

Pharmaceutical
Multiphase Reactors
CHE.782

Design of Multiphase
Flow Processes
669.266

Usage of the „ParScale“ Library

- A - Content Parcale
- B - Installation and Coupling
- C - Running Simulations

Ass.Prof. Dr. Stefan Radl,
M. Sc. Thomas Forgber
Email: radl@tugraz.at
Institute of Process and
Particle Engineering
Inffeldgasse 13/III
TU Graz

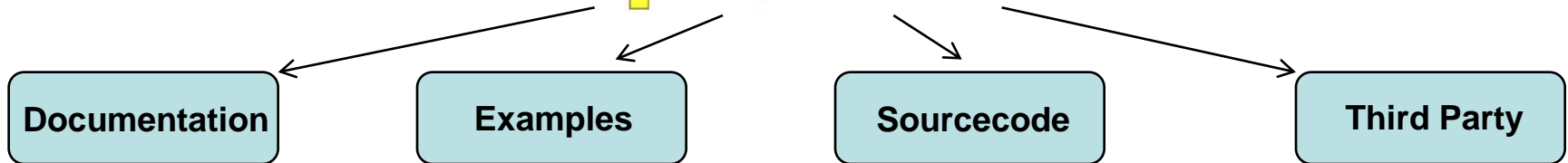
A part of this teaching material has been prepared
for NanoSim (<http://sintef.no/NanoSim/>)



NanoSim - A Multi-scale Simulation-Based Design Platform



What is part of ParScale?



- **doc/** - folder
- *.md files for all implemented features
- doc/pdf/ - folder holds theoretical documentation of equations implemented
- **examples/** - folder
- octaveFuntions/ holds functions used for caluclation and **post-processing**
- verificationCases/ holds verification studies including analytical solution
- testCases/ holds non- verified test cases
- **src/** - folder
- All source classes
- Routines for building ParScale (MAKE/)
- **INSTALL.md** holds all information for compiling ParScale
- Class structure can be seen using the doxygen_config file
- **thirdParty/** holds third party libraries ParScale depend on
- chemkinReader/ for interface to chemkin-II files
- sundials-2.5.0/ for integrator class

¹ Download from: <https://github.com/CFDEMproject/PARSCALE-PUBLIC.git>

Before you start...

...go through the README.md!

<https://github.com/CFDEMproject/ParScale-PUBLIC/blob/master/README.md>

README.md



A Compilation of Particle Scale Models.

ParScale is part of the [NanoSim Project](#)

Copyright Notice

- Copyright 2014 - Graz University of Technology (S. Radl, T. Forgber).
- Copyright 2014 - DCS Computing GmbH, Linz (C. Goniva, C. Kloss).

All rights reserved.

Note that some modules within the ParScale package are not under the copyright of the above copyright holders. These modules are included in the ParScale package with a valid open-source license. Please see the text in the source files, or the corresponding sub-directory, for more detailed information on copyright and licensing issues.

Basic Requirements for:

(1) Stand – alone

- Linux environment **or** Linux access from Windows computer (via Xming, putty, filezilla)
- gedit (file editing), git, Boost library, HDF 5.x, Qt 4.3x, Octave with JsonLab, (Markddown viewer)
- Access to Public ParScale git repository

(2) Coupling LIGGGHTS

- LIGGGHTS requirement/ Compliant
- Built as static library
- ParScale compiled as library

(3) Coupling LIGGGHTS, CFDEM

- CFDEM requirements
- OpenFOAM installed
- CFDEM compliant

(1) Download, compile and test Stand-alone

- Get your copy of ParScale using e.g.
„git clone <https://github.com/CFDEMproject/PARSCALE-PUBLIC.git>“
- Carefully read through **src/INSTALL.md**
- Make sure you have all additional libraries installed
- Make sure all environment variables are set correctly in you **.bashrc**
- Go to *src/* and run „**./refresh**“ to compile the stand alone version
- Take action in case of any errors (e.g. use the forum on **www.cfdem.com**)
- Check by running all verification and test cases:
 - Go to *examples/verificationCases/* and run „**./Allrun_convective**“ every sub folder
 - Go to *examples/testCases/* and run „**./Allrun_convective**“ in every sub folder
- Only continue if no error occurs

(2) Download, compile and test LIGGGHTS- coupling

- Go to *ParScale/src* and run „***./refreshLibrary***“ to compile ParScale as a library (see ***src/INSTALL.md*** for detailed information)
- Go to *LIGGGHTS/src* and run „***make yes-PASCAL***“. This will install the fix that couples LIGGGHTS with ParScale. You can now continue to compile LIGGGHTS (see the LIGGGHTS manual)
- Take action in case of any errors (e.g. use the forum on www.cfdem.com)
- Only continue if no error occurs
- **Test your compilation** by running the test case in „LIGGGHTS-PUBLIC/examples/LIGGGHTS/Tutorials_public/ParScale/heatTransferBed_p aScal“
- In case you make changes to the ParScale fixes in *LIGGGHTS/src*, please be sure you **copy back the files** to the *LIGGGHTS/src/PASCAL* folder

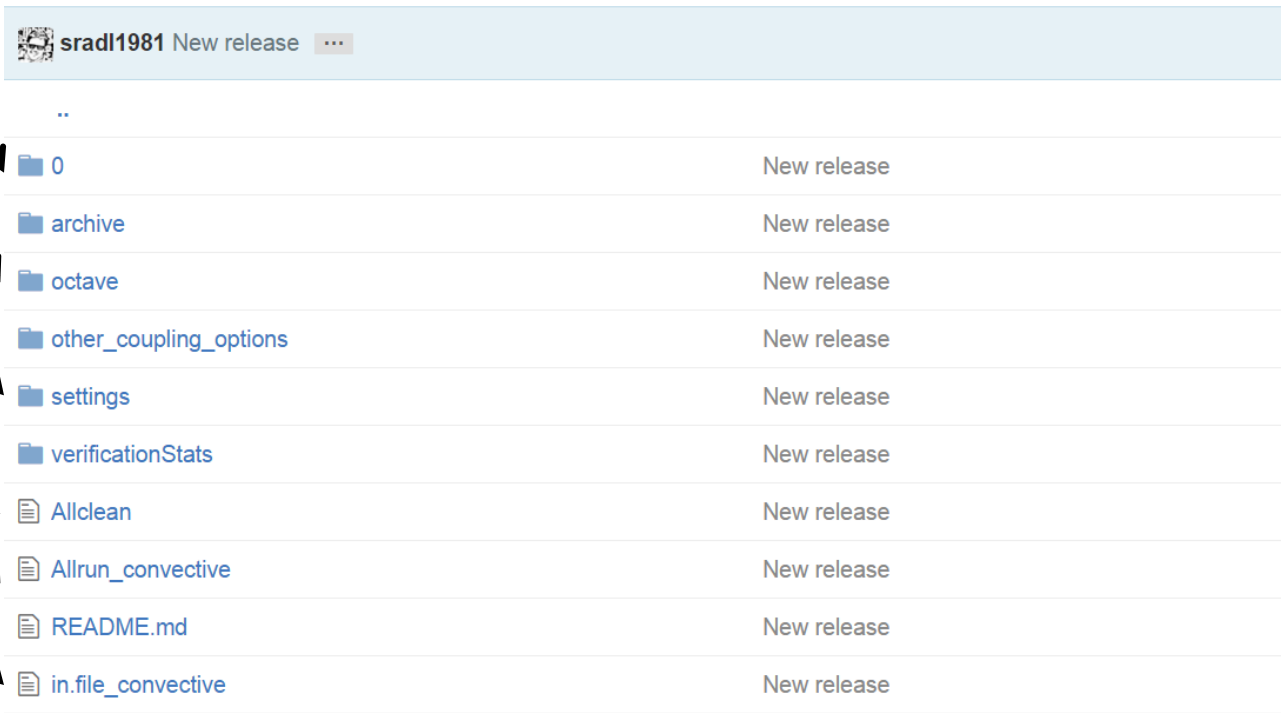
(3) Download, compile and test ParScale – CFDEM – LIGGGHTS coupling

- Make sure you have compiled the latest version of LIGGGHTS with coupling to ParScale
- Add **additional libraries** in „src/lagrangian/cfdemParticle/etc/additionalLibs“
- For example file see „./additionalLibs.C3POParScale“
- **Compile CFDEM** by typing „cfdemCompCFDEMail“
- **Check by running** test case in „tutorials/cfdemSolverPimpleImEx/fluidizedBedTempParScale/“

- All input/output is in the form of **text files**
- The **main input script** contains one command per line. It is similar to a Matlab input script, however, each command is NOT executed immediately.
- During a simulation, users **CANNOT directly interact** with a ParScale simulation.

initial conditions
holds auto-postprocessing routines
Most commands read a „settings“ file in JSON format
scripts to auto-start and clean
main input script

ParScale-PUBLIC / examples / verificationCases / heatConductionTransientBC /



File/Folder	Release Status
..	
0	New release
archive	New release
octave	New release
other_coupling_options	New release
settings	New release
verificationStats	New release
Allclean	New release
Allrun_convective	New release
README.md	New release
in.file_convective	New release

Pre-Process

- Problem definition (physical background, possible with ParScale/Coupling with LIGGGHTS?)
- Choosing models, identify relevant phenomena, parameters
- Coupling necessary?

Input

- Define boundary conditions
- Identify time scales, relevant output
- Mesh resolution
- Initial conditions

Run

- make sure convergence is reached
- checking data before running big simulation

Post-Process

- Post-process relevant data
- use routines/interfaces provided in verification/test cases

Impressum & Disclaimer

©2015 by Stefan Radl, and other members of the „Simulation Science“ Group at the Institute of Process and Particle Engineering, Graz University of Technology.

All rights reserved. No part of the material protected by this copyright notice may be reproduced or utilized in any form or by any means, electronically or mechanically, including photocopying, recording or by any information storage and retrieval system without written permission from the author.

LIGGGHTS® is a registered trade mark of DCS Computing GmbH, the producer of the LIGGGHTS® software. CFDEM® is a registered trade mark of DCS Computing GmbH, the producer of the CFDEM®coupling software. This offering is not approved or endorsed by DCS Computing GmbH, the producer of the LIGGGHTS® and CFDEM® coupling software and owner of the LIGGGHTS and CFDEM® trade marks.