SECTION C: CONTINUOUS OPTIMISATION
LECTURE 6: TRUST REGION METHODS

HONOUR SCHOOL OF MATHEMATICS, OXFORD UNIVERSITY
HILARY TERM 2006, DR RAPHAEL HAUSER
WITH A FEW ADDITIONS FROM DR. NICK GOULD

1. Trust Region Methods. All unconstrained optimisation methods we discussed so far in this course are based on line-searches

$$\min_{\alpha>0} f(x_k + \alpha d_k),$$

where d_k is a descent direction. Thus, in effect, in each iteration one replaces the n-dimensional minimisation problem

$$\min_{x\in\mathbb{R}^n} f(x) \tag{1.1}$$

by a simpler one-dimensional minimisation problem. Line-search methods are widely used in practical optimisation codes, but this is not the only useful principle for constructing iterative minimisation algorithms. *Trust region methods* constitute a second fundamental class of algorithms. In this approach (1.1) is again replaced by a sequence of easier problems, but instead of reducing the problem dimension the simplicity is achieved by replacing f with a degree 2 polynomial. Conceptually, the idea can be described as follows:

- In iteration k, replace $f(x)$ by a locally valid quadratic model function $m_k(x)$ (recall that we already encountered this idea in the context of quasi-Newton methods).
- Choose a neighbourhood R_k of the current iterate x_k in which $m_k(x)$ can be trusted to approximate f well (we do not care about how well m_k approximates f outside R_k).
- The next iterate x_{k+1} is found by approximately minimising the model function over the trust region,

$$x_{k+1} \approx \arg\min_{x\in R_k} m_k(x). \tag{1.2}$$

It may seem surprising that we propose to replace the unconstrained optimisation problem (1.1) by the constrained *trust region subproblem* (1.2), as constraints introduce additional difficulties. However, this is worthwhile doing because (1.2) need only be approximately solved, and this can be done efficiently when

$$m_k(x) = f(x_k) + \nabla f(x_k)^{\mathrm{T}}(x-x_k) + \frac{1}{2}(x-x_k)^{\mathrm{T}} B_k (x-x_k) \tag{1.3}$$

is a quadratic function and the trust region R_k is chosen judiciously, see Lecture 7.

The linear part of (1.3) coincides with the first order Taylor approximation of f around x_k, so that $m_k(x)$ will be a good local approximation of $f(x)$ if $B_k \approx D^2 f(x_k)$. To make the method work, we will thus have to worry about how to update B_k cheaply. But note that the quasi-Newton Hessian approximations discussed in Lecture 5 are perfect for this job!

1.1. Accepting and Rejecting Updates. Let y_{k+1} be the approximate minimiser of the trust region subproblem (1.2). In principle, this is the point we would like to select as our next iterate x_{k+1}. However, y_{k+1} is computed on the basis of the model function m_k, and it could happen that moving to y_{k+1} leads to an increase rather than decrease in of the *true* objective function f. Trust-region methods therefore accept y_{k+1} only if the decrease achieved in f is at least a fixed proportion of the decrease "promised" by m_k,

$$x_{k+1} = \begin{cases} y_{k+1} \text{ if } \frac{f(x_k)-f(y_{k+1})}{m_k(x_k)-m_k(y_{k+1})} > \eta, \\ x_k \text{ otherwise,} \end{cases} \tag{1.4}$$

where $\eta \in (0, 1/4)$ is fixed. Note that rejecting the update does not imply that the algorithm will stall, because we can still shrink the trust region so that $y_{k+2} \neq y_{k+1}$.

1.2. Updating the Trust Region. The easiest way to define a trust region R_k is to choose the closed ball of radius Δ_k around x_k in some norm $\|\cdot\|$,

$$R_k = \{x \in \mathbb{R}^n : \|x - x_k\| \le \Delta_k\}.$$

For simplicity, we will assume that $\|\cdot\|$ is the Euclidean norm. Δ_k is called the *trust region radius*.

In order to define a new trust region R_{k+1} around x_{k+1}, it suffices to fix a rule on how to select Δ_{k+1}. The following rule is a popular choice, where y_{k+1} is as in Section 1.1,

$$\Delta_{k+1} = \begin{cases} \frac{\Delta_k}{4} \text{ if } \frac{f(x_k)-f(y_{k+1})}{m_k(x_k)-m_k(y_{k+1})} < \frac{1}{4}, \\ \min(2\Delta_k, \Delta_{\max}) \text{ if } \frac{f(x_k)-f(y_{k+1})}{m_k(x_k)-m_k(y_{k+1})} > \frac{3}{4}, \\ \Delta_k \text{ otherwise.} \end{cases} \tag{1.5}$$

The rule is designed so that Δ_k never exceeds $\Delta_{\max}$, and it is motivated by comparing the objective function decrease $f(x_k) - f(y_{k+1})$ with the decrease $m_k(x_k) - m_k(y_{k+1})$ "promised" by the model function:

- If the actual decrease was below our expectations, this indicates that m_k should be regarded as a more local model than before. We thus find a reasonable Δ_{k+1} by shrinking Δ_k.
- If the actual decrease was above our expectations, we feel confident to expand the trust region by selecting Δ_{k+1} as an expansion of Δ_k.
- If there is neither reason for gloom nor euphoria, we stick to the previous value $\Delta_{k+1} = \Delta_k$.

1.3. The Algorithm. By now we assembled the necessary elements to formulate a generic trust region algorithm:

Algorithm 1.1 (Generic Trust region Method).
S0 *Choose* $\Delta_{\max} > 0$, $\Delta_0 \in (0, \Delta_{\max})$, $\eta \in (0, 1/4)$, $x_0 \in \mathbb{R}^n$, B_0, $\epsilon > 0$.
S1 *While* $\|\nabla f(x_k)\| \ge \epsilon$ *repeat*
    Compute y_{k+1} *as the approximate minimiser of* (1.2).
    Determine x_{k+1} *via* (1.4).

Compute Δ_{k+1} *using* (1.5).
Build a new model function $m_{k+1}(x)$.
$k \leftarrow k+1$.
end
S2 *Return* x_k.

2. The Cauchy Point. In step **S1** of the algorithm, the approximate minimiser y_{k+1} can be computed in many different ways. Some of these methods will be discussed in Lecture 7. We intend to use the remaining part of the present section to derive a rather general convergence result for Algorithm 1.1, see Section 3 below. For this to work out, we need to assume that the method chosen for computing y_{k+1} compares favourably to a specific benchmark, the so-called *Cauchy point*. This point is obtained when a steepest descent line-search is applied to m_k at x_k and is restricted to R_k.

An unrestricted line-search in the direction $-\nabla f(x_k)$ yields the step-length multiplier

$$\begin{aligned}
\alpha_k^u &:= \arg\min_{\alpha\geq 0} m_k(x_k - \alpha\nabla f(x_k)) \\
&= \arg\min_{\alpha\geq 0} f(x_k) - \alpha\nabla f(x_k)^{\mathrm{T}}\nabla f(x_k) + \frac{\alpha^2}{2}\nabla f(x_k)^{\mathrm{T}}B_k\nabla f(x_k) \\
&= \begin{cases} +\infty \text{ if } \nabla f(x_k)^{\mathrm{T}}B_k\nabla f(x_k)\leq 0, \\ \frac{\nabla f(x_k)^{\mathrm{T}}\nabla f(x_k)}{\nabla f(x_k)^{\mathrm{T}}B_k\nabla f(x_k)} \text{ otherwise.}\end{cases}
\end{aligned}$$

If we want to stay within R_k we have to "clip" α_k^u to a constrained step-length multiplier α_k^c. Note that $\alpha\mapsto m_k(x_k-\alpha\nabla f(x_k))$ is strictly decreasing on $[0,\alpha_k^u)$. Moreover, the radius $\|x_k-\alpha\nabla f(x_k)\|$ is strictly increasing over the same interval. Therefore, the correct clipping rule is given by

$$\alpha_k^c = \min\left(\frac{\Delta_k}{\|\nabla f(x_k)\|}, \alpha_k^u\right) \tag{2.1}$$

and $y_k^c := x_k - \alpha_k^c\nabla f(x_k)$ is the Cauchy point of the trust region subproblem (1.2).

3. Global Convergence of Trust Region Algorithms. Next we will show that Algorithm 1.1 converges globally.

Theorem 3.1. *Let Algorithm 1.1 be applied to the minimisation of* $f\in C^2(\mathbb{R}^n,\mathbb{R})$, *and for all* k *let* y_{k+1} *be computed such that* $m_k(y_{k+1})\leq m_k(y_k^c)$ *holds. Let there exist* $\beta>0$ *such that for all* k, $\|B_k\|,\|D^2f(x_k)\|\leq\beta$, *and finally, let* $\Delta_0\geq\epsilon/(14\beta)$. *Then exactly one of two following alternatives occurs:*

(i) The algorithm does not terminate, but $\lim_{k\to\infty} f(x_k) = -\infty$ *and* f *is unbounded below.*

(ii) The algorithm terminates in finite time, returning an approximate minimiser.

Proof. If $\|\nabla f(x_k)\|<\epsilon$ occurs for some $k\in\mathbb{N}$ then we are in case (ii) and nothing needs to be proven. We may therefore assume that $\|\nabla f(x_k)\|\geq\epsilon$ for all k, and it remains to show that this assumption implies $f(x_k)\to-\infty$.

Claim 1: The update is accepted, i.e., $x_{k+1}=y_{k+1}$ in (1.4), for infinitely many k.

Claim 2: Whenever $x_{k+1}=y_{k+1}$ occurs, we have $f(x_{k+1})-f(x_k)\leq -\eta\epsilon^2/(28\beta)$.

Claim 1 follows from Proposition 3.2 below; for Claim 2 see Problem Set 3. It follows from these two claims that

$$\lim_{k\to\infty} f(x_k) = \sum_{k=0}^{\infty} f(x_{k+1})-f(x_k) = -\infty,$$

since (1.4) guarantees that the series on the right hand side contains only nonpositive terms. ◻

We now set out to showing the validity of Claim 1. Intuitively it is clear that when $\|\nabla f(x_k)\|$ is bounded below and Δ_k becomes sufficiently small, then $f(y_{k+1})-f(x_k)\approx m_k(y_{k+1})-m_k(x_k)$ should hold. Indeed, in Lemma 3.5 below we will show that $\|\nabla f(x_k)\|\geq\epsilon$ and $\Delta_k<2\epsilon/(7\beta)$ imply

$$\frac{f(y_{k+1})-f(x_k)}{m_k(y_{k+1})-m_k(x_k)} > \frac{1}{4}. \tag{3.1}$$

Claim 1 then follows immediately from the following result:

Proposition 3.2. *There are at most* $\lfloor\log_4\frac{\Delta_{\max}7\beta}{2\epsilon}\rfloor$ *rejected updates between successive accepted updates.*

Proof. Suppose to the contrary that all updates y_{k+1} for $k=k_0,k_0+1,\dots,k_0+\lceil\log_4\frac{\Delta_{\max}7\beta}{2\epsilon}\rceil =: k_1$ are rejected. Then

$$\Delta_{k_1} = \Delta_{k_0}4^{-(k_1-k_0)} \leq \frac{2\epsilon}{7\beta},$$

and (3.1) contradicts our assumption that that y_{k_1+1} is rejected. ◻

It remains to prove (3.1). We divide the argument into several lemmas.

Lemma 3.3. *Let* $\|\nabla f(x_k)\|\geq\epsilon$ *and* $\Delta_k<\epsilon/\beta$. *Then*

$$y_k^c = x_k - \frac{\Delta_k}{\|\nabla f(x_k)\|}\nabla f(x_k). \tag{3.2}$$

Proof. If $\nabla f(x_k)^{\mathrm{T}}B_k\nabla f(x_k)\leq 0$ then (3.2) holds because of (2.1). So, we may assume that $\nabla f(x_k)^{\mathrm{T}}B_k\nabla f(x_k)>0$, and then

$$\Delta_k < \frac{\epsilon}{\beta} < \frac{\|\nabla f(x_k)\|}{\beta} = \frac{\|\nabla f(x_k)\|^3}{\beta\|\nabla f(x_k)\|^2} \leq \frac{\|\nabla f(x_k)\|^3}{\nabla f(x_k)^{\mathrm{T}}B_k\nabla f(x_k)},$$

But this implies that

$$\frac{\Delta_k}{\|\nabla f(x_k)\|} < \frac{\nabla f(x_k)^{\mathrm{T}}\nabla f(x_k)}{\nabla f(x_k)^{\mathrm{T}}B_k\nabla f(x_k)}.$$

The result now follows from (2.1). ◻

Lemma 3.4. *Let* $\|\nabla f(x_k)\|\geq\epsilon$ *and* $\Delta_k<\epsilon/(2\beta)$. *Then*

$$\nabla f(x_k)^{\mathrm{T}}(y_{k+1}-x_k) \leq -\frac{\Delta_k\|\nabla f(x_k)\|}{2}.$$

Proof. The relation $\Delta_k < \frac{\epsilon}{2\beta} \le \frac{\|\nabla f(x_k)\|}{2\beta}$ implies that

$$-\Delta_k \|\nabla f(x_k)\| + \Delta_k^2 \beta \le -\frac{\Delta_k \|\nabla f(x_k)\|}{2}. \tag{3.3}$$

Moreover, by Lemma 3.3, $\Delta_k < \frac{\epsilon}{2\beta} < \frac{\epsilon}{\beta}$ implies $y_k^c = x_k - \frac{\Delta_k}{\|\nabla f(x_k)\|}\nabla f(x_k)$, and hence,

$$m_k(y_k^c) = f(x_k) - \Delta_k \|\nabla f(x_k)\| + \frac{\Delta_k^2}{2} \frac{\nabla f(x_k)^{\mathrm{T}} B_k \nabla f(x_k)}{\|\nabla f(x_k)\|^2} \tag{3.4}$$

The assumption $m_k(y_{k+1}) \le m_k(y_k^c)$ from Theorem 3.1 implies

$$f(x_k) + \nabla f(x_k)^{\mathrm{T}}(y_{k+1} - x_k) + \frac{1}{2}(y_{k+1} - x_k)^{\mathrm{T}} B_k (y_{k+1} - x_k) \overset{(3.4)}{\le} f(x_k) - \Delta_k \|\nabla f(x_k)\| + \frac{\Delta_k^2}{2} \frac{\nabla f(x_k)^{\mathrm{T}} B_k \nabla f(x_k)}{\|\nabla f(x_k)\|^2},$$

so that

$$\begin{aligned}
\nabla f(x_k)^{\mathrm{T}}&(y_{k+1} - x_k) \\
&\le -\Delta_k \|\nabla f(x_k)\| + \frac{\Delta_k^2}{2} \frac{\nabla f(x_k)^{\mathrm{T}} B_k \nabla f(x_k)}{\|\nabla f(x_k)\|^2} - \frac{1}{2}(y_{k+1} - x_k)^{\mathrm{T}} B_k (y_{k+1} - x_k) \\
&\le -\Delta_k \|\nabla f(x_k)\| + \Delta^2 \beta \\
&\overset{(3.3)}{\le} -\frac{\Delta_k \|\nabla f(x_k)\|}{2}.
\end{aligned}$$

□

LEMMA 3.5. *Let $\|\nabla f(x_k)\| \ge \epsilon$ and $\Delta_k < 2\epsilon/(7\beta)$. Then*

$$\frac{f(y_{k+1}) - f(x_k)}{m_k(y_{k+1}) - m_k(x_k)} > \frac{1}{4}.$$

Proof. We have

$$\begin{aligned}
\Delta_k < \frac{2\epsilon}{7\beta} \le \frac{2\|\nabla f(x_k)\|}{7\beta} &\Rightarrow \beta\Delta_k < \frac{\|\nabla f(x_k)\|}{4} + \frac{\beta\Delta_k}{8} \\
&\Rightarrow \frac{\beta\Delta_k}{\|\nabla f(x_k)\| + \frac{1}{2}\beta\Delta_k} < \frac{1}{4} \\
&\Rightarrow \frac{\frac{1}{2}\|\nabla f(x_k)\|\Delta_k - \frac{1}{2}\beta\Delta_k^2}{\|\nabla f(x_k)\|\Delta_k + \frac{1}{2}\beta\Delta_k^2} = \frac{1}{2} - \frac{\beta\Delta_k}{\|\nabla f(x_k)\| + \frac{1}{2}\beta\Delta_k} > \frac{1}{4}.
\end{aligned} \tag{3.5}$$

On the other hand, since $\Delta_k < 2\epsilon/7\beta < \epsilon/2\beta$, Lemma 3.3 shows that

$$\begin{aligned}
0 < m_k(x_k) - m_k(y_{k+1}) &= \nabla f(x_k)^{\mathrm{T}}(x_k - y_{k+1}) - \frac{1}{2}(y_{k+1} - x_k)^{\mathrm{T}} B_k (y_{k+1} - x_k) \\
&\le \nabla f(x_k)^{\mathrm{T}}(x_k - y_{k+1}) + \frac{1}{2}\beta\Delta_k^2 \le \|\nabla f(x_k)\|\Delta_k + \frac{1}{2}\beta\Delta_k^2.
\end{aligned}$$

Furthermore, applying the mean value theorem (twice), we find

$$f(x_k) - f(y_{k+1}) = \nabla f(x_k)^{\mathrm{T}}(x_k - y_{k+1}) - \frac{1}{2}(y_{k+1} - x_k)^{\mathrm{T}} H (y_{k+1} - x_k),$$

where $H = D^2 f(z)$ for some $z \in \operatorname{conv}(x_k, y_{k+1}) \subset R_k$. Lemma 3.4 therefore implies

$$f(x_k) - f(y_{k+1}) \ge \nabla f(x_k)^{\mathrm{T}}(x_k - y_{k+1}) - \frac{1}{2}\beta\Delta_k^2 \ge \frac{1}{2}\|\nabla f(x_k)\|\Delta_k - \frac{1}{2}\beta\Delta_k^2.$$

Therefore,

$$\frac{f(x_k) - f(y_{k+1})}{m_k(x_k) - m_k(y_{k+1})} \ge \frac{\frac{1}{2}\|\nabla f(x_k)\|\Delta_k - \frac{1}{2}\beta\Delta_k^2}{\|\nabla f(x_k)\|\Delta_k + \frac{1}{2}\beta\Delta_k^2} \overset{(3.5)}{>} \frac{1}{4}.$$

□