ENSEMBLE SIMULATIONS

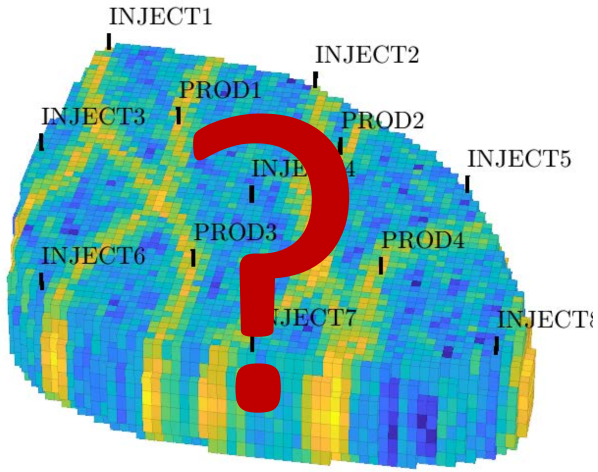
IN MRST USING THE ENSEMBLE MODULE

Håvard Heitlo Holm, Øystein Strengehagen Klemetsdal

September 15th 2021

MRST Symposium

Uncertainties in subsurface simulation



- Numerical model of subsurface reservoir with parameters θ transforming the initial state x_0 into well responses y at time t:

$$y = G(t;\, x_0, \theta)$$

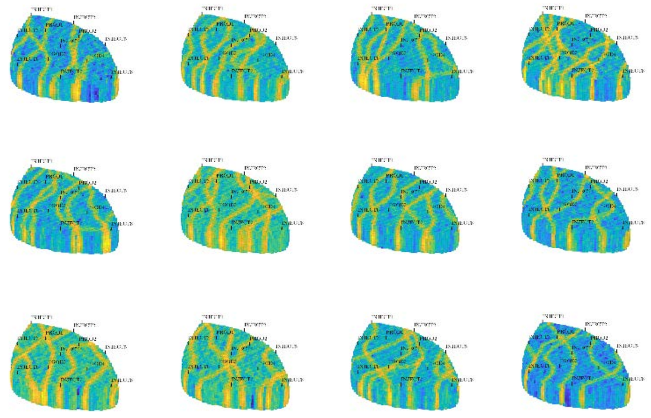
- x_0, θ not really known, and therefore y is subject to uncertainty.
- Classical Monte Carlo methods give mean and variance of

$$\bar{y} = \sum_{i=1}^{N_e} y_i \quad \text{and} \quad \mathrm{Var}(y) = \sum_{i=1}^{N_e} (y_i - \bar{y})(y_i - \bar{y})^T$$

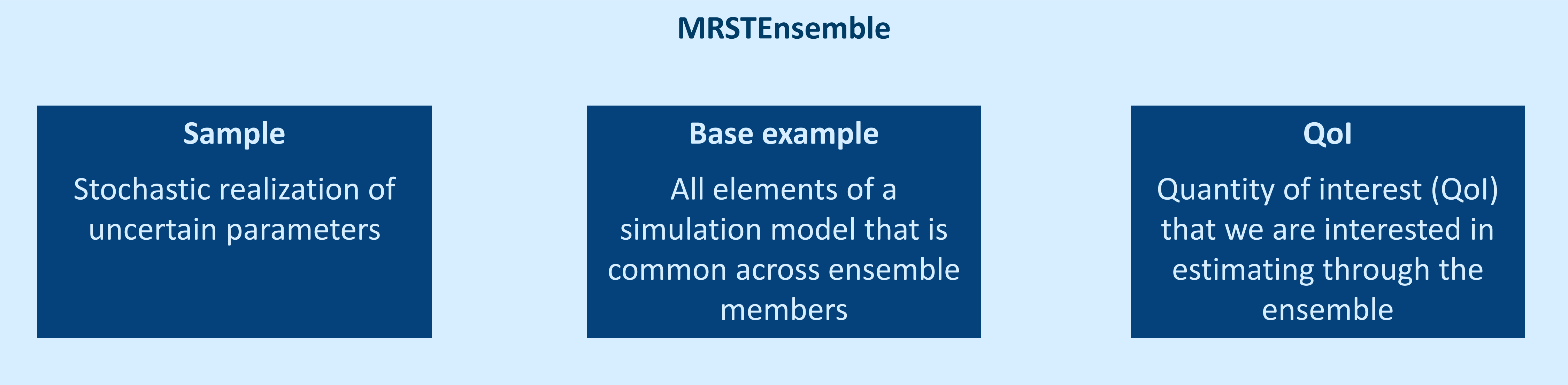
in which

$$y_i = G(t; x_{0,i}, \theta_i)$$

with $x_{0,i} \sim p(x_0)$ and $\theta_i \sim p(\theta)$.



MRSTEnsemble class



MRSTEnsemble

Sample

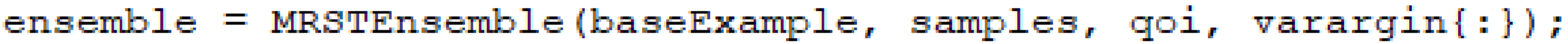
Stochastic realization of uncertain parameters

Base example

All elements of a simulation model that is common across ensemble members

QoI

Quantity of interest (QoI) that we are interested in estimating through the ensemble



```
ensemble = MRSTEnsemble(baseExample, samples, qoi, varargin{:});
```

Base example

- Defines everything that is common to all ensemble members

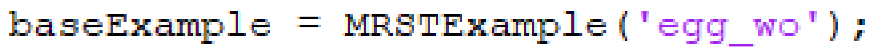
- Instance of `MRSTExample` from the `example-suite` module

→Contains `model`, `state0`, `schedule`, and more

→`getPackedSimulationProblem()`

- Build ensembles from existing examples, or easily define your own

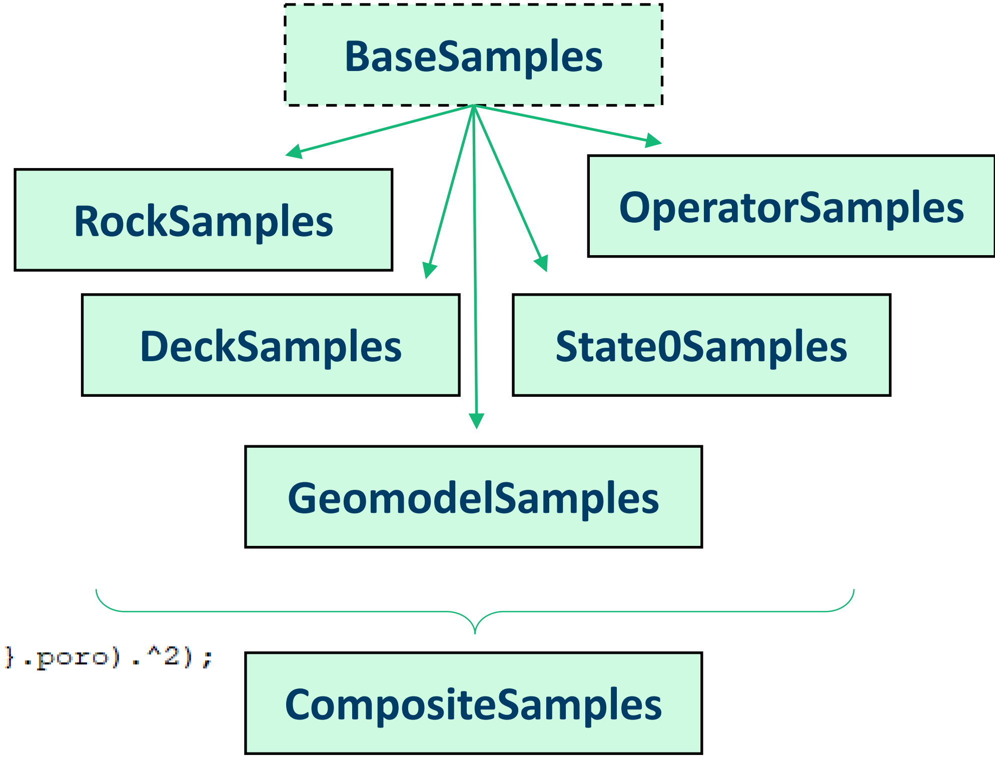
→ Full simulation models defined in dedicated functions



```
baseExample = MRSTExample('egg_wo');
```

Samples

- The stochastic component that makes each ensemble member unique
- Can be defined in three ways:
  - Cell array of precomputed data
  - Function that generate stochastic realizations on the fly
  - Result handlers pointing to pre-generated data on disk
- Defines how specific sample types are mapped onto the base example


BaseSamples

RockSamples

OperatorSamples

DeckSamples

State0Samples

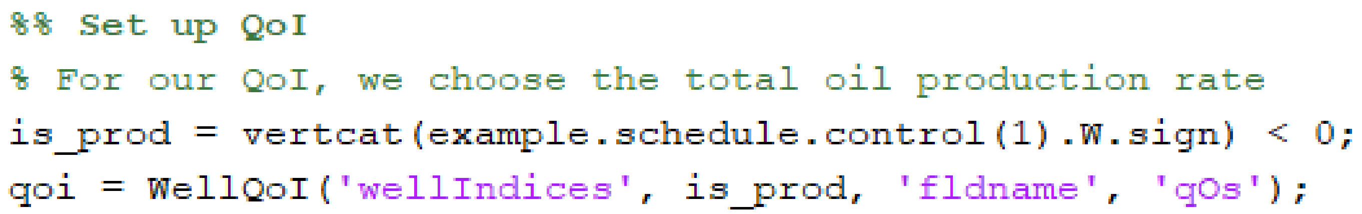
GeomodelSamples

CompositeSamples

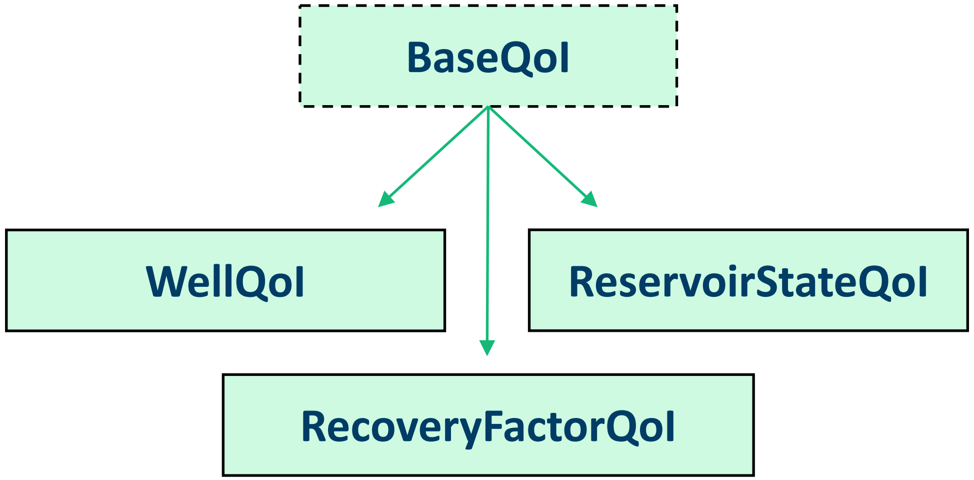
```
configData = cell(ensembleSize, 1);
for i = 1:ensembleSize
    configData{i}.poro = gaussianField(baseExample.model.G.cartDims, [0.2 0.4]);
    configData{i}.perm = configData{i}.poro.^3.*(1e-5)^2./(0.81*72*(1-configData{i}.poro).^2);
end
samples = RockSamples('data', configData);
```

Quantity of interest

- Storing all data from all simulated ensemble members can be very expensive
- Define subsets of the simulation results that we want to estimate



```
%% Set up QoI
% For our QoI, we choose the total oil production rate
is_prod = vertcat(example.schedule.control(1).W.sign) < 0;
qoi = WellQoI('wellIndices', is_prod, 'fldname', 'qOs');
```



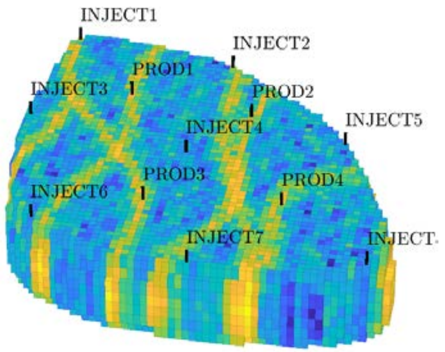
BaseQoI

WellQoI

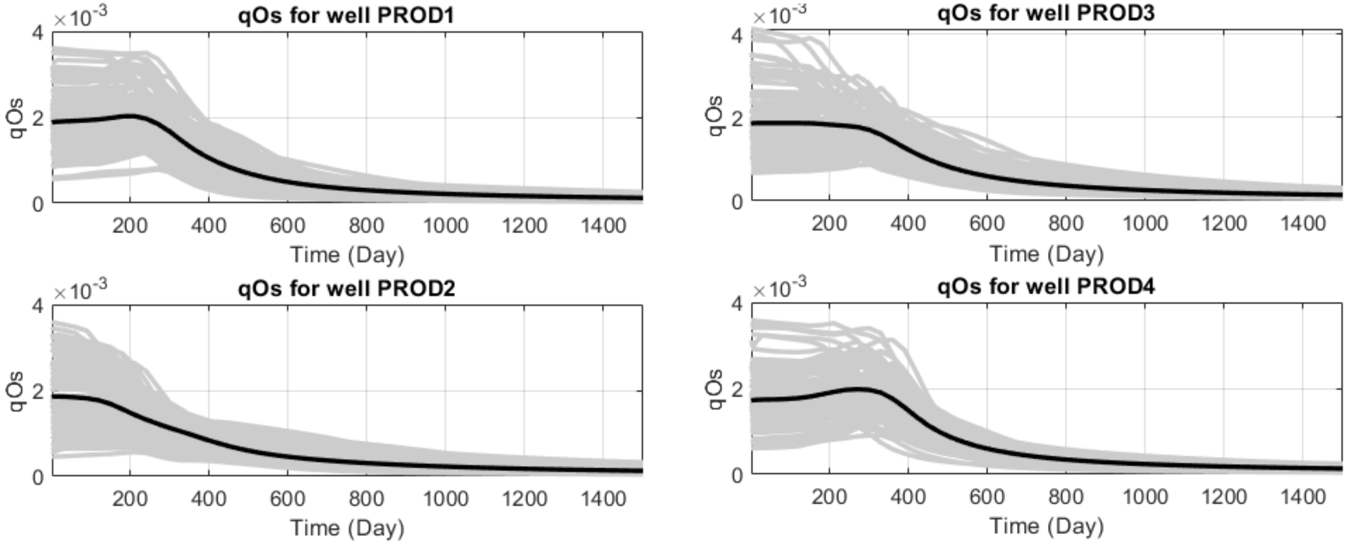
ReservoirStateQoI

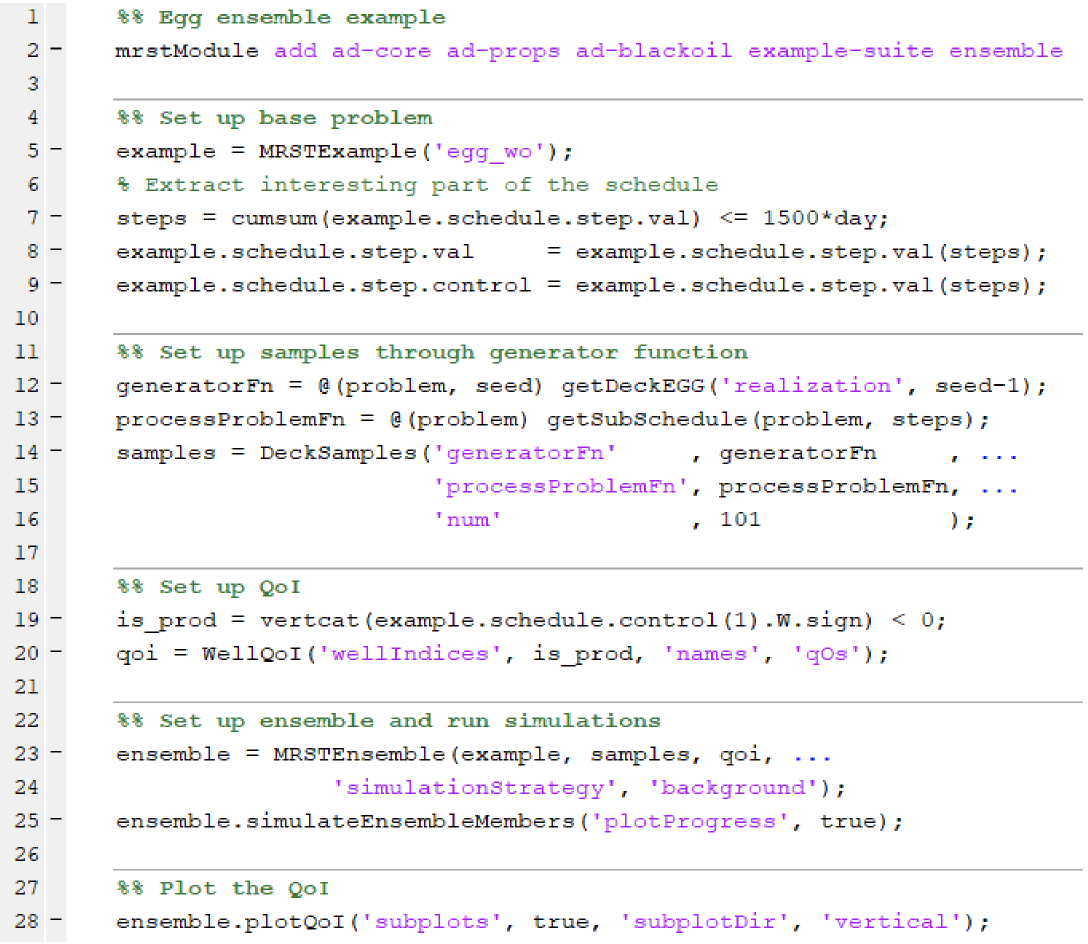
RecoveryFactorQoI

Example: Running the Egg ensemble



- The Egg model [1] is highly channelized water-oil reservoir
- 101 ensemble realizations with different permeability





```
%% Egg ensemble example
mrstModule add ad-core ad-props ad-blackoil example-suite ensemble

%% Set up base problem
example = MRSTExample('egg_wo');
% Extract interesting part of the schedule
steps = cumsum(example.schedule.step.val) <= 1500*day;
example.schedule.step.val     = example.schedule.step.val(steps);
example.schedule.step.control = example.schedule.step.val(steps);

%% Set up samples through generator function
generatorFn = @(problem, seed) getDeckEGG('realization', seed-1);
processProblemFn = @(problem) getSubSchedule(problem, steps);
samples = DeckSamples('generatorFn'     , generatorFn     , ...
                      'processProblemFn', processProblemFn, ...
                      'num'             , 101             );

%% Set up QoI
is_prod = vertcat(example.schedule.control(1).W.sign) < 0;
qoi = WellQoI('wellIndices', is_prod, 'names', 'qOs');

%% Set up ensemble and run simulations
ensemble = MRSTEnsemble(example, samples, qoi, ...
               'simulationStrategy', 'background');
ensemble.simulateEnsembleMembers('plotProgress', true);

%% Plot the QoI
ensemble.plotQoI('subplots', true, 'subplotDir', 'vertical');
```

[1] Jansen, J. D., et al., "The egg model – a geological ensemble for reservoir simulation." Geoscience Data Journal 1.2 (2014): 192-195.

SINTEF

Parallel strategies

Simulation strategies when running ensembles

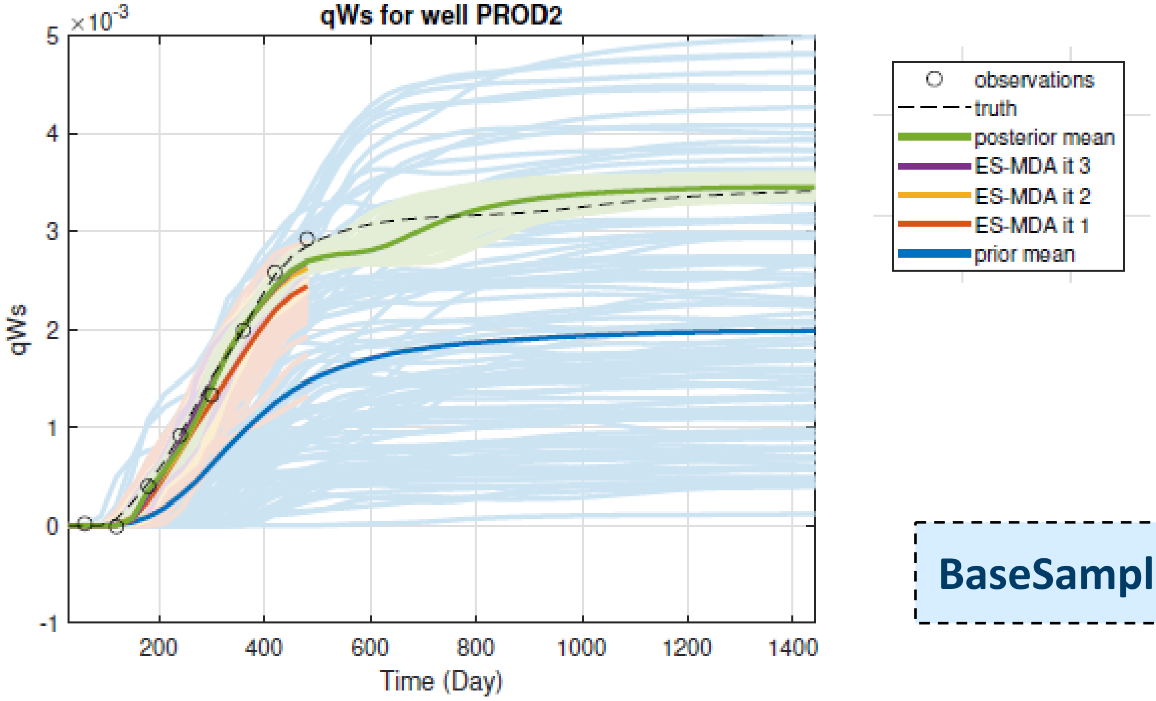
Standard MRST and MATLAB

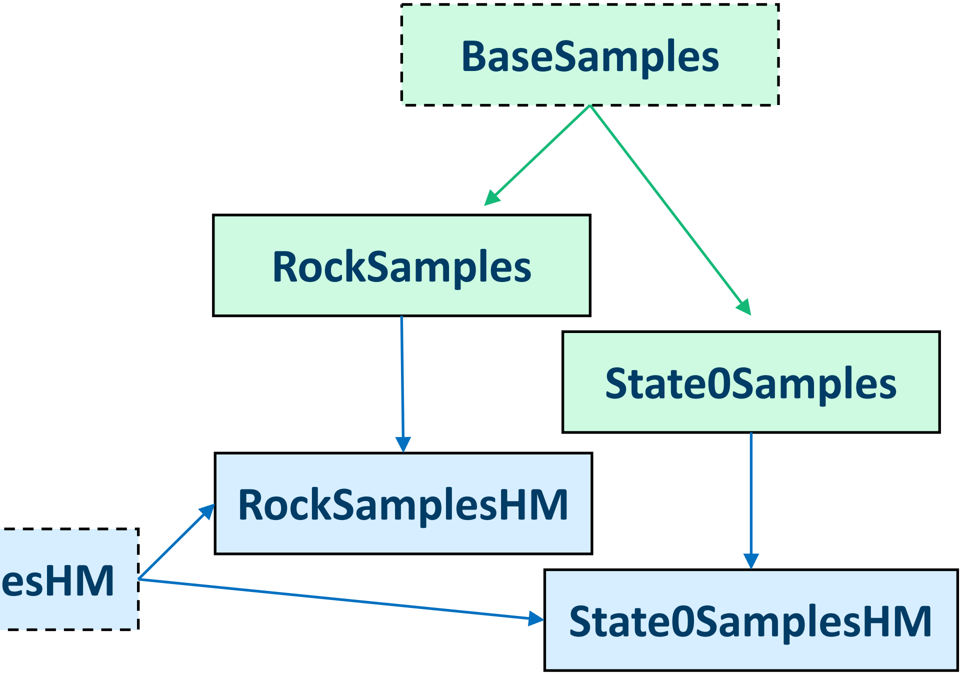
- 'Serial' – No parallelization, good for debugging
- 'background' – Spawn MATLAB sessions in the background, let the OS distribute the available resources on your system
- 'parallel' – MATLAB launches scripts to parallel workers using the 'batch' function from the Parallel Computing Toolbox. Distributed memory parallelization
- 'spmd' – Creates parallel workers that share the ensemble member executions. Shared memory parallelization.

Parallel Computing Toolbox

Module extensions

- Multi-level Monte Carlo
- History matching (HM)
  - Problem description: $p(\text{samples}|\text{qoi}) \propto p(\text{qoi}|\text{samples})p(\text{samples})$
  - Additional HM classes that inherits from the ensemble classes
  - Reuse ensemble functionality but extend with functionality required for ES-MDA


qWs for well PROD2



Technology for a
better society