Sequential Quadratic Programming

(Alternative to Augmented Lagrangian Approach)

Consider the equality constrained problem

$$\boxed{\min f(x) \quad \text{s.t.} \quad c_i(x) = 0 \quad i \in \mathcal{E}}$$

The Lagrangian Form is

$$\min_{x,p} \; L(x,p) = f(x) \underbrace{- \sum_{i \in \mathcal{E}} p_i c_i(x)}_{-p^T c(x)}$$

And suppose we wish to perform a Newton Step from (x_k, p_k) to (x_{k+1}, p_{k+1}) which satisfies the stationarity condition to second order in $L(x,p)$.

$$x_{k+1} = x_k + s^x$$

$$p_{k+1} = p_k + s^p$$

$$s = \begin{bmatrix} s^x \\ s^p \end{bmatrix}$$

- $\nabla L(x+s^x, p+s^p) = 0$
- $\nabla L(x+s^x, p+s^p) = \nabla L(x,p) + \nabla^2 L(x,p) s$

$$\Rightarrow \quad \nabla^2 L(x,p)\, s = -\nabla L(x,p)$$

$$\begin{bmatrix} \nabla^2_{xx} L(x,p) & \nabla^2_{xp} L(x,p) \\ \nabla^2_{px} L(x,p) & \nabla^2_{pp} L(x,p) \end{bmatrix} \begin{bmatrix} s^x \\ s^p \end{bmatrix} = - \begin{bmatrix} \nabla_x L(x,p) \\ \nabla_p L(x,p) \end{bmatrix}$$

Newton step $s_k = \begin{bmatrix} s_k^x \\ s_k^p \end{bmatrix}$.

We then have

$$\nabla^2_{xp} L(x,p) = \nabla^2_{px} L(x,p) = -\nabla c(x)$$

$$\nabla^2_{pp} L(x,p) = 0$$

$$\nabla_p L(x,p) = -c(x)$$

where $c(x) := \begin{bmatrix} c_1(x) & c_2(x) & \cdots & c_m(x) \end{bmatrix}$

$\nabla c(x) := \begin{bmatrix} \nabla c_1(x) & \nabla c_2(x) & \cdots & \nabla c_m(x) \end{bmatrix}$

transpose of Jacobian

$$\boxed{\begin{bmatrix} \nabla^2_{xx} L(x,p) & \nabla c(x) \\ \nabla c(x)^T & 0 \end{bmatrix} \begin{bmatrix} s^x \\ -s^p \end{bmatrix} = - \begin{bmatrix} \nabla_x L(x,p) \\ c(x) \end{bmatrix}} \quad *$$

- Symmetric!
- Positive Definite if
  - (a) LICQ ($\nabla c(x)$ is L.I.)
  - (b) $\nabla^2_{xx} L$ pos. def. on null($\nabla c(x)^T$)

$\Big\}$ normal KKT condition requirements

An interesting observation is that this result also identifies a stationary point of the following quadratic program

$$\boxed{\begin{aligned} \min_{s^x} \quad & f(x) + (s^x)^T \nabla_x L(x,p) + \frac{1}{2}(s^x)^T \nabla^2_{xx} L(x,p)\, s^x \\ \text{s.t.} \quad & \nabla c(x)^T s^x + c(x) = 0 \end{aligned}} \quad (*)$$

That is, $(*)$ are the KKT conditions of $(*)$

We can make one more alteration that simplifies the computation.

The first equation:

$$\nabla_{xx} L(x,p)\, s^x - \nabla c(x)\, s^p = -\nabla_x L(x,p)$$

$$\nabla_{xx} L(x,p)\, s^x - \nabla c(x)\, s^p = -\nabla_x f(x) + \nabla c(x)\, p$$

$$\nabla_{xx} L(x,p)\, s^x - \nabla c(x)\, (s^p + p) = -\nabla_x f(x)$$

But $s^p + p$ is the updated Lagrange multiplier vector.

This observation leads to the update $\longrightarrow$

$$\boxed{\begin{bmatrix} \nabla^2_{xx} L(x_k, p_k) & \nabla c(x_k) \\ \nabla c(x_k)^T & 0 \end{bmatrix} \begin{bmatrix} s^x \\ -p_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla f(x_k) \\ c(x_k) \end{bmatrix} \qquad \text{with} \quad x_{k+1} = x_k + s^x}$$

The form $\circledast$ is a useful form as the problem appears as a local linear approximation of the constraints and a local quadratic approximation of the Lagrangian objective.

We make one simplification and one generalization.

(S) $(S^x)^T \nabla L_x(x,P) = (S^x)^T \nabla f(x)$
when $c(x) = 0$

(g) linearize inequality constraints

not clear why this last idea works — employ complementarity condition.

solves for S^* with P as the Lagrange multipliers

$$\boxed{\begin{aligned} \min_{s} \quad & f(x_k) + S^T \nabla f(x_k) + \tfrac{1}{2} S^T \nabla^2_{xx} L(x_k, P_k) S \\ \text{s.t.} \quad & S^T \nabla c_i(x_k) + c_i(x_k) = 0 \quad i \in \mathcal{E} \\ & S^T \nabla c_i(x_k) + c_i(x_k) \geq 0 \quad i \in \mathcal{I} \\ & x_{k+1} = x_k + S^*, \quad P_{k+1} = \lambda^* \end{aligned}}$$

$\circledast$

An SQP Algorithm Concept

Given: x_0, p_0

Set: $k \leftarrow 0$

while convergence not satisfied

    Evaluate: $f(x_k)$, $\nabla f(x_k)$, $c(x_k)$, $\nabla c(x_k)$, $\nabla^2_{xx} L(x_k, p_k)$

    Solve $*$ for s^*, λ^*

    $x_{k+1} = x_k + s^*$, $p_{k+1} = \lambda^*$

    $k \leftarrow k+1$

    Other algorithmic updates

end

Putting it all together

P_k here is S^*

Algorithm 18.3 (Line Search SQP Algorithm).

Armijo parameter

backtracking shrink

Choose parameters $\eta \in (0, 0.5)$, $\tau \in (0, 1)$, and an initial pair (x_0, λ_0);
Evaluate $f_0, \nabla f_0, c_0, A_0$; $A = c(x)^T$
If a quasi-Newton approximation is used, choose an initial $n \times n$ symmetric positive definite Hessian approximation B_0, otherwise compute $\nabla^2_{xx}\mathcal{L}_0$;
repeat until a convergence test is satisfied

Compute p_k by solving (18.11); let $\hat{\lambda}$ be the corresponding multiplier; QP solver
Set $p_\lambda \leftarrow \hat{\lambda} - \lambda_k$;
Penalty parameter
Choose μ_k to satisfy (18.36) with $\sigma = 1$;
Set $\alpha_k \leftarrow 1$;
this line search test must be examined next
while $\phi_1(x_k + \alpha_k p_k; \mu_k) > \phi_1(x_k; \mu_k) + \eta\alpha_k D_1(\phi(x_k; \mu_k) p_k)$
Reset $\alpha_k \leftarrow \tau_\alpha \alpha_k$ for some $\tau_\alpha \in (0, \tau]$;
end (while) linesearch
Set $x_{k+1} \leftarrow x_k + \alpha_k p_k$ and $\lambda_{k+1} \leftarrow \lambda_k + \alpha_k p_\lambda$;
Evaluate $f_{k+1}, \nabla f_{k+1}, c_{k+1}, A_{k+1}$, (and possibly $\nabla^2_{xx}\mathcal{L}_{k+1}$); general updates
If a quasi-Newton approximation is used, set
$s_k \leftarrow \alpha_k p_k$ and $y_k \leftarrow \nabla_x \mathcal{L}(x_{k+1}, \lambda_{k+1}) - \nabla_x \mathcal{L}(x_k, \lambda_{k+1})$,
and obtain B_{k+1} by updating B_k using a quasi-Newton formula; BFGS update

end (repeat)

Quasi-Newton Update by Damped BFGS

$\left(\text{for} \quad B_k := \nabla^2_{xx} L(x_k, p_k) \right)$

$$s_k = x_{k+1} - x_k$$

$$y_k = \nabla_x L(x_{k+1}, \lambda_{k+1}) - \nabla_x L(x_k, \lambda_{k+1})$$

$$r_k = \theta y_k + (1-\theta) B_k s_k$$

$$\theta_k = \begin{cases} 1 & \text{if } s_k^T y_k \geq \frac{1}{5} s_k^T B_k s_k \\ \frac{4}{5} (s_k^T B_k s_k) / (s_k^T B_k s_k - s_k^T y_k) & \text{otherwise} \end{cases}$$

$$B_{k+1} = B_k - \frac{B_k s_k s_k^T B_k^T}{s_k^T B_k s_k} + \frac{r_k r_k^T}{s_k r_k^T}$$

THE QP SOLVER

For solving

$$\min \tfrac{1}{2} s^T G s + s^T c \quad \text{s.t.} \quad A_e x = b_e, \quad A x \geq b$$

we have

$$G = \nabla^2_{xx} L(x, P)$$

$$c = \nabla_x f(x)$$

$$A_e = \nabla_x C_E(x)^T$$

$$b_e = -C_E(x)$$

$$A = \nabla_x C_I(x)^T$$

$$b = -C_I(x)$$

$$\boxed{\begin{aligned} \min_s \quad & f(x_k) + s^T \nabla f(x_k) + \tfrac{1}{2} s^T \nabla^2_{xx} L(x_k, P_k) s \\ \text{s.t.} \quad & s^T \nabla C_i(x_k) + C_i(x_k) = 0 \quad i \in \mathcal{E} \\ & s^T \nabla C_i(x_k) + C_i(x_k) \geq 0 \quad i \in \mathcal{I} \\ & x_{k+1} = x_k + s^*, \quad P_{k+1} = \lambda^* \end{aligned}}$$

$$C_E(x) = [\cdots \; C_i(x) \; \cdots]^T \quad i \in \mathcal{E}$$

$$\nabla_x C_E(x) = [\cdots \; \nabla_x C_i(x) \; \cdots] \quad i \in \mathcal{E}$$

$$C_I(x) = [\cdots \; C_i(x) \; \cdots]^T \quad i \in \mathcal{I}$$

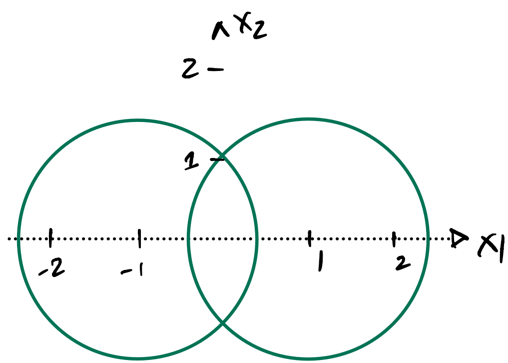
$$\nabla_x C_I(x) = [\cdots \; \nabla_x C_i(x) \; \cdots] \quad i \in \mathcal{I}$$

The subproblem may be infeasible

Example.

$$c_1(x) = -(x_1+1)^2 - x_2^2 + 2 \geq 0$$

$$c_2(x) = -(x_1-1)^2 - x_2^2 + 2 \geq 0$$



at $x=(0,0)$ we have

$s_1 = 1/2$, $s_1 = -1/2$, s_2 free ✗

at $x=(1,0)$ we have

$s_1 = -1/2$, $2=0$ ✗

at $x=(0,1)$ we have

$s_1+s_2=0$, $s_1-s_2=0$ ✓

at $x=(0,2)$ we have

$$\left.\begin{aligned} s_1+2s_2 &= -3/2 \\ s_1-2s_2 &= 3/2 \end{aligned}\right\} \Rightarrow s_1=0,\ s_2=-3/4$$ ✓

$$\nabla c_1(x) = \begin{bmatrix} -2(x_1+1) \\ -2x_2 \end{bmatrix} \rightarrow -2(x_1+1)s_1 - 2x_2 s_2 - (x_1+1)^2 - x_2^2 + 2 = 0$$

$$\nabla c_2(x) = \begin{bmatrix} -2(x_1-1) \\ -2x_2 \end{bmatrix} \rightarrow -2(x_1-1)s_1 - 2x_2 s_2 - (x_1-1)^2 - x_2^2 + 2 = 0$$

We can always relax the constraints using a penalty function approach.

For any (possibly infeasible) problem, we can solve the relaxed problem

$$\begin{aligned} \min_{x} \quad & f(x) \\ \text{s.t.} \quad & c_i(x) = 0 \quad i \in \mathcal{E} \\ & c_i(x) \geq 0 \quad i \in \mathcal{I} \end{aligned}$$

$\longrightarrow$

$$\begin{aligned} \min_{x,e,d,b} \quad & f(x) + \mu \sum_{i \in \mathcal{E}} (e_i + d_i) + \mu \sum_{i \in \mathcal{I}} b_i \\ \text{s.t.} \quad & c_i(x) = e_i - d_i \quad i \in \mathcal{E} \\ & c_i(x) \geq -b_i \quad i \in \mathcal{I} \\ & e, d, b \geq 0 \end{aligned}$$

General QP Solve Procedure:

Determine if the un-relaxed QP subproblem is feasible. If so then solve. If not then relax the QP subproblem and solve. Either method returns (s^*, λ^*). Then $x_{k+1} = x_k + s^*$, $p_{k+1} = \lambda^*$.

What Does SQP use as a line Search?

Define $\phi_1(x;\mu) = f(x) + \mu [c(x)]_+$

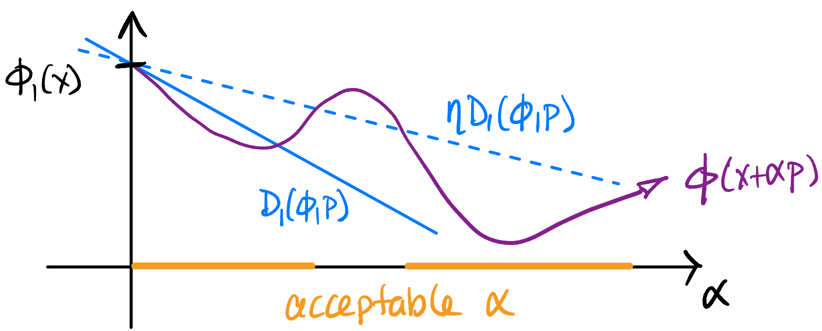
where the total constraint violation is

$$[c(x)]_+ := \sum_{i\in\mathcal{E}} |c_i(x)| + \sum_{i\in\mathcal{I}} \max\{0, -c_i(x)\}.$$

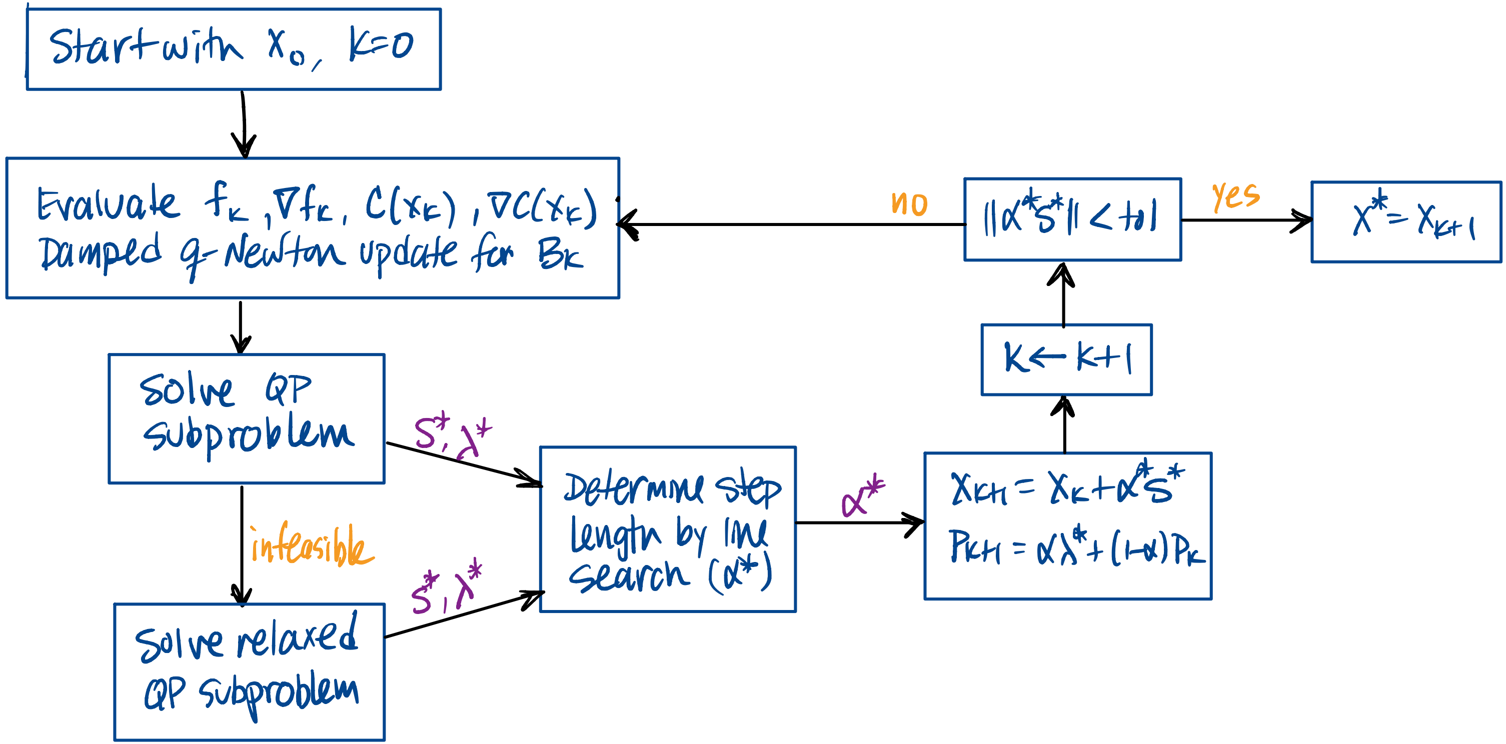
(this is an ℓ_1 penalty if we have only equality constraints)

Also, $D_1(\phi; d)$ is the directional derivative of ϕ in direction d.

$$\phi_1(x+\alpha p;\mu) \le \phi_1(x;\mu) + \eta\alpha D_1(\phi(x;\mu), p)$$ (Armijo-like condition)



(use simple backtracking)


Start with x_0, $k=0$
Evaluate f_k, ∇f_k, $C(x_k)$, $\nabla C(x_k)$
Damped q-Newton update for B_k
Solve QP subproblem
infeasible
Solve relaxed QP subproblem
s^*, λ^*
s^*, λ^*
Determine step length by line search (α^*)
α^*
$x_{k+1} = x_k + \alpha^* s^*$
$p_{k+1} = \alpha \lambda^* + (1-\alpha) p_k$
$k \leftarrow k+1$
$\|\alpha^* s^*\| < tol$
no
yes
$x^* = x_{k+1}$

Notes on updating μ.

The goal is to choose μ large enough so that the we take advantage of the exact penalty term in $\phi(x;\mu) = f(x) + \mu[c(x)]_+$.

When $[c(x)]_+ = 0$ we have $D(\phi(x;\mu);p) = \nabla f(x)^T p < 0$ (p is descent dir.)

However, when $[c(x)]_+ > 0$, the KKT conditions and Taylor's Theorem lead to (18.31):

$$D(\phi(x;\mu);p) = \nabla f(x)^T p - \mu[c(x)]_+$$

Thus, to have a descent direction p, we choose

$$\mu > \frac{\nabla f(x)^T p}{[c(x)]_+} \quad \text{or (see 18.33)} \quad \mu \geq \frac{\nabla f(x)^T p}{(1-\rho)[c(x)]_+}, \quad 0<\rho<1.$$

(if $\nabla f(x)^T p \leq 0$ then no update is required!)

$$\text{let } \mu = \max\left\{ \mu_0, \frac{\nabla f(x)^T p}{(1-\rho)[c(x)]_+} \right\}$$