

Numerical Optimization

Unit 3: Methods That Guarantee Convergence

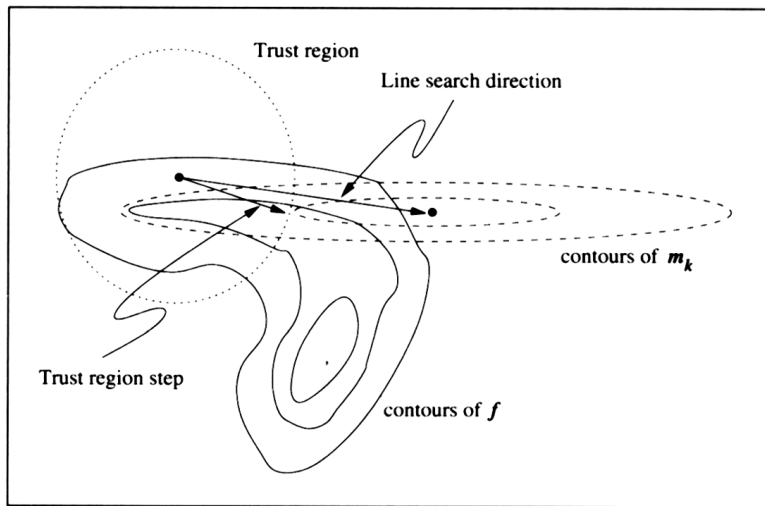
Che-Rung Lee

Department of Computer Science
National Tsing Hua University

September 22, 2021

Line search and trust region

We will talk about two types of algorithms that guarantee convergence:
Line search and trust region.



Where are we?

Three problems of Newton's method:

- ① Hessian matrix H may not be positive definite.
- ② Hessian matrix H is expensive to compute.
- ③ The system $\vec{p} = -H^{-1}\vec{g}$ is expensive to compute.

We will discuss methods to solve the first problem.

- ① The consequence of the first problem is that Newton's direction is not a descent direction.
- ② We need to find directions that are similar to the Newton's direction but also descent.
- ③ Do descent directions guarantee convergence? The answer is NO.
- ④ Line search algorithms: descent directions+good step sizes, which guarantee convergence.

Modified Newton's method

- When the Hessian H is not positive definite, what can we do?
 - Use another \hat{H} , similar to H , but positive definite.
 - How can this work?

$$\begin{aligned}\vec{p} &= -\hat{H}^{-1}\vec{g} \\ \vec{g}^T \vec{p} &= -\vec{g}^T \hat{H} \vec{g} < 0\end{aligned}$$

\vec{p} is a descent direction.

Theorem (The convergence of the modified Newton)

If f is twice continuously differentiable in a domain D and $\nabla^2 f(\vec{x}^)$ is positive definite. Assume \vec{x}_0 is sufficiently close to \vec{x}^* and the modified \hat{H}_k is well-conditioned. Then*

$$\lim_{k \rightarrow \infty} \nabla f(\vec{x}_k) = 0.$$

Conditionness of a matrix

- For a matrix, what is “well-conditioned”?
 - A matrix A 's condition number is $\kappa(A) = \|A\| \|A^{-1}\|$. If $\kappa(A)$ is small, we call A well-conditioned. If $\kappa(A)$ is large, we call A ill-conditioned.
- But what is the meaning of $\kappa(A)$?
 - The condition number $\kappa(A)$ measures the “sensitivity” of the matrix when solving $Ax = b$.

$$(A + E)\tilde{x} = b = Ax$$

$$A\tilde{x} - Ax = -E\tilde{x}$$

$$\tilde{x} - x = -A^{-1}E\tilde{x}$$

$$\|\tilde{x} - x\| = \|A^{-1}E\tilde{x}\| \leq \|A^{-1}\| \|E\| \|\tilde{x}\|$$

$$\frac{\|\tilde{x} - x\|}{\|\tilde{x}\|} \leq \|A\| \|A^{-1}\| \frac{\|E\|}{\|A\|} = \kappa(A) \frac{\|E\|}{\|A\|}$$

Requirements of good modifications

- Three requirements of a good modification:
 - ① Matrix \hat{H} is positive definite and well-conditioned, so the convergence theorem holds.
 - ② Matrix \hat{H} is similar to H , $\|\hat{H} - H\|$ small, so \vec{p} is close to the Newton's direction, and the fast convergence can be hopefully preserved.
 - ③ The modification can be easily computed.
- We will see three algorithms, and each has its pros and cons.
 - ① Eigenvalue modification.
 - ② Shift modification.
 - ③ Modification with LDL decomposition.

First method: eigenvalue modification

Algorithm 1: Eigenvalue modification

- 1 Compute H 's eigenvalue decomposition, $H = V\Lambda V^{-1}$, $\Lambda = \text{diag}(\lambda_1, \lambda_2, \dots, \lambda_n)$.
- 2 Make the modification for a given small $\epsilon > 0$,

$$\hat{\lambda}_i = \begin{cases} \lambda_i, & \text{if } \lambda_i > 0 \\ \epsilon, & \text{if } \lambda_i < 0 \end{cases}$$

- 3 $\hat{H} = V\hat{\Lambda}V^{-1}$, $\hat{\Lambda} = \text{diag}(\hat{\lambda}_1 \ \hat{\lambda}_2 \ \dots \ \hat{\lambda}_n)$.

- It satisfies requirement 1 and 2 (why?), but eigenvalue decomposition is expensive to compute: $O(n^3)$ with big constant coefficient.

Second method: shift modification

Algorithm 2: Shift modification

- ① Let $H_0 = H$.
- ② For $k = 0, 1, 2, \dots$
 - ① If H_k can have Cholesky decomposition, then return $\hat{H} = H_k$.
 - ② Otherwise, $H_{i+1} = H_i + \mu I$ for some small $\mu > 0$.

- Why does that work?

$$H + \mu I = V\Lambda V^{-1} + \mu I = V\Lambda V^{-1} + \mu VV^{-1} = V(\Lambda + \mu I)V^{-1}$$

$$\Lambda + \mu I = \begin{pmatrix} \lambda_1 + \mu & & & \\ & \lambda_2 + \mu & & \\ & & \ddots & \\ & & & \lambda_n + \mu \end{pmatrix}, \quad \mu > 0$$

- Matrix H_k is symmetric positive definite if and only if its Cholesky definition exists. (See note 2.)
- Which requirements this method satisfies?

Third method: using LDL^T decomposition

Algorithm 3: Modified LDL^T Decomposition

- 1 Compute $H = LDL^T$.
 - 2 Update D to \hat{D} so that all \hat{d}_i are positive.
(If $d_i < 0$, replace it by $\epsilon > 0$.)
 - 3 $\hat{H} = L\hat{D}L^T$.
- The LDL decomposition of a symmetric matrix H is $H = LDL^T$, where L is lower triangular and D is diagonal.
 - Additional advantage of LDL decomposition: we can use that to solve $\hat{H}\vec{p} = -\vec{g}$,
$$\vec{p} = -L^{-T}\hat{D}^{-1}L^{-1}\vec{g}.$$
 - But it is not numerically stable (the updates can be very large).

Some properties of descent direction

Why are we so obsessed with the "descent direction"?

- Let $\phi_k(\alpha) = f(\vec{x}_k + \alpha\vec{p}_k)$.
- Since \vec{p}_k is a decent direction, $\phi_k(\varepsilon) < \phi_k(0)$ for some small $\varepsilon > 0$.
- $\phi'_k(0) = \nabla f_k^T \vec{p}_k$. (Why?)
- $\phi'_k(\alpha) = \nabla f_k(\vec{x}_k + \alpha\vec{p}_k)^T \vec{p}_k$. (Why?)

Problems of descent directions

- The descent directions guarantee that $f(\vec{x}_{k+1}) < f(\vec{x}_k)$, which however do not guarantee to converge to the optimal solution.
- Here are two examples. ¹

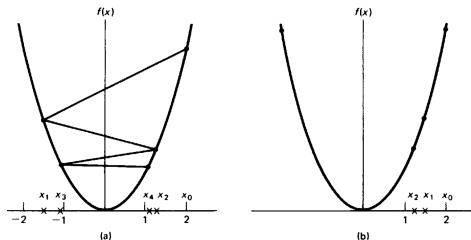


Figure 6.3.2 Monotonically decreasing sequences of iterates that don't converge to the minimizer

- $f(x) = x^2$, $x_0 = 2$, $p_k = (-1)^{k+1}$ and $\alpha_k = 2 - 3 \times 2^{-k-1}$,
 $\{x_k\} = \{2, -3/2, 5/4, -9/8, \dots\} = \{(-1)^k(1 + 2^{-k})\}$.
- $f(x) = x^2$, $x_0 = 2$, $p_k = -1$ and $\alpha_k = 2^{-k-1}$,
 $\{x_k\} = \{2, 3/2, 5/4, 9/8, \dots\} = \{1 + 2^{-k}\}$.

¹Example and figures are from chapter 6 of *Numerical Methods for Unconstrained Optimization and Nonlinear Equations* by J. Dennis and R. Schnabel

First example

- What's the problem of the first example?
 - The *relative decrease* is $\frac{|\phi_k(\alpha_k) - \phi_k(0)|}{\alpha_k} \approx 2^{-k}$ which becomes too small before reaching the optimal solution.
 - The relative decrease is the absolute value of the slope of the line segment $(\vec{x}_k, f(\vec{x}_k)), (\vec{x}_{k+1}, f(\vec{x}_{k+1}))$.
 - How large should the relative decrease be? The slope of the tangent line at $\alpha = 0$ provides good information about f 's trend. (What is $\phi'(0)$? What is the sign of $\phi'(0)$?)
 - The sufficient decrease condition:

Sufficient decrease condition

$$f(\vec{x}_k + \alpha \vec{p}_k) \leq f(\vec{x}_k) + c_1 \alpha \vec{g}_k^T \vec{p}_k,$$

for some $c_1 \in (0, 1)$.

Second example

- What's the problem of the second example?
 - The *relative decrease* of the second problem is $\frac{|\phi_k(\alpha_k) - \phi_k(0)|}{\alpha_k} \approx 1$ is large enough, but *the step is too small*.
 - How large should the step size at least to be? Remember that α should be shrunken as f converges to the optimal solution. $\Rightarrow f'$ converges to 0.
 - So the step size should be proportional to the change of ϕ' , which leads to the curvature condition:

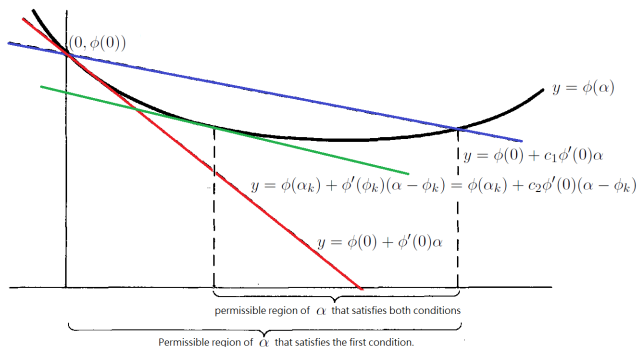
Curvature condition

$$\phi'_k(\alpha_k) = \nabla f(\vec{x}_k + \alpha_k \vec{p}_k)^T \vec{p}_k \geq c_2 \nabla f_k^T \vec{p}_k = c_2 \phi'_k(0)$$

for some $c_2 \in (c_1, 1)$.

Wolfe conditions

- Condition 1 and condition 2 together are called *the Wolfe conditions*.²



- Typical values: $c_1 = 0.1$ and $c_2 = 0.9$.
- Can both conditions be satisfied simultaneously for any smooth function?

²Figure is also from D&S's book.

Existence of feasible region for the Wolfe conditions

- 1 The function $\phi_k(\alpha)$ must be bounded below, which means it will go up eventually (why?). Therefore, the line $y = \phi_k(0) + c_1\phi'_k(0)\alpha$ must intersect with $y = \phi_k(\alpha)$, say at α_1 .
- 2 Since \vec{p}_k is a descent direction, $\phi'_k(0) < c_1\phi'_k(0) < 0$ for some $c_1 \in (0, 1)$.
- 3 By the mean value theorem, $\exists \alpha_2 \in [0, \alpha_1]$, such that

$$c_1\phi'_k(0) = \frac{\phi_k(\alpha_1) - \phi_k(0)}{\alpha_1 - 0} = \phi'_k(\alpha_2).$$

- 4 Since the curvature condition requires $c_2 > c_1$, between $[\alpha_2, \alpha_1]$, there must be some regions in which there exists α_3 such that $\phi'_k(\alpha_3) \geq c_2\phi'_k(0)$. (why?)

Convergence guarantee

- Do Wolfe conditions guarantee convergence?

Theorem

If \vec{p}_k is a descent direction, α_k satisfies Wolfe conditions, f is bounded below and continuously differentiable, and ∇f is Lipschitz continuous, then

$$\sum_{k \geq 0} \cos^2 \theta_k \|\nabla f_k\|^2 < \infty$$

where $\cos \theta_k = \frac{-\nabla f_k^T \vec{p}_k}{\|\nabla f_k\| \|\vec{p}_k\|}$.

Definition (Lipschitz continuous)

A vector function $f : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is Lipschitz continuous if $\|f(\vec{x}) - f(\vec{y})\| < L \|\vec{x} - \vec{y}\|$ for some constant $L > 0$.

Implications of the theorem

- The convergence theorem implies $\lim_{k \rightarrow \infty} \cos^2 \theta_k \|\nabla f_k\|^2 = 0$. (why?)
- To show the convergence, we need to show that $|\cos \theta_k| > \delta > 0$ when $k \rightarrow \infty$.
- For the steepest descent method, this condition satisfies automatically since \vec{p}_k is parallel to \vec{g}_k .
- How about the Newton's method or the modified Newton's method?
For them, $\vec{p}_k = -H_k^{-1} \vec{g}_k$ or $\vec{p}_k = -\hat{H}_k^{-1} \vec{g}_k$.

$$\vec{g}_k^T \vec{p}_k = -\vec{g}_k^T H_k^{-1} \vec{g}_k.$$

One can show that if H_k is well-conditioned, $\kappa(H) < M$, then $|\cos \theta_k| > 1/M$.

Goldstein condition

- Problems of the Wolfe conditions are the need to evaluate

$$\phi'(\alpha_k) = \nabla f(\vec{x}_k + \alpha_k \vec{p}_k)^T \vec{p}_k.$$

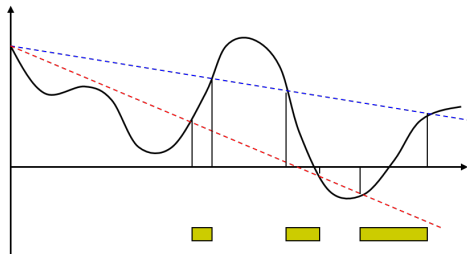
- Another frequently used conditions is the Goldstein condition:

Goldstein condition

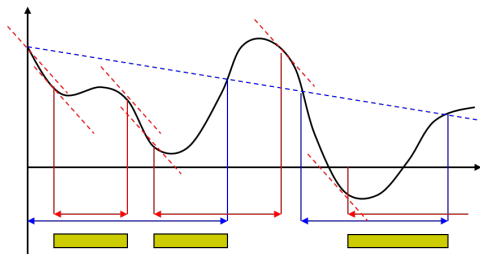
$$f(\vec{x}_k) + (1 - c)\alpha_k \nabla f_k^T \vec{p}_k \leq f(\vec{x}_k + \alpha \vec{p}_k) \leq f(\vec{x}_k) + c\alpha_k \nabla f_k^T \vec{p}_k$$

for $c \in [0, 1/2]$.

Examples of Wolfe conditions and Goldstein condition



The Goldstein conditions



The Wolfe conditions

Algorithm 4: Backtracking line search algorithm

- ➊ Guess an initial α_0 (For Newton's method, usually $\alpha_0 = 1$.)
- ➋ For $k = 1, 2, \dots$ until α_k satisfies the required conditions.
 - Using interpolation methods to model function $\phi(\alpha)$ in the desired interval and then search the feasible solution of the model function.

What is the interpolation method?

- Initially, we know $\phi(0) = f(\vec{x}_k)$, $\phi'(0) = \nabla f(\vec{x}_k)^T \vec{p}_K$, and $\phi(1)$. We can use that build a quadratic polynomial $q_0(\alpha)$ such that $q_0(0) = \phi(0)$, $q'_0(0) = \phi'(0)$ and $q_0(1) = \phi(1)$.
- Use q_0 to find a solution α_1 . Check if α_1 satisfies the required conditions.
- Now we know four things: $\phi(0) = f(\vec{x}_k)$, $\phi'(0) = \nabla f(\vec{x}_k)^T \vec{p}_K$, $\phi(1)$, and $\phi(\alpha_1)$. Use them to build a cubic polynomial $q_1(\alpha)$ such that $q_1(0) = \phi(0)$, $q'_1(0) = \phi'(0)$, $q_1(\alpha_1) = \phi(\alpha_1)$ and $q_1(1) = \phi(1)$.
- Use q_1 to find a solution α_2 . Check if α_2 satisfies the required conditions.

Trust region method

- The line search method finds a descent direction \vec{p}_k first, and then search a suitable step length α_k that satisfies some conditions.
- The idea of the trust region method is to build a model for the function, and then specifies a region in which this model works. It then solves constrained model problem.

Algorithm 5: The trust region framework

- 1 Guess an initial trust region Δ_0 and an initial \vec{x}_0 .
- 2 For $k = 0, 1, 2, \dots$ until convergence
 - 1 Build a model m_k of f at \vec{x}_k
 - 2 Solve the constrained minimization problem: $\min_{\vec{p}} m_k(\vec{p})$ s.t. $\|\vec{p}\| \leq \Delta_k$.
 - 3 Evaluate the trust region Δ_k . If not satisfied, update Δ_k and goto (2-2).
 - 4 Set $\vec{x}_{k+1} = \vec{x}_k + \vec{p}_k$ where \vec{p}_k is the solution of the model problem.

Details of the trust region method

- How to build a model for a function $f(\vec{x})$?
 - Most are based on the Taylor expansions. For example, the quadratic model

$$m_k(\vec{p}) = f_k + \vec{g}_k^T \vec{p} + \frac{1}{2} \vec{p}^T H_k \vec{p}.$$

- How to evaluate and update the trust region Δ_k ?
 - The trust region is evaluated by the given $\vec{p}_k \neq \vec{0}$. Let

$$\rho_k = \frac{f(\vec{x}_k) - f(\vec{x}_k + \vec{p}_k)}{m_k(\vec{0}) - m_k(\vec{p}_k)}.$$

- If $\rho_k < 0$, reject the solution, and let $\Delta_k = \sigma_k \Delta_k$ for some $0 < \sigma_k < 1$.
 - If ρ_k is close to 1, increase $\Delta_k = \tau_k \Delta_k$ for some $\tau_k > 1$.
- The trust region method is also guaranteeing convergence.

Convergence of trust region framework

Theorem (The convergence of trust region framework)

Suppose $\|B_k\|$ is bounded, and f is bounded below on the level set $S = \{x | f(x) \leq f(x_0)\}$ and Lipschitz continuously differentiable in the neighborhood of S . If

$$m_k(\vec{0}) - m_k(\vec{p}_k) \geq c_1 \|\vec{g}_k\| \min \left(\Delta_k, \frac{\|\vec{g}_k\|}{\|B_k\|} \right)$$

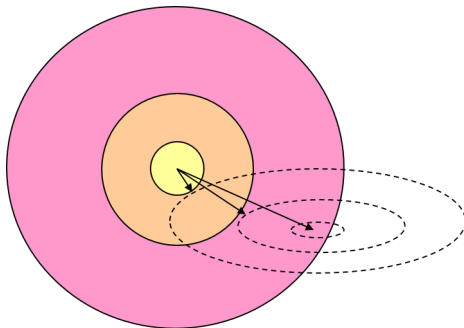
and $\|\vec{p}_k\| \leq \gamma \Delta_k$ for some $c_1 \in (0, 1]$ and $\gamma \geq 1$. Then

$$\lim_{k \rightarrow \infty} \inf \|\vec{g}_k\| = 0.$$

Solving the model problem m_k

$$\begin{aligned} \min_{\vec{p}} m_k(\vec{p}) &= f_k + \vec{g}_k^T \vec{p} + \frac{1}{2} \vec{p}^T B_k \vec{p}. \\ \text{s.t. } \|\vec{p}\| &\leq \Delta \end{aligned}$$

- If $\|B_k^{-1} \vec{g}\|$ and B_k is positive definite, $\vec{p} = -B_k^{-1} \vec{g}$.
- Otherwise, the direction varies for different Δ .



- \vec{p}^* is the optimal solution if and only if it satisfies

$$(B_k + \lambda I)\vec{p}^* = -\vec{g}$$

$$\lambda(\Delta - \|\vec{p}^*\|) = 0$$

where $B_k + \lambda I$ is positive definite.

- $\lambda \geq 0$ is called the Lagrangian modifier (chap 12).
- Assume $\|B^{-1}\vec{g}\| \geq \Delta$ for $\lambda \geq 0$. Define

$$\phi(\lambda) = \|(B + \lambda I)^{-1}\vec{g}\| - \Delta$$

and solve $\phi(\lambda) = 0$.

- This is a univariable nonlinear equation. (chap 11)

Example

$$\min_{x,y} f(x,y) = x^4 + 2x^3 + 24x^2 + y^4 + 12y^2 \text{ s.t. } \Delta = \sqrt{x^2 + y^2} \leq 1$$

$$\text{At } (2,1), f(2,1) = 141, \nabla f(2,1) = \begin{bmatrix} 152 \\ 28 \end{bmatrix}, \nabla^2 f(2,1) = \begin{bmatrix} 120 & 0 \\ 0 & 36 \end{bmatrix}$$

The quadratic model at $(2,1)$ is

$$m(x,y) = 60(2-x)^2 + 18(1-y)^2 + 152(2-x) + 28(1-y) + 141$$

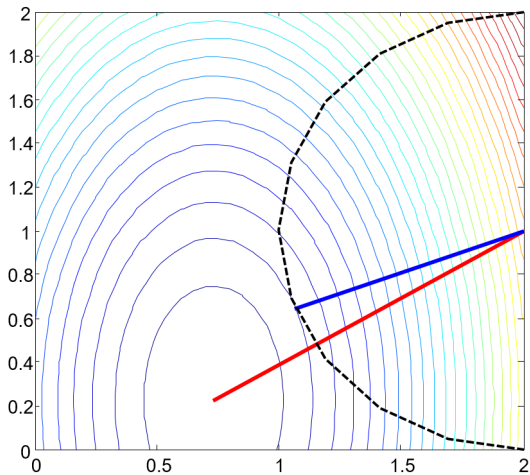
$$\text{The Newton's direction is } \vec{p}^N = -(\nabla^2 f)^{-1} \nabla f = \begin{bmatrix} -1.22 \\ -0.77 \end{bmatrix}, \|\vec{p}^N\| > \Delta$$

Find λ such that $(B_k + \lambda I)\vec{p} = -\vec{g}$ and $\lambda(\Delta - \|\vec{p}\|) = 0$.

$$B + \lambda I = \begin{bmatrix} 120 + \lambda & 0 \\ 0 & 26 + \lambda \end{bmatrix}, \vec{p} = -(B + \lambda I)^{-1} \vec{g} = \begin{bmatrix} -152/(120 + \lambda) \\ -28/(26 + \lambda) \end{bmatrix}$$

$$\text{Let } \|\vec{p}\| = \Delta = 1. \text{ We can solve } \lambda^* = 42.655 \text{ and } \vec{p}^* = \begin{bmatrix} -0.93 \\ -0.36 \end{bmatrix}$$

Example: continue



Approximate solutions and scaling

- The problem $\phi(\lambda) = \| -(B + \lambda I)^{-1} \vec{g} \| - \Delta = 0$ is difficult to solve. Approximate methods are used instead
 - Cauchy point
 - The dogleg method
 - Two-dimensional subspace minimization
- Poor scaled problems are sensitive to certain directions. The solution is to make the trust region elliptical (scaling).

$$\min_{\vec{p}} m_k(\vec{p}) = f_k + \vec{g}_k^T \vec{p} + \frac{1}{2} \vec{p}^T B_k \vec{p}.$$

$$\text{s.t. } \|D\vec{p}\| \leq \Delta$$

where D is a diagonal matrix.

Cauchy point

- The steepest descent direction (gradient + line search)
- The solution is

$$\vec{p}_k = -\tau_k \Delta \frac{\vec{g}_k}{\|\vec{g}_k\|} \text{ (Cauchy point)}$$

where

$$\tau_k = \begin{cases} 1 & \vec{g}_k^T B_k \vec{g}_k \leq 0 \\ \min(\|\vec{g}_k\|^3 / (\Delta \vec{g}_k^T B_k \vec{g}_k), 1) & \text{otherwise} \end{cases}$$

- Pros and Cons
 - Slow convergence
 - Easy to compute
 - Use as a reference direction

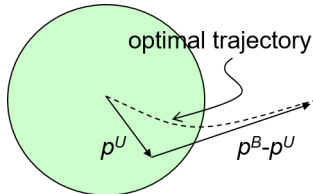
Dogleg method

- Use the combination of Cauchy point and Newton's direction to approximate the optimal trajectory. (Require B_k be positive definite.)
- Find τ such that $\|\vec{p}(\tau)\|^2 = \Delta^2$

$$\vec{p}(\tau) = \begin{cases} \tau \vec{p}^U & 0 \leq \tau \leq 1 \\ \vec{p}^U + (\tau - 1)(\vec{p}^B - \vec{p}^U) & \text{otherwise} \end{cases}$$

where

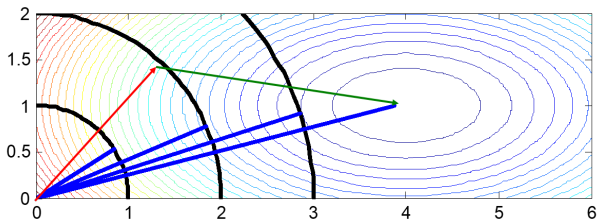
- $\vec{p}^U = -\frac{\vec{g}_k^T \vec{g}_k}{\vec{g}_k^T B_k \vec{g}_k} \vec{g}_k$ is Cauchy point.
- $\vec{p}^B = -B_k^{-1} \vec{g}_k$ is Newton's direction.
- If B_k is positive definite, $\|\vec{p}(\tau)\|$ is an increasing function and $m(\vec{p}(\tau))$ is a decreasing function.



Dogleg method: example

Consider $f(x) = \frac{1}{2}\vec{x}^T \begin{bmatrix} 1/4 & 0 \\ 0 & 1 \end{bmatrix} \vec{x} - \begin{bmatrix} 1 \\ 1 \end{bmatrix}^T \vec{x}$. Let $x_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$.

$$\vec{p}^U = \begin{bmatrix} 1.6 \\ 1.6 \end{bmatrix}, \vec{p}^B = \begin{bmatrix} 4 \\ 1 \end{bmatrix}, \text{ and } \vec{p}^B - \vec{p}^U = \begin{bmatrix} 2.4 \\ -0.6 \end{bmatrix}$$



Δ	λ	$\min f(x)$	τ	$f(p(\tau))$
1	.9212	-1.1478	0.625	-1.1017
2	.2912	-1.8935	1.064	-1.7116
3	.0998	-2.3331	1.551	-2.3193

Two-dimensional subspace minimization

- Use the linear combination of \vec{g} and $B^{-1}\vec{g}$.

$$\min_{\vec{p}} m_k(\vec{p}) = f_k + \vec{g}_k^T \vec{p} + \frac{1}{2} \vec{p}^T B_k \vec{p}.$$

$$\text{s.t. } \|\vec{p}\| \leq \Delta \text{ and } \vec{p} \in \text{span}(\vec{g}, B^{-1}\vec{g})$$

- Matrix B_k can be indefinite. In that case,
 - Find α so that $B_k + \alpha I$ is positive definite
 - If $\|(B_k + \alpha I)^{-1}\vec{g}_k\| \leq \Delta$, let $\vec{p} = (B_k + \alpha I)^{-1}\vec{g}_k + \vec{v}$ where \vec{v} satisfies $\vec{v}^T (B_k + \alpha I)^{-1}\vec{g}_k \leq 0$
 - Otherwise, let $\vec{p} \in \text{span}(\vec{g}, (B + \alpha I)^{-1}\vec{g})$