Performance Estimation Problems (PEPs):

Tutorial – Session 3/3

Aymeric Dieuleveut, Adrien Taylor





SMAI-MODE tutorial – March 2026

Designing algorithms

A "generic" first-order method

$$
\begin{aligned}
w_1 &= w_0 - \alpha_{1,0}\nabla f(w_0) \\
w_2 &= w_1 - \alpha_{2,0}\nabla f(w_0) - \alpha_{2,1}\nabla f(w_1) \\
w_3 &= w_2 - \alpha_{3,0}\nabla f(w_0) - \alpha_{3,1}\nabla f(w_1) - \alpha_{3,2}\nabla f(w_2) \\
&\vdots \\
w_n &= w_{n-1} - \sum_{i=0}^{n-1} \alpha_{n,i}\nabla f(w_i),
\end{aligned}
\tag{FOM}
$$

for some coefficients $\{\alpha_{i,j}\}$. Generic **non-adaptive** first-order method.

Designing algorithms

A "generic" first-order method

$$
\begin{aligned}
w_1 &= w_0 - \alpha_{1,0}\nabla f(w_0) \\
w_2 &= w_1 - \alpha_{2,0}\nabla f(w_0) - \alpha_{2,1}\nabla f(w_1) \\
w_3 &= w_2 - \alpha_{3,0}\nabla f(w_0) - \alpha_{3,1}\nabla f(w_1) - \alpha_{3,2}\nabla f(w_2) \\
&\vdots \\
w_n &= w_{n-1} - \sum_{i=0}^{n-1} \alpha_{n,i}\nabla f(w_i),
\end{aligned}
\tag{FOM}
$$

for some coefficients $\{\alpha_{i,j}\}$. Generic **non-adaptive** first-order method.

How to choose $\{\alpha_{i,j}\}$?

Designing algorithms

A "generic" first-order method

$$
\begin{aligned}
w_1 &= w_0 - \alpha_{1,0}\nabla f(w_0) \\
w_2 &= w_1 - \alpha_{2,0}\nabla f(w_0) - \alpha_{2,1}\nabla f(w_1) \\
w_3 &= w_2 - \alpha_{3,0}\nabla f(w_0) - \alpha_{3,1}\nabla f(w_1) - \alpha_{3,2}\nabla f(w_2) \\
&\vdots \\
w_n &= w_{n-1} - \sum_{i=0}^{n-1} \alpha_{n,i}\nabla f(w_i),
\end{aligned}
\tag{FOM}
$$

for some coefficients $\{\alpha_{i,j}\}$. Generic **non-adaptive** first-order method.

How to choose $\{\alpha_{i,j}\}$?

- ◇ pick a performance criterion, for instance $\frac{\|w_n - w_\star\|^2}{\|w_0 - w_\star\|^2}$,

Designing algorithms

A "generic" first-order method

$$
\begin{aligned}
w_1 &= w_0 - \alpha_{1,0}\nabla f(w_0) \\
w_2 &= w_1 - \alpha_{2,0}\nabla f(w_0) - \alpha_{2,1}\nabla f(w_1) \\
w_3 &= w_2 - \alpha_{3,0}\nabla f(w_0) - \alpha_{3,1}\nabla f(w_1) - \alpha_{3,2}\nabla f(w_2) \\
&\vdots \\
w_n &= w_{n-1} - \sum_{i=0}^{n-1} \alpha_{n,i}\nabla f(w_i),
\end{aligned}
\tag{FOM}
$$

for some coefficients $\{\alpha_{i,j}\}$. Generic **non-adaptive** first-order method.

How to choose $\{\alpha_{i,j}\}$?

- ◇ pick a performance criterion, for instance $\frac{\|w_n - w_\star\|^2}{\|w_0 - w_\star\|^2}$,
- ◇ solve the minimax (minimize worst-case): $\min_{\{\alpha_{i,j}\}_{i,j}} \max_{f\in\mathcal{F},\{w_i\}} \frac{\|w_n - w_\star\|^2}{\|w_0 - w_\star\|^2}$.

Traditional design

Algorithm design (with guarantees):

Traditional design

Algorithm design (with guarantees):

- ◇ unconstrained quadratic programming:

  first-order methods $\Leftrightarrow$ polynomials $\rightarrow$ optimal algorithms via optimal polynomials.[1,2,3]

[1] Golub and Varga (1961). "Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods." Numerische Mathematik.

[2] Polyak (1987). "Introduction to optimization." Optimization Software.

[3] Nemirovski (1995). "Information-based complexity of convex programming." (Lecture notes)

Traditional design

Algorithm design (with guarantees):

- ◇ unconstrained quadratic programming:

  first-order methods $\Leftrightarrow$ polynomials $\rightarrow$ optimal algorithms via optimal polynomials.[1,2,3]

- ◇ Beyond quadratics: traditionally more "hand-crafted"
  - – Analogy with conjugate gradients.[4,5]
  - – Lyapunov function-type analyses.[6,7]

---

[1] Golub and Varga (1961). "Chebyshev semi-iterative methods, successive overrelaxation iterative methods, and second order Richardson iterative methods." Numerische Mathematik.

[2] Polyak (1987). "Introduction to optimization." Optimization Software.

[3] Nemirovski (1995). "Information-based complexity of convex programming." (Lecture notes)

[4] Nemirovski (1982). "Orth-method for smooth convex optimization." (in Russian) Engineering Cybernetics.

[5] Narkiss and Zibulevsky (2005). "Sequential subspace optimization method for large-scale unconstrained problems."

[6] Nesterov (1983). "A method for solving the convex programming problem with convergence rate $O(1/k^2)$." Dokl. akad. nauk. Sssr 269.

[7] Beck and Teboulle (2009). "A fast iterative shrinkage-thresholding algorithm for linear inverse problems." SIAM journal on imaging sciences 2(1).

Design problem

How to solve the design problem (or proxy of it)?

$$\min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{\|w_n - w_\star\|^2}{\|w_0 - w_\star\|^2} ?$$

Design problem

How to solve the design problem (or proxy of it)?

$$\min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{\|w_n - w_\star\|^2}{\|w_0 - w_\star\|^2}? \qquad \min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{f(w_n) - f(w_\star)}{f(w_0) - f(w_\star)}?$$

Design problem

How to solve the design problem (or proxy of it)?

$$\min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{\|w_n - w_\star\|^2}{\|w_0 - w_\star\|^2}? \quad \min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{f(w_n) - f(w_\star)}{f(w_0) - f(w_\star)}? \quad \min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{\|\nabla f(w_n)\|^2}{\|\nabla f(w_0)\|^2}?$$

Design problem

How to solve the design problem (or proxy of it)?

$$\min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{\|w_n - w_\star\|^2}{\|w_0 - w_\star\|^2}? \quad \min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{f(w_n) - f(w_\star)}{f(w_0) - f(w_\star)}? \quad \min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{\|\nabla f(w_n)\|^2}{\|\nabla f(w_0)\|^2}?$$

- Convex relaxations,

Design problem

How to solve the design problem (or proxy of it)?

$$\min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{\|w_n - w_\star\|^2}{\|w_0 - w_\star\|^2}? \quad \min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{f(w_n) - f(w_\star)}{f(w_0) - f(w_\star)}? \quad \min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{\|\nabla f(w_n)\|^2}{\|\nabla f(w_0)\|^2}?$$

- ◇ Convex relaxations,
- ◇ analogies (e.g., with conjugate gradient methods)

$$w_{k+1} \in \operatorname*{argmin}_{x \in \mathbb{R}^d} \{f(w) : \ w \in w_0 + \operatorname{span}\{\nabla f(w_0), \ldots, \nabla f(w_k)\}\},$$

Design problem

How to solve the design problem (or proxy of it)?

$$\min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{\|w_n - w_\star\|^2}{\|w_0 - w_\star\|^2}? \quad \min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{f(w_n) - f(w_\star)}{f(w_0) - f(w_\star)}? \quad \min_{\{\alpha_{i,j}\}} \max_{f \in \mathcal{F}} \frac{\|\nabla f(w_n)\|^2}{\|\nabla f(w_0)\|^2}?$$

- ◇ Convex relaxations,
- ◇ analogies (e.g., with conjugate gradient methods)

$$w_{k+1} \in \operatorname*{argmin}_{x \in \mathbb{R}^d} \{f(w) : \ w \in w_0 + \operatorname{span}\{\nabla f(w_0), \ldots, \nabla f(w_k)\}\},$$

- ◇ brutal approaches.

Primal problem ($n = 1$)

Primal problem ($n = 1$)

Recall primal problem, with step-size optimization

$$
\begin{aligned}
\min_{\alpha_{1,0}} \max_{G,\, F} \quad & G_{1,1} + \alpha_{1,0}^2 G_{2,2} - 2\alpha_{1,0} G_{1,2} \\
\text{subject to} \quad & F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{L}{L-\mu} G_{1,2} \leqslant 0 \\
& -F + \tfrac{L\mu}{2(L-\mu)} G_{1,1} + \tfrac{1}{2(L-\mu)} G_{2,2} - \tfrac{\mu}{L-\mu} G_{1,2} \leqslant 0 \\
& G_{1,1} = 1 \\
& G \succcurlyeq 0.
\end{aligned}
$$

Primal problem ($n = 1$)

Recall primal problem, with step-size optimization

$$
\begin{array}{ll}
\min\limits_{\alpha_{1,0}} \max\limits_{G,\, F} & G_{1,1} + \alpha_{1,0}^2 G_{2,2} - 2\alpha_{1,0} G_{1,2} \\
\text{subject to} & F + \frac{L\mu}{2(L-\mu)} G_{1,1} + \frac{1}{2(L-\mu)} G_{2,2} - \frac{L}{L-\mu} G_{1,2} \leqslant 0 \\
& -F + \frac{L\mu}{2(L-\mu)} G_{1,1} + \frac{1}{2(L-\mu)} G_{2,2} - \frac{\mu}{L-\mu} G_{1,2} \leqslant 0 \\
& G_{1,1} = 1 \\
& G \succcurlyeq 0.
\end{array}
$$

"Simple" minimization problem by dualizing inner maximization.

Primal problem ($n = 1$)

Recall primal problem, with step-size optimization

$$
\begin{array}{ll}
\min\limits_{\alpha_{1,0}} \max\limits_{G,\, F} & G_{1,1} + \alpha_{1,0}^2 G_{2,2} - 2\alpha_{1,0} G_{1,2} \\
\text{subject to} & F + \frac{L\mu}{2(L-\mu)} G_{1,1} + \frac{1}{2(L-\mu)} G_{2,2} - \frac{L}{L-\mu} G_{1,2} \leqslant 0 \\
& -F + \frac{L\mu}{2(L-\mu)} G_{1,1} + \frac{1}{2(L-\mu)} G_{2,2} - \frac{\mu}{L-\mu} G_{1,2} \leqslant 0 \\
& G_{1,1} = 1 \\
& G \succcurlyeq 0.
\end{array}
$$

"Simple" minimization problem by dualizing inner maximization.

Dualize inner maximization $\rightarrow$ min min.

Optimizing the step-sizes ($n = 1$)

Optimizing the step-sizes ($n = 1$)

For $N = 1$, optimizing over step-size $\alpha_{1,0}$ remains convex!

Optimizing the step-sizes ($n = 1$)

For $N = 1$, optimizing over step-size $\alpha_{1,0}$ remains convex!

$$\min_{\tau,\lambda\geqslant 0} \quad \tau$$

$$\text{subject to} \begin{bmatrix} \tau - 1 + \frac{\lambda L\mu}{L-\mu} & \alpha_{1,0} - \frac{\lambda(\mu+L)}{2(L-\mu)} \\ \alpha_{1,0} - \frac{\lambda(\mu+L)}{2(L-\mu)} & \frac{\lambda}{L-\mu} - \alpha_{1,0}^2 \end{bmatrix} \succcurlyeq 0.$$

Optimizing the step-sizes ($n = 1$)

For $N = 1$, optimizing over step-size $\alpha_{1,0}$ remains convex!

$$\min_{\tau,\lambda\geqslant 0,\alpha_{1,0}} \quad \tau$$

$$\text{subject to} \quad \begin{bmatrix} \tau - 1 + \frac{\lambda L\mu}{L-\mu} & \alpha_{1,0} - \frac{\lambda(\mu+L)}{2(L-\mu)} \\ \alpha_{1,0} - \frac{\lambda(\mu+L)}{2(L-\mu)} & \frac{\lambda}{L-\mu} - \alpha_{1,0}^2 \end{bmatrix} \succcurlyeq 0.$$

Optimizing the step-sizes ($n = 1$)

For $N = 1$, optimizing over step-size $\alpha_{1,0}$ remains convex!

$$\min_{\tau,\lambda\geqslant 0,\alpha_{1,0}} \quad \tau$$
$$\text{subject to} \quad \begin{bmatrix} \tau - 1 + \frac{\lambda L\mu}{L-\mu} & \alpha_{1,0} - \frac{\lambda(\mu+L)}{2(L-\mu)} \\ \alpha_{1,0} - \frac{\lambda(\mu+L)}{2(L-\mu)} & \frac{\lambda}{L-\mu} - \alpha_{1,0}^2 \end{bmatrix} \succcurlyeq 0.$$

Optimize $\alpha_{1,0}$ "for free" (linear SDP via Schur complement):

Optimizing the step-sizes ($n = 1$)

For $N = 1$, optimizing over step-size $\alpha_{1,0}$ remains convex!

$$\min_{\tau,\lambda\geqslant 0,\alpha_{1,0}} \tau$$

$$\text{subject to } \begin{bmatrix} \tau - 1 + \frac{\lambda L\mu}{L-\mu} & \alpha_{1,0} - \frac{\lambda(\mu+L)}{2(L-\mu)} \\ \alpha_{1,0} - \frac{\lambda(\mu+L)}{2(L-\mu)} & \frac{\lambda}{L-\mu} - \alpha_{1,0}^2 \end{bmatrix} \succcurlyeq 0.$$

Optimize $\alpha_{1,0}$ "for free" (linear SDP via Schur complement):

Symmetric $M = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$

if $C \succ 0$: $M \succcurlyeq 0 \Leftrightarrow A - BC^{-1}B^T \succcurlyeq 0$

Optimizing the step-sizes ($n = 1$)

For $N = 1$, optimizing over step-size $\alpha_{1,0}$ remains convex!

$$\min_{\tau,\lambda\geqslant 0,\alpha_{1,0}} \tau$$
$$\text{subject to } \begin{bmatrix} \tau - 1 + \frac{\lambda L\mu}{L-\mu} & \alpha_{1,0} - \frac{\lambda(\mu+L)}{2(L-\mu)} \\ \alpha_{1,0} - \frac{\lambda(\mu+L)}{2(L-\mu)} & \frac{\lambda}{L-\mu} - \alpha_{1,0}^2 \end{bmatrix} \succcurlyeq 0.$$

Optimize $\alpha_{1,0}$ "for free" (linear SDP via Schur complement):

Symmetric $M = \begin{pmatrix} A & B \\ B^T & C \end{pmatrix}$
if $C \succ 0$: $M \succcurlyeq 0 \Leftrightarrow A - BC^{-1}B^T \succcurlyeq 0$

$$\min_{\tau,\lambda\geqslant 0,\alpha_{1,0}} \tau$$
$$\text{subject to } \begin{bmatrix} \tau - 1 + \frac{\lambda L\mu}{L-\mu} & -\frac{\lambda(\mu+L)}{2(L-\mu)} & 1 \\ -\frac{\lambda(\mu+L)}{2(L-\mu)} & \frac{\lambda}{L-\mu} & -\alpha_{1,0} \\ 1 & -\alpha_{1,0} & 1 \end{bmatrix} \succcurlyeq 0.$$

Optimizing the step-sizes ($n = 2$)

When $N = 2$, the problem becomes

$$\min_{\substack{\tau,\lambda_1,\ldots,\lambda_6 \geqslant 0 \\ \{\alpha_{i,j}\}}} \tau$$

subject to

Optimizing the step-sizes ($n = 2$)

When $N = 2$, the problem becomes

$$\min_{\substack{\tau,\lambda_1,\ldots,\lambda_6 \geqslant 0 \\ \{\alpha_{i,j}\}}} \tau$$

subject to

$$\begin{bmatrix} \lambda_1 + \lambda_2 - \lambda_3 - \lambda_5 \\ -\lambda_1 + \lambda_3 + \lambda_4 - \lambda_6 \end{bmatrix} = 0,$$

Optimizing the step-sizes ($n = 2$)

When $N = 2$, the problem becomes

$$\min_{\substack{\tau,\lambda_1,\ldots,\lambda_6 \geqslant 0 \\ \{\alpha_{i,j}\}}} \tau$$

$$\text{subject to} \begin{bmatrix} S_{1,1} & S_{1,2} & S_{1,3} \\ S_{1,2} & S_{2,2} & S_{2,3} \\ S_{1,3} & S_{2,3} & S_{3,3} \end{bmatrix} \succcurlyeq 0$$

$$\begin{bmatrix} \lambda_1 + \lambda_2 - \lambda_3 - \lambda_5 \\ -\lambda_1 + \lambda_3 + \lambda_4 - \lambda_6 \end{bmatrix} = 0,$$

Optimizing the step-sizes ($n = 2$)

When $N = 2$, the problem becomes

$$\min_{\substack{\tau,\lambda_1,\ldots,\lambda_6 \geqslant 0 \\ \{\alpha_{i,j}\}}} \tau$$

$$\text{subject to} \begin{bmatrix} S_{1,1} & S_{1,2} & S_{1,3} \\ S_{1,2} & S_{2,2} & S_{2,3} \\ S_{1,3} & S_{2,3} & S_{3,3} \end{bmatrix} \succcurlyeq 0$$

$$\begin{bmatrix} \lambda_1 + \lambda_2 - \lambda_3 - \lambda_5 \\ -\lambda_1 + \lambda_3 + \lambda_4 - \lambda_6 \end{bmatrix} = 0,$$

for some $S_{1,1}, S_{1,2}, \ldots, S_{3,3}$ (functions of $\tau, \lambda_1, \ldots, \lambda_6$ and $\{\alpha_{i,j}\}$).

Optimizing the step-sizes ($n = 2$)

When $N = 2$, the problem becomes

$$\min_{\substack{\tau,\lambda_1,\dots,\lambda_6\geqslant 0\\ \{\alpha_{i,j}\}}} \tau$$
$$\text{subject to } \begin{bmatrix} S_{1,1} & S_{1,2} & S_{1,3} \\ S_{1,2} & S_{2,2} & S_{2,3} \\ S_{1,3} & S_{2,3} & S_{3,3} \end{bmatrix} \succcurlyeq 0$$
$$\begin{bmatrix} \lambda_1+\lambda_2-\lambda_3-\lambda_5 \\ -\lambda_1+\lambda_3+\lambda_4-\lambda_6 \end{bmatrix} = 0,$$

for some $S_{1,1}, S_{1,2}, \dots, S_{3,3}$ (functions of $\tau, \lambda_1, \dots, \lambda_6$ and $\{\alpha_{i,j}\}$).

In particular

$$S_{1,2} = -\frac{L\lambda_3 - 2(L-\mu)\alpha_{2,0} + \mu\lambda_1 + L\mu(\lambda_2+\lambda_5)\alpha_{1,0}}{L-\mu}$$
$$S_{2,2} = \frac{-2(\mu\lambda_6 + L\lambda_4)\alpha_{1,0} - 2(L-\mu)\alpha_{2,0}^2 + L\mu(\lambda_2+\lambda_4+\lambda_5+\lambda_6)\alpha_{1,0}^2 + \lambda_1+\lambda_3+\lambda_4+\lambda_6}{L-\mu}$$

Optimizing the step-sizes ($n = 2$)

When $N = 2$, the problem becomes

$$\min_{\substack{\tau,\lambda_1,\ldots,\lambda_6\geqslant 0\\ \{\alpha_{i,j}\}}} \tau$$
$$\text{subject to } \begin{bmatrix} S_{1,1} & S_{1,2} & S_{1,3} \\ S_{1,2} & S_{2,2} & S_{2,3} \\ S_{1,3} & S_{2,3} & S_{3,3} \end{bmatrix} \succcurlyeq 0$$
$$\begin{bmatrix} \lambda_1+\lambda_2-\lambda_3-\lambda_5 \\ -\lambda_1+\lambda_3+\lambda_4-\lambda_6 \end{bmatrix} = 0,$$

for some $S_{1,1}, S_{1,2}, \ldots, S_{3,3}$ (functions of $\tau, \lambda_1, \ldots, \lambda_6$ and $\{\alpha_{i,j}\}$).

In particular

$$S_{1,2} = -\frac{L\lambda_3 - 2(L-\mu)\alpha_{2,0} + \mu\lambda_1 + L\mu(\lambda_2+\lambda_5)\alpha_{1,0}}{L-\mu}$$
$$S_{2,2} = \frac{-2(\mu\lambda_6 + L\lambda_4)\alpha_{1,0} - 2(L-\mu)\alpha_{2,0}^2 + L\mu(\lambda_2+\lambda_4+\lambda_5+\lambda_6)\alpha_{1,0}^2 + \lambda_1+\lambda_3+\lambda_4+\lambda_6}{L-\mu}$$

LMI convex in some step-sizes ($\alpha_{2,0}$ and $\alpha_{2,1}$) but not in the others.

Forewords and problem statement

Numerical examples

Design via idealized algorithms

Concluding remarks

Numerical examples I

Example for $L = 1$ and $\mu = .1$

Numerical examples I

Example for $L = 1$ and $\mu = .1$

- For $n = 1$, we reach $\frac{\|w_1 - w_\star\|^2}{\|w_0 - w_\star\|^2} \leqslant 0.6694$ with step-sizes

$$[\alpha_{i,j}^\star] = \begin{bmatrix} 1.8182 \end{bmatrix}.$$

Numerical examples I

Example for $L = 1$ and $\mu = .1$

- For $n = 1$, we reach $\frac{\|w_1 - w_\star\|^2}{\|w_0 - w_\star\|^2} \leqslant 0.6694$ with step-sizes

$$[\alpha_{i,j}^\star] = \begin{bmatrix} 1.8182 \end{bmatrix}.$$

- For $n = 2$, we reach $\frac{\|w_2 - w_\star\|^2}{\|w_0 - w_\star\|^2} \leqslant 0.3769$ with

$$[\alpha_{i,j}^\star] = \begin{bmatrix} 1.5466 & \\ 0.2038 & 2.4961 \end{bmatrix}.$$

Numerical examples I

Example for $L = 1$ and $\mu = .1$

- For $n = 1$, we reach $\frac{\|w_1 - w_\star\|^2}{\|w_0 - w_\star\|^2} \leqslant 0.6694$ with step-sizes

$$[\alpha_{i,j}^\star] = \begin{bmatrix} 1.8182 \end{bmatrix}.$$

- For $n = 2$, we reach $\frac{\|w_2 - w_\star\|^2}{\|w_0 - w_\star\|^2} \leqslant 0.3769$ with

$$[\alpha_{i,j}^\star] = \begin{bmatrix} 1.5466 & \\ 0.2038 & 2.4961 \end{bmatrix}.$$

- For $n = 3$, we reach $\frac{\|w_3 - w_\star\|^2}{\|w_0 - w_\star\|^2} \leqslant 0.1932$ with

$$[\alpha_{i,j}^\star] = \begin{bmatrix} 1.5466 & & \\ 0.1142 & 1.8380 & \\ 0.0642 & 0.4712 & 2.8404 \end{bmatrix}.$$

Numerical examples I

Example for $L = 1$ and $\mu = .1$

- For $n = 1$, we reach $\frac{\|w_1 - w_\star\|^2}{\|w_0 - w_\star\|^2} \leqslant 0.6694$ with step-sizes

$$[\alpha_{i,j}^\star] = \begin{bmatrix} 1.8182 \end{bmatrix}.$$

- For $n = 2$, we reach $\frac{\|w_2 - w_\star\|^2}{\|w_0 - w_\star\|^2} \leqslant 0.3769$ with

$$[\alpha_{i,j}^\star] = \begin{bmatrix} 1.5466 & \\ 0.2038 & 2.4961 \end{bmatrix}.$$

- For $n = 3$, we reach $\frac{\|w_3 - w_\star\|^2}{\|w_0 - w_\star\|^2} \leqslant 0.1932$ with

$$[\alpha_{i,j}^\star] = \begin{bmatrix} 1.5466 & & \\ 0.1142 & 1.8380 & \\ 0.0642 & 0.4712 & 2.8404 \end{bmatrix}.$$

- For $n = 4$, we reach $\frac{\|w_4 - w_\star\|^2}{\|w_0 - w_\star\|^2} \leqslant 0.0944$ with

$$[\alpha_{i,j}^\star] = \begin{bmatrix} 1.5466 & & & \\ 0.1142 & 1.8380 & & \\ 0.0331 & 0.2432 & 1.9501 & \\ 0.0217 & 0.1593 & 0.6224 & 3.0093 \end{bmatrix}.$$

Numerical examples II

What about different performance measure? Example $\frac{f(w_n)-f_\star}{f(w_0)-f_\star}$ and $L = 1$, $\mu = .1$.

Numerical examples II

What about different performance measure? Example $\frac{f(w_n)-f_\star}{f(w_0)-f_\star}$ and $L = 1$, $\mu = .1$.

- For $n = 1$, we obtain $\frac{f(w_1)-f_\star}{f(w_0)-f_\star} \leqslant 0.6694$ with step-size

$$[\alpha_{i,j}] = \begin{bmatrix} 1.8182 \end{bmatrix}.$$

Numerical examples II

What about different performance measure? Example $\frac{f(w_n)-f_\star}{f(w_0)-f_\star}$ and $L = 1$, $\mu = .1$.

- For $n = 1$, we obtain $\frac{f(w_1)-f_\star}{f(w_0)-f_\star} \leqslant 0.6694$ with step-size

$$[\alpha_{i,j}] = \begin{bmatrix} 1.8182 \end{bmatrix}.$$

- For $n = 2$, we obtain $\frac{f(w_2)-f_\star}{f(w_0)-f_\star} \leqslant 0.3554$ with

$$[\alpha_{i,j}] = \begin{bmatrix} 2.0095 & \\ 0.4229 & 2.0095 \end{bmatrix}.$$

Numerical examples II

What about different performance measure? Example $\frac{f(w_n)-f_\star}{f(w_0)-f_\star}$ and $L = 1$, $\mu = .1$.

- For $n = 1$, we obtain $\frac{f(w_1)-f_\star}{f(w_0)-f_\star} \leqslant 0.6694$ with step-size

$$[\alpha_{i,j}] = \begin{bmatrix} 1.8182 \end{bmatrix}.$$

- For $n = 2$, we obtain $\frac{f(w_2)-f_\star}{f(w_0)-f_\star} \leqslant 0.3554$ with

$$[\alpha_{i,j}] = \begin{bmatrix} 2.0095 & \\ 0.4229 & 2.0095 \end{bmatrix}.$$

- For $n = 3$, we obtain $\frac{f(w_3)-f_\star}{f(w_0)-f_\star} \leqslant 0.1698$ with

$$[\alpha_{i,j}] = \begin{bmatrix} 1.9470 & & \\ 0.4599 & 2.2406 & \\ 0.1705 & 0.4599 & 1.9470 \end{bmatrix}.$$

Numerical examples II

What about different performance measure? Example $\frac{f(w_n)-f_\star}{f(w_0)-f_\star}$ and $L = 1$, $\mu = .1$.

- For $n = 1$, we obtain $\frac{f(w_1)-f_\star}{f(w_0)-f_\star} \leqslant 0.6694$ with step-size

$$[\alpha_{i,j}] = \begin{bmatrix} 1.8182 \end{bmatrix}.$$

- For $n = 2$, we obtain $\frac{f(w_2)-f_\star}{f(w_0)-f_\star} \leqslant 0.3554$ with

$$[\alpha_{i,j}] = \begin{bmatrix} 2.0095 & \\ 0.4229 & 2.0095 \end{bmatrix}.$$

- For $n = 3$, we obtain $\frac{f(w_3)-f_\star}{f(w_0)-f_\star} \leqslant 0.1698$ with

$$[\alpha_{i,j}] = \begin{bmatrix} 1.9470 & & \\ 0.4599 & 2.2406 & \\ 0.1705 & 0.4599 & 1.9470 \end{bmatrix}.$$

- For $n = 4$, we obtain $\frac{f(w_4)-f_\star}{f(w_0)-f_\star} \leqslant 0.0789$ with

$$[\alpha_{i,j}] = \begin{bmatrix} 1.9187 & & & \\ 0.4098 & 2.1746 & & \\ 0.1796 & 0.5147 & 2.1746 & \\ 0.0627 & 0.1796 & 0.4098 & 1.9187 \end{bmatrix}.$$

Numerical example III

Worst-case performance $\frac{f(w_n)-f_\star}{\|w_0-w_\star\|^2}$ with $L = 1$ and $\mu = .01$. We compare

Numerical example III

Worst-case performance $\frac{f(w_n)-f_\star}{\|w_0-w_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

- worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,

Numerical example III

Worst-case performance $\frac{f(w_n)-f_\star}{\|w_0-w_\star\|^2}$ with $L = 1$ and $\mu = .01$. We compare

- ◇ worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,
- ◇ worst-case performance of optimized method, conjugate gradient-based method (both numerically generated),

Numerical example III

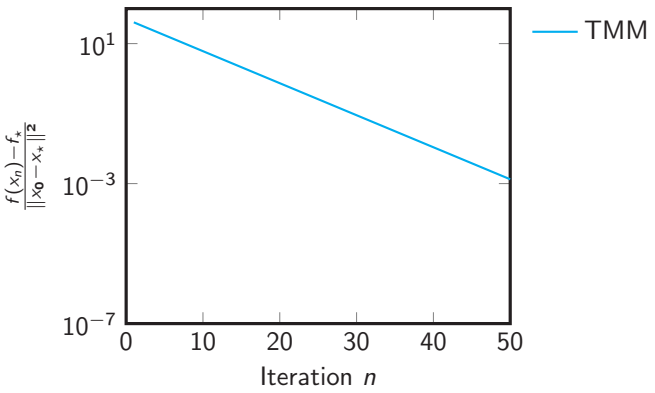
Worst-case performance $\frac{f(w_n)-f_\star}{\|w_0-w_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

- ◇ worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,
- ◇ worst-case performance of optimized method, conjugate gradient-based method (both numerically generated),
- ◇ Lower complexity bound (numerically generated).

Numerical example III

Worst-case performance $\frac{f(w_n)-f_\star}{\|w_0-w_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

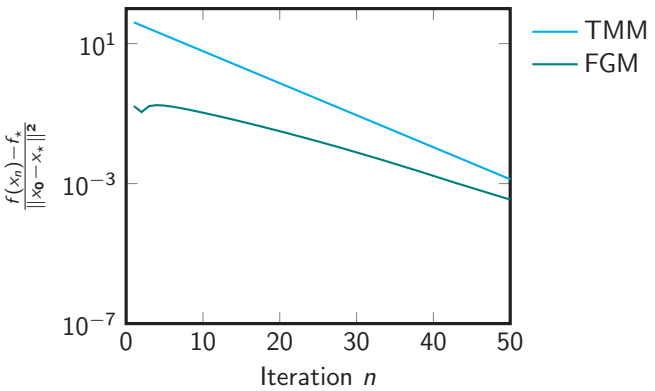
- ◇ worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,
- ◇ worst-case performance of optimized method, conjugate gradient-based method (both numerically generated),
- ◇ Lower complexity bound (numerically generated).



Numerical example III

Worst-case performance $\frac{f(w_n)-f_\star}{\|w_0-w_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

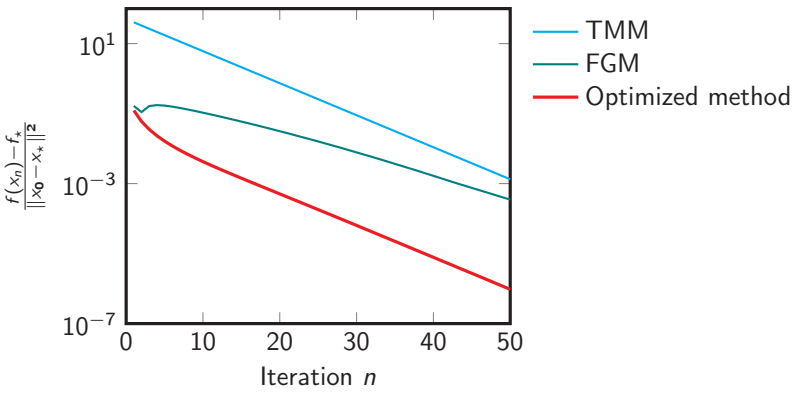
- worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,
- worst-case performance of optimized method, conjugate gradient-based method (both numerically generated),
- Lower complexity bound (numerically generated).



Numerical example III

Worst-case performance $\frac{f(w_n)-f_\star}{\|w_0-w_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

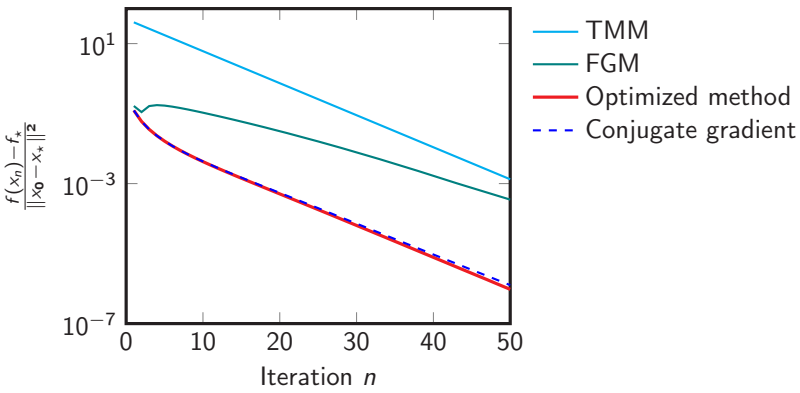
- ◇ worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,
- ◇ worst-case performance of optimized method, conjugate gradient-based method (both numerically generated),
- ◇ Lower complexity bound (numerically generated).



Numerical example III

Worst-case performance $\frac{f(w_n)-f_\star}{\|w_0-w_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

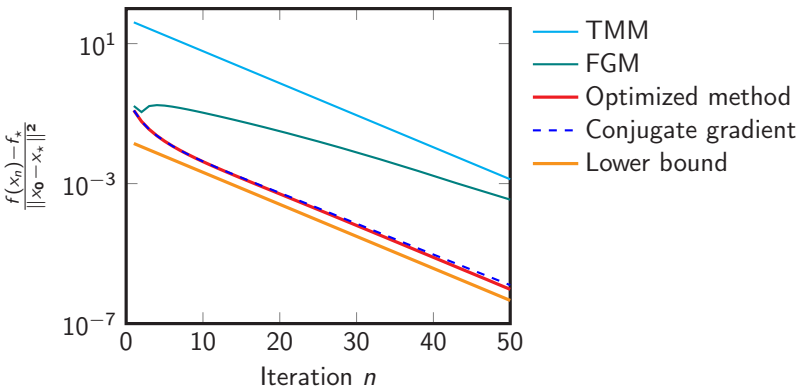
- ◇ worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,
- ◇ worst-case performance of optimized method, conjugate gradient-based method (both numerically generated),
- ◇ Lower complexity bound (numerically generated).



Numerical example III

Worst-case performance $\frac{f(w_n)-f_\star}{\|w_0-w_\star\|^2}$ with $L=1$ and $\mu=.01$. We compare

- worst-case performance of known methods, namely Triple Momentum Method (TMM) and Accelerated/Fast Gradient Method (FGM) computed using PEPs,
- worst-case performance of optimized method, conjugate gradient-based method (both numerically generated),
- Lower complexity bound (numerically generated).



Minimax design beyond quadratics

From my humble current understanding:

- ◇ approaches to minimax (beyond quadratics) are less direct,

Minimax design beyond quadratics

From my humble current understanding:

- ◇ approaches to minimax (beyond quadratics) are less direct,
- ◇ traditionally follows a two-stage procedure:

Minimax design beyond quadratics

From my humble current understanding:

- ◇ approaches to minimax (beyond quadratics) are less direct,
- ◇ traditionally follows a two-stage procedure:
  - (i) algorithm-dependent upper bound, and

Minimax design beyond quadratics

From my humble current understanding:

- ◇ approaches to minimax (beyond quadratics) are less direct,
- ◇ traditionally follows a two-stage procedure:
  - (i) algorithm-dependent upper bound, and
  - (ii) algorithm-independent lower bound.

Minimax design beyond quadratics

From my humble current understanding:

- ◇ approaches to minimax (beyond quadratics) are less direct,
- ◇ traditionally follows a two-stage procedure:
  - (i) algorithm-dependent upper bound, and
  - (ii) algorithm-independent lower bound.
- ◇ Similar approach with PEPs, helped by computers and SDPs.

Minimax design beyond quadratics

From my humble current understanding:

- ◇ approaches to minimax (beyond quadratics) are less direct,
- ◇ traditionally follows a two-stage procedure:
  - (i) algorithm-dependent upper bound, and
  - (ii) algorithm-independent lower bound.
- ◇ Similar approach with PEPs, helped by computers and SDPs.
- ◇ Optimal methods are known for the two notions

Minimax design beyond quadratics

From my humble current understanding:

- ◇ approaches to minimax (beyond quadratics) are less direct,
- ◇ traditionally follows a two-stage procedure:
  - (i) algorithm-dependent upper bound, and
  - (ii) algorithm-independent lower bound.
- ◇ Similar approach with PEPs, helped by computers and SDPs.
- ◇ Optimal methods are known for the two notions
  - $\frac{f(w_n)-f_\star}{\|w_0-w_\star\|^2}$ (for L-smooth convex functions),[9]

[9] Kim, Fessler (2016). "Optimized first-order methods for smooth convex minimization."

Minimax design beyond quadratics

From my humble current understanding:

- ◇ approaches to minimax (beyond quadratics) are less direct,
- ◇ traditionally follows a two-stage procedure:
  - (i) algorithm-dependent upper bound, and
  - (ii) algorithm-independent lower bound.
- ◇ Similar approach with PEPs, helped by computers and SDPs.
- ◇ Optimal methods are known for the two notions
  - $\frac{f(w_n)-f_\star}{\|w_0-w_\star\|^2}$ (for L-smooth convex functions),[9]
  - $\frac{\|w_n-w_\star\|^2}{\|w_0-w_\star\|^2}$ (for L-smooth μ-strongly convex functions),[10]

[9] Kim, Fessler (2016). "Optimized first-order methods for smooth convex minimization."

[10] T, Drori (2023). "An optimal gradient method for smooth strongly convex minimization."

Minimax design beyond quadratics

From my humble current understanding:

- ◇ approaches to minimax (beyond quadratics) are less direct,
- ◇ traditionally follows a two-stage procedure:
  - (i) algorithm-dependent upper bound, and
  - (ii) algorithm-independent lower bound.
- ◇ Similar approach with PEPs, helped by computers and SDPs.
- ◇ Optimal methods are known for the two notions$\rightarrow$ incredibly close to Nesterov's method.[8]
  - $\frac{f(w_n)-f_\star}{\|w_0-w_\star\|^2}$ (for L-smooth convex functions),[9]
  - $\frac{\|w_n-w_\star\|^2}{\|w_0-w_\star\|^2}$ (for L-smooth μ-strongly convex functions),[10]

[8] Nesterov (2004). "Introductory lectures on convex optimization: A basic course.".

[9] Kim, Fessler (2016). "Optimized first-order methods for smooth convex minimization."

[10] T, Drori (2023). "An optimal gradient method for smooth strongly convex minimization."

Minimax design beyond quadratics

From my humble current understanding:

- ◇ approaches to minimax (beyond quadratics) are less direct,
- ◇ traditionally follows a two-stage procedure:
  - (i) algorithm-dependent upper bound, and
  - (ii) algorithm-independent lower bound.
- ◇ Similar approach with PEPs, helped by computers and SDPs.
- ◇ Optimal methods are known for the two notions$\rightarrow$ incredibly close to Nesterov's method.[8]
  - $\frac{f(w_n)-f_\star}{\|w_0-w_\star\|^2}$ (for L-smooth convex functions),[9]
  - $\frac{\|w_n-w_\star\|^2}{\|w_0-w_\star\|^2}$ (for L-smooth μ-strongly convex functions),[10]
- ◇ Beyond that, a few criterion/settings/methods for which "perfectly optimal" algorithms might be known, but matching lower bounds are still missing.
  - $\frac{\|\nabla f(w_n)\|^2}{f(w_0)-f_\star}$,[11]
  - a few numerically-generated methods, proximal variants, etc.

[8] Nesterov (2004). "Introductory lectures on convex optimization: A basic course.".

[9] Kim, Fessler (2016). "Optimized first-order methods for smooth convex minimization."

[10] T, Drori (2023). "An optimal gradient method for smooth strongly convex minimization."

[11] Kim, Fessler (2021). "Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions."

Optimized gradient method (OGM)

Example I: the optimal method for $\frac{f(y_n)-f_\star}{\|y_0-y_\star\|^2}$ is called the "optimized gradient method".[12]

[12] Kim, Fessler (2016). "Optimized methods for smooth convex optimization."

Optimized gradient method (OGM)

Example I: the optimal method for $\frac{f(y_n)-f_\star}{\|y_0-y_\star\|^2}$ is called the "optimized gradient method".[12]

$$y_k = \tfrac{1}{\theta_k} z_k + \left(1 - \tfrac{1}{\theta_k}\right)\left(y_{k-1} - \tfrac{1}{L}\nabla f(y_{k-1})\right)$$
$$z_{k+1} = z_k - \tfrac{2\theta_k}{L}\nabla f(y_k)$$

[12] Kim, Fessler (2016). "Optimized methods for smooth convex optimization."

Optimized gradient method (OGM)

Example I: the optimal method for $\frac{f(y_n)-f_\star}{\|y_0-y_\star\|^2}$ is called the "optimized gradient method".[12]

$$y_k = \tfrac{1}{\theta_k} z_k + \left(1 - \tfrac{1}{\theta_k}\right)\left(y_{k-1} - \tfrac{1}{L}\nabla f(y_{k-1})\right)$$
$$z_{k+1} = z_k - \tfrac{2\theta_k}{L}\nabla f(y_k)$$

which rely on some exotic sequence

$$\theta_{k+1} = \begin{cases} \frac{1+\sqrt{4\theta_k^2+1}}{2} & \text{if } k \leqslant n-2 \\ \frac{1+\sqrt{8\theta_k^2+1}}{2} & \text{if } k = n-1, \end{cases}$$

[12] Kim, Fessler (2016). "Optimized methods for smooth convex optimization."

Optimized gradient method (OGM)

Example I: the optimal method for $\frac{f(y_n)-f_\star}{\|y_0-y_\star\|^2}$ is called the "optimized gradient method".[12]

$$y_k = \tfrac{1}{\theta_k} z_k + \left(1 - \tfrac{1}{\theta_k}\right)\left(y_{k-1} - \tfrac{1}{L}\nabla f(y_{k-1})\right)$$
$$z_{k+1} = z_k - \tfrac{2\theta_k}{L}\nabla f(y_k)$$

which rely on some exotic sequence

$$\theta_{k+1} = \begin{cases} \frac{1+\sqrt{4\theta_k^2+1}}{2} & \text{if } k \leqslant n-2 \\ \frac{1+\sqrt{8\theta_k^2+1}}{2} & \text{if } k = n-1, \end{cases}$$

where $\theta_{-1} = 0$, and roughly $\theta_k \approx \frac{k}{2}$.

[12] Kim, Fessler (2016). "Optimized methods for smooth convex optimization."

Optimized gradient method (OGM)

Example I: the optimal method for $\frac{f(y_n)-f_\star}{\|y_0-y_\star\|^2}$ is called the "optimized gradient method":[12]

$$y_k = \tfrac{1}{\theta_k} z_k + \left(1 - \tfrac{1}{\theta_k}\right)\left(y_{k-1} - \tfrac{1}{L}\nabla f(y_{k-1})\right)$$
$$z_{k+1} = z_k - \tfrac{2\theta_k}{L}\nabla f(y_k)$$

which rely on some exotic sequence

$$\theta_{k+1} = \begin{cases} \frac{1+\sqrt{4\theta_k^2+1}}{2} & \text{if } k \leqslant n-2 \\ \frac{1+\sqrt{8\theta_k^2+1}}{2} & \text{if } k = n-1, \end{cases}$$

where $\theta_{-1} = 0$, and roughly $\theta_k \approx \frac{k}{2}$.

The (tight) worst-case guarantee: $\frac{f(y_n)-f_\star}{L\|y_0-y_\star\|^2} \leqslant \frac{1}{2\theta_n^2} \approx \frac{2}{n^2}$. Matches exactly lower bound.[13]

[12] Kim, Fessler (2016). "Optimized methods for smooth convex optimization."

[13] Drori (2017). "The exact information-based complexity of smooth convex minimization."

Information-Theoretic Exact Method (ITEM)

Example II: optimal method for $\frac{\|z_n - z_\star\|^2}{\|z_0 - z_\star\|^2}$ is "Information-Theoretic Exact Method":[14]

[14] T, Drori (2023). "An optimal gradient method for smooth strongly convex minimization."

Information-Theoretic Exact Method (ITEM)

Example II: optimal method for $\frac{\|z_n - z_\star\|^2}{\|z_0 - z_\star\|^2}$ is "Information-Theoretic Exact Method":[14]

$$y_k = (1-\beta_k)z_k + \beta_k\left(y_{k-1} - \frac{1}{L}\nabla f(y_{k-1})\right)$$
$$z_{k+1} = (1-\tfrac{\mu}{L}\delta_k)z_k + \tfrac{\mu}{L}\delta_k\left(y_k - \frac{1}{\mu}\nabla f(y_k)\right),$$

[14] T, Drori (2023). "An optimal gradient method for smooth strongly convex minimization."

Information-Theoretic Exact Method (ITEM)

Example II: optimal method for $\frac{\|z_n - z_\star\|^2}{\|z_0 - z_\star\|^2}$ is "Information-Theoretic Exact Method":[14]

$$y_k = (1-\beta_k)z_k + \beta_k\left(y_{k-1} - \frac{1}{L}\nabla f(y_{k-1})\right)$$
$$z_{k+1} = (1-\tfrac{\mu}{L}\delta_k)z_k + \tfrac{\mu}{L}\delta_k\left(y_k - \frac{1}{\mu}\nabla f(y_k)\right),$$

where the sequences $\{\beta_k\}$ and $\{\delta_k\}$ depends on some external sequence $\{A_k\}$ with $A_k \geqslant \left(1-\sqrt{\frac{\mu}{L}}\right)^{-2k}$.

[14] T, Drori (2023). "An optimal gradient method for smooth strongly convex minimization."

Information-Theoretic Exact Method (ITEM)

Example II: optimal method for $\frac{\|z_n - z_\star\|^2}{\|z_0 - z_\star\|^2}$ is "Information-Theoretic Exact Method":[14]

$$y_k = (1-\beta_k)z_k + \beta_k\left(y_{k-1} - \frac{1}{L}\nabla f(y_{k-1})\right)$$
$$z_{k+1} = (1-\tfrac{\mu}{L}\delta_k)z_k + \tfrac{\mu}{L}\delta_k\left(y_k - \frac{1}{\mu}\nabla f(y_k)\right),$$

where the sequences $\{\beta_k\}$ and $\{\delta_k\}$ depends on some external sequence $\{A_k\}$ with $A_k \geqslant \left(1-\sqrt{\frac{\mu}{L}}\right)^{-2k}$.

The (tight) guarantee is $\frac{\|z_n - z_\star\|^2}{\|z_0 - z_\star\|^2} \leqslant \frac{1}{1+\frac{\mu}{L}A_n} = O\left(\left(1-\sqrt{\frac{\mu}{L}}\right)^{2n}\right)$. Matches exact lower bound.[15]

[14] T, Drori (2023). "An optimal gradient method for smooth strongly convex minimization."

[15] Drori, T (2022). "On the oracle complexity of smooth strongly convex minimization."

A few instructive examples

Design first-order methods via PEPs:

- Drori, Teboulle (2014). "Performance of first-order methods for smooth convex minimization: a novel approach." Mathematical Programming 145(1).
- Kim, Fessler (2016). "Optimized methods for smooth convex optimization." Mathematical programming 159.
- Van Scoy, Freeman, Lynch (2017). "The fastest known globally convergent first-order method for minimizing strongly convex functions." IEEE Control Systems Magazine 39(3).
- Kim, Fessler (2021). "Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions." Journal of Optimization Theory and Applications 188(1).
- Altschuler, Parrilo (2023). "Acceleration by Stepsize Hedging I: Multi-Step Descent and the Silver Stepsize Schedule." Preprint.

A few instructive examples

Design first-order methods via PEPs:

- ◇ Drori, Teboulle (2014). "Performance of first-order methods for smooth convex minimization: a novel approach." Mathematical Programming 145(1).
- ◇ Kim, Fessler (2016). "Optimized methods for smooth convex optimization." Mathematical programming 159.
- ◇ Van Scoy, Freeman, Lynch (2017). "The fastest known globally convergent first-order method for minimizing strongly convex functions." IEEE Control Systems Magazine 39(3).
- ◇ Kim, Fessler (2021). "Optimizing the efficiency of first-order methods for decreasing the gradient of smooth convex functions." Journal of Optimization Theory and Applications 188(1).
- ◇ Altschuler, Parrilo (2023). "Acceleration by Stepsize Hedging I: Multi-Step Descent and the Silver Stepsize Schedule." Preprint.

... including "brutal" examples:

- ◇ Grimmer (2024). "Provably faster gradient descent via long steps." SIAM Journal on Optimization 34(3).
- ◇ Gupta, Van Parijs, Ryu (2024). "Branch-and-Bound Performance Estimation Programming: A Unified Methodology for Constructing Optimal Methods." Mathematical Programming 204(1).

Forewords and problem statement

Numerical examples

Design via idealized algorithms

Concluding remarks

Conjugate gradient-based design

Alternate way to design $\{\alpha_{i,j}\}$: via the study of

Conjugate gradient-based design

Alternate way to design $\{\alpha_{i,j}\}$: via the study of

$$w_{k+1} \in \operatorname*{argmin}_{x \in \mathbb{R}^d} \{f(w) : \ w \in w_0 + \operatorname{span}\{\nabla f(w_0), \ldots, \nabla f(w_k)\}\},$$

Conjugate gradient-based design

Alternate way to design $\{\alpha_{i,j}\}$: via the study of

$$w_{k+1} \in \operatorname*{argmin}_{x \in \mathbb{R}^d} \{f(w) : \ w \in w_0 + \operatorname{span}\{\nabla f(w_0), \ldots, \nabla f(w_k)\}\},$$

Let's exemplify:

$$w_{k+1} \in \operatorname*{argmin}_{x \in \mathbb{R}^d} \{f(w) : \ w \in w_k + \operatorname{span}\{\nabla f(w_k)\}\}.$$

Fixed stepsize policy from WC analysis of linesearch?

Target convergence rate (example):

Fixed stepsize policy from WC analysis of linesearch?

Target convergence rate (example):

$$\rho \overset{\text{(def)}}{=} \max_{w_0, w_1, f \in \mathcal{F}_{\mu,L}} \frac{f(w_1) - f_\star}{f(w_0) - f_\star} \text{ s.t. } \langle \nabla f(w_1), \nabla f(w_0) \rangle = 0, \ \langle \nabla f(w_1), w_1 - w_0 \rangle = 0,$$

Fixed stepsize policy from WC analysis of linesearch?

Target convergence rate (example):

$$\rho \overset{\text{(def)}}{=} \max_{w_{\mathbf{0}}, w_{\mathbf{1}}, f \in \mathcal{F}_{\mu,\mathrm{L}}} \frac{f(w_1) - f_\star}{f(w_0) - f_\star} \text{ s.t. } \langle \nabla f(w_1), \nabla f(w_0) \rangle = 0, \ \langle \nabla f(w_1), w_1 - w_0 \rangle = 0,$$

upper bound from a Lagrangian relaxation with $\lambda_1, \lambda_2 \in \mathbb{R}$:

$$\rho \leqslant \bar{\rho}(\lambda_1, \lambda_2) \overset{\text{(def)}}{=} \max_{x_{\mathbf{0}}, x_{\mathbf{1}}, f \in \mathcal{F}_{\mu,\mathrm{L}}} \left\{ \frac{f(w_1) - f_\star}{f(w_0) - f_\star} + \lambda_1 \langle \nabla f(w_1), \nabla f(w_0) \rangle + \lambda_2 \langle \nabla f(w_1), w_1 - w_0 \rangle \right\}.$$

Fixed stepsize policy from WC analysis of linesearch?

Target convergence rate (example):

$$\rho \overset{\text{(def)}}{=} \max_{w_0, w_1, f \in \mathcal{F}_{\mu,L}} \frac{f(w_1) - f_\star}{f(w_0) - f_\star} \text{ s.t. } \langle \nabla f(w_1), \nabla f(w_0) \rangle = 0, \ \langle \nabla f(w_1), w_1 - w_0 \rangle = 0,$$

upper bound from a Lagrangian relaxation with $\lambda_1, \lambda_2 \in \mathbb{R}$:

$$\rho \leqslant \bar{\rho}(\lambda_1, \lambda_2) \overset{\text{(def)}}{=} \max_{x_0, x_1, f \in \mathcal{F}_{\mu,L}} \left\{ \frac{f(w_1) - f_\star}{f(w_0) - f_\star} + \lambda_1 \langle \nabla f(w_1), \nabla f(w_0) \rangle + \lambda_2 \langle \nabla f(w_1), w_1 - w_0 \rangle \right\}.$$

We can also create an intermediary problem

$$\rho \leqslant \qquad\qquad \leqslant \bar{\rho}(\lambda_1, \lambda_2).$$

Fixed stepsize policy from WC analysis of linesearch?

Target convergence rate (example):

$$\rho \overset{\text{(def)}}{=} \max_{w_{\mathbf{0}}, w_{\mathbf{1}}, f \in \mathcal{F}_{\mu,\mathrm{L}}} \frac{f(w_1) - f_\star}{f(w_0) - f_\star} \text{ s.t. } \langle \nabla f(w_1), \nabla f(w_0) \rangle = 0, \ \langle \nabla f(w_1), w_1 - w_0 \rangle = 0,$$

upper bound from a Lagrangian relaxation with $\lambda_1, \lambda_2 \in \mathbb{R}$:

$$\rho \leqslant \bar{\rho}(\lambda_1, \lambda_2) \overset{\text{(def)}}{=} \max_{x_{\mathbf{0}}, x_{\mathbf{1}}, f \in \mathcal{F}_{\mu,\mathrm{L}}} \left\{ \frac{f(w_1) - f_\star}{f(w_0) - f_\star} + \lambda_1 \langle \nabla f(w_1), \nabla f(w_0) \rangle + \lambda_2 \langle \nabla f(w_1), w_1 - w_0 \rangle \right\}.$$

We can also create an intermediary problem

$$\rho \leqslant \max_{w_{\mathbf{0}}, w_{\mathbf{1}}, f \in \mathcal{F}_{\mu,\mathrm{L}}} \left\{ \frac{f(w_1) - f_\star}{f(w_0) - f_\star} \text{ s.t. } \lambda_1 \langle \nabla f(w_1), \nabla f(w_0) \rangle + \lambda_2 \langle \nabla f(w_1), w_1 - w_0 \rangle = 0 \right\} \leqslant \bar{\rho}(\lambda_1, \lambda_2).$$

Fixed stepsize policy from WC analysis of linesearch?

Target convergence rate (example):

$$\rho \overset{\text{(def)}}{=} \max_{w_{\mathbf{0}}, w_{\mathbf{1}}, f \in \mathcal{F}_{\mu,\mathrm{L}}} \frac{f(w_1) - f_\star}{f(w_0) - f_\star} \text{ s.t. } \langle \nabla f(w_1), \nabla f(w_0) \rangle = 0, \ \langle \nabla f(w_1), w_1 - w_0 \rangle = 0,$$

upper bound from a Lagrangian relaxation with $\lambda_1, \lambda_2 \in \mathbb{R}$:

$$\rho \leqslant \bar{\rho}(\lambda_1, \lambda_2) \overset{\text{(def)}}{=} \max_{x_{\mathbf{0}}, x_{\mathbf{1}}, f \in \mathcal{F}_{\mu,\mathrm{L}}} \left\{ \frac{f(w_1) - f_\star}{f(w_0) - f_\star} + \lambda_1 \langle \nabla f(w_1), \nabla f(w_0) \rangle + \lambda_2 \langle \nabla f(w_1), w_1 - w_0 \rangle \right\}.$$

We can also create an intermediary problem

$$\rho \leqslant \max_{w_{\mathbf{0}}, w_{\mathbf{1}}, f \in \mathcal{F}_{\mu,\mathrm{L}}} \left\{ \frac{f(w_1) - f_\star}{f(w_0) - f_\star} \text{ s.t. } \lambda_1 \langle \nabla f(w_1), \nabla f(w_0) \rangle + \lambda_2 \langle \nabla f(w_1), w_1 - w_0 \rangle = 0 \right\} \leqslant \bar{\rho}(\lambda_1, \lambda_2).$$

So, for any pair $\lambda_1, \lambda_2 \in \mathbb{R}$ we get:

Fixed stepsize policy from WC analysis of linesearch?

Target convergence rate (example):

$$\rho \overset{\text{(def)}}{=} \max_{w_{\mathbf{0}}, w_{\mathbf{1}}, f \in \mathcal{F}_{\mu,\mathrm{L}}} \frac{f(w_1) - f_\star}{f(w_0) - f_\star} \text{ s.t. } \langle \nabla f(w_1), \nabla f(w_0) \rangle = 0, \ \langle \nabla f(w_1), w_1 - w_0 \rangle = 0,$$

upper bound from a Lagrangian relaxation with $\lambda_1, \lambda_2 \in \mathbb{R}$:

$$\rho \leqslant \bar{\rho}(\lambda_1, \lambda_2) \overset{\text{(def)}}{=} \max_{x_{\mathbf{0}}, x_{\mathbf{1}}, f \in \mathcal{F}_{\mu,\mathrm{L}}} \left\{ \frac{f(w_1) - f_\star}{f(w_0) - f_\star} + \lambda_1 \langle \nabla f(w_1), \nabla f(w_0) \rangle + \lambda_2 \langle \nabla f(w_1), w_1 - w_0 \rangle \right\}.$$

We can also create an intermediary problem

$$\rho \leqslant \max_{w_{\mathbf{0}}, w_{\mathbf{1}}, f \in \mathcal{F}_{\mu,\mathrm{L}}} \left\{ \frac{f(w_1) - f_\star}{f(w_0) - f_\star} \text{ s.t. } \lambda_1 \langle \nabla f(w_1), \nabla f(w_0) \rangle + \lambda_2 \langle \nabla f(w_1), w_1 - w_0 \rangle = 0 \right\} \leqslant \bar{\rho}(\lambda_1, \lambda_2).$$

So, for any pair $\lambda_1, \lambda_2 \in \mathbb{R}$ we get:

- an upper bound $\bar{\rho}(\lambda_1, \lambda_2)$ (possibly $+\infty$) on ρ,

Fixed stepsize policy from WC analysis of linesearch?

Target convergence rate (example):

$$\rho \overset{\text{(def)}}{=} \max_{w_{\mathbf{0}}, w_{\mathbf{1}}, f \in \mathcal{F}_{\mu,\mathrm{L}}} \frac{f(w_1) - f_\star}{f(w_0) - f_\star} \text{ s.t. } \langle \nabla f(w_1), \nabla f(w_0) \rangle = 0, \ \langle \nabla f(w_1), w_1 - w_0 \rangle = 0,$$

upper bound from a Lagrangian relaxation with $\lambda_1, \lambda_2 \in \mathbb{R}$:

$$\rho \leqslant \bar{\rho}(\lambda_1, \lambda_2) \overset{\text{(def)}}{=} \max_{x_{\mathbf{0}}, x_{\mathbf{1}}, f \in \mathcal{F}_{\mu,\mathrm{L}}} \left\{ \frac{f(w_1) - f_\star}{f(w_0) - f_\star} + \lambda_1 \langle \nabla f(w_1), \nabla f(w_0) \rangle + \lambda_2 \langle \nabla f(w_1), w_1 - w_0 \rangle \right\}.$$

We can also create an intermediary problem

$$\rho \leqslant \max_{w_{\mathbf{0}}, w_{\mathbf{1}}, f \in \mathcal{F}_{\mu,\mathrm{L}}} \left\{ \frac{f(w_1) - f_\star}{f(w_0) - f_\star} \text{ s.t. } \lambda_1 \langle \nabla f(w_1), \nabla f(w_0) \rangle + \lambda_2 \langle \nabla f(w_1), w_1 - w_0 \rangle = 0 \right\} \leqslant \bar{\rho}(\lambda_1, \lambda_2).$$

So, for any pair $\lambda_1, \lambda_2 \in \mathbb{R}$ we get:

- an upper bound $\bar{\rho}(\lambda_1, \lambda_2)$ (possibly $+\infty$) on ρ,
- all methods satisfying $\langle \nabla f(w_1), \lambda_1 \nabla f(w_0) + \lambda_2 (w_1 - w_0) \rangle = 0$ have convergence rate at most $\bar{\rho}(\lambda_1, \lambda_2)$.

Fixed stepsize policy from WC analysis of linesearch?

Target convergence rate (example):

$$\rho \overset{\text{(def)}}{=} \max_{w_{\mathbf{0}}, w_{\mathbf{1}}, f \in \mathcal{F}_{\mu,\mathrm{L}}} \frac{f(w_1) - f_\star}{f(w_0) - f_\star} \text{ s.t. } \langle \nabla f(w_1), \nabla f(w_0) \rangle = 0, \ \langle \nabla f(w_1), w_1 - w_0 \rangle = 0,$$

upper bound from a Lagrangian relaxation with $\lambda_1, \lambda_2 \in \mathbb{R}$:

$$\rho \leqslant \bar{\rho}(\lambda_1, \lambda_2) \overset{\text{(def)}}{=} \max_{x_{\mathbf{0}}, x_{\mathbf{1}}, f \in \mathcal{F}_{\mu,\mathrm{L}}} \left\{ \frac{f(w_1) - f_\star}{f(w_0) - f_\star} + \lambda_1 \langle \nabla f(w_1), \nabla f(w_0) \rangle + \lambda_2 \langle \nabla f(w_1), w_1 - w_0 \rangle \right\}.$$

We can also create an intermediary problem

$$\rho \leqslant \max_{w_{\mathbf{0}}, w_{\mathbf{1}}, f \in \mathcal{F}_{\mu,\mathrm{L}}} \left\{ \frac{f(w_1) - f_\star}{f(w_0) - f_\star} \text{ s.t. } \lambda_1 \langle \nabla f(w_1), \nabla f(w_0) \rangle + \lambda_2 \langle \nabla f(w_1), w_1 - w_0 \rangle = 0 \right\} \leqslant \bar{\rho}(\lambda_1, \lambda_2).$$

So, for any pair $\lambda_1, \lambda_2 \in \mathbb{R}$ we get:

- an upper bound $\bar{\rho}(\lambda_1, \lambda_2)$ (possibly $+\infty$) on ρ,
- all methods satisfying $\langle \nabla f(w_1), \lambda_1 \nabla f(w_0) + \lambda_2 (w_1 - w_0) \rangle = 0$ have convergence rate at most $\bar{\rho}(\lambda_1, \lambda_2)$.

Bonus: there exists a choice $\lambda_1^\star, \lambda_2^\star$ such that

$$\rho = \bar{\rho}(\lambda_1^\star, \lambda_2^\star).$$

Design procedure

Algorithm constraints, and associated Lagrange multipliers:

$$\langle \nabla f(w_i), \nabla f(w_j) \rangle = 0, \qquad \text{for all } 0 \leqslant j < i = 1, \dots n \qquad : \beta_{i,j},$$
$$\langle \nabla f(w_i), w_j - w_0 \rangle = 0, \qquad \text{for all } 1 \leqslant j \leqslant i = 1, \dots n \qquad : \gamma_{i,j}.$$

Design procedure

Algorithm constraints, and associated Lagrange multipliers:

$$\langle \nabla f(w_i), \nabla f(w_j) \rangle = 0, \qquad \text{for all } 0 \leqslant j < i = 1, \dots n \qquad : \beta_{i,j},$$
$$\langle \nabla f(w_i), w_j - w_0 \rangle = 0, \qquad \text{for all } 1 \leqslant j \leqslant i = 1, \dots n \qquad : \gamma_{i,j}.$$

... replaced by:

$$\langle \nabla f(w_i), \sum_{j=1}^{i} \beta_{i,j}(w_j - w_0) + \sum_{j=0}^{i-1} \gamma_{i,j} \nabla f(w_j) \rangle = 0 \text{ for all } i = 1, \dots, n$$

Design procedure

Algorithm constraints, and associated Lagrange multipliers:

$$\langle \nabla f(w_i), \nabla f(w_j) \rangle = 0, \qquad \text{for all } 0 \leqslant j < i = 1, \dots n \qquad : \beta_{i,j},$$
$$\langle \nabla f(w_i), w_j - w_0 \rangle = 0, \qquad \text{for all } 1 \leqslant j \leqslant i = 1, \dots n \qquad : \gamma_{i,j}.$$

... replaced by:

$$\langle \nabla f(w_i), \sum_{j=1}^{i} \beta_{i,j}(w_j - w_0) + \sum_{j=0}^{i-1} \gamma_{i,j} \nabla f(w_j) \rangle = 0 \text{ for all } i = 1, \dots, n$$

Subspace-search elimination (abstract)

Design procedure

Algorithm constraints, and associated Lagrange multipliers:

$$\langle \nabla f(w_i), \nabla f(w_j) \rangle = 0, \qquad \text{for all } 0 \leqslant j < i = 1, \dots n \qquad : \beta_{i,j},$$
$$\langle \nabla f(w_i), w_j - w_0 \rangle = 0, \qquad \text{for all } 1 \leqslant j \leqslant i = 1, \dots n \qquad : \gamma_{i,j}.$$

... replaced by:

$$\langle \nabla f(w_i), \sum_{j=1}^{i} \beta_{i,j}(w_j - w_0) + \sum_{j=0}^{i-1} \gamma_{i,j} \nabla f(w_j) \rangle = 0 \text{ for all } i = 1, \dots, n$$

Subspace-search elimination (abstract)

1. Choose $n \geqslant 0$, $\mathcal{F}$.

Design procedure

Algorithm constraints, and associated Lagrange multipliers:

$$\langle \nabla f(w_i), \nabla f(w_j) \rangle = 0, \qquad \text{for all } 0 \leqslant j < i = 1, \dots n \qquad : \beta_{i,j},$$
$$\langle \nabla f(w_i), w_j - w_0 \rangle = 0, \qquad \text{for all } 1 \leqslant j \leqslant i = 1, \dots n \qquad : \gamma_{i,j}.$$

... replaced by:

$$\langle \nabla f(w_i), \sum_{j=1}^{i} \beta_{i,j}(w_j - w_0) + \sum_{j=0}^{i-1} \gamma_{i,j} \nabla f(w_j) \rangle = 0 \text{ for all } i = 1, \dots, n$$

Subspace-search elimination (abstract)

1. Choose $n \geqslant 0$, $\mathcal{F}$.
2. Find a feasible solution $\{\beta_{i,j}\}, \{\gamma_{i,j}\}$ to (dual) PEP for the greedy method with rate $\bar{\rho}$.

Design procedure

Algorithm constraints, and associated Lagrange multipliers:

$$\langle \nabla f(w_i), \nabla f(w_j) \rangle = 0, \qquad \text{for all } 0 \leqslant j < i = 1, \dots n \qquad : \beta_{i,j},$$
$$\langle \nabla f(w_i), w_j - w_0 \rangle = 0, \qquad \text{for all } 1 \leqslant j \leqslant i = 1, \dots n \qquad : \gamma_{i,j}.$$

... replaced by:

$$\langle \nabla f(w_i), \sum_{j=1}^{i} \beta_{i,j}(w_j - w_0) + \sum_{j=0}^{i-1} \gamma_{i,j} \nabla f(w_j) \rangle = 0 \text{ for all } i = 1, \dots, n$$

Subspace-search elimination (abstract)

1. Choose $n \geqslant 0$, $\mathcal{F}$.
2. Find a feasible solution $\{\beta_{i,j}\}, \{\gamma_{i,j}\}$ to (dual) PEP for the greedy method with rate $\bar{\rho}$.
3. Any method satisfying

$$\langle \nabla f(w_i), \sum_{j=1}^{i} \beta_{i,j}(w_j - w_0) + \sum_{j=0}^{i-1} \gamma_{i,j} \nabla f(w_j) \rangle = 0 \text{ for all } i = 1, \dots n$$

benefits from the same worst-case convergence rate $\bar{\rho}$.

Optimized gradient methods

Smooth convex minimization setting:

$$\min_{x \in \mathbb{R}^d} f(x)$$

with f being L-smooth and convex.

Optimized gradient methods

Smooth convex minimization setting:

$$\min_{x \in \mathbb{R}^d} f(x)$$

with f being L-smooth and convex.

Lower bound for large-scale setting ($d \geqslant n+2$) by Drori (2017):

$$f(x_n) - f(x_\star) \geqslant \frac{L\|x_0 - x_\star\|^2}{2\theta_n^2},$$

with $\theta_0 = 1$, and:

$$\theta_{i+1} = \begin{cases} \frac{1+\sqrt{4\theta_i^2+1}}{2} & \text{if } i \leqslant n-2, \\ \frac{1+\sqrt{8\theta_i^2+1}}{2} & \text{if } i = n-1. \end{cases}$$

As a result from subspace-search elimination, all methods satisfying (for $i = 1, \dots, n$)

$$\langle \nabla f(x_i); x_i - \left[\left(1 - \frac{1}{\theta_i}\right) \left(x_{i-1} - \tfrac{1}{L} \nabla f(x_{i-1})\right) + \frac{1}{\theta_i} \left(x_0 - \tfrac{2}{L} \sum_{j=0}^{i-1} \theta_j \nabla f(x_j) \right) \right] \rangle \leqslant 0$$

benefit from the same guarantee:

$$f(x_n) - f(x_\star) \leqslant \frac{L \|x_0 - x_\star\|^2}{2\theta_n^2},$$

Greedy First-order Method (GFOM)

Inputs: f, x_0, n.

For $i = 1, 2, \ldots, n$

$$x_{i+1} = \operatorname*{argmin}_{x \in \mathbb{R}^d} \{f(x) : \; x \in x_0 + \operatorname{span}\{\nabla f(x_0), \ldots, \nabla f(x_{i-1})\}\}.$$

Worst-case guarantee:

$$f(x_n) - f(x_\star) \leqslant \frac{L\|x_0 - x_\star\|^2}{2\theta_n^2}.$$

Optimized gradient methods

Three methods with the same (optimal) worst-case behavior

Optimized gradient method with exact line-search

Inputs: f, x_0, n.

For $i = 1, \ldots, n$

$$y_i = \left(1 - \frac{1}{\theta_i}\right) x_{i-1} + \frac{1}{\theta_i} x_0$$

$$d_i = \left(1 - \frac{1}{\theta_i}\right) \nabla f(x_{i-1}) + \frac{1}{\theta_i} \left(2 \sum_{j=0}^{i-1} \theta_j \nabla f(x_j)\right)$$

$$\alpha = \operatorname*{argmin}_{\alpha \in \mathbb{R}} f(y_i + \alpha d_i)$$

$$x_i = y_i + \alpha d_i$$

Worst-case guarantee: $f(x_n) - f(x_\star) \leqslant \frac{L\|x_0 - x_\star\|^2}{2\theta_n^2}$

Optimized gradient methods

Three methods with the same (optimal) worst-case behavior

Optimized gradient method

Inputs: f, x_0, n.

For $i = 1, \dots, n$

$$y_i = x_{i-1} - \frac{1}{L}\nabla f(x_{i-1})$$

$$z_i = x_0 - \frac{2}{L}\sum_{j=0}^{i-1} \theta_j \nabla f(x_j)$$

$$x_i = \left(1 - \frac{1}{\theta_i}\right) y_i + \frac{1}{\theta_i} z_i$$

Worst-case guarantee: $f(x_n) - f(x_\star) \leqslant \frac{L\|x_0 - x_\star\|^2}{2\theta_n^2}$.

Forewords and problem statement

Numerical examples

Design via idealized algorithms

Concluding remarks

Feedback and challenging questions to the others

https://forms.gle/r5sPXw7ak1wXM3ie6

Concluding remarks

Performance estimation's philosophy

Concluding remarks

Performance estimation's philosophy

- ◇ numerically allows obtaining tight bounds (rigorous baselines),
  - – fast prototyping
  - – worth checking before trying to prove a method works.

Concluding remarks

Performance estimation's philosophy

- ◇ numerically allows obtaining tight bounds (rigorous baselines),
  - – fast prototyping
  - – worth checking before trying to prove a method works.
- ◇ algebraic insights into performance analyses:
  - – analyses are dual feasible points,
  - – analyses are linear combinations of certain specific inequalities.

Concluding remarks

Performance estimation's philosophy

- ◇ numerically allows obtaining tight bounds (rigorous baselines),
  - – fast prototyping
  - – worth checking before trying to prove a method works.
- ◇ algebraic insights into performance analyses:
  - – analyses are dual feasible points,
  - – analyses are linear combinations of certain specific inequalities.

Byproducts:

- ◇ computer-assisted design of analyses,
- ◇ computer-assisted design of numerical methods,
- ◇ step towards reproducible theory
  - – validation & benchmark tool for proofs (also for reviews ☺).

Take-home messages

Worst-cases are solutions to optimization problems.

Take-home messages

Worst-cases are solutions to optimization problems.

Sometimes, those optimization problems are tractable.

Take-home messages

Worst-cases are solutions to optimization problems.

Sometimes, those optimization problems are tractable.

Often tractable for first-order methods in convex optimization!

Take-home messages

Worst-cases are solutions to optimization problems.

Sometimes, those optimization problems are tractable.

Often tractable for first-order methods in convex optimization!

Acceleration/optimal methods by optimizing worst-cases.