Minimizing with Nonlinear Constraints

- Start to Finish

Phase 1 Steps

Choose an initial point on any constraint boundary.

1. Find the steepest descent vector for f(x) at that point.
2. Test to see if the steepest descent vector is feasible. If not, skip to phase 2. If it is, move to step 3.
3. Write coordinates for a “new point”, find the maximum a value, minimize f(a), and calculate the new point.
4. If the new point is interior, find the steepest descent vector and repeat step 3. If it is on a constraint, move to phase 2.

Phase 2 Steps

Given a point on a constraint boundary whose steepest-descent vector for f(x) is not feasible,

1. Push off from the boundary using the steepest-descent vector for the active constraint g(x).*
2. Find the steepest descent vector for f(x) from the interior point.
3. From the interior point, find the maximum a value*, minimize f(a), then calculate the new point.

* (with modifications for corner regions)

Iterations/End Conditions

Repeat phase 2 (push off, steepest descent, minimize f(a)) until:

- two successive edge points are very close together, and/or
- the determinant of the steepest descent + constraint gradient matrix is 0 or very close to 0.

Problem 1

Minimize $f(x) = (x_1 - 3)^2 + x_2^2$

subject to $g_1(x) = x_1^2 - 4x_1 - x_2 \leq 0$

$g_2(x) = x_2^2 + x_1 - 10 \leq 0$

$g_3(x) = 2^{x_1 - 3} - x_2 \leq 0$

from starting point (1, 3) on g_2.

You will find it helpful to graph the feasible region before beginning.

Automate anything you like. You will need to perform a lot of iterations, so it's best to write programs and use loops.

Problem 2

Minimize $f(x) = x_1^2 + x_2$

subject to $g_1(x) = x_1^2 - 4x_1 - x_2 \leq 0$

$g_2(x) = x_2^2 + x_1 - 10 \leq 0$

$g_3(x) = 2^{x_1 - 3} - x_2 \leq 0$

from starting point (1, 3) on g_2.

Note that this has the same constraints as problem 1, but a different objective function.

Continue to use a graph.

A note: beyond 3D

You may have noticed, particularly in problem 2, a heavy reliance on the graph.

The main reason for this, however, was for humans to avoid extra calculations (like finding which constraint is the closest). But these sort of calculations can be done easily by a computer.

Nothing else in the process requires a visual or graph. These procedures scale easily into non-graphable dimensions.