Optimization Method Coding Project 3: Non-linear Least Square Problems

Li, Ziyao
1500017776 / Yuanpei College
leeeezy@pku.edu.cn

Abstract

This work focuses on methods solving non-linear least square problems. Six test functions are evaluated in this work, namely Meyer Function, Osborne Exponential I Function, Osborne Exponential II Function, Jennrich and Sampson Funtion, Freudenstein and Roth Function and Davidon Function. Numerical Experiments are conducted on three formerly mentioned problems. Six methods are examined with these problems. Generally Gauss-Newton (GN) method have higher convergence speed, while the exact line search used in it adds greatly to its function-evaluation times. Levenberg-Marquardt-Fletcher (LMF) method and Levenberg-Marquardt-Nielson (LMN) method show similar performances. Dogleg and Twofold Dogleg methods do not perform well on the problem. Gauss-Newton method with the BFGS adjustment for Large-Residual problem is the most robust method with more iterations to converge in different problems.

1 Least Square Problem

Least Square Problems are defined as

$$\min_x f(x) = \sum_{i=1}^{m} r_i(x).$$

This kind of problems are special because the gradients of the problem can be transformed to

$$g(x) = J'(x)r(x),$$

and approximations over the Hesse matrices of original problems can be made by calculating the Jacobian matrices, and Newton directions can be estimated over the Gauss-Newton equation

$$J_k'J_kd_k = -J_k'r_k.$$

Levenberg-Marquardt methods adjust the matrix $J_k'J_k$ into $J_k'J_k + \nu_k I_n$ to avoid problems caused by the singularity of the original matrix; Dogleg methods adopt the idea of Trust Region Algorithms when deriving the descend directions and step-lengths by combining the Gauss-Newton direction and negative gradient direction; for problems with large residuals, ignoring the matrix $S(x)$ can cause problems, and BFGS methods can be used to calculate the $B(x)$ matrices to approximate $S(x)$, leading to robustness and faster convergence.

2 Numerical Performance

Performance of six different methods are tested. The general convergence criterion, without specific notifications, are defined as

$$\begin{aligned} |f_k - f_{k-1}| &\leq \epsilon \\ ||g_k||_\infty &\leq \epsilon \\ \epsilon &= 10^{-8}. \end{aligned}$$

However, due to the property of some of the problems, g_k hardly converges to 10^{-8} no matter what method is used. Therefore, an alternative criterion without constraints over g_k is used sometimes and the factor indicating performance, $\|g_k\|_\infty/\|g_0\|_\infty$, is monitored when such criterion is used. Relevant results are shown in Table 1.

Due to the calculation of exponential, sine and cosine functions, the computational accuracy in this projects are relatively low, causing much trouble when line search methods are adopted. Figure 1 shows the function values versus step lengths of the Meyer Function, from which we can see

	GN			LMF			LMN		
Problem	iter	feva	g	iter	feva	g	iter	feva	g
Meyer	*8	350	10	*160	321	13	*495	991	10
Osborne I	10	442	-	27	55	-	22	45	-
Osborne II	13	570	-	19	39	-	14	29	-
Jennrich and Sampson	/	/	/	91	183	-	159	319	-
Freudenstein and Roth	41	1,892	-	51	103	-	*32	65	6
Davidon	*625	29,375	7	*75	151	9	*77	155	8
	Dogleg			**Twofold-Dogleg**			**GN+BFGS**		
Problem	iter	feva	g	iter	feva	g	iter	feva	g
Meyer	*95	154	11	*86	150	14	*110	7,603	9
Osborne I	9	17	-	18	32	-	22	1,243	-
Osborne II	13	26	-	18	33	-	27	1,493	-
Jennrich and Sampson	*27	40	8	*38	58	7	8	656	-
Freudenstein and Roth	41	66	-	*31	49	7	7	306	-
Davidon	*255	455	7	*100	176	7	13	991	-

/: Algorithm cannot continue.

$*$: Criterion without constraining g adopted.

g: $-lg(\frac{\|g_k\|_\infty}{\|g_0\|_\infty})$, calculated if * is adopted.

Table 1: Numerical performance of six methods over six problems. The functions can be found in *Testing Unconstrained Optimization Software*, or relevant references attached to this report.

obvious fluctuations caused by machine accuracy. Therefore, if valid step lengths cannot be derived, we set $\alpha = 1$ instead of speculating that the direction is non-descending and exiting, as we may do in the Quasi-Newton Methods.

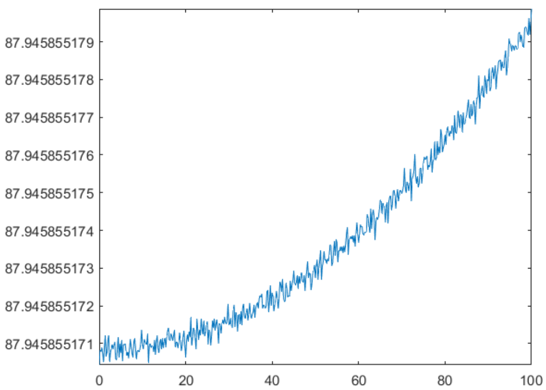

Figure 1: A Figure of function value versus step length during some iteration of the Meyer Function. The function suffers from giant fluctuations caused by limited machine accuracy, which is mainly caused by the exponential operation in the Meyer Function.

The gradient of Meyer function never reaches a level below 10^{-5}. This is mainly caused by the exponential part of the function, evaluation errors and small denominators (x_3). GN method behaves surprisingly good over this problem, while it cannot continue at iteration 9. Other methods all have poor performance over this problem.

The Osborne Exponential problems have rather good numerical properties. Although the dimensionality of the problem is the most during all six problems, they still remain the two easiest problems. All methods show good performances over these problems, among which the Dogleg method's is the best.

Surprisingly, the methods targeting at small residual problems are able to solve large residual problems as well, while the gradient usually do not converge in reasonable iterations, and the performances are poor. GN+BFGS performs extremely good over these three problems, while the function-evaluation times are massive due to the exact line search process.

3 `lsqnonlin()` Performance

`lsqnonlin()` is an inner function defined in Matlab. We examined the performance of this function over these six problems, finding that processes terminate with different reasons. For Meyer, Osborne Exponential II and Freudenstein

& Roth functions, the iterations stop because function values stop changing (g not converged). For Osborne Exponential II function, both function value and g converge. For Jennrich & Sampson function, the step length is so small to continue. For Davidon function, the iteration maximun (400) reached. Besides, for Freudenstein & Roth function, the local minimum is obtained.