Implicit Filtering

C. T. Kelley
NC State University
tim.kelley@ncsu.edu
Joint with Xiaojun Chen
Supported by ARO, NSF, DOE(CASL/LANL)

Copper, April 8, 2014
Outline

1. What is this for?
2. Implicit Filtering
3. Hidden Constraints
4. Embedded Monte Carlo Simulations
5. Example
6. Conclusions
What is the problem?

Ideally we like to solve

$$\min_{\Omega} f(x)$$

where

$$\Omega = \{x \mid L \leq x \leq U\} \subset \mathbb{R}^N$$

First order necessary conditions:

$$x = \mathcal{P}(x - \nabla f(x)), \text{ where } \mathcal{P}(x) = \max(L, \min(x, U)).$$

But we have a few problems...
Implicit Filtering

What is this for?

\( f \) is unfriendly because . . .

- \( f \) is a “black box”, so gradients are not available
- \( f \) is not everywhere defined in \( \Omega \)
  - \( f \) can fail to return a value
  - You get a failure flag instead
- You don’t even get the right \( f \) when you call the function
  - You get an error-infested approximation \( \hat{f} \)

We will deal with these one at a time.
Implicit Filtering

What is this for?

Two Landscapes

![Graph 1](image1.png)

![Graph 2](image2.png)
Implicit Filtering and Coordinate Search

Who needs gradients when you can throw darts?
From a current point $x$ and scale $h$ evaluate $f$ on the stencil

$$S(x, h) = \{ z \mid z = x \pm he_i \} \cap \Omega$$

If you find a better point than $x$, take it.
If the stencil fails to find a better point, i.e.

$$f(x) \leq \min_{z \in S(x, h)} f(z)$$

reduce $h$, say $h \leftarrow h/2$. 
Theory for Coordinate Search: due to many people

If \( f \) is Lipschtiz continuously differentiable and \( \{x_n, h_n\} \) are the points/scales from coordinate search, then

- The stencil fails infinitely often, and so . . .
  - \( h_n \to 0 \)
  - \( \lim \inf \|x_n - P(x_n - \nabla f(x_n))\| = 0. \)

Nice, but it’s as slow as steepest descent.
Implicit Filtering

After the function evaluations on the stencil either

- Shrink $h$ if the stencil fails or . . .
  - build a finite difference gradient
  - maintain a quasi-Newton model Hessian
  - see if the quasi-Newton direction leads to a better point

Much better than coordinate search.
If $f$ is Lipschitz continuously differentiable and $\{x_n, h_n\}$ are the points/scales from implicit filtering, and

- The stencil fails infinitely often then
  - $h_n \to 0$
  - $\lim \inf \|x_n - P(x_n - \nabla f(x_n))\| = 0$.

Note: stencil failure is now an assumption instead of a conclusion. Reason: quasi-Newton point may leave the grid.
Hidden Constraints

\( f \) is defined on \( \mathcal{D} \subset \Omega \)

- You know \( x \notin \mathcal{D} \) when \( f(x) = NaN \).
- The cost of an evaluation of \( f \) for \( x \notin \mathcal{D} \) may vary.
- Sources of hidden constraints
  - failure of internal solvers
  - internal tests and sanity checks
  - stiffness, risk, reliability
  - non-physical intermediate results
First-order Necessary Conditions: Audet-Dennis 06

Assume $\mathcal{D}$ is regular. This means that the Tangent cone

$$T_{\mathcal{D}}^{CL}(x) = \text{cl}\{v \mid x + tv \in \mathcal{D} \text{ for all sufficiently small } t > 0\},$$

is the closure of its non-empty interior.

First-order necessary conditions at $x \in \mathcal{D}$ are

$$\frac{\partial f(x)}{\partial v} \geq 0 \text{ for all } v \in T_{\mathcal{D}}(x)$$

if $\nabla f$ is Lipschitz continuous.
Extra Directions
Missing Directions and the Stencil Gradient

Not all points in $S$ need be in $D$.
Define the stencil gradient $\nabla f(x, V, h)$ as the solution of

$$
\min_{y \in \mathbb{R}^N} \| hV^T y - \delta(f, x, V, h) \|
$$

where $V$ is the matrix of directions and

$$
\delta(f, x, V, h) = \begin{pmatrix}
  f(x + h v_1) - f(x) \\
  f(x + h v_2) - f(x) \\
  \vdots \\
  f(x + h v_K) - f(x)
\end{pmatrix}.
$$

We use $\nabla f(x, V, h)$ in the quasi-Newton method.
So what’s $V$?
Here are the rules

- The call to $f$ must work, so

$$x + hv_j \in \mathcal{D}$$

- If $x$ is the only point in $\mathcal{D}$, shrink.
- You have to have enough directions to avoid missing $\mathcal{D}$.

So, your direction set has to be “rich” and must vary with the iteration.
\( \mathcal{V} = \{ V_n \} \) is rich if

- for any unit vector \( v \) and
- any subsequence \( \mathcal{W} = \{ W_{n_j} \} \) of \( \mathcal{V} \)

\[
\liminf_{j \to \infty} \min_{w \in W_{n_j}} \| w - v \| = 0.
\]

Example: add one or more random directions to the coordinate directions.
Convergence for Implicit Filtering

If

- $\nabla f$ Lipschitz
- Search and simplex gradient use $V_n$ at iteration $n$
- $\mathcal{D}$ is regular
- Stencil fails infinitely often

then any limit point of the implicit filtering iteration satisfies the necessary conditions.
Suppose we can’t evaluate $f$, but instead evaluate

$$\tilde{f}(x, N_{MC})$$

where $N_{MC}$ is the number of “trials”.

We assume that the errors are like Monte Carlo integration.

Unconstrained stuff: Trosset 00, Anderson-Ferris 01, Zhang-Kim 03, Deng-Ferris 07
Just like MC high-dimensional integration

There is \( c_F : (0, \infty) \rightarrow (0, \infty) \) such that
For all \( \delta > 0 \), and \( x \in \mathcal{D} \)

\[
Prob \left( |f(x) - \tilde{f}(x, N_{MC})| > \frac{c_F(\delta)}{\sqrt{N_{MC}}} \right) < \delta
\]

and

\[
Prob \left( \tilde{f}(x, N_{MC}) = NaN \right) \leq \frac{c_F(\delta)}{\sqrt{N_{MC}}}.
\]
Algorithm and Theory

If $x \notin D$, 

$$\text{Prob} \left( \tilde{f}(x, N_{MC}) = NaN \right) \leq \frac{c_F(\delta)}{\sqrt{N_{MC}}}.$$ 

The algorithm uses $\tilde{f}$ and increases $N_{MC}$ as $h$ decreases.

$$\lim_{n \to \infty} \left( h_n \sqrt{N_{MC}^n} \right)^{-1} = 0.$$ 

Do this and the theory still holds with probability one.
Example: Water Resource Policy
Dillard, Characklis, Kirsch, Ramsey, K: 06-11

![Graphs showing different NMC values]
Properties of the Example

- six variables
- two linear constraints
- two real hidden constraints

Does the theory reflect the practice?
Implicit Filtering

Example

Software: imfil.m, K-11

- MATLAB implicit filtering software
- Handles linear constraints via tangent directions
- Rich stencils by adding random directions
- \( f \) can be scale aware and change \( N_{MC} \) as \( h \) varies
- Code for this example LRGV*
Do Random Directions Help?

Add $k$ random directions with $N_{MC} = 500$. 
Scale Aware Computation; $N_{MC} = 100, \ldots, 4.9M$

12 runs; 24 random directions; 1891 calls to $f$; over 1000 failures
Conclusions

- Sampling methods for black-box functions
- Hidden constraints and random noise
- Asymptotic convergence theory
- Examples