simulation (Price, 2012). Recall that $d\mathbf{x}\prime = \dfrac{dm\prime}{\rho\prime}d\mathbf{x}\prime = \dfrac{dm(\mathbf{x}\prime)}{\rho(\mathbf{x}\prime)}$, the integration equation, Eq. (31), can be approximated by summation and lead to an approximation of the function A:

10 $$<A(\mathbf{x})> \approx \sum_b m_b \frac{A_b}{\rho_b} w\left(\mathbf{x}-\mathbf{x}_b,h\right) \tag{32}$$

where the summation is over all the particles within the region of compact support (See Eq. (29)) of the weighting function. Gradient terms may be straightforwardly calculated by taking the derivative of Eq. (32), giving

$$<\nabla A(\mathbf{x})> = \frac{\partial}{\partial \mathbf{x}}\int_\Omega A\left(\mathbf{x}\prime\right)w\left(\mathbf{x}-\mathbf{x}\prime,h\right)d\mathbf{x}\prime + O\left(h^2\right)$$
$$\approx \sum_b m_b \frac{A_b}{\rho_b}\nabla w\left(\mathbf{x}-\mathbf{x}_b,h\right) \tag{33}$$

For vector quantities the expressions are similar, simply replacing A with $\mathbf{A}$ in Eq. (32) and Eq. (33), giving

15 $$<\mathbf{A}(\mathbf{x})> \approx \sum_b m_b \frac{\mathbf{A}_b}{\rho_b}w\left(\mathbf{x}-\mathbf{x}_b,h\right) \tag{34}$$

$$<\nabla \cdot \mathbf{A}(\mathbf{x})> \approx \sum_b m_b \frac{\mathbf{A}_b}{\rho_b}\cdot\nabla w\left(\mathbf{x}-\mathbf{x}_b,h\right) \tag{35}$$

$$<\nabla \times \mathbf{A}(\mathbf{x})> \approx \sum_b m_b \frac{\mathbf{A}_b}{\rho_b}\times\nabla w\left(\mathbf{x}-\mathbf{x}_b,h\right) \tag{36}$$

$$<\nabla^j \mathbf{A}^i(\mathbf{x})> \approx \sum_b m_b \frac{\mathbf{A}_b^i}{\rho_b}\nabla^j w\left(\mathbf{x}-\mathbf{x}_b,h\right) \tag{37}$$

3.2 ~~Weighting function~~ Artificial Viscosity

20 ~~As described in the previous section, the weighting function (or kernel function) is used to replace the Dirac function and should satisfy Eq. (26). So it can be viewed as an approximate form of the Dirac function and satisfies the normalization~~

condition:

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

Besides normalization, the weighting function of particle a has to be symmetric with respect to a to ensure that neighbor particleslocated at the same distance away from a contribute equally to SPH summation equation, see Eq. (28). The weighting function also needs to satisfy conditions such aspositivity and compact support. In addition, the kernel function must be monotonically decreasing with the distance between particles.

$$w\left(\mathbf{x}-\mathbf{x}\prime,h\right)=w\left(\mathbf{x}\prime-\mathbf{x},h\right)$$

In classical SPH, shock waves are handled by introducing artificial viscosity, a term that is defined based on second derivatives of velocity, to smear out discontinuities. As in the case of first order derivatives, second order derivatives can be estimated by differentiating a SPH interpolation twice. However, such a formulation has two disadvantages: first, it is very sensitive to irregular distribution of particles, second, the second derivative of the kernel can change sign and lead to unphysical representations (for example, viscous dissipation causes decrease of the entropy).

There is a wide variety of possible weighting functions, such as spline functions (with different orders) and Gaussian functions. Generally, the accuracy increases with the order of the polynomials of the kernel function, but the computational cost also increases as number of interactions increase. We are adopting a truncated Gaussian function as the weighting function One of the most commonly used models of artificial viscosity (Monaghan and Gingold, 1983) is:

$$\Pi_{ab}=-\frac{\nu}{\bar{\rho}_{ab}}\frac{\mathbf{v}_{ab}\cdot\mathbf{x}_{ab}}{\mathbf{x}_{ab}^{2}+\left(\eta h\right)^{2}} \tag{38}$$

The coefficient ν is defined as:

$$\nu=\alpha\bar{h}_{ab}\bar{c}_{ab} \tag{39}$$

where

$$\bar{c}_{ab}=\frac{c_{a}+c_{b}}{2} \tag{40}$$

$$\bar{\rho}_{ab}=\frac{\rho_{a}+\rho_{b}}{2} \tag{41}$$

$$\mathbf{v}_{ab}=\mathbf{v}_{a}-\mathbf{v}_{b} \tag{42}$$

$$\mathbf{x}_{ab}=\mathbf{x}_{a}-\mathbf{x}_{b} \tag{43}$$

The artificial viscosity term Π_{ab} is a Galilean invariant and vanishes for rigid rotation. It produces 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 (Monaghan, 2005). It has been shown that shear viscosity coefficient $\eta = \frac{\rho \alpha h c}{8}$ and bulk viscosity coefficient $\zeta = \frac{5\eta}{3}$ are appropriate for two dimensional flows. $\eta = \frac{\rho \alpha h c}{10}$ and $\zeta = \frac{5\eta}{3}$ are appropriate for three dimensional flows. An extra term was added to ν considering aspects of the dissipative term in shock solutions based on Riemann solvers and lead to a new formulation of artificial viscosity.

5 We adopt this new formulation in our simulation:

$$\Pi_{ab}^{\beta} = \begin{cases} \dfrac{-\alpha \mu_{ab} \bar{c}_{ab} + \beta \mu_{ab}^2}{\bar{\rho}_{ab}} & \mathbf{v}_{ab} \cdot \mathbf{x}_{ab} < 0 \\ \\ 0 & \mathbf{v}_{ab} \cdot \mathbf{x}_{ab} > 0 \end{cases} \tag{44}$$

where

$$\mu_{ab} = \frac{h \mathbf{v}_{ab} \cdot \mathbf{x}_{ab}}{\mathbf{x}_{ab}^2 + (\eta h)^2} \tag{45}$$

α and β are two parameters that can be adjusted for different cases. $\alpha = 1$ and $\beta = 2$ are recommended by Monaghan for best

10 results. In our simulation, these two parameters are calibrated to $\alpha = 0.3$ and $\beta = 0.6$.

$$w(\mathbf{x} - \mathbf{x}_b) = \begin{cases} \dfrac{1}{(h\sqrt{\pi})^d} exp\left[-\left(\dfrac{\mathbf{x} - \mathbf{x}\prime}{h}\right)^2\right] & |\mathbf{x} - \mathbf{x}\prime| \le 3h \\ \\ 0 & \text{Otherwise} \end{cases}$$

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

$$\nabla w(\mathbf{x} - \mathbf{x}_b) = \begin{cases} -2\left(\dfrac{\mathbf{x} - \mathbf{x}\prime}{h^2}\right) \dfrac{1}{(h\sqrt{\pi})^d} exp\left[-\left(\dfrac{\mathbf{x} - \mathbf{x}\prime}{h}\right)^2\right] & |\mathbf{x} - \mathbf{x}\prime| \le 3h \\ \\ 0 & \text{Otherwise} \end{cases}$$

η is usually taken as 0.1 to prevent singularities.

15 ### 3.3 Discretization of governing equations and extensibility

The basic interpolation given in Eq. (32) to Eq. (37) provides a general way to obtain SPH expressions of governing equations. The problem is that using these expressions "as is" in general leads to quite poor gradient estimates. Various tricks can be used to conserve linear and angular momentum and thermal energy (Monaghan, 1992). Special treatments are also needed for second order derivative terms (Monaghan, 2005). We only refer here to one of these possible discretizations of compressible

Euler equations with SPH:

$$< \rho_a >= \sum_b m_b w_{ab}(h) \tag{46}$$

$$\left\langle \frac{d\mathbf{v}_a}{dt} \right\rangle = -\sum_b m_b \left(\frac{p_b}{\rho_b^2} + \frac{p_a}{\rho_a^2} + \Pi_{ab} \right) \nabla_a w_{ab}(h) + \mathbf{g} \tag{47}$$

$$\left\langle \frac{de_a}{dt} \right\rangle = 0.5 \sum_b m_b \mathbf{v}_{ab} \left(\frac{p_b}{\rho_b^2} + \frac{p_a}{\rho_a^2} + \Pi_{ab} \right) \cdot \nabla_a w_{ab}(h) \tag{48}$$

5   where, a is the SPH particle index~~, $\mathbf{v}_{ab} = \mathbf{v}_a - \mathbf{v}_b$~~. Π is an artificial viscosity term, which ~~will be~~ is discussed in section 3.2. $w_{ab}(h)$ is a concise form of $w(\mathbf{x}_a - \mathbf{x}_b, h)$ and from here on, we will use this concise form. As a Lagrangian method, particle position is also updated at every time step.

$$\left\langle \frac{d\mathbf{x}_a}{dt} \right\rangle = \mathbf{v}_a \tag{49}$$

We highlight an important feature of the SPH methodology. Adding new physics and new phases into the model is trivial in
10   terms of discretization. For example, adding of new source (or sink) into Eq. (46), adding a drag force into Eq. (47) and adding a heat exchange term into Eq. (48) leads to the new discretization form:

$$< \rho_a >= \sum m_b w_{ab}(h) + \dot{\rho}(\mathbf{x}, t) \tag{50}$$

$$\left\langle \frac{d\mathbf{v}_a}{dt} \right\rangle = -\sum_b m_b \left(\frac{p_b}{\rho_b^2} + \frac{p_a}{\rho_a^2} + \Pi_{ab} \right) \nabla_a w_{ab}(h) + \mathbf{g} + D \sum_b m_b \frac{\mathbf{v}_b - \mathbf{v}_a}{\rho_b} \tag{51}$$

$$\left\langle \frac{de_a}{dt} \right\rangle = 0.5 \sum_b m_b \mathbf{v}_{ab} \left(\frac{p_b}{\rho_b^2} + \frac{p_a}{\rho_a^2} + \Pi_{ab} \right) \cdot \nabla_a w_{ab}(h) + \sum_b \frac{m_b}{\rho_b} (\kappa_a + \kappa_b) \frac{(T_a - T_b)}{\mathbf{r}_a - \mathbf{r}_b} w_{ab}(h) \tag{52}$$

15   where the source term $\dot{\rho}$ can be a "sink" of erupted vapor due to its phase change. D is a drag force coefficient. κ is the heat conduction coefficient. T is the temperature. Other physics can be added easily in a similar way. Adding of these new terms leads to modification of ~~several~~ only a few lines in the source code. The drag force term should show up ~~only when dynamics~~ when dynamic disequilibrium between different phases is considered. In that case, each phase needs one set of governing equations of Navier-Stokes type. Adding of new phase into SPH code only needs adding few new lines for the new phase
20   besides interaction terms introduced by the new phase.

3.4   ~~Artificial viscosity~~

~~In classical SPH, shock waves are handled by introducing artificial viscosity to smear out discontinuities. Π, in discretized momentum equation, Eq. (47), and energy equation, Eq. (48), represents artificial viscosity term which is essentially a second order derivative term. As in the case of first order derivatives, second order derivatives can be estimated by differentiating a SPH~~
25   ~~interpolation twice. However, such a formulation has two disadvantages: First, it is very sensitive to particle disorder. Second, the second derivative of the kernel can change sign and lead to unphysical representations (For example, viscous dissipation causes decrease of the entropy).~~

One of the most commonly used models of artificial viscosity is:

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

The coefficient ν is defined as:

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

5 Where

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

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

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

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

10 The artificial viscosity term Π_{ab} is a Galilean invariant and vanishes for rigid rotation. It produces 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 (Monaghan, 2005). It has been shown that the shear viscosity coefficient $\eta = \frac{\rho \alpha h c}{8}$ and the bulk viscosity coefficient $\zeta = \frac{5\eta}{3}$ for two dimensional and $\eta = \frac{\rho \alpha h c}{10}$, $\zeta = \frac{5\eta}{3}$ for three dimensional. An extra term was added to ν considering aspects of the dissipative term in shock

15 solutions based on Riemann solvers and lead to a new formulation of artificial viscosity. We adopt this new formulation in our simulation:

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

Where

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

20 α and β are two parameters that can be adjusted for different cases. $\alpha = 1$ 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 to prevent singularities.

3.4 Time step

The physical quantities (velocity, density and pressure) and particle position change every time step. The Courant condition, which is in spirit similar to the Courant condition for the mesh-based methods, is used to determine the time step Δt.

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

5 ~~Where~~ where c_a is sound speed at particle ~~a, d is number of dimensions.~~ a and calculated based on heat specific ration of the mixture γ_m (See Eq. (54)).

$$c_a = \left(\gamma_m \frac{p}{\rho} \right)^{0.5} \tag{54}$$

First order Euler integration, with CFL $= 0.2$, is used to advance in time.

3.5 Tensile instability and corrected derivatives

10 The classical SPH method was known to suffer from tensile instability and boundary deficiency. Tests of the standard SPH method indicate an instability in the tensile regime, while the calculations are stable in compression. A simple example of such a test calculation exhibiting the instability involves a body which is subject to an uniform initial stress, either compressive or tensile. If the initial stress is tensile, a very small velocity perturbation on a single particle can lead to particles clumping together, forming large voids and seriously corrupting density distribution. But if the initial stress is compressive, the small

15 velocity perturbation on a single particle can not lead to any changes in particle distribution (Swegle et al., 1995) . To address these difficulties, ~~Chen (Chen et al., 1999)~~ Chen et al. (1999) proposed a corrected SPH formulation. For 1D case, employing a Taylor expansion for $A(x)$ about x_a, multiplying both sides by kernel function and then doing integration over the domain gives

$$\int_\Omega A(x) \, w(x - x_a, h) \, dx = A_a \int_\Omega w(x - x_a, h) \, dx + \underline{A_{xa}} \frac{\partial A}{\partial x}(x_a) \int_\Omega (x - x_a) \, w(x - x_a, h) \, dx + \dots \tag{55}$$

20 Ignoring derivative terms higher than first order, and writing the integral in particle approximation form leads to:

$$A_a = \frac{\sum_b m_b \frac{A_b}{\rho_b} w(x_a - x_b, h)}{\sum_b m_b \frac{1}{\rho_b} w(x_a - x_b, h)} \frac{\sum_b m_b \frac{A_b}{\rho_b} w_{ab}(h)}{\sum_b m_b \frac{1}{\rho_b} w_{ab}(h)} \tag{56}$$

~~Equation~~ Notice that the denominator in Eq. (56) ~~implies that~~ $\sum_b m_b \frac{1}{\rho_b} w(x - x_b, h) = 1$, ~~which can be viewed as the approximation form of~~ is actually summation approximation of the left side in Eq. (27). That is to say, Eq. (56) and Eq. (32) are the same for particles far away from boundaries as the denominator in Eq. (56) becomes 1 in that case. The first order derivative term can

be obtained in a similar way:

$$\nabla A_a = \frac{\sum_b m_b \dfrac{A_b - A_a}{\rho_b} \nabla_a w(x_a - x_b, h)}{\sum_b m_b \dfrac{x_b - x_a}{\rho_b} \nabla_a w(x_a - x_b, h)} \frac{\sum_b m_b \dfrac{A_b - A_a}{\rho_b} \nabla_a w_{ab}(h)}{\sum_b m_b \dfrac{x_b - x_a}{\rho_b} \nabla_a w_{ab}(h)} \tag{57}$$

For problems of higher dimension, the expressions for function approximation are exactly the same as Eq. (56), even though the derivation is different. The first order derivative can be obtained by solving system of equations explicitly or numerically (Chen et al., 1999).

3.6 ~~Multiphase SPH~~Mass fraction update

~~Numerical simulation of multiphase flows is usually difficult due to the existence of complex evolving interfaces between phases. This presents severe challenges to conventional Eulerian grid-based numerical methods. As we discussed in the introduction section, special algorithms to treat and track the interface between different phases is required for mesh based methods to deal with multiphase problems. What's worse, the engulfment process during plume development makes geometry of interfaces very complicated. SPH is able to handle multiphase problems without additional numerical effort for dealing with interfaces by simply tagging particles of each phase, which makes it suitable for volcanic plume simulation. Take the governing equations adopted in our model as an example. Different phases of fluid (air and erupted material)~~ Air and erupted material are represented by two different sets of SPH particles (or discretization points) in the model. Based on assumptions we made in section 2, only density needs to be updated respectively for each phase. The updating of density is exactly the same as Eq. (46) in spirit. Particles of phase 1 are not ~~be~~ counted while evaluating density of phase 2 and vice versa. Updating of density is then based on the following discretized equations.

$$< \rho_\alpha^a > = \frac{\sum m_b w_{\alpha b}(h)}{\sum \frac{m_b}{\rho_b} w_{\alpha b}(h) + \sum \frac{m_j}{\rho_j} w_{\alpha j}(h)} \tag{58}$$

$$< \rho_\alpha^{sg} > = \frac{\sum_j m_j w_{\alpha j}(h)}{\sum \frac{m_b}{\rho_b} w_{\alpha b}(h) + \sum \frac{m_j}{\rho_j} w_{\alpha j}(h)} \frac{\sum m_j w_{\alpha j}(h)}{\sum \frac{m_b}{\rho_b} w_{\alpha b}(h) + \sum \frac{m_j}{\rho_j} w_{\alpha j}(h)} \tag{59}$$

~~Where~~ where the subscript a and b represents air particles (phase 1) while i and j represents particles of erupted material. β and α represents either erupted material ~~or air~~particles or air particles.

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 air and erupted material. By definition, the mass fraction is updated according to Eq. (60).

$$< \xi_\alpha > = \frac{\rho_\alpha^{sg}}{\rho_\alpha} \tag{60}$$

In areas far away from the interface, updating of density is exactly the same as that for single phase flow. For example, in the right side and left side (or blue areas) in Fig. 1, where there are only air particles, Eq. (59) evaluates to zero and total density

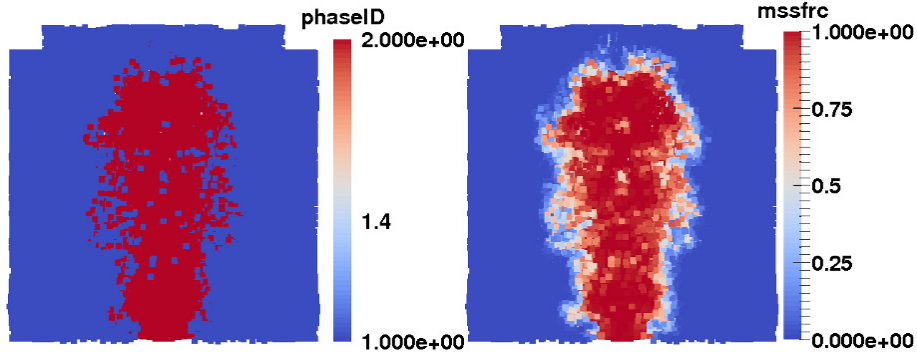



Figure 1. In the left figure, the blue particles (phaseID=1) represent air particles the red ones (phaseID=2) represent erupted material. The right figure shows corresponding mass fraction. Mass fraction are evaluated based on Eq. (58) and Eq. (~~58~~59) without any other interface track or capture method.

is:

$$\rho_\alpha = \rho_\alpha^a = \frac{\sum m_b w_{\alpha b}(h)}{\sum \frac{m_b}{\rho_b} w_{\alpha b}} \frac{\sum m_b w_{\alpha b}(h)}{\sum \frac{m_b}{\rho_b} w_{\alpha b}(h)} \tag{61}$$

Which is a special case of Eq. (56). For these areas occupied by only particles of phase 2 and far away from the interface, similarly, the equation for density update becomes:

$$\rho_\alpha = \rho_\alpha^{sg} = \frac{\sum m_j w_{\alpha j}(h)}{\sum \frac{m_j}{\rho_j} w_{\alpha j}} \frac{\sum m_j w_{\alpha j}(h)}{\sum \frac{m_j}{\rho_j} w_{\alpha j}(h)} \tag{62}$$

That is to say, the same density updating equation can be applied for both phases and no additional numerical treatment needed to locate where the interface is.

Interface construction will become necessary and important when attempt to include the effects of mixing by resolving the detailed interface structure and dynamics of turbulence. As a Lagrangian method, interface tracking in SPH is explicit through capturing of the locations of the particles, much simplier than Eleruan methods. The existence of complex evolving interfaces between phases presents severe challenges to conventional Eulerian grid-based numerical methods. Either interface tracking (Lagrangian) (Harlow et al., 1965; Wrobel and Brebbia, 1991; Cheng and Armfield, 1995) or interface capturing (Eulerian) (Hirt and Nich 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 tracking techniques (Hirt and Nichols, 1981; Unverdi and Tryggvason, 1992; Anderson et al., 1998) . For interface capturing (Eulerian) method, the surrendering of surface detail before the phase transport calculation means that interface reconstruction is required between time steps to recover the interface information, which needs additional numerical effort (Hirt and Nichols, 1981; Youngs, 1982) . Since SPH is able to adaptively adjust the discretization and automatically construct the interface. SPH requires less additional numerical effort for interface construction and therefore is more suitable for volcanic plume simulation.

3.7 Turbulence modeling with SPH

~~Typically, turbulencefluctuations in a volcanic column occurs at very different length scales, ranging from millimeters up to hundreds of meters. The entrainment of air and~~ For high speed shearing flow, the momentum exchange and heat transfer are dominated by ~~such~~ turbulent fluctuations as turbulent exchange coefficients are several magnitudes larger than corresponding

5 physical coefficients (molecular viscosity and heat conduction coefficient). ~~One way to include enough turbulence in the model is to use the~~ In addition to momentum and enrgy exchange, mixing between plume and air is important in plume modeling. Quantifying these mixing processes in real implementation is challenging because of the scale disparity between the large-scale fluid motion and the diffusion processes on interface that ultimately lead to mixing. Ideally, one would like to be able to include the effects of mixing on the large scale dynamics without resolving the detailed interface structure and dynamics of turbulence

10 to reduce computational cost. To resolve all turbulent exchange at all different scales and sub-particle scale mixing with relative coarse resolution, a SPH-SPS ~~(SPH sub-particle-scale) turbulence model. Another way is to use fine enough resolution (namely, the direct numerical simulation method) at the expense of much higher computational costwhich results from both increase in number of particles and decrease in time steps constrained by the CFL (Courant–Friedrichs–Lewy) condition. Here we choose the SPH-SPS method~~turbulence models should be included. Among existing SPH-SPS (SPH sub-particle-scale) turbulence

15 models (Holm, 1999; Monaghan, 2002; Violeau and Issa, 2007; Monaghan, 2011), we adopt a LANS type turbulence model, the $SPH-\varepsilon$ turbulence model (Monaghan, 2011). However, the $SPH-\varepsilon$ turbulence model was proposed only for incompressible flow. In the following section, we will extend it for compressible flow. It is necessary to mention that all other existing SPH-SPS turbulence models (Holm, 1999; Monaghan, 2002; Violeau and Issa, 2007) also only focus on incompressible flow.

3.7.1 Langrangian average in $SPH-\varepsilon$

20 Monaghan (2011) constructed $SPH-\varepsilon$ turbulence model 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 LANS turbulence modeling. The basic idea of $SPH-\varepsilon$ is to determine a smoothed (averaged in space) velocity $\widehat{\mathbf{v}}$ by a linear operation on the unsmoothed velocity $\mathbf{v}$. The SPH particles move with this smoothed velocity and hence the average motion of the fluid is determined by the averaged velocity $\widehat{\mathbf{v}}$:

$$\frac{d\mathbf{x}_a}{dt} = \widehat{\mathbf{v}}_a \tag{63}$$

Average of physical quantities over space introduces extra terms into the governing equations. Once the form of the smoothing (average) is chosen these extra terms are determined. The typical LANS model uses a smoothed velocity $\widehat{\mathbf{v}}$ defined in terms of the unsmoothed velocity $\mathbf{v}$ by:

$$\widehat{\mathbf{v}}(\mathbf{x}) = \int \mathbf{v}(\mathbf{x}\prime)\,G\left(|\mathbf{x}\prime - \mathbf{x}|, l\right) d\mathbf{x}\prime \tag{64}$$

30 ~~Where~~ where G satisfies:

$$\int G\left(|\mathbf{x}\prime - \mathbf{x}|, l\right) d\mathbf{x}\prime = 1 \tag{65}$$

and is a member of a sequence of functions which tends to the δ function in the limit when $l \to 0$. 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 a common practice in LANS to use a differential equation for the smoothing rather than the integral form and finally reach to a system of equations that need to be solved implicitly. In $SPH - \varepsilon$ method, a XSPH (Monaghan, 1989) smoothing is adopted which conserves linear and angular momentum. In this way, solving of a system of equations is avoided and it also makes the method simple to implement and cheap for computation. The discretized form of the momentum equation is obtained through lengthy derivation. Derivation and other discussions are available in the literature – see for e.g. (Monaghan, 2011). Here we provide a brief summary of key steps.

The smoothing adopted by Monaghan (2011) is:

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

As function G has the same feature as kernel function w, SPH approximation of the integration leads to:

$$\widehat{\mathbf{v}}(\mathbf{x}) = \mathbf{v}(\mathbf{x}) + \epsilon \sum_b m_b \frac{(\mathbf{v}_b - \mathbf{v})}{\rho_b} G\left(|\mathbf{x}_b - \mathbf{x}|, l \right) \tag{67}$$

By making the replacement:

$$\frac{G\left(|\mathbf{x}_b - \mathbf{x}_a|, l \right)}{\rho_b} \to \frac{K_{ab}}{M} \tag{68}$$

~~Where~~ 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 lengthy derivation:

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b \left[m_b \left(\frac{p_b}{\rho_b^2} + \frac{p_a}{\rho_a^2} \right) \nabla_a w_{ab}(h) \right] + \sum_b m_b \frac{\varepsilon}{2} \frac{\mathbf{v}_{ab} \cdot \mathbf{v}_{ab}}{M} \nabla_a K_{ab} \tag{69}$$

Notice that if l is ~~uniform~~constant:

$$\nabla K_{ab} = \nabla \left(l^d G_{ab} \right) = l^d \nabla G_{ab} \tag{70}$$

The discretized momentum equation with $SPH - \varepsilon$ turbulence model can be written in terms of G_{ab} instead of K_{ab}:

$$\frac{d\mathbf{v}_a}{dt} = -\sum_b \left[m_b \left(\frac{p_b}{\rho_b^2} + \frac{p_a}{\rho_a^2} \right) \nabla_a w_{ab}(h) \right] + \sum_b m_b \Phi_{ab} \nabla_a G_{ab}(l) \tag{71}$$

where

$$\Phi_{ab} = \frac{\varepsilon}{2} \frac{\mathbf{v}_{ab} \cdot \mathbf{v}_{ab}}{\rho_b} \tag{72}$$

Which is the extra stress term induced by average. We take coefficient ε as 0.8 following Monaghan (2011).

For compressible flow, the energy equation is coupled with the momentum equation and mass conservation equation. Averaging of thermal energy over space introduces some additional terms besides the stress term induced by velocity average.

(Rumsey, 2014). The averaged momentum equation for compressible flow are in the same form as that for incompressible flow, all of other additional terms, besides the corresponding velocity average induced stress term, show up in the energy equation. Most turbulence modeling focuses on the stress terms induced by average of velocity. These stress terms are usually either solved directly (for example, LANS methods) or defined via a constitutive relation (for example, large eddy simulation method). Less attention is typically given to the other terms. Most commonly, a Reynolds analogy is used to model the turbulent exchange. 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 (Cebeci, 2013). We adopt this strategy. The additional terms associated with molecular diffusion and turbulent transport in the energy equation are either modeled in different ways or neglected sometimes (Rumsey, 2014). We neglect these terms in our simulation.

3.7.2 Turbulent heat transfer

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

$$Pr = \frac{C_p \mu}{\kappa} \tag{73}$$

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

$$\mu = \rho \nu \tag{74}$$

Then

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

Typical value of Pr_t for air is $0.7 \sim 0.9$. We take $Pr_t = 0.85$ for gases as recommended Kays (1994) from summarizing of experimental results.

Monaghan (2005) 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, Eq. (71), is the turbulent shear stress term. Recall that molecular viscosity can be discretized with SPH as shown in Eq. (38). It has been shown that the discretized molecular viscosity has both bulk viscosity and shear viscosity, where shear viscosity coefficient is (Monaghan, 2005):

$$\nu_t = S\nu \tag{76}$$

with

$$S = \begin{cases} \dfrac{1}{10} & if \quad d = 3 \\[2ex] \dfrac{1}{8} & if \quad d = 2 \end{cases} \tag{77}$$

The turbulent viscosity coefficient can be inferred from that formulation if we can reformulate the turbulent shear stress term in a form which is similar to the molecular shear term. ~~Reformulate~~ Reformulating the turbulent shear stress term:

$$\sum_b \frac{\varepsilon}{2} \frac{m_b}{\rho_b} \mathbf{v}_{ab} \cdot \mathbf{v}_{ab} \nabla_a G_{ab}(l_a) = \sum_b \frac{\varepsilon}{2S} m_b \frac{\mathbf{v}_{ab}}{\rho_b} \frac{S\mathbf{v}_{ab} \cdot \mathbf{x}_{ab}}{x_{ab}^2} \frac{x_{ab}^2}{\mathbf{x}_{ab}} \nabla_a G_{ab}(l_a) \tag{78}$$

~~Then~~ then the turbulent viscosity coefficient can be inferred from Eq. (78).

$$\nu_t = \frac{\varepsilon}{2S} \frac{\mathbf{v}_{ab} \cdot \mathbf{x}_{ab}}{\rho_b} \tag{79}$$

Please note that turbulent viscosity term has opposite sign with molecular viscosity term in discretized momentum equation and there is a minus sign in the expression of Π_{ab}, and they cancel out.

However, the above equation is correct only for the 1D ~~situation~~ situations. For 2D or 3D, it is not easy to get an explicit expression. We adopt an alternative way: obtaining a value for each pair of particles instead of persisting on getting an analytical expression. Choosing the smoothing function as the same as SPH kernel and the smoothing length scale l as the same as smoothing length h, the ratio between turbulent shear stress and physical shear stress is:

$$\Upsilon_{ab} = \frac{\dfrac{\varepsilon}{2} \dfrac{\mathbf{v}_{ab} \cdot \mathbf{v}_{ab}}{\rho_b}}{\dfrac{S\nu}{\rho_{ab}} \dfrac{\mathbf{v}_{ab} \cdot \mathbf{x}_{ab}}{x_{ab}^2 + \eta^2 h_{ab}^2}}$$
$$= \frac{\varepsilon \left(x_{ab}^2 + \eta^2 h_{ab}^2 \right)}{2S\nu} \frac{\mathbf{v}_{ab} \cdot \mathbf{v}_{ab}}{\mathbf{v}_{ab} \cdot \mathbf{x}_{ab}} \tag{80}$$

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

$$\nu_{t,ab} = \nu \Upsilon_{ab}$$
$$= \frac{\varepsilon \left(x_{ab}^2 + \eta^2 h_{ab}^2 \right)}{2S} \frac{\mathbf{v}_{ab} \cdot \mathbf{v}_{ab}}{\mathbf{v}_{ab} \cdot \mathbf{x}_{ab}} \tag{81}$$

The corresponding turbulent thermal conductivity should be

$$\kappa_{t,ab} = \frac{\varepsilon \overline{C_{p,ab}} \, \overline{\rho_{ab}} \left(x_{ab}^2 + \eta^2 h_{ab}^2 \right) \mathbf{v}_{ab} \cdot \mathbf{v}_{ab}}{2S Pr_t \mathbf{v}_{ab} \cdot \mathbf{x}_{ab}} \tag{82}$$

$\overline{C_{p,ab}}$ and $\overline{\rho_{ab}}$ are simply the arithmetic means of specific heat and density. The term used to prevent singularity now can be removed.

$$\kappa_{t,ab} = \frac{\varepsilon \overline{C_{p,ab}} \, \overline{\rho_{ab}} x_{ab}^2 \mathbf{v}_{ab} \cdot \mathbf{v}_{ab}}{2S Pr_t \mathbf{v}_{ab} \cdot \mathbf{x}_{ab}} \tag{83}$$

We also need to prevent singularity, so:

$$\kappa_{t,ab} = \begin{cases} 0 & if \quad \mathbf{v}_{ab} = 0 \quad or \quad \mathbf{x}_{ab} = 0 \\ \dfrac{\varepsilon \overline{C_{p,ab}} \, \overline{\rho_{ab}} x_{ab}^2 \mathbf{v}_{ab} \cdot \mathbf{v}_{ab}}{2S Pr_t \mathbf{v}_{ab} \cdot \mathbf{x}_{ab}} & \text{otherwise} \end{cases} \tag{84}$$

Heat conduction equation without source term is:

$$C_p \frac{dT}{dt} = \frac{1}{\rho} \nabla (\kappa \nabla T) \tag{85}$$

Second spatial derivative can be approximated with SPH by following Monaghan (2005)~~.~~:

$$C_p \frac{dT}{dt} = \sum_b \frac{m_b}{\rho_a \rho_b} (\kappa_a + \kappa_b)(T_a - T_b) F_{ab}(h) \tag{86}$$

~~Where~~ where $F_{ab}(h)$ is short for $F(\mathbf{x}_a - \mathbf{x}_b, h)$, whose definition is:

$$F_{ab}(h)\mathbf{x}_{ab} = \nabla_a w_{ab}(h) \tag{87}$$

F_{ab} is always nonpositive, which guarantees that heat flux flows from hot to cold. Plug the turbulent thermal conductivity into the heat conduction equation:

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

$$= 2 \sum_b \frac{m_b}{\rho_a \rho_b} \frac{\overline{C_{p,ab}} \overline{\rho_{ab}} \varepsilon x_{ab}^2 \mathbf{v}_{ab} \cdot \mathbf{v}_{ab}}{2 Pr_t S \mathbf{v}_{ab} \cdot \mathbf{x}_{ab}} (T_a - T_b) F_{ab}(h) \tag{88}$$

Notice that the number "2" in the front of Eq. (88) comes from integration approximation of second order derivative (Cleary and Monaghan, 1999). By further simplification, we get:

$$C_p \frac{dT}{dt} = \frac{\varepsilon}{S Pr_t} \sum_b \frac{m_b}{\rho_a \rho_b} \frac{\overline{C_{p,ab}} \overline{\rho_{ab}} x_{ab}^2 \mathbf{v}_{ab} \cdot \mathbf{v}_{ab}}{\mathbf{v}_{ab} \cdot \mathbf{x}_{ab}} (T_a - T_b) F_{ab}(h) \tag{89}$$

3.7.3 Discretized governing equations with $SPH - \varepsilon$ turbulence model

Plugging in the discretized turbulent stress term and turbulent heat transfer term into the momentum and energy equation, we get new discretized governing equations:

$$\left\langle \frac{d\mathbf{v}_\alpha}{dt} \right\rangle = -\sum_b \left[m_b \left(\frac{p_b}{\rho_b^2} + \frac{p_\alpha}{\rho_\alpha^2} + \Pi_{\alpha b}^\beta - \Phi_{\alpha b} \right) \nabla_\alpha w_{\alpha b}(h) \right] - \sum_j \left[m_j \left(\frac{p_j}{\rho_j^2} + \frac{p_\alpha}{\rho_\alpha^2} + \Pi_{\alpha j}^\beta - \Phi_{\alpha j} \right) \nabla_\alpha w_{\alpha j}(h) \right] + \mathbf{g} \tag{90}$$

With

$$\Phi_{\alpha\beta} = \frac{\varepsilon}{2} \frac{\mathbf{v}_{\alpha\beta} \cdot \mathbf{v}_{\alpha\beta}}{\rho_\beta} \tag{91}$$

$$\left\langle \frac{de_\alpha}{dt} \right\rangle = 0.5 \sum_b \left[m_b \widehat{\mathbf{v}_{\alpha b}} \left(\frac{p_b}{\rho_b^2} + \frac{p_\alpha}{\rho_\alpha^2} + \Pi_{\alpha b}^\beta - \Phi_{\alpha b} \right) \nabla_\alpha w_{\alpha b}(h) \right] + 2 \sum_b \frac{m_b}{\rho_\alpha \rho_b} \kappa_{t,\alpha b} (T_\alpha - T_b) F_{\alpha b}(h)$$

$$+ 0.5 \sum_j \left[m_j \widehat{\mathbf{v}_{\alpha b}} \left(\frac{p_j}{\rho_j^2} + \frac{p_\alpha}{\rho_\alpha^2} + \Pi_{\alpha j}^\beta - \Phi_{\alpha j} \right) \nabla_\alpha w_{\alpha j}(h) \right] + 2 \sum_j \frac{m_j}{\rho_\alpha \rho_j} \kappa_{t,\alpha j} (T_\alpha - T_j) F_{\alpha j}(h) \tag{92}$$

with $\kappa_{t,\alpha\beta}$ given by Eq. (84). 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 is done according to Eq. (67). Position of particles is updated according to Eq. (63). Smoothed velocity is also used while computing artificial viscosity.

3.8 Boundary conditions

All boundary conditions are imposed by ghost particles. Subfigure b of Fig. 2 shows how boundaries are deployed.

3.8.1 Wall boundary condition

Traditionally either ghost particles that mirror real particles across the boundary (Ferrari et al., 2009) or boundary forces (Monaghan and Kajtar, 2009) have been used to impose the wall boundary conditions. One disadvantage of the latter is that the boundary forces tend to corrupt the solution in the local neighborhood. In addition, a 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 the ghost particle method (Kumar et al., 2013) for wall boundary conditions. Stationary wall ghost particles are deployed in the same way as real particles. Instead of enforcing symmetry particle by particle, a symmetric field across the boundary is explicitly enforced. Ghost particles are reflected into the domain and physical quantities are calculated at these reflected positions by SPH interpolations. It should be noted that wall ghost particles should not be counted when computing physical properties of these reflected positions. Assign all properties, except for velocity, at the corresponding reflected position to the ghost particle. The velocity of each wall ghost particles is set to have the same value but opposite direction of the interpolated velocity at its corresponding reflection. By this way, the no-slip wall boundary condition (Eq. (22)) is imposed naturally. These wall ghost particles serve as neighbors in momentum and energy update. More implementation details about this method can be found in (Kumar et al., 2013). As these wall ghost particles are stationary, there is no mass flux on the boundary (Eq. (20) and (21)). In addition, as temperature is also symmetric with respect to the boundary, the gradient of temperature vanishes and hence there is no internal energy flux on the wall boundary (Eq. (23)). In our current model, the ground is assumed to be flat. For more complicated topography, it has been shown in other work (Kumar et al., 2013) that this method works as well. We do have concern regarding potential limitation of this method of deployment of ghost particles for more complicated boundaries in three dimensions. Fortunately, current model does not involve complicated wall boundary.

3.8.2 Eruption boundary condition

A natural way of imposing eruption boundary condition is using ghost particles that move with the eruption velocity and bear the temperature of the erupted material. A parabolic velocity profile that represents a fully developed Hagen-Poiseuille flow is used to determine the inlet particle velocity. The detailed shape of the parabolic profile is determined based on an averaged eruption velocity (Eq. (18)). The mass of eruption ghost particles is set to a value so that evaluation of Eq. (46) can return us a density that is consistent with the value given in Eq. (16). The internal energy associated with these particles are set to a value so that Eq. (19) is satisfied. The mass fraction of erupted material (Eq. (17)) is automatically satisfied as all particles in the eruption conduit are of phase 2. The density, momentum, and internal energy of these eruption ghost particles are not updated before they move above ground. As soon as they move out from ~~erupt~~ eruption conduit, these ghost particles will be shifted to real particles and their physical quantities and position will be updated based on discretized governing equations. New ghost particles need to be added at the bottom of the eruption conduit as these existing ghost particles move upwards.

3.8.3 Pressure boundary condition

Another boundary condition in our model is the 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 condition (Eq. (24)) is more suitable for plume simulation as the outlet is open atmosphere. The way we impose pressure boundary condition is: adding several layers of pressure ghost particles surrounding the real atmosphere particles. Pressure, density and temperature are determined based on the elevation of particles. Velocity is set to zero for static atmosphere. The physical quantities for pressure ghost particles are not updated while these for real particles are updated at every time step. As position of all pressure ghost particles ~~keep~~ keeps constant, we essentially impose a static pressure boundary condition. Real particles are removed as soon as they move out of the pressure boundary.

As simulations progress, changes in position and physical quantities of real particles near pressure boundaries might corrupt pressure boundary condition that was established initially. This shortcoming is relieved by choosing a larger computational domain so that boundaries that might be corrupted locate far away from turbulent mixing area. In addition, to avoid enlarging fluctuations, we add another constraint on the time step:

$$\Delta t \leq CFL_p \frac{h}{v} \tag{93}$$

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

3.9 Parallellism and performance

One disadvantage of the 3D model is that it usually takes a much longer time than 1D models to complete one simulation. This disadvantage further prevents doing simulation with finer resolution and accounting for more physics in one model. Non-intrusive uncertainty analysis, which is commonly adopted in hazard forecasting, requires finishing multiple simulations within a given time window. High performance computing is therefore essential. Among existing CPU parallel SPH schemes, most of them focus on neighbor ~~searching~~ search algorithm and dynamic load balancing ~~(eg. Ferrari et al., 2009; Crespo et al., 2015)~~ (Ferrari et al. . Less attention has been paid to developing more flexible data management schemes for more complicated problems. Motivated by techniques developed for mesh based methods, we develop a complete framework for parallelizing SPH with distrubted memory parallelism (MPI) allowing flexible and efficient data access.

~~Any implementation of SPH code requires efficient searching and updating of neighbors during simulation~~T The time complexity for an SPH method is $O(N^2)$, where, N is the total number of SPH particles. Efficient neighbor searches and compact-supported kernel functions can help to reduce computational cost.. Of the many possible choices we adopt a background grid which was proposed by Monaghan and Lattanzio (1985) and is quite popular in parallel SPH. Then the time complexity is reduced to $O(MN + mN)$. Where m is the average of number of particles within the compact support of kernel

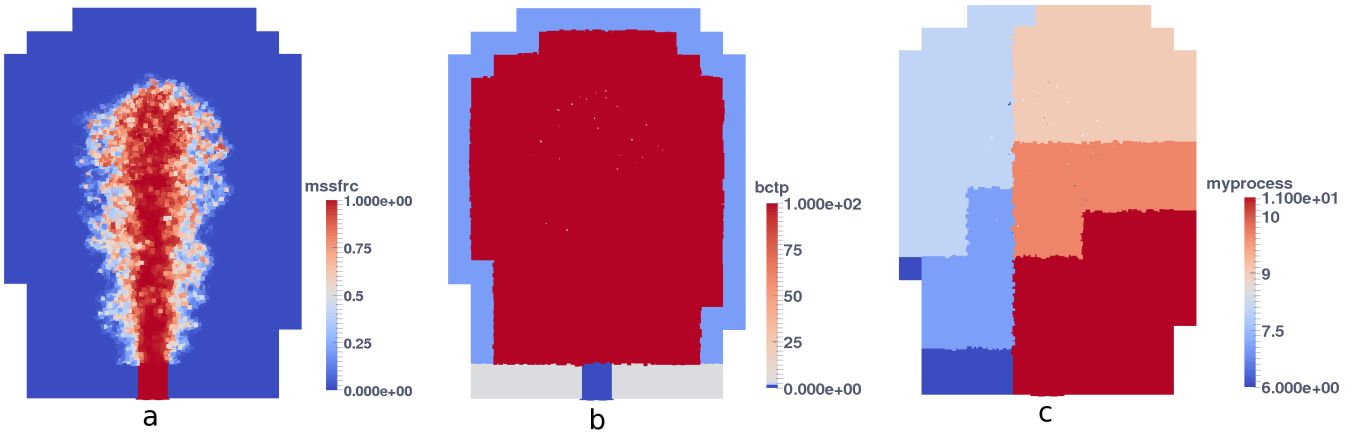



Figure 2. A cross section view of the simulation domain in $y - z$ plane at 66 seconds. Subfigure a shows the mass fraction. Subfigure b shows all boundary conditions: the dark blue region are occupied by eruption ghost particles with "Ghost particle ID" of 0, the light blue area are occupied by pressure ghost particles with "Ghost particle ID" of 1, the gray area is filled with wall ghost particles with "Ghost particle ID" of 2, "Ghost particle ID" of all real particles are set to 100, they occupy the major portion of the domain in subfigure b. Subfigure c shows the cross section view of domain decomposition based on SFC. The simulation is conducted on 12 processors, so there are 12 subdomains in total. The crosss section view shows a portion of them.

function, M is average number of particles in sub-domain within which neighbour searching is carried out. This background grid is also used for domain decomposition~~in SPH~~. We refer to the elements of background grid, namely squares for two dimensions and cubic for three dimensions, as buckets. As for the actual storage of the physical quantities associated to each particle, different strategies have been adopted in existing implementations of SPH.

5   In DualSPHysics (Crespo et al., 2015), the physical quantities of each particle (position, velocity, density...) are stored in arrays. 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 easy 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 et al. (2009) adopted linked lists using pointers so that particles can be deleted or added during the simulation. Storage

10   problems caused by fixed-size arrays are thereby also eliminated. We define C++ classes which contain all data of particles and buckets. As for the management of data, we adopt hash tables to store pointers to particles and buckets, which gives us not only flexibility of deleting and adding of elements, but also quicker access compared with linked lists. Instead of using the "natural order" to number particles, we adopt a space filling curve (SFC) based index to give each particle and background bucket an unique identifier – a strategy known to preserve data locality at minimal cost. The SFC based numbering strategy is

15   further extended to include time step information so that particles added at the same position but different time have different identifiers.

~~As for domain decomposition, even~~ The parallelization is achieved by splitting computational domain into subdomains. Each subdomain is computed by a single processor. For any subdomain, information from its neighbouring subdomains is required

when updating physical quantities. To guarantee consistency, data is synchronized if physical quantities are updated. Even though more complicated graph-based partitioning tools (Biswas and Oliker, 1999) might get higher quality decomposition, they requires much more effort in programming and computation. So we adopt an easy-programming scheme based on SFC (Patra and Kim, 1999) to decompose the computational domain. Subfigure c of Fig. 2 shows a cross section view of domain decomposition.

More details about the data structure, domain decomposition, load balancing~~and domain adjusting will be published separately~~, domain adjusting and performance benchmarking has been published separately (Cao et al., 2017) .

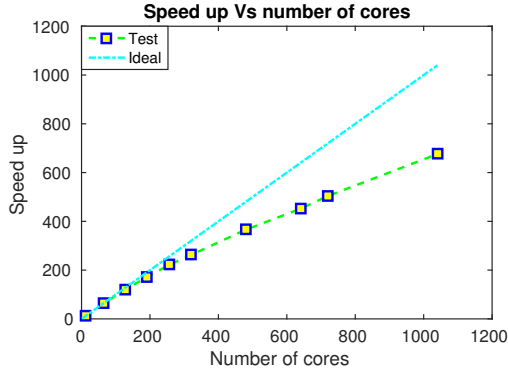
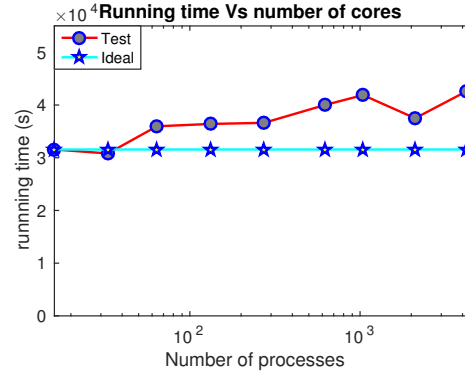



Figure 3. Strong scalability test result   **Figure 4.** Weak scalability test result

Performance tests have been carried out on the computational cluster of CCR (Center for Computational Research) at University at Buffalo. Intel Xeon 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 is comprised of two sockets with 6 of these cores. Memory and level 3 cache are shared on each node. The initial domain is ~~$[-4.8km, 4.8km] \times [-4.8km, 4.8km] \times [0km, 6km]$~~$[-4.8km, 4.8km] \times [-4.8km, 4.8km] \times [0km, 6km$ Almost linear speed up is observed in our strong scalability test (Fig. 3).

The weak scalability test is conducted with the same initial domain and various smoothing lengths. Each simulation runs for 400 time steps. The average number of real particles of each process keeps constant at 25900. As shown in Fig. 4, simulation times increase around $1/3$ when number of cores increase from 16 to 4208. For the test problem in this section, the volcanic plume finally reaches to a region of ~~$[-30km\ 30km] \times [-30km\ 30km] \times [1.5km\ 40km]$~~$[-30km, 30km] \times [-30km, 30km] \times [1.5km, 40$ after around 400 seconds of eruption. When numerical simulation goes up to 90 seconds, the plume is still within a region of ~~$[-10km\ 10km] \times [-10km\ 10km] \times [0km\ 25km]$~~$[-10km, 10km] \times [-10km, 10km] \times [0km, 25km]$. This implies that adjusting of domain can avoid computing large number of uninfluenced air particles, especially for the beginning stage of simulation. A domain adjusting algorithm (Cao et al., 2017) is designed and implemented in our code. Figure 5 shows that simulation time of the test problem is greatly reduced when we adopt the domain adjusting strategy.

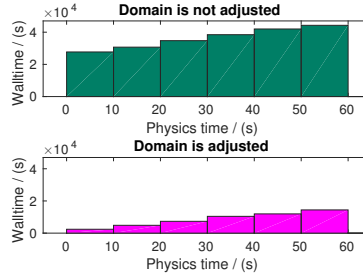



Figure 5. The effect of domain adjusting on simulation time. The figure on the top shows execution time without domain adjusting, the figure on the bottom shows execution time with domain adjusting. Different bins represent execution time up to specific physical time indicated by horizontal axis.

4 Verification and validation

We present a series of numerical simulations to verify and validate our model in this section. Plume-SPH is first verified by 1D shock tube tests then by a JPUE (jet or plume that is ejected from a nozzle into a uniform environment) simulation. Velocity and mass fraction distribution both along the central axis and cross transverse are compared with experimental results. The pattern of ambient particles entrainment is also clearly shown. Then a simulation of representative strong volcanic plume is conducted. ~~Both global variables and~~ Integrated local variable are comparable with simulation results from existing 3D plume models.

4.1 1D Shock Tube Tests

1D shock tube tests are first conducted to verify our code. Input parameter of each tests can be found in Table 1. These tests can represent typical cases in 1D. Test 1 consists of a left rarefaction, a right travelling contact and a right shock. Density decreases at down wind of contact wave. Test 2 also consists of a left rarefaction, a right travelling contact and a right shock. Density increases at down wind of contact wave. Test 3 is a double expansion test with different initial dnesity.

Table 1. Input parameters of 1D shock tube tests

	ρ_L	p_L	v_L	ρ_R	p_R	v_R	t_f
Test 1	1.0	1.0	0	0.5	0.2	0	0.2
Test 2	1.0	1.0	0	0.25	0.1795	0	0.17
Test 3	2.0	1.95	1.0	1.0	1.95	-1.0	0.13

In table 1, subscript L refers left side and R for right side, t_f is the total simulation time. The initial interval between two adjcent particles is 0.03. The computational domain is $[-0.4, 0.4]$. The specific internal energy is is compared against exact

solutions. As shown in Fig. 6, the position and magnitude of the waves are correctly predicted. The fluctuations near the contact discontinuity are caused by sharp change of smoothing length.

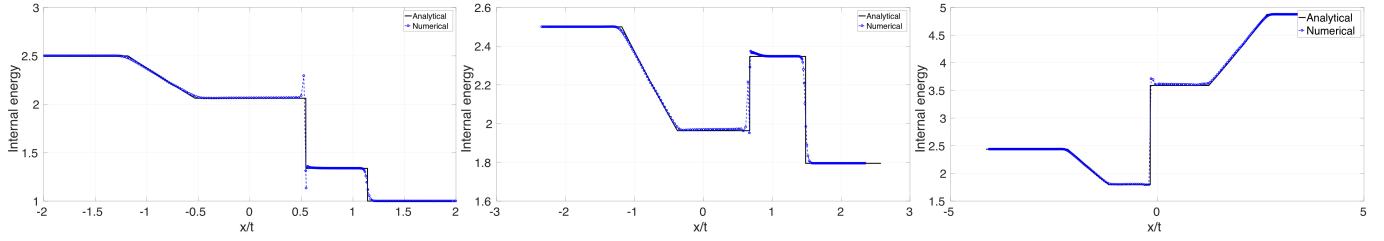



Figure 6. Comparison of specific internal energy of simulation results agaist analytical results for shock tube tests. The plots from left to right are corresponding to test 1, test 2 and test 3 respectively.

4.2 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. There exist consistently good experimental data ~~(List, 1982; Dimotakis et al., 1983; Papanicolaou and List, 1988)~~ (List, 1982; Dimotakis et al., 1983; Papanicolaou and List, 1988; Ezza describe the JPUE flow field giving an insight into details of JPUE, such as transverse velocity and concentration profile. In this section, we verify that our code and the $SPH - \varepsilon$ turbulence model is able to reproduce feature of turbulent entrainment by a JPUE simulation.

As many of these experiments were conducted with liquid, we replace the original equation of state (Eq. (5)) with a weakly compressible Tait equation of state (Becker and Teschner, 2007) (see Eq. (94)) to avoid solving the Poisson equation.

$$p = B \left[\left(\frac{\rho}{\rho_0} \right)^{\gamma} - 1 \right] \tag{94}$$

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

$$B = \frac{\rho_0 c^2}{\gamma} \tag{95}$$

~~Where~~ where c is the speed of sound in the liquid. The energy equation is actually decoupled from the momentum conservation equation and the mass conservation equation by using this ~~equation of state (EOS)~~ EOS. In addition, the "atmosphere" is assumed to be uniform and gravity is set to be zero. We set the temperature and density of ejected material the same as surrounding ambient. This further simplifies the scenario for the convenience of studying turbulent mixing.

One overall feature of JPUE is "self-similarity" which means that the evolution of the JPUE is determined solely 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 velocity at each height. As a results, physical and numerical experiments do not necessary to have exactly the same setups and are compared on a non-dimensional basis.

Table 2. List of eruption condition for the test cases

Parameters	Units	JPUE	Plume
Vent velocity	$m \cdot s^{-1}$	500	275
Vent gas mass fraction		1.0	~~0.5~~ 0.05
Vent Temperature	K	273	1053
Vent height	m	0	1500
Mass discharge rate	$kg \cdot s^{-1}$	5.47×10^{7}	1.5×10^{9}

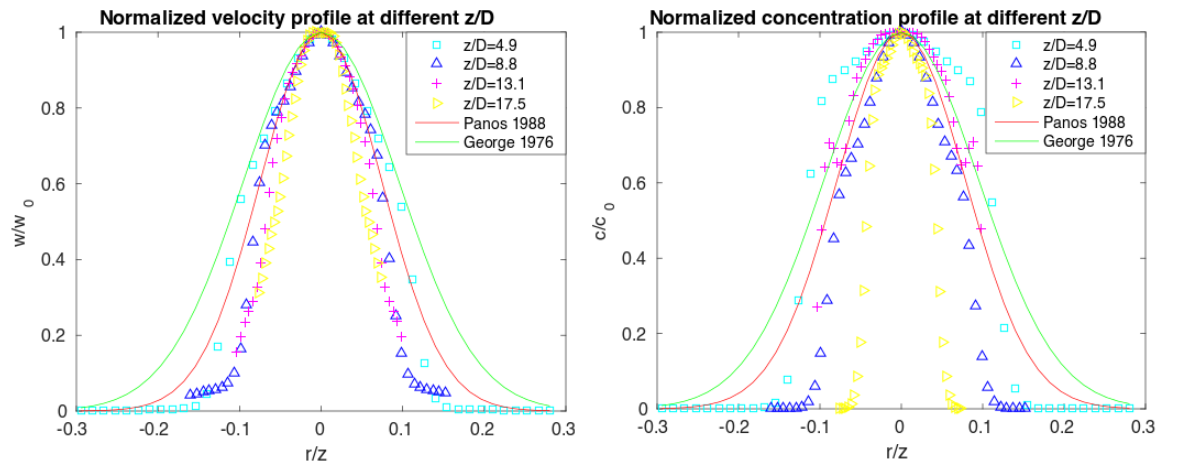



Figure 7. Dimensionless concentration and velocity distribution across the cross-section.

~~Dimensionless concentration distribution across the cross-section.~~

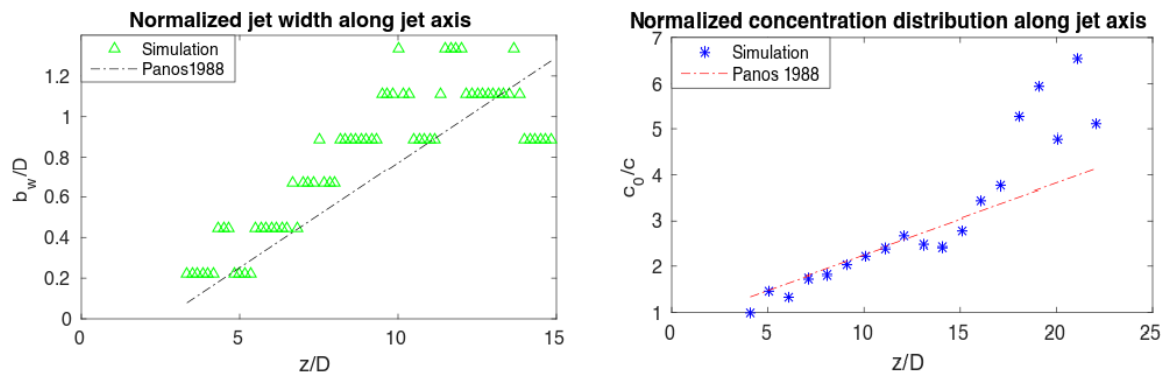



Figure 8. ~~Dimensionless~~ The left plot is normalized jet width (which determined based on velocity~~distribution~~) along ~~centraline~~the centerline. The right plot shows normalized concentration along the centerline.

A three dimensional axisymmetric JPUE which ejects from a round vent is simulated with eruption parameters listed in Table 2. Material properties of water are used as material properties for both phases. The results are compared with experiments (George et al., 1977; Papanicolaou and List, 1988) for ~~validation~~ verification purposes. Experimental data of concentration and velocity distribution across the cross-section were fit into a Gaussian profile ~~:~~ $\frac{\varphi}{\varphi_c} = exp\left[-coef\left(\left(\frac{r}{z}\right)^2\right)\right]$ ~~(solid line)by Papanicolaou and List (1988)~~ (See Eq. (96)) by Papanicolaou and List (1988) and George et al. (1977) even though the actual profile are slightly different from the Gaussian profile. ~~Papanicolaou and List (1988) also fit concentration distribution and jet width based on velocity along central line into a straight line :~~ $\frac{\varphi_0}{\varphi_c} = slope\left(\frac{z}{D} + intercept\right)$ ~~(dash line).~~

$$\frac{\varphi}{\varphi_c} = exp\left[-coef\left(\left(\frac{r}{z}\right)^2\right)\right] \qquad (96)$$

where φ is either velocity or concentration, the subscript c represents the centreline~~, subscript 0 represents the cross-sectionally averaged exit value~~. r is the distance from the centreline on any cross-section. z is the axial distance from the origin of the jet transverse section under consideration. ~~D is the diameter of vent.~~ The coefficient $coef$ for concentration is 80 and 50 respectively according to George et al. (1977) and Papanicolaou and List (1988). $coef$ for velocity is 90 and 55 respectively according to George et al. (1977) and Papanicolaou and List (1988).

Papanicolaou and List (1988) also fit concentration distribution and jet width based on velocity along centerline into a straight line (See Eq. (97))..

$$\frac{\varphi_0}{\varphi_c} = slope\left(\frac{z}{D} + intercept\right) \qquad (97)$$

where subscript 0 represents the cross-sectionally averaged exit value, D is the diameter of vent. $slope$ for jet width based on velocity is 0.104 and for concentration is 0.157. $intercept$ for jet width based on velocity is 2.58 while that for the concentration is 4.35.

Although both velocity and concentration are found to be well matched with experimental results, a small disparity in both velocity and concentration are observed near the boundary of the jet. Which is possibly caused by overestimating of the drag effect by standard SPH (Ritchie and Thomas, 2001). Ritchie and Thomas (2001) also proposed an alternative way for density update which relieved the overestimating of the drag effect. However, how well does his method conserve mass is not clear. There are several ~~factors that would~~ other factors that might also attribute to such disparity. Reynolds number is not reported in many experiments assuming a high enough Reynolds number. In addition, some details of the experiments, such as exit velocity profile and viscosity of the experimental liquid, are not reported. These factors prevent us from numerically reproducing these experiments in an exact way as they were conducted. However, the features of JPUE ~~is~~ are correctly reproduced with our code.

We also investigated the mixing due to turbulence in JPUE simulation by checking the mixture of the two phases. It is shown in Fig. 9 that the ejected material and ambient fluids are mixed through eddies at the outer shear region. And the inner dense

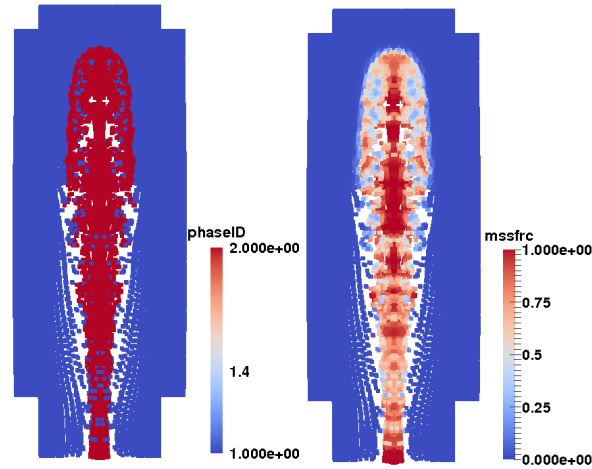



Figure 9. The left figure shows particle distribution. Particles of phase 1 (blue) are gradually entrained and mixed with erupted particles (red) as jet flows down stream. The right figure shows the mass fraction of erupted material at the moment corresponding to the left figure.

core dispersed gradually due to erosion of the outer shear region. Hence, our confidence in the numerical correctness of our code is greatly reinforced.

4.3 Simulation of a volcanic plume

The development of a volcanic plume is more complicated than JPUE in several aspects. Besides turbulent entrainment of am-
bient fluids, development of volcanic plume also involves heating up of entrained air and expanding in a stratified atmosphere. A strong eruption column without wind is tested in this section for the purpose of further verification and validation. Both global variable and local variables are compared with existing models.

4.3.1 Input parameters

Eruption parameters, material properties and atmosphere are chosen to be the same as the strong plume no wind case in a
comparison study on eruptive column models (Costa et al., 2016) by Costa et al. (2016) DIFaddend . Eruption conditions are listed in Table 2. As our model does not distinguish solid particles of different sizes, only mass fraction of solids of all size is used in simulation even though two size classes were provided by Costa et al. (2016) . The density of erupted material at the vent and radius of the vent can be computed from the given parameters. The eruption pressure is assumed to be as the same as pressure of ambient at the vent and hence is not given in the table. The vertical profiles of atmospheric properties were
obtained based on the reanalysis data from ECMWF (European Centre for Medium-Range Weather Forecasts) for the period corresponding to the climactic phase of the Pinatubo eruption (Philippines, 15 June 1991). These conditions are more typical of a tropical atmosphere (see Fig. 1B in (Costa et al., 2016)). Vertical distribution of temperature, pressure and density is used to establish stratified atmosphere. Wind velocity and specific humidity are not used in our simulation even though they were

also provided by Costa et al. (2016) (see Fig. 1B). Material properties, shown in Table 3, are selected based on properties of Pinatubo and Shinmoe-dake eruption. Other material properties not given in the table can be ~~infered~~ inferred from these given parameters based on their relationships.

Table 3. List of material properties

Parameters	Units	Value
Specific heat of gas at constant volume	$J \cdot kg^{-1} \cdot K^{-1}$	717
Specific heat of air at constant volume	$J \cdot kg^{-1} \cdot K^{-1}$	1340
Specific heat of solid	$J \cdot kg^{-1} \cdot K^{-1}$	1100
Specific heat of gas at constant pressure	$J \cdot kg^{-1} \cdot K^{-1}$	1000
Specific heat of air at constant pressure	$J \cdot kg^{-1} \cdot K^{-1}$	1810
Density of air at vent height	~~$kg \cdot m^{-2}$~~ $kg \cdot m^{-3}$	1.104
Pressure at vent height	Pa	84363.4

4.3.2 ~~Global and local variables~~

5 Figure 10 shows the mass fraction of simulated volcanic plume at 500s after eruption, at which time the plume starts spreading radially. A contour plot of the mass fraction on the vertical cross-section (x-z plane) were also provided. The zoomed view of the quiver plot shows detailed entrainment of air at the margin of the plume.
~~Density of the strong plume without wind after reaching its top height~~

4.3.2 Global and local variables

10 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 40 km which agrees with other plume models. For example, the height predicted by PDAC is 42500 m, by SK-3D is 39920 m, by ATHAM is 33392 m, by AHSEE is 36700 m. As for local variables, the profiles of integrated temperature, density, mass fraction of entrained air, gas mass fraction, mass fraction of solid and the radius of the plume as a function of height are compared with existing 3D models in Fig. 11 $\sim$ 14. To get rid of significant fluctuations in time and space we conducted a time 15 averaging and spatial integration of the dynamic 3D flow fields by following Cerminara et al. (2016b).

As particles distribute ~~in a disordered manner in~~ irregularly in the space in SPH simulation results. We ~~first~~ need to project simulation results (on ~~disordered~~ irregular particles) onto a pre-defined grid before doing time average and spatial integration. ~~The project method is the basic SPH kernel based interpolation~~See appendix A for more details of post processing.

The profiles of local variables match well with simulation results of existing 3D models in a general sense. The basic 20 phenomena in volcanic plume development is correctly captured by our model.
~~Radius of the strong plume without wind after reaching its top height.~~

As the height increases, the amount of entrained air also increases. Around the neutral height, where the umbrella expands, the entrainment of air shows a slight decrease due to lack of air surrounding the column at that height. The profile for gas,

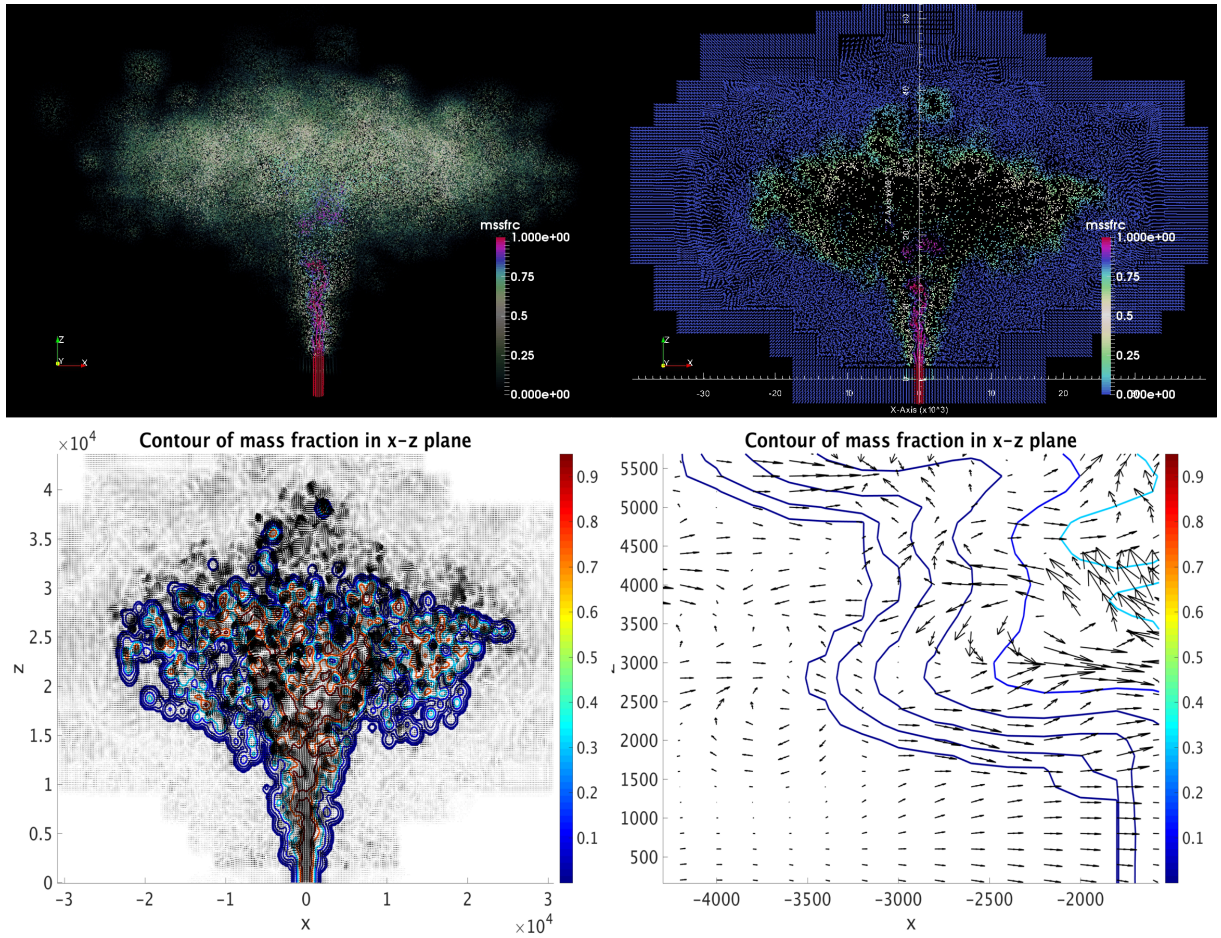



Figure 10. ~~Temperature as~~ Mass fraction for $t = 500s$ after eruption. Figures on the first row are visualization of SPH simulation results. The figure on the up right corner is visualization of a ~~function~~ slice of ~~height~~the computational, whose thickness is around $10000m$. The lowest portion of the plume represents erupted material in eruption vent (the underground portition). Figures at the second row show contour of mass fraction and velocity quiver on x-z plane. These figures are plotted utilizing post-processed data (see Appendix A). The contour level in the plot are $0.00001, 0.0001, 0.001, 0.1, 0.3, 0.75, 0.95$. The figure on the lower right corner is a zoomed view of velocity quivier showing plume expansion and entrainment of air.

which account for both air and vapor, shows a very similar tendency as that of entrained air. Recall that vapor condensation is not considered in our model. In addition, we assume that erupted material behaves like a single phase fluid. So the mass fraction of gas is simply a function of entrained air (Eq. (98)).

$$\xi_a + \xi_g = \xi_a + (1 - \xi_a)\xi_{g0} \tag{98}$$

5   Among these 3D models, ATHAM takes vapor condensation into account and Eq. (98) does not hold for ATHAM. However, the profile of entrained air and profile of gas predicted by ATHAM are still very close to each other which implies that ignoring

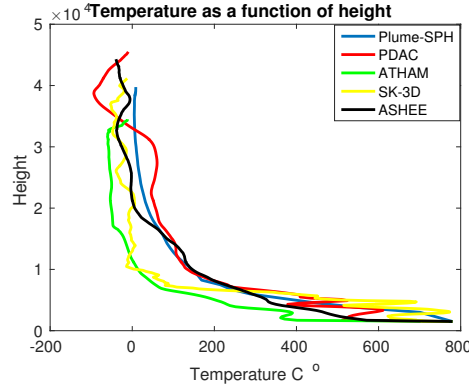



Figure 11. Temperature as a function of height

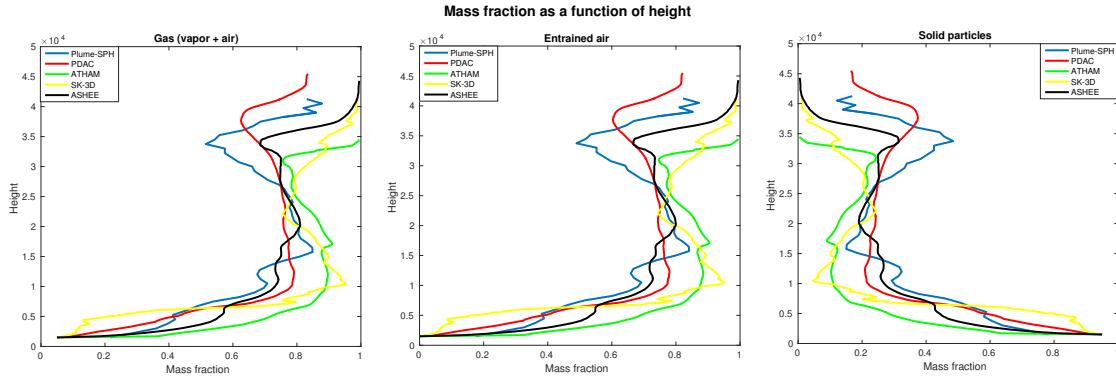



Figure 12. The mass fraction of entrained air, gas, and solid as a function of height.

of water phase change is a valid assumption for eruptions similar to this test case (strong plume with erupted water fraction in erupted material less than 5%). This observation can be explained by the fact that air occupies a larger portion of the gas and ignoring of phase change of vapor (which is only a small portion of gas) causes slight influence on plume development. As for mass fraction of solid, similarly, Eq. (99) and Eq. (100) hold for our model.

5 $$\xi_s = (1 - \xi_a)(1 - \xi_{g0}) \tag{99}$$

$$\xi_s = 1 - (\xi_a + \xi_g) \tag{100}$$

PDAC, which ~~treat~~ treats particles of two different sizes as two separate phases, predicted a similar mass fraction profile. That implies that assumption of dynamic equilibrium in our model is at least valid for eruptions similar to the test case.

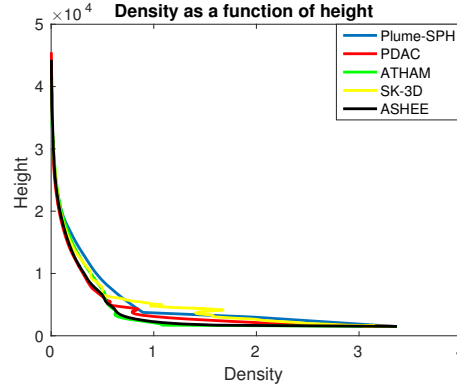



Figure 13. Density of the strong plume without wind after reaching its top height

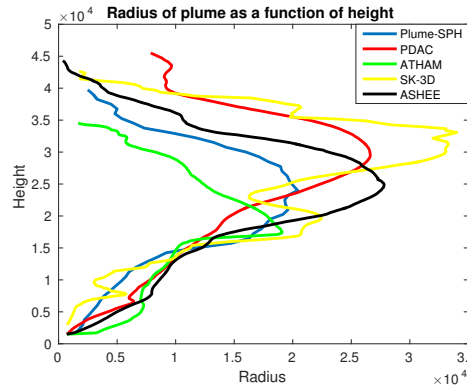



Figure 14. Radius of the strong plume without wind after reaching its top height.

$$\xi_a + \xi_g = \xi_a + (1 - \xi_a)\,\xi_{g0}$$

$$\xi_s = (1 - \xi_a)\,(1 - \xi_{g0})$$

5    $$\xi_s = 1 - (\xi_a + \xi_g)$$

    With more cool air entrained into the plume and mixing with the hot erupted material, the temperature of the plume decreases as the height increases as shown in Fig. 11. In the meanwhile, bulk density decrease due to entrainment and expansion (Fig. 13).

    Our model adopts the same assumptions and governing equations as SK-3D. However, there is still ~~a big~~ an obvious disparity

10   between the profiles of local variables of our model and SK-3D. One of the big differences between these two models is that we adopt a LANS type of turbulence model while SK-3D adopts a LES (large eddy simulation) turbulence model. This implies that choice of turbulence model might play a critical role in plume simulation.

5 Conclusions

A new plume model was developed based on the SPH method. Extensions necessary for Lagrangian methodology and compressible flow were made in the formulation of the equations of motion and turbulence models. Advanced numerical techniques in SPH were exploited and tailored for this model. High performance computing was used to mitigate the tradeoff between
5   accuracy (depends on comprehensiveness of the model, resolution, order of accuracy of numerical methods, scheme for time upgrading.) and simulation time (depends on comprehensiveness of model, resolution, order of accuracy of numerical methods, scheme for time upgrading ... and computational techniques). The correctness of the code and model was verified and validated by a series of test simulations. Typical 1D shock tube problems were simulated and compared against analytical results showing good agreement. Dimensionless velocity and concentration distribution across the cross-section and along the jet axis
10 match well with experimental results of JPUE. Top height and integrated local variables simulated by our model are consistent with simulation results of existing 3D plume models. Comparison between our results with these those of SK-3D implies that turbulence model plays a significant role in plume modeling.

Currently existing 3D models focus on certain aspect of the volcanic plume (PDAC on pyroclastic flow, ATHAM on microphysics, and SK-3D on entrainment with higher accuracy resolution and higher order of accuracy) and hence, naturally,
15 different assumptions were made in these models. However, these different aspects of volcanic plumes are not independent, but are actually coupled. For example, it has been illustrated by Cerminara et al. (2016b) that gas-particle non-equilibrium would introduce a previously unrecognized jet-dragging effect, which imposes great influence on plume development, especially for weak plumes. In addition, there is no absolute boundary to determine which kind of hazard is dominant in certain eruptions. So it is necessary to simulate all associated hazards in one model. Actually, effort has already been put on developing more
20 comprehensive plume models. For example, a large-particle module (LPM) was added to ATHAM to track the paths of rocky particles (pyroclastic or tephra) within the plume and predict where these particles fall (Kobs, 2009). We were also motivated by such an evolution of plume modeling to choose SPH as our numerical tool. Besides its ability in dealing with interfaces for multiphase flows, as mentioned in the introduction section, SPH method has good extensibility and adding new physics and phases requires much less modification of the code compared with mesh based methods. Last but not least, the dramatic devel-
25 opment of computational power makes it possible to establish a comprehensive model. While current computational capacity may not allow us to have a fully comprehensive model, the easy-extension feature of SPH makes it convenient to keep adding new physics into the model when necessary and computationally feasible.

We have presented in this paper an initial effort and results towards developing a first principle based plume model with comprehensive physics, adopting proper numerical tools and high performance computing. More advanced numerical techniques,
30 such as adaptive particle size, Godunov-SPH, semi-explicit time advancing scheme and better data management strategies and algorithms are on our list to exploit in the future. In the near future, effect of wind field will be take into account. Our code will also be made available in the open source form for the community to enhance. Besides improving plume model, coupling volcanic plume model with magma reservoir models (e.g. Terray et al., 2018) , which could provide more accurate eruption conditions, might improve accuracy of volcanic plume simulation.

5.1 Code availability

The Plume-SPH code, together with a user manual providing instructions for installation, running and visualization are archived at https://zenodo.org/record/572819#.WRCy7xiZORs (DOI: 10.5281/zenodo.572819). The input data for all simulations presented in this work are archived in the same repository. Output of simulations presented in this paper are archived in Box.
5   Access permission will be given upon request.

Appendix A: Post processing of particle data

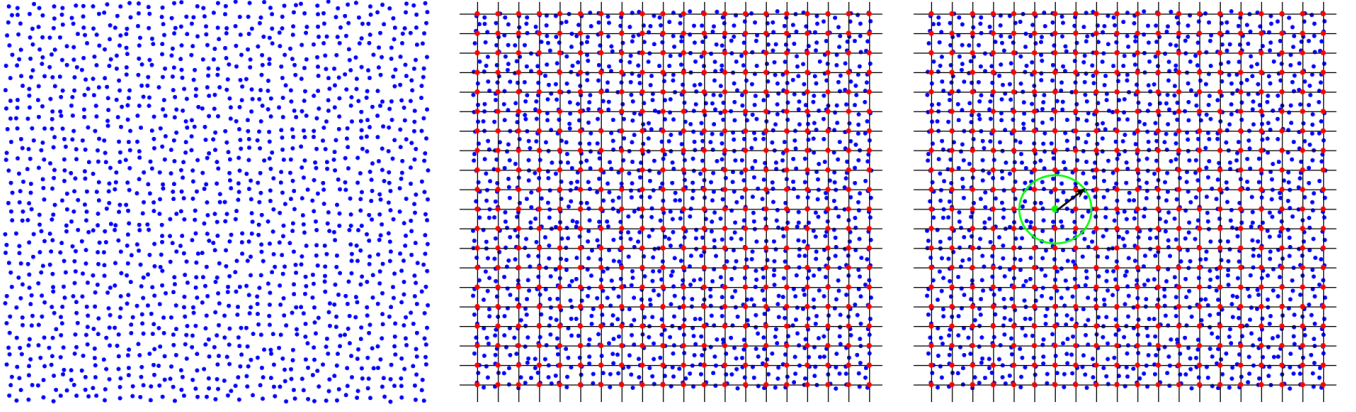



Figure A1. Procedure of projection of simulation results carried by particles onto regular grids. as shown in these figures from left to right: 1) raw data of SPH simulation results (irregularly distributed in space), 2) Add regular mesh, 3) searching for neighbours of each node (the blue SPH particles within the green circle around the red dot). The last step is not shown in these pictures, which is treating each node on the regular mesh as a SPH particle and projecting data on particles onto nodes utilizing "SPH interpolation" (see Eq. (31)).

Particles distribute irregularly in SPH simulation results. To adapt post processing originally proposed for mesh-based method, we need to project simulation results onto a pre-defined regular mesh. As shown in Fig. A1, the basic steps for such projection are:

10   – obtain raw simulation results carried by particles that irregularly distribute in the space

– create regular grids

– search for neighbour paricles for each node of the regular grids.

– interpolate physical quanties from neighbour particle onto corresponding node of regular grids according to Eq. (31)

Competing interests. Authors have the standard conflict with employers and listed sponsors

Acknowledgements. All developers of Titan-2D, especially Dinesh Kumar who developed the GSPH version of Titan-2D, are greatly appreciated as Plume-SPH is based on their code. Advice and data (which is not shown in this paper) given by Suzuki Yujiro gave us great help at the initial stage of model establishment and therefor are greatly appreciated. We appreciate Tomaso Esposti Ongaro for providing simulation data of PDAC (were also provided by Antonio Costa later together with simulation results of other plume models) which helped us in doing early verification and improvement. We appreciate Antonio Costa for providing simulation results of existing 3D models, which are used in verification and validation section of this paper. We thank Matteo Cerminara for his helping on post-processing of plume simulation results. Computational results reported here were performed at the Center for Computational Research at the University at Buffalo. This project is supported by Grants No. NSF ACI/1131074 from the National Science Foundation.

References

Adami, S., Hu, X., and Adams, N.: A new surface-tension formulation for multi-phase SPH using a reproducing divergence approximation, J. Comput. Phys., 229, 5011–5021, 2010.

Anderson, D., McFadden, G. B., and Wheeler, A.: Diffuse-interface methods in fluid mechanics, Annu. Rev. Fluid Mech., 30, 139–165, 1998.

Becker, M. and Teschner, M.: Weakly compressible SPH for free surface flows, in: Proceedings of the 2007 ACM SIGGRAPH/Eurographics symposium on Computer animation, pp. 209–217, Eurographics Association, 2007.

Biswas, R. and Oliker, L.: Experiments with repartitioning and load balancing adaptive meshes, in: Grid Generation and Adaptive Algorithms, pp. 89–111, Springer, 1999.

Bursik, M.: Effect of wind on the rise height of volcanic plumes, Geophys. Res. Lett, 28, 3621–3624, 2001.

Bursik, M., Martınez-Hackert, B., Delgado, H., and Gonzalez-Huesca, A.: A smoothed-particle hydrodynamic automaton of landform degradation by overland flow, Geomorphology, 53, 25–44, 2003.

Bursik, M. I.: bent: A model of plumes in crossflow, https://vhub.org/resources/267, 2010.

Bursik, M. I., Carbonara, A. U., and Zawicki, S. M.: puffin, https://vhub.org/resources/114, 2013.

Cao, Z., Patra, A., and Jones, M.: Data Management and Volcano Plume Simulation with Parallel SPH Method and Dynamic Halo Domains, Procedia Comput. Sci.,, 108, 786–795, 2017.

Carcano, S., Bonaventura, L., Esposti Ongaro, T., and Neri, A.: A semi-implicit, second-order-accurate numerical model for multiphase underexpanded volcanic jets, Geosci. Model Dev., 6, 1905–1924, 2013.

Carey, S. N. and Sigurdsson, H.: Influence of particle aggregation on deposition of distal tephra from the May 18, 1980, eruption of Mount St. Helens volcano, J. Geophys. Res-Sol. Ea., 87, 7061–7072, 1982.

Cebeci, T.: Analysis of turbulent flows with computer programs, Butterworth-Heinemann, 2013.

Cerminara, M., Esposti Ongaro, T., and Berselli, L.: ASHEE-1.0: a compressible, equilibrium-Eulerian model for volcanic ash plumes, Geosci. Model Dev., 9, 697–730, 2016a.

Cerminara, M., Esposti Ongaro, T., and Neri, A.: Large Eddy Simulation of gas–particle kinematic decoupling and turbulent entrainment in volcanic plumes, J. Volcanol Geoth. Res., 2016b.

Chen, J., Beraun, J., and Jih, C.: An improvement for tensile instability in smoothed particle hydrodynamics, Comput. Mech., 23, 279–287, 1999.

Chen, Z., Zong, Z., Liu, M., Zou, L., Li, H., and Shu, C.: An SPH model for multiphase flows with complex interfaces and large density differences, J. Comput. Phys., 283, 169–188, 2015.

Cheng, L. and Armfield, S.: A simplified marker and cell method for unsteady flows on non-staggered grids, International journal for numerical methods in fluids, 21, 15–34, 1995.

Cleary, P. W. and Monaghan, J. J.: Conduction modelling using smoothed particle hydrodynamics, J. Comput. Phys., 148, 227–264, 1999.

Colagrossi, A. and Landrini, M.: Numerical simulation of interfacial flows by smoothed particle hydrodynamics, J. Comput. Phys., 191, 448–475, 2003.

Costa, A., Suzuki, Y., Cerminara, M., Devenish, B., Esposti Ongaro, T., Herzog, M., Van Eaton, A., Denby, L., Bursik, M., de' Michieli Vitturi, M., et al.: Results of the eruptive column model inter-comparison study, J. Volcanol Geoth. Res., 2016.

Costa, A., Suzuki, Y., and Koyaguchi, T.: Understanding the plume dynamics of explosive super-eruptions, Nat. Commun., 9, 654, 2018.

Crespo, A., Domínguez, J., Rogers, B., Gómez-Gesteira, M., Longshaw, S., Canelas, R., Vacondio, R., Barreiro, A., and García-Feal, O.: DualSPHysics: Open-source parallel CFD solver based on Smoothed Particle Hydrodynamics (SPH), Comput. Phys. Commun., 187, 204–216, 2015.

de' Michieli Vitturi, M.: PlumeMoM, https://vhub.org/resources/3541, 2015.

5 Degruyter, W. and Bonadonna, C.: Improving on mass flow rate estimates of volcanic eruptions, Geophys. Res. Lett., 39, 2012.

de'Michieli Vitturi, M., Neri, A., and Barsotti, S.: PLUME-MoM 1.0: A new integral model of volcanic plumes based on the method of moments, Geosci. Model Dev., 8, 2447–2463, 2015.

Devenish, B.: Using simple plume models to refine the source mass flux of volcanic eruptions according to atmospheric conditions, J. Volcanol Geoth. Res., 256, 118–127, 2013.

10 Dimotakis, P. E., Miake-Lye, R. C., and Papantoniou, D. A.: Structure and dynamics of round turbulent jets, Phys. Fluids, 26, 3185–3192, 1983.

Draxler, R. and Rolph, G.: HYSPLIT (HYbrid Single-Particle Lagrangian Integrated Trajectory) Model access via NOAA ARL READY Website. NOAA Air Resources Laboratory, Silver Spring, http://www.arl.noaa.gov/HYSPLIT_info.php, 2015.

Ernst, G. G., Sparks, R. S. J., Carey, S. N., and Bursik, M. I.: Sedimentation from turbulent jets and plumes, J. Geophys. Res-Sol. Ea., 101, 15 5575–5589, 1996.

Esposti Ongaro, T., Cavazzoni, C., Erbacci, G., Neri, A., and Salvetti, M.-V.: A parallel multiphase flow code for the 3D simulation of explosive volcanic eruptions, Parallel Comput., 33, 541–560, 2007.

Ezzamel, A., Salizzoni, P., and Hunt, G. R.: Dynamical variability of axisymmetric buoyant plumes, J. Fluid Mech., 765, 576–611, 2015.

Ferrari, A., Dumbser, M., Toro, E. F., and Armanini, A.: A new 3D parallel SPH scheme for free surface flows, Comput. & Fluids, 38, 20 1203–1217, 2009.

Folch, A., Costa, A., and Macedonio, G.: FPLUME-1.0: An integral volcanic plume model accounting for ash aggregation, Geosci. Model Dev., 9, 431, 2016.

George, W. K., Alpert, R. L., and Tamanini, F.: Turbulence measurements in an axisymmetric buoyant plume, International Journal of Heat and Mass Transfer, 20, 1145–1154, 1977.

25 Gerlach, D., Tomar, G., Biswas, G., and Durst, F.: Comparison of volume-of-fluid methods for surface tension-dominant two-phase flows, International Journal of Heat and Mass Transfer, 49, 740–754, 2006.

Gingold, R. A. and Monaghan, J. J.: Smoothed particle hydrodynamics: theory and application to non-spherical stars, Mon. Not. R. Astron. Soc., 181, 375–389, 1977.

Gopala, V. R. and van Wachem, B. G.: Volume of fluid methods for immiscible-fluid and free-surface flows, Chem. Eng. J., 141, 204–221, 30 2008.

Haddad, B., Palacios, D., Pastor, M., and Zamorano, J. J.: Smoothed particle hydrodynamic modeling of volcanic debris flows: Application to Huiloac Gorge lahars (Popocatépetl volcano, Mexico), J. Volcanol Geoth. Res., 324, 73–87, 2016.

Harlow, F. H., Welch, J. E., et al.: Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface, Phys. Fluids, 8, 2182, 1965.

35 Hérault, A., Bilotta, G., Del Negro, C., Russo, G., and Vicari, A.: SPH modeling of lava flows with GPU implementation, From physics to control through an emergent view, World Scientific Series on Nonlinear Science, Series B, 15, 183–188, 2010.

Herzog, M., Graf, H.-F., Textor, C., and Oberhuber, J. M.: The effect of phase changes of water on the development of volcanic plumes, J. Volcanol Geoth. Res., 87, 55–74, 1998.

Herzog, M., Oberhuber, J. M., and Graf, H.-F.: A prognostic turbulence scheme for the nonhydrostatic plume model ATHAM, Journal of the Atmospheric Sciences, 60, 2783–2796, 2003.

Hirt, C. W. and Nichols, B. D.: Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys., 39, 201–225, 1981.

Holm, D. D.: Fluctuation effects on 3D Lagrangian mean and Eulerian mean fluid motion, Physica. D, 133, 215–269, 1999.

5   Hu, X. and Adams, N. A.: An incompressible multi-phase SPH method, J. Comput. Phys., 227, 264–278, 2007.

Issa, R.: Numerical assessment of the Smoothed Particle Hydrodynamics gridless method for incompressible flows and its extension to turbulent flows, 2005.

Kays, W. M.: Turbulent Prandtl number—where are we?, J. Heat Transf., 116, 284–295, 1994.

Kobs, S.: Modeling particle motion and near-vent deposition in explosive volcanic eruptions, State University of New York at Buffalo, 2009.

10   Koyaguchi, T., Ochiai, K., and Suzuki, Y. J.: The effect of intensity of turbulence in umbrella cloud on tephra dispersion during explosive volcanic eruptions: Experimental and numerical approaches, J. Volcanol Geoth. Res., 186, 68–78, 2009.

Kumar, D., Patra, A. K., Pitman, E. B., and Chi, H.: Parallel Godunov smoothed particle hydrodynamics (SPH) with improved treatment of Boundary Conditions and an application to granular flows, Comput. Phys. Commun., 184, 2277–2286, 2013.

List, E.: Turbulent jets and plumes, Annu. Rev. Fluid Mech., 14, 189–212, 1982.

15   Lucy, L. B.: A numerical approach to the testing of the fission hypothesis, Astron. J., 82, 1013–1024, 1977.

Mastin, L. G.: A user-friendly one-dimensional model for wet volcanic plumes, Geoche. Geophy. Geosy., 8, 2007.

Mastin, L. G.: Plumeria 2.3.1, https://vhub.org/resources/1194, 2011.

Monaghan, J.: On the problem of penetration in particle methods, J. Comput. Phys., 82, 1–15, 1989.

Monaghan, J.: SPH compressible turbulence, Mon. Not. R. Astron. Soc., 335, 843–852, 2002.

20   Monaghan, J.: Smoothed particle hydrodynamics, Rep. Prog. Phys., 68, 1703, 2005.

Monaghan, J.: Smoothed particle hydrodynamics and its diverse applications, Annu. Rev. Fluid Mech., 44, 323–346, 2012.

Monaghan, J. and Gingold, R.: Shock simulation by the particle method SPH, J. Comput. Phys., 52, 374–389, 1983.

Monaghan, J. and Kocharyan, A.: SPH simulation of multi-phase flow, Comput. Phys. Commun., 87, 225–235, 1995.

Monaghan, J. and Rafiee, A.: A simple SPH algorithm for multi-fluid flow with high density ratios, International Journal for Numerical
25   Methods in Fluids, 71, 537–561, 2013.

Monaghan, J. J.: Smoothed particle hydrodynamics, Annu. Rev. Astron. Astr., 30, 543–574, 1992.

Monaghan, J. J.: A turbulence model for Smoothed Particle Hydrodynamics, Eur. J. Mech. B-Fluid, 30, 360–370, 2011.

Monaghan, J. J. and Kajtar, J. B.: SPH particle boundary forces for arbitrary boundaries, Comput. Phys. Commun., 180, 1811–1820, 2009.

Monaghan, J. J. and Lattanzio, J. C.: A refined particle method for astrophysical problems, Astron. Astrophys., 149, 135–143, 1985.

30   Neri, A., Esposti Ongaro, T., Macedonio, G., and Gidaspow, D.: Multiparticle simulation of collapsing volcanic columns and pyroclastic flow, J. Geophys. Res-Sol. Ea. (1978–2012), 108, 2003.

Oberhuber, J. M., Herzog, M., Graf, H.-F., and Schwanke, K.: Volcanic plume simulation on large scales, J. Volcanol Geoth. Res., 87, 29–53, 1998.

Papanicolaou, P. N. and List, E. J.: Investigations of round vertical turbulent buoyant jets, J. Fluid Mech., 195, 341–391, 1988.

35   Patra, A. and Kim, D.: Efficient mesh partitioning for adaptive hp finite element meshes, in: In International Conference on Domain Decomposition Methods, Citeseer, 1999.

Pouget, S., Bursik, M., Singla, P., and Singh, T.: Sensitivity analysis of a one-dimensional model of a volcanic plume with particle fallout and collapse behavior, J. Volcanol Geoth. Res., 2016.

Price, D. J.: Smoothed particle hydrodynamics and magnetohydrodynamics, J. Comput. Phys., 231, 759–794, 2012.

Ritchie, B. W. and Thomas, P. A.: Multiphase smoothed-particle hydrodynamics, Mon. Not. R. Astron. Soc., 323, 743–756, 2001.

Rosswog, S.: Astrophysical smooth particle hydrodynamics, New Astron. Rev., 53, 78–104, 2009.

Rumsey, C.: Implementing Turbulence Models into the Compressible RANS Equations, http://turbmodels.larc.nasa.gov/implementrans.html, 2014.

Sigurdsson, H., Carey, S., and Espindola, J.: The 1982 eruptions of El Chichón volcano, Mexico: stratigraphy of pyroclastic deposits, J. Volcanol Geoth. Res., 23, 11–37, 1984.

Suzuki, Y. and Koyaguchi, T.: A three-dimensional numerical simulation of spreading umbrella clouds, J. Geophys. Res-Sol. Ea. (1978–2012), 114, 2009.

Suzuki, Y. and Koyaguchi, T.: Numerical determination of the efficiency of entrainment in volcanic eruption columns, Geophys. Res. Lett., 37, 2010.

Suzuki, Y. J. and Koyaguchi, T.: 3D numerical simulation of volcanic eruption clouds during the 2011 Shinmoe-dake eruptions, Earth, Planets and Space, 65, 581–589, 2013.

Suzuki, Y. J., Koyaguchi, T., Ogawa, M., and Hachisu, I.: A numerical study of turbulent mixing in eruption clouds using a three-dimensional fluid dynamics model, J. Geophys. Res-Sol. Ea., 110, 2005.

Swegle, J., Hicks, D., and Attaway, S.: Smoothed particle hydrodynamics stability analysis, J. Comput. Phys., 116, 123–134, 1995.

Taddeucci, J., Scarlato, P., Montanaro, C., Cimarelli, C., Del Bello, E., Freda, C., Andronico, D., Gudmundsson, M., and Dingwell, D.: Aggregation-dominated ash settling from the Eyjafjallajökull volcanic cloud illuminated by field and laboratory high-speed imaging, Geology, 39, 891–894, 2011.

Terray, L., Gauthier, P.-J., Salerno, G., Caltabiano, T., Spina, A. L., Sellitto, P., and Briole, P.: A New Degassing Model to Infer Magma Dynamics from Radioactive Disequilibria in Volcanic Plumes, Geosci. J, 8, 27, 2018.

Textor, C., Graf, H.-F., Herzog, M., and Oberhuber, J.: Injection of gases into the stratosphere by explosive volcanic eruptions, J. Geophys. Res-Atmos. (1984–2012), 108, 2003.

Textor, C., Graf, H.-F., Herzog, M., Oberhuber, J. M., Rose, W. I., and Ernst, G.: Volcanic particle aggregation in explosive eruption columns. Part II: Numerical experiments, J. Volcanol Geoth. Res., 150, 378–394, 2006a.

Textor, C., Graf, H.-F., Herzog, M., Oberhuber, J. M., Rose, W. I., and Ernst, G. G.: Volcanic particle aggregation in explosive eruption columns. Part I: Parameterization of the microphysics of hydrometeors and ash, J. Volcanol Geoth. Res., 150, 359–377, 2006b.

Trentmann, J., Andreae, M., Graf, H.-F., Hobbs, P., Ottmar, R., and Trautmann, T.: Simulation of a biomass-burning plume: Comparison of model results with observations, J. Geophys. Res-Atmos. (1984–2012), 107, AAC–5, 2002.

Unverdi, S. O. and Tryggvason, G.: A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys., 100, 25–37, 1992.

Violeau, D. and Issa, R.: 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.

Weller, H. G., Tabor, G., Jasak, H., and Fureby, C.: A tensorial approach to computational continuum mechanics using object-oriented techniques, Comput. Phys., 12, 620–631, 1998.

Woodhouse, M., Hogg, A., Phillips, J., and Sparks, R.: Interaction between volcanic plumes and wind during the 2010 Eyjafjallajökull eruption, Iceland, J. Geophys. Res-Sol. Ea., 118, 92–109, 2013.

Woods, A.: The fluid dynamics and thermodynamics of eruption columns, B. Volcanol, 50, 169–193, 1988.

Wrobel, L. C. and Brebbia, C.: Computational Modelling of Free and Moving Boundary Problems: Fluid flow, vol. 1, Walter de Gruyter, 1991.

Youngs, D. L.: Time-dependent multi-material flow with large fluid distortion, Numerical methods for fluid dynamics, 24, 273–285, 1982.