Documentation of Objective-First Numerical Methods

Jesse Lu

November 3, 2011

Contents

1 Interior-point algorithm

As taken from section 19.3 of [1], the interior point method obtains step direction p by solving

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} & 0 & A_E^T(x) & A_I^T(x) \\ 0 & \Sigma & 0 & -I \\ A_E(x) & 0 & 0 & 0 \\ A_I(x) & -I & 0 & 0 \end{bmatrix} \begin{bmatrix} p_x \\ p_s \\ -p_y \\ -p_z \end{bmatrix} = - \begin{bmatrix} \nabla f(x) - A_E^T(x)y - A_I^T(x)z \\ z - \mu S^{-1}e \\ c_E(x) \\ c_I(x) - s \end{bmatrix}. \tag{1}$$

This equation can be simplified by removing p_s and then p_z. The reduced system is then

$$\begin{bmatrix} \nabla^2_{xx}\mathcal{L} + A_I^T(x)\Sigma A_I^T(x) & A_E^T(x) \\ A_E(x) & 0 \end{bmatrix} \begin{bmatrix} p_x \\ -p_y \end{bmatrix} = - \begin{bmatrix} \nabla f(x) - A_E^T(x)y - A_I(x)(z - \Sigma c_I(x) + \mu S^{-1}e) \\ c_E(x) \end{bmatrix}, \tag{2}$$

where

$$p_s = A_I(x)p_x + c_I(x) - s \tag{3}$$

$$p_z = -\Sigma A_I(x)p_x - \Sigma c_I(x) + \mu S^{-1}e. \tag{4}$$

We can focus the problem by only considering simple bound inequality constraints $l \leq x \leq u$, and affine equality constraints $Ax - b = 0$. Then our problem is written down as

$$\begin{bmatrix} \nabla^2 f(x) + \Sigma_0 + \Sigma_1 & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} p_x \\ -p_y \end{bmatrix} = - \begin{bmatrix} \nabla f(x) - A^T y + h_0 + h_1 \\ Ax - b \end{bmatrix}, \tag{5}$$

where

$$h_0 = -z_0 + \Sigma_0(x - l) - \mu S_0^{-1} e \tag{6}$$

$$h_1 = z_1 - \Sigma_1(u - x) + \mu S_1^{-1} e, \tag{7}$$

the other components of p are

$$p_{s_0} = p_x + (x - l) - s_0 \tag{8}$$

$$p_{z_0} = -\Sigma_0 p_x - \Sigma_0(x - l) + \mu S_0^{-1} e \tag{9}$$

$$p_{s_1} = -p_x + (u - x) - s_1 \tag{10}$$

$$p_{z_1} = \Sigma_1 p_x - \Sigma_1(u - x) + \mu S_1^{-1} e, \tag{11}$$

and the error function used is

$$\begin{aligned} E(x, s_0, s_1, y, z_0, z_1, \mu) = \max\{&\|\nabla f(x) - A^T y - z_0 + z_1\|, \\ &\|S_0 z_0 - \mu e\|, \|S_1 z_1 - \mu e\|, \|Ax - b\|, \|(x - l) - s_0\|, \|(u - x) - s_1\|\}. \end{aligned} \tag{12}$$

Lastly, inspired from section 11.7.3 of [2], we perform a backtracking line search (see section 9.2 or [2]) in order to guarantee decrease of the residual $r(x^+, s_0^+, s_1^+, y^+, z_0^+, z_1^+, \mu)$ where,

$$\begin{aligned} x^+ &= x + t\alpha_p p_x \\ s_0^+ &= s_0 + t\alpha_p p_{s_0} \\ s_1^+ &= s_1 + t\alpha_p p_{s_1} \\ y^+ &= y + \alpha_d p_y \\ z_0^+ &= z_0 + \alpha_d p_{z_0} \\ z_1^+ &= z_1 + \alpha_d p_{z_1} \end{aligned}$$

and,

$$r(x, s_0, s_1, y, z_0, z_1, \mu) = \left\| \begin{bmatrix} \nabla f(x) - A^T y + (-z_0 + \Sigma_0(x - l) - \mu S_0^{-1} e) + (z_1 - \Sigma_1(u - x) + \mu S_1^{-1} e) \\ Ax - b \end{bmatrix} \right\|_2. \tag{13}$$

The exit condition for the line search is

$$r(x^+, s_0^+, s_1^+, y^+, z_0^+, z_1^+, \mu) \leq (1 - \alpha t) r(x, s_0, s_1, y, z_0, z_1, \mu). \tag{14}$$

where t is initially set to $t = \alpha_p$.

2 Solving an augmented-arrow system

We now address the solution of a system with the following form:

$$\left(\begin{bmatrix} D & A^T \\ A & 0 \end{bmatrix} + UV^T\right) x = (\hat{A} + UV^T)x = b. \tag{15}$$

First, we obtain a method to solve $\hat{A}^{-1}x$. We choose to use block substitution to do so. Such a method solves

$$\hat{A}y = \begin{bmatrix} D & A^T \\ A & 0 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} d_1 \\ d_2 \end{bmatrix} \tag{16}$$

by computing

$$y_2 = S^{-1}(d_2 - AD^{-1}d_1) \tag{17}$$
$$y_1 = D^{-1}(d_1 - A^T y_2). \tag{18}$$

Next, we solve 15 by employing the matrix inversion lemma,

$$(A + UV^T)^{-1} = A^{-1} - A^{-1}U(I + V^T A^{-1}U)^{-1}V^T A^{-1}, \tag{19}$$

in the following way:

$$Y = \hat{A}^{-1}U \tag{20}$$
$$z = \hat{A}^{-1}b \tag{21}$$
$$x = z - Y(I + V^T Y)^{-1}V^T z. \tag{22}$$

3 Outer-product form of L-BFGS

We use an outer-product form of the limited-memory Broyden-Fletcher-Goldfarb-Shanno (L-BFGS) algorithm (See chapter 7.2 of [1]). This form produces a diagonal plus low-rank approximation of the Hessian of a function,

$$\tilde{H} = \delta I + WMW^T \tag{23}$$

based on a limited sampling of previous gradients of the function.

4 Backtracking-line search

References

[1] Nocedal and Wright, Numerical Optimization, Second Edition (Cambridge 2004)

[2] Boyd and Vandenberghe, Convex Optimization (Cambridge 2004)