Faculty of Science

Numerical Methods for Linear Complementarity Problems in Physics-based Animation

Kenny Erleben
kenny@diku.dk

Department of Computer Science
University of Copenhagen

2013

The Definition

The Complementarity Problem

If given $x, y \in \mathbb{R}$ where

$$x \geq 0$$
$$y \geq 0$$

and

$$x > 0 \Rightarrow y = 0$$
$$y > 0 \Rightarrow x = 0$$

Then we have a complementarity problem. If for some $a, b \in \mathbb{R}$

$$y = ax + b$$

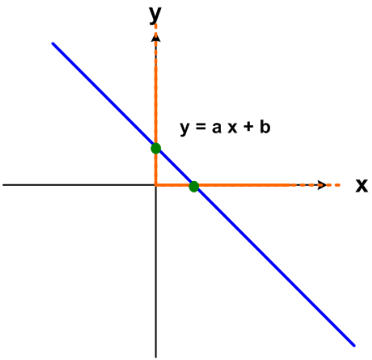
We have a Linear Complementarity Problem (LCP).

The Linear Complementarity Problem

More compact notation

$$x \geq 0$$
$$ax + b \geq 0$$
$$x(ax + b) = 0$$

The Geometry



How many #Solutions of LCP?

Hint: Try to examine signs of a and b

	$b < 0$	$b = 0$	$b > 0$
$a < 0$			
$a = 0$			
$a > 0$			

Answer of # Solutions

Verify this using geometry

	$b < 0$	$b = 0$	$b > 0$
$a < 0$	0	1	2
$a = 0$	0	∞	1
$a > 0$	1	1	1

Going to Higher Dimensions

Let $\mathbf{b}, \mathbf{x} \in \mathbb{R}^n$ and $\mathbf{A} \in \mathbb{R}^{n \times n}$ so $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$,

$$\begin{aligned} \mathbf{x}_i &\geq 0 \quad \forall i \in [1..n] \\ (\mathbf{A}\mathbf{x} + \mathbf{b})_i &\geq 0 \quad \forall i \in [1..n] \\ \mathbf{x}_i(\mathbf{A}\mathbf{x} + \mathbf{b})_i &= 0 \quad \forall i \in [1..n] \end{aligned}$$

In Matrix-Vector Notation

$$\begin{aligned} \mathbf{x} &\geq 0 \\ (\mathbf{A}\mathbf{x} + \mathbf{b}) &\geq 0 \\ \mathbf{x}^T(\mathbf{A}\mathbf{x} + \mathbf{b}) &= 0 \end{aligned}$$

Or compactly

$$\mathbf{0} \leq \mathbf{A}\mathbf{x} + \mathbf{b} \quad \perp \quad \mathbf{x} \geq \mathbf{0}$$

This talk is about how we can solve this type of problem.

What kind of problem is a LCP formulation?

What do you think?

- A constrained minimization problem?
- A nonlinear equation (root search problem) ?
- A fixed-point problem?
- A combinatorial problem?
- Something else?

LCP for water animation

Separating Solid Wall Boundary Conditions

If fluid separates at a solid wall then

$$\mathbf{n} \cdot \mathbf{u} > 0$$

where $\mathbf{n}$ is unit outward surface normal of solid wall boundary. However, this means we have a free surface and $p = 0$. If we do not have a free surface then $p > 0$ and then we have

$$\mathbf{n} \cdot \mathbf{u} = 0$$

By Gauss divergence theorem

$$\int_V \nabla \cdot \mathbf{u} dV = \int_S \mathbf{n} \cdot \mathbf{u} dS$$

In summary we find the complementarity condition

$$0 \leq \nabla \cdot \mathbf{u} \quad \perp \quad p \geq 0$$

Time Discretization

From Navier-Stokes equations

$$\frac{\partial \mathbf{u}}{\partial t} = -\frac{1}{\rho}\nabla p$$

Using first-order explicit Euler

$$\mathbf{u}^{t+\Delta t} = \mathbf{u}^{t} - \frac{\Delta t}{\rho}\nabla p$$

Inserting into Continuity equation $\nabla \cdot \mathbf{u} = 0$

$$\nabla \cdot \mathbf{u}^{t+\Delta t} = \nabla \cdot \mathbf{u}^{t} - \nabla \cdot \left(\frac{\Delta t}{\rho}\right)\nabla p = 0$$

Hence

$$\underbrace{\frac{\Delta t}{\rho}\nabla^2}_{\mathbf{A}} \underbrace{p}_{\mathbf{x}} - \underbrace{\nabla \cdot \mathbf{u}^{t}}_{-\mathbf{b}} = \mathbf{0}$$

We have a LCP

Inserting $\nabla \cdot \mathbf{u}^{t+\Delta t} = \mathbf{A}\mathbf{x} + \mathbf{b}$ into the complementarity condition

$$0 \leq \nabla \cdot \mathbf{u}^{t+1} \quad \perp \quad p \geq 0$$

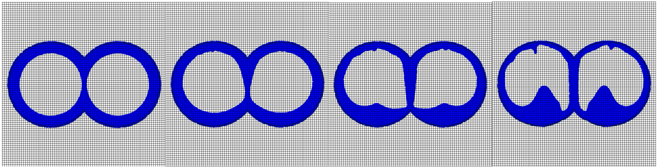
gives

$$\mathbf{0} \leq \mathbf{A}\mathbf{x} + \mathbf{b} \quad \perp \quad \mathbf{x} \geq \mathbf{0}$$

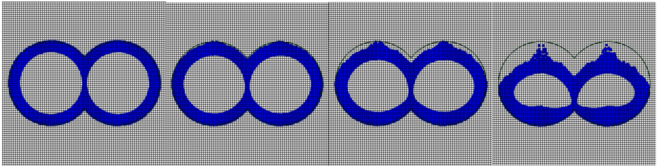
This is a LCP.

Visual Comparison of Boundary Conditions

Free slip solid wall boundary conditions



Separating solid wall boundary conditions



LCP for rigid body contact force computations

The Position Level Non-Penetrating Constraint

Consider two rigid bodies at a single point of contact

- Let d be the minimum distance
- Penetration is not allowed hence $d \geq 0$
- A normal force λ_n prevents penetration and it is non-sticking $\lambda_n \geq 0$
- If the rigid bodies are separated $d > 0$ then there can be no normal force $\lambda_n = 0$ between the two bodies.
- If there is a normal force $\lambda_n > 0$ then the rigid bodies must be touching $d = 0$.

We write

$$\lambda_n > 0 \Rightarrow d = 0$$
$$d > 0 \Rightarrow \lambda_n = 0$$

This is known as a complementarity condition.

The Velocity Level Non-Penetrating Constraint

Assume we have touching contact $d = 0$ and let $\mathbf{v} = \frac{d}{dt}\mathbf{d}$ be the time derivative of the minimum distance vector. Let $v_n = \mathbf{n} \cdot \mathbf{v}$ be the projection onto the normal direction at point of contact.

- if the contact is about to separate then $v_n > 0$
- If we have sustained contact then $v_n = 0$
- If $v_n > 0$ then at any future time we have $d > 0$ due to continuity of physics this implies $\lambda_n = 0$
- If $\lambda_n > 0$ then the contact can not be separating and we must have sustained contact $v_n = 0$

We write

$$\lambda_n > 0 \Rightarrow v_n = 0$$
$$v_n > 0 \Rightarrow \lambda_n = 0$$

Again a complementarity condition.

The Contact Frame

Let the contact normal be $\mathbf{n}$ and the orthogonal contact plane vector $\mathbf{t}$. We assume

$$\begin{aligned} \| \mathbf{n} \| &= 1 \\ \| \mathbf{t} \| &= 1 \\ \mathbf{n} \cdot \mathbf{t} &= 0 \end{aligned}$$

Define the contact plane velocities as

$$\begin{aligned} v_1 &= \mathbf{t} \cdot \mathbf{v} \\ v_2 &= -\mathbf{t} \cdot \mathbf{v} \end{aligned}$$

and $v_n = \mathbf{n} \cdot \mathbf{v}$.

The Friction Force

- The friction force $\mathbf{f}_t$ is given by

$$\mathbf{f}_t = \lambda_1 \mathbf{t} - \lambda_2 \mathbf{t}$$

where

$$\lambda_1 > 0 \Rightarrow \lambda_2 = 0$$
$$\lambda_2 > 0 \Rightarrow \lambda_1 = 0$$

The λ's are the "components" of the friction force projected onto the positive span defined by $\mathbf{t}$ and $-\mathbf{t}$.

Coulomb Friction in a 2D World

- If there is a normal force λ_n then the friction force $\mathbf{f}_t$ is bounded by the cone

$$\| \mathbf{f}_t \| = \lambda_1 + \lambda_2 \leq \mu \lambda_n$$

  where $\mu > 0$ is a positive constant known as the coefficient of friction.
- If there is sliding v_1 or $v_2 \neq \mathbf{0}$ then the friction force $\mathbf{f}_t$ works against the sliding direction and attains it maximum possible value.
- If there is no sliding then the friction force can have any value bounded by the friction cone.

Coulomb Friction in a 2D World Continued

Let us introduce the scalar $\beta \geq 0$ then we may write

$$\begin{aligned} \lambda_1 > 0 &\Rightarrow (\beta + v_1) = 0 \\ \lambda_2 > 0 &\Rightarrow (\beta + v_2) = 0 \\ (\beta + v_1) > 0 &\Rightarrow \lambda_1 = 0 \\ (\beta + v_2) > 0 &\Rightarrow \lambda_2 = 0 \end{aligned}$$

That is

$$0 \leq (\beta + v_i) \quad \perp \quad \lambda_i \geq 0 \quad \forall i$$

If there is sliding (a λ is positive) then β estimates the magnitude of the sliding speed

Coulomb Friction in a 2D World Continued

Now we combine knowledge of β with Coulomb friction cone model

$$\beta > 0 \Rightarrow (\mu\lambda_n - \lambda_1 - \lambda_2) = 0$$
$$(\mu\lambda_n - \lambda_1 - \lambda_2) > 0 \Rightarrow \beta = 0$$

Again we recover complementarity conditions.

$$0 \leq \beta \quad \perp \quad (\mu\lambda_n - \lambda_1 - \lambda_2) \geq 0$$

The role of β is not only to measure if there is sliding but also to pick the direction of the friction force.

How does β really work?

A 3D world example, let the contact frame be spanned by

$$
\begin{aligned}
\mathbf{n} &= \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T \\
\mathbf{t}_1 &= \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T \\
\mathbf{t}_2 &= \begin{bmatrix} 0 & 1 & 0 \end{bmatrix}^T \\
\mathbf{t}_3 &= \begin{bmatrix} -1 & 0 & 0 \end{bmatrix}^T \\
\mathbf{t}_4 &= \begin{bmatrix} 0 & -1 & 0 \end{bmatrix}^T
\end{aligned}
$$

Without loss of generality we make a random pick

$$
\mathbf{v} = \begin{bmatrix} 5 \\ 1 \\ 0 \end{bmatrix}
$$

How does β really work?

Now

$$\begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ v_4 \end{bmatrix} = \begin{bmatrix} \mathbf{v} \cdot \mathbf{t}_1 \\ \mathbf{v} \cdot \mathbf{t}_2 \\ \mathbf{v} \cdot \mathbf{t}_3 \\ \mathbf{v} \cdot \mathbf{t}_4 \end{bmatrix} = \begin{bmatrix} 5 \\ 1 \\ -5 \\ -1 \end{bmatrix}$$

Next consider the non-negative constraints (remember $\beta \geq 0$ must hold)

$$\begin{bmatrix} \beta + v_1 \\ \beta + v_2 \\ \beta + v_3 \\ \beta + v_4 \end{bmatrix} = \begin{bmatrix} \beta + 5 \\ \beta + 1 \\ \beta + (-5) \\ \beta + (-1) \end{bmatrix} \geq 0$$

Hence we must have $\beta \geq 5$ for these constraints to be true.

How does β really work?

For $\beta = 5$ we have

$$
\begin{aligned}
0 &< (\beta + v_1) & \perp & \quad \lambda_1 = 0 \\
0 &< (\beta + v_2) & \perp & \quad \lambda_2 = 0 \\
0 &= (\beta + v_3) & \perp & \quad \lambda_3 > 0 \\
0 &< (\beta + v_4) & \perp & \quad \lambda_4 = 0
\end{aligned}
$$

The size of λ_3 is then determined from

$$0 \leq \beta \quad \perp \quad (\mu\lambda_n - \lambda_1 - \lambda_2 - \lambda_3 - \lambda_4) \geq 0$$

Since $\beta > 0$ we have

$$0 = (\mu\lambda_n - \lambda_1 - \lambda_2 - \lambda_3 - \lambda_4) = \mu\lambda_n - \lambda_3$$

Hence the solution is $\lambda_3 = \mu\lambda_n$

How does β really work?

What if we choose $\beta > 5$ then

$$\begin{aligned} 0 < (\beta + v_1) &\quad \perp \quad \lambda_1 = 0 \\ 0 < (\beta + v_2) &\quad \perp \quad \lambda_2 = 0 \\ 0 < (\beta + v_3) &\quad \perp \quad \lambda_3 = 0 \\ 0 < (\beta + v_4) &\quad \perp \quad \lambda_4 = 0 \end{aligned}$$

and since $\beta > 0$ we get

$$0 = (\mu\lambda_n - \lambda_1 - \lambda_2 - \lambda_3 - \lambda_4) = \mu\lambda_n$$

This is only possible if $\lambda_n = 0$ that we have a separating contact. The insight

- If $\lambda_n > 0$ then there is unique solution for β
- If $\lambda_n = 0$ then there are infinitely many solutions for β

Study Group Work

- Assume $\lambda_n > 0$ then by case-by-case analysis show that $\beta > 5$ is not possible
- Replace **v** with $\begin{bmatrix} -35 & 15 & 0 \end{bmatrix}^T$, what is β? which λ_i's are positive?
- Replace **v** with $\begin{bmatrix} -35 & -35 & 0 \end{bmatrix}^T$, what is β? which λ_i's are positive?
- Trick question: What happens if $\beta = 0$? (Hint remember that there is complementarity conditions between the λ_i's. If one direction is positive then the opposite direction must be zero. With this knowledge only two things can happen model-wise).
- More trick question: replace **v** with $\begin{bmatrix} -35 & 15 & 0.01 \end{bmatrix}^T$ what happens now?

Time Discretization – The Contact LCP Model

Skipping details and cheating with notation we have from time-discretization

$$\begin{bmatrix} v_n \\ v_1 \\ v_2 \end{bmatrix} = \begin{bmatrix} B_{nn} & B_{n1} & B_{n2} \\ B_{1n} & B_{11} & B_{12} \\ B_{2n} & B_{21} & B_{22} \end{bmatrix} \begin{bmatrix} \lambda_n \\ \lambda_1 \\ \lambda_2 \end{bmatrix} + \begin{bmatrix} b_n \\ b_1 \\ b_2 \end{bmatrix}$$

Stating that final contact velocities at the end of a time-step are an affine function of the contact impulse. Using this we have

$$\mathbf{0} \leq \underbrace{\begin{bmatrix} B_{nn} & B_{n1} & B_{n2} & 0 \\ B_{1n} & B_{11} & B_{12} & 1 \\ B_{2n} & B_{21} & B_{22} & 1 \\ \mu & -1 & -1 & 0 \end{bmatrix}}_{\mathbf{A}} \underbrace{\begin{bmatrix} \lambda_n \\ \lambda_1 \\ \lambda_2 \\ \beta \end{bmatrix}}_{\mathbf{x}} + \underbrace{\begin{bmatrix} b_n \\ b_1 \\ b_2 \\ 0 \end{bmatrix}}_{\mathbf{b}} \quad \perp \quad \underbrace{\begin{bmatrix} \lambda_n \\ \lambda_1 \\ \lambda_2 \\ \beta \end{bmatrix}}_{\mathbf{x}} \geq \mathbf{0}$$

A LCP

A NAÏVE Active Set/Pivoting Method

Guessing A Solution

Given the index set $\mathcal{I} = \{1, \dots, n\}$ define

$$\mathcal{F} = \{i \quad | \quad i \in \mathcal{I} \quad \wedge \quad \mathbf{y}_i > 0\}$$
$$\mathcal{A} = \{i \quad | \quad i \notin \mathcal{F} \quad \wedge \quad \mathbf{y}_i = 0\}$$

Make "lucky" guess of $\mathcal{F}$ and $\mathcal{A}$,

$$\begin{bmatrix} \mathbf{y}_\mathcal{A} \\ \mathbf{y}_\mathcal{F} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\mathcal{A}\mathcal{A}} & \mathbf{A}_{\mathcal{A}\mathcal{F}} \\ \mathbf{A}_{\mathcal{F}\mathcal{A}} & \mathbf{A}_{\mathcal{F}\mathcal{F}} \end{bmatrix} \begin{bmatrix} \mathbf{x}_\mathcal{A} \\ \mathbf{x}_\mathcal{F} \end{bmatrix} + \begin{bmatrix} \mathbf{b}_\mathcal{A} \\ \mathbf{b}_\mathcal{F} \end{bmatrix}$$

By assumption $\mathbf{y}_\mathcal{F} > 0 \Rightarrow \mathbf{x}_\mathcal{F} = 0$

$$\begin{bmatrix} 0 \\ \mathbf{y}_\mathcal{F} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\mathcal{A}\mathcal{A}} & \mathbf{A}_{\mathcal{A}\mathcal{F}} \\ \mathbf{A}_{\mathcal{F}\mathcal{A}} & \mathbf{A}_{\mathcal{F}\mathcal{F}} \end{bmatrix} \begin{bmatrix} \mathbf{x}_\mathcal{A} \\ 0 \end{bmatrix} + \begin{bmatrix} \mathbf{b}_\mathcal{A} \\ \mathbf{b}_\mathcal{F} \end{bmatrix}$$

Verify if Guess was a Solution

So

$$\begin{bmatrix} 0 \\ \mathbf{y}_{\mathcal{F}} \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{\mathcal{A}\mathcal{A}}\mathbf{x}_{\mathcal{A}} + \mathbf{b}_{\mathcal{A}} \\ \mathbf{A}_{\mathcal{F}\mathcal{A}}\mathbf{x}_{\mathcal{A}} + \mathbf{b}_{\mathcal{F}} \end{bmatrix}$$

Compute

$$\mathbf{x}_{\mathcal{A}} = -\mathbf{A}_{\mathcal{A}\mathcal{A}}^{-1}\mathbf{b}_{\mathcal{A}}$$

Verify

$$\mathbf{x}_{\mathcal{A}} \geq 0$$

Compute

$$\mathbf{y}_{\mathcal{F}} = \mathbf{A}_{\mathcal{F}\mathcal{A}}\mathbf{x}_{\mathcal{A}} + \mathbf{b}_{\mathcal{F}}$$

Verify

$$\mathbf{y}_{\mathcal{F}} > 0$$

How Many Guesses?

We only need

$$\mathbf{A}^{-1}_{\mathcal{A}\mathcal{A}}$$

Hopefully

$$\| \mathcal{A} \| \ll n$$

Cool this will be fast!

How many guesses do we need?

Answer: Non-Polynomial Complexity

Worst case time complexity of guessing

$$\mathcal{O}(n^3 2^n)$$

Not computational very efficient!

Connection to Positive Cones (Linear Programming)

First some algebra on $\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$,

$$\mathbf{I}\mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b}$$

$$\begin{bmatrix} \mathbf{I} & -\mathbf{A} \end{bmatrix} \begin{bmatrix} \mathbf{y} \\ \mathbf{x} \end{bmatrix} = \mathbf{b}$$

Make guesses $\mathcal{F} = \{i | \mathbf{y}_i \geq 0\}$ and $\mathcal{A} = \{i | \mathbf{x}_i \geq 0\}$ so $\mathcal{F} \cap \mathcal{A} = \emptyset$ and $\mathcal{F} \cup \mathcal{A} = \{1, \ldots, n\}$

$$\underbrace{\begin{bmatrix} \mathbf{I}_{\cdot\mathcal{F}} & -\mathbf{A}_{\cdot\mathcal{A}} \end{bmatrix}}_{\mathbf{M}} \underbrace{\begin{bmatrix} \mathbf{y}_{\mathcal{F}} \\ \mathbf{x}_{\mathcal{A}} \end{bmatrix}}_{\mathbf{z}} = \mathbf{b}$$

Verify if LP

$$\mathbf{M}\mathbf{z} = \mathbf{b} \quad \text{subject to} \quad \mathbf{z} \geq 0$$

Has a solution (same as $\mathbf{b}$ in positive cone of $\mathbf{M}$).

The Projected Gauss–Seidel Method

Use a Splitting Method

Use the splitting

$$\mathbf{A} = \mathbf{M} - \mathbf{N}$$

then

$$\begin{aligned} \mathbf{Mx} - \mathbf{Nx} + \mathbf{b} &\geq 0 \\ \mathbf{x} &\geq 0 \\ (\mathbf{x})^T(\mathbf{Mx} - \mathbf{Nx} + \mathbf{b}) &= 0 \end{aligned}$$

Introduce the iteration indices

Assume $\mathbf{x}^k \to \mathbf{x}^{k+1}$ for $k \to \infty$ then

$$
\begin{aligned}
\mathbf{M}\mathbf{x}^{k+1} - \mathbf{N}\mathbf{x}^k + \mathbf{b} &\geq 0 \\
\mathbf{x}^{k+1} &\geq 0 \\
(\mathbf{x}^{k+1})^T(\mathbf{M}\mathbf{x}^{k+1} - \mathbf{N}\mathbf{x}^k + \mathbf{b}) &= 0
\end{aligned}
$$

Use Discretization $\Rightarrow$ Fixed Point Formulation

Thus, we have created a sequence of sub problems

$$\begin{aligned} \mathbf{M}\mathbf{x}^{k+1} + \mathbf{c}^k &\geq 0 \\ \mathbf{x}^{k+1} &\geq 0 \\ (\mathbf{x}^{k+1})^T(\mathbf{M}\mathbf{x}^{k+1} + \mathbf{c}^k) &= 0 \end{aligned}$$

where

$$\mathbf{c}^k = \mathbf{b} - \mathbf{N}\mathbf{x}^k$$

This is a fixed point problem.

Use Minimum Map Reformulation

Given sub problem

$$\begin{aligned} \mathbf{M}\mathbf{x}^{k+1} + \mathbf{c}^k &\geq 0 \\ \mathbf{x}^{k+1} &\geq 0 \\ (\mathbf{x}^{k+1})^T(\mathbf{M}\mathbf{x}^{k+1} + \mathbf{c}^k) &= 0 \end{aligned}$$

Same as (Why?)

$$\underbrace{\min(\mathbf{x}^{k+1}, \mathbf{M}\mathbf{x}^{k+1} + \mathbf{c}^k)}_{\mathbf{H}(\mathbf{x}^{k+1})} = 0$$

A root search problem: $\mathbf{H}(\mathbf{x}^{k+1}) = 0$.

The Minimum Map Formulation

Say $a, b \in \mathbb{R}$ are complementary

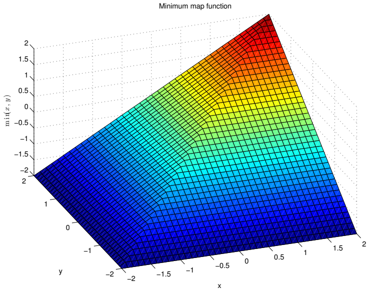
$$a > 0 \Rightarrow b = 0$$
$$b > 0 \Rightarrow a = 0$$

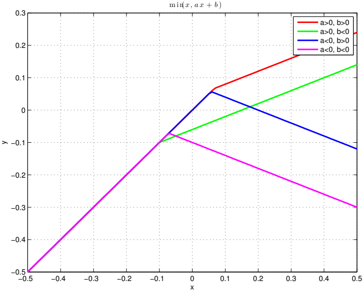
Let us look at the minimum map

$\min(a, b)$	$a > 0$	$a = 0$	$a < 0$
$b > 0$	$+$	0	$-$
$b = 0$	0	0	$-$
$b < 0$	$-$	$-$	$-$

Same solutions as complementarity problem.

Visualization of 1D Problem





$$0 \leq y = ax + b \quad \perp \quad x \geq 0$$

More Clever Manipulation

So

$$\min(\mathbf{x}^{k+1}, \mathbf{M}\mathbf{x}^{k+1} + \mathbf{c}^k) = 0$$

Subtract $\mathbf{x}^{k+1}$

$$\min(0, \mathbf{M}\mathbf{x}^{k+1} + \mathbf{c}^k - \mathbf{x}^{k+1}) = -\mathbf{x}^{k+1}$$

Multiply by minus one

$$\underbrace{\max(0, -\mathbf{M}\mathbf{x}^{k+1} - \mathbf{c}^k + \mathbf{x}^{k+1})}_{\mathbf{F}(\mathbf{x}^{k+1})} = \mathbf{x}^{k+1}$$

A fixed point formulation: $\mathbf{F}(\mathbf{x}^{k+1}) = \mathbf{x}^{k+1}$.

Do A Case-by-Case Analysis

So

$$\max(0, -\mathbf{M}\mathbf{x}^{k+1} - \mathbf{c}^k + \mathbf{x}^{k+1}) = \mathbf{x}^{k+1}$$

If

$$(-\mathbf{M}\mathbf{x}^{k+1} - \mathbf{c}^k + \mathbf{x}^{k+1})_i \leq 0$$

Then

$$\mathbf{x}_i^{k+1} = 0$$

Else

$$(-\mathbf{M}\mathbf{x}^{k+1} - \mathbf{c}^k + \mathbf{x}^{k+1})_i > 0$$

and

$$(\mathbf{x}^{k+1} - \mathbf{M}\mathbf{x}^{k+1} - \mathbf{c}^k)_i = \mathbf{x}_i^{k+1}$$

That is

$$(\mathbf{M}\mathbf{x}^{k+1})_i = -c_i^k$$

Putting it Together

For suitable choice of **M**

$$(\mathbf{M}\mathbf{x}^{k+1})_i = -c_i^k \Rightarrow \mathbf{x}_i^{k+1} = (-\mathbf{M}^{-1}\mathbf{c}^k)_i$$

Back-substitution of $\mathbf{c}^k = \mathbf{b} - \mathbf{N}\mathbf{x}^k$ we have

$$\left(\mathbf{M}^{-1}\left(\mathbf{N}\mathbf{x}^k - \mathbf{b}\right)\right)_i = \mathbf{x}_i^{k+1}$$

Insert in fixed point formulation

$$\underbrace{\max\left(0, \left(\mathbf{M}^{-1}\left(\mathbf{N}\mathbf{x}^k - \mathbf{b}\right)\right)\right)}_{\mathbf{G}(\mathbf{x}^k)} = \mathbf{x}^{k+1}$$

Closed form solution for sub problem: $\mathbf{x}^{k+1} \leftarrow \mathbf{G}(\mathbf{x}^k)$.

Final Iterative Scheme – The Projected Gauss–Seidel (PGS) method

Given $\mathbf{x}^1$ set $k = 1$

Step 1 Compute

$$\mathbf{z}^k = \left(\mathbf{M}^{-1}\left(\mathbf{N}\mathbf{x}^k - \mathbf{b}\right)\right)$$

Step 2 Compute

$$\mathbf{x}^{k+1} = \max(0, \mathbf{z}^k)$$

Step 3 If convergene then return $\mathbf{x}^{k+1}$ otherwise goto Step 1

Study Group Work

Derive an algebraic equation of the PGS method for the i^{th} component only. That is rewrite

$$\mathbf{x}_i^{k+1} = \max(0, \mathbf{z}_i^k)$$

using $\mathbf{r} = \mathbf{A}\mathbf{x}^k + \mathbf{b}$ and letting

$$\mathbf{M} = \mathbf{diag}(\mathbf{diag}(\mathbf{A})) + \mathbf{tril}(\mathbf{A}, -1)$$
$$\mathbf{N} = -\mathbf{triu}(\mathbf{A}, 1)$$

Here we used Matlab like notation. The final result should only include the terms $\mathbf{r}_i$, $\mathbf{A}_{ii}$ and $\mathbf{x}_i$.

The Projected Gauss–Seidel Method...

AGAIN!!!

Connection to Quadratic Programming (QP) Problems

Consider the minimization problem

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \geq 0} \frac{1}{2}\mathbf{x}^T \mathbf{A}\mathbf{x} + \mathbf{b}^T \mathbf{x}$$

where $\mathbf{A}$ is symmetric. First order optimality (KKT) conditions

$$\begin{aligned} \mathbf{A}\mathbf{x} + \mathbf{b} - \mathbf{I}\mathbf{y} &= 0 \\ \mathbf{x} &\geq 0 \\ \mathbf{y} &\geq 0 \\ \mathbf{y}^T \mathbf{x} &= 0 \end{aligned}$$

Same as LCP problem.

More on QP relation

From optimization theory we know

- If **A** is positive definite then we have a strict convex QP with a unique solution
- If **A** is positive semi-definite then we have a convex QP where a solution exist but it is no longer unique

This is cool if we have this prior knowledge of **A** ([1]).

[1] The LCP solution needs not be a global solution of the QP or even a minimizer. It is sufficient that the LCP solution fulfills first-order optimality only.

The QP problem – Again

The LCP problem can be restated as

$$\mathbf{x}^* = \arg\min_{\mathbf{x} \geq \mathbf{0}} f(\mathbf{x})$$

where

$$f(\mathbf{x}) \equiv \frac{1}{2}\mathbf{x}^T \mathbf{A}\mathbf{x} + \mathbf{x}^T \mathbf{b}$$

The Idea – Sweep over Coordinates

The i^{th} unit axis vector

$$\mathbf{e}^i_j = \begin{cases} 1 & i = i \\ 0 & i \neq j \end{cases}$$

The i^{th} relaxation step solves the one dimensional problem

$$\tau^* = \arg \min_{\mathbf{x}+\tau\mathbf{e}^i \geq \mathbf{0}} f(\mathbf{x} + \tau \mathbf{e}^i)$$

and computing $\mathbf{x} \leftarrow \mathbf{x} + \tau \mathbf{e}^i$ same as

$$x_i \leftarrow x_i + \tau$$

One relaxation cycle consist of one sequential sweep over all i's.

A Closed Form Solution for the i^{th} Coordinate

The object function of the one-dimensional problem

$$\begin{aligned} f(\mathbf{x}+\tau\mathbf{e}^i) &= \frac{1}{2}(\mathbf{x}+\tau\mathbf{e}^i)^T\mathbf{A}(\mathbf{x}+\tau\mathbf{e}^i)+(\mathbf{x}+\tau\mathbf{e}^i)^T\mathbf{b}, \\ &= \underbrace{\frac{1}{2}\tau^2(\mathbf{A})_{ii}+\tau(\underbrace{\mathbf{A}\mathbf{x}+\mathbf{b}}_{\mathbf{r}})_i}_{g(\tau)}+\underbrace{\frac{1}{2}\mathbf{x}^T\mathbf{A}\mathbf{x}+\mathbf{x}^T\mathbf{b}}_{f(\mathbf{x})\equiv\text{const}}. \end{aligned}$$

So

$$f(\mathbf{x}+\tau\mathbf{e}^i)=g(\tau)+f(\mathbf{x})$$

We just need to minimize $g(\tau)$

Found PGS Again

The unconstrained minimizer of $g(\tau)$

$$\tau^u = -\frac{\mathbf{r}_i}{(\mathbf{A})_{ii}}.$$

Considering the constraint $\mathbf{x}_i + \tau \geq 0$ we have

$$\tau^c = \max\left(-\frac{\mathbf{r}_i}{(\mathbf{A})_{ii}}, -\mathbf{x}_i\right)$$

The final update rule

$$\mathbf{x}_i \leftarrow \max\left(0, \mathbf{x}_i - \frac{\mathbf{r}_i}{\mathbf{A}_{ii}}\right)$$

Algebraic equivalent to the i^{th} step in PGS shown by splitting.

Lessons Learned

- PGS can be derived from a QP reformulation or by a splitting method
- Each derivation assumes different matrix properties of the $\mathbf{A}$-matrix

Study Group Work

- Show that both derivations end up in algebraic equivalent update formulas for $\mathbf{x}_i$.
- What properties should the $\mathbf{A}$ have for splitting derivation to work?
- What properties should the $\mathbf{A}$ have for QP derivation to work?
- What properties should the $\mathbf{A}$ have for PGS to converge?
- Speculate what to do if $\mathbf{A}_{ii} \ngtr 0$ (Hint: look at how to minimize $g(\tau)$)?

The Projected Succesive Over Relaxation Method

Relaxing the Steps

Given the polynomial

$$g(\tau) \equiv \frac{1}{2}\tau^2 \mathbf{A}_{ii} + \tau \mathbf{r}_i$$

where $\mathbf{A}_{ii} > 0$ then

- One trival root at $\tau^1 = 0$
- One global minima at $\tau^u = -\frac{\mathbf{r}_i}{\mathbf{A}_{ii}}$ where $g(\tau^0) < 0$
- Second root at $\tau^2 = -2\frac{\mathbf{r}_i}{\mathbf{A}_{ii}}$.

For any τ between τ^1 and τ^2

$$\tau^\lambda = -\lambda \frac{\mathbf{r}_i}{\mathbf{A}_{ii}} \quad \Rightarrow \quad g(\tau^\lambda) < 0, \quad \forall \lambda \in [0..2].$$

The Projected SOR Method

From this it follows that

$$f(\mathbf{x} + \tau^\lambda \mathbf{e}^i) = g(\tau^\lambda) + f(\mathbf{x}) \leq f(\mathbf{x}), \quad \forall \lambda \in [0..2]$$

With equality for $\tau^\lambda = 0$. This results in the over-relaxed version

$$\mathbf{x}_i \leftarrow \max\left(0, \mathbf{x}_i - \lambda \frac{\mathbf{r}_i}{\mathbf{A}_{ii}}\right).$$

Algebraic equivalent to the i^{th} step in the projected SOR[2].

[2] PGS is special case when $\lambda = 1$

The Fischer–Newton Method

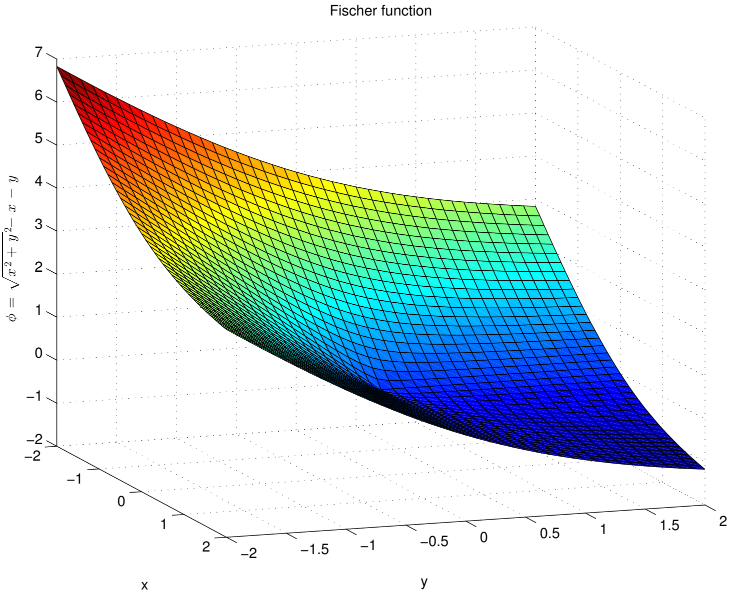
The Fischer Function

The Fischer function

$$\phi(x, y) = \sqrt{x^2 + y^2} - (x + y) \quad \text{for some } x, y \in \mathbb{R}.$$

If one has the complementarity problem $0 \leq x \perp y \geq 0$ then a solution (x^*, y^*) is only a solution if and only if $\phi(x^*, y^*) = 0$

How does it look like?



The Fischer Reformulation

Given the LCP

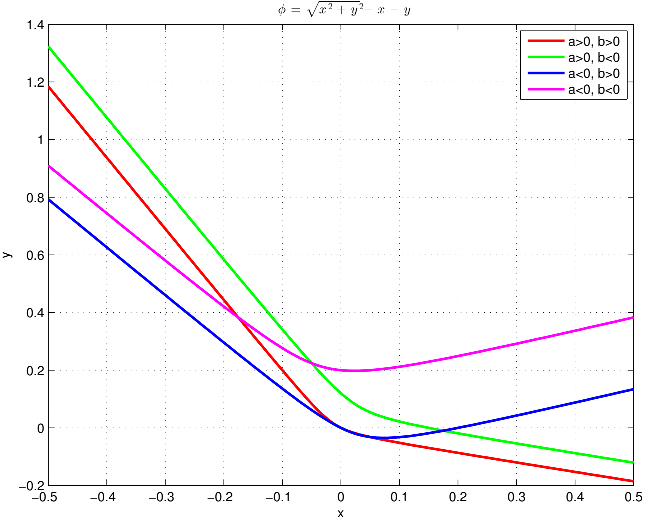
$$\mathbf{0} \leq \mathbf{x} \quad \perp \quad \mathbf{y} = \mathbf{A}\mathbf{x} + \mathbf{b} \geq 0$$

We reformulate

$$\mathbf{F}(\mathbf{x}) = \mathbf{F}(\mathbf{x}, \mathbf{y}) = \begin{bmatrix} \phi(\mathbf{x}_1, \mathbf{y}_1) \\ \vdots \\ \phi(\mathbf{x}_n, \mathbf{y}_n) \end{bmatrix} = \mathbf{0}$$

This is a nonsmooth root search problem

How does it look like?



Pop Quiz

Could one find the roots using "textbook" root-search methods such as Newton-Raphson?

The Fischer–Newton Method

Solved using a generalized Newton–Method. In an iterative fashion solves the generalized Newton subsystem

$$\mathbf{J}\Delta\mathbf{x}^k = -F(\mathbf{x}^k)$$

for the Newton direction $\Delta\mathbf{x}^k$. Here $\mathbf{J} \in \partial F(\mathbf{x}^k)$ is any member from the generalized Jacobian $\partial F(\mathbf{x}^k)$. Then the Newton update yields

$$\mathbf{x}^{k+1} = \mathbf{x}^k + \tau^k \Delta\mathbf{x}^k$$

where τ^k is the step length of the k^{th} iteration.

The B–subdifferential

Given $\mathbf{F}$

- let $\mathcal{D} \subset \mathbb{R}^n$ be the set of all $\mathbf{x} \in \mathbb{R}^n$ where $\mathbf{F}$ is continuously differentiable
- Assume $\mathbf{F}$ is Lipschitz continuous at $\mathbf{x}$

The B–subdifferential of $\mathbf{F}$ at $\mathbf{x}$ is defined as

$$\partial_B \mathbf{F}(\mathbf{x}) \equiv \left\{ \mathbf{H} \in \mathbb{R}^{n \times n} \quad | \quad \exists (\mathbf{x}_k) \subset \mathcal{D} \quad \text{and} \quad \lim_{\mathbf{x}_k \rightarrow \mathbf{x}} \frac{\partial \mathbf{F}(\mathbf{x}_k)}{\partial \mathbf{x}} = \mathbf{H} \right\}$$

Huh, exactly what is this?

The Generalized Jacobian

Clarke's generalized Jacobian of $\mathbf{F}$ at $\mathbf{x}$ is defined as the convex hull of the B–subdifferential,

$$\partial \mathbf{F}(\mathbf{x}) \equiv \mathbf{co}\left(\partial_B \mathbf{F}(\mathbf{x})\right)$$

Huh, exactly what is this?

Example – The Euclidean Norm

Consider the Euclidean norm $e : \mathbb{R}^2 \mapsto \mathbb{R}$ then for $\mathbf{z} = \begin{bmatrix} x & y \end{bmatrix}^T \in \mathbb{R}^2 \setminus \{\mathbf{0}\}$ we have

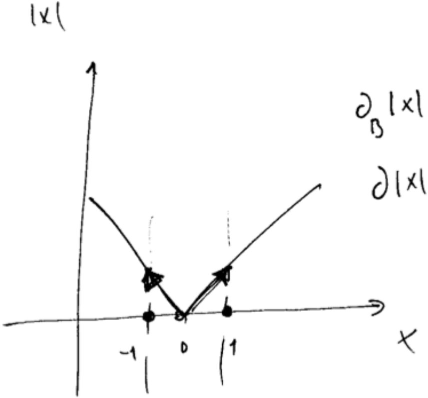
$$\partial e(\mathbf{z}) = \partial_B e(\mathbf{z}) = \frac{\partial e(\mathbf{z})}{\partial \mathbf{z}} = \frac{\mathbf{z}^T}{\| \mathbf{z} \|} \quad \forall \mathbf{z} \neq \mathbf{0}$$

For $\mathbf{z} = \mathbf{0}$ we have

$$\begin{aligned} \partial_B e(\mathbf{0}) &= \{\mathbf{v}^T \quad | \quad \mathbf{v} \in \mathbb{R}^2 \quad \text{and} \quad \| \mathbf{v} \| = 1\} \\ \partial e(\mathbf{0}) &= \{\mathbf{v}^T \quad | \quad \mathbf{v} \in \mathbb{R}^2 \quad \text{and} \quad \| \mathbf{v} \| \leq 1\} \end{aligned}$$

Illustration of Generalized Jacobian in 1D

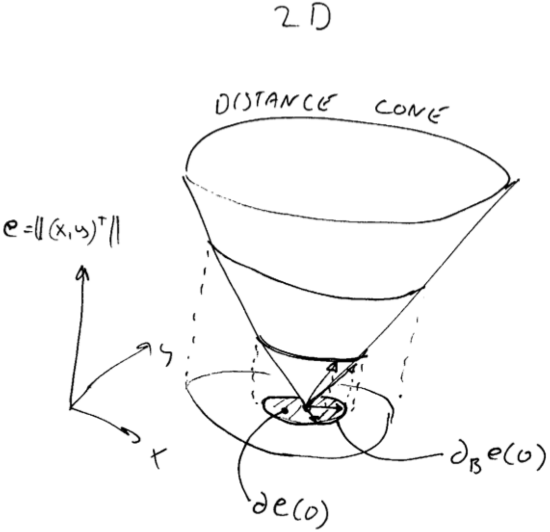
1 D



$\partial_B |x| = \{-1, 1\}$

$\partial |x| = [-1, 1]$

Illustration of Generalized Jacobian in 2D



Example – The Fischer Function

For $\mathbf{x} = \begin{bmatrix} x_1 & x_2 \end{bmatrix}^T \in \mathbb{R}^2$ we may write the Fischer function as

$$\phi(\mathbf{x}) = e(\mathbf{x}) - f(\mathbf{x})$$

where $f(\mathbf{x}) = \left(\begin{bmatrix} 1 & 1 \end{bmatrix}^T \mathbf{x} \right)$. From this we find

$$\partial_B \phi(\mathbf{x}) = \partial_B e(\mathbf{x}) - \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$$
$$\partial \phi(\mathbf{x}) = \partial e(\mathbf{x}) - \frac{\partial f(\mathbf{x})}{\partial \mathbf{x}}$$

Hence for $\mathbf{x} \neq 0$,

$$\partial \phi(\mathbf{x}) = \partial_B \phi(\mathbf{x}) = \left\{ \frac{\mathbf{x}^T}{\| \mathbf{x} \|} - \begin{bmatrix} 1 & 1 \end{bmatrix}^T \right\}$$

and

$$\partial_B \phi(\mathbf{0}) = \{ \mathbf{v}^T - \begin{bmatrix} 1 & 1 \end{bmatrix}^T \quad | \quad \mathbf{v} \in \mathbb{R}^2 \quad \text{and} \quad \| \mathbf{v} \| = 1 \}$$
$$\partial \phi(\mathbf{0}) = \{ \mathbf{v}^T - \begin{bmatrix} 1 & 1 \end{bmatrix}^T \quad | \quad \mathbf{v} \in \mathbb{R}^2 \quad \text{and} \quad \| \mathbf{v} \| \leq 1 \}$$

Generalized Jacobian of The Fischer Reformulation

Written as

$$\partial F(\mathbf{x}) \equiv \mathbf{D}_p(\mathbf{x}) + \mathbf{D}_q(\mathbf{x})\mathbf{A}$$

where $\mathbf{D}_p(\mathbf{x}) = \mathbf{diag}\left(p_1(\mathbf{x}), \ldots, p_n(\mathbf{x})\right)$ and $\mathbf{D}_q(\mathbf{x}) = \mathbf{diag}\left(q_1(\mathbf{x}), \ldots, q_n(\mathbf{x})\right)$ are diagonal matrices. If $\mathbf{y}_i \neq 0$ or $\mathbf{x}_i \neq 0$ then

$$p_i(\mathbf{x}) = \frac{\mathbf{x}_i}{\sqrt{\mathbf{x}_i^2 + \mathbf{y}_i^2}} - 1,$$

$$q_i(\mathbf{x}) = \frac{\mathbf{y}_i}{\sqrt{\mathbf{x}_i^2 + \mathbf{y}_i^2}} - 1,$$

else if $\mathbf{y}_i = \mathbf{x}_i = 0$ then

$$p_i(\mathbf{x}) = \alpha_i - 1,$$

$$q_i(\mathbf{x}) = \beta_i - 1$$

for any $\alpha_i, \beta_i \in \mathbb{R}$ such that $\| \begin{bmatrix} \alpha_i & \beta_i \end{bmatrix}^T \| \leq 1$

Proof of $\partial F(\mathbf{x})$

Assume $\mathbf{y}_i \neq 0$ or $\mathbf{x}_i \neq 0$ then the differential is

$$d\mathbf{F}_i(\mathbf{x}, \mathbf{y}) = d\left(\left(\mathbf{x}_i^2 + \mathbf{y}_i^2\right)^{\frac{1}{2}}\right) - d\left(\mathbf{x}_i + \mathbf{y}_i\right)$$

By chain rule

$$\begin{aligned}
d\mathbf{F}_i(\mathbf{x}, \mathbf{y}) &= \frac{1}{2}\left(\mathbf{x}_i^2 + \mathbf{y}_i^2\right)^{-\frac{1}{2}} d\left(\mathbf{x}_i^2 + \mathbf{y}_i^2\right) - d\mathbf{x}_i - d\mathbf{y}_i \\
&= \frac{\mathbf{x}_i d\mathbf{x}_i + \mathbf{y}_i d\mathbf{y}_i}{\sqrt{\mathbf{x}_i^2 + \mathbf{y}_i^2}} - d\mathbf{x}_i - d\mathbf{y}_i \\
&= \left[\underbrace{\left(\frac{\mathbf{x}_i}{\sqrt{\mathbf{x}_i^2 + \mathbf{y}_i^2}} - 1\right)}_{p_i(\mathbf{x})} \quad \underbrace{\left(\frac{\mathbf{y}_i}{\sqrt{\mathbf{x}_i^2 + \mathbf{y}_i^2}} - 1\right)}_{q_i(\mathbf{x})}\right] \begin{bmatrix} d\mathbf{x}_i \\ d\mathbf{y}_i \end{bmatrix}
\end{aligned}$$

Proof of $\partial F(\mathbf{x})$ (Contd)

Finally $d\mathbf{y} = \mathbf{A}d\mathbf{x}$, so $d\mathbf{y}_i = \mathbf{A}_{i*}d\mathbf{x}$ by substitution

$$d\mathbf{F}_i(\mathbf{x}, \mathbf{y}) = \underbrace{\left(p_i(\mathbf{x})\mathbf{e}_i^T + q_i(\mathbf{x})\mathbf{A}_{i*}\right)}_{\partial F_i(\mathbf{x})} d\mathbf{x}$$

The case $\mathbf{x}_i = \mathbf{y}_i = 0$ follows from the previous examples.

How to solve Generalized Newton System

- whenever $\mathbf{x}_i = \mathbf{y}_i = 0$ one would use $\mathbf{x}'_i = \varepsilon$ in-place $\mathbf{x}_i$ when evaluating the generalized Jacobian where $0 < \varepsilon \ll 1$
- If Newton system is solved with iterative method (GMRES) then we only need to compute $\mathbf{J}\Delta\mathbf{x}$. By definition of directional derivative

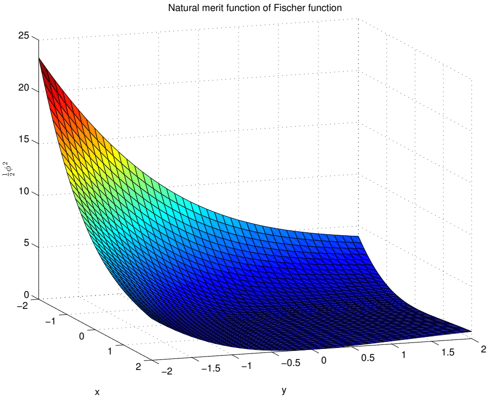
$$\mathbf{J}\Delta\mathbf{x} = \lim_{h \to 0^+} \frac{\mathbf{F}(\mathbf{x} + h\Delta\mathbf{x}) - \mathbf{F}(\mathbf{x})}{h}$$

  So we can numerically approximate $\mathbf{J}\Delta\mathbf{x}$ using finite differences
- A projected Armijo back-tracking can be very usefull to globalize the Newton method and to ensure feasibility of all iterates

Natural Merit Function

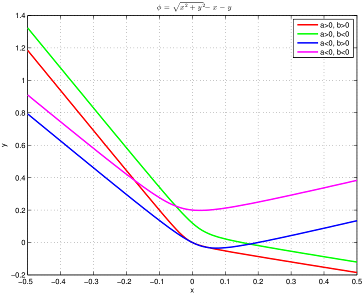
Define natural merit function

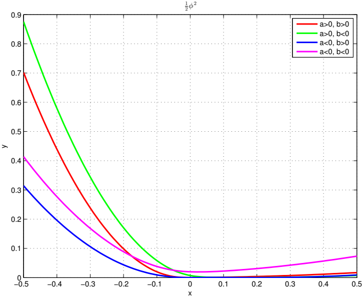
$$\theta(\mathbf{x}) = \frac{1}{2}\mathbf{F}(\mathbf{x})^T\mathbf{F}(\mathbf{x})$$



Looks nice...but?

Let us look at 1D cases





Anything to worry about here?

Projected Armijo Backtracking Line Search

Project Newton Search Direction

$$\Delta x \leftarrow \max\left(\mathbf{0}, \Delta \mathbf{x}\right)$$

Find smallest $k \in \mathbf{Z}_0$ such that

$$\theta(\mathbf{x} + \alpha^k \Delta \mathbf{x}) \leq \theta(\mathbf{x}) + \underbrace{\left(\beta \frac{\partial \theta(\mathbf{x})}{\partial \mathbf{x}} \Delta \mathbf{x}\right)}_{c \equiv \text{const}} \alpha^k$$

for some user defined constants $0 \leq \beta < \alpha < 1$. Now

$$\tau = \alpha^k$$

$$\mathbf{x} \leftarrow \mathbf{x} + \tau \Delta \mathbf{x}$$

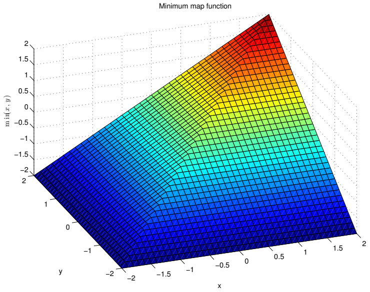
Study Group Work

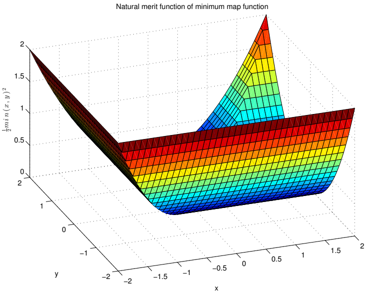
As a challenge if you have time. Derive a Nonsmooth Newton method for the minimum map reformulation of the LCP

$$\min(\mathbf{x}, \mathbf{y}) = \mathbf{0}$$

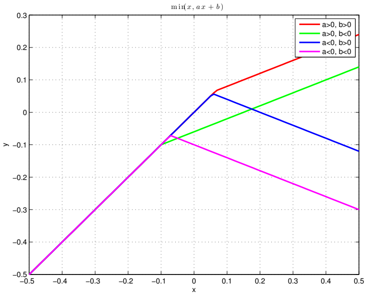
(Hint: Look at Erleben and Ortiz: A non-smooth newton method for multibody dynamics, In Proc. of ICNAAM 06')

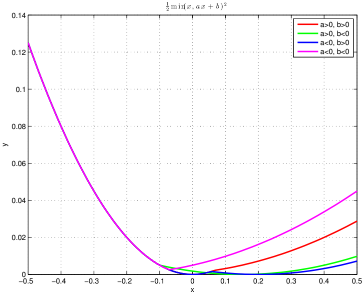
How does the Minimum Map function look like?





What if we throw in $y = ax + b$?





Further Reading

- K. G. Murty: Linear complementarity, linear and nonlinear programming. Sigma Series in Applied Mathematics. 3. Berlin: Heldermann Verlag, 1988. (Chapter 1 and 9)
- R. W. Cottle, J.-S. Pang, R. E. Stone. The Linear Complementarity Problem. Academic Press. 1992. (Chapter 1 and 5)
- Stephen C. Billups and Katta G. Murty: Complementarity problems, Journal of Computational and Applied Mathematics Volume 124, Issues 1-2, Pages 1-373 (1 December 2000)
- D.E. Stewart and J.C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with inelastic collisions and coulomb friction. International Journal of Numerical Methods in Engineering, 39:2673-2691

More Reading

- D.E. Stewart and J.C. Trinkle. Dynamics, friction, and complementarity problems. In M.C. Ferris and J.S. Pang, editors, Complementarity and Variational Problems, pages 425-439. SIAM, 1997.
- D.E. Stewart and J.C. Trinkle. An implicit time-stepping scheme for rigid body dynamics with coulomb friction. In Proceedings, IEEE International Conference on Robots and Automation, pages 162-169, 2000.
- M. Silcowitz, S. Niebe, and K. Erleben. Nonsmooth newton method for fischer function reformulation of contact force problems for interactive rigid body simulation. In Proceedings of VRIPHYS, 2009.
- M. Silcowitz, S. Niebe, and K. Erleben. A nonsmooth nonlinear conjugate gradient method for interactive contact force problems. The Visual Computer, 2010.

Study Group Work

- Examine the Stewart–Trinkle (ST) LCP formulation of the contact force problem.
- What matrix properties can you identify?
- What do you know about the right hand side vector?
- What kind of reformulations are applicable to the ST LCP formulation?
- What kind of methods can be used to solve a ST LCP formulation problem?

Basic Programming Exercise

- Obtain Lemke's method from CPNET[3]
- Create a routine that can generate random N-dimensional LCP problems.
- Generate a sequence of random LCP test problems with increasing number of variables, $N = 2^k, k = 2, 3, \ldots, 10$.
- Use Lemke to solve random sequences of LCP test problems (run 10 sequences at least).
- Make plots of computing time as a function of increasing number of variables and make a histogram showing the fraction of solved problems.
- Discuss your results – what do you think of Lemke's method?

[3] http://www.cs.wisc.edu/cpnet/

Intermediate Programming Exercise

- Try to implement a PGS solver, a Fischer–Newton solver,and use a QP reformulation solved by using the Matlab **quadprog** function.
- Create a seqeunce of random problems of increasing size and use the solvers to find solutions.
  - Compare the accuracy of each iterative solver with the true solution, what is the error
  - Plot how the error of each solver behaves as a function of the number of iterations (Hint: make a log plot and determine convergence rate)
  - Try to measure the computing cost per iteration of each solver as a function of increasing variables (Hint: compare plot with your complexity analysis)
- Based on your experiments evaluate if you implementations behaves as expected, speculate for what purposes you want to use different solvers for.

Advanced Programming Exercise

Create a simple 2D rigid body simulator using sperical objects of varying size and mass. Ignore friction and use simple first-order time stepping. In each simulation step solve a LCP for the normal penetration constraints.

- Determine the eigenvalue spectrum of the coefficient-matrix
- Determine if the coefficient matrix is symmetric or not
- Try experimenting with using different LCP solvers (take those from the intermediate programming exercise)
- Which solver do you think is best for this particular simulator and why?
- If you have more time try to add friction to the contact forces and rerun all your tests. Did this change on your conclusion on which solver is the best?

Hints if you get Stuck

Go have a look at

http://code.google.com/p/num4lcp/

Its an open source project for numerical methods (NUM) for (4) linear complementarity problems (LCP) in physics-based animation.

- It has MATLAB implementations for all solvers in these slides (and the notes)
- Some of the solvers are also in python
- We are soon releasing CUDA/CUSP solvers of Newton methods