Resolvent Splitting with Minimal Lifting

Matthew K. Tam

School of Mathematics and Statistics
The University of Melbourne

One World Optimisation Seminar
November 8th, 2021.

Resolvent Splitting with Minimal Lifting

Joint work with Yura Malitsky (Linköping).

Reference for the talk:

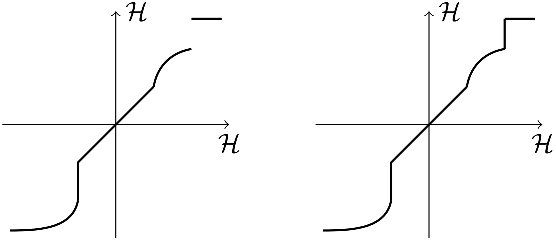
Resolvent splitting for sums of monotone operators with minimal lifting. Preprint: *arXiv:2108.02897*.

Monotone Operators

Let $\mathcal{H}$ be a real Hilbert space. An operator $B\colon \mathcal{H} \rightrightarrows \mathcal{H}$ is monotone if

$$\langle x - y, u - v \rangle \geq 0 \quad \forall (x, u), (y, v) \in \operatorname{graph} B.$$

A monotone operator B is maximally monotone if there exists no monotone operator whose graph properly contains $\operatorname{graph} B$.



Bauschke, H. H., & Combettes, P. L. (2011). *Convex analysis and monotone operator theory in Hilbert spaces*. New York: Springer.

Monotone Inclusions

Problem (*n*-operator monotone inclusion)

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \sum_{i=1}^{n} A_i(x),$$

where $A_i \colon \mathcal{H} \rightrightarrows \mathcal{H}$ is maximally monotone for all $i \in \{1, \ldots, n\}$.

Some important examples (potentially nonsmooth, finite sum):

- *Minimisation:* $A_i = \partial f_i$ for convex f_i gives

$$\min_x \sum_{i=1}^{n} f_i(x).$$

- *Minimax:* $A_i = \binom{\partial_u \Phi_i}{\partial_v(-\Phi_i)}$ for convex-concave $\Phi_i(u, v)$ gives

$$\min_u \max_v \sum_{i=1}^{n} \Phi_i(u, v).$$

(But examples are not the main focus of this talk.)

Monotone Inclusions

Problem (n-operator monotone inclusion)

$$\text{find } x \in \mathcal{H} \text{ such that } 0 \in \sum_{i=1}^{n} A_i(x),$$

where $A_i \colon \mathcal{H} \rightrightarrows \mathcal{H}$ is maximally monotone for all $i \in \{1, \ldots, n\}$.

Some important examples (potentially nonsmooth, finite sum):

- *Minimisation:* $A_i = \partial f_i$ for convex f_i gives
$$\min_x \sum_{i=1}^{n} f_i(x).$$
- *Minimax:* $A_i = \binom{\partial_u \Phi_i}{\partial_v(-\Phi_i)}$ for convex-concave $\Phi_i(u, v)$ gives
$$\min_u \max_v \sum_{i=1}^{n} \Phi_i(u, v).$$

(But examples are not the main focus of this talk.)

Solving Monotone Inclusions ($n = 1$)

Recall that the resolvent of an operator $B : \mathcal{H} \rightrightarrows \mathcal{H}$ is defined as

$$J_B := (\mathrm{Id} + B)^{-1}.$$

If B is maximally monotone, then J_B is a single-valued with full domain.

Proximal Point Algorithm

Let $A_1 : \mathcal{H} \rightrightarrows \mathcal{H}$ be maximally monotone with $\operatorname{zer} A_1 \neq \emptyset$. Given $z^0 \in \mathcal{H}$, consider the sequence (z^k) given by

$$z^{k+1} = J_{A_1}(z^k) \quad \forall k \in \mathbb{N}.$$

Then $z^k \rightharpoonup z \in \operatorname{Fix} J_{A_1} = \operatorname{zer} A_1$.

Solving Monotone Inclusions ($n = 2$)

Douglas–Rachford Splitting

Let $A_1, A_2 \colon \mathcal{H} \rightrightarrows \mathcal{H}$ be maximally monotone with $\operatorname{zer}(A_1 + A_2) \neq \emptyset$. Given $z^0 \in \mathcal{H}$, consider the sequence (z^k) given by

$$z^{k+1} = T_{\mathrm{DR}}(z^k) := z^k + J_{A_2}\big(2J_{A_1}(z^k) - z^k\big) - J_{A_1}(z^k) \quad \forall k \in \mathbb{N}.$$

Then $z^k \rightharpoonup z \in \operatorname{Fix} T_{\mathrm{DR}}$ and $J_{A_1}(z^k) \rightharpoonup J_{A_1}(z) \in \operatorname{zer}(A_1 + A_2)$.

Note, two different sequences are involved:

- T_{DR} generates (z^k), but this sequence does not solve the problem.
- The resolvent J_{A_1} is applied to (z^k) to get the solution sequence.

Solving Monotone Inclusions ($n \geq 3$)

Let $A = (A_1, \ldots, A_n)$ be an n-tuple of maximally monotone operators.

Reformulate the n-operator inclusion as a two operator inclusion:

$$x \in \operatorname{zer}\left(\sum_{i=1}^{n} A_i\right) \subseteq \mathcal{H} \iff \mathbf{x} = (x, \ldots, x) \in \operatorname{zer}(N_{\Delta_n} + A) \subseteq \mathcal{H}^n,$$

where N_{Δ_n} is normal cone to the diagonal subspace given by

$$\Delta_n := \{(x_1, \ldots, x_n) \in \mathcal{H}^n : x_1 = \cdots = x_n\}.$$

Douglas–Rachford Splitting in the Product Space

Apply Douglas–Rachford splitting in $\mathcal{H}^n$ to the two operator inclusion involving N_{Δ_n} and A. The DR operator $T_{\mathrm{DR}} : \mathcal{H}^n \to \mathcal{H}^n$ can be expressed in terms of $J_A = (J_{A_1}, \ldots, J_{A_n})$ and

$$J_{N_{\Delta_n}}(\mathbf{z}) = P_{\Delta_n}(\mathbf{z}) = \left(\frac{1}{n}\sum_{i=1}^{n} z_i, \ldots, \frac{1}{n}\sum_{i=1}^{n} z_i\right).$$

Solving Monotone Inclusions ($n \geq 3$)

Let $A = (A_1, \ldots, A_n)$ be an n-tuple of maximally monotone operators.

Reformulate the n-operator inclusion as a two operator inclusion:

$$x \in \operatorname{zer}\left(\sum_{i=1}^{n} A_i\right) \subseteq \mathcal{H} \iff \mathbf{x} = (x, \ldots, x) \in \operatorname{zer}(N_{\Delta_n} + A) \subseteq \mathcal{H}^n,$$

where N_{Δ_n} is normal cone to the diagonal subspace given by

$$\Delta_n := \{(x_1, \ldots, x_n) \in \mathcal{H}^n : x_1 = \cdots = x_n\}.$$

Douglas–Rachford Splitting in the Product Space

Apply Douglas–Rachford splitting in $\mathcal{H}^n$ to the two operator inclusion involving N_{Δ_n} and A. The DR operator $T_{\mathrm{DR}} : \mathcal{H}^n \to \mathcal{H}^n$ can be expressed in terms of $J_A = (J_{A_1}, \ldots, J_{A_n})$ and

$$J_{N_{\Delta_n}}(\mathbf{z}) = P_{\Delta_n}(\mathbf{z}) = \left(\frac{1}{n}\sum_{i=1}^{n} z_i, \ldots, \frac{1}{n}\sum_{i=1}^{n} z_i\right).$$

Solving Monotone Inclusions ($n \geq 3$)

Let $A = (A_1, \dots, A_n)$ be an n-tuple of maximally monotone operators.

Reformulate the n-operator inclusion as a two operator inclusion:

$$x \in \operatorname{zer}\left(\sum_{i=1}^{n} A_i\right) \subseteq \mathcal{H} \iff \mathbf{x} = (x, \dots, x) \in \operatorname{zer}(N_{\Delta_n} + A) \subseteq \mathcal{H}^n,$$

where N_{Δ_n} is normal cone to the diagonal subspace given by

$$\Delta_n := \{(x_1, \dots, x_n) \in \mathcal{H}^n : x_1 = \dots = x_n\}.$$

Douglas–Rachford Splitting in the Product Space

Apply Douglas–Rachford splitting in $\mathcal{H}^n$ to the two operator inclusion involving N_{Δ_n} and A. The DR operator $T_{\mathrm{DR}} : \mathcal{H}^n \to \mathcal{H}^n$ can be expressed in terms of $J_A = (J_{A_1}, \dots, J_{A_n})$ and

$$J_{N_{\Delta_n}}(\mathbf{z}) = P_{\Delta_n}(\mathbf{z}) = \left(\frac{1}{n}\sum_{i=1}^{n} z_i, \dots, \frac{1}{n}\sum_{i=1}^{n} z_i\right).$$

Goal of this Talk

- The literature is mostly devoted establishing what *is possible*:
  - Developing and analysing algorithms for solving monotone inclusions.
  - Algorithms distinguished based on the properties of $A_1, \dots, A_n$:
    - Set or single-valued, Lipschitz, cocoercive, strongly monotone, etc.

- Very little work concerned with examining what *is not possible*.
  - Statements like "There exists no algorithm with the following properties."
  - To be able to make such statements, we need to formalise the "rules".

⋆ **Main goal of talk belongs to the second category.**

Roughly speaking, our rules are:

- Fixed point algorithms which employ the resolvents of $A_1, \dots, A_n$.

Goal of this Talk

- The literature is mostly devoted establishing what *is possible*:
  - Developing and analysing algorithms for solving monotone inclusions.
  - Algorithms distinguished based on the properties of $A_1, \dots, A_n$:
    - Set or single-valued, Lipschitz, cocoercive, strongly monotone, etc.

- Very little work concerned with examining what *is not possible*.
  - Statements like "There exists no algorithm with the following properties."
  - To be able to make such statements, we need to formalise the "rules".

$\star$ **Main goal of talk belongs to the second category.**

Roughly speaking, our rules are:

- Fixed point algorithms which employ the resolvents of $A_1, \dots, A_n$.

Structure of Resolvent Splitting Algorithms

Definitions in this section from:

Ryu, E. K. (2020). Uniqueness of DRS as the 2 operator resolvent-splitting and impossibility of 3 operator resolvent-splitting. *Mathematical Programming*, 182(1), 233-273.

Fixed Point Encodings

Let $\mathcal{A}_n$ denote the set of n-tuples of maximally monotone operators.
i.e., $A = (A_1, \ldots, A_n) \in \mathcal{A}_n$ when all A_i's are maximally monotone.

Definition (Fixed point encoding)

A pair of single-valued operators (T_A, S_A) is a fixed point encoding for $\mathcal{A}_n$ if, for all $A \in \mathcal{A}_n$, the following hold:

1. $\operatorname{Fix} T_A \neq \emptyset \iff \operatorname{zer}\left(\sum_{i=1}^n A_i\right) \neq \emptyset$.
2. $\mathbf{z} \in \operatorname{Fix} T_A \implies S_A(\mathbf{z}) \in \operatorname{zer}\left(\sum_{i=1}^n A_i\right)$.

In addition, a fixed point encoding is said to be convergent if:[1]

3. For all initial points $\mathbf{z}^0$, we have $\mathbf{z}^{k+1} = T_A(\mathbf{z}^k) \rightharpoonup \mathbf{z} \in \operatorname{Fix} T_A$.

To interpret this definition, it helps to keep the following in mind:

- The fixed point operator, T_A, is the basis for the iterative algorithm:

$$\mathbf{z}^{k+1} = T_A(\mathbf{z}^k) \quad \forall k \in \mathbb{N}.$$

- The solution operator, S_A, maps fixed points of T_A to solutions.

[1] Ryu's original paper uses the terminology "unconditionally convergent" for (3).

Fixed Point Encodings

Let $\mathcal{A}_n$ denote the set of n-tuples of maximally monotone operators.
i.e., $A = (A_1, \ldots, A_n) \in \mathcal{A}_n$ when all A_i's are maximally monotone.

Definition (Fixed point encoding)

A pair of single-valued operators (T_A, S_A) is a fixed point encoding for $\mathcal{A}_n$ if, for all $A \in \mathcal{A}_n$, the following hold:

1. $\operatorname{Fix} T_A \neq \emptyset \iff \operatorname{zer}\left(\sum_{i=1}^n A_i\right) \neq \emptyset$.
2. $\mathbf{z} \in \operatorname{Fix} T_A \implies S_A(\mathbf{z}) \in \operatorname{zer}\left(\sum_{i=1}^n A_i\right)$.

In addition, a fixed point encoding is said to be convergent if:[1]

3. For all initial points $\mathbf{z}^0$, we have $\mathbf{z}^{k+1} = T_A(\mathbf{z}^k) \rightharpoonup \mathbf{z} \in \operatorname{Fix} T_A$.

To interpret this definition, it helps to keep the following in mind:

- The fixed point operator, T_A, is the basis for the iterative algorithm:

$$\mathbf{z}^{k+1} = T_A(\mathbf{z}^k) \quad \forall k \in \mathbb{N}.$$

- The solution operator, S_A, maps fixed points of T_A to solutions.

[1] Ryu's original paper uses the terminology "unconditionally convergent" for (3).

Fixed Point Encodings

Let $\mathcal{A}_n$ denote the set of n-tuples of maximally monotone operators. *i.e.*, $A = (A_1, \ldots, A_n) \in \mathcal{A}_n$ when all A_i's are maximally monotone.

Definition (Fixed point encoding)

A pair of single-valued operators (T_A, S_A) is a fixed point encoding for $\mathcal{A}_n$ if, for all $A \in \mathcal{A}_n$, the following hold:

1. $\operatorname{Fix} T_A \neq \emptyset \iff \operatorname{zer}\left(\sum_{i=1}^n A_i\right) \neq \emptyset$.
2. $\mathbf{z} \in \operatorname{Fix} T_A \implies S_A(\mathbf{z}) \in \operatorname{zer}\left(\sum_{i=1}^n A_i\right)$.

In addition, a fixed point encoding is said to be convergent if:[1]

3. For all initial points $\mathbf{z}^0$, we have $\mathbf{z}^{k+1} = T_A(\mathbf{z}^k) \rightharpoonup \mathbf{z} \in \operatorname{Fix} T_A$.

To interpret this definition, it helps to keep the following in mind:

- The fixed point operator, T_A, is the basis for the iterative algorithm:

$$\mathbf{z}^{k+1} = T_A(\mathbf{z}^k) \quad \forall k \in \mathbb{N}.$$

- The solution operator, S_A, maps fixed points of T_A to solutions.

[1] Ryu's original paper uses the terminology "unconditionally convergent" for (3).

Examples of Fixed Point Encodings

- Proximal point algorithm is a fixed point encoding for $\mathcal{A}_1$ with

$$T_A = J_{A_1} \quad \text{and} \quad S_A = \mathrm{Id}\,.$$

- Douglas–Rachford splitting is a fixed point encoding for $\mathcal{A}_2$ with

$$T_A = \mathrm{Id} + J_{A_2}(2J_{A_1} - \mathrm{Id}) - J_{A_1} \quad \text{and} \quad S_A = J_{A_1}\,.$$

- DR splitting in the product space is a fixed point encoding for $\mathcal{A}_n$ with

$$T_A = \mathrm{Id} + J_A(2P_{\Delta_n} - \mathrm{Id}) - P_{\Delta_n} \quad \text{and} \quad S_A(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} z_i,$$

where we note that, for this case, $T_A \colon \mathcal{H}^n \to \mathcal{H}^n$ and $S_A \colon \mathcal{H}^n \to \mathcal{H}$.

⋆ All these examples are actually convergent fixed point encodings.

Examples of Fixed Point Encodings

- Proximal point algorithm is a fixed point encoding for $\mathcal{A}_1$ with

$$T_A = J_{A_1} \quad \text{and} \quad S_A = \mathrm{Id}\,.$$

- Douglas–Rachford splitting is a fixed point encoding for $\mathcal{A}_2$ with

$$T_A = \mathrm{Id} + J_{A_2}(2J_{A_1} - \mathrm{Id}) - J_{A_1} \quad \text{and} \quad S_A = J_{A_1}\,.$$

- DR splitting in the product space is a fixed point encoding for $\mathcal{A}_n$ with

$$T_A = \mathrm{Id} + J_A(2P_{\Delta_n} - \mathrm{Id}) - P_{\Delta_n} \quad \text{and} \quad S_A(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} z_i,$$

where we note that, for this case, $T_A \colon \mathcal{H}^n \to \mathcal{H}^n$ and $S_A \colon \mathcal{H}^n \to \mathcal{H}$.

⋆ All these examples are actually convergent fixed point encodings.

Examples of Fixed Point Encodings

- Proximal point algorithm is a fixed point encoding for $\mathcal{A}_1$ with

$$T_A = J_{A_1} \quad \text{and} \quad S_A = \mathrm{Id}\,.$$

- Douglas–Rachford splitting is a fixed point encoding for $\mathcal{A}_2$ with

$$T_A = \mathrm{Id} + J_{A_2}(2J_{A_1} - \mathrm{Id}) - J_{A_1} \quad \text{and} \quad S_A = J_{A_1}\,.$$

- DR splitting in the product space is a fixed point encoding for $\mathcal{A}_n$ with

$$T_A = \mathrm{Id} + J_A(2P_{\Delta_n} - \mathrm{Id}) - P_{\Delta_n} \quad \text{and} \quad S_A(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} z_i,$$

  where we note that, for this case, $T_A\colon \mathcal{H}^n \to \mathcal{H}^n$ and $S_A\colon \mathcal{H}^n \to \mathcal{H}$.

⋆ All these examples are actually convergent fixed point encodings.

Examples of Fixed Point Encodings

- Proximal point algorithm is a fixed point encoding for $\mathcal{A}_1$ with

$$T_A = J_{A_1} \quad \text{and} \quad S_A = \operatorname{Id}.$$

- Douglas–Rachford splitting is a fixed point encoding for $\mathcal{A}_2$ with

$$T_A = \operatorname{Id} + J_{A_2}(2J_{A_1} - \operatorname{Id}) - J_{A_1} \quad \text{and} \quad S_A = J_{A_1}.$$

- DR splitting in the product space is a fixed point encoding for $\mathcal{A}_n$ with

$$T_A = \operatorname{Id} + J_A(2P_{\Delta_n} - \operatorname{Id}) - P_{\Delta_n} \quad \text{and} \quad S_A(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} z_i,$$

where we note that, for this case, $T_A \colon \mathcal{H}^n \to \mathcal{H}^n$ and $S_A \colon \mathcal{H}^n \to \mathcal{H}$.

⋆ All these examples are actually convergent fixed point encodings.

Examples of Fixed Point Encodings

- Proximal point algorithm is a fixed point encoding for $\mathcal{A}_1$ with

$$T_A = J_{A_1} \quad \text{and} \quad S_A = \mathrm{Id}\,.$$

- Douglas–Rachford splitting is a fixed point encoding for $\mathcal{A}_2$ with

$$T_A = \mathrm{Id} + J_{A_2}(2J_{A_1} - \mathrm{Id}) - J_{A_1} \quad \text{and} \quad S_A = J_{A_1}\,.$$

- DR splitting in the product space is a fixed point encoding for $\mathcal{A}_n$ with

$$T_A = \mathrm{Id} + J_A(2P_{\Delta_n} - \mathrm{Id}) - P_{\Delta_n} \quad \text{and} \quad S_A(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} z_i,$$

where we note that, for this case, $T_A\colon \mathcal{H}^n \to \mathcal{H}^n$ and $S_A\colon \mathcal{H}^n \to \mathcal{H}$.

⋆ All these examples are actually convergent fixed point encodings.

An Example which is not a Fixed Point Encoding

- The forward-backward algorithm given by

$$T_{\mathrm{FB}} = J_{\lambda A_1}\left(\mathrm{Id} - \lambda A_2\right)$$

  is **not** a convergent fixed point encoding for $\mathcal{A}_2$ because:
  - Convergence of fixed point iteration requires A_2 to be cocoercive.
  - T_{FB} is single-valued only when A_2 is single-valued.

- To be a fixed point encoding for $\mathcal{A}_2$, the properties must hold **for all** pairs of maximally monotone operators $A = (A_1, A_2) \in \mathcal{A}_2$.

Resolvent Splitting and Frugality

Definition (Resolvent splitting and frugality)

A fixed point encoding (T_A, S_A) is a resolvent splitting if, for all $A \in \mathcal{A}_n$, there is a procedure that evaluates T_A and S_A at a point that uses only:

1. Vector addition.
2. Scalar multiplication.
3. The resolvents $J_{A_1}, \ldots, J_{A_n}$.

In addition, if the procedure uses each resolvent only once, then the resolvent splitting is said to be frugal.

Implications for T_A and S_A:

- Allows for a kind of canonical form in terms of coefficient matrices.
- Informally, non-linearities in T_A can only arise from the resolvents.

Resolvent Splitting and Frugality

Definition (Resolvent splitting and frugality)

A fixed point encoding (T_A, S_A) is a resolvent splitting if, for all $A \in \mathcal{A}_n$, there is a procedure that evaluates T_A and S_A at a point that uses only:

1. Vector addition.
2. Scalar multiplication.
3. The resolvents $J_{A_1}, \ldots, J_{A_n}$.

In addition, if the procedure uses each resolvent only once, then the resolvent splitting is said to be frugal.

Implications for T_A and S_A:

- Allows for a kind of canonical form in terms of coefficient matrices.
- Informally, non-linearities in T_A can only arise from the resolvents.

Examples for (Frugal) Resolvent Splittings

- All fixed point encoding on previous slide are frugal resolvent splittings.
- The convergent fixed point encoding (T_A, S_A) for $\mathcal{A}_1$ given by

$$T_A = \mathrm{Id} + J_{A_1}(2J_{A_1} - \mathrm{Id}) - J_{A_1} \quad \text{and} \quad S_A = J_{A_1}.$$

  is a resolvent splitting but it is **not** a frugal resolvent splitting.
- Methods whose iterations project onto separating hyperplanes are **not** resolvent splittings, even though they use $J_{A_1}, \dots, J_{A_n}$.
  - Haugazeau-type methods, Projective splitting (Eckstein–Svaiter '08), etc.
- Methods whose iterations use the resolvents $J_{\lambda_1 A_1}, \dots, J_{\lambda_n A_n}$ with different values for $\lambda_1, \dots, \lambda_n > 0$ are **not** resolvent splittings
  - Parallel Douglas–Rachford with reduced dimension (Condat–Kitaharra–Contreras–Hirabayashi '20, Campoy '21).

Examples for (Frugal) Resolvent Splittings

- All fixed point encoding on previous slide are frugal resolvent splittings.
- The convergent fixed point encoding (T_A, S_A) for $\mathcal{A}_1$ given by

$$T_A = \mathrm{Id} + J_{A_1}(2J_{A_1} - \mathrm{Id}) - J_{A_1} \quad \text{and} \quad S_A = J_{A_1}.$$

  is a resolvent splitting but it is **not** a frugal resolvent splitting.
- Methods whose iterations project onto separating hyperplanes are **not** resolvent splittings, even though they use $J_{A_1}, \dots, J_{A_n}$.
  - Haugazeau-type methods, Projective splitting (Eckstein–Svaiter '08), etc.
- Methods whose iterations use the resolvents $J_{\lambda_1 A_1}, \dots, J_{\lambda_n A_n}$ with different values for $\lambda_1, \dots, \lambda_n > 0$ are **not** resolvent splittings
  - Parallel Douglas–Rachford with reduced dimension (Condat–Kitaharra–Contreras–Hirabayashi '20, Campoy '21).

Examples for (Frugal) Resolvent Splittings

- All fixed point encoding on previous slide are frugal resolvent splittings.
- The convergent fixed point encoding (T_A, S_A) for $\mathcal{A}_1$ given by

$$T_A = \mathrm{Id} + J_{A_1}(2J_{A_1} - \mathrm{Id}) - J_{A_1} \quad \text{and} \quad S_A = J_{A_1}.$$

  is a resolvent splitting but it is **not** a frugal resolvent splitting.
- Methods whose iterations project onto separating hyperplanes are **not** resolvent splittings, even though they use $J_{A_1}, \dots, J_{A_n}$.
  - Haugazeau-type methods, Projective splitting (Eckstein–Svaiter '08), etc.
- Methods whose iterations use the resolvents $J_{\lambda_1 A_1}, \dots, J_{\lambda_n A_n}$ with different values for $\lambda_1, \dots, \lambda_n > 0$ are **not** resolvent splittings
  - Parallel Douglas–Rachford with reduced dimension (Condat–Kitaharra–Contreras–Hirabayashi '20, Campoy '21).

Examples for (Frugal) Resolvent Splittings

- All fixed point encoding on previous slide are frugal resolvent splittings.
- The convergent fixed point encoding (T_A, S_A) for $\mathcal{A}_1$ given by

$$T_A = \mathrm{Id} + J_{A_1}(2J_{A_1} - \mathrm{Id}) - J_{A_1} \quad \text{and} \quad S_A = J_{A_1}.$$

  is a resolvent splitting but it is **not** a frugal resolvent splitting.

- Methods whose iterations project onto separating hyperplanes are **not** resolvent splittings, even though they use $J_{A_1}, \dots, J_{A_n}$.
  - Haugazeau-type methods, Projective splitting (Eckstein–Svaiter '08), etc.
- Methods whose iterations use the resolvents $J_{\lambda_1 A_1}, \dots, J_{\lambda_n A_n}$ with different values for $\lambda_1, \dots, \lambda_n > 0$ are **not** resolvent splittings
  - Parallel Douglas–Rachford with reduced dimension (Condat–Kitaharra–Contreras–Hirabayashi '20, Campoy '21).

The Structure of the Solution Map

Recall that a fixed point encoding must satisfy:

$$\mathbf{z} \in \operatorname{Fix} T_A \implies S_A(\mathbf{z}) \in \operatorname{zer}\left(\sum_{i=1}^{n} A_i\right).$$

Using the canonical form of a frugal resolvents splitting, properties about the general structure of all such algorithms can be derived.

Proposition (Malitsky–T.)

Let (T_A, S_A) be a frugal resolvent splitting for $\mathcal{A}_n$. Suppose $\mathbf{z} \in \operatorname{Fix} T_A$ and let y_i denote the point where J_{A_i} is evaluated in the procedure for evaluating $T_A(\mathbf{z})$. Then, necessarily, we have

$$S_A(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} y_i = J_{A_1}(y_1) = \cdots = J_{A_n}(y_n).$$

The Structure of the Solution Map

Recall that a fixed point encoding must satisfy:

$$\mathbf{z} \in \operatorname{Fix} T_A \implies S_A(\mathbf{z}) \in \operatorname{zer}\left(\sum_{i=1}^{n} A_i\right).$$

Using the canonical form of a frugal resolvents splitting, properties about the general structure of all such algorithms can be derived.

Proposition (Malitsky–T.)

Let (T_A, S_A) be a frugal resolvent splitting for $\mathcal{A}_n$. Suppose $\mathbf{z} \in \operatorname{Fix} T_A$ and let y_i denote the point where J_{A_i} is evaluated in the procedure for evaluating $T_A(\mathbf{z})$. Then, necessarily, we have

$$S_A(\mathbf{z}) = \frac{1}{n}\sum_{i=1}^{n} y_i = J_{A_1}(y_1) = \cdots = J_{A_n}(y_n).$$

Lifting

Definition (Lifting)

Let $d \in \mathbb{N}$. A fixed point encoding (T_A, S_A) is a *d-fold lifting* for $\mathcal{A}_n$ if $T_A \colon \mathcal{H}^d \to \mathcal{H}^d$ and $S_A \colon \mathcal{H}^d \to \mathcal{H}$.

- Value of d represents number of copies of variable needed to use T_A.
- Smaller d means the corresponding algorithm needs less memory.
- Methods which attain the smallest value of d (for a given n) are said to have "minimal lifting". That is, they have the lowest memory requirements for algorithm class.

$\star$ **We focus on (minimal) lifting for frugal resolvent splittings.**

Examples of Lifting for Frugal Resolvent Splittings

- Proximal point algorithm for $\mathcal{A}_1$ has 1-fold lifting (*i.e.*, no lifting).
- Douglas–Rachford splitting for $\mathcal{A}_2$ has 1-fold lifting (*i.e.*, no lifting): $T_A\colon \mathcal{H} \to \mathcal{H}$ where

$$T_A = \mathrm{Id} + J_{A_2}(2J_{A_1} - \mathrm{Id}) - J_{A_1}.$$

- DR splitting in the product space for $\mathcal{A}_n$ has n-fold lifting: $T_A\colon \mathcal{H}^n \to \mathcal{H}^n$ where

$$T_A = \mathrm{Id} + J_A(2P_{\Delta_n} - \mathrm{Id}) - P_{\Delta_n}.$$

- Primal-dual hybrid gradient (PDHG) for $\mathcal{A}_2$ has 2-fold lifting: $T_A\colon \mathcal{H}^2 \to \mathcal{H}^2$ given by

$$\begin{bmatrix} u^{k+1} \\ v^{k+1} \end{bmatrix} := T_A\left(\begin{bmatrix} u^k \\ v^k \end{bmatrix}\right) = \begin{bmatrix} J_{A_1}(u^k - \lambda v^k) \\ (\mathrm{Id} - J_{A_2})(v^k + \lambda(2u^{k+1} - v^k)) \end{bmatrix}$$

Examples of Lifting for Frugal Resolvent Splittings

- Proximal point algorithm for $\mathcal{A}_1$ has 1-fold lifting (*i.e.*, no lifting).
- Douglas–Rachford splitting for $\mathcal{A}_2$ has 1-fold lifting (*i.e.*, no lifting): $T_A \colon \mathcal{H} \to \mathcal{H}$ where

$$T_A = \mathrm{Id} + J_{A_2}(2J_{A_1} - \mathrm{Id}) - J_{A_1}.$$

- DR splitting in the product space for $\mathcal{A}_n$ has n-fold lifting: $T_A \colon \mathcal{H}^n \to \mathcal{H}^n$ where

$$T_A = \mathrm{Id} + J_A(2P_{\Delta_n} - \mathrm{Id}) - P_{\Delta_n}.$$

- Primal-dual hybrid gradient (PDHG) for $\mathcal{A}_2$ has 2-fold lifting: $T_A \colon \mathcal{H}^2 \to \mathcal{H}^2$ given by

$$\begin{bmatrix} u^{k+1} \\ v^{k+1} \end{bmatrix} := T_A\left(\begin{bmatrix} u^k \\ v^k \end{bmatrix}\right) = \begin{bmatrix} J_{A_1}(u^k - \lambda v^k) \\ (\mathrm{Id} - J_{A_2})(v^k + \lambda(2u^{k+1} - v^k)) \end{bmatrix}$$

Examples of Lifting for Frugal Resolvent Splittings

- Proximal point algorithm for $\mathcal{A}_1$ has 1-fold lifting (*i.e.*, no lifting).
- Douglas–Rachford splitting for $\mathcal{A}_2$ has 1-fold lifting (*i.e.*, no lifting): $T_A \colon \mathcal{H} \to \mathcal{H}$ where

$$T_A = \mathrm{Id} + J_{A_2}(2J_{A_1} - \mathrm{Id}) - J_{A_1}.$$

- DR splitting in the product space for $\mathcal{A}_n$ has n-fold lifting: $T_A \colon \mathcal{H}^n \to \mathcal{H}^n$ where

$$T_A = \mathrm{Id} + J_A(2P_{\Delta_n} - \mathrm{Id}) - P_{\Delta_n}.$$

- Primal-dual hybrid gradient (PDHG) for $\mathcal{A}_2$ has 2-fold lifting: $T_A \colon \mathcal{H}^2 \to \mathcal{H}^2$ given by

$$\begin{bmatrix} u^{k+1} \\ v^{k+1} \end{bmatrix} := T_A\left(\begin{bmatrix} u^k \\ v^k \end{bmatrix}\right) = \begin{bmatrix} J_{A_1}(u^k - \lambda v^k) \\ (\mathrm{Id} - J_{A_2})(v^k + \lambda(2u^{k+1} - v^k)) \end{bmatrix}$$

Examples of Lifting for Frugal Resolvent Splittings

- Proximal point algorithm for $\mathcal{A}_1$ has 1-fold lifting (*i.e.*, no lifting).
- Douglas–Rachford splitting for $\mathcal{A}_2$ has 1-fold lifting (*i.e.*, no lifting): $T_A \colon \mathcal{H} \to \mathcal{H}$ where

$$T_A = \mathrm{Id} + J_{A_2}(2J_{A_1} - \mathrm{Id}) - J_{A_1}.$$

- DR splitting in the product space for $\mathcal{A}_n$ has n-fold lifting: $T_A \colon \mathcal{H}^n \to \mathcal{H}^n$ where

$$T_A = \mathrm{Id} + J_A(2P_{\Delta_n} - \mathrm{Id}) - P_{\Delta_n}.$$

- Primal-dual hybrid gradient (PDHG) for $\mathcal{A}_2$ has 2-fold lifting: $T_A \colon \mathcal{H}^2 \to \mathcal{H}^2$ given by

$$\begin{bmatrix} u^{k+1} \\ v^{k+1} \end{bmatrix} := T_A\left(\begin{bmatrix} u^k \\ v^k \end{bmatrix}\right) = \begin{bmatrix} J_{A_1}(u^k - \lambda v^k) \\ (\mathrm{Id} - J_{A_2})(v^k + \lambda(2u^{k+1} - v^k)) \end{bmatrix}$$

Examples of Lifting for Frugal Resolvent Splittings

- Proximal point algorithm for $\mathcal{A}_1$ has 1-fold lifting (*i.e.*, no lifting).
- Douglas–Rachford splitting for $\mathcal{A}_2$ has 1-fold lifting (*i.e.*, no lifting): $T_A \colon \mathcal{H} \to \mathcal{H}$ where

$$T_A = \mathrm{Id} + J_{A_2}(2J_{A_1} - \mathrm{Id}) - J_{A_1}.$$

- DR splitting in the product space for $\mathcal{A}_n$ has n-fold lifting: $T_A \colon \mathcal{H}^n \to \mathcal{H}^n$ where

$$T_A = \mathrm{Id} + J_A(2P_{\Delta_n} - \mathrm{Id}) - P_{\Delta_n}.$$

- Primal-dual hybrid gradient (PDHG) for $\mathcal{A}_2$ has 2-fold lifting: $T_A \colon \mathcal{H}^2 \to \mathcal{H}^2$ given by

$$\begin{bmatrix} u^{k+1} \\ v^{k+1} \end{bmatrix} := T_A\left(\begin{bmatrix} u^k \\ v^k \end{bmatrix}\right) = \begin{bmatrix} J_{A_1}(u^k - \lambda v^k) \\ (\mathrm{Id} - J_{A_2})(v^k + \lambda(2u^{k+1} - v^k)) \end{bmatrix}$$

Lifting for Frugal Resolvent Splittings

For convenience, denote:

$$d^*(n) := \begin{cases} \text{minimal amount of lifting needed by} \\ \text{convergent frugal resolvent splittings for } \mathcal{A}_n \end{cases}$$

Upper bound: DR splitting applied to the product space gives

$$d^*(n) \leq n \quad \forall n \in \mathbb{N}.$$

The first few values of $d^*(n)$ are known in the literature:

- $d^*(1) = 1$ due to the proximal point algorithm.
- $d^*(2) = 1$ due to Douglas–Rachford splitting.
- $d^*(3) = ??$

Lifting for Frugal Resolvent Splittings

For convenience, denote:

$$d^*(n) := \begin{cases} \text{minimal amount of lifting needed by} \\ \text{convergent frugal resolvent splittings for } \mathcal{A}_n \end{cases}$$

Upper bound: DR splitting applied to the product space gives

$$d^*(n) \leq n \quad \forall n \in \mathbb{N}.$$

The first few values of $d^*(n)$ are known in the literature:

- $d^*(1) = 1$ due to the proximal point algorithm.
- $d^*(2) = 1$ due to Douglas–Rachford splitting.
- $d^*(3) = ??$

Lifting for Frugal Resolvent Splittings

For convenience, denote:

$$d^*(n) := \begin{cases} \text{minimal amount of lifting needed by} \\ \text{convergent frugal resolvent splittings for } \mathcal{A}_n \end{cases}$$

Upper bound: DR splitting applied to the product space gives

$$d^*(n) \leq n \quad \forall n \in \mathbb{N}.$$

The first few values of $d^*(n)$ are known in the literature:

- $d^*(1) = 1$ due to the proximal point algorithm.
- $d^*(2) = 1$ due to Douglas–Rachford splitting.
- $d^*(3) = ??$

Ryu's Splitting Algorithm for $n = 3$

Let $\gamma \in (0, 1)$. Ryu's splitting algorithm is given by $T_A \colon \mathcal{H}^2 \to \mathcal{H}^2$ where

$$T_A(\mathbf{z}) = \mathbf{z} + \gamma \begin{pmatrix} x_3 - x_1 \\ x_3 - x_2 \end{pmatrix} \quad \text{where} \quad \begin{cases} x_1 = J_{A_1}(z_1) \\ x_2 = J_{A_2}(z_2 + x_1) \\ x_3 = J_{A_3}(x_1 - z_1 + x_2 - z_2). \end{cases}$$

Ryu, E. K. (2020). Uniqueness of DRS as the 2 operator resolvent-splitting and impossibility of 3 operator resolvent-splitting. *Math. Program.*, 182(1), 233–273.

Aragón Artacho, F. J., Campoy, R., & Tam, M. K. (2021). Strengthened splitting methods for computing resolvents. *COAP*, 80(2), 549–585.

Ryu's Splitting Algorithm for $n = 3$

Let $\gamma \in (0,1)$. Ryu's splitting algorithm is given by $T_A\colon \mathcal{H}^2 \to \mathcal{H}^2$ where

$$\mathbf{z}^{k+1} = T_A(\mathbf{z}^k) = \mathbf{z}^k + \gamma \begin{pmatrix} x_3^k - x_1^k \\ x_3^k - x_2^k \end{pmatrix} \quad \text{where} \quad \begin{cases} x_1^k = J_{A_1}(z_1^k) \\ x_2^k = J_{A_2}(z_2^k + x_1^k) \\ x_3^k = J_{A_3}(x_1^k - z_1^k + x_2^k - z_2^k). \end{cases}$$

Ryu, E. K. (2020). Uniqueness of DRS as the 2 operator resolvent-splitting and impossibility of 3 operator resolvent-splitting. *Math. Program.*, 182(1), 233–273.

Aragón Artacho, F. J., Campoy, R., & Tam, M. K. (2021). Strengthened splitting methods for computing resolvents. *COAP*, 80(2), 549–585.

Ryu's Splitting Algorithm for $n = 3$

Let $\gamma \in (0,1)$. Ryu's splitting algorithm is given by $T_A \colon \mathcal{H}^2 \to \mathcal{H}^2$ where

$$\mathbf{z}^{k+1} = T_A(\mathbf{z}^k) = \mathbf{z}^k + \gamma \begin{pmatrix} x_3^k - x_1^k \\ x_3^k - x_2^k \end{pmatrix} \quad \text{where} \quad \begin{cases} x_1^k = J_{A_1}(z_1^k) \\ x_2^k = J_{A_2}(z_2^k + x_1^k) \\ x_3^k = J_{A_3}(x_1^k - z_1^k + x_2^k - z_2^k). \end{cases}$$

Theorem (Ryu, Aragón–Campoy–T.)

Let $A_1, A_2, A_3 \colon \mathcal{H} \rightrightarrows \mathcal{H}$ be maximally monotone with $\operatorname{zer}(\sum_{i=1}^3 A_i) \neq \emptyset$. Given an initial $\mathbf{z}^0 \in \mathcal{H}^2$, define the sequences $(\mathbf{z}^k)$ and $(\mathbf{x}^k)$ as above. Then the following assertions hold.

1. $\mathbf{z}^k \rightharpoonup \mathbf{z} \in \operatorname{Fix} T_A$.
2. $\mathbf{x}^k \rightharpoonup (x,x,x) \in \mathcal{H}^3$ with $x \in \operatorname{zer}(\sum_{i=1}^3 A_i)$.

Ryu, E. K. (2020). Uniqueness of DRS as the 2 operator resolvent-splitting and impossibility of 3 operator resolvent-splitting. *Math. Program.*, 182(1), 233–273.

Aragón Artacho, F. J., Campoy, R., & Tam, M. K. (2021). Strengthened splitting methods for computing resolvents. *COAP*, 80(2), 549–585.

Lifting for Frugal Resolvent Splittings

For convenience, denote:

$$d^*(n) := \begin{cases} \text{minimal amount of lifting needed by} \\ \text{convergent frugal resolvent splittings for } \mathcal{A}_n \end{cases}$$

Upper bound: DR splitting applied to the product space gives

$$d^*(n) \leq n \quad \forall n \in \mathbb{N}.$$

The first few values of $d^*(n)$ are known in the literature:

- $d^*(1) = 1$ due to the proximal point algorithm.
- $d^*(2) = 1$ due to Douglas–Rachford splitting.
- $d^*(3) = ??$

Lifting for Frugal Resolvent Splittings

For convenience, denote:

$$d^*(n) := \begin{cases} \text{minimal amount of lifting needed by} \\ \text{convergent frugal resolvent splittings for } \mathcal{A}_n \end{cases}$$

Upper bound: DR splitting applied to the product space gives

$$d^*(n) \leq n \quad \forall n \in \mathbb{N}.$$

The first few values of $d^*(n)$ are known in the literature:

- $d^*(1) = 1$ due to the proximal point algorithm.
- $d^*(2) = 1$ due to Douglas–Rachford splitting.
- $d^*(3) = 2$ due to Ryu's splitting algorithm.

Lifting for Frugal Resolvent Splittings

For convenience, denote:

$$d^*(n) := \begin{cases} \text{minimal amount of lifting needed by} \\ \text{convergent frugal resolvent splittings for } \mathcal{A}_n \end{cases}$$

Upper bound: DR splitting applied to the product space gives

$$d^*(n) \leq n \quad \forall n \in \mathbb{N}.$$

The first few values of $d^*(n)$ are known in the literature:

- $d^*(1) = 1$ due to the proximal point algorithm.
- $d^*(2) = 1$ due to Douglas–Rachford splitting.
- $d^*(3) = 2$ due to Ryu's splitting algorithm.

Conjecture

The minimal amount of lifting is given by $d^*(n) = n - 1$ for $n \geq 2$.

Resolvent Splitting with Lifting

As a first step towards resolving the conjecture, we showed the following.

Theorem (Malitsky-T.)

Let $n \geq 2$. If (T_A, S_A) is a frugal resolvent splitting for $\mathcal{A}_n$ with d-fold lifting, then $d \geq n-1$.

- Proof is by contradiction and uses the rank-nullity theorem applied to the coefficient matrices in the canonical form of T_A.
- No need to consider S_A directly – already determined by proposition.
- **Consequence of theorem:** $d^*(n) = n-1$ or $d^*(n) = n$.

⋆ Do frugal resolvents splittings with $(n-1)$-lifting exist for $n \geq 3$?

Resolvent Splitting with Lifting

As a first step towards resolving the conjecture, we showed the following.

Theorem (Malitsky-T.)

Let $n \geq 2$. If (T_A, S_A) is a frugal resolvent splitting for $\mathcal{A}_n$ with d-fold lifting, then $d \geq n-1$.

- Proof is by contradiction and uses the rank-nullity theorem applied to the coefficient matrices in the canonical form of T_A.
- No need to consider S_A directly – already determined by proposition.
- **Consequence of theorem:** $d^*(n) = n-1$ or $d^*(n) = n$.

⋆ Do frugal resolvents splittings with $(n-1)$-lifting exist for $n \geq 3$?

Resolvent Splitting with Lifting

As a first step towards resolving the conjecture, we showed the following.

Theorem (Malitsky-T.)

Let $n \geq 2$. If (T_A, S_A) is a frugal resolvent splitting for $\mathcal{A}_n$ with d-fold lifting, then $d \geq n-1$.

- Proof is by contradiction and uses the rank-nullity theorem applied to the coefficient matrices in the canonical form of T_A.
- No need to consider S_A directly – already determined by proposition.
- **Consequence of theorem:** $d^*(n) = n-1$ or $d^*(n) = n$.

⋆ Do frugal resolvents splittings with $(n-1)$-lifting exist for $n \geq 3$?

Resolvent Splitting with Lifting

As a first step towards resolving the conjecture, we showed the following.

Theorem (Malitsky-T.)

Let $n \geq 2$. If (T_A, S_A) is a frugal resolvent splitting for $\mathcal{A}_n$ with d-fold lifting, then $d \geq n-1$.

- Proof is by contradiction and uses the rank-nullity theorem applied to the coefficient matrices in the canonical form of T_A.
- No need to consider S_A directly – already determined by proposition.
- **Consequence of theorem:** $d^*(n) = n-1$ or $d^*(n) = n$.

$\star$ Do frugal resolvents splittings with $(n-1)$-lifting exist for $n \geq 3$?

Frugal Resolvent Splitting with Minimal Lifting

Extending Ryu's Splitting from $n = 3$ to $n = 4$

We tried (unsuccessfully) to extend Ryu's scheme to $n \geq 4$ operators.

Let $\gamma \in (0, 1)$. Define $T : \mathcal{H}^2 \to \mathcal{H}^2$ according to

$$T(\mathbf{z}) = \mathbf{z} + \gamma \begin{pmatrix} x_4 - x_1 \\ \\ x_4 - x_3 \end{pmatrix} \quad \text{where} \quad \begin{cases} x_1 = J_{A_1}(z_1) \\ x_2 = J_{A_2}(z_2 + x_1) \\ \\ x_4 = J_{A_4}(x_1 - z_1 + x_2 - z_2) \end{cases},$$

- Possible four operator extension of Ryu's splitting (red terms new).
- If $\mathbf{z} \in \operatorname{Fix} T$, then $x_1 = x_2 = x_3 = x_4 \in \operatorname{zer}(A_1 + A_2 + A_3 + A_4)$.
- However, fixed point iteration of T does not always converge.
- In fact, it only converges when A_4 is 1-strongly monotone.

Extending Ryu's Splitting from $n = 3$ to $n = 4$

We tried (unsuccessfully) to extend Ryu's scheme to $n \geq 4$ operators.

Let $\gamma \in (0,1)$. Define $T\colon \mathcal{H}^3 \to \mathcal{H}^3$ according to

$$T(\mathbf{z}) = \mathbf{z} + \gamma \begin{pmatrix} x_4 - x_1 \\ x_4 - x_2 \\ x_4 - x_3 \end{pmatrix} \quad \text{where} \quad \begin{cases} x_1 = J_{A_1}(z_1) \\ x_2 = J_{A_2}(z_2 + x_1) \\ x_3 = J_{A_3}(z_3 + x_2 - x_1) \\ x_4 = J_{A_4}(x_1 - z_1 + x_2 - z_2 + x_3 - z_3), \end{cases}$$

- Possible four operator extension of Ryu's splitting (red terms new).
- If $\mathbf{z} \in \operatorname{Fix} T$, then $x_1 = x_2 = x_3 = x_4 \in \operatorname{zer}(A_1 + A_2 + A_3 + A_4)$.
- However, fixed point iteration of T does not always converge.
- In fact, it only converges when A_4 is 1-strongly monotone.

Extending Ryu's Splitting from $n = 3$ to $n = 4$

We tried (unsuccessfully) to extend Ryu's scheme to $n \geq 4$ operators.

Let $\gamma \in (0,1)$. Define $T\colon \mathcal{H}^3 \to \mathcal{H}^3$ according to

$$T(\mathbf{z}) = \mathbf{z} + \gamma \begin{pmatrix} x_4 - x_1 \\ x_4 - x_2 \\ x_4 - x_3 \end{pmatrix} \quad \text{where} \quad \begin{cases} x_1 = J_{A_1}(z_1) \\ x_2 = J_{A_2}(z_2 + x_1) \\ x_3 = J_{A_3}(z_3 + x_2 - x_1) \\ x_4 = J_{A_4}(x_1 - z_1 + x_2 - z_2 + x_3 - z_3), \end{cases}$$

- Possible four operator extension of Ryu's splitting (red terms new).
- If $\mathbf{z} \in \operatorname{Fix} T$, then $x_1 = x_2 = x_3 = x_4 \in \operatorname{zer}(A_1 + A_2 + A_3 + A_4)$.
- However, fixed point iteration of T does not always converge.
- In fact, it only converges when A_4 is 1-strongly monotone.

Searching for New Methods

Let $\mathbf{z} = (z_1, \ldots, z_{n-1}) \in \mathcal{H}^{n-1}$ and $\mathbf{x} = (x_1, \ldots, x_n), \mathbf{y} = (y_1, \ldots, y_n) \in \mathcal{H}$.

Consider candidates for the operator T_A of the form

$$T_A(\mathbf{z}) = \mathbf{z} + \gamma M \mathbf{x} \quad \text{where} \quad x_i = J_{A_i}(y_i) \quad \forall i \in \{1, \ldots, n\}$$

for some matrix $M \in \mathbb{R}^{(n-1)\times n}$.

Assuming T_A is a frugal resolvent splitting, we can deduce that:

1. $\mathbf{z} \in \operatorname{Fix} T \iff \mathbf{x} \in \ker M$.
2. If $\mathbf{x} \in \ker M$, then $x^* := x_1 = \cdots = x_n$.

So after choosing such an M, we need to investigate expressions for $\mathbf{y}$:

3. If x^* is a solution, then $x^* = \frac{1}{n}\sum_{i=1}^{n} y_i^*$ (by sol'n map proposition).
4. y_i must be a linear combination of $z_1, \ldots, z_{n-1}$ and $x_1, \ldots, x_{i-1}$.
5. In addition, need the resulting fixed point iteration to be convergent!

Searching for New Methods

Let $\mathbf{z} = (z_1, \dots, z_{n-1}) \in \mathcal{H}^{n-1}$ and $\mathbf{x} = (x_1, \dots, x_n), \mathbf{y} = (y_1, \dots, y_n) \in \mathcal{H}$.

Consider candidates for the operator T_A of the form

$$T_A(\mathbf{z}) = \mathbf{z} + \gamma M\mathbf{x} \quad \text{where} \quad x_i = J_{A_i}(y_i) \quad \forall i \in \{1, \dots, n\}$$

for some matrix $M \in \mathbb{R}^{(n-1)\times n}$.

Assuming T_A is a frugal resolvent splitting, we can deduce that:

1. $\mathbf{z} \in \operatorname{Fix} T \iff \mathbf{x} \in \ker M$.
2. If $\mathbf{x} \in \ker M$, then $x^* := x_1 = \cdots = x_n$.

So after choosing such an M, we need to investigate expressions for $\mathbf{y}$:

3. If x^* is a solution, then $x^* = \frac{1}{n}\sum_{i=1}^n y_i^*$ (by sol'n map proposition).
4. y_i must be a linear combination of $z_1, \dots, z_{n-1}$ and $x_1, \dots, x_{i-1}$.
5. In addition, need the resulting fixed point iteration to be convergent!

Searching for New Methods

Let $\mathbf{z} = (z_1, \ldots, z_{n-1}) \in \mathcal{H}^{n-1}$ and $\mathbf{x} = (x_1, \ldots, x_n), \mathbf{y} = (y_1, \ldots, y_n) \in \mathcal{H}$.

Consider candidates for the operator T_A of the form

$$T_A(\mathbf{z}) = \mathbf{z} + \gamma M\mathbf{x} \quad \text{where} \quad x_i = J_{A_i}(y_i) \quad \forall i \in \{1, \ldots, n\}$$

for some matrix $M \in \mathbb{R}^{(n-1)\times n}$.

Assuming T_A is a frugal resolvent splitting, we can deduce that:

1. $\mathbf{z} \in \operatorname{Fix} T \iff \mathbf{x} \in \ker M$.
2. If $\mathbf{x} \in \ker M$, then $x^* := x_1 = \cdots = x_n$.

So after choosing such an M, we need to investigate expressions for $\mathbf{y}$:

3. If x^* is a solution, then $x^* = \frac{1}{n}\sum_{i=1}^{n} y_i^*$ (by sol'n map proposition).
4. y_i must be a linear combination of $z_1, \ldots, z_{n-1}$ and $x_1, \ldots, x_{i-1}$.
5. In addition, need the resulting fixed point iteration to be convergent!

A New Family of Resolvent Splitting

We propose the following family of splitting algorithms.

Let $\gamma \in (0,1)$ and $T_A\colon \mathcal{H}^{n-1} \to \mathcal{H}^{n-1}$ be given by

$$T_A(\mathbf{z}) = \mathbf{z} + \gamma \begin{pmatrix} x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \end{pmatrix},$$

where $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{H}^n$ is given by

$$\begin{cases} x_1 = J_{A_1}(z_1), \\ x_i = J_{A_i}(z_i + x_{i-1} - z_{i-1}) & \forall i \in \{2, \dots, n-1\} \\ x_n = J_{A_n}(x_1 + x_{n-1} - z_{n-1}). \end{cases}$$

- If $n = 2$, then T_A is the same as T_{DR} for A_1 and A_2.
- When $n = 3$, it is different to Ryu's splitting method.

A New Family of Resolvent Splitting

We propose the following family of splitting algorithms.

Let $\gamma \in (0,1)$ and $T_A \colon \mathcal{H}^{n-1} \to \mathcal{H}^{n-1}$ be given by

$$T_A(\mathbf{z}) = \mathbf{z} + \gamma \begin{pmatrix} x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \end{pmatrix},$$

where $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{H}^n$ is given by

$$\begin{cases} x_1 = J_{A_1}(z_1), \\ x_i = J_{A_i}(z_i + x_{i-1} - z_{i-1}) & \forall i \in \{2, \dots, n-1\} \\ x_n = J_{A_n}(x_1 + x_{n-1} - z_{n-1}). \end{cases}$$

- If $n = 2$, then T_A is the same as T_{DR} for A_1 and A_2.
- When $n = 3$, it is different to Ryu's splitting method.

A New Family of Resolvent Splitting

We propose the following family of splitting algorithms.

Let $\gamma \in (0,1)$ and $T_A \colon \mathcal{H}^{n-1} \to \mathcal{H}^{n-1}$ be given by

$$T_A(\mathbf{z}) = \mathbf{z} + \gamma \begin{pmatrix} x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \end{pmatrix},$$

where $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{H}^n$ is given by

$$\begin{cases} x_1 = J_{A_1}(z_1), \\ x_i = J_{A_i}(z_i + x_{i-1} - z_{i-1}) & \forall i \in \{2, \dots, n-1\} \\ x_n = J_{A_n}(x_1 + x_{n-1} - z_{n-1}). \end{cases}$$

- If $n = 2$, then T_A is the same as T_{DR} for A_1 and A_2.
- When $n = 3$, it is different to Ryu's splitting method.

A New Family of Resolvent Splitting

We propose the following family of splitting algorithms.

Let $\gamma \in (0,1)$ and $T_A\colon \mathcal{H}^{n-1} \to \mathcal{H}^{n-1}$ be given by

$$T_A(\mathbf{z}) = \mathbf{z} + \gamma \begin{pmatrix} x_2 - x_1 \\ x_3 - x_2 \\ \vdots \\ x_n - x_{n-1} \end{pmatrix},$$

where $\mathbf{x} = (x_1, \dots, x_n) \in \mathcal{H}^n$ is given by

$$\begin{cases} x_1 = J_{A_1}(z_1), \\ x_i = J_{A_i}(z_i + x_{i-1} - z_{i-1}) \qquad \forall i \in \{2, \dots, n-1\} \\ x_n = J_{A_n}(x_1 + x_{n-1} - z_{n-1}). \end{cases}$$

- If $n = 2$, then T_A is the same as T_{DR} for A_1 and A_2.
- When $n = 3$, it is different to Ryu's splitting method.

The Fixed Point Set of T_A

Lemma (Malitsky–T.)

Let $n \geq 2$, $A = (A_1, \ldots, A_n) \in \mathcal{A}_n$ and $\gamma > 0$. Then we have:

1. $\operatorname{Fix} T_A \neq \emptyset \iff \operatorname{zer}\left(\sum_{i=1}^n A_i\right) \neq \emptyset$.
2. If $\mathbf{z} = (z_1, \ldots, z_{n-1}) \in \operatorname{Fix} T_A$, then $x \in \operatorname{zer}\left(\sum_{i=1}^n A_i\right)$ where

$$x := J_{A_1}(z_1) = J_{A_i}(z_i + x - z_{i-1}) = J_{A_n}(2x - z_{n-1}) \tag{1}$$

for all $i \in \{2, \ldots, n-1\}$.

- Shows that T_A can be used to define a fixed point encoding for $\mathcal{A}_n$.
- Any resolvents in (1) can be used solution map. For instance:

$$S_A(\mathbf{z}) := J_{A_1}(z_1).$$

- How about (weak) convergence of the fixed point iteration?

The Fixed Point Set of T_A

Lemma (Malitsky–T.)

Let $n \geq 2$, $A = (A_1, \dots, A_n) \in \mathcal{A}_n$ and $\gamma > 0$. Then we have:

1. $\operatorname{Fix} T_A \neq \emptyset \iff \operatorname{zer}\left(\sum_{i=1}^n A_i\right) \neq \emptyset$.
2. If $\mathbf{z} = (z_1, \dots, z_{n-1}) \in \operatorname{Fix} T_A$, then $x \in \operatorname{zer}\left(\sum_{i=1}^n A_i\right)$ where

$$x := J_{A_1}(z_1) = J_{A_i}(z_i + x - z_{i-1}) = J_{A_n}(2x - z_{n-1}) \qquad (1)$$

for all $i \in \{2, \dots, n-1\}$.

- Shows that T_A can be used to define a fixed point encoding for $\mathcal{A}_n$.
- Any resolvents in (1) can be used solution map. For instance:

$$S_A(\mathbf{z}) := J_{A_1}(z_1).$$

- How about (weak) convergence of the fixed point iteration?

Nonexpansivity Properties of T_A

Lemma (Malitsky–T.)

Let $n \geq 2$, $A = (A_1, \dots, A_n) \in \mathcal{A}_n$ and $\gamma > 0$. Then we have

$$\|T_A(\mathbf{z}) - T_A(\bar{\mathbf{z}})\|^2 + \frac{1-\gamma}{\gamma}\|(\mathrm{Id} - T_A)(\mathbf{z}) - (\mathrm{Id} - T_A)(\bar{\mathbf{z}})\|^2 + \frac{1}{\gamma}\Big\|\sum_{i=1}^{n-1}(\mathrm{Id} - T_A)(\mathbf{z})_i - \sum_{i=1}^{n-1}(\mathrm{Id} - T_A)(\bar{\mathbf{z}})_i\Big\|^2 \leq \|\mathbf{z} - \bar{\mathbf{z}}\|^2,$$

where $\mathbf{z} = (z_1, \dots, z_n) \in \mathcal{H}^{n-1}$ and $\bar{\mathbf{z}} = (\bar{z}_1, \dots, \bar{z}_n) \in \mathcal{H}^{n-1}$.

- If $\gamma \in (0,1)$, then the operator T_A is γ-averaged nonexpansive.
- Counter-example in paper: in general, we cannot take $\gamma = 1$.
- However, if $n = 2$, then inequality simplifies and can take $\gamma \in (0,2)$.

Nonexpansivity Properties of T_A

Lemma (Malitsky–T.)

Let $n \geq 2$, $A = (A_1, \dots, A_n) \in \mathcal{A}_n$ and $\gamma > 0$. Then we have

$$\|T_A(\mathbf{z}) - T_A(\bar{\mathbf{z}})\|^2 + \frac{1-\gamma}{\gamma}\|(\mathrm{Id} - T_A)(\mathbf{z}) - (\mathrm{Id} - T_A)(\bar{\mathbf{z}})\|^2 + \frac{1}{\gamma}\Big\|\sum_{i=1}^{n-1}(\mathrm{Id} - T_A)(\mathbf{z})_i - \sum_{i=1}^{n-1}(\mathrm{Id} - T_A)(\bar{\mathbf{z}})_i\Big\|^2 \leq \|\mathbf{z} - \bar{\mathbf{z}}\|^2,$$

where $\mathbf{z} = (z_1, \dots, z_n) \in \mathcal{H}^{n-1}$ and $\bar{\mathbf{z}} = (\bar{z}_1, \dots, \bar{z}_n) \in \mathcal{H}^{n-1}$.

- If $\gamma \in (0, 1)$, then the operator T_A is γ-averaged nonexpansive.
- Counter-example in paper: in general, we cannot take $\gamma = 1$.
- However, if $n = 2$, then inequality simplifies and can take $\gamma \in (0, 2)$.

Main Convergence Theorem

Fixed point algorithm:

$$\mathbf{z}^{k+1} = T_A(\mathbf{z}^k) = \mathbf{z}^k + \gamma \begin{pmatrix} x_2^k - x_1^k \\ x_3^k - x_2^k \\ \vdots \\ x_n^k - x_{n-1}^k \end{pmatrix} \text{ where } \begin{cases} x_1^k = J_{A_1}(z_1^k), \\ x_i^k = J_{A_i}(z_i^k + x_{i-1}^k - z_{i-1}^k \\ x_n^k = J_{A_n}(x_1^k + x_{n-1}^k - z_{n-1}^k). \end{cases}$$

Theorem (Malitsky–T.)

Let $n \geq 2$, $A = (A_1, \dots, A_n) \in \mathcal{A}_n$ with $\operatorname{zer}\left(\sum_{i=1}^n A_i\right) \neq \emptyset$, and $\gamma \in (0,1)$. Given $\mathbf{z}^0 \in \mathcal{H}^{n-1}$, let $(\mathbf{z}^k) \subseteq \mathcal{H}^{n-1}$ and $(\mathbf{x}^k) \subseteq \mathcal{H}^n$ be given as above. Then, the following assertions hold.

1. $\mathbf{z}^k \rightharpoonup \mathbf{z} \in \operatorname{Fix} T_A$.
2. $\mathbf{x}^k \rightharpoonup (x, \dots, x) \in \mathcal{H}^n$ with $x \in \operatorname{zer}\left(\sum_{i=1}^n A_i\right)$.

Main Convergence Theorem

Fixed point algorithm:

$$\mathbf{z}^{k+1} = T_A(\mathbf{z}^k) = \mathbf{z}^k + \gamma \begin{pmatrix} x_2^k - x_1^k \\ x_3^k - x_2^k \\ \vdots \\ x_n^k - x_{n-1}^k \end{pmatrix} \text{ where } \begin{cases} x_1^k = J_{A_1}(z_1^k), \\ x_i^k = J_{A_i}(z_i^k + x_{i-1}^k - z_{i-1}^k \\ x_n^k = J_{A_n}(x_1^k + x_{n-1}^k - z_{n-1}^k). \end{cases}$$

Some further refinements:

- If $A_2, \ldots, A_n$ are uniformly monotone (but not necessarily A_1), then $(\mathbf{x}^k)$ converges strongly. This holds in the limiting case $\gamma = 1$
  - In contrast, Peaceman–Rachford splitting (= limiting case of DR) in the product space requires all operators to be uniformly monotone.
- If A_i's are normal cones to subspaces S_i, then $(\mathbf{x}^k)$ converges strongly and $x := P_{\cap_{i=1}^n S_i}\left(\frac{1}{n-1}\sum_{i=1}^n z_i^0\right)$. (Bauschke–Singh–Wang)

Bauschke, H. H., Singh, S., & Wang, X. (2021). The splitting algorithms by Ryu and by Malitsky–Tam applied to normal cones of linear subspaces converge strongly to the projection onto the intersection. *arXiv:2109.11072*.

Minimal Lifting for Frugal Resolvent Splitting

Conjecture

The minimal amount of lifting is given by $d^*(n) = n - 1$ for $n \geq 2$.

Combining everything in this talk so far, gives the following answer.

Corollary (Malitsky–T.)

Suppose $n \geq 2$. There exists a convergent frugal resolvent splitting for $\mathcal{A}_n$ with $(n-1)$-fold lifting. Moreover, this is the minimal amount of lifting possible with frugal resolvent splittings for $\mathcal{A}_n$.

Algorithmic consequences:

- In general, it is not possible to do too much better than the product space: n-fold lifting vs $(n-1)$-fold lifting.
- For small n, the difference is more significant. For large n, less so.

Minimal Lifting for Frugal Resolvent Splitting

Conjecture

The minimal amount of lifting is given by $d^*(n) = n - 1$ for $n \geq 2$.

Combining everything in this talk so far, gives the following answer.

Corollary (Malitsky–T.)

Suppose $n \geq 2$. There exists a convergent frugal resolvent splitting for $\mathcal{A}_n$ with $(n-1)$-fold lifting. Moreover, this is the minimal amount of lifting possible with frugal resolvent splittings for $\mathcal{A}_n$.

Algorithmic consequences:

- In general, it is not possible to do too much better than the product space: n-fold lifting vs $(n-1)$-fold lifting.
- For small n, the difference is more significant. For large n, less so.

Minimal Lifting for Frugal Resolvent Splitting

Conjecture

The minimal amount of lifting is given by $d^*(n) = n - 1$ for $n \geq 2$.

Combining everything in this talk so far, gives the following answer.

Corollary (Malitsky–T.)

Suppose $n \geq 2$. There exists a convergent frugal resolvent splitting for $\mathcal{A}_n$ with $(n-1)$-fold lifting. Moreover, this is the minimal amount of lifting possible with frugal resolvent splittings for $\mathcal{A}_n$.

Algorithmic consequences:

- In general, it is not possible to do too much better than the product space: n-fold lifting vs $(n-1)$-fold lifting.
- For small n, the difference is more significant. For large n, less so.

Concluding Remarks

Concluding Remarks

In this work, we have shown:

- Minimal amount of lifting for n-operator inclusion is $n-1$.
- New n-operator resolvent splitting method that generalises DR.

Directions for future work:

- Finer properties of new splitting algorithm (*e.g.*, inconsistent prob).
- How does frugality affect the amount of lifting needed? Trade off?
- Characterise all frugal resolvent splittings for n-operators?

# of Operators	Minimal Amount of Lifting	Algorithm(s)
1	1	Proximal Point algorithm
2	1	Douglas–Rachford algorithm
3	2	Ryu's algorithm + This Work + Others?
$n \geq 2$	$n-1$	This Work + Others?

⋆ Perhaps we should more often examine what might *not be possible*

Resolvent splitting for sums of monotone operators with minimal lifting
with Yura Malitsky. *arXiv:2108.02897*.

Concluding Remarks

In this work, we have shown:

- Minimal amount of lifting for n-operator inclusion is $n-1$.
- New n-operator resolvent splitting method that generalises DR.

Directions for future work:

- Finer properties of new splitting algorithm (*e.g.*, inconsistent prob).
- How does frugality affect the amount of lifting needed? Trade off?
- Characterise all frugal resolvent splittings for n-operators?

# of Operators	Minimal Amount of Lifting	Algorithm(s)
1	1	Proximal Point algorithm
2	1	Douglas–Rachford algorithm
3	2	Ryu's algorithm + This Work + Others?
$n \geq 2$	$n-1$	This Work + Others?

⋆ Perhaps we should more often examine what might *not be possible*

Resolvent splitting for sums of monotone operators with minimal lifting
with Yura Malitsky. *arXiv:2108.02897*.

Concluding Remarks

In this work, we have shown:

- Minimal amount of lifting for n-operator inclusion is $n-1$.
- New n-operator resolvent splitting method that generalises DR.

Directions for future work:

- Finer properties of new splitting algorithm (*e.g.*, inconsistent prob).
- How does frugality affect the amount of lifting needed? Trade off?
- Characterise all frugal resolvent splittings for n-operators?

# of Operators	**Minimal Amount of Lifting**	**Algorithm(s)**
1	1	Proximal Point algorithm
2	1	Douglas–Rachford algorithm
3	2	Ryu's algorithm + This Work + Others?
$n \geq 2$	$n-1$	This Work + Others?

⋆ Perhaps we should more often examine what might *not be possible*

Resolvent splitting for sums of monotone operators with minimal lifting
with Yura Malitsky. *arXiv:2108.02897*.