Lecture -12

MODELING COMPLEX SYSTEMS: SURROGATE MODELING AND DESIGN SPACE REDUCTION

Reference: Book Chapter 15

INTRODUCTION

- *M**athematical approaches to model the behavior of complex systems, in the context of optimizing the system is introduced.***

- **Important challenges** to quantitative optimization of complex systems are
  - **high dimensionality** of the design space
  - **prohibitive computational time** (or expense)
  - **lack of mathematical models**

MODELING

- **Modeling** is one of the primary activities in optimization.
- The design of complex systems, such as aircrafts, cars, and smart-grid networks, are being increasingly performed using **simulation-based design and analysis tools** (e.g., Finite Element Analysis (FEA) and Computational Fluid Dynamics (CFD)).

Example

1-Modern aerospace systems

→ **Design objectives**
- improving performance
- reducing costs
- enhancing the safety

2-optimization of the vehicle structure

→ **Design objectives**
- absorb crash energy
- passenger space
- control crash deceleration pulse

CHALLENGES IN PRACTICAL OPTIMIZATION PROBLEMS

1. **Design-space dimensionality,**
2. **Computational expense, and**
3. **Lack of mathematical models.**

CHALLENGES IN PRACTICAL OPTIMIZATION PROBLEMS

1. *Design-space dimensionality:*

- ❑ Complex problems could involve a very large number of variables

Example:

- nearly 300 variables are involved in designing a vehicle powertrain system
- more than 1000 design variables are involved in designing large-scale wind farms

- ❑ Models that are formulated to operate only on the entire (high-dimensional) variable space in such cases **could pose challenging-to-prohibitive numerical burden on the optimization process**.

CHALLENGES IN PRACTICAL OPTIMIZATION PROBLEMS

2. Computational expense:

- ❑ The behavior of **complex physical systems** are often modeled using computational simulations (e.g., FEA or CFD), which are basically **computer-based implementation** of complex mathematical models.
- ❑ These simulations can take hours (even days) to run - hence, their usage could become **computationally/time-wise prohibitive in the context of optimization**.

Example:

- large eddy simulation (which is a CFD model) is used to model the flow inside a wind farm, it could demand approximately **600 million CPU-hours** for optimizing the configuration of a 25- turbine wind farm

CHALLENGES IN PRACTICAL OPTIMIZATION PROBLEMS

3. *Lack of mathematical models:*

- ❑ The mathematical models that define the underlying physics/behavior of the system (or the dynamics of the problem) **may not even exist** for certain systems.

- ❑ This could be due to the lack of a clear understanding of the underlying physics **or** the relationship between the design parameters and the criteria functions.

Example:

- The **empirical models (based on experimental data)** is used to represent the constitutive relationship of Ti-25V-15Cr-0.2 Si alloy during hot deformation process due to lack of physical models.

CHALLENGES IN PRACTICAL OPTIMIZATION PROBLEMS

- **Design space reduction or design variable linking approach** is used to address the **Design-space dimensionality.** These approaches seek to reduce the design dimensionality of a problem.
- **Mathematical functions** known by names such as metamodeling, surrogate modeling, response surface methodology, and function fitting are used to address the **computational expense** and **lack of mathematical models** in practical optimization problems.

PROBLEM DIMENSIONALITY

- In certain optimization problems, when the number of design variables is large, it is possible to reduce the dimensionality through **design variable linking** by reducing the number of independent design variables.
- **Equality constraints** are used to define this linking. In this technique, one design variable can be expressed in terms of one or more design variables.
- This reduction of the number of design variables is generally intended to **reduce computational cost of optimization**.

DESIGN VARIABLE LINKING

- Assume that an optimization problem is defined by *n*-dimensional design variable vector *x*, where *n* is deemed large and computationally undesirable.
- It may be possible to redefine the optimization problem in terms of a new *m*-dimensional design variable vector ex, where ***m* < *n***, **through a linear transformation defined by the constant matrix T**.
- This transformation can be expressed as

$$x_{n\times 1} = T_{n\times m}\,\widetilde{x}_{m\times 1}$$

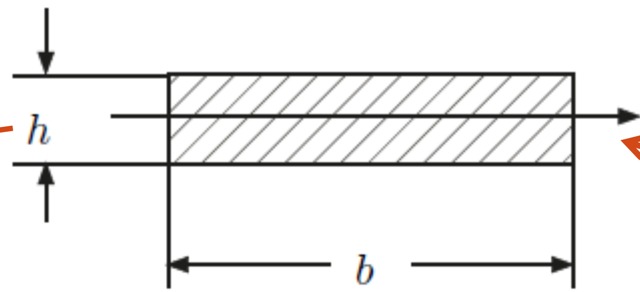
DESIGN VARIABLE LINKING

Example:

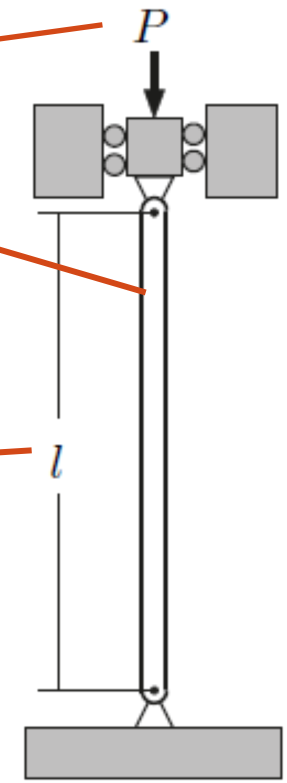
Buckling is a common failure mode when a structure is subjected to high compressive stresses. The objective of this problem is to minimize the volume of a vertical aluminum column is pinned at both ends:

P = 35.5N

h=b/4



Rectangular Cross Section



height of the column is L= 0.5m

- The area moment of inertia of a rectangle is given by $I_0 = \frac{bh^3}{12}$
- The critical force that the column can accept is given by $P_{cr} = \frac{\pi^2 E I_0}{(Kl)^2}$
- The Young's modulus, *E=69 Gpa,* and the effective length factor, *K=1*

DESIGN VARIABLE LINKING

Example:

The optimization problem is formulated as

$$\min f(h,b) = lhb$$

subject to

$$h - 0.25b = 0$$

$$P - P_{cr} \leq 0$$

$$h, b \geq 0$$

- The above formulation of the optimization problem involves **two design variables**. Using **design variable linking**, we can **reduce the number of design variables to one**.
- Using the equality constraint (*h-0.25b=0*), the design variable *h* can be replaced by *h=0.25b*.
- The optimization problem is reduced to: $$\min f(h,b) = 0.25lb^2$$

subject to

$$P - P_{cr} \leq 0$$

$$h, b \geq 0$$

DESIGN VARIABLE LINKING

Example:

The optimization problem is reduced to:

$$\min f(h,b) = 0.25lb^2$$

subject to

$$P - P_{cr} \leq 0$$
$$h, b \geq 0$$

- The optimal solution to the above optimization problem is

b = 0.01m and h = 0.0025m

DESIGN OF EXPERIMENTS

- ***Design of experiments (DoE)*** techniques were originally developed to study the behavior of systems through physical experiments.

- ***DoE*** can be defined as the design of a controlled information gathering exercise for a system or phenomena where variation is known to be present.

- ***DoE*** techniques have existed in some form since the late 1700s, and is considered a discipline with very broad application across the different natural and social sciences, and engineering.

- ***The primary objective of DoE*** is to decide multiple combinations of the controlled parameters (or conditions) at which the experiments will be conducted.

DESIGN OF EXPERIMENTS

- Traditionally controlled experiments referred only to physical experiments, in modern times, it would include **both physical experiments and computational experiments** .

- Once, experiments have been performed for each planned sample condition, the data acquired can be used to
    - investigate a theoretical hypothesis,
    - analyze the system behavior (e.g., sensitivity analysis), and/or
    - **develop empirical or surrogate models to represent the relationship between different system parameters.**

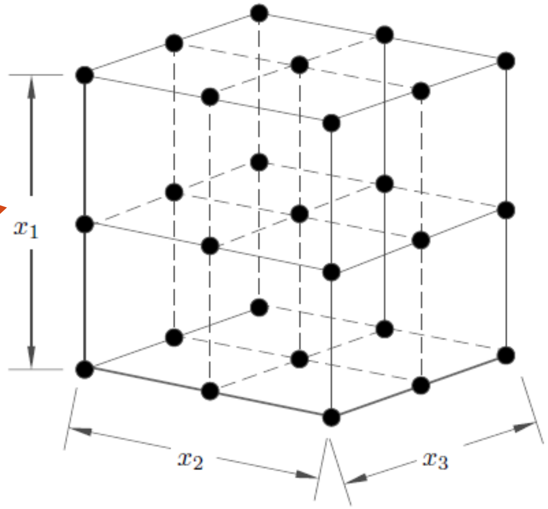
DESIGN OF EXPERIMENTS

- DoE techniques have a large influence on the accuracy of the surrogate model developed thereof.
- **To develop effective surrogate models**, it is necessary to acquire adequate information about the underlying system.
- Assuming <u>no prior knowledge of the system behavior</u>, the typical DoE strategy is to generate a distribution of sample points **throughout the design space in an uniform fashion**.
- There are several techniques available to distribute the sample points, so as to have an adequate coverage of the design space without any particular variable bias including:
    - Factorial design,
    - Central Composite designs , and
    - Latin Hypercube design.

Factorial design

- The most straightforward approach to uniform sampling is the **factorial design method**.
- In this technique, the range of each design variable is divided into **different levels** between the upper and lower limits of a design space.
- In a **full factorial design**, sample points are located at all the combination of the different levels of all the design variables.

A 3-Level and 3-Dimensional **Full Factorial Design** (27 Points)



Factorial design

- In high-dimensional problems, the full factorial design might be **cost/time-wise unaffordable**.

Example:

For example, in a **10 dimension problem**, even a **2-level full factorial design** would require as many as $2^{10} = 1024$ sample points; if we assume that each experiment takes only 1 hr, even then 42 days will be required to run through the entire sample set.

- In such situations, **only a fraction of the sample points can be used** for conducting experiments. Designs that use a fraction of the full factorial design sample points are called **fractional factorial design**.

Factorial design

- The Matlab functions available to perform full factorial design and fractional factorial design are

fullfract → **full factorial design**

fracfact → **Fractional factorial design**

Latin Hypercube design (LHD)

- In a Latin Hypercube design (LHD) or Latin Hypercube Sampling (LHS), the range of each design variable is divided into ***n*** **non-overlapping intervals with equal probability**.

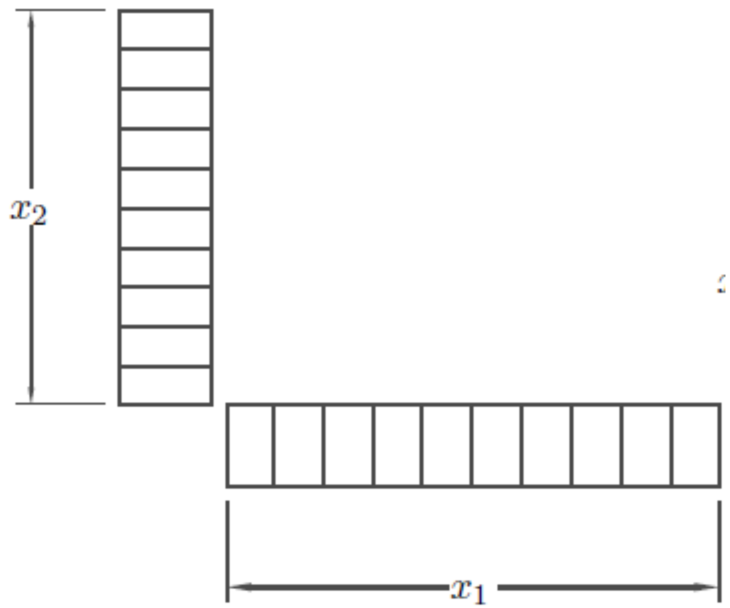
- A sample point is then located randomly on each interval of every design variable.

Latin Hypercube design (LHD)

Example:

It is desired to generate a LHD of size n = 10 for two design variables.

1. The **ten intervals** of the variable x_1 and x_2 in that case are



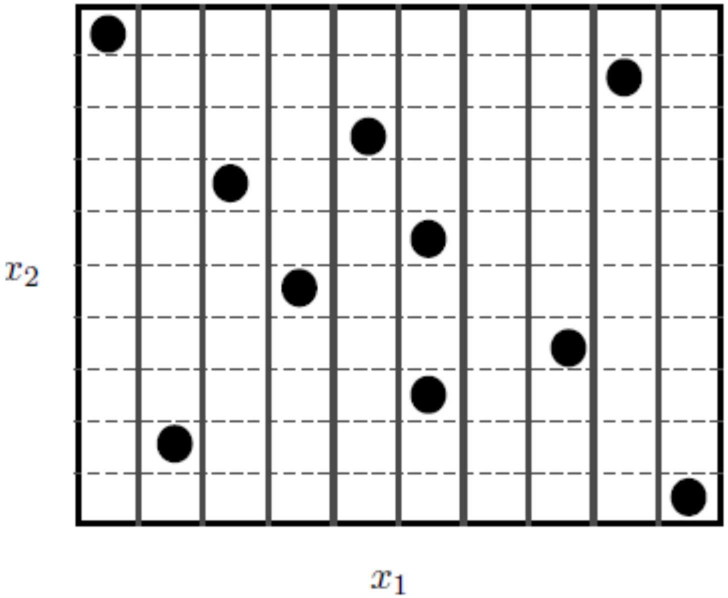
Design of Experiments

Latin Hypercube design (LHD)

Example:

It is desired to generate a LHD of size n = 10 for two design variables.

1. The ten intervals of the variable x_1 and x_2 in that case are
2. The next step is to randomly select specific values for x_1 and x_2 in each of their ten intervals



Latin Hypercube design (LHD)

- **Matlab** provides a set of in-built functions to perform different types of Latin Hypercube Sampling (LHS).

- The most straight-forward implementation of LHS can be performed using **lhsdesign**.

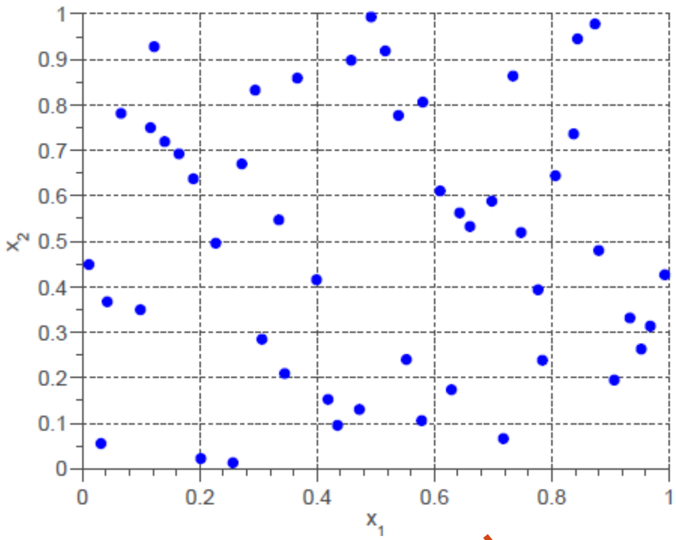
- Direct implementation of this function generates a sample set in the range of [0, 1] for each variable.

Latin Hypercube design (LHD)

Example:

To generate and plot a LHS set for two variables:

```
x = lhsdesign(50,2);
plot(x(:,1),x(:,2))
```



LHS provides a uniform yet non-deterministic coverage of the multi-dimensional design space.

DESIGN OF EXPERIMENTS

- **DoE or effective sampling** is the first step in the development of surrogate models. This is because, surrogate models are **trained** by directly using the **input data generated by the DoE** and the **output data generated by the experiments conducted under that DoE**.

- It is therefore evident that the greater the number of samples or more dense the coverage of the design space, the greater is the expected accuracy of the surrogate models trained thereof.

- Unfortunately, using a generously large number of samples contradicts the very purpose of developing surrogate models, which is **to avoid the unreasonable cost of running too many expensive computational/physical experiments**.

DESIGN OF EXPERIMENTS

- To develop accurate surrogate models, **performing a comprehensive set of simulations or experiments is desirable but often unreasonable.**

- As evident from the discussion of the different DoE methods, <u>the size of the sample set is closely related to the dimension of the problem.</u>

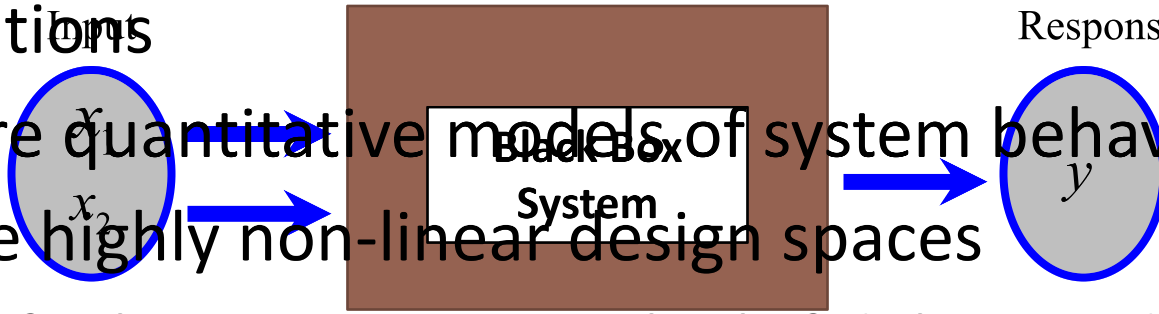
- The DoE techniques that you learned in this section are helpful in **planning effective sample sets** for the surrogate modeling of systems of different design dimensions.

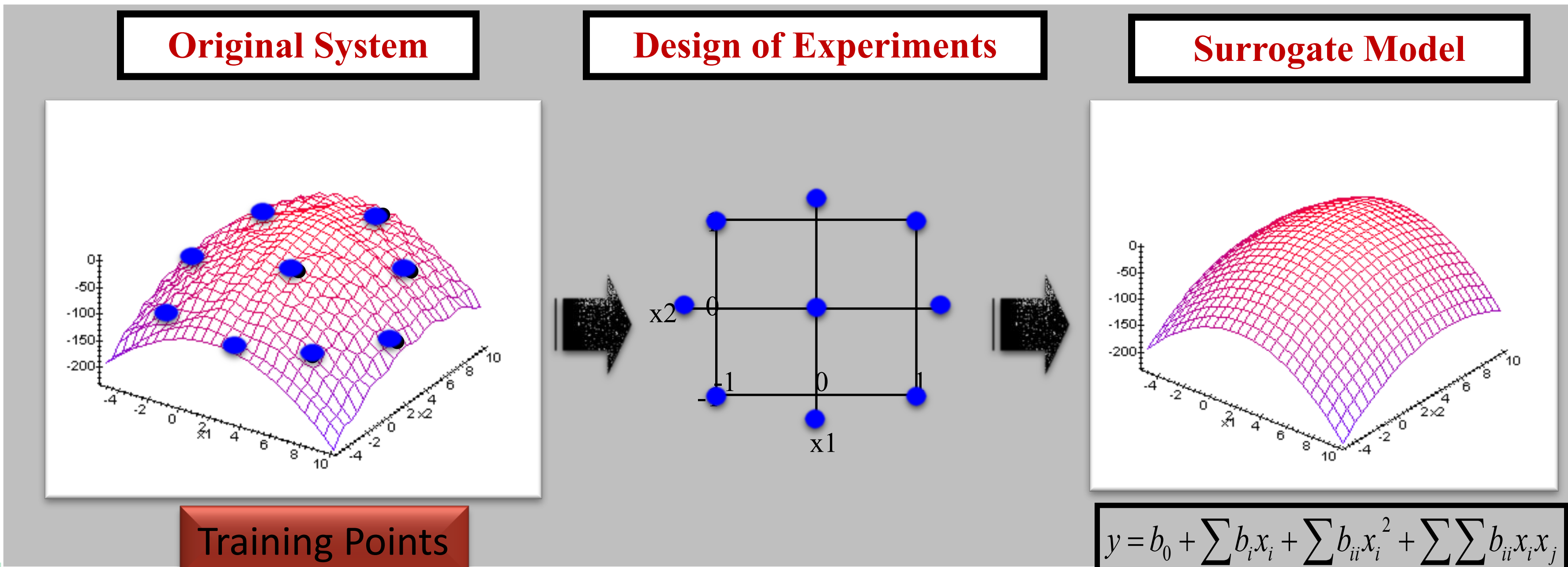
SURROGATE MODELING

- The need to quantify economic and engineering performance of complex systems often demands **highly complex and computationally/time-wise expensive simulations and expensive experiments**.
- The direct use of these computational simulations or experiments in optimization could be anywhere between **challenging to prohibitive**.
- **Surrogate models** are one of the most popular methodologies to deal with this issue, i.e., <u>provide a significantly less expensive and often more tractable alternative towards model-based optimization</u>.
- Surrogate modeling is concerned with **the construction of purely mathematical models** <u>to estimate system performance</u>, or in other words <u>to define relationships between specific system inputs and outputs</u>.

SURROGATE MODELING IN COMPLEX SYSTEMS ENGINEERING

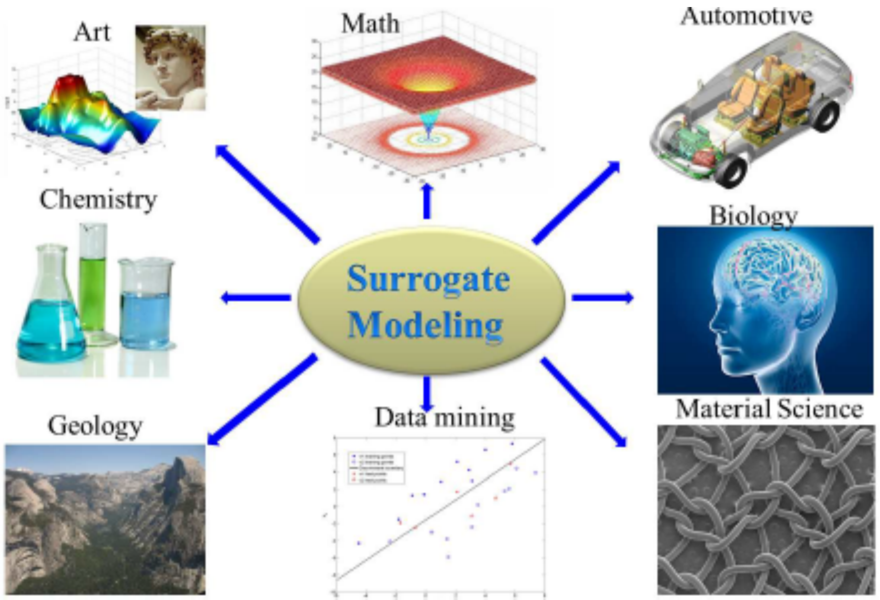
- Reduce computationally expensive physical simulations
- Prepare quantitative models of system behavior
- Handle highly non-linear design spaces
- Quantify the uncertainty in high-fidelity models





SURROGATE MODELING

- Surrogate models are being extensively used in <u>the analysis and optimization of complex systems</u> or in <u>the solution of complex problems</u>.
- **Surrogate modeling techniques have been used for a variety of applications** from multidisciplinary design optimization to the reduction of analysis time and to the improvement of the tractability of complex analysis codes.



Applications of surrogate modeling

SURROGATE MODELING

The general surrogate modeling problem can be stated as follows:

- Given a set of data points $x^i \in R^m, i = 1, \cdots, n_p$;
- the corresponding function values $f(x^i)$

global approximation function $\tilde{f}(x)$

represents the original/actual functional relationship over a given design domain

Surrogate Modeling Process

The process of surrogate modeling generally involves three stages:

(i) design of experiments (DoE);

(ii) construction or training of the surrogate model; and

(iii) model validation.

Surrogate Modeling Process

The process of surrogate modeling generally involves three stages:

(i) design of experiments (DoE);

- The **sample points** used to construct surrogate models are generally called **training points**, and the construction of surrogate models is often called **model training**.
- Once, the training points have been defined (through **DoE** or other sources), the next step is to choose an appropriate surrogate model or functional form.

Surrogate Modeling Process

<u>The process of surrogate modeling generally involves three stages</u>:

(i) design of experiments (DoE);

(ii) construction or training of the surrogate model;

- four major surrogate models are
  - (i) polynomial response surfaces,
  - (ii) radial basis functions (RBF),
  - (iii) kriging, and
  - (iv) artificial neural network,

Polynomial Response Surface Methodology

- The Polynomial Response Surface (PRS) methodology is motivated by the Taylor series expansion.
- A **Taylor series** generally requires an infinite number of terms to obtain the exact value of the real function. The **number of term**s included in a PRS depends on up to what order we want to approximate.
- The approximation is accurate around a chosen point, and become progressively inaccurate as we move away from that point.
- Depending on the form of a real function, the availability of training data, and users' accuracy requirement, PRS can be a linear, second-order, or higher-order polynomial of the vector of design variables.

Polynomial Response Surface Methodology

- In n-dimensional space, variable *x* has ***n*** components:

$$\boldsymbol{x_j}, \qquad \boldsymbol{j = 1, \dots, n}$$

- The linear PRS of n-dimensional variable x has the following form

$$\tilde{f}(x) = a_0 + \sum_{k=1}^{K} a_k x^k$$

$a_j \longrightarrow$ arbitrary coefficients to be determined by training

SURROGATE MODELING

Polynomial Response Surface Methodology

- The most popular PRS is the 2nd-order PRS or the Quadratic Response Surface (QRS). The QRS of an n-dimensional variable x has the following form.

$$\tilde{f}(x) = a_0 + \sum_{j=1}^{n} a_j x_j + \sum_{j=1}^{n} \sum_{i=j}^{n} a_{ji} x_j x_i$$

a_j and a_{ji} are arbitrary coefficients to be determined by training.

Polynomial Response Surface Methodology

- The PRS methodology is frequently used in regression analyses. In regression analyses, the number of training points is generally greater than that of the unknown coefficients, ***a***, since only lower order PRs are used in practice.
- The resulting PRS does not necessarily pass through the training sample data, i.e., does not necessarily have a zero error at the training points.
- One of the approaches to evaluate the unknown coefficients, ***a*** is **the least squares method**.

Polynomial Response Surface Methodology

- The **least squares method** solves regression as an optimization problem.
- The overall solution minimizes the sum of the squares of the errors between the real function values and the corresponding estimated PRS values, over all training points.

Polynomial Response Surface Methodology

Example:

- The following four training points and their corresponding real function values are given

$$f(x_1 = 2) = 4.3, \qquad f(x_2 = 3) = 8.7,$$
$$f(x_3 = 4) = 10.3, \qquad f(x_4 = 5) = 6.7$$

- The second-order response surface of one variable is given as

$$\tilde{f}(x) = a_0 + a_1 x + a_2 x^2$$

Polynomial Response Surface Methodology

Example:

- The parameters, a_0, a_1, and a_2, need to be determined using the **least squares method**.

$$\min_{a_0,a_1,a_2} h(a_0, a_1, a_2) = \sum_{j=1}^{4} \left(\tilde{f}(x_j) - f(x_j)\right)^2$$

- The solution to the optimization problem is $a_0 = -17.6$, $a_1 = 14.9$, and $a_2 = -2$. The squared error is estimated to be **0.29**.

Polynomial Response Surface Methodology

Example:

- To find the maximum value of the fitted function, we need to solve the following optimization problem.

$$\max_{x} \tilde{f}(x) = a_0 + a_1 x + a_2 x^2$$

- The optimal point is ***x = 3.73***, and the corresponding maximum function value is **10.15**.

Polynomial Response Surface Methodology

- The process to fit PRS for practical optimization problems is the same as shown in the example.
- First, the least squares method or other methods are used to fit an appropriate PRS based to the training data.
- The accuracy of the PRS is generally given by the RMSE error.
- The in-built Matlab function, **polyfit** can be used to fit a PRS of a desired order.

Polynomial Response Surface Methodology

- The process of fitting a QRS for the above example using polyfit is shown below.

```
x = [2 3 4 5];
y = [4.3 8.7 10.3 6.7];
P = polyfit(x,y,2)
P =
   -2.0000   14.8800   -17.5800
```

Radial Basis Function

- The idea of using Radial Basis Functions (RBFs) as approximation functions was first proposed by Hardy in 1971, where he used multiquadric RBFs to fit irregular topographical data.
- Radial Basis Function (RBF) expresses surrogate models as linear combinations of a particular **type of basis function** ($\psi(r)$).
- Each constituent basis function is defined in terms of the Euclidean distance (*r*) between a training point and the point of evaluation.
- The Euclidean distance (r) can be expressed as

$$r = \|x - x_i\|$$

point of evaluation

ith training point

Radial Basis Function

- The commonly used non-parametric basis functions are:

1. **Linear:** $\psi(r) = r$,
2. **Cubic:** $\psi(r) = r^3$, and
3. **Thin plate spline**: $\psi(r) = r^2 \ln r$

- The commonly used parametric basis functions are:

1. **Gaussian:** $\psi(r) = \mathrm{e}^{-r^2/(2\delta^2)}$,
2. **Multiquadric**: $\psi(r) = (r^2 + \delta^2)^{1/2}$, and
3. **Inverse multiquadric:** $\psi(r) = (r^2 + \delta^2)^{-1/2}$

Radial Basis Function

- The RBF model is then expressed as a linear combination of the basis functions across all training points

$$\tilde{h}(x) = \sum_{i=1}^{m} w_i \psi(\|x - x_i\|)$$

the generic weights of the basis functions (to be determined by training)

Radial Basis Function

- **The weights**, ***w_i***, are evaluated using all **the training points x_i** and their corresponding **function values $f(x_i)$.**
- Assuming Ψ is used to represent the matrix of the basis function values at the training points, as given by

$$\Psi = \begin{pmatrix} \psi(\|x_1 - x_1\|) & \psi(\|x_1 - x_2\|) & \cdots & \psi(\|x_1 - x_m\|) \\ \psi(\|x_2 - x_1\|) & \psi(\|x_2 - x_2\|) & \cdots & \psi(\|x_2 - x_m\|) \\ \vdots & \vdots & \ddots & \vdots \\ \psi(\|x_m - x_1\|) & \psi(\|x_m - x_2\|) & \cdots & \psi(\|x_m - x_m\|) \end{pmatrix}$$

- The vector ***W*** is defined to represent the vector of the weights:

$$W = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_m \end{pmatrix}$$

Radial Basis Function

- The vector **Y** is defined to represent the vector of the function values at the training points (**f(x$_i$)**), as given below.

$$Y = \begin{pmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_m) \end{pmatrix}$$

- The weights **w$_i$** are then evaluated by solving the following matrix equation.

$$\Psi W = Y$$

- It is important to note that RBFs are essentially interpolating functions, i.e., a trained RBF model will pass through all the training points.

Radial Basis Function

Example:

- The training points and their corresponding real function values are:

$$f(x_1 = 2) = 4.2,\ f(x_2 = 3.5) = 7.1,\ f(x_3 = 4) = 14.8, \text{ and } f(x_4 = 5) = 11.1$$

Fit a RBF model $\tilde{f}(x)$ of a single variable *x*, and find the maximum value of the fitted function in the range [2,5]

In this example, the multiquadric basis function, $\psi(r) = (r^2 + \delta^2)^{1/2}$, is used. The parameter, δ, is set to 0.9..

Radial Basis Function

Example:

- The matrix Ψ of the basis function values at training points, and the vectors ***Y*** and ***W*** are given below.

$$\Psi = \begin{pmatrix} \psi(\|x_1 - x_1\|) & \psi(\|x_1 - x_2\|) & \psi(\|x_1 - x_3\|) & \psi(\|x_1 - x_4\|) \\ \psi(\|x_2 - x_1\|) & \psi(\|x_2 - x_2\|) & \psi(\|x_2 - x_3\|) & \psi(\|x_2 - x_4\|) \\ \psi(\|x_3 - x_1\|) & \psi(\|x_3 - x_2\|) & \psi(\|x_3 - x_3\|) & \psi(\|x_3 - x_4\|) \\ \psi(\|x_4 - x_1\|) & \psi(\|x_4 - x_2\|) & \psi(\|x_4 - x_3\|) & \psi(\|x_4 - x_4\|) \end{pmatrix}$$

$$= \begin{pmatrix} (0^2 + 0.9^2)^{1/2} & (1.5^2 + 0.9^2)^{1/2} & (2.0^2 + 0.9^2)^{1/2} & (3.0^2 + 0.9^2)^{1/2} \\ (1.5^2 + 0.9^2)^{1/2} & (0^2 + 0.01^2)^{1/2} & (0.5^2 + 0.9^2)^{1/2} & (1.5^2 + 0.9^2)^{1/2} \\ (2.0^2 + 0.9^2)^{1/2} & (0.5^2 + 0.9^2)^{1/2} & (0^2 + 0.9^2)^{1/2} & (1.0^2 + 0.9^2)^{1/2} \\ (3.0^2 + 0.9^2)^{1/2} & (1.5^2 + 0.9^2)^{1/2} & (1.0^2 + 0.9^2)^{1/2} & (0^2 + 0.9^2)^{1/2} \end{pmatrix}$$

Radial Basis Function

Example:

- The matrix Ψ of the basis function values at training points, and the vectors ***Y*** and ***W*** are given below.

$$W = \begin{pmatrix} w_1 \\ w_2 \\ w_3 \\ w_4 \end{pmatrix} \qquad Y = \begin{pmatrix} 4.2 \\ 7.1 \\ 14.8 \\ 11.1 \end{pmatrix}$$

Radial Basis Function

Example:

- By Solving the **following matrix equation** for the weights

$$\Psi W = Y$$

we obtain w_1 = -4.947, w_2 = 66.437, w_3 = −82.098, and w_4 = 23.144.

- To find the maximum of the RBF function, we solve the following optimization problem.

$$\max_x \tilde{h}(x) = \sum_{l=1}^{4} w_l \psi(\|x - x_i\|)$$

The optimal point is found to be *x = 4.28,* and the corresponding maximum function value is *16.29*.

Kriging

- Kriging is an approach to approximate irregular data, which is most commonly (but not restricted to be) used as **an interpolating model**.
- The kriging approximation function consists of two components:
  - **(i)** **a global trend function**, and
  - **(ii)** **a deviation function** representing the departure from the trend function.
- The general form of the Kriging surrogate model is given by

$$\tilde{f}(x) = G(x) + Z(x)$$

SURROGATE MODELING

Kriging

$$\tilde{f}(x) = G(x) + Z(x)$$

the unknown function of interest

the trend function, generally a polynomial (e.g., constant, linear, or quadratic).

the realization of a stochastic process with the mean equal to zero, and a nonzero covariance.

- The i, j – th element of the covariance matrix of Z(x) is given as

$$COV[Z(x^i), Z(x^j)] = \sigma_z^2 R_{ij}$$

the process variance

the correlation function between the i^{th} and the j^{th} data points

Surrogate Modeling Process

The process of surrogate modeling generally involves three stages:

(i) design of experiments (DoE);

(ii) construction or training of the surrogate model; and

(iii) model validation.

- Once the surrogate model has been constructed, the final step is to evaluate the performance or expected accuracy of the surrogate model. The two most popular measures of model error are the **root mean squared error (RMSE**) and the **maximum absolute error (MAE).**

Model validation

- **The root mean squared error (RMSE)** is a global error measure, which provides an understanding of the model accuracy over the entire design domain.
- **The maximum absolute error (MAE)** provides an understanding of the maximum local deviations of the model from the actual function.

➢ The most prominent approaches to estimate these error measures are:
  (i) split sample,
  (ii) cross-validation

Model validation

- In **split sample strategy**, the sample data is divided into **training** and **test points**.
- The former is used to <u>construct the surrogate</u>; and the later is used to <u>test the performance of the surrogate</u>.

Model validation

- The **cross-validation technique** operates through the following five steps:
  1. Splits the sample points randomly into q (approximately) equal subsets;
  2. Removes each of these subsets in turn (one at a time);
  3. Trains a intermediate surrogate model to the remaining $q - 1$ subsets;
  4. Computes the error of the intermediate surrogate using the omitted subset;
  5. Once, each one of the q subsets has been used as the omitted subset, the q sets of errors evaluated thereof are generally aggregated to yield a global error measure.

SURROGATE MODELING TOOLS

- SUrrogate MOdeling (SUMO) Toolbox: http://www.sumowiki.intec.ugent.be/index.php/Main_Page
- SURROGATES Toolbox: https://sites.google.com/site/srgtstoolbox/
- UQLab: http://www.uqlab.com/
- Kriging: http://www2.imm.dtu.dk/projects/dace/