Surrogate and Reduced-Order Models

**Problem:** Difficult to obtain sufficient number of realizations of discretized PDE models for Bayesian model calibration, design and control.

\[
\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho v) = 0
\]

\[
\frac{\partial v}{\partial t} = -v \cdot \nabla v - \frac{1}{\rho} \nabla p - g \hat{k} - 2\Omega \times v
\]

\[
\rho c_v \frac{\partial T}{\partial t} + p \nabla \cdot v = -\nabla \cdot F + \nabla \cdot (k \nabla T) + \rho \dot{q}(T, p, \rho)
\]

\[
p = \rho RT
\]

\[
\frac{\partial m_j}{\partial t} = -v \cdot \nabla m_j + S_m(\rho, T, m_j, \chi_j, \rho), \quad j = 1, 2, 3,
\]

\[
\frac{\partial \chi_j}{\partial t} = -v \cdot \nabla \chi_j + S_{\chi}(T, \chi_j, \rho), \quad j = 1, \ldots, J,
\]

**Solution:** Construct surrogate models

- Also termed data-fit models, response surface models, emulators, meta-models
- Projection-based models often called reduced-order models
Surrogate Models: Motivation

Example: Consider the model

\[
\frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = f(q)
\]

Boundary Conditions

Initial Conditions

with the response

\[
y(q) = \int_0^1 \int_0^1 \int_0^1 \int_0^1 u(t, x, y, z; q) dx dy dz dt
\]

Notes:
- Requires approximation of PDE in 3-D
- What would be a simple surrogate?
Surrogate Models

Example: Consider the model

$$\frac{\partial u}{\partial t} + \frac{\partial^2 u}{\partial x^2} + \frac{\partial^2 u}{\partial y^2} + \frac{\partial^2 u}{\partial z^2} = f(q)$$

Boundary Conditions
Initial Conditions

with the response

$$y(q) = \int_0^1 \int_0^1 \int_0^1 \int_0^1 u(t, x, y, z; q) dx dy dz dt$$

Surrogate: Quadratic

$$y_s(q) = (q - 0.25)^2 + 0.5$$
Surrogate and Reduced-Order Models

Issues:

• Techniques for regression versus interpolation
• Grid choices for interpolation
• Techniques for time-dependent problems
Data-Fit Models

Notes:

• Often termed response surface models, emulators, meta-models;
• Rely on interpolation or regression;
• Data can consist of high-fidelity simulations or experiments.

• Common techniques: polynomial models, kriging (Gaussian process regression), stochastic collocation.

Strategy: Consider high fidelity model

\[ y = f(q) \]

with M model evaluations

\[ y_m = f(q^m), \ m = 1, \ldots, M \]

Statistical Model: \( f_s(q) \): Emulator for \( f(q) \)

\[ y_m = f_s(q^m) + \varepsilon_m, \ m = 1, \ldots, M \]
Data-Fit Models – Polynomial Emulator

**Quadratic Emulator:** Regression

\[ f_s(q; \beta) = \beta_0 + \beta_1 q + \beta_2 q^2 \]

**Deterministic System:** \( y_{obs} = X \beta \)

\[
\begin{bmatrix}
  y_1 \\
  \vdots \\
  y_M
\end{bmatrix} =
\begin{bmatrix}
  1 & q^1 & (q^1)^2 \\
  \vdots & \vdots & \vdots \\
  1 & q^M & (q^M)^2
\end{bmatrix}
\begin{bmatrix}
  \beta_0 \\
  \beta_1 \\
  \beta_2
\end{bmatrix}
\]

**Least Squares Estimate:**

\[
\beta = [X^T X]^{-1} X^T y_{obs}
\]

**Notes:**

- Good choice for optimization;
- Accurate approximation may require high-order polynomials;
- Does not provide uncertainty bounds for uncertainty quantification.
Data-Fit Models – Collocation

**Strategy:** Consider high fidelity model

\[ y = f(q) \]

with M model evaluations

\[ y_m = f(q^m), \ m = 1, \cdots, M \]

**Collocation Surrogate:**

\[ y_s(q) = f_s(q) = \sum_{m=1}^{M} y_m L_m(q) \]

where \( L_m(q) \) is a Lagrange polynomial, which in 1-D is represented by

\[ L_m(q) = \prod_{\substack{j=0 \atop j \neq m}}^{M} \frac{q - q^j}{q^m - q^j} = \frac{(q - q^1) \cdots (q - q^{m-1})(q - q^{m+1}) \cdots (q - q^M)}{(q^m - q^1) \cdots (q^m - q^{m-1})(q^m - q^{m+1}) \cdots (q^m - q^M)} \]

**Note:**

\[ L_m(q^j) = \delta_{jm} = \begin{cases} 
0 & , \ j \neq m \\
1 & , \ j = m 
\end{cases} \]

**Result:**

\[ y_s(q^m) = f(q^m) \]
Data-Fit Models – Collocation

**Strategy:** Consider high fidelity model

\[ y = f(q) \]

with M model evaluations

\[ y_m = f(q^m), \ m = 1, \cdots, M \]

**Collocation Surrogate:**

\[ y_s(q) = f_s(q) = \sum_{m=1}^{M} y_m L_m(q) \]

where \( L_m(q) \) is a Lagrange polynomial, which in 1-D is represented by

\[
L_m(q) = \prod_{\substack{j=0 \atop j \neq m}}^{M} \frac{q - q^j}{q^m - q^j} = \frac{(q - q^1) \cdots (q - q^{m-1})(q - q^{m+1}) \cdots (q - q^M)}{(q^m - q^1) \cdots (q^m - q^{m-1})(q^m - q^{m+1}) \cdots (q^m - q^M)}
\]

**Note:**

\[ L_m(q^j) = \delta_{jm} = \begin{cases} 
0, & j \neq m \\
1, & j = m
\end{cases} \]

**Result:**

\[ y_s(q^m) = f(q^m) \]

**Note:** Method is nonintrusive and treats code as blackbox.
Data-Fit Models – Gaussian Process Emulator

**Kriging (Gaussian Process):**

\[
f_s(q; \beta) = g^T(q)\beta + Z(q)
\]

- \(g^T(q)\beta\): Trend function
- \(Z(q)\): Gaussian process with

\[
\text{cov}(Z(q^i), Z(q^j)) = \sigma^2 R(q^i, q^j) + \sigma_0^2 \delta(q^i, q^j)
\]

\[
R(q^i, q^j) = \exp\left(-\sum_{k=1}^{p} |\theta_k(q^i_k - q^j_k)|^{\gamma_k}\right)
\]

**Error Bounds:**

![Error Bound Diagram](image)
Surrogate Models – Grid Choice

Example: Consider the Runge function $f(q) = \frac{1}{1 + 25q^2}$ with points

$$q^j = -1 + (j - 1) \frac{2}{M_1}, \; j = 1, \ldots, M_1 + 1$$
Surrogate Models – Grid Choice

Example: Consider the Runge function $f(q) = \frac{1}{1 + 25q^2}$ with points

\[ q^j = -1 + (j - 1) \frac{2}{M_1}, \quad j = 1, \ldots, M_1 + 1 \]

\[ q^j = -\cos \frac{\pi(j - 1)}{M_1 - 1}, \quad j = 1, \ldots, M_1 \]
Sparse Grid Techniques

**Tensored Grids:** Exponential growth as a function of dimension

**Sparse Grids:** Same accuracy with significantly reduce number of points

Motivation: Do not need full set of points to achieve same degree of accuracy

<table>
<thead>
<tr>
<th>R</th>
<th>0</th>
<th>1</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>$x$</td>
<td>$y$</td>
</tr>
<tr>
<td>2</td>
<td>$x^2$</td>
<td>$xy$</td>
</tr>
<tr>
<td>3</td>
<td>$x^3$</td>
<td>$x^2y$</td>
</tr>
<tr>
<td>4</td>
<td>$x^4$</td>
<td>$x^3y$</td>
</tr>
</tbody>
</table>
Sparse Grid Techniques

**Tensored Grids:** Exponential growth

```
<table>
<thead>
<tr>
<th>p</th>
<th>( R_\ell )</th>
<th>Sparse Grid ( \mathcal{R} )</th>
<th>Tensored Grid ( R = (R_\ell)^p )</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>9</td>
<td>29</td>
<td>81</td>
</tr>
<tr>
<td>5</td>
<td>9</td>
<td>241</td>
<td>59,049</td>
</tr>
<tr>
<td>10</td>
<td>9</td>
<td>1581</td>
<td>&gt; 3 \times 10^9</td>
</tr>
<tr>
<td>50</td>
<td>9</td>
<td>1,719,001</td>
<td>&gt; 5 \times 10^{47}</td>
</tr>
<tr>
<td>100</td>
<td>9</td>
<td>1,353,801</td>
<td>&gt; 2 \times 10^{95}</td>
</tr>
</tbody>
</table>
```
Reduced-order Models

Based on Projections: e.g., Evolution model: for all \( v \in V \)

\[
\int_D \frac{\partial u}{\partial t} v dx + \int_D N(u, q) S(v) dx = \int_D F(q) v dx
\]

Full-Order Approximate Solution

\[
u^J(t, x; q) = \sum_{j=1}^{J} u_j(t) \phi_j(x)
\]

satisfies

\[
\int_D \frac{\partial u^J}{\partial t} \phi_\ell dx + \int_D N(u^J, q) S(\phi_\ell) dx = \int_D F(q) \phi_\ell dx
\]

on high-dimensional space \( V^J = \text{span}(\phi_j) \).

Goal: Construct reduced-order solution

\[
u^{J_r}(t, x; q) = \sum_{j=1}^{J_r} u_j(t) \phi_j(x)
\]

on low-dimensional space \( V^{J_r} \) where \( J_r << J \)

Note:

- Basis construction crux of method.
- Eigenfunctions
- Proper Orthogonal Decomposition (POD)
Reduced-order Models

Example: Eigenfunctions for heat equation

\[ \frac{\partial T}{\partial t} = \alpha \frac{\partial^2 T}{\partial x^2}, \quad 0 < x < L, \, t > 0 \]

\[ T(t,0) = T(t,L) = 0, \quad t > 0 \]

\[ T(0,x) = T_0(x), \quad 0 < x < L \]

Weak Formulation: For all \( \phi \in H^1_0(0,L) \)

\[ \int_0^L \frac{\partial T}{\partial t} \phi \, dx + \alpha \int_0^L \frac{\partial T}{\partial x} \frac{d\phi}{dx} \, dx = 0 \]

Reduced-Order Basis: \( \phi^r_j(x) = \sin \left( \frac{j\pi x}{L} \right) \)

Reduced-Order Model:

\[ T^{J_r}(t,x) = \sum_{j=1}^{J_r} T_j(t) \sin \left( \frac{j\pi x}{L} \right) \]

satisfying

\[ \int_0^L \frac{\partial T^{J_r}}{\partial t} \phi^r_i \, dx + \alpha \int_0^L \frac{\partial T^{J_r}}{\partial x} \frac{d\phi^r_i}{dx} \, dx = 0 \]

Reduced-Order System:

\[ \frac{dT_j}{dt} = -(j\pi/L)^2 T_j \]

\[ T_j(0) = \gamma_j \]
Reduced-order Models

Proper Orthogonal Decomposition (POD):

- Based on snapshots \( v_n = u^J(t_n, x; q) \) or \( u^J(t, x; q^n) \)
- Set typically highly redundant;
- POD extracts coherent structure having largest mean square projection onto set of observations.
- Related to Karhunen-Loeve expansion and singular value decomposition.

Strategy: Seek basis function of the form

\[
\phi(x) = \sum_{i=1}^{N} a_i v_i(x)
\]

where the coefficients \( a_i \) are chosen to maximize

\[
\frac{1}{N} \sum_{i=1}^{N} |\langle v_i, \phi \rangle|^2 \quad \text{subject to } \langle \phi, \phi \rangle = \| \phi \|^2 = 1.
\]

Reduced-Order Basis:

\[
\phi^*_j(x) = \sum_{i=1}^{N} \frac{1}{\sqrt{N \lambda_j}} a_{i}^j v_i(x)
\]

where \((\lambda_j, a_{i}^j)\) are eigenpair
Reduced-order Models

**Strategy:** Seek basis function of the form

\[ \phi(x) = \sum_{i=1}^{N} a_i v_i(x) \]

where the coefficients \( a_i \) are chosen to maximize

\[ \frac{1}{N} \sum_{i=1}^{N} |\langle v_i, \phi \rangle|^2 \quad \text{subject to} \quad \langle \phi, \phi \rangle = \|\phi\|^2 = 1. \]

**Note:**

\[ C(x, y) = \frac{1}{N} \sum_{i=1}^{N} v_i(x) v_i(y) \]

\[ \langle R\phi, \phi \rangle = \frac{1}{N} \sum_{i=1}^{N} |\langle v_i, \phi \rangle|^2 \quad \Rightarrow \quad R \text{ is positive, self-adjoint operator} \]

**Equivalent Problem:** Find largest eigenvalue of

\[ R\phi = \lambda \phi \quad \text{subject to} \quad \|\phi\| = 1 \]

or

\[ \int_{D} C(x, y) \phi(y) dy = \lambda \phi \quad \text{with} \quad \|\phi\| = 1 \]

**Reduced-Order Basis:**

\[ \phi^r_j(x) = \sum_{i=1}^{N} \frac{1}{\sqrt{N\lambda_j}} a_i^j v_i(x) \]

where \((\lambda_j, a_i^j)\) are eigenpair