Forward-Backward L-BFGS for large scale nonsmooth convex optimization

Panos Patrinos
IMT Lucca
(KU Leuven starting Sept. 1st, 2015)
joint work with Lorenzo Stella
PhD student, IMT Lucca

IMTEK, May 12, 2015

Nonsmooth convex optimization

composite form

$$\text{minimize} \;\; f(x) + g(x)$$

separable form

$$\begin{aligned} &\text{minimize} \;\; f(x) + g(z) \\ &\text{subject to} \; Ax + Bz = b \end{aligned}$$

- f, g convex functions (possibly nonsmooth)
- norm regularized problems

f is loss function $\qquad$ g is regularizer

- constrained problems

$$g(x) = \delta_C(x) = \begin{cases} 0, & \text{if } x \in C \\ +\infty, & \text{otherwise} \end{cases}$$

- statistics, machine learning, signal processing, communications, control, system identification, image analysis...

Operator splitting methods

$$\text{minimize} \;\; f(x) + g(x) \qquad\qquad \begin{array}{l}\text{minimize} \;\; f(x) + g(z) \\ \text{subject to} \; Ax + Bz = b\end{array}$$

- solve problem by handling each term separately (or its conjugate)
  - ✓ perform either a gradient step (explicit or forward)

$$z = x - \gamma \nabla h(x)$$

  - ✓ or a proximal step (implicit or backward)

$$z = \mathrm{prox}_{\gamma h}(x) = \underset{z \in \mathbb{R}^n}{\arg\min} \left\{ h(z) + \tfrac{1}{2\gamma} \|z - x\|^2 \right\}$$

- forward-backward splitting (FBS), Douglas-Rachford splitting (DRS)
- AMM, ADMM (dual versions of FBS and DRS)
- simple, exploit structure; distributed implementations
- **but**: very prone to ill-conditioning, as any first-order method

Large-scale smooth convex optimization

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \; f(x) \qquad n \text{ large}$$

- a single iteration of Newton's method might take forever
- first-order methods seem to be the only option
- (fast) gradient method: simple to implement, easy to parallelize
- "*Subsequently, Nesterov [1983] presented an algorithm that attains this optimal bound...* **We feel that this approach is unlikely to be effective in practice...**"[1]
- **L-BFGS/nonlinear CG**: methods of choice for large-scale problems

Can we extend L-BFGS to nonsmooth problems while maintaining favorable structural properties of splitting methods?

[1] Nocedal, J., and S. J. Wright. "Nonlinear optimization." Springer (2006), pp. 133

Goals

- interpret **forward-backward splitting** as a **gradient method**...
- on a smooth function we call **forward-backward envelope (FBE)**
- generalizes the notion of Moreau envelope

- can apply any method for **unconstrained smooth optimization**...
- to solve **nonsmooth problems**...
- while retaining favorable features of splitting methods
  - ✓ forward-backward L-BFGS
  - ✓ alternating minimization L-BFGS
- require exactly the **same oracle** with FBS
- optimal global complexity estimates but **much faster in practice**

Outline

- FBS and forward-backward envelope (FBE)
- Forward-Backward L-BFGS and acceleration
- AMM and alternating minimization L-BFGS
- **ForBES** (Forward Backward Envelope Solver)
- conclusions

Outline

- FBS and forward-backward envelope (FBE)

Forward-Backward L-BFGS and acceleration

AMM and alternating minimization L-BFGS

ForBES (Forward Backward Envelope Solver)

conclusions

Problem Definition

$$\text{minimize} \;\; \varphi(x) = f(x) + g(x)$$

- f, g closed proper convex
- $f : \mathbb{R}^n \to \mathbb{R}$ is L_f-strongly smooth

$$f(z) \leq Q^f_{1/L_f}(z; x) = f(x) + \langle \nabla f(x), z - x \rangle + \frac{L_f}{2}\|z - x\|^2 \qquad \forall x, z$$

- $g : \mathbb{R}^n \to \mathbb{R} \cup \{+\infty\}$ has easily computable proximal mapping

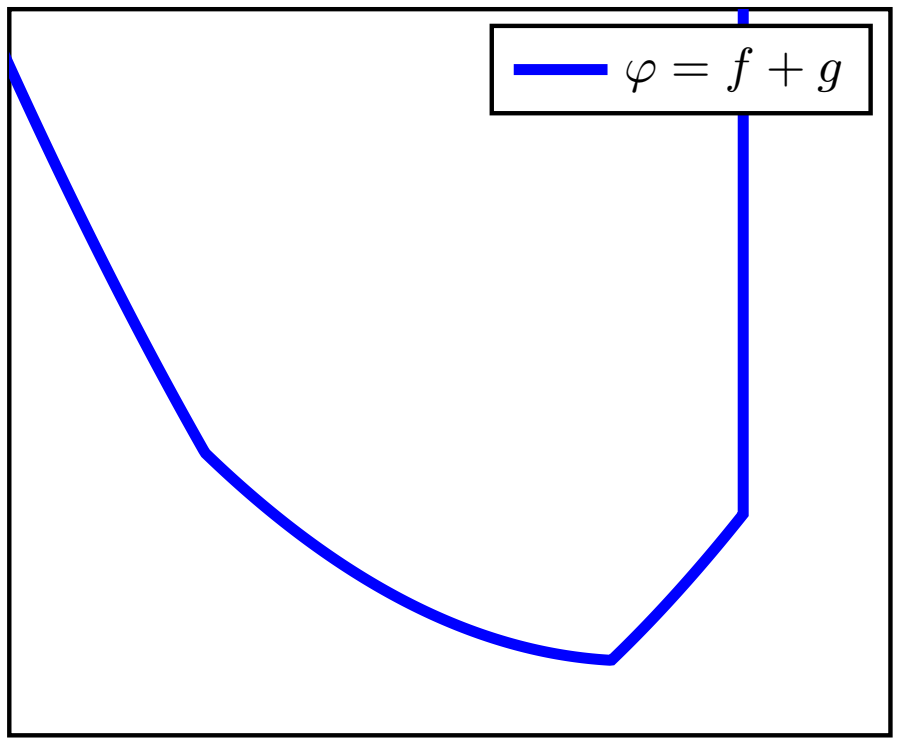
$$\text{prox}_{\gamma g}(x) = \operatorname*{arg\,min}_{z \in \mathbb{R}^n} \left\{ g(z) + \frac{1}{2\gamma}\|z - x\|^2 \right\} \qquad \gamma > 0$$

Forward-Backward Splitting (FBS)

$$\text{minimize } \varphi(x) = f(x) + g(x)$$

- **forward-backward splitting** (aka **proximal gradient**) ($\gamma \leq 1/L_f$)

$$x^{k+1} = \underset{z \in \mathbb{R}^n}{\arg\min} \{ \underbrace{f(x^k) + \langle \nabla f(x^k), z - x^k \rangle + \frac{1}{2\gamma} \|z - x^k\|^2}_{Q_\gamma^f(z; x^k)} + g(z) \}$$

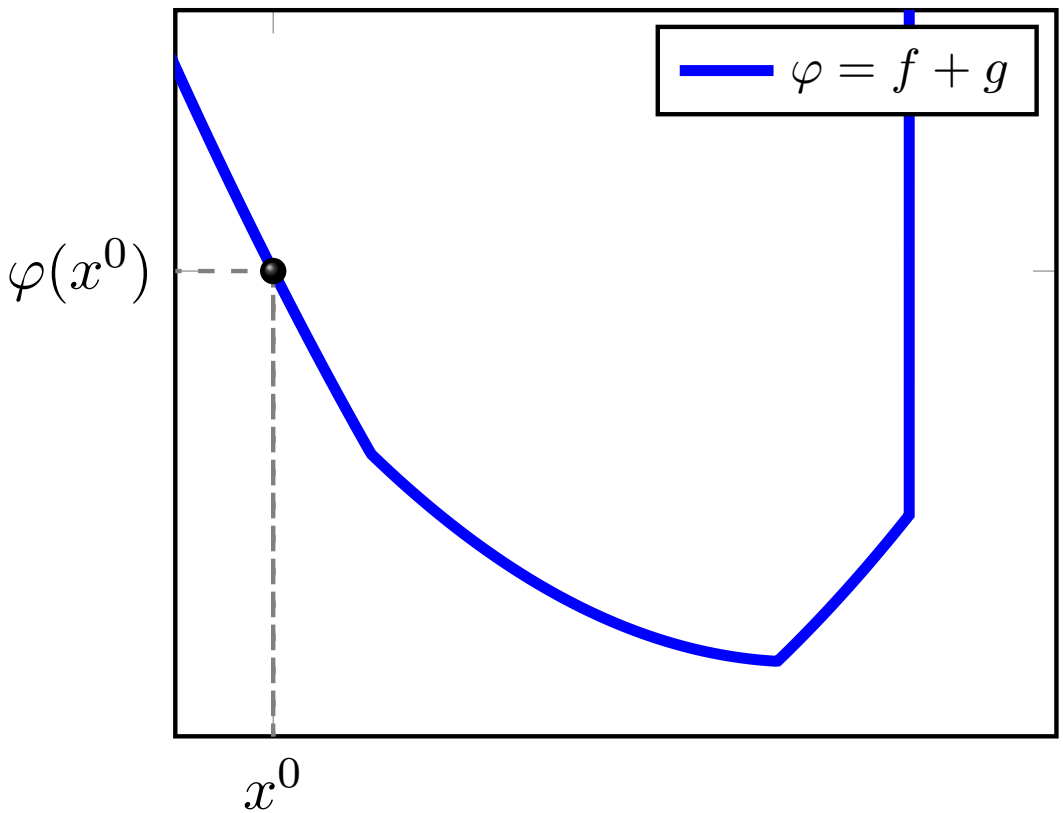


Forward-Backward Splitting (FBS)

$$\text{minimize } \varphi(x) = f(x) + g(x)$$

- **forward-backward splitting** (aka **proximal gradient**) ($\gamma \leq 1/L_f$)

$$x^{k+1} = \arg\min_{z \in \mathbb{R}^n} \{ \underbrace{f(x^k) + \langle \nabla f(x^k), z - x^k \rangle + \frac{1}{2\gamma} \|z - x^k\|^2}_{Q_\gamma^f(z; x^k)} + g(z) \}$$

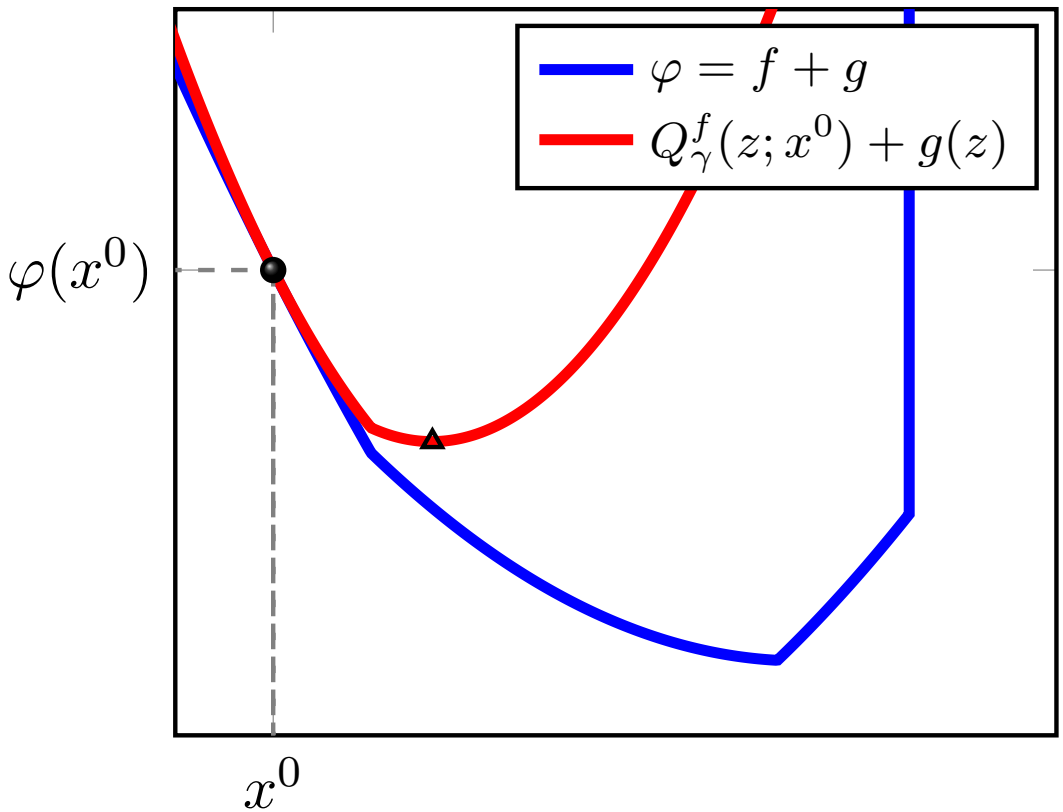


Forward-Backward Splitting (FBS)

$$\text{minimize } \varphi(x) = f(x) + g(x)$$

- **forward-backward splitting** (aka **proximal gradient**) ($\gamma \leq 1/L_f$)

$$x^{k+1} = \underset{z \in \mathbb{R}^n}{\arg\min} \{ \underbrace{f(x^k) + \langle \nabla f(x^k), z - x^k \rangle + \frac{1}{2\gamma} \|z - x^k\|^2}_{Q_\gamma^f(z; x^k)} + g(z) \}$$

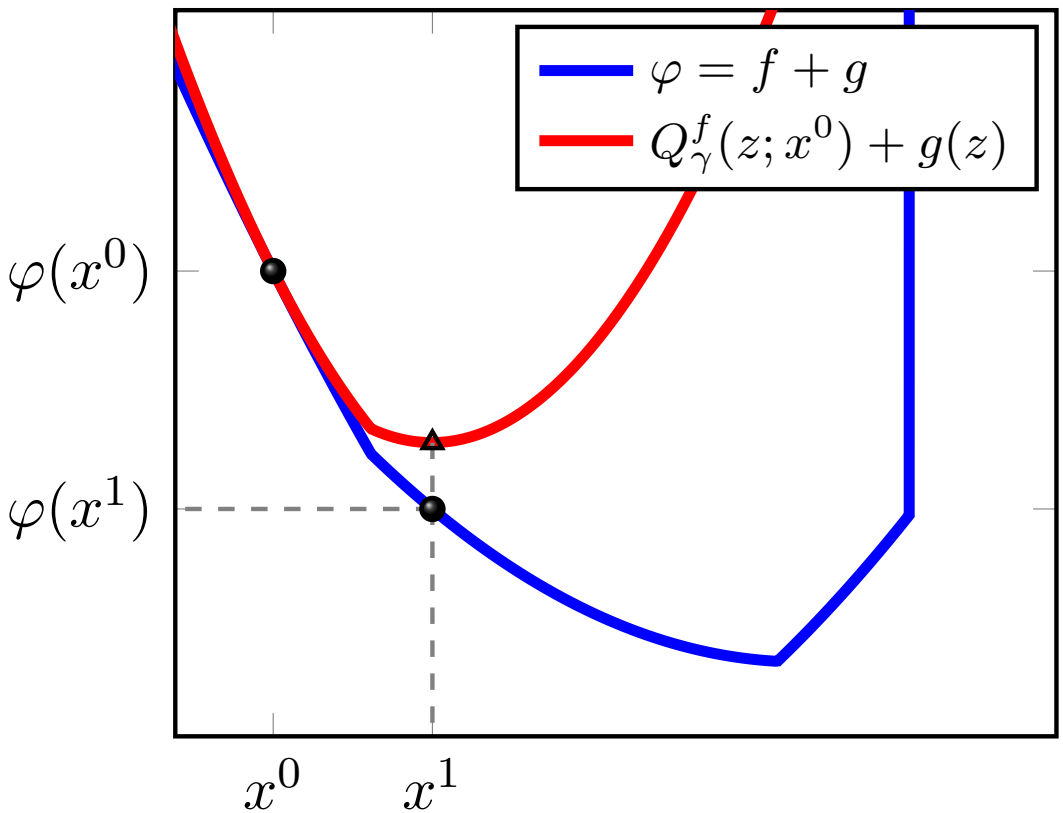


Forward-Backward Splitting (FBS)

$$\text{minimize } \varphi(x) = f(x) + g(x)$$

- **forward-backward splitting** (aka **proximal gradient**) ($\gamma \leq 1/L_f$)

$$x^{k+1} = \underset{z \in \mathbb{R}^n}{\arg\min} \{ \underbrace{f(x^k) + \langle \nabla f(x^k), z - x^k \rangle + \frac{1}{2\gamma} \|z - x^k\|^2}_{Q_\gamma^f(z; x^k)} + g(z) \}$$

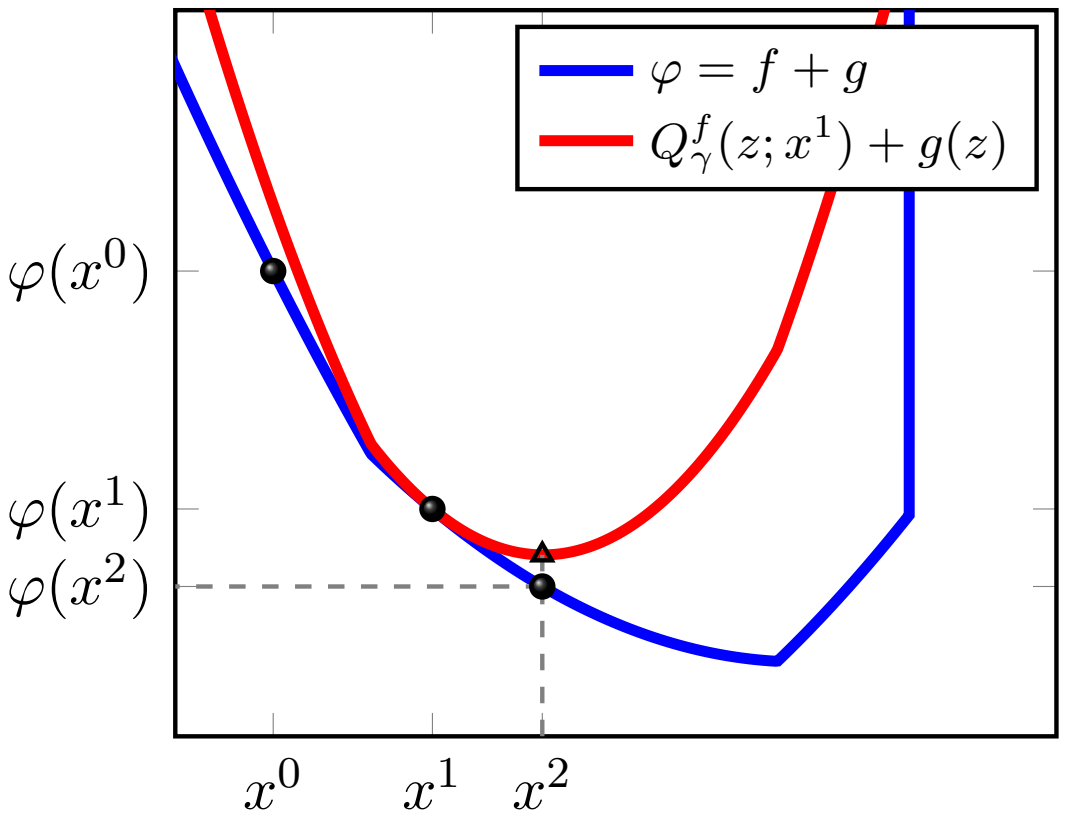


Forward-Backward Splitting (FBS)

$$\text{minimize } \varphi(x) = f(x) + g(x)$$

- **forward-backward splitting** (aka **proximal gradient**) ($\gamma \leq 1/L_f$)

$$x^{k+1} = \underset{z \in \mathbb{R}^n}{\arg\min} \{ \underbrace{f(x^k) + \langle \nabla f(x^k), z - x^k \rangle + \frac{1}{2\gamma} \|z - x^k\|^2}_{Q_\gamma^f(z; x^k)} + g(z) \}$$

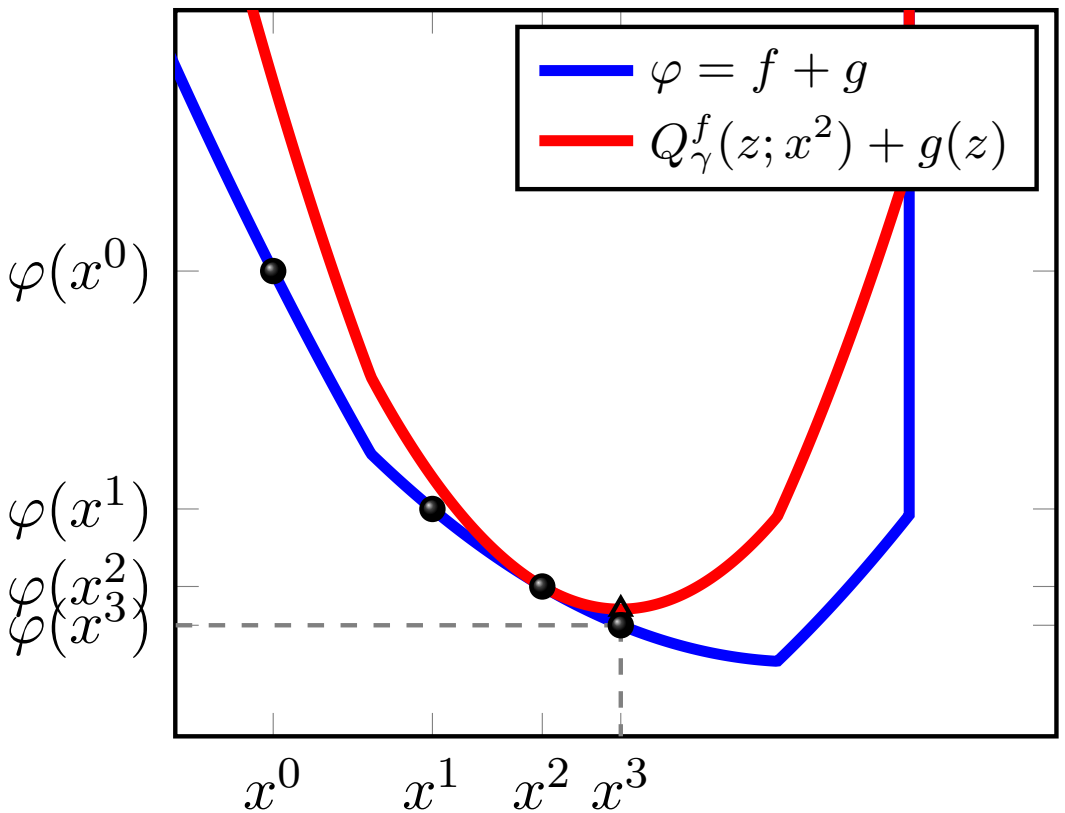


Forward-Backward Splitting (FBS)

$$\text{minimize } \varphi(x) = f(x) + g(x)$$

- **forward-backward splitting** (aka **proximal gradient**) ($\gamma \leq 1/L_f$)

$$x^{k+1} = \underset{z \in \mathbb{R}^n}{\arg\min} \{ \underbrace{f(x^k) + \langle \nabla f(x^k), z - x^k \rangle + \frac{1}{2\gamma} \|z - x^k\|^2}_{Q_\gamma^f(z; x^k)} + g(z) \}$$



Forward-Backward Splitting (FBS)

$$\text{minimize } \varphi(x) = f(x) + g(x)$$

- iteration can be expressed as

$$x^{k+1} = \text{prox}_{\gamma g}(x^k - \gamma \nabla f(x^k))$$

- fixed point iteration for optimality condition

$$x = \text{prox}_{\gamma g}(x - \gamma \nabla f(x)), \qquad \gamma > 0$$

- generalizes classical methods

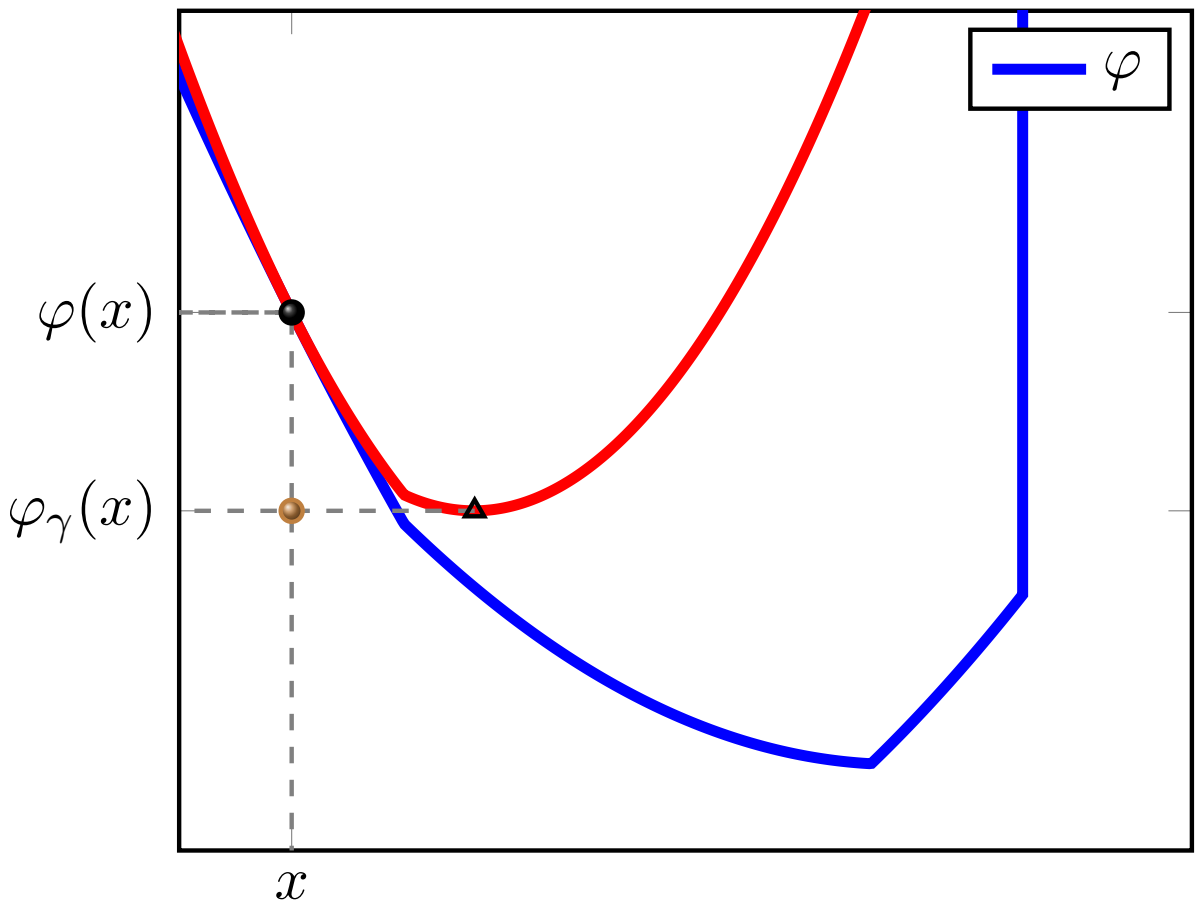
$$x^{k+1} = \begin{cases} x^k - \gamma \nabla f(x^k) & g = 0 \quad \text{gradient method} \\ \Pi_C(x^k - \gamma \nabla f(x^k)) & g = \delta_C \quad \text{gradient projection} \\ \text{prox}_{\gamma g}(x^k) & f = 0 \quad \text{proximal point algorithm} \end{cases}$$

- accelerated versions-FISTA (Nesterov, Beck, Teboulle)

Forward-Backward Envelope (FBE)

value function of FBS subproblem ($\gamma \leq 1/L_f$)

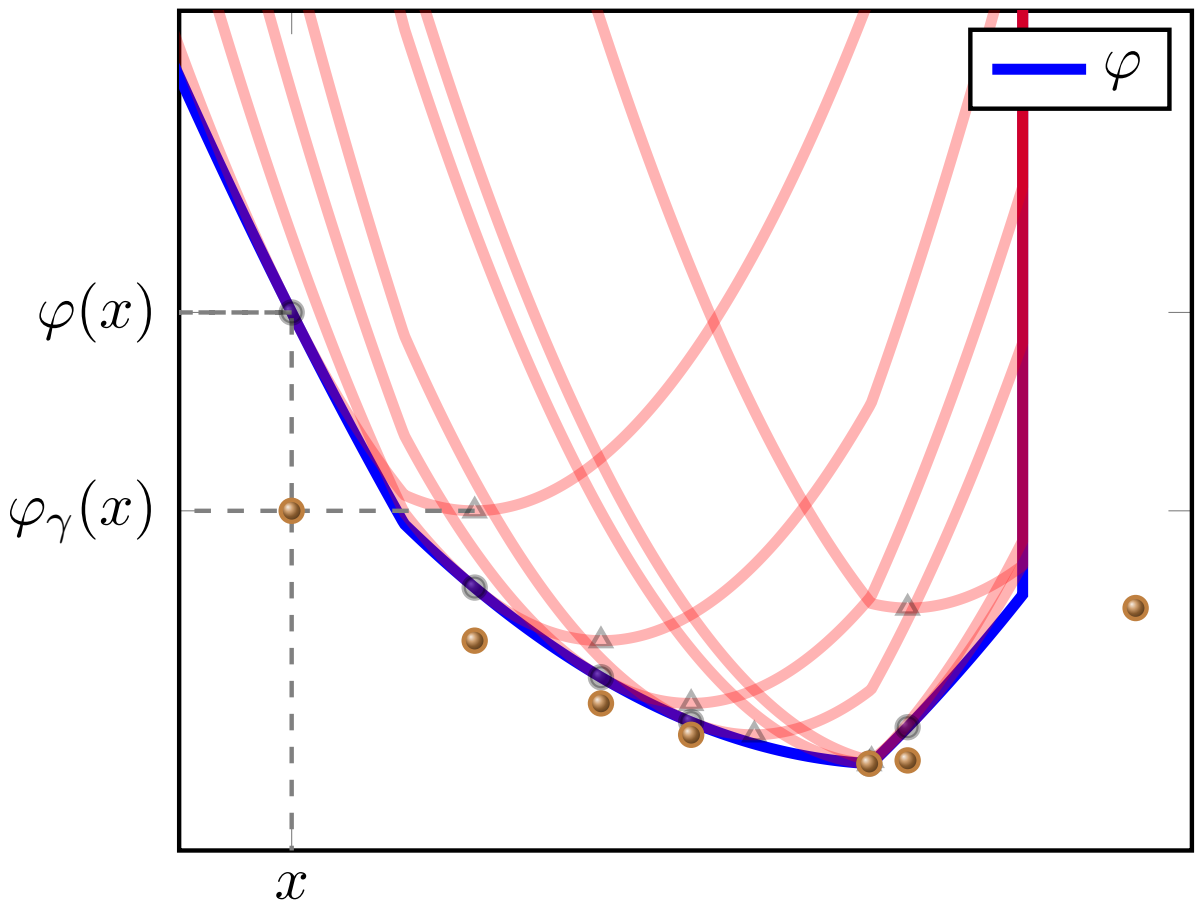
$$\varphi_\gamma(x) = \min_z \left\{ f(x) + \langle \nabla f(x), z - x \rangle + g(z) + \tfrac{1}{2\gamma}\|z - x\|^2 \right\}$$



Forward-Backward Envelope (FBE)

value function of FBS subproblem ($\gamma \leq 1/L_f$)

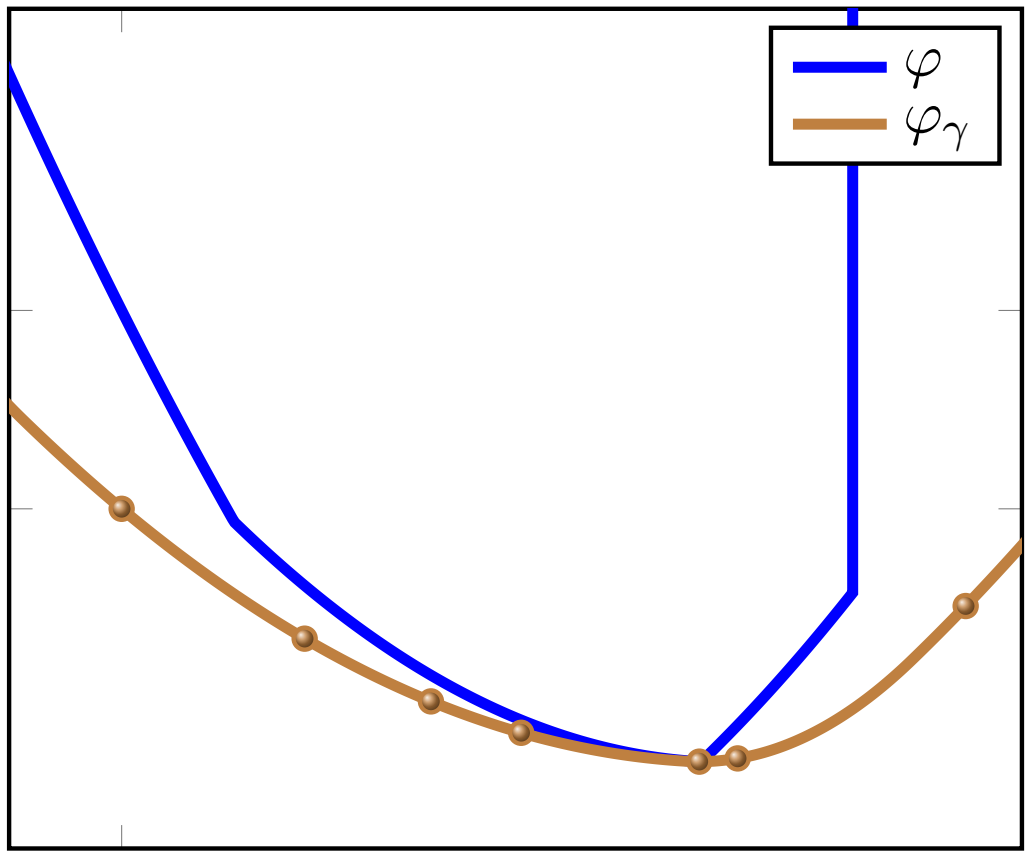
$$\varphi_\gamma(x) = \min_z \left\{ f(x) + \langle \nabla f(x), z - x \rangle + g(z) + \tfrac{1}{2\gamma}\|z - x\|^2 \right\}$$



Forward-Backward Envelope (FBE)

value function of FBS subproblem ($\gamma \leq 1/L_f$)

$$\varphi_\gamma(x) = \min_z \left\{ f(x) + \langle \nabla f(x), z - x \rangle + g(z) + \tfrac{1}{2\gamma}\|z - x\|^2 \right\}$$



Key property of FBE

upper bound

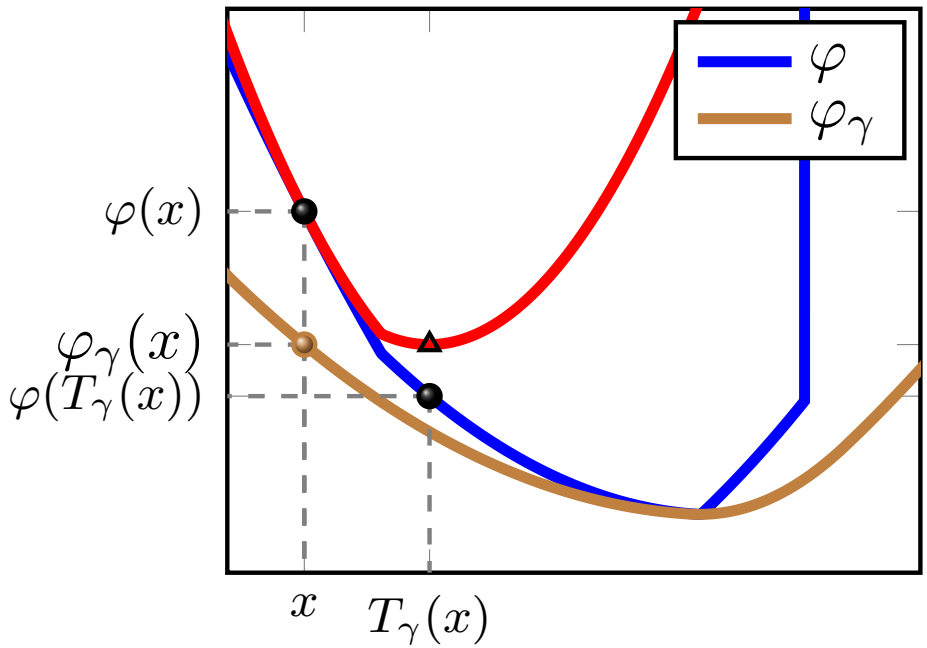
$$\varphi_\gamma(x) \le \varphi(x) - \tfrac{1}{2\gamma}\|R_\gamma(x)\|^2$$

lower bound

$$\varphi_\gamma(x) \ge \varphi(T_\gamma(x)) + \tfrac{1-\gamma L_f}{2\gamma}\|R_\gamma(x)\|^2$$

where $R_\gamma(x) = x - T_\gamma(x)$

$T_\gamma(x) = \operatorname{prox}_{\gamma g}(x - \gamma\nabla f(x))$



- minimizing FBE is equivalent to minimizing φ if $\gamma \in (0, 1/L_f]$

$$\inf \varphi = \inf \varphi_\gamma \qquad \arg\min \varphi = \arg\min \varphi_\gamma$$

Key property of FBE

upper bound

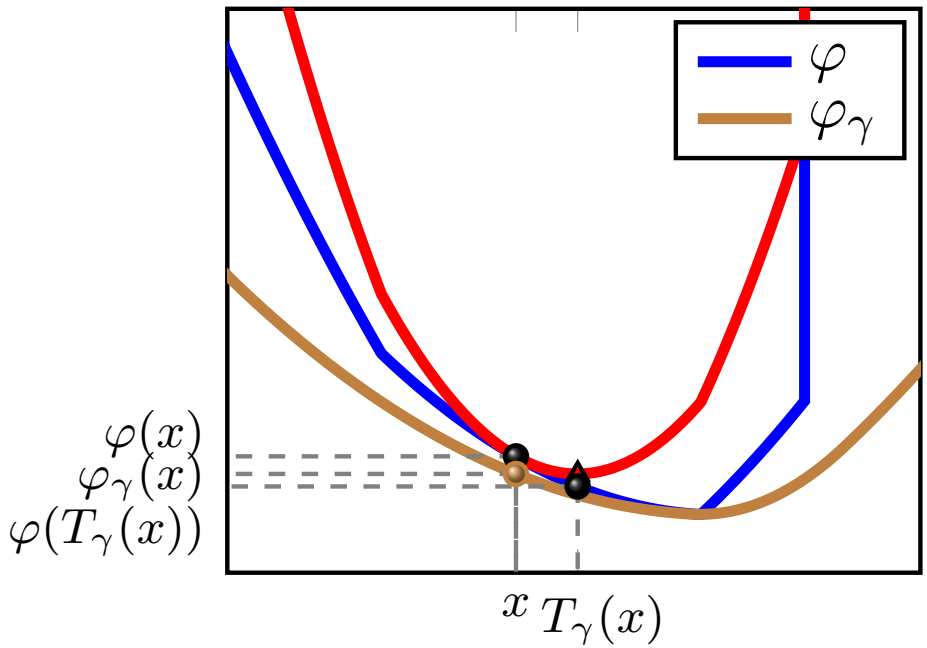
$$\varphi_\gamma(x) \le \varphi(x) - \tfrac{1}{2\gamma}\|R_\gamma(x)\|^2$$

lower bound

$$\varphi_\gamma(x) \ge \varphi(T_\gamma(x)) + \tfrac{1-\gamma L_f}{2\gamma}\|R_\gamma(x)\|^2$$

where $R_\gamma(x) = x - T_\gamma(x)$

$T_\gamma(x) = \operatorname{prox}_{\gamma g}(x - \gamma\nabla f(x))$



- minimizing FBE is equivalent to minimizing φ if $\gamma \in (0, 1/L_f]$

$$\inf \varphi = \inf \varphi_\gamma \qquad \arg\min \varphi = \arg\min \varphi_\gamma$$

Key property of FBE

upper bound

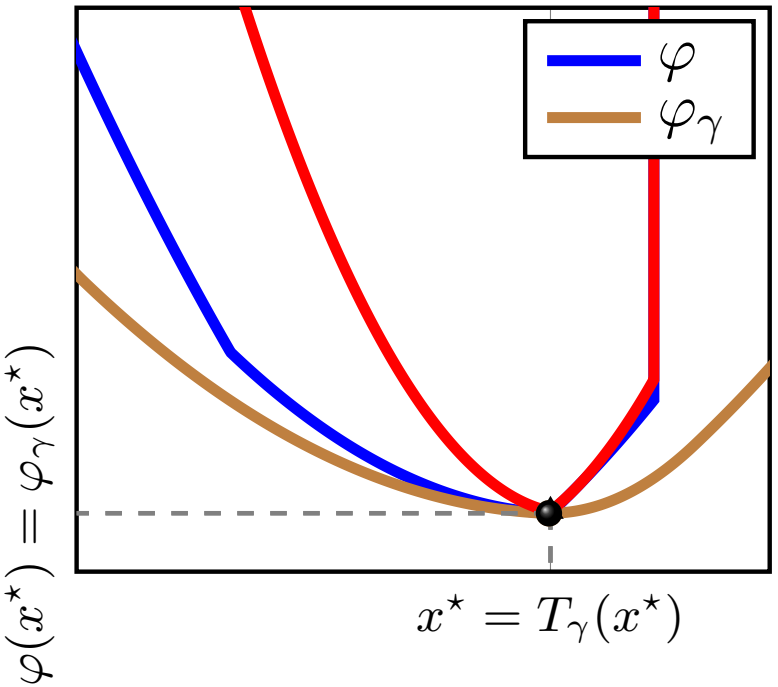
$$\varphi_\gamma(x) \le \varphi(x) - \tfrac{1}{2\gamma}\|R_\gamma(x)\|^2$$

lower bound

$$\varphi_\gamma(x) \ge \varphi(T_\gamma(x)) + \tfrac{1-\gamma L_f}{2\gamma}\|R_\gamma(x)\|^2$$

where $R_\gamma(x) = x - T_\gamma(x)$

$T_\gamma(x) = \operatorname{prox}_{\gamma g}(x - \gamma\nabla f(x))$



- minimizing FBE is equivalent to minimizing φ if $\gamma \in (0, 1/L_f]$

$$\inf \varphi = \inf \varphi_\gamma \qquad \arg\min \varphi = \arg\min \varphi_\gamma$$

Differentiability of FBE

FBE can be expressed as

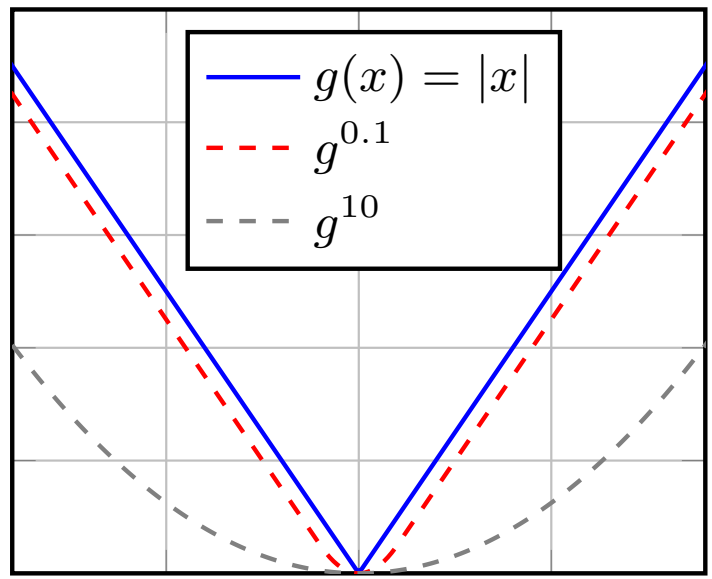
$$\varphi_\gamma(x) = f(x) - \tfrac{\gamma}{2}\|\nabla f(x)\|^2 + g^\gamma(x - \gamma\nabla f(x))$$

g^γ is Moreau envelope of g

$$g^\gamma(x) = \inf_{z\in\mathbb{R}^n}\{g(z) + \tfrac{1}{2\gamma}\|z - x\|^2\}$$

smooth approximation of g

$$\nabla g^\gamma(x) = \gamma^{-1}(x - \operatorname{prox}_{\gamma g}(x))$$



- if f is $\mathcal{C}^2$ then FBE is $\mathcal{C}^1$ with

$$\nabla\varphi_\gamma(x) = \gamma^{-1}(I - \gamma\nabla^2 f(x))(x - \operatorname{prox}_{\gamma g}(x - \gamma\nabla f(x))$$

Problem equivalence

if f is $\mathcal{C}^2$ then FBE is $\mathcal{C}^1$ with

$$\nabla\varphi_\gamma(x) = \gamma^{-1}(I - \gamma\nabla^2 f(x))(x - \operatorname{prox}_{\gamma g}(x - \gamma\nabla f(x))$$

- for $\gamma \in (0, 1/L_f)$ stationary points of φ_γ =minimizers of φ

$$\arg\min \varphi = \arg\min \varphi_\gamma = \operatorname{zer} \nabla\varphi_\gamma$$

- FBS is a **variable metric gradient method** for FBE

$$x^{k+1} = x^k - \gamma(I - \gamma\nabla^2 f(x))^{-1}\nabla\varphi_\gamma(x^k)$$

- extends classical result of Rockafellar (put $f = 0$)

$$x^{k+1} = \operatorname{prox}_{\gamma g}(x^k) = x^k - \gamma\nabla g^\gamma(x^k)$$

- paves the way for applying L-BFGS to nonsmooth composite problems

Outline

FBS and forward-backward envelope (FBE)

- Forward-Backward L-BFGS and acceleration

AMM and alternating minimization L-BFGS

ForBES (Forward Backward Envelope Solver)

conclusions

Algorithm 1 Forward-Backward L-BFGS

1: choose $\gamma \in (0, 1/L_f)$, starting point $x^0 \in \mathbb{R}^n$, integer $m > 0$
2: compute $d^k = -H^k \nabla \varphi_\gamma(x^k)$
3: update $x^{k+1} = x^k + \tau_k d^k$ where τ_k satisfies Wolfe condition
4: **If** $k > m$ **then** discard $\{s_{k-m}, y_{k-m}\}$
5: store $s^k = x^{k+1} - x^k$, $y^k = \nabla \varphi_\gamma(x^{k+1}) - \nabla \varphi_\gamma(x^k)$, $\rho_k = 1/\langle y_k, s_k \rangle$
6: $k \leftarrow k + 1$ and go to 2

- H^k can be seen as approximation of inverse Hessian (when it exists)
- no need to form H^k
- $H^k \nabla \varphi_\gamma(x^k) \leftarrow r$ where r comes from L-BFGS two-loop recursion

$q \leftarrow \nabla \varphi_\gamma(x^k)$
for $i = k-1 : -1 : k-m$
$\alpha_i \leftarrow \rho_i \langle s^i, q \rangle, \; q \leftarrow q - \alpha_i y_i$

$r \leftarrow H_0^k q$
for $i = k-m : 1 : k-1$
$r \leftarrow r + s^i(\alpha_i - \rho_i \langle y^i, r \rangle)$

- requires $4mn$ multiplications (m usually between $3 - 20$)

Black-box Oracle

- evaluating

$$\varphi_\gamma(x) = \min_{z \in \mathbb{R}^n} \left\{ f(x) + \langle \nabla f(x), z - x \rangle + g(z) + \tfrac{1}{2\gamma} \|z - x\|^2 \right\}$$

  requires one call to f, ∇f, g, $\mathrm{prox}_{\gamma g}$ (same as FBS)

- evaluating

$$\nabla \varphi_\gamma(x) = \gamma^{-1}(I - \gamma \nabla^2 f(x))(x - \mathrm{prox}_{\gamma g}(x - \gamma \nabla f(x))$$

  requires an extra hessian-vector product

- **hessian-free evaluation**: only one extra call to ∇f

  forward differences

$$\nabla^2 f(x) d \approx \frac{\nabla f(x + \epsilon d) - \nabla f(x)}{\epsilon}$$

  ✓ prone to cancellation errors

  complex step differentiation

$$\nabla^2 f(x) d \approx \frac{\mathrm{imag}(\nabla f(x + i\epsilon d))}{\epsilon}$$

  ✓ requires ∇f to be analytic

  ✓ $\epsilon = 10^{-8}$– machine precision accuracy

similar complexity per iteration as FBS

(Accelerated) Forward-Backward L-BFGS

- L-BFGS: convergence only under strict conditions (but widely used for non convex)
- *"What we would like to have, is a* **method with rate of convergence** *which is* **guaranteed never to be worse**, *and* **'typically' is much better than the 'optimal' rate."**[2]
- can we have an algorithm with benefits of L-BFGS and strong convergence properties?
- yes. combine L-BFGS with forward-backward steps

[2] Ben-Tal, Nemirovski, Non-euclidean restricted memory level method for large-scale convex optimization [Math. Program., Ser. A 102: 407-456 (2005)]

Algorithm 2 Global Forward-Backward L-BFGS

1: Choose $\gamma \in (0, 1/L_f)$, $x^0 \in \operatorname{dom} g$.
2: Choose w^k such that $\varphi_\gamma(w^k) \leq \varphi_\gamma(x^k)$
3: $x^{k+1} = \operatorname{prox}_{\gamma g}(w^k - \gamma \nabla f(w^k))$
4: $k \leftarrow k + 1$ and go to 2

- in step **??** can use any descent step (L-BFGS, nonlinear CG,. . .)
- minor extra computations: $\varphi_\gamma(w^k)$ entails computation of x^{k+1}
- can adjust dynamically γ when L_f is unknown
- φ bounded level sets: $\{\varphi(x^k)\}$ converges to $\varphi^\star$ with rate $O(1/k)$
- linear convergence for strongly convex φ

Algorithm 3 Accelerated Forward-Backward L-BFGS

1: Choose $\gamma \in (0, 1/L_f)$, $\theta_0 = 1$, $\theta_{k+1} = \frac{\sqrt{\theta_k^4 + 4\theta_k^2} - \theta_k^2}{2}$, $x^0 = v^0 \in \operatorname{dom} g$
2: $\hat{y} = \theta_k v^k + (1 - \theta_k) x^k$
3: $z^k = \operatorname{prox}_{\gamma g}(\hat{y} - \gamma \nabla f(\hat{y}))$
4: $y^k = \begin{cases} \hat{y}, & \text{if } \varphi_\gamma(\hat{y}) \le \varphi_\gamma(w^{k-1}) \\ x^k, & \text{otherwise} \end{cases}$
5: choose w^k such that $\varphi_\gamma(w^k) \le \varphi_\gamma(y^k)$
6: $x^{k+1} = \operatorname{prox}_{\gamma g}(w^k - \gamma \nabla f(w^k))$
7: $v^{k+1} = x^k + \theta_k^{-1}(z^k - x^k)$
8: $k \leftarrow k + 1$ and go to 2

- step **??** enforces monotonicity on FBE
- in step **??** can use any descent step (L-BFGS, nonlinear CG)
- $y^k = \hat{y}$, $w^k = y^k$: algorithm becomes FISTA[3]
- $\{\varphi(x^k)\}$ converges to $\varphi^\star$ with the **optimal rate** $O(1/k^2)$

[3] Beck, Amir, and Marc Teboulle. "A fast iterative shrinkage-thresholding algorithm for linear inverse problems." SIAM Journal on Imaging Sciences 2.1 (2009): 183-202.

Outline

FBS and forward-backward envelope (FBE)

Forward-Backward L-BFGS and acceleration

- AMM and alternating minimization L-BFGS

ForBES (Forward Backward Envelope Solver)

conclusions

Separable convex problems

$$\begin{aligned} \text{minimize} \quad & f(x) + g(z) \\ \text{subject to} \quad & Ax + Bz = b \end{aligned}$$

- f is **strongly convex** with convexity parameter μ_f
- f, g are nice, e.g. separable. coupling introduced through constraints

Alternating Minimization Method (AMM): FBS applied to the dual

$$\begin{aligned} x^{k+1} &= \arg\min_x \; L_0(x, z^k, y^k) \\ z^{k+1} &= \arg\min_z \; L_\gamma(x^{k+1}, z, y^k) \qquad\qquad \gamma \in (0, 2\mu_f/\|A\|^2) \\ y^{k+1} &= y^k + \gamma(Ax^{k+1} + Bz^{k+1} - b) \end{aligned}$$

Augmented Lagrangian

$$L_\gamma(x, z, \lambda) = f(x) + g(z) + \langle y, Ax + Bz - b\rangle + \tfrac{\gamma}{2}\|Ax + Bz - b\|^2$$

Alternating Minimization Envelope (AME)

- dual problem

$$\text{maximize} \;\; \psi(y) = -\left(f^\star(-A^\top y) + b^\top y + g^\star(-B^\top y) \right)$$

where $f^\star(v) = \sup_{x \in \mathbb{R}^n} \{\langle v, x \rangle - f(x)\}$ is the conjugate of f

- **AME: FBE for dual problem is augmented Lagrangian**

$$\psi_\gamma(y) = L_\gamma(x(y), z(y), y)$$

where $x(y)$, $z(y)$ are the **AMM updates**

$$x(y) = \arg\min_x \;\; f(x) + \langle y, Ax \rangle$$
$$z(y) = \arg\min_z \;\; g(z) + \langle y, Bz \rangle + \tfrac{\gamma}{2} \|Ax(y) + Bz - b\|^2$$

Connection between AMM and AME

dual problem is equivalent to

$$\underset{y}{\text{maximize}}\ \psi_\gamma(y) = L_\gamma(x(y), z(y), y) \qquad \gamma \in \left(0, \mu_f / \|A\|^2\right)$$

- f is C^2 on $\operatorname{int}\operatorname{dom} f \implies \psi_\gamma$ is C^1

$$\nabla\psi_\gamma(y) = \overbrace{\left(I - \gamma A \nabla^2 f(x(y))^{-1} A^\top\right)}^{D(y)}(Ax(y) + Bz(y) - b)$$

- **AMM** $=$ **variable metric gradient method** on **AME**

$$y^{k+1} = y^k + \gamma D(y^k)^{-1} \nabla\psi_\gamma(y^k)$$

- **Alternating minimization L-BFGS**: Forward-Backward L-BFGS applied to the dual

Outline

FBS and forward-backward envelope (FBE)

Forward-Backward L-BFGS and acceleration

AMM and alternating minimization L-BFGS

- **ForBES** (Forward Backward Envelope Solver)

conclusions

ForBES

- **For**ward-**B**ackward **E**nvelope **S**olver
- MATLAB solver for convex composite and separable problems
- **generic** and **efficient**
- implements several algorithms for minimizing the FBE
  - ✓ L-BFGS
  - ✓ nonlinear CG methods
  - ✓ Barzilai-Borwein
  - ✓ global and accelerated versions
- on Github

`http://lostella.github.io/ForBES/`

ForBES

composite problems

$$\text{minimize} \;\; f(Cx + d) + g(z)$$

- f convex, strongly smooth and $\mathcal{C}^2$
- g proper, closed, convex function with computable prox

separable problems

$$\begin{aligned} &\text{minimize } f(x) + g(z) \\ &\text{subject to } Ax + Bz = b \end{aligned}$$

- f strongly convex and $\mathcal{C}^2$ in the interior of its domain
- g any proper, closed, convex function with computable prox
- B is a tight frame

$$B^\top B = \alpha I \qquad \alpha > 0$$

ForBES

composite problems

$$\text{minimize} \;\; f(Cx + d) + g(z)$$

- `out = forbes(f, g, init, aff);`
- can have the sum of multiple f's and g's
- `aff = {C, d};`

separable problems

$$\begin{aligned} &\text{minimize } f(x) + g(z) \\ &\text{subject to } Ax + Bz = b \end{aligned}$$

- `out = forbes(f, g, init, [], constr);`
- can have the sum of multiple f's and g's
- `constr = {A, B, b};`

Example: Lasso

$$\text{minimize} \quad \underbrace{\tfrac{1}{2}\|Ax-b\|_2^2}_{f(Cx+d)} + \underbrace{\lambda\|x\|_1}_{g(x)}$$

- regularization term enforces sparsity in the solution $x_\star$
- used for regression and feature selection
- denser solution, harder problem as $\lambda \to 0$

proximal mapping of ℓ_1-norm: soft thresholding

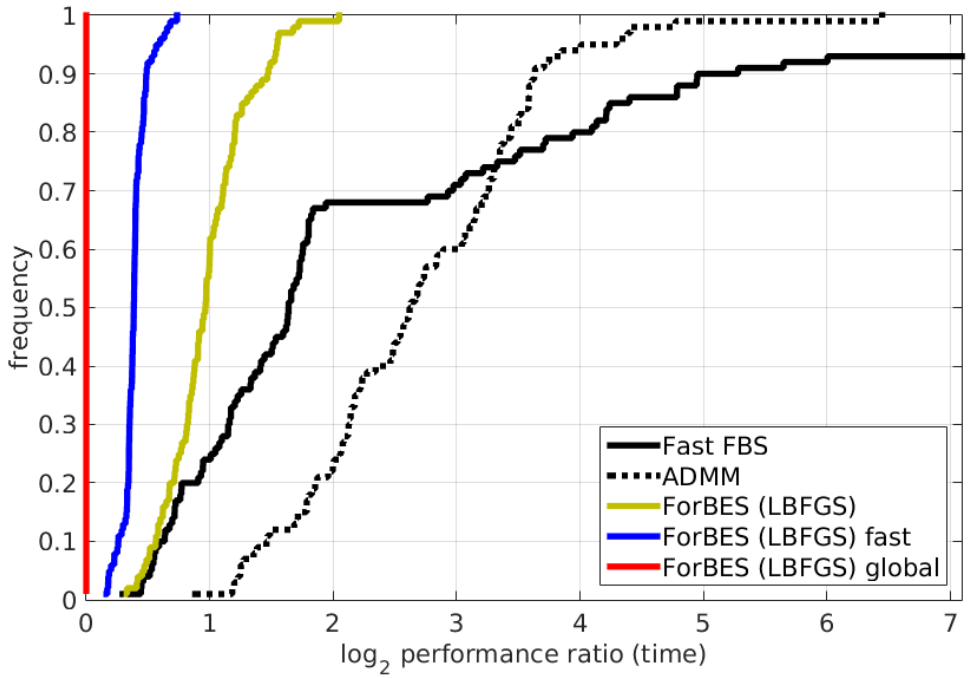
$$\text{prox}_{\gamma\|\cdot\|_1}(x) = \text{sign}(x) \cdot \max\{0, |x| - \gamma\}$$

```
out = forbes(quadLoss(), l1Norm(lam), x0, {A, -b});
 out =
        message: 'reached optimum (up to tolerance)'
           flag: 0
              x: [5000x1 double]
     iterations: 352
             [..]
```

Example: Lasso

performance profile on HDR problems taken from the SPEAR dataset.

- HDR: high dynamic range $\Longrightarrow$ large ratio $\max x_\star / \min x_\star$
- 274 problems, size ranging from 512×1024 to 8192×49152



- point (x, y) of graph for solver s gives percentage of problems y on which solver's running time was within 2^x times the best running time

dataset at `wwwopt.mathematik.tu-darmstadt.de/spear`

Example: Lasso

Compute regularization path warm-starting each problem

```
x0 = zeros(n, 1);
lambdas = lambda_max*logspace(-2, 0, 10);
f = quadLoss();
for i = length(lambdas):-1:1 % start from the larger
    out = forbes(f, l1Norm(lam(i)), x0, {A, -b});
    x0 = out.x;
end
```

- random example, 3000 observations, 500K features, 7.5M nnzs

	iter.	matvecs	backsolves	time
Fast FBS	22406	44812	-	1054s
ADMM	2214	4428	2214	261s
ForBES (LBFGS)	1050	6260	-	196s

Example: Sparse robust regression

$$\text{minimize} \sum_{i=1}^{m} h_\delta(\langle a_i, x\rangle - b_i) + \lambda\|x\|_1, \quad h_\delta(r) = \begin{cases} \frac{1}{2\delta}r^2 & \text{if } |r| \le \delta \\ |r| - \frac{\delta}{2} & \text{otherwise} \end{cases}$$

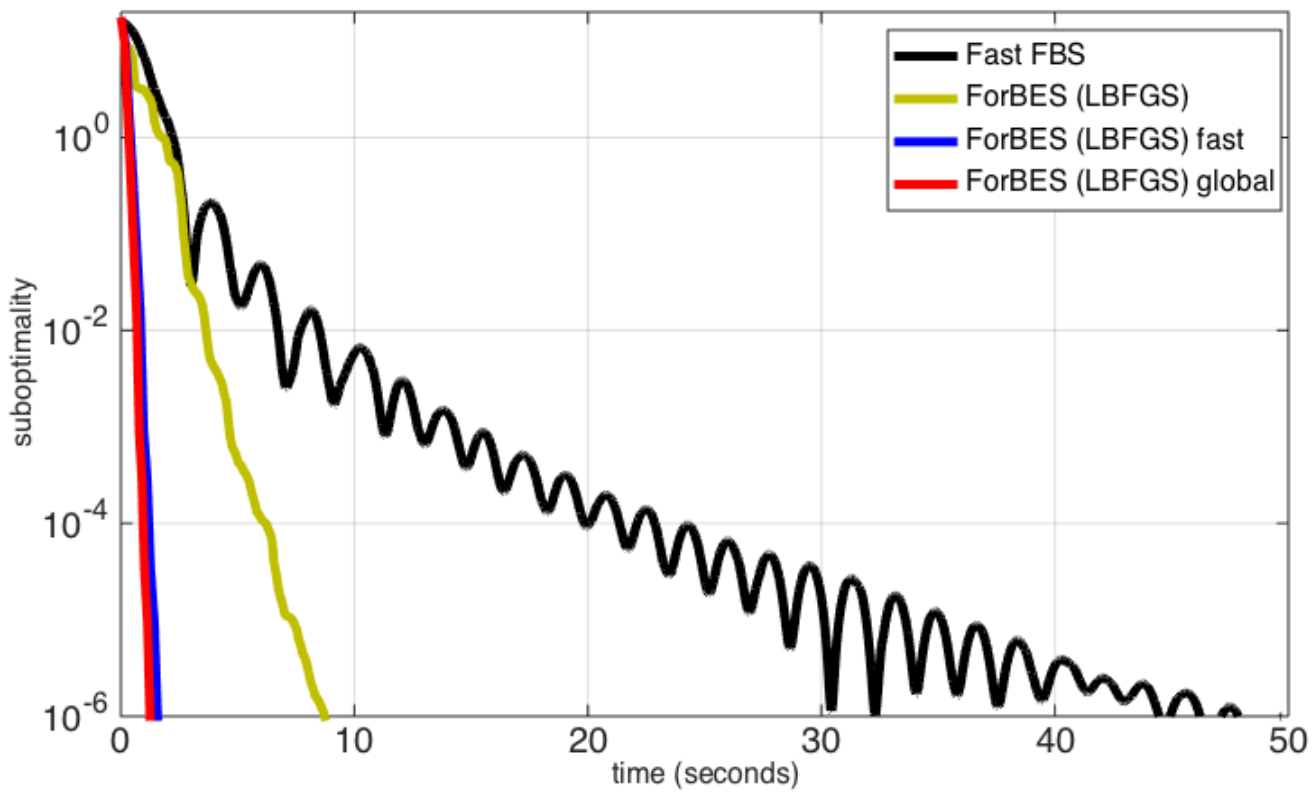
- Huber loss is Moreau envelope (with parameter δ) of $|\cdot|$ (not $\mathcal{C}^2$)
- quadratic penalty for small residuals, linear penalty for large ones
- useful when data contains outliers

$$A = \begin{bmatrix} a_1, & \ldots, & a_m \end{bmatrix}^\top, \quad b = \begin{bmatrix} b_1, & \ldots, & b_m \end{bmatrix}^\top$$

```
f = huberLoss(del);
g = l1Norm(lam);
out = forbes(f, g, zeros(n,1), {A, -b});
```

Example: Sparse robust regression

- random problem: 5000 samples, 200K features, 1M nonzeros
- artificial outliers in $b = (b_{\text{in}}, b_{\text{out}})$ (added large noise to b_{out})



$$\text{nnz}(x_\star) = 92, \|(Ax_\star - b)_{\text{in}}\|/\|b_{\text{in}}\| = 0.16$$

$$\text{with Lasso: } \text{nnz}(x_\star) = 93, \|(Ax_\star - b)_{\text{in}}\|/\|b_{\text{in}}\| = 0.58$$

Example: Sparse logistic regression

$$\text{minimize } \underbrace{(1/m)\sum_{i=1}^{m}\log(1+\exp(-y_i\langle a_i, x\rangle))}_{f(Cx+d)} + \underbrace{\lambda\|x\|_1}_{g(x)}$$

- labels $y_i \in \{-1, 1\}$, $i = 1, \ldots, m$
- used for classification and feature selection
- again, as $\lambda \to 0$ denser solution, harder problem

```
f = logLoss(1/m);
C = diag(sparse(y))*A;
g = l1Norm(lam);
out = forbes(f, g, zeros(n, 1), C);
```

Example: Sparse logistic regression

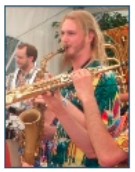
Dataset	Features	nnz	Fast FBS	ForBES	**Speedup**
w4a	299	86K	3.54s	1.01s	**x3.5**
w5a	299	115K	3.73s	1.08s	**x3.5**
w6a	300	200K	5.73s	1.99s	**x2.9**
w7a	300	288K	7.97s	2.15s	**x3.7**
w8a	300	580K	15.85s	4.17s	**x3.8**
rcv1	45K	910K	15.94s	1.63s	**x9.8**
real-sim	21K	1.5M	24.39s	3.16s	**x7.7**
news20	1.4M	3.7M	427.77s	53.55s	**x8.0**

Datasets taken from LIBSVM:

`www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

Example: Matrix completion

$$\text{minimize} \;\; \tfrac{1}{2}\|\mathcal{A}(X) - b\|_2^2 \;+\; \lambda\|X\|_\star$$



user j

$$X = \begin{pmatrix} ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & 4 & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \\ ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? & ? \end{pmatrix}$$

item i

- $\|X\|_\star = \sum \sigma(X)$ forces $X_\star$ to have low rank
- $\mathcal{A} : \mathbb{R}^{m\times n} \to \mathbb{R}^k$ is linear. example: $\mathcal{A}$ selects k entries from X
- proximal mapping of $\|\cdot\|_\star$ is *singular value soft-thresholding*

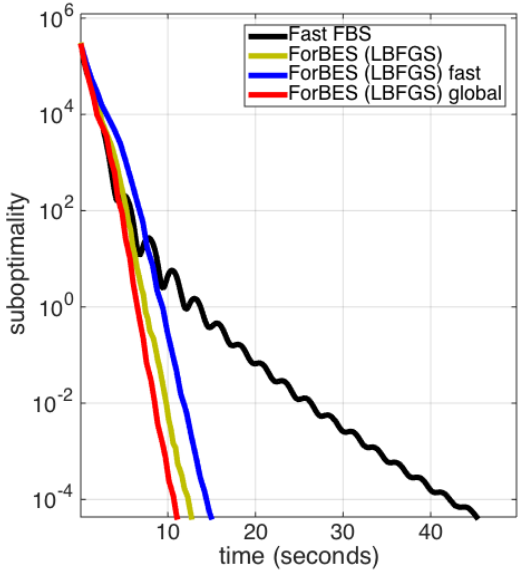
$$\text{prox}_{\gamma\|\cdot\|_\star}(X) = U\,\text{diag}(\max\{0, \sigma_i - \gamma\}, i = 1, \ldots, r)V^T$$

where $\sigma_1, \ldots, \sigma_r$ are the singular values of X

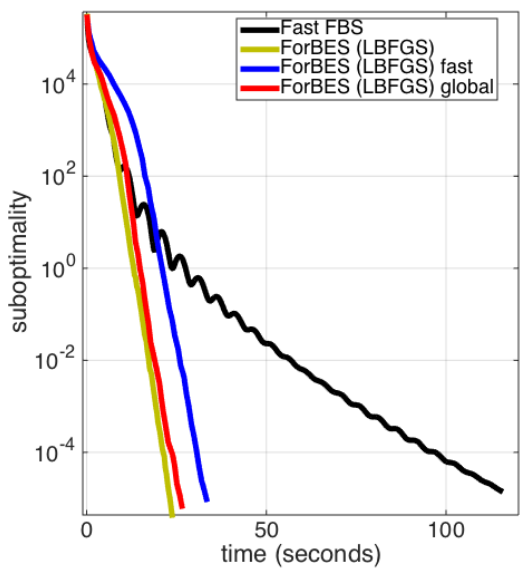
Example: Matrix completion

```
% P binary matrix, B contains observations
f = quadLoss(P(:), B(:));
g = nuclearNorm(m, n, lam);
out = forbes(f, g, zeros(m*n, 1));
```

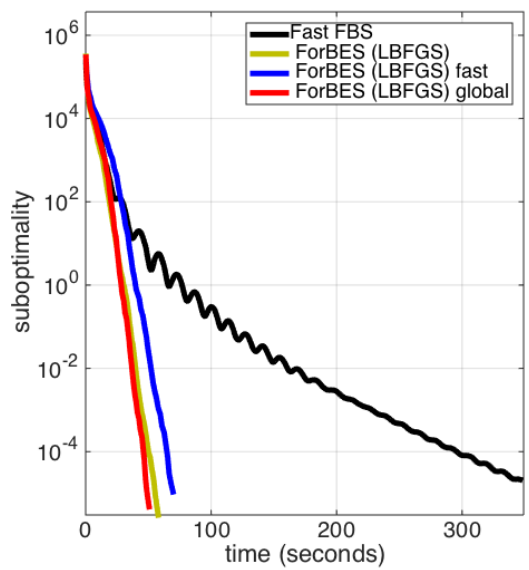
MovieLens 100k dataset: 943×1682, **1.5M** unknowns



$\lambda = 50$
$\text{rank}(X_\star) = 3$



$\lambda = 30$
$\text{rank}(X_\star) = 8$



$\lambda = 20$
$\text{rank}(X_\star) = 38$

Frequency Domain Subspace identification[4]

$$\text{minimize } \tfrac{1}{2}\|x-b\|_2^2 \;+\; \lambda\|Z\|_\star, \text{ subject to } \mathcal{H}(x)=Z$$

- $\mathcal{H}$ is a linear map returning the (block) Hankel matrix

$$\mathcal{H}(x_1,x_2,\ldots,x_n)=\begin{pmatrix} x_1 & x_2 & x_3 & \cdots & x_q \\ x_2 & x_3 & \cdots & & x_{q+1} \\ \vdots & \vdots & & & \vdots \\ x_p & x_{p+1} & \cdots & & x_{p+q-1} \end{pmatrix} \in \mathbb{R}^{pn_y \times qn_u}$$

- b_i is IDFT of a noisy frequency sample for a MIMO, LTI system

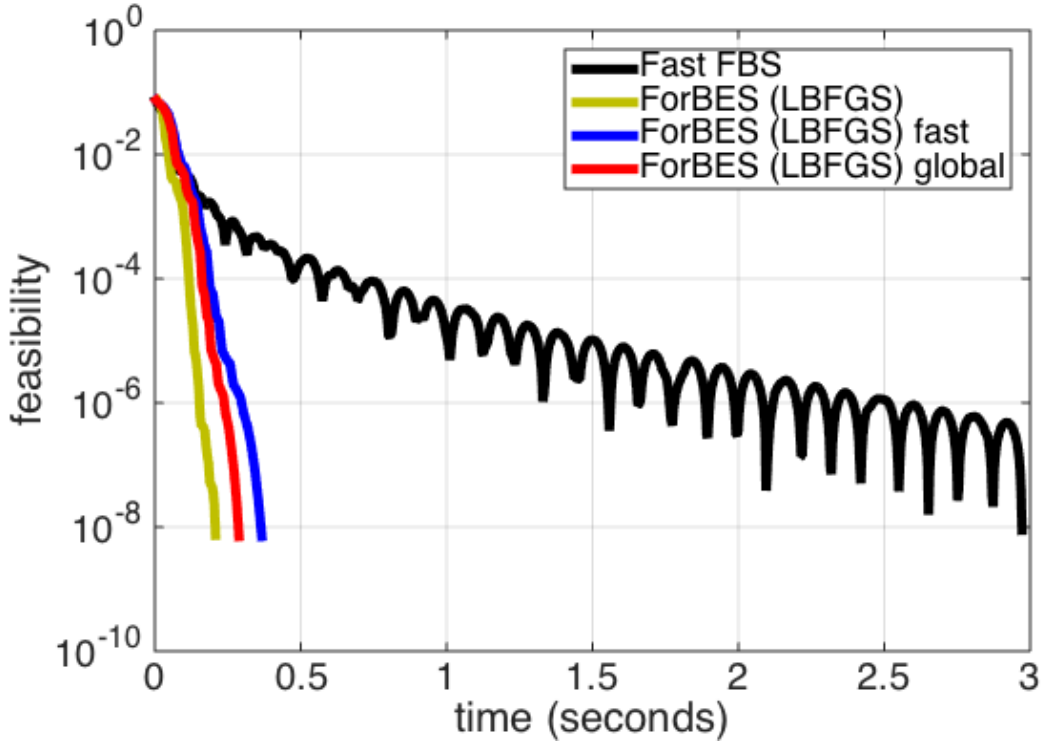
$$b_i = \tfrac{1}{2M}\sum_{k=0}^{2M-1} G_k e^{j2\pi ik/2M}, \quad i=0,\ldots,2M-1$$

$$G_k = C(e^{j\omega_k}I-A)^{-1}B+D, \quad \omega_k = k\pi/M, \quad k=0,\ldots,M$$

```
f = quadLoss(1, b);
g = nuclearNorm(lam);
A = blockHankel(p, q, ny, nu);
b = zeros(p*q, 1);
out = forbes(f, g, zeros((p*ny)*(q*nu), 1), [], {A, -1,
```

[4] Graf Plessen, Semeraro, Wood and Smith, *Optimization Algorithms for Nuclear Norm Based Subspace Identification with Uniformly Spaced Frequency Domain Data, ACC, 2015*

Frequency Domain Subspace identification



More efficient use of operations

	iterations	Hankel	SVD
Fast FBS	850	1700	850
ForBES (LBFGS)	39	154	43
ForBES (LBFGS) fast	30	234	95
ForBES (LBFGS) global	30	176	66

Example: Support vector machines

$$\text{minimize } \underbrace{\tfrac{\lambda}{2}\|x\|_2^2}_{f(x)} + \underbrace{\textstyle\sum_{i=1}^m \max\{0, 1 - b_i z_i\}}_{g(z)}, \text{ subject to } Ax = z$$

- used for classification, nonsmooth term is the *hinge* loss
- $A = (a_1, \ldots, a_m)^T$ matrix of features, $b \in \{-1, 1\}^m$ vector of labels
- $g(z) = \sum_{i=1}^m g_i(z_i)$ and

$$\text{prox}_{\gamma g_i}(z_i) = \begin{cases} b_i \min\{1, b_i z_i + \gamma\} & \text{if } b_i z_i < 1 \\ z_i & \text{otherwise} \end{cases}$$

```
f = quadLoss(lam);
g = hingeLoss(1, b);
constr = {A, -1, zeros(m,1)};
out = forbes(f, g, zeros(m, 1), [], constr);
```

Example: Support vector machines

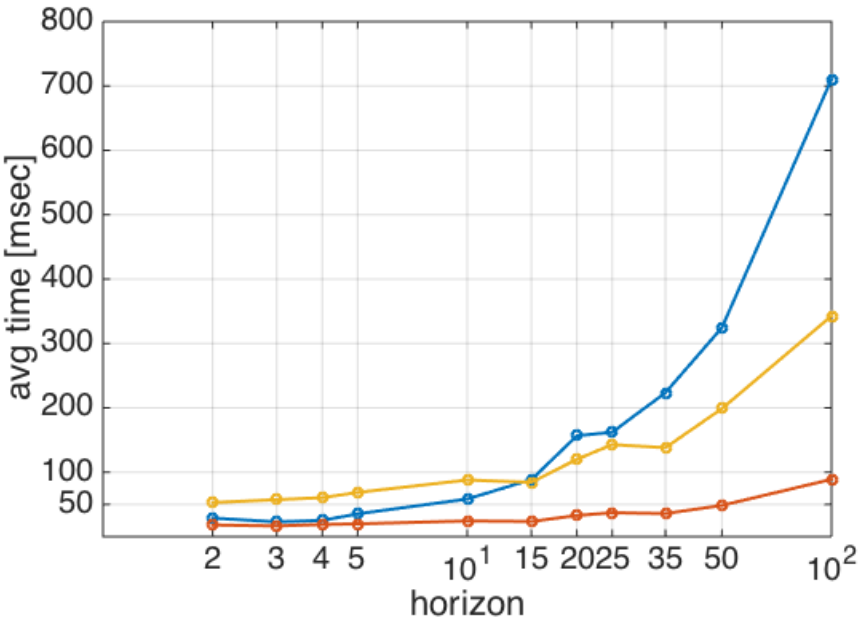
Dataset	Features	nnz	Fast AMM	ForBES	**Speedup**
w4a	299	86K	16.51s	1.44s	**x11.5**
w5a	299	115K	21.88s	2.69s	**x8.1**
w6a	300	200K	55.52s	4.61s	**x12.0**
w7a	300	288K	96.68s	7.29s	**x13.3**
w8a	300	580K	273.89s	20.61s	**x13.3**
rcv1	45K	910K	617.71s	96.47s	**x6.4**
real-sim	21K	1.5M	39.73s	14.38s	**x2.8**
news20	1.4M	3.7M	148.01s	18.73s	**x7.9**

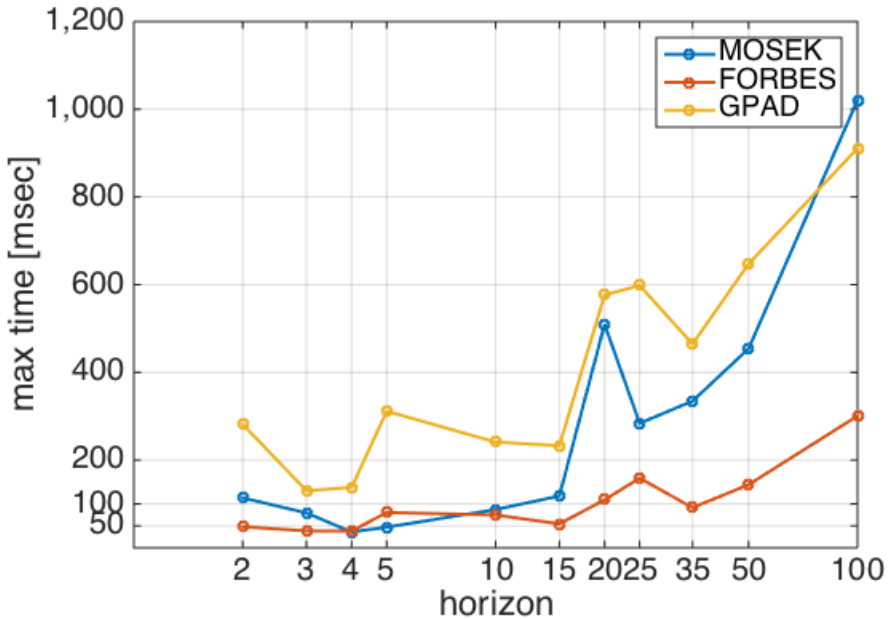
Datasets taken from LIBSVM:

`www.csie.ntu.edu.tw/~cjlin/libsvmtools/datasets/`

Example: MPC

- Random problems with 30 states, 10 inputs, increasing horizon N
- Box constraints, weight matrices with $\kappa(Q) \approx \kappa(R) \approx \kappa(Q_f) \approx 10^2$
- For every N, average and max time for 50 random x^0





- MOSEK: commercial QP solver (interior point)
- GPAD: accelerated dual proximal gradient

Summary

- proximal minimization = gradient method on Moreau envelope (70's)
- **FBS** = (variable metric) gradient method on **FBE** (this talk)
- Forward-Backward L-BFGS for nonsmooth problems
- Optimal complexity estimates, much faster convergence in practice
- **ForBES**: Forward-Backward Envelope solver

Sources

based on

1. Patrinos and Bemporad (2013) *Proximal Newton methods for convex composite optimization*. 52nd IEEE CDC, Florence, Italy
2. Patrinos, Stella and Bemporad (2014), *Forward-backward truncated Newton methods for convex composite optimization*, `arXiv:1402.6655`
3. Patrinos and Stella (2014), *Forward-backward L-BFGS for large-scale convex composite optimization*, working paper

ForBES code available on github

`http://lostella.github.io/ForBES/`