Unit 13

~~Week 14~~

Iterative Solution Algorithms for Nonlinear Optimization

~~14~~3.1 Iterative Solution Algorithm for Constrained Nonlinear Optimization Problems

Let us remember that a **constrained NLPP** can be generically written as:

$$\begin{array}{ll} \min & f(x) \\ \text{s.t.} & x \in C, \end{array}$$

where $C := \{x \in \mathbb{R}^n \ ; \ h_1(x) = 0, \ldots, h_p(x) = 0, g_1(x) \leq 0, \ldots, g_q(x) \leq 0\}$ is the feasible set, $h_1, \ldots, h_p : \mathbb{R}^n \to \mathbb{R}$ are the equality-constraint functions, $g_1, \ldots, g_q : \mathbb{R}^n \to \mathbb{R}$ are the less-than-or-equal-to-constraint functions, $p, q \in \{0, 1, 2, \ldots\}$ and $f : \mathbb{R}^n \to \mathbb{R}$ is the objective function.

14.1.1 Penalty-Based Method

Example 14.1. Consider $f, h, g : \mathbb{R}^2 \to \mathbb{R}$ defined by $f(x_1, x_2) = (x_1 + 2)^2 + (x_2 - 3)^2$, $h(x_1, x_2) = x_1 + 2 \cdot x_2 - 4$ and $g(x_1, x_2) = -x_1$, respectively. Solve the problem by the **Penalty-based method** using:

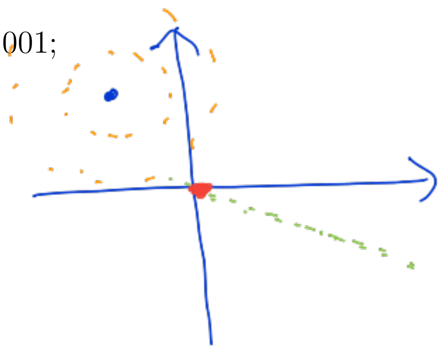
1. a tolerance of 0.001;
2. $x^0 = (-1, -1)$;

$\min \ (x_1+2)^2 + (x_2-3)^2$

s.a. $\boxed{x_1 + 2 \cdot x_2 - 4 = 0}$

$-x_1 \leq 0$

$y = 2 - \frac{1}{2}x$



3. initial ρ value of 1;
4. initial ϕ value of 1;
5. the steepest descent search direction with the Armijo rule with $\sigma = 0.1$, $\beta = 0.1$ and $s = 1$;

And if a constraint is violated, multiply the penalty weight on that constraint by 3.

Solution. Indeed, denoting the penalized objective function

$$F_{\rho,\phi}(x) := f(x) + \rho \cdot h(x)^2 + \phi \cdot \max(0, g(x))^2,$$

after conducting the following:

Algorithm 1: Penalty-based method for problem stated in Example 14.1.

$k \leftarrow 0$
$x^0 \leftarrow (-1, -1)$
$\rho \leftarrow 1$
$\phi \leftarrow 1$
Update penalized objective function $F_{\rho,\phi}$
$grad \leftarrow \nabla F_{\rho,\phi}(x^k)$
while $grad \neq 0$ or x^k is not in C **do**
    $d^k \leftarrow -grad$
    Determine step size, α^k
    $x^{k+1} \leftarrow x^k + \alpha^k \cdot d^k$
    $k \leftarrow k + 1$
    if $h(x^k) \neq 0$ **then**
        $\rho \leftarrow 3 \cdot \rho$
    end
    if $g(x^k) > 0$ **then**
        $\phi \leftarrow 3 \cdot \phi$
    end
    Update penalized objective function $F_{\rho,\phi}$
    $grad \leftarrow \nabla F_{\rho,\phi}(x^k)$
end

We have:

```
Iteration 0
x               :       Matrix([[-1], [-1]])
f(x)            :       17.0000000000000
direction       :       Matrix([[14.0000000000000], [36.0000000000000]])
alpha           :       0.1


Iteration 1
x               :       Matrix([[0.400000000000000], [2.60000000000000]])
f(x)            :       5.92000000000000
direction       :       Matrix([[-14.4000000000000], [-18.4000000000000]])
alpha           :       0.010000000000000002


Iteration 2
x               :       Matrix([[0.256000000000000], [2.41600000000000]])
f(x)            :       5.43059200000000
direction       :       Matrix([[-24.0960000000000], [-38.0000000000000]])
alpha           :       0.010000000000000002
```

Figure 14.1: Results of first iterations.

```
Iteration 54
x               :       Matrix([[-4.70426835633760e-6], [2.00000282317768]])
f(x)            :       4.99997553660132
direction       :       Matrix([[-0.00123574830289818], [-0.00266027827621979]])
alpha           :       1.0000000000000005e-07


Iteration 55
x               :       Matrix([[-4.70439193116789e-6], [2.00000282291165]])
f(x)            :       4.99997553663907
direction       :       Matrix([[-0.000407545140831244], [-0.00126656283986663]])
alpha           :       1.0000000000000005e-07


x               :       Matrix([[-4.70443268568197e-6], [2.00000282278499]])
f(x)            :       4.99997553672937
```

Figure 14.2: Results of last iterations.