10.34, Numerical Methods Applied to Chemical Engineering
Professor William H. Green
Lecture #17: Constrained Optimization.

Notation

"second derivative of $\underline{f}(\underline{x})$.": We normally mean

f_{xx} Hessian Matrix $\underline{\underline{H}} = \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & \frac{\partial^2 f}{\partial x_1 \partial x_2} \\ \frac{\partial^2 f}{\partial x_2 \partial x_1} & \frac{\partial^2 f}{\partial x_2^2} \end{pmatrix}$ $\nabla^2 f$ in BEERS

but second derivative can also mean:

f_{xx} Laplacian $Tr\{\underline{\underline{H}}\} = \frac{\partial^2 f}{\partial x_1^2} + \frac{\partial^2 f}{\partial x_2^2}$ - scalar $\nabla^2 f$ in Physics Texts

$\underline{\nabla} \cdot (\underline{\nabla} f)$

Constrained Optimization

Equality Constraints: $\min_{\underline{x}} \underline{f}(\underline{x})$ such that $\underline{g}(\underline{x}) = 0$

May be able to invert this statement as: $x_N = \underline{G}(x_1, x_2, \ldots, x_{N-1})$

Then we can state min as: min $\underline{f}(x_1, x_2, \ldots, x_{N-1}, \underline{G}(x_1, x_2, \ldots, x_{N-1}))$

Notice the x_N is gone. Constrained becomes unconstrained. Solve with previous methods.

Other way to do this:

Lagrange Multipliers

Unconstrained $\left.\frac{\partial f}{\partial x_n}\right|_{\underline{x}_{mn}} = 0$ at the minimum

- constrained problems do not work that way!
    - BOUNDARIES GET IN THE WAY

Constrained: $\left.\frac{\partial f}{\partial x_n}\right|_{\underline{x}\,const.\min} = \lambda \left.\frac{\partial g}{\partial x_n}\right|_{\underline{x}\,const.\min}$ $\underline{\nabla} f|_{const.\ min} = \lambda \underline{\nabla} g|_{\underline{x}\ const.\ min}$

Gradient of f equals 0 in directions parallel to constraint but not perpendicular

Create a new function $L(\underline{x}, \lambda) = f(\underline{x}) - \lambda \underline{g}(\underline{x})$ (λ is unknown before you do the problem)

$\nabla_x L = 0$	at constrained min
$\partial L / \partial \lambda = \underline{g}(\underline{x}) \rightarrow 0$	at constrained min

Second derivatives not necessarily all positive

Augmented Lagrangian

$L_A = \underline{f}(\underline{x}) - \lambda \underline{g}(\underline{x}) + \frac{1}{2\mu^{(0)}}(g(x))^2$

$\min_{\underline{x}} L_{\underline{A}}$ given initial guess $\lambda^{[0]}, \mu^{[0]} \rightarrow \underline{x}_{min}^{[0]}$

$$\nabla_x L_A(\underline{x}_{min}^{[0]}, \lambda^{[0]}) = \underline{\nabla f}|_{x_{min}^{[0]}} - \lambda^{[0]} \underline{\nabla g}(\underline{x}_{min}^{[0]}) - \frac{1}{\mu^{(0)}} \underline{g}(\underline{x}) \underline{\nabla g}(\underline{x}) \Rightarrow \underline{\nabla f} - \underbrace{(\lambda^{[0]} - \frac{g(x_{min}^{[0]})}{\mu^{(0)}})}_{\lambda^{[1]}} \underline{\nabla g}(\underline{x})$$

As $\mu^{[0]}$ shrinks, $\frac{1}{2\mu^{(0)}}$ gets large, magnifying $(\underline{g}(\underline{x}))^2$ term, and thus holding the constraint more strictly. $\min_{\underline{x}} L_{\underline{A}}$ using $\lambda^{[1]}$ get a new $\underline{x}_{min}$. In quantum mechanics, λ corresponds to orbital energies. Most of the time, λ does not have a physical meaning. $\mu^{[0]}$ is a mathematical trick.

More Than One Constraint

Suppose you have >1 constraints:

$$\left.\begin{aligned} g_1(\underline{x}) &= 0 \\ g_2(\underline{x}) &= 0 \\ g_3(\underline{x}) &= 0 \end{aligned}\right\}$$

make sure these are compatible i.e. there is a "feasible space" – set of $\underline{x}$ that satisfies all constraints

$L = \underline{f}(\underline{x}) - \sum_i \lambda_i g_i(\underline{x})$

$\underline{\nabla}L = 0 \qquad \underline{\nabla f} = \sum_i \lambda_i \nabla g_i$

Inequality Constraints

very common

$\min \underline{f}(\underline{x})$, s.t. $\underline{g}(\underline{x}) = 0$, $h(\underline{x}) \geq 0$

Active inequality constraints: $h(\underline{x}_{min}) = 0$

Inactive inequality constraints: $h(\underline{x}_{min}) > 0$

Usually, we do not know whether h's are active or inactive before doing a problem, but must leave in during optimization process, to allow finding of solution:

$\underline{\nabla f} = K_j \underline{\nabla} h_j$, $K \geq 0$ when h_j is active; also $h_j = 0$ and $k_j \geq 0$.

$K_j h_j(\underline{x}_{\text{const. min}}) = 0$ when h_j is inactive if inactive, $h_j \neq 0$ and $k_j = 0$. $\underline{\nabla h_j}$ can be anything; it does not affect the problem

Karash-Kahn-Tucker (KKT) conditions:

$L = \underline{f}(\underline{x}) - \Sigma\lambda_i g_i(\underline{x}) - \Sigma k_j h_j(\underline{x})$

$\underline{\nabla}L(\underline{x}_{min}) = \underline{0}$ $\quad\quad$ $\underline{h}(\underline{x}_{min}) \geq \underline{0}$

$\underline{g}(\underline{x}_{min}) = \underline{0}$ $\quad\quad$ $K_j \geq 0$

$K_j h_j = 0$

To handle active-inactive constraints, add *slack variables*:

$h_j(\underline{x}) \geq 0 \rightarrow h_j(\underline{x}) - S_j = 0, \; S_j \geq 0$

Augmented Method L_A:

Optimal $S_j = \max\{h_j(\underline{x}) - \mu^{[k]}K_j^{[k]}; 0\}$

$L_A = \underline{f}(\underline{x}) - \Sigma\lambda_i g_i - \Sigma k_j h_j - \Sigma\mu^{[k]}(k_j^{[k]})^2 + (^1/_{2\mu^{[0]}})(g_i^2+h_j^2+(\mu^{[0]}k_j)^2)$

$\underline{F}(\underline{x}) = \underline{\nabla}L_A = 0$ Use Newton's Method with Broyden to approximate the Hessian matrix.

Trying to solve: $\underline{\underline{J}}_{L_A} * \underline{\Delta x} = -\underline{\nabla}L_A$ Use Newton's method to find x

Jacobian is messy:

$$\begin{pmatrix} \left(\frac{\partial^2 L}{\partial x_i \partial x_j}\right)\Bigg|_{old} & \left(-\frac{\partial C_m}{\partial x_j}\right)^T\Bigg|_{old} \\ \left(\frac{\partial C_n}{\partial x_j}\right)\Bigg|_{old} & \underline{\underline{0}} \end{pmatrix} \begin{pmatrix} \underline{\Delta x} \\ \underline{\Delta \lambda} \end{pmatrix} = \begin{pmatrix} -\nabla_x f\big|_{x^{old}} \\ \underline{S} - \underline{C(x^{old})} \end{pmatrix}$$

If we want to: $\min_{\underline{p}} \underline{f}(\underline{p}) = (^1/_2)\underline{p}^T \left(\frac{\partial^2 L}{\partial x_i \partial x_j}\right)\Bigg|_{x^{old}} \underline{p} + \underline{\nabla}f|_{\underline{x}^{old}} \cdot \underline{p}$

such that $\sum \frac{\partial c_m}{\partial x_j}\bigg|_{x^{old}} p_j + c_m(x^{old}) = 0 \quad \forall_m = 1,\ldots,N_{\text{constraints}}$

- can easily get $\underline{p}$ (same as $\underline{\Delta x}$ above) "quadratic program"

Sequential Quadratic Programming (SQP)

In MATLAB: fmincon