Introduction to Design Optimization

16.90

12 May, 2014

Today's Topics

- Unconstrained optimization algorithms (cont.)
- Computing gradients
- The 1D search in an optimization algorithm
- Surrogate models
- Least squares fitting of a response surface

Design Optimization Problem Statement

The design problem may be formulated as a problem of Nonlinear Programming (NLP)

$$\min \mathbf{J}\left(\mathbf{x},\mathbf{p}\right)$$

$$\text{s.t.} \quad \mathbf{g}(\mathbf{x},\mathbf{p}) \le 0$$

$$\mathbf{h}(\mathbf{x},\mathbf{p}) = 0$$

$$x_{i,LB} \le x_i \le x_{i,UB} \quad \left(i = 1,\ldots,n\right)$$

where

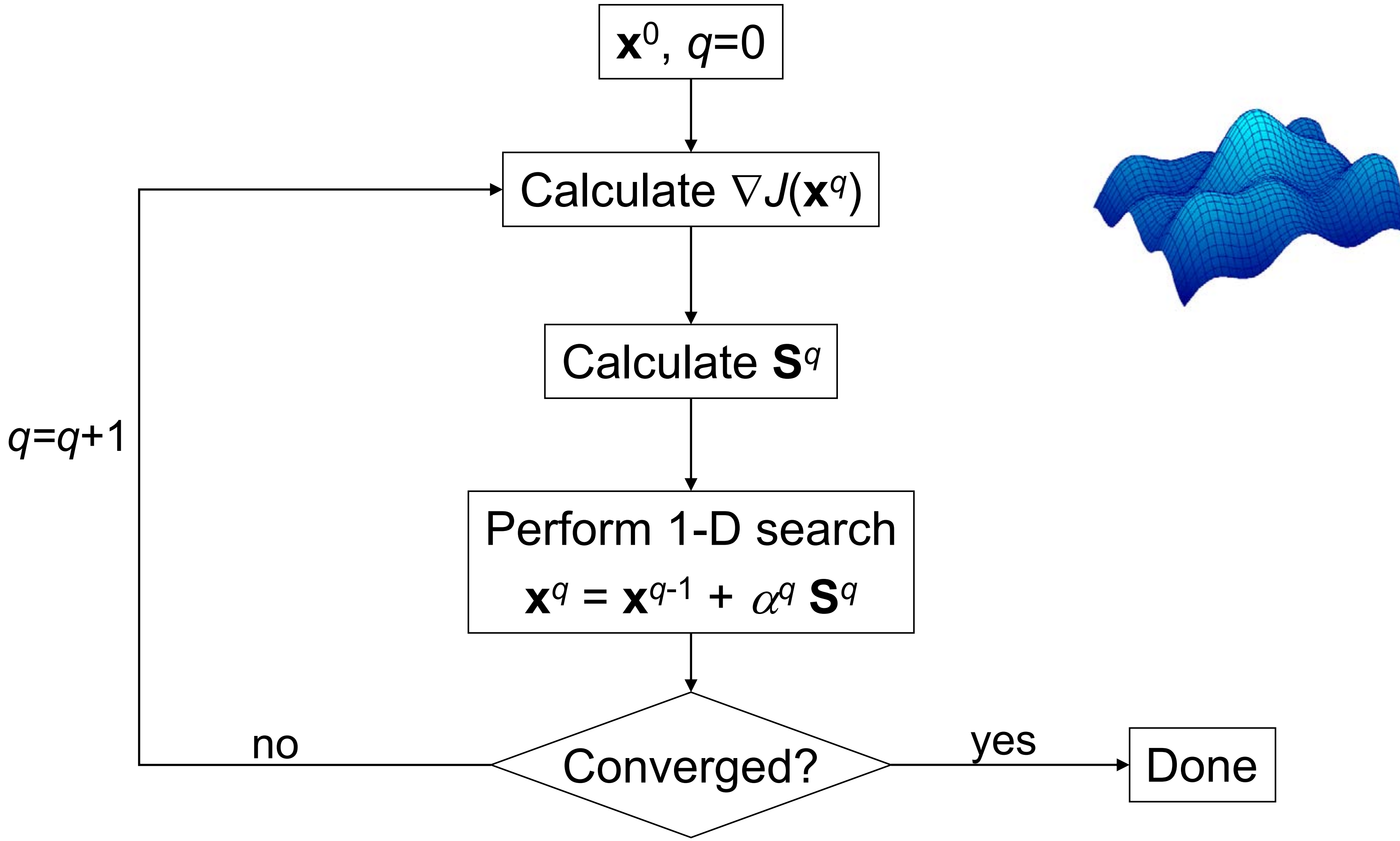
$$\mathbf{J} = \left[J_1\left(\mathbf{x}\right) \quad \cdots \quad J_z\left(\mathbf{x}\right)\right]^T$$

$$\mathbf{x} = \left[x_1 \quad \cdots \quad x_i \quad \cdots \quad x_n\right]^T$$

$$\mathbf{g} = \left[g_1(\mathbf{x}) \cdots g_{m_1}(\mathbf{x})\right]^T$$

$$\mathbf{h} = \left[h_1(\mathbf{x}) \cdots h_{m_2}(\mathbf{x})\right]^T$$

Gradient-Based Optimization Process



Unconstrained Problems: Gradient-Based Optimization Methods

- First-Order Methods
  - use gradient information to calculate **S**
  - steepest descent method
  - conjugate gradient method
  - quasi-Newton methods
- Second-Order Methods
  - use gradients and Hessian to calculate **S**
  - Newton method

- Often, a constrained problem can be cast as an unconstrained problems and these techniques used.

Steepest Descent

$$\boldsymbol{S^q = -\nabla J(x^{q-1})}$$

$-\nabla J(\mathbf{x})$ is the direction of max decrease of J at $\mathbf{x}$

Algorithm:

choose $\mathbf{x}^0$, set $\mathbf{x}=\mathbf{x}^0$

repeat until converged:

$\mathbf{S} = -\nabla J(\mathbf{x})$

choose α to minimize $J(\mathbf{x}+\alpha\mathbf{S})$

$\mathbf{x} = \mathbf{x} + \alpha\mathbf{S}$

- doesn't use any information from previous iterations
- converges slowly
- α is chosen with a 1-D search (interpolation or Golden section)

Conjugate Gradient

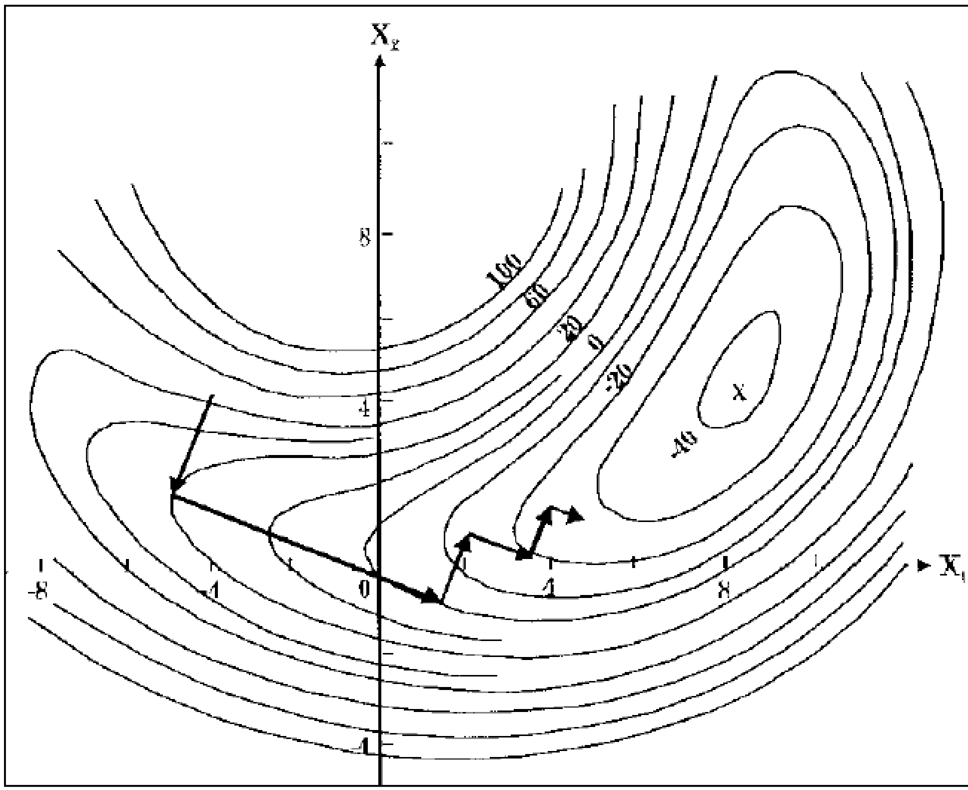
$$\mathbf{S}^1 = -\nabla J(\mathbf{x}^0)$$
$$\mathbf{S}^q = -\nabla J(\mathbf{x}^{q-1}) + \beta^q \mathbf{S}^{q-1}$$

$$\beta^q = \frac{\left|\nabla J(\mathbf{x}^{q-1})\right|^2}{\left|\nabla J(\mathbf{x}^{q-2})\right|^2}$$

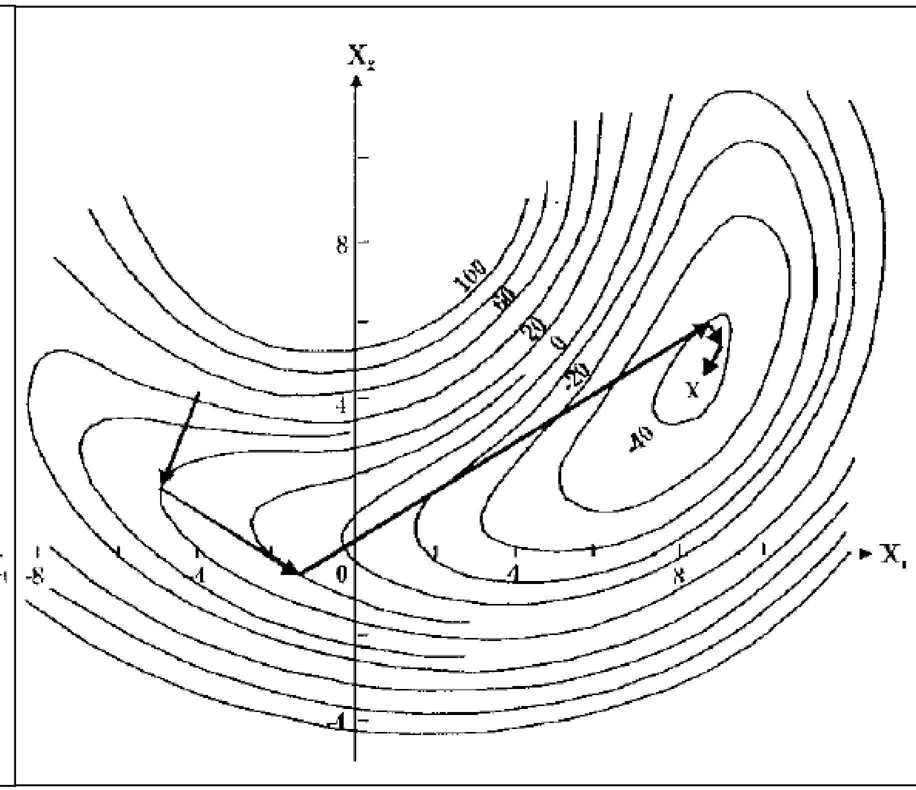
- search directions are now conjugate
- directions $\mathbf{S}^j$ and $\mathbf{S}^k$ are conjugate if $\mathbf{S}^{jT}\,\mathbf{H}\,\mathbf{S}^k = 0$ (also called H-orthogonal)
- makes use of information from previous iterations without having to store a matrix

Geometric Interpretation

Steepest descent



Conjugate gradient



Figures from “Optimal Design in Multidisciplinary Systems,” AIAA Professional Development Short Course Notes, September 2002.

Newton's Method

Taylor series:

$$J(\mathbf{x}) \approx J(\mathbf{x}^0) + \nabla J(\mathbf{x}^0)^T \delta\mathbf{x} + \frac{1}{2}\delta\mathbf{x}^T \mathbf{H}(\mathbf{x}^0)\delta\mathbf{x}$$

where $\delta\mathbf{x} = \mathbf{x} - \mathbf{x}^0$

differentiate: $\nabla J(\mathbf{x}) \approx \nabla J(\mathbf{x}^0) + \mathbf{H}(\mathbf{x}^0)\delta\mathbf{x}$

at optimum $\nabla J(\mathbf{x}^*)=0$

$$\Rightarrow \nabla J(\mathbf{x}^0) + \mathbf{H}(\mathbf{x}^0)\delta\mathbf{x} = 0$$

$$\boxed{\delta\mathbf{x} = -\left[\mathbf{H}(\mathbf{x}^0)\right]^{-1} \nabla J(\mathbf{x}^0)}$$

Newton's Method

$$\mathbf{S} = -\left[\mathbf{H}(\mathbf{x}^0)\right]^{-1} \nabla J(\mathbf{x}^0)$$

- if $J(\mathbf{x})$ is quadratic, method gives exact solution in one iteration
- if $J(\mathbf{x})$ not quadratic, perform Taylor series about new point and repeat until converged
- a very efficient technique if started near the solution
- **H** is not usually available analytically, and finite difference is too expensive ($n \times n$ matrix)
- **H** can be singular if J is linear in a design variable

Quasi Newton

$$\mathbf{S}^q = -\mathbf{A}^q \nabla J(\mathbf{x}^{q-1})$$

- Also known as variable metric methods
- Objective and gradient information is used to create an approximation to the inverse of the Hessian
- **A** approaches $\mathbf{H}^{-1}$ during optimization of quadratic functions
- Convergence is similar to second-order methods (strictly 1st order)

- Initially: $A = \mathbf{I}$, so $\mathbf{S}^1$ is steepest descent direction

  then: $\mathbf{A}^{q+1} = \mathbf{A}^q + D^q$

  where **D** is a symmetric update matrix

$$\mathbf{D}^q = \text{fn}(\mathbf{x}^q - \mathbf{x}^{q-1}, \nabla J(\mathbf{x}^q) - \nabla J(\mathbf{x}^{q-1}), \mathbf{A}^q)$$

- Various methods to determine **D**

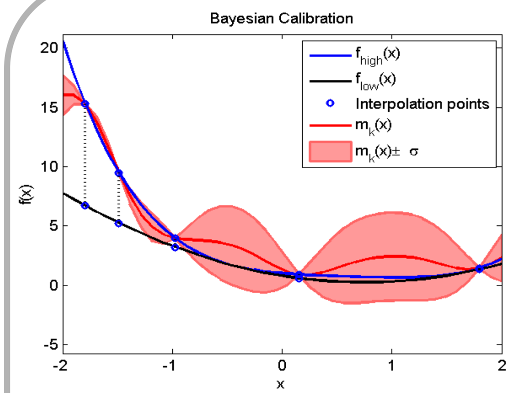
  e.g., Davidon-Fletcher-Powell (DFP)

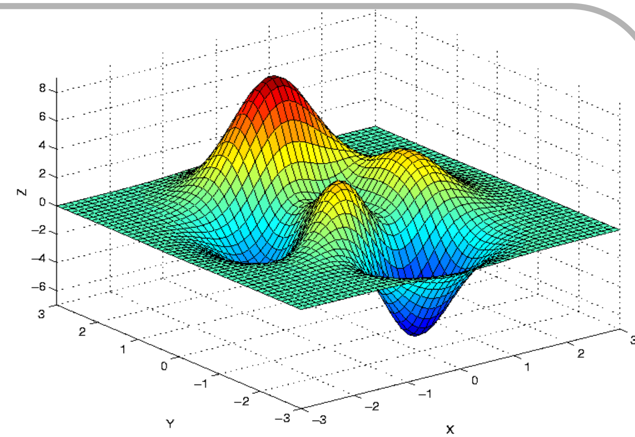
  Broydon-Fletcher-Goldfarb-Shanno (BFGS)

Computing Gradients Using Finite Difference Approximation

The 1D Search

Surrogate Models

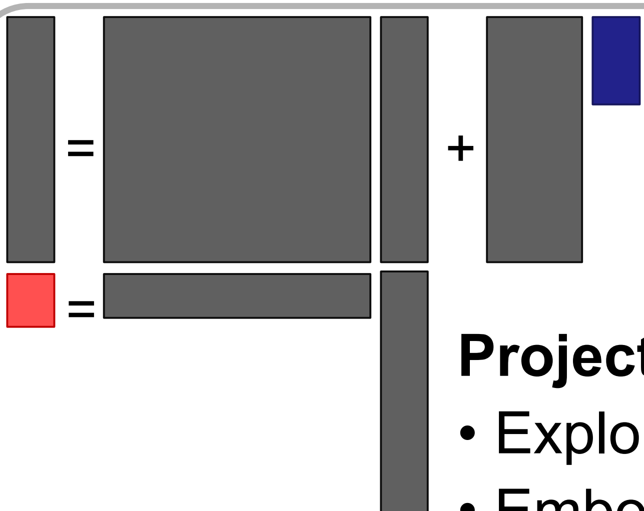


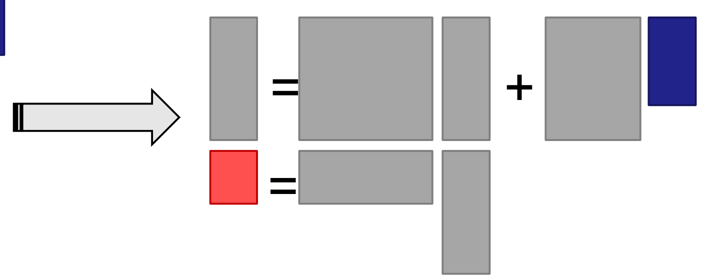


Data-fit models

from Choi et al.

Simplified physics models





Projection-based reduced models

- Exploit problem structure
- Embody underlying physics

Data Fit Methods

- Sample the simulation at some number of design points
  - Use DOE methods (e.g., Latin hypercube) to select the points
- Fit a surrogate model using the sampled information
- Surrogate may be global (e.g., quadratic response surface) or local (e.g., Kriging interpolation)
- Surrogate may be updated adaptively by adding sample points based on surrogate performance (e.g., Efficient Global Optimization, EGO)

Polynomial Response Surface Method

- Surrogate model is a local or global polynomial model
- Can be of any order
  - Most often quadratic; higher order requires many samples
- Advantages: Simple to implement, visualize, and understand, easy to find the optimum of the response surface
- Disadvantages: May be too simple, doesn’t capture multimodal functions well

Global Polynomial Response Surface

- Fit objective function with a polynomial
- *e.g.,* quadratic approximation to a function of n design variables $x_1, x_2, \ldots, x_n$

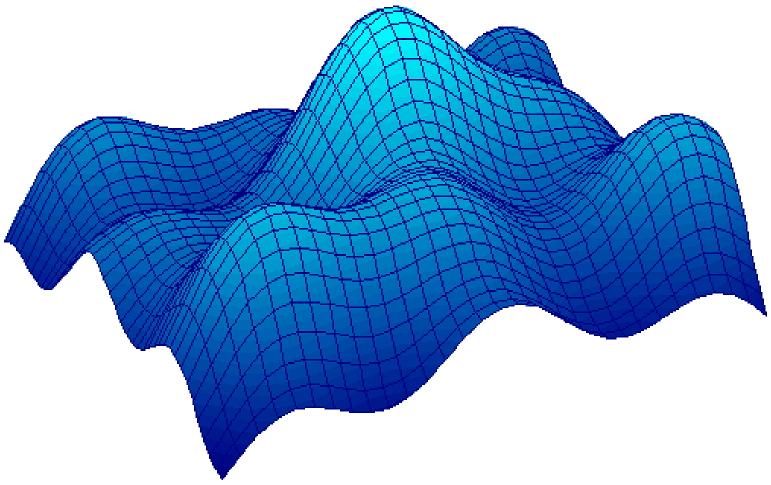
$$
\begin{aligned}
J(\mathbf{x}) \quad = \quad & c_0 + c_1 x_1 + c_2 x_2 + \ldots + c_n x_n \\
& + c_{n+1} x_1^2 + c_{n+2} x_1 x_2 + \ldots + c_{p-1} x_n^2
\end{aligned}
$$

- Coefficients determined using a least squares fit to available data
- Update model by including a new function evaluation then doing least squares fit to compute the new coefficients

Fitting a Polynomial Response Surface

Polynomial Response Surface Method

Matlab demo: Peaks function



Summary

- From this lecture and the online notes you should:
  - Have an understanding of how a design problem can be posed as an optimization problem
  - Have a basic understanding of the steps in the gradient-based unconstrained optimization algorithms
  - Be able to estimate gradient and Hessian information using finite difference approximation
  - Understand how to construct a polynomial response surface using least-squares regression and how to measure the quality of fit.