

Numerical Optimization

Lecture Notes #8 — Trust-Region Methods 2-D Subspace Minimization & Iterative “Nearly Exact” Methods

Fall 2024

Outline

- 1 Trust Region Methods
 - Recap
 - 2-D Subspace Minimization
- 2 Iterative “Nearly Exact” Solution to the Subproblem
 - Fundamentals...
 - Building a Scheme to Solve the Problem...

Quick Recap: Last Time — Trust-Region / Cauchy Point

- Introduction to Trust-region (TR) methods.
 - Use a quadratic model $m_k(\bar{\mathbf{p}})$ around the current point $\bar{\mathbf{x}}_k$; we trust the model in some neighborhood (\Rightarrow name of this class of methods).
 - Optimize the model $m_k(\bar{\mathbf{p}})$ to find the appropriate step $\bar{\mathbf{p}}_k$.
 - We developed a macro algorithm for expanding and contracting the size of the TR, depending on how well the model approximates the objective locally.
 - We spend quite some effort in finding the **Cauchy point**, which if selected in each iteration guarantees global convergence to a stationary point.
 - We realized that the Cauchy point was just a rediscovery of the steepest descent algorithm, with a particular (possibly sub-optimal) step size.
 - We introduced the first improvement over the Cauchy point, the Dogleg Method.

Improvement #1: The Dogleg Method (Review)

1 of 3

Method: Dogleg for Trust-region

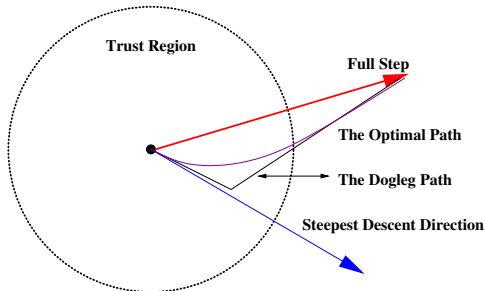
Use When: The model Hessian B_k is positive definite.

Figure: The dogleg path tries to approximate the optimal path, which describes the optimal solution to the trust-region sub-problem as a function of the trust-region radius. The dogleg step is simply the exit point of the dogleg path, or — in the case when it is feasible — the full step.

Improvement #1: The Dogleg Method (Review)

2 of 3

The **trust region sub-problem** is given by

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

The **full step** is 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

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

Improvement #1: The Dogleg Method (Review)

3 of 3

Algorithm: The Dogleg Step

If($\|\bar{\mathbf{p}}_k^U\| \geq \Delta_k$), then

$$\bar{\mathbf{p}}_k^{\text{DL}} = \Delta_k \cdot \bar{\mathbf{p}}_k^U / \|\bar{\mathbf{p}}_k^U\|,$$

elseif($\|\bar{\mathbf{p}}_k^{\text{FS}}\| \leq \Delta_k$), then

$$\bar{\mathbf{p}}_k^{\text{DL}} = \bar{\mathbf{p}}_k^{\text{FS}},$$

else

$$\bar{\mathbf{p}}_k^{\text{DL}} = \bar{\mathbf{p}}_k^U + (\tau^* - 1)(\bar{\mathbf{p}}_k^{\text{FS}} - \bar{\mathbf{p}}_k^U)$$

where $\tau^* \in [1, 2]$ so that $\|\bar{\mathbf{p}}_k^U + (\tau^* - 1)(\bar{\mathbf{p}}_k^{\text{FS}} - \bar{\mathbf{p}}_k^U)\|^2 = \Delta_k^2$

end

Next we look at improvements of the dogleg method.

Lookahead: Improvements to the Cauchy Point / Dogleg Method

We seek other improvements to the Cauchy point, with the goal of developing trust region methods with better convergence properties.

In this lecture, we look at:

2-D Subspace Method:

Another simplified model for the optimal solution to the model problem in the trust-region. Here, the search for an optimal solution of the model is restricted to a plane ($\mathbb{R}^n \rightarrow \mathbb{R}^2$).

Iterative “Nearly Exact” Solutions to the Subproblem:

Useful **only** for small problems; costlier per iteration, but needs fewer iterations. (*) Definition of “small” may vary.

2-D Subspace Minimization

1 of 9

Method: 2-D Subspace Minimization for Trust-region

Use When: The model Hessian B_k is positive definite, and (modified version) when B_k is indefinite.

Without too much extra effort, we can expand the minimization from the dogleg path to the entire plane spanned by the steepest descent vector $\bar{\mathbf{p}}_k^U$ and the full step vector $\bar{\mathbf{p}}_k^{FS}$,

$$\arg \min_{\bar{\mathbf{p}}} \left[\underbrace{f(\bar{\mathbf{x}}_k) + \bar{\mathbf{p}}^T \nabla f(\bar{\mathbf{x}}_k) + \frac{1}{2} \bar{\mathbf{p}}^T B_k \bar{\mathbf{p}}}_{m_k(\bar{\mathbf{p}})} \right], \text{ s.t. } \|\bar{\mathbf{p}}\| \leq \Delta_k, \bar{\mathbf{p}} \in \text{span}[\bar{\mathbf{p}}_k^U, \bar{\mathbf{p}}_k^{FS}]$$

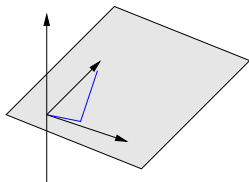


Figure: The two-dimensional subspace spanned by $\bar{\mathbf{p}}_k^U$ and $\bar{\mathbf{p}}_k^{FS}$, with the dogleg path indicated.

2-D Subspace Minimization

2 of 9

The plane is parameterized by (η_1, η_2) :

$$\bar{\mathbf{p}}(\eta_1, \eta_2) = \eta_1 \bar{\mathbf{p}}_k^U + \eta_2 \bar{\mathbf{p}}_k^{FS}.$$

Hence we are reduced to solving the problem

$$\bar{\eta}_k = \arg \min_{\eta_1, \eta_2} \left[f(\bar{\mathbf{x}}_k) + (\eta_1 \bar{\mathbf{p}}_k^U + \eta_2 \bar{\mathbf{p}}_k^{FS})^T \nabla f(\bar{\mathbf{x}}_k) + \frac{1}{2} (\eta_1 \bar{\mathbf{p}}_k^U + \eta_2 \bar{\mathbf{p}}_k^{FS})^T B_k (\eta_1 \bar{\mathbf{p}}_k^U + \eta_2 \bar{\mathbf{p}}_k^{FS}) \right],$$

subject to $\|\bar{\mathbf{p}}(\eta_1, \eta_2)\| \leq \Delta_k$.

This is a minimization of a quadratic model in two variables — $m_k(\eta_1, \eta_2)$; to find the minimum, we set

$$\left[\frac{\partial}{\partial \eta_1} m_k(\eta_1, \eta_2), \quad \frac{\partial}{\partial \eta_2} m_k(\eta_1, \eta_2) \right]^T = \bar{\mathbf{0}}.$$

2-D Subspace Minimization

3 of 9

$$\left[\begin{array}{cc} \frac{\partial}{\partial \eta_1} m_k(\eta_1, \eta_2), & \frac{\partial}{\partial \eta_2} m_k(\eta_1, \eta_2) \end{array} \right]^T = \bar{\mathbf{0}}$$

is a set (2-by-2) of linear equations, and with “some” algebraic manipulation we can explicitly write down the solution (η_1, η_2) .

The optimal parameters (η_1, η_2) are given by⁽¹⁾

$$\left[\begin{array}{c} \eta_1 \\ \eta_2 \end{array} \right] = - \left[\begin{array}{cc} (\bar{\mathbf{p}}_k^U)^T B_k \bar{\mathbf{p}}_k^U & (\bar{\mathbf{p}}_k^U)^T B_k \bar{\mathbf{p}}_k^{FS} \\ (\bar{\mathbf{p}}_k^{FS})^T B_k \bar{\mathbf{p}}_k^U & (\bar{\mathbf{p}}_k^{FS})^T B_k \bar{\mathbf{p}}_k^{FS} \end{array} \right]^{-1} \left[\begin{array}{c} (\bar{\mathbf{p}}_k^U)^T \nabla f(\bar{\mathbf{x}}_k) \\ (\bar{\mathbf{p}}_k^{FS})^T \nabla f(\bar{\mathbf{x}}_k) \end{array} \right],$$

as long as this gives $\bar{\mathbf{p}}(\eta_1, \eta_2) \in T_k$; otherwise⁽²⁾ we perform a one-parameter search $\eta_1 \in [0, \Delta_k / \|\bar{\mathbf{p}}_k^U\|] \Leftrightarrow \eta_2 \in [0, \Delta_k / \|\bar{\mathbf{p}}_k^{FS}\|]$ so that $\eta_1^2 \|\bar{\mathbf{p}}_k^U\|^2 + \eta_2^2 \|\bar{\mathbf{p}}_k^{FS}\|^2 + 2\eta_1\eta_2 \langle \bar{\mathbf{p}}_k^U, \bar{\mathbf{p}}_k^{FS} \rangle = \Delta_k^2$.

2-D Subspace Minimization

Some basic linear algebra can greatly simplify the one-parameter search “ $\eta_1 \in [0, \Delta_k / \|\bar{\mathbf{p}}_k^U\|] \Leftrightarrow \eta_2 \in [0, \Delta_k / \|\bar{\mathbf{p}}_k^{FS}\|]$ so that $\eta_1^2 \|\bar{\mathbf{p}}_k^U\|^2 + \eta_2^2 \|\bar{\mathbf{p}}_k^{FS}\|^2 + 2\eta_1\eta_2 \langle \bar{\mathbf{p}}_k^U, \bar{\mathbf{p}}_k^{FS} \rangle = \Delta_k^2$ ”:

2-D Subspace Minimization

Indefinite B_k

4 of 9

The main advantage of the 2-D Subspace Minimization is that we can extend it to the case where B_k is **indefinite**.

When B_k has negative eigenvalues, the 2-D subspace minimization is performed over a different subspace: —

Since we no longer safely can compute the full step $\bar{\mathbf{p}}_k^{\text{FS}} = -B_k^{-1} \nabla f(\bar{\mathbf{x}}_k)$, we define a new (modified) step $\bar{\mathbf{p}}_k^*$ to replace the full step

$$\text{span}(\bar{\mathbf{p}}_k^{\text{U}}, \bar{\mathbf{p}}_k^{\text{FS}}) \rightarrow \text{span}(\bar{\mathbf{p}}_k^{\text{U}}, \bar{\mathbf{p}}_k^*)$$

The goal is that $\bar{\mathbf{p}}_k^*$ should be “close” to $\bar{\mathbf{p}}_k^{\text{FS}}$, but guaranteed safe to compute.

2-D Subspace Minimization

Indefinite B_k

5 of 9

How do we select $\bar{\mathbf{p}}_k^*$?

Let λ_1 denote the most negative eigenvalue of B_k , and select a parameter $\beta \in (-\lambda_1, -2\lambda_1]$. Then the matrix $(B_k + \beta I)$ must be positive definite.

We now define

$$\bar{\mathbf{p}}_k^* = -(B_k + \beta I)^{-1} \nabla f(\bar{\mathbf{x}}_k)$$

Note: In the case $\|\bar{\mathbf{p}}_k^*\| \leq \Delta_k$ the modified full step is feasible, so we do not have to perform the subspace optimization. **However**, we do not use the step $\bar{\mathbf{p}}_k^*$...

Instead, the step is defined to be $\bar{\mathbf{p}}_k = \bar{\mathbf{p}}_k^* + \bar{\mathbf{v}}$ where the vector $\bar{\mathbf{v}}$ satisfies $\bar{\mathbf{v}}^T \bar{\mathbf{p}}_k^* \geq 0$ (so that $\|\bar{\mathbf{p}}_k\| \geq \|\bar{\mathbf{p}}_k^*\|$). This is a safeguard which makes sure that $\bar{\mathbf{p}}_k$ does not approach 0, in which case the algorithm is not making any progress.

2-D Subspace Minimization

Indefinite B_k

6 of 9

Note: The computation of β for $(B_k + \beta I)$ can be computed by e.g. the Lanczos method^(*).

For a description leading up to, but not completely covering, the Lanczos method see and *Numerical Linear Algebra*, Lloyd N. Trefethen and David Bau, III, Society for Industrial and Applied Mathematics (SIAM), 1997. ISBN 0-89871-361-7.

Note: For descriptions on the selection of $\bar{\mathbf{v}}$, see e.g.

- R.H. Byrd, R.B. Schnabel, and G.A. Schultz, *Approximate solution of the trust regions problem by minimization over two-dimensional subspaces*, Mathematical Programming, 40 (1988) pp. 247–263.
- G.A. Schultz, R.B. Schnabel, and R.H. Byrd, *A family of trust-region-based algorithms for the unconstrained minimization with strong global convergence properties*, SIAM Journal on Numerical Analysis, 22 (1985), pp. 47–67.

Interpretation: $+\beta I$

1 of 2

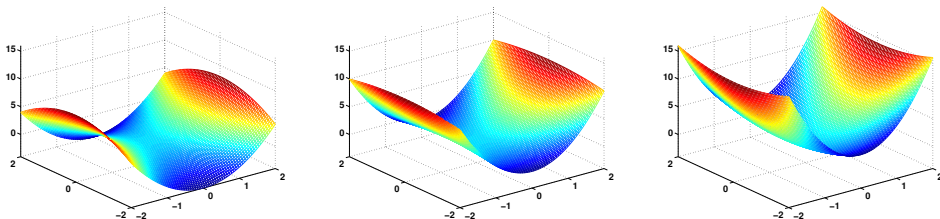


Figure: The addition of βI to B_k can be seen as adding positive curvature β in all directions. In the left panel we see a saddle with eigenvalues $\{-1, 2\}$; in the center panel, we have added $\beta = 0.75$ and we can see that the convex direction has become more convex, and the concave direction less concave; finally, in the right panel we have added $\beta = 1.5$ so that both directions in the “modified saddle” now are convex.

We note that the absolute distances between the eigenvalues are the same in all three cases, but that the condition numbers change — in this example they are 2, 11, and 7.

Interpretation: $+\beta I$

2 of 2

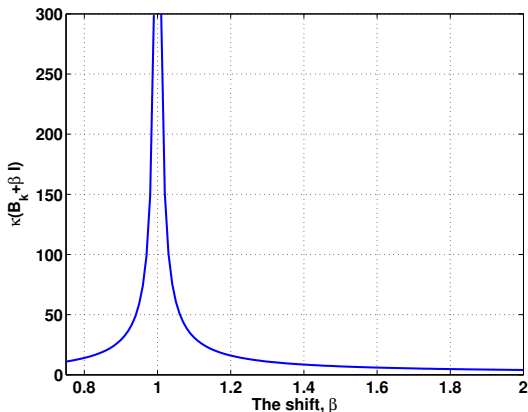


Figure: We see the condition number $\kappa(B_k + \beta I)$ has a spike (to ∞) at $\beta = 1$, so somehow we need to make sure that we make the modified matrix positive definite “enough” that the resulting condition number does not destroy our computational scheme.

Roadmap to Minimization (So Far)

The Global Minimization Problem

Iterative Solution — From $x(k)$ to a better $x(k+1)$

LINESEARCH

Search Direction $p(k)$
 Coordinate Descent
 Steepest Descent
 [Quasi-Newton]
 Newton

Step Length

Initial Step
 Methods
 Backtracking
 Interpolation
 Quadratic
 Cubic
 Sufficient Decrease Condition
 Wolfe Conditions
 Strong Wolfe Conditions
 [Goldstein Conditions]

Convergence

Global
 Rates

TRUSTREGION

Cauchy Point
 Steepest Descent in TR framework

Dogleg
 1D subproblem

2D Subspace Minimization
 2D subproblem
 Model Hessian can be indefinite

[Steihaug's Method]

Nearly Exact Subproblem
 For low-dimensional problems

[Convergence]

Iterative “Nearly Exact” Methods for “Small” Problems

The methods we have looked at: Cauchy Point, Dogleg, and 2-D Subspace Minimization **do not** try to solve the subproblem

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

exactly. — They use some of the information in the model Hessian B_k , and guarantee global convergence to a stationary point at a relatively low cost.

For **small problems**, when the number of unknowns n is not too large, it is sometimes worth the effort to solve the subproblem more accurately.

The cost (per iteration) of the method we describe here is about three factorizations of the (model) Hessian Matrix. The hope is that, by working harder in each step, significantly fewer iterations will be necessary.

Iterative “Nearly Exact” Methods: Setup

The basis of the nearly exact solution of the subproblem is:

- A good characterization of the exact solution.
- Application of Newton’s method in 1-D.

Essentially we are looking for a solution of

$$(B_k + \lambda I)\bar{\mathbf{p}} = -\nabla f(\bar{\mathbf{x}}_k), \quad (\text{The Convexified Problem})$$

for an appropriate $\lambda \geq 0$.

Iterative "Nearly Exact" Methods: Key Result

Theorem

The vector $\bar{\mathbf{p}}^$ is a global solution of the trust-region problem*

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

if and only if $\bar{\mathbf{p}}^$ is feasible, and there is a scalar $\lambda \geq 0$ such that the following conditions are satisfied:*

1. $(B_k + \lambda I) \bar{\mathbf{p}}^* = -\nabla f(\bar{\mathbf{x}}_k),$
2. $\lambda(\Delta_k - \|\bar{\mathbf{p}}^*\|) = 0,$
3. $(B_k + \lambda I)$ is positive semi-definite.

For the proof, see NW^{1st} pp.84–87, or NW^{2nd} pp.89–91.

Iterative “Nearly Exact” Methods: Key Result — Discussion

The Conditions

1. $(B_k + \lambda I)\bar{\mathbf{p}}^* = -\nabla f(\bar{\mathbf{x}}_k)$
2. $\lambda(\Delta_k - \|\bar{\mathbf{p}}^*\|) = 0$
3. $(B_k + \lambda I)$ is positive semi-definite

Condition 1

implies that $\lambda\bar{\mathbf{p}}^* = -B_k\bar{\mathbf{p}}^* - \nabla f(\bar{\mathbf{x}}_k) = -\nabla m_k(\bar{\mathbf{p}}^*)$, i.e. $\bar{\mathbf{p}}^*$ is collinear with the negative gradient of the model m_k , and hence normal to its contour lines. (Steepest Descent for the model, in $\bar{\mathbf{p}}^*$.)

Condition 2

says that either $\lambda = 0$ (in which case B_k is positive semidefinite and $B_k\bar{\mathbf{p}}^* = -\nabla f(\bar{\mathbf{x}}_k)$), or $\bar{\mathbf{p}}^*$ lies on the boundary of the trust region.

Iterative “Nearly Exact” Methods: The Scheme

Based on the theorem, we construct an algorithm for finding the solution of the local subproblem:

Case #1: $\lambda = 0$ works, i.e. B_k is positive semi-definite, and $\|\bar{\mathbf{p}}^*\| = \|B_k^\dagger \nabla f(\bar{\mathbf{x}}_k)\| \leq \Delta_k$. [pseudo-inverse, B_k^\dagger : next slide].

Case #2: Otherwise, we define a function

$$\bar{\mathbf{p}}(\lambda) = -(B_k + \lambda I)^{-1} \nabla f(\bar{\mathbf{x}}_k)$$

which is defined for λ sufficiently large that $(B_k + \lambda I)$ is positive semi-definite. Now, we seek the value of λ so that $\|\bar{\mathbf{p}}(\lambda)\| = \Delta_k$.

Notice: this is a 1-D root-finding problem in λ .

Iterative “Nearly Exact” Methods: The Scheme — Pseudo-Inverse

Definition (The Pseudo-Inverse)

Given $A \in \mathbb{C}^{m \times n}$, the pseudo-inverse is the unique matrix $A^\dagger \in \mathbb{C}^{n \times m}$ that satisfies the Moore-Penrose conditions:

- (i) $AA^\dagger A = A$
- (ii) $A^\dagger AA^\dagger = A^\dagger$
- (iii) $(AA^\dagger)^* = AA^\dagger$
- (iv) $(A^\dagger A)^* = A^\dagger A$

Computationally, the Pseudo-Inverse is typically implemented using the QR -decomposition $[A^\dagger \rightsquigarrow R^{-1}Q^*]$, or the Singular Value Decomposition (SVD) $[A^\dagger \rightsquigarrow V\Sigma^{-1}U^*]$ (i.e. from Linear Algebra).

Iterative “Nearly Exact” Methods: Properties of $\|\bar{\mathbf{p}}(\lambda)\|$

1 of 3

We take a closer look at the properties of $\|\bar{\mathbf{p}}(\lambda)\|$ —

Since B_k is symmetric, it is **unitarily diagonalizable**, (i.e. from Linear Algebra), i.e. there exists an orthonormal matrix Q and a diagonal matrix $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$ where λ_i are the eigenvalues of B_k listed in increasing order, such that $B_k = Q\Lambda Q^T$. Now,

$$B_k + \lambda I = Q(\Lambda + \lambda I)Q^T,$$

and we can write

$$\bar{\mathbf{p}}(\lambda) = -Q(\Lambda + \lambda I)^{-1}Q^T \nabla f(\bar{\mathbf{x}}_k) = -\sum_{j=1}^n \frac{\bar{\mathbf{q}}_j^T \nabla f(\bar{\mathbf{x}}_k)}{\lambda_j + \lambda} \bar{\mathbf{q}}_j,$$

where $\bar{\mathbf{q}}_j$ is the j th column of Q .

Iterative “Nearly Exact” Methods: Properties of $\|\bar{\mathbf{p}}(\lambda)\|$

2 of 3

The “definition” of a small problem is a problem for which we can compute the unitary diagonalization $B_k = Q\Lambda Q^T$ in reasonable amount of time.

If B_k is of size $n \times n$, then the number of operations required to find the decomposition is $\sim \mathcal{O}(n^3)$.

The details of the decomposition (which is a 2-phase algorithm, first converting B_k into tridiagonal form T_k , and then iteratively diagonalizing T_k using the QR-algorithm) is usually taught in a Linear Algebra class.

Iterative “Nearly Exact” Methods: Properties of $\|\bar{\mathbf{p}}(\lambda)\|$

3 of 3

Now, we can write

$$\|\bar{\mathbf{p}}(\lambda)\|^2 = \sum_{j=1}^n \frac{(\bar{\mathbf{q}}_j^T \nabla f(\bar{\mathbf{x}}_k))^2}{(\lambda_j + \lambda)^2}.$$

Note that the numerators are all constants independent of λ . If $\lambda > -\lambda_1$, then $(\lambda_j + \lambda) > 0, \forall j$ and $\|\bar{\mathbf{p}}(\lambda)\|$ is a decreasing function of λ in the interval $(-\lambda_1, \infty)$.

We have the following

$$\lim_{\lambda \rightarrow \infty} \|\bar{\mathbf{p}}(\lambda)\| = 0,$$

and when $\bar{\mathbf{q}}_j^T \nabla f(\bar{\mathbf{x}}_k) \neq 0$

$$\lim_{\lambda \rightarrow -\lambda_j} \|\bar{\mathbf{p}}(\lambda)\| = \infty.$$

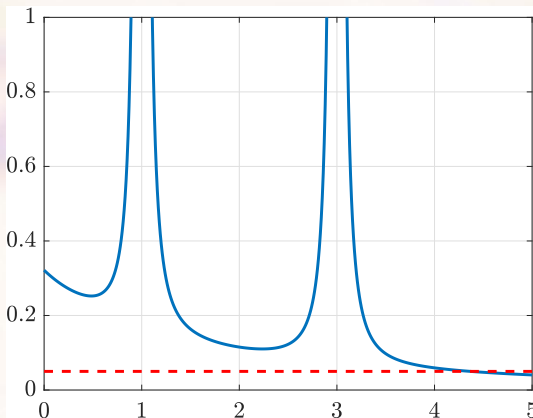
Illustration: $\|\bar{\mathbf{p}}(\lambda)\|$ 

Figure: Illustration, here $\lambda_1 = -3$, $\lambda_2 = -1$, and all other $\lambda_j > 0$. We see the blow-ups of $\|\bar{\mathbf{p}}(\lambda)\|$ at $\lambda = 1$, and $\lambda = 3$. The dash-dotted line illustrates the trust-region bound $\Delta_k = 0.05$.

Iterative “Nearly Exact” Methods: The Scheme, Revisited

1 of 2

Case #2a: If B_k is positive definite, but $\|B_k^{-1}\nabla f(\bar{\mathbf{x}}_k)\| > \Delta_k$, there is a strictly positive λ for which $\|\bar{\mathbf{p}}(\lambda)\| = \Delta_k$, so we search for a $\lambda \in (0, \infty)$.

Case #2b: If B_k is indefinite, and $\bar{\mathbf{q}}_1^T \nabla f(\bar{\mathbf{x}}_k) \neq 0$ we must search in the interval $(-\lambda_1, \infty)$...

Case #2c: If B_k is indefinite, and $\bar{\mathbf{q}}_1^T \nabla f(\bar{\mathbf{x}}_k) = 0 \rightsquigarrow$ “The Hard Case.”

We would like to apply Newton’s method to

$$\Phi_1(\lambda) = \|\bar{\mathbf{p}}(\lambda)\| - \Delta_k = 0,$$

but since $\|\bar{\mathbf{p}}(\lambda)\|$ (and therefore $\Phi_1(\lambda)$) blows up around λ_1 , Newton’s method may be slow and unreliable.

Iterative “Nearly Exact” Methods: The Scheme, Revisited

2 of 2

Instead, we apply Newton's method to the root finding problem

$$\Phi_2(\lambda) = \frac{1}{\Delta_k} - \frac{1}{\|\bar{\mathbf{p}}(\lambda)\|} = 0.$$

$\Phi_2(\lambda)$ behaves nicely (linearly!) around λ_1 :

$$\Phi_2(\lambda) \approx \frac{1}{\Delta_k} - \frac{\lambda + \lambda_1}{C}, \quad C > 0, \quad \lambda \approx \lambda_1$$

Hence, root-finding with Newton's method will work.

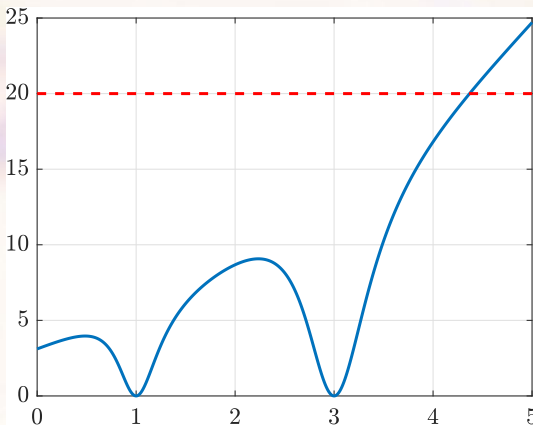
Illustration: Searching for $1/\|\bar{\mathbf{p}}(\lambda)\| = 1/\Delta_k$ 

Figure: Illustration, here $\lambda_1 = -3$, $\lambda_2 = -1$, and all other $\lambda_j > 0$. We see that the blow-ups of $\|\bar{\mathbf{p}}(\lambda)\|$ at $\lambda = 1$, and $\lambda = 3$ have been “converted” into zeros. The dashed line illustrates the converted trust-region bound $1/\Delta_k = 1/0.05 = 20.0$.

Exact Trust Region: Newton-Iteration Module

We are now ready to write down the Newton iteration

$$\lambda^{(n+1)} = \lambda^{(n)} - \frac{\Phi_2(\lambda^{(n)})}{\Phi_2'(\lambda^{(n)})}$$

Algorithm: Exact Trust Region

1. Given $\lambda^{(0)}$, $\Delta_k > 0$
2. LOOP until convergence
3. TRY to factor $B + \lambda^{(n)}I = L^T L$
4. IF factorization failed
5. increase $\lambda^{(n)}$ and RETRY (3)
6. ELSE
7. Solve $L^T L \bar{\mathbf{p}}_n = -\nabla f(\bar{\mathbf{x}}_k)$, $L^T \bar{\mathbf{q}}_n = \bar{\mathbf{p}}_n$
8. Update

$$\lambda^{(n+1)} = \lambda^{(n)} + \left[\frac{\|\bar{\mathbf{p}}_n\|}{\|\bar{\mathbf{q}}_n\|} \right]^2 \left[\frac{\|\bar{\mathbf{p}}_n\| - \Delta}{\Delta} \right]$$

9. ENDIF, ENDLLOOP ($n = n + 1$)

Cholesky Factorization —

“TRY to factor $B + \lambda^{(n)}I = L^T L$ ”

Algorithm: Cholesky Factorization

```

for i = 1, 2, ..., n
     $L_{ii} = \sqrt{A_{ii}}$ , this Fails if  $A_{ii} < 0$ , (really if  $A_{ii} < \epsilon_*$ )
    for j = i+1, i+2, ..., n
         $L_{ji} = A_{ji} / L_{ii}$ 
        for k = i+1, i+2, ..., j
             $A_{jk} = A_{jk} - L_{ji} L_{ki}$ 
        end-for-k
    end-for-j
end-for-i

```

(*) André-Louis Cholesky (15 October 1875 – 31 August 1918). He served the French military as artillery officer, and was killed in battle a few months before the end of World War I.



Exact Trust Region: Success and Failure

This algorithm will work nicely for Case #2a and Case #2b, unfortunately the story does not end there. (In Case #2a we can apply Newton’s Method directly to the un-converted $\Phi_1(\lambda)$, as long as we enforce $\lambda > 0$ in the Newton iteration).

Recall that in Case #2b we assumed $\bar{\mathbf{q}}_1^T \nabla f(\bar{\mathbf{x}}_k) \neq 0$. If that is not true there may not be a value $\lambda \in (-\lambda_1, \infty)$ for which $\|\bar{\mathbf{p}}(\lambda)\| = \Delta_k$. (See **illustration** on next slide.)

Since root finding on $(-\lambda_1, \infty)$ fails, and the theorem guarantees that $\lambda \in [-\lambda_1, \infty)$, we must have $\lambda = -\lambda_1$.

Finding $\bar{\mathbf{p}}$ in this case requires a little more work...

Illustration: The Hard Case

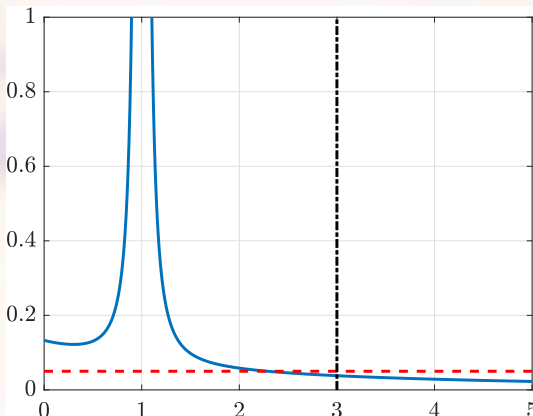


Figure: Illustration, here $\lambda_1 = -3$, $\lambda_2 = -1$, and all other $\lambda_j > 0$. Here $\bar{\mathbf{q}}_1^T \nabla f(\bar{\mathbf{x}}_k) = 0$, and we end up with the hard case where $\|\bar{\mathbf{p}}(\lambda)\| = \Delta$ is not solvable in the region $\lambda \in (-\lambda_1, \infty)$. (We see the blow-up at $-\lambda_2$, and the location of $-\lambda_1$ has been marked with a vertical dash-dotted line.)

The Hard Case: Computing $\bar{\mathbf{p}}$

The matrix $(B - \lambda_1 I)$ is singular, therefore there exists a vector $\bar{\mathbf{z}}$ such that $\|\bar{\mathbf{z}}\| = 1$ and $(B - \lambda_1 I)\bar{\mathbf{z}} = 0$.

Hence $\bar{\mathbf{z}}$ is the eigenvector of B corresponding to λ_1 , by the orthogonality of Q , we have $\bar{\mathbf{q}}_j^T \bar{\mathbf{z}} = 0$ for $\lambda_j \neq \lambda_1$. If we set

$$\bar{\mathbf{p}}(\tau, \lambda) = \tau \bar{\mathbf{z}} + \sum_{\{j: \lambda_j \neq \lambda_1\}} \frac{\bar{\mathbf{q}}_j^T \nabla f(\bar{\mathbf{x}}_k)}{\lambda_j + \lambda} \bar{\mathbf{q}}_j,$$

for any $\tau \in \mathbb{R}$, then

$$\|\bar{\mathbf{p}}(\tau, \lambda)\|^2 = \tau^2 + \sum_{\{j: \lambda_j \neq \lambda_1\}} \frac{(\bar{\mathbf{q}}_j^T \nabla f(\bar{\mathbf{x}}_k))^2}{(\lambda_j + \lambda)^2}.$$

Since $\|\bar{\mathbf{p}}(0, \lambda_1)\| < \Delta_k$ and $\|\bar{\mathbf{p}}(\tau, \lambda_1)\|$ is monotonically increasing in τ , we can find the unique τ^* for which $\|\bar{\mathbf{p}}(\tau^*, \lambda_1)\| = \Delta_k$.

Index

2D subspace minimization, 8
 indefinite Hessian, 12
Cholesky factorization, 38
minimization road map, 22
trust region subproblem
 iterative solution, 24

Reference(s):

- BSS-1988** R.H. Byrd, R.B. Schnabel, and G.A. Schultz, *Approximate solution of the trust regions problem by minimization over two-dimensional subspaces*, Mathematical Programming, 40 (1988) pp. 247–263.
- SSB-1985** G.A. Schultz, R.B. Schnabel, and R.H. Byrd, *A family of trust-region-based algorithms for the unconstrained minimization with strong global convergence properties*, SIAM Journal on Numerical Analysis, 22 (1985), pp. 47–67.
- TB-1997** *Numerical Linear Algebra*, Lloyd N. Trefethen and David Bau, III, Society for Industrial and Applied Mathematics (SIAM), 1997. ISBN 0-89871-361-7.