Algorithms for nonlinear constrained problems: (12/8)

Basic idea for unconstrained problem:

$$\text{minimize } f(x)$$

Find x^* st $\nabla f(x^*) = 0$ iteratively. Leads to Newton's method. So long as e.g. Armijo conditions hold, convergence to local minimum guaranteed.

Consider equality-constrained optimization problem:

$$\begin{aligned} &\text{minimize } f(x) \\ &\text{subject to } g(x) = 0, \end{aligned} \qquad (*)$$

where $g : \mathbb{R}^n \to \mathbb{R}^m$, $m < n$, with Dg full rank.

Lagrange function for $(*)$ is:

$$L(x, \lambda) = f(x) + \lambda^T g(x).$$

To solve, need to satisfy KKT conditions:

$$\begin{cases} \left.\dfrac{\partial L}{\partial x}\right|_{(x^*, \lambda^*)} = \nabla f(x^*) + Dg(x^*)\lambda^* = 0 \\ g(x^*) = 0 \end{cases}$$

Note, second condition is equivalent to:

$$\left.\frac{\partial L}{\partial \lambda}\right|_{(x^*, \lambda^*)} = g(x^*) = 0.$$

So, KKT conditions can be written more compactly as:

$$\nabla L = 0.$$

Apply Newton's method... Have:

$$\frac{\partial^2 L}{\partial x \partial x^T} = \nabla^2 f(x) + \nabla^2 g(x)\lambda$$

$$\frac{\partial^2 L}{\partial x \partial \lambda^T} = \nabla g(x) = \frac{\partial^2 L}{\partial \lambda \partial x^T}^T$$

$$\frac{\partial^2 L}{\partial \lambda \partial \lambda^T} = 0$$

Letting $p_k = x_{k+1} - x_k$ and $\nu_k = \lambda_{k+1} - \lambda_k$, the Newton system is:

$$\begin{bmatrix} \nabla^2 f(x_k) + \nabla^2 g(x_k)\lambda_k & \nabla g(x_k) \\ \nabla g(x_k)^T & 0 \end{bmatrix} \begin{bmatrix} p_k \\ \nu_k \end{bmatrix} = - \begin{bmatrix} \nabla f(x_k) + \nabla g(x_k)\lambda_k \\ g(x_k) \end{bmatrix}$$

Using Newton here has the same trade-offs as using it for solving unconstrained problems:

- Quadratic convergence if in basin of attraction
- Can diverge
- No guarantee on global optimality

Additional worries:

- No guarantee that x_k's are feasible!
- Can't use $L(x_{k+1}, \lambda_{k+1}) < L(x_k, \lambda_k)$ to measure progress.

Example: Consider:

$$\text{minimize } x^2$$
$$\text{subject to } x = 1$$

The Lagrange function for this problem is:

$$L(x,\lambda) = x^2 + \lambda(1-x)$$

Hence:

$$\nabla L = (2x - \lambda, 1 - x)$$

$$\nabla^2 L = \begin{bmatrix} 2 & -1 \\ -1 & 0 \end{bmatrix}$$

Let's check the eigenvalues of $\nabla^2 L$:

$$\det \begin{bmatrix} 2-\lambda & -1 \\ -1 & -\lambda \end{bmatrix} = (\lambda - 2)\lambda - 1 = \lambda^2 - 2\lambda - 1 = 0.$$

$$\Rightarrow \quad \lambda = \frac{2 \pm \sqrt{4+4}}{2} = \frac{2 \pm 2\sqrt{2}}{2} = 1 \pm \sqrt{2}.$$

One eigenvalue is positive, and the other is negative $\Rightarrow \nabla^2 L$ is indefinite.

Another way to check $\nabla^2 L$ is indefinite: find two vectors u, v st $u^T \nabla^2 L u > 0$ and $v^T \nabla^2 L v < 0$. For ">0", choice is easy. Let $u = (1, 0)$. Then $u^T \nabla^2 L u = 2 > 0$. For v, let $v = (\varepsilon, 1)$. Then:

$$v^T \nabla^2 L v = \begin{bmatrix} \varepsilon \\ 1 \end{bmatrix}^T \begin{bmatrix} 2 & -1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} \varepsilon \\ 1 \end{bmatrix} = \begin{bmatrix} \varepsilon \\ 1 \end{bmatrix}^T \begin{bmatrix} 2\varepsilon - 1 \\ -\varepsilon \end{bmatrix} = 2\varepsilon^2 - 2\varepsilon = 2\varepsilon(\varepsilon - 1).$$

So, if $\varepsilon < 1$, then $v^T \nabla^2 L v < 0$.

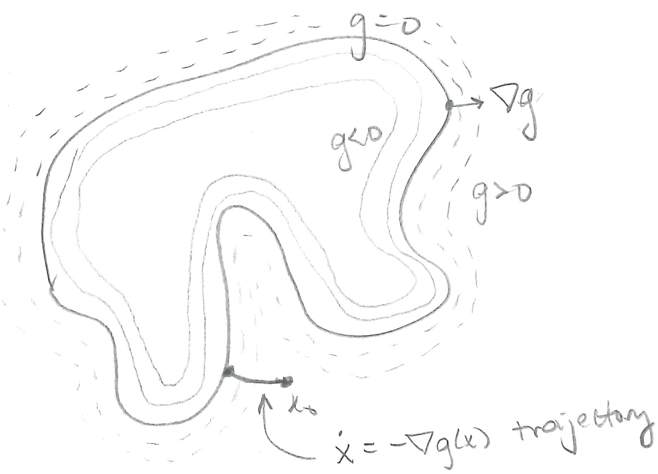
One way to proceed is to simply require each iterate to be feasible. Then, progress is ensured, since:

$$L(x_{k+1}, \lambda_{k+1}) = f(x_{k+1}) < f(x_k) = L(x_k, \lambda_k).$$

You used an approach a bit like this for the Thomson problem, where you used projected gradient descent to keep iterates feasible.

This idea can be made a bit more systematic as follows.

Recall that since g is a level set function, ∇g is the normal vector for the surface. E.g.:



Provided that g is C^1, at each point ∇g gives us a local direction to follow to reach the $g=0$ level set. For example, under suitable assumptions, the flow $\dot{x} = \nabla g(x)$ will reach $g=0$ if $g(x_0) < 0$, and $\dot{x} = -\nabla g(x)$ will go to $g=0$ if $g(x_0) > 0$.

A way we can use this idea is:

1) For feasible x_k, solve Newton system for (p_k, ν_k).
2) Set $\tilde{x}_{k+1} = x_k + p_k$, $\tilde{\lambda}_{k+1} = \tilde{\lambda}_k + \tilde{\nu}_k$.
3) Find t st $g(\tilde{x}_{k+1} + t\nabla g(\tilde{x}_{k+1})) = 0$.

4) Set $x_{k+1} = \tilde{x}_{k+1} + t\nabla g(\tilde{x}_{k+1})$.

5) Solve for λ_{k+1} with x_{k+1} fixed : From KKT stationary condition, have $\nabla f(x_{k+1}) + \nabla g(x_{k+1})\lambda_{k+1} = 0$.
Since $\nabla g(x_{k+1}) \in \mathbb{R}^{n\times m}$ with $m<n$, solve normal equations to get: $\lambda_{k+1} = -\left(\nabla g(x_{k+1})^T \nabla g(x_{k+1})\right)^{-1} \nabla g(x_{k+1})^T \nabla f(x_{k+1})$.

Question : do we actually need to compute λ_k above? What role does it play?

Simple case: $m=1 \Rightarrow g(x)=0$ defines a $n-1$ dimensional hypersurface.
Then,

$$\nabla^2 L = \begin{bmatrix} \nabla^2 f + \lambda \nabla^2 g & \nabla g \\ \nabla g^T & 0 \end{bmatrix}, \quad \nabla L = \begin{bmatrix} \nabla f + \lambda \nabla g \\ g \end{bmatrix}$$

Try using block Gaussian elimination to solve. (Exercise.)

Other ways of solving: saw interior point methods already.

Merit functions: Replace (*) with:

$$\text{minimize} \quad \mathcal{M}(x) = f(x) + \rho\|g(x)\|$$

where $\rho > 0$. Here, $\mathcal{M}(x)$ is called a merit function.

Usually hard to choose the parameter ρ. For fixed ρ, minimizing $\mathcal{M}$ will have a tendency to minimize $\|g(x)\|$, too, which encourages $g(x) \approx 0$ — but this is no guarantee. Usually, ρ is changed from iteration to iteration.

<u>Example</u>: Consider the minimization problem:

$$\begin{aligned} &\text{minimize} \quad f(x) \\ &\text{subject to} \quad \|x\|^2 \leq r^2 \end{aligned}$$

Using a Merit function, we get the unconstrained optimization problem:

$$\text{minimize} \quad f(x) + \rho \|x\|^2.$$

This is just an ℓ_2 regularized optimization problem.