Optimization

This is another vast and deep subject, and we will only take a brief peek. We have of course already seen some forms of optimization, in the form of error minimization. No doubt you are familiar as well with the optimization of linear systems. In this set of notes we will focus on optimizing arbitrary non-linear functions. And in the next section of notes we will consider optimization of functionals, that is, functions of functions.

Applications (Let's focus on robotics-related optimization.)

- Minimize energy/power consumption.
- Minimize torque, null space forces.
- Path planning via potential fields.
- Computation of stable resting configurations.
- Maximize sensing information (e.g., where to probe next?)
- Optimize execution time

  (Actually this leads to Markov Decision Theory, Dynamic Programming, and Calculus of Variations. We will discuss Cof V in detail later.)

We will now become increasingly more analytical and less numerical. You should also put on your geometric thinking caps, since much of the analysis is best understood in terms of geometry.

The basic problem: Given a function of n variables

$$f: \mathbb{R}^n \longrightarrow \mathbb{R}$$

we would like to find its minimum (or maximum), if it exists. In many cases it is difficult to find a global minimum, so we will often settle for local minima.

Recall how this process works in one dimension ($n=1$). We have a function $f: \mathbb{R} \to \mathbb{R}$. First we might compute the critical set of f,

$$C_f = \{ x \mid f'(x) = 0 \}$$

By examining this set we can determine those x that are global minima.

Notice that computation of C_f seems to entail finding the zeros of a function, namely the zeros of f. In other words, we have reduced the optimization problem to the root-finding problem. So, I guess we are done. Well, not quite. In higher dimensions, it is often easier to find (local) minima than one would think, based on our experience in finding simultaneous zeros in higher dimensions. Intuitively, this is because f' is not an arbitrary function, but rather a derivative.

We will thus have a lot more to say about optimization in higher dimensions than we did about root-finding.

To show the analogy with root-finding, let's stick with the one-dimensional case for a minute.

Golden Section Search (analogy to bisection)

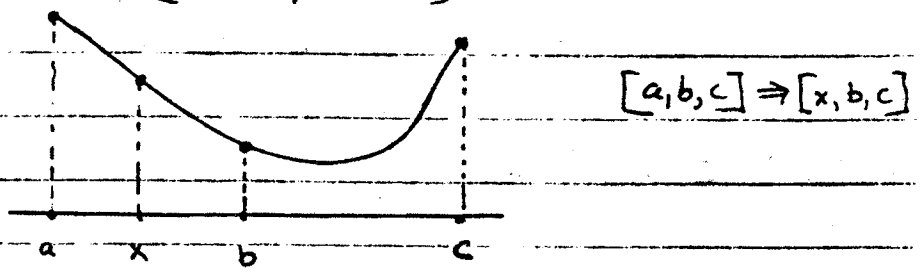
Basic idea: Want to bracket a minimum, then shrink the bracket.

The problem is that we don't know the value of the function at the minimum, so we can't know whether we have bracketed a minimum. Fortunately, if we are willing to settle for a local minimum, then we really just want to bracket a zero of f'. That we can do, by a numerical approximation to the derivative of f. The inner part of the loop works as follows.

We have three points a, b, c such that $a < b < c$ and

$$f(b) < \min\{f(a), f(c)\}$$

Eg.:



Now choose a point x, say halfway between a and b.
If $f(x) > f(b)$, then the new bracketing triple becomes $[x, b, c]$.
If $f(x) < f(b)$, then the new bracketing triple becomes $[a, x, b]$.

See NRiC p. 398 for wise words regarding limitations on how small you can make the bracketing interval. And see the rest of NRiC for other methods, e.g. Parabolic Interpolation.

Now, let's turn to the n-dimensional case

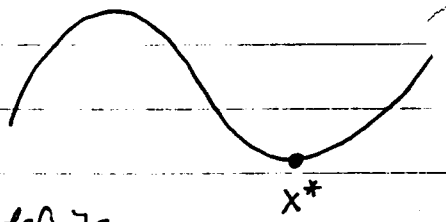
First, recall the following conditions for a relative (local) minimum.

Necessary Conditions for a Relative Minimum

Suppose $f: \mathbb{R}^n \to \mathbb{R}$ is C^2. (C^2 means f has continuous second partials.)
Let x^* be a relative minimum of f.
Then:
i) $\nabla f(x^*) = 0$
ii) for every vector $d \in \mathbb{R}^n$, $d^T \nabla^2 f(x^*) d \geq 0$

Notation: Here $\nabla^2 f(x^*)$ is the Hessian of f at x^* $\left(\frac{\partial^2 f}{\partial x_i \partial x_j}\right)$
Don't confuse this with the Laplacian.

Note:
- In 1D this just says the usual thing:
  i) $f'(x^*) = 0$
  ii) $f''(x^*) \geq 0$



- $\nabla^2 f(x^*)$ is symmetric positive semi-definite.
  If x^* is a strict relative minimum then $\nabla f(x^*)$ is positive-definite.

We can almost get a converse to these conditions:

Sufficient Conditions for a Relative Minimum

If $\nabla f(x^*) = 0$ and $\nabla^2 f(x^*)$ is positive definite
then x^* is a strict relative minimum.

Ex Suppose c is a real number
b is a vector $(n \times 1)$
A is a symmetric positive definite matrix $(n \times n)$

Consider the function $f: \mathbb{R}^n \to \mathbb{R}$ given by

$$f(x) = c + b^T x + \frac{1}{2} x^T A x$$

It is easy to see that

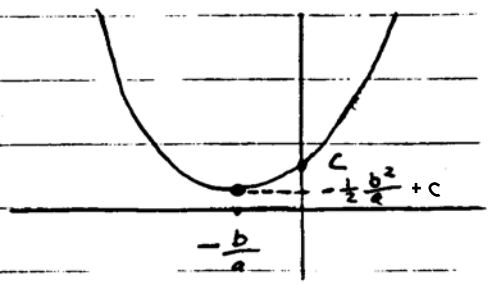
$$\nabla f(x) = b + Ax$$

$$\nabla^2 f(x) = A$$

So, there is a single extremum, located at $x = x^*$, where x^* is the solution to the system $\boxed{Ax = -b.}$
Since A is positive definite this extremum is a strict local minimum. Since it is the only one, it is in fact a global minimum.
Notice how once again the problem of finding a minimum is reduced to the problem of finding a zero.

This example is of course just the generalization of the following 1D example. The point is that locally every function[1] behaves like this (locally near a minimum).

$y = c + bx + \frac{1}{2}ax^2$
$y' = b + ax$
$y'' = a > 0$
$y' = 0$ iff $x = -\frac{b}{a}$



(1) every function with a Taylor series expansion

Global Minima

We just saw that $\nabla^2 f$ is positive definite at and near a strict local minimum. By Taylor's theorem every function looks like a quadratic near a strict local minimum. Furthermore, if f happens to be quadratic globally, formed from a symmetric positive definite matrix A as on the previous page, then it has a unique local minimum. This local minimum is therefore a global minimum.

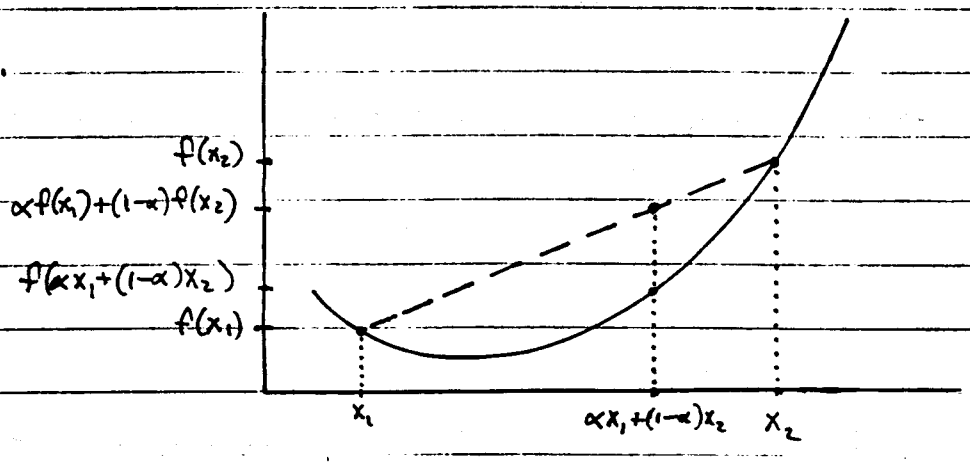
Can we say more about functions whose local minima are global minima?

A broad class of such functions are the **convex functions**. Let's take a look.

Def A function defined on a convex domain $\Omega \subseteq \mathbb{R}^n$, $f: \Omega \to \mathbb{R}$, is said to be **convex** if for every pair of points x_1, x_2 in Ω and for every α, $0 \le \alpha \le 1$,

$$f(\alpha x_1 + (1-\alpha) x_2) \le \alpha f(x_1) + (1-\alpha) f(x_2).$$

E.g.



Convex functions describe functions near local minima, as the following proposition shows (we omit the proof).

Prop Let $f \in C^2$. Then f is convex over a convex set Ω containing an interior point if and only if the Hessian matrix $\nabla^2 f$ is positive semi-definite throughout Ω.

And the minima of convex functions are global minima, as the following theorem shows.

Th^m Let f be a convex function defined on a convex set Ω. Then the set Γ where f achieves its minimum value is convex. Furthermore, any relative minimum is a global minimum.

Pf If f has no relative minima then the theorem is valid by default. Assume therefore that c_0 is the minimum value of f on Ω. Then $\Gamma = \{x \in \Omega \mid f(x) = c_0\}$

$= \{x \in \Omega \mid -f(x) \leq c_0\}$ ← this is easily seen to be convex, since f is convex.

Suppose now that $x^* \in \Omega$ is a relative minimum point of f. Suppose it is not a global minimum. Then there exists a $y \in \Omega$ such that $f(y) < f(x^*)$.

On the line $\{\alpha y + (1-\alpha)x^* \mid 0 \leq \alpha \leq 1\}$ we have

$$f(\alpha y + (1-\alpha)x^*) \leq \alpha f(y) + (1-\alpha) f(x^*)$$
$$< \alpha f(x^*) + (1-\alpha) f(x^*) = f(x^*)$$

which contradicts the fact that x^* is a relative minimum. //

Steepest Descent

Now let's return to the general problem of minimizing a function $f: \mathbb{R}^n \to \mathbb{R}$.

We saw previously that we want to find the critical values where $\nabla f = 0$. This is a system of n equations:

$$\begin{aligned} \frac{\partial f}{\partial x_1} &= 0 \\ &\vdots \\ \frac{\partial f}{\partial x_n} &= 0 \end{aligned}$$

We might expect to encounter the usual difficulties associated with higher-dimensional root-finding. Fortunately the derivative nature of the equations imposes some helpful structure to the problem.

In 1D, to find a local minimum, we might employ the following rule:

- If $f'(x) < 0$ then move to the right
- If $f'(x) > 0$ then move to the left
- If $f'(x) = 0$ then stop.

In higher dimensions we use the gradient ∇f to point us toward a minimum. This is called <u>steepest descent</u>. In particular, the algorithm repeatedly performs 1D minimizations along the direction of steepest descent.

The algorithm:

We start with $x^{(0)}$, as an approximation to a local minimum of $f: \mathbb{R}^n \to \mathbb{R}$.

For $m = 0, 1, 2, \ldots$ until satisfied do:

$\vec{u} = \nabla f(x^{(m)})$

If $\vec{u} = \underline{0}$ then stop

Else:
- minimize the function $g(t) = f(x^{(m)} - t\vec{u})$
- let $t^* > 0$ be the closest such minimum to zero.
- $x^{(m+1)} \leftarrow x^{(m)} - t^*\vec{u}$

How does one perform the line minimization of $g(t)$?

Anyway you want. For instance, step along the line until you produce a bracket, then refine the bracket.

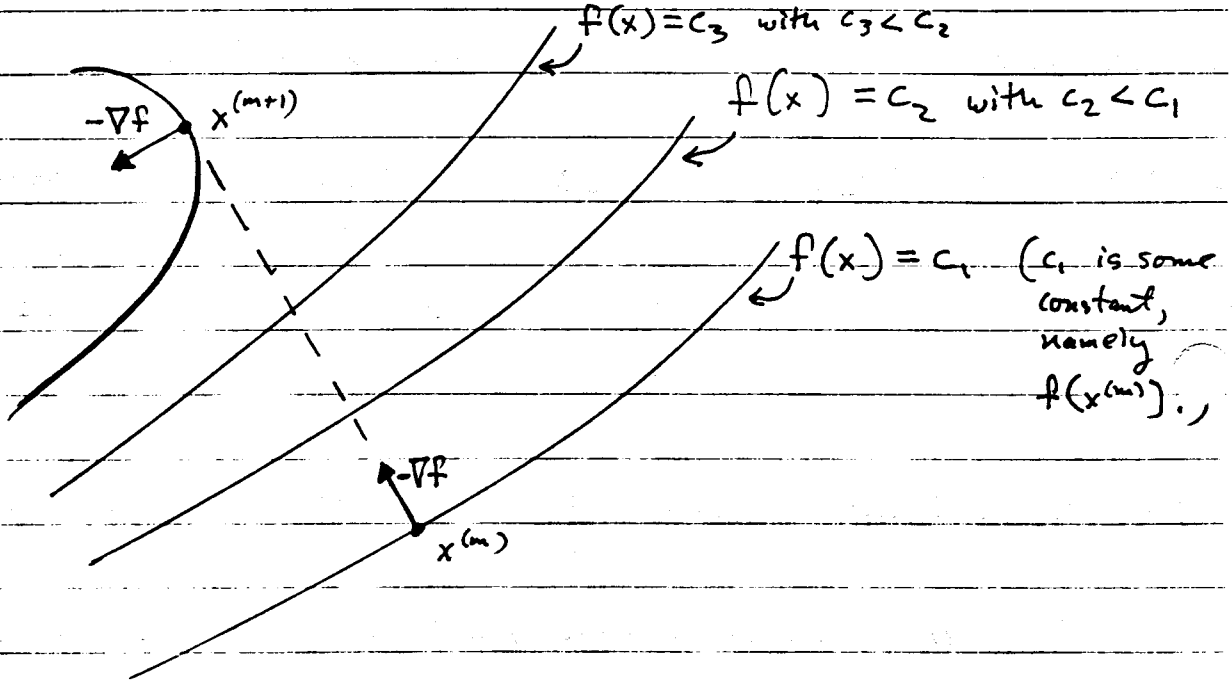
It turns out that steepest descent converges globally to a relative minimum (of course, a relative minimum might be a point at infinity). Convergence is linear. In fact

$$\frac{|e_{m+1}|}{|e_m|} \sim \left(\frac{A-a}{A+a}\right)^2$$

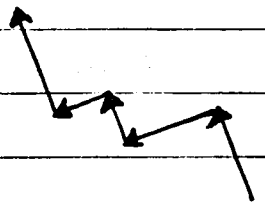
where A and a are the largest and smallest eigenvalues, respectively, of the Hessian $\nabla^2 f$ at the local minimum.

Problem with steepest descent

The method can take many steps. The problem is that the method repeatedly moves in a steepest direction to a minimum along that direction. Consequently consecutive steps are perpendicular to each other. Graphically, we see the following behavior.



So, we see a bunch of back and forth steps



that only slowly converge to a minimum. This problem is especially bad in a narrow valley. Successive steps undo some of their previous progress. — Ideally in $\mathbb{R}^n$ we would like to take n perpendicular steps, each of which attains a minimum in its direction without undoing any other step's previous progress.

This idea leads to the Conjugate Gradient Method

The basic idea is to move in non-interfering directions.

Suppose we have just performed a line minimization along direction u. Then ∇f is perpendicular to u, or else we could have moved further along u. Next we should move along some direction v. In Steepest Descent we let v be $-\nabla f$. In Conjugate Gradient we perturb $-\nabla f$, i.e., v is $-\nabla f$ plus something

We want to choose v in such a way that it doesn't "undo" our minimization along u. In other words, we want ∇f to be perpendicular to u before and after we move along v. In short, the local condition is that the change in ∇f be perpendicular to u.

Now observe that a small change in x, call it δx, will produce a small change in ∇f, call it $\delta(\nabla f)$. The connection between these changes is given by the Hessian:

$$\delta(\nabla f) = [\nabla^2 f]\,\delta x$$

So, our idea of moving along non-interfering directions leads to the condition

$$u^T \delta(\nabla f) = 0,$$

i.e., $$u^T [\nabla^2 f] v = 0.$$

So, v isn't orthogonal to u, but it is $[\nabla^2 f]$-orthogonal to u.

Of course, we must worry about a slight technicality. The connection between δx and $\delta(\nabla f)$ in terms of the Hessian $\nabla^2 f$ is a differential relationship. Can we really use it for finite motions? The answer is "sort of," namely to the extent that Taylor's approximation of order 2 is valid.

Suppose we expand a function $f: \mathbb{R}^n \to \mathbb{R}$ around a point $y \in \mathbb{R}^n$:

$$f(x+y) = f(y) + \sum_{i=1}^{n} \frac{\partial f}{\partial x_i}\bigg|_y x_i + \frac{1}{2} \sum_{i,j} \frac{\partial^2 f}{\partial x_i \partial x_j}\bigg|_y x_i x_j + \cdots$$

$$\approx f(y) + \nabla f(y)^T x + \frac{1}{2} x^T [\nabla^2 f] x$$

So, locally f looks like a quadratic. If we focus on quadratics, then the Hessian doesn't vary as we move along different directions u and v. Thus the condition $u^T [\nabla^2 f] v = 0$ makes sense.

With this reasoning as background, one develops Conjugate Gradient for quadratic functions formed from symmetric positive definite matrices. For such quadratic functions, the Conjugate Gradient method converges to the unique global minimum in at most n steps, by moving along successive non-interfering directions.

For general functions, Conjugate Gradient repeatedly executes "packages" of n steps. Once near a local minimum the algorithm converges quadratically. We will see more of the details shortly.

Newton's Method

Before looking at further details of Conjugate Gradient, let's mention Newton's method. Recall that in finding a minimum of f we may wish to consider the set of zeros of f'. So, in principle we could apply Newton-Raphson to f'.

This would yield the update rule

$$x^{(m+1)} = x^{(m)} - \left[\nabla^2 f(x^{(m)})\right]^{-1} \nabla f(x^{(m)})$$

Comments:

- Suppose f is quadratic: $f(x) = c + b^T x + \frac{1}{2} x^T A x$, with A symmetric positive definite. Then $\nabla f = b + Ax$, $\nabla^2 f = A$, and the global minimum of f satisfies $Ax = -b$. In this case, Newton's method converges in a single step.

- For general f, the Hessian $\nabla^2 f$ often is unknown. To remedy this, there exist methods called Quasi-Newton methods that build $\left[\nabla^2 f\right]^{-1}$ iteratively as they move.

Conjugate Gradient is an intermediate between steepest descent and Newton's method. It tries to achieve the quadratic convergence of Newton's method without incurring the cost of computing $\nabla^2 f$. At the same time, Conjugate Gradient will execute at least one gradient descent step per cacheee of n steps thereby ensuring convergence in the worst case.

Details

Let's look at Conjugate Gradient more carefully

Our previous musings concerning non-interfering directions lead to the following.

Def Suppose that Q is a symmetric matrix. Two vectors d_1 and d_2 are said to be Q-orthogonal or conjugate with respect to Q if

$$d_1^T Q d_2 = 0$$

In our case we are interested in local minima, so the following proposition applies:

Prop If Q is symmetric positive definite and the non-zero vectors $d_0, \dots, d_k$ are Q-orthogonal, then they are independent.

Pf Suppose $\alpha_0 d_0 + \dots + \alpha_k d_k = 0$.

We must show that $\alpha_i = 0$, $i = 0, \dots, k$.

Well, clearly

$$\underbrace{\alpha_0 d_i^T Q d_0 + \alpha d_i^T Q d_1 + \dots + \alpha_k d_i^T Q d_k}_{= \alpha_i d_i^T Q d_i \text{ by def of } Q\text{-orthogonal}} = 0$$

So $\alpha_i d_i^T Q d_i = 0$ for each i. But $d_i^T Q d_i > 0$ since Q is positive definite. So $\alpha_i = 0$ for each i. ∎

Deriving the algorithm (for quadratic f)

Suppose we wish to solve the following problem (in $\mathbb{R}^n$)

$$\text{minimize} \quad \tfrac{1}{2} x^T Q x + b^T x \qquad (\text{with } Q \text{ sym pos def})$$

Suppose further that we "happen" to have n non-zero Q-orthogonal vectors $d_0, \ldots, d_{n-1}$. By the proposition on page 14, these vectors are independent. Therefore $\{d_0, \ldots, d_{n-1}\}$ is a Q-orthogonal basis for $\mathbb{R}^n$.

Now let x^* be the unique vector that minimizes $\frac{1}{2} x^T Q x + b^T x$.
We can write $x^* = \alpha_0 d_0 + \cdots + \alpha_{n-1} d_{n-1}$ for some set of real numbers $\alpha_0, \ldots, \alpha_{n-1}$.
We know that x^* satisfies $Q x^* = -b$.
So

$$Q(\alpha_0 d_0 + \cdots + \alpha_{n-1} d_{n-1}) = -b,$$

and $$d_i^T Q(\alpha_0 d_0 + \cdots + \alpha_{n-1} d_{n-1}) = -d_i^T b.$$

Therefore $$\alpha_i d_i^T Q d_i = -d_i^T b.$$

In other words, $$\alpha_i = \frac{-d_i^T b}{d_i^T Q d_i}$$

We thus have the explicit formula

$$x^* = -\sum_{i=0}^{n-1} \frac{d_i^T b}{d_i^T Q d_i} d_i.$$

Notice two important tricks:

(1) By choosing the $\{d_i\}$ to be Q-orthogonal we've set things up so we can determine the coefficients $\{\alpha_i\}$ easily, using inner products.

(2) Of course (1) is possible for any positive-definite matrix. In particular, we could simply have chosen the $\{d_i\}$ to be orthogonal (i.e., I-orthogonal). Then $x^* = \sum_{i=0}^{n-1} \frac{d_i^T x^*}{d_i^T d_i} d_i$. However, by choosing the $\{d_i\}$ to be Q-orthogonal we can determine the coefficients $\{\alpha_i\}$ in terms of the known quantity b, not the unknown quantity x^*.

How does this generate an algorithm?

- One view is purely algebraic, namely, we compute $\alpha_0, \alpha_1, \ldots, \alpha_{n-1}$.
- Another view is to think of these computations as an n-step search. We start the search at the origin. On the i^{th} iteration we move in direction d_i by α_i. After n iterations we have found the unique minimum x^*.

- We can generalize this second view to devise a numerical minimizer for arbitrary functions f. For such f, the Hessian of f plays the role of Q. The algorithm executes packages of n search step. Each step builds a coordinate $\alpha_i d_i$ in a search for the minimum x^*. After n steps, the algorithm resets, using its current x location as a new "origin" from which to start another n-step search. — The resulting search as pseudo quadratic convergence:

$$|e_{k+n}| \leq c|e_k|^2$$ — (For large n, this is not so great.

Two important issues remain:

1) How do we construct the Q-orthogonal vectors $\{d_0, \dots, d_{n-1}\}$?
2) How do we deal with the reality that the matrix Q (i.e., the Hessian of f) is often unknown?

The following algorithm addresses the first issue. We will deal with the other issue shortly.

The following algorithm generates the d_i iteratively, as descent vectors. The d_i are modified versions of the gradient ∇f, modified in a way to ensure Q-orthogonality.

Notation:
- We formulate the algorithm for quadratic cost $\frac{1}{2}x^TQx + b^Tx$.
- g_k means $Qx_k + b$. In other words, g_k is the gradient of $f(x) = \frac{1}{2}x^TQx + b^Tx$, evaluated at $x = x_k$.

1. Let $d_0 = -g_0 = -b_0 - Qx_0$ ($x_0 \in \mathbb{R}^n$ is arbitrary)

2. For $k = 0, \dots, n-1$ do:

2a. $\alpha_k \leftarrow -\dfrac{g_k^T d_k}{d_k^T Q d_k}$

2b. $x_{k+1} \leftarrow x_k + \alpha_k d_k$

2c. $\beta_k \leftarrow \dfrac{g_{k+1}^T Q d_k}{d_k^T Q d_k}$

2d. $d_{k+1} \leftarrow -g_{k+1} + \beta_k d_k$

3. Return x_n.

Observe: • Step 1 is a pure steepest descent step. The presence of this step will ensure convergence when we pass from quadratic cost functions to arbitrary cost functions.

• Step 2d is the generalization of steepest descent that will give us Q-orthogonality of the descent vectors $d_0, \dots, d_{n-1}$.

Correctness

To see that the algorithm is correct, we need to show that:

1) The $\{d_i\}$ produced by the algorithm are Q-orthogonal.
2) $x^* - x_0 = \alpha_0 d_0 + \cdots + \alpha_{n-1} d_{n-1}$
   where x_0 is our starting point and x^* is the minimum of $\frac{1}{2}x^TQx + b^Tx$.

Proof of (2):

In deriving the coefficients $\{\alpha_i\}$ on page 15, we assumed effectively that the search starts at the origin. For a search that starts at x_0, i.e., for a search in which we are trying to compute $x^* - x_0$, the formula for α_i on page 15 becomes

$$\alpha_i = -\frac{d_i^T(b + Qx_0)}{d_i^T Q d_i}$$

The algorithm, on the other hand, computes

$$\alpha_i = = \frac{g_i^T d_i}{d_i^T Q d_i}$$

It is easy to see that these expressions are the same.

To wit, $g_i^T d_i = d_i^T g_i$

$$= d_i^T (Q x_i + b)$$

$$= d_i^T \left(Q (x_0 + \alpha_0 d_0 + \cdots + \alpha_{i-1} d_{i-1}) + b \right)$$

$$= d_i^T (Q x_0 + b) + \alpha_0 d_i^T Q d_0 + \cdots + \alpha_{i-1} d_i^T Q d_{i-1}$$

$$= d_i^T (b + Q x_0) \quad //$$

This establishes (2).

Proving property (1) is considerably more difficult. We will not really do so, but instead state two very important theorems. These theorems will give us:

- A little more intuition about the nature of Conjugate Gradient.
- A proof of (1).
- A method for computing the $\{d_i\}$ & $\{\alpha_i\}$ without requiring explicit knowledge of Q.

And then we'll be done with Conjugate Gradient.

Expanding Subspace Theorem

Let $d_0, \ldots, d_{n-1}$ be a sequence of non-zero Q-orthogonal vectors in $\mathbb{R}^n$. For every $x_0 \in \mathbb{R}^n$, the sequence $\{x_k\}$ generated by the rule

$$x_{k+1} = x_k + \alpha_k d_k$$

where $$\alpha_k = -\frac{g_k^T d_k}{d_k^T Q d_k}$$

& $$g_k = Q x_k + b$$

has the property that x_{k+1} minimizes the function $f(x) = \frac{1}{2} x^T Q x + b^T x$ on the line $x = x_k + \alpha d_k$, $-\infty < \alpha$, as well as on the affine variety

$$x_0 + \mathcal{B}_{k+1}$$

where $$\mathcal{B}_{k+1} = \text{span}\{d_0, \ldots, d_k\}$$

This tells us that the Conjugate Gradient algorithm really is a generalization of Steepest Descent. Each step of adding $\alpha_k d_k$ to the previous estimate is the same as doing a line minimization along the direction d_k. Furthermore, the offset $\alpha_k d_k$ does not undo previous progress, that is, the minimization is in fact a minimization over $x_0 + \mathcal{B}_{k+1}$.

Corollary $\quad g_k \perp \mathcal{B}_k$

Conjugate Gradient Theorem

In the algorithm of page 17, we have that:

a) $\text{span}\{g_0, \ldots, g_k\} = \text{span}\{g_0, Qg_0, \ldots, Q^k g_0\}$

b) $\text{span}\{d_0, \ldots, d_k\} = \text{span}\{g_0, Qg_0, \ldots, Q^k g_0\}$

c) $d_k^T Q d_i = 0$ for all i less than k.

d) $\alpha_k = \dfrac{g_k^T g_k}{d_k^T Q d_k}$

e) $\beta_k = \dfrac{g_{k+1}^T g_{k+1}}{g_k^T g_k}$

Comments:

- Part (c) is of course point (1) from page 18, that is, the $\{d_i\}$ are Q-orthogonal. — Notice by the way that $d_{k+1}^T Q d_k$ is easily seen to be zero, since step (2d) on page 17 purposefully constructs d_{k+1} to satisfy this Q-orthogonality.
- Part (e) is very important, because it provides us a way to compute β_k without knowing Q.
- How do we compute α_k without knowing Q? The Expanding Subspace Theorem already gave us the answer; we perform a line minimization.

We can now state the Conjugate Gradient algorithm for general C^2 functions $f: \mathbb{R}^n \to \mathbb{R}$.

Conjugate Gradient Algorithm

STEP 1. Start at some x_0.
Set $d_0 = -\nabla f(x_0)$

STEP 2. For $k = 0, 1, \ldots, n-1$ Do:

a) Set $x_{k+1} = x_k + \alpha_k d_k$,
where α_k minimizes
$g(\alpha) = f(x_k + \alpha d_k)$

b) Set $d_{k+1} = -\nabla f(x_{k+1}) + \beta_k d_k$
where $\beta_k = \dfrac{|\nabla f(x_{k+1})|^2}{|\nabla f(x_k)|^2}$

STEP 3. Replace x_0 by x_n and Repeat from Step 1, until satisfied with the results.

Note: Step 1. ensures that there is at least one descent direction every n iterations.
Step 2a. ensures that no step increases f.

Omitted Topics

- Quasi-Newton Methods
- Partial Conjugate Gradient Methods
  (Run the inner loop for m steps, with $m \leq n$, then restart STEP 1 and repeat.)
- Variation of Brent's method for line minimization.

Tammy Abell

Constrained Optimization

Problem: minimize (maximize) $f(x)$

subject to $h_1(x) = 0$

$\vdots$

$h_m(x) = 0$

where $x \in \Omega \subset R^n$; $f: R^n \to R$; $h_j: R^n \to R$ for $j = 1, \ldots, m$ and $m \leq n$.

Usually we also assume that f, h_j in C^2

If f and the h_j are linear, we can use linear programming, which also handles linear inequality constraints of the form

$$g_j(x) \leq 0 \quad \text{for} \quad j = 1, \ldots, p$$

However, we would like to solve the problem for arbitrary nonlinear functions f, h_j.

The method we will use to do this is called

Lagrange Multipliers.

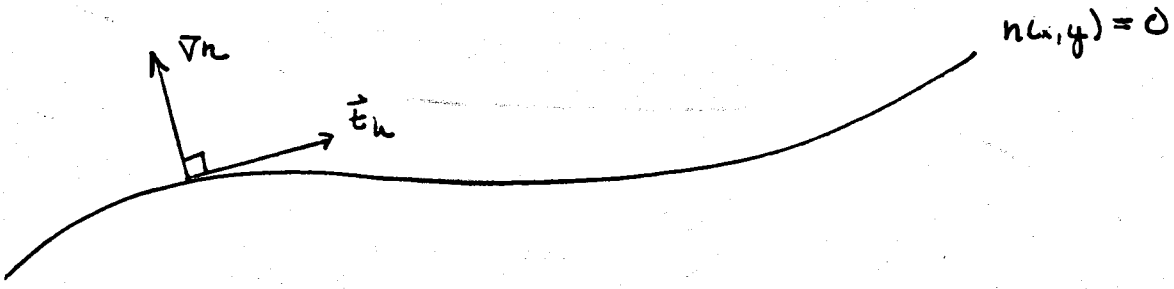
Note: In the following derivations, we will generally disregard the set constraint $x \in \Omega$, assuming that either $\Omega = R^n$ or the solution to the optimization lies in the interior of Ω.

In order to gain some intuition, let's look at the case where $n = 2$ and $m = 1$. The problem becomes

minimize $f(x,y)$

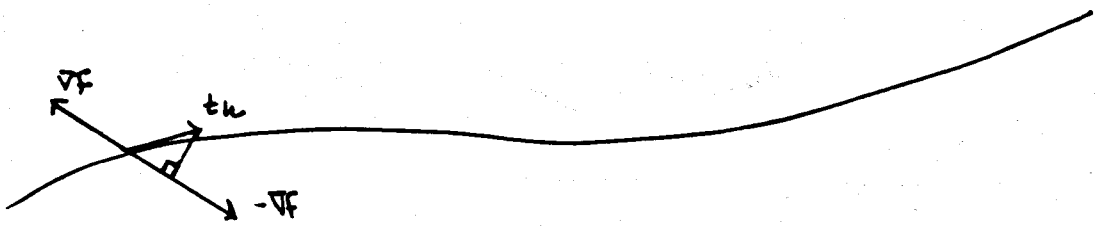
subject to $h(x,y) = 0$ $\qquad x, y \in R$

Suppose we are at a <u>feasible point</u>, that is a point on the constraint curve $h(x,y) = 0$. To stay on this curve, motion must be along the tangent $\vec{t}_h(x,y)$. Recall that the tangent of a curve is normal to the gradient, thus $\nabla h \cdot \vec{t}_h = 0$



In order to increase or decrease $F(x,y)$, motion along the constraint curve must produce a component of motion along the gradient of f, i.e.

$$\nabla f \cdot \vec{t}_h \neq 0$$



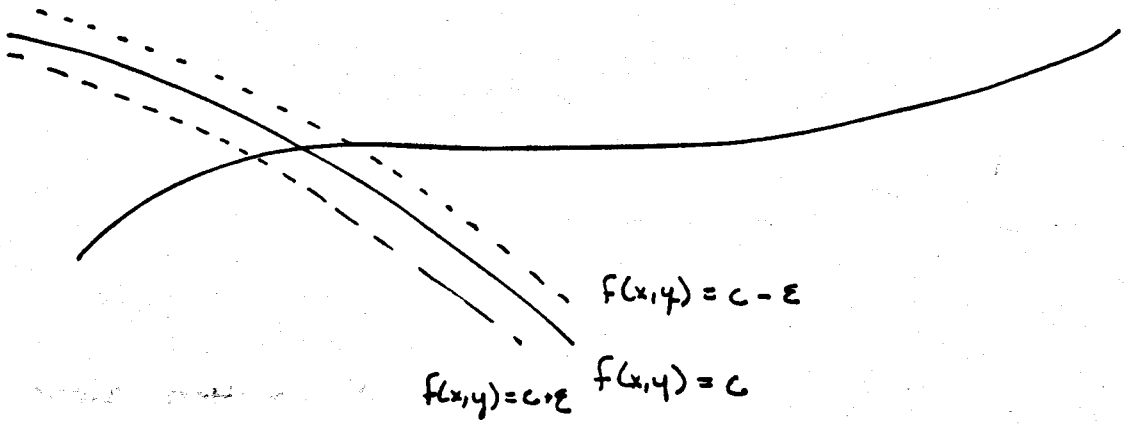
At an extremal value of f, a differential motion should not produce a component of motion along ∇f. Thus $\vec{t}_h$ is orthogonal to ∇f and

$$\nabla f \cdot \vec{t}_h = 0$$ ..... at an extremal value of f.

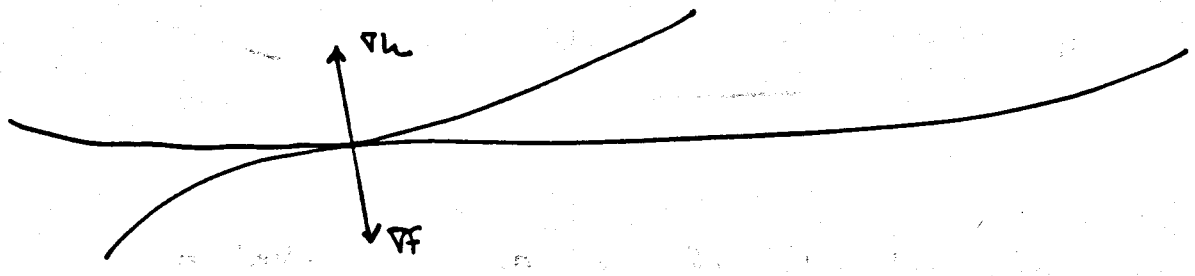
Since $\vec{t}_w$ is orthogonal to both ∇f and ∇h at an extrema, ∇f and ∇h must be parallel. Thus there exists a $\lambda \in \mathbb{R}$ s.t.

$$\nabla f + \lambda \nabla h = 0$$

Another way to see this is to note that the constraint curve must be tangent to a constant contour of f when f is extremal. Otherwise infinitesimal motions along the constraint curve result in an increase or decrease in f as the point moves to a new constant contour.



At their mutual point of tangency, the normals to the curves $h(x,y) = 0$ and $f(x,y) = c^*$ must be parallel. But these normals are simply the gradients ∇h and ∇f.



Thus we again conclude that $\nabla F + \lambda \nabla h = 0$ at the extremal values of f.

Now, suppose we find the set $C_{f,h}$ of points (x,y) satisfying

$$h(x,y) = 0$$
$$\nabla f + \lambda \nabla h = 0 \qquad \text{for some value } \lambda$$

$C_{f,h}$ then contains the extremal points of f subject to the constraint $h(x,y) = 0$.

But these equations describe a nonlinear system in the parameters x, y, λ. It can be solved using numerical techniques, for example Newton-Raphson.

Now consider the Lagrangian associated with this constraint problem.

$$F(x,y,\lambda) = f(x,y) + \lambda h(x,y) \qquad F: \mathbb{R}^3 \to \mathbb{R}$$

Note that

$$\nabla F = \begin{bmatrix} \frac{df}{dx} + \lambda \frac{dh}{dx} \\ \frac{df}{dy} + \lambda \frac{dh}{dy} \\ h \end{bmatrix} = \begin{bmatrix} \nabla f + \lambda \nabla h \\ h \end{bmatrix}.$$

Thus setting $\nabla F = 0$ yields the same system of nonlinear equations derived above.

The value λ is known as the Lagrange multiplier. The approach of constructing the Lagrangian and setting its gradient to 0 is know as the Method of Lagrange Multipliers.

Now for an example:

Ex Find the extremal values of the function $f(x,y) = xy$ subject to the constraint

$$h(x,y) = \frac{x^2}{8} + \frac{y^2}{2} - 1 = 0$$

Solution

Constructing the Lagrangian:

$$F(x,y,\lambda) = xy + \lambda\left(\frac{x^2}{8} + \frac{y^2}{2} - 1\right)$$

$$\nabla F(x,y,\lambda) = \begin{bmatrix} y + \frac{\lambda x}{4} \\ x + \lambda y \\ \frac{x^2}{8} + \frac{y^2}{2} - 1 \end{bmatrix} = \vec{0}$$

This may be written as

$$y = \frac{-\lambda x}{4} \qquad x = -\lambda y \qquad x^2 + 4y^2 = 8$$

Combining the left pair of equations

$$y = \frac{-\lambda(-\lambda y)}{4} \qquad \lambda^2 = 4 \qquad \lambda = \pm 2$$

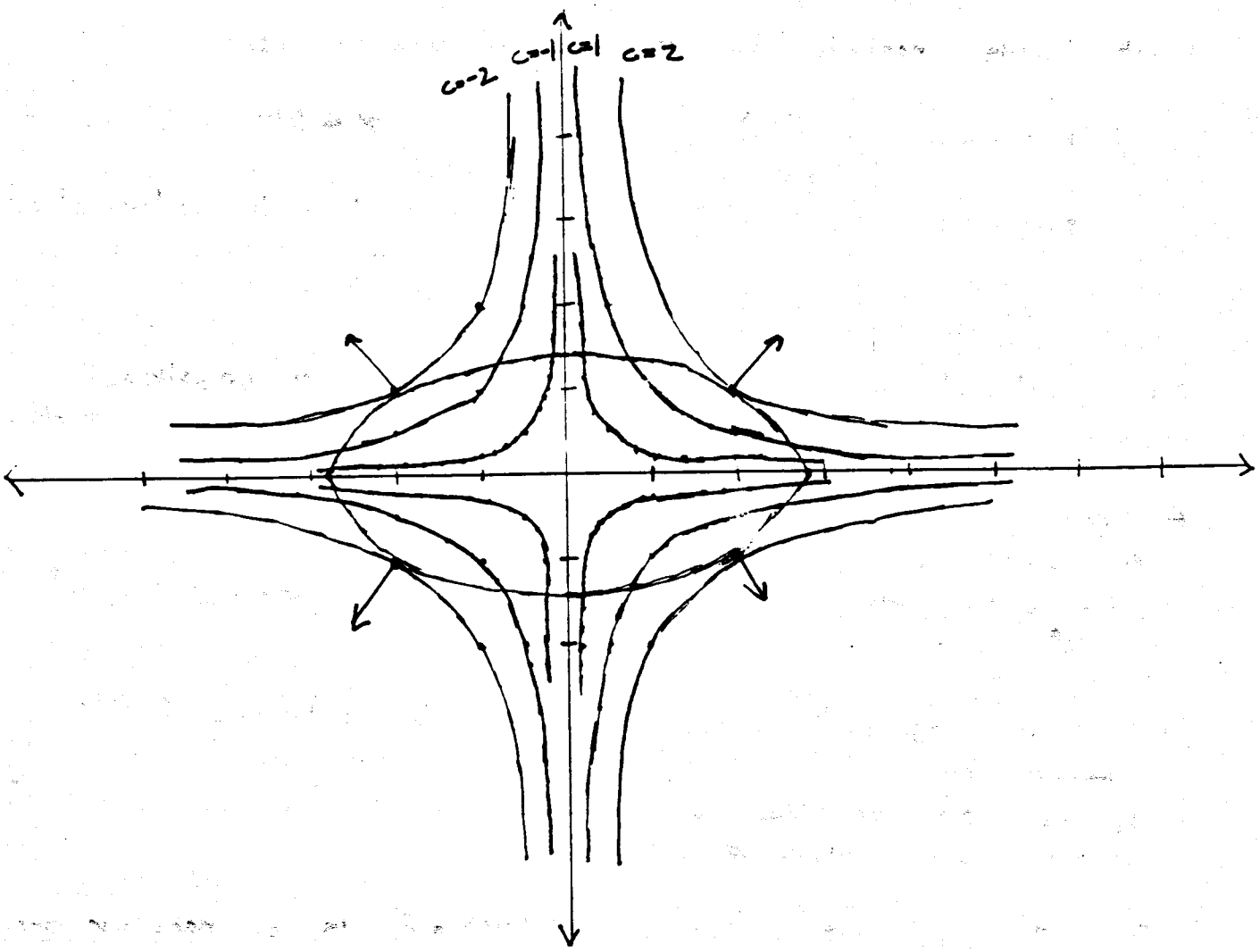
Thus $x = \pm 2y$. Substituting into rightmost equation

$$4y^2 + 4y^2 = 8$$

$$y = \pm 1 \qquad x = \pm 2$$

So there are four extremal points of f subject to h: $(2,1)$, $(-2,1)$, $(2,-1)$, $(-2,-1)$. f subject to h is maximal at the points $(2,1)$, $(-2,-1)$ and minimal at $(-2,1)$, $(2,-1)$.

Graphically, the constraint h defines an ellipse. The constant contours of f are the hyperbolas $xy = c$, with $|c|$ increasing as the curves move out from the origin.



Now let's derive the general Lagrange Multipliers formulation. This may seem painstaking, but it is necessary to understand when and how the method can fail.

First, let's restate the problem in vector form:

minimize $f(x)$ $\quad x \in R^n$, $f: R^n \to R$

subject to $h(x) = \vec{0}$ $\quad$ where h is the vector-valued function $h = (h_1, \ldots, h_m)$, $h: R^n \to R^m$

The constraint equations $h_j(x) = 0$ define a <u>constraint hypersurface</u> S in R^n. If $h_j(x) \in C^1$, this surface is smooth.

A <u>curve on S</u> is a family of pts $x(t) \in S$, $a \le t \le b$. The curve is <u>differentiable</u> if $\dot{x} \triangleq d/dt\, x(t)$ exists, and <u>twice differentiable</u> if $\ddot{x}$ exists. <u>$x(t)$ passes thru x^*</u> if $x^* = x(t^*)$ for some t^*, $a \le t^* \le b$.

<u>Def</u> The <u>tangent space</u> at x^* is the subspace of R^n spanned by $\{\dot{x}(t^*) : x(t^*) = x^*, x(t)$ a curve on $S\}$. That is, the span of the derivatives of all surface curves thru x^*. Denote this subspace as T.

<u>Def</u> A point x^* satisfying $h(x^*) = \vec{0}$ is a <u>regular point</u> of the constraint if the gradient vectors $\nabla h_1(x^*), \ldots \nabla h_m(x^*)$ are linearly independent.

From our previous intuitions, we expect that $\nabla f \cdot v = 0$ for all $v \in T$ at an extremum. This implies that ∇f lies in the orthogonal compliment of T, call it $T^\perp$. We would like to claim that ∇f can be composed from a linear combination of the ∇h_j. This is only valid provided that the ∇h_j span $T^\perp$. The ∇h_j span $T^\perp$ provided the extremal pt is regular.

Thm At a regular point x^* of the surface S defined by $h(x) = 0$, the tangent space is equal to

$$M = \{ y : \nabla h(x^*) y = 0 \}$$

proof:

$$\frac{d}{dt} h(x) \Big|_{t^*} = \nabla h(x) \dot{x}(t) \Big|_{t^*} = \nabla h(x^*) \underbrace{\dot{x}(t^*)}_{\text{tangent vector}} = 0$$

Since this is true for all curves x and tangent vectors $\dot{x}$ through x^*, we see that

$$T \subset M.$$

To show $M \subset T$, need to be able to construct a curve through x^* whose tangent is y for all $y \in M$.

Consider the curve $x(t) = x^* + ty + \nabla h(x^*)^T u(t)$ where $u(t) \in R^m$.

First we want to show that $x(t)$ is a curve on S, or in other words

$$\langle 1 \rangle \qquad h(x(t)) = 0$$

to pass through x^*, we see that $u(0) = 0$.
Now consider the Jacobian of $\langle 1 \rangle$ w.r.t. u at $t=0$.

$$\frac{d}{du} h(x(t)) \Big|_{t=0} = \nabla h(x(t)) \frac{d}{du} x(t) \Big|_{t=0}$$

$$= \nabla h(x(t)) \nabla h(x^*)^T \Big|_{t=0} = \underbrace{\nabla h(x^*)}_{m \times n} \underbrace{\nabla h(x^*)^T}_{n \times m}$$

This matrix is nonsingular since $\nabla h(x^*)$ is full rank (m) when x^* regular. Thus by the Implicit Function Thm there is a continuous solution $u(t)$ on some region $-a \le t \le a$ s.t. $\langle 1 \rangle$ holds. Thus $x(t)$ is a curve on S.

Differentiating <1> w.r.t. t at $t=0$, we get

$$\frac{d}{dt} h(x(t)) \Big|_{t=0} = \nabla h(x(t))\left(y + \nabla h(x^*)^T \dot{w}(\,)\right)\Big|_{t=0} = 0$$

$$= \underbrace{\nabla h(x^*)\, y}_{0} + \underbrace{\nabla h(x^*)\nabla h(x^*)^T \dot{w}(0)}_{\text{nonsingular}} = 0$$

thus we conclude that $\dot{w}(0) = 0$

Therefore $\dot{x}(0) = y + \nabla h(x^*)^T \dot{w}(0) = y$. We have constructed a curve that has derivative y at x^*. //

So, what is this theorem saying?

The matrix $\nabla h(x^*)$ has as its rows the gradient vectors $\nabla h_j(x^*)$. The theorem says that the tangent space at x^*, T, is equal to the nullspace of $\nabla h(x^*)$. Thus $T^\perp$, the orthogonal complement of T, must equal the rowspace of $\nabla h(x^*)$. This means that the vectors $\nabla h_j(x^*)$ span $T^\perp$ at the minimum x^*.

Note that the condition of being a regular point is not a condition of the surface S itself but of the parameterization h. In particular, <u>the tangent space is defined independent of h, while M is not.</u>

<u>ex</u> Let $h(x_1, x_2) = x_1$ Then $\nabla h = 1$ at all points, so $x^* \in R^2$ is regular.

If instead $h(x_1, x_2) = x_1^2$ then $\nabla h = 2x_1 = 0$ on the surface defined by $h(x_1, x_2) = 0$. Thus no point is regular. The dimension of T is still 1, but dimension of M is 2.

Lemma: Let x^* be a local extremum of f subject to the constraints $h(x) = \vec{0}$. Then for all y in the tangent space of the constraint surface at x^*,

$$\nabla f(x^*) y = 0$$

proof: Let $x(t)$ be any curve on constraint surface S with tangent y at x^*. Thus $x(0) = x^*$ and $\dot{x}(0) = y$. Since x^* is a constrained local minimum we have

$$\frac{d}{dt} f(x(t)) \Big|_{t=0} = 0 \quad \text{fc} \quad \| \; y, x(t)$$

$$\nabla f(x(t)) \dot{x}(t) \Big|_{t=0} = \nabla f(x^*) y = 0 \quad //$$

The next theorem states the Lagrange multiplier method as a necessary condition on an extremum

Thm Let x^* be a local extremum of f subject to the constraints $h(x) = 0$. Assume further that x^* is a regular point of these constraints. Then $\exists \lambda \in \mathbb{R}^m$ s.t.

$$\nabla f(x^*) + \lambda^T \nabla h(x^*) = 0$$

handwaving proof:

We know that the $\nabla h_i(x^*)$ span $T^\perp$ since x^* regular. The lemma says effectively that $\nabla f(x^*)$ lies in $T^\perp$. Thus we can write $\nabla f(x^*)$ as a linear combination of the $\nabla h_i(x^*)$. Therefore

$$\nabla f(x^*) = -\sum_{i=1}^{m} \lambda_i \nabla h_i(x^*) = -\lambda^T \nabla h(x^*)$$

or $$\nabla f(x^*) + \lambda^T \nabla h(x^*) = 0$$

The next two theorems describe the second order necessary and sufficient conditions for a critical point derived using the previous theorem to be a minimum.

Second order necessary conditions

Suppose x^* is a local minimum of f subject to $h(x) = 0$ and that x^* is a regular point of these constraints. Then $\exists\, \lambda \in R^m$ s.t.

$$\nabla f(x^*) + \lambda^T \nabla h(x^*) = 0$$

If we denote by M the subspace $M = \{y : \nabla h(x^*) y = 0\}$, then the matrix

$$L(x^*) = \nabla^2 f(x^*) + \lambda^T \nabla^2 h(x^*)$$

is positive semi-definate on M, that is $y^T L y \geq 0$ for all $y \in M$.

proof

similar to prev. theorem.

Second order sufficiency conditions

Suppose $\exists\, x^*$ satisfying $h(x^*) = 0$ and $\exists\, \lambda \in R^m$ s.t.

$$\nabla f(x^*) + \lambda^T \nabla h(x^*) = 0$$

Suppose also that matrix $L(x^*) = \nabla^2 f(x^*) + \lambda^T \nabla^2 h(x^*)$ is positive definate on $M = \{y : \nabla h(x^*) y = 0\}$.

Then x^* is a strict local minimum of f, subject to $h(x) = 0$.

proof

hairy.

So, what happens when an extremal point of f, x^*, is not regular?

The result stated in the theorem on pg 8 fails. In particular, $T \neq m$ where m is the Nullspace of $\nabla h(x^*)$.

$$m = \{y : \nabla h(x^*) y = 0\}.$$

It is still true that $T \subset m$, but $m \not\subset T$, thus the dimension of m may be larger than that of T.

Stated another way, since $T \subset m$, we can conclude that $m^\perp \subset T^\perp$, that is the gradient vectors $\nabla h_j(x^*)$ lie in the space $T^\perp$. However they may not span it.

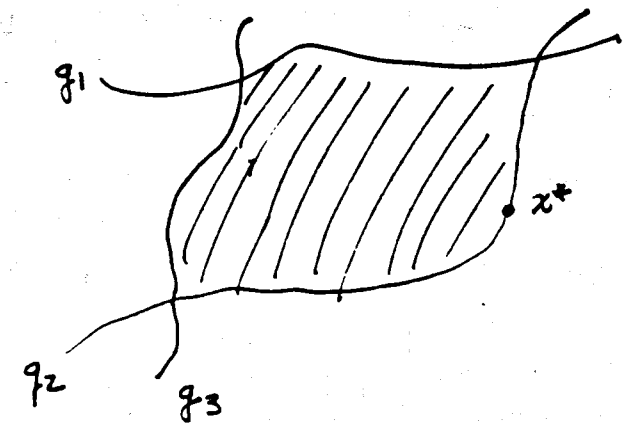
The result from the lemma tells us that $\nabla f(x^*)$ also lies in $T^\perp$. However, if $\nabla h_j(x^*)$ do not span $T^\perp$, $\nabla f(x^*)$ may have a component outside the span of $\nabla h_j(x^*)$. In particular, it may not be possible to find a $\lambda \in \mathbb{R}^m$ s.t.

$$\nabla f(x^*) + \lambda^T \nabla h(x^*) = 0.$$

Omitted topics

- Inequality constraints $g_j(x) \leq 0$ for $j = 1, \ldots, p$.

  Just for intuition, think about searching for a constrained optimum in a space bounded by inequality constraints.



  At x^*, it is clear that we don't need to worry much about g_1 and g_3, but we do need to be concerned about g_2. This gives rise to the notion of <u>active constraints</u>, those where $g_j(x) = 0$, and <u>inactive</u> <u>constraints</u> where $g_j(x) < 0$.

  The intuition is that we can use an approach like Lagrange multipliers applied to the active constraints with some additional work to figure out which constraints are active.

- Convergence rates