Can One Genuinely Split $m > 2$ Monotone Operators?

P. L. Combettes

Laboratoire Jacques-Louis Lions
Faculté de Mathématiques
Université Pierre et Marie Curie – Paris 6
75005 Paris, France

Playa Blanca – 14 Octubre 2013

Notation

- $\mathcal{H}$, $\mathcal{H}_i$, $\mathcal{G}$, $\mathcal{G}_i$: real Hilbert spaces.
- $\mathcal{B}(\mathcal{H},\mathcal{G})$ bounded linear operators from $\mathcal{H}$ to $\mathcal{G}$.
- $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ a set-valued operator.
- Graph of A: $\operatorname{gra} A = \big\{(x,u) \in \mathcal{H} \times \mathcal{H} \mid u \in Ax\big\}$.
- Zeros of A: $\operatorname{zer} A = \big\{x \in \mathcal{H} \mid 0 \in Ax\big\}$.
- Inverse of A: $\operatorname{gra} A^{-1} = \big\{(u,x) \in \mathcal{H} \times \mathcal{H} \mid u \in Ax\big\}$.
- Resolvent of A:
$$J_A = (\mathrm{Id} + A)^{-1}.$$
- Parallel sum of A and B: $A \,\Box\, B = (A^{-1} + B^{-1})^{-1}$.

Monotone operators

- $A\colon \mathcal{H} \to 2^{\mathcal{H}}$ is *monotone* if

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

  and *maximally monotone* if there exists no monotone operator $B\colon \mathcal{H} \to 2^{\mathcal{H}}$ such that $\operatorname{gra} A \subset \operatorname{gra} B \neq \operatorname{gra} A$.

- If A is maximally monotone, its resolvent $J_A = (\mathrm{Id} + A)^{-1}$ is single-valued, defined everywhere (Minty), and *firmly nonexpansive*:

$$\|J_A x - J_A y\|^2 + \|(\mathrm{Id} - J_A)x - (\mathrm{Id} - J_A)y\|^2 \leqslant \|x - y\|^2.$$

  Moreover,

$$J_A + J_{A^{-1}} = \mathrm{Id} \quad \text{and} \quad \operatorname{Fix} J_A = \operatorname{zer}(A).$$

- H. H. Bauschke and PLC, *Convex Analysis and Monotone Operator Theory in Hilbert Spaces*, Springer, 2011.

The proximal point algorithm

- Many problems in nonlinear analysis can be reduced to

$$\text{find } \; x \in \operatorname{zer} C, \quad \text{where } \; C\colon \mathcal{H} \to 2^{\mathcal{H}} \text{ is maximally monotone.}$$

The proximal point algorithm

- Many problems in nonlinear analysis can be reduced to

$$\text{find } \ x \in \operatorname{zer} C, \quad \text{where } \ C\colon \mathcal{H} \to 2^{\mathcal{H}} \text{ is maximally monotone.}$$

- This inclusion can be solved by the proximal point algorithm

$$x_{n+1} = J_{\gamma_n C} x_n, \tag{1}$$

where $(\gamma_n)_{n\in\mathbb{N}}$ lies in $]0,+\infty[$ and $\sum_{n\in\mathbb{N}} \gamma_n^2 = +\infty$.

  - H. Brézis and P.-L. Lions, Produits infinis de résolvantes, *Israel J. Math.,* vol. 29, pp. 329-345, 1978.

The proximal point algorithm

- Many problems in nonlinear analysis can be reduced to

  find $x \in \operatorname{zer} C$, where $C\colon \mathcal{H} \to 2^{\mathcal{H}}$ is maximally monotone.

- This inclusion can be solved by the proximal point algorithm

$$x_{n+1} = J_{\gamma_n C} x_n, \tag{1}$$

  where $(\gamma_n)_{n\in\mathbb{N}}$ lies in $]0,+\infty[$ and $\sum_{n\in\mathbb{N}} \gamma_n^2 = +\infty$.

  - H. Brézis and P.-L. Lions, Produits infinis de résolvantes, *Israel J. Math.,* vol. 29, pp. 329-345, 1978.

- Unfortunately, in most situations, (1) is not implementable because the resolvents of C are too hard to compute.

The proximal point algorithm

- Many problems in nonlinear analysis can be reduced to

$$\text{find} \quad x \in \operatorname{zer} C, \qquad \text{where} \quad C \colon \mathcal{H} \to 2^{\mathcal{H}} \text{ is maximally monotone.}$$

- This inclusion can be solved by the proximal point algorithm

$$x_{n+1} = J_{\gamma_n C} x_n, \tag{1}$$

where $(\gamma_n)_{n\in\mathbb{N}}$ lies in $]0,+\infty[$ and $\sum_{n\in\mathbb{N}} \gamma_n^2 = +\infty$.

  - H. Brézis and P.-L. Lions, Produits infinis de résolvantes, *Israel J. Math.,* vol. 29, pp. 329-345, 1978.

- Unfortunately, in most situations, (1) is not implementable because the resolvents of C are too hard to compute.
- **Splitting methods:** Decompose C in terms of operators which are simpler (i.e., they can be used explicitly or have easily computable resolvents), and devise an algorithm which employs these operators individually.

Splitting methods: Some hard facts of life

- One knows how to split only two operators: $0 \in Ax + Bx$.

Splitting methods: Some hard facts of life

- One knows how to split only two operators: $0 \in Ax + Bx$.
- There exist only only three splitting schemes.

Splitting methods: Some hard facts of life

- One knows how to split only two operators: $0 \in Ax + Bx$.
- There exist only only three splitting schemes.
- Yet, we want to solve systems of monotone inclusions such as

find $\overline{x_1} \in \mathcal{H}_1, \ldots, \overline{x_m} \in \mathcal{H}_m$ such that

$$\begin{cases} z_1 & \in A_1\overline{x_1} + \displaystyle\sum_{k=1}^{K} L_{k1}^* \bigg((B_k \,\square\, D_k) \Big(\sum_{i=1}^{m} L_{ki}\overline{x_i} - r_k \Big) \bigg) + C_1\overline{x_1} \\ & \vdots \\ z_m & \in A_m\overline{x_m} + \displaystyle\sum_{k=1}^{K} L_{km}^* \bigg((B_k \,\square\, D_k) \Big(\sum_{i=1}^{m} L_{ki}\overline{x_i} - r_k \Big) \bigg) + C_m\overline{x_m}, \end{cases}$$

for instance [inf-convolution: $g_k \,\square\, \ell_k \colon y \mapsto \inf_t g_k(t) + \ell_k(y-t)$]

$$\underset{x_1 \in \mathcal{H}_1, \ldots, x_m \in \mathcal{H}_m}{\text{minimize}} \sum_{i=1}^{m} f_i(x_i) + \sum_{k=1}^{K} (g_k \,\square\, \ell_k) \Big(\sum_{i=1}^{m} L_{ki}x_i - r_k \Big) + \sum_{i=1}^{m} h_i(x_i) - \langle x_i \mid z_i \rangle$$

Early example: Legendre's method of least squares

- Set $m = 1$, $z_1 = 0$, $\mathcal{H}_1 = \mathbb{R}^N$, $L_{k1} = \mathrm{Id}$, $A_1 = C_1 = 0$, $D_k = \mathrm{Id}$, and

$$B_k \colon x \mapsto \begin{cases} \operatorname{span}\{u_k\}, & \text{if } \langle x \mid u_k\rangle = \rho_k; \\ \varnothing, & \text{if } \langle x \mid u_k\rangle \neq \rho_k, \end{cases} \quad \text{where} \quad \begin{cases} u_k \in \mathbb{R}^N \\ \|u_k\| = 1 \\ \rho_k \in \mathbb{R}. \end{cases}$$

  Then the problem becomes

$$\underset{x\in\mathbb{R}^N}{\text{minimize}} \sum_{k=1}^{m} |\langle x \mid u_k\rangle - \rho_k|^2,$$

  which is precisely Legendre's least squares method for solving the overdetermined system $\langle x \mid u_k\rangle = \rho_k$, $1 \leqslant k \leqslant K$.

  - A. M. Legendre, *Nouvelles Méthodes pour la Détermination de l'Orbite des Comètes.* Courcier, Paris, 1805.
  - C. F. Gauss, *Theoria Motus Corporum Coelestium.* Perthes and Besser, Hamburg, 1809.

Basic splitting schemes for $0 \in Ax + Bx$

- **Douglas-Rachford algorithm:** $\gamma \in \left]0, +\infty\right[$.
  - $\operatorname{zer}(A+B) = J_{\gamma B}\Big(\operatorname{Fix}\big(\frac{1}{2}((2J_{\gamma A} - \operatorname{Id}) \circ (2J_{\gamma B} - \operatorname{Id}) + \operatorname{Id})\big)\Big)$.
  - Iterate

$$
\left\lfloor
\begin{array}{lll}
x_n & = J_{\gamma B} y_n & \text{(backward step)} \\
y_{n+1} & = J_{\gamma A}\big(2x_n - y_n\big) + y_n - x_n & \text{(backward step)}
\end{array}
\right.
$$

  Then $y_n \rightharpoonup y$ and $z = J_{\gamma B} y \in \operatorname{zer}(A+B)$ (Lions&Mercier, **1979**), and $x_n \rightharpoonup z \in \operatorname{zer}(A+B)$.
  - ADMM, method of partial inverses are essentially special cases.
  - There are tricks to reduce m-operator problems to 2-operator problems in product spaces [Spingarn (1983), PLC (2009), Briceño-PLC (2011)] and use Douglas-Rachford splitting.

Basic splitting schemes for $0 \in Ax + Bx$

- **Forward-Backward algorithm:** $\gamma \in \,]0, +\infty[$.
  - $B\colon \mathcal{H} \to \mathcal{H}$ is β-cocoercive: $\langle x - y \mid Bx - By \rangle \geqslant \beta \|Bx - By\|^2$; $\gamma \in \,]0, 2\beta[$.
  - $\operatorname{zer}(A + B) = \operatorname{Fix}\Big(J_{\gamma A}(\operatorname{Id} - \gamma B)\Big)$.
  - Iterate

$$
\left\lfloor
\begin{array}{lll}
y_n & = x_n - \gamma B x_n & \text{(forward step)} \\
x_{n+1} & = J_{\gamma A} y_n & \text{(backward step)}
\end{array}
\right.
$$

  Then $x_n \rightharpoonup z \in \operatorname{zer}(A + B)$ (Mercier, **1979**)
  - There are tricks to use the forward-backward algorithm (on the dual problem if the primal is strongly monotone, in primal-dual spaces, in renormed spaces) to solve *m*-operator problems; see [PLC&Vũ, (2013)]

Basic splitting schemes for $0 \in Ax + Bx$

- **Forward-Backward-Forward algorithm:** $\gamma \in]0, +\infty[$.
  - $\mathrm{zer}(A + B) = \mathrm{Fix}\,\Big(J_{\gamma A}(\mathrm{Id} - \gamma B)\Big)$.
  - $B\colon \mathcal{H} \to \mathcal{H}$ is $1/\beta$-Lipschitzian; $0 < \gamma_n < \beta$.
  - Iterate

$$\left\lfloor \begin{array}{lcll} y_n & = & x_n - \gamma B x_n & \text{(forward step)} \\ p_n & = & J_{\gamma A}\, y_n & \text{(backward step)} \\ q_n & = & p_n - \gamma B p_n & \text{(forward step)} \\ x_{n+1} & = & x_n - y_n + q_n & \end{array} \right.$$

  Then $x_n \rightharpoonup z \in \mathrm{zer}(A + B)$ [Tseng (2000)]
  - There are tricks to use the forward-backward-forward algorithm to obtain fully split algorithms for rather complex structured monotone inclusion problems, such as...

Multivariate structured inclusion problem

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

$$z \in Ax + Bx \tag{2}$$

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

Multivariate structured inclusion problem

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

$$z \in Ax + L^* B(Lx - r) \tag{2}$$

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, $r \in \mathcal{G}$, $L \in \mathcal{B}(\mathcal{H}, \mathcal{G})$

Multivariate structured inclusion problem

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

$$z \in Ax + \sum_{k=1}^{K} L_k^* B_k(L_k x - r_k) \tag{2}$$

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, $r_k \in \mathcal{G}_k$, $L_k \in \mathfrak{B}(\mathcal{H}, \mathcal{G}_k)$

Multivariate structured inclusion problem

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

$$z \in Ax + \sum_{k=1}^{K} L_k^*(B_k \,\square\, D_k)(L_k - r_k x) \tag{2}$$

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

Multivariate structured inclusion problem

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

$$z \in Ax + \sum_{k=1}^{K} L_k^*(B_k \,\square\, D_k)(L_k - r_k x) + Cx \tag{2}$$

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, $r_k \in \mathcal{G}_k$, $L_k \in \mathcal{B}(\mathcal{H}, \mathcal{G}_k)$
- $D_k \colon \mathcal{G}_k \to 2^{\mathcal{G}_k}$ 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

Multivariate structured inclusion problem

find $\overline{x_1} \in \mathcal{H}_1, \ldots, \overline{x_m} \in \mathcal{H}_m$ such that

$$\begin{cases} z_1 & \in A_1\overline{x_1} + \sum_{k=1}^{K} L_{k1}^*\Bigg((B_k \,\square\, D_k)\Bigg(\sum_{i=1}^{m} L_{ki}\overline{x_i} - r_k\Bigg)\Bigg) + C_1\overline{x_1} \\ \quad\vdots & \\ z_m & \in A_m\overline{x_m} + \sum_{k=1}^{K} L_{km}^*\Bigg((B_k \,\square\, D_k)\Bigg(\sum_{i=1}^{m} L_{ki}\overline{x_i} - r_k\Bigg)\Bigg) + C_m\overline{x_m} \end{cases} \tag{2}$$

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

Multivariate structured inclusion problem

- **Primal problem:**

find $\overline{x_1} \in \mathcal{H}_1, \ldots, \overline{x_m} \in \mathcal{H}_m$ such that

$$\begin{cases} z_1 \in A_1\overline{x_1} + \sum_{k=1}^{K} L_{k1}^* \Big((B_k \,\square\, D_k) \Big(\sum_{i=1}^{m} L_{ki}\overline{x_i} - r_k \Big) \Big) + C_1\overline{x_1} \\ \quad\vdots \\ z_m \in A_m\overline{x_m} + \sum_{k=1}^{K} L_{km}^* \Big((B_k \,\square\, D_k) \Big(\sum_{i=1}^{m} L_{ki}\overline{x_i} - r_k \Big) \Big) + C_m\overline{x_m}, \end{cases}$$

- **Dual problem:**

find $\overline{v_1} \in \mathcal{G}_1, \ldots, \overline{v_K} \in \mathcal{G}_K$ such that

$$\begin{cases} -r_1 \in -\sum_{i=1}^{m} L_{1i}(A_i + C_i)^{-1} \Big(z_i - \sum_{k=1}^{K} L_{ki}^*\overline{v_k} \Big) + B_1^{-1}\overline{v_1} + D_1^{-1}\overline{v_1} \\ \quad\vdots \\ -r_K \in -\sum_{i=1}^{m} L_{Ki}(A_i + C_i)^{-1} \Big(z_i - \sum_{k=1}^{K} L_{ki}^*\overline{v_K} \Big) + B_K^{-1}\overline{v_K} + D_K^{-1}\overline{v_K}. \end{cases}$$

- PLC, Systems of structured monotone inclusions: Duality, algorithms, and applications, *SIAM J. Optim.*, to appear.

Reformulation in primal-dual space

- $\boldsymbol{\mathcal{H}} = \mathcal{H}_1 \oplus \cdots \oplus \mathcal{H}_m$, $\boldsymbol{\mathcal{G}} = \mathcal{G}_1 \oplus \cdots \oplus \mathcal{G}_K$, $\boldsymbol{\mathcal{K}} = \boldsymbol{\mathcal{H}} \oplus \boldsymbol{\mathcal{G}}$

Reformulation in primal-dual space

- $\boldsymbol{\mathcal{H}} = \mathcal{H}_1 \oplus \cdots \oplus \mathcal{H}_m$, $\boldsymbol{\mathcal{G}} = \mathcal{G}_1 \oplus \cdots \oplus \mathcal{G}_K$, $\boldsymbol{\mathcal{K}} = \boldsymbol{\mathcal{H}} \oplus \boldsymbol{\mathcal{G}}$
- $\boldsymbol{A} \colon \boldsymbol{\mathcal{H}} \to 2^{\boldsymbol{\mathcal{H}}} \colon \boldsymbol{x} \mapsto \bigtimes_{i=1}^{m} A_i x_i$, $\boldsymbol{C} \colon \boldsymbol{\mathcal{H}} \to \boldsymbol{\mathcal{H}} \colon \boldsymbol{x} \mapsto (C_i x_i)_{1 \leqslant i \leqslant m}$
- $\boldsymbol{B} \colon \boldsymbol{\mathcal{G}} \to 2^{\boldsymbol{\mathcal{G}}} \colon \boldsymbol{v} \mapsto \bigtimes_{k=1}^{K} B_k v_k$, $\boldsymbol{D} \colon \boldsymbol{\mathcal{G}} \to 2^{\boldsymbol{\mathcal{G}}} \colon \boldsymbol{v} \mapsto \bigtimes_{k=1}^{K} D_k v_k$
- $\boldsymbol{L} \colon \boldsymbol{\mathcal{H}} \to \boldsymbol{\mathcal{G}} \colon \boldsymbol{x} \mapsto \left(\sum_{i=1}^{m} L_{ki} x_i \right)_{1 \leqslant k \leqslant K}$, $\boldsymbol{z} = (z_i)_{1 \leqslant i \leqslant m}$, $\boldsymbol{r} = (r_k)_{1 \leqslant k \leqslant K}$

Reformulation in primal-dual space

- $\boldsymbol{\mathcal{H}} = \mathcal{H}_1 \oplus \cdots \oplus \mathcal{H}_m$, $\boldsymbol{\mathcal{G}} = \mathcal{G}_1 \oplus \cdots \oplus \mathcal{G}_K$, $\boldsymbol{\mathcal{K}} = \boldsymbol{\mathcal{H}} \oplus \boldsymbol{\mathcal{G}}$
- $\boldsymbol{A} \colon \boldsymbol{\mathcal{H}} \to 2^{\boldsymbol{\mathcal{H}}} \colon \boldsymbol{x} \mapsto \bigtimes_{i=1}^{m} A_i x_i$, $\boldsymbol{C} \colon \boldsymbol{\mathcal{H}} \to \boldsymbol{\mathcal{H}} \colon \boldsymbol{x} \mapsto (C_i x_i)_{1 \leqslant i \leqslant m}$
- $\boldsymbol{B} \colon \boldsymbol{\mathcal{G}} \to 2^{\boldsymbol{\mathcal{G}}} \colon \boldsymbol{v} \mapsto \bigtimes_{k=1}^{K} B_k v_k$, $\boldsymbol{D} \colon \boldsymbol{\mathcal{G}} \to 2^{\boldsymbol{\mathcal{G}}} \colon \boldsymbol{v} \mapsto \bigtimes_{k=1}^{K} D_k v_k$
- $\boldsymbol{L} \colon \boldsymbol{\mathcal{H}} \to \boldsymbol{\mathcal{G}} \colon \boldsymbol{x} \mapsto \left(\sum_{i=1}^{m} L_{ki} x_i \right)_{1 \leqslant k \leqslant K}$, $\boldsymbol{z} = (z_i)_{1 \leqslant i \leqslant m}$, $\boldsymbol{r} = (r_k)_{1 \leqslant k \leqslant K}$
- $\boldsymbol{P} \colon \boldsymbol{\mathcal{K}} \to 2^{\boldsymbol{\mathcal{K}}} \colon (\boldsymbol{x}, \boldsymbol{v}) \mapsto (-\boldsymbol{z} + \boldsymbol{A}\boldsymbol{x}) \times (\boldsymbol{r} + \boldsymbol{B}^{-1}\boldsymbol{v})$ (max. mon.)
- $\boldsymbol{Q} \colon \boldsymbol{\mathcal{K}} \to \boldsymbol{\mathcal{K}} \colon (\boldsymbol{x}, \boldsymbol{v}) \mapsto \big(\boldsymbol{C}\boldsymbol{x} + \boldsymbol{L}^* \boldsymbol{v}, \boldsymbol{D}^{-1} \boldsymbol{v} - \boldsymbol{L}\boldsymbol{x} \big)$ (mon. and Lips.)

Reformulation in primal-dual space

- $\boldsymbol{\mathcal{H}} = \mathcal{H}_1 \oplus \cdots \oplus \mathcal{H}_m$, $\boldsymbol{\mathcal{G}} = \mathcal{G}_1 \oplus \cdots \oplus \mathcal{G}_K$, $\boldsymbol{\mathcal{K}} = \boldsymbol{\mathcal{H}} \oplus \boldsymbol{\mathcal{G}}$
- $\boldsymbol{A} \colon \boldsymbol{\mathcal{H}} \to 2^{\boldsymbol{\mathcal{H}}} \colon \boldsymbol{x} \mapsto \bigtimes_{i=1}^{m} A_i x_i$, $\boldsymbol{C} \colon \boldsymbol{\mathcal{H}} \to \boldsymbol{\mathcal{H}} \colon \boldsymbol{x} \mapsto (C_i x_i)_{1 \leqslant i \leqslant m}$
- $\boldsymbol{B} \colon \boldsymbol{\mathcal{G}} \to 2^{\boldsymbol{\mathcal{G}}} \colon \boldsymbol{v} \mapsto \bigtimes_{k=1}^{K} B_k v_k$, $\boldsymbol{D} \colon \boldsymbol{\mathcal{G}} \to 2^{\boldsymbol{\mathcal{G}}} \colon \boldsymbol{v} \mapsto \bigtimes_{k=1}^{K} D_k v_k$
- $\boldsymbol{L} \colon \boldsymbol{\mathcal{H}} \to \boldsymbol{\mathcal{G}} \colon \boldsymbol{x} \mapsto \left(\sum_{i=1}^{m} L_{ki} x_i \right)_{1 \leqslant k \leqslant K}$, $\boldsymbol{z} = (z_i)_{1 \leqslant i \leqslant m}$, $\boldsymbol{r} = (r_k)_{1 \leqslant k \leqslant K}$
- $\boldsymbol{P} \colon \boldsymbol{\mathcal{K}} \to 2^{\boldsymbol{\mathcal{K}}} \colon (\boldsymbol{x}, \boldsymbol{v}) \mapsto (-\boldsymbol{z} + \boldsymbol{A}\boldsymbol{x}) \times (\boldsymbol{r} + \boldsymbol{B}^{-1}\boldsymbol{v})$ (max. mon.)
- $\boldsymbol{Q} \colon \boldsymbol{\mathcal{K}} \to \boldsymbol{\mathcal{K}} \colon (\boldsymbol{x}, \boldsymbol{v}) \mapsto \big(\boldsymbol{C}\boldsymbol{x} + \boldsymbol{L}^* \boldsymbol{v}, \boldsymbol{D}^{-1}\boldsymbol{v} - \boldsymbol{L}\boldsymbol{x}\big)$ (mon. and Lips.)
- Any zero of $\boldsymbol{P} + \boldsymbol{Q}$ is a primal-dual solution.
- $\longrightarrow$ Apply the forward-backward-forward algorithm to get...

Splitting algorithm

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

Open question

- All existing splitting methods are, in the end, an instance of the 3 basic splitting schemes.

Open question

- All existing splitting methods are, in the end, an instance of the 3 basic splitting schemes.
- In some very special cases, it is possible to devise methods which cannot be reduced to a 2-operator scheme, for instance if $\operatorname{zer}(A) \cap \operatorname{zer}(B) \cap \operatorname{zer}(C) \neq \varnothing$, iterate

$$x_{n+1} = (J_A \circ J_B \circ J_C)x_n \rightharpoonup z \in \operatorname{zer}(A + B + C).$$

Open question

- All existing splitting methods are, in the end, an instance of the 3 basic splitting schemes.
- In some very special cases, it is possible to devise methods which cannot be reduced to a 2-operator scheme, for instance if $\operatorname{zer}(A) \cap \operatorname{zer}(B) \cap \operatorname{zer}(C) \neq \varnothing$, iterate

$$x_{n+1} = (J_A \circ J_B \circ J_C)x_n \rightharpoonup z \in \operatorname{zer}(A + B + C).$$

- ***Open problem:*** Can we devise a genuine (not reducible to a 2-operator scheme through some reformulation or transformation) splitting scheme for $m > 2$?

Open question: possible bad news?

- 34 years have elapsed since 1979.

Open question: possible bad news?

- 34 years have elapsed since 1979.
- $\in$ is a binary relation.

Open question: possible bad news?

- 34 years have elapsed since 1979.
- $\in$ is a binary relation.
- $2 \neq 3$. For a proof see:

Open question: possible bad news?

- 34 years have elapsed since 1979.
- $\in$ is a binary relation.
- $2 \neq 3$. For a proof see:
    - J.-B. Baillon, PLC, and R. Cominetti, There is no variational characterization of the cycles in the method of periodic projections, *J. Funct. Anal.*, vol. 262, pp. 400–408, 2012.

Open question: possible bad news?

- 34 years have elapsed since 1979.
- $\in$ is a binary relation.
- $2 \neq 3$. For a proof see:
  - J.-B. Baillon, PLC, and R. Cominetti, There is no variational characterization of the cycles in the method of periodic projections, *J. Funct. Anal.*, vol. 262, pp. 400–408, 2012.
- ***Open problem 2:*** Can we formally show that any splitting method for $m > 2$ operator is reducible to a 2-operator method?