Parallel, Block-Iterative, Primal-Dual Monotone Operator Splitting

Patrick L. Combettes

Department of Mathematics
North Carolina State University
Raleigh, NC 27695, USA

Joint work with Jonathan Eckstein, Rutgers University

Dedicated to the memory of Jon Borwein

Workshop on Splitting Algorithms and Modern Operator Theory
Oaxaca, Mexico, September 19, 2017

Monotone operators

- $\mathcal{H}$ a real Hilbert space.
- $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ is maximally monotone: for every $(x, x^*) \in \mathcal{H}^2$,

$$(x, x^*) \in \operatorname{gra} A \quad \Leftrightarrow \quad (\forall (y, y^*) \in \operatorname{gra} A) \quad \langle x - y \mid x^* - y^* \rangle \geqslant 0$$

- The resolvent of A, $J_A = (\mathrm{Id} + A)^{-1}\colon \mathcal{H} \to \mathcal{H}$, is firmly nonexpansive and $\operatorname{Fix} J_A = \operatorname{zer} A = \{x \in \mathcal{H} \mid 0 \in Ax\}$
- Minty's parametrization:

$$(\forall x \in \mathcal{H}) \quad (J_A x, x - J_A x) = (J_A x, J_{A^{-1}} x) \in \operatorname{gra} A$$

Solving monotone inclusions

- $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ maximally monotone,
- Problem: solve: $0 \in Ax$
- Conceptual solution methods when A is simple:
  - The proximal point algorithm (implicit):

$$x_{n+1} = (\mathrm{Id} + \gamma_n A)^{-1} x_n = J_{\gamma_n A} x_n, \quad \text{where} \quad \gamma_n > 0.$$

  - If $A\colon \mathcal{H} \to \mathcal{H}$ is β-cocoercive ($A^{-1} - \beta\mathrm{Id}$ is monotone), the explicit iteration

$$x_{n+1} = x_n - \gamma_n A x_n, \quad \text{where} \quad 0 < \gamma_n < 2\beta.$$

- For "real" problems **splitting** is required.

Splitting structured problems: 3 basic methods

$A, B\colon \mathcal{H} \to 2^{\mathcal{H}}$ maximally monotone, solve $0 \in A\overline{x} + B\overline{x}$.

- Douglas-Rachford splitting (1979)

$$\left\lfloor \begin{array}{l} y_n = J_{\gamma B} x_n \\ z_n = J_{\gamma A}(2y_n - x_n) \\ x_{n+1} = x_n + z_n - y_n \end{array} \right.$$

- $B\colon \mathcal{H} \to \mathcal{H}$ β-cocoercive: forward-backward splitting (1979+)

$$\left\lfloor \begin{array}{l} 0 < \gamma_n < 2\beta \\ y_n = x_n - \gamma_n B x_n \\ x_{n+1} = J_{\gamma_n A} y_n \end{array} \right.$$

- $B\colon \mathcal{H} \to \mathcal{H}$ μ-Lipschitzian: forward-backward-forward splitting (2000)

$$\left\lfloor \begin{array}{l} 0 < \gamma_n < 1/\mu \\ y_n = x_n - \gamma_n B x_n \\ z_n = J_{\gamma_n A} y_n \\ r_n = z_n - \gamma_n B z_n \\ x_{n+1} = x_n - y_n + r_n \end{array} \right.$$

- Spingarn's method (1983) for $0 \in A_1\overline{x} + \cdots + A_n\overline{x}$.

Splitting algorithms (1979-2000)

find $\overline{x} \in \mathcal{H}$ such that

$$z^* \in A\overline{x} + B\overline{x}$$

where:

- $z^* \in \mathcal{H}$, $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ is maximally monotone
- $B\colon \mathcal{H} \to 2^{\mathcal{H}}$ is maximally monotone

Splitting algorithms (Briceño-Arias-PLC, 2011)

find $\overline{x} \in \mathcal{H}$ such that

$$z^* \in A\overline{x} + L^* B(L\overline{x} - r)$$

where:

- $z^* \in \mathcal{H}$, $A \colon \mathcal{H} \to 2^{\mathcal{H}}$ is maximally monotone
- $B \colon \mathcal{G} \to 2^{\mathcal{G}}$ is maximally monotone, $r \in \mathcal{G}$, $L \in \mathscr{B}(\mathcal{H}, \mathcal{G})$

Splitting algorithms (Briceño-Arias-PLC, 2011)

find $\overline{x} \in \mathcal{H}$ such that

$$z^* \in A\overline{x} + \sum_{k=1}^{K} L_k^* B_k(L_k\overline{x} - r_k)$$

where:

- $z^* \in \mathcal{H}$, $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ is maximally monotone
- $B_k\colon \mathcal{G}_k \to 2^{\mathcal{G}_k}$ is maximally monotone, $r_k \in \mathcal{G}_k$, $L_k \in \mathscr{B}(\mathcal{H}, \mathcal{G}_k)$

Splitting algorithms (PLC-Pesquet, 2012)

find $\overline{x} \in \mathcal{H}$ such that

$$z^* \in A\overline{x} + \sum_{k=1}^{K} L_k^*(B_k \,\Box\, D_k)(L_k\overline{x} - r_k)$$

where:

- $z^* \in \mathcal{H}$, $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ is maximally monotone
- $B_k\colon \mathcal{G}_k \to 2^{\mathcal{G}_k}$ is maximally monotone, $r_k \in \mathcal{G}_k$, $L_k \in \mathscr{B}(\mathcal{H}, \mathcal{G}_k)$
- $D_k\colon \mathcal{G}_k \to 2^{\mathcal{G}_k}$ is maximally monotone, D_k^{-1} is ν_k-Lipschitzian, $B_k \,\Box\, D_k = (B_k^{-1} + D_k^{-1})^{-1}$

Splitting algorithms (PLC-Pesquet, 2012)

find $\overline{x} \in \mathcal{H}$ such that

$$z^* \in A\overline{x} + \sum_{k=1}^{K} L_k^*(B_k \,\square\, D_k)(L_k - r_k\overline{x}) + C\overline{x}$$

where:

- $z^* \in \mathcal{H}$, $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ is maximally monotone
- $B_k\colon \mathcal{G}_k \to 2^{\mathcal{G}_k}$ is maximally monotone, $r_k \in \mathcal{G}_k$, $L_k \in \mathscr{B}(\mathcal{H}, \mathcal{G}_k)$
- $D_k\colon \mathcal{G}_k \to 2^{\mathcal{G}_k}$ is maximally monotone, D_k^{-1} is ν_k-Lipschitzian, $B_k \,\square\, D_k = (B_k^{-1} + D_k^{-1})^{-1}$
- $C\colon \mathcal{H} \to \mathcal{H}$ is monotone and μ-Lipschtizian

Splitting algorithms (PLC, 2013)

find $\overline{x}_1 \in \mathcal{H}_1, \ldots, \overline{x}_m \in \mathcal{H}_m$ such that

$$\begin{cases} z_1^* & \in A_1\overline{x}_1 + \displaystyle\sum_{k=1}^{K} L_{k1}^*\Bigg((B_k \,\square\, D_k)\Bigg(\sum_{i=1}^{m} L_{ki}\overline{x}_i - r_k\Bigg)\Bigg) + C_1\overline{x}_1 \\ & \vdots \\ z_m^* & \in A_m\overline{x}_m + \displaystyle\sum_{k=1}^{K} L_{km}^*\Bigg((B_k \,\square\, D_k)\Bigg(\sum_{i=1}^{m} L_{ki}\overline{x}_i - r_k\Bigg)\Bigg) + C_m\overline{x}_m \end{cases}$$

where:

- $z_i^* \in \mathcal{H}_i$, $A_i \colon \mathcal{H}_i \to 2^{\mathcal{H}_i}$ is maximally monotone
- $B_k \colon \mathcal{G}_k \to 2^{\mathcal{G}_k}$ is maximally monotone, $r_k \in \mathcal{G}_k$, $L_{ki} \in \mathscr{B}(\mathcal{H}_i, \mathcal{G}_k)$
- $D_k \colon \mathcal{G}_k \to 2^{\mathcal{G}_k}$ is maximally monotone, D_k^{-1} is ν_k-Lipschitzian, $B_k \,\square\, D_k = (B_k^{-1} + D_k^{-1})^{-1}$
- $C_i \colon \mathcal{H}_i \to \mathcal{H}_i$ is monotone and μ_i-Lipschtizian

Splitting algorithms (PLC, 2013)

- $\mathcal{K} = \mathcal{H}_1 \oplus \cdots \oplus \mathcal{H}_m \oplus \mathcal{G}_1 \oplus \cdots \oplus \mathcal{G}_p$
- $\boldsymbol{M}\colon \mathcal{K} \to 2^{\mathcal{K}}\colon (x_1, \ldots, x_m, v_1^*, \ldots, v_p^*) \mapsto$ $(-z_1^* + A_1 x_1) \times \cdots \times (-z_m^* + A_m x_m) \times (r_1 + B_1^{-1} v_1^*) \times \cdots \times (r_p + B_p^{-1} v_p^*)$
- $\boldsymbol{Q}\colon \mathcal{K} \to \mathcal{K}\colon (x_1, \ldots, x_m, v_1^*, \ldots, v_p^*) \mapsto$ $\big(C_1 x_1 + \sum_{k=1}^K L_{k1}^* v_k^*, \ldots, C_m x_m + \sum_{k=1}^K L_{km}^* v_k^*, -\sum_{i=1}^m L_{1i} x_i + D_1^{-1} v_1^*, \ldots, \sum_{i=1}^m L_{Ki} x_i + D_K^{-1} v_K^*\big)$
- $\boldsymbol{M}$ and $\boldsymbol{Q}$ are maximally monotone, $\boldsymbol{Q}$ is Lipschitzian, the zeros of $\boldsymbol{M} + \boldsymbol{Q}$ are primal-dual solutions
- Solve $\mathbf{0} \in \boldsymbol{M}\boldsymbol{x} + \boldsymbol{Q}\boldsymbol{x}$, where $\boldsymbol{x} = (x_1, \ldots, x_m, v_1^*, \ldots, v_p^*)$ via Tseng's forward-backward-forward splitting algorithm

$$
\begin{array}{l}
\boldsymbol{y}_n = \boldsymbol{x}_n - \boldsymbol{Q}\boldsymbol{x}_n \\
\boldsymbol{p}_n = (\mathrm{Id} + \boldsymbol{M})^{-1} \boldsymbol{y}_n \\
\boldsymbol{q}_n = \boldsymbol{p}_n - \boldsymbol{Q}\boldsymbol{p}_n \\
\boldsymbol{x}_{n+1} = \boldsymbol{x}_n - \boldsymbol{y}_n + \boldsymbol{q}_n
\end{array}
$$

in $\mathcal{K}$ to get...

Splitting algorithms (PLC, 2013)

$$
\begin{array}{l}
\text{For } n = 0, 1, \ldots \\
\left\lfloor
\begin{array}{l}
\varepsilon \leqslant \gamma_n \leqslant (1-\varepsilon)/\left(\max\left\{\max\limits_{1\leqslant i\leqslant m} \mu_i, \max\limits_{1\leqslant k\leqslant K} \nu_k\right\} + \sqrt{\sum_{k=1}^K \sum_{i=1}^m \|L_{ki}\|^2}\right) \\
\text{For } i = 1, \ldots, m \\
\left\lfloor
\begin{array}{l}
s_{1,i,n} = x_{i,n} - \gamma_n\left(C_i x_{i,n} + \sum_{k=1}^K L_{ki}^* v_{k,n}^*\right) \\
p_{1,i,n} = J_{\gamma_n A_i}(s_{1,i,n} + \gamma_n z_i)
\end{array}
\right. \\
\text{For } k = 1, \ldots, K \\
\left\lfloor
\begin{array}{l}
s_{2,k,n} = v_{k,n}^* - \gamma_n\left(D_k^{-1} v_{k,n}^* - \sum_{i=1}^m L_{ki} x_{i,n}\right) \\
p_{2,k,n} = s_{2,k,n} - \gamma_n\left(r_k + J_{\gamma_n^{-1} B_k}(\gamma_n^{-1} s_{2,k,n} - r_k)\right) \\
q_{2,k,n} = p_{2,k,n} - \gamma_n\left(D_k^{-1} p_{2,k,n} - \sum_{i=1}^m L_{ki} p_{1,i,n}\right) \\
v_{k,n+1}^* = v_{k,n}^* - s_{2,k,n} + q_{2,k,n}
\end{array}
\right. \\
\text{For } i = 1, \ldots, m \\
\left\lfloor
\begin{array}{l}
q_{1,i,n} = p_{1,i,n} - \gamma_n\left(C_i p_{1,i,n} + \sum_{k=1}^K L_{ki}^* p_{2,k,n}\right) \\
x_{i,n+1} = x_{i,n} - s_{1,i,n} + q_{1,i,n}
\end{array}
\right.
\end{array}
\right.
\end{array}
$$

Some limitations of the state-of-the-art

We present a new framework that circumvents simultaneously the limitations of current methods, which require:

- inversions of linear operators or knowledge of bounds on norms of all the L_{ki}
- the proximal parameters must be the same for all the monotone operators
- activation of the resolvents of all the monotone operators: impossible in huge-scale problems
- synchronicity: all resolvent operator evaluations must be computed and used during the current iteration

and, in general,

- converge only weakly

Asynchronous, block-iterative splitting

- For every $i \in I$ (finite), $\mathcal{H}_i$ a Hilbert space, $A_i \colon \mathcal{H}_i \to 2^{\mathcal{H}_i}$ maximally monotone, $z_i^* \in \mathcal{H}_i$
- For every $k \in K$ (finite), $\mathcal{G}_k$ a Hilbert space, $B_k \colon \mathcal{G}_k \to 2^{\mathcal{G}_k}$ maximally monotone, $r_k \in \mathcal{G}_k$, $\mathrm{L}_{ki} \in \mathscr{B}(\mathcal{H}_i, \mathcal{G}_k)$
- **Initial problem:** find $(\overline{x}_i)_{i\in I} \in \boldsymbol{\mathcal{H}} = \bigoplus_{i\in I} \mathcal{H}_i$ such that

$$(\forall i \in I) \quad z_i^* \in A_i\overline{x}_i + \sum_{k\in K} L_{ki}^*\Bigg(B_k\Bigg(\sum_{j\in I} L_{kj}\overline{x}_j - r_k\Bigg)\Bigg)$$

Asynchronous, block-iterative splitting

- For every $i \in I$ (finite), $\mathcal{H}_i$ a Hilbert space, $A_i \colon \mathcal{H}_i \to 2^{\mathcal{H}_i}$ maximally monotone, $z_i^* \in \mathcal{H}_i$
- For every $k \in K$ (finite), $\mathcal{G}_k$ a Hilbert space, $B_k \colon \mathcal{G}_k \to 2^{\mathcal{G}_k}$ maximally monotone, $r_k \in \mathcal{G}_k$, $\mathrm{L}_{ki} \in \mathscr{B}(\mathcal{H}_i, \mathcal{G}_k)$
- **Initial problem:** find $(\overline{x}_i)_{i\in I} \in \boldsymbol{\mathcal{H}} = \bigoplus_{i\in I} \mathcal{H}_i$ such that

$$(\forall i \in I) \quad z_i^* \in A_i\overline{x}_i + \sum_{k\in K} L_{ki}^*\Bigg(B_k\bigg(\sum_{j\in I} L_{kj}\overline{x}_j - r_k\bigg)\Bigg)$$

- **Dual problem:** find $(\overline{v}_k^*)_{k\in K} \in \boldsymbol{\mathcal{G}} = \bigoplus_{k\in K} \mathcal{G}_k$ such that

$$(\forall k \in K) \quad -r_k \in -\sum_{i\in I} L_{ki}\Bigg(A_i^{-1}\bigg(z_i^* - \sum_{l\in K} L_{li}^*\overline{v}_l^*\bigg)\Bigg) + B_k^{-1}\overline{v}_k^*$$

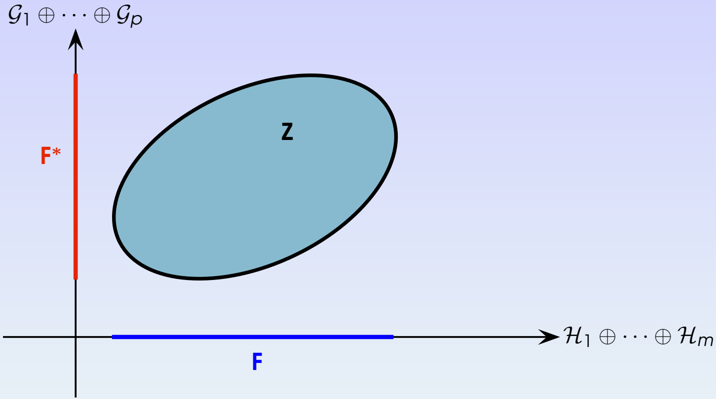
Asynchronous, block-iterative splitting

- **Solutions set:** the associated Kuhn-Tucker set

$$\boldsymbol{Z} = \Big\{ \big((\overline{x}_i)_{i\in I}, (\overline{v}_k^*)_{k\in K}\big) \;\Big|\; \overline{x}_i \in \mathcal{H}_i \;\text{ and }\; z_i^* - \sum_{k\in K} L_{ki}^*\overline{v}_k^* \in A_i\overline{x}_i, \quad \overline{v}_k^* \in \mathcal{G}_k \;\text{ and }\; \sum_{i\in I} L_{ki}\overline{x}_i - r_k \in B_k^{-1}\overline{v}_k^* \Big\}$$

- $\boldsymbol{Z}$ is a closed convex set
- The projection of $\boldsymbol{Z}$ onto $\boldsymbol{\mathcal{H}}$ is the set $\boldsymbol{F}$ of primal solutions
- The projection of $\boldsymbol{Z}$ onto $\boldsymbol{\mathcal{G}}$ is the set $\boldsymbol{F}^*$ of dual solutions

The Kuhn-Tucker set



With proper CQ, this framework includes..

- Let $\boldsymbol{F}$ be the set of solutions to the problem

$$\underset{(x_i)_{i\in I}\in\boldsymbol{\mathcal{H}}}{\text{minimize}} \sum_{i\in I}\big(f_i(x_i)-\langle x_i \mid z_i^*\rangle\big)+\sum_{k\in K} g_k\Bigg(\sum_{i\in I} L_{ki}x_i - r_k\Bigg)$$

where $f_i \in \Gamma_0(\mathcal{H}_i)$, $g_k \in \Gamma_0(\mathcal{G}_k)$, $L_{ki} \in \mathscr{B}(\mathcal{H}_i,\mathcal{G}_k)$

- Let $\boldsymbol{F}^*$ be the set of solutions to the dual problem

$$\underset{(v_k^*)_{k\in K}\in\bigoplus_{k\in K}\mathcal{G}_k}{\text{minimize}} \sum_{i\in I} f_i^*\Bigg(z_i^* - \sum_{k\in K} L_{ki}^* v_k^*\Bigg)+\sum_{k\in K}\big(g_k^*(v_k^*)+\langle v_k^* \mid r_k\rangle\big)$$

- Associated Kuhn-Tucker set

$$\boldsymbol{Z}=\Bigg\{\big((\overline{x}_i)_{i\in I},(\overline{v}_k^*)_{k\in K}\big)\;\Bigg|\;\overline{x}_i\in\mathcal{H}_i \text{ and } z_i^*-\sum_{k\in K}L_{ki}^*\overline{v}_k^*\in\partial f_i(\overline{x}_i), \quad \overline{v}_k^*\in\mathcal{G}_k \text{ and } \sum_{i\in I}L_{ki}\overline{x}_i-r_k\in\partial g_k^*(\overline{v}_k^*)\Bigg\}$$

Methodology for 2 operators

- $0 \in A\overline{x} + L^*\big(BL\overline{x}\big)$ and $0 \in -L\big(A^{-1}\big(-L^*\overline{v}^*\big)\big) + B^{-1}\overline{v}^*$

Methodology for 2 operators

- $0 \in A\overline{x} + L^*\big(BL\overline{x}\big)$ and $0 \in -L\big(A^{-1}\big(-L^*\overline{v}^*\big)\big) + B^{-1}\overline{v}^*$
- Take $(\overline{x}, \overline{v}^*) \in \boldsymbol{Z}$. Then $-L^*\overline{v}^* \in A\overline{x}$ and $L\overline{x} \in B^{-1}\overline{v}^*$, i.e.,

$$\big(\overline{x}, -L^*\overline{v}^*\big) \in \operatorname{gra} A \quad \text{and} \quad \big(L\overline{x}, \overline{v}^*\big) \in \operatorname{gra} B$$

Methodology for 2 operators

- $0 \in A\overline{x} + L^*\big(BL\overline{x}\big)$ and $0 \in -L\big(A^{-1}\big(-L^*\overline{v}^*\big)\big) + B^{-1}\overline{v}^*$
- Take $(\overline{x}, \overline{v}^*) \in \boldsymbol{Z}$. Then $-L^*\overline{v}^* \in A\overline{x}$ and $L\overline{x} \in B^{-1}\overline{v}^*$, i.e.,

$$\big(\overline{x}, -L^*\overline{v}^*\big) \in \operatorname{gra} A \quad \text{and} \quad \big(L\overline{x}, \overline{v}^*\big) \in \operatorname{gra} B$$

- Suppose that at iteration n

$$(a_n, a_n^*) \in \operatorname{gra} A \quad \text{and} \quad (b_n, b_n^*) \in \operatorname{gra} B$$

Methodology for 2 operators

- $0 \in A\overline{x} + L^*\big(BL\overline{x}\big)$ and $0 \in -L\big(A^{-1}\big(-L^*\overline{v}^*\big)\big) + B^{-1}\overline{v}^*$
- Take $(\overline{x}, \overline{v}^*) \in \boldsymbol{Z}$. Then $-L^*\overline{v}^* \in A\overline{x}$ and $L\overline{x} \in B^{-1}\overline{v}^*$, i.e.,

$$\big(\overline{x}, -L^*\overline{v}^*\big) \in \operatorname{gra} A \quad \text{and} \quad \big(L\overline{x}, \overline{v}^*\big) \in \operatorname{gra} B$$

- Suppose that at iteration n

$$(a_n, a_n^*) \in \operatorname{gra} A \quad \text{and} \quad (b_n, b_n^*) \in \operatorname{gra} B$$

- By monotonicity of A and B,

$$\langle a_n - \overline{x} \mid a_n^* + L^*\overline{v}^*\rangle + \langle b_n - L\overline{x} \mid b_n^* - \overline{v}^*\rangle \geqslant 0$$

i.e.,

$$\langle (\overline{x}, \overline{v}) \mid (a_n^* + L^* b_n^*, b_n - La_n)\rangle \leqslant \langle a_n \mid a_n^*\rangle + \langle b_n \mid b_n^*\rangle$$

Methodology for 2 operators

- $0 \in A\overline{x} + L^*\big(BL\overline{x}\big)$ and $0 \in -L\big(A^{-1}\big(-L^*\overline{v}^*\big)\big) + B^{-1}\overline{v}^*$
- Take $(\overline{x}, \overline{v}^*) \in \boldsymbol{Z}$. Then $-L^*\overline{v}^* \in A\overline{x}$ and $L\overline{x} \in B^{-1}\overline{v}^*$, i.e.,

$$\big(\overline{x}, -L^*\overline{v}^*\big) \in \operatorname{gra} A \quad \text{and} \quad \big(L\overline{x}, \overline{v}^*\big) \in \operatorname{gra} B$$

- Suppose that at iteration n

$$(a_n, a_n^*) \in \operatorname{gra} A \quad \text{and} \quad (b_n, b_n^*) \in \operatorname{gra} B$$

- By monotonicity of A and B,

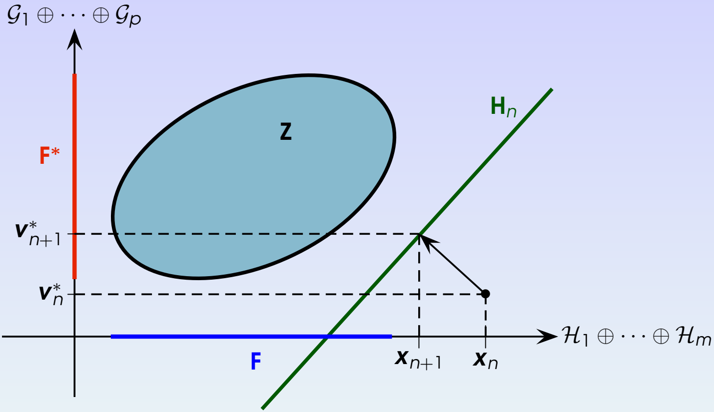
$$\langle a_n - \overline{x} \mid a_n^* + L^*\overline{v}^* \rangle + \langle b_n - L\overline{x} \mid b_n^* - \overline{v}^* \rangle \geqslant 0$$

i.e.,

$$\langle (\overline{x}, \overline{v}) \mid (a_n^* + L^* b_n^*, b_n - La_n) \rangle \leqslant \langle a_n \mid a_n^* \rangle + \langle b_n \mid b_n^* \rangle$$

- This places $\boldsymbol{Z}$ in the closed affine half-space with outer normal $(a_n^* + L^* b_n^*, b_n - La_n)$!

Fejér monotone scheme in the general case



- Choose suitable points in the graphs of $(A_i)_{i\in I}$ and $(B_k)_{k\in K}$ to construct a half-space $\mathbf{H}_n$ containing $\mathbf{Z}$
- Algorithm: $(\boldsymbol{x}_{n+1}, \boldsymbol{v}^*_{n+1}) = P_{\mathbf{H}_n}(\boldsymbol{x}_n, \boldsymbol{v}^*_n) \rightharpoonup (\boldsymbol{x}, \boldsymbol{v}^*) \in \mathbf{Z} \subset \mathbf{F} \times \mathbf{F}^*$

Main novelties

- **Block iterations:** At iteration n, we require calculation of new points in the graphs of only some the operators $(A_i)_{i\in I_n\subset I}$ and $(B_k)_{k\in K_n\subset K}$. The control sequences $(I_n)_{n\in\mathbb{N}}$ and $(K_n)_{n\in\mathbb{N}}$ dictate how frequently the various operators are used.
- **Asynchronicity:** A new point $(a_{i,n}, a^*_{i,n}) \in \operatorname{gra} A_i$ being incorporated into the calculations at iteration n may be based on data $x_{i,c_i(n)}$ and $(v^*_{k,c_i(n)})_{k\in K}$ available at some possibly earlier iteration $c_i(n) \leqslant n$. Therefore, the calculation of $(a_{i,n}, a^*_{i,n})$ could have been initiated at iteration $c_i(n)$, with its results becoming available only at iteration n. Likewise, for $(b_{k,n}, b^*_{k,n}) \in \operatorname{gra} B_k$.

Also:

- No knowledge of the $\|L_{ki}\|$s is required
- No linear operator inversion is required
- No bounds required on the proximal parameters

Asynchronous block-iterative proximal splitting I

for $n = 0, 1, \ldots$

- for every $i \in I_n$
  - $l^*_{i,n} = \sum_{k\in K} L^*_{ki} v^*_{k,c_i(n)}$
  - $(a_{i,n}, a^*_{i,n}) = \Big(J_{\gamma_{i,c_i(n)} A_i}\big(x_{i,c_i(n)} + \gamma_{i,c_i(n)}(z_i - l^*_{i,n})\big), \gamma^{-1}_{i,c_i(n)}(x_{i,c_i(n)} - a_{i,n}) - l^*_{i,n}\Big)$
- for every $i \in I \smallsetminus I_n$
  - $(a_{i,n}, a^*_{i,n}) = (a_{i,n-1}, a^*_{i,n-1})$
- for every $k \in K_n$
  - $l_{k,n} = \sum_{i\in I} L_{ki} x_{i,d_k(n)}$
  - $(b_{k,n}, b^*_{k,n}) = \Big(r_k + J_{\mu_{k,d_k(n)} B_k}\big(l_{k,n} + \mu_{k,d_k(n)} v^*_{k,d_k(n)} - r_k\big), v^*_{k,d_k(n)} + \mu^{-1}_{k,d_k(n)}(l_{k,n} - b_{k,n})\Big)$
- for every $k \in K \smallsetminus K_n$
  - $(b_{k,n}, b^*_{k,n}) = (b_{k,n-1}, b^*_{k,n-1})$
- $\big((t^*_{i,n})_{i\in I}, (t_{k,n})_{k\in K}\big) = \big((a^*_{i,n} + \sum_{k\in K} L^*_{ki} b^*_{k,n})_{i\in I}, (b_{k,n} - \sum_{i\in I} L_{ki} a_{i,n})_{k\in K}\big)$
- $\tau_n = \sum_{i\in I} \|t^*_{i,n}\|^2 + \sum_{k\in K} \|t_{k,n}\|^2$
- if $\tau_n > 0$
  - $$\theta_n = \frac{\lambda_n}{\tau_n} \max\Big\{0, \sum_{i\in I} \big(\langle x_{i,n} \mid t^*_{i,n}\rangle - \langle a_{i,n} \mid a^*_{i,n}\rangle\big) + \sum_{k\in K} \big(\langle t_{k,n} \mid v^*_{k,n}\rangle - \langle b_{k,n} \mid b^*_{k,n}\rangle\big)\Big\}$$
- else $\theta_n = 0$
- for every $i \in I$
  - $x_{i,n+1} = x_{i,n} - \theta_n t^*_{i,n}$
- for every $k \in K$
  - $v^*_{k,n+1} = v^*_{k,n} - \theta_n t_{k,n}$

Convergence

- $(I_n)_{n\in\mathbb{N}}$ is a sequence of nonempty subsets of I, and $(K_n)_{n\in\mathbb{N}}$ is a sequence of nonempty subsets of K such that $I_0 = I$, $K_0 = K$, and

$$(\forall n \in \mathbb{N}) \quad \left(\bigcup_{j=n}^{n+M-1} I_j = I \quad \text{and} \quad \bigcup_{j=n}^{n+M-1} K_j = K \right). \tag{1}$$

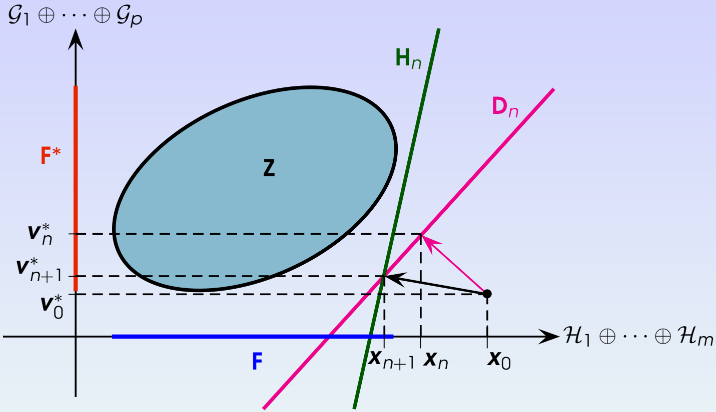
- $(c_i(n))_{n\in\mathbb{N}}$ and $(d_k(n))_{n\in\mathbb{N}}$ are sequences in $\mathbb{N}$ such that

$$(\forall i \in I) \;\; n - D \leqslant c_i(n) \leqslant n \quad \text{and} \quad (\forall k \in K) \;\; n - D \leqslant d_k(n) \leqslant n$$

- $\varepsilon \in \,]0,1[$ and $(\gamma_{i,n})_{n\in\mathbb{N}}$ and $(\mu_{k,n})_{n\in\mathbb{N}}$ are sequences in $[\varepsilon, 1/\varepsilon]$.

Set $x_n = (x_{i,n})_{i\in I}$ and $v_n^* = (v_{k,n}^*)_{k\in K}$. Then $(x_n)_{n\in\mathbb{N}}$ converges weakly to a point $\overline{x} \in \boldsymbol{F}$, $(v_n^*)_{n\in\mathbb{N}}$ converges weakly to a point $\overline{v} \in \boldsymbol{F}^*$, and $(\overline{x}, \overline{v}^*) \in \boldsymbol{Z}$.

Asynchronous block-iterative proximal splitting II



- Construct $\mathbf{H}_n$ as before
- The half-space $\mathbf{D}_n$ satisfies $(\boldsymbol{x}_n, \boldsymbol{v}_n^*) = P_{\mathbf{D}_n}(\boldsymbol{x}_0, \boldsymbol{v}_0^*)$
- Algorithm: $(\boldsymbol{x}_{n+1}, \boldsymbol{v}_{n+1}^*) = P_{\mathbf{H}_n \cap \mathbf{D}_n}(\boldsymbol{x}_0, \boldsymbol{v}_0^*) \to P_{\mathbf{Z}}(\boldsymbol{x}_0, \boldsymbol{v}_0^*) \in \mathbf{F} \times \mathbf{F}^*$

References

- PLC and J. Eckstein, Asynchronous block-iterative primal-dual decomposition methods for monotone inclusions, *Math. Programing*, published on line.
- A. Alotaibi, PLC, and N. Shahzad, Solving coupled composite monotone inclusions by successive Fejér approximations of their Kuhn-Tucker set, *SIAM J. Optim.*, vol. 24, 2014.
- H. Attouch, L. M. Briceño-Arias, PLC, A strongly convergent primal-dual method for nonoverlapping domain decomposition, *Numer. Math.*, vol. 133, 2016.
- L. M. Briceño-Arias and PLC, A monotone+skew splitting model for composite monotone inclusions in duality, *SIAM J. Optim.*, vol. 21, 2011.
- PLC and J.-C. Pesquet, Primal-dual splitting algorithm for solving inclusions with mixtures of composite, Lipschitzian, and parallel-sum type monotone operators, *Set-Valued Var. Anal.*, vol. 20, 2012.
- H. H. Bauschke and PLC, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces.* **2nd ed.**, Springer, New York, 2017.