Lecture 11: Successive Quadratic Programming (SQP) Methods

Prof. Krishna R. Pattipati

Dept. of Electrical and Computer Engineering
University of Connecticut
Contact: krishna@engr.uconn.edu (860) 486-2890

ECE 6437
Computational Methods for Optimization

Fall 2009
November 3 & 10 , 2009

Outline of Lecture 11

- Motivation for Successive Quadratic Programming (SQP) Methods
- Key SQP Ideas
- Newton Version of SQP
- Descent Property of Merit Function $f + cP$
- Quasi-Newton Version of SQP
- SQP with second order correction

Motivation for SQP - 1

- Consider unconstrained minimization problem: $\min_{\underline{x}} f(\underline{x})$
  - Given the current estimate $\underline{x}_k$ the next estimate $\underline{x}_{k+1}$ is obtained via a quadratic approximation of $f(\underline{x}^*)$ around $\underline{x}_k$:

$$f(\underline{x}^*) \approx f(\underline{x}_k) + \nabla \underline{f}^T(\underline{x}_k)(\underline{x}^* - \underline{x}_k) + \frac{1}{2}(\underline{x}^* - \underline{x}_k)^T \nabla^2 f(\underline{x}_k)(\underline{x}^* - \underline{x}_k) \quad \text{min at } \underline{x}_{k+1} \cong \underline{x}^*$$

$$\underline{x}_{k+1} = \underline{x}_k - [\nabla^2 f(\underline{x}_k)]^{-1} \nabla \underline{f}(\underline{x}_k)$$ "PURE NEWTON ITERATION"

An alternate viewpoint is to consider solving the first order necessary condition :

$$\nabla \underline{f}(\underline{x}^*) = \underline{0}$$

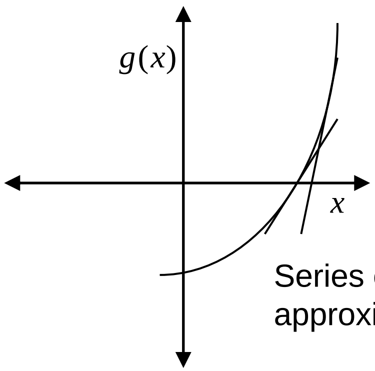
  - Consider scalar iteration first:

Consider

Solving $g(x) = 0$, a scalar non-linear function

$$0 = g(x^*) = g(x_k) + g'(x_k)(x^* - x_k)$$ LINEARIZATION

$$\Rightarrow x_{k+1} = x_k - \frac{g(x_k)}{g'(x_k)}$$



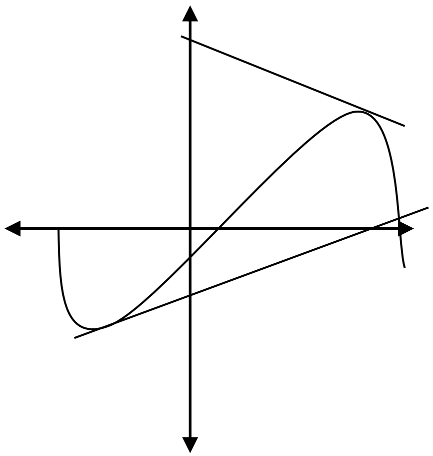
Series of straight line approximations

Motivation for SQP - 2

- Now consider solving the necessary conditions:

$$\nabla \underline{f}(\underline{x}^*) \approx \nabla \underline{f}(\underline{x}_k) + \nabla^2 f(\underline{x}_k)(\underline{x}^* - \underline{x}_k) \approx \underline{0}$$

$$\Rightarrow \boxed{\underline{x}_{k+1} = \underline{x}_k - [\nabla^2 f(\underline{x}_k)]^{-1} \nabla \underline{f}(\underline{x}_k)}$$



Quadratic approximations of $f(\underline{x})$ around $\underline{x}_k \Leftrightarrow$ linearization of first order necessary conditions around $\underline{x}_k$

- Also, know that Newton's method is locally convergent and that we need to modify it via step size selection or trust region approach and employ strategies for indefinite Hessian (e.g., modified Cholesky, Levenberg-Marquardt, double dog-leg, trust region)
- Quasi-Newton methods to avoid having to compute the Hessian ($\Rightarrow$ secant approximation)

SQP for Constrained Optimization

- Can we extend this idea to constrained minimization problems: Yes!
  - Consider $\min f(\underline{x})$ such that $\underline{h}(\underline{x}) = \underline{0}$. Lagrangian function is given as: $L(\underline{x},\underline{\lambda}) = f(\underline{x}) + \underline{\lambda}^T \underline{h}(\underline{x})$. First order necessary conditions of optimality:

$$n \text{ equations: } \nabla_x L(\underline{x}^*,\underline{\lambda}^*) = \nabla \underline{f}(\underline{x}^*) + \nabla \underline{\underline{h}}(\underline{x}^*)\underline{\lambda} = \nabla \underline{f}(\underline{x}^*) + \sum_{i=1}^{m} \lambda_i \nabla \underline{h}_i(\underline{x}^*) = \underline{0}$$

$$m \text{ equations: } \nabla_\lambda L(\underline{x}^*,\underline{\lambda}^*) = \underline{h}(\underline{x}^*) = \underline{0}; \quad (m+n) \text{ unknowns}: \underline{x}^*, \underline{\lambda}^*$$

  - Recall Newton's method for solving a system of non-linear equations:

Given $(\underline{x}_k, \underline{\lambda}_k)$, the current estimates, want to find new estimates $(\underline{x}_{k+1}, \underline{\lambda}_{k+1})$

$$\Rightarrow \nabla_x L(\underline{x}_k,\underline{\lambda}_k) + \nabla^2_{xx} L(\underline{x}_k,\underline{\lambda}_k)(\underline{x}_{k+1} - \underline{x}_k) + \nabla^2_{x\lambda} L(\underline{x}_k,\underline{\lambda}_k)(\underline{\lambda}_{k+1} - \underline{\lambda}_k) = \underline{0}$$

$$\nabla_\lambda L(\underline{x}_k,\underline{\lambda}_k) + \nabla^2_{\lambda x} L(\underline{x}_k,\underline{\lambda}_k)(\underline{x}_{k+1} - \underline{x}_k) = \underline{0}$$

Using: $\nabla_x L(\underline{x}_k,\underline{\lambda}_k) = \nabla \underline{f}(\underline{x}_k) + \nabla \underline{\underline{h}}(\underline{x}_k)\underline{\lambda}_k, \nabla_\lambda L(\underline{x}_k,\underline{\lambda}_k) = \underline{h}(\underline{x}_k)$

$$\nabla^2_{xx} L(\underline{x}_k,\underline{\lambda}_k) = \nabla^2 f(\underline{x}_k) + \sum_{i=1}^{m} (\underline{\lambda}_k)_i \nabla^2 h_i(\underline{x}_k) = H_k$$

$$\nabla^2_{x\lambda} L(\underline{x}_k,\underline{\lambda}_k) = \nabla \underline{\underline{h}}(\underline{x}_k) = [\nabla^2_{\lambda x} L(\underline{x}_k,\underline{\lambda}_k)]^T = N_k$$

$$\Rightarrow \begin{bmatrix} \nabla^2_{xx} L(\underline{x}_k,\underline{\lambda}_k) & \nabla \underline{\underline{h}}(\underline{x}_k) \\ \nabla \underline{\underline{h}}^T(\underline{x}_k) & 0 \end{bmatrix} \begin{bmatrix} \underline{x}_{k+1} - \underline{x}_k \\ \underline{\lambda}_{k+1} - \underline{\lambda}_k \end{bmatrix} = \begin{bmatrix} -\nabla \underline{f}(\underline{x}_k) - \nabla \underline{\underline{h}}(\underline{x}_k)\underline{\lambda}_k \\ -\underline{h}(\underline{x}_k) \end{bmatrix}$$

add $\frac{-1}{c_k} I$ to (2,2) block if N_k is not full rank

Solution of Linearized Equations = QPP

- Let $\underline{x}_{k+1} - \underline{x}_k = \underline{d}_k$ and add $\nabla \underline{\underline{h}}(\underline{x}_k)\underline{\lambda}_k$ to first equation:

$$\begin{bmatrix} \nabla^2_{xx} L(\underline{x}_k, \underline{\lambda}_k) & \nabla \underline{\underline{h}}(\underline{x}_k) \\ \nabla \underline{\underline{h}}^T(\underline{x}_k) & 0 \end{bmatrix} = \begin{bmatrix} \underline{d}_k \\ \underline{\lambda}_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla \underline{f}(\underline{x}_k) \\ -\underline{h}(\underline{x}_k) \end{bmatrix} \quad (*)$$

Claim: These are the necessary conditions of optimality for the following quadratic programming problem:

$$\min_{\underline{d}_k} \frac{1}{2} \underline{d}_k^T \nabla^2_{xx} L(\underline{x}_k, \underline{\lambda}_k) \underline{d}_k + \nabla \underline{f}^T(\underline{x}_k) \underline{d}_k \qquad \text{(QPP)}$$

$$\text{s.t. } \nabla \underline{\underline{h}}^T(\underline{x}_k) \underline{d}_k + \underline{h}(\underline{x}_k) = \underline{0}$$

First order necessary conditions of optimality:

Define: $L(\underline{d}, \underline{\gamma}) = \frac{1}{2} \underline{d}^T \nabla^2_{xx} L(\underline{x}_k, \underline{\lambda}_k) \underline{d} + \nabla \underline{f}^T(\underline{x}_k) \underline{d}_k + \underline{\gamma}^T [\nabla \underline{h}^T(\underline{x}_k) \underline{d}_k + \underline{h}(\underline{x}_k)]$

Optimality Conditions of $(\underline{d}_k, \underline{\gamma}^* = \underline{\lambda}_{k+1})$

$$\left.\begin{aligned} \nabla^2_{xx} L(\underline{x}_k, \underline{\lambda}_k) \underline{d}_k + \nabla \underline{\underline{h}}(\underline{x}_k) \underline{\lambda}_{k+1} &= -\nabla \underline{f}(\underline{x}_k) \\ \nabla \underline{\underline{h}}^T(\underline{x}_k) \underline{d}_k &= -\underline{h}(\underline{x}_k) \end{aligned}\right\} \text{same as } *$$

Summary of SQP Ideas - 1

- Let us summarize results so far and list unresolved issues:

1) Can obtain $\underline{d}_k$ and the multiplier vector $\underline{\lambda}_{k+1}$ form the solution of a quadratic programming problem with linear equality constraints.
2) In essence, we are approximating the nonlinear equality problem by a series of quadratic programming problems, one at each iteration.
3) Again, can get only local convergence. Need strategies for:
   a) Indefinite $\nabla^2_{xx} L \Rightarrow$ Modified Cholesky, Quasi-Newton, Augmented Lagrangian
   b) Global convergence – Line search. Q: Line Search on What?
4) What about inequality constraints?
   - One way of ensuring positive definiteness of $\nabla^2_{xx} L$ is to convexify the Lagrangian by adding a quadratic penalty term:

$$L_c(\underline{x}, \underline{\lambda}) = f(\underline{x}) + \underline{\lambda}^T \underline{h}(\underline{x}) + \frac{1}{2} c \underline{h}^T(\underline{x}) \underline{h}(\underline{x})$$

$$\text{Use } \nabla^2_{xx} L_c(\underline{x}_k, \underline{\lambda}_k) = \nabla^2_{xx} L_o(\underline{x}_k, \underline{\lambda}_k) + \sum_{i=1}^{m} c_i h_i(\underline{x}_k) \nabla^2 h_i(\underline{x}_k) + c_k \nabla \underline{\underline{h}}(\underline{x}_k) \nabla \underline{\underline{h}}^T(\underline{x}_k)$$

Summary of SQP Ideas - 2

- Alternatively, use only

$$\nabla^2_{xx} L_c(\underline{x}_k,\underline{\lambda}_k) \approx \nabla^2_{xx} L_o(\underline{x}_k,\underline{\lambda}_k) + c_k \nabla \underline{\underline{h}}(\underline{x}_k) \nabla \underline{\underline{h}}^T(\underline{x}_k),$$

RHS will be modified as: $-\nabla \underline{f}(\underline{x}_k) \to -[\nabla \underline{f}(\underline{x}_k) + c_k \nabla \underline{\underline{h}}(\underline{x}_k) \underline{h}(\underline{x}_k)]$

- Extension to inequality constraints:

$$\min f(\underline{x}), \quad \text{s.t.} \quad \underline{h}(\underline{x}) = \underline{0}; \; \underline{h} \in R^m; \underline{g}(\underline{x}) \le \underline{0}; \quad \underline{g} \in R^r$$

Lagrangian Function: $L(\underline{x},\underline{\lambda},\underline{\mu}) = f(\underline{x}) + \underline{\lambda}^T \underline{h}(\underline{x}) + \underline{\mu}^T \underline{g}(\underline{x})$

Necessary Conditions:

$$\nabla \underline{f}(\underline{x}^*) + \nabla \underline{\underline{h}}(\underline{x}^*) \underline{\lambda}^* + \nabla \underline{\underline{g}}(\underline{x}^*) \underline{\mu}^* = \underline{0}$$

$$\underline{h}(\underline{x}^*) = \underline{0}$$

$$\mu_i^* g_i(\underline{x}^*) = 0 \qquad i = 1, 2, \ldots, r \;\; (or) \;\; \underline{g}(\underline{x}^*) \le \underline{0}$$

$$\underline{\mu} \ge \underline{0}$$

Linearization leads to:

$$\begin{bmatrix} \nabla^2{}_{xx} L(\underline{x}_k,\underline{\lambda}_k,\underline{\mu}_k) & \nabla \underline{\underline{h}}(\underline{x}_k) & \nabla \underline{\underline{g}}(\underline{x}_k) \\ \nabla \underline{h}^T(\underline{x}_k) & 0 & 0 \\ \nabla \underline{g}^T(\underline{x}_k) & 0 & 0 \end{bmatrix} \begin{bmatrix} \underline{d}_k \\ \underline{\lambda}_{k+1} \\ \underline{\mu}_{k+1} \end{bmatrix} \begin{matrix} = \\ = \\ \le \end{matrix} \begin{bmatrix} -\nabla \underline{f}(\underline{x}_k) \\ -\underline{h}(\underline{x}_k) \\ -\underline{g}(\underline{x}_k) \end{bmatrix}$$

where $\nabla^2{}_{xx} L(\underline{x}_k,\underline{\lambda}_k,\underline{\mu}_k) = \nabla^2 f(\underline{x}_k) + \sum_{i=1}^{m} (\underline{\lambda}_k)_i \nabla^2 h_i(\underline{x}_k) + \sum_{j=1}^{r} (\underline{\mu}_k)_j \nabla^2 g_j(\underline{x}_k)$

Summary of SQP Ideas - 3

This is equivalent to the following QPP with linear equality and inequality constraints :

$$\min_{\underline{d}_k} \; \underline{d}_k^T \nabla^2_{xx} L(\underline{x}_k, \underline{\lambda}_k, \underline{\mu}_k)\underline{d}_k + \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k$$

$$\text{s.t. } \nabla \underline{h}^T(\underline{x}_k)\underline{d}_k + \underline{h}(\underline{x}_k) = \underline{0}$$

$$\nabla \underline{g}^T(\underline{x}_k)\underline{d}_k + \underline{g}(\underline{x}_k) \le \underline{0}$$

Questions:

- How do we use this idea in a general SQP algorithm?
- Need to solve a quadratic programming problem at each iteration. How to solve QPP?
- How to ensure global convergence? Line search on what function?

❑ General Algorithm: Newton Version

Step 1: Given an initial estimate $\underline{x}_0, \underline{\lambda}_0, \underline{\mu}_0$ compute $\nabla^2_{xx} L(\underline{x}_0, \underline{\lambda}_0, \underline{\mu}_0), \nabla \underline{h}(\underline{x}_0), \nabla \underline{g}(\underline{x}_0)$. Set $k = 0$

Step 2: Solve the QPP

$$\left.\begin{aligned} &\min_{\underline{d}_k} \frac{1}{2} \underline{d}_k^T \nabla^2_{xx} L(\underline{x}_k, \underline{\lambda}_k, \underline{\mu}_k)\underline{d}_k + \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k \\ &\text{s.t. } \nabla \underline{h}^T(\underline{x}_k)\underline{d}_k + \underline{h}(\underline{x}_k) = \underline{0} \\ &\nabla \underline{g}^T(\underline{x}_k)\underline{d}_k + \underline{g}(\underline{x}_k) \le \underline{0} \end{aligned}\right\} \Rightarrow \text{RESULT: } \underline{d}_k, \underline{\lambda}_{k+1}, \underline{\mu}_{k+1}$$

General SQP Algorithm

❑ General Algorithm: Newton Version (continued)

Step 3: Select a step size α_k along $\underline{d}_k$ to minimize a penalty (merit) function $= f + cP$

Choices for P: 1. $P = \max\{0, g_1(\underline{x}), g_2(\underline{x}), \dots, g_r(\underline{x}), |h_1(\underline{x})|, |h_2(\underline{x})|, \dots, |h_m(\underline{x})|\}$

2. $P = \sum_{j=1}^{r} \max(0, g_r(\underline{x})) + \sum_{i=1}^{m} |h_i(\underline{x})|$

3. $f + cP = \underbrace{f(\underline{x}) + \underline{\lambda}^T \underline{h}(\underline{x}) + \underline{\mu}^T \underline{g}(\underline{x})}_{L(\underline{x},\underline{\lambda},\underline{\mu})} + \frac{1}{2} c \sum_{i=1}^{r} [\max(0, g_r(\underline{x}))]^2 + \frac{1}{2} c \underline{h}^T(\underline{x}) \underline{h}(\underline{x})$

4. $f + cP = L(\underline{x}, \underline{\lambda}, \underline{\mu}) + \frac{1}{2} c \underline{h}^T(\underline{x}) \underline{h}(\underline{x}) + \frac{1}{2} \| \nabla_x L(\underline{x}, \underline{\lambda}, \underline{\mu}) \|^2$

5. $\beta f + cP = 0. f(\underline{x}) + \frac{1}{2} \nabla_x^T \underline{L}(\underline{x}, \underline{\lambda}) \nabla_x \underline{L}(\underline{x}, \underline{\lambda}) + \frac{1}{2} c \underline{h}^T(\underline{x}) \underline{h}(\underline{x})$

In any case, $\alpha_k = \arg\min_{\alpha} \{ f(\underline{x}_k + \alpha \underline{d}_k) + cP(\underline{x}_k + \alpha \underline{d}_k) \}$

Note that some of the Penalty functions are non-differentiable.

DO NOT USE LINE SEARCH TECHNIQUES THAT USE DERIVATIVE INFO.

ONLY THOSE THAT USE FUNCTION EVALUATIONS (e.g., GS+QI)

Step 4: Check for convergence. If not converged, $k = k + 1$ and go back to Step 2

Descent Property of f+cP - 1

- Descent Property of f+cP

Consider the inequality constrained case: $f + cP = f(\underline{x}) + c\sum_{j=1}^{r}[g_j(\underline{x})]^+$

No loss of generality since $h_i(\underline{x}) = 0 \Rightarrow \begin{cases} h_i(\underline{x}) \le 0 \\ -h_i(\underline{x}) \le 0 \end{cases}$

Proof: $a(\underline{x}) = f(\underline{x}) + c\sum_{j=1}^{r}[g_j(\underline{x})]^+$, Let $J(\underline{x}) = \{j : g_j(\underline{x}) > 0\}$

$$a(\underline{x}+\alpha\underline{d}) = f(\underline{x}+\alpha\underline{d}) + c\sum_{j=1}^{r}[g_j(\underline{x}+\alpha\underline{d})]^+$$

$$= f(\underline{x}) + \alpha\nabla \underline{f}^T(\underline{x})\underline{d} + c\sum_{j=1}^{r}[g_j(\underline{x}) + \alpha\nabla \underline{g}_j^T(\underline{x})\underline{d})]^+ + O(\alpha)$$

$$= f(\underline{x}) + \alpha\nabla \underline{f}^T(\underline{x})\underline{d} + c\sum_{j=1}^{r}[g_j(\underline{x})]^+ + \alpha c\sum_{j\in J(\underline{x})}\nabla \underline{g}_j^T(\underline{x})\underline{d} + O(\alpha)$$

This is because $\nabla \underline{g}_j^T(\underline{x})\underline{d} + g_j(\underline{x}) \le 0 \Rightarrow \nabla \underline{g}_j^T(\underline{x})\underline{d} \le 0$ if $g_j(\underline{x}) \ge 0$

So, $a(\underline{x}+\alpha\underline{d}) = a(\underline{x}) + \alpha[\nabla \underline{f}^T(\underline{x})\underline{d} + c\sum_{j\in J(\underline{x})}\nabla \underline{g}_j^T(\underline{x})\underline{d}]$

Since $c\sum_{j\in J(\underline{x})}\nabla \underline{g}_j^T(\underline{x})\underline{d} \le c\sum_{j\in J(\underline{x})} -g_j(\underline{x}) = -c\sum_{j=1}^{r}[g_j(\underline{x})]^+$

Descent Property of f+cP - 2

- Descent Property of $f+cP$ (continued)

So, $a(\underline{x}+\alpha\underline{d}) \le a(\underline{x}) + \alpha\nabla \underline{f}^T(\underline{x})\underline{d} - c\sum_{j=1}^{r}[g_j(\underline{x})]^+$

From the necessary conditions of optimality, we have

$$\nabla \underline{f}^T(\underline{x})\underline{d} = -\underline{d}^T\nabla_{xx}^2 L\underline{d} - \sum_{j=1}^{r}\mu_j \nabla \underline{g}_j^T(\underline{x})\,\underline{d}$$

$$= -\underline{d}^T\nabla_{xx}^2 L\underline{d} + \sum_{j=1}^{r}\mu_j g_j(\underline{x})$$

$$\boxed{\mu_j[\nabla \underline{g}_j^T(\underline{x})\underline{d} + g_j(\underline{x})] = 0 \Rightarrow -\mu_j\nabla \underline{g}_j^T(\underline{x})\underline{d} = \mu_j g_j(\underline{x})}$$

$$\le -\underline{d}^T\nabla_{xx}^2 L\underline{d} + \max_j(\mu_j)\sum_{j=1}^{r}[g_j(\underline{x})]^+$$

$$a(\underline{x}+\alpha\underline{d}) \le a(\underline{x}) + \alpha\{-\underline{d}^T\nabla_{xx}^2 L\underline{d} - [c - \max(\mu_j)]\sum_{j=1}^{r}[g_j(\underline{x})]^+\} + O(\alpha)$$

Since $\nabla_{xx}^2 L$ is PD and if $c > \max_j \mu_j$, $\exists$ an $\alpha \ni\ a(\underline{x}+\alpha\underline{d}) < a(\underline{x})$

- What if $\nabla_{xx}^2 L$ is not PD? Use Augmented Lagrangian
- If don't want to compute Hessian, use Quasi-Newton Method

Quasi-Newton Version of SQP - 1

General Algorithm: Quasi-Newton Version

Step 1: Given an initial estimate of $\underline{x}_0$ and a PD matrix B_0 (approximation to the Hessian) or its square root L_0 for square root version

Step 2: Solve the QPP:
$$\left.\begin{aligned} &\min \frac{1}{2}\underline{d}_k^T B_k \underline{d}_k + \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k \\ &\text{s.t. } \nabla \underline{\underline{h}}^T(\underline{x}_k)\underline{d}_k + \underline{h}(\underline{x}_k) = \underline{0} \\ &\nabla \underline{\underline{g}}^T(\underline{x}_k)\underline{d}_k + \underline{g}(\underline{x}_k) \le \underline{0} \end{aligned}\right\} \Rightarrow \text{Get } \underline{d}_{k+1}, \underline{\lambda}_{k+1}, \underline{\mu}_{k+1}$$

Step 3: Perform Line search to obtain α_k, where $\alpha_k = \arg\min_{\alpha}\{f + cP\}$

Step 4: Check for convergence. If not go to Step 5

Step 5: Update B_k (or L_k) via generalized BFGS update

$$B_{k+1} = B_k + \frac{\hat{\underline{\omega}}_k \hat{\underline{\omega}}_k^T}{\underline{p}_k^T \hat{\underline{\omega}}_k} - \frac{B_k \underline{p}_k \underline{p}_k^T B_k}{\underline{p}_k^T B_k \underline{p}_k}; \quad \hat{\underline{\omega}}_k = \theta_k \underline{q}_k + (1-\theta_k) B_k \underline{p}_k;$$

$$\boxed{\begin{aligned} \underline{q}_k &= \nabla_x L(\underline{x}_{k+1}, \underline{\lambda}_{k+1}, \underline{\mu}_{k+1}) - \nabla_x L(\underline{x}_k, \underline{\lambda}_{k+1}, \underline{\mu}_{k+1}) \\ \underline{p}_k &= \underline{x}_{k+1} - \underline{x}_k = \alpha_k \underline{d}_k \end{aligned}}$$

suggested value of θ_k (empirical):
$$\theta_k = \begin{cases} 1 \text{ if } \underline{p}_k^T \underline{q}_k \ge 0.2\, \underline{p}_k^T B_k \underline{p}_k \\ \dfrac{0.8\, \underline{p}_k^T B_k \underline{p}_k}{\underline{p}_k^T B_k \underline{p}_k - \underline{p}_k^T \underline{q}_k} \text{ if } \underline{p}_k^T \underline{q}_k \le 0.2\, \underline{p}_k^T B_k \underline{p}_k \end{cases}$$

Quasi-Newton Version of SQP - 2

- General Algorithm: Quasi-Newton Version (continued)

θ_k is used to ensure $\underline{p}_k^T \hat{\underline{\omega}}_k > 0 \Rightarrow \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k < \nabla \underline{f}^T(\underline{x}_{k+1})\underline{d}_k$

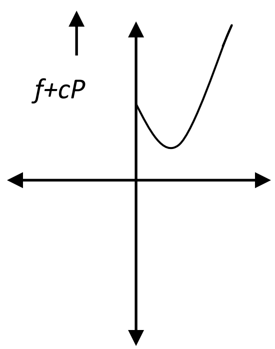
$$\theta_k = 1 \Rightarrow \text{BFGS update } B_{k+1} = B_k + \frac{\underline{q}_k \underline{q}_k^T}{\underline{p}_k^T \underline{q}_k} - \frac{B_k \underline{p}_k \underline{p}_k^T B_k}{\underline{p}_k^T B_k \underline{p}_k}$$

$$\theta_k \neq 1 \Rightarrow \underline{p}_k^T \hat{\underline{\omega}}_k = \frac{0.8\,\underline{p}_k^T B_k \underline{p}_k}{\underline{p}_k^T B_k \underline{p}_k - \underline{p}_k^T \underline{q}_k}\,\underline{p}_k^T \underline{q}_k + \frac{0.2\,\underline{p}_k^T B_k \underline{p}_k - \underline{p}_k^T \underline{q}_k}{\underline{p}_k^T B_k \underline{p}_k - \underline{p}_k^T \underline{q}_k}\,.\underline{p}_k^T B_k \underline{p}_k$$

$$= 0.2\,\underline{p}_k^T B_k \underline{p}_k > 0$$

- Powell (Math Programming, Vol. 15, 1978) shows that if $\alpha_k \approx 1$, the method has super linear convergence. However, one can find problems where $\alpha_k \neq 1$ for $\underline{x}_k$ arbitrarily close to $\underline{x}^*$. Known as Maratos effect

$$f(\underline{x}+\underline{d}) + c\sum_j |g_j^+(\underline{x}+\underline{d})| > f(\underline{x}) + c\sum_j |g_j^+(\underline{x})|$$



Maratos Effect

Maratos Effect

$$\min\ f(\underline{x}) = 2(x_1^2 + x_2^2 - 1) - x_1 \ \textit{subject to}\ (x_1^2 + x_2^2 - 1) = 0$$

$$\text{Optimal solution: } \underline{x}^* = (1,0),\ \lambda^* = 1.5, \nabla^2 L_{xx}(\underline{x}^*, \lambda^*) = I$$

$$\text{At iteration } k,\ \underline{x}_k = (\cos\theta, \sin\theta)^T \Rightarrow \textit{Feasible}$$

$$f(\underline{x}_k) = -\cos\theta;\ \nabla \underline{f}(\underline{x}_k) = \begin{bmatrix} 4\cos\theta - 1 \\ 4\sin\theta \end{bmatrix}, \nabla h(\underline{x}_k) = \begin{bmatrix} 2\cos\theta \\ 2\sin\theta \end{bmatrix}$$

$$QPP: \min\ -\cos\theta + (4\cos\theta - 1)d_1 + 4\sin\theta d_2 + \frac{1}{2}d_1^2 + \frac{1}{2}d_2^2$$

$$\text{subject to: } d_1\cos\theta + d_2\sin\theta = 0$$

$$\underline{d}_k = \begin{bmatrix} \sin^2\theta \\ -\sin\theta\cos\theta \end{bmatrix}; \lambda_{k+1} = \cos\theta \Rightarrow \underline{x}_k + \underline{d}_k = \begin{bmatrix} \cos\theta + \sin^2\theta \\ \sin\theta(1-\cos\theta) \end{bmatrix}$$

$$\text{Can show } \frac{\|\underline{e}_{k+1}\|_2}{\|\underline{e}_k\|_2} = \frac{2\sin^2(\frac{\theta}{2})}{2|\sin(\frac{\theta}{2})|} = |\sin(\frac{\theta}{2})| < 1 \Rightarrow \textit{converging}$$

$$\textit{However},\ f(\underline{x}_k + \underline{d}_k) = \sin^2\theta - \cos\theta > -\cos\theta = f(\underline{x}_k)$$

$$h(\underline{x}_k + \underline{d}_k) = \sin^2\theta > h(\underline{x}_k) = 0$$

Solutions:

1. Use Augmented Lagrangian-based Merit Function

2. Second order correction

3. Allow merit function to increase in some iterations

SQP Algorithm with 2nd Order Correction - 1

- Solve two quadratic programs to improve convergence rate:

$$(1)\ \min \nabla \underline{f}^T(\underline{x}_k)\underline{d}_k + \frac{1}{2}\underline{d}_k^T B_k \underline{d}_k \qquad (2)\ \min \frac{1}{2}\underline{p}_k^T \underline{p}_k$$

$$\text{s.t. } \underline{h}(\underline{x}_k) + \nabla \underline{\underline{h}}^T(\underline{x}_k)\underline{d}_k = \underline{0} \qquad \text{s.t. } \underline{h}(\underline{x}_k + \underline{d}_k) + \nabla \underline{\underline{h}}^T(\underline{x}_k)\underline{p}_k = \underline{0}$$

$$\underline{g}(\underline{x}_k) + \nabla \underline{\underline{g}}^T(\underline{x}_k)\underline{d}_k \le \underline{0} \qquad \underline{g}(\underline{x}_k + \underline{d}_k) + \nabla \underline{\underline{g}}^T(\underline{x}_k)\underline{p}_k \le \underline{0}$$

$$\underline{x}_{k+1} = \underline{x}_{k+1} + \alpha_k \underline{d}_k + \alpha_k^2 \underline{p}_k ; \alpha_k = \arg\min_{\alpha}(f + cP)$$

- Solution of QPP: At step k, we have

$$\min_{\underline{d}} \underline{g}_k^T \underline{d} + \frac{1}{2}\underline{d}^T B_k \underline{d} \qquad \underline{g}_k \sim \nabla \underline{f}(\underline{x}_k)$$

$$B_k \sim \nabla_{xx}^2 L,\ QN,\ \nabla_{xx}^2 L + c_k \nabla \underline{\underline{h}} \nabla \underline{\underline{h}}^T$$

$$\text{s.t. } A_1^T \underline{d} = \underline{b}_1 \qquad A_1 \sim \nabla \underline{\underline{h}},\ A_2 \sim \nabla \underline{\underline{g}}$$

$$A_2^T \underline{d} \le \underline{b}_2 \qquad \underline{b}_1 = -\underline{h},\ \underline{b}_2 = -\underline{g}$$

Suppose we have a feasible point $\underline{d}_l$,

$$\underline{d}_l \ni \left.\begin{array}{l} A_1^T \underline{d}_l = \underline{b}_1 \\ A_2^T \underline{d}_l \le \underline{b}_2 \end{array}\right\} \Rightarrow$$

Solve Phase I LP

$$\min \sum_{i=1}^{m} z_i + \sum_{j=1}^{r} y_j$$

$$\text{s.t. } A_1^T \underline{d}_l + \underline{z} = \underline{b}_1,\ A_2^T \underline{d}_l + \underline{y} = \underline{b}_2, \underline{y} \ge \underline{0}$$

SQP Algorithm with 2nd Order Correction - 2

- Solution of QPP (continued):

At $\underline{d}_l$: Equality constraints are satisfied and some inequality constraints

$$\text{Define } \hat{A}^T = \begin{bmatrix} A_1^T \\ \hat{A}_2^T \end{bmatrix} \begin{matrix} m \text{ equality} \\ r^* \text{ active inequality} \end{matrix}$$

$$\text{At optimum } \underline{g}_k + B_k \underline{d}^* = -A_1 \underline{\lambda}^* - \hat{A}_2 \underline{\mu}^*$$

If we know active constraints at $\underline{d}^*$, we can actually solve an equality constrained

$$\text{problem: } \min \ \underline{g}_k^T \underline{d} + \frac{1}{2} \underline{d}^T B_k \underline{d} \quad \text{s.t. } \hat{A}^T \underline{d} = \hat{\underline{b}}; \ \hat{\underline{b}} = \begin{pmatrix} \underline{b}_1 \\ \hat{\underline{b}}_2^* \end{pmatrix}$$

Unfortunately don't know r^*, so our procedure is iterative:

Repeat until Convergence

- Start with the current working set S_l
- Go to the next point $\underline{d}_{l+1} = \underline{d}_l + \underline{p}_l$
- See if we need to update $S_l \to S_{l+1}$

SQP Algorithm with 2nd Order Correction - 3

- How to get the best $\underline{p}_l$?

$$\hat{A} = [\hat{A}_1 \;\; \hat{A}_2]; \text{ Suppose } \hat{A} = [Q_l \;\; \bar{Q}_l]\begin{bmatrix} R_l \\ 0 \end{bmatrix} = Q_l R_l \Rightarrow Q_l^T \hat{A} = R_l$$

$$Q_l = R(\hat{A}) \text{ column space of } \hat{A}$$

$$\bar{Q}_l = \text{Orthogonal to } \hat{A} \Rightarrow \bar{Q}_l^T \hat{A} = 0 \Rightarrow \hat{A}^T \bar{Q}_l = 0$$

Also, $Q_l^T \bar{Q}_l = 0$; Since $\underline{d}_l$ and $\underline{d}_{l+1}$ are feasible

$$\hat{A}^T \underline{d}_{l+1} = \hat{A}^T \underline{d}_l + \hat{A}^T \underline{p}_l = \underline{0} \Rightarrow \hat{A}^T \underline{p}_l = \underline{0}$$

Since columns of Q_l and $\bar{Q}_l$ span R^n, we can write

$$\underline{p}_l = Q_l \underline{y}_l + \bar{Q}_l \underline{z}_l$$

$$\hat{A}^T \underline{p}_l = \hat{A}^T Q_l \underline{y}_l + \hat{A}^T \bar{Q}_l \underline{z}_l = \underline{0} \Rightarrow R_l^T \underline{y}_l = 0 \Rightarrow \underline{y}_l = \underline{0}$$

$$\therefore \boxed{\underline{p}_l = \bar{Q}_l \underline{z}_l}$$

SQP Algorithm with 2nd Order Correction - 4

- The problem of finding $\underline{p}_l$ can be written as:

$$\min \underline{g}_k^T(\underline{d}_l+\underline{p}_l)+\frac{1}{2}(\underline{d}_l+\underline{p}_l)^T B_k(\underline{d}_l+\underline{p}_l)$$

$$\text{s.t. } \hat{A}^T \underline{p}_l = \underline{0}$$

$$\Rightarrow \min(\underline{g}_k + B_k\underline{d}_l)^T \underline{p}_l + \frac{1}{2}\underline{p}_l^T B_k \underline{p}_l$$

$$\text{s.t. } \hat{A}^T \underline{p}_l = \underline{0}$$

$$\Rightarrow \min \tilde{\underline{g}}_k^T \underline{p}_l + \frac{1}{2}\underline{p}_l^T B_k \underline{p}_l$$

$$\text{s.t. } \hat{A}^T \underline{p}_l = \underline{0}$$

where $\tilde{\underline{g}}_k = \underline{g}_k + B_k\underline{d}_l$

So, the problem of finding $\underline{p}_l$ is another QPP, but simpler constraints $\Rightarrow$ can solve very easily!!!

SQP Algorithm with 2nd Order Correction - 5

- Optimality Conditions:

$$\begin{bmatrix} B_k & \hat{A} \\ \hat{A}^T & 0 \end{bmatrix} \begin{bmatrix} \underline{p}_l \\ \underline{\tilde{\lambda}}_{l+1} \end{bmatrix} = \begin{bmatrix} -\underline{g}_k - B_k \underline{d}_l \\ \underline{0} \end{bmatrix}$$

of QPP

$$\text{or} \quad \begin{bmatrix} B_k & \hat{A} \\ \hat{A}^T & 0 \end{bmatrix} \begin{bmatrix} \underline{d}_{l+1} \\ \underline{\tilde{\lambda}}_{l+1} \end{bmatrix} = \begin{bmatrix} -\underline{g}_k \\ \underline{\hat{b}} \end{bmatrix}$$

Since $\underline{d}_{l+1} = Q_l \underline{c}_{l+1} + \bar{Q}_l \underline{a}_{l+1}$

$$\Rightarrow \begin{bmatrix} B_k Q_l & B_k \bar{Q}_l & Q_l R_l \\ R_l^T Q_l^T Q_l & 0 & 0 \end{bmatrix} \begin{bmatrix} \underline{c}_{l+1} \\ \underline{a}_{l+1} \\ \underline{\tilde{\lambda}}_{l+1} \end{bmatrix} = \begin{bmatrix} -\underline{g}_k \\ \underline{\hat{b}} \end{bmatrix}$$

$\Rightarrow R_l^T \underline{c}_{l+1} = \underline{\hat{b}} \Rightarrow$ solve for $\underline{c}_{l+1}$ in $O(\frac{n^2}{2})$ operations.

$$\bar{Q}_l^T B_k \bar{Q}_l \underline{a}_{l+1} = -\bar{Q}_l^T [\underline{g}_k + B_k Q_l \underline{c}_{l+1}]$$

SQP Algorithm with 2nd Order Correction - 6

Do Cholesky on $\bar{Q}_l^T B_k \bar{Q}_l = \bar{U}_l \bar{U}_l^T$, so $\underline{a}_{l+1} = -[\bar{U}_l \bar{U}_l^T]^{-1} \bar{Q}_l^T [\underline{g}_k + B_k Q_l \underline{c}_{l+1}]$

$\Rightarrow \underline{d}_{l+1} = Q_l \underline{c}_{l+1} - \bar{Q}_l [\bar{U}_l \bar{U}_l^T]^{-1} \bar{Q}_l^T [\underline{g}_k + B_k Q_l \underline{c}_{l+1}]$

Finally, $R_l \underline{\lambda}_{l+1} = -Q_l^T [\underline{g}_k + B_k Q_l \underline{c}_{l+1} + B_k \bar{Q}_l \underline{a}_{l+1}] = -Q_l^T [\underline{g}_k + B_k \underline{d}_{l+1}]$

or $\underline{\lambda}_{l+1} = -R_l^{-1} Q_l^T [\underline{g}_k + B_k \underline{d}_{l+1}]$ Multiplier vector in $O(n^2)$ operations.

- Update of working set:
  - If $\underline{d}_{l+1} = \underline{d}_l \Rightarrow \underline{p}_l = 0 \Rightarrow \underline{d}_l$ is optimal w.r.t current set of constraints S_l
  - If $\underline{p}_l \neq 0$ but is feasible for all constraints, $\underline{d}_{l+1} = \underline{d}_l + \underline{p}_l$ is the new point.

    If $\mu_q \geq 0 \; \forall q$ of inequality constraints, stop $\Rightarrow$ Optimal
  - If $\underline{d}_{l+1}$ is not feasible $\Rightarrow$ some constraint is voilated. So, let $\underline{d}_{l+1} = \underline{d}_l + \alpha_l \underline{p}_l$

where: $$\alpha_l = \min_{\substack{\underline{a}_i^T \underline{p}_l > 0 \\ i \notin S_l}} \{1, \frac{b_i - \underline{a}_i^T \underline{d}_l}{\underline{a}_i^T \underline{p}_l}\}$$

$$i_a = \arg \min_{\substack{\underline{a}_i^T \underline{p}_l > 0 \\ i \notin S_l}} \{1, \frac{b_i - \underline{a}_i^T \underline{d}_l}{\underline{a}_i^T \underline{p}_l}\} \Rightarrow S_{l+1} = S_l \cup \{i_a\}$$

SQP Algorithm with 2nd Order Correction - 7

- How to drop active inequality constraints:
  - $\underline{d}_{l+1}$ is feasible for all constraints in S_l
  - If all $\mu_q > 0 \Rightarrow$ Optimal
  - Find $i_d = \arg\min_q \{\mu_q\}, S_{l+1} = S_l - \{i_d\}$
- Algorithm:

Step1: Start with an initial feasible $\underline{d}_0$ and the corresponding working set S_0. Set $l = 0$.

Step 2: Solve for $\underline{p}_l \Rightarrow \underline{d}_{l+1}$

Step 3: Find step length α_l. If $\alpha_l < 1$, append corresponding constraint i_a. So

$$\tilde{A} = [\hat{A} \quad \underline{\hat{a}}] = Q\begin{pmatrix} R & Q^{T'}\underline{\hat{a}} \\ 0 & 1 \end{pmatrix} = \bar{Q}\begin{bmatrix} \bar{R} \\ 0 \end{bmatrix}$$

$$\tilde{Q} = QQ_l, \text{ New } Q_l = [Q_l \vdots \underline{q}_{m+r_l+1}]$$

New $\bar{Q}_l$ complete change

Return to Step 2 ; else go to Step 4

SQP Algorithm with 2nd Order Correction - 8

- Algorithm (continued)

Step 4: If $\alpha_j = 1$, compute $\underline{\lambda}$ (last r components are $\underline{\mu}$)

Find $\mu_{i_d} = \min_i (\mu_i)$

If $\mu_{i_d} \geq 0$ Stop

else drop constraint corresponding to i_d

$\tilde{A} = [\underline{a}_1 \ \underline{a}_2 \ldots \underline{a}_{i_d - 1} \ \underline{a}_{i_d + 1} \ldots \underline{a}_{m+\hat{r}}]$

$Q^T \tilde{A} = \begin{bmatrix} M \\ 0 \end{bmatrix}$

$M =$ upper triangular cols. 1 to $i_d - 1$. Has elements in subdiagonals for columns i_d to $m + \hat{r} - 1$

New $\mathrm{Q}_l = [\underline{q}_1 \ \underline{q}_2 \ldots \underline{q}_{i_d - 1}, \hat{Q}]$,

New $\bar{\mathrm{Q}}_l = [\hat{g}, \ \bar{Q}_{l\ old}]$

Go to Step 2

SQP Algorithm with 2nd Order Correction - 9

- What if QPP is infeasible? Add artificial variables to detect it.

$$\left.\begin{aligned} &\min \frac{1}{2}\underline{d}^T B_k \underline{d} + \underline{g}_k^T \underline{d} + C\underline{\xi}^T \underline{1} \\ &\text{s.t. } \nabla \underline{h}^T(\underline{x}_k)\underline{d}_k + \underline{h}(\underline{x}_k) = \underline{0} \\ &\quad \nabla \underline{g}^T(\underline{x}_k)\underline{d}_k + \underline{g}(\underline{x}_k) \le \underline{\xi} \\ &\quad \underline{\xi} \ge \underline{0} \end{aligned}\right\} \text{Always feasible}$$

Other Methods:

- M.JD Powell, "On the QP Algorithm of Goldfarb and Idnani", MP, 1985, pp.46-61
- Goldfarb and Idnani, "A numerically stable dual method for solving strictly quadratic programs convex"MP,1983,pp. 1-33

Summary

- Motivation for Successive Quadratic Programming (SQP) Methods
- Key SQP Ideas
- Newton Version of SQP
- Descent Property of Merit Function $f+cP$
- Quasi-Newton Version of SQP
- SQP with second order correction