

The OPM Initiative

Kristin M. Flornes



opm-project.org



OPM

The Open Porous Media Initiative

Information
News
About
Gallery
Mailing List
Publications
Modules
opm-core
opm-polymer
dune-cornerpoint
opm-porsol
opm-upscaling
eWoms
ResInsight
Experimental
Development
Download
Tutorials
Documentation
Contributing
Wiki
License
Related
MRST
DuMuX

OPM in a nutshell

The Open Porous Media (OPM) initiative provides a set of open-source tools centered around the simulation of flow and transport of fluids in porous media. The goal of the initiative is to establish a sustainable environment for the development of an efficient and well-maintained software suite based on the following principles:

- All OPM modules are free software available under the terms of the GNU General Public License (GPL) version 3.
- OPM strives to use an open development model. This means that anyone can contribute on equal grounds:
 - All source-code is hosted on public repositories on [github](#).
 - All developer infrastructure (e.g. mailing lists, bug tracking system, wiki) is open to the general public.
- OPM aims to be useful for applications in many industrially relevant fields including CO₂ storage, environmental engineering, and reservoir engineering.
- OPM also tries to be easy to extend with new functionality; for this reason most OPM modules use the DUNE C++ template library to achieve good computational performance while keeping the code as easy to understand and to maintain as possible.

Currently, OPM development is focused on oil reservoir engineering, enhanced oil recovery and CO₂ sequestration, although contributions aimed at different fields are always welcome.

Latest News

Program of OPM Symposium

May 15, 2013

The program for this year's OPM symposium in Bergen/Norway has been published. It is available [here](#).

OPM Symposium May 2013

April 29, 2013

This year, a one and a half day symposium focused on the Open Porous Media Initiative will be organized in Bergen, Norway from May 28 to May 29. It is open to everyone directly or indirectly interested in the OPM initiative. The idea of the symposium is to be forum to exchange ideas. For this reason, its setting will be informal and it will be centered around short presentations and discussions.

Call for Presentations

We strongly encourage everyone with an interest in OPM to contribute a presentation. Please [notify us](#) as soon as possible if you would like to participate.

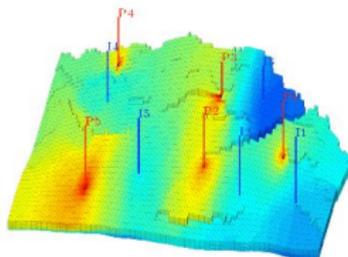
Deadline

If you intend to participate, please send the title and a short abstract of your presentation to [Kristin Flornes](#) or [Atgeirr Rasmussen](#) no later than May 10, 2013.

Date and Venue

The event will be taking place on 28-29 May 2013 at ViVite, Thormøhlensgate 51, next to the IRIS office in Bergen, Norway.

Schedule



Pressure distribution of a synthetic test case. Cornerpoint grid with faults drawn from the Saigup project. This case is explored in a [tutorial](#) for the opm polymer simulator.

OPM – Open-source simulators of Porous Media flow



THE VISION FROM 2009

A long-lasting, efficient, and well-maintained, open-source software for flow and transport in porous media.

The resulting software should:

- › be built on modern software principles,
- › have functionality supporting multiple application areas,
- › be easy to extend with new functionality,
- › be built on open-source code principles,
- › have a relatively low user threshold.

The software should be used/maintained based on a collaborative effort and involve groups with different research focus

Where are we today?



In a Nutshell, Open Porous Media...

- ... has had 10,172 commits made by 61 contributors representing 525,528 lines of code
- ... is mostly written in C++ with a well-commented source code
- ... has a well established, mature codebase maintained by a very large development team with increasing Y-O-Y commits
- ... took an estimated 140 years of effort (COCOMO model) starting with its first commit in February, 2008 ending with its most recent commit 5 days ago

Open Porous Media

Analyzed 4 days ago

The principal objective of the Open Porous Media (OPM) initiative is to develop a simulation suite that is capable of modeling industrially and scientifically relevant flow and transport processes in porous media and bridge the gap between the different application areas of porous media modeling, including reservoir mechanics, CO₂ ... [More]

Mostly written in C++

Licenses: GPL-3.0+

visualization cfd simulation porousmedia

525K lines of code

43 current contributors

5 days since last commit

5 users on Ohloh

Compare

Very High Activity

0 Reviews

Application areas: Upscaling, EOR simulation, CO₂ storage, black oil simulation, visualization,...

OPM software on GitHub



The software under the OPM umbrella on GitHub now contains parts originating from:

- › Uni Stuttgart (eWoms)
- › Sintef, IRIS (dune-cornerpoint, opm-core, opm-porsol, opm-upscaling, opm-benchmarks, opm-polymer, opm-autodiff)
- › Statoil (opm-benchmarks, opm-parser, opm-upscaling)
- › CIPR (opm-verteq)
- › Ceetron (ResInsight)

Many pieces, nontrivial dependencies.

GitHub

Explore

Features

Enterprise

Blog

Sign up

Sign in


OPM
OPM

<http://www.opm-project.org>

Joined on Jun 07, 2012

11

public repos

19

members

Repositories

Members

All Sources Forks Mirrors

**ResInsight**

3D viewer and post processing of reservoir models

Last updated 3 hours ago

C ★ 7 10

**opm-autodiff**

Utilities for automatic differentiation and simulators based on AD.

Last updated 3 days ago

C++ ★ 0 5

**opm-benchmarks**

DUNE module containing benchmarks for the OPM project

Last updated 4 days ago

C++ ★ 1 4

**opm-core**

Open Porous Media Core Library

Last updated 4 days ago

C++ ★ 8 17

**ewoms**

The eWoms Porous Media Simulation Framework

Last updated 5 days ago

C++ ★ 2 4

**opm-upscaling**

DUNE module containing single-phase and steady-state upscaling methods

Last updated 12 days ago

C++ ★ 4 10

**opm-parser**

Last updated 21 days ago

C++ ★ 0 3

Status of OPM Initiative

Atgeirr Flø Rasmussen

SINTEF ICT, Applied Mathematics

May 28 2013

Overview

opm-core

opm-polymer

The Dune modules

opm-autodiff

Overview

opm-core

opm-polymer

The Dune modules

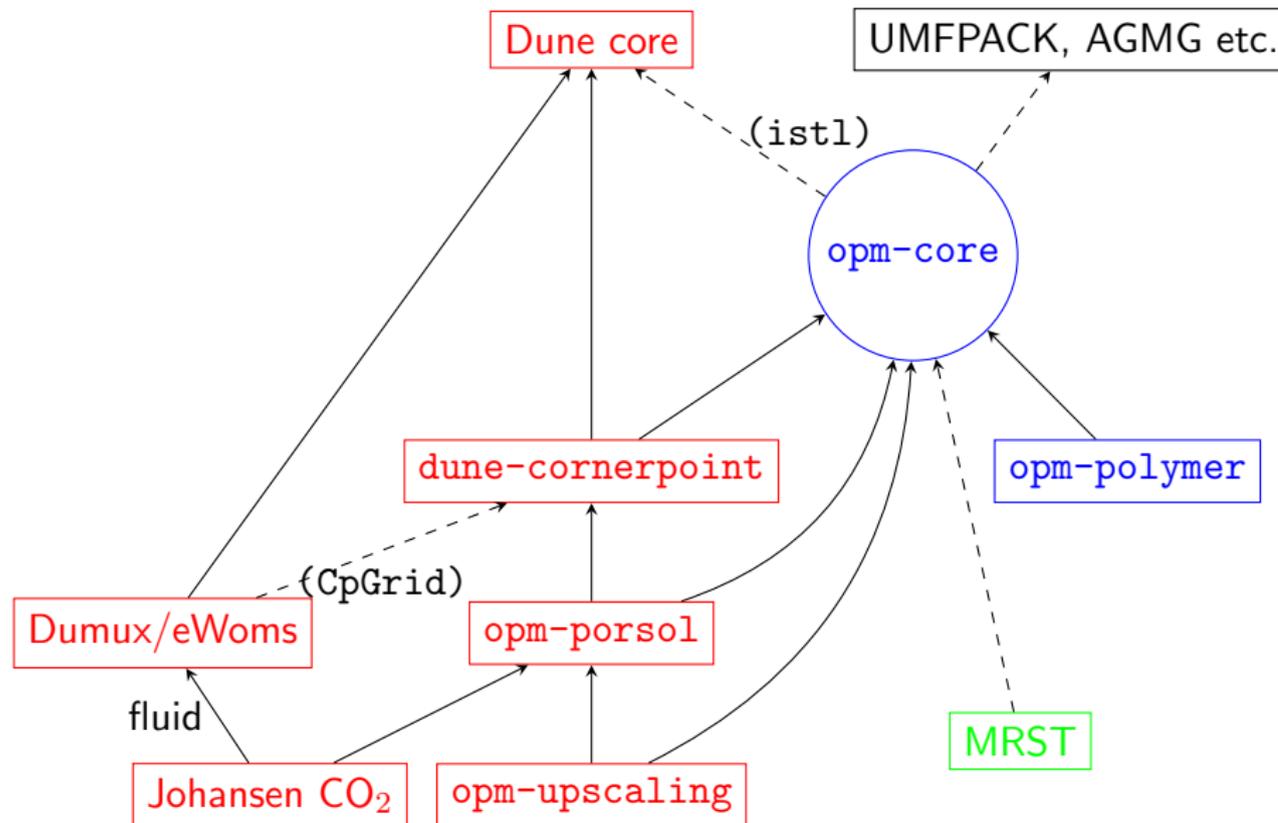
opm-autodiff

The software under the OPM umbrella on GitHub now contains parts originating from:

- ▶ Uni Stuttgart (eWoms)
- ▶ Sintef, IRIS (dune-cornerpoint, opm-core, opm-porsol, opm-upscaling, opm-benchmarks, opm-polymer, opm-autodiff)
- ▶ Statoil (opm-benchmarks, opm-parser)
- ▶ CIPR (opm-verteq)
- ▶ Ceetron (ResInsight)

Many pieces, nontrivial dependencies.

Dependencies



Overview

opm-core

opm-polymer

The Dune modules

opm-autodiff

Utilities

- ▶ parameter input
- ▶ Eclipse deck input
- ▶ output: vtk, matlab, Eclipse
- ▶ scalar solver, interpolation
- ▶ generalized barycentric coordinates
- ▶ velocity interpolation
- ▶ timers, units, etc.

Grid handling

- ▶ struct `UnstructuredGrid`
- ▶ corner-point grid processing

Pressure solvers

- ▶ incompressible TPFA
- ▶ incompressible mimetic
- ▶ compressible TPFA
- ▶ compressible TPFA, residual formulation

Transport solvers

- ▶ incompressible explicit upwind Euler
- ▶ incompressible implicit upwind Euler
- ▶ incompressible implicit upwind Euler with reordering
- ▶ compressible implicit upwind Euler with reordering
- ▶ black-oil explicit upwind Euler (as part of pressure solver)
- ▶ column gravity segregation solver

Well handling

- ▶ initialize from deck
- ▶ handles most well control types
- ▶ group controls
- ▶ well constraints
- ▶ switching controls

Linear algebra interfaces

- ▶ UMFPACK
- ▶ dune-istl
- ▶ AGMG

Properties

- ▶ incompressible fluid
- ▶ black-oil fluid
- ▶ initialize from parameters or deck
- ▶ choice of 3-phase model

Other

- ▶ FV time-of-flight **and tracer partitioning** calculation
- ▶ DG(0/1) time-of-flight **and tracer partitioning** calculation
- ▶ **A CMake-based build system**
- ▶ **Release 2013.03**
- ▶ **Improved documentation**

External dependencies of opm-core:

- ▶ BLAS, LAPACK
- ▶ Boost
- ▶ Linear solvers (optional, but at least one)
 - ▶ UMFPACK
 - ▶ dune-istl
 - ▶ AGMG
- ▶ ERT (optional)

We want opm-core to have only easy-to-satisfy dependencies.

Overview

opm-core

opm-polymer

The Dune modules

opm-autodiff

A simulator for the two-phase polymer flow problem.

Features:

- ▶ Polymer model:
 - ▶ Dead pore space
 - ▶ Viscosity modification
 - ▶ Adsorption
 - ▶ Mixing
 - ▶ Residual reduction
- ▶ Gravity
- ▶ Wells
- ▶ Rock compressibility
- ▶ Weak fluid compressibility (no gas)

(More depth in its own presentation)

Overview

opm-core

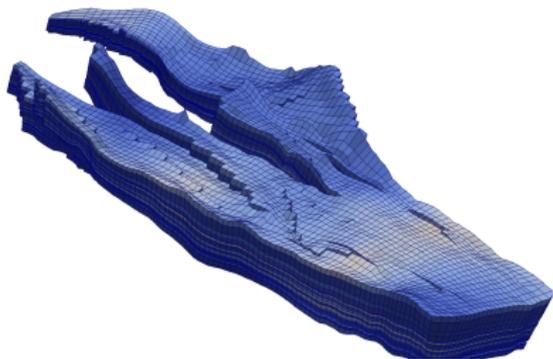
opm-polymer

The Dune modules

opm-autodiff

The dune-cornerpoint module

Implements corner-point grids with the Dune grid interface.



- ▶ Depends on opm-core.
- ▶ Generic functionality moved to opm-core.
- ▶ Some Dune-using groups (Münster, Stuttgart) using/testing.
- ▶ Process underway with aim to make it a dune core module.
 - ▶ May include a parallelization effort.
 - ▶ (Unknown status)

Implements porous media solvers using the Dune grid interface and linear solvers.

- ▶ Now depends on opm-core.
- ▶ Generic functionality moved to opm-core.
- ▶ Now implements CO₂ storage simulator in addition to black-oil simulator.
- ▶ Black-oil simulator gives results similar to commercial product (but is still slower).

The opm-upscaling module

Implements upscaling features using the solvers provided by opm-core and dune-porsol.

- ▶ Implicit transport for steady-state upscaling.
- ▶ Steady stream of minor improvements to tools from Statoil.
- ▶ Testing and verification ongoing.

Recently released version 2.2.

- ▶ A fork of DuMuX.
- ▶ Fully implicit methods.
- ▶ Compositional, black-oil fluids.

(Ask Andreas...)

Almost no overlap in capabilities with opm-porsol, opm-core...

Overview

opm-core

opm-polymer

The Dune modules

opm-autodiff

A testing-ground for using automatic differentiation (AD) for simulator construction.

To be considered **experimental** for now.

Features:

- ▶ AD C++ classes for computation with sparse structures
- ▶ Interface based on Eigen library (vector and matrix arithmetics)
- ▶ Utilities for writing solvers in a vectorized style
- ▶ Prototype solvers for:
 - ▶ Operator-splitting pressure and transport.
 - ▶ Fully implicit black-oil.