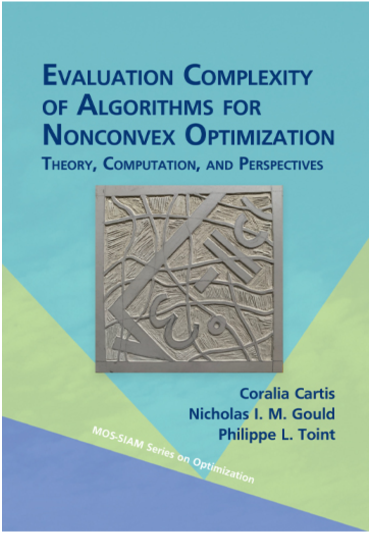
Some recent proposals for nonconvex optimization

Serge Gratton, Sadok Jerad, Philippe Toint

INP - ANITI / UNamur

BRAZOPT 2024 Rio de Janeiro, March 2024

First: a brief publicity break :-)



The problem

Once more, the standard unconstrained nonconvex optimization problem

$$\min_{x \in \mathbf{R}^n} f(x)$$

where the objective function f is

- "sufficiently" smooth
- bounded below

Remarkable one can still say (hopefully) interesting things on this subject!

A brief outline

Yet another fast variant of Newton's method

- The full-space AN2C... and its complexity
- A subspace version
- Numerical illustration

Objective Function Free Optimization (OFFO)

- Noise and nonlinear optimization
- Adagrad (and friends) as trust-region method(s)
- A fully second-order variant
- OFFAR: a "fast" second-order OFFO method

Outline

AN2C: a fast regularized Newton's method

A story of OFFO

Motivation and inspiration

Newton's method:

$$x_{k+1} = x_k - {H_k}^{-1} g_k$$

where $H_k = \nabla_x^2 f(x_k)$ and $g_k = \nabla_x^1 f(x_k)$

$\Longrightarrow$ the workhorse of nonlinear optimization, but...

- local convergence only for the vanilla version
- can be (very) slow $\mathcal{O}(\epsilon^{-2})$ when convergent (Cartis, Gould, T, 2010), even with exact linesearch (Cartis,Gould, T., 2022)

Globalizations:

- quadratic regularization (Goldfeldt, Quandt, Trotter, 1966): simple subproblem but can be just as slow (Ueda,Yamashita, 2014)
- trust-region (Moré, 1983, Conn, Gould, T, 2000): more complicated subproblem... and also slow
- cubic regularization (Nesterov, Polyak, 2006, Cartis, Gould, T., 2011): more complicated subproblem, but "fast" $\mathcal{O}(\epsilon^{-3/2})$

Today's question...

Can one combine fast convergence and simple subproblem?

(simple = a single linear solve?)

A previous proposal: (Birgin, Martinez, 2017): pick a subproblem to ensure fast convergence

Recent progress (Doikov, Nesterov, 2023, Mischenko, 2023) for convex problems: a combination of quadratic regularization (à la GQT) and gradient-dependent scaling (Fan, Yuan, 2001). Consider

$$x_{k+1} = x_k - (H_k + \sqrt{\alpha \|g_k\|} I)^{-1} g_k$$

Not enough for nonconvex problems! Can we improve it?

- Cannot ignore possible negative eigenvalues in H_k!

Our aim: use this idea with minimal consideration of eigenvalues

Adaptive Newton with Negative Curvature (AN2C) (1)

The idea (using the generic constant κ)

1) first try an *a priori* regularization using $\sqrt{\kappa\sigma_k\|g_k\|}$:

REGSTEP($g_k, H_k, \sigma_k, \kappa$)
Attempt to solve the linear system

$$(H_k + \sqrt{\kappa\sigma_k\|g_k\|}I)s_k^{def} = -g_k.$$

If s_k^{def} can be obtained such that

$$(s_k^{def})^T(H_k + \sqrt{\kappa\sigma_k\|g_k\|}I)s_k^{def} > 0,$$

$$\|s_k^{def}\| \leq \kappa\sqrt{\frac{\|g_k\|}{\sigma_k}},$$

$$\|r_k^{def}\| \leq \min\left(\kappa\sqrt{\kappa\sigma_k\|g_k\|}\|s_k^{def}\|, \kappa\|g_k\|\right)$$

where $r_k^{def} = (H_k + \sqrt{\kappa\sigma_k\|g_k\|}I)s_k^{def} + g_k$, return s_k^{def}.

Adaptive Newton with Negative Curvature (AN2C) (2)

2) if unsuccessful and curvature not too negative:

NWTSTEP($g_k, H_k, \sigma_k, \kappa$)
(Approximately) solve

$$\left(H_k + (\sqrt{\sigma_k \|g_k\|} + [-\lambda_{\min}(H_k)]_+)I\right) s_k^{neig} = -g_k$$

such that

$$\left\| \left[H_k + (\sqrt{\sigma_k \|g_k\|} + [-\lambda_{\min}(H_k)]_+)I\right] s_k^{neig} + g_k \right\| \le \min\left(\kappa \sqrt{\sigma_k \|g_k\|} \|s_k^{neig}\|, \kappa \|g_k\|\right).$$

Return s_k^{neig}

Adaptive Newton with Negative Curvature (AN2C) (3)

3) if still unsuccessful, take a negative curvature step:

EIGENSTEP($g_k, H_k, \sigma_k, \kappa$)
Compute u_k such that

$$g_k^\intercal u_k \le 0, \;\; \|u_k\| = 1 \;\; \text{and} \;\; u_k^T H_k u_k \le \kappa \lambda_{\min}(H_k)$$

and set

$$s_k = \kappa \sqrt{\frac{\|g_k\|}{\sigma_k}} u_k.$$

Return s_k.

(This is the case we wish to avoid as much as possible)

An overview of the full AN2C

Step 0: Initialization x_0, $\sigma_0 > 0$ $\epsilon \in (0, 1]$, κ. Set $k = 0$.

Step 1: Check termination Terminate if $\|g_k\| \leq \epsilon$.

Step 2 (Optional): Attempt an *a priori* regularization step
$s_k = \mathrm{REGSTEP}(\mathrm{g_k}, \mathrm{H_k}, \sigma_\mathrm{k}, \kappa)$. If successful, go to Step 5.

Step 3 : Newton Step Computation If $\lambda_{\min}(H_k) > -\kappa\sqrt{\sigma_k\|g_k\|}$,
$s_k = \mathrm{NWTSTEP}(\mathrm{g_k}, \mathrm{H_k}, \sigma_\mathrm{k}, \kappa)$ and go to Step 5.

Step 4 : Else take an eigen-step $s_k = \mathrm{EIGENSTEP}(\mathrm{g_k}, \mathrm{H_k}, \sigma_\mathrm{k}, \kappa)$.

Step 5: Acceptance test Evaluate $f(x_k + s_k)$ and

$$\rho_k = \frac{f(x_k) - f(x_k + s_k)}{-(g_k^\mathsf{T} s_k + \frac{1}{2} s_k^\mathsf{T} H_k s_k)}.$$

If $\rho_k \geq \eta_1$, set $x_{k+1} = x_k + s_k$ else $x_{k+1} = x_k$.

Step 6: Regularization parameter update Set

$$\sigma_{k+1} \in \begin{cases} [\max(\sigma_{\min}, \gamma_1\sigma_k), \sigma_k] & \text{if } \rho_k \geq \eta_2, \\ [\sigma_k, \gamma_2\sigma_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_2\sigma_k, \gamma_3\sigma_k] & \text{if } \rho_k < \eta_1. \end{cases}$$

AN2C: comments

- Step 2 not necessary for the theory, but instrumental in reducing the number of eigenvalue computations
- In the full-space context, checking positive definiteness can be achieved by attempting a Cholesky factorization...
- ...but can also be checked if a Krylov solver is used
- Acceptance rule and regularization parameter update standard (as in adaptive cubic)

AN2C: worst-case complexity

AS.1 f is two times continuously differentiable in $\mathbb{R}^n$.
AS.2 $f(x) \geq f_{\rm low}$ for all $x \in \mathbb{R}^n$.
AS.3 $\nabla_x^2 f$ is globally Lipschitz continuous
AS.4 There exists a constant $\kappa_B > 0$ such that
$\max(0, -\lambda_{\min}(\nabla_x^2 f(x))) \leq \kappa_B$ for all $x \in \{y \in \mathbb{R}^n \mid f(y) \leq f(x_0)\}$.

Suppose AS.1-AS.4 hold. Then AN2C (with suitable choices for κ!) requires at most

$$\mathcal{O}\left(\epsilon^{-3/2} + |\log(\epsilon)|\right)$$

iterations (and evaluations of f and its derivatives) to produce an iterate x_k such that $\|g_k\| \leq \epsilon$, and at most an additional

$$\mathcal{O}\left(\epsilon^{-3}(1 + |\log(\epsilon)|)\right)$$

iterations and evaluations to ensure that $\lambda_{\min}(\nabla_x^2 f(x_k)) \geq -\epsilon$.

Numerical illustration (1)

Environment: Matlab

Criteria: performance (π_{algo}) (# of iterations) and reliability (ρ_{algo})

Test problems:

119 small, 74 medium, 59 "largish" problems from the OPM/CUTEst collection

Algorithms::

AN2CER: Uses REGSTEP + Cholesky factorization + eig() for eigenvalue computations

AN2C: Does not use REGSTEP + Cholesky factorization + eig()

AR2: Adaptive cubic regularization (AR2) with modified subproblem termination

TR2M: l_2 trust-region with Moré-Sorensen subproblem solver

Numerical illustration (2)

	small pbs.		medium pbs.		largish pbs.	
`algo`	$\pi_{\texttt{algo}}$	$\rho_{\texttt{algo}}$	$\pi_{\texttt{algo}}$	$\rho_{\texttt{algo}}$	$\pi_{\texttt{algo}}$	$\rho_{\texttt{algo}}$
AN2CER	0.88	96.64	0.85	93.24	0.85	94.92
AN2CE	0.91	96.64	0.91	95.95	0.81	86.44
AR2	0.92	97.48	0.85	93.24	0.84	93.22
TR2M	0.91	94.96	0.86	93.24	0.83	91.53

Efficiency and reliability statistics for the OPM problems (full-space variants)

- AN2CER; NWTSTEP for 6.4% of all iterations and EIGENSTEP for $< 1\%$
- AN2C: NWTSTEP at all iterations, but never EIGENSTEP
- results for second-order points undistinguishable

A subspace variant (AN2CKU)

Ideas:

- at each iteration, choose a subspace $\mathcal{S}_k$
- compute steps/eigenvalues/eigenvectors in $\mathcal{S}_k$ (potentially much cheaper)
- in each subspace ensure that the step yields

$$\|\text{subspace-residual}\| \leq \kappa \, \|\text{full-space residual}\|$$

The same complexity results continue to hold.

Specialized Lanczos-based implementation for Krylov subspaces!

Numerical illustration (3)

- Lanczos-based subproblem solvers for all algos
- AN2CKYU uses a slightly modified eigen-step

	small pbs.		medium pbs.		largish pbs.	
algo	$\pi_{\texttt{algo}}$	$\rho_{\texttt{algo}}$	$\pi_{\texttt{algo}}$	$\rho_{\texttt{algo}}$	$\pi_{\texttt{algo}}$	$\rho_{\texttt{algo}}$
AN2CKU	0.86	96.64	0.81	93.24	0.77	86.44
AN2CKYU	0.91	96.64	0.90	95.95	0.85	91.53
AR2K	0.92	97.48	0.87	93.24	0.89	93.22
TR2K	0.94	96.64	0.85	87.84	0.77	84.75

Efficiency and reliability statistics for the OPM problems (Krylov-space variants)

AN2CKYU uses the eigen-step for 0.25% of all iterations for small problems, 0.23% for medium problems and never for largish ones. In all other case, it reduces to a Lanczos-based approximate linear system solver.

Outline

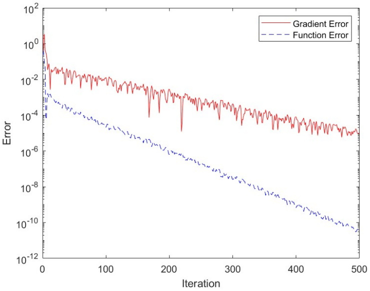
AN2C: a fast regularized Newton's method

A story of OFFO

And now something very different. . .

Our target: robust algorithms for noisy functions/inexact arithmetic

For convergence, standard methods (TR, AR) requires an error on function values which is the square (!) of that on the gradient (e.g. Bellavia et al, 22)



$\Rightarrow$ Design algorithms that do not evaluate the function

Adaptive gradient methods:

- Adagrad (Duchi et al, 2011)
- WNGrad (Wu, Ward, Bottou, 2018)
- Adam (Kingma, Ba, 2014)

A trust-region method:

- : Adatr (Grapiglia, 2022)

$\Rightarrow$ Objective Function Free Optimization = OFFO

ASTR1 an adaptive trust-region algorithm

Step 0: Initialization. x_0 is given. Set $k = 0$.

Step 1: Define the TR. Compute $g_k = g(x_k)$ and define

$$\Delta_{i,k} = \frac{|g_{i,k}|}{w_{i,k}}$$

where $w_{i,k} \geq \varsigma_i > 0$ are weights.

Step 2: Hessian approximation. Select a symmetric B_k.

Step 3: GCP. Define

$$s^L_{i,k} = -\mathrm{sgn}(g_{i,k})\Delta_{i,k} \quad \text{and} \quad s^Q_k = \gamma_k s^L_k$$

with

$$\gamma_k = \begin{cases} \min\left[1, \dfrac{|g_k^T s_k^L|}{(s_k^L)^T B_k s_k^L}\right] & \text{if } (s_k^L)^T B_k s_k^L > 0, \\ 1 & \text{otherwise.} \end{cases}$$

Step 3: Step. Compute a step s_k such that $|s_{i,k}| \leq \Delta_{i,k}$ $(\forall i)$ and

$$g_k^T s_k + \tfrac{1}{2} s_k^T B_k s_k \leq g_k^T s_k^Q + \tfrac{1}{2}(s_k^Q)^T B_k s_k^Q$$

Step 5: New iterate. Set $x_{k+1} = x_k + s_k$, increment k, and go to Step 1.

ASTR1: comments

- the objective function is not evaluated $\Rightarrow$ OFFO ... and thus the TR radius cannot depend on ared/prered.
- large weights $\Rightarrow$ short steps
- γ_k minimize the quadratic model between 0 and s_k^L

Suppose that $f \in C^1$, has Lipschitz gradient with constant L and that $\|B_k\| \le \kappa_B$. Then

$$f(x_{k+1}) \le f(x_k) - \sum_{i=1}^{n} \frac{\varsigma_{\min} g_{i,j}^2}{2\kappa_B w_{i,j}} + \tfrac{1}{2}(\kappa_B + L) \sum_{i=1}^{n} \frac{g_{i,j}^2}{w_{i,j}^2}$$

$\Rightarrow$ descent for large enough weights $w_{i,k}$

ASTR1 with ADAGRAD-like weights (1)

For given $\varsigma \in (0,1]$, $\vartheta \in (0,1]$ and $\mu \in (0,1)$, define

$$\boxed{w_{i,k} \in \left[\vartheta\left(\varsigma + \sum_{\ell=0}^{k} g_{i,\ell}^2\right)^{\mu}, \left(\varsigma + \sum_{\ell=0}^{k} g_{i,\ell}^2\right)^{\mu}\right]}$$

For $\vartheta = 1$ and $\mu = \frac{1}{2}$, $w_{i,k} = \sqrt{\varsigma + \sum_{\ell=0}^{k} g_{i,\ell}^2}$ and

ASTR1 with $\vartheta = 1$, $\mu = \frac{1}{2}$ and $B_k = 0$ is ADAGRAD

Suppose that $f \in C^1$, has Lipschitz gradient with constant L and is bounded below. Then ASTR1 with ADAGRAD-like weights, $\mu \in (0,1]$ and $\|B_k\|$ uniformly bounded requires at most

$$\mathcal{O}\left(\epsilon^{-1}\right)$$

iterations to produce an iterate k such that $\text{average}_{0,\ldots,k} \|g_\ell\|^2 \le \epsilon$.

More on ASTR1

- Extends known result by (Wu, Ward, Bottou, 2018)
- Allows the use of curvature information in an ADAGRAD-like method (Barzilai-Borwein, LBFGS, quasi-Newton, ...true Hessian)
- The above bound is essentially sharp.

Also possible with the "divergent" weights

$$w_{i,k} \in [v_{i,k}(k+1)^{\nu}, v_{i,k}(k+1)^{\mu}]$$

for $0 < \nu \leq \mu < 1$ and

$$v_{i,k} = \max_{0,\ldots,k} |g_{i,\ell}| \quad \text{or} \quad v_{i,k} = \underset{0,\ldots,k}{\text{average}} |g_{i,\ell}|$$

Slightly weaker (sharp) complexity result

Some results on the small noiseless OPM problems

Method	$\pi_{\texttt{algo}}$	$\rho_{\texttt{algo}}$
adagbfgs3	0.75	69.75
sdba (using f)	0.73	68.91
adagH	0.72	69.75
adagrad	0.69	73.11
maxg	0.66	66.39
adagbb	0.63	64.71
adam	0.54	30.25

Performance and reliability statistics for deterministic OFFO and steepest descent algorithms on small OPM problems ($\epsilon = 10^{-6}$)

The impact of noise

algo	$\rho_{\texttt{algo}}$/relative noise level				
	0%	5%	15%	25%	50%
adagH	83.19	84.96	84.20	84.71	82.18
adagbfgs3	78.15	80.50	80.50	80.84	80.18
adagrad	77.31	80.50	80.25	80.17	80.17
adagbb	75.69	80.08	80.17	79.58	79.41
maxg	74.79	74.37	75.55	78.15	78.07
adam	40.34	35.55	36.30	44.03	45.80
sdba	81.51	30.92	31.85	34.87	29.58

Reliability of OFFO algorithms and steepest descent as a function of the level of relative Gaussian noise ($\epsilon = 10^{-3}$)

Towards second-order criticality

Use a similar mechanism for second-order criticality?

At x_k, let

$$T_{f,2}(x_k, d) = f(x_k) + g(x)_k^T d + \frac{1}{2} d^T H(x_k) d.$$

and the second-order criticality measure

$$\phi_{f,2}^{\delta}(x_k) = \max_{\|d\| \leq \delta} -\left(g(x_k)^T d + \frac{1}{2} d^T H(x_k) d \right) = \max_{\|d\| \leq \delta} \Delta q_k(d)$$

Define:

x_k is ϵ-second-order critical if $\phi_{f,2}^{\delta}(x_k) \leq \epsilon$

Idea: Use $\phi_{f,2}^{\delta}(x_k)$ to define weights for the trust-region

ASTR2: a TR OFFO method for 2nd-order criticality

Step 0: Initialization. Given: x_0 and algo constants. Set $k = 0$.

Step 1: Compute derivatives. Compute g_k and H_k, as well as ϕ_k and $\widehat{\phi}_k \stackrel{\rm def}{=} \min[\phi^{\delta}_{f,2}(x_k), \kappa]$.

Step 2: Define the TR radii. For weights w_k^L and w_k^Q, set

$$\Delta_k^L = \frac{\|g_k\|}{w_k^L} \quad \text{and} \quad \Delta_k^Q = \frac{\widehat{\phi}_k}{w_k^Q}.$$

Step 3: Step computation. If $\|g_k\|^2 \geq \widehat{\phi}_k^3$, set $s_k = -g_k/w_k^L$.
Otherwise, set s_k such that

$$\|s_k\| \leq \Delta_k^Q \quad \text{and} \quad \Delta q_k(s_k) \geq \max\left[\Delta q_k^C, \Delta q_k^E\right]$$

where $\Delta q_k^C = \max_{\alpha \geq 0, \alpha\|g_k\| \leq \Delta_k^Q} \Delta q_k(-\alpha g_k)$ and

$$\Delta q_k^E = \begin{cases} \max_{\alpha \geq 0, \alpha \leq \Delta_k^Q} \Delta q_k(\alpha u_k) & \text{if } \lambda_{\min}[H_k] < 0 \\ 0 & \text{if } \lambda_{\min}[H_k] \geq 0 \end{cases}$$

with

$$u_k^T H_k u_k \leq \kappa \lambda_{\min}[H_k], \quad u_k^T g_k \leq 0 \quad \text{and} \quad \|u_k\| = 1,$$

Step 4: New iterate. Define $x_{k+1} = x_k + s_k$, increment k and return to Step 1.

Function decrease for ASTR2

Suppose that $f \in C^2$ and has Lipschitz continuous gradient and Hessian. Then, if $\|g_k\|^2 \geq \widehat{\phi}_k^3$,

$$f_{k+1} \leq f_k - \frac{\|g_k\|^2}{w_k^L} + \frac{L_1}{2}\frac{\|g_k\|^2}{(w_k^L)^2}$$

while, if $\|g_k\|^2 < \widehat{\phi}_k^3$,

$$f_{k+1} \leq f_k - \kappa \min\left[\frac{1}{2(1+L_1)}, \frac{1}{w_k^Q}, \frac{1}{(w_k^Q)^2}\right] \widehat{\phi}_k^3 + \frac{L_2}{6}\frac{\widehat{\phi}_k^3}{(w_k^Q)^3}.$$

$\Rightarrow$ roles of w_k^L and w_k^Q complementary

Complexity of ASTR2 for ADAGRAD-like weights

When using

$$w_k^L \in \left[\vartheta\left(\varsigma + \sum_{\ell=0,\ell\in\mathcal{K}^L}^{k} \|g_\ell\|^2\right)^\mu, \left(\varsigma + \sum_{\ell=0,\ell\in\mathcal{K}^L}^{k} \|g_\ell\|^2\right)^\mu\right]$$

$$w_k^Q \in \left[\vartheta\left(\varsigma + \sum_{\ell=0,\ell\in\mathcal{K}^Q}^{k} \widehat{\phi}_k^3\right)^\mu, \left(\varsigma + \sum_{\ell=0,\ell\in\mathcal{K}^Q}^{k} \widehat{\phi}_k^3\right)^\mu\right]$$

Suppose that $f \in C^2$ with Lipschitz gradient and Hessian and is bounded below. Then ASTR2 with the above weights and $\mu \in (0,1]$ requires at most $\mathcal{O}\left(\epsilon^{-1}\right)$ iterations to produce an iterate k such that $\text{average}_{0,\ldots,k} \|g_\ell\|^2 \leq \epsilon$ and $\text{average}_{0,\ldots,k} \widehat{\phi}_\ell^3 \leq \epsilon$. [Essentially sharp!]

. . . and now for an OFFO regularization algorithm!

Consider now the more general

$$T_{f,p}(x,s) = f(x) + \sum_{i=1}^{p} \frac{1}{i!} \nabla_x^i f(x)[s]^i.$$

and the derived regularized model

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

We assume that $\nabla_x^p f$ is globally Lipschitz.

The OFFAR algorithm

(again using generic κ)

Step 0: Initialization: x_0, $\nu_0 > 0$, ϵ and constants. Set $k = 0$.

Step 1: Check for termination: Evaluate $g_k = \nabla_x^1 f(x_k)$ and terminate if $\|g_k\| \leq \epsilon$. Else, evaluate $\{\nabla_x^i f(x_k)\}_{i=2}^p$.

Step 2: Step calculation: If $k = 0$, set $\sigma_0 = \mu_0 = \nu_0$. Else set

$$\mu_k = \frac{p!\|g_k\|}{\|s_{k-1}\|^p} - \kappa\sigma_{k-1} \;\text{ and }\; \sigma_k \in [\kappa\nu_k, \max(\nu_k, \mu_k)] .$$

Then compute a step s_k such that

$$m_k(s_k) < m_k(0) \;\text{ and }\; \|\nabla_s^1 T_{f,p}(x_k, s_k)\| \leq \kappa \frac{\sigma_k}{p!} \|s_k\|^p.$$

Step 3: Updates. Set $x_{k+1} = x_k + s_k$ and $\nu_{k+1} = \nu_k + \nu_k \|s_k\|^{p+1}$. Increment k by one and go to Step 1.

Complexity of OFFAR

- No objective function evaluation $\Rightarrow$ OFFO
- The use of μ_k is optional: one could simply set $\mu_k = 0$ without altering the theory. But it is important for performance.
- The definition of μ_k promotes fast growth of the regularization parameter up the problem's Lispchitz constant
- The definition of σ_k helps to limit this growth once the value of the Lipschitz constant has been reached.
- If $p = 1$, $\nu_{k+1} = \nu_k + \nu_k \|s_k\|^2$, recovering WNGrad (Wu, Ward, Bottou, 2018)

Suppose that $f \in C^p$ with $\nabla_x^p f$ Lipschitz gradient, is bounded below and is such that $\min_{\|d\| \leq 1} \nabla_x^i [d]^i \geq \kappa$ for $i = 2, \ldots, p$. Then OFFAR (with suitable constants) requires at most $\mathcal{O}\left(\epsilon^{-\frac{p+1}{p}}\right)$ iterations to produce an iterate k such that $\|g_k\| \leq \epsilon$. [Sharp!]

More on OFFAR

- Same rate as ARp using function values (Birgin et al, 2016)
- For $p = 2$, same rate as ARC/AR2 (Cartis, Gould, T. 2011). Optimal rate for second order methods
- Optimal rates for exact pth order methods (Carmon et al. 2019).

MOFFAR: If one requires that the step also satisfies

$$\max\left(0, -\lambda_{\min}[\nabla_s^2 T_{f,p}(x_k, s_k)]\right) \le \frac{\kappa\sigma_k}{(p-1)!}\|s_k\|^{p-1}$$

Suppose that $f \in C^p$ with $\nabla_x^p f$ Lipschitz gradient, is bounded below and is such that $\min_{\|d\|\le 1} \nabla_x^i [d]^i \ge \kappa$ for $i = 2, \ldots, p$. Then MOFFAR (with suitable constants) requires at most $\mathcal{O}\left(\epsilon^{-\frac{p+1}{p}}\right)$ iterations to produce an iterate k such that $\|g_k\| \le \epsilon$ and $\widehat{\phi}_k \le \epsilon$. [Sharp]

Numerical illustration

For AR2 and two variants of OFFAR with $p = 2$, differing on how aggressively μ_k forces growth in σ_k (b more aggressive than a)

	AR2	OFFAR2a	OFFAR2b
$\pi_{\mathtt{algo}}$	0.99	0.78	0.83
$\rho_{\mathtt{algo}}$	97.48	81.51	88.24

Performance and reliability statistics on the small OPM problems without noise

	5%	15%	25%	50%
AR2	40.67	30.84	24.54	6.81
OFFAR2a	80.76	75.38	70.76	56.30
OFFAR2b	85.97	80.67	72.69	47.98

Reliability statistics $\rho_{\mathtt{algo}}$ for 5%, 15%, 25% and 50% relative random Gaussian noise (averaged on 10 runs)

Conclusions

AN2C promising, both in full-space and subspace versions

Computing the value of f is not necessary for (theoretical) fast convergence

The use of curvature information is possible (and beneficial) in standard OFFO adaptive methods

OFFO creates some interesting challenges in convergence theory!

In particular stochastic variants are of interest.

Thank you for your interest. . . and patience!

Details in. . .

S. Gratton and S. Jerad and Ph. L. Toint, "Yet Another Fast Variant of Newton's Method for Nonconvex Optimization", to appear in IMA Journal of Numerical Analysis, 2024, arXiv:2203.10065.

S. Gratton and S. Jerad and Ph. L. Toint, "Parametric Complexity Analysis for a Class of First-Order Adagrad-like Algorithms", to appear in Optimization Methods and Software, 2023, arXiv:2203.01647.

S. Gratton and Ph. L. Toint, "OFFO minimization algorithms for second-order optimality and their complexity", Computational Optimization and Applications, vol. 84, pp. 573—607, 2022.

S. Gratton and S. Jerad and Ph. L. Toint, "Convergence properties of an Objective-Function-Free Optimization regularization algorithm, including an $\mathcal{O}(\epsilon^{-3/2})$ complexity bound", SIAM Journal on Optimization, vol. 33(3), pp. 1621–1646, 2023.