Local Computation

ME555 Lecture

(Max) Yi Ren

Department of Mechanical Engineering, University of Michigan

March 23, 2014

Outline

1. Global and local convergence
2. Line search
3. Quasi-Newton Methods
4. Active set strategies
5. Penalty and barriers
6. Augmented Lagrangian
7. Sequential quadratic programming

Global and local convergence

Global convergence refers to the ability of the algorithm to reach the neighborhood of some local solution $\mathbf{x}_*$ from an arbitrary initial point $\mathbf{x}_0$, which is not close to $\mathbf{x}_*$. The convergence of a globally convergent algorithm should not be affected by the choice of initial point.

Local convergence refers to the ability of the algorithm to approach $\mathbf{x}_*$, rapidly from a point in the neighborhood of $\mathbf{x}_*$.

convergence ratio γ:

- Linear convergence: $||\mathbf{x}_{k+1} - \mathbf{x}_*|| \leq \gamma||\mathbf{x}_k - \mathbf{x}_*||, 0 < \gamma < 1$
- Quadratic convergence: $||\mathbf{x}_{k+1} - \mathbf{x}_*|| \leq \gamma||\mathbf{x}_k - \mathbf{x}_*||^2, \gamma \in \mathbb{R}$

Newton's method has quadratic convergence rate but is not globally convergent; Gradient descent has global convergence but in some cases can be inefficient.

Line search: Bisection

Bisection is used to find the root of a single-variable function. We can apply this method to the derivative of a function to find its stationary point (and local minimum, how?)

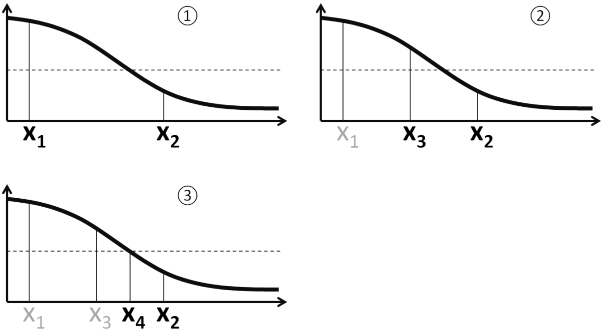


Figure: Bisection

Line search: Interpolation

A line search can also be carried out by interpolation. This is a simpler version of the derivative-free response surface method for optimization. The following figure shows the procedure of finding a local minimum by a series of quadratic interpolations.

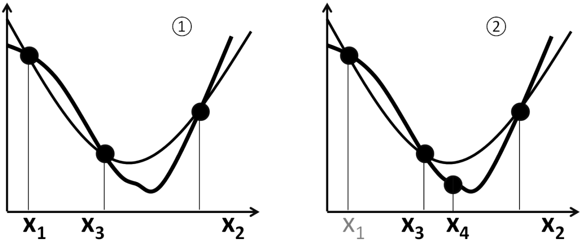


Figure: Bisection

Inexact line search

Recall that in an iterative search process we update with

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{s}_k,$$

where $\mathbf{s}_k$ is the current search direction and α_k is the step size. An exact line search finds the optimal α_k (in the feasible domain) but can be costly and unnecessary. Therefore an inexact line search is more commonly used that finds an acceptable sufficient decrease from f_k.

The Armijo-Goldstein criteria

$$f_k + \varepsilon \alpha \mathbf{g}_k^T \mathbf{s}_k \geq f_{k+1}$$
$$f_k + (1 - \varepsilon) \alpha \mathbf{g}_k^T \mathbf{s}_k \leq f_{k+1}$$

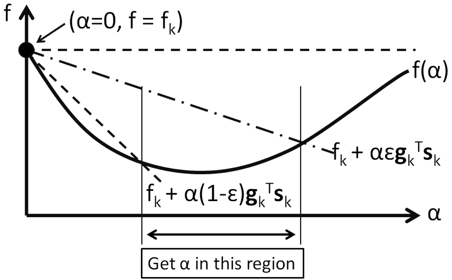


Figure: Armijo-Goldstein criteria

Armijo line search

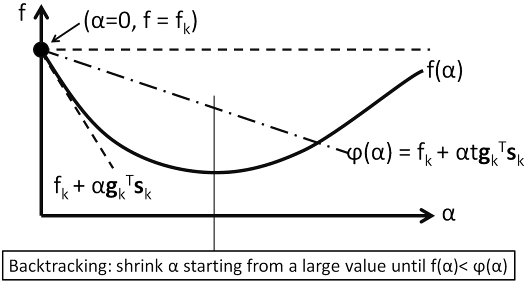


Figure: Armijo rule

In Armijo line search, we construct a function

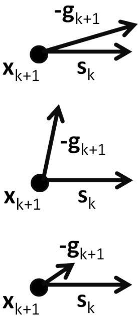
$$\phi(\alpha) = f_k + \alpha \varepsilon \mathbf{g}_k^T \mathbf{s}_k,$$

and denote $f(\alpha) := f(\mathbf{x}_k + \alpha \mathbf{s}_k)$. Starting with a large value, α is halved until $f(\alpha) < \phi(\alpha)$, at which point it is guaranteed that $f(\alpha) < f_k$, since $\phi(\alpha) < f_k$ by nature. Practical values for ε are $10^{-}1$ to $10^{-}4$.

Curvature condition

The following curvature condition ensures that the slope has been reduced sufficiently:

$$\mathbf{s}_k^T \mathbf{g}_{k+1} \geq \varepsilon_2 \mathbf{s}_k^T \mathbf{g}_k.$$



$\mathbf{s}_k$ is still close to $\mathbf{g}_{k+1}$, keep searching in this direction

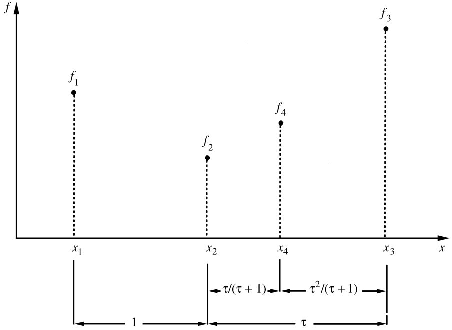
$\mathbf{s}_k$ is different from $\mathbf{g}_{k+1}$, stop searching

$\mathbf{g}_{k+1}$ is small, stop searching

The Armijo rule and curvature condition together constitutes the Wolfe conditions and are necessary to guarantee convergence. We will revisit the curvature condition in Quasi-Newton methods.

Exercise 7.10

A popular purely sectioning procedure is the *golden section search.* Consider a function as shown in the figure with three points x_1, x_2, x_3 already placed with the ratio $(x_3 - x_2)/(x_2 - x_1)$ fixed at τ, where $\tau > 1$ is a constant. Insert the fourth trial point x_4 so that both *potential* new brackets $[x_1, x_4]$ and $[x_2, x_3]$ have intervals in the same ratio τ. Based on this construction, evaluate the lengths of the intervals $[x_2, x_4]$ and $[x_4, x_3]$. Prove that $\tau \cong 1.618$. Write out the steps of an algorithm based on repeating this sectioning, noting that in the figure the interval $[x_4, x_3]$ is discarded.



Exercise 7.13

Consider the function $f = 1 - x\exp(-x)$.

1. Find the minimum using the golden section method, terminating when $|x_{k+1} - x_k| < 0.1$ and starting from [0, 2].
2. Find the value(s) for x that satisfy the Armijo–Goldstein criteria with ε_1=0.1.
3. Find a value for x using the Armijo Line Search of Example 7.5.

Quasi-Newton methods

Recall the Newton's method

$$\mathbf{x}_{k+1} = \mathbf{x}_k - \alpha_k \mathbf{H}_k^{-1} \mathbf{g}_k,$$

where $\mathbf{H}_k$ and $\mathbf{g}_k$ are the Hessian and gradient at iteration k. The method has quadratic convergence when $\mathbf{H}_k$ remains positive-definite. Quasi-Newton methods build a positive-definite approximation of the Hessian using f_k and $\mathbf{g}_k$, and is regarded as the best general-purpose methods for solving unconstrained problems.

Calculating $\mathbf{H}_k$ can be time consuming. Therefore we wish to approximate $\mathbf{H}_k$ as $\hat{\mathbf{H}}_k$ iteratively:

$$\hat{\mathbf{H}}_{k+1} = \hat{\mathbf{H}}_k + \text{something},$$

to get the second-order approximation at $\mathbf{x}_{k+1}$:

$$f(\mathbf{x}) = f(\mathbf{x}_{k+1}) + \mathbf{g}_{k+1}^T \partial \mathbf{x}_{k+1} + \frac{1}{2} \partial \mathbf{x}_{k+1}^T \hat{\mathbf{H}}_{k+1} \partial \mathbf{x}_{k+1}, \tag{1}$$

where $\partial \mathbf{x}_{k+1} = \mathbf{x} - \mathbf{x}_{k+1}$.

The DFP method (1/3)

Three conditions need to be imposed on Equation (1).

First, $\hat{\mathbf{H}}$ needs to be symmetric and positive-definite.

Second, the approximated $f(\mathbf{x})$ must match the true gradients at $\mathbf{x}_k$ and $\mathbf{x}_{k+1}$. For $\mathbf{x}_{k+1}$, the approximation from Equation (1) naturally follows that

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}|_{k+1} = \mathbf{g}_{k+1}^T.$$

Therefore the approximated gradient is the true gradient at $\mathbf{x}_{k+1}$.

For $\mathbf{x}_k$, considering a general search $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{s}_k$, we have

$$\frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}|_{k} = \mathbf{g}_{k+1}^T - \alpha_k \mathbf{s}_k^T \mathbf{H}_{k+1}.$$

By rearranging terms we have

$$\mathbf{g}_{k+1} - \mathbf{g}_k = \alpha_k \mathbf{H}_{k+1} \mathbf{s}_k \qquad (2)$$

Equation (2) is called the **secant equation** and key to approximate $\mathbf{H}_{k+1}$.

The DFP method (2/3)

Multiply both ends of the secant equation by $\mathbf{s}_k^T$ to have

$$\mathbf{s}_k^T(\mathbf{g}_{k+1} - \mathbf{g}_k) = \alpha_k \mathbf{s}_k^T \mathbf{H}_{k+1} \mathbf{s}_k > 0,$$

when the Hessian is positive-definite. This leads to the curvature condition.

In fact, the curvature condition ensures a positive-definite approximation $\hat{\mathbf{H}}$ of $\mathbf{H}$.

However, there are infinitely many symmetric positive-definite matrices that satisfy the secant equation.

The last condition: We will select $\hat{\mathbf{H}}_{k+1}$ that is closest to $\hat{\mathbf{H}}_k$ in the weighted Frobenius norm. Overall, we find $\hat{\mathbf{H}}_{k+1}$ that solves the following convex problem

$$\begin{aligned} \min_{\mathbf{H}} \quad & ||\hat{\mathbf{H}} - \hat{\mathbf{H}}_k||_F \\ \text{subject to} \quad & \hat{\mathbf{H}} = \hat{\mathbf{H}}^T \\ & \mathbf{g}_{k+1} - \mathbf{g}_k = \alpha_k \hat{\mathbf{H}} \mathbf{s}_k \end{aligned}$$

The DFP method (3/3)

Solve Problem (14) and denote $\mathbf{B} = \mathbf{H}^{-1}$ to have the DFP update

$$\mathbf{B}_{k+1}^{\text{DFP}} = \mathbf{B}_k + \left[\frac{\partial\mathbf{x}\partial\mathbf{x}^T}{\partial\mathbf{x}^T\partial\mathbf{g}}\right]_k - \left[\frac{(\mathbf{B}\partial\mathbf{g})(\mathbf{B}\partial\mathbf{g})^T}{\partial\mathbf{g}^T\mathbf{B}\partial\mathbf{g}}\right]_k, \tag{3}$$

where $\partial\mathbf{x}_k = \alpha\mathbf{s}_k$ and $\partial\mathbf{g} = \mathbf{g}_{k+1} - \mathbf{g}_k$.

The Davidon-Fletcher-Powell (DFP) method was originally proposed by W.C. Davidon in 1959.

The BFGS method (1/3)

Instead of imposing conditions on the Hessian as in DFP, the BFGS method directly work on the inverse of the Hessian. The secant equation therefore is in the form

$$\mathbf{B}_{k+1}(\mathbf{g}_{k+1} - \mathbf{g}_k) = \alpha_k \mathbf{s}_k.$$

The revised conditions lead to the BFGS update

$$\mathbf{B}_{k+1}^{BFGS} = \mathbf{B}_k + \left[1 + \frac{\partial \mathbf{g}^T \mathbf{B} \partial \mathbf{g}}{\partial \mathbf{x}^T \partial \mathbf{g}}\right]_k \left[\frac{\partial \mathbf{x} \partial \mathbf{x}^T}{\partial \mathbf{x}^T \partial \mathbf{g}}\right]_k - \left[\frac{\partial \mathbf{x} \partial \mathbf{g}^T \mathbf{B} + \mathbf{B} \partial \mathbf{g} \partial \mathbf{x}^T}{\partial \mathbf{x}^T \partial \mathbf{g}}\right]_k.$$

The Broyden-Fletcher-Goldfarb-Shanno (BFGS) method is more commonly used than DFP.

The BFGS method (2/3)

There are several ways to set the initial value for $\mathbf{H}$ (or $\mathbf{B}$):

- A finite difference approximation at $\mathbf{x}_0$.
- Use the identity matrix.
- Use $\text{diag}(\lambda_1, \lambda_2, ...)$, where $\boldsymbol{\lambda}$ captures the scaling of the variables.

The BFGS method (3/3)

Algorithm: The BFGS method in unconstrained optimization

Given starting point $\mathbf{x}_0$, convergence tolerance $\varepsilon > 0$, and initial inverse Hessian approximation $\mathbf{B}_0$:

while $||\mathbf{g}_k|| > \varepsilon$

1. $\mathbf{s}_k = -\mathbf{B}_k\mathbf{g}_k$
2. $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k\mathbf{s}_k$, where α_k comes from a line search following the Wolfe conditions
3. Let $\partial\mathbf{x} = \mathbf{x}_{k+1} - \mathbf{x}_k$ and $\partial\mathbf{g} = \mathbf{g}_{k+1} - \mathbf{g}_k$
4. $\mathbf{B}_{k+1}^{BFGS} = \mathbf{B}_k + \left[1 + \frac{\partial\mathbf{g}^T\mathbf{B}\partial\mathbf{g}}{\partial\mathbf{x}^T\partial\mathbf{g}}\right]_k \left[\frac{\partial\mathbf{x}\partial\mathbf{x}^T}{\partial\mathbf{x}^T\partial\mathbf{g}}\right]_k - \left[\frac{\partial\mathbf{x}\partial\mathbf{g}^T\mathbf{B}+\mathbf{B}\partial\mathbf{g}\partial\mathbf{x}^T}{\partial\mathbf{x}^T\partial\mathbf{g}}\right]_k$
5. k = k + 1

end-while

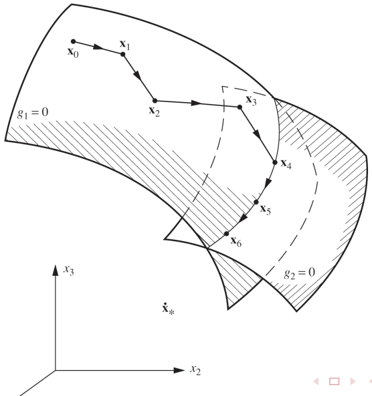
Active set strategy (1/4)

We now discuss how we apply a reduced gradient algorithm to a problem with *inequality* constraints. The difficulty is that we do not know at the beginning which inequality constraints will be active at an optimal solution. The strategy is to maintain a *working set* of active constraints (along with equality constraints) and keep adding or deleting constraints in this working set.

Active set strategy (2/4)

Adding constraints

Starting at an initial feasible point and an initial working set, we minimize the objective function subject to the equalities in the working set. When hitting a new inequality constraint (here moving from $\mathbf{x}_3$ to $\mathbf{x}_4$), that constraint will be added to the working set and the step size is reduced to retain feasibility.



Active set strategy (3/4)

Removing constraints

When arrived at a point where no progress is possible by adding constraints, we check the KKT conditions and estimate the Lagrangian multipliers (since we may not have arrived at an optimal solution yet, these multipliers are only estimated). If the Lagrangian multipliers for some active constraints are negative, these constraints will become candidate for deletion. A common heuristic is to delete one constraint with the most negative multiplier. (Why?)

Active set strategy with GRG (4/4)

The active set algorithm

1. Input initial feasible point and working set.
2. Compute a feasible search vector $\mathbf{s}_k$.
3. Compute a step length α_k along $\mathbf{s}_k$, such that $f(\mathbf{x}_k + \alpha_k \mathbf{s}_k) < f(\mathbf{x}_k)$. If α_k violates a constraint, continue; otherwise go to 6.
4. Add a violated constraint to the constraint set and reduce α_k to the maximum possible value that retains feasibility.
5. Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \mathbf{s}_k$.
6. Check the norm of reduced gradient. If not zero, go to step 2. Otherwise, check if estimates of Lagrangian multipliers for active constraints are positive or not. If not all positive, delete a constraint that has the most negative multiplier, and go to step 2. Otherwise, terminate.

Barrier method (1/2)

Instead of solving the constrained problem, we can construct a *barrier function* to be optimized

$$T(\mathbf{x}, r) := f(\mathbf{x}) + rB(\mathbf{x}), \quad r > 0,$$

where $B(\mathbf{x}) := -\sum_{j=1}^{m} ln[-g_j(\mathbf{x})]$ (logarithmic) or $B(\mathbf{x}) := -\sum_{j=1}^{m} g_j^{-1}(\mathbf{x})$ (inverse). The barrier method only works for problems with inequality constraints.

Barrier function algorithm

1. Find an interior point $\mathbf{x}_0$. Select a monotonically decreasing sequence $\{r_k\} \to 0$ for $k \to \infty$. Set $k = 0$.
2. At iteration k, minimize the function $T(\mathbf{x}, r_k)$ using an unconstrained method and $\mathbf{x}_k$ as the starting point. the solution $\mathbf{x}_*(r_k)$ is set equal to $\mathbf{x}_{k+1}$.
3. Perform a convergence test. If the test is not satisfied, set $k = k + 1$ and return to step 2.

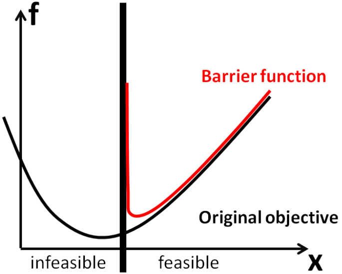
Barrier method (2/2)

When using the barrier method, we can estimate Lagrange multipliers as

- $\mu_j(r_k) = -r_k/g_j$, (for the logarithmic barrier)
- $\mu_j(r_k) = -r_k/g_j^2$ (for the inverse barrier)

The actual multiplier can be obtained at the limit.

Note that this basic barrier method has a major computational difficulty: A small r_k leads to an ill-conditioned Hessian, making the optimization difficult. (What can we do?)



Penalty method (1/2)

A typical penalty function has the form

$$T(\mathbf{x}, r) := f(\mathbf{x}) + r^{-1}P(\mathbf{x}), \quad r > 0,$$

where the *penalty function* $P(\mathbf{x})$ can take a quadratic form

$$P(\mathbf{x}) := \sum_{j=1}^{m} [\max\{0, g_j(\mathbf{x})\}]^2$$

for inequality constraints, and

$$P(\mathbf{x}) := \sum_{j=1}^{m} [h_j(\mathbf{x})]^{\mathbf{2}}$$

for equality constraints.

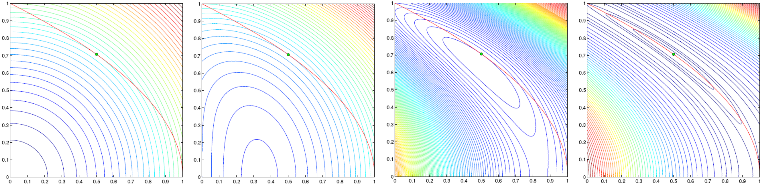
Lagrange multipliers can be estimated as

$$\mu_j(r_k) = (2/r_k)\max\{0, g_j(\mathbf{x})_{\mathbf{k}}\}$$

for a decreasing sequence $\{r_k\}$.

Penalty method (2/2)

An example of an ill-conditioned Hessian when using the penalty method (from lecture notes of Nick Gould)



(a) $r = 100$ (b) $r = 1$ (c) $r = 0.1$ (d) $r = 0.01$

Figure: Quadratic penalty function for min $x_1^2 + x_2^2$ subject to $x_1 + x_2^2 = 1$

Augmented Lagrangian (1/6)

Recall that the drawback of the penalty method (as well as the barrier method) is that we can only find a good approximation of the true solution when the penalty is high, i.e., $r \to 0$, in which case the convergence of the problem will suffer from ill-conditioned Hessian.

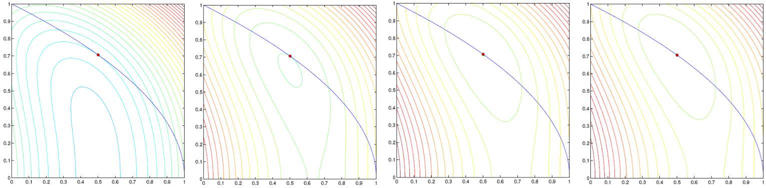
With that in mind, we introduce the augmented Lagrangian function:

$$\Phi(\mathbf{x}, \boldsymbol{\lambda}, r) = f(\mathbf{x}) + \boldsymbol{\lambda}^T \mathbf{h}(\mathbf{x}) + \frac{1}{r} ||\mathbf{h}(\mathbf{x})||^2$$

When the Lagrangian multipliers $\boldsymbol{\lambda}$ are close to their true values, a reasonable small value of r allows us to find the true optimal solution $\mathbf{x}$ without encountering an ill-conditioned Hessian.

Augmented Lagrangian (2/6)

An example where we can find the true optimal solution for a constrained problem without setting $r \to 0$. (When we guessed correctly on λ_*, we can find the solution $\mathbf{x}_*$ without $r \to 0$)



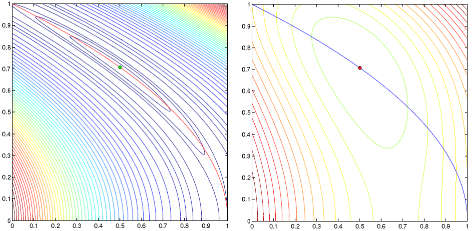
(a) $\lambda = -0.5$ (b) $\lambda = -0.9$ (c) $\lambda = -0.99$ (d) $\lambda = \lambda_* = -1$

Figure: Augmented Lagrangian function for min $x_1^2 + x_2^2$ subject to $x_1 + x_2^2 = 1$ with fixed $r = 1$

Augmented Lagrangian (3/6)

Two ways of understanding augmented Lagrangian under equality constraints only:

1. **Shifted quadratic penalty function**: The augmentation shifts the origin of the penalty term so that an optimum value for the transformed function can be found without the penalty parameter going to the limit.



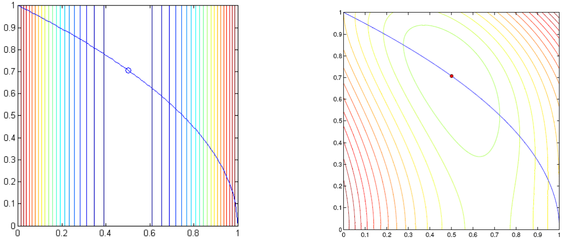
(a) $r = 0.01$ (b) $\lambda = \lambda_* = -1, r = 1$

Figure: Comparison between penalty and Augmented Lagrangian

Augmented Lagrangian (4/6)

Two ways of understanding augmented Lagrangian under equality constraints only:

2. **Convexification of the Lagrangian function**: If $\mathbf{x}_*$ is a local solution for the original problem, it follows KKT conditions and the second-order sufficiency condition (which is what?). We want to construct a unconstrained problem which has $\mathbf{x}_*$ as its local solution. The first-order necessary condition of this problem should be the original KKT condition and its Hessian should be positive-definite. The augmented Lagrangian function satisfies these requirements.



(a) $\lambda = \lambda_* = -1, r = \infty$ (b) $\lambda = \lambda_* = -1, r = 1$

Augmented Lagrangian (5/6)

The augmented Lagrangian method requires tuning of $\boldsymbol{\lambda}$ and r together in some way so that $\{\mathbf{x}_k\} \to \mathbf{x}_*$.

- Check if $||\mathbf{h}(\mathbf{x})|| \leq \eta_k$ where $\{\eta_k\} \to 0$.
  - if so, set $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + 2\mathbf{h}(\mathbf{x}_k)/r_k$ and $r_{k+1} = r_k$. It is proved that such a series $\{\boldsymbol{\lambda}_k\}$ converges to $\boldsymbol{\lambda}_*$.
  - if not, set $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k$ and $r_{k+1} = \tau r_k$ for some $\tau \in (0, 1)$. Often choose $\tau = \min\{0.1, \sqrt{r_k}\}$.
- update on η_k: $\eta_k = r_k^{0.1+0.9j}$ where j iterations since r_k last changed.

Augmented Lagrangian (6/6)

The augmented Lagrangian algorithm (from Nick Gould's lecture notes)

1. Given $r_0 > 0$ and $\boldsymbol{\lambda}_0$, set $k = 0$
2. While KKT conditions are not met
   - 2.1 Starting from $\mathbf{x}_k^s = \mathbf{x}_{k-1}$, use an unconstrained minimization algorithm to find an "approximate" minimizer $\mathbf{x}_k$ so that $||\nabla_{\mathbf{x}}\Phi(\mathbf{x}_k, \boldsymbol{\lambda}_k, r_k)|| \leq \varepsilon_k$
   - 2.2 If $||\mathbf{h}(\mathbf{x}_k)|| \leq \eta_k$, set $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k + 2\mathbf{h}(\mathbf{x}_k)/r_k$ and $r_{k+1} = r_k$
   - 2.3 Otherwise set $\boldsymbol{\lambda}_{k+1} = \boldsymbol{\lambda}_k$ and $r_{k+1} = \tau r_k$
   - 2.4 $k = k + 1$. Set $\varepsilon_k = r_k^{j+1}$ and $\eta_k = r_k^{0.1+0.9j}$ where j iterations since r_k last changed

This method can be extended to inequalities with the aid of an active set strategy. Details of implementation can be found in Pierre and Lowe (1975) and Bertsekas (1982). An alternative way proposed by Nocedal and Wright (2006) is to convert inequalities to equalities by introducing slack variables, which can be optimized separately and eliminated.

Exercise 7.27

Using the penalty transformation $T = f + \frac{1}{2} r\mathbf{h}^T\mathbf{h}$, evaluate and sketch the progress of the penalty method (sequential unconstrained minimization) for the problem $\{\min f = x$, subject to $h = x - 1 = 0\}$, with $r = 1, 10, 100, 1000$. Repeat, using the augmented Lagrangian transformation $T = f + \boldsymbol{\lambda}^T\mathbf{h} + \frac{1}{2} r\mathbf{h}^T\mathbf{h}$. (From Fletcher 1981.)

Exercise 7.27 Solution

Penalty method: $\min x + 1/2r(x-1)^2$

Stationary point: $x_*(r) = 1 - 1/r$

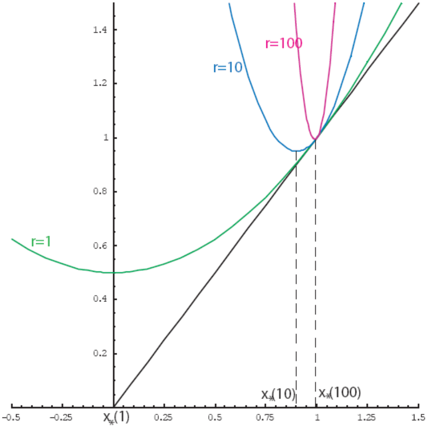
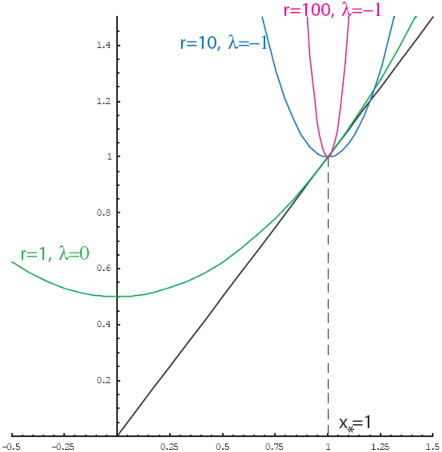


Figure: Penalty method

Exercise 7.27 Solution

Augmented Lagrangian method: $\min x + \lambda(x-1) + 0.5r(x-1)^2$

Stationary point: $x_*(r) = 1 - 1/r(1+\lambda)$. Let $\lambda_0 = 0$, $r_0 = 1$, we have $x_* = 0$, $h_0 = -1$. Then update $\lambda_1 = \lambda_0 + r_0 h_0 = -1$ and $r_1 = r_0 = 1$. Then we have $x_* = 1$ and $h_1 = 0$.



What have we learned so far?

- Unconstrained optimization
  - gradient descent with line search
  - Newton's method with line search
  - trust region (why?)
  - quasi-Newton (why?)
- Constrained optimization
  - generalized reduced gradient
  - barrier and penalty (why not?)
  - augmented Lagrangian
  - (active set)

Comparisons

- Active set method should be attached to other algorithms and thus will not be compared with.
- GRG is the most reliable but requires the most implementation effort. It is also not the most efficient and requires a lot of function evaluation.
- Augmented Lagrangian is less reliable than GRG. A widely used package of this method is LANCELOT, which deals with large-scale optimization problems with bounds and equality constraints. The idea of augmented Lagrangian is also used in SQP type of algorithm to improve line search and Hessian approximation.
- SQP is the most widely used algorithm and can deal with large-scale problems (up to the scale of 10000 variables and constraints). It is more reliable than augmented Lagrangian and more efficient than GRG.

The Lagrange-Newton Equations (1/2)

Consider the equality constrained problem

$$\begin{aligned} \min \quad & f(\mathbf{x}) \\ \text{subject to} \quad & \mathbf{h}(\mathbf{x}) = \mathbf{0} \end{aligned} \tag{4}$$

The stationary condition for this problem is

$$\nabla L(\mathbf{x}_*, \boldsymbol{\lambda}_*) = \mathbf{0}^T.$$

We may solve this equation using Newton-Ralphson to update $\mathbf{x}$ and $\boldsymbol{\lambda}_*$:

$$[\nabla L(\mathbf{x}_k + \partial\mathbf{x}_k, \boldsymbol{\lambda}_k + \partial\boldsymbol{\lambda}_k)]^T = \nabla L_k^T + \nabla^2 L_k(\partial\mathbf{x}_k, \partial\boldsymbol{\lambda}_k)^T, \tag{5}$$

where $\nabla L_k^T = \nabla f_k^T + \nabla \mathbf{h}_k^T \boldsymbol{\lambda}_k$ and $\nabla^2 L_k = [\nabla^2 f + \boldsymbol{\lambda}^T \nabla^2 \mathbf{h},\ \nabla \mathbf{h}^T;\ \nabla \mathbf{h},\ \mathbf{0}]_k$.

The Lagrange-Newton Equations (2/2)

Define $\mathbf{W} = \nabla^2 f + \boldsymbol{\lambda}^T \nabla^2 \mathbf{h}$ and $\mathbf{A} = \nabla \mathbf{h}$ to have

$$\nabla^2 L_k = [\mathbf{W}\ \mathbf{A}^T; \mathbf{A}\ \mathbf{0}]_k.$$

Denote the step as $\mathbf{s}_k := \partial \mathbf{x}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$ and set the left-hand side of Equation (5) to zero to have

$$\begin{aligned} \mathbf{W}_k \mathbf{s}_k + \mathbf{A}_k^T \boldsymbol{\lambda}_{k+1} + \nabla f_k^T &= \mathbf{0} \\ \mathbf{A}_k \mathbf{s}_k + \mathbf{h}_k &= \mathbf{0} \end{aligned} \tag{6}$$

Equation (6) is referred to as a Lagrange-Newton method for solving the constrained problem (4).

What are the conditions of $\mathbf{W}_*$ and $\mathbf{A}_*$ for the solution to be unique?

Quadratic programming subproblem

Note that Equation (6) can be viewed as the KKT conditions for the quadratic model

$$\begin{aligned} \min_{\mathbf{s}_k} \quad & q(\mathbf{s}_k) = f_k + \nabla_{\mathbf{x}} L_k \mathbf{s}_k + \frac{1}{2}\mathbf{s}_k^T \mathbf{W}_k \mathbf{s}_k \\ \text{subject to} \quad & \mathbf{A}_k \mathbf{s}_k + \mathbf{h}_k = \mathbf{0}, \end{aligned} \tag{7}$$

where $\nabla_{\mathbf{x}} L_k = \nabla f_k + \boldsymbol{\lambda}_k^T \nabla \mathbf{h}_k$ and the multiplier of problem (7) are $\partial \boldsymbol{\lambda}_k$.

It can be shown that solving the Lagrange-Newton equations from Equation (6) is equivalent to solving the quadratic programming subproblem (7).

Another equivalent QP subproblem is as follows

$$\begin{aligned} \min_{\mathbf{s}_k} \quad & q(\mathbf{s}_k) = f_k + \nabla f_k \mathbf{s}_k + \frac{1}{2}\mathbf{s}_k^T \mathbf{W}_k \mathbf{s}_k \\ \text{subject to} \quad & \mathbf{A}_k \mathbf{s}_k + \mathbf{h}_k = \mathbf{0}, \end{aligned} \tag{8}$$

which also gives a solution $\mathbf{s}_k$ with multipliers $\boldsymbol{\lambda}_{k+1}$ directly, rather than $\partial \boldsymbol{\lambda}_k$. What is the meaning of this QP subproblem?

SQP Algorithm (without line search)

1. Select initial point $\mathbf{x}_0$, $\boldsymbol{\lambda}_0$; let $k = 0$.
2. For $k = k + 1$, solve the QP subproblem and determine $\mathbf{s}_k$ and $\boldsymbol{\lambda}_{k+1}$.
3. Set $\mathbf{x}_{k+1} = \mathbf{x}_k + \mathbf{s}_k$.
4. If KKT condition not satisfied, return to 2.

Advantage:

- Simple
- Fast, locally quadratically convergent

Enhancements of the basic algorithm (1/3)

The basic SQP algorithm may not have global convergence. For points far from $\mathbf{x}_*$, the QP subproblem may have an unbounded solution.

It is shown that for the QP subproblem to have a well-defined solution, the following is needed:

- $\mathbf{A}$ has full rank
- $\mathbf{W}$ has to be positive definite in feasible perturbations

One possibility is to use the QP solution $\mathbf{s}_k$ as a search direction and find the step size α_k that minimizes a *merit function*, which is a penalty function that properly weighs objective function decrease and constraint violations.

Enhancements of the basic algorithm (2/3)

One merit function (exact penalty function, Powell 1978a) that is widely implemented has the following form

$$\phi(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \sum_{j=1}^{m_1} w_j |h_j| + \sum_{j=1}^{m_2} w_j |\max\{0, g_j\}|,$$

where m_1 and m_2 are the numbers of equality and inequality constraints and w_j are weights used to balance the infeasibilities. The suggested values are

$$\begin{aligned} w_{j,0} &= |\lambda_j| \quad \text{for } k = 0 \text{ first iteration}, \\ w_{j,k} &= \max\{|\lambda_{j,k}|, 0.5(w_{j,k-1} + |\lambda_{j,k}|)\} \quad \text{for } k \geq 1, \end{aligned}$$

where μ_j would be used for the inequalities.

One can also use a quadratic penalty function

$$\phi(\mathbf{x}, \boldsymbol{\lambda}, \boldsymbol{\mu}) = f(\mathbf{x}) + \sum_{j=1}^{m_1} w_j h_j^2 + \sum_{j=1}^{m_2} w_j (\max\{0, g_j\})^2.$$

Enhancements of the basic algorithm (3/3)

The evaluation of $\mathbf{W}_k$ can be approximated by a quasi-Newton method. The BFGS approximation of a positive-definite $\mathbf{W}_k$ is as follows:

$$\partial \mathbf{g}_k = \theta_k \mathbf{y}_k + (1-\theta_k)\hat{\mathbf{W}}_k \partial \mathbf{x}_k, \quad 0 \le \theta \le 1,$$

where

$$\mathbf{y}_k = \nabla L(\mathbf{x}_{k+1}, \boldsymbol{\lambda}_{k+1})^T - \nabla L(\mathbf{x}_k, \boldsymbol{\lambda}_{k+1})^T,$$

and

$$\theta_k = \begin{cases} 1 & \text{if } \partial \mathbf{x}_k^T \mathbf{y}_k \ge (0.2)\partial \mathbf{x}_k^T \hat{\mathbf{W}}_k \partial \mathbf{x}_k, \\ \frac{(0.8)\partial \mathbf{x}_k^T \hat{\mathbf{W}}_k \partial \mathbf{x}_k}{\partial \mathbf{x}_k^T \hat{\mathbf{W}}_k \partial \mathbf{x}_k - \partial \mathbf{x}_k^T \mathbf{y}_k} & \text{if } \partial \mathbf{x}_k^T \mathbf{y}_k < (0.2)\partial \mathbf{x}_k^T \hat{\mathbf{W}}_k \partial \mathbf{x}_k, \end{cases}$$

and $\partial \mathbf{x}_k = \mathbf{x}_{k+1} - \mathbf{x}_k$, $\hat{\mathbf{W}}_k$ is the current BFGS approximation to the Hessian of the Lagrangian.

Extension to inequalities

Active set can be applied in two ways

- An active set strategy may be employed on the original problem so that the QP subproblem always have only equality constraints.
- The second way is to pose the QP subproblem with the linearized inequalities included ($\mathbf{A}_k\mathbf{s}_k + \mathbf{g}_k \leq \mathbf{0}$), and use an active set strategy on the subproblem.

The merit function must then include all constraints, active and inactive, to guard against failure when the wrong active set is used to determine the step direction.

Solving the quadratic subproblem

consider the QP problem

$$\begin{aligned} \min \quad & \frac{1}{2}\mathbf{x}^T\mathbf{Q}\mathbf{x} + \mathbf{c}^T\mathbf{x} \\ \text{subject to} \quad & \mathbf{A}\mathbf{x} - \mathbf{b} = \mathbf{0}. \end{aligned} \tag{9}$$

The Lagrange-Newton equations (KKT conditions) for this problem is

$$\begin{pmatrix} \mathbf{Q} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{pmatrix} \begin{pmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} -\mathbf{c} \\ \mathbf{b} \end{pmatrix}$$

When the Lagrangian matrix is invertible (e.g., $\mathbf{Q}$ positive-definite and $\mathbf{A}$ full rank), the solution to the QP problem is

$$\begin{pmatrix} \mathbf{x} \\ \boldsymbol{\lambda} \end{pmatrix} = \begin{pmatrix} \mathbf{Q} & \mathbf{A}^T \\ \mathbf{A} & \mathbf{0} \end{pmatrix}^{-1} \begin{pmatrix} -\mathbf{c} \\ \mathbf{b} \end{pmatrix}$$

exercise 7.30

Consider the problem of Example 5.9. Apply an SQP algorithm with line search, starting from $\mathbf{x}_0 = (1,1)^T$. Solve the QP subproblem using (7.72) and BFGS approximation (7.73), (7.74) for the Hessian of the Lagrangian. Use the merit function (7.76) and the Armijo Line Search to find step sizes. Perform at least three iterations. Discuss the results.

$$\begin{array}{ll} \min & x_1^2 + (x_2 - 3)^2 \\ \text{subject to} & x_2^2 - 2x_1 \leq 0 \end{array}$$