Evaluation Complexity In Nonlinear Optimization Using Lipschitz-Continuous Hessians

Philippe Toint (with Coralia Cartis and Nick Gould)

Namur Center for Complex Systems (naXys), University of Namur, Belgium

(philippe.toint@fundp.ac.be)

Florianopolis, March 2014

The problem

We consider the unconstrained nonlinear programming problem:

$$\text{minimize} \quad f(x)$$

for $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \rightarrow \mathbb{R}$ smooth.

Important special case: the nonlinear least-squares problem

$$\text{minimize} \quad f(x) = \tfrac{1}{2}\|F(x)\|^2$$

for $x \in \mathbb{R}^n$ and $F : \mathbb{R}^n \rightarrow \mathbb{R}^m$ smooth.

A useful observation

Note the following: if

- f has gradient g and globally Lipschitz continuous Hessian H with constant $2L$

Taylor, Cauchy-Schwarz and Lipschitz imply

$$\begin{aligned} f(x+s) &= f(x) + \langle s, g(x)\rangle + \tfrac{1}{2}\langle s, H(x)s\rangle \\ &\quad + \int_0^1 (1-\alpha)\langle s, [H(x+\alpha s) - H(x)]s\rangle \, d\alpha \\ &\le \underbrace{f(x) + \langle s, g(x)\rangle + \tfrac{1}{2}\langle s, H(x)s\rangle + \tfrac{1}{3}L\|s\|_2^3}_{m(s)} \end{aligned}$$

$\Longrightarrow$ reducing m from $s = 0$ improves f since $m(0) = f(x)$.

Approximate model minimization

Lipschitz constant L unknown $\Rightarrow$ replace by adaptive parameter σ_k in the model :

$$m(s) \stackrel{\rm def}{=} f(x) + s^T g(x) + \tfrac{1}{2} s^T H(x) s + \tfrac{1}{3}\sigma_k \|s\|_2^3$$

Computation of the step:

1. minimize $m(s)$ until an approximate first-order minimizer is obtained:

$$\|\nabla_s m(s)\| \le \min[\kappa_{\rm stop}, \|s\|] \, \|g_k\| \quad \text{and "(before) line minimizer"}$$

(s-rule)

Note: no global optimization involved.

Adaptive Regularization with Cubic (ARC)

Algorithm 1.1: The ARC2 Algorithm

Step 0: Initialization: x_0 and $\sigma_0 > 0$ given. Set $k = 0$

Step 1: Step computation: Compute s_k for which

$$\|\nabla_s m(s_k)\| \le \min[\kappa_{\text{stop}}\|s_k\|]\|g_k\| \text{ and "(before) line minimizer"}$$

Step 2: Step acceptance: Compute $\rho_k = \dfrac{f(x_k) - f(x_k + s_k)}{f(x_k) - m_k(s_k)}$

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

Step 3: Update the regularization parameter:

$$\sigma_{k+1} \in \begin{cases} (0, \sigma_k] & = \frac{1}{2}\sigma_k & \text{if } \rho_k > 0.9 & \textit{very successful} \\ [\sigma_k, \gamma_1\sigma_k] & = \sigma_k & \text{if } 0.1 \le \rho_k \le 0.9 & \textit{successful} \\ [\gamma_1\sigma_k, \gamma_2\sigma_k] & = 2\sigma_k & \text{otherwise} & \textit{unsuccessful} \end{cases}$$

Cubic regularization highlights

$$f(x+s) \leq m(s) \equiv f(x) + s^T g(x) + \tfrac{1}{2} s^T H(x) s + \tfrac{1}{3} L \|s\|_2^3$$

- Nesterov and Polyak minimize m globally and exactly
  - N.B. m may be non-convex!
  - efficient scheme to do so if H has sparse factors
- global (ultimately rapid) convergence to a 2nd-order critical point of f
- better worst-case function-evaluation complexity than previously known

Obvious questions:

- can we avoid the global Lipschitz requirement? YES!
- can we approximately minimize m and retain good worst-case function-evaluation complexity? YES !
- does this work well in practice? yes

Function-evaluation complexity (1)

How many function evaluations (iterations) are needed to ensure that

$$\|g_k\| \leq \epsilon?$$

If H is globally Lipschitz, the s-rule is applied and additionally s_k is the global (line) minimizer of $m_k(\alpha s_k)$ as a function of α, the ARC2 algorithm requires at most

$$\left\lceil \frac{\kappa_S}{\epsilon^{3/2}} \right\rceil \text{ function evaluations}$$

for some κ_S independent of ϵ.

c.f. Nesterov & Polyak

Note: an $O(\epsilon^{-3})$ bound holds for convergence to second-order critical points.

Function-evaluation complexity (2)

Is the bound in $O(\epsilon^{-3/2})$ sharp? YES!!!

Construct a unidimensional example with

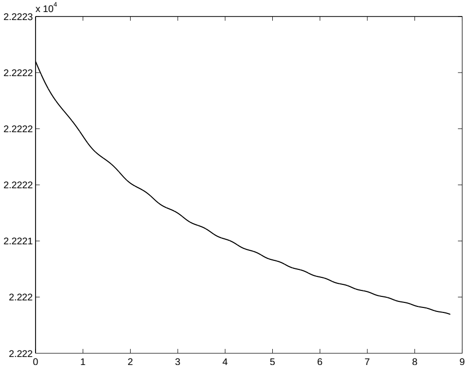
$$x_0 = 0, \quad x_{k+1} = x_k + \left(\frac{1}{k+1}\right)^{\frac{1}{3}+\eta},$$

$$f_0 = \frac{2}{3}\zeta(1+3\eta), \quad f_{k+1} = f_k - \frac{2}{3}\left(\frac{1}{k+1}\right)^{1+3\eta},$$

$$g_k = -\left(\frac{1}{k+1}\right)^{\frac{2}{3}+2\eta}, \quad H_k = 0 \text{ and } \sigma_k = 1,$$

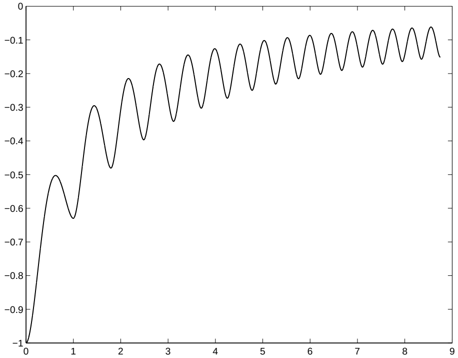
Use Hermite interpolation on $[x_K, x_{k+1}]$.

An example of slow ARC2 (1)



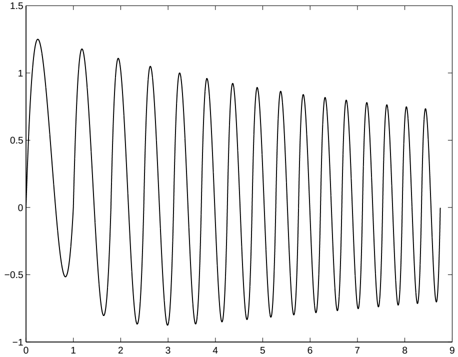
The objective function

An example of slow ARC2 (2)



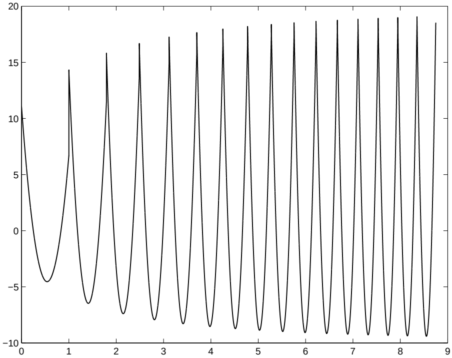
The first derivative

An example of slow ARC2 (3)



The second derivative

An example of slow ARC2 (4)



The third derivative

Slow steepest descent (1)

The steepest descent method with requires at most

$$\left\lceil \frac{\kappa_C}{\epsilon^2} \right\rceil \text{ function evaluations}$$

for obtaining $\|g_k\| \leq \epsilon$.

Nesterov
Sharp??? YES

Newton's method (when convergent) requires at most

$$O(\epsilon^{-2}) \text{ function evaluations}$$

for obtaining $\|g_k\| \leq \epsilon$!!!!

Slow Newton (1)

Choose $\tau \in (0,1)$

$$g_k = -\begin{pmatrix} \left(\frac{1}{k+1}\right)^{\frac{1}{2}+\eta} \\ \left(\frac{1}{k+1}\right)^{2} \end{pmatrix} \qquad H_k = \begin{pmatrix} 1 & 0 \\ 0 & \left(\frac{1}{k+1}\right)^{2} \end{pmatrix},$$

for $k \geq 0$ and

$$f_0 = \zeta(1+2\eta) + \frac{\pi^2}{6}, \qquad f_k = f_{k-1} - \frac{1}{2}\left[\left(\frac{1}{k+1}\right)^{1+2\eta} + \left(\frac{1}{k+1}\right)^{2}\right] \quad \text{for} \quad k \geq 1,$$

$$\eta = \eta(\tau) \stackrel{\rm def}{=} \frac{\tau}{4-2\tau} = \frac{1}{2-\tau} - \frac{1}{2}.$$

Slow Newton (2)

$$H_k s_k = -g_k,$$

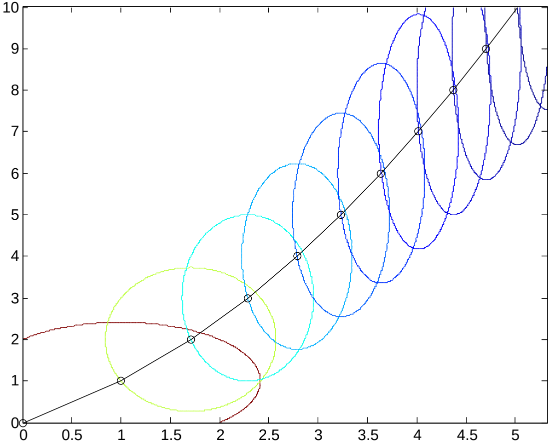
and thus

$$s_k = \begin{pmatrix} \left(\frac{1}{k+1}\right)^{\frac{1}{2}+\eta} \\ 1 \end{pmatrix},$$

$$x_0 = \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \qquad x_k = \begin{pmatrix} \sum_{j=0}^{k-1} \left(\frac{1}{j+1}\right)^{\frac{1}{2}+\eta} \\ k \end{pmatrix}.$$

Slow Newton (3)

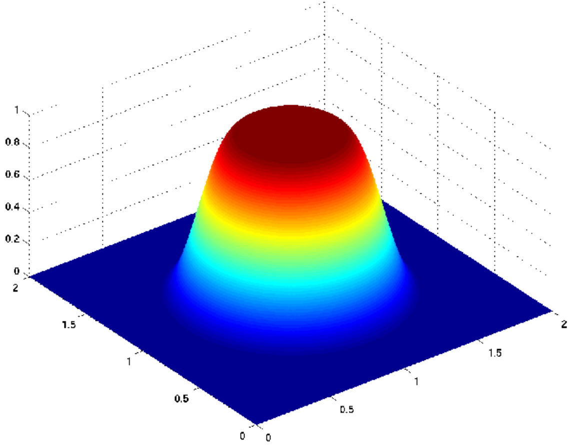
$$q_k(x_{k+1}, y_{k+1}) = f_k + \langle g_k, s_k \rangle + \tfrac{1}{2}\langle s_k, H_k s_k \rangle = f_{k+1}$$



The shape of the successive quadratic models

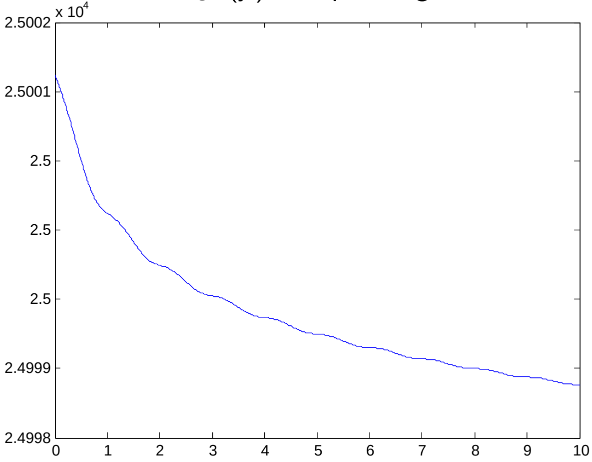
Slow Newton (4)

Define a support function $s_k(x, y)$ around (x_k, y_k)



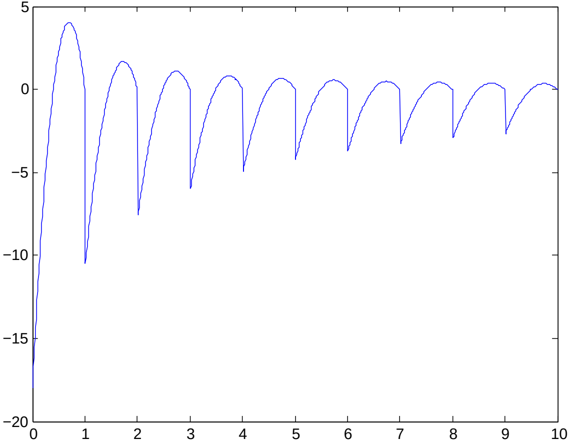
Slow Newton (5)

A background function $f_{BCK}(y)$ interpolating f_k values...



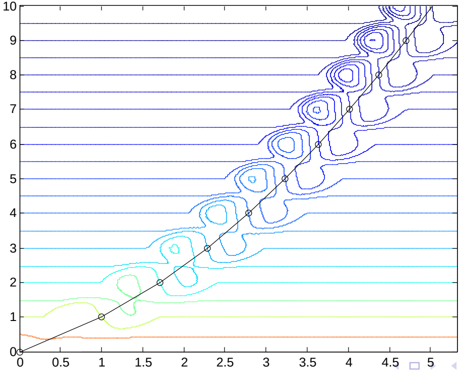
Slow Newton (6)

...with bounded third derivative



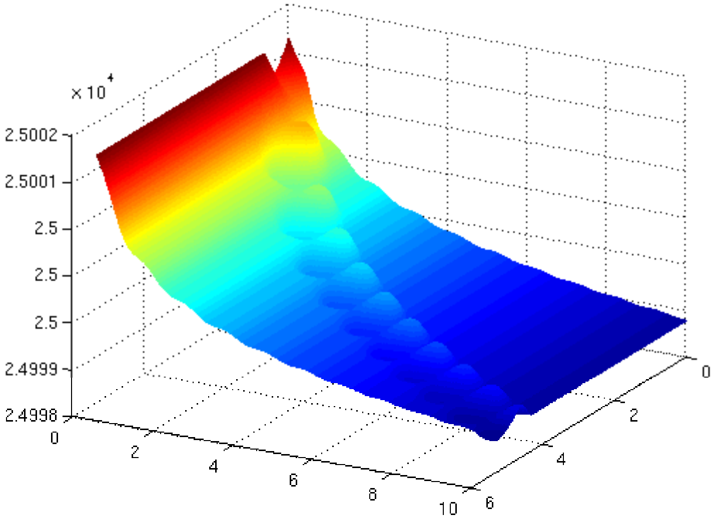
Slow Newton (7)

$$f_{SN1}(x,y) = \sum_{k=0}^{\infty} s_k(x,y) q_k(x,y) + \left[1 - \sum_{k=0}^{\infty} s_k(x,y)\right] f_{BCK}(x,y)$$



Slow Newton (8)

Some steps on a sandy dune. . .



More general second-order methods

Assume that, for $\beta \in (0,1]$, the step is computed by

$$(H_k + \lambda_k I)s_k = -g_k \quad \text{and} \quad 0 \le \lambda_k \le \kappa_s \|s_k\|^\beta$$

(ex: Newton, ARC2, Levenberg-Morrison-Marquardt, (TR2), ...)

The corresponding method may require as much as

$$\left\lceil \frac{\kappa_C}{\epsilon^{-(\beta+2)/(\beta+1)}} \right\rceil \text{ function evaluations}$$

to obtain $\|g_k\| \le \epsilon$ on functions with bounded and (segment-wise) β-Hölder continuous Hessians.

Note: ranges form ϵ^{-2} to $\epsilon^{-3/2}$

ARC2 is optimal within this class

The constrained case

Can we apply regularization to the constrained case?

Consider the constrained nonlinear programming problem:

$$\begin{array}{ll} \text{minimize} & f(x) \\ & x \in \mathcal{F} \end{array}$$

for $x \in \mathbb{R}^n$ and $f : \mathbb{R}^n \to \mathbb{R}$ smooth, and where

$$\mathcal{F} \text{ is convex.}$$

Ideas:

- exploit (cheap) projections on convex sets
- use appropriate termination criterion

$$\chi_f(x_k) \stackrel{\text{def}}{=} \left| \min_{x+d\in\mathcal{F}, \|d\|\leq 1} \langle \nabla_x f(x_k), d \rangle \right|,$$

Constrained step computation

$$\min_s \quad f(x) + \langle s, g(x) \rangle + \tfrac{1}{2}\langle s, H(x)s \rangle + \tfrac{1}{3}\sigma\|s\|^3$$

subject to

$$x + s \in \mathcal{F}$$

- minimization of the cubic model until an approximate first-order critical point is met, as defined by

$$\chi_m(s) \le \min(\kappa_{\text{stop}}, \|s\|)\,\chi_f(x_k)$$

c.f. the "s-rule" for unconstrained

Note: OK at local constrained model minimizers

A constrained regularized algorithm

Algorithm 3.1: ARC for Convex Constraints (ARC2CC)

Step 0: Initialization. $x_0 \in \mathcal{F}$, σ_0 given. Compute $f(x_0)$, set $k = 0$.

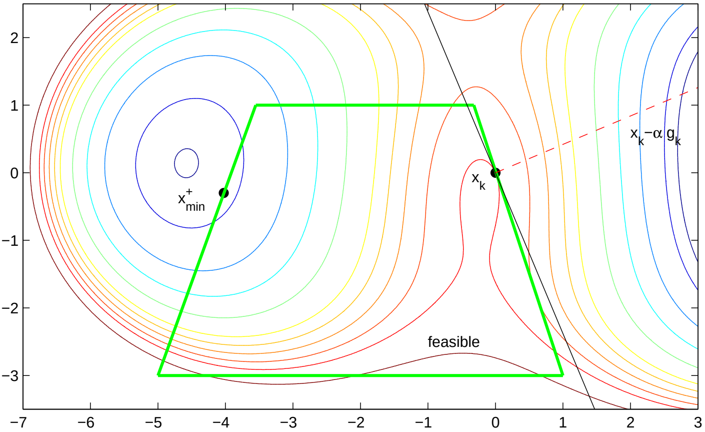
Step 1: Step calculation. Compute s_k and $x_k^+ \stackrel{\rm def}{=} x_k + s_k \in \mathcal{F}$ such that $\chi_m(s_k) \leq \min(\kappa_{\rm stop}, \|s_k\|)\, \chi_f(x_k)$.

Step 2: Acceptance of the trial point. Compute $f(x_k^+)$ and ρ_k. If $\rho_k \geq \eta_1$, then $x_{k+1} = x_k + s_k$; otherwise $x_{k+1} = x_k$.

Step 3: Regularisation parameter update. Set

$$\sigma_{k+1} \in \left\{ \begin{array}{ll} (0, \sigma_k] & \text{if } \rho_k \geq \eta_2, \\ [\sigma_k, \gamma_1 \sigma_k] & \text{if } \rho_k \in [\eta_1, \eta_2), \\ [\gamma_1 \sigma_k, \gamma_2 \sigma_k] & \text{if } \rho_k < \eta_1. \end{array} \right.$$

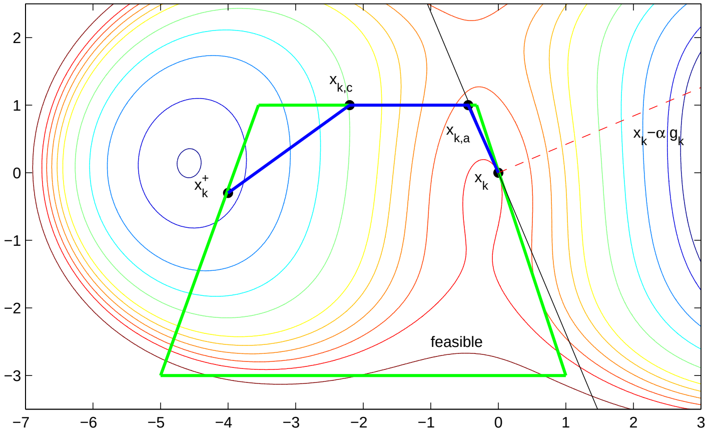
Walking through the pass...



A "beyond the pass" constrained problem with

$$m(x,y) = -x - \frac{42}{100}y - \frac{3}{10}x^2 - \frac{1}{10}y^3 + \frac{1}{3}[x^2+y^2]^{\frac{3}{2}}$$

Walking through the pass...with a sherpa



A piecewise descent path from x_k to x_k^+ on

$$m(x,y) = -x - \tfrac{42}{100}y - \tfrac{3}{10}x^2 - \tfrac{1}{10}y^3 + \tfrac{1}{3}[x^2+y^2]^{\frac{3}{2}}$$

Function-Evaluation Complexity for ARC2CC

Assume also

- $x_k \leftarrow x_k^+$ in a bounded number of feasible descent substeps
- $\|H_k - \nabla_{xx} f(x_k)\| \leq \kappa \|s_k\|^2$
- $\nabla_{xx} f(\cdot)$ is globally Lipschitz continuous
- $\{x_k\}$ bounded

The ARC2CC algorithm requires at most

$$\left\lceil \frac{\kappa_C}{\epsilon^{3/2}} \right\rceil \text{ function evaluations}$$

(for some κ_C independent of ϵ) to achieve $\chi_f(x_k) \leq \epsilon$

Caveat: cost of solving the subproblem! c.f. unconstrained case!!!

The general constrained case

Consider now the general NLO (slack variables formulation):

$$\begin{array}{ll} \text{minimize}_x & f(x) \\ \text{such that} & c(x) = 0 \quad \text{and} \quad x \in \mathcal{F} \end{array}$$

Ideas for a second-order algorithm:

1. get feasible (if possible) by minimizing $\|c(x)\|^2$ such that $x \in \mathcal{F}$
2. track the trajectory

$$\mathcal{T}(t) \stackrel{\text{def}}{=} \{x \in \mathbb{R}^n \mid c(x) = 0 \quad \text{and} \quad f(x) = t\}$$

for values of t decreasing from f(first feasible iterate) while preserving $x \in \mathcal{F}$

A detour via unconstrained nonlinear least-squares (1)

Consider

$$\text{minimize} \quad f(x) = \tfrac{1}{2}\|F(x)\|^2$$

Apply ARC2 to obtain $O(\epsilon^{-3/2})$ complexity?

- only yields $\|J(x_k)F(x_k)\| \le \epsilon$!
- requires unpalatably strong conditions on $J(x)$!

Turn to the "scaled residual"

$$\nabla_x \|F(x_k)\| \stackrel{\text{def}}{=} \begin{cases} \dfrac{\|J(x_k)^T F(x_k)\|}{\|F(x_k)\|} & \text{if } \|F(x_k)\| > 0 \\ 0 & \text{otherwise} \end{cases}$$

Copes with both zero and nonzero residuals !

A detour via unconstrained nonlinear least-squares (2)

Assume f has Lipschitz Hessian. Then the ARC2 algorithm takes at most

$$O(\epsilon^{-3/2}) \text{ function evaluations}$$

to find an iterate x_k with either

$$\nabla_x \|F(x_k)\| \leq \epsilon \quad \text{or} \quad \|F(x_k)\| \leq \epsilon.$$

- No requirement on regularity for $J(x)$!

... and via constrained nonlinear least-squares (1)

Consider now

$$\text{minimize} \quad f(x) = \tfrac{1}{2}\|F(x)\|^2 \quad \text{such that} \quad x \in \mathcal{F}$$

Remember termination rules:

$$\chi_f(x_k) \le \epsilon \qquad \text{(convex inequality constraints)}$$

$$\nabla_x \|F(x_k)\| \le \epsilon \qquad \text{(NLSQ)}$$

For inequality-constrained nonlinear least-squares, combine these into

$$\chi_{\|F(x)\|}(x_k) = \left| \min_{x+d\in\mathcal{F}, \|d\|\le 1} \langle \nabla_x \|F(x_k)\|, d\rangle \right| \le \epsilon$$

... and via constrained nonlinear least-squares (2)

Assume f has Lipschitz Hessian. Then the ARC2CC algorithm takes at most

$$O(\epsilon^{-3/2}) \text{ function evaluations}$$

to find an iterate x_k with either

$$\chi_{\|F(x)\|}(x_k) \leq \epsilon \quad \text{or} \quad \|F(x_k)\| \leq \epsilon.$$

Second-order complexity for general NLO (1)

Sketch of a short-step ARC2 (ARC2GC) algorithm

feasibility: apply ARC2CC (with $\nabla_x \|F(x_k)\|$ stopping rule) to

$$\min_x \|c(x)\|^2 \quad \text{such that} \quad x \in \mathcal{F}$$

at most $O(\epsilon^{-3/2})$ function evaluations

tracking: successively

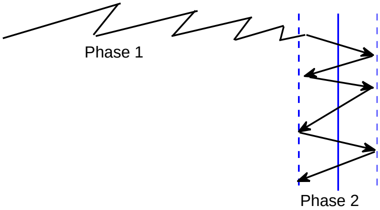
- apply one (successful) step of ARC2CC (with $\nabla_x \|F(x_k)\|$ stopping rule) to

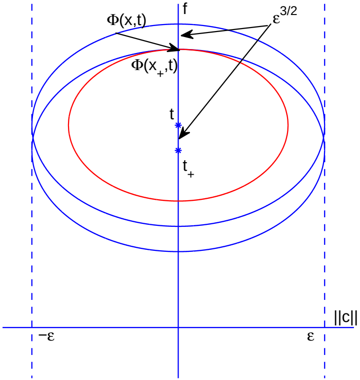
$$\min_x \phi(x) \stackrel{\text{def}}{=} \|c(x)\|^2 + (f(x) - t)^2 \quad \text{such that} \quad x \in \mathcal{F}$$

- decrease t (proportionally to the decrease in $\phi(x)$)

at most $O(\epsilon^{-3/2})$ function evaluations !

A view of Algorithm ARC2CC





Second-order complexity for general NLO (2)

Under the "conditions stated above", the ARC2CC algorithm takes at most

$$O(\epsilon^{-3/2}) \text{ function evaluations}$$

to find an iterate x_k with either

$$\|c(x_k)\| \leq \delta\epsilon \quad \text{and} \quad \chi_{\mathcal{L}} \leq \|(y,1)\|\epsilon^{2/3}$$

for some Lagrange multiplier y and where

$$\mathcal{L}(x,y) = f(x) + \langle y, c(x) \rangle,$$

or

$$\|c(x_k)\| > \delta\epsilon \quad \text{and} \quad \chi_{\|c\|} \leq \epsilon.$$

Conclusions

- Complexity analysis for first-order critical points using second-order methods complete !

$$O(\epsilon^{-3/2}) \quad \text{(unconstrained, general constraints !)}$$

- Available also for first order methods :

$$O(\epsilon^{-2}) \quad \text{(unconstrained, general constraints !)}$$

- Jarre's example $\Rightarrow$ global optimization much harder
- smooth functions littered with approximate critical points !
- ARC2 is optimal amongst second-order method
- More also known (unconstrained 2nd order criticality, DFO, etc)

Many thanks for your attention!