Constrained Optimization Algorithms I

Yinyu Ye
Department of Management Science and Engineering
Stanford University
Stanford, CA 94305, U.S.A.

http://www.stanford.edu/~yyye

Chapters 12.1-3, 12.5, 12.7

Methods for Solving Constrained Optimization

The gradient projection method: project the gradient-solution onto the feasible set (first-order).

The feasible-direction method: search along a feasible and descent direction (first or second order).

The reduced gradient method: like the simplex method by changing non-basic variables (first-order).

Newton's method for computing the roots of the KKT equations (second order)

The Lagrangian method: relax equality constraints into the objective function and update the multipliers (first or second order).

The barrier function or interior-point method: Force the iterative points inside of the feasible region while improve the objective function value (first or second-order).

Equality and Inequality Constrained Problems

$$\boxed{\begin{array}{ll} \min & f(\boldsymbol{x}) \\ \text{s.t.} & A\boldsymbol{x} = \boldsymbol{b} \\ & \boldsymbol{x} \ge \boldsymbol{0} \end{array}}$$

KKT Conditions:

$$\nabla f(\mathbf{x}) - A^T\boldsymbol{y} - \boldsymbol{r} = \boldsymbol{0}$$

$$A\boldsymbol{x} = \boldsymbol{b},\ (\boldsymbol{x},\boldsymbol{r}) \ge \boldsymbol{0}$$

$$\boldsymbol{x}_j\boldsymbol{r}_j = 0, \text{ for all } j$$

Thus, we are interested in finding an ε-solution pair $(\boldsymbol{x},\boldsymbol{y},\boldsymbol{r})$ such that

Approximate KKT Conditions: $(\boldsymbol{x},\boldsymbol{r}) \ge \boldsymbol{0}$ *and* $\boldsymbol{y}$

$$\|\nabla f(\mathbf{x}) - A^T\boldsymbol{y} - \boldsymbol{r}\| \le \epsilon$$

$$\|A\boldsymbol{x} - \boldsymbol{b}\| \le \epsilon$$

$$x_j r_j \le \epsilon, \text{ for all } j$$

Gradient-Projection for Cone-Constrained Optimization

$$\begin{aligned} &\min \quad f(\boldsymbol{x}) \\ &\text{s.t.} \quad \boldsymbol{x} \geq \boldsymbol{0} \end{aligned}$$

KKT Conditions

$\nabla f(\mathbf{x}) - \mathbf{y} = \mathbf{0},\ \mathbf{y} \geq \mathbf{0}$ (LDC)

$\mathbf{x}.^{*}\ \mathbf{y} = \mathbf{0}$ (CSC)

The Gradient-solution projection on to the feasible region would be:

$$\mathbf{x}^{k+1} = \max\{\boldsymbol{0},\ \mathbf{x}^{k} - \beta^{-1}\nabla f(\mathbf{x}^{k})\}$$

$$\boldsymbol{y}^{k+1} = \max\{\boldsymbol{0},\ \nabla f(\mathbf{x}^{k})\}\ \text{(dual)}$$

The step-size could be chosen to minimize the function value.
What to do if there are up-bounds on the decision variables?

Gradient-Projection for Cone Constrained QP

```
% start from the given initial solution
x=x0;
norm(Q*x+c)
for k=1:100,
  g=(Q*x+c);
  x=max(0, x-(1/beta)*g);
end;
y=max(0,g);
norm(x.*y)
% Steepest Descent and Projection Method for solving
convex QP
%
%     minimize    0.5x'*Q*x+c'*x, s.t.  x>=0
```

Gradient-Projection for Equality-Constrained Optimization

$$\boxed{\begin{array}{ll} \min & f(\mathbf{x}) \\ \text{s.t.} & A\mathbf{x}-\mathbf{b} = \mathbf{0} \end{array}} \Rightarrow \boxed{\begin{array}{c} KKT \\ \nabla f(\mathbf{x}) - A^T\mathbf{y} = \mathbf{0} \\ \textbf{(LDC)} \end{array}}$$

$$\mathbf{x}^{k+1} = \mathbf{x}^k - \beta^{-1}(I - A^T(AA^T)^{-1}A)\nabla f(\mathbf{x}^k)$$

$$\mathbf{y}^{k+1} = (AA^T)^{-1}A\nabla f(\mathbf{x}^k) \text{ (dual)}$$

Assuming that the initial solution is feasible, then all following solutions are feasible

What to do if there are also bounds on the decision variables such as $\mathbf{x} \geq \mathbf{0}$?

An alternating (projection) method would be discussed later.

The GP Method for ECQP: Convergence Speed

First-Order β-Lipschitz *f*: One can simply let $\mathbf{d}^k=-(I-A^T(AA^T)^{-1}A)\nabla f(\mathbf{x}^k)$ and stepsize to be fixed for all iterations at β^{-1}, that is,

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \beta^{-1}\mathbf{d}^k \text{ with } \mathbf{y}^k= (AA^T)^{-1}A\nabla f(\mathbf{x}^k) \text{ (dual)}$$

Then, the following theorem can be proved.

Theorem. Let the problem admit a minimizer $\mathbf{x}^*$ and it satisfies the first-order β-Lipschitz condition. Stating from $\mathbf{x}^0$ such that $A\mathbf{x}^0=\mathbf{b}$, then in at most $2\beta(f(x^0)-f(x^*))/\varepsilon^2$ steps

$$\|\nabla f(\mathbf{x}^k)-A^T\mathbf{y}^k\| \le \epsilon \text{ and } A\mathbf{x}^k=\mathbf{b}.$$

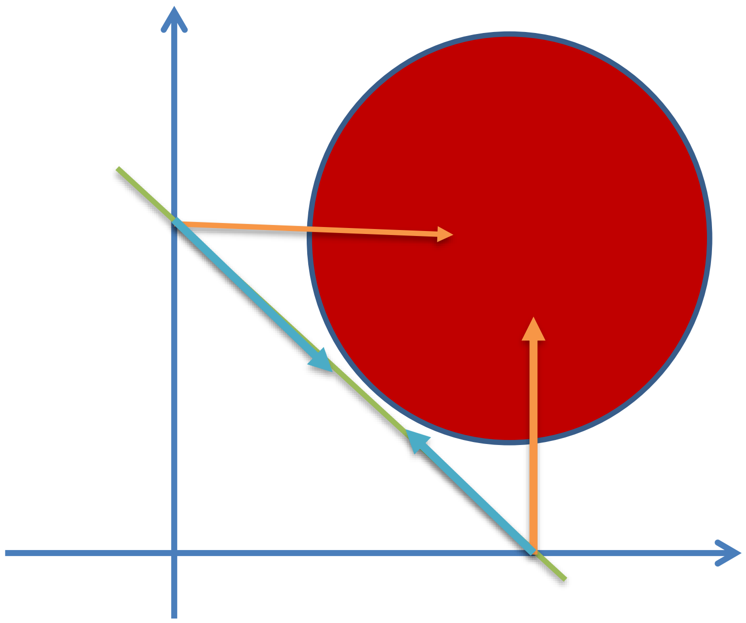
Here $P_A=(I-A^T(AA^T)^{-1}A)$ is called the projection matrix. Note $(P_A)^2=P_A$.

$$\begin{aligned}
f(\mathbf{x}^{k+1}) - f(\mathbf{x}^*) &= f(\mathbf{x}^k + \beta^{-1}\mathbf{d}^k)) - f(\mathbf{x}^*)\\
&\le f(\mathbf{x}^k) + \beta^{-1}\nabla f(\mathbf{x}^k)^T\mathbf{d}^k + \frac{\beta}{2}\beta^{-2}\|\mathbf{d}^k\|^2 - f(\mathbf{x}^*)\\
&= f(\mathbf{x}^k) - \beta^{-1}\|\mathbf{d}^k\|^2 + \frac{1}{2}\beta^{-1}\|\mathbf{d}^k\|^2 - f(\mathbf{x}^*)\\
&= f(\mathbf{x}^k) - f(\mathbf{x}^*) - \frac{1}{2}\beta^{-1}\|\mathbf{d}^k\|^2
\end{aligned}$$

Thus, in at most that number of steps, $\|\mathbf{d}^k\| \le \epsilon$. Let $\mathbf{y}^k= (AA^T)^{-1}A\nabla f(\mathbf{x}^k)$ we have the desired result. Finally, $A\mathbf{x}^{k+1}=A(\mathbf{x}^k + \beta^{-1}\mathbf{d}^k)=A\mathbf{x}^k=\mathbf{b}$ for all *k*.

A Linearly Constrained Example

$$\begin{aligned} \min \quad & 0.5(x_1-1)^2+0.5(x_2-1)^2 \\ \text{s.t.} \quad & x_1+x_2-1=0 \end{aligned}$$



$$A=(1 \quad 1), \qquad x=\begin{pmatrix}1\\0\end{pmatrix}, \qquad \nabla f=\begin{pmatrix}0\\-1\end{pmatrix}$$

$$P=(I-A^T(AA^T)^{-1}A)=\begin{pmatrix}0.5 & -0.5\\ -0.5 & 0.5\end{pmatrix}$$

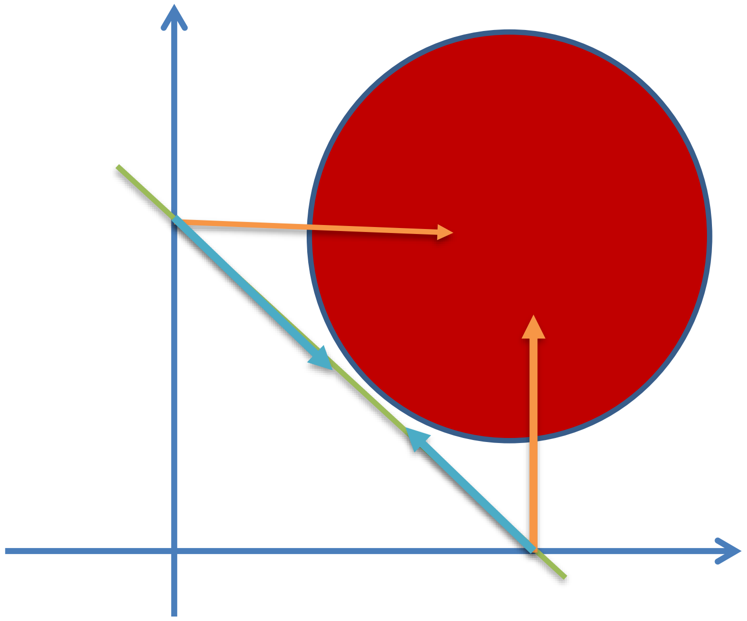
$$\text{The projected direction}: \text{-}P\nabla f=\sqrt{2}\begin{pmatrix}\text{-}0.5\\ 0.5\end{pmatrix}$$

$$\text{or simply} \qquad \begin{pmatrix}\text{-}0.5\\ 0.5\end{pmatrix}$$

If the line search is applied, one can find the minimizer in one step.

SLP Matlab Code for the Linearly Constrained Example

$$\begin{aligned} \min \quad & 0.5(x_1-1)^2+0.5(x_2-1)^2 \\ \text{s.t.} \quad & x_1+x_2-1=0 \end{aligned}$$



```
%
%P=eye(2)-A'*inv(A*A')*A;
g=[x(1)-1;x(2)-1];
y=(A*A')\(A*g);
gp=g-A'*g;
x=x-gp
% Gradient projection method for solving
%
%      minimize    0.5(x(1)-1)^2+0.5(x(2)-1)^2
%      subject to    x(1)+x(2)-1=0
%
%      Input: any initial feasible x
%
```

Fixing the step-size and starting from any feasible solution

The multipliers would be given by y=(A*A')\(Ag)

Machine Learning: The Wassestein Barycenter Problem I

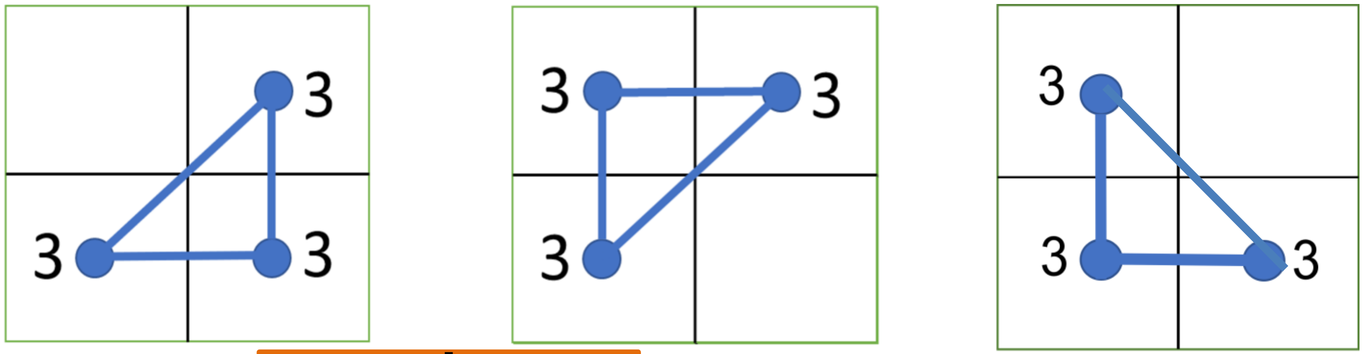
WD(s, d^k)=

$$\begin{array}{ll} \min & \sum_{i=1}^{N}\sum_{j=1}^{N} c_{ij}x_{ij} \\ \text{s.t.} & \sum_{j=1}^{N} x_{ij} = s_i, \quad \forall\, i = 1,\dots,N \\ & \sum_{i=1}^{N} x_{ij} = d_j, \quad \forall\, j = 1,\dots,N \\ & x_{ij} \ge 0, \quad \forall\, i,j \end{array}$$

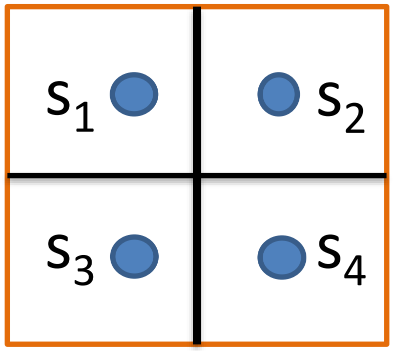
The minimal transportation cost in Data Science is called the Wasserstein distance between a supply distribution and a demand distribution.

The **Wasserstein Barycenter Problem** is to find a distribution/points such that the sum of its Wasserstein distances to each of a set of distributions/points would be minimized

$\min_s \sum_k$ **WD(s, d^k) s.t. total mass constraint**



Three possible demand distribution scenario of 4 cities



Constraints:

$s_1+s_2+s_3+s_4=9$

$(s_1,s_2,s_3,s_4)>=0$

$$C=\begin{pmatrix} 0 & 1 & 1 & 2 \\ 1 & 0 & 2 & 1 \\ 1 & 2 & 0 & 1 \\ 2 & 1 & 1 & 0 \end{pmatrix}$$

General Feasible-Descent Direction Methods

Starting with a feasible solution $\mathbf{x}^k$, we consider again an iterative scheme of the form

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \alpha_k \mathbf{d}^k$$

where $\mathbf{d}^k$ is a search direction vector, both feasible and descent, and scalar α_k is again the step-size.

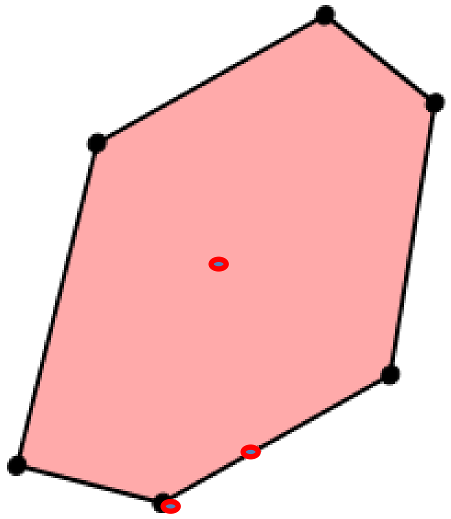
In fact, once the search direction is chosen, the objective function can be written as $f(\mathbf{x}^k + \alpha \mathbf{d}^k)$, which is just function of α. Thus one can choose step-size α as α_k according to some line (one-variable optimization) search to minimize the function WHILE keep $\mathbf{x}^k + \alpha_k \mathbf{d}^k$ feasible.

This is exact what the LP Simplex method does, where the descent direction is chosen with a negative reduced cost and the step-size is the largest possible to keep the iterate feasible.

Feasible Direction Space

Again, at feasible point $\mathbf{x}$, a feasible direction is

$F_{\mathbf{x}} := \{\mathbf{d} \in R^n \quad : \mathbf{d} \neq \mathbf{0},\ \mathbf{x} + \lambda\mathbf{d} \in F \text{ for all small } \lambda > 0\}.$



Examples:

Unconstrained: $R^n \quad \Rightarrow \quad F_x = R^n.$

Linear Equality: $\{\ \mathbf{x} : A\mathbf{x} = \mathbf{b}\} \Rightarrow F_x = \{\mathbf{d} : A\mathbf{d} = \mathbf{0}\ \}.$

Linear Inequality: $\{\ \mathbf{x} : A\mathbf{x} \geq \mathbf{b}\} \Rightarrow F_x = \{\mathbf{d} : \boldsymbol{a}_i \mathbf{d} \geq \mathbf{0},\ \forall i \in B(\boldsymbol{x})\},$

where $B(\boldsymbol{x})$ is the binding constraint index set

$$B(\boldsymbol{x}) := \{\ i : \boldsymbol{a}_i \boldsymbol{x} = b_i\ \}.$$

Linear Equality and Nonnegativity: $\{\ \mathbf{x} : A\mathbf{x} = \mathbf{b}, \boldsymbol{x} \geq \mathbf{0}\} \Rightarrow$

$$F_x = \{\mathbf{d} : A\mathbf{d} = \mathbf{0},\ d_i \geq 0,\ \forall i \in B(\boldsymbol{x})\ \},$$

where $\qquad B(\boldsymbol{x}) := \{\ i : x_i = 0\ \}.$

Find a Feasible-and-Descent Direction: Sequential LP

$$\boxed{\begin{array}{ll} \min & \nabla f(\mathbf{x}^k)^T \boldsymbol{d} \\ \text{s.t.} & \mathbf{d} \in F_{x^k} \\ & \|\boldsymbol{d}\| = 1 \end{array}} \quad \Rightarrow \quad \boxed{\begin{array}{ll} \min & \nabla f(\mathbf{x}^k)^T \boldsymbol{d} \\ \text{s.t.} & \mathbf{d} \in F_{x^k} \\ & \|\boldsymbol{d}\| \leq 1 \end{array}}$$

The minimizer would be the $\boldsymbol{d}^k$. This problem can be relaxed to a convex optimization problem

If the norm is 1 or ∞, it becomes linear program. Such a method is also called sequential LP method (SLP or Frank-Wolf).
If the norm is 2, it becomes a quadratic program, and sometime the solution has a close form if only equality constraints present.
If the direction minimizer $\nabla f(\mathbf{x}^k)^T \boldsymbol{d}^* < 0$, then we find a feasible and descent direction so that the objective can be reduced.

The Reduced Gradient/Newton Method

$$\boxed{\begin{array}{ll} \min & f(\boldsymbol{x}) \\ \text{s.t.} & \boldsymbol{Ax}-\boldsymbol{b} = \boldsymbol{0} \\ \nabla f(\mathbf{x}) - A^T\mathbf{y} = \mathbf{0} & \end{array}} \Longrightarrow \boxed{\begin{array}{ll} \min & f(\boldsymbol{b}-(\boldsymbol{A}_B)^{-1}\boldsymbol{A}_N\boldsymbol{x}_N, x_N) \\ \text{s.t.} & \text{Substitute } \mathbf{x}_B \text{ using} \\ & \boldsymbol{x}_B=\boldsymbol{b}-(\boldsymbol{A}_B)^{-1}\boldsymbol{A}_N\boldsymbol{x}_N \end{array}}$$

$$\boxed{\min \quad f(\boldsymbol{b}-(\boldsymbol{A}_B)^{-1}\boldsymbol{A}_N\boldsymbol{x}_N, x_N)}$$

This is also similar to what the LP Simplex method does (there the basis is updated in the reduced form). The gradient vector of the function is also called the reduced gradient of $\mathbf{x}_N$ only, and it is identical to the reduced cost coefficient vector of $\mathbf{x}_N$ only.

Direct Newton's Method for Nonlinear ECOP

The KKT Equations and its Jacobian Matrix:

$$\mathbf{g}(x,y) = \begin{pmatrix} \nabla f(x) - A^T y \\ Ax - b \end{pmatrix} (=0)$$

$$\nabla \mathbf{g}(x,y) = \begin{pmatrix} \nabla^2 f(x) & -A^T \\ A & 0 \end{pmatrix}$$

$$\begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ y^k \end{pmatrix} \boxed{- \nabla \mathbf{g}(x^k, y^k)^{-1} \mathbf{g}(x^k, y^k)}$$

Newton Direction Vector

The Portfolio QP Example

$$\begin{aligned} \min_{(x_1,x_2)} \quad & (x_1)^2 + 2(x_2)^2 - 2x_1x_2 - 0.5x_1 - 0.5x_2 \\ \text{s.t.} \quad & x_1 + x_2 = 1, \end{aligned}$$

Newton's Method for Linear Equality Constrained QP

$$\begin{aligned} \min \quad & \frac{1}{2} x^T Q x + c^T x \\ \text{s.t.} \quad & Ax = b \end{aligned}$$

KKT System

$$\begin{pmatrix} 2 & -2 & -1 \\ -2 & 4 & -1 \\ 1 & 1 & 0 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ y \end{pmatrix} = \begin{pmatrix} 0.5 \\ 0.5 \\ 1 \end{pmatrix}$$

$$\begin{pmatrix} Q & -A^T \\ A & 0 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -c \\ b \end{pmatrix}$$

Newton's Method for Nonlinear ECOP

$$\min f(\boldsymbol{x}) \quad \text{s.t.} \quad h(\boldsymbol{x}) = 0$$

The KKT Equations and its Jacobian Matrix:

$$\mathbf{g}(x,y) = \begin{pmatrix} \nabla f(x) - \sum_i y_i \nabla h_i(x) \\ \mathbf{h}(x) \end{pmatrix} (=0)$$

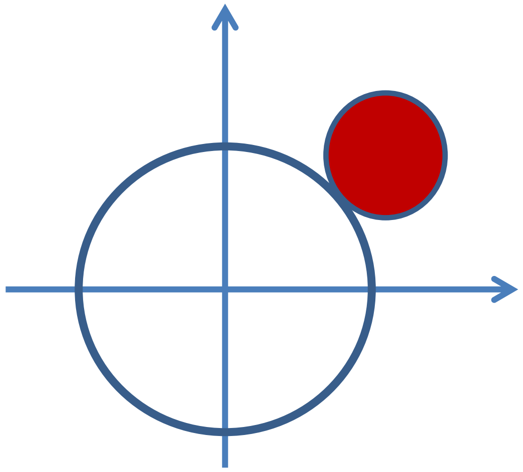
$$\nabla \mathbf{g}(x,y) = \begin{pmatrix} \nabla^2 f(x) - \sum_i y_i \nabla^2 h_i(x) & -\nabla \mathbf{h}(x)^T \\ \nabla \mathbf{h}(x) & 0 \end{pmatrix}$$

$$\begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ y^k \end{pmatrix} - \boxed{\nabla \mathbf{g}(x^k, y^k)^{-1} \mathbf{g}(x^k, y^k)}$$

Newton Direction Vector

A Spherical Constraint QP Example I

$$\begin{array}{ll} \min & (x_1-1)^2+(x_2-1)^2 \\ \text{s.t.} & (x_1)^2+(x_2)^2 \quad -1=0 \end{array}$$



$$\mathbf{g}(x,y)=\begin{pmatrix} \begin{pmatrix} 2(x_1-1) \\ 2(x_2-1) \end{pmatrix} - y\begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix} \\ (x_1)^2+(x_2)^2 \quad -1 \end{pmatrix} (=0)$$

$$\nabla\mathbf{g}(x,y)=\begin{pmatrix} \begin{pmatrix} 2-2y & 0 \\ 0 & 2-2y \end{pmatrix} & -\begin{pmatrix} 2x_1 \\ 2x_2 \end{pmatrix} \\ (\,2x_1 \quad 2x_2\,) & 0 \end{pmatrix}$$

$$\begin{pmatrix} x^{k+1} \\ y^{k+1} \end{pmatrix} = \begin{pmatrix} x^k \\ y^k \end{pmatrix} - \nabla\mathbf{g}(x^k,y^k)^{-1}\mathbf{g}(x^k,y^k)$$

A Spherical Constraint QP Example II

MATLAB Implementation

```
G=[2-2*y 0 -2*x(1);0 2-2*y -2*x(2);2*x(1)   2*x(2) 0];
g=[2*(x(1)-1)-2*y*x(1);2*(x(2)-1)-2*y*x(2);x(1)^2+x(2)^2-1];
G\g;
x=x-ans(1:2);y=y-ans(3);
% Newton for
%
%      minimize    (x(1)-1)^2+(x(2)-1)^2
%      subject to    x(1)^2+x(2)^2-1=0
%
%      Input: good initial x and y
```

Descent-First Feasible-Second Approach: Steepest-Descent Projection Method? See ProjQCQPexample211.m