1. Good morning, thanks everyone for coming to my defense. The topic of my dissertation is “smoothed particle hydrodynamics (SPH) method for compressible multiphase turbulent flow with application to volcanic plume model”
2. This is outline of today’s presentation, after an overview, I will basically follow the steps for developing the computational software, which is one outcome my dissertation research. That is, start from physics model, then numerical discretization with SPH method, and code architecture, data structure and parallelization. After coding work done, I will need to verify my programming and then validate the physics model. Then, the software is used to create initial conditions for volcano ash transport and dispersion simulation. Finally some comments on future.
3. Explosive eruption of volcano usually generate ash plume that raises up to several km or even tens of km. The deposition of volcanic ash, would pose great threaten to communities, destroy plants and crops, kill animals and threats the health, life and property of humans. These ashes can transport to a vast area in the atmosphere. If aircraft flying through the ash, its engine might be seriously damaged. You can see the blade of this engine is destroyed by ash. (So flights are usually re-scheduled or cancelled to avoid flying through the region coved by ash, causing big economic loss). In current work, we focus more on threats of volcanic ash to aviation.

(Reducing risk and the losses caused by explosive volcanic hazards using tools from mathematical and computational modeling is thus of great importance. ) It is a common practice to using volcanic plume model combined with an ash transport and dispersion models, which called as VATDs, to forecast the transportation of volcanic ash.

(3D volcanic plume model can also be used to investigate subtle feature of the plume development. )

There are four 3D plume models and around ten 1D plume models. There are also many VATDs available.

One major goal of my research is to develop a new 3D plume model based on SPH.

1. Then, someone might ask the question why do you need a new 3D model as there seem to be enough models already exist.

I can argue that all these existing 3D models are adopt mesh-based method. None of them is using mesh free method. So there is a gap.

However, this is not convictive enough. I was actually motivated by a good match between features of SPH and demands of volcano plume modeling. For example, volcano plume is in nature multiple phase and has no predefined boundary, while SPH, as a Lagrangian method is suitable for free boundary flow. And compared with mesh-based method, multiphase can be easily handled by SPH while it needs more numerical effort for mesh based method. The most attractive feature of SPH is its easy extensibility, which means it requires much less programming work to extend SPH solver to include more physics and more phases . This will make the software to be more sustainable in terms of adopting a more comprehensive model.

1. The main challenge of doing this work comes from two aspects. First of all SPH is still a developing tool. SPH still has unknown properties, and many questions remain unanswered on a purely theoretical ground, such as convergence, numerical stability, boundary conditions, kernel properties, time marching, existence, and properties of solutions. In addition, implementation of SPH in CFD field is limited to relative simple scenarios. The second aspect is the complexity of the object that I am trying to modeling. As have been mentioned, compressible, supersonic, turbulent and multiphase coupled with microphysics. (There are some researches on one of them, but no implementation accounting all of them. )

Particularly, there are few research on pressure outlet boundary condition and velocity inlet boundary condition, turbulence model in SPH are developed only for incompressible flow, explicit artificial viscosity is added to handle shocks and stabilize the simulation, however, it might introduce too much dissipation, smearing the discontinuities and overwhelm turbulent mixing. In addition, a so called “mixing issue” have been reported with SPH. There is still no conclusive solution for this issues, and mean while, there are still different opinions regarding the source of the mixing issue . (As turbulent mixing is one of the fundamental mechanism during plume development, I have to somehow live along with this issue.) In terms of software development, I encountered similar challenges, most implementation of SPH does not require flexible data management. So there is few research on that. But here, it is required.

1. Now let’s look at more details, I need to first establish the physics
2. Based on a series of assumption, including 1) well mixed, 2) thermal dynamic equilibrium, 3) Dynamic equilibrium 4) Ignorable wind effect

Finally got the governing equations for our model. Which is very similar to Euler equation besides an extra mass conservation equation. Two phases are used, one phase for atmosphere and another phase for erupted material. We assume that the mixture of erupted material and atmosphere will reach to a dynamic and thermal dynamic equilibrium states. So there is no separate moment and energy equations for each phase. Here, we only have one extra unknown, the mass fraction of the erupted material introduced by the second mass conservation equation.

The system of equations are closed by EOS for the mixture, where gama\_m is the specific heat ration of the mixture and it is updated at every time step based on mass fraction of the erupted material.

1. Now I have a batch of PDES, a closed system, mathematically I will need boundary conditions and initial conditions to get particular solution.

The computational domain is box and three different types of boundary conditions are required. The red circular area at the bottom is the eruption vent where erupted material comes out. So this is the eruption boundary condition, which is actually a velocity inlet boundary. All these green areas at the bottom are no-slip wall boundary. Pressure outlet boundary condition on all other five faces of the box.

With the boundary conditions together with governing equations, I have a complete mathematical description of the phenomena.

1. Since the governing equations are not the same as well studied PDEs in CFD, such as NS equation or Euler equation. It is necessary to investigate the basic properties of the system.

The eigenstructure analysis of Jacobian, shows that there are four characteristic fields, two non-linear, corresponding to eigenvalue u-c and u+c. The system also has two linear characteristic fields, both have the same eigenvalue. So this eigenstructure is very similar to that of Euler equations. The only extra characteristic field introduced by the extra equation is a linear field.

Elementary wave solution for Riemann problem demonstrate that

1. Only rho changes --> This is exact the same as Euler equations
2. Only xi changes
3. Xi does not change across shock (one type of waves associated with non-linear characteristic field)
4. Xi does not changes across rarefaction (Another type of waves associated with non-linear characteristic field)

Based on the last three conclusions, we can say that xi only changes when cross the second linear wave. In addition, rho, u, e (p) do not change across the contact discontinuity corresponding to xi. So the system is equivalent to two sub-systems that weakly coupled together, the first one is a system very similar to Euler equation, the second one is a linear hyperbolic system for updating mass fraction of erupted material. The updating of c\_m and gamma\_m requires mass fraction, while updating of mass fraction is based particles’ movement. (That is two say, updating of xi is decoupled from updating of rho, u, p.

However, sound speed and the specific heat ratio depend on xi, so the second PDE is not completely decoupled from the system. I would call it decoupled within each time step, so it is a weak coupling.)

1. We are done the physics model, to solve the PDEs numerically, I need to discretize it
2. Here is brief review of the classical SPH.

Any function A …. can be approximated by a weighted summation over all particles within the compact support of the weighting function, (Where w is …., usually the weighting function has a compact support,) w, whose radius is determined by smoothing length h.

The weighting function should be an approximation of delta function, when smoothing length tends to zero, it should tend to delta function. And the integration of the weighing function should be unit. There are several candidates for weighing functions, we choose the truncated Gaussian.

1. Here is one discretized formulation of Euler equation using SPH. It is derived based on SPH approximation of derivatives. Some tricks are used to get this symmetric formulation. And one advantage of such symmetric formulation is that it can guarantee momentum and energy conservation.

The time step is also constrained by CLF condition.

As a Lagarian method, the position of SPH particles are updated based on its velocity.

1. All what I talked about in previous slides are classical SPH. In now days, different types of remedies have be adopted in many implementation of SPH. In our plume modeling, we also include several remedies of regarding SPH

One issue of classical SPH is ….. To address this issue, we adopted a corrected formulation of SPH.

Starting from a talylor series expansion, ignore all these terms higher than first order derivative. We get this corrected formulation of SPH.

The difference between this corrected formulation and classical SPH formulation is that the new formulation can be viewed as a normalized form of SPH.  Recall that one property that the weighting function has to satisfy is the normalization condition. Namely, …

The weighting function that we choose would always satisfies the normalization condition in the integration form, but not necessary in the summation form. By using the corrected formulation we essentially enforce the normalization condition in the discretized formulation.

Similar story for derivatives.

1. Two phase are involved in my model, as for multiphase flow, one fundamental question is how to determine the interface? Actually as you will see, I do not need to worries about the interface at all in SPH.

We first estimate the single phase density of for each phase. For the single phase density of air, we do weighted summation over all air particle. And the same for erupted material. Then the mass fraction of erupted material is updated according to its definition.

The top figure shows the contour of mass fraction and the bottom figure shows contour of particle type. Interface is kind of automatically constructed.

1. One of the fundamental mechanism in volcanic plume development is entrainment of air due to turbulent mixing. And such turbulent mixing happens in various length scales ranging from mm to maybe to hundreds of meters. To capture the turbulent mixing, we can use either very fine resolution or some sub-particle scale turbulence model. We adopt the later option to avoid the high computational cost of the first method.

The turbulence model that we are using is the SPH-elpslon model.

Here is the basic idea this turbulence model. The velocity is filtered or averaged in space by a filtering operator. And such filtering will induce an extra stress term, which can be viewed as turbulent stress

The picture on the top shows a JPUE simulation without turbulence model, as you can see, there is no mixing

After adopting the SPH-epslon turbulence model, as you can I got mixing between two phases.

1. For compressible flow, the energy equation is coupled with momentum equation. If we carry out a similar filtering over internal energy. It will induce some other extra terms in the discretized energy equation. But it is a common practice to simply using a Reynolds analogy to get an equivalent turbulent heat exchange coefficient. That is the strategy that I adopted here.

Here is the idea of Reynolds analogy . The prandtl number indicates the ratio between turbulent momentum exchange rate and turbulent energy exchange rate. Usually, the prandtl number is obtained by experiments for different fluids. So it is given. Then if we know the equivalent turbulent viscosity, we can easily get the equivalent heat exchange coefficient.

We calculate the turbulent viscosity in this way. First calculate the ration between turbulent stress and physics stress.  And this ratio is actually also the ratio between turbulent viscosity and physical viscosity. Then the turbulent viscosity can be obtained by multiplying physics viscosity with the ratio. Then plug equivalent viscosity into the definition of pradtl number, we get equivalent turbulent heat exchange coefficient.

1. If you can remember, an artificial viscosity term was added in the discretized momentum and energy equation. This term, usually introduce too much dissipation, smearing the discontinuity and over damping mixing. So the artificial viscosity coefficients in out implementation is minimized to relieve such issue. We also tried another scheme of SPH, the Godunov SPH, which combines the idea of Godunov with SPH, and that method introduces artificial viscosity implicitly. However it turned out that GSPH still introduces more dissipation than needs. Here we proposed new scheme, named as RSPH by combining SPH with random choice method
2. This is the discretized formulation using RSPH, which is the same as that of GSPH. I also copied the discretized formulation using classical SPH below. As you can see, there is no explicit artificial viscosity term. And the pressure in the brackets is not outside and replaced by a starred value p\*. The key point is how to calculate the starred value of pressure and velocity.
3. We first create an local coordinate system that …, then project physical quantities in global coordinate system onto this local coordinate system, and then adopting linear constant construction of Riemann problem and define a local Riemann problem, solve this local Riemann problem with approximate HLLC Riemann solver. Randomly sample the approximate solution, we got starred value in the local coordinate system, and then project back to get the starred value in the global coordinate system. These starred value.

I also need to mention the difference between GSPH and RSPH, while GSPH takes solution of local Riemann problem at fixed point as the local starred value, RSPH sample the local Riemann problem randomly to get the solution.

1. First, let’s check this RSPH method with different types of 1D shock tube problem. As you can see, RSPH works well with all these benchmark tests.
2. In this slides, I am gonna compare RSPH with classical SPH and GSPH.

These are zoomed view of the first shock tube test.

In the first pic, RSPH is compared with SPH, which uses different artificial viscosity coefficient.

The second pic shows the zoomed view of around the shock, which confirms that RSPH causes much less smearing than classical SPH.

The third one shows that RSPH can get rid of pressure wiggle around contact discontinuity

Then we compared the RSPH with GSPH, For 1D test, RSPH introduces less smearing and less dissipation. In 3D test, RSPH introduces less equivalent dissipation.

1. Finally, the momentum and energy equations are discretized into the following form using SPH. I have to mention that, the discretization form of using RSPH and GSPH straight forward to get.

This term is the artificial viscosity term …

This term is the turbulent stress term …

You can see similar terms in the discretized energy equation.

The last term in the energy equation is the turbulent heat exchange term.

1. We have three different boundary conditions. How do we impose all those boundary conditions in SPH?

All of these three boundary conditions are imposed using several layers of ghost particles. One advantage of imposing boundary condition in this way is there will always be enough particle within the support of weighting functions no matter whether it is close to the boundary or not.

For example. We set up pressure boundary conditions at the initial step and do not update its physical quantities after that.

As for the eruption boundary condition. All these eruption ghost particles have the same physical quantities as the eruption condition. And they are moving upward with the eruption velocity. As soon as they move out of the eruption vent, they will be shifted to real particle. Simultaneously, new eruption ghost particles will be added at the bottom of the eruption vent.

The way to impose wall boundary condition is as following. We did not imposing no-slip wall boundary condition by mirroring really particle with respect to the wall. In stead, we find a reflection of the wall ghost particles inside the computational domain and then calculate the physical quantities of the reflected point by SPH interpolation. Then assign the physical quantities of the reflected point to the corresponding wall ghost particles except for velocity. As for the velocity, we assign the same value but opposite direction to the wall ghost particles.

Overview

1. The volcano plume raising up process is complicated. Basically it is free boundary flow accompanied by turbulent mixing. During that process, multiple phase involved, other physics, liked heat transfer, aggregation is coupled. It also interact with a stratified atmosphere and finally reach to a much larger scale. Some times it is locally supersonic. And there are great uncertainty in the atmosphere, eruption conditions, and material property. Compared with traditional implementations of SPH, modeling of plume need pressure outlet and velocity inlet boundary conditions.
2. 1) To complete simulation of such complicated phenomena within a time window with acceptable accuracy requires parallel computing.
3. 2)  Computational efficiency is also critical as we might need to do assemble simulation to qualify the propagation of uncertainties.  —> For SPH, neighbor search is a computationally expensive, we have to find some efficient neighbor search algorithm.
4. 3) In addition, to  impose eruption boundary condition, the data structure should allows efficient data deleting and adding.
5. 4)  Flexible data access is always needed for computational software of complicated system.
6. 5) As volcano plume is kind of inject flow which finally reach to a large area. The movement of particles can cause load imbalance. So a dynamic load balancing strategy is needed.
7. There are many implementations of SPH, some of them are parallel, some of them are serial. Due to the complexity of plume simulation, parallel solver is needed so that simulation can be completed in a required time window. So our solver will be parallel.
8. As for targeted hardware, we are targeting at CPUs instead of GPU.
9. The parallel mode for CPUs can be distributed memory, sheared memory or Hybrid. We choose Distributed memory.
10. Most implementations of the parallel SPH method presented to date are limited to standard SPH benchmark problems like dam breaks, or relatively simple scenarios like breaking-waves, flooding. Most of them are free boundary flow and only needs wall boundary condition. Our model is for compressible, multiple phase injection flow. which has  a higher requirement on data accessing flexibility —>That is the main challenge we are going to address in this paper.
11. Secondly, by adding a background mesh, the particle is divided into small groups. The cost of neighbour searching is reduced. For example, to find neighbours of this particle, we only need to search through the bucket that containing it and its neighbour bucket.
12. SFC-based index
13. There are several information I want to delivery through the left picture. First of all: look the SFC that pass through all particles. Using SFC, particles that distributed in a, in this example, 2D space is mapped on a 1 dimensional curve.
14. The third information I want to show is domain decomposition.
15. For the background bucket, We also have SFC, the bold dash line in this picture. This SFC passing through the centroids of all buckets. Again, it maps centroids in a 2D space onto a one dimensional curve. If we cut the 1D SFC into small piece, the background bucket is divided into sub-domain. We can do domain decomposition in this way. For example, the domain is decomposed into two if we cut the SFC from here.
16. Time dependent SFC
17. In the previous slides, the SFC is only based on location. The SFC can guarantee uniqueness of particle index if all particles are added at the same time. But for plume simulation, need particles need to be added during the simulation. For example,  this is a cross-section view of the computational domain. Different colours represent different particles.  The light blue particles are pressure ghost particles that used to impose pressure outlet boundary conditions.
18. The grey particles .......
19. The dark blue particles
20. The red particles are real particles that used to
21. For the eruption boundary conditions, new particles need to be added at the same place during simulation. For this situation, as SFC is only based on location, these particles added at the same location will have the same ID. To guarantee uniqueness of particle index, we use a time-dependent SFC.  One natural way of defining the time-dependent index is simply adding time as a new dimension. If particles are added uniformly, it would be good idea. But for plume modelling, new particles will only be added at a small area -- the bottom. It is OK for SFC, but it will make it harder to design an efficient hash table. So instead of using SFC in a higher dimensional space, we adopt the following way of
22. Hash table
23. As for data management, we use hash table. In our data structure, pointer to particle will be stored in certain slot of the hash table. Every time when need a particle is given, usually by key (or time-dependent SFC based index), we can find in which slot the pointer to that particle is store and the access that particle. The map or the function that used to determine in which slot the particle is stored is hash function. It maps particle key to slot number.
24. A good hash function should fill up most of the slots and avoid hash conflict. For our case, the hash function is designed as following.