Numerical Optimization

Lecture Notes #7

Trust-Region Methods: Introduction / Cauchy Point

Peter Blomgren,
⟨blomgren.peter@gmail.com⟩

Department of Mathematics and Statistics
Dynamical Systems Group
Computational Sciences Research Center
San Diego State University
San Diego, CA 92182-7720

http://terminus.sdsu.edu/

Fall 2018

Outline

Quick Recap: Last Time — Step Length Selection 1 of 2

We improved on **Backtracking Line Search** — introducing interpolation based alternatives for finding a new trial step length when the old one is rejected.

Interpolation #1: (No extra gradient evaluations: $\nabla f(\bar{\mathbf{x}}_k + \alpha\bar{\mathbf{p}}_k)$) — First use the optimizer α_1 of the quadratic model interpolating $\Phi(0)$, $\Phi'(0)$, and $\Phi(\alpha_0)$. If that fails, try the optimizer α_2 of the cubic model interpolating $\Phi(0)$, $\Phi'(0)$, $\Phi(\alpha_0)$, and $\Phi(\alpha_1)$. If α_2 fails, keep building similar cubic models.

Interpolation #2: (Evaluations of $\nabla f(\bar{\mathbf{x}}_k + \alpha\bar{\mathbf{p}}_k)$ if not excessively expensive) — First use the optimizer α_1 of the cubic model interpolating $\Phi(0)$, $\Phi'(0)$, $\Phi(\alpha_0)$, and $\Phi'(\alpha_0)$ (Hermite polynomial). If that fails, try the optimizer α_2 of the cubic model interpolating $\Phi(\alpha_0)$, $\Phi'(\alpha_0)$, $\Phi(\alpha_1)$, and $\Phi'(\alpha_1)$. If α_2 fails, keep building similar cubic models.

Quick Recap: Last Time — Step Length Selection 2 of 2

Strategies for the initial step α_0. — Newton and quasi-Newton have a sense of scale, use $\alpha_0 = 1$.

For other search directions (lacking a sense of scale) —

Strategy #1: Assume the rate of change in the current iteration will be the same as in the previous iteration.

$$\alpha_0^{[k]} = \alpha^{[k-1]} \frac{\bar{\mathbf{p}}_{k-1}^T \nabla f(\bar{\mathbf{x}}_{k-1})}{\bar{\mathbf{p}}_k^T \nabla f(\bar{\mathbf{x}}_k)}.$$

Strategy #2: Use the minimizer of the quadratic interpolant of $f(\bar{\mathbf{x}}_{k-1})$, $f(\bar{\mathbf{x}}_k)$, and $\bar{\mathbf{p}}_k^T \nabla f(\bar{\mathbf{x}}_k)$.

$$\alpha_0^{[k]} = \frac{2[f(\bar{\mathbf{x}}_k) - f(\bar{\mathbf{x}}_{k-1})]}{\bar{\mathbf{p}}_k^T \nabla f(\bar{\mathbf{x}}_k)}.$$

Finally, we looked at a full implementation of a Line Search algorithm yielding steps satisfying the **Strong Wolfe conditions**.

Lookahead: This Time — Trust Region Methods

The Idea:

- Build a, usually quadratic, model around the current point $\bar{\mathbf{x}}_k$.
- Along with the model we define a region in which we trust the model to be a good representation of the objective f.
- Let the next iterate $\bar{\mathbf{x}}^*_{k+1}$ be the (approximate) optimizer of the **model** in the **"trust region."**
- The step α and the direction $\bar{\mathbf{p}}$ are selected simultaneously.
- If the new point $\bar{\mathbf{x}}^*_{k+1}$ is not acceptable, we reduce the size of the trust region, and repeat.

Trust Region Methods — Introduction

Clearly, we want our algorithm to have some **"memory"** of what happened in the past.

— If the first point was accepted in the previous iteration, we may want to increase the size of the trust region in the current iteration. This way, we can allow large steps when we have a good model of the objective.

— If, on the other hand, many reductions of the trust region were required in the previous iteration, then we probably do not have a very good model; hence we start with a small trust region in the current iteration.

Trust Region Methods — The Quadratic Model

The **"model"** is based on (surprise, surprise!) the Taylor expansion of the objective f at the current point $\bar{\mathbf{x}}_k$ —

$$m_k(\bar{\mathbf{p}}) = f(\bar{\mathbf{x}}_k) + \bar{\mathbf{p}}^T \nabla f(\bar{\mathbf{x}}_k) + \frac{1}{2}\bar{\mathbf{p}}^T B_k \bar{\mathbf{p}},$$

where B_k is a symmetric matrix.

We see that the first two terms agree with the Taylor expansion, and that if $B_k = \nabla^2 f(\bar{\mathbf{x}}_k)$ the model agrees with the first three terms of the expansion.

In the first case $B_k \neq \nabla^2 f(\bar{\mathbf{x}}_k)$ the **error** in the model **is quadratic** in $\bar{\mathbf{p}}$, *i.e.*

$$\|m_k(\bar{\mathbf{p}}) - f(\bar{\mathbf{x}}_k + \bar{\mathbf{p}})\| \sim \mathcal{O}\left(\|\bar{\mathbf{p}}\|^2\right),$$

and in the second case it is **cubic**

$$\|m_k(\bar{\mathbf{p}}) - f(\bar{\mathbf{x}}_k + \bar{\mathbf{p}})\| \sim \mathcal{O}\left(\|\bar{\mathbf{p}}\|^3\right).$$

Trust Region Methods — The Quadratic Model

When the first three terms of the quadratic model agrees with the Taylor expansion, *i.e.* $B_k = \nabla^2 f(\bar{\mathbf{x}}_k)$, the algorithm is called **the trust-region Newton Method**.

In general, all we need to assume about the matrices B_k is that they are symmetric, and $\|B_k\| < M$ (uniformly bounded).

The locally constrained **trust region problem** is

$$\min_{\bar{\mathbf{p}} \in T_k} m_k(\bar{\mathbf{p}}) = \min_{\bar{\mathbf{p}} \in T_k} \left[f(\bar{\mathbf{x}}_k) + \bar{\mathbf{p}}^T \nabla f(\bar{\mathbf{x}}_k) + \frac{1}{2}\bar{\mathbf{p}}^T B_k \bar{\mathbf{p}} \right],$$

where T_k is the trust region.

Note: If B_k is positive definite, and $\bar{\mathbf{p}}_k^B = -B_k^{-1}\nabla f(\bar{\mathbf{x}}_k) \in T_k$, then the **full step** is allowed.

Illustration: The Quadratic Model

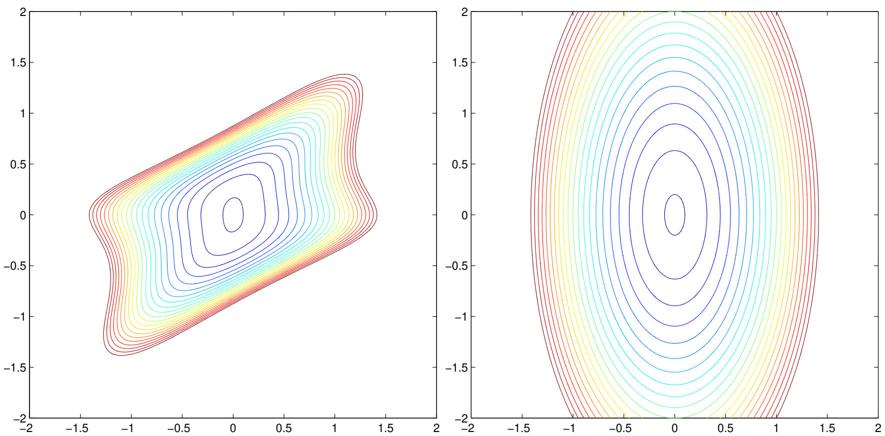


Figure: The picture to the left shows the contour lines of the objective $f(\bar{\mathbf{x}}) = x_1^2 + x_2^2/4 + 4(x_1 - x_2)^2 \cdot \sin^2(x_2)$ and the picture to the right shows the same contour lines for the model $m_k(\bar{\mathbf{p}})$ whose first three terms agree with the Taylor expansion of the objective.

Illustration: The Quadratic Model

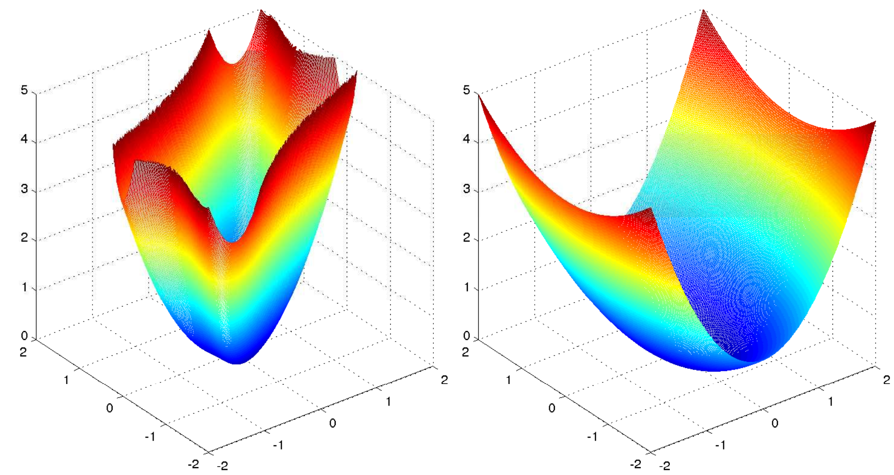


Figure: The picture to the left mesh plot of the objective $f(\bar{\mathbf{x}}) = x_1^2 + x_2^2/4 + 4(x_1 - x_2)^2 \cdot \sin^2(x_2)$ and the picture to the right shows the mesh plot for the model $m_k(\bar{\mathbf{p}})$ whose first three terms agree with the Taylor expansion of the objective.

Illustration: The Quadratic Model

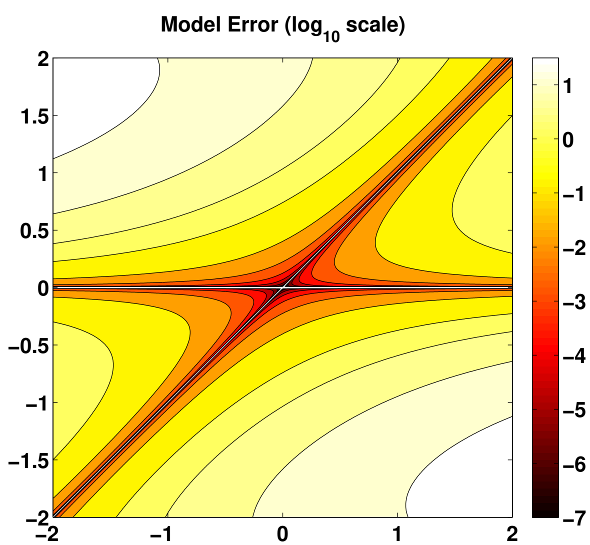


Figure: The model error $\|m_k(\bar{\mathbf{p}}) - f(\bar{\mathbf{x}} + \bar{\mathbf{p}})\|$, at $\bar{\mathbf{x}} = \bar{\mathbf{0}}$.

Trust Region Methods — The Trust Region

Usually, the **Trust Region** T_k is defined by its radius Δ_k:

$$T_k = \{\bar{\mathbf{x}} \in \mathbb{R}^n \, : \, \|\bar{\mathbf{x}}\| \leq \Delta_k\}.$$

Note: If $\|B_k^{-1}\nabla f(\bar{\mathbf{x}}_k)\| > \Delta_k$ then the full step is not allowed, and we must find the optimal solution to the (locally) **constrained problem**

$$\min_{\|\bar{\mathbf{p}}\| \leq \Delta_k} \left[f(\bar{\mathbf{x}}_k) + \bar{\mathbf{p}}^T \nabla f(\bar{\mathbf{x}}_k) + \frac{1}{2}\bar{\mathbf{p}}^T B_k \bar{\mathbf{p}} \right].$$

The solution is not immediately obvious.

In practice we only need an approximate solution which yields sufficient decrease in the objective.

The Base-Line Trust Region Algorithm

As in the line search case, we start out by formulating a working Trust-Region algorithm. Then we study the different components of the algorithm, and substitute more clever solutions to the various subproblems.

First, we define a ratio measuring the success of a step —

Definition

Given a step $\bar{\mathbf{p}}_k$ we define the ratio

$$\rho_k = \frac{\text{actual reduction}}{\text{predicted reduction}} = \frac{f(\bar{\mathbf{x}}_k) - f(\bar{\mathbf{x}}_k + \bar{\mathbf{p}}_k)}{m_k(0) - m_k(\bar{\mathbf{p}}_k)}$$

The predicted reduction is always non-negative (the step $\bar{\mathbf{p}}_k = 0$ is part of the trust region). Thus if $\rho_k < 0$ the step must be rejected (since $f(\bar{\mathbf{x}}_k + \bar{\mathbf{p}}_k) > f(\bar{\mathbf{x}}_k)$).

The Base-Line Trust Region Algorithm

If $\rho_k < 0$	We shrink the size of the trust region.
If $\rho_k \approx 0$	Then we shrink the size of the trust region.
If $\rho_k \approx 1$	Then the model is in good agreement with the objective; in this case it is (probably) safe to expand the trust region for the next iteration.

Otherwise we keep the size of the trust region.

The Base-Line Trust Region Algorithm

Algorithm: Trust Region

```
[ 1] Set k = 1, Δ̂ > 0, Δ_0 ∈ (0, Δ̂), and η ∈ (0, 1/4)
[ 2] While optimality condition not satisfied
[ 3]   Get p̄_k (approximate solution)
[ 4]   Evaluate ρ_k = (f(x̄_k) − f(x̄_k + p̄_k)) / (m_k(0) − m_k(p̄_k))
[ 5]   if ρ_k < 1/4
[ 6]     Δ_{k+1} = (1/4) Δ_k
[ 7]   else
[ 8]     if ρ_k > 3/4 and ||p̄_k|| = Δ_k
[ 9]       Δ_{k+1} = min(2Δ_k, Δ̂)
[10]     else
[11]       Δ_{k+1} = Δ_k
[12]     endif
[13]   endif
[14]   if ρ_k > η
[15]     x̄_{k+1} = x̄_k + p̄_k
[16]   else
[17]     x̄_{k+1} = x̄_k
[18]   endif
[19]   k = k + 1
[20] End-While
```


The Base-Line Trust Region Algorithm: Missing Parts

Clearly, in order to make use of this "algorithm" we must turn our attention to the solution of

$$\min_{\|\bar{\mathbf{p}}\| \le \Delta_k} \left[f(\bar{\mathbf{x}}_k) + \bar{\mathbf{p}}^T \nabla f(\bar{\mathbf{x}}_k) + \frac{1}{2} \bar{\mathbf{p}}^T B_k \bar{\mathbf{p}} \right]. \qquad (\textbf{Get } \bar{\mathbf{p}}_k)$$

We look at the easiest approximation:

— the **Cauchy point**, the minimizer of $m_k(\bar{\mathbf{p}})$ in the steepest descent direction.

Then we study three improvements to the Cauchy point:

— **Dogleg method**; used when B_k is positive definite.

— **2-D Subspace Minimization**; can be used when B_k is indefinite.

— **Steihaug's Method**; appropriate when $B_k = \nabla^2 f(\bar{\mathbf{x}}_k)$ and this matrix is large and sparse (most entries are zeros.)

The Cauchy Point

For global convergence we can be quite sloppy in the minimization of the model $m_k(\bar{\mathbf{p}})$ — all we must require is **sufficient reduction** in the model. This is quantified in terms of the Cauchy point $\bar{\mathbf{p}}_k^c$ —

Algorithm: Cauchy Point Calculation

Find the minimizer for the linear model $l_k(\bar{\mathbf{p}}) = f(\bar{\mathbf{x}}_k) + \bar{\mathbf{p}}^T \nabla f(\bar{\mathbf{x}}_k)$

$$\bar{\mathbf{p}}_k^s = \arg\min_{\|\bar{\mathbf{p}}\| \le \Delta_k} \left[f(\bar{\mathbf{x}}_k) + \bar{\mathbf{p}}^T \nabla f(\bar{\mathbf{x}}_k) \right].$$

Let $\tau_k > 0$ be the scalar that minimizes $m_k(\tau \bar{\mathbf{p}}_k^s)$ subject to satisfying the trust-region constraint, *i.e.*

$$\tau_k = \arg\min_{\tau > 0} m_k(\tau \bar{\mathbf{p}}_k^s), \quad \text{such that} \;, \; \|\tau \bar{\mathbf{p}}_k^s\| \le \Delta_k.$$

Let $\bar{\mathbf{p}}_k^c = \tau_k \bar{\mathbf{p}}_k^s$. This is the Cauchy point.

The Cauchy Point — Explicit Expressions

We can write down some of the quantities explicitly, *e.g.*

$$\bar{\mathbf{p}}_k^s = -\Delta_k \frac{\nabla f(\bar{\mathbf{x}}_k)}{\|\nabla f(\bar{\mathbf{x}}_k)\|},$$

is the full step to the trust-region boundary.

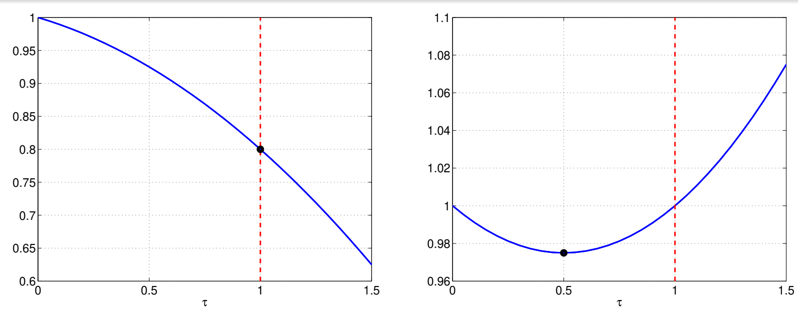
Case: $\nabla f(\bar{\mathbf{x}}_k)^T B_k \nabla f(\bar{\mathbf{x}}_k) \le 0$

$m_k(\tau \bar{\mathbf{p}}_k^s)$ decreases monotonically with τ, whenever $\nabla f(\bar{\mathbf{x}}_k) \neq 0$. Hence, τ_k is the largest τ which keeps satisfies the trust-region condition; by construction of $\bar{\mathbf{p}}_k^s$, this means $\tau_k = 1$.

Case: $\nabla f(\bar{\mathbf{x}}_k)^T B_k \nabla f(\bar{\mathbf{x}}_k) > 0$

$m_k(\tau \bar{\mathbf{p}}_k^s)$ is a convex quadratic in τ; hence τ_k is the smaller of the minimizer of the quadratic, or 1.

The Cauchy Point — Explicit Expressions



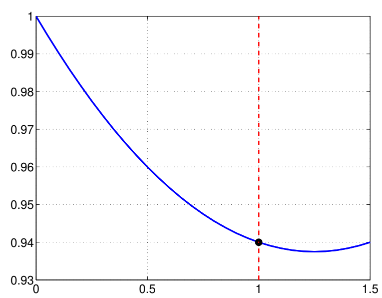


Figure: The three possible scenarios for selection of τ.

The Cauchy Point — Explicit Expressions

The unconstrained minimizer of the quadratic is

$$\tau_k^* = \frac{\|\nabla f(\bar{\mathbf{x}}_k)\|^3}{\Delta_k \nabla f(\bar{\mathbf{x}}_k)^T B_k \nabla f(\bar{\mathbf{x}}_k)}.$$

Hence we have, for the **Cauchy point**

$$\left\{ \begin{array}{lcl} \bar{\mathbf{p}}_k^c & = & -\tau_k \dfrac{\Delta_k}{\|\nabla f(\bar{\mathbf{x}}_k)\|} \nabla f(\bar{\mathbf{x}}_k) \\ \text{where} & & \\ \tau_k & = & \left\{ \begin{array}{ll} 1 & \text{if } \nabla f(\bar{\mathbf{x}}_k)^T B_k \nabla f(\bar{\mathbf{x}}_k) \le 0 \\ \min\left(1, \frac{\|\nabla f(\bar{\mathbf{x}}_k)\|^3}{\Delta_k \nabla f(\bar{\mathbf{x}}_k)^T B_k \nabla f(\bar{\mathbf{x}}_k)}\right) & \text{otherwise.} \end{array} \right. \end{array} \right.$$

The Cauchy point is cheap to calculate — no matrix inversions, or factorizations are required.

A trust-region method will be globally convergent if its steps $\bar{\mathbf{p}}_k$ give reductions in the models $m_k(\bar{\mathbf{p}})$ that is at least some fixed multiple of the decrease attained by the Cauchy point in each iteration.

The Cauchy Point — Are We Done?

The Cauchy point $\bar{\mathbf{p}}_k^c$ gives us sufficient reduction for global convergence and it is cheap-and-easy to compute. Is there any reason to look for other (approximate) solutions of

$$\underset{\|\bar{\mathbf{p}}\|\leq\Delta_k}{\arg\min}\left[f(\bar{\mathbf{x}}_k)+\bar{\mathbf{p}}^T\nabla f(\bar{\mathbf{x}}_k)+\frac{1}{2}\bar{\mathbf{p}}^T B_k\bar{\mathbf{p}}\right] \quad ???$$

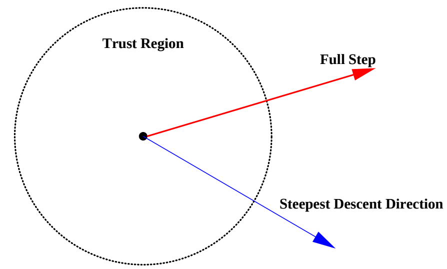
Well, yes. Using the Cauchy point as our step means that we have implemented the **Steepest Descent** method, with a particular step length. From previous discussion (and HW#1) we know that steepest descent converges slowly (linearly) even when the step length is chosen optimally.

$\therefore$ there is room for improvement (a.k.a. rotten-tomato-moment™.)

The Dogleg Method

Strategy: Dogleg
Method: Dogleg (for Trust-region).
Use When: The model Hessian B_k is positive definite.

At a point $\bar{\mathbf{x}}_k$ we have already looked at two steps — a step in the steepest descent direction, and the full step.



The Dogleg Method

The **full step** is given by the unconstrained minimum of the quadratic model

$$\bar{\mathbf{p}}_k^{\text{FS}} = -B_k^{-1}\nabla f(\bar{\mathbf{x}}_k).$$

The step in the **steepest descent direction** is given by the unconstrained minimum of the quadratic model along the steepest descent direction

$$\bar{\mathbf{p}}_k^U = -\frac{\nabla f(\bar{\mathbf{x}}_k)^T\nabla f(\bar{\mathbf{x}}_k)}{\nabla f(\bar{\mathbf{x}}_k)^T B_k\nabla f(\bar{\mathbf{x}}_k)}\nabla f(\bar{\mathbf{x}}_k).$$

When the trust region is small, the quadratic term is small, so the minimum of

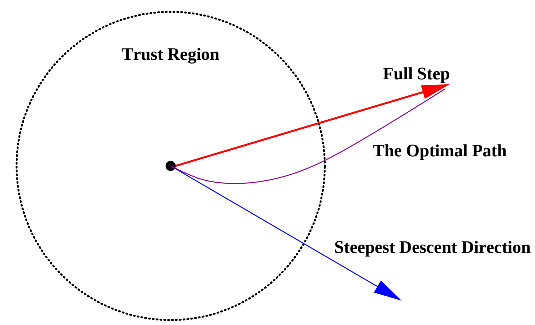
$$\underset{\|\bar{\mathbf{p}}\|\leq\Delta_k}{\arg\min}\left[f(\bar{\mathbf{x}}_k)+\bar{\mathbf{p}}^T\nabla f(\bar{\mathbf{x}}_k)+\frac{1}{2}\bar{\mathbf{p}}^T B_k\bar{\mathbf{p}}\right],$$

is achieved very close to the steepest descent direction.

The Dogleg Method

On the other hand, as the trust region gets larger ($\Delta_k \to \infty$) the optimum will move to the full step.

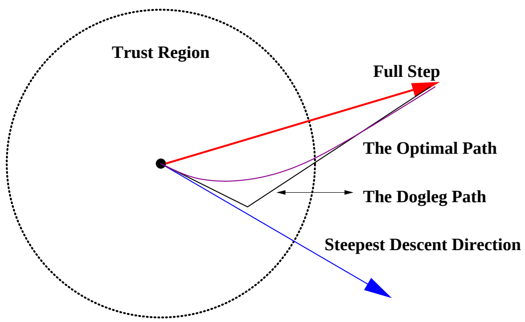
If we plot the optimum as a function of the size of the trust region, we get a smooth path:



The Dogleg Method

The idea of the dogleg method is to **(i)** approximate this path, since the analytical expression for it is quite expensive; and **(ii)** to optimize the model $m_k(\bar{\mathbf{p}})$ along the approximate path subject to the trust region constraint.

The approximate path is a line segment running from $\bar{\mathbf{0}}$ to $\bar{\mathbf{p}}_k^U$, connected to a second line segment running from $\bar{\mathbf{p}}_k^U$ to $\bar{\mathbf{p}}_k^{\text{FS}}$, something like



The Dogleg Method

Formally, the dogleg path can be described by one parameter τ

$$\tilde{\bar{p}}(\tau) = \begin{cases} \tau \bar{\mathbf{p}}_k^U & 0 \leq \tau \leq 1 \\ \bar{\mathbf{p}}_k^U + (\tau - 1)(\bar{\mathbf{p}}_k^{\text{FS}} - \bar{\mathbf{p}}_k^U) & 1 \leq \tau \leq 2 \end{cases}$$

The following result can be shown —

Lemma

Let B_k be positive definite, then

(i) $\|\tilde{\bar{p}}(\tau)\|$ *is an increasing function of* τ.

(ii) $m_k(\tilde{\bar{p}}(\tau))$ *is a decreasing function of* τ.

This means that the optimum along the dogleg path is achieved at the point where the path exits the trust-region (if it does), otherwise the full step is allowed and optimal.

The Dogleg Method

If the full step is not allowed, then the exit point for the dogleg path is given by the scalar quadratic equation

$$\left\| \bar{\mathbf{p}}_k^U + (\tau - 1)(\bar{\mathbf{p}}_k^{\text{FS}} - \bar{\mathbf{p}}_k^U) \right\|^2 = \Delta_k^2, \quad \tau \in [1, 2]$$

assuming that $\bar{\mathbf{p}}_k^U$ is allowable, otherwise the exit point is along the steepest descent path

$$\left\| \tau \bar{\mathbf{p}}_k^U \right\|^2 = \Delta_k^2, \quad \tau \in [0, 1].$$

Next time we look at dealing with indefinite model Hessians B_k...

Index