Golden Ratio Algorithms for Variational Inequalities

Yura Malitsky

University of Göttingen

Splitting Algorithms, Modern Operator Theory, and Applications
Oaxaca, September–2017

Introduction

Variational inequality problem (VIP):

find $z^* \in Z = \mathbb{R}^d$ such that

$$\langle F(z^*), z - z^* \rangle + G(z) - G(z^*) \geq 0 \quad \forall z \in Z,$$

where

- $F\colon Z \to Z$ is monotone: $\langle F(z) - F(z'), z - z' \rangle \geq 0 \quad \forall z, z'$
- $G\colon Z \to (-\infty, +\infty]$ is a proper lsc convex function

Introduction

Variational inequality problem (VIP):

find $z^* \in Z = \mathbb{R}^d$ such that

$$\langle F(z^*), z - z^* \rangle + G(z) - G(z^*) \geq 0 \quad \forall z \in Z,$$

where

- $F\colon Z \to Z$ is monotone: $\langle F(z) - F(z'), z - z' \rangle \geq 0 \quad \forall z, z'$
- $G\colon Z \to (-\infty, +\infty]$ is a proper lsc convex function

VIP as a monotone operator inclusion:

$$0 \in F(z^*) + \partial G(z^*)$$

Motivation–1

Composite minimization:

$$\min_x f(x) + g(x)$$

- $f: X \to \mathbb{R}$ is a convex smooth function
- $g: X \to (-\infty, +\infty]$ is a proper lsc convex function

Motivation–1

Composite minimization:

$$\min_x f(x) + g(x)$$

- $f: X \to \mathbb{R}$ is a convex smooth function
- $g: X \to (-\infty, +\infty]$ is a proper lsc convex function

First-order optimality condition:

$$\langle \nabla f(x^*), x - x^* \rangle + g(x) - g(x^*) \geq 0 \quad \forall x \in X.$$

Motivation–2

Saddle point problem:

$$\min_x \max_y \mathcal{L}(x, y) := g(x) + K(x, y) - f^*(y)$$

- $K \colon X \times Y \to \mathbb{R}$ is smooth convex-concave
- $g \colon X \to (-\infty, +\infty]$, $f \colon Y \to (-\infty, +\infty]$ are convex lsc

Motivation–2

Saddle point problem:

$$\min_x \max_y \mathcal{L}(x,y) := g(x) + K(x,y) - f^*(y)$$

- $K\colon X \times Y \to \mathbb{R}$ is smooth convex-concave
- $g\colon X \to (-\infty, +\infty]$, $f\colon Y \to (-\infty, +\infty]$ are convex lsc

Let

$$z = \begin{pmatrix} x \\ y \end{pmatrix}, \quad F(z) = \begin{pmatrix} \nabla_x K(x,y) \\ -\nabla_y K(x,y) \end{pmatrix}, \quad G(z) = g(x) + f^*(y)$$

Motivation–2

Saddle point problem:

$$\min_x \max_y \mathcal{L}(x,y) := g(x) + K(x,y) - f^*(y)$$

- $K\colon X \times Y \to \mathbb{R}$ is smooth convex-concave
- $g\colon X \to (-\infty, +\infty]$, $f\colon Y \to (-\infty, +\infty]$ are convex lsc

Let

$$z = \begin{pmatrix} x \\ y \end{pmatrix}, \quad F(z) = \begin{pmatrix} \nabla_x K(x,y) \\ -\nabla_y K(x,y) \end{pmatrix}, \quad G(z) = g(x) + f^*(y)$$

First-order optimality condition:

$$\langle F(z^*), z - z^* \rangle + G(z) - G(z^*) \geq 0 \quad \forall z \in X \times Y$$

Goal

find $z^* \in Z$ such that

$$\langle F(z^*), z - z^* \rangle + G(z) - G(z^*) \geq 0 \quad \forall z \in Z,$$

where

- $F\colon Z \to Z$ is monotone, (L–Lipschitz) continuous
- $G\colon Z \to (-\infty, +\infty]$ is a proper lsc convex

Goal

find $z^* \in Z$ such that

$$\langle F(z^*), z - z^* \rangle + G(z) - G(z^*) \geq 0 \quad \forall z \in Z,$$

where

- $F\colon Z \to Z$ is monotone, (L–Lipschitz) continuous
- $G\colon Z \to (-\infty, +\infty]$ is a proper lsc convex
- G is prox-friendly: $\operatorname{prox}_G \equiv (\operatorname{Id} + \partial G)^{-1}$ is "easy" to compute
- F is not

How we can solve such problems?

Classic

Forward-backward method:

$$z^{k+1} = \operatorname{prox}_{\lambda G}(z^k - \lambda F(z^k))$$

Classic

Forward-backward method:

$$z^{k+1} = \operatorname{prox}_{\lambda G}(z^k - \lambda F(z^k))$$

or for saddle point problems:

$$\begin{aligned} x^{k+1} &= \operatorname{prox}_{\lambda g}(x^k - \lambda \nabla_x K(x^k, y^k)) \\ y^{k+1} &= \operatorname{prox}_{\lambda f^*}(y^k + \lambda \nabla_y K(x^k, y^k)) \end{aligned}$$

Classic

Forward-backward method:

$$z^{k+1} = \operatorname{prox}_{\lambda G}(z^k - \lambda F(z^k))$$

or for saddle point problems:

$$\begin{aligned} x^{k+1} &= \operatorname{prox}_{\lambda g}(x^k - \lambda \nabla_x K(x^k, y^k)) \\ y^{k+1} &= \operatorname{prox}_{\lambda f^*}(y^k + \lambda \nabla_y K(x^k, y^k)) \end{aligned}$$

Arrow-Hurwicz method:

$$\begin{aligned} x^{k+1} &= \operatorname{prox}_{\lambda g}(x^k - \lambda \nabla_x K(x^k, y^k)) \\ y^{k+1} &= \operatorname{prox}_{\lambda f^*}(y^k + \lambda \nabla_y K(x^{k+1}, y^k)) \end{aligned}$$

Classic

Forward-backward method:

$$z^{k+1} = \operatorname{prox}_{\lambda G}(z^k - \lambda F(z^k))$$

or for saddle point problems:

$$\begin{aligned} x^{k+1} &= \operatorname{prox}_{\lambda g}(x^k - \lambda \nabla_x K(x^k, y^k)) \\ y^{k+1} &= \operatorname{prox}_{\lambda f^*}(y^k + \lambda \nabla_y K(x^k, y^k)) \end{aligned}$$

Arrow-Hurwicz method:

$$\begin{aligned} x^{k+1} &= \operatorname{prox}_{\lambda g}(x^k - \lambda \nabla_x K(x^k, y^k)) \\ y^{k+1} &= \operatorname{prox}_{\lambda f^*}(y^k + \lambda \nabla_y K(x^{k+1}, y^k)) \end{aligned}$$

Convergence: under quite restrictive assumptions

Classic that works

Extragradient method (Korpelevich, 1976):

$$\begin{aligned} w^{k+1} &= \text{prox}_{\lambda G}(z^k - \lambda F(z^k)) \\ z^{k+1} &= \text{prox}_{\lambda G}(z^k - \lambda F(w^{k+1})) \end{aligned}$$

Convergence: $\lambda < \frac{1}{L}$

Popov's method, 1978:

$$\begin{aligned} w^{k+1} &= \text{prox}_{\lambda G}(w^k - \lambda F(z^k)) \\ z^{k+1} &= \text{prox}_{\lambda G}(w^{k+1} - \lambda F(z^k)) \end{aligned}$$

Convergence: $\lambda < \frac{\sqrt{2}-1}{L}$

Recent methods

Forward-backward-forward method (Tseng, 2000):

$$w^{k+1} = \operatorname{prox}_{\lambda G}(z^k - \lambda F(z^k))$$
$$z^{k+1} = w^{k+1} + \lambda(F(z^k) - F(w^{k+1}))$$

Convergence: $\lambda < \frac{1}{L}$

Recent methods

Forward-backward-forward method (Tseng, 2000):

$$w^{k+1} = \operatorname{prox}_{\lambda G}(z^k - \lambda F(z^k))$$
$$z^{k+1} = w^{k+1} + \lambda(F(z^k) - F(w^{k+1}))$$

Convergence: $\lambda < \frac{1}{L}$

Proximal reflected gradient method, (M. 2015):

$$z^{k+1} = \operatorname{prox}_{\lambda G}(z^k - \lambda F(2z^k - z^{k-1})).$$

Convergence: $\lambda < \frac{\sqrt{2}-1}{L}$

Golden Ratio Algorithm (GRAAL)

Find $z^* \in Z$ such that

$$\langle F(z^*), z - z^* \rangle + G(z) - G(z^*) \geq 0 \quad \forall z \in Z$$

Let $\varphi = \frac{\sqrt{5}+1}{2} = 1.618\ldots$

Golden Ratio Algorithm (GRAAL)

Find $z^* \in Z$ such that

$$\langle F(z^*), z - z^* \rangle + G(z) - G(z^*) \geq 0 \quad \forall z \in Z$$

Let $\varphi = \frac{\sqrt{5}+1}{2} = 1.618\dots$

GRAAL:

$$\bar{z}^k = \frac{(\varphi - 1)z^k + \bar{z}^{k-1}}{\varphi}$$
$$z^{k+1} = \operatorname{prox}_{\lambda G}(\bar{z}^k - \lambda F(z^k))$$

Convergence: $\lambda \leq \frac{\varphi}{2L}$

Linesearch

Why do we need non-stationary methods?

Linesearch

Why do we need non-stationary methods?

- expensive to compute L

Linesearch

Why do we need non-stationary methods?

- expensive to compute L
- conservative estimation provides smaller steps

Linesearch

Why do we need non-stationary methods?

- expensive to compute L
- conservative estimation provides smaller steps
- F might be non-Lipschitz

Linesearch

Why do we need non-stationary methods?

- expensive to compute L
- conservative estimation provides smaller steps
- F might be non-Lipschitz
- local information of F may give larger steps

Linesearch

Why do we need non-stationary methods?

- expensive to compute L
- conservative estimation provides smaller steps
- F might be non-Lipschitz
- local information of F may give larger steps

[Khobotov, Konnov, Solodov, Tseng, Iusem, Svaiter] et.al.

Linesearch

Why do we need non-stationary methods?

- expensive to compute L
- conservative estimation provides smaller steps
- F might be non-Lipschitz
- local information of F may give larger steps

[Khobotov, Konnov, Solodov, Tseng, Iusem, Svaiter] et.al.

Algorithms with linesearch are more expensive!

Linesearch

Why do we need non-stationary methods?

- expensive to compute L
- conservative estimation provides smaller steps
- F might be non-Lipschitz
- local information of F may give larger steps

[Khobotov, Konnov, Solodov, Tseng, Iusem, Svaiter] et.al.

Algorithms with linesearch are more expensive!

Tseng's method with linesearch:

find λ_k s.t.

$$w^{k+1} = \text{prox}_{\lambda_k G}(z^k - \lambda_k F(z^k))$$
$$z^{k+1} = w^{k+1} + \lambda_k (F(z^k) - F(w^{k+1}))$$

until

$$\lambda_k \|F(z^{k+1}) - F(z^k)\| \leq \|z^{k+1} - z^k\|$$

Explicit Golden Ratio Algorithm

Initialization: Choose $z_0, z_1 \in Z$, $\lambda_0 > 0$, $\phi \in (1, \varphi]$.
Set $\theta_0 = 1$, $r = \frac{1}{\phi} + \frac{1}{\phi^2}$.

Main iteration:

1. Find the stepsize:

$$\lambda_k = \min\Big\{r\lambda_{k-1},\ \frac{\phi\theta_{k-1}}{4\lambda_{k-1}} \frac{\|z^k - z^{k-1}\|^2}{\|F(z^k) - F(z^{k-1})\|^2}\Big\}$$

2. Compute next iterates:

$$\bar{z}^k = \frac{(\phi - 1)z^k + \bar{z}^{k-1}}{\phi}$$
$$z^{k+1} = \operatorname{prox}_{\lambda_k G}(\bar{z}^k - \lambda_k F(z^k)).$$

3. Update: $\theta_k = \dfrac{\phi\lambda_k}{\lambda_{k-1}}$.

Explicit Golden Ratio Algorithm

Initialization: Choose $z_0, z_1 \in Z$, $\lambda_0 > 0$, $\phi \in (1, \varphi]$.
Set $\theta_0 = 1$, $r = \frac{1}{\phi} + \frac{1}{\phi^2}$.

Main iteration:

1. Find the stepsize:

$$\lambda_k = \min\left\{ r\lambda_{k-1},\ \frac{\phi\theta_{k-1}}{4\lambda_{k-1}} \frac{\|z^k - z^{k-1}\|^2}{\|F(z^k) - F(z^{k-1})\|^2} \right\}$$

2. Compute next iterates:

$$\bar{z}^k = \frac{(\phi - 1)z^k + \bar{z}^{k-1}}{\phi}$$
$$z^{k+1} = \operatorname{prox}_{\lambda_k G}(\bar{z}^k - \lambda_k F(z^k)).$$

3. Update: $\theta_k = \dfrac{\phi\lambda_k}{\lambda_{k-1}}$.

Explicit Golden Ratio Algorithm

Initialization: Choose $z_0, z_1 \in Z$, $\lambda_0 > 0$, $\phi \in (1, \varphi]$.
Set $\theta_0 = 1$, $r = \frac{1}{\phi} + \frac{1}{\phi^2}$.

Main iteration:

1. Find the stepsize:

$$\lambda_k = \min\left\{ r\lambda_{k-1},\ \frac{\phi\theta_{k-1}}{4\lambda_{k-1}} \frac{\|z^k - z^{k-1}\|^2}{\|F(z^k) - F(z^{k-1})\|^2} \right\}$$

2. Compute next iterates:

$$\bar{z}^k = \frac{(\phi - 1)z^k + \bar{z}^{k-1}}{\phi}$$
$$z^{k+1} = \operatorname{prox}_{\lambda_k G}(\bar{z}^k - \lambda_k F(z^k)).$$

3. Update: $\theta_k = \dfrac{\phi\lambda_k}{\lambda_{k-1}}$.

Explicit Golden Ratio Algorithm

Initialization: Choose $z_0, z_1 \in Z$, $\lambda_0 > 0$, $\phi \in (1, \varphi]$.
Set $\theta_0 = 1$, $r = \frac{1}{\phi} + \frac{1}{\phi^2}$.

Main iteration:

1. Find the stepsize:

$$\lambda_k = \min\left\{ r\lambda_{k-1},\ \frac{\phi\theta_{k-1}}{4\lambda_{k-1}} \frac{\|z^k - z^{k-1}\|^2}{\|F(z^k) - F(z^{k-1})\|^2} \right\}$$

2. Compute next iterates:

$$\bar{z}^k = \frac{(\phi - 1)z^k + \bar{z}^{k-1}}{\phi}$$
$$z^{k+1} = \operatorname{prox}_{\lambda_k G}(\bar{z}^k - \lambda_k F(z^k)).$$

3. Update: $\theta_k = \dfrac{\phi\lambda_k}{\lambda_{k-1}}$.

Explicit Golden Ratio Algorithm

Initialization: Choose $z_0, z_1 \in Z$, $\lambda_0 > 0$, $\phi \in (1, \varphi]$.
Set $\theta_0 = 1$, $r = \frac{1}{\phi} + \frac{1}{\phi^2}$.

Main iteration:

1. Find the stepsize:

$$\lambda_k = \min\left\{ r\lambda_{k-1},\ \frac{\phi\theta_{k-1}}{4\lambda_{k-1}} \frac{\|z^k - z^{k-1}\|^2}{\|F(z^k) - F(z^{k-1})\|^2} \right\}$$

2. Compute next iterates:

$$\bar{z}^k = \frac{(\phi - 1)z^k + \bar{z}^{k-1}}{\phi}$$
$$z^{k+1} = \operatorname{prox}_{\lambda_k G}(\bar{z}^k - \lambda_k F(z^k)).$$

3. Update: $\theta_k = \dfrac{\phi\lambda_k}{\lambda_{k-1}}$.

Explicit Golden Ratio Algorithm

Initialization: Choose $z_0, z_1 \in Z$, $\lambda_0 > 0$, $\phi \in (1, \varphi]$.
Set $\theta_0 = 1$, $r = \frac{1}{\phi} + \frac{1}{\phi^2}$.

Main iteration:

1. Find the stepsize:

$$\lambda_k = \min\left\{ r\lambda_{k-1},\ \frac{\phi\theta_{k-1}}{4\lambda_{k-1}} \frac{\|z^k - z^{k-1}\|^2}{\|F(z^k) - F(z^{k-1})\|^2} \right\}$$

2. Compute next iterates:

$$\bar{z}^k = \frac{(\phi - 1)z^k + \bar{z}^{k-1}}{\phi}$$

$$z^{k+1} = \text{prox}_{\lambda_k G}(\bar{z}^k - \lambda_k F(z^k)).$$

3. Update: $\theta_k = \dfrac{\phi\lambda_k}{\lambda_{k-1}}$.

Explicit Golden Ratio Algorithm

Benefits:

- One F, one prox_G per iteration, no linesearch

Explicit Golden Ratio Algorithm

Benefits:

- One F, one prox_G per iteration, no linesearch
- No need in Lipschitz continuity of F, only locally one

Explicit Golden Ratio Algorithm

Benefits:

- One F, one prox_G per iteration, no linesearch
- No need in Lipschitz continuity of F, only locally one
- Steps are always larger than predicted by Lipschitz constant L

Explicit Golden Ratio Algorithm

Benefits:

- One F, one prox_G per iteration, no linesearch
- No need in Lipschitz continuity of F, only locally one
- Steps are always larger than predicted by Lipschitz constant L
- Easy to implement: few lines of code

Example: Nash equlibrium

n number of firms
$x = (x_i)$ production vector
$X = \sum_i x_i$ the total sum of goods
$f_i(x_i)$ the production cost for i-th firm
$p(X)$ the inverse demand function

$$\max_{x_i \geq 0} x_i p(X) - f_i(x_i) \qquad \forall i$$

Equivalent to the VIP:

$$\text{find} \quad x^* \in \mathbb{R}^n_+ \quad \text{s.t.} \ \sum_{i=1}^{n} \langle F_i(x^*), x_i - x_i^* \rangle \geq 0, \quad \forall x \in \mathbb{R}^n_+,$$

where

$$F_i(x) = f_i'(x_i) - p(X) - x_i p'(X)$$

Example: Nash equlibrium

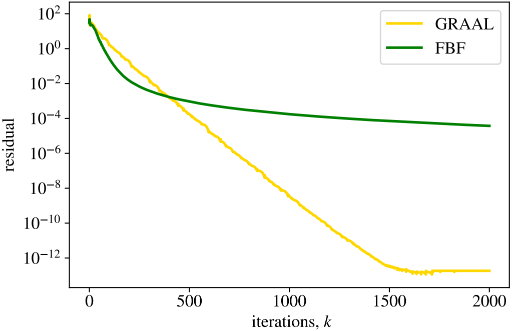
$$
\begin{aligned}
n &= 1000 \\
p(X) &= 5000^{1/\gamma} X^{-1/\gamma} && \text{inverse demand} \\
f_i(x_i) &= c_i x_i + \frac{\delta_i}{\delta_i + 1} L_i^{\frac{1}{\delta_i}} x_i^{\frac{\delta_i+1}{\delta_i}} && \text{production cost}
\end{aligned}
$$

Example: Nash equlibrium

$$n = 1000$$

$$p(X) = 5000^{1/\gamma} X^{-1/\gamma} \qquad \text{inverse demand}$$

$$f_i(x_i) = c_i x_i + \frac{\delta_i}{\delta_i + 1} L_i^{\frac{1}{\delta_i}} x_i^{\frac{\delta_i+1}{\delta_i}} \qquad \text{production cost}$$



Unbearable inefficiency of VI methods

$$\min_x \max_y g(x) + \langle Ax, y \rangle - f^*(y)$$

Unbearable inefficiency of VI methods

$$\min_x \max_y g(x) + \langle Ax, y\rangle - f^*(y)$$

Primal-dual algorithm: (Chambolle-Pock, 2011)

$$x^{k+1} = \text{prox}_{\tau g}(x^k - \tau A^* y^k)$$
$$\bar{x}^{k+1} = 2x^{k+1} - x^k$$
$$y^{k+1} = \text{prox}_{\sigma f^*}(y^k + \sigma A\bar{x}^{k+1})$$

Convergence: $\sigma\tau L^2 < 1$

Unbearable inefficiency of VI methods

$$\min_x \max_y g(x) + \langle Ax, y\rangle - f^*(y)$$

Primal-dual algorithm: (Chambolle-Pock, 2011)

$$\begin{aligned} x^{k+1} &= \operatorname{prox}_{\tau g}(x^k - \tau A^* y^k) \\ \bar{x}^{k+1} &= 2x^{k+1} - x^k \\ y^{k+1} &= \operatorname{prox}_{\sigma f^*}(y^k + \sigma A\bar{x}^{k+1}) \end{aligned}$$

Convergence: $\sigma\tau L^2 < 1$

Why primal-dual methods are much better for such problems than general algorithms for monotone VI?

Example: projection onto a polygon

$$\min_x \|x - u\|^2 \qquad \text{s.t.} \quad Ax \leq b,$$

where $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $m = 100$, $n = 1000$

Example: projection onto a polygon

$$\min_x \|x - u\|^2 \qquad \text{s.t.} \quad Ax \le b,$$

where $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $m = 100$, $n = 1000$

$$\min_x \max_y \|x - u\|^2 + \langle Ax, y\rangle - \langle b, y\rangle - \delta_{\mathbb{R}^m_+}(y)$$

Example: projection onto a polygon

$$\min_x \|x-u\|^2 \qquad \text{s.t.} \quad Ax \le b,$$

where $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $m = 100$, $n = 1000$

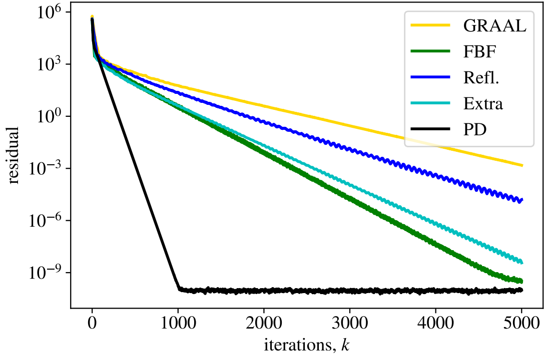
$$\min_x \max_y \underbrace{\|x-u\|^2}_{g(x)} + \langle Ax, y\rangle - \underbrace{\langle b, y\rangle - \delta_{\mathbb{R}^m_+}(y)}_{f^*(y)}$$

Example: projection onto a polygon

$$\min_x \|x - u\|^2 \qquad \text{s.t.} \quad Ax \le b,$$

where $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $m = 100$, $n = 1000$

$$\min_x \max_y \underbrace{\|x-u\|^2}_{g(x)} + \langle Ax, y\rangle - \underbrace{\langle b, y\rangle - \delta_{\mathbb{R}^m_+}(y)}_{f^*(y)}$$



Example: projection onto a polygon

$$\min_x \|x-u\|^2 \qquad \text{s.t.} \quad Ax \le b,$$

where $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $m = 100$, $n = 1000$

$$\min_x \max_y \underbrace{\|x-u\|^2}_{g(x)} + \langle Ax, y\rangle - \underbrace{\langle b, y\rangle - \delta_{\mathbb{R}^m_+}(y)}_{f^*(y)}$$

PDA

$$x^{k+1} = \operatorname{prox}_{\tau g}(x^k - \tau A^* y^k)$$
$$y^{k+1} = \operatorname{prox}_{\sigma f^*}(y^k + \sigma A\bar{x}^{k+1})$$

GRAAL

$$x^{k+1} = \operatorname{prox}_{\lambda_k g}(\bar{x}^k - \lambda_k A^* y^k)$$
$$y^{k+1} = \operatorname{prox}_{\lambda_k f^*}(\bar{y}^k + \lambda_k A x^{k+1})$$

Unbearable inefficiency of VI methods

$$\min_x \max_y g(x) + \langle Ax, y \rangle - f^*(y)$$

Why primal-dual methods usually are much better for such problems than general algorithms for VI?

- Because the former use structure.

Unbearable inefficiency of VI methods

$$\min_x \max_y g(x) + \langle Ax, y \rangle - f^*(y)$$

Why primal-dual methods usually are much better for such problems than general algorithms for VI?

- Because the former use structure.

What can we do the VI methods?

Unbearable inefficiency of VI methods

$$\min_x \max_y g(x) + \langle Ax, y \rangle - f^*(y)$$

Why primal-dual methods usually are much better for such problems than general algorithms for VI?

- Because the former use structure.

What can we do the VI methods?

Unbearable inefficiency of VI methods

$$\min_x \max_y g(x) + \langle Ax, y\rangle - f^*(y)$$

Why primal-dual methods usually are much better for such problems than general algorithms for VI?

- Because the former use structure.

What can we do the VI methods?

$$z^{k+1} = \operatorname{prox}_{\lambda G}(\bar{z}^k - \lambda F(z^k))$$

where

$$F(z) = \begin{pmatrix} A^* y \\ -Ax \end{pmatrix}, \quad G(z) = g(x) + f^*(y)$$

Unbearable inefficiency of VI methods

$$\min_x \max_y g(x) + \langle Ax, y\rangle - f^*(y)$$

Why primal-dual methods usually are much better for such problems than general algorithms for VI?

- Because the former use structure.

What can we do the VI methods?

$$z^{k+1} = \text{prox}_G^M(\bar{z}^k - M^{-1}F(z^k))$$

where

$$F(z) = \begin{pmatrix} A^*y \\ -Ax \end{pmatrix}, \quad G(z) = g(x) + f^*(y)$$

M is a positive definite matrix

But?

Before $\lambda < \frac{1}{L}$, i.e.,

$$\lambda^2 \|F(z_1) - F(z_2)\|^2 \le \|z_1 - z_2\|^2 \quad \forall z_1, z_2$$

But?

Before $\lambda < \frac{1}{L}$, i.e.,

$$\lambda^2 \|F(z_1) - F(z_2)\|^2 \le \|z_1 - z_2\|^2 \quad \forall z_1, z_2$$

Now

$$\|F(z_1) - F(z_2)\|^2_{M^{-1}} \le \|z_1 - z_2\|^2_M \quad \forall z_1, z_2$$

But?

Before $\lambda < \frac{1}{L}$, i.e.,

$$\lambda^2 \|F(z_1) - F(z_2)\|^2 \le \|z_1 - z_2\|^2 \quad \forall z_1, z_2$$

Now

$$\|F(z_1) - F(z_2)\|^2_{M^{-1}} \le \|z_1 - z_2\|^2_M \quad \forall z_1, z_2$$

We do not want to check this!

Structural Golden Ratio Algorithm

Initialization: Choose $z_0, z_1 \in Z$, $\lambda_0 > 0$, $\phi \in (1, \varphi]$. Set $\theta_0 = 1$, $r = \frac{1}{\phi} + \frac{1}{\phi^2}$.

Main iteration:

1. Find the metric:

$$\lambda_k = \min\Big\{ r\lambda_{k-1},\ \frac{\phi\theta_{k-1}}{4\lambda_{k-1}} \frac{\|z^k - z^{k-1}\|_M^2}{\|F(z^k) - F(z^{k-1})\|_{M^{-1}}^2} \Big\}$$

$$M_k = \lambda_k M$$

2. Compute the next iterates:

$$\bar{z}^k = \frac{(\phi - 1)z^k + \bar{z}^{k-1}}{\phi}$$

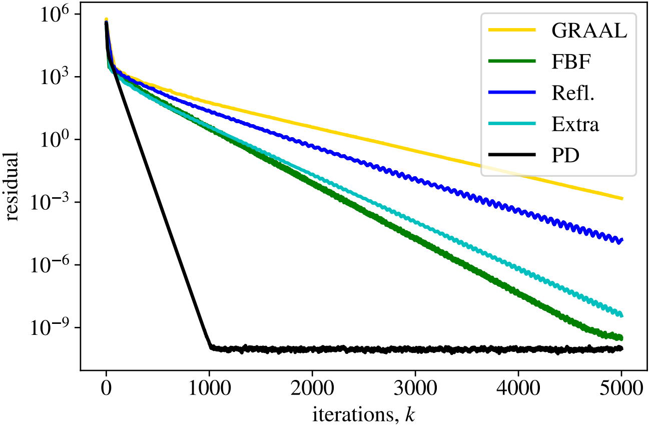
$$z^{k+1} = \mathrm{prox}_G^{M_k}(\bar{x}^k - M_k^{-1} F(z^k)).$$

3. Update: $\theta_k = \dfrac{\phi\lambda_k}{\lambda_{k-1}}$.

Example: projection onto a polygon–2

$$\min_x \|x - u\|^2 \qquad \text{s.t.} \quad Ax \le b,$$

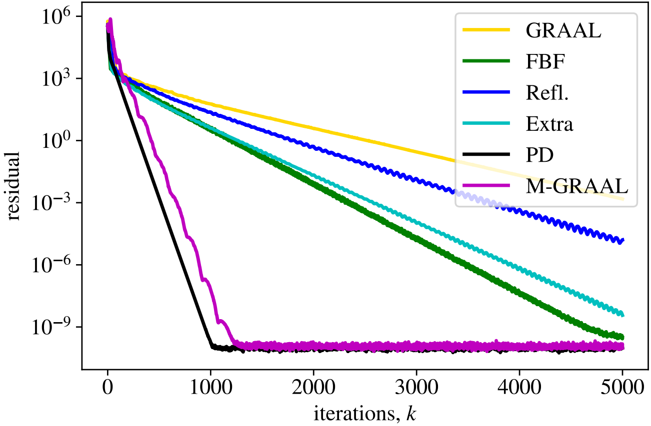
where $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $m = 100$, $n = 1000$



Example: projection onto a polygon–2

$$\min_x \|x-u\|^2 \qquad \text{s.t.} \quad Ax \le b,$$

where $A \in \mathbb{R}^{m\times n}$, $x \in \mathbb{R}^n$, $b \in \mathbb{R}^m$, $m = 100$, $n = 1000$



Rate of convergence

Dual gap function:

$$e(v) = \max_{u \in \operatorname{dom} G} \Psi(u, v) := \langle F(u), v - u \rangle + G(v) - G(u)$$

z^* is a solution of VI $\quad \Leftrightarrow \quad e(z^*) = 0$ and $z^* \in \operatorname{dom} G$

Rate of convergence

Dual gap function:

$$e(v) = \max_{u \in \operatorname{dom} G} \Psi(u, v) := \langle F(u), v - u \rangle + G(v) - G(u)$$

z^* is a solution of VI $\quad \Leftrightarrow \quad e(z^*) = 0$ and $z^* \in \operatorname{dom} G$

$$\Psi(z, Z^N) \leq O(1/N)$$

where Z^N is the ergodic sequence of (z^k).

Fixed points

Let $T\colon X \to X$

$$x = Tx \qquad \Leftrightarrow \qquad 0 = (\mathrm{Id} - T)x =: F(x)$$

Fixed points

Let $T\colon X \to X$

$$x = Tx \qquad \Leftrightarrow \qquad 0 = (\mathrm{Id} - T)x =: F(x)$$

T is nonexpansive $\|Tx - Ty\|^2 \le \|x - y\|^2$ $\Rightarrow$ F is monotone $\langle F(y) - F(x), x - y\rangle \ge 0$

Fixed points

Let $T\colon X \to X$

$$x = Tx \qquad \Leftrightarrow \qquad 0 = (\mathrm{Id} - T)x =: F(x)$$

T is nonexpansive
$\|Tx - Ty\|^2 \le \|x - y\|^2$

$\Rightarrow$

F is monotone
$\langle F(y) - F(x), x - y\rangle \ge 0$

Krasnoselskii-Mann scheme

$$x^{k+1} = \frac{x^k + Tx^k}{2}$$

Fixed points

Let $T\colon X \to X$

$$x = Tx \qquad \Leftrightarrow \qquad 0 = (\mathrm{Id} - T)x =: F(x)$$

T is nonexpansive

$\|Tx - Ty\|^2 \le \|x - y\|^2$

$\Rightarrow$

F is monotone

$\langle F(y) - F(x), x - y\rangle \ge 0$

Krasnoselskii-Mann scheme

$$x^{k+1} = \frac{x^k + Tx^k}{2}$$

Explicit GRAAL

$$x^{k+1} = \frac{(\phi - 1 - \lambda_k \phi)x^k + \bar{x}^{k-1}}{\phi} + \lambda_k Tx^k$$

Fixed points

Let $T\colon X \to X$

$$x = Tx \qquad \Leftrightarrow \qquad 0 = (\mathrm{Id} - T)x =: F(x)$$

T is nonexpansive

$$\|Tx - Ty\|^2 \le \|x - y\|^2$$

$\Rightarrow$

F is monotone

$$\langle F(y) - F(x), x - y\rangle \ge 0$$

Krasnoselskii-Mann scheme

$$x^{k+1} = \frac{x^k + Tx^k}{2}$$

Explicit GRAAL

$$x^{k+1} = \frac{(\phi - 1 - \lambda_k \phi)x^k + \bar{x}^{k-1}}{\phi} + \lambda_k Tx^k$$

Goal: to accelerate fixed point algorithms

Convex feasibility problem

$$\text{find } x \in S := \bigcap_{i=1}^m C_i$$

Let $T = \frac{1}{m}(P_{C_1} + \cdots + P_{C_m})$

If $S \neq \varnothing \quad \Rightarrow \quad x = Tx \Leftrightarrow x \in S$
if $S = \varnothing \quad \Rightarrow \quad x = Tx \Leftrightarrow x \in \operatorname{argmin}_u \sum_{i=1}^m \operatorname{dist}(u, C_i)^2 =: f(u)$

$$x^{k+1} = Tx^k$$

Krasnoselskii-Mann scheme (1953) = Cimmino method (1938) = method of parallel projections = gradient descent for $\min_u f(u)$

Tomography reconstruction

$$Ax = b,$$

$A \in \mathbb{R}^{m \times n}$ is the projection matrix, $b \in \mathbb{R}^m$ is the observed sinogram, $m = 2^{14}$, $n = 2^{16}$

CFP: find $x \in \bigcap_{i=1}^m C_i$, where $C_i = \{u \colon \langle a_i, u \rangle = b_i\}$.

$T = \frac{1}{m}(P_{C_1} + \dots + P_{C_m})$ residual$(x) = \|x - Tx\|$.

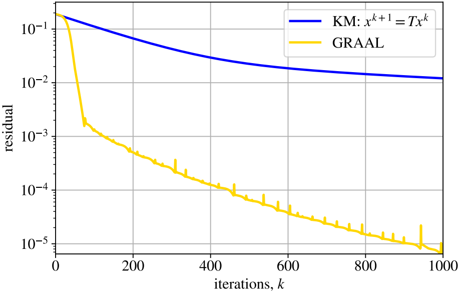
Tomography reconstruction

$$Ax = b,$$

$A \in \mathbb{R}^{m\times n}$ is the projection matrix, $b \in \mathbb{R}^m$ is the observed sinogram, $m = 2^{14}$, $n = 2^{16}$

CFP: find $x \in \bigcap_{i=1}^m C_i$, where $C_i = \{u\colon \langle a_i, u\rangle = b_i\}$.

$T = \frac{1}{m}(P_{C_1} + \cdots + P_{C_m})$ residual$(x) = \|x - Tx\|$.



GRAAL: $x^{k+1} = \frac{(\phi - 1 - \lambda_k\phi)x^k + \bar{x}^{k-1}}{\phi} + \lambda_k Tx^k$

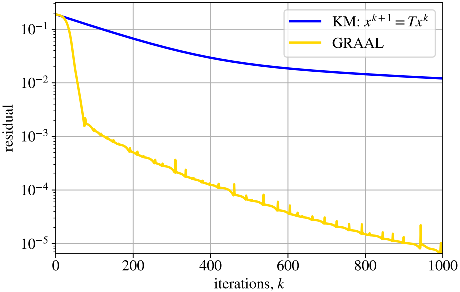
Tomography reconstruction

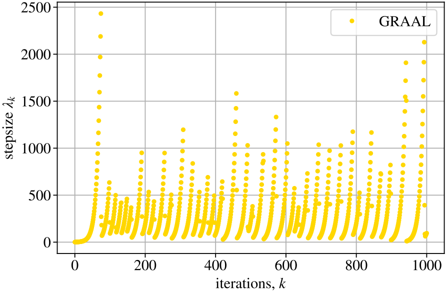
$$Ax = b,$$

$A \in \mathbb{R}^{m\times n}$ is the projection matrix, $b \in \mathbb{R}^m$ is the observed sinogram, $m = 2^{14}$, $n = 2^{16}$

CFP: find $x \in \bigcap_{i=1}^m C_i$, where $C_i = \{u\colon \langle a_i, u\rangle = b_i\}$.

$T = \frac{1}{m}(P_{C_1} + \cdots + P_{C_m})$ residual$(x) = \|x - Tx\|$.





GRAAL: $x^{k+1} = \frac{(\phi-1-\lambda_k\phi)x^k+\bar{x}^{k-1}}{\phi} + \lambda_k Tx^k$

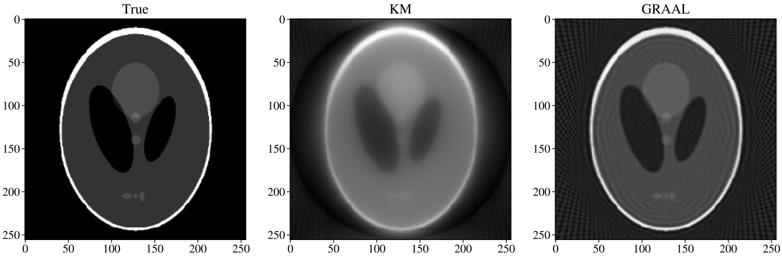
Tomography reconstruction

$$Ax = b,$$

$A \in \mathbb{R}^{m\times n}$ is the projection matrix, $b \in \mathbb{R}^m$ is the observed sinogram, $m = 2^{14}$, $n = 2^{16}$

CFP: find $x \in \bigcap_{i=1}^m C_i$, where $C_i = \{u \colon \langle a_i, u\rangle = b_i\}$.

$T = \frac{1}{m}(P_{C_1} + \dots + P_{C_m})$ residual$(x) = \|x - Tx\|$.



Conclusions

- VI are still cool

Conclusions

- VI are still cool
- New efficient algorithms are important

Conclusions

- VI are still cool
- New efficient algorithms are important
- Pay attention on the structure of the problem

Conclusions

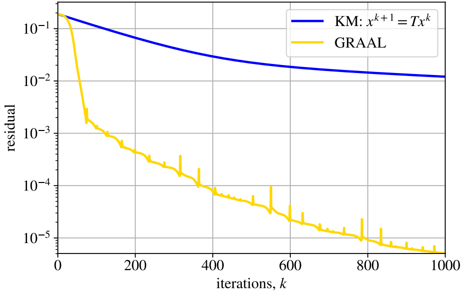
- VI are still cool
- New efficient algorithms are important
- Pay attention on the structure of the problem
- Step size matters

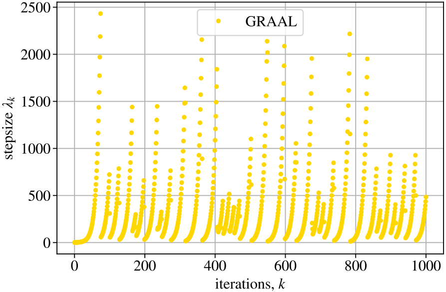
Thanks for attention!

$$Ax = b,$$

$A \in \mathbb{R}^{m \times n}$ is the projection matrix, $b \in \mathbb{R}^m$ is the observed noisy sinogram, noise $\sim \mathcal{N}(0, 0.1)$, $m = 2^{14}$, $n = 2^{16}$

CFP: find $x \in \bigcap_{i=1}^m C_i$, where $C_i = \{u \colon \langle a_i, u \rangle = b_i\}$.
$T = \frac{1}{m}(P_{C_1} + \dots + P_{C_m})$ residual$(x) = \|x - Tx\|$.





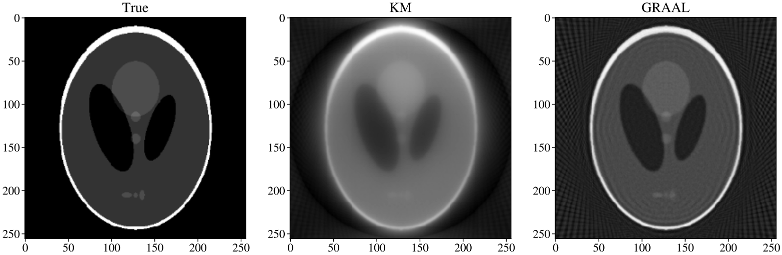
GRAAL: $x^{k+1} = \frac{(\phi - 1 - \lambda_k \phi) x^k + \bar{x}^{k-1}}{\phi} + \lambda_k T x^k$

Tomography reconstruction: noisy case

$$Ax = b,$$

$A \in \mathbb{R}^{m\times n}$ is the projection matrix, $b \in \mathbb{R}^m$ is the observed noisy sinogram, noise $\sim \mathcal{N}(0, 0.1)$, $m = 2^{14}$, $n = 2^{16}$

CFP: find $x \in \bigcap_{i=1}^m C_i$, where $C_i = \{u\colon \langle a_i, u\rangle = b_i\}$.
$T = \frac{1}{m}(P_{C_1} + \cdots + P_{C_m})$ residual$(x) = \|x - Tx\|$.



GRAAL: $x^{k+1} = \frac{(\phi - 1 - \lambda_k \phi)x^k + \bar{x}^{k-1}}{\phi} + \lambda_k T x^k$

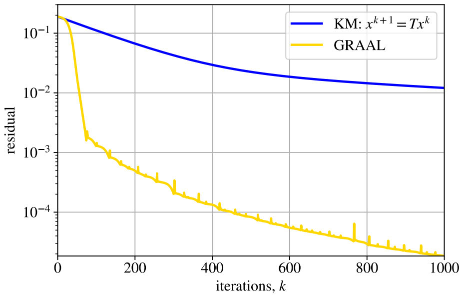
Tomography reconstruction: noisy case

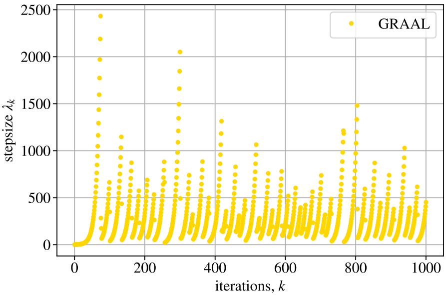
$$Ax = b,$$

$A \in \mathbb{R}^{m\times n}$ is the projection matrix, $b \in \mathbb{R}^m$ is the observed noisy sinogram, noise $\sim \mathcal{N}(0,1)$, $m = 2^{14}$, $n = 2^{16}$

CFP: find $x \in \bigcap_{i=1}^m C_i$, where $C_i = \{u \colon \langle a_i, u\rangle = b_i\}$.

$T = \frac{1}{m}(P_{C_1} + \dots + P_{C_m})$ residual$(x) = \|x - Tx\|$.





GRAAL: $x^{k+1} = \frac{(\phi - 1 - \lambda_k\phi)x^k + \bar{x}^{k-1}}{\phi} + \lambda_k Tx^k$

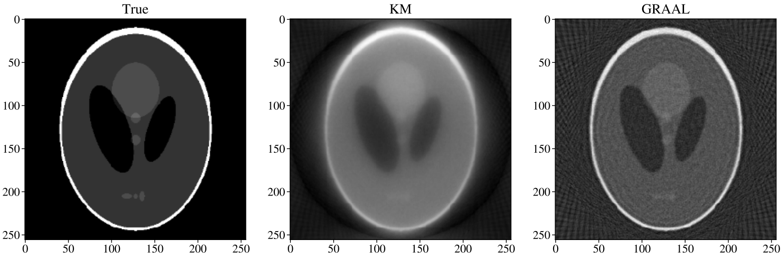
Tomography reconstruction: noisy case

$$Ax = b,$$

$A \in \mathbb{R}^{m\times n}$ is the projection matrix, $b \in \mathbb{R}^m$ is the observed noisy sinogram, noise $\sim \mathcal{N}(0,1)$, $m = 2^{14}$, $n = 2^{16}$

CFP: find $x \in \bigcap_{i=1}^m C_i$, where $C_i = \{u\colon \langle a_i, u\rangle = b_i\}$.

$T = \frac{1}{m}(P_{C_1} + \cdots + P_{C_m})$ residual$(x) = \|x - Tx\|$.



GRAAL: $x^{k+1} = \frac{(\phi - 1 - \lambda_k\phi)x^k + \bar{x}^{k-1}}{\phi} + \lambda_k Tx^k$