Plume-SPH : Volcanic Plume Simulation Software

User Guide

Version: 1.0.0

Geophysical Mass Flow Group (GMFG)

University at Buffalo, NY, USA

Contents

1 Introduction to Plume-SPH

Plume-SPH is a computer program developed by the GMFG (Geophysical Mass Flow Group) at State University of New York at Buffalo, for the purpose of simulating development of volcano plume. It 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.

The governing equations solved by Plume-SPH are a 3D dusty-gas Navier-Stokes type of governing equations. Erupted material is represented by a single phase while air is another phase. Immediate dynamic and thermal dynamic equilibrium between air and erupted material are assumed as soon as they get mixed. SPH (smoothed particle hydrodynamics) method is adopted as the numerical method for discretizing the governing equations.

Either a realistic atmosphere data or an atmosphere model can be used for establishing initial atmosphere. The direct outputs of Plume-SPH are mass fraction of erupted material, internal energy (which can be easily converted into temperature), velocity, density, pressure, particle mass, particle phase ID, particle type ID and coordinates for each discretized points (particles). With these field properties on discretized points, properties at any other point can be obtained by a SPH interpolation.

The solver is parallelized with MPI (message passing interface) http://www-unix.mcs.anl.gov/mpi/]. MPI allows for computing on multiple processors, increased computational power, decreased computing time, and allows for the use of higher resolution. The feature of domain adjusting avoids simulating of uninvolved atmosphere and hence decreases simulation time greatly.

This user guide provides information for installing and running Plume-SPH. For convenience, the Table of Contents contains hyperlinks to all listed sections.

2 Getting Started

System requirements

The Plume-SPH software is open source and is built using only other open source systems. It is designed for both low end single processor use and high end distributed/shared memory multi-processor use. The installation and use procedure is largely similar but there are small differences on each system.

Plume-SPH program File

The latest version of Plume-SPH can be obtained from the Plume-SPH GitHub repository [https://github.com/Plume-SPH/plume-sph.git].

Plume-SPH can be installed from source code. Please see instructions below on how to install the Plume-SPH executable from source code. After successfully installing Plume-SPH, the preprocess executable, "preprocess" and the Plume-SPH executable, "particler", will appear in the its "bin" directory, you will need to access this "bin" directory to run Plume-SPH. You can also copy the "bin" folder to any directory you want and rename it. Please do not forget to modify the "simulation.data" in bin to indicated the correct directory of meteorological data folder "Meteo_data".

Installation from source

Currently, installation of Plume-SPH is only tested on Mac and linux system. Plume-SPH has several dependencies and only a small subset of their versions were tested and some versions are not compatible with Plume-SPH. Therefore, please try to compile Plume-SPH with the version of dependencies that used in the example installation if possible. Trying of different version of dependencies is of course encouraged but no guarantee on success.

Dependencies:

1) HDF5, which is a general purpose library and file format for storing scientific data.
For downloading and more information, please refer to:
Hdf5 1.8.1 works well with Plume-SPH.

2) MPI library, either mpich or Intel MPI works well with our code. For downloading mpich library and more detail, see the link: . Intel MPI might requires license. For more information, see the link:
Other MPI library, such as MVAPICH2 (
) and OpenMPI() is not tested yet.
Plume-SPH works well with intel-mpi 5.0.2 and mpich-3.1.

Setting up environment variables

```
setenv PATH /path/to/mpicxx/:$PATH
setenv HDF5 "/path/to/dir/where_hdf5_is_installed"
```

Download the source code:

```
git clone  https://github.com/Plume-SPH/plume-sph.git
```

Configure:

```
cd plume-sph

make clean
make distclean

aclocal
autoconf
automake --add-missing

./configure --without-gdal --with-hdf5 --enable-debug --enable-parallel CC=mpicc
CXX=mpicxx FC=mpifort
```

Note:

1) It is recommended that turn off debug by remove "--enable-debug" from the configure option.
2) It is recommended to use the mpi version of Plume-SPH, as you will see, the serial version is too slow to do real simulation. So always configure with "--enable-parallel"

Install:

```
Make && make install
```

check installation:

1) if there is some error, there will be some error message print on the screan.
2) cd bin
   ls -l
   There should be two recently generated binary file: "particler" and "preprocess"

Running Plume-SPH

The current version of Plume-SPH is purely command line based. The procedure of running Plume-SPH are as following

1) Have all data ready (meteorological data, material property, eruption parameters, simulation setting parameters), we will cover all details about these data late.
2) Put the meteorological data in the directory named Meteo_data, modify root/src/plume/parameters.h put material property and eruption parameter in the corresponding place. Modify root/src/plume/parameters.h, place simulation parameters at the corresponding place.

3) Modify root/bin/simulate.data.

4) configure if necessary. (If you have already configured, even though you made some modification on root/src/plume/parameters.h, it is not necessary to re-do the configure.). Then install (make && make install)

5) You can copy “bin” and “Meteo_data” to your running directory. You are free to change the name of “bin” to anything. but DO NOT FORGET to modify the line “the directory and name of meteo data” in “simulation.data”. Or you can run Plume-SPH in the compile directory.

6) Run preprocess to generate buckets

   cd bin

   ./preprocess #_of_cores smoothing_length 0

   Please note there are three command line options for running “preprocess”, the first one is the number of cores, the second one is smoothing length, the last one MUST be zero for current version, which indicates that “do not read wind field data while running simulation”. We preserve the last option for future development of Plume-SPH.

   If successful, you will see some files named as: funkyxxxx.h5 in the running dorectory.

7) Run Plume-SPH

   Run the serial version:

   ./particler

   Run the MPI version

   mpirun –np #_of_cores ./particler

   NOTE: #_of_cores while running particler should be the consistent with that for running preprocess.

   Check running output.

   The outputs are files named as pvplotxxxx.h5part (And maybe also pvplot_showxxx.h5part). Each file is corresponding to each sub-domain.

Visualization

The recommended tool for visualization is paraview, which can be download from,

Before loading data in paraview GUI, you need first add plugin

Tools ->Manage Plugins , select “H5PartReader”, and click “Load Selected”

Then load data into paraview:

File->Open

Navigate to the directory where output file of Plume-SPH stores (named as pvplotxxxx.h5part or pvplot_showxxxx.h5part), select them and click open.

Set x coordinate to be “x”, y coordinate to be “y”, z coordinate as “z”

Then you can play with it.

It is recommended to write some python scripts to open batch of pvplotxxxx.h5part (or pvplot_showxxxx.h5part) files. Because it is really boring and time consuming to set coordinates for each input file (corresponding to each domain), especially when you are running you job with hundreds of processors. You can use “trace” under “Tools” to facilitate writing of your scripts. When your scripts is ready, you can run the python scripts through Tools->Python Shell.

Two test run example

1) Pinatubo Eruption

   This test run is based on climactic phase of the Pinatubo eruption (Philippines, 15 June 1991). Material properties are selected based on properties of Pinatubo and Shinmoe-dake.
   All input data for this test run is ready in source code. Smoothing length for this running should be 200m. If you want to use a different smoothing length, you might need to modify Lx_p, Ly_p and num_erupt_particles in root/src/plume/parameters.h and smoothing length of air particles in root/bin/simulation.data.
   After configure and installation
   What you need is just following “Running Plume-SPH” section starts from step 6).

2) A JPUE simulation

   This is A three dimensional axisymmetric JPUE (jet or plume ejected into a uniform environment) which ejects from a round vent.
   All input data for this test run is included in source code. You need to do the following to shift from default Pinatubo running to JPUE.
   a)Modify root/src/plume/options.h:
      #define FLUID_COMPRESSIBILITY 0 -> #define FLUID_COMPRESSIBILITY 1. By doing this, you are actually shifting from EOS (equation of state) for ideal gas to that of weakly compressible liquid
      #define ATMOSPHERE_TYPE 4 -> #define ATMOSPHERE_TYPE 2, by doing this you are actually shifting from a stratified atmosphere to a uniform non-gravity atmosphere.
   b) Modify root/src/plume/parameters.h:

Comments off these lines below the line “These parameters are for repeat the most recent excise” and above “These parameters are for JOUE of incompressible flow”

Comments on these lines between “These parameters are for JOUE of incompressible flow” and “Parameters for running RP1 with sml1=200”

c) Use simulation.data for JPUE.

cd root/bin

cp simulation_JPUE_incomp_back.data simulation.data

The file simulation_JPUE_incomp_back.data are for running JPUE. We replace old simulation.data with this back data file.

Smoothing length for this running is 8m. If you want to use a different smoothing length, you might need to modify Lx_P, Ly_P and num_erupt_particles_P in root/src/plume/parameters.h and smoothing length of air particles in root/bin/simulation.data.

Do installation (make && make install). Then follow “Running Plume-SPH” section starts from step 6).

3 Data files for Plume-SPH

Meteorological data

Meteorological data should be put into format the same as: root/Meteo_data/Meteo_Profiles-W.dat. You should remove header of that file while using it. One example is root/Meteo_data/Meteo_Profiles-W-V.dat, which is the "no header" version of root/Meteo_data/Meteo_Profiles-W.dat. Unit of your data should be consist with the unit listed in the header.

Here is what the header of meteorological data header looks like:

Z(asl) (km)	Density (kg/m^3)	Pressure (hPa)	Temperature (K)	Specific-humidity (g/kg)	Wind-velocity West->East(m/s)	Wind-velocity North->South(m/s)
1.400	1.120	863.203	268.420	0.992	6.567	-11.389
...	...	...	...	...	...	...

Eruption parameters

Eruption parameters are in root/src/plume/parameters.h, Primitive parameters are list as following:

1) double ng0_P: Initial mass fraction of volcanic gas: (mass of volcanic gas)/(total mass of erupted material, No unit
2) double Uv0_P: Velocity in horizontal direction, in ms^{-1}.
3) double Vv0_P: Velocity in verticle direction, in ms^{-1}.
4) double Tv0_P: Temperature of erupted material, in K.
5) double pv0_P: Pressure of erupted material at the vent, for pressure-balanced eruption, usually set to be the same as ambient pressure at the vent height, in Pa.
6) double Mv_P: mass flow rate, it is not directly used in simulation, in kgs^{-1}.
7) double Pos_v_P[3]: Coordinate of vent center, in m;

Other parameters such eruption radius, internal energy of the erupted material ect. can be calculated based on these primitive variables. So when run a new simulation, only these seven primitive eruption parameters should be modified.

Other properties

Other properties, including material properties, gravity, and atmosphere at the vent height, will also be different for different eruptions. However, at most time, you can just use the default material property for new eruptions. Of course, you are free to try different material properties.

All Other properties are in root/src/plume/parameters.h. Primitive properties are list as following:

1) double g_P : Gravity acceleration, in ms-2.
2) double Rg_P: Gas constant for volcanic gases (vapor).
3) double Ta0_P: Temperature of atmosphere at the eruption vent, in K
4) double pa0_P: Pressure of the atmosphere at the eruption vent, in Pa.
5) double rhoa0_P: Density of the atmosphere at the eruption vent, in kgm-3.
6) double Ra_P: Gas constant for volcanic air. Please be noted that propertis 4)-7) are not independent, they should satisfy the EOS.
7) double Cvs_P: Specific heat of solid in erupted material under constant volume, in Jkg-1K-1.
8) double Cvg_P: Specific heat of erupted gas in erupted material under constant volume, in Jkg-1K-1.
9) double Cva_P: Specific heat of air under constant volume, in Jkg-1K-1.
10) double gamma_P: Specific heat ratio.

Simulation Parameters

Most of the simulation parameters are given in root/bin/simulation.data except for : Lx_P, Ly_P, Lz_P and num_erupt_particles_P which will also used by preprocess to setup the domain. We list all simulation parameters here.

1) double Lx_P[2], Ly_P[2], Lz_P[2]: There three parameter defined the maximum domain of simulation. Each parameter has two values, the first one is the lower boundary, the second one is the up boundary. Unit should be m.
2) int num_erupt_particles: Number of eruption particles in the eruption conduit.
3) maximum simulation time(s)
4) maximum simulation time step
5) output time interval(s)
6) start eruption time(s)
7) end eruption time(s)
8) output format indicator, 0 for matlab readable data, 1 for h5part data
9) P_CONSTANT, the constant in equation of state for ideal gas
10) the ratio between specific heat of constant pressure and that of constant volume
11) smoothing length for air particles
12) minimun x coordinate of initial domain
13) maximum x coordinate of initial domain
14) minimun y coordinate of initial domain
15) maximum y coordinate of initial domain
16) minimum z coordinate of initial domain
17) maximum z coordinate of initial domain

18) number of rows in meteo data
19) number of columns in meteo data
20) the directory and name of meteo data

The meaning of these parameters in root/bin/simulation.data is self-explanatory and does not need more explain.

4 Note

Smoothing length for air and erupted material

In the examples, we are using different smoothing length for erupted material and air. This is a trade off between computational cost and numerical accuracy. Smooth length of erupted material should not be too large. Otherwise, there will be no enough particles to represent the structure of plume. But smaller smoothing length for air means larger number of air particles which will increase computational cost. Ratio of smoothing length between air particles and erupted material should not be larger than 2.

Initial domain size

Using a smaller initial domain will avoid simulating of stationary uninvolved air particles and speed up simulation. But there are two constrains: 1) Number of buckets in initial domain should be larger than 4*number_of_cores. 2) The eruption vent should be contained in the initial domain.

Number of processors for running the job

Number of processors need to accomplish the job in an accepted time depends on domain size and smoothing length. Run the job with too less processor will take very long time, and you might even run out memory. Larger number of processors usually requires a larger initial domain, which essentially reduce the effect of domain adjusting feature. Here is a rough criteria for determine how many cores you need: it would be better to give each processors around 200~500 bukets (43200 ~ 108000 particles each processor). Every time you run “preprocess” given smoothing length and number of processors, the total number of buckets and number of buckets per processor will print on the screen.

Checkpoint and restart

Our code does not have module for checkpoint. Users are encouraged to use third party checkpoint tools. One of the checkpoint tools that works well along with Plume-SPH is: dmtcp. Please refer to : for more details.

In root/bin/, there is a slurm scripts for running Plume-SPH together with dmtcp on a cluster. As the slurm scripts greatly depends on system and paltform, that this slurm script can only be used as a reference.