

Contents

Preface	xii
How to Get the Software	xiii
I Preliminaries	1
1 Introduction	3
1.1 What Is the Problem?	3
1.2 Sampling Methods	4
1.3 When to Use Sampling Methods	4
1.4 When to Avoid Sampling Methods	6
2 Getting Started with <code>imfil.m</code>	9
2.1 Computing Environment and Installation	9
2.2 What <code>imfil.m</code> does	9
2.2.1 Constraints	10
2.2.2 The Budget for the Iteration	11
2.2.3 The Objective Function	11
2.3 Basic Usage	12
2.3.1 Termination	12
2.4 A Very Simple Example	13
2.5 Setting Options	15
2.5.1 Nonlinear Least Squares	16
2.5.2 Parallel Computing	17
2.5.3 Scaling f	19
2.5.4 Changing the Scales	20
2.5.5 Looking at the Iteration History	20
2.5.6 Scale-Aware Functions	20
2.6 Passing Data to the Function	21
3 Notation and Preliminaries	23
3.1 Numerical Linear Algebra	23

3.1.1	The LU Factorization and Gaussian Elimination . . .	24
3.1.2	The Cholesky Factorization	24
3.1.3	Linear Least Squares	24
3.2	Preliminaries from Calculus	26
3.2.1	The Fundamental Theorem of Calculus and Taylor's Theorem	27
3.3	Unconstrained and Bound Constrained Optimization	28
3.4	Necessary and Sufficient Conditions for Optimality	29
3.4.1	Unconstrained Problems	29
3.4.2	Bound Constrained Problems	29
3.5	Steepest Descent and Gradient Projection Algorithms	31
3.5.1	The Method of Steepest Descent	31
3.5.2	The Gradient Projection Method	34
3.6	Fast Local Convergence	38
3.7	Finite Difference Approximations	39
3.8	Quasi-Newton Methods	43
3.8.1	Unconstrained Problems	44
3.8.2	Projected Quasi-Newton Methods	45
3.9	Nonlinear Least Squares	47
3.9.1	Gauss–Newton Iteration	47
3.9.2	Projected Gauss–Newton Iteration	49
3.9.3	Levenberg–Marquardt Iteration	50
3.9.4	Projected Levenberg–Marquardt Iteration	50
II Algorithms and Theory		53
4	The Implicit Filtering Algorithm	55
4.1	Coordinate Search	55
4.2	Primitive Implicit Filtering	59
4.2.1	The Stencil Gradient	60
4.2.2	Implementation of Primitive Implicit Filtering . . .	61
4.3	Model Hessians and Bounds	63
4.3.1	Quasi-Newton and Gauss–Newton model Hessians .	63
4.3.2	Bound Constraints and Scaling	64
4.4	The Implementation in imfil.m	65
4.4.1	The Outer Iteration	65
4.4.2	The Inner Iteration	66
4.4.3	Hidden Constraints	67
4.4.4	Explicit Constraints	68
5	Convergence Theory	71
5.1	Consequences of Stencil Failure	71
5.2	Stencil Failure and Coordinate Search	74
5.3	Positive Spanning Sets and Positive Bases	75
5.4	A Convergence Theorem for Implicit Filtering	76

5.5	Convergence Rates	78
5.5.1	Identification of the Binding Constraints	78
5.5.2	Gauss–Newton Iteration	80
5.6	Enriched Stencils	84
III Software Reference		89
6	Using <code>imfil.m</code>	91
6.1	Installation and Testing	91
6.2	Input	91
6.2.1	The Initial Iterate	92
6.2.2	The Input Function <code>f</code>	92
6.2.3	The Budget	93
6.2.4	The Bounds	93
6.3	Output and Troubleshooting	94
6.3.1	The <code>histout</code> array	94
6.3.2	The <code>complete_history</code> Structure	94
6.3.3	Slow Convergence or No Convergence	95
6.4	Setting Options	95
6.5	The Inner Iteration	96
6.5.1	Scaling f with <code>fscale</code>	96
6.5.2	Quasi-Newton Methods for General Problems	97
6.5.3	Nonlinear Least Squares	97
6.5.4	Which Best Point to Take?	97
6.5.5	Limiting the Quasi-Newton Step	98
6.6	Managing and Using the Scales	98
6.6.1	<code>Scalestart</code> and <code>Scaleddepth</code>	98
6.6.2	<code>custom_scales</code>	99
6.6.3	Scale-Aware Functions	99
6.6.4	Smooth Problems	99
6.7	Parallel Computing	100
6.8	Passing Data to f	100
6.9	Stencils	101
6.9.1	<code>vstencil</code>	101
6.9.2	<code>random_stencil</code>	101
6.10	Terminating the Outer Iteration	101
6.10.1	<code>target</code>	102
6.10.2	<code>stencil_delta</code>	102
6.10.3	<code>function_delta</code>	102
6.10.4	<code>maxfail</code>	103
6.11	Terminating the Inner Iteration	103
6.11.1	<code>maxit</code>	103
6.11.2	<code>maxitarm</code>	103
6.11.3	Noise-Aware Functions and the <code>svarmin</code> Option	103

6.11.4	Terminating the Quasi-Newton Iteration with <code>termtol</code>	104
6.12	<code>verbose</code>	104
7	Advanced Options	105
7.1	Adding New Directions to the Stencil	105
7.2	The <code>iteration_data</code> Structure	108
7.2.1	Internal Scaling and <code>f_internal</code>	108
7.3	Updating the <code>complete_history</code> Structure	110
7.4	Testing More Points with the <code>explore_function</code> Option	111
7.4.1	Random Search Example	112
7.5	The Executive Function	116
7.5.1	Input to the Executive Function	117
7.5.2	Output from the Executive Function	117
7.5.3	Levenberg–Marquardt Example	118
IV	Case Studies	119
8	Harmonic Oscillator	121
8.1	Problem Formulation	121
8.1.1	Calling <code>imfil.m</code> and Looking at Results	124
8.2	Parallelism	126
8.2.1	Parallelizing the Serial Code	126
8.2.2	Looking at the Parallel Results	127
8.3	Using the <code>scale_aware</code> Option	129
8.4	Termination Revisited	130
8.4.1	Using <code>function_delta</code> to Terminate the Iteration	131
8.4.2	Using the Executive Function	132
9	Hydraulic Capture Problem	133
9.1	Problem Formulation	134
9.1.1	Saturated Flow Equations	134
9.1.2	Objective Function	135
9.1.3	Constraints	136
9.1.4	Design Variables and Initial Iterate	138
9.2	MATLAB Codes and MODFLOW	139
9.2.1	Working with MODFLOW	139
9.3	Results	141
10	Water Resources Policy	145
10.1	Problem Formulation	145
10.1.1	Objective Function	146
10.1.2	Hydrologic Constraints	146
10.1.3	Purchasing Decisions	147
10.1.4	Evaluating the Objective	149

10.1.5	The Reliability and CVaR Constraints	150
10.2	Objective Function Code	152
10.3	Results	153
10.3.1	Hiding the Linear Constraints	153
10.3.2	Using the <code>add_new_directions</code> Option	155
Bibliography		157
Index		167