

softwareunderground.org presents



TRANSFORM 2021

Virtual Conference on the Digital Subsurface, 16–23 April

supported by



Welcome! The livestream will begin soon...

Reservoir Simulation and Modelling with MRST

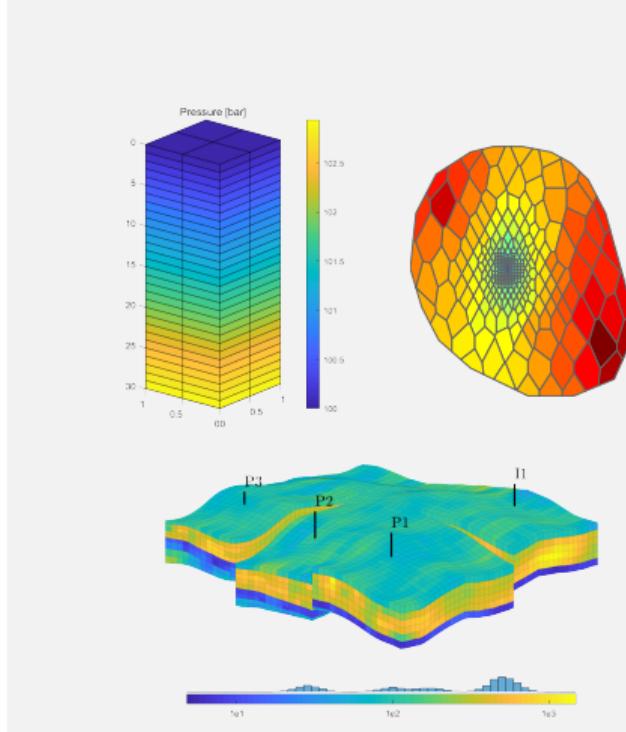
Øystein S. Klemetsdal

Computational Geosciences, SINTEF Digital, Oslo, Norway

TRANSFORM 2021, Software underground, April 20, 2021

Slack channel: t21-tue-mrst

- Short overview of MRST
 - what is the purpose of this software?
 - how is the software organized?
 - where can I find help?
- Getting started
 - download or clone MRST
 - using and navigating the modules
 - solving an incompressible flow problem
- Numerical framework
 - discrete operators and automatic differentiation
- More complex example
 - creating and simulating a sector model



Short overview of MRST

MATLAB Reservoir Simulation Toolbox (MRST)

Transforming research on reservoir modelling

Unique prototyping platform:

- Standard data formats
- Data structures/library routines
- Fully unstructured grids
- Rapid prototyping:
 - discrete operators
 - automatic differentiation
 - object-oriented framework
- Industry-standard simulation

MATLAB Reservoir Simulation Toolbox (MRST)

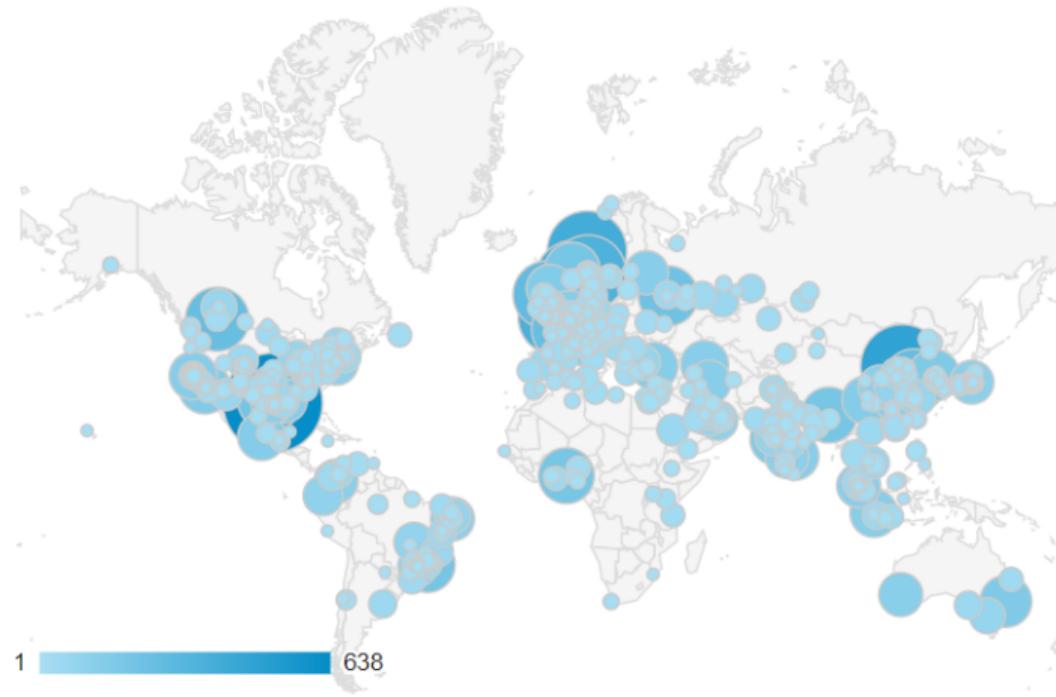
Transforming research on
reservoir modelling

Large international user base:

- downloads from the whole world
- 125 master theses
- 62 PhD theses
- 270 journal papers (not by us)
- 150 proceedings papers

Numbers are from Google Scholar notifications

Used both by academia and industry

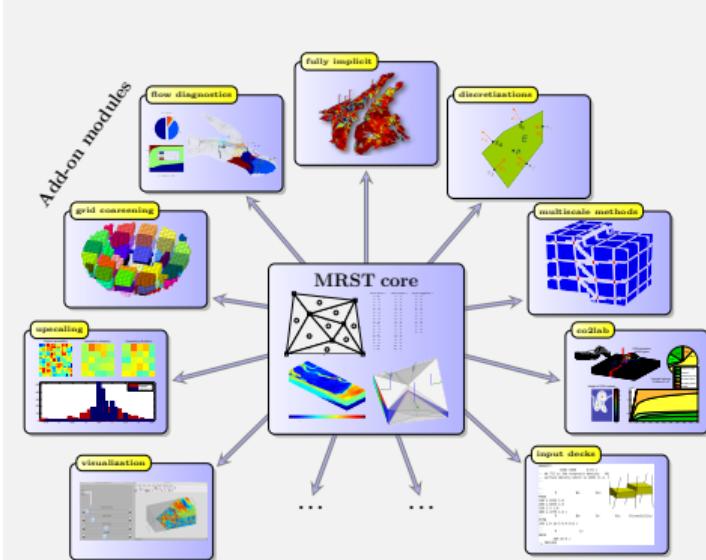


Google Analytics: access pattern for www.mrst.no
Period: 1 July 2018 to 31 December 2019

Modular design:

- **small core** with mature and well-tested functionality used in *many* programs or modules
- **semi-independent modules** extend core functionality
- in-source documentation like in MATLAB
- all modules must have code examples and/or tutorials

This simplifies how we distinguish public and in-house or client-specific functionality



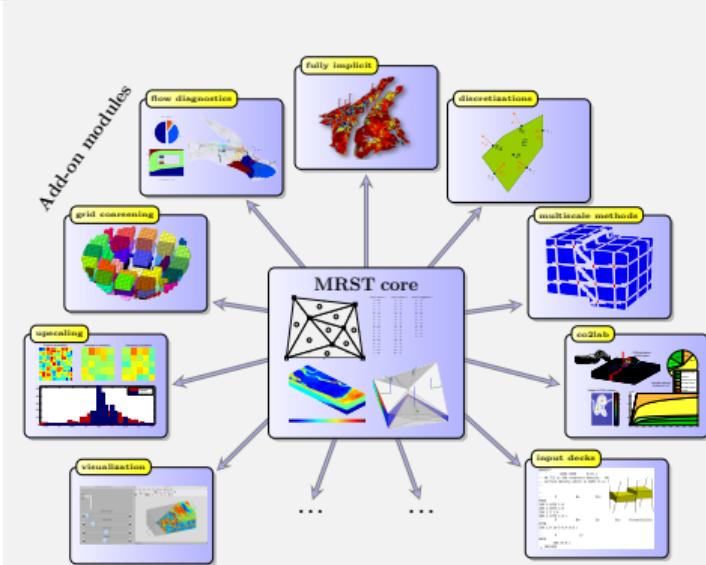
Modular design:

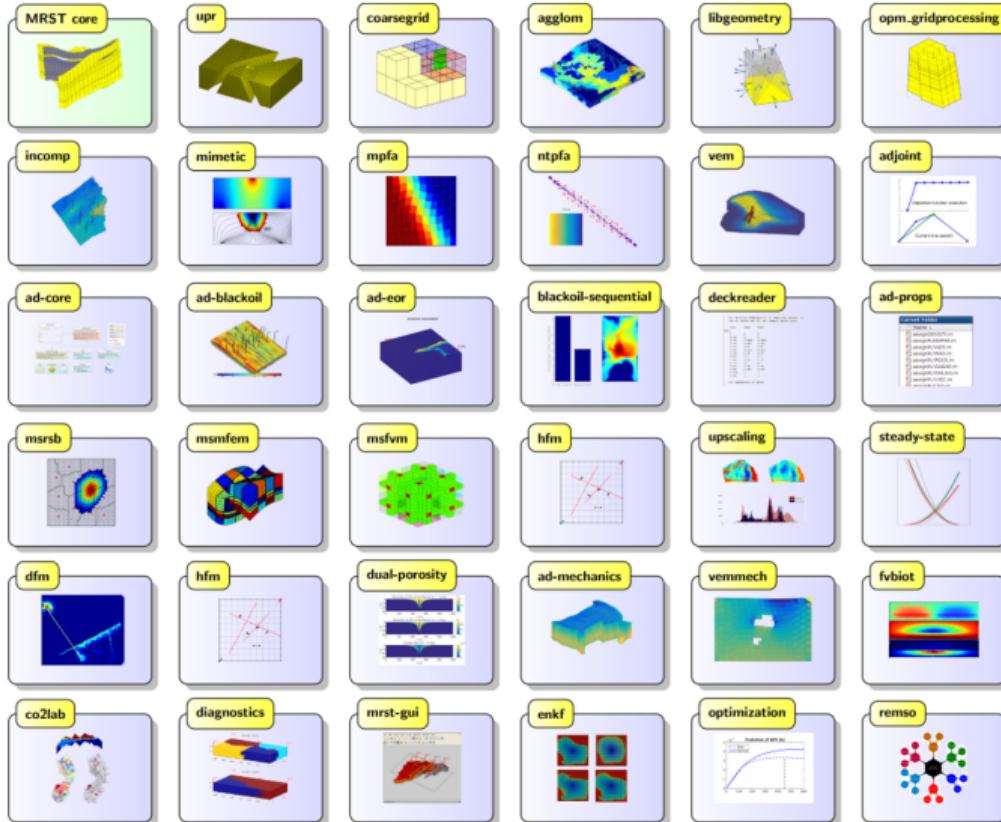
- **small core** with mature and well-tested functionality used in *many* programs or modules
- **semi-independent modules** extend core functionality
- in-source documentation like in MATLAB
- all modules must have code examples and/or tutorials

This simplifies how we distinguish public and in-house or client-specific functionality

Core module:

grid structure, grid factory routines, petrophysical data, basic fluid models, automatic differentiation library, setting boundary/wells/sources, reservoir state, visualization, etc





- Grid generation and coarsening
- ECLIPSE input and output
- Upscaling / multiscale solvers
- Consistent discretizations
- Black-oil, EOR, compositional
- Fractures: DFM, EDFM, DPDP
- Geomechanics, geochemistry, geothermal
- Unsaturated media (Richards eq.)
- Multisegment wells (general network)
- CO2 storage laboratory
- Adjoints, optimization, ensembles
- Pre/postprocessing/visualization
- Flow diagnostics
- ...

User resources (getting help)

The screenshot shows the homepage of the MRST - MATLAB Reservoir Simulation Toolbox. The top navigation bar includes links for MRST, FAQ, Forum, Modules, Gallery, Download, Documentation, Publications, Contact, and Downloadable Resources. Below the navigation is a search bar. The main content area is divided into several sections: 'Basic functionality' (with sub-sections like 'Create data structures', 'Visualization', and 'Paths and data sets'), 'Discretizations and solvers' (with sub-sections like 'Stability', 'Matrix', 'Geometry', 'Grid', 'Dense', 'Matrix', 'Adaptive methods', and 'MFDIPE'), and 'Workflow tools' (with sub-sections like 'Visual modeling', 'Workflow', 'Visualizations', 'Scripted simulation', 'Refine input', 'MRST-embed', and 'Visual analysis'). Each section contains a grid of small images illustrating specific features or tools. A large 'website' button is located in the bottom right corner.

-  Welcome to the MRST User Group (1)
Av MRST-users The Matlab Reservoir Simulation Toolbox User Group - 6 innlegg - 648 visninger └
 -  Could you get the data? (9)
Av jaehw...@gmail.com - 9 innlegg - 58 visninger
-  Set wells in geology model (and perforation intervals) (3)
Av Алексей Ишков - 3 innlegg - 7 visninger
-  EXAMPLE (2)
Av 崔杰民 - 2 innlegg - 18 visninger
-  Invalid MEX-file in the mac os platform (3)
Av franko...@hotmail.com - 3 innlegg - 11 visninger
-  Rate constraints for Compositional models (4)
Av xcu...@yahoo.com - 4 innlegg - 35 visninger
-  error during interpolation of vaporized oil table (2)
Av hui...@gmail.com - 2 innlegg - 22 visninger
-  well equation in TwoPhaseWaterGasModel model (2)
Av kai.wang - 2 innlegg - 16 visninger
-  IncompTPFA generating pressure and flux Nan's..... (3)
Av Paul Morris - 3 innlegg - 4 visninger

 **An Introduction to Reservoir Simulation Using MATLAB/GNU Octave**
User Guide for the MATLAB Reservoir Simulation Toolbox (MRST)
Knut-Andreas Lie

Access  Open access
Knut-Andreas Lie, SINTEF, Norway

Publisher: Cambridge University Press
Online publication date: July 2019
Print publication year: 2019
Online ISBN: 9781108591416

DOI: <https://doi.org/10.1017/9781108591416>

```

>> help computeTrans
Compute transmissibilities.

SYNOPSIS:
T = computeTrans(G, rock)
T = computeTrans(G, rock, 'pa', pr, ...)

PARAMETERS:
G - Grid structure as described by grid_structures.

rock = Rock data structure with valid field 'pers'. The permeability
is assumed to be in measured in units of metres squared (m^2).
Use function 'darcy' to convert from darcies to m^2, e.g.:

    perm = convertFrom(perm, millidarcies)

    if the permeability is provided in units of millidarcies.

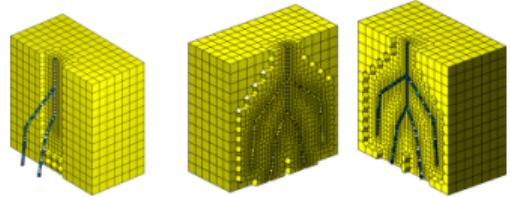
The field rock.pers may have ONE column for a scalar
permeability in each cell, TWO/THREE columns for a diagonal
permeability in each cell (in 2/3 D) and THREE/SIX columns for a
symmetric full tensor permeability. In the latter case, each
cell gets the permeability tensor

    R_d = [ k1 k2 ]           in two space dimensions
          [ k2 k3 ]
:
RETURNS:
T - half-transmissibilities for each local face of each grid
    The number of half-transmissibilities equals the number

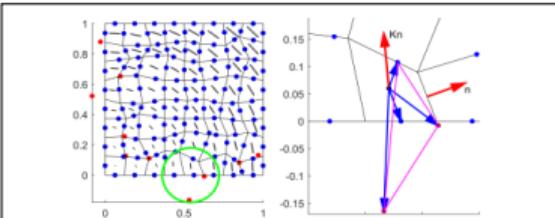
```

New book: Advanced modeling with MRST

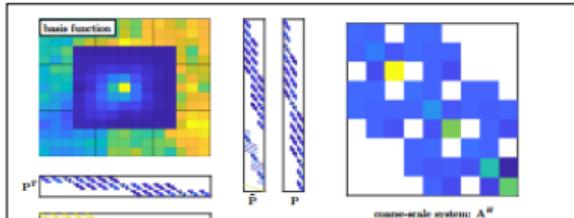
8 / 26



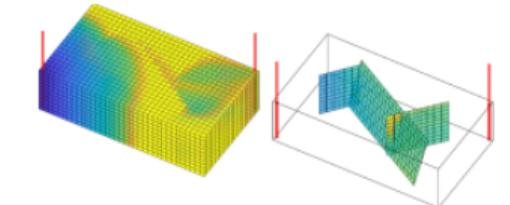
Berge et al.: Constrained Voronoi grids



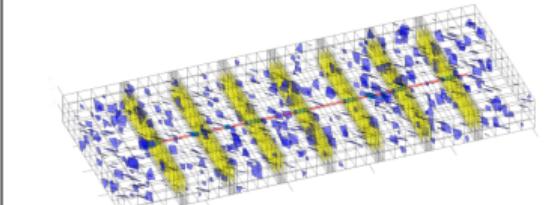
Al Kobaisi & Zhang: nonlinear FVM



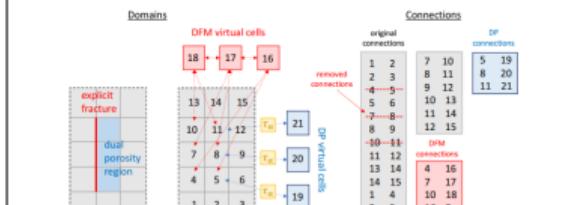
Lie & Møyner: multiscale methods



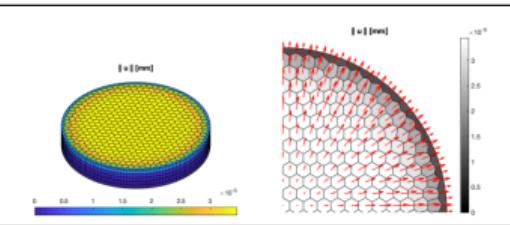
Wong et al.: embedded discrete fractures



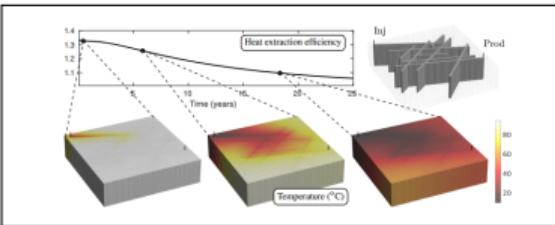
Olorode et al.: fractured unconventional



March et al.: unified framework, fractures



Varela et al.: unsaturated poroelasticity



Collignon et al.: geothermal systems

- Klemetsdal & Lie: discontinuous Galerkin
- Møyner: state functions, AD backends
- Sun et al.: chemical EOR
- Møyner: compositional
- Andersen: coupled flow & geomechanics

Minimal requirement is MATLAB version 7.4 (R2007a). No toolboxes!

Certain modules use features that were not present in R2007a:

- Automatic differentiation relies upon new-style classes (`classdef`) from R2008a.
- Various scripts and examples use new syntax for random numbers from R2007b.
- Some scripts use tilde operator from R2009b to ignore return values.
- Some solvers (e.g., fully implicit) are not efficient on versions older than R2011b.

Most of MRST can be used with GNU Octave, maybe except for some GUIs. The AD-OO solvers are somewhat slow, but will get better performance in the next release.

External dependencies:

- AGMG or AMGCL for iterative linear solvers (multigrid, etc)
- MATLAB-BGL (MATLAB Boost Graph Library) for graph algorithms
- METIS for partitioning of fully unstructured grids, etc.

Getting started

Getting the software

11 / 26

MRST - MATLAB Reservoir Simulation Toolbox

 SEARCH

MRST - MATLAB Reservoir Simulation Toolbox (MRST)

MRST is a free open-source software for reservoir modeling and simulation, developed primarily by the Computational Geosciences group in the Department of Mathematics and Cybernetics at SINTEF Digital. The software has a large international user base and also includes third-party modules developed by researchers from Heriot-Watt University, NTNU, University of Bergen, TNO, and TU Delft.

Basic functionality

Discretizations and solvers

Workflow tools

Visualization

Public data sets

TPFA

MaMFE

Grid coarsening

Flow diagnostics

Mimetic/MPFA

MaFV

Upscaling

Black-oil simulators

DFM

MaRSB

Eclipse input

MRST-geo3Lab

Adjoint methods

MsTPFA

EnKF methods

Download MRST



- URL: www.mrst.no
- Free software with GNU GPL license
- Released twice per year
- Latest release: MRST 2021a, from 19/04/2021 (yesterday!)
- Provided as a self-contained archive file (e.g., `mrst-2021a.zip`)
- We recommend that you join the MRST-announce Google group

Follow development actively:
bitbucket.org/mrst/mrst-core

NEWS

The GeoScience & GeoEnergy Webinars recently featured a talk on MRST: "Building an Open-Source Community Code: the MATLAB Reservoir Simulation Toolbox (MRST)" by Knut-Andreas Lie. [Link to video](#)

Talk at the MATLAB Energy Conference 2020: "Reservoir Modeling Using MATLAB - The MATLAB Reservoir Simulation Toolbox (MRST)" by Knut-Andreas Lie. [Link to video](#)

Talk at the MATLAB Energy Conference 2020: "MATLAB Reservoir Simulation Toolbox in Action" by Francesca Watson. [Link to video](#)

Version 2020b was released on the 30th of November 2020, and can be [downloaded](#) under the terms of the GNU General Public License (GPL). To be notified of new MRST releases please join the [MRST-announce](#) google group.

The MRST textbook by Lie has been available [online](#) since mid July 2019 and has been downloaded more than 10.000 times since then. A new [MRST book](#) that describes more advanced functionality is currently being copy-edited by Cambridge University Press. [Read more ...](#)

Unzip the software to a subfolder `mrst-2021a` of your current working directory:

```
unzip('mrst-2021a.zip')
```

Once MRST is extracted to a directory, you must navigate MATLAB there. On Linux/Mac OS,

```
cd /home/username/mrst-2021a/
```

or on Windows,

```
cd C:\Users\username\mrst-2021a\
```

assuming that the files were extracted to the home directory. The `startup.m` file must then be run to activate MRST,

```
startup;
```

or you can call the `startup` script directly

```
run /home/username/mrst-2021a/startup
```

If you start MATLAB in the directory containing MRST, or run the `startup.m` file, you will see the following message

```
Command Window

Welcome to the Matlab Reservoir Simulation Toolbox (MRST) !
You are using the release version 2020b. To download other versions of MRST
and view examples and relevant publications, please visit www.mrst.no

Useful commands for getting started:
- List all introductory examples: mrstExamples\(\)
- List all modules: mrstPath\('list'\)
- Load modules using GUI: mrstModule\('qui'\)
- Explore all available data sets mrstDatasetGUI\(\)
- List examples of a module: mrstExamples\('ad-blackoil'\)
- Explore modules and publications: mrstExploreModules\(\)
- Show all examples in all modules: mrstExamples\('all'\)
- Display this message: mrstStartupMessage\(\)

For assistance and discussions about MRST, please visit our mailing list at
www.sintef.no/projectweb/mrst/forum/ (sintef-mrst@googlegroups.com)
For some common queries, see our FAQ: www.sintef.no/projectweb/mrst/faq/

fx >>
< >
```

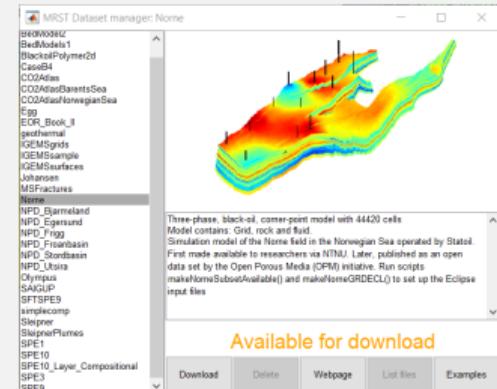
Interactive demonstration

The core module of MRST offers a number of examples that introduce you to data structures and data sets, how to set up basic solvers, how to visualize input data and simulation results, etc

```
>> mrstExamples
Module "core" has 14 examples:
    flowSolverTutorial1.m
    flowSolverTutorialAD.m
    tutorialAD.m
    tutorialBasicObjects.m
    tutorialPlotting.m
    datasets\showCaseB4.m
    datasets\showJohansen.m
    datasets\showNorne.m
    datasets\showSAIGUP.m
    datasets\showSPE10.m
    grids\gridTutorialCornerPoint.m
    grids\gridTutorialIntro.m
    grids\gridTutorialStruct.m
    grids\gridTutorialUnstruct.m
```

We go through the following:

- Visit the website, forum, and FAQ
- List basic tutorials, `mrstExample()`
- Bring up `mrstExploreModules()`
- Bring up `mrstDatasetGUI()`



Graphical user interface to modules:

```
mrstModule('gui')  
moduleGUI
```

List all modules and their path

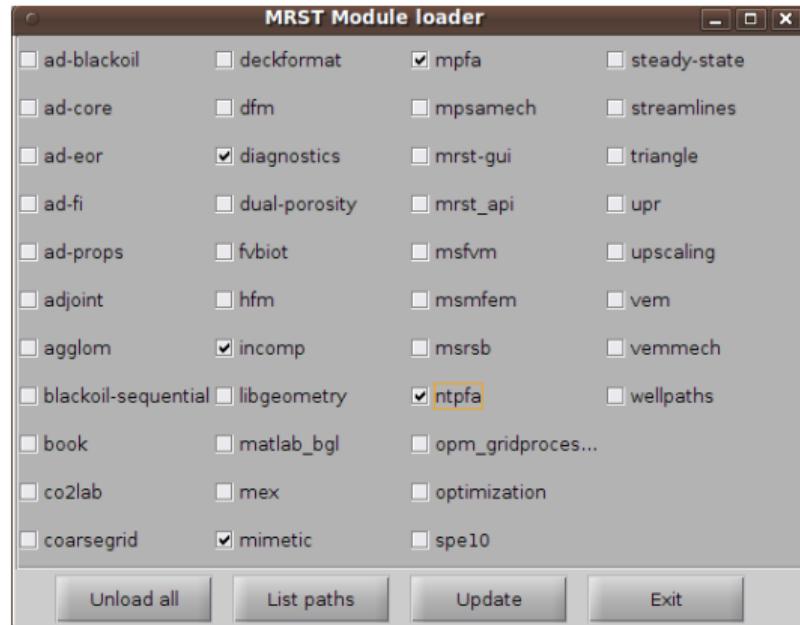
```
mrstPath
```

Load new modules

```
mrstModule add mimetic mpfa
```

Adding your own modules

```
mrstPath reregister distmesh ...  
/home/username/mrst-2016b/utils/3rdparty/distmesh
```

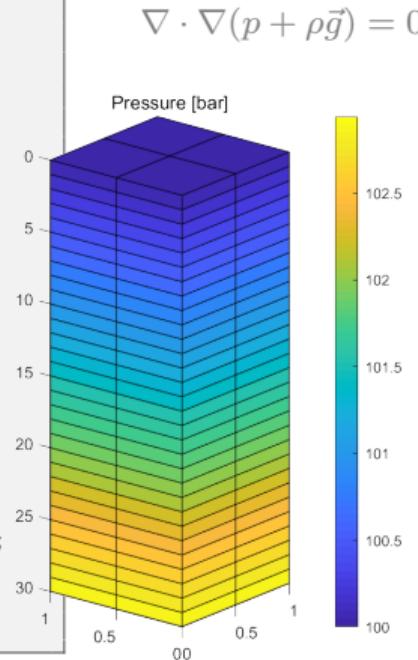


```
% Activate module for incompressible solvers
mrstModule add incomp

%% Define the model
gravity reset on
G = cartGrid([2, 2, 30], [1, 1, 30]);
G = computeGeometry(G);
rock = makeRock(G, 0.1*darcy, 1);
fluid = initSingleFluid('mu', 1*centi*poise, ...
    'rho', 1014*kilogram/meter^3);
bc = pside([], G, 'TOP', 100.*barsa());

%% Assemble and solve the linear system
T = computeTrans(G, rock);
sol = incompTPFA(initResSol(G, 0.0), ...
    G, T, fluid, 'bc', bc);

%% Plot the face pressures
newplot;
plotFaces(G, 1:G.faces.num, sol.facePressure./barsa);
set(gca, 'ZDir', 'reverse', 'DataAspectRatio', [1 1 10]);
title('Pressure [bar]')
view(3), colorbar
```



Code: flowSolverTutorial1.m

Will go through this code in detail.

Earliest parts of MRST:

- Procedural programming
- Structs for reservoir state, rock parameters, wells, b.c., and source term
- Fluid behavior: struct with function pointers

Advantages:

- **hide specific details** of geomodel and fluid model
- **vectorization**: efficient/compact code
- **unified access** to key parameters

softwareunderground.org presents



TRANSFORM 2021

Virtual Conference on the Digital Subsurface, 16–23 April

supported by



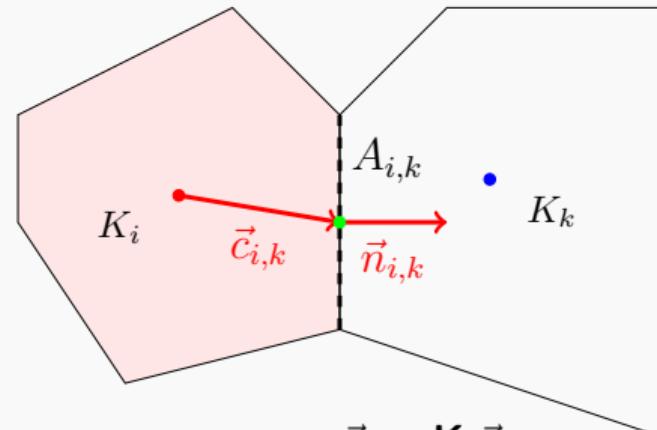
Taking a break... back soon.

Numerical framework

Conservation of momentum (Darcy's law):

$$\int_{\Gamma_f} \vec{v}(x) \cdot \vec{n}_f \, ds = - \int_{\Gamma_f} \mathbf{K}(x) \nabla p \cdot \vec{n}_f \, ds$$

Two-point flux approximation



$$T_{i,k} = A_{i,k} \frac{\vec{c}_{i,k} \cdot \mathbf{K}_i \vec{n}_{i,k}}{|\vec{c}_{i,k}|^2}$$

$$T_{ik} = [T_{i,k}^{-1} + T_{k,i}^{-1}]^{-1}$$

$$v_{ik} = T_{ik}(p_i - p_k)$$

Conservation of momentum (Darcy's law):

$$\int_{\Gamma_f} \vec{v}(x) \cdot \vec{n}_f \, ds = - \int_{\Gamma_f} \mathbf{K}(x) \nabla p \cdot \vec{n}_f \, ds$$

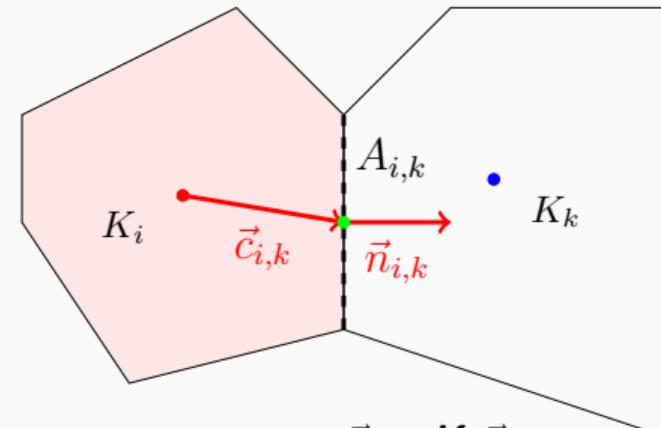
discrete: $\mathbf{v}[f] = -\mathbf{T}[f] \operatorname{grad}(\mathbf{p})[f]$

Conservation of mass:

$$\int_{\partial\Omega_c} \vec{v} \cdot \vec{n} \, ds = \int_{\Omega_c} \nabla \cdot \vec{v} \, d\vec{x} = \int_{\Omega_c} q \, d\vec{x}$$

discrete: $\operatorname{div}(\mathbf{v})[c] = \mathbf{q}[c]$

Two-point flux approximation

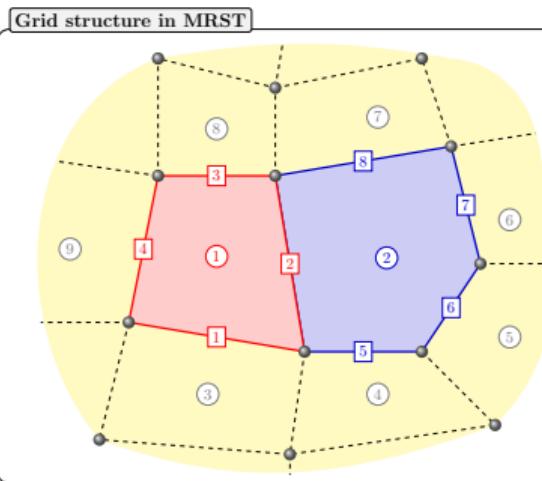
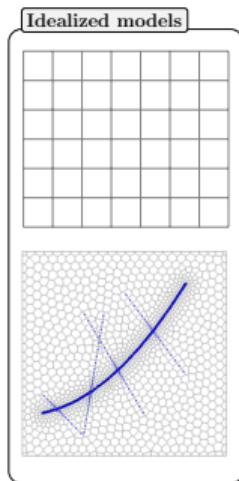


$$T_{i,k} = A_{i,k} \frac{\vec{c}_{i,k} \cdot \mathbf{K}_i \vec{n}_{i,k}}{|\vec{c}_{i,k}|^2}$$

$$T_{ik} = [T_{i,k}^{-1} + T_{k,i}^{-1}]^{-1}$$

$$v_{ik} = T_{ik}(p_i - p_k)$$

Rapid prototyping: discrete differentiation operators

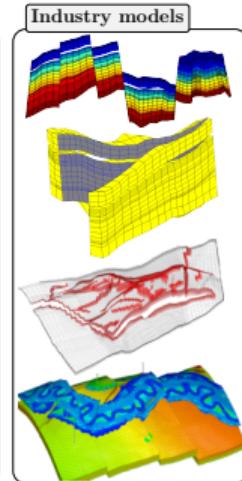


c	F(c)
1	1
1	2
1	3
1	4
2	5
2	6
2	7
2	8
2	9
3	1
⋮	⋮
⋮	⋮

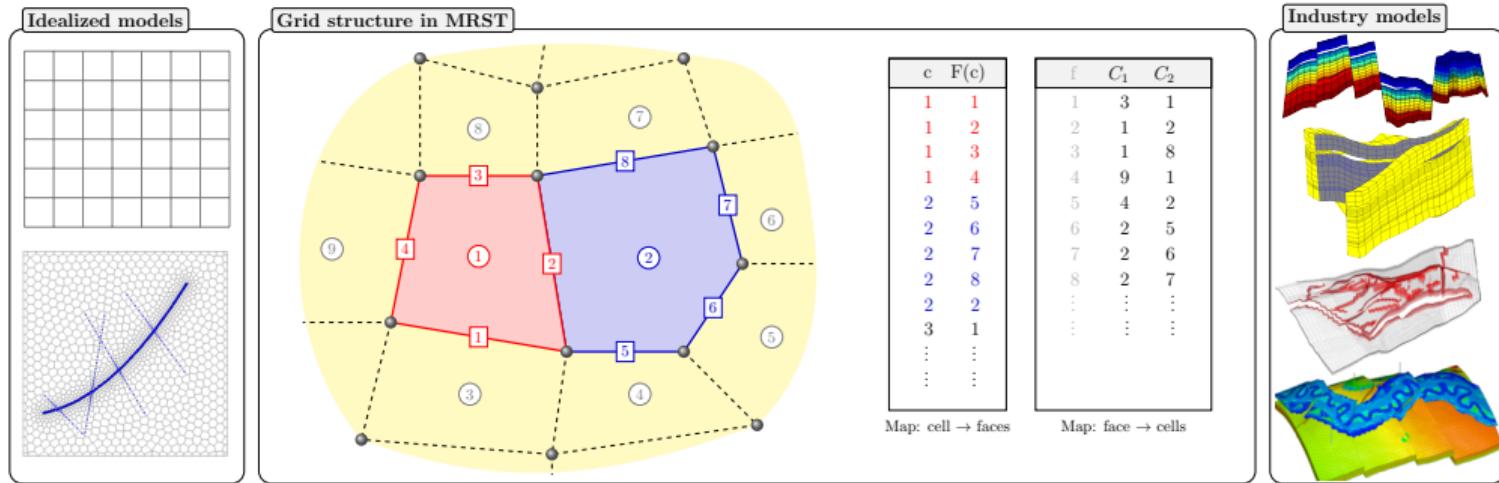
Map: cell \rightarrow faces

\mathbb{F}	C_1	C_2
1	3	1
2	1	2
3	1	8
4	9	1
5	4	2
6	2	5
7	2	6
8	2	7
⋮	⋮	⋮
⋮	⋮	⋮

Map: face \rightarrow cells



Rapid prototyping: discrete differentiation operators



For finite volumes, the discrete grad operator maps from cell pair $C_1(f), C_2(f)$ to face f :

$$\text{grad}(\mathbf{p})[f] = \mathbf{p}[C_2(f)] - \mathbf{p}[C_1(f)],$$

where $\mathbf{p}[c]$ is a scalar quantity associated with cell c . Discrete div maps from faces to cells.

Both are linear operators and can be represented as sparse matrix multiplications.

Continuous

Incompressible flow:

$$\nabla \cdot (\mathbf{K} \nabla p) + q = 0$$

Compressible flow:

$$\frac{\partial(\phi\rho)}{\partial t} + \nabla \cdot (\rho \mathbf{K} \nabla p) + q = 0$$

Discrete in MATLAB

Incompressible flow:

```
eq = div(T .* grad(p)) + q;
```

Compressible flow:

```
eq = (pv(p).*rho(p)-pv(p0).*rho(p0))/dt ...  
+ div(avg(rho(p)).*T.*grad(p))+q;
```

Continuous

Incompressible flow:

$$\nabla \cdot (\mathbf{K} \nabla p) + q = 0$$

Compressible flow:

$$\frac{\partial(\phi\rho)}{\partial t} + \nabla \cdot (\rho \mathbf{K} \nabla p) + q = 0$$

Discrete in MATLAB

Incompressible flow:

```
eq = div(T .* grad(p)) + q;
```

Compressible flow:

```
eq = (pv(p).*rho(p)-pv(p0).*rho(p0))/dt ...  
+ div(avg(rho(p)).*T.*grad(p))+q;
```

Discretization of flow models leads to large systems of nonlinear equations. Can be linearized and solved with Newton's method

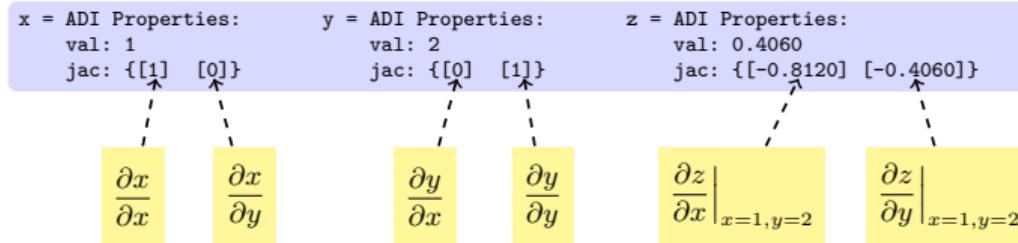
$$\mathbf{F}(\mathbf{u}) = \mathbf{0} \quad \Rightarrow \quad \frac{\partial \mathbf{F}}{\partial \mathbf{u}}(\mathbf{u}^i)(\mathbf{u}^{i+1} - \mathbf{u}^i) = -\mathbf{F}(\mathbf{u}^i)$$

Coding necessary Jacobians is time-consuming and error prone

General idea:

- Any code consists of a limited set of arithmetic operations and elementary functions
- Introduce an extended pair, $\langle x, 1 \rangle$, i.e., the value x and its derivative 1
- Use chain rule and elementary derivative rules to mechanically accumulate derivatives *at specific values of x*
 - Elementary: $v = \sin(x) \rightarrow \langle v \rangle = \langle \sin x, \cos x \rangle$
 - Arithmetic: $v = fg \rightarrow \langle v \rangle = \langle fg, fg_x + f_x g \rangle$
 - Chain rule: $v = \exp(f(x)) \rightarrow \langle v \rangle = \langle \exp(f(x)), \exp(f(x))f'(x) \rangle$
- Use operator overloading to avoid messing up code

```
[x,y] = initVariablesADI(1,2);
z = 3*exp(-x*y)
```



Solving the Poisson equation: unstructured grid

23 / 26

% Grid and grid information

```
load seamount
G = pebi(triangleGrid([x(:) y(:)]));
G = computeGeometry(G);
rock = makeRock(G, 1, 1);
nc = G.cells.num;
```

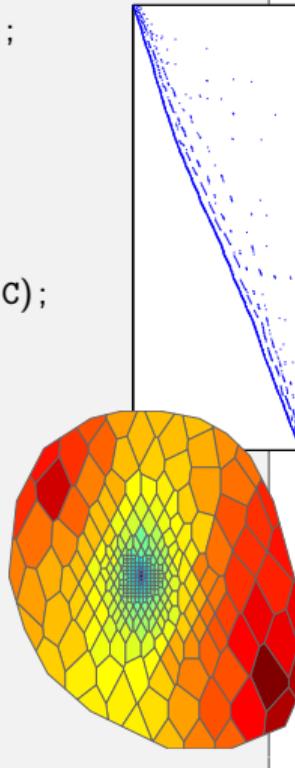
% Operators

```
S = setupOperatorsTPFA(G,rock); spy(S.C);
```

% Assemble and solve equations

```
p = initVariablesADI(zeros(nc,1));
q = zeros(nc, 1)
q([135 282 17]) = [-1 .5 .5];
```

```
eq = S.Div(S.T.*S.Grad(p))+q;
eq(1) = eq(1) + p(1);
p = -eq.jac{1}\eq.val;
plotCellData(G,p);
```



Discretization of $-\nabla \cdot (\mathbf{K} \nabla p) = q$ gives the residual flow equation

$$\mathbf{0} = \operatorname{div}(\mathbf{T} \operatorname{grad}(\mathbf{p})) + \mathbf{q} = \mathbf{F}(\mathbf{p})$$

Automatic differentiation gives us $\partial \mathbf{F} / \partial \mathbf{p}$

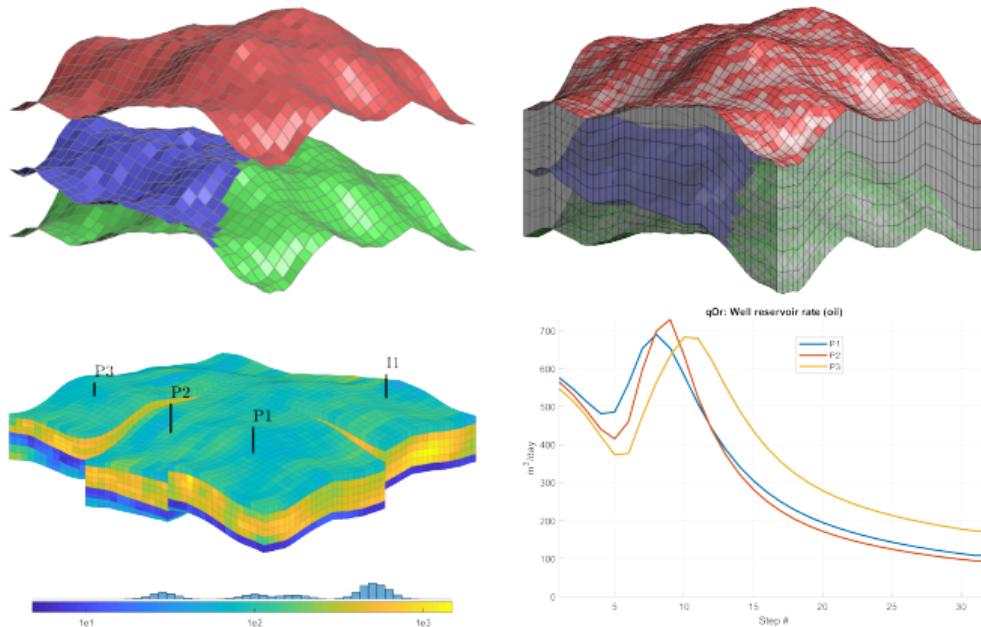
Go through the code in detail:

- Explain the gridding
- Show the grid structure
- Look at the operators
- Look at Jacobians, etc

More complex example

Example: from geological horizons to simulation

25 / 26



Worked simulation example:

- Define horizons
- Extrude grid to mimic stratigraphy
- Introduce structural architecture
- Add petrophysics
- Define wells
- Setup fluid model
- Initial state
- Simulation schedule (well controls)
- Simulate with computational steering

softwareunderground.org presents



TRANSFORM 2021

Virtual Conference on the Digital Subsurface, 16–23 April

supported by



The livestream is over. Thanks for taking part!