TU Munich, 16-20 September 2013, RB Summer School

rbMIT software library: Heat Transfer Examples

Andrea Manzoni
Gianluigi Rozza

release developed at MIT, TLO 12600
(A.T. Patera, D.B.P. Huynh, C.N. Nguyen)

Overview and Methodology

Overview

- Certified Reduced Basis method and associated software package rbMIT [†]
  - Problem Formulation and Reduced Basis approximation
- Examples of steady and unsteady conduction worked problems
  - a thermal fin
  - a thermal analysis of a delamination crack

rbMIT Methodology

- Input parameter (problem data) and desired outputs (thermal quantities)
- Computational stages
  - Offline ("Instructor" level)
  - Online ("Lecturer/Student" level): rapid and reliable prediction of outputs and rigorous error bounds

[†] available for educational and academic use at http://augustine.mit.edu

Motivation

Heat Transfer Education:

- classical approaches such as **finite element method** are often too **slow** and low order heuristic approaches are often unreliable

Goal: to achieve the **accuracy** and **reliability** of a high fidelity approximation but at greatly **reduced cost** of a **low order model**

Pedagogical prospects:

- interactive in-class visualization and parametric exploration
- rapid assessment of classical engineering approximations and interpretations
- more realistic examples in homework assignment and design projects
- collection/catalogue of many worked problems, available on line *

Way: certified Reduced Basis Method for **rapid** and **reliable** prediction of engineering outputs associated with parametrized PDEs

* `http://augustine.mit.edu`

Motivation

Heat Transfer Education:

- classical approaches such as **finite element method** are often too **slow** and low order heuristic approaches are often unreliable

Goal: to achieve the **accuracy** and **reliability** of a high fidelity approximation but at greatly **reduced cost** of a **low order model**

Pedagogical prospects:

- interactive in-class visualization and parametric exploration
- rapid assessment of classical engineering approximations and interpretations
- more realistic examples in homework assignment and design projects
- collection/catalogue of many worked problems, available on line *

rapid = minimiziation of the marginal cost in I/O evaluation

reliable = error bounds of input/output evaluation or field variable

useful in real-time/interactive or many queries context such as robust parameter estimation, design, optimization and control

* http://augustine.mit.edu

Motivation

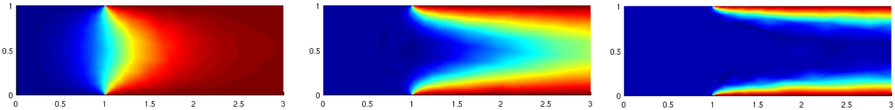
Input and Output

- Input parameter: $\mu \in D \subset \mathbb{R}^p \rightarrow$ geometry, material prop., BCs, sources, ...
- Output of interest: $s(\mu) = \ell(u(\mu)) \rightarrow$ related to temperature or fluxes
- Field variable: temperature $u(\mu) \rightarrow$ satisfies a μ-parametrized PDE

Motivation

Input and Output

- Input parameter: $\mu \in D \subset \mathbb{R}^p \rightarrow$ geometry, material prop., BCs, sources, ...
- Output of interest: $s(\mu) = \ell(u(\mu)) \rightarrow$ related to temperature or fluxes
- Field variable: temperature $u(\mu) \rightarrow$ satisfies a μ-parametrized PDE



- Rapidly convergent global reduced basis (RB) approximations (Galerkin projection onto a space spanned by solution of governing PDE at N selected $\mu^1, \dots, \mu^N$)
- Rigorous a posteriori error estimation procedures (inexpensive yet sharp bounds for the error in the RB field-variable and output approximations)
- Offline/Online computational procedures (very extensive and parameter independent Offline stage / inexpensive Online calculations for new I/O evaluation)

The "Game"

- $(\ )^{\mathcal{N}}$: "truth" finite element – to be accelerated
- $(\ )_N$: reduced basis – *the accelerator*

⋆ Input parameter: μ (geometry, physical properties, …)

⋆ Output: $$s(t;\mu) \approx \underbrace{s^{\mathcal{N}}(t;\mu)}_{\text{finite element}} \approx \underbrace{s_N(t;\mu)}_{\text{reduced basis}}$$

⋆ Input-Output evaluation: $\mu \rightarrow s^{\mathcal{N}}(t;\mu) \rightarrow s_N(t;\mu)$

- **Offline**: very expensive pre-processing
- **Online**: extremely fast (reduced basis) input-output valuation

$$\mu \quad \rightarrow \quad \underbrace{s_N(t;\mu)}_{\text{reduced basis output}} \quad \rightarrow \quad \underbrace{\Delta_N^s(t;\mu)}_{\text{reduced basis error bound}}$$

such that (certification)

$$\underbrace{s^{\mathcal{N}}(t;\mu)}_{\text{"truth" finite element output}} \in \underbrace{[s_N(t;\mu)-\Delta_N^s(t;\mu), s_N(t;\mu)+\Delta_N^s(t;\mu)]}_{\text{reduced basis "error bar"}}$$

Heat Conduction: Problem Formulation

Steady Heat Conduction

Given $\mu \in \mathscr{D} \subset \mathbb{R}^P$, we evaluate

$$s_o(\mu) = \int_{B_{oL}} u_o(\mu)$$

where $u_o(\mu)$ satisfies

$$\begin{cases} -\dfrac{\partial}{\partial x_{o\,i}}\left(\kappa_{o\,ij}\dfrac{\partial u_o}{\partial x_{o\,j}}\right) = f_o & \text{in } \Omega_o(\mu) \\ +\text{BCs} & \text{on } \partial\Omega_o(\mu) \end{cases}$$

Unsteady Heat Conduction

Given $\mu \in \mathscr{D} \subset \mathbb{R}^P$, we evaluate

$$s_o(\mu) = \int_0^{t_f}\left(h(t)\int_{B_L} u_o(t;\mu)\right)dt$$

where $u_o(t;\mu)$ satisfies for $t \in [0, t_f]$

$$\begin{cases} \dfrac{\partial u_o}{\partial t} - \dfrac{\partial}{\partial x_{o\,i}}\left(\kappa_{o\,ij}\dfrac{\partial u_o}{\partial x_{o\,j}}\right) = g(t) f_o & \text{in } \Omega_o(\mu) \\ u_o(t=0;\mu) = u^0 & \text{in } \Omega_o(\mu) \\ +\text{BCs} & \text{on } \partial\Omega_o(\mu) \end{cases}$$

- The μ–dependent problem has to be formulated on a μ–independent reference domain Ω
- A domain decomposition of $\Omega_o(\mu)$ and proper piecewise-affine mappings are automatically built by rbMIT©
- The problem is then reduced to a parametric PDE on reference domain: geometric variations are now captured by the coefficients of the equation

Heat Conduction: Finite Element Discretization

FE discretization (Steady case)

Given $\mu \in \mathscr{D} \subset \mathbb{R}^P$, we evaluate

$$s^{\mathcal{N}}(\mu) = \{\mathbf{L}^{\mathcal{N}}(\mu)\}^T \{\mathbf{u}^{\mathcal{N}}(\mu)\}$$

where $\{\mathbf{u}^{\mathcal{N}}\} \in \mathbb{R}^{\mathcal{N}}$ satisfies

$$[\mathbf{K}^{\mathcal{N}}(\mu)]\{\mathbf{u}^{\mathcal{N}}(\mu)\} = \{\mathbf{F}^{\mathcal{N}}(\mu)\}$$

Semi-discrete FE discretization (Unsteady case)

Given $\mu \in \mathscr{D} \subset \mathbb{R}^P$, we evaluate

$$s^{\mathcal{N}}(\mu) = \int_0^{t_f} \left(h(t)\{\mathbf{L}^{\mathcal{N}}\}^T \{\mathbf{u}^{\mathcal{N}}(t;\mu)\} \right) dt$$

where $\mathbf{u}^{\mathcal{N}}(t;\mu) \in \mathbb{R}^{\mathcal{N}}$ satisfies for $t \in (0, t_f]$

$$[\mathbf{M}^{\mathcal{N}}(\mu)]\{\dot{\mathbf{u}}^{\mathcal{N}}(t;\mu)\} + [\mathbf{K}^{\mathcal{N}}(\mu)]\{\mathbf{u}^{\mathcal{N}}(t;\mu)\} = g(t)\{\mathbf{F}^{\mathcal{N}}\}$$

with $\mathbf{u}^{\mathcal{N}}(t=0;\mu) = \mathbf{u}_0^{\mathcal{N}}$

- The dimension of the FE approximation $\mathcal{N}$ is sufficiently large so that the FE output $s^{\mathcal{N}}(\mu)$ is indistinguishable from the exact output $s(\mu)$ at the accuracy level of interest
- The matrix $[\mathbf{K}^{\mathcal{N}}(\mu)]$ is "affine" in the parameter μ, by which we mean

  $$[\mathbf{K}^{\mathcal{N}}(\mu)] = \sum_{q=1}^{Q} \Theta_q(\mu)[\mathbf{K}_q^{\mathcal{N}}]$$

  where for $q = 1, \ldots, Q$, the $\Theta_q : \mathscr{D} \to \mathbb{R}$ are (typically very smooth) μ-dependent functions, and the $[\mathbf{K}_q^{\mathcal{N}}]$ are μ-independent matrices
- The affine-parameter decomposition is crucial to the computational performance of the Offline-Online procedure (but it may be relaxed)

Heat Conduction: Reduced Basis Approximation

RB formulation (Steady case)

Given $\mu \in \mathscr{D}$, we evaluate

$$s_N(\mu) = \{\mathbf{F}_N\}^T \{\mathbf{u}_N(\mu)\}$$

where $\{\mathbf{u}_N(\mu)\}$ satisfies

$$[\mathbf{K}_N(\mu)]\{\mathbf{u}_N(\mu)\} = \{\mathbf{F}_N\}$$

RB formulation (Unsteady case)

Given $\mu \in \mathscr{D}$, we evaluate

$$s_N(\mu) = \int_0^{t_f} \left(h(t)\{\mathbf{L}_N\}^T \{\mathbf{u}_N(t;\mu)\} \right) dt$$

where $\{\mathbf{u}_N(t;\mu)\}$ satisfies

$$\sum_{j=1}^{J} \Phi_j(\mu)[\mathbf{M}_{j\,N}]\{\dot{\mathbf{u}}_N(t;\mu)\} + \sum_{q=1}^{Q} \Theta_q(\mu)[\mathbf{K}_{q\,N}]\{\mathbf{u}_N(t;\mu)\} = g(t)\{\mathbf{F}_N\}$$

- Snapshot FEM solutions with $\mu \in S_N = \{\mu^1, \dots, \mu^N\}, 1 \le N \le N_{\max}$ span a subspace

$$W_N^{\mathcal{N}} = \mathrm{span}\{u^{\mathcal{N}}(\mu^n), 1 \le n \le N\} = \mathrm{span}\{\zeta_n^{\mathcal{N}}, 1 \le n \le N\}$$

- Reduced Basis formulaion is obtained by a Galerkin projection on $W_N^{\mathcal{N}}$
- $[\mathbf{Z}_N] \equiv [\mathbf{Z}_N^{\mathcal{N}}] = [\{\zeta_1^{\mathcal{N}}\}|\cdots|\{\zeta_N^{\mathcal{N}}\}]$ is the orthonormalized–snapshot $\mathcal{N} \times N$ matrix
- The following affine representations for stiffness and mass matrices is used:

$$\{\mathbf{L}_N\} = [\mathbf{Z}_N]^T\{\mathbf{L}^{\mathcal{N}}\}, \qquad \{\mathbf{F}_N\} = [\mathbf{Z}_N]^T\{\mathbf{F}^{\mathcal{N}}\}, \qquad [\mathbf{K}_N(\mu)] = [\mathbf{Z}_N]^T[\mathbf{K}^{\mathcal{N}}(\mu)][\mathbf{Z}_N]$$

$$[\mathbf{K}_{q\,N}] = [\mathbf{Z}_N]^T[\mathbf{K}_q^{\mathcal{N}}][\mathbf{Z}_N], 1 \le q \le Q, \qquad [\mathbf{M}_{j\,N}] = [\mathbf{Z}_N]^T[\mathbf{M}_j^{\mathcal{N}}][\mathbf{Z}_N], 1 \le j \le J$$

Heat Conduction: RB Error Estimation

A posteriori error estimator is a certificate of fidelity that rigorously bounds the error in the RB prediction relative to the highly accurate truth finite element solution

Steady case

$$|s^{\mathcal{N}}(\mu) - s_N(\mu)| \leq \Delta_N^s(\mu) = \varepsilon^2(\mu)/\alpha_{\mathrm{LB}}^{\mathcal{N}}(\mu)$$

- $\varepsilon^2(\mu) = \{\mathbf{R}^{\mathcal{N}}\}^T[\mathbf{Y}^{\mathcal{N}}]^{-1}\{\mathbf{R}^{\mathcal{N}}\}$ is the square of the dual norm of the residual vector
$$\{\mathbf{R}^{\mathcal{N}}\} = \{\mathbf{F}^{\mathcal{N}}\} - [\mathbf{K}^{\mathcal{N}}(\mu)][\mathbf{Z}_N]\{\mathbf{u}_N(\mu)\}$$
- $[\mathbf{Y}^{\mathcal{N}}] = [\mathbf{K}^{\mathcal{N}}(\overline{\mu})]$ for some $\overline{\mu} \in \mathscr{D}$
- $\alpha_{\mathrm{LB}}^{\mathcal{N}}(\mu)$ is a lower bound for the discrete coercivity constant (SCM method)

Unsteady case

$$|s^{\mathcal{N}}(t,\mu) - s_N(t,\mu)| \leq \Delta_N^s(t,\mu) = \frac{\sigma_0}{\alpha_{\mathrm{LB}}^{\mathcal{N}}(\mu)}\left(\left(\int_0^{t_f} h^2(t)dt\right)\left(\int_0^{t_f} \varepsilon^2(t;\mu)dt\right)\right)^{1/2}$$

- $\sigma_0^2 = \{\mathbf{L}^{\mathcal{N}}\}^T[\mathbf{Y}^{\mathcal{N}}]^{-1}\{\mathbf{L}^{\mathcal{N}}\}$ = square of the dual norm of the output vector $\mathbf{L}^{\mathcal{N}}$
- $\varepsilon^2(t;\mu) = \{\mathbf{R}^{\mathcal{N}}\}^T[\mathbf{Y}^{\mathcal{N}}]^{-1}\{\mathbf{R}^{\mathcal{N}}\}$ = square of the dual norm of the residual vector

The rbMIT software package

- The rbMIT© software package implements in Matlab® all general RB algorithms
- The user must describe the problem. The input can be separated into three parts:

The User Input

- **geometry**: $\Omega_o(\mu)$ is defined by providing points coordinates, straight/curvy edges describing all regionsand regions $\Omega_o^k(\mu)$
- **material properties**: coefficients are provided for differential operator in each region $\Omega_o^k(\mu)$ and for boundary conditions.
- **parameter control and settings**: parameter domain $\mathscr{D}$, reference parameters and other RB information (e.g. $N_{\max}$)

The rbMIT software package

- The rbMIT© software package implements in Matlab® all general RB algorithms
- The user must describe the problem. The input can be separated into three parts:

The User Input

- **geometry**: $\Omega_o(\mu)$ is defined by providing points coordinates, straight/curvy edges describing all regionsand regions $\Omega_o^k(\mu)$
- **material properties**: coefficients are provided for differential operator in each region $\Omega_o^k(\mu)$ and for boundary conditions.
- **parameter control and settings**: parameter domain $\mathscr{D}$, reference parameters and other RB information (e.g. $N_{\max}$)

- The rbMIT© Software architecture can be divided into three steps:
  - ⋆ the Problem Formulation Step ("Instructor/Lecturer" level)
  - ⋆ the RB Offline Step ("Instructor" level)
  - ⋆ the RB Online Step ("Student" level)

Problem Formulation, Offline and Online Steps

The Problem Formulation Step

- Domain Decomposition and geometric transformations are built: (coupled with material input properties) coefficients $\Theta_q(\mu)$ are generated for each sub-domain
- A FE mesh is generated and discrete FE stiffness matrices and vectors are assembled for each sub-domain (and then combined) to form the μ-independent components

Problem Formulation, Offline and Online Steps

The Problem Formulation Step

- Domain Decomposition and geometric transformations are built: (coupled with material input properties) coefficients $\Theta_q(\mu)$ are generated for each sub-domain
- A FE mesh is generated and discrete FE stiffness matrices and vectors are assembled for each sub-domain (and then combined) to form the μ-independent components

The RB Offline Step

- RB parameter sample set $S_{N_{\max}}$ and $[\mathbf{Z}_{N_{\max}}]$ are obtained (greedy algorithm)
- $\{\mathbf{F}_{N_{\max}}\}$, $[\mathbf{K}_{q\,N_{\max}}]$ are saved into a "Online Database" to be used Online

Problem Formulation, Offline and Online Steps

The Problem Formulation Step

- Domain Decomposition and geometric transformations are built: (coupled with material input properties) coefficients $\Theta_q(\mu)$ are generated for each sub-domain
- A FE mesh is generated and discrete FE stiffness matrices and vectors are assembled for each sub-domain (and then combined) to form the μ-independent components

The RB Offline Step

- RB parameter sample set $S_{N_{\max}}$ and $[\mathbf{Z}_{N_{\max}}]$ are obtained (greedy algorithm)
- $\{\mathbf{F}_{N_{\max}}\}$, $[\mathbf{K}_{q\,N_{\max}}]$ are saved into a "Online Database" to be used Online

The RB Online Step

- Given $\mu \in \mathscr{D}$, the RB Online Evaluator returns output prediction and error bound

  `Online_RB (probname,` μ`, outputname, ...)`: $\mu \rightarrow s_N^{\mathcal{N}}(\mu), \Delta_N^s(\mu)$
- The RB Visualizer renders the relevant field variable and provides the error bound

  `Vis_RB (probname,` μ`)`: $\mu \rightarrow \Omega,\ u_N^{\mathcal{N}}(x;\mu)$ for all x in $\Omega_{\mathbf{o}}(\mu)$

rbMIT Users' Interface

Example of rbUfile for User Problem Formulation

```
% -----        rbMIT  Software Copyright MIT 2006-09       -----
% --------- DBP Huynh, NC Nguyen, AT Patera, G Rozza ---------

probname = 'Tfin';

points = '[0,0; 1/2,0; 1/2,3/5; 3/20,3/5; 0,3/5; 3/20,3/5+mu2/2;
           3/20,3/5+mu2; 0,3/5+mu2; 0,3/5+mu2/2]';

edge = [1,2;2,3;3,4;4,5;4,6;6,7;7,8;8,9;9,5;5,1];

geometry{1} = [1,2,3,4,10];
geometry{2} = [4,5,6,7,8,9];
gflag = [1,1];

muref  = [.1,4,1];
mu_min = [.01,2,1];
mu_max = [0.5,8,10];
mu_bar = [.1,4,1];

kappa{1} = '[mu3, 0, 0; 0, mu3, 0; 0, 0, 0]' ;
kappa{2} = '[1, 0, 0; 0, 1, 0; 0, 0, 0]' ;

dirichlet = '[1,0; 2,1; 4,0]' ;
nload     = '[3,0,0,1; 5, mu1,0,0;6,mu1,0,0]' ;

outputname = 'basetemp';
oload='[1,2]';
```

rbMIT Users' Interface

Example of rbUfile for User Problem Formulation

- **Geometry**

```
% -----        rbMIT  Software Copyright MIT 2006-09       -----
% --------- DBP Huynh, NC Nguyen, AT Patera, G Rozza ---------

probname = 'Tfin';

points = '[0,0; 1/2,0; 1/2,3/5; 3/20,3/5; 0,3/5; 3/20,3/5+mu2/2;
           3/20,3/5+mu2; 0,3/5+mu2; 0,3/5+mu2/2]';

edge = [1,2;2,3;3,4;4,5;4,6;6,7;7,8;8,9;9,5;5,1];

geometry{1} = [1,2,3,4,10];
geometry{2} = [4,5,6,7,8,9];
gflag = [1,1];

muref  = [.1,4,1];
mu_min = [.01,2,1];
mu_max = [0.5,8,10];
mu_bar = [.1,4,1];

kappa{1} = '[mu3, 0, 0; 0, mu3, 0; 0, 0, 0]' ;
kappa{2} = '[1, 0, 0; 0, 1, 0; 0, 0, 0]' ;

dirichlet = '[1,0; 2,1; 4,0]' ;
nload     = '[3,0,0,1; 5, mu1,0,0;6,mu1,0,0]' ;

outputname = 'basetemp' ;
oload='[1,2]';
```

rbMIT Users' Interface

Example of rbUfile for User Problem Formulation

- **Geometry**
- **Parameters**

```
% -----        rbMIT  Software Copyright MIT 2006-09       -----
% --------- DBP Huynh, NC Nguyen, AT Patera, G Rozza ---------

probname = 'Tfin';

points = '[0,0; 1/2,0; 1/2,3/5; 3/20,3/5; 0,3/5; 3/20,3/5+mu2/2;
          3/20,3/5+mu2; 0,3/5+mu2; 0,3/5+mu2/2]';

edge = [1,2;2,3;3,4;4,5;4,6;6,7;7,8;8,9;9,5;5,1];

geometry{1} = [1,2,3,4,10];
geometry{2} = [4,5,6,7,8,9];
gflag = [1,1];

muref  = [.1,4,1];
mu_min = [.01,2,1];
mu_max = [0.5,8,10];
mu_bar = [.1,4,1];

kappa{1} = '[mu3, 0, 0; 0, mu3, 0; 0, 0, 0]' ;
kappa{2} = '[1, 0, 0; 0, 1, 0; 0, 0, 0]' ;

dirichlet = '[1,0; 2,1; 4,0]' ;
nload     = '[3,0,0,1; 5, mu1,0,0;6,mu1,0,0]' ;

outputname = 'basetemp' ;
oload='[1,2]';
```

rbMIT Users' Interface

Example of rbUfile for User Problem Formulation

- **Geometry**
- **Parameters**
- **PDE/BCs**

```
% -----        rbMIT  Software Copyright MIT 2006-09       -----
% --------- DBP Huynh, NC Nguyen, AT Patera, G Rozza ---------

probname = 'Tfin';

points = '[0,0; 1/2,0; 1/2,3/5; 3/20,3/5; 0,3/5; 3/20,3/5+mu2/2;
           3/20,3/5+mu2; 0,3/5+mu2; 0,3/5+mu2/2]';

edge = [1,2;2,3;3,4;4,5;4,6;6,7;7,8;8,9;9,5;5,1];

geometry{1} = [1,2,3,4,10];
geometry{2} = [4,5,6,7,8,9];
gflag = [1,1];

muref  = [.1,4,1];
mu_min = [.01,2,1];
mu_max = [0.5,8,10];
mu_bar = [.1,4,1];

kappa{1} = '[mu3, 0, 0; 0, mu3, 0; 0, 0, 0]' ;
kappa{2} = '[1, 0, 0; 0, 1, 0; 0, 0, 0]' ;

dirichlet = '[1,0; 2,1; 4,0]' ;
nload     = '[3,0,0,1; 5, mu1,0,0;6,mu1,0,0]' ;

outputname = 'basetemp';
oload='[1,2]';
```

rbMIT Users' Interface

Example of rbUfile for User Problem Formulation

- **Geometry**
- **Parameters**
- **PDE/BCs**
- **Output**

```
% -----        rbMIT  Software Copyright MIT 2006-09        -----
% --------- DBP Huynh, NC Nguyen, AT Patera, G Rozza ---------

probname = 'Tfin';

points = '[0,0; 1/2,0; 1/2,3/5; 3/20,3/5; 0,3/5; 3/20,3/5+mu2/2;
           3/20,3/5+mu2; 0,3/5+mu2; 0,3/5+mu2/2]';

edge = [1,2;2,3;3,4;4,5;4,6;6,7;7,8;8,9;9,5;5,1];

geometry{1} = [1,2,3,4,10];
geometry{2} = [4,5,6,7,8,9];
gflag = [1,1];

muref  = [.1,4,1];
mu_min = [.01,2,1];
mu_max = [0.5,8,10];
mu_bar = [.1,4,1];

kappa{1} = '[mu3, 0, 0; 0, mu3, 0; 0, 0, 0]' ;
kappa{2} = '[1, 0, 0; 0, 1, 0; 0, 0, 0]' ;

dirichlet = '[1,0; 2,1; 4,0]' ;
nload     = '[3,0,0,1; 5, mu1,0,0;6,mu1,0,0]' ;

outputname = 'basetemp';
oload='[1,2]';
```

Problem Formulation, Offline and Online Steps

- Example of geometry and field variable visualizations provided by `rbMIT` package

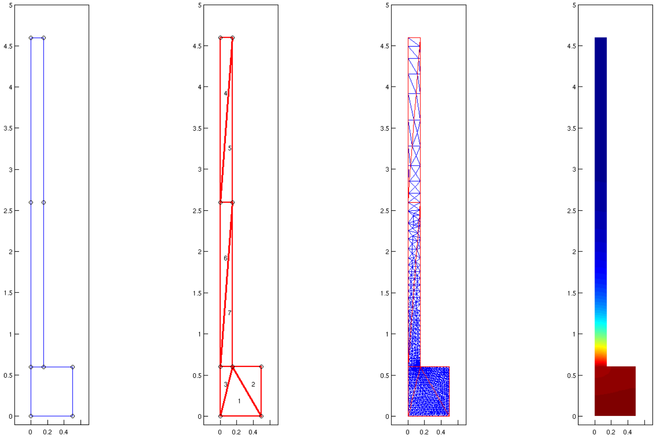
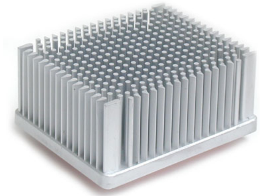


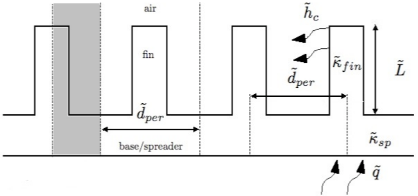
Figure: Initial geometry, domain decompostion, FE mesh and RB solution visualization for a thermal fin problem

The Thermal Fin problem

Engineeristic aspects

- Heat sink designed for thermal management of high-density electronic components
- Shaded domain due to assumed periodicity and symmetry (multi-fin sink)
- Flowing air is modelled though a simple convection HT coefficient: to compute temperature at the base of the spreader





Physical and geometrical parametrization

$\mu_1 = \mathrm{Bi} = \tilde{h}_c \tilde{d}_{\mathrm{per}} / \tilde{\kappa}_{\mathrm{fin}}$	Biot number	$\mu_1 \in [0.01, 0.5]$
$\mu_2 = L = \tilde{L} / \tilde{d}_{\mathrm{per}}$	nondimensional fin height	$\mu_2 \in [2, 8]$
$\mu_3 = \kappa = \tilde{\kappa}_{\mathrm{sp}} / \tilde{\kappa}_{\mathrm{fin}}$	spreader-to-fin conductivity ratio	$\mu_3 \in [1, 10]$

The Thermal Fin problem

- Modeling: temperature $u_o(\mu)$ over $\Omega_o(\mu)$ satisfies a steady conduction equation
- Output: average temperature over the base of the spreader (component to be cooled, being the hottest location in the system)

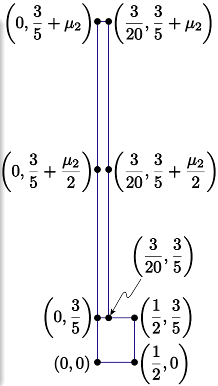
$$-\frac{\partial}{\partial x_{oi}}\left(\underbrace{\begin{bmatrix}\mu_3 & 0\\ 0 & \mu_3\end{bmatrix}}_{\kappa^1_{o\,ij}}\frac{\partial}{\partial x_{oj}}u_o(\mu)\right)=0 \quad \text{in } \Omega^1_o$$

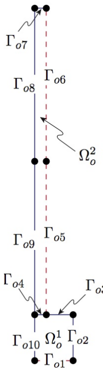
$$-\frac{\partial}{\partial x_{oi}}\left(\underbrace{\begin{bmatrix}1 & 0\\ 0 & 1\end{bmatrix}}_{\kappa^2_{o\,ij}}\frac{\partial}{\partial x_{oj}}u_o(\mu)\right)=0 \quad \text{in } \Omega^2_o(\mu_2)$$

$$\mathbf{n}_{o\,i}\kappa^1_{o\,ij}\frac{\partial u_o}{\partial x_{oj}}(\mu)=1 \quad \text{on } \Gamma_{o1}$$

$$\mathbf{n}_{o\,i}\kappa^2_{o\,ij}\frac{\partial u_o}{\partial x_{oj}}(\mu)+(\mu_1)u_o=0 \quad \text{on } \Gamma_R=\Gamma_{o5}\cup\Gamma_{06}$$

$$\mathbf{n}_{o\,i}\kappa_{o\,ij}\frac{\partial}{\partial x_{oj}}u_o(\mu)=0 \quad \text{on } \Gamma\setminus(\Gamma_{o1}\cup\Gamma_R)$$





output: $$T_{o\,\mathrm{av}}(\mu)=2\int_{\Gamma_{o1}}u_o(\mu)$$

The Thermal Fin problem

Approximation property	
# of mesh nodes $\mathcal{N}$	4198
# of RB functions N	≈ 10

Reduced Basis vs Finite Elements	
RB online evaluation time	$0.13s$ $(N = 7)$ $0.15s$ $(N = 13)$
FEM solution $\mu \rightarrow s^{\mathcal{N}}(\mu)$	$1.96s$

- ⋆ Reduction of 400:1 in linear system dimension
- ⋆ Online evaluation requires only $5-6\%$ of the FEM cpu cost

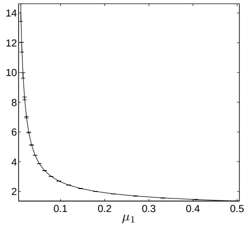


Figure: RB output and RB error bars — defined as the interval $[s_N(\mu) - \Delta_N^s(\mu), s_N(\mu) + \Delta_N^s(\mu)]$ — as a function of μ_1 for $\mu_2 = 2$, $\mu_3 = 1$ and $N = 6$.

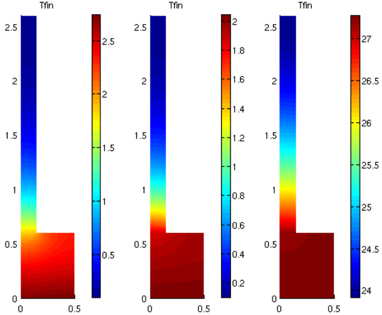
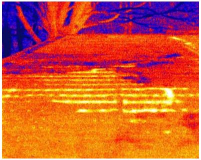


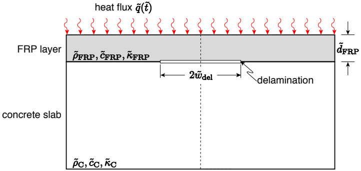
Figure: RB temperature field for different choices of parameters: $\mu = (0.5,\ 2,\ 1)$, $\mu = (0.5,\ 2,\ 5)$, $\mu = (0.01,\ 2,\ 10)$.

The Delamination Crack problem

Engineeristic aspects

- Analysis of the transient evolution of the temperature field near the surface of a Fiber-Reinforced-Polymer (FRP) Concrete (C) slab
- Application of transient conduction to real-time non-destructive crack detection
- Dependence of temperature field evolution on material/geometric inhomogeneities





Physical and geometrical parametrization

$\mu_1 = \tilde{w}_{\mathrm{del}}/\tilde{d}_{\mathrm{FRP}}^{\max}$	nondim. delamination crack width	$\mu_1 \in [0.01, 1]$
$\mu_2 = \tilde{d}_{\mathrm{FRP}}/\tilde{d}_{\mathrm{FRP}}^{\max}$	nondim. crack location (FRP layer thickness)	$\mu_2 \in [0.1, 1]$
$\mu_3 = \kappa = \tilde{\kappa}_{\mathrm{FRP}}/\tilde{\kappa}_{\mathrm{C}}$	ratio of FRP/Concrete thermal conductivities	$\mu_3 \in [0.4, 1.8]$

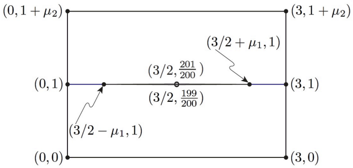
The Delamination Crack problem

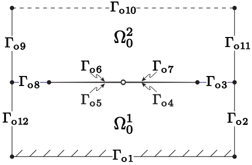
- Modeling: unsteady heat equation for temperature $u_o(\mu)$ over $\Omega_o(\mu) \times [0, t_f]$
- Output: integral of average temperature of the FRP layer over time interval $[0, t_f]$

$$\frac{\partial u_o(\mu)}{\partial t} - \frac{\partial}{\partial x_{oi}}\left(\underbrace{\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}}_{\kappa^1{}_{oij}} \frac{\partial}{\partial x_{oj}} u_o(\mu)\right) = 0 \quad \text{in } \Omega_o^1(\mu),$$

$$\frac{\partial u_o(\mu)}{\partial t} - \frac{\partial}{\partial x_{oi}}\left(\underbrace{\begin{bmatrix} \mu_3 & 0 \\ 0 & \mu_3 \end{bmatrix}}_{\kappa^2_{oij}} \frac{\partial}{\partial x_{oj}} u_o(\mu)\right) = 0 \quad \text{in } \Omega_o^2(\mu)$$

$$\begin{aligned} u_o(t=0) &= 0 \quad && \text{in } \Omega_o(\mu) \\ u_o(\mu) &= 0 && \text{on } \Gamma_{o1} \\ \mathbf{n}_{oi}\kappa^2_{oij}\frac{\partial u_o(\mu)}{\partial x_{oj}} &= g(t) && \text{on } \Gamma_{o10} \\ \mathbf{n}_{oi}\kappa_{oij}\frac{\partial}{\partial x_{oj}} u_o(\mu) &= 0 && \text{on } \Gamma \setminus (\Gamma_{o1} \cup \Gamma_{o10}) \end{aligned}$$





output: $s(\mu) = \frac{1}{3\mu_2}\int_0^{t_f}\left(h(t)\int_{\Omega_o^2} u_o(t;\mu)\right)dt$

The Delamination Crack problem

Approximation property	
# of mesh nodes $\mathcal{N}$	1912
# of time steps $\mathcal{K}$	50
# of RB functions N	25

⋆ Reduction of 80:1 in linear system dimension (at each time step)

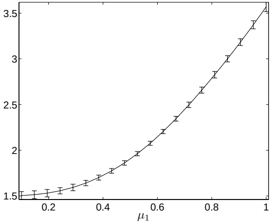


Figure: RB output and RB error bars — defined as the interval $[s_N(\mu) - \Delta^s_N(\mu), s_N(\mu) + \Delta^s_N(\mu)]$ — as a function of μ_1 for $\mu_2 = 0.2$, $\mu_3 = 1$ and $N = 25$.

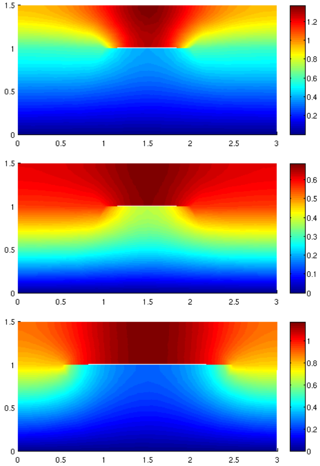


Figure: RB temperature field for different choices of parameters: $\mu = (0.5,\ 0.5,\ 0.4)$, $\mu = (0.5,\ 0.5,\ 1.8)$, $\mu = (1,\ 0.5,\ 1.8)$.

References

- A. Quarteroni, G. R. and A. M., 2011. Certified Reduced Basis Approximation for Parametrized PDEs and Applications. *J. Math. Industry*, 1:3.
- G. R., D. B. P. Huynh and A. T. Patera, 2008. "Reduced basis approximation and a posteriori error estimation for affinely parametrized elliptic coercive PDEs: Application to transport and continuum mechanics", *Arch. Comput. Methods Engrg.*, 15 (3), pp. 229–275.
- A.T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized PDEs*. to appear in MIT Pappalardo Graduate Monographs in Mechanical Engineering, 2009. ©Massachusetts Institute of Technology, Version 1.0.
- A. M., A. Quarteroni and G. R., 2012. Computational reduction for parametrized PDEs: strategies and applications. Milan J. Math. 80, pp. 283–309.
- G. R., C. N. Nguyen, D. B. P. Huynh, and A. T. Patera, 2009. Real-time reliable simulation of heat transfer phenomena, *Proceedings of HT2009, 2009 ASME Summer Heat Transfer Conference*, S. Francisco, USA, paper HT 2009–88212.
- M. Grepl, A.T. Patera, 2005. A posteriori error bounds for reduced-basis approximations of parametrized parabolic partial differential equations. *ESAIM: Math. Model. Numer. Anal.*, 39(1):157–181.
- M. Grepl, Y. Maday, N.C. Nguyen, and A.T. Patera, 2007. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM: Math. Model. Numer. Anal.*, 41(3):575–605.
- B. Haasdonk and M. Ohlberger, 2008. Reduced basis method for finite volume approximations of parametrized linear evolution equations. *ESAIM: Math. Model. Numer. Anal.*, 42:277–302.
- D. B. P. Huynh, C. N. Nguyen, G. R. and A. T. Patera, 2007-09. Documentation for rbMIT Software. `http://augustine.mit.edu/methodology/` ©MIT, Tech. Lic. Office 12600, Cambridge, MA, US.

Steady Heat Conduction: Formulation (1/2)

Scalar problem formulation

Given $\mu \in \mathscr{D} \subset \mathbb{R}^P$, we evaluate the output

$$s_o(\mu) = \int_{B_{oL}} u_o(\mu)$$

where the temperature field $u_o(\mu)$ satisfies

$$\begin{cases} -\frac{\partial}{\partial x_{oi}}\left(\kappa^k_{oij}\frac{\partial u_o(\mu)}{\partial x_{oj}}\right) + r^k_o u = f^k_o & \text{in } \Omega_o(\mu) \\ u_o = u_{oD} & \text{on } \Gamma_{oD} \\ n_{oi}\kappa^k_{oij}\frac{\partial u_o(\mu)}{\partial x_{oj}} + g_{o1}(u_o(\mu) - g_{o2}) = g_{o3} & \text{on } \Gamma_{oN} \end{cases}$$

Assumptions:

- ⋆ κ^k_{oij} is a 2×2 SPD tensor conductivity
- ⋆ $r^k_o \geq 0$ (reaction) and f^k_o (field) are scalars
- ⋆ g_{o1} is the Robin coefficient, g_{o2} is the "sink" field value, and g_{o3} is the flux

We must formulate our μ–dependent problem on a μ–indep. reference domain Ω:

- A decomposition of the domain $\Omega_o(\mu)$ in subdomains $\Omega^k_o(\mu)$, $1 \leq k \leq K_{\mathrm{reg}}$ is automatically built
- A piecewise-affine mapping which maps the μ-dependent $\bar{\Omega}_o(\mu) \equiv \bigcup_{k=1}^{K_{\mathrm{reg}}} \bar{\Omega}^k_o(\mu)$ to a reference μ-independent $\bar{\Omega} \equiv \bar{\Omega}(\mu_{\mathrm{ref}}) \equiv \bigcup_{k=1}^{K_{\mathrm{reg}}} \bar{\Omega}^k$ is built
- The problem is reduced to a parametric PDE on reference domain: geometric variations are now captured by the coefficients of the equation

Steady Heat Conduction: Formulation (2/2)

Finite Element discretization

Given $\mu \in \mathscr{D} \subset \mathbb{R}^P$, we evaluate

$$s^{\mathcal{N}}(\mu) = \{\mathbf{L}^{\mathcal{N}}(\mu)\}^T \{\mathbf{u}^{\mathcal{N}}(\mu)\}$$

where the FE temperature solution $\{\mathbf{u}^{\mathcal{N}}\} \in \mathbb{R}^{\mathcal{N}}$ satisfies

$$[\mathbf{K}^{\mathcal{N}}(\mu)]\{\mathbf{u}^{\mathcal{N}}(\mu)\} = \{\mathbf{F}^{\mathcal{N}}(\mu)\}$$

Finite Element Assumptions:

- ⋆ $[\mathbf{K}^{\mathcal{N}}(\mu)] \in \mathbb{R}^{\mathcal{N}\times\mathcal{N}}$: FE stiffness matrix $\{\mathbf{F}^{\mathcal{N}}(\mu)\} \in \mathbb{R}^{\mathcal{N}}$: force vector $\{\mathbf{L}^{\mathcal{N}}(\mu)\} \in \mathbb{R}^{\mathcal{N}}$: output vector
- ⋆ $[\mathbf{K}^{\mathcal{N}}(\mu)]$ is SPD and $\{\mathbf{L}^{\mathcal{N}}(\mu)\} = \{\mathbf{F}^{\mathcal{N}}(\mu)\}$ ("compliant" problem)

- The dimension of the FE approximation $\mathcal{N}$ is sufficiently large so that the FE output $s^{\mathcal{N}}(\mu)$ is indistinguishable from the exact output $s(\mu)$ at the accuracy level of interest
- The matrix $[\mathbf{K}^{\mathcal{N}}(\mu)]$ is "affine" in the parameter μ, by which we mean

$$[\mathbf{K}^{\mathcal{N}}(\mu)] = \sum_{q=1}^{Q} \Theta_q(\mu)[\mathbf{K}_q^{\mathcal{N}}]$$

  where for $q = 1, \ldots, Q$, the $\Theta_q : \mathscr{D} \to \mathbb{R}$ are (typically very smooth) μ-dependent functions, and the $[\mathbf{K}_q^{\mathcal{N}}]$ are μ-independent matrices
- The affine-parameter decomposition is crucial to the computational performance of the Offline-Online procedure (but it may be relaxed)

Steady Heat Conduction: Reduced Basis Approximation

- Snapshot FEM solutions with $\mu \in S_N = \{\mu^1, \ldots, \mu^N\}, 1 \le N \le N_{\max}$ give

$$W_N^{\mathcal{N}} = \mathrm{span}\{u^{\mathcal{N}}(\mu^n), 1 \le n \le N\} = \mathrm{span}\{\zeta_n^{\mathcal{N}}, 1 \le n \le N\}$$

RB formulation (Galerkin projection)

Given $\mu \in \mathcal{D}$, we evaluate the RB output as

$$s_N(\mu) = \{\mathbf{F}_N\}^T \{\mathbf{u}_N(\mu)\}$$

where the RB coefficient N–vector $\{\mathbf{u}_N(\mu)\}$ satisfies

$$[\mathbf{K}_N(\mu)]\{\mathbf{u}_N(\mu)\} = \{\mathbf{F}_N\}.$$

- $[\mathbf{Z}_N] \equiv [\mathbf{Z}_N^{\mathcal{N}}] = [\{\zeta_1^{\mathcal{N}}\}| \cdots |\{\zeta_N^{\mathcal{N}}\}]$ is the orthonormalized–snapshot $\mathcal{N} \times N$ matrix
- $[\mathbf{K}_N(\mu)] = [\mathbf{Z}_N]^T[\mathbf{K}^{\mathcal{N}}(\mu)][\mathbf{Z}_N]$
- $\{\mathbf{F}_N\} = [\mathbf{Z}_N]^T\{\mathbf{F}^{\mathcal{N}}\}$

- With affine form of $[\mathbf{K}_N(\mu)]$ the RB problem becomes $\sum_{q=1}^{Q} \Theta_q(\mu)[\mathbf{K}_{q\,N}]\{\mathbf{u}_N(\mu)\} = \{\mathbf{F}_N\}$ where the $[\mathbf{K}_{q\,N}] = [\mathbf{Z}_N]^T[\mathbf{K}_q^{\mathcal{N}}][\mathbf{Z}_N]$ are parameter-independent $N \times N$ matrices

Offline stage: compute the $\{\mathbf{u}^{\mathcal{N}}(\mu^n)\}$, $1 \le n \le N_{\max}$, form the matrix $[\mathbf{Z}_{N_{\max}}]$ and then form and store $\{\mathbf{F}_{N_{\max}}\}$ and $[\mathbf{K}_{q\,N_{\max}}]$

Online stage: for a given μ and N retrieve the pre-computed $[\mathbf{K}_{q\,N}]$ and $\{\mathbf{F}_N\}$, form $[\mathbf{K}_N(\mu)]$, solve the $N \times N$ system to obtain $\{\mathbf{u}_N(\mu)\}$, and evaluate s_N

Unsteady Heat Conduction: Formulation

Problem formulation

Given $\mu \in \mathscr{D} \subset \mathbb{R}^P$, we evaluate the output

$$s(\mu) = \int_0^{t_f} \left(h(t) \int_{B_L} u(t;\mu) \right) dt$$

where the temperature field $u(t;\mu)$ solves

$$\begin{cases} \frac{\partial u}{\partial t} - \frac{\partial}{\partial x_i}\left(\kappa_{ij}^k \frac{\partial u}{\partial x_j}\right) + r^k u = g(t) f^k & \text{in } \Omega(\mu) \\ u(t=0;\mu) = u_0 & \text{in } \Omega(\mu) \\ +\text{boundary conditions} & \text{on } \partial\Omega(\mu) \end{cases}$$

Assumptions:

- ⋆ Same as before on κ_{ij}^k, r^k, f^k
- ⋆ $h(t), g(t) \in L^2((0,t_f])$ are the output and input (control) functions of t
- ⋆ g_1, g_2, and g_3 may depend on t as well

FE Assumptions:

- ⋆ $[\mathbf{M}^{\mathcal{N}}(\mu)]$ FE mass matrix, SPD and affine in the parameter: $[\mathbf{M}^{\mathcal{N}}(\mu)] = \sum_{j=1}^{J} \Phi_j(\mu)[\mathbf{M}_j^{\mathcal{N}}]$
- ⋆ Trapezoidal rule for $s^{\mathcal{N}}(\mu)$

Semi-discrete Finite Element approximation

Given $\mu \in \mathscr{D} \subset \mathbb{R}^P$, we evaluate

$$s^{\mathcal{N}}(\mu) = \int_0^{t_f} \left(h(t)\{\mathbf{L}^{\mathcal{N}}\}^T\{\mathbf{u}^{\mathcal{N}}(t;\mu)\} \right) dt$$

where the FE temperature vector $\mathbf{u}^{\mathcal{N}}(t;\mu) \in \mathbb{R}^{\mathcal{N}}$ satisfies for $t \in (0,t_f]$

$$[\mathbf{M}^{\mathcal{N}}(\mu)]\{\dot{\mathbf{u}}^{\mathcal{N}}(t;\mu)\} + [\mathbf{K}^{\mathcal{N}}(\mu)]\{\mathbf{u}^{\mathcal{N}}(t;\mu)\} = g(t)\{\mathbf{F}^{\mathcal{N}}\}$$

with initial condition $\mathbf{u}^{\mathcal{N}}(t=0;\mu) = \mathbf{u}_0^{\mathcal{N}}$

Unsteady Heat Conduction: Reduced Basis Approximation

- For generation of RB spaces $W_N^{\mathcal{N}} = \text{span}\{\zeta_n^{\mathcal{N}}, 1 \le n \le N\}$, $1 \le N \le N_{\max}$, a **POD-Greedy sampling procedure** combines spatial snapshots in time and μ
  - ⋆ POD in t captures the causality associated with the evolution equation
  - ⋆ Greedy in μ treats efficiently more extensive ranges of parameter variation

Reduced Basis formulation (Galerkin projection)

Given $\mu \in \mathscr{D}$, we evaluate the RB output as

$$s_N(\mu) = \int_0^{t_f} \left(h(t) \{\mathbf{L}_N\}^T \{\mathbf{u}_N(t;\mu)\} \right) dt$$

where $\{\mathbf{u}_N(t;\mu)\}$ satisfies the evolution equation

$$\sum_{j=1}^{J} \Phi_j(\mu)[\mathbf{M}_{jN}]\{\dot{\mathbf{u}}_N(t;\mu)\} + \sum_{q=1}^{Q} \Theta_q(\mu)[\mathbf{K}_{qN}]\{\mathbf{u}_N(t;\mu)\} = g(t)\{\mathbf{F}_N\}$$

- The following affine representations for stiffness/mass matrices is used:

$$\{\mathbf{L}_N\} = [\mathbf{Z}_N]^T \{\mathbf{L}^{\mathcal{N}}\}, \qquad \{\mathbf{F}_N\} = [\mathbf{Z}_N]^T \{\mathbf{F}^{\mathcal{N}}\}$$

$$[\mathbf{K}_{qN}] = [\mathbf{Z}_N]^T [\mathbf{K}_q^{\mathcal{N}}][\mathbf{Z}_N], 1 \le q \le Q, \qquad [\mathbf{M}_{jN}] = [\mathbf{Z}_N]^T [\mathbf{M}_j^{\mathcal{N}}][\mathbf{Z}_N], 1 \le j \le J$$

- Offline-Online procedure is straightforward and very similar to the steady case

Steady/Unsteady Heat Conduction: RB Error Estimation

A posteriori error estimator is a certificate of fidelity that rigorously bounds the error in the RB prediction relative to the highly accurate truth finite element solution

Steady case

$$|s^{\mathcal{N}}(\mu) - s_N(\mu)| \leq \Delta_N^s(\mu) = \varepsilon^2(\mu)/\alpha_{\mathrm{LB}}^{\mathcal{N}}(\mu)$$

- $\varepsilon^2(\mu) = \{\mathbf{R}^{\mathcal{N}}\}^T[\mathbf{Y}^{\mathcal{N}}]^{-1}\{\mathbf{R}^{\mathcal{N}}\}$ = square of the dual norm of the residual vector
$$\{\mathbf{R}^{\mathcal{N}}\} = \{\mathbf{F}^{\mathcal{N}}\} - [\mathbf{K}^{\mathcal{N}}(\mu)][\mathbf{Z}_N]\{\mathbf{u}_N(\mu)\}$$
- $\alpha_{\mathrm{LB}}^{\mathcal{N}}(\mu)$ is a lower bound for the discrete coercivity constant (SCM method)

Unsteady case

$$|s^{\mathcal{N}}(t,\mu) - s_N(t,\mu)| \leq \Delta_N^s(t,\mu) = \frac{\sigma_0}{\alpha_{\mathrm{LB}}^{\mathcal{N}}(\mu)}\left(\left(\int_0^{t_f} h^2(t)dt\right)\left(\int_0^{t_f} \varepsilon^2(t;\mu)dt\right)\right)^{1/2}$$

- $\sigma_0^2 = \{\mathbf{L}^{\mathcal{N}}\}^T[\mathbf{Y}^{\mathcal{N}}]^{-1}\{\mathbf{L}^{\mathcal{N}}\}$ = square of the dual norm of the output vector $\mathbf{L}^{\mathcal{N}}$
- $\varepsilon^2(t;\mu) = \{\mathbf{R}^{\mathcal{N}}\}^T[\mathbf{Y}^{\mathcal{N}}]^{-1}\{\mathbf{R}^{\mathcal{N}}\}$ = square of the dual norm of the residual vector

The computation of $\varepsilon^2(\mu)$ readily admits an Offline-Online strategy: all the underbraced matrix-matrix or matrix-vector products can be pre-computed Offline