Global rates of convergence of algorithms for nonconvex smooth optimization

Coralia Cartis (University of Oxford)

joint with

Nick Gould (RAL, UK) & **Philippe Toint** (Namur, Belgium)

Katya Scheinberg (Lehigh, USA)

ICML Workshop on Optimization Methods for the Next Generation of Machine Learning
ICML New York City, June 23–24, 2016

Unconstrained optimization — a “mature” area?

Nonconvex local unconstrained optimization:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) \quad \text{where} \quad f \in C^1(\mathbb{R}^n) \text{ or } C^2(\mathbb{R}^n).$$

Currently two main competing methodologies:

- Linesearch methods
- Trust-region methods

to globalize gradient and (approximate) Newton steps.

Much reliable, efficient software for (large-scale) problems.

- Is there anything more to say?...

Unconstrained optimization — a "mature" area?

Nonconvex local unconstrained optimization:

$$\underset{x \in \mathbb{R}^n}{\text{minimize}} \quad f(x) \quad \text{where} \quad f \in C^1(\mathbb{R}^n) \text{ or } C^2(\mathbb{R}^n).$$

Currently two main competing methodologies:

- Linesearch methods
- Trust-region methods

to globalize gradient and (approximate) Newton steps.

Much reliable, efficient software for (large-scale) problems.

- Is there anything more to say?...
  - Global rates of convergence of optimization algorithms

$\Longleftrightarrow$ Evaluation complexity of methods (from any initial guess)

[well-studied for convex problems, but unprecedented for nonconvex until recently]

Evaluation complexity of unconstrained optimization

Relevant analyses of iterative optimization algorithms:

- Global convergence to first/second-order critical points (from any initial guess)
- Local convergence and local rates (sufficiently close initial guess, well-behaved minimizer)

  [Newton's method: Q-quadratic; steepest descent: linear]

- Global rates of convergence (from any initial guess)
  $\Longleftrightarrow$ Worst-case function evaluation complexity
  - evaluations are often expensive in practice (climate modelling, molecular simulations, etc)
  - black-box/oracle computational model (suitable for the different 'shapes and sizes' of nonlinear problems)

    [Nemirovskii & Yudin ('83); Vavasis ('92), Sikorski ('01), Nesterov ('04)]

Overview

- Evaluation complexity of standard methods
- Improved complexity for cubic regularization
- Regularization - and other methods - with only occasionally accurate information

Global efficiency of steepest-descent methods

Steepest descent method (with linesearch or trust-region):

- $f \in \mathcal{C}^1(\mathbb{R}^n)$ with Lipschitz continuous gradient.
- to generate gradient $\|g(x_k)\| \leq \epsilon$, requires at most

[Nesterov ('04); Gratton, Sartenaer & Toint ('08)]

$$\left\lceil \kappa_{\mathbf{sd}} \cdot \mathbf{Lips}_g \cdot (f(x_0) - f_{\mathbf{low}}) \cdot \epsilon^{-2} \right\rceil \ \textbf{function evaluations.}$$

Global efficiency of steepest-descent methods

Steepest descent method (with linesearch or trust-region):

- $f \in \mathcal{C}^1(\mathbb{R}^n)$ with Lipschitz continuous gradient.
- to generate gradient $\|g(x_k)\| \leq \epsilon$, requires at most

[Nesterov ('04); Gratton, Sartenaer & Toint ('08)]

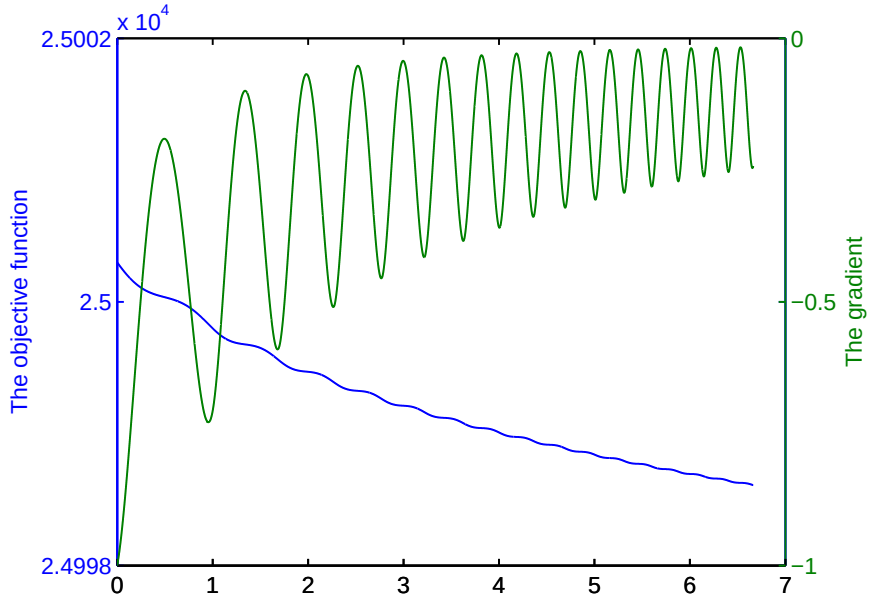
$$\lceil \kappa_{\mathrm{sd}} \cdot \mathbf{Lips}_g \cdot (f(x_0) - f_{\mathrm{low}}) \cdot \epsilon^{-2} \rceil \ \textbf{function evaluations.}$$

The worst-case bound is sharp for steepest descent: [CGT('10)]

For any $\epsilon > 0$ and $\tau > 0$, (inexact-linesearch) steepest descent applied to this f takes precisely

$$\lceil \epsilon^{-2+\tau} \rceil \ \textbf{function evaluations}$$

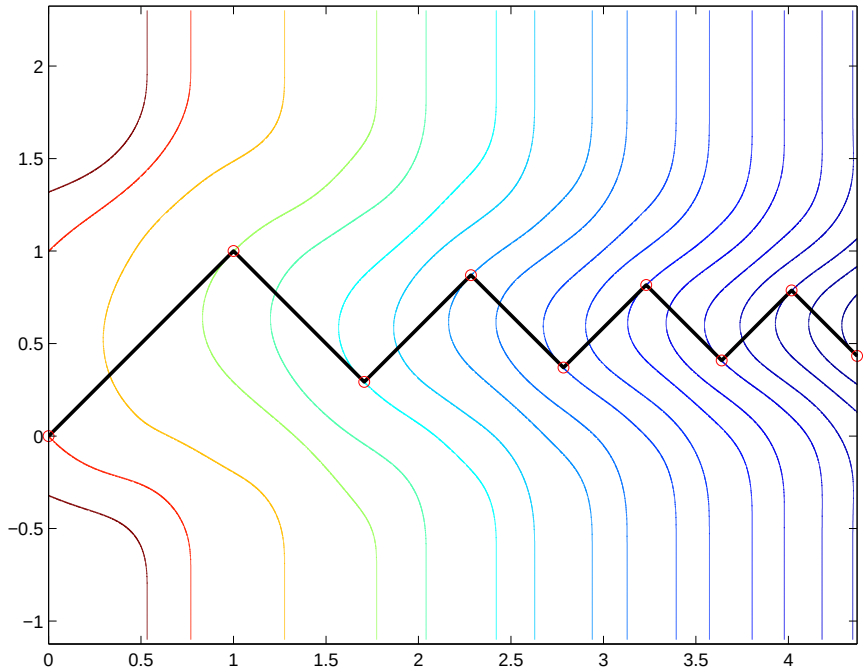
to generate $|g(x_k)| \leq \epsilon$.



Worst-case bound is sharp for steepest descent

Steepest descent method with exact linesearch

- $x_{k+1} = x_k - \alpha_k g(x_k)$ with $\alpha_k = \arg\min_{\alpha \geq 0} f(x_k - \alpha g(x_k))$
- takes $\lceil \epsilon^{-2+\tau} \rceil$ iterations to generate $\|g(x_k)\| \leq \epsilon$



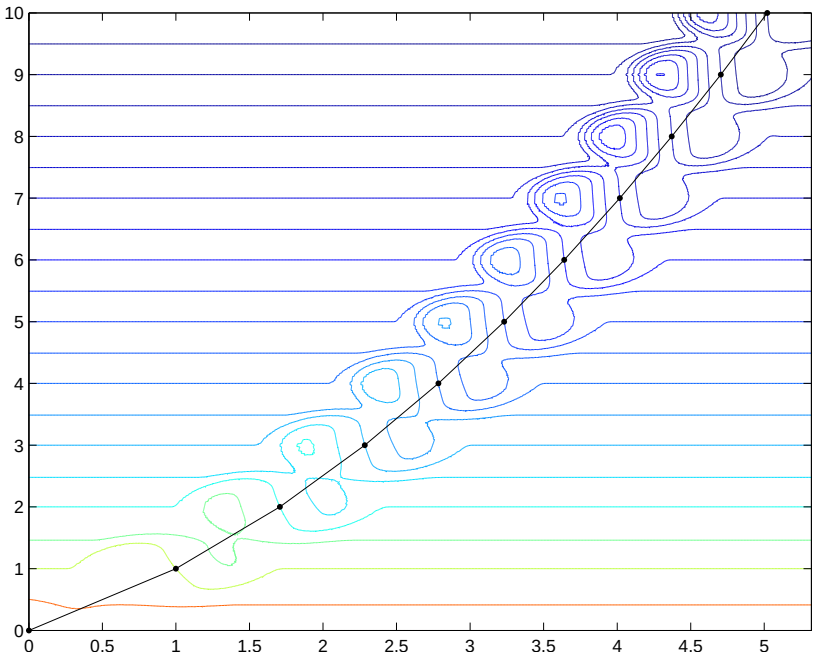
Contour lines of $f(x_1, x_2)$ and path of iterates.

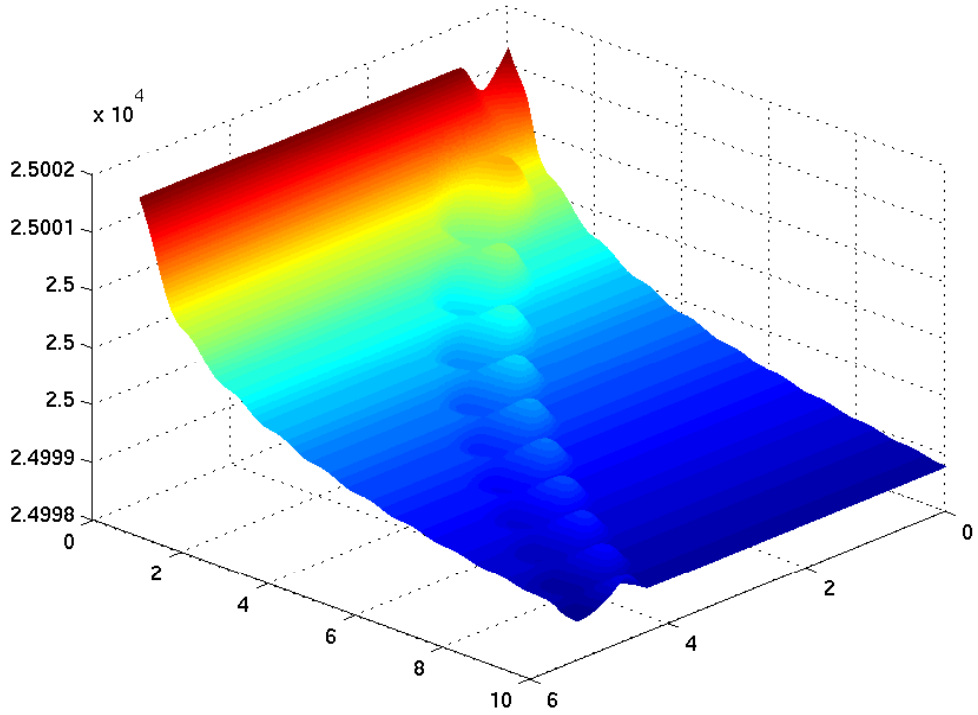
Global efficiency of Newton's method

- Newton's method: $x_{k+1} = x_k - H_k^{-1} g_k$ with $H_k \succ 0$.

Newton's method: as slow as steepest descent

- may require $\lceil \epsilon^{-2+\tau} \rceil$ evaluations/iterations, same as steepest descent method





Globally Lipschitz continuous gradient and Hessian

Worst-case bound for Newton's method

- when globalized with trust-region or linesearch, Newton's method will take at most

$$\lceil \kappa_N \epsilon^{-2} \rceil$$

evaluations to generate $\|g_k\| \leq \epsilon$

- similar worst-case complexity for classical trust-region and linesearch methods

Is there any method with better evaluation complexity than steepest-descent?

Improved complexity for cubic regularization

Improved complexity for cubic regularization

A cubic model: [Griewank ('81, TR), Nesterov & Polyak ('06), Weiser et al ('07)]

H is globally Lipschitz continuous with Lipschitz constant 2σ:

Taylor, Cauchy-Schwarz and Lipschitz $\Longrightarrow$

$$f(x_k + s) \quad \leq \quad \underbrace{f(x_k) + s^T g(x_k) + \tfrac{1}{2} s^T H(x_k) s + \tfrac{1}{3}\sigma \|s\|_2^3}_{m_k(s)}$$

$\Longrightarrow$ reducing m_k from $s = 0$ decreases f since $m_k(0) = f(x_k)$.

Cubic regularization method: [Nesterov & Polyak ('06)]

- $x_{k+1} = x_k + s_k$
- compute $s_k \longrightarrow \mathbf{min}_s\, m_k(s)$ globally: [tractable, even if m_k nonconvex!]

Improved complexity for cubic regularization

A cubic model: [Griewank ('81, TR), Nesterov & Polyak ('06), Weiser et al ('07)]

H is globally Lipschitz continuous with Lipschitz constant 2σ:

Taylor, Cauchy-Schwarz and Lipschitz $\Longrightarrow$

$$f(x_k + s) \leq \underbrace{f(x_k) + s^T g(x_k) + \tfrac{1}{2} s^T H(x_k) s + \tfrac{1}{3}\sigma \|s\|_2^3}_{m_k(s)}$$

$\Longrightarrow$ reducing m_k from $s = 0$ decreases f since $m_k(0) = f(x_k)$.

Cubic regularization method: [Nesterov & Polyak ('06)]

- $x_{k+1} = x_k + s_k$
- compute $s_k \longrightarrow \mathbf{min}_s\, m_k(s)$ globally: [tractable, even if m_k nonconvex!]

Worst-case evaluation complexity: at most $\lceil \kappa_{\mathbf{cr}} \cdot \epsilon^{-3/2} \rceil$ function evaluations to ensure $\|g(x_k)\| \leq \epsilon$. [Nesterov & Polyak ('06)]

Can we make cubic regularization computationally efficient ?

Adaptive cubic regularization – a practical method

Use [C, Gould & Toint (CGT): Math Programming (2011)]

- cubic regularization model at x_k

$$m_k(s) \equiv f(x_k) + s^T g(x_k) + \tfrac{1}{2} s^T B_k s + \tfrac{1}{3} \sigma_k \|s\|^3$$

  - $\sigma_k > 0$ is the iteration-dependent **regularization weight**
  - B_k is an approximate Hessian

Adaptive cubic regularization – a practical method

Use [C, Gould & Toint (CGT): Math Programming (2011)]

- cubic regularization model at x_k

$$m_k(s) \equiv f(x_k) + s^T g(x_k) + \tfrac{1}{2} s^T B_k s + \tfrac{1}{3}\sigma_k \|s\|^3$$

  - $\sigma_k > 0$ is the iteration-dependent **regularization weight**
  - B_k is an approximate Hessian

- compute $s_k \approx \arg\min_s m_k(s)$ [details to follow]

- compute $\rho_k = \dfrac{f(x_k) - f(x_k + s_k)}{f(x_k) - m_k(s_k)}$

- set $x_{k+1} = \begin{cases} x_k + s_k & \text{if } \rho_k > \eta = 0.1 \\ x_k & \text{otherwise} \end{cases}$

- $\sigma_{k+1} = \sigma_k/\gamma = 2\sigma_k$ when $\rho_k < \eta$; else $\sigma_{k+1} = \max\{\gamma\sigma_k, \sigma_{\min}\}$

Adaptive Regularization with Cubics (ARC)

ARC: s_k =global min of $m_k(s)$ over $s \in \mathcal{S} \leq \mathbb{R}^n$, with $g \in \mathcal{S}$

$\longrightarrow$ increase subspaces to satisfy termination criteria:

$$\|\nabla_s m_k(s_k)\| \leq \min(1, \|s_k\|)\|g_k\|$$

ARC has excellent convergence properties: globally, to second-order critical points and locally, Q-quadratically.

Adaptive Regularization with Cubics (ARC)

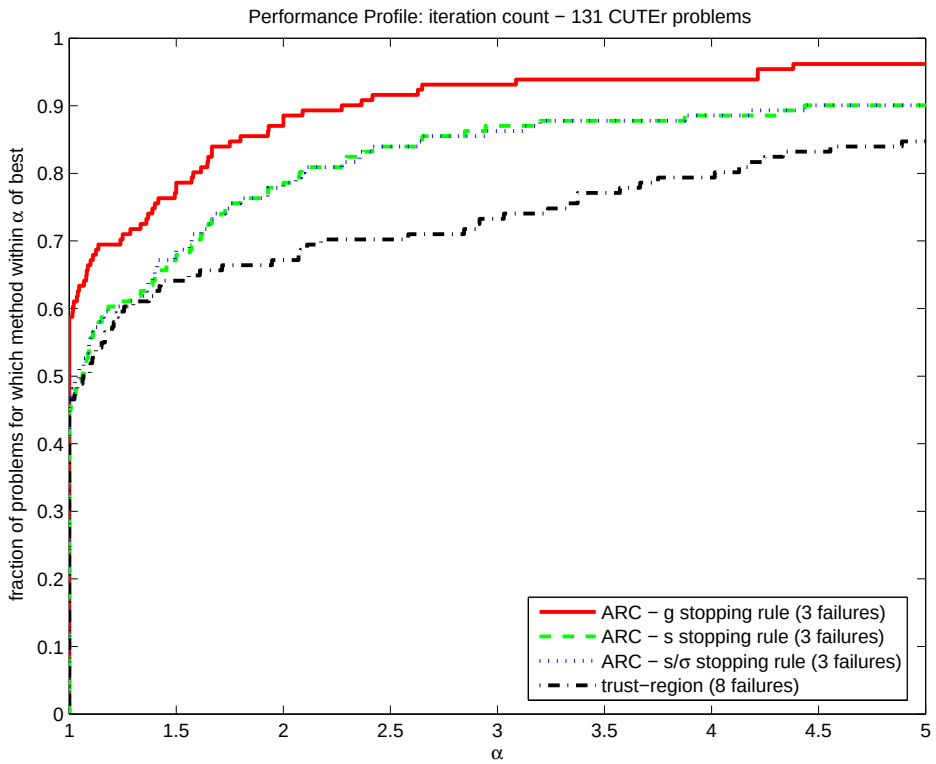
ARC: s_k =global min of $m_k(s)$ over $s \in \mathcal{S} \leq \mathbb{R}^n$, with $g \in \mathcal{S}$

$\longrightarrow$ increase subspaces to satisfy termination criteria:

$$\|\nabla_s m_k(s_k)\| \leq \min(1, \|s_k\|)\|g_k\|$$

ARC has excellent convergence properties: globally, to second-order critical points and locally, Q-quadratically.

'Average-case' performance of ARC variants (preliminary numerics)



Worst-case performance of ARC

If H is Lipschitz continuous on iterates' path and $\|(B_k - H_k)s_k\| = O(\|s_k\|^2)^{(*)}$, then ARC requires at most

$$\left\lceil \kappa_{\mathbf{arc}} \cdot \mathbf{L_H}^{\frac{3}{2}} \cdot (f(x_0) - f_{\mathbf{low}}) \cdot \epsilon^{-\frac{3}{2}} \right\rceil \ \textbf{function evaluations}$$

to ensure $\|g_k\| \leq \epsilon$. [cf. Nesterov & Polyak]

$(*)$ achievable when $B_k = H_k$ or when B_k is computed by gradient finite differences

Worst-case performance of ARC

If H is Lipschitz continuous on iterates' path and $\|(B_k - H_k)s_k\| = O(\|s_k\|^2)^{(*)}$, then ARC requires at most

$$\left\lceil \kappa_{\mathbf{arc}} \cdot \mathbf{L_H}^{\frac{3}{2}} \cdot (f(x_0) - f_{\mathbf{low}}) \cdot \epsilon^{-\frac{3}{2}} \right\rceil \ \textbf{function evaluations}$$

to ensure $\|g_k\| \leq \epsilon$. [cf. Nesterov & Polyak]

$(*)$ achievable when $B_k = H_k$ or when B_k is computed by gradient finite differences

Key ingredients:

- sufficient function decrease: $f(x_k) - f(x_{k+1}) \geq \frac{\eta_1}{6}\sigma_k\|s_k\|^3$

  [local, approximate model minimization is sufficient here]

- long successful steps: $\|s_k\| \geq C\|g_{k+1}\|^{\frac{1}{2}}$ (and $\sigma_k \geq \sigma_{\mathbf{min}} > 0$)

$\Longrightarrow$ while $\|g_k\| \geq \epsilon$ and k successful,

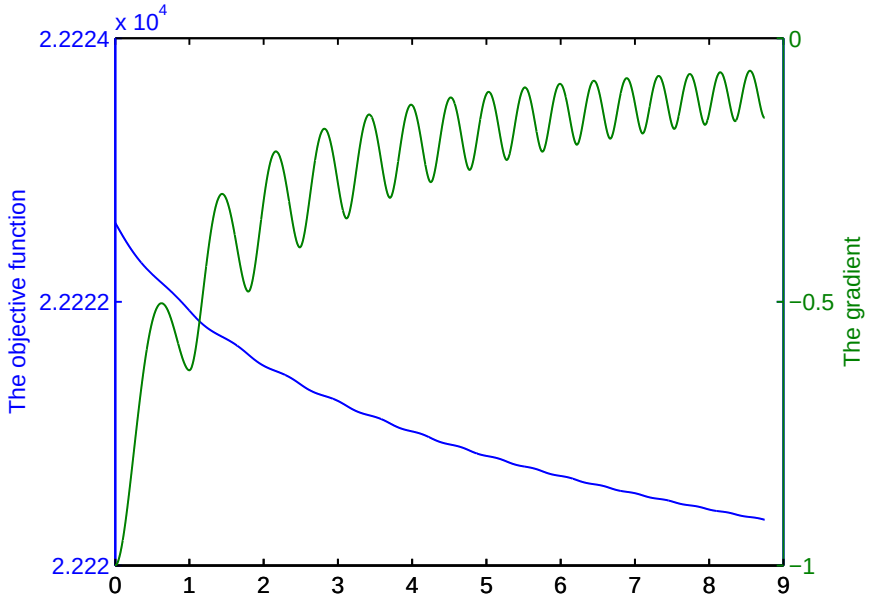
$$f(x_k) - f(x_{k+1}) \geq \frac{\eta_1}{6}\sigma_{\mathbf{min}} C \cdot \epsilon^{\frac{3}{2}}$$

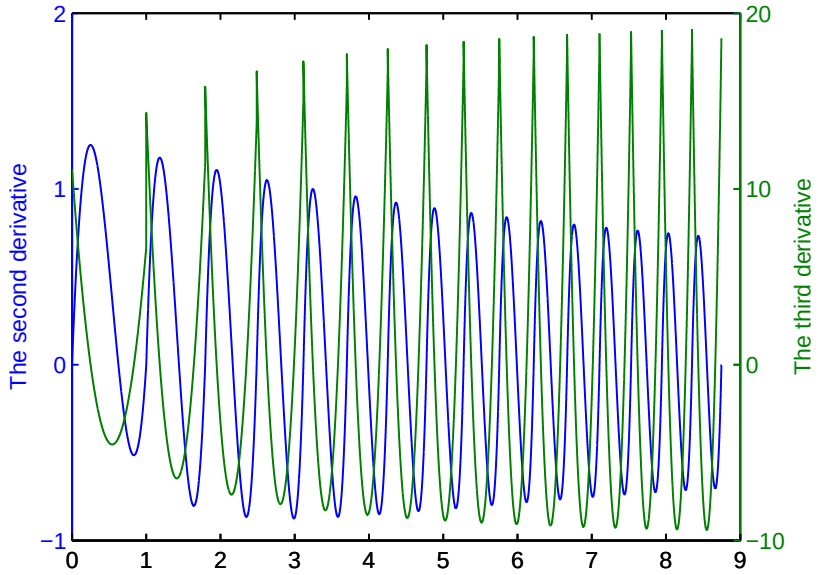
summing up over k successful: $f(x_0) - f_{\mathbf{low}} \geq k_{\mathbf{S}} \frac{\eta_1 \sigma_{\mathbf{min}} C}{6} \epsilon^{\frac{3}{2}}$

Cubic regularization: worst-case bound is optimal

Sharpness: for any $\epsilon > 0$ and $\tau > 0$, to generate $|g(x_k)| \leq \epsilon$, cubic regularization/ARC applied to this f takes precisely

$$\left\lceil \epsilon^{-\frac{3}{2}+\tau} \right\rceil \quad \textbf{function evaluations}$$

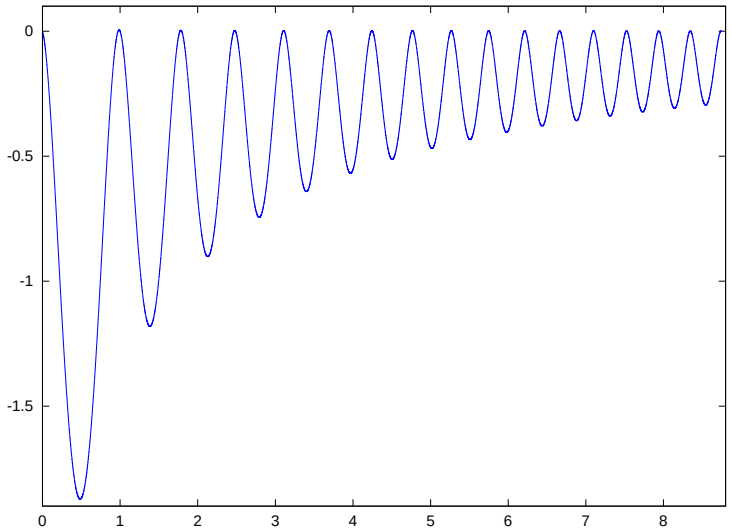




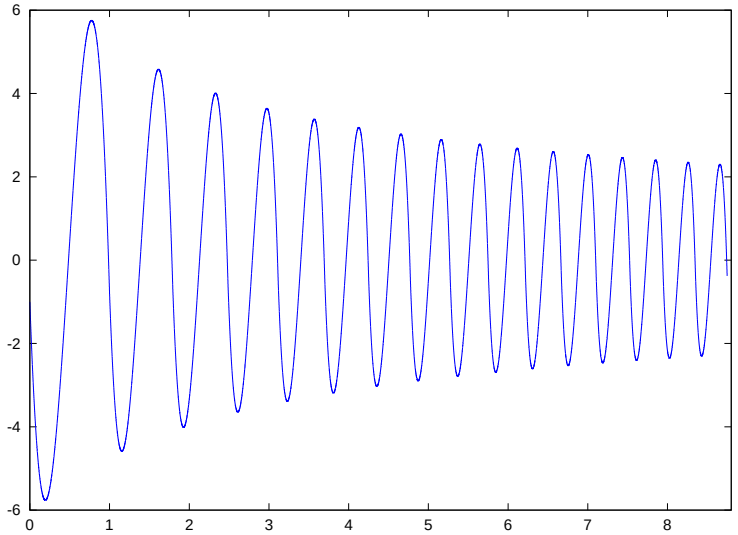
ARC's worst-case bound is optimal within a large class of second-order methods for f with Lipschitz continuous H.[CGT'11]

Second-order optimality complexity bounds

- $\mathcal{O}(\epsilon^{-3})$ evaluations for ARC and trust-region to ensure both $\|g_k\| \leq \epsilon$ and $\lambda_{\min}(B_k) \geq -\epsilon$. [CGT'12]
- this bound is tight for each method.



The gradient g.



The Hessian H.

Regularization methods with only occasionally accurate models

Probabilistic local models and methods

Context/purpose: $f \in C^1$ or $f \in C^2$ but derivatives are inaccurate/impossible/expensive to compute.

- Use model-based derivative-free optimization algorithms
- Models may be "good"/ "sufficiently accurate" only with certain probability, for example:

  $\longrightarrow$ models based on random sampling of function values (within a ball)

  $\longrightarrow$ finite-difference schemes in parallel, with total probability of any processor failing less than 0.5
- Consider cubic regularization local models with approximate first and second derivatives.
- Expected number of iterations to generate sufficiently small 'true' gradients?

Probabilistic ARC

In the ARC framework, each (realization of) the cubic regularization model [C & Scheinberg, 2015]

$$m_k(s) = f(x_k) + s^T g_k + \tfrac{1}{2} s^T b_k s + \tfrac{1}{3}\sigma_k \|s\|^3$$

has $g_k \approx \nabla f(x_k)$ and $b_k \approx H(x_k)$.

Random model/variable $M_k \longrightarrow m_k(\omega_k)$ realization;
random vars $X_k, S_k, \Sigma_k \longrightarrow x_k, s_k, \sigma_k$ realizations

Probabilistic ARC

In the ARC framework, each (realization of) the cubic regularization model [C & Scheinberg, 2015]

$$m_k(s) = f(x_k) + s^T g_k + \tfrac{1}{2} s^T b_k s + \tfrac{1}{3}\sigma_k \|s\|^3$$

has $g_k \approx \nabla f(x_k)$ and $b_k \approx H(x_k)$.

Random model/variable $M_k \longrightarrow m_k(\omega_k)$ realization; random vars $X_k, S_k, \Sigma_k \longrightarrow x_k, s_k, \sigma_k$ realizations

$\{M_k\}$ is (p)-probabilistically 'sufficiently accurate' for P-ARC if for $\{\Sigma_k, X_k\}$, the events

$$I_k = \{\|\nabla f(X_k) - G_k\| \leq \kappa_g \|S_k\|^2 \ \& \ \|(H(X_k) - B_k)S_k\| \leq \kappa_H \|S_k\|^2\}$$

hold with probability at least p (conditioned on the past). I_k occurs $\longrightarrow$ k true iteration; otherwise, false.

Probabilistic ARC - complexity

If $\nabla f(x)$ and H are Lipschitz continuous, then the expected number of iterations that P-ARC takes until $\|\nabla f(x^k)\| \leq \epsilon$ satisfies

$$\mathbb{E}(N_\epsilon) \leq \frac{1}{2p-1} \cdot \kappa_{\mathrm{p-arc}} \cdot (f(x_0) - f_{\mathrm{low}}) \cdot \epsilon^{-\frac{3}{2}}$$

provided the probability of sufficiently accurate models is $p > \frac{1}{2}$.

Analysis

- Four types of iterations (successful, unsuccessful, true and false)
- Analysis of joint stochastic processes $\{\Sigma_k, F(x_0) - F(X_k)\}$

Probabilistic ARC - analysis

Let N_ϵ hitting time for $\|\nabla f(X^k)\| \leq \epsilon$

Measure of progress towards optimality: $F_k = f(X^0) - f(X^k)$

As $F_{k+1} \leq F_k$ & $F_k \leq F_* = f(X^0) - f_{\text{low}}$: $\mathbb{E}(N_\epsilon) \leq \mathbb{E}(T_{F_*}^{F_k})$.

- If k is a true and successful iteration, then

$$f_{k+1} \geq f_k + \frac{\kappa}{(\max\{\sigma_k, \sigma_c\})^{3/2}} \|\nabla f(x^{k+1})\|^{3/2}$$

  and $\sigma_{k+1} = \max\{\gamma\sigma_k, \sigma_{\min}\}$

- If $\sigma_k \geq \sigma_c$, and iteration k is true, then it is also successful.

Probabilistic ARC - analysis

Let N_ϵ hitting time for $\|\nabla f(X^k)\| \leq \epsilon$

Measure of progress towards optimality: $F_k = f(X^0) - f(X^k)$

As $F_{k+1} \leq F_k$ & $F_k \leq F_* = f(X^0) - f_{\text{low}}$: $\mathbb{E}(N_\epsilon) \leq \mathbb{E}(T_{F_*}^{F_k})$.

- If k is a true and successful iteration, then

$$f_{k+1} \geq f_k + \frac{\kappa}{(\max\{\sigma_k, \sigma_c\})^{3/2}} \|\nabla f(x^{k+1})\|^{3/2}$$

  and $\sigma_{k+1} = \max\{\gamma\sigma_k, \sigma_{\min}\}$

- If $\sigma_k \geq \sigma_c$, and iteration k is true, then it is also successful.

Split iterations into $K' = \{k : \sigma_k > \sigma_c\}$ and $K'' = \{k : \sigma_k \leq \sigma_c\}$; analyze joint stochastic processes $\{\Sigma_k, F_k\}$ for $k \in K'$ and $k \in K''$.

Over K': σ_k is a random walk (goes 'up' w.p. $1-p$; 'down' w.p. p). Hence $\sigma_k = \sigma_c$ on average every $\frac{1}{2p-1}$ iterations.

F_k increases by $\kappa \left(\frac{\epsilon}{\sigma_c}\right)^{\frac{3}{2}}$ on average every $\frac{1}{2p-1}$ iterations.

Linesearch methods with occasionally accurate models

A probabilistic linesearch method

Initialization: Choose parameters $\gamma, \eta \in (0, 1)$. At iteration k, do:

- (Model and step calculation) Compute random model $m_k(s) = f(x_k) + s^T g_k$ and use it to generate direction g_k. Set $s_k = -\alpha_k g_k$.
- (Sufficient decrease) Check if $\rho_k = \dfrac{f(x_k) - f(x_k + s_k)}{f(x_k) - m_k(s_k)} \geq \eta$

  [this is equivalent to the Armijo condition]
- (Successful step) If $\rho_k \geq \eta$, set

  $x_{k+1} = x_k + s_k$ and $\alpha_{k+1} = \min\{\gamma^{-1}\alpha_k, \alpha_{\max}\}$.
- (Unsuccessful step) Else, set

  $x_{k+1} = x_k$ and $\alpha_{k+1} = \gamma\alpha_k$. □

More general models m_k and directions d_k possible.

A Probabilistic LineSearch (P-LS) method

The model $\{M_k\}$ is (p)-probabilistically 'sufficiently accurate' for P-LS if for corresponding $\{\mathcal{A}_k, X_k\}$, the events

$$I_k = \{\|\nabla f(X_k) - G_k\| \leq \kappa_g \mathcal{A}_k \|G_k\|\}$$

hold with probability at least p (conditioned on the past).
I_k occurs $\longrightarrow$ k true iteration; otherwise, false.

Complexity: If ∇f is Lipschitz continuous, then the expected number of iterations that P-LS takes until $\|\nabla f(x^k)\| \leq \epsilon$ satisfies

$$\mathbb{E}(N_\epsilon) \leq \frac{1}{2p-1} \cdot \kappa_{\mathbf{p-ls}} \cdot (f(x_0) - f_{\mathbf{low}}) \cdot \epsilon^{-2}$$

provided the probability of sufficiently accurate models is $p > \frac{1}{2}$.

P-LS method - complexity for special cases

f convex with bounded level sets: the expected number of iterations that P-LS takes until $f(x^k) - f_{\text{low}} \leq \epsilon$ is

$$\mathbb{E}(N_\epsilon) \leq \frac{1}{2p-1} \cdot \kappa_{\text{p-ls-c}} \cdot D^2 \cdot \epsilon^{-1}.$$

measure of progress: $F_k = \frac{1}{f(X_k) - f_{\text{low}}}$; $\mathbb{E}(N_\epsilon) = \mathbb{E}(T_{\epsilon^{-1}}^{F_k})$.

f strongly convex: the expected number of iterations that P-LS takes until $f(x^k) - f_{\text{low}} \leq \epsilon$ is

$$\mathbb{E}(N_\epsilon) \leq \frac{1}{2p-1} \cdot \kappa_{\text{pls-cc}} \cdot \frac{L}{\mu} \cdot \log(\epsilon^{-1}).$$

measure of progress: $F_k = \log \frac{1}{f(X_k) - f_{\text{low}}}$; $\mathbb{E}(N_\epsilon) = \mathbb{E}(T_{F_\epsilon}^{F_k})$ where $F_\epsilon = \log \frac{1}{\epsilon}$.

A generic algorithmic framework

Initialization: Choose a class of (possibly random) models m_k; and parameters $\gamma, \eta \in (0,1)$. At iteration k, do:

- (Model and step calculation) Compute m_k of f around x_k; and $s_k = s_k(\alpha_k)$ to reduce $m_k(s)$.
- (Sufficient decrease) Check if
$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{f(x_k) - m_k(s_k)} \geq \eta$$
- (Successful step) If $\rho_k \geq \eta$, set
$$x_{k+1} = x_k + s_k \text{ and } \alpha_{k+1} = \min\{\gamma^{-1}\alpha_k, \alpha_{\max}\}.$$
- (Unsuccessful step) Else, set
$$x_{k+1} = x_k \text{ and } \alpha_{k+1} = \gamma\alpha_k. \quad \square$$

Examples: linesearch methods ($s_k = \alpha_k d_k$); adaptive regularization ($\alpha_k = 1/\sigma_k$); trust-region.

A generic algorithmic framework...

$\{M_k\}$ is (p)-probabilistically 'sufficiently accurate' for P-Alg: I_k=$\{$ M_k 'sufficiently accurate' | $\mathcal{A}_k$ and $X_k\}$ holds with prob. p.

I_k occurs $\longrightarrow$ k true iteration; otherwise, false.

Assumption: P-Alg construction and M_k probabilistically accurate must ensure: there exists $C > 0$ s.t. if $\alpha_k \leq C$ and iteration k is true then k is also successful. Hence $\alpha_{k+1} = \min\{\gamma^{-1}\alpha_k, \alpha_{\max}\}$ and $f_{k+1} \geq f_k + h(\alpha_k)$.

Result: For P-Alg with (p)-probabilistically accurate models, the expected number of iterations to reach desired accuracy can be founded as follows

$$\mathbb{E}(N_\epsilon) \leq \frac{1}{2p-1} \cdot \kappa_{\mathrm{p-alg}} \cdot \frac{F_\epsilon}{h(C)},$$

where $p > \frac{1}{2}$ and $F_\epsilon \geq F_k$ total function decrease.

Generating (p)-sufficiently accurate models

- Stochastic gradient and batch sampling [Byrd et al, 2012]

$$\|\nabla f_{S_k}(x^k) - \nabla f(x^k)\| \leq \mu \|\nabla f_{S_k}(x^k)\|$$

with $\mu \in (0,1)$ and fixed, sufficiently small α_k.

- Models formed by sampling of function values in a ball $B(x_k, \Delta_k)$ (model-based dfo)
  M_k (p)-fully linear model: if the event

$$I_k^l = \{\|\nabla f(X^k) - G^k\| \leq \kappa_g \Delta_k\}$$

holds at least w.p. p (conditioned on the past).
M_k (p)-fully quadratic model: if the event

$$I_k^q = \{\|\nabla f(X^k) - G^k\| \leq \kappa_g \Delta_k^2 \quad \text{and} \quad \|H(X^k) - B^k\| \leq \kappa_H \Delta_k\}$$

holds at least w.p. p (conditioned on the past).

Conclusions and future directions

Some results not covered (but existent/in progress):

- high-order adaptive regularization methods: [Birgin et al. ('15)]

$$m_k(s) = T_{p-1}(x_k, s) + \frac{\sigma_k}{p}\|s\|^p$$

  where $T_{p-1}(x_k, s)$ (p-1)-order Taylor polynomial of f.

  Complexity: $O(\epsilon^{-\frac{p}{p-1}})$ to ensure $\|g(x_k)\| \leq \epsilon$ [approx, local model min]
  Complexity of pth order criticality? [in progress]

- Complexity of constrained optimization (with convex, nonconvex constraints): for carefully devised methods, it is the same as for the unconstrained case [CGT ('12,'16)]

- Optimization with occasionally accurate models:
  - second-order criticality (in progress)
  - stochastic function values - trust-region approach [Arxiv, with Blanchet, Meninckelly, Scheinberg'16]