SECTION C: CONTINUOUS OPTIMISATION
LECTURE 7: THE DOGLEG AND STEIHAUG METHODS

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

1. Variants of Trust-Region Methods. The generic trust region method we introduced in Lecture 6 is a fairly general algorithmic framework:

(i) Although we made a specific choice for defining and updating the trust region R_k, other choices are possible, for example by considering balls in the norms $\|\cdot\|_1$ or $\|\cdot\|_\infty$. We will not pursue this matter further.
(ii) There is freedom in the choice of the model function m_k. We chose to investigate only quadratic model functions whose linear part coincides with the first order Taylor approximation of f, but this leaves many possibilities for choosing the matrix B_k. We discuss this issue in Section 2 below.
(iii) The point y_{k+1} should be obtained via an approximate solution of the trust region subproblem

$$\min_{y\in R_k} m_k(y). \tag{1.1}$$

Theorem 1.2 of Lecture 6 shows that it is desirable to choose an approximate computation that uses the Cauchy point as a benchmark, but other than that there is complete freedom in choosing a method for this computation. Two of the most widely used methods in this context are the dogleg method of Section 3.1 and Steihaug's method of Section 3.2.

2. Choice of the model function. Let us discuss a few methods for choosing the matrix B_k that determines the model function

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

2.1. Trust-Region Newton Methods. If the problem dimension is not too large, the choice

$$B_k = D^2 f(x_k)$$

is reasonable and leads to a model function m_k that is simply the second order Taylor approximation of the objective function f around the current iterate x_k. Methods based on this choice of model function are called *trust-region Newton methods*.

It is important to understand that trust-region Newton methods are not simply the Newton-Raphson method with an additional step-size restriction. In fact, trust-region Newton methods overcome most of the unwanted aspects of the dynamical behaviour of the Newton-Raphson method while retaining all its advantages with regards to convergence speed:

(i) In the neighbourhood of a saddle point or a local maximiser x^* of f, the Newton-Raphson method is attracted to x^*. This is unwanted, because x^* is a spurious solution of the minimisation problem $\min f(x)$. Trust-region Newton methods are not attracted to such solutions because the trust-region framework ensures that the sequence $(f(x_k))_{\mathbb{N}}$ is decreasing.
(ii) The Newton-Raphson update $x_{k+1} = x_k - D^2 f(x_k)^{-1}\nabla f(x_k)$ is not defined when the Hessian $D^2 f(x_k)$ is singular. However, the trust-region subproblem (1.1) is still well-defined and y_{k+1} can be computed.
(iii) Even in situations where the Newton-Raphson update x_{k+1} is well-defined and $f(x_{k+1}) < f(x_k)$, y_{k+1} may still differ from x_{k+1} because x_{k+1} can lie outside the trust region R_k.
(iv) When x_k enters a sufficiently small neighbourhood of a local minimiser x^* of f where $D^2 f(x^*) \succ 0$, the updates $x_{k+1},\dots$ generated by trust-region Newton methods start coinciding with those produced by the Newton-Raphson method. The two approaches have therefore the same asymptotic convergence rate which is Q-quadratic.

2.2. Quasi-Newton Trust-Region Methods. When the problem dimension n is large, the natural choice for the model function m_k is to use quasi-Newton updates for the approximate Hessians B_k. The only difference is that x_{k+1} is now obtained by approximately solving the trust region subproblem (1.1) rather than by a line-search. Again, these methods are qualitatively different from the corresponding quasi-Newton line-search methods:

(i) When B_k is updated using the SR1 rule (see Lecture 4), it is not guaranteed to be positive definite for all k. This poses a problem for the SR1 line-search method that depends on solving the linear system $B_k d_k = -\nabla f(x_k)$, because d_k may fail to be a descent direction or B_k may be nearly singular. In contrast, the SR1 trust-region method is not affected by this, because the trust-region subproblem (1.1) is still well defined and an approximate minimiser y_{k+1} can be obtained via Steihaug's method (see Section 3.2).
(ii) When x_k enters a sufficiently small neighbourhood of a local minimiser x^* of f, the output sequences produced by quasi-Newton trust-region methods and their line-search counterparts again start coinciding, and the asymptotic convergence rate is Q-superlinear for both approaches.

Note: (i) shows that while there are good reasons to prefer BFGS to SR1 updates in line-search methods, there is no such obvious choice when it comes to quasi-Newton trust-region methods. In fact, when the approximate solver of the trust-region subproblem does not depend on B_k to be positive definite, SR1 updates are preferable because they are allowed to become indefinite and can model the true Hessian $D^2 f(x_k)$ better. Moreover, they are cheaper to evaluate.

3. Solving the Trust-Region Subproblem. In this section we will discuss two of the most widely used methods for computing an approximate minimiser y_{k+1} of the trust-region subproblem (1.1).

3.1. The Dogleg Method. This method is very simple and cheap to compute, but it works only when B_k is positive definite. Therefore, when this approach is used in connection with quasi-Newton trust-region methods, BFGS updates for B_k are a good choice, but SR1 updates are not.

The method is motivated as follows: consider the exact solution of the trust region subproblem as a function of the trust region radius,

$$x(\Delta) = \arg \min_{\{x\in\mathbb{R}^n : \|x-x_k\|\le\Delta\}} m_k(x). \tag{3.1}$$

If $B_k \succ 0$ then $\Delta \mapsto x(\Delta)$ describes a curvilinear path from $x(0) = x_k$ to the exact minimiser of the unconstrained problem $\min_{x\in\mathbb{R}^n} m_k(x)$, that is, to the quasi-Newton point

$$y_k^{qn} = x_k - B_k^{-1}\nabla f(x_k). \tag{3.2}$$

Moreover, we have $x(\Delta) = y_k^{qn}$ for all $\Delta \ge \|y_k^{qn} - x_k\|$, see Lemma 3.1 (iii) below.

The dogleg idea is to replace the curvilinear path $\Delta \mapsto x(\Delta)$ by a polygonal path $\tau \mapsto y(\tau)$ and to determine y_{k+1} as the minimiser of $m_k(y)$ among the points on the path $\{y(\tau) : \tau \ge 0\}$. That is, $y_{k+1} = y(\tau_k)$, where $\tau_k = \arg\min_{\tau\ge 0} m_k(y(\tau))$. We call y_{k+1} the dogleg minimiser.

The simplest and most interesting version of such a method works with a polygon consisting of just two line segments, which reminds some people of the leg of a dog. The "knee" of this leg is located at the steepest descent minimiser $y_k^u = x_k - \alpha_k^u \nabla f(x_k)$, where α_k^u is defined as in Lecture 6. Note that unless x_k is a stationary point, we have $\nabla f(x_k)^{\mathrm{T}} B_k \nabla f(x_k) > 0$, and hence

$$y_k^u = x_k - \frac{\|\nabla f(x_k)\|^2}{\nabla f(x_k)^{\mathrm{T}} B_k \nabla f(x_k)} \nabla f(x_k), \tag{3.3}$$

as shown in Lecture 6. From y_k^u the dogleg path continues along a straight line segment to the quasi-Newton minimiser y_k^{qn}, see Figures 3.1 and 3.2

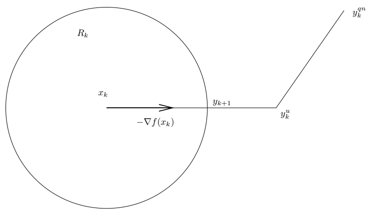

Fig. 3.1. *The dogleg path in the case where y_{k+1} lies on the first section of the leg.*

The dogleg path is thus described by

$$y(\tau) = \begin{cases} x_k + \tau(y_k^u - x_k) & \text{for } \tau \in [0,1], \\ y_k^u + (1-\tau)(y_k^{qn} - y_k^u) & \text{for } \tau \in [1,2]. \end{cases} \tag{3.4}$$

This choice of path is motivated in part by the following lemma.

Lemma 3.1. *If B_k is positive definite symmetric, then*

i) the model function m_k is strictly decreasing along the path $y(\tau)$,

ii) $\|y(\tau) - x_k\|$ is strictly increasing along the path $y(\tau)$,

iii) if $\Delta \ge \|B_k^{-1}\nabla f(x_k)\|$ then $y(\Delta) = y_k^{qn}$,

iv) if $\Delta \le \|B_k^{-1}\nabla f(x_k)\|$ then $\|y(\Delta) - x_k\| = \Delta$,

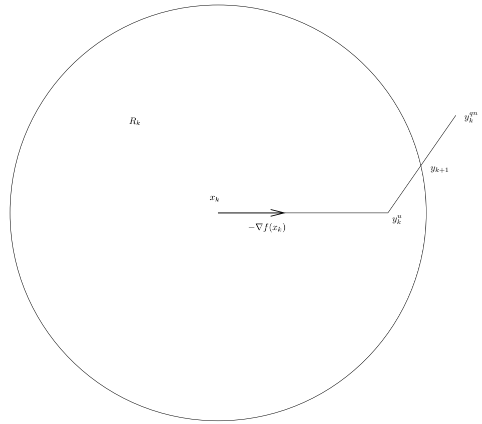

Fig. 3.2. *The dogleg path in the case where y_{k+1} lies on the second section of the leg.*

v) the two paths $x(\Delta)$ and $y(\tau)$ have first order contact at x_k, that is, the derivatives at $\Delta = 0$ are colinear:

$$\lim_{\Delta\to 0+} \frac{x(\Delta) - x_k}{\Delta} = -\frac{\nabla f(x_k)}{\|\nabla f(x_k)\|} \sim \frac{-\|\nabla f(x_k)\|^2}{\nabla f(x_k)^{\mathrm{T}} B_k \nabla f(x_k)} \nabla f(x_k)$$
$$= \lim_{\tau\to 0+} \frac{y(\tau) - y(0)}{\tau}.$$

Proof. See Problem Set 4. ∎

Parts i) and ii) of the Lemma show that the dogleg minimiser y_{k+1} is easy to compute: if $y_k^{qn} \in R_k$ then $y_{k+1} = y_k^{qn}$, and otherwise y_{k+1} is the unique intersection point of the dogleg path with the boundary of R_k, see Figures 3.1 and 3.2. The dogleg calculation of y_{k+1} can thus be summed up as follows:

Algorithm 3.2 (Dogleg).
compute y_k^u as in (3.3)
if $\|y_k^u - x_k\| \ge \Delta_k$ stop with $y_{k+1} = x_k + \frac{\Delta_k}{\|y_k^u - x_k\|}(y_k^u - x_k)$ (*)
compute y_k^{qn} as in (3.2)
if $\|y_k^{qn} - x_k\| \le \Delta_k$ stop with $y_{k+1} = y_k^{qn}$
else begin
  find τ^* s.t. $\|y_k^u + \tau^*(y_k^{qn} - y_k^u) - x_k\| = \Delta_k$
  stop with $y_{k+1} = y_k^u + \tau^*(y_k^{qn} - y_k^u)$
end

If the algorithm stops in (*) then the dogleg minimiser lies on the first part of the leg and equals the Cauchy point. Otherwise the dogleg minimiser lies on the second part of the leg and is better than the Cauchy point. Therefore, we have

$m_k(y_{k+1}) \le m_k(y_k^c)$ in both cases, and Theorem 3.1 of Lecture Note 6 can be applied.

3.2. Steihaug's Method. This is the most widely used method for the approximate solution of the trust-region subproblem. The method works for quadratic models m_k defined by an arbitrary symmetric B_k. Positive definiteness is therefore not required and SR1 updates can be used for B_k.

One of the strengths of the dogleg method is that the method takes the quasi-Newton step $y_{k+1} = y_k^{qn}$ when y_k^{qn} lies in the trust region. If B_k converges to $D^2 f(x^*) \succ 0$ as x_k approaches a strict local minimiser x^* of f, this allows $(x_k)_{\mathbb{N}}$ to converge Q-superlinearly. Steihaug's method is designed to inherit this desirable property. However, when B_k is not positive definite, it is not necessarily desireable to move to y_k^{qn} because $m_k(y_k^{qn})$ might be larger than $m_k(x_k) = f(x_k)$. Steihaug's method overcomes this problem as follows:

- Draw the polygon traced by the iterates $x_k = z_0, z_1, \dots, z_j, \dots$ obtained by applying the conjugate gradient algorithm to the minimisation of the quadratic function $m_k(x)$ for as long as the updates are defined, i.e., as long as $d_j^{\mathrm{T}} B_k d_j > 0$.
- This terminates in the quasi-Newton point $z_n = y_k^{qn}$, unless $d_j^{\mathrm{T}} B_k d_j \le 0$. In the second case, continue to draw the polygon from z_j to infinity along d_j, as m_k can be pushed to $-\infty$ along that path.
- Minimise m_k along this polygon and select y_{k+1} as the minimiser.

The polygon is constructed so that $m_k(z)$ decreases along its path, while Theorem 3.4 below shows that $\|z - x_k\|$ increases. Therefore, if the polygon ends at $z_n \in R_k$ then $y_{k+1} = z_n$, and otherwise y_{k+1} is the unique point where the polygon crosses the boundary ∂R_k of the trust region. Stated more formally, Steighaug's method proceeds as follows:

Algorithm 3.3 (Steihaug).

S0 Initialisation:
  choose tolerance parameter $\epsilon > 0$
  set $z_0 = x_k$, $d_0 = -\nabla m_k(x_k)$
S1 For $j = 0, \dots, n-1$ *repeat*
  if $d_j^{\mathrm{T}} B_k d_j \le 0$ *begin*
    find $\tau^* \ge 0$ *s.t.* $\|z_j + \tau^* d_j - x_k\| = \Delta_k$
    stop with $y_{k+1} = z_j + \tau^* d_j$
  end
  else begin
    find $\tau_j := \arg\min_{\tau \ge 0} m_k(z_j + \tau d_j)$
    set $z_{j+1} := z_j + \tau_j d_j$
    if $\|z_{j+1} - x_k\| \ge \Delta_k$ *begin*
      find $\tau^* \ge 0$ *s.t.* $\|z_j + \tau^* d_j - x_k\| = \Delta_k$
      stop with $y_{k+1} = z_j + \tau^* d_j$
    end
    if $\|\nabla m_k(z_{j+1})\| \le \epsilon$ *stop with* $y_{k+1} = z_{j+1}$ *(*)*
    else compute $d_{j+1} = -\nabla m_k(z_{j+1}) + \frac{\|\nabla m_k(z_{j+1})\|^2}{\|\nabla m_k(z_j)\|^2} d_j$
  end
end

Algorithm 3.3 stops with $y_{k+1} = z_n$ in iteration $n-1$ at the latest, since in this case $d_j^{\mathrm{T}} B_k d_j > 0$ for $j = 0, \dots, n-1$ and this implies that $B_k \succ 0$. Furthermore, since $d_0 = -\nabla m_k(x_k) = -\nabla f(x_k)$, the algorithm stops at the Cauchy point $y_{k+1} = y_k^c$ if it stops in iteration 0, and if it stops later then $m_k(y_{k+1}) < m_k(y_k^c)$. Therefore, Theorem 3.1 of Lecture Note 6 is applicable.

Theorem 3.4. *Let the conjugate gradient algorithm be applied to the minimisation of $m_k(x)$ with starting point $z_0 = x_k$, and suppose that $d_j^{\mathrm{T}} B_k d_j > 0$ for $j = 0, \dots, i$. Then we have $0 = \|z_0 - x_k\| \le \|z_1 - x_k\| \le \dots \le \|z_i - x_k\|$.*

Proof. The restriction of B_k to $\mathrm{span}\{d_0, \dots, d_i\}$ is a positive definite operator,

$$\Big(\sum_{j=0}^{i} \lambda_j d_j\Big)^{\mathrm{T}} B_k \Big(\sum_{j=0}^{i} \lambda_j d_j\Big) = \sum_{j=0}^{i} \lambda_j^2 d_j^{\mathrm{T}} B_k d_j > 0,$$

where we used the B_k-conjugacy property $d_j^{\mathrm{T}} B_k d_l = 0 \ \forall\, j \ne l$. Therefore, up to iteration i all the properties we derived for the conjugate gradient algorithm are valid. Since $z_j - x_k = \sum_{l=0}^{j-1} \tau_l d_l$ for $(j = 1, \dots, i)$, we have

$$\|z_{j+1} - x_k\|^2 = \|z_j - x_k\|^2 + \sum_{l=0}^{j-1} \tau_j \tau_l d_j^{\mathrm{T}} d_l.$$

Moreover, $\tau_j > 0$ for all j. Therefore, it suffices to show that $d_j^{\mathrm{T}} d_l > 0$ for all $l \le j$. For $j = 0$ this is trivially true. We can thus assume that the claim holds for $j-1$ and proceed by induction. For $l < j$ have

$$d_j^{\mathrm{T}} d_l = -\nabla m_k(z_j)^{\mathrm{T}} d_l + \frac{\|\nabla m_k(z_j)\|^2}{\|\nabla m_k(z_{j-1})\|^2} d_{j-1}^{\mathrm{T}} d_l.$$

The second term on the right-hand side is positive because of the induction hypothesis, and it was established in the proof of Lemma 2.3 from Lecture 5 that the first term is zero. Furthermore, if $l = j$ then we have of course $d_j^{\mathrm{T}} d_l > 0$. ▯