Chapter 9

Line-search methods based on simplex derivatives

The implicit-filtering method of Kelley et al. in [229] (see also [141]) can be viewed as a line-search method based on the simplex gradient. In this chapter, we present a modified version of the implicit-filtering method that is guaranteed to be globally convergent to first-order stationary points of (1.1). The main modification relies on the application of the backtracking scheme used to achieve sufficient decrease, which can be shown to eventually be successful. Under such a modification, this derivative-free algorithm can be seen as a line-search counterpart of the trust-region derivative-free method considered in Chapter 10.

9.1 A line-search framework

For simplicity let us consider, at each iteration k of the algorithm, a sample set $Y_k = \{y_k^0, y_k^1, \ldots, y_k^n\}$, formed by $n+1$ points. We will assume that this set is poised in the sense of linear interpolation. In other words, the points $y_k^0, y_k^1, \ldots, y_k^n$ are assumed to be the vertices of a simplex set.

Now we consider the simplex gradient based at y_k^0 and computed from this sample set. Such a simplex gradient is given by

$$\nabla_s f(x_k) \;=\; L_k^{-1} \delta f(Y_k),$$

where

$$L_k \;=\; \left[y_k^1 - y_k^0 \cdots y_k^n - y_k^0 \right]^\top$$

and

$$\delta f(Y_k) \;=\; \begin{bmatrix} f(y_k^1) - f(y_k^0) \\ \vdots \\ f(y_k^n) - f(y_k^0) \end{bmatrix}.$$

Let us also define

$$\Delta_k \;=\; \max_{1 \le i \le n} \|y_k^i - y_k^0\|.$$

As we will see later in this chapter, other more elaborate simplex gradients could be used, following a regression approach (see also Chapters 2 and 4) or a centered difference scheme.

No matter which simplex gradient calculation is chosen, all we need to ensure is that the following error bound can be satisfied:

$$\|\nabla f(x_k) - \nabla_s f(x_k)\| \leq \kappa_{eg}\, \Delta_k, \tag{9.1}$$

where, as we have seen in Chapters 2 and 3, κ_{eg} is a positive constant depending on the geometry of the sample points. The algorithmic framework presented in Chapter 6 can be used to improve the geometry of the sample set. In order to make the statement of the algorithm simultaneously rigorous and close to a practical implementation, we introduce the following assumption, where by an improvement step of the simplex geometry we mean recomputation of one of the points in the set $\{y_k^1, \ldots, y_k^n\}$.

Assumption 9.1. *We assume that* (9.1) *can be satisfied for some fixed positive* $\kappa_{eg} > 0$ *and for any value of* $\Delta_k > 0$ *in a finite, uniformly bounded number of improvement steps of the sample set.*

Basically, one geometry improvement step consists of applying the algorithms described in Chapter 6 to replace one vertex of the simplex. Thus, in the interpolation case ($n+1$ sample points), for any positive value of Δ_k, the error bound (9.1) can be achieved using at most n improvement steps.

At each iteration of the line-search derivative-free method that follows, a sufficient decrease condition is imposed on the computation of the new point. When using the simplex gradient $\nabla_s f(x_k)$, with $x_k = y_k^0$, this sufficient decrease condition is of the form

$$f(x_k - \alpha \nabla_s f(x_k)) - f(x_k) \leq -\eta\alpha \|\nabla_s f(x_k)\|^2, \tag{9.2}$$

where η is a constant in the interval $(0,1)$ for all k and $\alpha > 0$. The new point x_{k+1} is *in principle* of the form $x_{k+1} = x_k - \alpha_k \nabla_s f(x_k)$, where α_k is chosen by a backtracking procedure to ensure (9.2) with $\alpha = \alpha_k$. However, the line-search version of the method analyzed here considers the possibility of accepting a point different from $x_k - \alpha_k \nabla_s f(x_k)$ as long as it provides a lower objective value.

The line-search derivative-free method is presented below and includes a standard backtracking scheme. However, this line search can fail. When it fails, the size of Δ_k is reduced compared to the size of $\|\nabla_s f(x_k)\|$ (which involves a number of improvement steps and the recomputation of the simplex gradient) and the line search is restarted from the same point (with a likely more accurate simplex gradient).

Algorithm 9.1 (Line-search derivative-free method based on simplex gradients).

Initialization: Choose an initial point x_0 and an initial poised sample set $\{y_0^0 (= x_0), y_0^1, \ldots, y_0^n\}$. Choose β, η, and ω in $(0,1)$. Select $j_{max} \in \mathbb{N}$.

For $k = 0, 1, 2, \ldots$

1. **Simplex gradient calculation:** Compute a simplex gradient $\nabla_s f(x_k)$ such that $\Delta_k \leq \|\nabla_s f(x_k)\|$ (apply Algorithm 9.2 below). Set $j_{current} = j_{max}$ and $\mu = 1$.

2. **Line search:** For $j = 0, 1, 2, \ldots, j_{current}$
   (a) Set $\alpha = \beta^j$. Evaluate f at $x_k - \alpha \nabla_s f(x_k)$.
   (b) If the sufficient decrease condition (9.2) is satisfied for α, then stop this step with $\alpha_k = \alpha$ (and go to Step 4).

3. **Line-search failure:** If the line search failed, then divide μ by two, recompute a simplex gradient $\nabla_s f(x_k)$ such that $\Delta_k \leq \mu \|\nabla_s f(x_k)\|$ (apply Algorithm 9.2 below), increase $j_{current}$ by one, and repeat the line search (go back to Step 2).

4. **New point:** Set

$$x_{k+1} = \operatorname*{argmin}_{x \in \mathcal{X}_k} \{f(x_k - \alpha_k \nabla_s f(x_k)), f(x)\},$$

   where $\mathcal{X}_k$ is the set of points where f has possibly been evaluated during the course of Steps 1 and 3. Set $y^0_{k+1} = x_{k+1}$. Update $y^1_{k+1}, \ldots, y^n_{k+1}$ from $y^0_k, y^1_k, \ldots, y^n_k$ by dropping one of these points.

A possible stopping criterion is to terminate the run when Δ_k becomes smaller than a chosen tolerance $\Delta_{tol} > 0$ (for instance $\Delta_{tol} = 10^{-5}$).

The algorithm that recomputes the sample set at x_k and the corresponding simplex gradient such that $\Delta_k \leq \mu \|\nabla_s f(x_k)\|$ is described next.

Algorithm 9.2 (Criticality step). *This algorithm is applied only when* $\Delta_k > \mu \|\nabla_s f(x_k)\|$. *The constant* $\omega \in (0, 1)$ *should be chosen in the initialization of Algorithm* 9.1.

Initialization: Set $i = 0$. Set $\nabla_s f(x_k)^{(0)} = \nabla_s f(x_k)$.

Repeat Increment i by one. Compute a new simplex gradient $\nabla_s f(x_k)^{(i)}$ based on a sample set containing x_k and contained in $B(x_k; \omega^i \mu \|\nabla_s f(x_k)^{(0)}\|)$ such that

$$\|\nabla f(x_k) - \nabla_s f(x_k)^{(i)}\| \leq \kappa_{eg} \left(\omega^i \mu \|\nabla_s f(x_k)^{(0)}\|\right)$$

(notice that this can be done in a finite, uniformly bounded number of steps). Set $\Delta_k = \omega^i \mu \|\nabla_s f(x_k)^{(0)}\|$ and $\nabla_s f(x_k) = \nabla_s f(x_k)^{(i)}$.

Until $\Delta_k \leq \mu \|\nabla_s f(x_k)^{(i)}\|$.

9.2 Global convergence for first-order critical points

We need to assume that ∇f is Lipschitz continuous on the level set (where the iterates must necessarily lie):

$$L(x_0) = \{x \in \mathbb{R}^n : f(x) \leq f(x_0)\}.$$

However, we also need to take into account that the points used in the simplex gradient calculations might lie outside $L(x_0)$, especially at the early iterations of the method. Thus, we

need to enlarge $L(x_0)$. If we impose that Δ_k never exceeds a given positive constant Δ_{max}, then all points used by the algorithm will lie in the following set:

$$L_{enl}(x_0) \;=\; L(x_0) \cup \bigcup_{x \in L(x_0)} B(x; \Delta_{max}) \;=\; \bigcup_{x \in L(x_0)} B(x; \Delta_{max}).$$

We will assume that f is a continuously differentiable function in an open set containing $L_{enl}(x_0)$ and that ∇f is Lipschitz continuous on $L_{enl}(x_0)$ with constant $\nu > 0$.

The purpose of the first lemma is to show that Steps 1 and 3 of Algorithm 9.1 are well defined, in the sense that Algorithm 9.2 will take a finite number of steps.

Lemma 9.1. *If $\nabla f(x_k) \neq 0$, Steps 1 and 3 of Algorithm 9.1 will satisfy $\Delta_k \leq \mu \|\nabla_s f(x_k)\|$ in a finite number of improvement steps (by applying Algorithm 9.2).*

Proof. The proof is postponed to Chapter 10 (see Lemma 10.5), where it is done in the trust-region environment, in a slightly more general context. The simplex gradient $\nabla_s f(x_k)$ plays the same role here as the gradient g_k of the model plays there. □

Now we need to analyze under what conditions the sufficient decrease (9.2) is attained and to make sure that the line search in Step 2 of Algorithm 9.1 can be accomplished after a finite number of reductions of α and μ. In other words, we will prove that one cannot loop infinitely between Steps 2 and 3 unless the point is stationary.

Lemma 9.2. *Let f be a continuously differentiable function in an open set containing $L_{enl}(x_0)$. Assume that ∇f is Lipschitz continuous on $L_{enl}(x_0)$ with constant $\nu > 0$. Let x_k be such that $\nabla f(x_k) \neq 0$. The sufficient decrease condition* (9.2) *is satisfied for all α and μ such that*

$$0 \;<\; \alpha \;\leq\; \frac{2(1-\eta-\kappa_{eg}\mu)}{\nu} \tag{9.3}$$

and

$$\mu \;<\; \frac{1-\eta}{\kappa_{eg}}. \tag{9.4}$$

Proof. First, we know that (see, e.g., the proof of Theorem 2.8)

$$\begin{aligned} & f(x_k + \alpha d) - f(x_k) \\ \leq\; & \alpha \nabla f(x_k)^\top d + \tfrac{\nu\alpha^2}{2}\|d\|^2 \\ =\; & \alpha(\nabla f(x_k) - \nabla_s f(x_k))^\top d + \alpha \nabla_s f(x_k)^\top d + \tfrac{\nu\alpha^2}{2}\|d\|^2. \end{aligned}$$

By replacing d by $-\nabla_s f(x_k)$ and using (9.1) and $\alpha > 0$, we obtain

$$\begin{aligned} & f(x_k - \alpha \nabla_s f(x_k)) - f(x_k) \\ \leq\; & \alpha(\kappa_{eg}\Delta_k/\|\nabla_s f(x_k)\| - 1 + \nu\alpha/2)\|\nabla_s f(x_k)\|^2. \end{aligned}$$

Combining this inequality with $\Delta_k \leq \mu\|\nabla_s f(x_k)\|$ and $\alpha > 0$ yields

$$f(x_k - \alpha \nabla_s f(x_k)) - f(x_k) \;\leq\; \alpha(\kappa_{eg}\mu - 1 + \nu\alpha/2)\|\nabla_s f(x_k)\|^2.$$

Thus, the sufficient decrease condition (9.2) is satisfied if

$$\kappa_{eg}\mu - 1 + \frac{\nu\alpha}{2} \leq -\eta,$$

and the proof is completed. □

As a result of Lemma 9.2 and of the scheme in Step 2 to update α, one can guarantee that α_k is bounded from below by

$$\alpha_k \geq \bar{\alpha} = \frac{2(1-\eta-\kappa_{eg}\bar{\mu})\beta}{\nu}, \tag{9.5}$$

where $\bar{\mu}$ is any number such that

$$0 < \bar{\mu} < \frac{1-\eta}{\kappa_{eg}}.$$

The global convergence of Algorithm 9.1 to stationary points is stated in the following theorem.

Theorem 9.3. *Let f be a continuously differentiable function in an open set containing $L_{enl}(x_0)$. Assume that f is bounded from below in $L(x_0)$ and that ∇f is Lipschitz continuous on $L_{enl}(x_0)$ with constant $\nu > 0$. Then the sequence of iterates generated by Algorithm* 9.1 *satisfies*

$$\lim_{k \longrightarrow +\infty} \|\nabla f(x_k)\| = 0. \tag{9.6}$$

Proof. The proof of

$$\lim_{k \longrightarrow +\infty} \|\nabla_s f(x_k)\| = 0 \tag{9.7}$$

follows the classical arguments for line-search methods. The sequence $\{f(x_k)\}$ is decreasing (by construction) and bounded from below in $L(x_0)$ (by hypothesis). Thus, the left-hand side in (9.2) (with $\alpha = \alpha_k$) converges to zero. The limit (9.7) then follows from

$$f(x_{k+1}) - f(x_k) \leq f(x_k - \alpha_k \nabla_s f(x_k)) - f(x_k) \leq -\eta\bar{\alpha}\|\nabla_s f(x_k)\|^2,$$

where $\bar{\alpha}$ is given in (9.5), and the first inequality comes from Step 4 of the algorithm.

Since $\nabla_s f(x_k)$ converges to zero, we know from (9.1) and Steps 1 and 3 of the algorithm that, for k sufficiently large,

$$\|\nabla f(x_k) - \nabla_s f(x_k)\| \leq \kappa_{eg}\|\nabla_s f(x_k)\|.$$

This inequality together with (9.7) implies (9.6). □

9.3 Analysis for noise

In many applications the noise level in the objective function is not zero, and what is evaluated in practice can be represented as

$$f(x) = f_{smooth}(x) + \varepsilon(x), \tag{9.8}$$

where f_{smooth} is an underlying smooth function and $\varepsilon(x)$ represents the noise in its evaluation.

We are interested in extending the analysis of the line-search derivative-free method to a noisy function f of the form (9.8). So, let us consider Algorithm 9.1 applied to the minimization of f in (9.8).

In order to retain global convergence to first-order critical points, we require the noise level in the objective function to satisfy

$$\max_{0\leq i\leq n} |\varepsilon(y_k^i)| \;\leq\; c\,\Delta_k^2 \tag{9.9}$$

for all iterations, where $c > 0$ is independent of k. In addition, since for sufficient decrease purposes one needs to evaluate f at points which might not be used for simplex gradient calculations, we also ask the noise level to satisfy

$$\varepsilon(x_k - \alpha \nabla_s f(x_k)) \;\leq\; c\,\Delta_k^2 \tag{9.10}$$

for all values of α considered in the backtracking scheme.

When the noise level satisfies the two conditions stated above, it is possible to prove, similarly to Theorem 9.3, that the limit of the gradient of the underlying function f_{smooth} converges to zero.

9.4 The implicit-filtering algorithm

The implicit-filtering method of Kelley et al. in [229] (see also [141]) differs from Algorithm 9.1 in a number of aspects. First, the sample set is not dynamically updated on an iteration base. Instead, a new sample set is chosen at each iteration for the purpose of the simplex gradient calculation. Such a choice may make the algorithm appealing for parallel computation but might compromise its efficiency in a serial environment.

No provision is made to link the accuracy of the simplex gradient and the quality of the geometry of the sample set to the line-search scheme itself, as in Algorithm 9.1. Therefore, it is not possible to guarantee success for the line search (which can terminate unsuccessfully after the predetermined finite number of steps j_{max}).

In addition to the basic line-search scheme, implicit filtering incorporates a quasi-Newton update (see, e.g., [76, 178]) for the Hessian approximation based on the simplex gradients that are being computed. When the line search fails the quasi-Newton matrix is reset to the initial choice. The method has been applied to problems with noisy functions, for which it has been shown to be numerically robust. We present it below especially having in mind the case where f is of the form (9.8) and the noise obeys (9.9) and (9.10).

Algorithm 9.3 (Implicit filtering method).

Initialization: Choose β and η in $(0,1)$. Choose an initial point x_0 and an initial Hessian approximation H_0 (for instance, the identity matrix). Select $j_{max} \in \mathbb{N}$.

For $k = 0, 1, 2, \ldots$

1. **Simplex gradient calculation:** Compute a simplex gradient $\nabla_s f(x_k)$ such that $\Delta_k \leq \|\nabla_s f(x_k)\|$. Compute $d_k = -H_k^{-1} \nabla_s f(x_k)$.

2. **Line search:** For $j = 0, 1, 2, \ldots, j_{max}$
   (a) Set $\alpha = \beta^j$. Evaluate f at $x_k + \alpha d_k$.
   (b) If the sufficient decrease condition
   $$f(x_k + \alpha d_k) - f(x_k) \leq \eta \alpha \nabla_s f(x_k)^\top d_k$$
   is satisfied for α, then stop this step with $\alpha_k = \alpha$.
3. **New point:** If the line search succeeded, then $x_{k+1} = x_k + \alpha_k d_k$.
4. **Hessian update:** If the line search failed set $H_{k+1} = H_0$. Otherwise, update H_{k+1} from H_k using a quasi-Newton update based on $x_{k+1} - x_k$ and $\nabla_s f(x_{k+1}) - \nabla_s f(x_k)$.

Other provisions may be necessary to make this algorithm practical when line-search failures occur. For instance, one might have to recompute the sampling points (by scaling them towards x_k so that $\Delta_k \leq \mu \|\nabla_s f(x_k)\|$ for some $\mu > 0$ smaller than one, as in Algorithm 9.1) and to repeat an iteration after a line-search failure.

9.5 Other simplex derivatives

There exist alternatives for the computation of simplex gradients based on $n+1$ points. For example, if the function is evaluated at more than $n+1$ points, one can compute simplex gradients in the regression sense, as explained in Chapter 2. In both cases, the order of accuracy is linear in Δ (the size of the ball containing the points).

Centered simplex gradients

When the number of points is $2n+1$ and the sampling set Y is of the form

$$\left\{y^0, y^0 + (y^1 - y^0), \ldots, y^0 + (y^n - y^0), y^0 - (y^1 - y^0), \ldots, y^0 - (y^n - y^0)\right\}$$

it is possible to compute a centered simplex gradient with Δ^2 accuracy. First, note that the sampling set given above is obtained by retaining the original points and adding their reflection through y^0. This geometrical structure has been seen before:

$$\left[\, y^1 - y^0 \;\cdots\; y^n - y^0 \;\; -(y^1 - y^0) \;\cdots\; -(y^n - y^0) \,\right]$$

forms a (maximal) positive basis (see Section 2.1). When $y^0 = 0$, this sampling set reduces to

$$\left\{0, y^1, \ldots, y^n, -y^1, \ldots, -y^n\right\}.$$

Consider, again, the matrix $L = \left[y^1 - y^0 \cdots y^n - y^0\right]^\top$. The centered simplex gradient is defined by

$$\nabla_{cs} f(y^0) \;=\; L^{-1} \delta_{cs} f(Y),$$

where

$$\delta_{cs} f(Y) = \frac{1}{2} \begin{bmatrix} f(y^0 + (y^1 - y^0)) - f(y^0 - (y^1 - y^0)) \\ \vdots \\ f(y^0 + (y^n - y^0)) - f(y^0 - (y^n - y^0)) \end{bmatrix}.$$

One can easily show that if $\nabla^2 f$ is Lipschitz continuous with constant $\nu > 0$ in an open set containing the ball $B(y^0; \Delta)$, where

$$\Delta = \max_{1 \le i \le n} \|y^i - y^0\|,$$

then

$$\|\nabla f(y^0) - \nabla_{cs} f(y^0)\| \le \kappa_{eg} \Delta^2,$$

where $\kappa_{eg} = n^{\frac{1}{2}} \nu \|\hat{L}^{-1}\|$ and $\hat{L} = L/\Delta$. It is important to stress that this improvement in the order of accuracy of the gradient approximation does not have consequences for second-order approximations. In fact, it is not possible in general, given only a number of points linear in n, to compute a simplex Hessian (see the coming paragraphs) that approximates the true Hessian within an error of the order of Δ.

Simplex Hessians

Given a sample set $Y = \{y^0, y^1, \ldots, y^p\}$, with $p = (n+1)(n+2)/2 - 1$, poised in the sense of quadratic interpolation, one can compute a simplex gradient $\nabla_s f(y^0)$ and a simplex (symmetric) Hessian $\nabla_s^2 f(y^0)$ from the system of linear equations

$$(y^i - y^0)^\top \nabla_s f(y^0) + \frac{1}{2}(y^i - y^0)^\top \nabla_s^2 f(y^0)(y^i - y^0) = f(y^i) - f(y^0), \tag{9.11}$$

$i = 1, \ldots, p$.

One can observe that the simplex gradient and simplex Hessian defined above are nothing else than the coefficients of the quadratic interpolation model $m(x) = c + g^\top x + (1/2) x^\top H x$:

$$\nabla_s f(y^0) = g \quad \text{and} \quad \nabla_s^2 f(y^0) = H.$$

When $p = 2n$, it is possible to neglect all the off-diagonal elements of the Hessian and compute a simplex gradient and a diagonal simplex Hessian.

9.6 Other notes and references

The implicit-filtering algorithm was first described in the already cited paper [229] and later, in more detail, in the journal publications by Stoneking at al. [212] and Gilmore and Kelley [105]. The global convergence properties of the method were analyzed by Bortz and Kelley [41]. Choi and Kelley [52] studied the rate of local convergence.

Mifflin [171] suggested in 1975 a line-search algorithm based on centered simplex gradients and approximated simplex Hessians, for which he proved global convergence to first-order stationary points and studied the rate of local convergence. Mifflin's algorithm is a hybrid approach, sharing features with direct-search methods (use of the coordinate-search directions to compute the simplex derivatives) and with line-search algorithms.

9.7 Exercises

1. Prove that Theorem 9.3 remains true for f_{smooth} when Algorithm 9.1 is applied to f given in (9.8) if conditions (9.9) and (9.10) are satisfied.