Assignment - Constrained optimization Interior Point Method applied to linear programming problems

Matteo Bunino

January 21, 2020

1 Problem definition

The Predictor-Corrector is an Interior Point Method built to iteratively solve a nonlinear system of equations, obtained by defining the KKT condition for the pair primal-dual problem.
The linear system to solve is:

$$F(x, \lambda, s) = \begin{bmatrix} Ax - b \\ s + A^T \lambda - c \\ XSe \end{bmatrix} = 0$$

According to Newton method for nonlinear system of equation, the system can locally linearized and solved. To linearize this system is necessary to compute the Jacobian of F at each iteration and solve the linear system:

$$F'(x_k, \lambda_k, s_k) \cdot \begin{pmatrix} \Delta x_k \\ \Delta \lambda_k \\ \Delta s_k \end{pmatrix} + F(x_k, \lambda_k, s_k) = 0$$

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S_k & 0 & X_k \end{pmatrix} \cdot \begin{pmatrix} \Delta x_k \\ \Delta \lambda_k \\ \Delta s_k \end{pmatrix} + F(x_k, \lambda_k, s_k) = 0$$

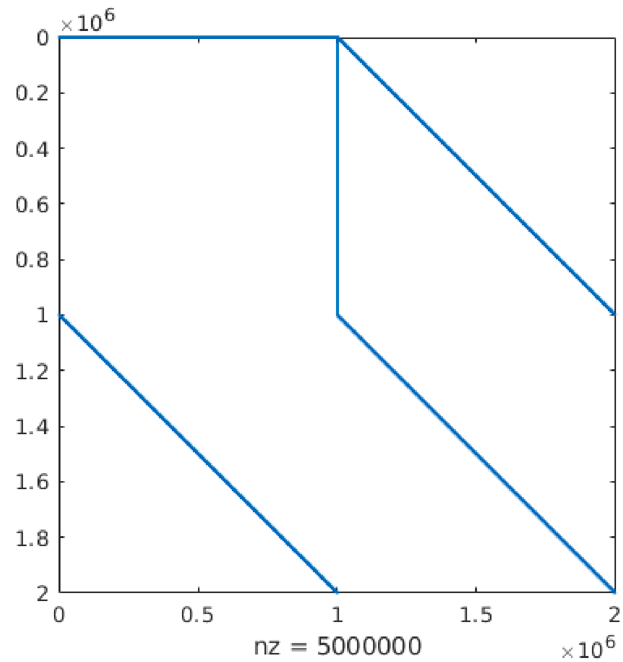
Where $X_k = diag(x_k)$ and $S_k = diag(s_k)$.
Interior point methods are variants of Newton method that guarantee that each iterate (x_k, λ_k, s_k) satisfy the constraint $x, s \geq 0$.

2 Problem data

$A = (1, \dots 1) \in \Re^{1*n}$,
$b = 1$,

$c \in \Re^n : c_i = a$ if i is odd, otherwise 1.
$x \in \Re^n$.

After having built the sparse jacobian of F, using the command spy() it's possible to inspect its structure:



Using the command [R,p] = chol(J) it's possible to notice, thanks to the value of p, that the matrix isn't positive definite. Hence I could not use: gradient method, conjugate gradient or Cholesky decomposition.
Since this matrix isn't symmetric, I didn't use the LDL decomposition neither.

Analyzing the structure of the block matrix, I could notice that it's possible to make some reductions, rewriting the left hand part in a more compact way:

$$\begin{pmatrix} A & 0 & 0 \\ 0 & A^T & I \\ S_k & 0 & X_k \end{pmatrix} \cdot \begin{pmatrix} \Delta x_k \\ \Delta \lambda_k \\ \Delta s_k \end{pmatrix} = \begin{pmatrix} r_a \\ r_b \\ r_c \end{pmatrix}$$

Since X is invertible ($x > 0$ is guarateed by the IPM):

$$\begin{pmatrix} A & 0 \\ -X^{-1}S & A^T \end{pmatrix} \cdot \begin{pmatrix} \Delta x_k \\ \Delta \lambda_k \end{pmatrix} = \begin{pmatrix} r_a \\ r_b - X^{-1} r_c \end{pmatrix}$$

S is invertible for the same reason:

$$AS^{-1}XA^T\Delta\lambda = r_a + AXS^{-1}r_b - AS^{-1}r_c$$
$$\Delta s = r_b - A^T\Delta\lambda$$
$$\Delta x = S^{-1}r_c - XS^{-1}\Delta s$$

The matrix $AS^{-1}XA^T \in \Re^{mxm}$ is a real number, since in this case $m = 1$. This makes the computation of the system trivial, avoiding the problem of working with singular or indefinite matrices.

3 Results

n iter		a			
		2	20	200	2000
n	1,00E+04	7	9	9	10
	1,00E+06	8	10	9	11

time		a			
		2	20	200	2000
n	1,00E+04	0,1542	0,1038	0,0988	0,1635
	1,00E+06	9,1605	11,1476	10,1838	12,089

Figure 1: Starting point of all ones.

n iter		a			
		2	20	200	2000
n	1,00E+04	9	10	10	11
	1,00E+06	9	10	10	12

time		a			
		2	20	200	2000
n	1,00E+04	0,166	0,1329	0,1328	0,1449
	1,00E+06	10,2799	11,78	13,3218	13,8344

Figure 2: Starting point of all 100.

n iter		a			
		2	20	200	2000
n	1,00E+04	4	4	4	4
	1,00E+06	4	4	4	4

Time		a			
		2	20	200	2000
n	1,00E+04	0,1044	0,0677	0,0662	0,0657
	1,00E+06	5,2523	5,274	5,193	5,2321

Figure 3: Starting point as suggested on the book.