Software concepts for coupling reduced and detailed simulations

Workshop on Reduced Basis Methods, Ulm

Martin Drohmann, Bernard Haasdonk, Sven Kaulmann, Mario Ohlberger
Motivation

PDE Discretizations (FEM, FV, DG):
Navier-Stokes, Groundwater-Flow, Convection-Diffusion, Poisson, Maxwell, ...

Parametrization

Magical RB-Tools

Reduced model
 Goals Specification

- Abstract Reduced Basis Framework
- Re-use of (existing) efficient implementations of PDE discretizations
- Short time for implementing the detailed simulations as reduced simulations
Abstract Software Concept

Low dimensional computations: Abstract RB - Framework

High dimensional computations

- Linear problems
- Non-linear problems
- ...

PDE Discretizations (FEM, FV, DG):
- Navier-Stokes
- Groundwater-Flow
- Convection-Diffusion
- Poisson
- Maxwell
- ...
Low dimensional Computations

- Control of Offline-Algorithms
  - POD-Greedy Algorithm, Empirical Interpolation Algorithm
  - Gathering and post-processing of reduced matrices

- Low-Dimensional Computations
  - Reduced Simulations
  - A posteriori error estimation

- Low dimensional data visualization

- (Implemented in MATLAB)
High dimensional computations (DUNE-RB)

- Storage / Manipulation of reduced spaces
- Efficient high dimensional linear algebra algorithms
  - POD, Orthonormalization, Gram-Matrix computations
- Parametrization
- Implemented in C++, based on
  - DUNE core modules (http://dune-project.org) and
  - DUNE-FEM (dune.mathematik.uni-freiburg.de)
The Glue

- Communication between DUNE-RB and RBMATLAB can be realized by
  1. compiling DUNE-RB as (mex-) library for matlab or:
  2. TCP/IP Communication between two stand-alone applications
Proof of Concept (linear evolution problem)

```
RBMATLAB > matlab

< MATLAB (R) >
Copyright 1984-2008 The MathWorks, Inc.
Version 7.6.0.324 (R2008a)
February 10, 2008

To get started, type one of these: helpwin, helpdesk, or demo

For product information, visit www.mathworks.com.

starting up rbmatlab in directory:
/home/martin/projects/rbm-results/rbmatlab
Using the following directory for large temporary data:
/tmp
Using the following directory for data files storing results:
/home/martin/projects/rbm-results/results
Using the following directory as RBMATLABHOME:
/home/martin/projects/rbm-results/rbmatlab
skipped clearing filecache for function-calls!

DUNE-RB > ./dunerbServer
YaspGridParameterBlock: Parameter 'overlap' not specified, defaulting to '0'.
server: waiting for connections...
```
Proof of Concept (linear evolution problem)

To get started, type one of these: helpwin, helpdesk, or demo

For product information, visit www.mathworks.com.

Starting up rbmatlab in directory:
/home/martin/projects/rbm-results/rbmatlab
Using the following directory for large temporary data:
/tmp
Using the following directory for data files storing results:
/home/martin/projects/rbm-results/results
Using the following directory as RBMATLABHOME:
/home/martin/projects/rbm-results/rbmatlab
skipped clearing filecache for function-calls!

>> [a,b,c] = ...
mexclient('echo', [1, 2], ...
    struct('field', [1,2]), ...
    [1,2], [3,4]));

client: connect: Connection refused
Warning: connection to ::1 failed

client connected to 127.0.0.1
copying argument no. 0
copying argument no. 1
copying argument no. 2

DUNE-RB > ./dunerrServer
YaspGridParameterBlock: Parameter 'overlap' not specified, defaulting to '0'.
server: waiting for connections...
server: got connection from 127.0.0.1

Received call for processing 'echo'
with 4 arguments and 3 return values.

抄成 argument no. 0
抄成 argument no. 1
抄成 argument no. 2
Proof of Concept (linear evolution problem)

```matlab
/home/martin/projects/rbm-results/rbmatlab
skipped clearing filecache for function-calls!
>> [a,b,c] = ...
    mexclient('echo', [1, 2], ...
        struct('field', [1,2]), ...
        { [1,2], [3,4] });
   client: connect: Connection refused
   Warning: connection to ::1 failed

client connected to 127.0.0.1
copying argument no. 0
copying argument no. 1
copying argument no. 2

>> a
a =
    1   2
>> b
b =
    field: [1 2]
>> c
c =
    [1x2 double]  [1x2 double]
```

```
DUNE-RB > ./dunerbServer
YaspGridParameterBlock: Parameter 'overlap' not specified, defaulting to '0'.
server: waiting for connections...
server: got connection from 127.0.0.1

Received call for processing 'echo'
with 4 arguments and 3 return values.

---

copying argument no. 0
copying argument no. 1
copying argument no. 2
```
Proof of Concept (linear evolution problem)

```
< M A T L A B (R) >
Copyright 1984-2008 The MathWorks, Inc.
Version 7.6.0.324 (R2008a)
February 10, 2008

To get started, type one of these: helpwin, helpdesk, or demo.
For product information, visit www.mathworks.com.

starting up rbmatlab in directory:
/home/martin/projects/rbm-results/rbmatlab
Using the following directory for large temporary data:
/tmp
Using the following directory for data files storing results:
/home/martin/projects/rbm-results/results
Using the following directory as RBMATLABHOME:
/home/martin/projects/rbm-results/rbmatlab
skipped clearing filecache for function-calls!

>> load model parameters
>> model = convdiff dune_model;
Warning: Name is nonexistent or not a directory: mexclient.
> In path at 110
  In addpath at 87
  In convdiff dune_model at 95
client: connect: Connection refused
Warning: connection to ::1 failed

client connected to 127.0.0.1

DUNE-RB > ./dunerrbServer
YaspGridParameterBlock: Parameter 'overlap' not specified, defaulting to '0'.
server: waiting for connections...
server: got connection from 127.0.0.1

Received call for processing 'init model'
with 1 arguments and 1 return values.

read discfunclist.xdr from headerfile, size = 20
Using the explicit ode solver! In order to use a different disc
retization, change the 'DISCRETIZATION' make variable

Received call for processing 'get Mu'
with 1 arguments and 1 return values.

Received call for processing 'rb_symbolic_coefficients'
with 1 arguments and 1 return values.
```

Martin Drohmann (mdrohmann@uni-muenster.de)
Proof of Concept (linear evolution problem)

```matlab
client connected to 127.0.0.1
>> model.rb_problem_type
ans =
    lin_evol

>> model.RB_generation_mode
ans =
    greedy_uniform_fixed

>> model.RB_stop_Nmax
ans =
    20

>> model.T  % this is read from DUNE-RB
ans =
    1
```

```
DUNE-RB > ./dunerrbServer
YaspGridParameterBlock: Parameter 'overlap' not specified, defaulting to '0'.
server: waiting for connections...
server: got connection from 127.0.0.1

Received call for processing 'init model'
with 1 arguments and 1 return values.

read discfucnlist.xdr from headerfile, size = 20
Using the explicit ode solver! In order to use a different discretization, change the 'DISCRETIZATION' make variable

Received call for processing 'get_mu'
with 1 arguments and 1 return values.

Received call for processing 'rb_symbolic_coefficients'
with 1 arguments and 1 return values.
```
Proof of Concept (linear evolution problem)

ans =
lin_ev

>> model.RB_generation_mode
ans =
  greedy_uniform_fixed

>> model.RB_stop_Nmax
ans =
    20

>> model.T % this is read from DUNE-RB
ans =
    1

>>
>>

>> % generate high dimensional model specific data, like e.g.
>> % the grid
>> model_data = gen_model_data(model);

>>
Proof of Concept (linear evolution problem)

```matlab
>> % Just for fun: Do a DETAILED simulation in DUNE-RB >>>>
>>
>> % first set the parameter mu
>> model = set_mu(model, [0.0 0.5 1.0]);
>>
>> % then run the simulation
>> tic; sim data = detailed_simulation(model, model_data); toc
Elapsed time is 11.795319 seconds.
```

```matlab
Received call for processing 'rb_symbolic_coefficients'
with 1 arguments and 1 return values.

---------------------------------------------------------------

Received call for processing 'gen_model_data'
with 1 arguments and 1 return values.

---------------------------------------------------------------

Received call for processing 'set_mu'
with 2 arguments and 0 return values.

---------------------------------------------------------------

Received call for processing 'detailed_simulation'
with 1 arguments and 1 return values.

---------------------------------------------------------------

opening file: ./grape//solution.series

Received call for processing 'set mu'
with 2 arguments and 0 return values.

---------------------------------------------------------------

opening file: ./grape//solution.series
```
Proof of Concept (linear evolution problem)

```matlab
>> % Just for fun: Do a DETAILED simulation in DUNE-RB >>>>>>
>>
>> % first set the parameter mu
>> model = set_mu(model, [0.0 0.5 1.0]);
>>
>> % then run the simulation
>> tic; sim data = detailed_simulation(model, model_data); toc
Elapsed time is 11.795319 seconds.
```

11.8 seconds!
Proof of Concept (linear evolution problem)

```plaintext
>> % Generate the reduced basis with the POD-Greedy algorithm
>> % in DUNE-RB >>>>>>>>
>>
>> detailed_data = gen_detailed_data(model, model_data);

Starting RB extension loop

Detected maximum error prediction 0.044006 for mu=[0.001 1 0.5]
Extended RB to length 2

Detected maximum error prediction 0.015456 for mu=[0 1 0.5]
Extended RB to length 3

Detected maximum error prediction 0.012877 for mu=[0 1 0.5]
Extended RB to length 4

Detected maximum error prediction 0.01064 for mu=[0 1 0.5]
Extended RB to length 5

Detected maximum error prediction 0.0084073 for mu=[0.001 0.5 1]
Extended RB to length 6

Detected maximum error prediction 0.0073233 for mu=[0 1 1]
Extended RB to length 7

Detected maximum error prediction 0.0055443 for mu=[0 1 1]
Extended RB to length 8

Detected maximum error prediction 0.0048443 for mu=[0 1 0.5]
```
Proof of Concept (linear evolution problem)

```matlab
>> >> detailed_data.RB_info
ans =

    M train: [3x64 double]
          max_err_sequence: [1x20 double]
          mu_sequence: [3x20 double]
          mu_ind_seq: [1x20 double]
       toc_value_sequence: [1x20 double]
          M_first errs: [64x1 double]
    stopped_on epsilon: 0
    stopped_on max val train ratio: 0
    stopped on timeout: 0
    stopped on Nmax: 1
    stopped on empty extension: 0
    stopped on Nlimit estimation: 0
          M last errs: [64x1 double]
       elapsed_time: 3.0000e-06
```

Opening file: ./grape//solution.series
Proof of Concept (linear evolution problem)

```matlab
>> % NOW: Generate reduced matrices for online computations
>> % and get them to RBMATLAB

% reduced_data = gen_reduced_data(model, detailed_data)

reduced_data =
    a0: {1x20 double}
    LL_I: {2x1 cell}
    LL_E: {5x1 cell}
    bb: {4x1 cell}
    K_II: {4x1 cell}
    K_IE: {10x1 cell}
    K_EE: {25x1 cell}
    m_I: {8x1 cell}
    m_E: {20x1 cell}
    m: {16x1 cell}
    N: 20
```

Received call for processing 'rb init_values' with 2 arguments and 1 return values.

Received call for processing 'rb operators' with 2 arguments and 1 return values.

Received call for processing 'set_mu' with 2 arguments and 0 return values.

Received call for processing 'reconstruct_and_compare' with 2 arguments and 0 return values.

opening file: ./grape//solution.series

Received call for processing 'rb init_values' with 2 arguments and 1 return values.

Received call for processing 'rb operators' with 2 arguments and 1 return values.
Proof of Concept (linear evolution problem)

```matlab
>> % The matrices are all of small sizes: e.g. the explicit
>> % discretization operator: LL_E
>>
>> sizes=cellfun(@(X) size(X), reduced_data.LL_E, ...  
    'UniformOutput', false);
>> sizes{;}

ans =
    20  20

ans =
    20  20

ans =
    20  20

ans =
    20  20

>>
```

Received call for processing 'rb_init_values'
with 2 arguments and 1 return values.

Received call for processing 'rb_operators'
with 2 arguments and 1 return values.

Received call for processing 'set_mu'
with 2 arguments and 0 return values.

Received call for processing 'reconstruct_and_compare'
with 2 arguments and 0 return values.

opening file: ./grape//solution.series

Received call for processing 'rb_init_values'
with 2 arguments and 1 return values.

Received call for processing 'rb_operators'
with 2 arguments and 1 return values.
Proof of Concept (linear evolution problem)

% Now fast reduced simulations are possible in RBMATLAB
% without any communication to DUNE-RB
model = model.set_mu(model, [0 0.5 1], true);
tic; rb_sim_data = rb_simulation(model, reduced_data); toc
Elapsed time is 0.020223 seconds.

rb_sim_data =
   a: [20x113 double]
   Delta: [1x113 double]
   LL_I: [20x20 double]
   LL_E: [20x20 double]

Error estimator at end time:
rb_sim_data.Delta(end)
ans =
   0.0019

Received call for processing 'rb_init_values'
with 1 arguments and 1 return values.

Received call for processing 'rb_operators'
with 2 arguments and 1 return values.

Received call for processing 'set_mu'
with 2 arguments and 0 return values.

opening file: ./grape/solution.series
Received call for processing 'reconstruct_and_compare'
with 2 arguments and 0 return values.
Received call for processing 'rb_init_values'
with 2 arguments and 1 return values.
Proof of Concept (linear evolution problem)

```matlab
>> % Of course, the solution can be reconstructed in
>> % DUNE-RB >>>>>>>

```

Received call for processing 'reconstruct_and_compare'
with 2 arguments and 0 return values.

opening file: ./grape/solution.series

Received call for processing 'rb init values'

```
# Linear Transport Problem

<table>
<thead>
<tr>
<th></th>
<th>High-dim. Solution (sec)</th>
<th>RB Generation</th>
<th>Gen. of Online Matrices</th>
<th>Reduced Sim.</th>
<th>Reconstruction</th>
<th>Grid Cells</th>
<th>$L^\infty - L^2$-Error$^1$</th>
</tr>
</thead>
<tbody>
<tr>
<td>2D Transport</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(25 Base Functions)</td>
<td>11</td>
<td>71</td>
<td>6.69</td>
<td>0.11</td>
<td>0.33</td>
<td>1,024</td>
<td>$1.42 \cdot 10^{-3}$</td>
</tr>
<tr>
<td>2D Transport</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(50 Base Functions)</td>
<td>11</td>
<td>2,250</td>
<td>21</td>
<td>0.15</td>
<td>0.42</td>
<td>1,024</td>
<td>$4.64 \cdot 10^{-4}$</td>
</tr>
<tr>
<td>3D Transport</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>(50 Base Functions)</td>
<td>944</td>
<td>$1.57 \cdot 10^5$</td>
<td>4,659</td>
<td>0.15</td>
<td>26</td>
<td>32,768</td>
<td>$9.11 \cdot 10^{-4}$</td>
</tr>
</tbody>
</table>

Table 1: Numerical results for a transport problem in 2D and 3D with non-divergence-free velocity.
So, we have a hammer for linear evolution problems ...
... and even one for non-linear problems ...
But how do you make your problems look like a nail?

- Poisson
- Navier-Stokes
- Groundwater-Flow
Interface to PDE discretizations

```plaintext
>> Generate the reduced basis with the POD-Greedy algorithm
>> in DUNE-RB >>>>>>>>
>>
>> detailed data = gen detailed data(model, model_data);

Starting RB extension loop

Detected maximum error prediction 0.044006 for mu=[0.001
    1   0.5]
Extended RB to length 2

Detected maximum error prediction 0.015456 for mu=[0 1 1]
Extended RB to length 3

Detected maximum error prediction 0.012877 for mu=[0 1 0.5]
Extended RB to length 4

Detected maximum error prediction 0.01064 for mu=[0 1 0.5]
Extended RB to length 5

Detected maximum error prediction 0.0084073 for mu=[0.001
    0.5 1]
Extended RB to length 6

Detected maximum error prediction 0.0073233 for mu=[0 1 1]
Extended RB to length 7

Detected maximum error prediction 0.0055443 for mu=[0 1 1]
Extended RB to length 8

Detected maximum error prediction 0.0048443 for mu=[0 1 0.5]
```

Received call for processing 'rb operators'
with 2 arguments and 1 return values.

Received call for processing 'get_mu'
with 1 arguments and 1 return values.

Received call for processing 'set_mu'
with 2 arguments and 0 return values.

Received call for processing 'rb_extension_PCA'
with 3 arguments and 1 return values.

opening file: ./grape//solution.series

Received call for processing 'set_mu'
with 2 arguments and 0 return values.

Received call for processing 'rb_init_values'
with 2 arguments and 1 return values.
Interface to PDE discretizations (linear)

- For PDE discretization of form

  For \( \mu \in \mathcal{P} \) compute \( \{u_h^k(\mu)\}_{k=0}^{K} \subset \mathcal{V}_h \subset L^2(\Omega) \) by

  \[
  u_h^0(\mu) := P[u_0(\mu)] \quad , \quad u_h^k(\mu) := u_h^{k-1}(\mu) + \Delta t \mathcal{L}^E(\mu)[u_h^{k-1}(\mu)].
  \]

- Implement affinely parameter dependent operators

  \[
  \mathcal{L} = \sum_{q=1}^{Q_L} \sigma^q_L(\mu) \, \mathcal{L}^q
  \]
Interface to PDE discretizations (linear)

Affinely parametrized operators

\[ \mathcal{L} = \sum_{q=1}^{Q_L} \sigma^q_L(\mu) \mathcal{L}^q \]

with methods

- Parametrization [set\_mu()]
- Parameter dependent coefficients [coefficient()]
- Parameter independent operators [component()]
Interface to PDE discretizations (linear)

Affinely parametrized operators

\[ \mathcal{L} = \sum_{q=1}^{Q_L} \sigma^q_L(\mu) \mathcal{L}^q \]

with methods

- Parametrization \([\text{set\_mu}()]\)
- Parameter dependent coefficients \([\text{coefficient}()]\)
- Parameter independent operators \([\text{component}()]\)

\[ (L^q)_{n,m} = \int_{\Omega} L^q [\varphi_m] \varphi_n, \quad q = 1, \ldots, Q_L, m, n = 1, \ldots, N \]
Interface to PDE discretizations (non-linear)

- For PDE discretizations of the form

For $\mu \in \mathcal{P}$ find $\{u_h^k\}_{k=0}^K \subset \mathcal{V}_h \subset L^2(\Omega)$, s.t.

$$u_h^0 := P_h [u_0(\mu)], \quad u_h^{k+1} := u_h^{k+1, \nu_{\text{max}}(k)}$$

with Newton iteration

$$u_{h}^{k+1,0} := u_{h}^{k}, \quad u_{h}^{k+1,\nu+1} := u_{h}^{k+1,\nu} + \delta_{h}^{k+1,\nu+1},$$

$$\left( \mathbf{I} + \Delta t \mathbf{D} \mathcal{L}_h^l \big|_{u_h^{k+1,\nu}} \right) \left[ \delta_{h}^{k+1,\nu+1} \right] = u_{h}^{k} - u_{h}^{k+1,\nu} - \Delta t \left( \mathcal{L}_h^l \left[ u_{h}^{k+1,\nu} \right] + \mathcal{L}_h^E \left[ u_{h}^{k} \right] \right).$$

- Implement empirical interpolation for discrete operators

$$\mathcal{L}_h(\mu) \left[ u_h^k(\mu) \right] \approx \mathcal{I}_M \left[ \mathcal{L}_h(\mu) \right] \left[ u_h^k(\mu) \right] = \sum_{m=1}^M l_m(\mu) \left[ u_h^k(\mu) \right] \xi_m$$
The coefficient functionals $l_m(\mu) = \mathcal{L}_h(\mu)[\cdot](x_m)$ can be computed efficiently, if

- Operator has localized structure (small stencil) and
- Local geometry information can be precomputed in offline phase.
Empirical Interpolation for Operators (local grid)
Automatic generation of localized operators

Usual dependencies for local operator evaluations:

\[ l_m(\mu) = L_h(\mu)[\cdot](x_m) \]

- Basis function evaluations at quadrature points
- (Local) geometry information of the grid
- ...
Automatic generation of localized operators

Therefore: Delegation of function calls concerning grid geometry and discrete function space to a wrapper:

\[ l_m(\mu) = L_h(\mu)[\cdot](x_m) \]

DUNE-RB Grid Wrapper

- Basis function evaluations at quadrature points
- (Local) geometry information of the grid
- ...
Behaviour of DUNE-RB Grid Wrapper

- During detailed simulation:
  Delegate calls directly to the grid and the discrete function space

- During offline phase:
  store all grid and discrete function space information concerning the subgrid in low-dimensional data structures

- During online phase:
  Delegate calls to low-dimensional data structures
Conclusion

- Generic implementation of reduced base tools for linear problems is possible
- Empirical interpolation for non-linear problems can be realized with subgrid extraction

Future work:
- More examples
- Tarball – Release with installation instructions is planned, and will be published on http://morepas.org.
Thank you for your attention!