user guide

Last update: April 8, 2021

Holger Grosshans

holger.grosshans@ptb.de

Working group "Analysis and Simulation in Explosion Protection"
Physikalisch-Technische Bundesanstalt (PTB), Bundesallee 100, 38116 Braunschweig, Germany

1 About this guide

This guide describes the usage of pafiX, i.e. the file structure, the content of input and result files and how to run the code. As regards the underlying mathematical model, numerical methods, and scientific contributions we refer to our publications [1–7].

2 About pafiX

pafiX (particle flow simulation in explosion protection) is a computational fluid dynamics (CFD) solver written in Fortran 90 optimized to analyze pneumatic transport of powders. It is developed by the *Analysis & Simulation in Explosion Protection* group at PTB in collaboration with the *Multi-phase & Reacting Flows* group at UCLouvain. While there are a number of commercial and open source software packages on the market with the capability to compute powder flows, the development of pafiX aims to address the specific need for the theoretical evaluation of various kinds of arising explosion hazards. These hazards may be due to deposit formation, triboelelectric charging, external heating, etc. which have initiated fatal events in process and chemical industries in the past (see Eckhoff [8] for an exhaustive discussion).

Another feature differentiating pafiX from comparable software is that we stripped of all parts from the code, input masks and output files which are not directly related to explosion protection. Therefore, pafiX is on the one hand a state-of-the-art tool for innovative research and on the other a easy to learn CFD code for students. The current version of pafiX is always available on gitlab (`https://gitlab1.ptb.de/Holger/pafix`). This is Since April 2021 a public repository, thus, **pafiX is open-source, and you are welcome to distribute it further!**

The latest tag of the *master* branch is fully validated and usually documented in a publication. The *beta* branch is the current development version, you are welcome to test it and inform the maintainer about bugs. Please generate your own branch for specific and complex implementations. Proposals for simplification and improvement of the beta version via merge requests are encouraged.

In case of bugs, questions or requests for new features or other architectures please contact H. Grosshans (holger.grosshans@ptb.de).

3 Release history

The pafiX release history is given in Table 3. If you use pafiX to generate images and/or movies please cite the given reference. Doing so helps to sustain funding for future improvements and on going maintenance.

Version	pafiX.001.00100
Release date	March 1, 2019
Architecture	Ubuntu 18.04 64 bit
Ref.	[9]
Brief case description	Powder flow through a squared duct, particles are initially assigned an electrostatic charge which is then constant in time. A fully-developed turbulent flow field and a converged field of uncharged particles is provided which can be used as starting condition. For example, charge can be be assigned to the particles and its influence on the particle trajectories observed. It is noted that the provided initial flow field is essentially under-resolved and is merely provided to demonstrate the capability of pafiX and less to provide accurate results.
Version	v1.0.0
Release date	February 29, 2020
Ref.	[10]
Brief description	First release of the source code. Migration to gitlab.
Version	v1.1.0
Release date	April 8, 2021
Ref.	tba
Brief description	Major update of all parts of the code. First open source release.

4 How to run pafiX

- open a terminal
- clone the code

```
git clone https://gitlab1.ptb.de/Holger/pafix
```

- checkout the development version

```
cd pafix
git checkout beta
```

- compile the code

```
cd src
make
cd ..
```

- create an input file

```
cp input/input_example.dat input/input.dat
```

- Execute pafiX

```
. run.sh
```

- Monitor the evolution of some parameters of the simulation

```
gnuplot monitor.plt
```

In case of error messages you probably miss some Linux packages.

5 File structure

5.1 Case structure

Each pafiX case is structured as follows:

```
pafix .2 documentation
 |  |__doxygen_codeGuide
 |  |  |__codeGuide
 |  |__userGuide
 |     |__pafiX_userGuide
 |__input
 |  |__input.dat
 |__restart
 |  |__fluidField_pmmm_ii_jj_ll_nnnnnn
 |  |__particleField_ii_jj_ll_pmmm_nnnnnn
 |  |__...
 |__results
 |  |__fluid_xz_pmmm_nnnnnn.vtk
 |  |__particles_pmmm_nnnnnn.vtk
 |  |__fluid_xz.visit
 |  |__particles.visit
 |  |__fluid_u_xz_pmmm_nnnnnn.dat
 |  |__...
 |__output
 |  |__grid_pmmm.vtk
 |  |__monitor.dat
 |  |__output.dat
 |__src
 |  |__pafiX
 |  |__makefile
 |  |__....f90
 |__clean.sh
 |__duct.sh
 |__monitor.plt
```

5.2 File contents

In all file names *mmm* denotes the number of the processor that wrote the file, *nnnnnn* the respective time-step and *ii*, *jj*, and *ll* the number of grid cells per processor in x, y, and z direction, respectively.

documentation/doxygen_codeGuide Generate a code guide using doxygen

```
doxygen codeGuide
```

documentation/userGuide/pafiX_userGuide.pdf This user guide.

input.dat This file is the only mask for the user to control all conditions of simulation.

restart/... This folder contains binary files for the fluid and particle phase which are written out during the computation and can be used later on to restart the simulation at time-step *nnnnnn*.

results/... The folder *results* contains all result files of the simulation. The **.vtk* files are for visualization. While most post-processing tools read vtk format we have tested with Visit [11] which is open source. Since it is elaborative to load a large amount of data files stemming from multiple processors and time-steps into Visit, additionally **.visit* files are provided. These files are containers, i.e. it is sufficient to open the container file to visualize all data at once. The *.dat* files essentially contain the same information than the **.vtk* but are optimized for post-processing instead of visualization.

output/... Folder containing files to monitor the status of the simulation. *grid_pmmm.vtk* can be read by Visit to visualize the grid, *monitor.dat* can be read by the script *monitor.plt* to plot the current status of key parameters of a running simulation, and in *output.dat* the simulation conditions are documented.

src The complete source code.

clean.sh This script cleans the case folder from all simulation results and output data.

duct.sh Script to start the simulation and to control the number of used processors.

monitor.plt A gnuplot [12] script to plot the current status of key parameters of a running simulation.

References

[1] Grosshans, H. and Papalexandris, M. V. (2016). *Large eddy simulation of triboelectric charging in pneumatic powder transport.* Powder Technol., 301:1008–1015.

[2] Grosshans, H. and Papalexandris, M. V. (2016). *Evaluation of the parameters influencing electrostatic charging of powder in a pipe flow.* J. Loss Prev. Process Ind., 43:83–91.

[3] Grosshans, H. and Papalexandris, M. V. (2016). *A model for the non-uniform contact charging of particles.* Powder Technol., 305:518–527.

[4] Grosshans, H. and Papalexandris, M. V. (2017). *Direct numerical simulation of triboelectric charging in a particle-laden turbulent channel flow.* J. Fluid Mech., 818:465–491.

[5] Grosshans, H. and Papalexandris, M. V. (2017). *On the accuracy of the numerical computation of the electrostatic forces between charged particles.* Powder Technol., 322:185–194.

[6] Grosshans, H. and Papalexandris, M. V. (2018). *Exploring the mechanism of inter-particle charge diffusion.* Eur. Phys. J. Appl. Phys., 82(1):11101.

[7] Grosshans, H. (2018). *Modulation of particle dynamics in dilute duct flows by electrostatic charges.* Phys. Fluids, 30(8):083303.

[8] Eckhoff, R. K. (2003). *Dust Explosions in the Process Industries.* Gulf Professional Publishing, 3rd edition.

[9] Grosshans, H. (2019). *Modeling the agglomeration of electrostatically charged particles.* In *BAM-PTB Kolloquium 2019.* Braunschweig, Germany.

[10] Grosshans, H., Bissinger, C., Calero, M., and Papalexandris, M. V. (2021). *Influence of electrostatic charges on particle-laden duct flows.* J. Fluid Mech., 909:A21.

[11] Childs, H., Brugger, E., and et al. (2012). *VisIt: An end-user tool for visualizing and analyzing very large data.* In *High Performance Visualization–Enabling Extreme-Scale Scientific Insight*, pages 357–372.

[12] Williams, T., Kelley, C., and many others (2017). *Gnuplot 5.2: an interactive plotting program.* `http://www.gnuplot.info`.