Some Recent Advances in Alternating Direction Methods: Practice and Theory

Yin Zhang (张寅)

Department of Computational and Applied Mathematics
Rice University, Houston, Texas, USA

The 5th Sino-Japanese Optimization Meeting
Beijing, China, September 28, 2011
Outline:

- Alternating Direction Method (ADM)
- Recent Revival and Extensions
- Local Convergence and Rate
- Global Convergence
- Summary

Contributors:
Xin Liu, Junfeng Yang, Zaiwen Wen, Yilun Wang, Chengbo Li, Yuan Shen, Wei Deng, Wotao Yin
Basic Ideas

To an extent, constructing algorithm \( \approx \) “Art of Balance”

- Optimization algorithms are “always” iterative
- Total cost = (number of iterations) \( \times \) (cost/iter)
- 2 objectives above

It’s more difficult to analyze iteration complexity.
A good iteration complexity \( \neq \) fast algorithm

ADM Idea: lower per-iteration complexity

Approach:
- “远交近攻”, “各个击破” — Sun-Tzu (400 BC)
- “Divide and Conquer” — Julius Caesar (100-44 BC)
Convex program with the 2-separability structure

\[
\begin{align*}
\min_{x,y} & \quad f_1(x) + f_2(y), \quad \text{s.t.} \quad Ax + By = b, \quad x \in \mathcal{X}, \quad y \in \mathcal{Y} \\
& \quad f(x,y)
\end{align*}
\]

Augmented Lagrangian (AL): penalty $\beta > 0$

\[
\mathcal{L}_A(x, y, \lambda) = f(x, y) - \lambda^\top (Ax + By - b) + \frac{\beta}{2} \|Ax + By - b\|^2
\]

Classic AL Multiplier Method (ALM): step $\gamma \in (0, 2)$

\[
\begin{cases}
(x^{k+1}, y^{k+1}) \leftarrow \arg\min_{x,y} \{ \mathcal{L}_A(x, y, \lambda^k) : x \in \mathcal{X}, y \in \mathcal{Y} \} \\
\lambda^{k+1} \leftarrow \lambda^k - \gamma \beta (Ax^{k+1} + By^{k+1} - b)
\end{cases}
\]

Hestines-69, Powell-69, Rockafellar-73

(It does not explicitly use 2-separability)
Classic Alternating Direction Method \((\text{交替方向法})\)

Replace joint minimization by alternating minimization once:

\[
\min_{x,y} \mathcal{L}_A \approx (\min_x \mathcal{L}_A) \oplus (\min_y \mathcal{L}_A)
\]

(AL)ADM: step \(\gamma \in (0, 1.618)\)

\[
\begin{align*}
\mathbf{x}^{k+1} & \leftarrow \arg \min_x \{ \mathcal{L}_A(x, y^k, \lambda^k) : x \in \mathcal{X} \} \\
y^{k+1} & \leftarrow \arg \min_y \{ \mathcal{L}_A(x^{k+1}, y, \lambda^k) : y \in \mathcal{Y} \} \\
\lambda^{k+1} & \leftarrow \lambda^k - \gamma \beta (A\mathbf{x}^{k+1} + B\mathbf{y}^{k+1} - \mathbf{b})
\end{align*}
\]

It does use 2-separability: ("远交近攻", "各个击破")

- \(x\)-subproblem:

\[
\min_x f_1(x) + \frac{\beta}{2} \|Ax - c_1(y^k)\|^2
\]

- \(y\)-subproblem:

\[
\min_y f_2(y) + \frac{\beta}{2} \|By - c_2(x^{k+1})\|^2
\]
ADM overview (I)

ADM as we know today

- Glowinski-Marocco-75 and Gabay-Mercier-76
- Glowinski *at el.* 81-89, Gabay-83...

Connections before Aug. Lagrangian

- Douglas, Peaceman, Rachford *(middle 1950’s)*
- operator splittings for PDE *(a.k.a. ADI methods)*
After PDE, subsequent studies in optimization

- variational inequality, proximal-point, Bregman, ...
  (Eckstein-Bertsekas-92 ......)
- inexact ADM (He-Liao-Han-Yang-02 ......)
- Tseng-91, Fukushima-92, ...
- proximal-like, Bregman (Chen and Teboulle-93)
- ......

ADM had been used in optimization to some extent, but not as widely used to be called “main-stream” algorithm
ADM overview (III)

Recent Revival in Signal/Image/Data Processing

- $\ell_1$-norm, total variation (TV) minimization
- convex, non-smooth, simple structures

Splitting + alternating:

  (split + quadratic penalty, 2007)
  (split + quadratic penalty + multiplier in code, 2008)
- Goldstein-Osher-2008, split Bregman
  (split + quadratic penalty + Bregman, earlier in 2008)
- ADM $\ell_1$-solver for 8 models: YALL1. Yang-Z-2010

Googled “split Bregman”: “found 16,300 results”.
Turns out that hot split Bregman = cool ALM
ADM Global Convergence

e.g., “Augmented Lagrangian methods ...” Fortin-Glowinski-83

Assumptions required by current theory:
- convexity over the entire domain
- separability for exactly 2 blocks, no more
- exact or high-accuracy minimization for each block

Strength:
- differentiability not required
- side-constraints allowed: $x \in X, y \in Y$

But
- why not 3 or more blocks?
- very rough minimization?
- rate of convergence?
Some Recent Applications

From PDE to:

Signal/Image Processing
Sparse Optimization
TV-minimization in Image Processing

TV/L^2 deconvolution model (Rudin-Osher-Fatemi-92):

\[
\min_u \sum_i \|D_i u\| + \frac{\mu}{2} \|Ku - f\|^2 \text{ (sum all pixels)}
\]

Splitting:

\[
\min_{u, w} \left\{ \sum_i \|w_i\| + \frac{\mu}{2} \|Ku - f\|^2 : w_i = D_i u, \forall i \right\}.
\]

Augmented Lagrangian function \( \mathcal{L}_A(w, u, \lambda) \):

\[
\sum_i \left( \|w_i\| - \lambda^T_i (w_i - D_i u) + \frac{\beta}{2} \|w_i - D_i u\|^2 \right) + \frac{\mu}{2} \|Ku - f\|^2.
\]

Closed formulas for minimizing w.r.t. \( w \) (shrinkage) and \( u \) (FFT) (almost linear-time per iteration)
Shrinkage (or Soft Thresholding)

Solution to a simple optimization problem:

\[ x(v, \mu) := \arg \min_{x \in \mathbb{R}^d} \|x\| + \frac{\mu}{2} \|x - v\|^2 \]

where \( \| \cdot \| \) is the Euclidean norm in \( \mathbb{R}^d \), \( v \neq 0 \) and \( \mu > 0 \).

\[ x(v, \mu) = \max \left( \|v\| - \frac{1}{\mu}, 0 \right) \frac{v}{\|v\|} \]

This formula was used at least 30 years ago.

Orders of magnitude faster than PDE-based methods.

Key: “splitting-alternating” takes advantage of the structure. Use of multiplier brings further speedup.
Example: Cross-channel blur + Gaussian noise

FTVd: $\min_u \text{TV}(u) + \mu \| Ku - f \|^2_2$, sizes $512^2$ and $256^2$

( computation by Junfeng Yang)
\( \ell_1 \)-minimization in Compressive Sensing

Signal acquisition/compression: \( A \in \mathbb{R}^{m \times n} \ (m < n) \)

\[
b \approx Ax^* \in \mathbb{R}^m
\]

where \( x^* \in \mathbb{R}^n \) is sparse or compressible under a orthogonal transformation \( \Psi \). \( \ell_1 \) norm is used as the surrogate of sparsity.

8 signal recovery models: \( A \in \mathbb{R}^{m \times n} \ (m < n) \)

1. \[
\min \| \Psi x \|_1, \text{ s.t. } Ax = b \quad (x \geq 0)
\]
2. \[
\min \| \Psi x \|_1, \text{ s.t. } \| Ax - b \|_2 \leq \delta \quad (x \geq 0)
\]
3. \[
\min \| \Psi x \|_1 + \mu \| Ax - b \|_2^2 \quad (x \geq 0)
\]
4. \[
\min \| \Psi x \|_1 + \mu \| Ax - b \|_1 \quad (x \geq 0)
\]

Can we solve these 8 model by \( \leq 30 \) lines of 1 Matlab code? YALL1 using ADM.
\( \ell_1 \)-minimization in Compressive Sensing (II)

Sparse signal recovery model: \( A \in \mathbb{R}^{m \times n} \ (m < n) \)

\[
\begin{align*}
\min \{ \|x\|_1 : Ax &= b \} & \iff \max \{ b^\top y : A^\top y = z \in [-1, 1]^n \}
\end{align*}
\]

Add splitting \( z \) to "free" \( A^\top y \) from the unit box:

\[
\max \{ b^\top y : A^\top y = z \in [-1, 1]^n \}
\]

ADM (1 of variants in Yang-Z-09): \( AA^\top = I \) (common in CS)

\[
\begin{align*}
y & \leftarrow A(z - x) + b / \beta \\
z & \leftarrow P_{[-1,1]^n}(A^\top y + x) \\
x & \leftarrow x - \gamma(z - A^\top y)
\end{align*}
\]
## Numerical Comparison

ADM solver package **YALL1**: [http://yall1.blogs.rice.edu/](http://yall1.blogs.rice.edu/)

Compared codes: SPGL1, NESTA, SpaRSA, FPC, FISTA, CGD

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Relative Error</th>
<th>m/n: 0.20, p/m: 0.20,</th>
<th>PADM</th>
<th>DADM</th>
<th>SPGL1</th>
<th>NESTA</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Iteration</th>
<th>Relative Error</th>
<th>m/n: 0.30, p/m: 0.20, μ: 1.0e−004</th>
<th>PADM</th>
<th>DADM</th>
<th>SpaRSA</th>
<th>FPC_BB</th>
<th>FISTA</th>
<th>CGD</th>
</tr>
</thead>
<tbody>
<tr>
<td>10</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>100</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000</td>
<td>10</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(noisy measurements, average of 50 runs)

Nonasymptotically, ADMs showed the fastest speed of convergence in reducing error $\|x^k - x^*\|$. 
Single Parameter $\beta$

In theory, $\beta > 0 \implies$ convergence

How to choose the penalty parameter in practice?

In YALL1: **Make the subproblems scalar scale invariant**

- Scale $A$ to “unit” size
- Scale $b$ accordingly.
- $\beta = m/\|b\|_1$.

Optimal choice is still an open theoretical question.
Signal Reconstruction with Group Sparsity

Group-sparse signal \( x = (x_1; \cdots; x_s) \), \( x_i \in \mathbb{R}^{n_i}, \sum_{i=1}^{s} n_i = n \)

\[
\min_x \sum_{i=1}^{s} \|x_i\|_2 \quad \text{s.t.} \quad Ax = b.
\]

Introduce splitting \( y \in \mathbb{R}^n \),

\[
\min_{x,y} \sum_{i=1}^{s} \|y_i\|_2 \quad \text{s.t.} \quad y = x, \quad Ax = b.
\]

ADM (Deng-Yin-Z-10):

\[
y \leftarrow \text{shrink}(x + \lambda_1, 1/\beta) \quad \text{(group-wise)}
\]

\[
x \leftarrow (I + A^T A)^{-1}((y - \lambda_1) + A^T (b + \lambda_2))
\]

\[
(\lambda_1, \lambda_2) \leftarrow (\lambda_1, \lambda_2) - \gamma(y - x, Ax - b)
\]

Easy if \( AA^T = I \); else take a steepest descent step in \( x \) (say).
Multi-Signal Reconstruction with Joint Sparsity

Recover a set of jointly sparse signals $X = [x_1 \cdots x_p] \in \mathbb{R}^{n \times p}$

$$\min_X \sum_{i=1}^{n} \| e_i^T X \| \quad \text{s.t.} \quad A_j x_j = b_j, \ \forall j.$$ 

Assume $A_j = A$ for simplicity. Introduce splitting $Z \in \mathbb{R}^{p \times n}$,

$$\min_X \sum_{i=1}^{n} \| Ze_i \| \quad \text{s.t.} \quad Z = X^T, \ AX = B.$$ 

ADM (Deng-Yin-Z-10):

$$Z \leftarrow \text{shrink}(X^T + \Lambda_1, 1/\beta) \quad \text{(column-wise)}$$

$$X \leftarrow (I + A^T A)^{-1}((Z - \Lambda_1)^T + A^T (B + \Lambda_2))$$

$$(\Lambda_1, \Lambda_2) \leftarrow (\Lambda_1, \Lambda_2) - \gamma(Z - X^T, AX - B)$$

Easy if $AA^T = I$; else take a steepest descent step in $X$. 
Extensions to Non-convex Territories

(as long as convexity exists in each direction)

Low-Rank/Sparse Matrix Models

Non-separable functions

More than 2 blocks
Matrix Fitting Models (I): Completion

Find low-rank $Z$ to fit data $\{M_{ij} : (i, j) \in \Omega\}$

Nuclear-norm minimization is good, but SVDs are expensive.

Non-convex model (Wen-Yin-Z-09): find $X \in \mathbb{R}^{m \times k}$, $Y \in \mathbb{R}^{k \times n}$

$$\min_{X, Y, Z} \|XY - Z\|_F^2 \text{ s.t. } P_\Omega(Z - M) = 0$$

An SOR scheme:

$$Z \leftarrow \omega Z + (1 - \omega)XY$$
$$X \leftarrow \text{qr}(ZY^\top)$$
$$Y \leftarrow X^\top Z$$
$$Z \leftarrow XY + P_\Omega(M - XY)$$

1 small QR ($m \times k$). No SVD. $\omega$ dynamically adjusted.

Much faster than nuclear-norm codes (when it is applicable)
Nonlinear GS vs SOR

(a) $n=1000, r=10, \text{SR} = 0.08$

(b) $n=1000, r=10, \text{SR}=0.15$

Alternating minimization, but no multiplier for storage reason

Is non-convexity a problem for global optimization of this problem?
— “Yes” in theory
— “Not really” in practice
Matrix Fitting Models (II): Separation

Given data \( \{D_{ij} : (i,j) \in \Omega \} \),

Find low-rank \( Z \) so that difference \( P_\Omega(Z - D) \) is sparse

Non-convex Model (Shen-Wen-Z-10):

\[
U \in \mathbb{R}^{m \times k}, \ V \in \mathbb{R}^{k \times n}
\]

\[
\min_{U,V,Z} \|P_\Omega(Z - D)\|_1 \quad \text{s.t.} \quad Z - UV = 0
\]

ADM scheme:

\[
U \leftarrow \text{qr}((Z - \Lambda/\beta)V^T)
\]

\[
V \leftarrow U^T(Z - \Lambda/\beta)
\]

\[
P_\Omega^c(Z) \leftarrow P_\Omega^c(UV + \Lambda/\beta)
\]

\[
P_\Omega(Z) \leftarrow P_\Omega(\text{shrink}(...)+D)
\]

\[
\Lambda \leftarrow \Lambda - \gamma \beta (Z - UV)
\]

— 1 small QR. No SVD. Faster.
— non-convex, 3 blocks. nonlinear constraint. convergence?
Nonnegative Matrix Factorization (Z-09)

Given \( A \in \mathbb{R}^{n \times n} \), find \( X, Y \in \mathbb{R}^{n \times k} \) \( (k \ll n) \),

\[
\min \|XY^T - A\|_F^2 \quad \text{s.t.} \quad X, Y \geq 0
\]

Splitting:

\[
\min \|XY^T - A\|_F^2 \quad \text{s.t.} \quad X = U_1, Y = U_2, U_1, U_2 \geq 0
\]

ADM scheme:

\[
X \leftarrow (AY + \beta(U_1 - \Lambda_1))(Y^T Y + \beta I)^{-1}
\]
\[
Y^T \leftarrow (X^T X + \beta I)^{-1}(X^T A + \beta (U_2 - \Lambda_2))
\]
\[
(U_1, U_2) \leftarrow P_+(X + \Lambda_1, Y + \Lambda_2)
\]
\[
(\Lambda_1, \Lambda_2) \leftarrow (\Lambda_1, \Lambda_2) - \gamma(X - U_1, Y - U_2)
\]

— cost/iter: 2 \( (k \times k) \) linear systems plus matrix arithmetics
— better performance than Matlab built-in function “nnmf”
— non-convex, non-separable, 3 blocks: convergence?
Theoretical Convergence Results

A general setting

Local $R$-linear convergence

Global convergence for linear constraints

(Liu-Yang-Z, work in progress)
Consider

$$\min_{x} f(x) \quad \text{s.t.} \quad c(x) = 0$$

where $f : \mathbb{R}^n \to \mathbb{R}$ and $c : \mathbb{R}^n \to \mathbb{R}^m (m < n)$ are $C^2$-mappings.

Augmented Lagrangian:

$$\mathcal{L}_\alpha(x, y) \triangleq \alpha f(x) - y^T c(x) + \frac{1}{2} \| c(x) \|^2$$

Augmented saddle point system:

$$\nabla_x \mathcal{L}_\alpha(x, y) = 0, \quad c(x) = 0.$$
Splitting and Iteration Scheme

\[ G : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n \] is a splitting of \( F : \mathbb{R}^n \rightarrow \mathbb{R}^n \) if

\[ G(x, x) \equiv F(x), \ \forall x \in \mathbb{R}^n. \]

e.g., if \( A = L - R \), \( G(x, x) \triangleq Lx - Rx \equiv Ax \triangleq F(x) \).

Let \( G(x, x, y) \) be a splitting of \( \nabla_x \mathcal{L}_\alpha(x, y) \) on \( x \)

Augmented saddle point system becomes

\[
\begin{align*}
G(x, x, y) &= 0 \\
c(x) &= 0
\end{align*}
\]

A general Split (gSS) Scheme for Saddle-point Systems:

\[
\begin{align*}
x^{k+1} &\leftarrow G(x, x^k, y^k) = 0 \\
y^{k+1} &\leftarrow y^k - \tau c(x^{k+1})
\end{align*}
\]
Block Jacobi for Square System $F(x) = 0$

Partition the system and variable into $s \leq n$ consistent blocks:

$$F = (F_1, F_2, \ldots, F_s), \quad x = (x_1, x_2, \ldots, x_s)$$

Block Jacobi iteration: given $x^k$, for $i = 1, 2, \ldots, s$

$$x_i^{k+1} \leftarrow F_i(x_1^k, \ldots, x_{i-1}^k, x_i, x_{i+1}^k, \ldots, x_s^k) = 0$$

or

$$x^{k+1} \leftarrow G(x, x^k) = 0$$

where

$$G(x, z) = \begin{pmatrix}
F_1(x_1, z_2, \ldots, z_s) \\
\vdots \\
F_i(z_1, \ldots, x_i, z_{i+1}, \ldots, z_s) \\
\vdots \\
F_s(z_1, \ldots, x_s)
\end{pmatrix}$$
Block Gauss-Seidel for Square System $F(x) = 0$

Block GS iteration: given $x^k$, for $i = 1, 2, \ldots, s$

$$x_i^{k+1} \leftarrow F_i(x_1^{k+1}, \ldots, x_{i-1}^{k+1}, x_i, x_{i+1}^{k}, \ldots, x_s^k) = 0$$

or $$x^{k+1} \leftarrow G(x, x^k) = 0$$

where

$$G(x, z) = \begin{pmatrix} F_1(x_1, z_2, \ldots, z_s) \\ \vdots \\ F_i(x_1, \ldots, x_i, z_{i+1}, \ldots, z_s) \\ \vdots \\ F_s(x_1, \ldots, x_s) \end{pmatrix}$$

(SOR can be similarly defined.)
Splitting for Gradient Descent: \( F(x) = \nabla f(x) \)

Gradient descent method (with a constant step size):

\[
x^{k+1} = x^k - \alpha F(x^k),
\]

or \( x^{k+1} \leftarrow G(x, x^k) = 0 \)

where

\[
G(x, z) = \frac{1}{\alpha} x - \left( \frac{1}{\alpha} z - F(z) \right).
\]

— gradient descent iterations can be done block-wise
— block GS, SOR and gradient descent can be mixed
  (e.g., 1st block: GS; 2nd block: gradient descent)
Assumptions

Let \( \partial_i G(x, x, y) \) be the partial Jacobian of the splitting \( G \) w.r.t. the \( i \)-th argument, and \( \partial_i G^* \triangleq \partial_i G(x^*, x^*, y^*) \) where \( x^* \) is a minimizer and \( y^* \) the associated multiplier.

**Assumption 1.** (2nd-order Sufficiency)
\( f, c \in C^2 \), and \( \alpha > 0 \) is chosen so that
\[
\nabla_x^2 \mathcal{L}_\alpha(x^*, y^*) \succ 0
\]

**Assumption 2.** (Requirement on splitting)
\( \partial_1 G \) is nonsingular in a neighborhood of \( (x^*, x^*, y^*) \), and
\[
\rho \left( [\partial_1 G^*]^{-1} \partial_2 G^* \right) < 1
\]
(e.g., for gradient descent: \( [\partial_1 G^*]^{-1} \partial_2 G^* = I - \alpha \nabla^2 f(x^*) \))
Assumptions are Reasonable

**A1.** 2nd-order sufficiency guarantees that $\alpha > 0$ exists so that

$$\alpha \left[ \nabla^2 f(x^*) - \sum_i \hat{y}_i^* \nabla^2 c_i(x^*) \right] + A(x^*)^\top A(x^*) \succ 0$$

where $A(x) = \partial c(x)$. Note

$$\nabla_x L_\alpha(x, y) = G(x, x, y) \implies \nabla^2_x L^*_\alpha = \partial_1 G^* + \partial_2 G^* \succ 0$$

**A2.** Any convergent linear splitting for matrices $\succ 0$ leads to a corresponding nonlinear splitting $G$ satisfying

$$\rho \left( [\partial_1 G^*]^{-1} \partial_2 G^* \right) < 1$$

Hence, **A2** is satisfied by block GS (i.e., ADM), SOR, gradient descent (with appropriate $\alpha$) and their mixtures.
The Error System

Recall gSS:

\[
x^{k+1} \leftarrow G(x, x^k, y^k) = 0
\]
\[
y^{k+1} \leftarrow y^k - \tau c(x^{k+1})
\]

Using Implicit Function Theorem, we derive an error system

\[
e^{k+1} = M^*(\tau)e^k + o(\|e^k\|)
\]

where \( e^k \triangleq (x^k, y^k) - (x^*, y^*) \),

\[
M^*(\tau) = \begin{bmatrix}
- [\partial_1 G^*]^{-1} \partial_2 G^* & [\partial_1 G^*]^{-1} A^*^\top \\
\tau A^* [\partial_1 G^*]^{-1} \partial_2 G^* & I - \tau A^* [\partial_1 G^*]^{-1} A^*^\top
\end{bmatrix}
\]

Key Lemma. (Z-2010) Under Assumptions 1-2, there exists \( \eta > 0 \) such that \( \rho(M^*(\tau)) < 1 \) for all \( \tau \in (0, 2\eta) \).
Convergence: $\tau \in (0, 2\eta)$

**Theorem [Local convergence].**
There exists an open neighborhood $U$ of a KKT point $(x^*, y^*)$ such that for any $(x^0, y^0) \in U$, the sequence $\{(x^k, y^k)\}$ generated by gSS stays in $U$ and converges to $(x^*, y^*)$.

**Theorem [R-linear rate].**
The asymptotic convergence rate of gSS is $R$-linear with $R$-factor $\rho(M^*(\tau))$, i.e.,

$$\limsup_{k \to \infty} \frac{\| (x^k, y^k) - (x^*, y^*) \|^{1/k}}{k} = \rho(M^*(\tau)).$$

— These follow from the Key Lemma and Ortega-Rockoff-70.

**Corollary [quadratic case].**
If $f$ is quadratic and $c$ is affine, then $U = \mathbb{R}^n \times \mathbb{R}^m$ and the convergence is globally $Q$-linear with $Q$-factor $\rho(M^*(\tau))$. 
Global Convergence: Linear Constraints

\[
\min_x f(x_1, \cdots, x_p), \quad \text{s.t.} \quad \sum A_i x_i = b
\]

1st-order optimality or saddle point system:

\[
\nabla f(x) = A^\top y \\
A x - b = 0
\]

Augmented saddle point system:

\[
\nabla f(x) + \beta A^\top (A x - b) = A^\top y \\
y - \tau \beta (A x - b) = y
\]

Splittings \((F(x) = G(x, x))\) can be applied to the 1st equation.
- Block Jacobi type give block diagonal split
- ADM: a block Gauss-Seidel type split
Global Convergence (preliminary)

\[ \min_{x} f(x_1, \ldots, x_p), \text{ s.t. } \sum A_i x_i = b \]

\( f \) is separable if \( f(x_1, \ldots, x_p) = \sum_{i}^p f_i(x_i) \). In this case, the Hessian is block diagonal.

**Block Jacobi scheme:**
If \( f \in C^2 \) is separable, and each

\[ \nabla^2 f_i(x_i) + \beta A_i^T A_i \succeq \epsilon I, \]

\( \nabla^2 \mathcal{L}_{\alpha} \) is uniformly block diagonally dominant, then the block Jacobi scheme converges to a KKT point.

It can be extended to more general settings (GS, ...) under further assumptions (still under scrutiny).

**The number of blocks can be arbitrary** without modification.
Other multi-block extensions exist with convexity and algorithm modifications (He and Yuan et al).
Summary: ADM \(\simeq\) Splitting + Alternating

A simple yet effective approach to exploiting structures:
- bypasses non-differentiability
- enables very cheap iterations
- has at least an R-linear rate
- great versatility, good efficiency

Many issues remain. Convergence theory needs more work.
References on Codes


