A Trust Region Algorithm with a Worst-Case Iteration Complexity of $\mathcal{O}(\epsilon^{-3/2})$ for Nonconvex Optimization

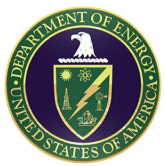
Frank E. Curtis, Lehigh University

joint work with

Daniel P. Robinson, Johns Hopkins University
Mohammadreza Samadi, Lehigh University

FoCM Conference — Montevideo, Uruguay

15 December 2014



Outline

Outline

Unconstrained (nonconvex) optimization

Given $f : \mathbb{R}^n \to \mathbb{R}$, consider the unconstrained optimization problem

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

In this talk, we are primarily interested in

- solving nonconvex instances;
- employing second-order methods;
- attaining global and fast local (i.e., superlinear) convergence;
- . . . but also ensuring good iteration (function evaluation, etc.) complexity.

Methods of interest (in this talk)

Trust region methods

- Decades of algorithmic development
- Levenberg (1944); Marquardt (1963); Powell (1970); many more!

Cubic regularization methods

- Relatively recent algorithmic development; fewer variants
- Griewank (1981); Nesterov & Polyak (2006); Cartis, Gould, & Toint (2011)

Methods of interest (in this talk)

Trust region methods

- Decades of algorithmic development
- Levenberg (1944); Marquardt (1963); Powell (1970); many more!

Cubic regularization methods

- Relatively recent algorithmic development; fewer variants
- Griewank (1981); Nesterov & Polyak (2006); Cartis, Gould, & Toint (2011)

Theoretical guarantees to assess a nonconvex optimization algorithm:

- Global convergence, i.e., $\nabla f(x_k) \to 0$
- Local convergence rate, i.e., $\|\nabla f(x_{k+1})\|_2 / \|\nabla f(x_k)\|_2 \to 0$ (or more)
- Worst-case complexity, i.e., upper bound on number of iterations[1] to achieve

$$\|\nabla f(x_k)\|_2 \leq \epsilon \quad \text{for some} \quad \epsilon > 0$$

[1] The complexity results we discuss also hold for function evaluations and subproblem solves

Methods of interest (in this talk)

Trust region methods

- Decades of algorithmic development
- Levenberg (1944); Marquardt (1963); Powell (1970); many more!
- Global convergence, local quadratic rate, $\mathcal{O}(\epsilon^{-2})$ complexity

Cubic regularization methods

- Relatively recent algorithmic development; fewer variants
- Griewank (1981); Nesterov & Polyak (2006); Cartis, Gould, & Toint (2011)
- Global convergence, local quadratic rate, $\mathcal{O}(\epsilon^{-3/2})$ complexity

Theoretical guarantees to assess a nonconvex optimization algorithm:

- Global convergence, i.e., $\nabla f(x_k) \to 0$
- Local convergence rate, i.e., $\|\nabla f(x_{k+1})\|_2 / \|\nabla f(x_k)\|_2 \to 0$ (or more)
- Worst-case complexity, i.e., upper bound on number of iterations[1] to achieve

$$\|\nabla f(x_k)\|_2 \leq \epsilon \quad \text{for some} \quad \epsilon > 0$$

[1] The complexity results we discuss also hold for function evaluations and subproblem solves

Goals and contributions

What are our goals in this work?

- Question: Can we design a TR method with improved complexity?
- (Whether this can improve practical performance remains to be seen.)

What are our contributions? A TR method that has

- global and quadratic local convergence guarantees;
- a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$.

How is this achieved?

- new step acceptance criteria;
- new mechanism for rejecting a step while expanding the TR radius;
- new updates that may involve sublinear TR radius decrease.

Goals and contributions

What are our goals in this work?

- Question: Can we design a TR method with improved complexity?
- (Whether this can improve practical performance remains to be seen.)

What are our contributions? A TR method that has

- global and quadratic local convergence guarantees;
- a worst-case iteration complexity of $\mathcal{O}(\epsilon^{-3/2})$.

How is this achieved?

- new step acceptance criteria;
- new mechanism for rejecting a step while expanding the TR radius;
- new updates that may involve sublinear TR radius decrease.

We discuss three algorithms:

- TTR: "Traditional" Trust Region algorithm
- ARC: Adaptive Regularisation algorithm using Cubics
  - Cartis, Gould, & Toint (2011)
- TRACE: Trust Region Algorithm with Contractions and Expansions
  - Curtis, Robinson, & Samadi (2014)

Outline

Algorithm basics

TTR
1: Solve to compute s_k: $\min_{s\in\mathbb{R}^n} \; q_k(s) := f_k + g_k^T s + \frac{1}{2} s^T H_k s$ s.t. $\|s\|_2 \leq \delta_k$ (dual: λ_k)
2: Compute ratio: $\rho_k^q \leftarrow \frac{f_k - f(x_k+s_k)}{f_k - q_k(s_k)}$
3: Update radius: $\rho_k^q \geq \eta$: accept and $\delta_k \nearrow$ $\rho_k^q < \eta$: reject and $\delta_k \searrow$

ARC
1: Solve to compute s_k: $\min_{s\in\mathbb{R}^n} \; c_k(s) := f_k + g_k^T s + \frac{1}{2} s^T H_k s + \frac{1}{3}\sigma_k \|s\|_2^3$
2: Compute ratio: $\rho_k^c \leftarrow \frac{f_k - f(x_k+s_k)}{f_k - c_k(s_k)}$
3: Update regularization: $\rho_k^c \geq \eta$: accept and $\sigma_k \searrow$ $\rho_k^c < \eta$: reject and $\sigma_k \nearrow$

Algorithm basics: Subproblem solution correspondence

TTR		ARC
1: Solve to compute s_k: $\min_{s\in\mathbb{R}^n} \; q_k(s) := f_k + g_k^T s + \frac{1}{2} s^T H_k s$ s.t. $\|s\|_2 \leq \delta_k$ (dual: λ_k)	$\sigma_k = \frac{\lambda_k}{\delta_k}$ →	1: Solve to compute s_k: $\min_{s\in\mathbb{R}^n} \; c_k(s) := f_k + g_k^T s + \frac{1}{2} s^T H_k s + \frac{1}{3}\sigma_k \|s\|_2^3$
2: Compute ratio: $\rho_k^q \leftarrow \frac{f_k - f(x_k+s_k)}{f_k - q_k(s_k)}$	← $\delta_k = \|s_k\|_2$	2: Compute ratio: $\rho_k^c \leftarrow \frac{f_k - f(x_k+s_k)}{f_k - c_k(s_k)}$
3: Update radius: $\rho_k^q \geq \eta$: accept and $\delta_k \nearrow$ $\rho_k^q < \eta$: reject and $\delta_k \searrow$		3: Update regularization: $\rho_k^c \geq \eta$: accept and $\sigma_k \searrow$ $\rho_k^c < \eta$: reject and $\sigma_k \nearrow$

Discussion

What are the similarities?

- algorithmic structures are almost identical
- one-to-one correspondence (essentially) between subproblem solutions

What are the key differences?

- step acceptance criteria
- trust region vs. regularization coefficient updates

Discussion

What are the similarities?

- algorithmic structures are almost identical
- one-to-one correspondence (essentially) between subproblem solutions

What are the key differences?

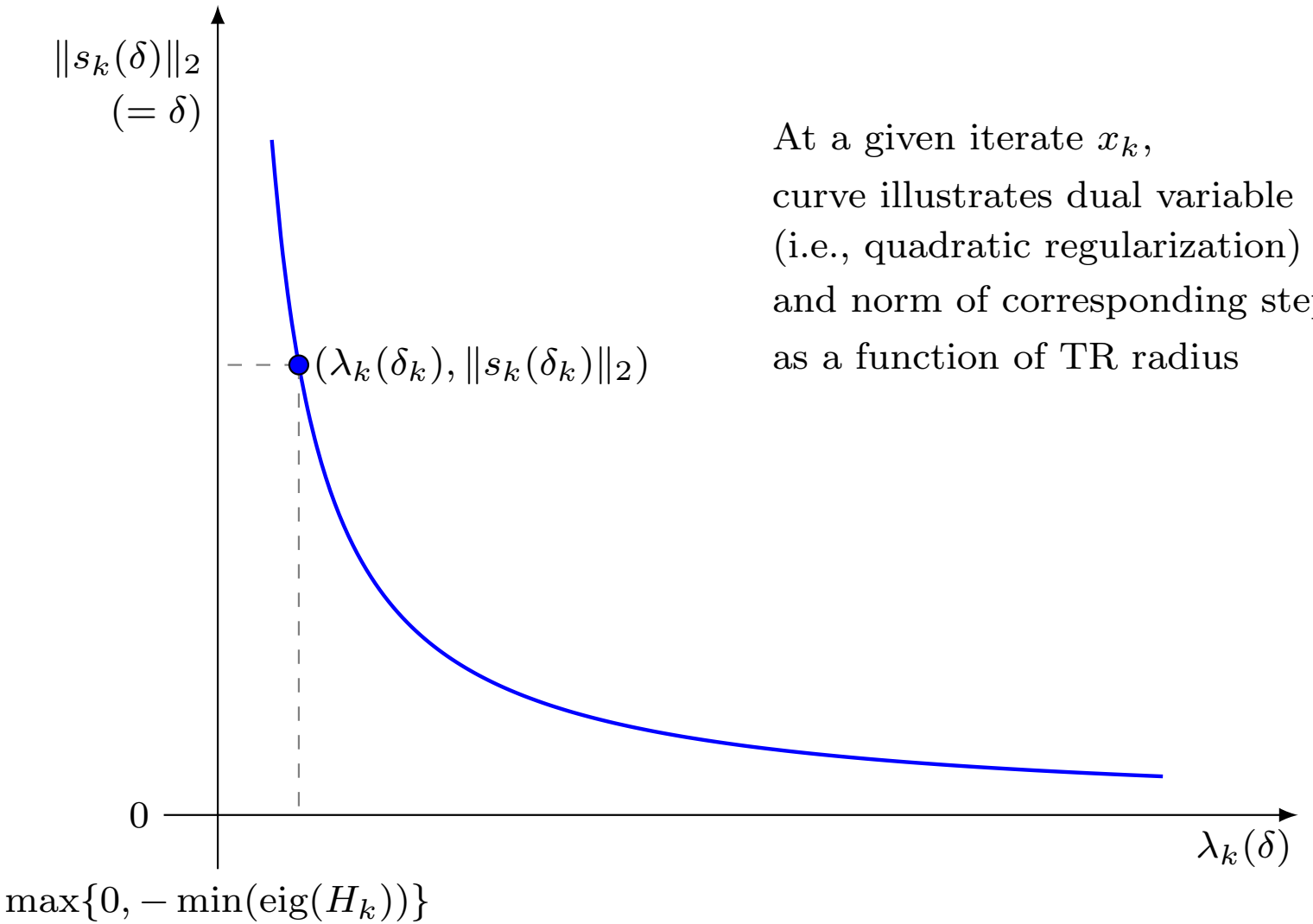
- step acceptance criteria
- trust region vs. regularization coefficient updates

Recall that a solution s_k of the TR subproblem is also a solution of

$$\min_{s \in \mathbb{R}^n} \; f_k + g_k^T s + \tfrac{1}{2} s^T (H_k + \lambda_k I) s,$$

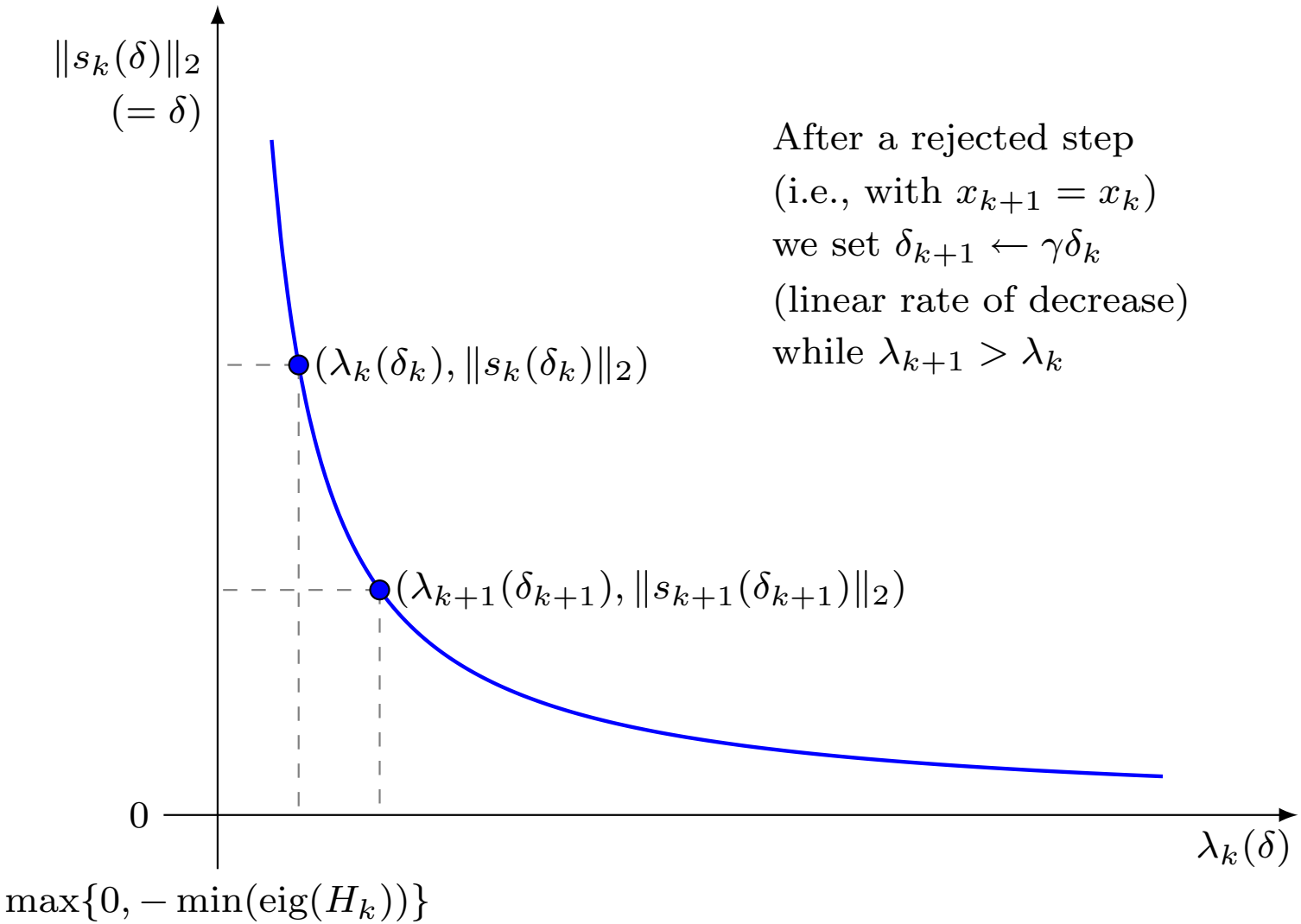
so the dual variable λ_k can be viewed as a quadratic regularization coefficient.

Regularization/stepsize trade-off: TTR



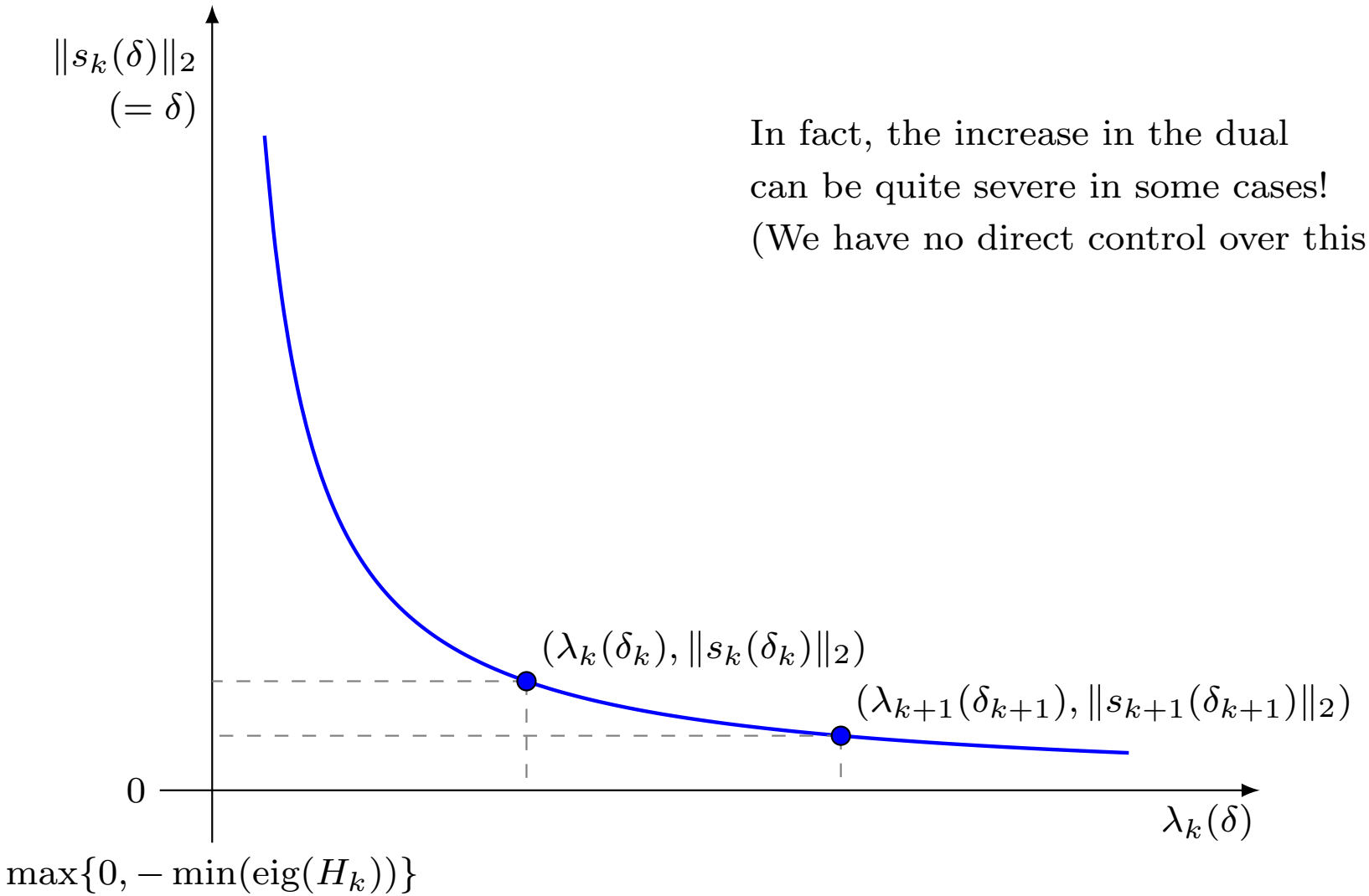
At a given iterate x_k,
curve illustrates dual variable
(i.e., quadratic regularization)
and norm of corresponding step
as a function of TR radius

Regularization/stepsize trade-off: TTR



After a rejected step
(i.e., with $x_{k+1} = x_k$)
we set $\delta_{k+1} \leftarrow \gamma\delta_k$
(linear rate of decrease)
while $\lambda_{k+1} > \lambda_k$

Regularization/stepsize trade-off: TTR



In fact, the increase in the dual can be quite severe in some cases! (We have no direct control over this.)

Intuition, please!

Intuitively, what is so important about $\frac{\lambda_k}{\|s_k\|_2} = \frac{\lambda_k}{\delta_k}$?

- Large δ_k implies s_k may not yield objective decrease.
- Small δ_k prohibits long steps.
- Small λ_k suggests the TR is not restricting us too much.
- Large λ_k suggests more objective decrease is possible.

So what is so bad (for complexity's sake) with the following?

$$\frac{\lambda_k}{\delta_k} \approx 0 \quad \text{and} \quad \frac{\lambda_{k+1}}{\delta_{k+1}} \gg 0.$$

It's that we may go from a

- large, but unproductive step to a
- productive, but (too) short step!

ARC magic

So what's the magic of ARC?

- It's not the types of steps you compute (since TR subproblem gives the same).
- It's that a simple update for σ_k gives a good regularization/stepsize balance.

ARC magic

So what's the magic of ARC?

- It's not the types of steps you compute (since TR subproblem gives the same).
- It's that a simple update for σ_k gives a good regularization/stepsize balance.

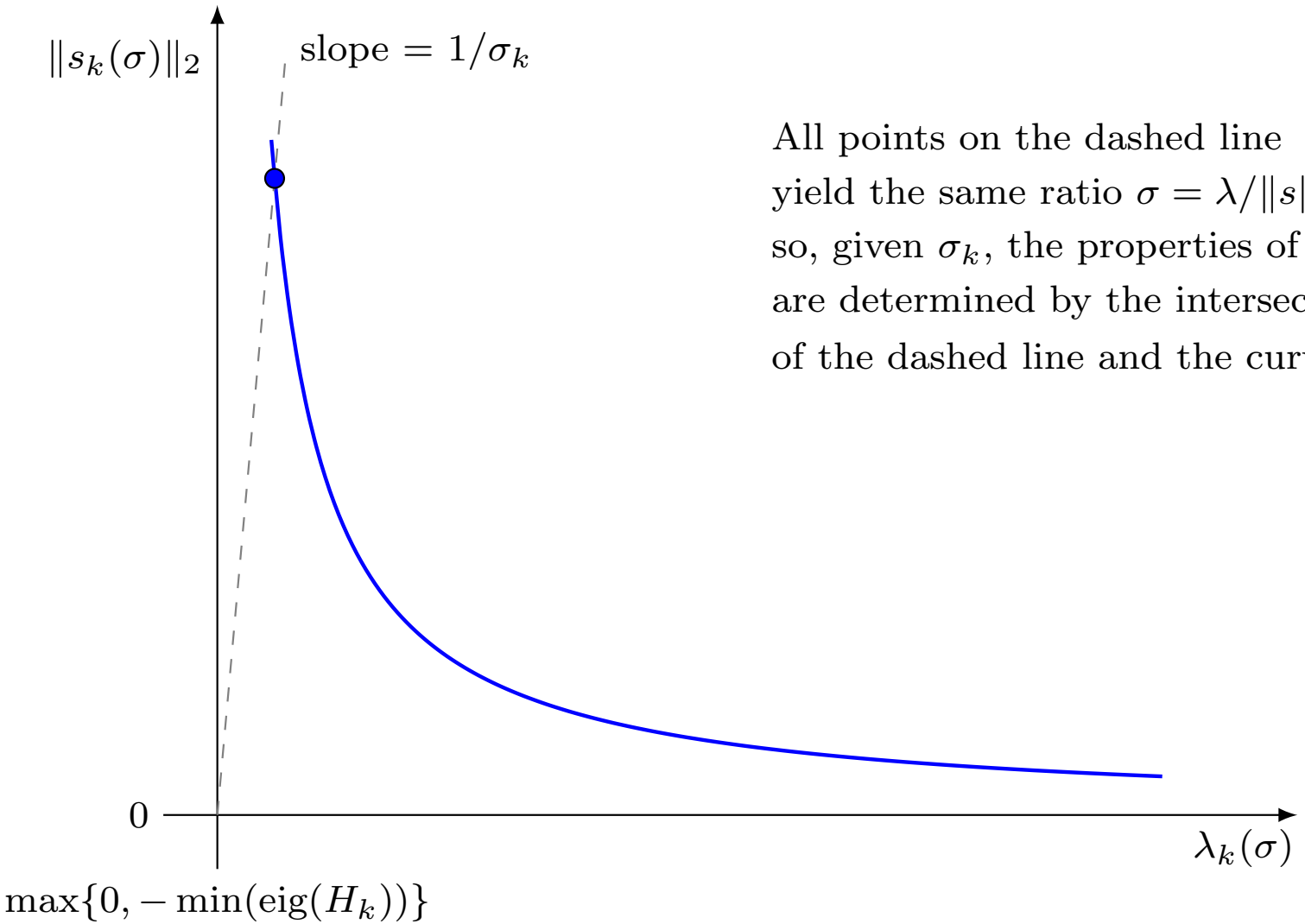
In ARC, accepted steps satisfy

$$f_k - f_{k+1} \geq c_1 \sigma_{\min} \|s_k\|_2^3 \quad \text{and} \quad \|s_k\|_2 \geq \left(\frac{c_2}{\sigma_{\max} + c_3} \right)^2 \|g_{k+1}\|_2^{1/2}.$$

Moreover, restricting $\{\sigma_k\}$ to be bounded below and proving that it remains bounded above ensures that, at any point, the number of rejected steps that can occur consecutively is bounded above by a constant (independent of k).

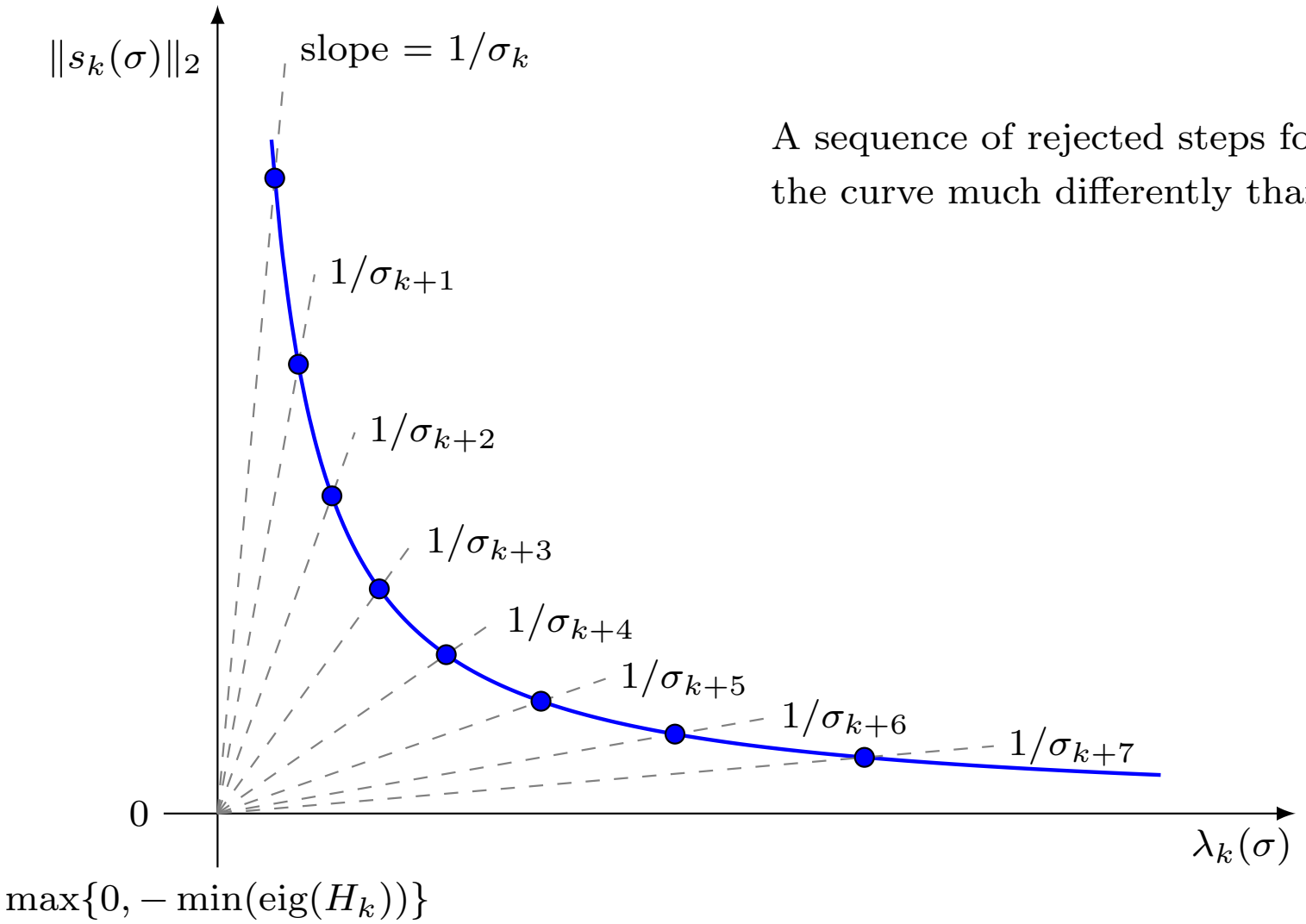
- Important to note that ARC always has the regularization "on."

Regularization/stepsize trade-off: ARC



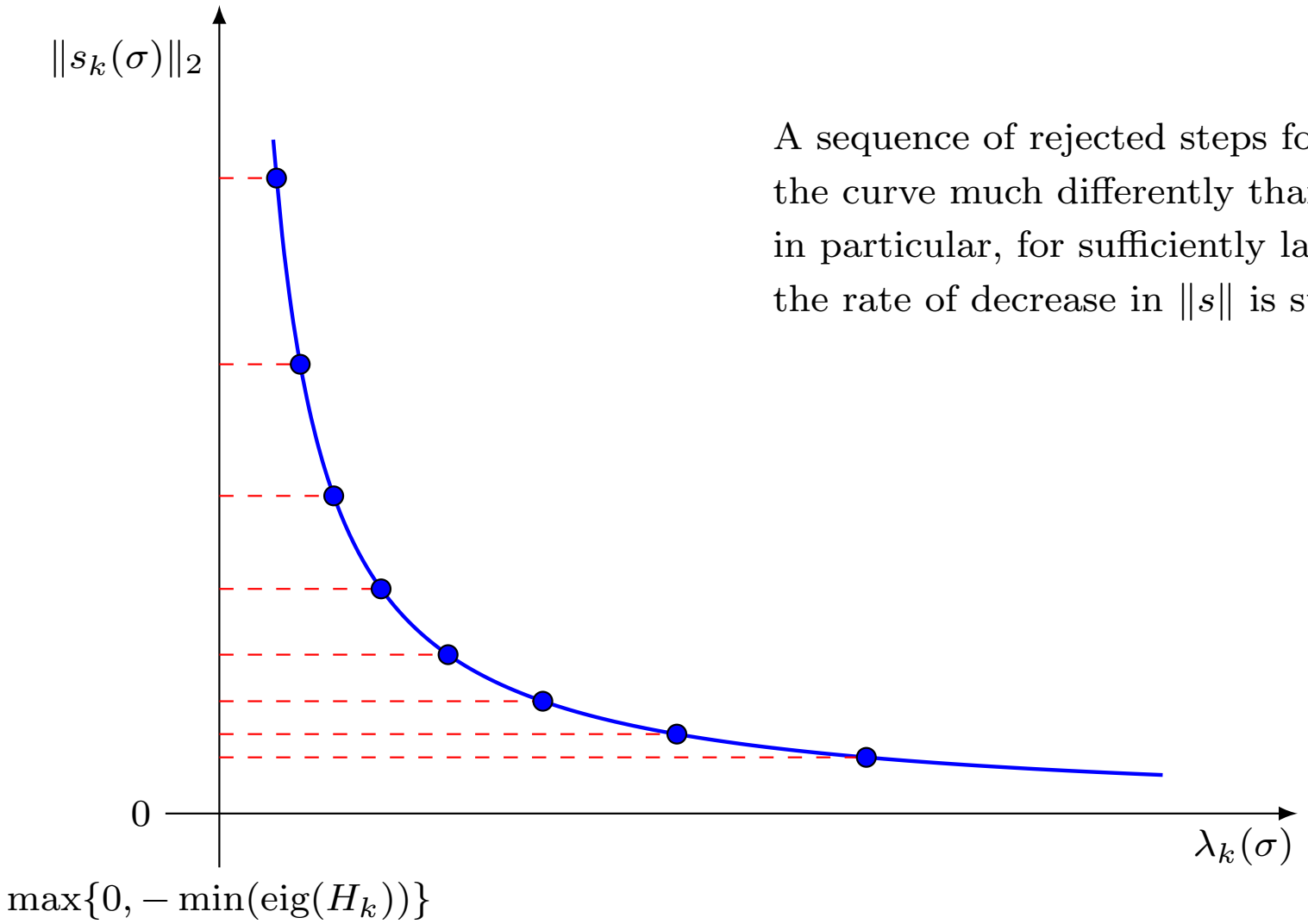
All points on the dashed line yield the same ratio $\sigma = \lambda/\|s\|_2$ so, given σ_k, the properties of s_k are determined by the intersection of the dashed line and the curve

Regularization/stepsize trade-off: ARC



A sequence of rejected steps follow the curve much differently than TTR;

Regularization/stepsize trade-off: ARC



A sequence of rejected steps follow the curve much differently than TTR; in particular, for sufficiently large σ, the rate of decrease in $\|s\|$ is sublinear

Outline

From TTR to TRACE

TRACE involves three key modifications of TTR.

1: Different step acceptance ratio

2: New expansion step: May reject step while increasing TR radius

3: New contraction procedure: Explicit or implicit (through update of λ)

Step acceptance ratio

1: Different step acceptance ratio

$$\text{TTR: } \rho_k^q = \frac{f_k - f(x_k + s_k)}{f_k - q_k(s_k)} \quad \Rightarrow \quad \text{TRACE: } \rho_k = \frac{f_k - f(x_k + s_k)}{\|s_k\|_2^3}$$

Motivations:

- With second-order model, error is third-order.
- Recall the first guarantee of accepted steps in ARC:

$$f_k - f_{k+1} \geq c_1 \sigma_{\min} \|s_k\|_2^3.$$

Expansion steps

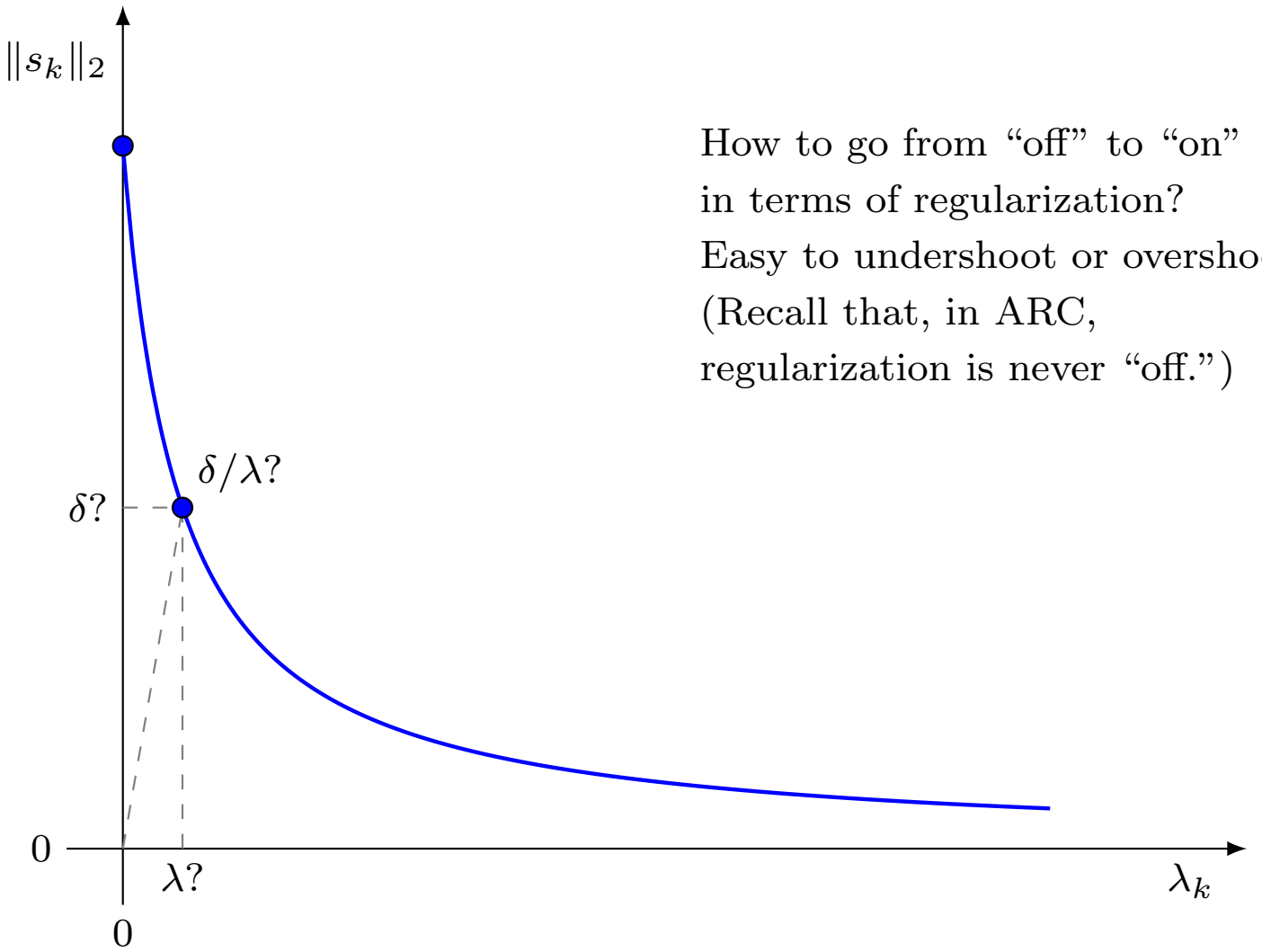
2: New expansion step: May reject step while increasing TR radius

- We define a monotonically increasing sequence $\{\sigma_k\}$.
- (Plays a similar theoretical role as the regularization coefficients in ARC.)
- If objective decrease is good, but dual suggests more decrease is possible, i.e.,

$$\rho_k \geq \eta \quad \text{but} \quad \lambda_k > \sigma_k \|s_k\|_2,$$

  then reject the step and increase the TR radius to allow more decrease.
- With $\delta_{k+1} \leftarrow \lambda_k/\sigma_k$, need at most one expansion between accepted steps.

Regularization/stepsize trade-off: "Off" to "on"



How to go from "off" to "on" in terms of regularization?
Easy to undershoot or overshoot!
(Recall that, in ARC, regularization is never "off.")

Contraction steps

3: New contraction procedure: Explicit or implicit (through update of λ)

$\lambda_k < \underline{\sigma}\|s_k\|_2$

- set $\lambda_{k+1} \leftarrow \lambda_k + (\underline{\sigma}\|g_k\|_2)^{1/2}$, or
- set $\lambda_{k+1} \in (\lambda_k, \lambda_k + (\underline{\sigma}\|g_k\|_2)^{1/2})$ so $\underline{\sigma} \leq \lambda_{k+1}/\|s_{k+1}\|_2 \leq \overline{\sigma}$

$\lambda_k \geq \underline{\sigma}\|s_k\|_2$

- set $\lambda_{k+1} \leftarrow \gamma_\lambda \lambda_k$ (with $\gamma_\lambda > 1$), or
- set $\delta_{k+1} \leftarrow \gamma_c \delta_k$ (with $\gamma_c \in (0, 1)$)

Update based on dual variable only requires a linear system solve!

$$(H_{k+1} + \lambda_{k+1} I)s = -g_{k+1}$$

Main algorithm

Algorithm 1 Trust Region Algorithm with Contraction and Expansion (TRACE)

Require: an acceptance constant $\eta \in \mathbb{R}_{++}$ with $0 < \eta < 1/2$
Require: update constants $\{\gamma_c, \gamma_e, \gamma_\lambda\} \subset \mathbb{R}_{++}$ with $0 < \gamma_c < 1 < \gamma_e$ and $\gamma_\lambda > 1$
Require: bound constants $\{\underline{\sigma}, \overline{\sigma}\} \subset \mathbb{R}_{++}$ with $0 < \underline{\sigma} \leq \overline{\sigma}$

procedure TRACE
  choose $x_0 \in \mathbb{R}^n$, $\{\delta_0, \Delta_0\} \subset \mathbb{R}_{++}$ with $\delta_0 \leq \Delta_0$, and $\sigma_0 \in \mathbb{R}_{++}$ with $\sigma_0 \geq \underline{\sigma}$
  compute (s_0, λ_0) by TR subproblem, then compute ρ_0
  for $k = 0, 1, 2, \ldots$ **do**
    if $\rho_k \geq \eta$ and either $\lambda_k \leq \sigma_k \|s_k\|_2$ or $\|s_k\|_2 = \Delta_k$ **then**
      set $x_{k+1} \leftarrow x_k + s_k$
      set $\Delta_{k+1} \leftarrow \max\{\Delta_k, \gamma_e \|s_k\|_2\}$
      set $\delta_{k+1} \leftarrow \min\{\Delta_{k+1}, \max\{\delta_k, \gamma_e \|s_k\|_2\}\}$
      set $\sigma_{k+1} \leftarrow \max\{\sigma_k, \lambda_k / \|s_k\|_2\}$
    else if $\rho_k < \eta$ **then**
      set $x_{k+1} \leftarrow x_k$
      set $\Delta_{k+1} \leftarrow \Delta_k$
      set $\delta_{k+1} \leftarrow \text{contract}(x_k, \delta_k, \sigma_k, s_k, \lambda_k)$
    else (i.e., **if** $\rho_k \geq \eta$, $\lambda_k > \sigma_k \|s_k\|_2$, and $\|s_k\|_2 < \Delta_k$)
      set $x_{k+1} \leftarrow x_k$
      set $\Delta_{k+1} \leftarrow \Delta_k$
      set $\delta_{k+1} \leftarrow \min\{\Delta_{k+1}, \lambda_k / \sigma_k\}$
      set $\sigma_{k+1} \leftarrow \sigma_k$
    compute (s_{k+1}, λ_{k+1}) by TR subproblem, then compute ρ_{k+1}
    if $\rho_k < \eta$ **then**
      set $\sigma_{k+1} \leftarrow \max\{\sigma_k, \lambda_{k+1} / \|s_{k+1}\|_2\}$

Contraction subroutine

Algorithm 2 Trust Region Contraction Subroutine (CONTRACT)

procedure CONTRACT($x_k, \delta_k, \sigma_k, s_k, \lambda_k$)
  if $\lambda_k < \underline{\sigma}\|s_k\|_2$ **then**
    set $\lambda \leftarrow \lambda_k + (\underline{\sigma}\|g_k\|_2)^{1/2}$
    set s as the solution of $(H_k + \lambda I)s = -g_k$
    set $\delta \leftarrow \|s\|_2$
    if $\lambda/\delta \leq \overline{\sigma}$ **then**
      return $\delta_{k+1} \leftarrow \delta$
    else
      compute $\hat{\lambda} \in (\lambda_k, \lambda)$ so $(H_k + \hat{\lambda} I)\hat{s} = -g_k$ yields $\underline{\sigma} \leq \hat{\lambda}/\|\hat{s}\|_2 \leq \overline{\sigma}$
      set $\hat{\delta} \leftarrow \|\hat{s}\|_2$
      return $\delta_{k+1} \leftarrow \hat{\delta}$
  else (i.e., **if** $\lambda_k \geq \underline{\sigma}\|s_k\|_2$)
    set $\lambda \leftarrow \gamma_\lambda \lambda_k$
    set s as the solution of $(H_k + \lambda I)s = -g_k$
    set $\delta \leftarrow \|s\|_2$
    if $\delta \geq \gamma_c\|s_k\|_2$ **then**
      return $\delta_{k+1} \leftarrow \delta$
    else
      return $\delta_{k+1} \leftarrow \gamma_c\|s_k\|_2$

Global and local quadratic convergence

Assumption 1

- *f twice continuously differentiable and bounded below by $f_{\min}$*
- *g Lipschitz continuous in open convex set containing $\{x_k\}$ and $\{x_k + s_k\}$*
- *$\{g_k\}$ has nonzero elements and bounded above*
- *$\{H_k\}$ bounded above*

Theorem 2

$\|g_k\|_2 \to 0$

Assumption 3 (in addition to Assumption 1)

$\{x_k\}_{\mathcal{S}} \to x_*$ around which H is positive definite and locally Lipschitz

Theorem 4

$\{x_k\} \to x_*$ with $g(x_*) = 0$ and, for sufficiently large k,

$$\|g_{k+1}\|_2 = \mathcal{O}(\|g_k\|_2^2) \quad \textit{and} \quad \|x_{k+1} - x_*\|_2 = \mathcal{O}(\|x_k - x_*\|_2^2)$$

Worst-case iteration complexity

Assumption 5 (in addition to Assumption 1)

H Lipschitz continuous in open convex set containing $\{x_k\}$ and $\{x_k + s_k\}$

Lemma 6

- $f_k - f_{k+1} \geq \eta \|s_k\|_2^3$ *for all accepted steps*
- $\{\sigma_k\}$ *bounded by* $\sigma_{\max} > 0$
- $\|s_k\|_2 \geq (H_{Lip} + \sigma_{\max})^{-1/2} \|g_{k+1}\|_2^{1/2}$

Theorem 7

Number of accepted steps prior to $\|g_k\|_2 \leq \epsilon$ *at most*

$$1 + \left\lfloor \frac{f_0 - f_{\min}}{\eta \Delta_0^3} \right\rfloor + \left\lfloor \frac{f_0 - f_{\min}}{\eta (H_{Lip} + \sigma_{\max})^{-3/2}} \epsilon^{-3/2} \right\rfloor .$$

Furthermore, number of rejected steps between consecutive accepted steps bounded by a constant independent of k. Overall, worst-case iteration complexity $\mathcal{O}(\epsilon^{-3/2})$.

Outline

Contributions

Question: Can we design a TR method with improved complexity?

- Yes, TRACE achieves the same convergence/complexity guarantees as ARC
- New step acceptance criteria
- New mechanism for rejecting a step while expanding the TR radius
- New updates that may involve sublinear TR radius decrease

F. E. Curtis, D. P. Robinson, and M. Samadi, "A Trust Region Algorithm with a Worst-Case Iteration Complexity of $\mathcal{O}(\epsilon^{-3/2})$ for Nonconvex Optimization," COR@L Laboratory, Department of ISE, Lehigh University, 14T-009, 2014.

Next question: Does TRACE perform better than TTR and/or ARC in practice?

- Numerical experiments forthcoming
- Preliminary results show it is at least competitive (not surprising)
- Note that an iteration of TRACE may only need a linear system solve!
- One may imagine algorithms like TRACE and ARC that achieve the same convergence/complexity guarantees and never fully solve a subproblem...