Solar Impulse, Optimization Framework and Reduced Basis

Christophe Prud'homme

christophe.prudhomme@epfl.ch

ModeFrontier User Meeting '04
Trieste
Oct 1st 2004

Contents

Collaborators

EPFL/MIT/Paris VI

A. Quarteroni	T. Patera	Y. Maday
G. Fourestey	K. Veroy	G. Turinici
P.O. Moix	N. Cuong	
A. Borshberg	M. Grepl	

1 Solar Impulse

1.1 Solar-Impulse Objectives

1.1.1 Solar-Impulse Objectives

<u>**Mission**</u>
Achieve a flight around the world with a solar powered airplane

Constraints

- Achieve a sustainable flight
- Use only renewable energies
- Eliminate polluting emissions

1.1.2 Solar-Impulse Objectives

Achieve "Aeronautical Firsts"

- Promote sustainable development in general;
- Promote the use of renewable energies in particular;
- Mobilize the enthousiasm of the public in order to change mentalities in relation with the problems of environment;
- Reinforce the perception that technologies can help achieve the objective of sustainable development.

1.2 Solar-Impulse Challenges

1.2.1 Energy Challenges

Optimize Energy Collection and Consumption

- 35kW peak power but only 10kW over 24hrs
- → energy stored in batteries and altitude
- → large wingspan for adequate night altitude
- → maximize minimum night altitude

Early Design
Wingspan ~ $80m$, Wingarea ~ $200m^2$

1.2.2 Structure Challenges

Solar Cells Encapsulation, Structure Weight, Batteries Insulation

- Reduce Airplane Weight
- Flexible and light solar cells
- Multi-functional encapsulation
- High transmittance level at high angle of incidence
- Physical characteristics maintained at high and low temperature
- Batteries temperature maintained in a narrow range
- Multi-functional battery isolation

1.2.3 Safety Challenges

Viable and safe environment at 12'000 meters

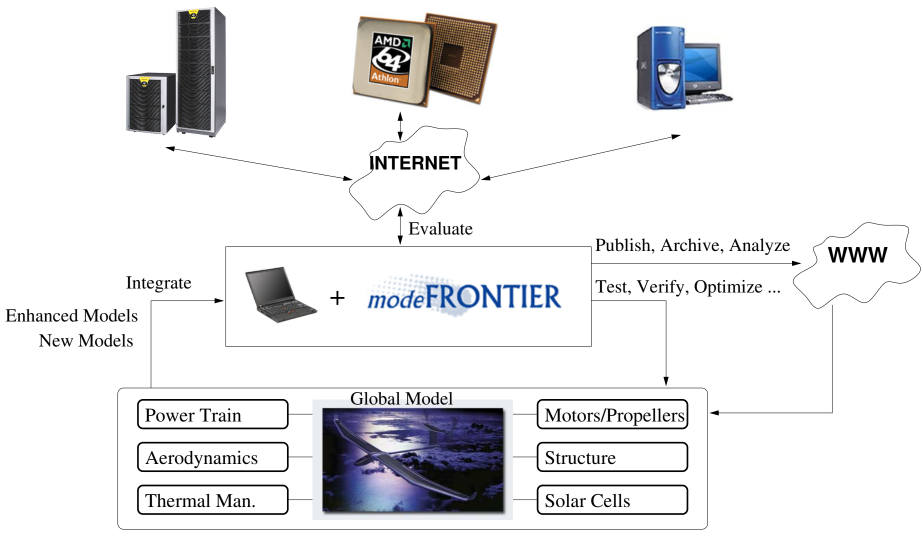
- High altitude: protection against gamma rays
- Low temperatures (-55 °C) : energy recuperation for heating purpose
- Little oxygen : light pressurization system with low energy consumption
- Long flights duration : automatic control and management of the airplane energy systems

1.3 Global Optimization

1.3.1 Global Optimization and MF

- Clear need for Global Multifunctional Optimization
- ModeFrontier is well suited especially as a plateform
- Build an environment revolving around modeFrontier
  - Dynamic website for results publishing and interaction with other groups
  - Usage of computing ressources available on campus from mainframes, to laptops

1.3.2 Global Optimization and MF



1.3.3 Global Optimization Challenges

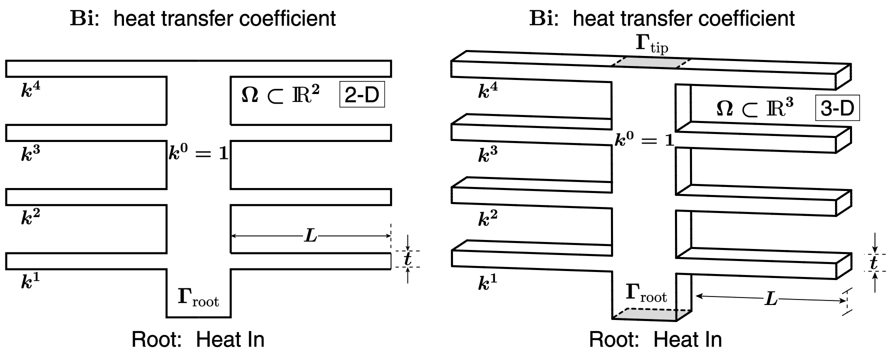
Challenges

- Result/data mining — Communication
- Expertise is not centralized — Communication
- Models order reduction — MF/RSM/RBOBM

2 Reduced Basis Output Bounds Methods (RBOBM)

2.1 3D Thermal Fin

2.1.1 Conduction Heat Transfer



Mathematical Model

2.1.2 Mathematical Model

Operator:

piecewise-constant-coefficient Laplacian, $-\nabla \cdot (k_i \nabla)$.

Domain: $\Omega^o(t_i, L_i)_{i=1\dots4}$

Interface Conditions:

continuity of temperature and flux.

Boundary Conditions:

Neumann (flux) on Γ_{root}; and
Robin (with coefficient Bi) on $\partial\Omega/\Gamma_{\text{root}}$.

2.1.3 Inputs and Outputs

Inputs :

$$\mu \equiv \big(\underbrace{(k^i)_{i=1\dots4}}_{\text{thermal conductivities}} , \underbrace{\text{Bi}}_{\text{convective coefficient}} , \underbrace{(t^i, \ell^i)_{i=1\dots4}}_{\text{geometry}} \big)$$

$$\mu \in \mathcal{D}^\mu \subset \mathbb{R}^{13}$$

Outputs : quantities of interest

$$s^1(\mu) \equiv \int_{\Gamma_{\text{root}}} T(\underline{x};\mu)^\dagger = \ell^1(T(\underline{x};\mu);\mu) \qquad \text{T\_ROOT}^\dagger$$

$$s^2(\mu) \equiv \int_\Omega = \ell^2(\mu) \qquad \text{VOLUME}$$

$$s^3(\mu) = \text{Bi} \qquad \text{“FAN\_POWER”}$$

† The temperature is defined relative to the ambient level, T_{amb}.

2.1.4 Engineering Goals

We wish to maintain

$$\text{T\_ROOT}\ [-T_{amb}] \le \underbrace{T_{per}}_{\substack{\text{OPERATING}\\ \text{CONSTRAINT}}} - \underbrace{T_{amb}}_{\substack{\text{ENVIRONMENTAL}\\ \text{CONDITION}}}$$

while minimizing

$$\underbrace{\text{VOLUME},\ \text{FAN\_POWER}}_{\text{COSTS}}.$$

Optimization should be performed both at design and (adaptively) in operation.

2.1.5 Optimization Problem: Design

For DESIGN ITERATION $1, 2, \ldots$

Choose	k_1, k_2, k_3, k_4	materials;
	c_1, c_2	preferences;
	$(T_{per} - T_{amb})_{min}$	scenarios.

Minimize c_1 VOLUME $(\mu) + c_2$ FAN_POWER(μ) .

$$\text{FEASIBLE} \begin{cases} \text{Bi}, (t_i, L_i)_{i,\ldots,4} \in \mathcal{D}^{\text{Bi},(t_i,L_i)_{i,\ldots,4}} \\ \text{T\_ROOT}\ (\mu) \le (T_{per} - T_{amb})_{min} \end{cases}$$

"Inspect" proposed optimal design.

2.1.6 Optimization Problem: Operation

Configuration

$$(k_i, t_i, L_i)_{i=1,\ldots,4} \quad \text{NOW FIXED}$$

Real-Time
For OPERATION (MISSION) $1, 2, \ldots$

"Measure" T_{amb};

Minimize FAN_POWER (μ). $(= \text{Bi})$

$$\text{FEASIBLE} \begin{cases} \text{Bi} \in \mathcal{D}^{\text{Bi}} \\ \text{T\_ROOT}\ (\mu) \le T_{per} - T_{amb} \end{cases}$$

Perform mission.

2.1.7 DEMO

DEMO

2.2 Methodology

2.2.1 Objectives

Develop methodology to

Evaluate $s(\mu)^\dagger$ rapidly and reliably in the limit of very many $\underline{\mu}$-queries;

with application to optimal design and operation.

Enablers:

- Restriction to parametric dependence;
- Acceptance of large initial (off-line) cost.

†Outputs of μ-parametrized partial differential equations.

2.2.2 Ingredients

- Reduced order approximation: $s_N(\mu)$
- Reliable and sharp error estimation: $\Delta_N(\mu)$
- Offline(expensive) / Online($O(1\text{ms})$) decomposition
- Fast automatic differentiation
- Maximum Error Control
- Maximum Computing Time Control

See for example [1, 3, 2]

2.2.3 Reduced Basis Approximation: FEM Approximation

Let $\mu \in \mathcal{D}^\mu$, evaluate

$$s^{\mathcal{N}}(\mu) = \ell(u^{\mathcal{N}}(\mu))\ , \qquad \text{“}= s(\mu)\text{”}$$

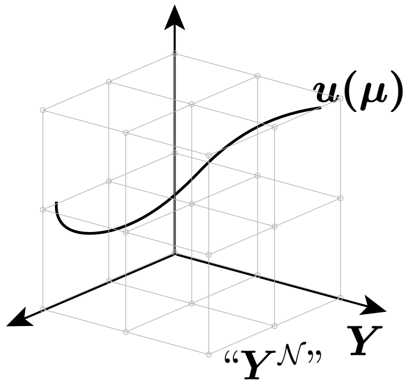
where $u^{\mathcal{N}}(\mu) \in Y^{\mathcal{N}}$ satisfies “$= u(\mu)$”

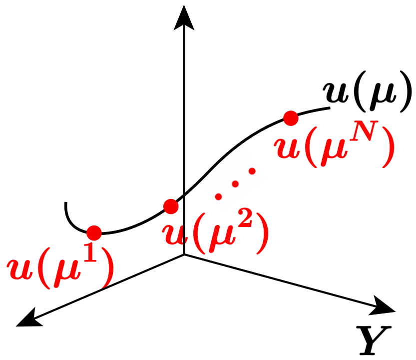
$$a(u^{\mathcal{N}}(\mu), v; \mu) = f(v), \quad \forall\, v \in Y^{\mathcal{N}}\ .$$

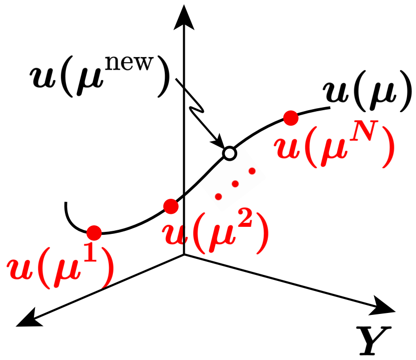
Here $Y^{\mathcal{N}} \subset Y$ is a Truth finite element approximation of dimension $\boxed{\mathcal{N} \gg 1}$.

2.2.4 Reduced Basis Approximation: Order Reduction

To approximate $u(\mu)$, and thus $s(\mu)$, we need not represent all functions in Y.







2.2.5 Reduced Basis Approximation: Formulation

Sample: $S_N = \{\mu^1 \in \mathcal{D}^\mu, \ldots, \mu^N \in \mathcal{D}^\mu\}$.

Space: $W_N = \text{span}\,\{\zeta_n \equiv \underbrace{u(\mu^n)}_{u^{\mathcal{N}}(\mu^n)}, n = 1, \ldots, N\}$. Galerkin Projection:

Given $\mu \in \mathcal{D}^\mu$, evaluate

$$s_N(\mu) = \ell(u_N(\mu))\,;$$

where $u_N(\mu) \in W_N$ statisfies

$$a(u_N(\mu), v; \mu) = f(v),\ \forall\, v \in W_N\,.$$

†Initial Work: Al, No, Rh, Po, Gu, ...

2.2.6 Reduced Basis Approximation: Matrix Form

Express

$$u_N(\mu) = \sum_{j=1}^{N} u_{Nj}(\mu)\,\zeta_j, \quad v = \zeta_i,\ i = 1, \ldots, N$$

To obtain

$$\underline{A}_N(\mu)\,\underline{u}_N(\mu) = \underline{F}_N$$

where $\underset{1 \le i,j \le N}{(A_N)_{i\,j}(\mu) = a(\zeta_j, \zeta_i; \mu)}$, $\quad \underset{1 \le i \le N}{F_{N\,i} = f(\zeta_i)}$.

then, $s_N(\mu) = \underline{L}_N^T\,\underline{u}_N(\mu)$, where $\underset{1 \le i \le N}{L_{N\,i} = \ell(\zeta_i)}$.

2.2.7 Fast and sharp Error Estimation: Ingredients

Introduce

$$\hat{\alpha}: \quad \mathcal{D}^\mu \to \mathbb{R} \qquad \text{positive}$$

$$\hat{a}: \quad Y \times Y \to \mathbb{R} \qquad \text{coercive over } Y$$

such that, $\forall\,\mu \in \mathcal{D}^\mu$,

$$\hat{\alpha}(\mu)\,\hat{a}(v,v) \le a(v,v;\mu), \qquad \forall\, v \in Y\,.$$

(For $\hat{a}(\,\cdot\,,\,\cdot\,) = (\,\cdot\,,\,\cdot\,)_Y$, $\hat{\alpha}(\mu)$ is a lower bound for $\alpha(\mu)$†.)

† $\alpha(\mu)$ is the coercity/inf-sup constant of $a(\cdot,\cdot;\mu)$

2.2.8 Fast and sharp Error Estimation: Residual Norm

Denote

$$r(v;\mu) \equiv f(v) - a(u_N, v; \mu), \qquad \forall\, v \in Y\,,$$

and $\hat{e}(\mu) \in Y$ such that

$$\hat{a}(\hat{e}(\mu), v) = r(v;\mu), \qquad \forall\, v \in Y\,.$$

Then

$$\|r(\,\cdot\,;\mu)\|_D \equiv [\hat{a}(\hat{e}(\mu), \hat{e}(\mu))]^{1/2}$$

is the dual norm of the residual.

2.2.9 Fast and sharp Error Estimation: Definition

For all $\mu \in \mathcal{D}^\mu$,

$$\begin{aligned} \Delta_N(\mu) &\equiv \hat{\alpha}^{-1}(\mu)\,\|r(\,\cdot\,;\mu)\|_D^2 \\ &= \hat{\alpha}^{-1}(\mu)\,\hat{a}(\hat{e}(\mu),\hat{e}(\mu))\,; \end{aligned}$$

and

$$\begin{aligned} s_N^-(\mu) &= s_N(\mu) \\ s_N^+(\mu) &= s_N(\mu) + \Delta_N(\mu)\,. \end{aligned}$$

2.2.10 Fast and sharp Error Estimation: Properties

For all $\mu \in \mathcal{D}^\mu$, and all $N \in \mathbb{N}$,

$$\underbrace{1}_{\substack{|s(\mu)-s_N(\mu)| \le \Delta_N(\mu) \\ \text{RELIABILITY}}} \le \frac{\Delta_N(\mu)}{|s(\mu)-s_N(\mu)|} \le \underbrace{\boxed{C_{[\hat{a},\hat{\alpha}]}}\,;}_{\substack{\Delta_N(\mu) \le C_{[\hat{a},\hat{\alpha}]}\,|s(\mu)-s_N(\mu)|^{\dagger} \\ \text{EFFICIENCY}}}$$

Moreover,

$$s_N^-(\mu) = s_N(\mu) \le s(\mu) \le s_N^+(\mu)\,.$$

Note . . .
. . . that $\mathrm{T\_ROOT}_N(\mu) = T_{\mathrm{per}} \Rightarrow \mathrm{T\_ROOT}(\mu) > T_{\mathrm{per}}$; but $\mathrm{T\_ROOT}_N^+(\mu) = T_{\mathrm{per}} \Rightarrow \mathrm{T\_ROOT}\,(\mu) \le T_{\mathrm{per}}$.

2.2.11 Fast and sharp Error Estimation: Questions

Can we compute

$$\hat{a}(\hat{e}(\mu),\hat{e}(\mu))\,,$$
$$\hat{e}(\mu) \in Y:\ \hat{a}(\hat{e}(\mu),v) = r(v;\mu),\ \forall\, v \in Y\,,$$
$$[\hat{e}(\mu) \in Y:\ \langle \hat{A}\,\hat{e}(\mu),v\rangle = \langle R(\mu),v\rangle,\ \forall\, v \in Y]$$

with an Online complexity independent of $\mathcal{N}$?

YES: $O(Q^2N^2)$;
or $O(QN^2)$ (domain decomposition).

2.2.12 Offline/Online Decomposition: Ingredient

Given $a(\,\cdot\,,\,\cdot\,;\mu)$,

$$a(w,v;\mu) = \sum_{q=1}^{Q} \Theta_q(\mu)\,a_q(w,v)\,,$$

We can write

$$\begin{aligned} (A_N)_{ij}(\mu) &= a(\zeta_j,\zeta_i;\mu) \\ &= \sum_{q=1}^{Q} \Theta_q(\mu)\,a_q(\zeta_j,\zeta_i) \\ &= \sum_{q=1}^{Q} \Theta_q(\mu)(A_{Nq})_{ij}\,. \end{aligned}$$

2.2.13 Offline/Online Decomposition: $s_N(\mu)$

OFF-LINE: $(\mathcal{N}, N, Q)$	Evaluate $\zeta_n,\ 1 \le n \le N$;
	Form $\underline{A}_{N\,q}, 1 \le q \le Q$; and $\underline{F}_N$.
ON-LINE: (N, Q)	Assemble $\underline{A}_N(\mu)$ — $O(QN^2)$;
	Solve for $\underline{u}_N(\mu)$ — $O(N^3)^{\ddagger}$;
	Evaluate $s_N(\mu)$ — $O(N)$.

2.2.14 Offline/Online Decomposition: $\Delta_N(\mu)$

Same Strategem as $s_N(\mu)$ but applied to $\hat{a}(\hat{e}, \hat{e})$.

OFF-LINE: Evaluate $\hat{z}_{00} = \hat{A}^{-1}F, \hat{z}_{qn} = \hat{A}^{-1}A_q\zeta_N$;
Form $\langle F, \hat{z}_{00}\rangle, \langle A_q\zeta_n, \hat{z}_{00}\rangle + \langle F, \hat{z}_{qn}\rangle$,
$\langle A_q\zeta_n, \hat{z}_{q'n'}\rangle$;
$1 \le q, q' \le Q,\ 1 \le n, n' \le N$.

ON-LINE: Evaluate $\Delta_N(\mu)$ — $O(Q^2N^2)$. †
INDEPENDENT OF $\mathcal{N}$

2.2.15 Fast Automatic Differentiation

Given $\mu \in \mathcal{D}^\mu$, compute

$$\begin{array}{l}(s_N(\mu), \nabla_\mu s_N(\mu), \nabla^2_\mu s_N(\mu)) \\ (\Delta_N(\mu), \nabla_\mu \Delta_N(\mu), \nabla^2_\mu \Delta_N(\mu))\end{array}$$

where

$$\begin{aligned}\nabla_\mu &= \left(\frac{\partial\cdot}{\partial\mu_i}\right)_{i=1\ldots P} \in \mathbb{R}^P \\ \nabla^2_\mu &= \left(\frac{\partial^2\cdot}{\partial\mu_i\partial\mu_j}\right)_{i,j=1\ldots P} \in \mathbb{R}^{P\times P}\end{aligned}$$

2.2.16 Maximum Error Control

Formulation

Input: Given a maximum error τ not to be exceeded
Objective: Compute $s_N(\mu)$ and $\Delta_N(\mu)$ such that with minimum complexity

$$\boxed{\forall\, \mu \in \mathcal{D}^\mu\ ,\ \Delta_N(\mu) < \tau}$$

Best accuracy not necessarily always needed. Can be used to accelerate optimization algorithm convergence.

2.2.17 Maximum Computing Time Control

Formulation

Input: maximum computing time τ

Objective: compute $s_N(\mu)$ and $\Delta_N(\mu)$ such that with minimum complexity

$$\boxed{\forall\, \mu \in \mathcal{D}^{\mu}, \text{ computing time}(s_N(\mu), \Delta_N(\mu)) < \tau}$$

Very important for hard constraints in real-time systems.

2.2.18 Application Domains

- Heat transfer
- Elasticity (linear and non linear)
- Fluid dynamics
- Acoustics
- And more generally *elliptic* and *parabolic* equations

2.3 RBOBM & MF

2.3.1 Current Status

Current Status

- Use script to evaluate outputs of interest
- Reload RBOBM DB/One $s(\mu)$-evaluation "slow"
- Handle error $s_N(\mu) + \Delta_N(\mu)$(upper) or $s_N(\mu)$ (lower)
- Finite difference derivatives "accuracy"

Proposal

MF plugin for:

- Evaluation of $s(\mu)$ using some approximate $\hat{s}(\mu)$ e.g. $s_N(\mu)$
- Handling of approximation error $|s(\mu) - \hat{s}(\mu)|^{\dagger}$ e.g. $\Delta_N(\mu)$
- Exact derivatives computation

† if available

Summary

Summary

- ModeFrontier is an excellent design *plateform*
- ModeFrontier allows to plug RBOBM algorithms using scripts
- ModeFrontier could allow plugins that provides $s(\mu)$-*rapid calculations* with (possibly)
  - Error handling
  - Automatic differentiation handling
  - Accuracy control

References

[1] C. Prud'homme, D. Rovas, K. Veroy, Y. Maday, A.T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. Journal of Fluids Engineering, 124(1):70–80, March 2002.

[2] K. Veroy, C. Prud'homme, and A.T. Patera. Reduced-basis approximation of the viscous equation: Rigorous a posteriori error bounds. C. R. Acad. Sci. Paris, 2003. Accepted for publication.

[3] K. Veroy, C. Prud'homme, D.V. Rovas, and A.T. Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In Proceedings of the 16th AIAA Computational Fluid Dynamics Conference, June 2003.