B Trust Region Methods

- As seen before, for the Newton method, we used the quadratic model to generate a search direction & then focus on finding a suitable step length α along this direction.
- On the other hand, trust region methods define a region round the current point within which they trust the model to be an adequate representation of the objective function and then choose the step to be the (approximate) minimiser of the model in this trust region. (A more conservative approach.)

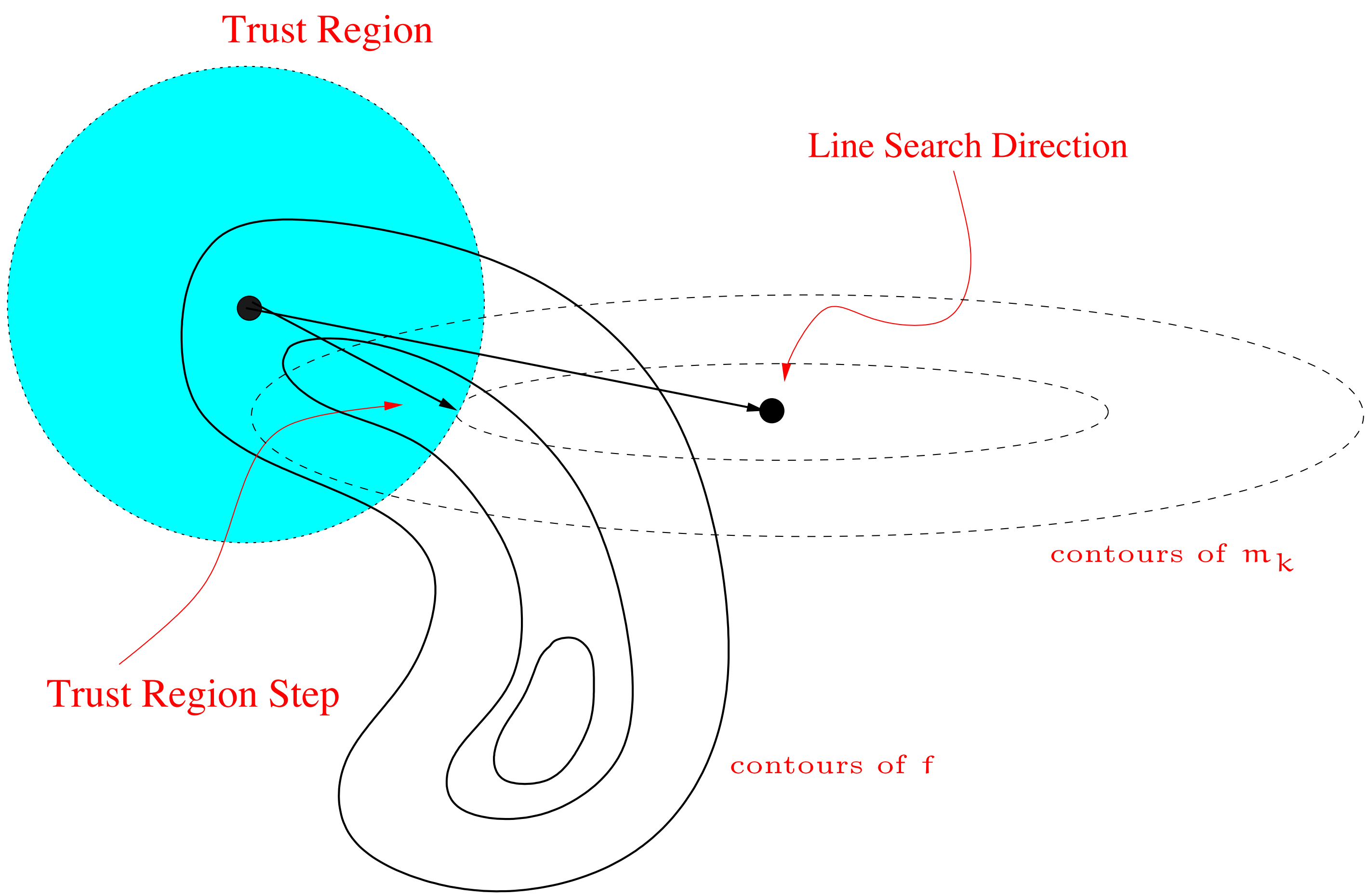


Figure 34: Trust-region vs. line-search methods

Obviously, the size of the trust region is crucial. Too small a region means a missed opportunity to take a large step while too large a region may mean a minimiser far from the minimiser of the objective function in the region.

I will assume that the first two terms of the quadratic model function m_k at each iterate x_k are identical to the first two terms of the Taylor Series expansion of f around x_k. I have

$$m_k(p) = f_k + {g_k}^T p + \frac{1}{2} p^T B_k p, \tag{B.1}$$

At each iteration, I seek a solution of the subproblem,

$$\min_{p\in\mathbb{R}^n} m_k(p) = f_k + {g_k}^T p + \frac{1}{2} p^T B_k p \quad \text{s.t.} \quad \|p\| \leq \Delta_k, \qquad \text{(B.2)}$$

where $\boxed{\Delta_k \text{ is the trust region radius.}}$

We have seen this problem in
Slide 5-14 (Boyd & Vandenberghe)

- For the moment, I will use the 2-norm; so the trust region is a ball centred at x_k , radius Δ_k.
- If B_k is positive definite and the Newton direction ${p_k}^B = -{B_k}^{-1} g_k$ has norm $\leq \Delta_k$, the solution of Eq. B.2 is just the **unconstrained** minimum to the subproblem, (B.2).
- In this case I call ${p_k}^B$ the **full step**.
- In other cases it is not so easy to find an approximate solution to (B.2).

B.1 Outline of the Algorithm

- The first decision to make is the strategy for choosing the trust region radius Δ_k at each iteration.
- I base the strategy on the agreement between the model function m_k and the objective function f at previous iterations.
- Given a step p_k , I define the ratio

$$r_k = \frac{f(x_k) - f(x_k + p_k)}{m_k(0) - m_k(p_k)} = \frac{\textbf{actual reduction}}{\textbf{predicted reduction}}. \tag{B.3}$$

- So, if $r_k < 0$, the new objective value $f(x_k + p_k)$ is greater than the current value $f(x_k)$, so the step must be rejected.
- On the other hand, if $r_k \approx 1$ then the quadratic model is a good approximation to f — so take the step to a new point and expand the trust region radius about the new point, confident that I can **trust** the quadratic model there too.
- If r_k is positive but much less than 1, I take the step but leave the radius unchanged.
- Finally, if r_k is close to zero or negative, I shrink the trust region radius, "stay where I am" and calculate a new step.

The following algorithm describes the process.

Algorithm B.1 (Trust Region)

```
begin
    Choose Δ̄ > 0, Δ_0 ∈ (0, Δ̄), η ∈ [0, 1/4) and k_max
    k ← 0
    while k < k_max do
        Find p_k by (approximately) solving B.2.
        Evaluate r_k from B.3.
        if r_k < 1/4
          then Δ_{k+1} ← 1/4 ‖p_k‖
          else
             if r_k > 3/4 & ‖p_k‖ = Δ_k
               then Δ_{k+1} ← min(2Δ_k, Δ̄)
               else
                  Δ_{k+1} ← Δ_k
             fi
        fi
        if r_k > η then x_{k+1} ← x_k + p_k
                   else x_{k+1} ← x_k
        fi
        k ← k + 1
    end (while)
end
```

To turn this into a practical algorithm, I need to focus on solving the quadratic subproblem (B.2).

B.2 The Cauchy Point

In this Section I first describe how to find the Cauchy point — the most easily calculated approximate solution to the quadratic subproblem (B.2)

B.2.1 Cauchy Point

As we have seen previously, a line search doesn't need to take the optimal step for the method to be globally convergent. In the same way, for a trust region method, it is enough for global convergence purposes to find an approximate solution p_k that lies in the trust region and gives a **sufficient reduction** in the value of the model function.

This sufficient reduction can be expressed using as a benchmark the **Cauchy point** which I refer to as p_k^c and define using the following simple steps:

1. Find the vector p_k^s which minimises a **linear** version of m_k , i.e.

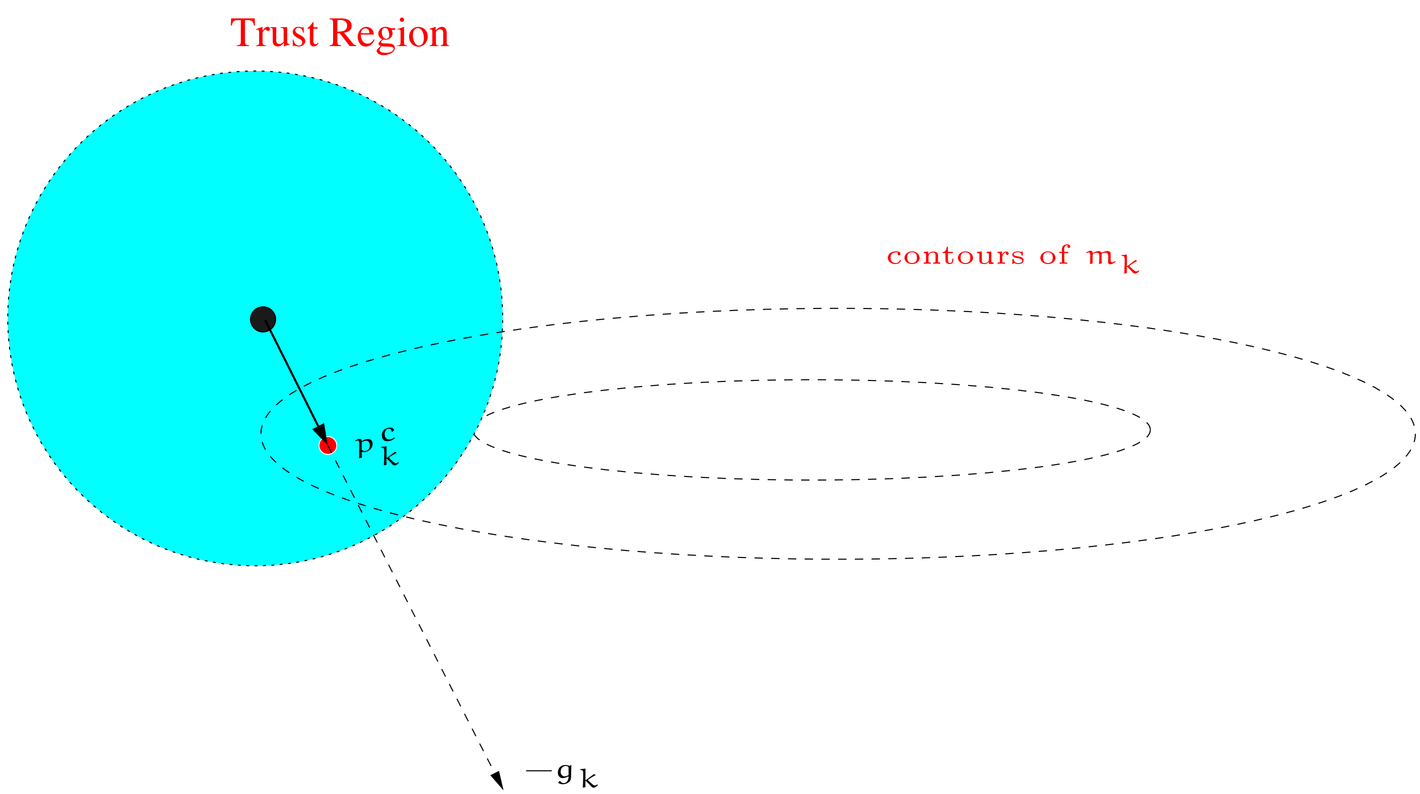
$$p_k^s = \arg\min_{p \in \mathbb{R}^n} f_k + {g_k}^T p, \quad \text{such that } \|p\| \le \Delta_k. \tag{B.4}$$

   Clearly p_k^s is not the "right answer" but a poor approximation to it, so:

2. Calculate the scalar $\tau_k > 0$ that minimises the full quadratic $m_k(\tau p_k^s)$, subject to satisfying the trust region bound, i.e.

$$\tau_k = \arg\min_{\tau > 0} m_k(\tau p_k^s), \quad \text{such that } \|\tau p_k^s\| \le \Delta_k. \tag{B.5}$$

3. Set $p_k^c = \tau_k p_k^s$.



p^s_k and p^c_k point in the direction of the negative gradient: -g_k

Lets remove the index k for clarity

FOR FINDING p^s in (B.4):

Primal problem: $\min g' p$, s.t. $\frac{1}{2} p' p \leq \frac{1}{2} \Delta^2$

$L(p,\lambda) = g' p + \frac{1}{2} \lambda p' p - \frac{1}{2} \lambda \Delta^2$
$dL/dp = g + \lambda p = 0$
$p^s = -1/\lambda\, g$

$2G(\lambda) = 2L(p^s,\lambda) = -2/\lambda\, g' g + 1/\lambda\, g' g - \lambda \Delta^2 = -2/\lambda\, g' g - \lambda \Delta^2$

Dual problem: $\max 2G(\lambda)$ s.t. $\lambda \geq 0$

$2dG/d\lambda = 1/\lambda^2\, g' g - \Delta^2 = 0$
$\lambda = |g| / \Delta$
which fulfills $\lambda \geq 0$

Finally: $p^s = -1/\lambda\, g = -\Delta\, g / |g|$

Note that $|p^s| = \Delta$

Recall $m(p) = f + g' p + \frac{1}{2} p' B p$, and that $p^s = -\Delta g / |g|$

FOR FINDING p^c in (B.5):

Primal problem: $\min m(\tau p^s) = \tau g' p^s + \frac{1}{2} \tau^2 p^{s\prime} B p^s$

s.t. $\tau | p^s | \leq \Delta$

Since $|p^s| = \Delta$, then $\tau \leq 1$ and recall that $\tau > 0$

The problem is quadratic and one-dimensional.

Can be either concave $p^{s\prime} B p^s \leq 0$, or convex $p^{s\prime} B p^s > 0$

Furthermore, note that $p^{s\prime} B p^s = c\, g' B g$, for $c = \Delta^2 / |g|^2 > 0$

Thus $g' B g$ controls whether $p^{s\prime} B p^s$ is positive or not

We put everything together in the next slide.

Recall that:

$$p_k^s = -\frac{\Delta_k}{\|g_k\|} g_k \,, \tag{B.6}$$

To calculate τ_k explicitly, consider the cases ${g_k}^T B_k g_k \leq 0$ and ${g_k}^T B_k g_k > 0$ separately.

- In the first case, the function $m_k(\tau p_k^s)$ decreases monotonically with positive τ provided $g_k \neq 0$.

  So τ_k is just the largest value that satisfies the trust region bound, namely $\tau_k = 1$.

- In the second case, $m_k(\tau p_k^s)$ is a convex quadratic in τ, so τ_k is either the unconstrained minimiser of $m_k(\tau p_k^s)$, $\tau_k = \frac{1}{\Delta_k} \frac{\|g_k\|^3}{{g_k}^T B_k g_k}$; or the boundary value 1, whichever is smaller.

In summary,

$$p_k^c = -\tau_k \frac{\Delta_k}{\|g_k\|} g_k \,, \tag{B.7}$$

where

$$\tau_k = \begin{cases} 1 & \text{if } {g_k}^T B_k g_k \leq 0 \,; \\ \min\left(1, \|g_k\|^3/(\Delta_k {g_k}^T B_k g_k)\right) & \text{otherwise.} \end{cases} \tag{B.8}$$

- The Cauchy point is quick to calculate — no linear systems have to be solved — and is crucial in deciding whether an approximate solution to the trust region subproblem is acceptable.
- It can be formally shown that the trust region method is globally convergent if its steps p_k attain a sufficient reduction in m_k , i.e. they give a reduction in m_k that is at least some fixed multiple of the decrease attained by the Cauchy step at each iteration.
- So the Cauchy point algorithm provides a benchmark against which other methods can be evaluated.
- Improvements include the dogleg method and 2-dimensional subspace minimization (See Chapter 4 of Nocedal & Wright)