Methods for Degenerate Nonlinear Programming

Olga Brezhneva

Miami University, Oxford, Ohio

in collaboration with

Alexey Tret'yakov

University of Podlasie in Siedlce, Poland

5th MOPTA Conference Modeling and Optimization: Theory and Applications

July 25-27, 2005 - University of Windsor, Windsor ON Canada

Outline

- Background
- Problem statement
- Motivation
- Optimality conditions and Lagrange–Newton method
- P-factor-Lagrange method
- Nonlinear complementarity problems
- Conclusion

Background

Consider the nonlinear programming (NLP) problem:

$$\begin{array}{ll} \underset{x \in \mathbb{R}^n}{\text{minimize}} & f(x) \\ \text{subject to} & g_1(x) \le 0, \ldots, g_m(x) \le 0, \end{array}$$

where f and g_j are smooth functions from $\mathbb{R}^n$ to $\mathbb{R}$.

We focus on finding a *local solution* x^* of the NLP.

The *Lagrangian* for the NLP problem is defined as

$$\mathcal{L}(x, \lambda) = f(x) + \sum_{j=1}^{m} \lambda_j g_j(x),$$

where $\lambda = (\lambda_1, \ldots, \lambda_m)$ is the vector of Lagrange multipliers.

Constraint qualification

Introduce the set of *active constraints:*

$$A(x^*) = \{j = 1, \ldots, m \,|\, g_j(x^*) = 0\}.$$

Constraint qualifications:

- *Linear Independence Constraint Qualification (LICQ)*: The gradients $\nabla g_j(x^*),\ j \in A(x^*)$, are linearly independent.

Constraint qualification

Introduce the set of *active constraints:*

$$A(x^*) = \{j = 1, \ldots, m \,|\, g_j(x^*) = 0\}.$$

Constraint qualifications:

- *Linear Independence Constraint Qualification (LICQ)*: The gradients $\nabla g_j(x^*),\ j \in A(x^*)$, are linearly independent.
- *Mangasarian-Fromovitz Constraint Qualification (MFCQ)*: There is a direction d such that $\nabla g_j(x^*)^T d < 0$ for all $j \in A(x^*)$.

First-Order Necessary Conditions

If x^* is a local minimizer of the NLP problem and a CQ is satisfied at x^*, then there is $\lambda^* \in \mathbb{R}^m$ such that

$$\nabla_x \mathcal{L}(x^*, \lambda^*) = \nabla f(x^*) + \sum_{j=1}^{m} \lambda_j^* \nabla g_j(x^*) = 0$$

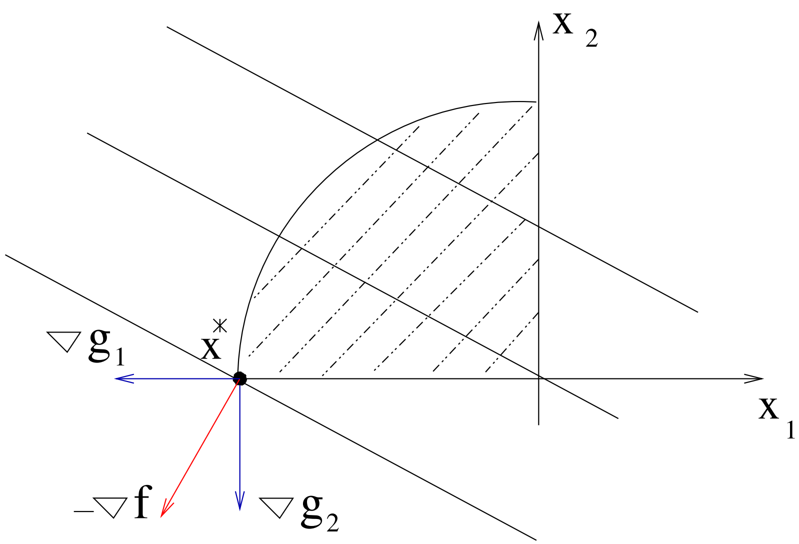
$$g(x^*) \leq 0, \quad \lambda^* \geq 0, \quad g_j(x^*)\lambda_j^* = 0, \quad j = 1, \dots, m$$

“Objective anti-gradient is a conic combination of the active constraint gradients"

Question 1: How to use the optimality conditions to solve the NLP?

Conditions for Optimality: *Example*

$$\begin{array}{ll} \underset{x\in\mathbb{R}^2}{\text{minimize}} & f(x_1,x_2) = x_1 + 2x_2 \\ \text{subject to} & g_1(x_1,x_2) = x_1^2 + x_2^2 - 2 \le 0 \\ & g_2(x_1,x_2) = -x_2 \le 0, \quad g_3(x_1,x_2) = x_1 \le 0 \end{array}$$



The minimizer $x^* = (-\sqrt{2}, 0)$ satisfies

$$-\nabla f(x^*) = \lambda_1^* \nabla g_1(x^*) + \lambda_2^* \nabla g_2(x^*), \quad \lambda_1^* \ge 0, \quad \lambda_2^* \ge 0.$$

Standard Assumptions

Local convergence analysis of algorithms *usually assumes*

- Constraint qualification (LICQ or MFCQ);
- Strict complementarity condition (SCC): exactly one of λ_j^* and $g_j(x^*)$ is zero for each index $j = 1, \ldots, m$.
- The second-order sufficient condition (SOSC): there exists $\nu > 0$ such that

$$\omega^T \mathcal{L}_{xx}(x^*, \lambda^*)\omega > \nu \|\omega\|^2$$

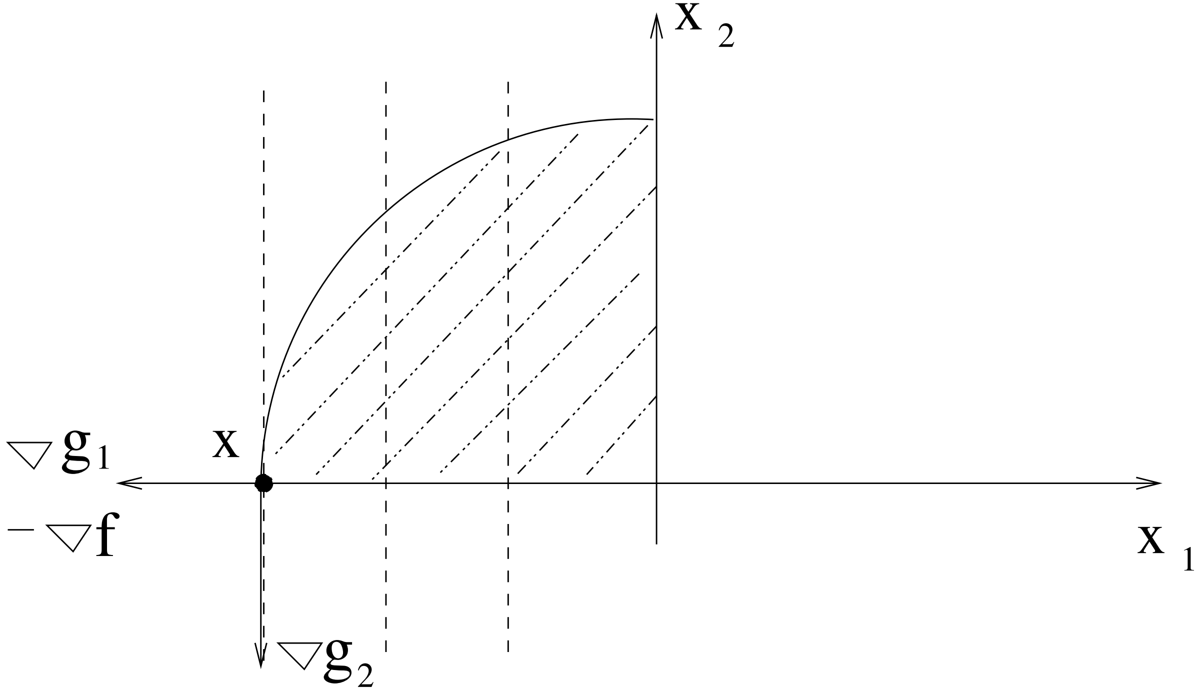
for all ω such that $\nabla g_j(x^*)^T \omega = 0, \quad$ for all $\quad j \in A(x^*)$.

Question 2: Are these assumptions always satisfied?

Example: No SCC

SCC: $\lambda_j^* > 0$ for each $g_j(x^*) = 0$.

Example. The SCC *does not hold:* $\lambda_2^* = 0, \; g_2(x^*) = 0.$



$$-\nabla f(x^*) = \lambda_1^* \nabla g_1(x^*) + 0 \nabla g_2(x^*)$$

Example: No SOSC

There are problems where the SOSC *does not hold*:

$$\begin{aligned} \underset{x}{\text{minimize}} \quad & x^4 \\ \text{subject to} \quad & x^3 \leq 0. \end{aligned}$$

In this problem $x^* = 0$ and

$$\mathcal{L}(x,\lambda) = x^4 + \lambda x^3, \quad \mathcal{L}_x(x,\lambda) = 12x^2 + 6\lambda x, \quad \mathcal{L}_{xx}(x^*,\lambda^*) = 0.$$

Thus, the SOSC is not satisfied at the point $(0, \lambda^*)$:

$$\omega^T \mathcal{L}_{xx}(x^*,\lambda^*)\omega = 0 \leq \nu\|\omega\|^2, \quad \nu > 0.$$

Problem statement

Find a local solution x^* of NLP in which

- the strict complementarity condition,
- a constraint qualification,
- a second-order sufficient condition for optimality

are *not* necessarily satisfied at the solution.

Motivation and Goals

Motivation:

Motivation and Goals

Motivation:

- Classical optimality conditions do not cover all NLP problems

Motivation and Goals

Motivation:

- Classical optimality conditions do not cover all NLP problems
- Numerical methods are not effective for some classes of NLP problems

Motivation and Goals

Motivation:

- Classical optimality conditions do not cover all NLP problems
- Numerical methods are not effective for some classes of NLP problems
- For NLP, it is often difficult to detect and remedy poor mathematical formulations of applied problems

Motivation and Goals

Motivation:

- Classical optimality conditions do not cover all NLP problems
- Numerical methods are not effective for some classes of NLP problems
- For NLP, it is often difficult to detect and remedy poor mathematical formulations of applied problems

Goals:

Motivation and Goals

Motivation:

- Classical optimality conditions do not cover all NLP problems
- Numerical methods are not effective for some classes of NLP problems
- For NLP, it is often difficult to detect and remedy poor mathematical formulations of applied problems

Goals:

- Propose new optimality conditions for some classes of NLP problems

Motivation and Goals

Motivation:

- Classical optimality conditions do not cover all NLP problems
- Numerical methods are not effective for some classes of NLP problems
- For NLP, it is often difficult to detect and remedy poor mathematical formulations of applied problems

Goals:

- Propose new optimality conditions for some classes of NLP problems
- Design algorithms that perform as well as possible in a variety of difficult circumstances

Conversion to the Equality Case

Convert the NLP problem into the equality constrained problem:

$$\begin{array}{ll} \underset{(x,s)}{\text{minimize}} & f(x) \\ \text{subject to} & g_1(x) + s_1^2 = 0, \ldots, g_m(x) + s_m^2 = 0, \end{array}$$

where $s_1, \ldots, s_m$ are additional variables.

Define the *Lagrangian function* as

$$L(x, s, \lambda) = f(x) + \sum_{j=1}^{m} \lambda_j (g_j(x) + s_j^2).$$

Lagrange system

If x^* is a minimizer, then (x^*, s^*, λ^*) is the solution to the *Lagrange system*:

$$F(x, s, \lambda) = \begin{bmatrix} L_x(x, s, \lambda) \\ 2\lambda s \\ g(x) + s^2 \end{bmatrix} = 0.$$

Lagrange-Newton method is the Newton method applied to the Lagrange system:

$$z^{k+1} = z^k - J(z^k)^{-1} F(z^k), \quad z = (x, s, \lambda).$$

Question 3: Where is the problem?

Lagrange–Newton Method

Write the Jacobian $J = F'(x, s, \lambda)$ explicitly:

$$
J = \left[\begin{array}{c|ccc|ccc}
L_{xx} & & 0 & & & \nabla g(x) & \\
\hline
 & 2\lambda_1 & \dots & 0 & 2s_1 & \dots & 0 \\
0 & \dots & \dots & \dots & \dots & \dots & \dots \\
 & 0 & \dots & 2\lambda_m & 0 & \dots & 2s_m \\
\hline
 & 2s_1 & \dots & 0 & & & \\
\nabla g(x)^T & \dots & \dots & \dots & & 0 & \\
 & 0 & \dots & 2s_m & & &
\end{array}\right]
$$

Problem: The *matrix J is singular* at (x^*, s^*, λ^*) if one of the standard assumptions (LICQ, SCC, SOSC) is not satisfied.

Other work in the area

- Under assumptions of SOSC and of existence of a KKT point, Izmailov and Solodov (2005) and Wright (2005) proposed an approach that is based on the correct identification of the active constraint set $A(x^*)$.

Other work in the area

- Under assumptions of SOSC and of existence of a KKT point, Izmailov and Solodov (2005) and Wright (2005) proposed an approach that is based on the correct identification of the active constraint set $A(x^*)$.
- They eliminate inactive constraints and thus reduce the NLP problem to an equality constrained problem. The difference between the methods is in the approach to solving the equality constrained problem.

Other work in the area

- Under assumptions of SOSC and of existence of a KKT point, Izmailov and Solodov (2005) and Wright (2005) proposed an approach that is based on the correct identification of the active constraint set $A(x^*)$.
- They eliminate inactive constraints and thus reduce the NLP problem to an equality constrained problem. The difference between the methods is in the approach to solving the equality constrained problem.

  BUT ...

Other work in the area

- Under assumptions of SOSC and of existence of a KKT point, Izmailov and Solodov (2005) and Wright (2005) proposed an approach that is based on the correct identification of the active constraint set $A(x^*)$.
- They eliminate inactive constraints and thus reduce the NLP problem to an equality constrained problem. The difference between the methods is in the approach to solving the equality constrained problem.

  BUT ...
- the set $A(x^*)$ is hard to identify for the degenerate problems;

Other work in the area

- Under assumptions of SOSC and of existence of a KKT point, Izmailov and Solodov (2005) and Wright (2005) proposed an approach that is based on the correct identification of the active constraint set $A(x^*)$.
- They eliminate inactive constraints and thus reduce the NLP problem to an equality constrained problem. The difference between the methods is in the approach to solving the equality constrained problem.

  BUT ...
- the set $A(x^*)$ is hard to identify for the degenerate problems;
- Jacobian of the Newton system is singular in the limit;

Other work in the area

- Under assumptions of SOSC and of existence of a KKT point, Izmailov and Solodov (2005) and Wright (2005) proposed an approach that is based on the correct identification of the active constraint set $A(x^*)$.
- They eliminate inactive constraints and thus reduce the NLP problem to an equality constrained problem. The difference between the methods is in the approach to solving the equality constrained problem.

  BUT ...
- the set $A(x^*)$ is hard to identify for the degenerate problems;
- Jacobian of the Newton system is singular in the limit;
- What if the SOSC does not hold?

Reformulation of the problem

Consider the system of equations

$$F(x) = \begin{pmatrix} F_1(x) \\ \dots \\ F_n(x) \end{pmatrix} = 0, \quad x \in \mathbb{R}^n.$$

Goal: Construct a local method for solving $F(x) = 0$ with a superlinear rate of convergence under assumption that the Jacobian matrix $F'(x^*)$ may be singular at a solution x^*, i.e.,

$$\operatorname{rank} F'(x^*) = \operatorname{rank} \begin{pmatrix} \nabla F_1^T(x^*) \\ \dots \\ \nabla F_n^T(x^*) \end{pmatrix} < n.$$

Nonlinear Equations

Notation:

For a given linear operator $A : \mathbb{R}^n \to \mathbb{R}^n$,

$$\operatorname{Im} A = \{y \in \mathbb{R}^n \,|\, y = Ax \text{ for some } x \in \mathbb{R}^n\}$$

denotes the image space.

$(\operatorname{Im} F'(x^*))^\perp$ is an orthogonal complementary subspace to the image of the first derivative of F;

$P^\perp$ is a matrix of the orthoprojector onto $(\operatorname{Im} F'(x^*))^\perp$.

Assumption:

$$\operatorname{Im}(F'(x^*) + P^\perp F''(x^*)h) = \mathbb{R}^n, \quad h \in \mathbb{R}^n.$$

2-factor-method

2-factor-method is the following iterative method:

$$x^{k+1} = x^k - \left[F'(x^k) + P^{\perp} F''(x^k)[h]\right]^{-1} \left[F(x^k) + P^{\perp} F'(x^k)[h]\right],$$

Remark: 2-factor-method is an application of the Newton method to the system $\Phi(x) = F(x) + P^{\perp} F'(x)[h] = 0$.

Under our assumption, there is a neighborhood $U(x^*)$ of x^* in $\mathbb{R}^n$ such that for any $x^0 \in U(x^*)$, the 2-factor-method converges to x^* and $\|x^{k+1} - x^*\| \leq C\|x^k - x^*\|^2, \quad C > 0.$

Remark: For details about 2-factor-method see Izmailov and Tret'yakov (1999) or Tret'yakov and Marsden (2003); about singular nonlinear equations – Brezhneva and Izmailov (2002).

2-factor-Lagrange Method

Recall that we consider the following problem:

Problem: Solve the Lagrange system

$$F(x, s, \lambda) = 0$$

for which the Jacobian matrix $F'(x^*, s^*, \lambda^*)$ might be *singular*.

2-factor-Lagrange method: Apply 2-factor-method to solve the Lagrange system. First, we will construct

$$\Phi(x, s, \lambda) = F(x, s, \lambda) + P^{\perp}(F'(x, s, \lambda)h)$$

such that if (x^*, s^*, λ^*) is a solution to $\Phi(x, s, \lambda) = 0$, then x^* is a local minimizer to the NLP problem.

Assumption

Assumption A1. If x^* is a solution of the NLP then for some λ^* and $z^* = (x^*, s^*, \lambda^*)$ the following condition holds:

$$\operatorname{Im}(F'(z^*) + P^{\perp} F''(z^*)h) = \mathbb{R}^{n+2m}, \quad h \in \mathbb{R}^{n+2m}.$$

Lemma. Let x^* be a solution to the NLP problem and the necessary optimality conditions are satisfied with some Lagrange multiplier λ^*.
If the standard second-order sufficient condition (SOSC) holds at x^*, then Assumption A1 is satisfied.

Constructing Φ

Assume that estimates of both x and λ are available.

First technique: Using singular-value decomposition of the matrix $F'(x, s, \lambda)$.

Second technique: Using an estimate of the set of the linearly independent gradients $\nabla F_1(z^*), \ldots, \nabla F_r(z^*)$, $r < n$.

The following defines a computable estimate of the distance to solution set $S = \{z \in \mathbb{R}^{n+2m} \mid F(z) = 0\}$:

$$\mu(z) \approx dist(z, S), \quad \mu(z) = \|F(z)\|^{1/2}.$$

Constructing Φ

Let $\mu = C\mu^{1/2}(z)$, $C > 0$. Then

1) The vectors $\nabla F_{i_1}(z^*), \dots, \nabla F_{i_s}(z^*)$ are linearly independent iff

$$\mathrm{dist}(\nabla F_{i_k}(z), L^s_{i_k}(z)) \geq \mu \quad \forall k = 1, \dots, s, \quad \forall z \in U(z^*),$$

where $L^s_{i_k}(z)$ is the linear span, $s \geq 2$,

$$L^s_{i_k}(z) = \mathrm{span}\{\nabla F_{i_1}(z), \dots, \nabla F_{i_{k-1}}(z), \nabla F_{i_{k+1}}(z), \dots, \nabla F_{i_s}(z)\}.$$

2) $\nabla F_i(z^*) = 0$ iff

$$\|\nabla F_i(z)\| < \mu, \quad \forall z \in U(z^*).$$

Constructing Φ

Assume that the first r vectors $\nabla F_1(z^*), \ldots, \nabla F_r(z^*)$ are linearly independent, and that others are zeros, i.e, $\nabla F_{r+1}(z^*) = 0$, ..., $\nabla F_{n+2m}(z^*) = 0$. If they were not zeros, we would transform the original system into another one that has the same solution z^* and that satisfies these assumptions.

Then $P^\perp$, which is the orthoprojector onto $(\operatorname{Im} F'(z^*))^\perp$, is a diagonal matrix $P^\perp = \operatorname{diag}(p_j)_{j=1}^{n+2m}$ that is given by

$$p_j = \begin{cases} 0, & j = 1, \ldots, r \\ 1, & j = r+1, \ldots, n+2m \end{cases}.$$

Constructing Φ

We get the mapping $\Phi(z)$:

$$\Phi(z) = F(z) + P^{\perp} F'(z)[h],$$

where the choice of the vector $h \in \mathbb{R}^{n+2m}$ is flexible and should be done in such a way that

$$\mathrm{rank}(\Phi'(z^*)) = n + 2m.$$

Having the mapping $\Phi(z)$ and a point z^0, we can define the 2-factor-Lagrange method:

$$z^{k+1} = z^k - \left[\Phi'(z^k)\right]^{-1} \left[\Phi(z^k)\right], \quad k = 0, 1, 2, \ldots$$

Theorem

Let x^* be a solution of NLP. Let $f, g \in C^3(\mathbb{R}^n)$ and let $z^* = (x^*, s^*, \lambda^*)$ satisfy the necessary optimality conditions:

$$F(x, s, \lambda) = \nabla L(x, s, \lambda) = 0.$$

Assume that Assumption A1 holds:

$$\mathrm{Im}(F'(z^*) + P^{\perp} F''(z^*)h) = \mathbb{R}^{n+2m}, \quad h \in \mathbb{R}^{n+2m}.$$

Then there is a neighborhood $V(x^*)$ of z^* in $\mathbb{R}^{n+2m}$ such that for any $z^0 \in V(z^*)$, the 2-factor-Lagrange-method converges to z^* and

$$\|z^{k+1} - z^*\| \leq C\|z^k - z^*\|^2, \quad C > 0.$$

Example

Consider the problem of minimizing the functional

$$J_0[y] = \int_{-3/2}^{3/2} y^2 dx$$

subject to the constraints

$$\int_{-3/2}^{3/2} ((y')^2 - y^2)dx \leq 0, \quad y(-3/2) = y(3/2) = 0.$$

Example

To discretize the problem, introduce the notation:

$$x_1 = y(-1/2), \quad x_2 = y(1/2),$$

and use the trapezoidal rule to approximate the integrals.

We get the following problem:

$$\begin{array}{ll} \underset{x \in \mathbb{R}^2}{\text{minimize}} & f(x_1, x_2) = x_1^2 + x_2^2 \\ \text{subject to} & g(x_1, x_2) = 3x_1^2 + x_2^2 - 4x_1 x_2 \leq 0 \end{array}$$

The solution to this problem is $x^* = (0,0)^T$.

Example

Reduce the problem to the equality constrained problem

$$\begin{array}{ll} \text{minimize} & x_1^2 + x_2^2 \\ \text{subject to} & 3x_1^2 + x_2^2 - 4x_1x_2 + s^2 = 0 \end{array}$$

The Lagrangian is

$$L(x_1, x_2, s, \lambda) = x_1^2 + x_2^2 + \lambda(3x_1^2 + x_2^2 - 4x_1x_2 + s^2).$$

The first-order conditions can be stated as

$$F(x, s, \lambda) = \begin{bmatrix} 2x_1 + 6\lambda x_1 - 4\lambda x_2 \\ 2x_2 + 2\lambda x_2 - 4\lambda x_1 \\ 2\lambda s \\ 3x_1^2 + x_2^2 - 4x_1x_2 + s^2 \end{bmatrix} = 0.$$

Example

The Jacobian of F is defined as

$$F'(x, s, \lambda) = \begin{bmatrix} 2+6\lambda & -4\lambda & 0 & 6x_1 - 4x_2 \\ -4\lambda & 2+2\lambda & 0 & 2x_2 - 4x_1 \\ 0 & 0 & 2\lambda & 2s \\ 6x_1 - 4x_2 & 2x_2 - 4x_1 & 2s & 0 \end{bmatrix}.$$

This matrix is singular at $x^* = (0,\ 0)^T$ with $\lambda^* = 0$ and $s^* = 0$. We apply the described method with $h = (0, 0, 1, 1)^T$ to get

$$\Phi(z) = F(z) + P^{\perp} F'(z)[h]$$

Example

The idea behind the algorithm is that instead of solving Lagrange system, which has the singular Jacobian, we are solving the following system:

$$\Phi(x, s, \lambda) = \begin{bmatrix} 2x_1 + 6\lambda x_1 - 4\lambda x_2 \\ 2x_2 + 2\lambda x_2 - 4\lambda x_1 \\ 2\lambda s + 2\lambda + 2s \\ 3x_1^2 + x_2^2 - 4x_1 x_2 + s^2 + 2s \end{bmatrix} = 0.$$

with the nonsingular Jacobian.
The system $\Phi = 0$ has the same solution $(0, 0, 0, 0)^T$ as the original system.

Example

At each iteration, the step $(\Delta x, \Delta s, \Delta\lambda)$ is a solution of the following system

$$\begin{bmatrix} 2+6\lambda & -4\lambda & 0 & 6x_1-4x_2 \\ -4\lambda & 2+2\lambda & 0 & 2x_2-4x_1 \\ 0 & 0 & 2\lambda+2 & 2s+2 \\ 6x_1-4x_2 & 2x_2-4x_1 & 2s+2 & 0 \end{bmatrix} \begin{bmatrix} \Delta x_1 \\ \Delta x_2 \\ \Delta s \\ \Delta\lambda \end{bmatrix}$$

$$= - \begin{bmatrix} 2x_1+6\lambda x_1-4\lambda x_2 \\ 2x_2+2\lambda x_2-4\lambda x_1 \\ 2\lambda s+2\lambda+2s \\ 3x_1^2+x_2^2-4x_1x_2+s^2+2s \end{bmatrix}.$$

NCP

Given a (smooth) mapping $g(x) : \mathbb{R}^n \rightarrow \mathbb{R}^n$. Find $x \in \mathbb{R}^n$ such that

$$g(x) \leq 0, \quad x \leq 0, \quad x^T g(x) = 0.$$

A. Economic Applications:

- General equilibrium problems
- Invariant capital stock problems
- Spatial price equilibrium problems

B. Physical (Engineering) Applications:

- Contact mechanics problems
- Obstacle (and free boundary) problems
- Traffic equilibrium problems

Method for Solving NCP

Convert the NCP problem into

$$F(x,s,y) = \begin{pmatrix} g(x) + s^2 \\ x + y^2 \\ sy \end{pmatrix} = 0.$$

The corresponding Jacobian is given by

$$F'(x,s,y) = \begin{pmatrix} \nabla g(x) & 2S & 0 \\ I & 0 & 2Y \\ 0 & Y & S \end{pmatrix},$$

where $S = \mathrm{diag}(s_j)_{j=1}^m$, $Y = \mathrm{diag}(y_j)_{j=1}^m$, and I is the identity matrix.

Method for Solving NCP

Introduce the set $J_0 = \{j = 1, \ldots, m \,|\, y_j^* = 0,\ s_j^* = 0\}$.
Define the vector $h = (h_1, \ldots, h_n)$ as

$$h_i = \begin{cases} 1, & i \in J_0 \\ 0, & i \notin J_0 \end{cases}$$

and a vector $\bar{h} \in \mathbb{R}^{3n}$ as $\bar{h} = (0, 0, h)$.
In NCP, the orthoprojector $P^\perp$ onto $(\operatorname{Im} F'(x, s, y))^\perp$ is a constant diagonal matrix $P^\perp = \operatorname{diag}(p_j)_{j=1}^{3n}$:

$$p_i = \begin{cases} 1, & i = n + j, \quad \text{where} \quad j \in J_0 \\ 1, & i = 2n + j, \quad \text{where} \quad j \in J_0 \\ 0, & \text{otherwise.} \end{cases}$$

Method for Solving NCP

Assume for simplicity that $J_0 = \{1, \ldots, n\}$. Then we get

$$F(x,s,y) + P^{\perp} F'(x,s,y)\bar{h} = \begin{pmatrix} g + s^2 \\ x + y^2 + 2y \\ sy + s \end{pmatrix}$$

and

$$\begin{pmatrix} x^{k+1} \\ s^{k+1} \\ y^{k+1} \end{pmatrix} = \begin{pmatrix} x \\ s \\ y \end{pmatrix} - \begin{bmatrix} (\nabla g(x))^T & 2S & 0 \\ I & 0 & 2(Y+I) \\ 0 & Y+I & S \end{bmatrix}^{-1}$$

$$\begin{pmatrix} g(x) + s^2 \\ x + y^2 + 2y \\ sy + s \end{pmatrix}.$$

Conclusion

- By introduction of additional (slack) variables, we have converted an optimization problem with inequality constraints into an equality constrained problem.

Conclusion

- By introduction of additional (slack) variables, we have converted an optimization problem with inequality constraints into an equality constrained problem.
- We proposed a new method for solving the Lagrange optimality system whose Jacobian might be singular at the solution.

Conclusion

- By introduction of additional (slack) variables, we have converted an optimization problem with inequality constraints into an equality constrained problem.
- We proposed a new method for solving the Lagrange optimality system whose Jacobian might be singular at the solution.
- We illustrated the proposed approach by constructing a method for nonlinear complementarity problems (NCP).

References

[1] A. F. Izmailov and M. V. Solodov (2005), *Newton–type methods for optimization problems without constraint qualifications*, SIAM Journal on Optimization, 15, pp. 210-228.

[2] S. J. Wright (2005), *An algorithm for degenerate nonlinear programming with rapid local convergence*, SIAM Journal on Optimization, 15, pp. 673-696.

References

[1] A. F. Izmailov and M. V. Solodov (2005), *Newton–type methods for optimization problems without constraint qualifications*, SIAM Journal on Optimization, 15, pp. 210-228.

[2] S. J. Wright (2005), *An algorithm for degenerate nonlinear programming with rapid local convergence*, SIAM Journal on Optimization, 15, pp. 673-696.

[3] A. F. Izmailov and A. A. Tret'yakov (1999), *2–regular solutions of nonregular problems*, Fizmatlit, Moscow (in Russian).

[4] A. A. Tret'yakov and J. E. Marsden (2003), *Factor-analysis of nonlinear mappings: p-regularity theory*, Communications on Pure and Applied Analysis, 2, pp. 425-445.

References

- More general approach to solving singular nonlinear equations is described in
  [5] O. Brezhneva and A. Izmailov (2002), *Construction of defining systems for finding singular solutions to nonlinear equations*, Comput. Math. and Math. Phys., 42, pp. 8–19.

References

- More general approach to solving singular nonlinear equations is described in
  [5] O. Brezhneva and A. Izmailov (2002), *Construction of defining systems for finding singular solutions to nonlinear equations*, Comput. Math. and Math. Phys., 42, pp. 8–19.
- This talk presents the paper:
  [6] O. Brezhneva and A. Tret'yakov, P*-factor-Lagrange methods for degenerate nonlinear programming*, in preparation.

End...

Thank you!