A Unified Framework for Flow Simulation in Fractured Reservoirs

Featured in the upcoming MRST Book

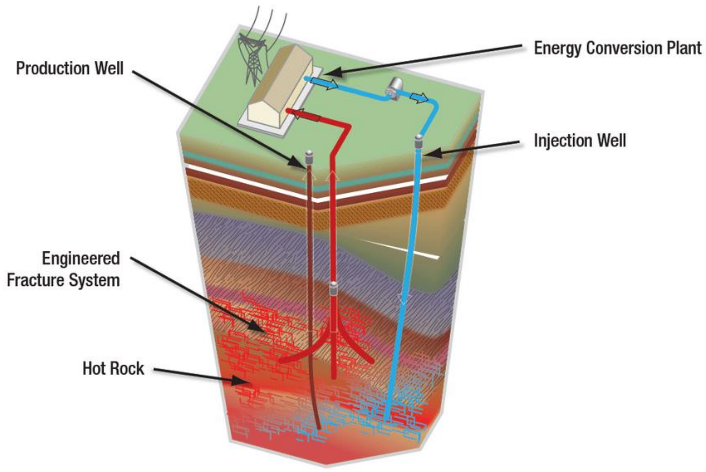
Christine Maier, Rafael March, Florian Doster, Sebastian Geiger

Fractured and Geothermal Reservoir Group
Institute of GeoEnergy Engineering
Heriot-Watt University
https://geofracs.hw.ac.uk/

'a foundation for the future'

Background and Motivation

- Modelling of multi-phase fluid flow in fractured rocks is relevant to several energy-related applications
- Some of these include design of Enhanced Oil Recovery operations, Enhanced Geothermal Systems and Carbon Sequestration in underground formations
- Over the last decades several approaches were developed to simulate/model flow in fractured reservoirs; each with its own advantages and pitfalls
- Here we present MRST's fracture module which provides a platform for quick implementation of fractured reservoir simulation models

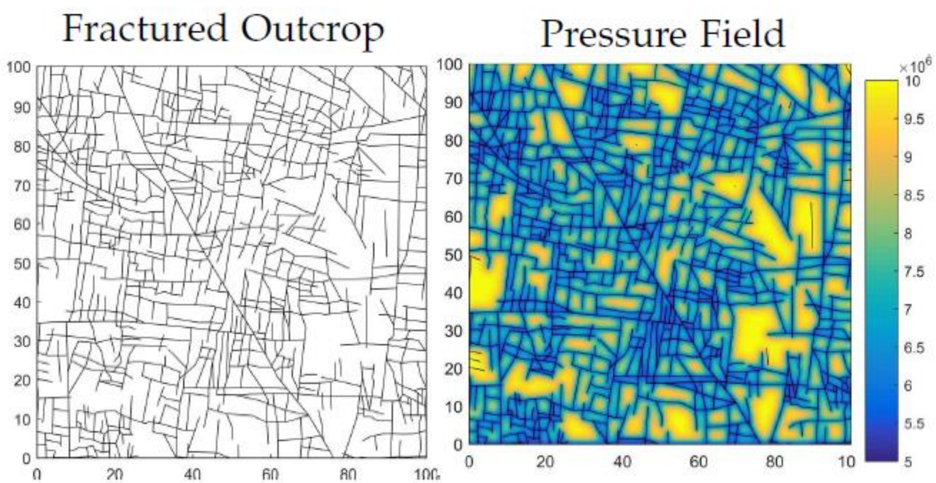


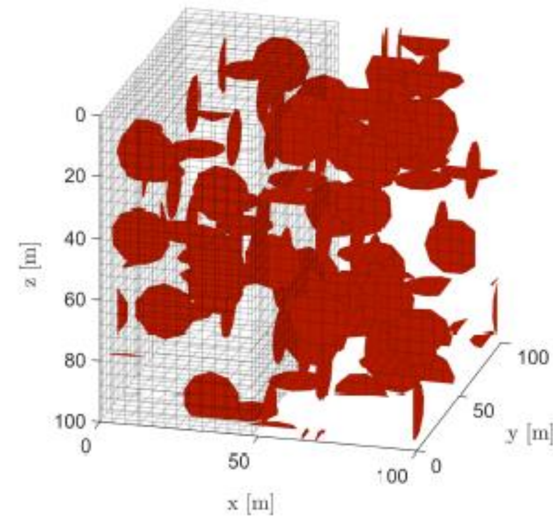
* Figures extracted from *Thiel, S. 2017 (*DOI:10.1007/s10712-017-9426-2) and from the upcoming MRST book (Advanced Modeling with the MATLAB Reservoir Simulation Toolbox)

Modelling Approaches for Fluid Flow in Fractured Media: Explicit Fracture Representation

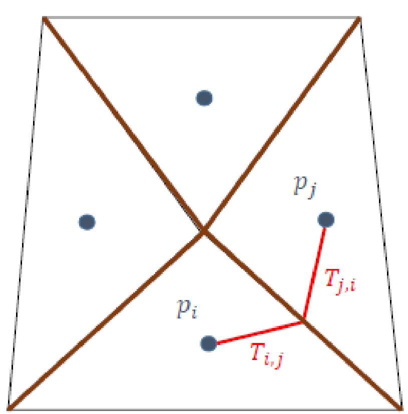
DFM

EDFM

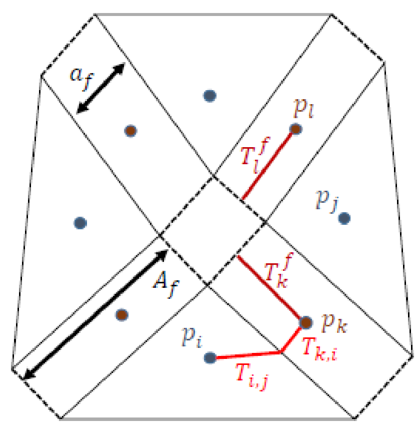


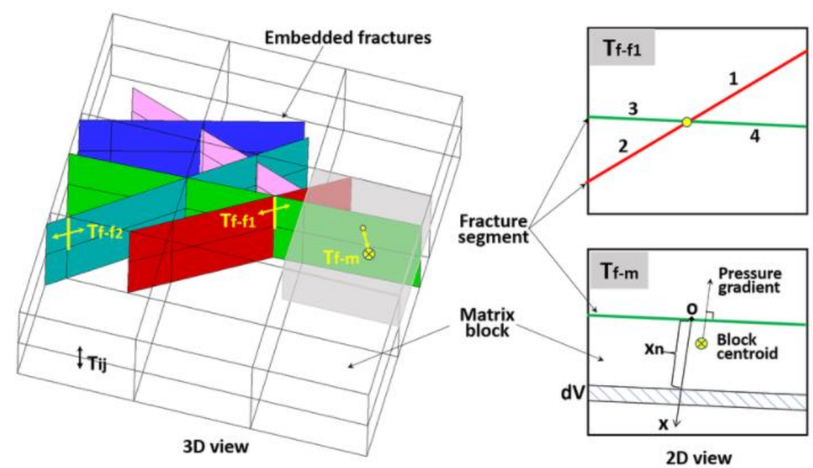


Geometrical domain



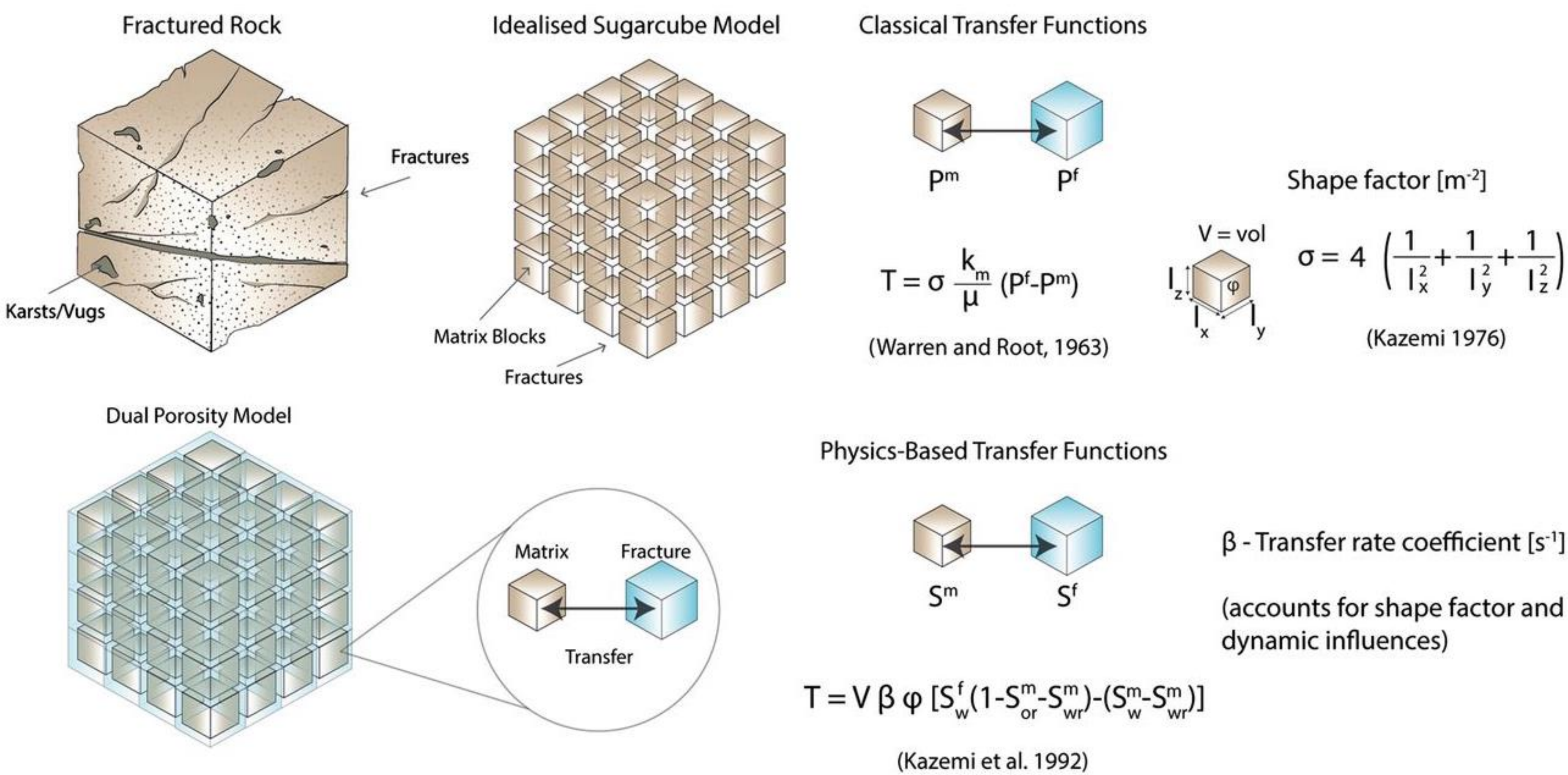
Numerical domain





* Figures extracted from *Li et al.* 2020 (DOI: 10.1029/2019WR025631), from Rafael March's and Daniel Wong's PhD Theses at Heriot-Watt University, and from the upcoming MRST book (Advanced Modeling with the MATLAB Reservoir Simulation Toolbox)

Modelling Approaches for Fluid Flow in Fractured Media: Multi-Continuum Methods



* Figure extracted from *Spooner et al., 2019* (doi.org/10.1144/petgeo2018-136)

Unified Framework in MRST (1/2)

How to combine into one single modelling concept?

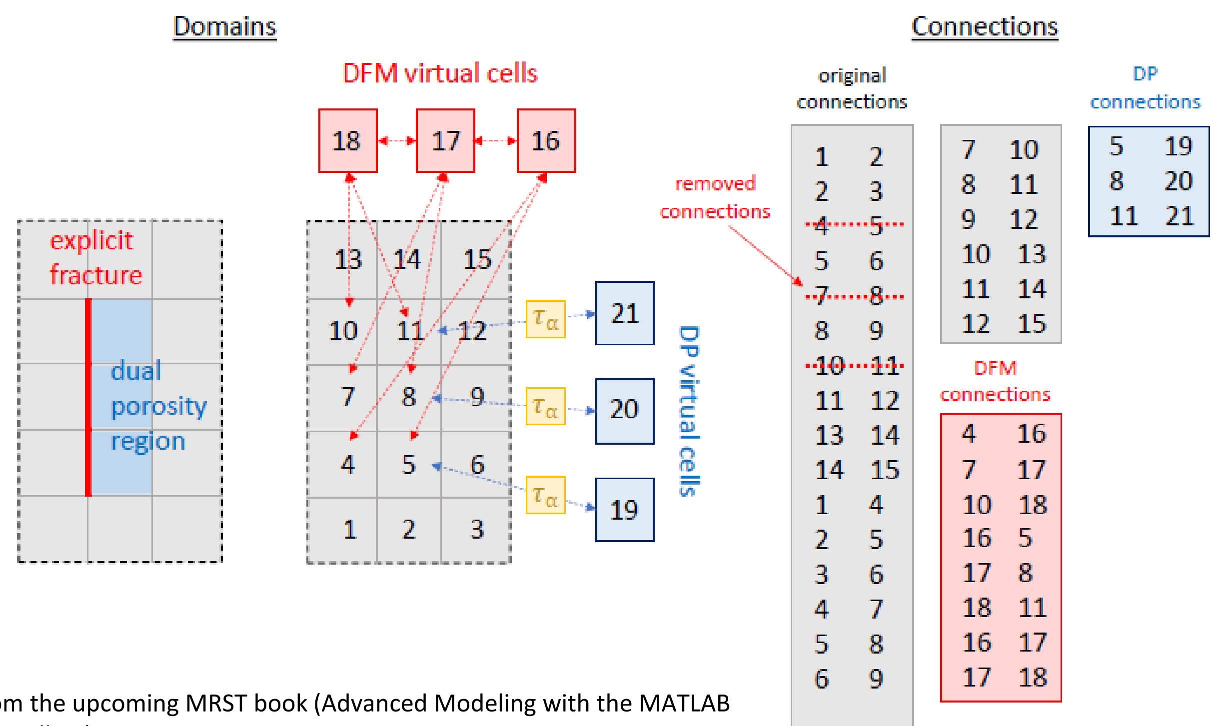
<u>DFM, EDFM</u>

$$\frac{\partial \phi \rho_\alpha S_\alpha}{\partial t} + \nabla \cdot \rho_\alpha \boldsymbol{v}_\alpha = \rho_\alpha q_\alpha$$

<u>Multi-Continuum</u>

$$\frac{\partial \phi_f \rho_\alpha S_{\alpha f}}{\partial t} + \nabla \cdot \rho_\alpha \boldsymbol{v}_{\alpha f} = \rho_\alpha q_{\alpha f} - \tau_\alpha$$

$$\frac{\partial \phi_m \rho_\alpha S_{\alpha m}}{\partial t} + \nabla \cdot \rho_\alpha \boldsymbol{v}_{\alpha m} = \tau_\alpha$$



* Figure extracted from the upcoming MRST book (Advanced Modeling with the MATLAB Reservoir Simulation Toolbox)

Unified Framework in MRST (2/2)

How are different models implemented in MRST?

```
[b, pv]    = model.getProps(state,  'ShrinkageFactors', 'PoreVolume');
[b0, pv0] = model.getProps(state0, 'ShrinkageFactors', 'PoreVolume');
[phaseFlux, flags] = model.getProps(state, 'PhaseFlux',  'PhaseUpwindFlag');
[pressures, mob, rho] = model.getProps(state, 'PhasePressures', 'Mobility',
                                              'Density');
```

```
[bW, bO]     = deal(b{:});
[bW0, bO0]   = deal(b0{:});
[vW, vO]     = deal(phaseFlux{:});
[upcw, upco] = deal(flags{:});

% Accumulation term for water and oil phase
water = (1/dt).*( pv.*bW.*sW - pv0.*bW0.*sW0 );
oil   = (1/dt).*( pv.*bO.*sO - pv0.*bO0.*sO0 );
eqs   = {water, oil};

% Fluxes across faces (connections)
eqs{1} = eqs{1} + s.Div(s.faceUpstr(upcw, bW).*vW);
eqs{2} = eqs{2} + s.Div(s.faceUpstr(upco, bO).*vO);
```

State functions specify how porosity, fluxes, density and others depend on the state variables

Fluxes between connections can be either vanilla Darcy fluxes or modelled through transfer functions

Fluxes between the different domains

```
function v = evaluateOnDomain(prop, model, state)
    [mob, kgrad] = prop.getEvaluatedDependencies(state, 'FaceMobility',
                                                  'PermeabilityPotentialGradient');
    % Standard flux evaluation
    v = cellfun(@(x,y)-x.*y, mob, kgrad, 'UniformOutput', false);

    % Compute fluxes between multi-continuum domains
    for j = 1:length(model.G.FracturedDomains.domains)
        dom = model.G.FracturedDomains.domains{j};
        if(strcmp(dom.type,'multi_continuum'))
            ids = dom.global_connection_ids;
            vf = dom.transfer_model.transfer(model, state, j);
            for i = 1:numel(mob)
                v{i}(ids) = vf{i};
            end
        end
    end
```

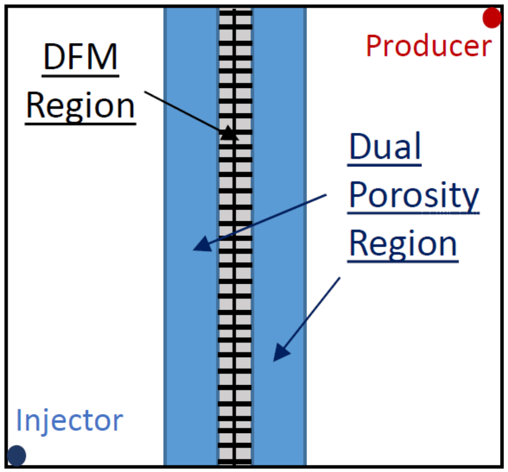
```
function v = transfer(obj, model, state, domain_id)
    [p, s, flag, mob, rho] = ...  % Get state variables
        model.getProps(state, 'PhasePressures', ...
                       'Saturation', 'PhaseUpwindFlag', 'Mobility', 'Density');
    % Get domain object
    dom = model.G.FracturedDomains.domains{domain_id};
    % Calculate shape factor
    sigma = obj.shape_factor_model.calculate_shape_factor(dom.rock);
    vb = model.G.cells.volumes(dom.region);
    % Compute fluxes (transfer)
    pf = {}; pm = {};         % fracture and matrix pressures
    for i = 1:length(p)
        % Pressures in fracture and matrix
        pf{i} = p{i}(dom.connections(:,1));
        pm{i} = p{i}(dom.connections(:,2));
        % Fluxes
        fmob = model.operators.faceUpstr(flag{i}, mob{i});
        fmob = fmob(dom.global_connection_ids);
        v{i} = vb.*fmob.*sigma.*(pf{i}-pm{i});
    end
end
```

Transfer Function (*"Kazemi"*)

* Figure extracted from the upcoming MRST book (Advanced Modeling with the MATLAB Reservoir Simulation Toolbox)

Example: DFM and Dual-Porosity in the Same Model

Model Definition

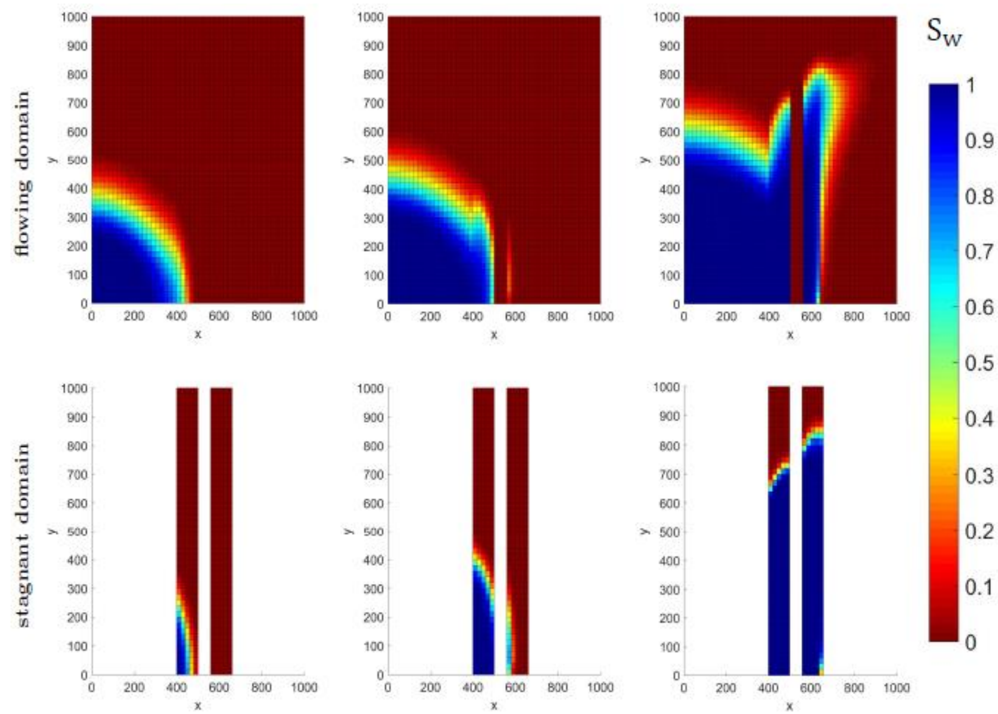



```
%% Dual-porosity cells, fault core cells and DFM edges
xc=G.cells.centroids(:,1); xf=G.faces.centroids(:,1); yf = G.faces.centroids(:,2));
dual_porosity_cells = find(((xc >= 400) & (xc <= 500)) | ...
                     ((xc >= 560) & (xc <= 650)));
fault_core_cells = find(xc>=500 & xc<=550);
dfm_edges = find((xf >= 490) & (xf <= 570) & (yf < y_size) & (yf > 0));
```

```
%% Fractured domain manager and transfer model
manager = FracturedDomainManager();
t_model = KazemiMultiphaseTransferFunction(KazemiShapeFactor([10, 10, 10]));

% Dual-porosity domain
model = manager.addFracturedDomain(model, 'multi_continuum',
                        dual_porosity_cells, dual_porosity_matrix_rock,
                        dual_porosity_matrix_fluid,
                        'transfer_model', t_model);
% DFM domain
model = manager.addFracturedDomain(model, 'DFM', dfm_edges,
                        dfm_fracture_rock, dfm_fracture_fluid);
```

Results

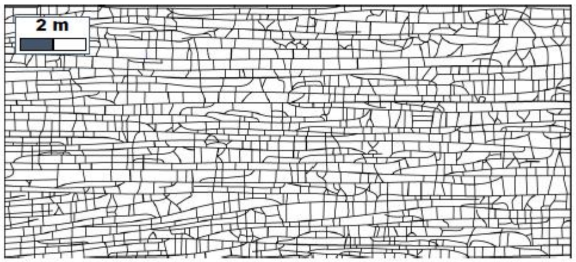


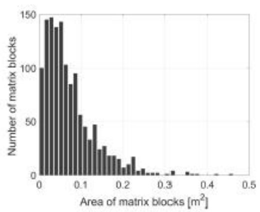
Water saturation profiles after 1500 days, 2500 days and 6000 days of simulation. Top row shows the water saturation at the flowing domain, which consists of the matrix of the single-porosity region, the fractures of the dual-porosity region and the DFM fractures. Bottom row shows the water saturation at the stagnant domain, which consists of the matrix of the dual-porosity region.

* Figures extracted from the upcoming MRST book (Advanced Modeling with the MATLAB Reservoir Simulation Toolbox)

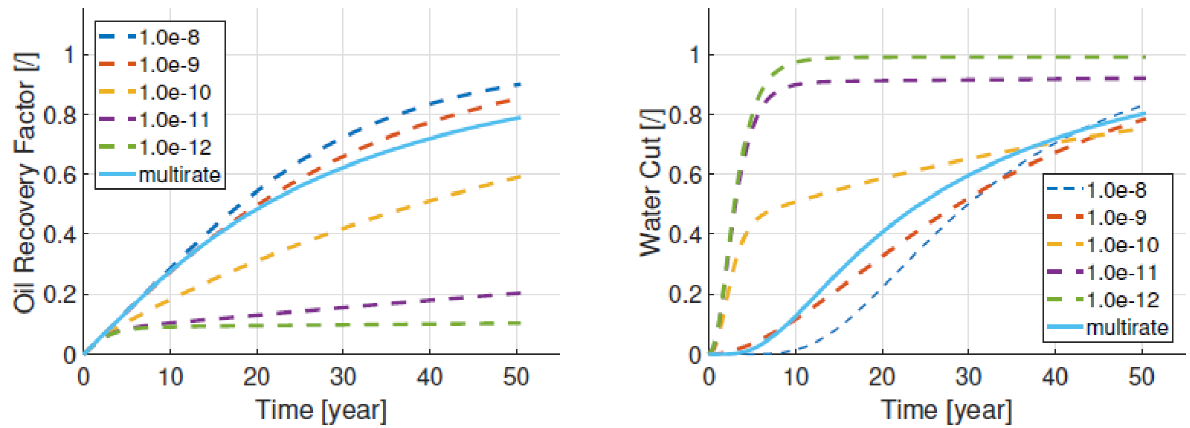
Example: Multi-rate Transfer in Multi-Continuum Models

Model Definition





Results



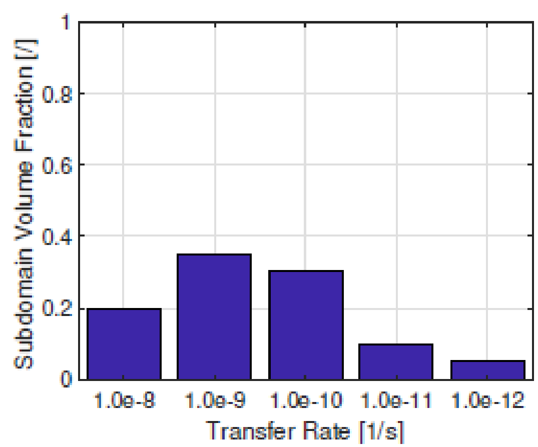
Oil recovery factor and water cut for different rates compared with the multi-rate case.

```
%% Setting pore volume multiplier to account for subdomain volume fractions
sub_domains_fraction = [0.2; 0.35; 0.3; 0.1; 0.05];
for i=1:length(sub_domains_fraction)
    mrtr_matrix_fluid{i}.pvMultR = @(p) p*0 + sub_domains_fraction(i);
end

%% Fractured domain manager
manager = FracturedDomainManager();
```

```
%% Transfer models
t_rates = [1e-8, 1e-9, 1e-10, 1e-11, 1e-12];
t_model = cell(length(sub_domains_fraction),1);
for i = 1:length(transfer_rates)
  t_model{i} = SaturationDifferenceTransferFunction(t_rates(i));
end

for i=1:length(sub_domains_fraction)
    model = manager.addFracturedDomain(model, 'multi_continuum', mrtr_cells,
                                       mrtr_matrix_rock, mrtr_matrix_fluid{i},
                                       'transfer_model', t_model{i});
end
```



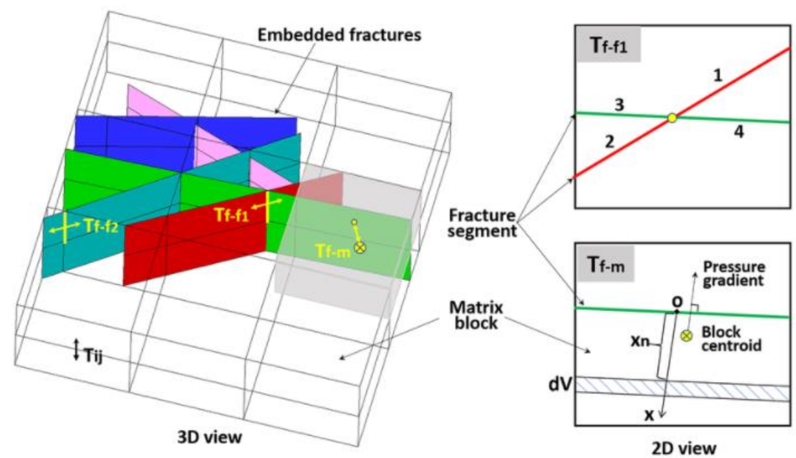
* Figures extracted from the upcoming MRST book (Advanced Modeling with the MATLAB Reservoir Simulation Toolbox)

Great Models that are not implemented yet (give it a go!)

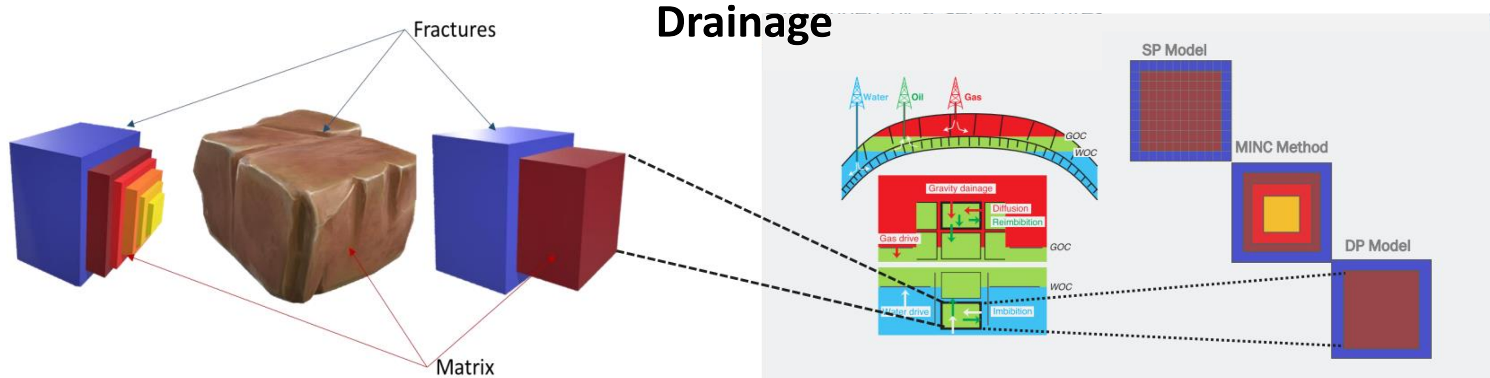
Package Structure

```
fractures
├── examples
│   ├── dfm-and-dual-porosity
│   ├── multirate-transfer
│   ├── pressure-buildup
│   └── validation-dfm
├── manager
│   ├── state-functions
│   └── FracturedDomainManager.m
├── transfer-functions
│   ├── shape-factors
│   ├── KazemiMultiphaseTransferFunction.m
│   └── SaturationDifferenceTransferFunction.m
└── utils
```

EDFM and pEDFM



MINC / Sub-Domain for Gravity Drainage



* Figures extracted from Al-Rudaini et al. 2018 (THE LIMITATIONS OF USING DUAL POROSITY MODELS WHEN SIMULATING CHEMICALLY ENHANCED OIL RECOVERY PROCESSES, Poster Presented at Heriot Watt's Energi Simulation Workshop) and *Li et al. 2020 (*DOI: 10.1029/2019WR025631)

Acknowledgments

Rafael March and Florian Doster thank the European Commission, the Research Council of Norway, the Rijksdienst voor Ondernemend Nederland, the Bundesministerium für Wirtschaft und Energie, and the Department for Business, Energy & Industrial Strategy, UK for funding through the ERANET Cofund ACT (Project no. 271497). Rafael March, Florian Doster, and Sebastian Geiger would also like to thank the U.S. Department of Energy (DOE) National Energy Technology Laboratory (NETL) who funded Rafael's PhD work under grant FE0023323. Neither the United States Government nor any agency thereof, nor any of their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise does not necessarily constitute or imply its endorsement, recommendation, or favoring by the U.S. Government or any agency thereof. The views and opinions of authors expressed herein do not necessarily state or reflect those of the U.S. Government or any agency thereof. Christine Maier and Sebastian Geiger thank Energi Simulation for supporting Christine's Research Associate position and Sebastian's Chair for Carbonate Reservoir Simulation at Heriot–Watt University.