Carnegie Mellon

18-660: Numerical Methods for Engineering Design and Optimization

Xin Li
Department of ECE
Carnegie Mellon University
Pittsburgh, PA 15213

Electrical & Computer ENGINEERING

Outline

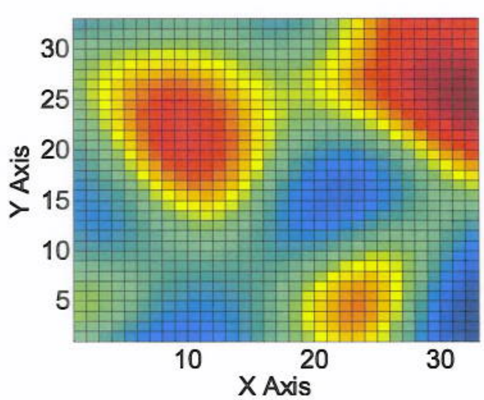
- Project 1: 2-D Thermal Analysis
  - Project overview
  - Thermal analysis review
  - Project details

Project Overview

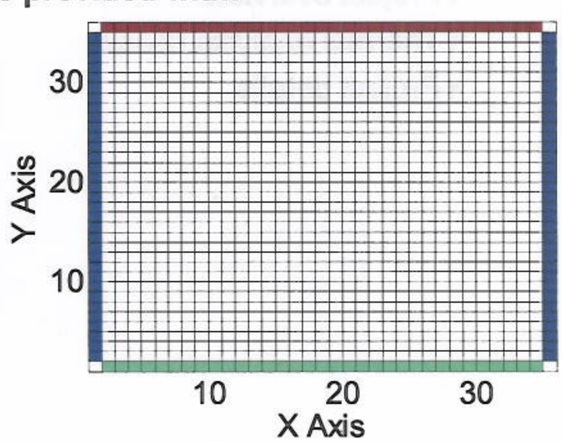
- In this project you are required to write a MATLAB program to simulate a 2-D steady-state thermal problem
- The program mainly consists of the following two steps:
  - Formulate and discretize thermal PDEs
  - Solve the resulting linear systems by both Gaussian elimination and Cholesky factorization

Project Overview

- You will be given 3 test cases
- In each test case, you will be provided with:



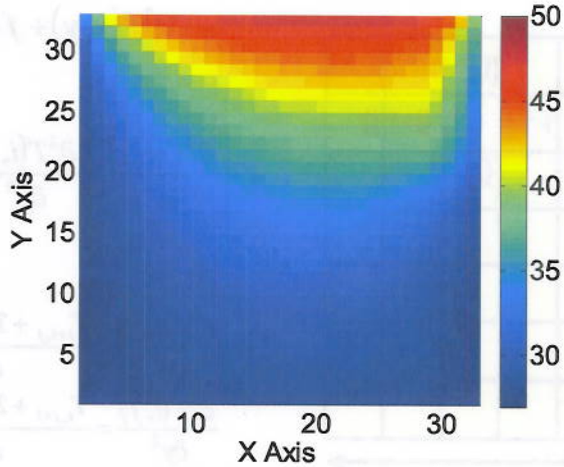
Power density within the medium



Temperature at the boundary

Project Overview

- **Your program will generate:**



Temperature plot within the medium

2-D Heat Equation

- **Heat equation is a 2nd-order linear PDE**

Density — Laplace operator

$$\rho \cdot C_p \cdot \frac{\partial T(x,y,t)}{\partial t} = \kappa \cdot \nabla^2 T(x,y,t) + f(x,y,t)$$

Thermal capacity — Thermal conductivity — Heat source

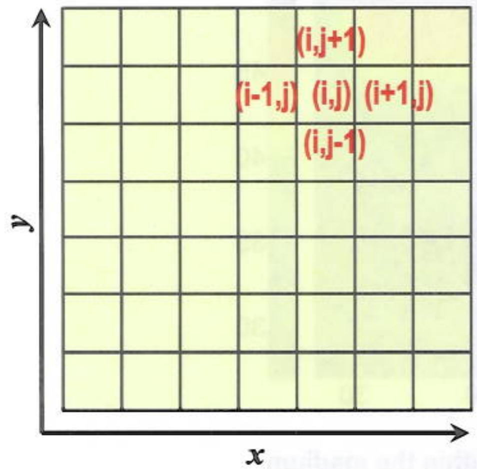
- **We assume that heat conduction has reached a steady state**
  - Heat equation can be simplified as

$$\kappa \cdot \nabla^2 T(x,y) + f(x,y) = 0$$

0 because it is steady state

Equation Discretization

- **Discretize 2-D space into a number of small panels**



$$\kappa \cdot \nabla^2 T(x,y) + f(x,y) = 0$$

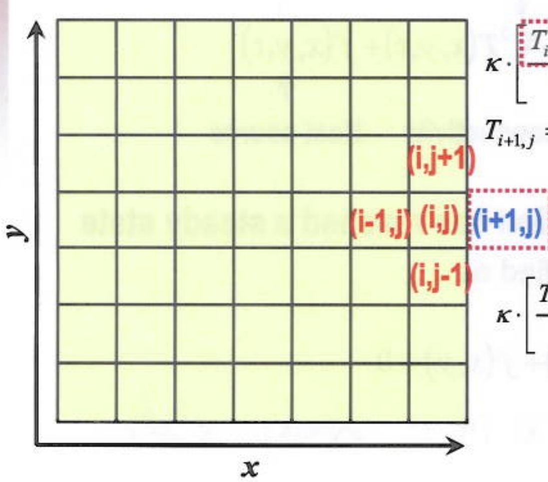
$$\kappa \cdot \left[\frac{\partial^2 T(i,j)}{\partial x^2} + \frac{\partial^2 T(i,j)}{\partial y^2} \right] = -f(i,j)$$

$$\frac{\partial^2 T(i,j)}{\partial x^2} = \frac{T_{i+1,j} + T_{i-1,j} - 2T_{i,j}}{\Delta x^2}$$

$$\frac{\partial^2 T(i,j)}{\partial y^2} = \frac{T_{i,j+1} + T_{i,j-1} - 2T_{i,j}}{\Delta y^2}$$

Boundary Condition

- **We assume a given temperature at the boundary**
  - You need to move the constant term to RHS of the equation



$$\kappa \cdot \left[\frac{\boxed{T_{i+1,j}} + T_{i-1,j} - 2T_{i,j}}{\Delta x^2} + \frac{T_{i,j+1} + T_{i,j-1} - 2T_{i,j}}{\Delta y^2} \right] = -f(i,j)$$

$$T_{i+1,j} = T_c$$

$$\kappa \cdot \left[\frac{T_{i-1,j} - 2T_{i,j}}{\Delta x^2} + \frac{T_{i,j+1} + T_{i,j-1} - 2T_{i,j}}{\Delta y^2} \right] = -f(i,j) - \frac{\kappa \cdot T_c}{\Delta x^2}$$

System of Linear Equations

- **Combine all linear equations**

$$\kappa \cdot \left[\frac{T_{i+1,j} + T_{i-1,j} - 2T_{i,j}}{\Delta x^2} + \frac{T_{i,j+1} + T_{i,j-1} - 2T_{i,j}}{\Delta y^2} \right] = -f(i,j)$$

$$(1 \le i \le N \quad 1 \le j \le M)$$

- **We get a system of linear equations**

$$A \cdot X = B$$

$$X = \left[T_{1,1}, T_{1,2}, \ldots, T_{N,M} \right]^T$$

- **The matrix A is symmetric and positive definite**
  - The linear system can be solved by using either Gaussian elimination or Cholesky factorization

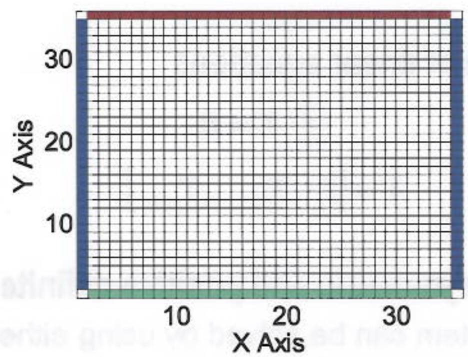
Project Files

- **All files for this project can be found from the distributed package**
  - A report template: Proj1.doc
  - A MATLAB function to generate thermal plot: thermalplot.m
  - Two templates: thermalsimGauss.m, thermalsimCholesky.m
  - Three sets of test data: case1.mat, case2.mat, case3.mat (use "load casex.mat" in MATLAB to import test data)

- **Each mat file contains the following variables:**
  - mediumX: x-dimension of the medium
  - mediumY: y-dimension of the medium
  - p: discretized power density
    p(i,j) means the power density at x = i, y = j
    size(p,1) is the number of panels in x-direction
    size(p,2) is the number of panels in y-direction

Project Data (cont'd)

- leftBound: temperature at the left boundary ($x = 0$)
  leftBound(j) means the temperature at T(0, j)
- rightBound: temperature at the right boundary ($x = N+1$)
- topBound: temperature at the top boundary ($y = M+1$)
- bottomBound: temperature at the bottom boundary ($y = 0$)



- **Thermal conductivity constant: κ = 157 W / m • K**

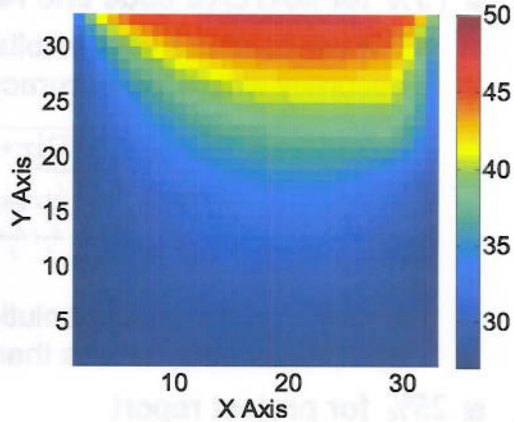
Project Requirements

- **You are required to implement MATLAB functions with the following format (templates included in the project package)**
  - [Temperature] = thermalsimGauss(p, mediumX, mediumY, leftBound, rightBound, topBound, bottomBound);
  - [Temperature] = thermalsimCholesky(p, mediumX, mediumY, leftBound, rightBound, topBound, bottomBound);
  - Temperature is a matrix with the same dimension as p
- **Your program must work on Windows or Linux computer without any modification**
- **You must implement Gaussian elimination and Cholesky factorization by yourself**
  - It is not allowed to use MATLAB functions such as *chol* or *backslash*

Temperature Plot

- **For each test case, you should generate the temperature plot using the following function**

  thermalplot(Temperature);



- **Please include the generated thermal plots in your report**

Project Submission

- **You should zip the MATLAB code into a single file and submit it to the course web site**

- **You should also submit a PDF report (at most 4 pages) to the course web site, including the following items**
  - A high level description of your implementation
  - Your approach for formulating the linear equation
  - Your approach for solving the linear equation
  - Temperature plot of each benchmark
  - Anything else that will make your program unique
  - A WORD template is provided for your project report

Grading Criteria

- **The total points will be distributed as the follows**
- **75% for MATLAB code and results**
  - We will compare your results to our "golden solution" for accuracy where the accuracy will be evaluated by:

$$Error = \sqrt{\frac{\sum_i \sum_j [T^*(i,j) - T(i,j)]^2}{\sum_i \sum_j [T^*(i,j)]^2}}$$

    where T^* is the golden solution and T is your solution. This error value must be less than 0.01
- **25% for project report**
  - You should complete all sections in the report template and clearly address all required points mentioned in the previous slide

Carnegie Mellon

18-660: Numerical Methods for Engineering Design and Optimization

Xin Li
Department of ECE
Carnegie Mellon University
Pittsburgh, PA 15213

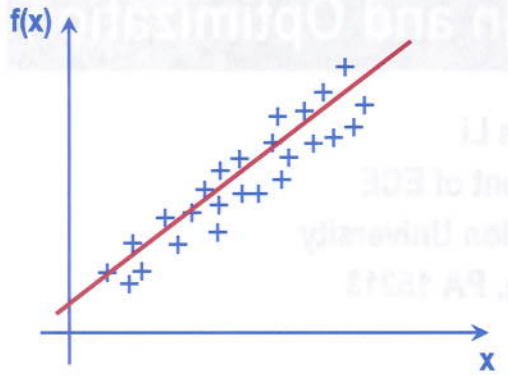
Electrical & Computer ENGINEERING

Overview

- Linear Regression
  - Ordinary least-squares regression
  - Minimax optimization
  - Design of experiments

Linear Regression

- **Linear regression (also referred to as response surface modeling) is widely used for many engineering problems**
  - We do not know the analytical form of f(x)
  - But we can generate a set of sampling points for f(x)
  - Fit an approximate function for f(x) from these sampling points



$f(x) = a_1 b_1(x) + a_2 b_2(x) + \ldots$

a_i: model coefficient

$b_i(x)$: basis function

Linear Regression

- **Major steps of linear regression**
  - Select a model template (e.g., polynomial function)
  - Generate a number of sampling points x
  - Compute performance values at these sampling points $f(x)$
  - Create a set of linear equations to solve model coefficients

- **A simple example**
  - f(x) = exp(x), $x \in [-1, 1]$
  - We will use this simple example to show how we can generally build a regression model from sampling data

Linear Regression Example

- **Step 1: select a model template**

$$f(x) \approx bx + c$$

- **Step 2: generate a number of sampling points**

Samples	1	2	3	4	5
x	-1	-0.5	0	0.5	1

- **Step 3: compute performance values at these sampling points**

Samples	1	2	3	4	5
f(x)	0.3679	0.6065	1.0000	1.6487	2.7183

Linear Regression Example

- **Step 4: create linear equations for model coefficients**

$$f(x) \approx bx + c$$

Samples	1	2	3	4	5
x	-1	-0.5	0	0.5	1
f(x)	0.3679	0.6065	1.0000	1.6487	2.7183

$b \cdot (-1) + c = 0.3679$

$$\underbrace{\begin{bmatrix} -1 & 1 \\ -0.5 & 1 \\ 0 & 1 \\ 0.5 & 1 \\ 1 & 1 \end{bmatrix}}_{\text{x values}} \cdot \begin{bmatrix} b \\ c \end{bmatrix} = \underbrace{\begin{bmatrix} 0.3679 \\ 0.6065 \\ 1.0000 \\ 1.6487 \\ 2.7183 \end{bmatrix}}_{\text{f(x) values}}$$

← i-th sampling point

over-determined

Linear Regression Example

- **Step 5: solve over-determined linear equations**
  - # of equations is greater than # of coefficients – over-determined
  - No exact solution exists to satisfy all equations, but we can find the least-squares solution:

$$A \cdot \alpha = B$$

$$\min_{\alpha} \quad \|A\alpha - B\|_2^2$$

Linear Regression Example

$$A \cdot \alpha = B$$

$$\begin{bmatrix} A \end{bmatrix} \cdot \begin{bmatrix} \alpha \end{bmatrix} - \begin{bmatrix} B \end{bmatrix} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_M \end{bmatrix}$$

ε_i = error at the ith sampling point

$$\min_{\alpha} \quad \|A \cdot \alpha - B\|_2^2 = \|\varepsilon\|_2^2 = \varepsilon_1^2 + \varepsilon_2^2 + \dots + \varepsilon_M^2$$

$\|\varepsilon\|_2$: L2-norm

Linear Regression Example

$$\text{M samples}\left\{\underbrace{\begin{bmatrix} \\ \quad A \quad \\ \\ \end{bmatrix}}_{\text{N coefficients}}\right. \cdot \alpha = \begin{bmatrix} \\ B \\ \\ \end{bmatrix} \qquad \text{(M > N)}$$

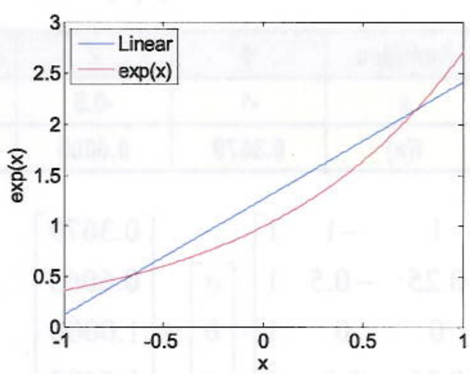
- **There are several possible ways to solve over-determined linear equations for linear regression**
  - We will explain these algorithms in detail in future lectures
  - For now, you can simply use “α = A\B” in MATLAB

Linear Regression Example

- **Step 5: solve over-determined linear equations**

$$\begin{bmatrix} -1 & 1 \\ -0.5 & 1 \\ 0 & 1 \\ 0.5 & 1 \\ 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} b \\ c \end{bmatrix} = \begin{bmatrix} 0.3679 \\ 0.6065 \\ 1.0000 \\ 1.6487 \\ 2.7183 \end{bmatrix}$$

$$b = 1.1486$$
$$c = 1.2683$$



Linear model results in large error

Quadratic Model Example

- **What if we build a quadratic model for y = exp(x)?**
  - Select a model template

$$f(x) \approx ax^2 + bx + c$$

  - Generate a number of sampling points

Samples	1	2	3	4	5
x	-1	-0.5	0	0.5	1

  - Compute performance values at these sampling points

Samples	1	2	3	4	5
f(x)	0.3679	0.6065	1.0000	1.6487	2.7183

Quadratic Model Example

- **Create a set of linear equations to solve model coefficients**

$$f(x) \approx ax^2 + bx + c$$

Samples	1	2	3	4	5
x	-1	-0.5	0	0.5	1
f(x)	0.3679	0.6065	1.0000	1.6487	2.7183

$$\begin{array}{c} \begin{bmatrix} 1 & -1 & 1 \\ 0.25 & -0.5 & 1 \\ 0 & 0 & 1 \\ 0.25 & 0.5 & 1 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0.3679 \\ 0.6065 \\ 1.0000 \\ 1.6487 \\ 2.7183 \end{bmatrix} \\ \downarrow x^2 \quad \downarrow x \qquad\qquad \downarrow \text{f(x) values} \end{array}$$

$a \cdot (-1)^2 + b(-1) + c = 0.3679$

$a - b + c = 0.3679$

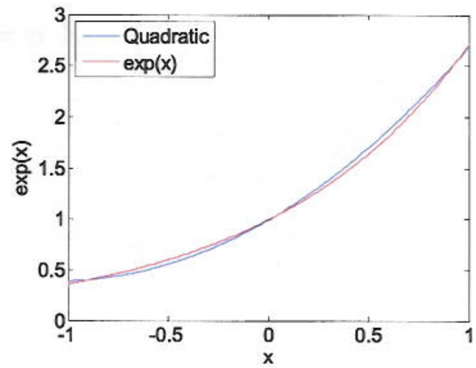
over-determined

Quadratic Model Example

- **Build quadratic model for y = exp(x)**

$$\begin{bmatrix} 1 & -1 & 1 \\ 0.25 & -0.5 & 1 \\ 0 & 0 & 1 \\ 0.25 & 0.5 & 1 \\ 1 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix} = \begin{bmatrix} 0.3679 \\ 0.6065 \\ 1.0000 \\ 1.6487 \\ 2.7183 \end{bmatrix}$$

$$a = 0.5477$$
$$b = 1.1486$$
$$c = 0.9944$$



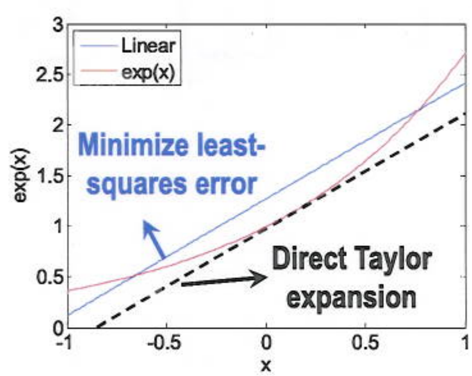
Quadratic model results in much better accuracy in this example

Linear Model vs. Quadratic Model

Linear RSM $\exp(x) \approx 1.1486x + 1.2683$

Quadratic RSM $\exp(x) \approx 0.5477x^2 + 1.1486x + 0.9944$

- **Regression model is different from direct Taylor expansion**
  - E.g., different constant terms in linear and quadratic models – they are selected to minimize the least-squares error



Linear model for exp(x)

Minimax Optimization

- **We can also solve over-determined linear equations to satisfy other optimality criteria (i.e., not ordinary least-squares)**

$$A \cdot \alpha = B$$

$$\begin{bmatrix} A \end{bmatrix} [\alpha] - \begin{bmatrix} B \end{bmatrix} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_M \end{bmatrix} \quad \min_{\alpha} \max_{i} |\varepsilon_i|$$

$$\min_{\alpha} \max_{i} |A(i,:)\alpha - B_i|$$

Minimax Optimization

- **Other optimality criteria can be similarly formulated**

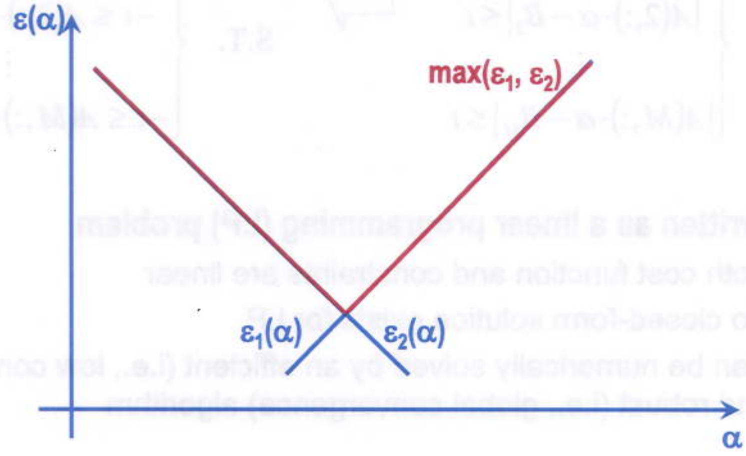
$$A \cdot \alpha = B$$

$$\begin{bmatrix} A \end{bmatrix} [\alpha] - \begin{bmatrix} B \end{bmatrix} = \begin{bmatrix} \varepsilon_1 \\ \varepsilon_2 \\ \vdots \\ \varepsilon_M \end{bmatrix} \quad \min_{\alpha} \max_{i} \left|\frac{\varepsilon_i}{B_i}\right|$$

$$\min_{\alpha} \max_{i} \left|\frac{A(i,:)\alpha - B_i}{B_i}\right|$$

Minimax Optimization

- **General minimax problems are difficult to solve**
  - Cost function does not have continuous derivative



Minimax Optimization

- **However, our minimax problem for regression modeling can be re-formulated into a special form**

- **Consider the example of absolute error minimization**

$$\min_{\alpha} \quad \max_{i} \left| A(i,:)\cdot\alpha - B_i \right|$$

$$\min_{\alpha, t} \quad t$$

$$\text{S.T.} \begin{cases} \left| A(1,:)\cdot\alpha - B_1 \right| \le t \\ \left| A(2,:)\cdot\alpha - B_2 \right| \le t \\ \vdots \\ \left| A(M,:)\cdot\alpha - B_M \right| \le t \end{cases}$$

Minimax Optimization

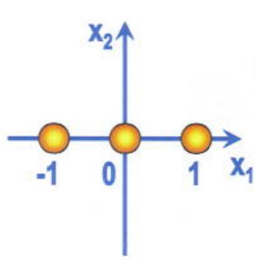
$$\min_{\alpha,t} \quad t \qquad\qquad\qquad\qquad \min_{\alpha,t} \quad t$$

$$\text{S.T.} \quad \begin{cases} |A(1,:)\cdot\alpha - B_1| \le t \\ |A(2,:)\cdot\alpha - B_2| \le t \\ \quad\vdots \\ |A(M,:)\cdot\alpha - B_M| \le t \end{cases} \quad\Rightarrow\quad \text{S.T.} \quad \begin{cases} -t \le A(1,:)\cdot\alpha - B_1 \le t \\ -t \le A(2,:)\cdot\alpha - B_2 \le t \\ \quad\vdots \\ -t \le A(M,:)\cdot\alpha - B_M \le t \end{cases}$$

- **Re-written as a linear programming (LP) problem**
  - Both cost function and constraints are linear
  - No closed-form solution exists for LP
  - Can be numerically solved by an efficient (i.e., low complexity) and robust (i.e., global convergence) algorithm

Design of Experiments (DOE)

- **We already know the basics for linear regression**
- **Open problem:**
  - How can we select few samples to achieve good accuracy?
- **A bad linear model example: $f(x_1, x_2) = a\cdot x_1 + b\cdot x_2 + c$**



$$\begin{aligned} &(x_1 = -1 \quad x_2 = 0 \quad f_1) \\ &(x_1 = 0 \quad x_2 = 0 \quad f_2) \\ &(x_1 = 1 \quad x_2 = 0 \quad f_3) \end{aligned}$$

Sampling points for linear model

Design of Experiments (DOE)

- **Linear model example (continued)**

$$f(x_1, x_2) = a \cdot x_1 + b \cdot x_2 + c$$

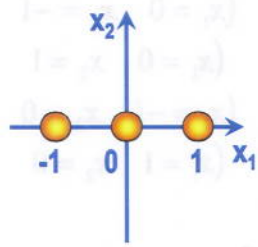
$$(x_1 = -1 \quad x_2 = 0 \quad f_1)$$
$$(x_1 = 0 \quad x_2 = 0 \quad f_2)$$
$$(x_1 = 1 \quad x_2 = 0 \quad f_3)$$

$$\begin{cases} -a + 0 + c = f_1 \\ 0 + 0 + c = f_2 \\ a + 0 + c = f_3 \end{cases}$$

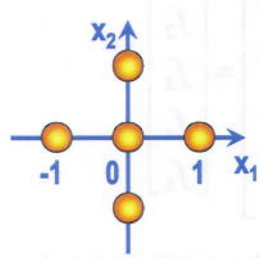
Design of Experiments (DOE)

- **Linear model example (continued)**

$f(x_1, x_2) = a x_1 + b x_2 + c$



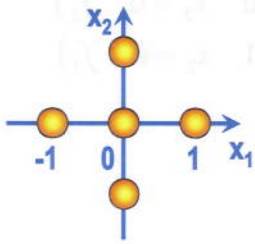
No variation is applied to x_2



Add additional sampling points for x_2

Design of Experiments (DOE)

- A **bad** quadratic model example: $f(x_1, x_2) = a_{11} \cdot x_1^2 + a_{12} \cdot x_1 \cdot x_2 + a_{22} \cdot x_2^2 + b_1 \cdot x_1 + b_2 \cdot x_2 + c$



$$
\begin{array}{l}
(x_1 = 0 \quad x_2 = 0 \quad f_1) \\
(x_1 = 0 \quad x_2 = -1 \quad f_2) \\
(x_1 = 0 \quad x_2 = 1 \quad f_3) \\
(x_1 = -1 \quad x_2 = 0 \quad f_4) \\
(x_1 = 1 \quad x_2 = 0 \quad f_5)
\end{array}
$$

Sampling points for quadratic model

Design of Experiments (DOE)

- Quadratic model example (continued)

$$
\begin{array}{l}
(x_1 = 0 \quad x_2 = 0 \quad f_1) \\
(x_1 = 0 \quad x_2 = -1 \quad f_2) \\
(x_1 = 0 \quad x_2 = 1 \quad f_3) \\
(x_1 = -1 \quad x_2 = 0 \quad f_4) \\
(x_1 = 1 \quad x_2 = 0 \quad f_5)
\end{array}
$$

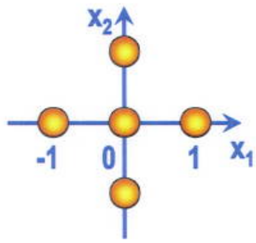
$$f(x_1, x_2) = a_{11} \cdot x_1^2 + a_{12} \cdot x_1 \cdot x_2 + a_{22} \cdot x_2^2 + b_1 \cdot x_1 + b_2 \cdot x_2 + c$$

$$
\begin{array}{c}
\begin{array}{cccccc} x_1^2 & x_1x_2 & x_2^2 & x_1 & x_2 & 1 \end{array} \\
\begin{bmatrix}
0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 1 & 0 & -1 & 1 \\
0 & 0 & 1 & 0 & 1 & 1 \\
1 & 0 & 0 & -1 & 0 & 1 \\
1 & 0 & 0 & 1 & 0 & 1
\end{bmatrix}
\end{array}
\cdot
\begin{bmatrix} a_{11} \\ a_{12} \\ a_{22} \\ b_1 \\ b_2 \\ c \end{bmatrix}
=
\begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \end{bmatrix}
$$

Singular matrix (cannot solve the coefficient a_{12})

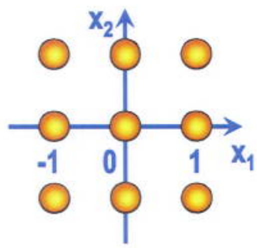
Design of Experiments (DOE)

- **Quadratic model example (continued)**



$\alpha_{12} x_1 x_2$

Cross-product terms cannot be captured



Add additional sampling points for x_1x_2

Design of Experiments (DOE)

- **Design of experiments (DOE) is a research area that studies how to optimally select sampling points for modeling**

- **Given a model template (e.g., linear or quadratic function), optimize sampling points for certain optimal criterion**
  - E.g., maximize modeling accuracy

- **Numerical optimization may be required to find the optimal sampling scheme**

D. Montgomery, Design and Analysis of Experiments, John Wiley & Sons, 2004

Summary

- **Linear regression**
  - Ordinary least-squares regression
  - Minimax optimization
  - Design of experiments