Chapter 18: Sequential Quadratic Programming

* One of the best for non-linearly constrained opt.
* Solves quadratic substeps at
* Works for both line-search and trust-region methods
* Small or large problems
* Active set methods for non-linear programming:
  - IQP approach: estimate ρ and active set together
  - EQP: estimate them separately.

* Basic ideas:
  - 18.1 for local problems
  - Extend to "global" using line-search and trust-region
  - attack large problems.

18.1 Local SQP Method

Equality only in:

$$\min f(x)$$

subject to $c(x) = 0$ (several constr.)

$$\begin{bmatrix} c_1(x) \\ c_2(x) \\ \vdots \\ c_m(x) \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

where $\left.\begin{array}{l} f: \mathbb{R}^n \to \mathbb{R} \\ c: \mathbb{R}^n \to \mathbb{R}^m \end{array}\right\}$ smooth functions

Basic idea:

- Apply Newton to KKT.

<u>KKT</u>: Start with $\mathcal{L}(x, \lambda) = f(x) - \lambda^T c(x)$

Define Jacobian:

$$A(x)^T = \left[\nabla c_1(x) \quad \nabla c_2(x) \quad \dots \quad \nabla c_m(x) \right]$$

<u>Note:</u> $\nabla_x \left(\lambda^T c(x) \right) = \nabla \left[\sum_{i=1}^{m} \lambda_i c_i(x) \right]$

$$= \sum_{i=1}^{m} \lambda_i \nabla c_i(x) = A(x)^T \lambda$$

corrected

Note the transposition

Apply the KKT conditions:

$$\mathcal{L}_x(x,\lambda) = \nabla f(x) - A(x)^T \lambda$$

For the second term, we have: $A^T(x)\lambda$.

$$\mathcal{L}_x(x,\lambda) = \nabla f(x) - A(x)^T \lambda = 0$$

In addition, from the constraints:

$$c(x) = 0.$$

Together, we define $F(x,\lambda)$ by:

$$F(x,\lambda) = \begin{bmatrix} \nabla f(x) - A(x)^T \lambda \\ c(x) \end{bmatrix} = 0. \quad \text{AA}$$

with $(n+m)$ equations

in $(n+m)$ unknowns x, λ.

For (1Δ), we note that this is a standard nonlinear problem for which we can use the multivariate Newton method for solving $F(x,\lambda)=0$.

In 1-D, recall the Newton algorithm:

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

For M-D, it is simply (see Chapter 11):

$$x_{k+1} = x_k - \left[\nabla f(x_k)\right]^{-1} f(x_k)$$

Or, rewrite this as:

$$(*)\ \begin{cases} x_{k+1} = x_k + P_k, \text{ where} \\ \left[\nabla f(x)\right] P_k = -f(x_k). \end{cases}$$

To apply (*) to $F(x,\lambda)=0$, we have:

$$\nabla F(x,\lambda) = \begin{bmatrix} \nabla_x F_1(x,\lambda) & \nabla_\lambda F_1(x,\lambda) \\ \nabla_x F_2(x,\lambda) & \nabla_\lambda F_2(x,\lambda) \end{bmatrix}$$

We have:

$$\nabla_x F_1(x,\lambda) = \nabla_x \left(\mathcal{L}_x(x,\lambda) \right)$$
$$= \nabla^2_{xx} \mathcal{L}(x,\lambda)$$
$$= W(x,\lambda) \text{ by definition of } W(x,\lambda).$$

$$\nabla_\lambda F_1(x,\lambda) = -A(x)^T$$

$$\nabla_x F_2(x,\lambda) = \nabla_x c(x) = A(x)$$

$$\nabla_\lambda F_2(x,\lambda) = \nabla_\lambda c(x) = 0.$$

We thus write:

$$\begin{bmatrix} x_{k+1} \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} x_k \\ \lambda_k \end{bmatrix} + \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix},$$

where p_k and p_λ solve:

$$\underbrace{\begin{bmatrix} W_k & -A_k^T \\ A_k & 0 \end{bmatrix}}_{[\nabla F(x,\lambda)]} \begin{bmatrix} p_k \\ p_\lambda \end{bmatrix} = \underbrace{\begin{bmatrix} -\nabla f_k + A_k^T \lambda_k \\ -c_k \end{bmatrix}}_{-F(x,\lambda)}$$

In chapter 16, "non-singularity" of the solutions is proven, based on:

$$\boxed{W_k = \nabla^2_{xx} \mathcal{L}(x, \lambda)}$$

Assumption 18.1

(a) The constraint Jacobian A_k has full row rank.

(b) The matrix W_k is positive definite for: $d^T W_k d > 0$ for $A_k d = 0$, $d \neq 0$.

(a) is LICQ, (b) is satisfied at the solution.

SQP Framework

We can define an SQP approach, based on what we just did.

At each iterate, compute (x_k, λ_k).

<u>Solve</u>: $\min_{p} \tfrac{1}{2} p^T W_k p + \nabla f_k^T p$

subject to: $A_k p + c_k = 0$

To solve this, form $\mathcal{L}(x, \lambda)$ locally:

$$\mathcal{L}(p_k, \mu_k) = \tfrac{1}{2} p_k^T W_k p_k + \nabla f_k^T p_k - \mu_k^T (A_k p + c_k)$$

$$\Rightarrow \mathcal{L}_{p_k}(p_k, \mu_k) = W_k p_k + \nabla f_k - A_k^T \mu_k = 0$$

<u>giving</u>:

$$W_k p_k + \nabla f_k - A_k^T \mu_k = 0$$

$$A_k p_k + c_k = 0$$

which is:

$$\begin{bmatrix} W_k & -A_k^T \\ A_k & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \lambda_{k+1} \end{bmatrix} = \begin{bmatrix} -\nabla f_k \\ -c_k \end{bmatrix} \quad \square$$

for $\lambda_{k+1} = \mu_k$.

Algorithm 18.1 (Local SQP Algorithm)

Choose an initial pair (x_0, λ_0)

for $k = 0, 1, 2, \ldots$ $\boxed{W_k = \nabla^2_{xx} L_k}$

Evaluate $f_k, \nabla f_k, W_k = W(x_k, \lambda_k)$, c_k, and A_k;

Solve $\Box$ for p_k, μ_k.

$x_{k+1} = x_k + p_k$;

If convergence test satisfied

STOP with solution (x_{k+1}, λ_{k+1});

end Update : $x_{k+1} = x_k + p_k$, λ_{k+1}

Evaluate $F(x, \lambda)$, and check if $\| F(x, \lambda) \| \simeq 0$.

... can be extended for inequalities, line-search, and trust-region methods

For inequalities:

$$\min f(x)$$

<u>subject to:</u> $c_i(x) = 0, \quad i \in \mathcal{E}$

$c_i(x) \geq 0, \quad i \in \mathcal{I}$

Linearize the problem to:

$$\min_{\underline{p}} \; f_k + \nabla f_k^T \underline{p} + \tfrac{1}{2} \underline{p}^T \nabla_{xx}^2 \mathcal{L}_k \underline{p}$$

<u>subject to:</u>

$$\nabla c_i(x_k)^T \underline{p} + c_i(x_k) = 0, \quad i \in \mathcal{E}$$

$$\nabla c_i(x_k)^T \underline{p} + c_i(x_k) \geq 0, \quad i \in \mathcal{I}$$

Solve by methods in chapter 16.