International Symposium on Nonlinear Theory and its Applications
Xi'an, PRC, October 7–11, 2002

A Simple Acceleration Technique of Levenberg-Marquartd Method

Hiroyasu Sakamoto *, Azusa Kuwahara * and Yoshiteru Hayami +

* Kyushu Inst. Design
Shiobaru, Minamiku, Fukuoka, Japan
Tel & Fax : +81-92-553-4506
Email : sakamoto@kyushu-id.ac.jp

+ Chikusi Jogakuen Jun.Coll.
Ishizaka, Dazaifu, Fukuoka, Japan
Tel : +81-92-925-3511, Fax : +81-92-924-4369
Email : hayami@chikushi-u.ac.jp

1. Introductin

Levenberg-Marquardt (LM) method is widely known as one of standard tools for nonlinear minimum mean square optimization problems [1],[2]. LM method has a parameter that smoothly connects between Newton-Gauss method and steepest descendent method. Because of its high converging speed and numerical stability, LM method has been introduced as an optimization part into many world wide famous software packages such as Matlab[3], gnuplot, etc. LMM method (Morrison is added to LM) is a more stabilized version. Here, we may refer to the both as LM method.

The method has solid theoretical justification[4] and users of the method (especially software package users) are implicitly supposed to take care of conditions that all the variables are almost equally scaled and has no correlated influence to cost function. However, in engineering applications the conditions are often laborious to maintain or just neglected and consequently slow and inaccurate solutions can result. The authors think that the responsibility for maintaining the conditions should be on the software side.

This paper discusses an ill-posed problem that often occurs when we apply LM and LMM methods to actual engineering applications, and proposes a simple technique which can accelerate the speed of convergence and stabilize LM iteration process. Our first engineering application of LMM method in this paper is a measurement system of interfacial tension between two fluids by MMSE curve fitting of Laplace's differential equations to drop shape of the fluid[5]. Numerical examples show that the proposed method is very effective.

2. LM method and LMM method

In this section we briefly summarize LM and LMM methods. Real functions $f_i(\boldsymbol{x})$, (i = 1, 2,…, l) of k variables vector $\boldsymbol{x}$ form a cost function $F(\boldsymbol{x})$ to be minimized.

$$F(\boldsymbol{x}) = \sum_{i=1}^{l} f_i^2(\boldsymbol{x}) \,, \quad (k \le l) . \tag{1}$$

Jacobian matrix of size $l \times k$ is denoted by $\boldsymbol{J}$, and a column vector of $f_i(\boldsymbol{x})$ by $\boldsymbol{f}(\boldsymbol{x})$.

Starting with an initial parameter $\boldsymbol{x}_0$, the n-th iteration of LM method solves the following equation to obtain a correction vector $\Delta\boldsymbol{x}_n$ and a new parameter vector $\boldsymbol{x}_{n+1} = \boldsymbol{x}_n + \Delta\boldsymbol{x}_n$.

$$\left[\boldsymbol{J}_n^T \boldsymbol{J}_n + \lambda_n^2 \boldsymbol{I} \right] \Delta\boldsymbol{x}_n = -\boldsymbol{J}_n^T \boldsymbol{f}(\boldsymbol{x}_n) . \tag{2}$$

In Eq.(2), $\boldsymbol{I}$ is identity matrix and λ_n (≥ 0) is Marquardt factor that is controlled at each iteration step and acts as accelerator and stabilizer of the iteration process.

In case of $\lambda_n = 0$, Eq. (2) is equivalent to Newton-Gauss method, and in the extreme case of $\lambda_n \to \infty$, direction of the solution $\Delta\boldsymbol{x}_n$ indicates the steepest descendent direction of the $F(\boldsymbol{x}_n)$.

In a standard control scheme for λ_n, it is increased or decreased in an iteration step n when $F(\boldsymbol{x}_n)$ increases or decreases, respectively. The initial value λ_0 is usually set to $\| \boldsymbol{J}_0 \|$.

LMM method solves the equation,

$$\begin{bmatrix} \boldsymbol{J}_n \\ \cdots\cdots \\ \lambda_n \boldsymbol{I} \end{bmatrix} \Delta\boldsymbol{x}_n = -\begin{bmatrix} \boldsymbol{f}(\boldsymbol{x}_n) \\ \cdots \\ \boldsymbol{0} \end{bmatrix}, \tag{3}$$

which is equivalent to Eq.(2). It is clear that the factors λ_n will constrain the solution $\Delta\boldsymbol{x}_n$ to be closer to zero. A Householder QR transformation for the left-hand-side matrix is employed to obtain stabler solution of $\Delta\boldsymbol{x}_n$.

In some implementations[3], LM methods are combined with simple line search algorithms for the minimum cost function along the direction indicated by $\Delta\boldsymbol{x}_n$. However, up to the authors' experiences, as the magnitude of Marquartd factor can also effectively control step size of the correction vector $\Delta\boldsymbol{x}_n$, additional scheme for controlling step size is not necessarily required.

3. The proposed method

When we apply LM method or LMM method to a real engineering problem, it often happens that each column of the matrix $\boldsymbol{J}$ has different physical units and shows quite different numerical magnitudes. This does because for each (i,j)-th element $\partial f_i(\boldsymbol{x})/\partial x_j$ of $\boldsymbol{J}$, all $f_i(\boldsymbol{x})$ have usually (not necessarily) the same unit in order to form an objective function of Eq.(1) and each x_j can have different units. Because the differences in numerical magnitudes can drive $\boldsymbol{J}$ to an ill-posed problem, it is preferable to suppress the differences by properly selecting the unit employed. However even after careful selection, good accordance of numerical magnitude is sometimes difficult to achieve. Rather, some users of software packages do not regard on the magnitude differences.

When $\boldsymbol{J}$ contains significant differences of column magnitude, the same value of λ_n for all the columns of $\boldsymbol{J}$ of Eqs. (2),(3) would slow down the convergence speed of the parameter $\boldsymbol{x}$. In fact, for the smallest column the constraint is too heavy to update corresponding variable. It is therefore reasonable to modify Eqs. (2) and (3) as follows in order to give different constraint to each column.

$$\left[\boldsymbol{J}_n^T\boldsymbol{J}_n+\mathrm{diag}(\lambda_{n,1}^2,\cdots,\lambda_{n,k}^2)\right]\Delta\boldsymbol{x}_n=-\boldsymbol{J}_n^T\boldsymbol{f}(\boldsymbol{x}_n)\,, \quad (4)$$

$$\begin{bmatrix}\boldsymbol{J}_n\\ \cdots\cdots\cdots\\ \mathrm{diag}(\lambda_{n,1},\cdots,\lambda_{n,k})\end{bmatrix}\Delta\boldsymbol{x}_n=-\begin{bmatrix}\boldsymbol{f}(\boldsymbol{x}_n)\\ \cdots\\ \boldsymbol{0}\end{bmatrix}. \quad (5)$$

Initial value of our new Marquardt factor $\lambda_{0,j}$ is obtained from the norm of the j-th column of the initial Jacobian matrix $\boldsymbol{J}_0$. At every iteration step n, values of $\lambda_{n,j}$ are adaptively updated in the same manner as traditional LM methods (see section 2.).

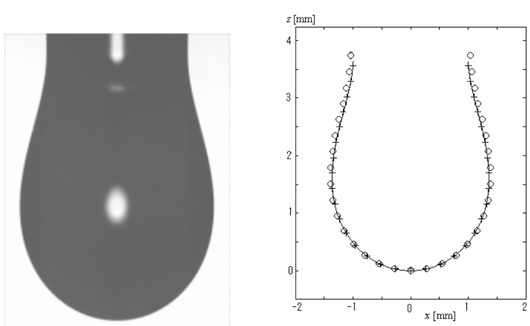

Fig.1. Left : Real image of a drop. Right : Initial synthesized curve with very carefully estimated parameters(O), final converged curve(+) of Laplace equation and observed edge (—). Dimension of one pixel is about 7.8[μm] square.

4. Numerical examples

4.1 Curve fitting in interfacial tension measurement

The first numerical example is obtained from a development of a measurement system for interfacial tension by processing images of pendant drops[5]. Fig.1 shows a few images from the system. Cost function $F(\boldsymbol{x})$ is RMSE of curve fitting with the parameter vector $\boldsymbol{x}$ = [horizontal position, vertical position, curvature at the drop's peak, fluids' constant which includes interfacial tension]. By using MKS unit system, four columns of $\boldsymbol{J}$ and $\boldsymbol{x}$ have units [1, 1, m^2, m^3] and [m, m, m^{-1}, m^{-2}], respectively. The dimensions of the drops (a few millimeters) and that of $\boldsymbol{f}(\boldsymbol{x})$ (less than 10^{-5} [m] initially and goes down to 10^{-7} [m] in subsequent image frames) introduce the maximum norm ratio of more than 10^{+7}. The 4-th column of $\boldsymbol{J}$ has the minimum norm.

Figure 2 (a) shows an error converging process of conventional LMM method (×) and the proposed method (O). Until all the elements of the parameter $\boldsymbol{x}$ converges to relative accuracy of more than 10^{-5}, conventional method requires 27 iterations and the proposed method does only 3 iterations. In this figure, we notice that the

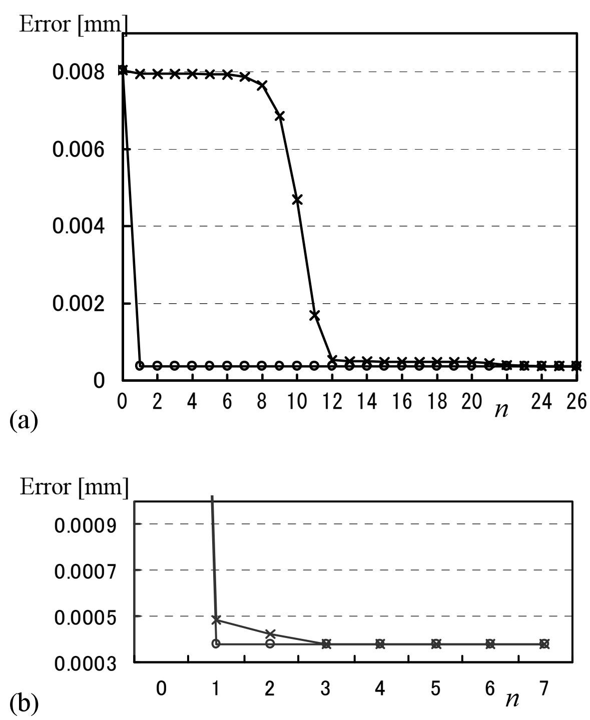

Fig.2. Convergence of RMS fitting error of conventional LMM method (×) and the proposed LMM method (o).
(a): Unit of the length is [m] (the maximum column norm ratio ≥ 10^{+7}), (b) : Unit is [cm] (the ratio is about 10^{+2}).

curve of the conventional method can be divided into 4 parts of convergence, i.e. $1 \le n \le 7$, $8 \le n \le 12$, $13 \le n \le 22$, and final. We will see later that these parts are strongly related to the norm difference of column vectors of $\boldsymbol{J}$.

In order to minimize the norm ratio, we had changed the unit of the length into [cm] and reduced the norm ratio to about 10^2. Fig.2(b) shows that the conventional method requires 7 iterations and our method does 3 iterations for the convergence with accuracy of 10^{-5} [m].

Figure 3(a) depicts parameters correction process of conventional LMM method that was shown in the curve ($\times$) of Fig.2(a). As the 4-th column of $\boldsymbol{J}$ has the minimum norm, the corresponding parameter x_4 is most strongly constrained to zero, and therefore it starts to effectively be corrected at the latest ($n \approx 13$) of all the parameters. For the iteration of $n < 13$, corrections to x_1, x_2 and x_3 are not either effective because they will be corrected later once again, but just preparing for the correction of x_4. Final convergence is attained at the 27-th iteration when corrections of all parameters become less than 10^{-5}.

The parameter correction in Fig.3(a) can exactly explain the four steps of convergence of the curve ($\times$) of Fig.2(a). The first step ($1 \le n \le 7$) is due to the correction for x_1 and x_2. In the 2-nd step ($8 \le n \le 12$) x_3 joins in the process, and in the 3-rd step ($13 \le n \le 22$) x_3 joins in, and in the 4-th step ($23 \le n \le 27$) is for final precise adjustment.

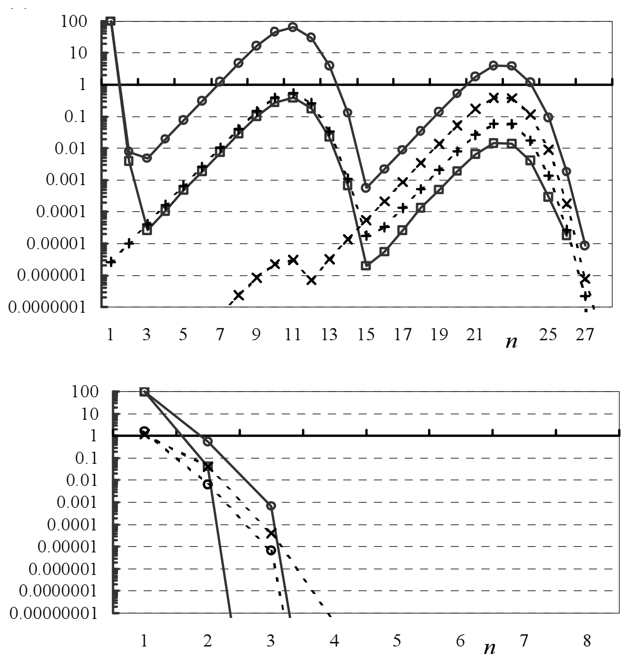

Fig.3. Amount of correction (% of converged quantity) for 4 elements of the vector $\varDelta \boldsymbol{x}_n$ vs. iteration count n. The marks are (□):horizontal position, (O): vertical position, (+): curvature, ($\times$): fluids' constant, arranged in descending order of column norms of $\boldsymbol{J}$. Fig.(a) shows conventional method, Fig.(b) the proposed method.

On the contrary, in the proposed method (the curves marked O in Fig.2(a),(b)), all parameters can be corrected from the first iteration as shown in Fig.3(b). Very quick convergence is therefore possible.

The authors have build the proposed LMM method into the measurement system, and achieved sub-micron curve fitting and reported that the accuracy of 1% confidential interval is ±0.015 [milli-N/m] and its real-time measuring rate is 3.3[1/sec]. However, due to the space limitation, little about the acceleration method was explained in the reference [5].

4.2 Extended Rosenbrock's functions

In order to perform a systematic and visible experiment, we have employed Rosenbrock's function [3] in an extended form,

$$R_m(x, y) = 100\,(my - x^2)^2 + (1 - x)^2 , \tag{6}$$

where m (= 1 for the original form) is a positive parameter that controls column norm of $\boldsymbol{J}$. The functions have a deep valley along a quadratic curve of the first term with the minimum value 0 at $(x, y) = (1, 1/\sqrt{m})$. The gradient along the valley has relatively small value.

Decomposing Eq. (6) into two, $f_1 = 10\,(my - x^2)$ and $f_2 = 1 - x$, $\boldsymbol{J}$ is given by,

$$\boldsymbol{J} = \begin{bmatrix} -20x & 10m \\ -1 & 0 \end{bmatrix}. \tag{7}$$

Figure 4 shows averaged LMM iteration counts for convergence of $\| \varDelta \boldsymbol{x}_n / \boldsymbol{x}_n \| < 10^{-15}$ when started from 1,000 random initial parameters (x_0, y_0) which are uniformly distributed in the range of ([-2, 2], [-2/m, 4/m]). See Fig.5.

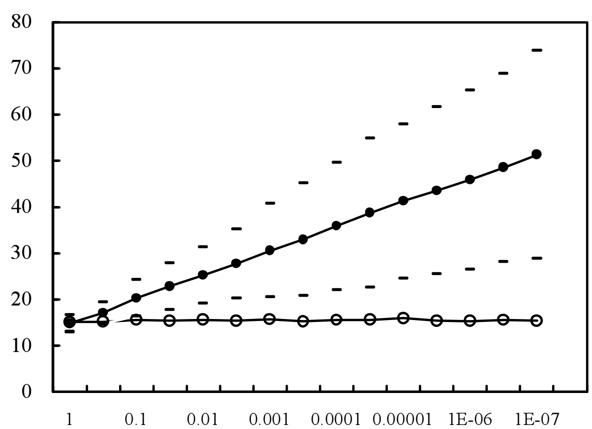

Fig.4. Averaged iteration counts vs. m. The conventional LMM method (•, the ranges of standard deviation are indicated by upper and lower bars -) and the proposed LMM method (O, all standard deviations are nearly 2).

It is clearly observed form the figure that the conventional LMM method requires larger iterations and has larger standard deviation for smaller values of m and larger difference of column norm of $\boldsymbol{J}$. The proposed LMM method keeps almost constant average counts and standard deviations (between 15.18~15.87 and 1.98 ~ 2.11, respectively) for a very wide range of m.

In more than 30,000 calculations of Fig.4, no failure of convergence is found without employing any line search algorithm [3].

When we switch from LMM method to LM method, almost the same curve as Fig.4 is obtained for both conventional and proposed LM methods.

As typical examples of convergence processes of the conventional and the proposed LMM methods, Fig.5 shows two loci for the case of $m = 10^{-4}$. The figure includes (x, y) area of $([-2, 2], [-2\times10^4, 4\times10^4])$. From Fig.4, averaged iterations for the conventional and the proposed methods are 35.85 and 15.52, respectively.

The marks + with dotted curve indicates the locus of the conventional method. It takes 48 iterations because the constraint to the variable y is so heavier (about 10^4 times than x) that had not been corrected until Marquardt factor became recursively smaller. The effective correction to y starts at about 28 iterations, at this time the initial value $y_0 = -3637$ is corrected only 2%.

The locus of the proposed method (× marks with real curve) moves quickly from the very first iteration step and after 14 iterations it converges to the solution $(1, 1\times10^4)$.

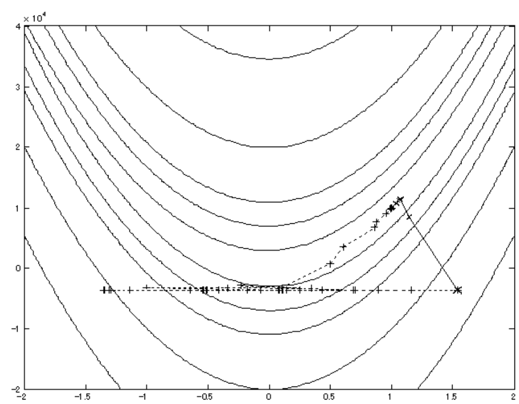

Fig.5. Examples of the loci of the conventional LMM method (+ -- +) and the proposed LMM method (× and real curve) for an extended Rosenbrock's function ($m = 10^{-4}$). The former took 48 iterations and the latter 14. The minimum value of the Rosenbrock's function is 0 at $(1, 1\times10^{-4})$.

5. Conclusions

A simple numerical technique for accelerating and stabilizing Levenberg-Marquardt(-Morrison) methods is proposed and examined. By compensating for the column norm of Jacobian matrix, the proposed method can significantly reduce computational requirements for a wide range of ill-posed problem such as Rosenbrock's function.

Although the theoretical principle is already known, many applications do not seem to properly use the LM and LMM methods. From the authors' opinion, a countermeasure should be supplied in the software side.

The authors are trying to compensate for the correlated influence of parameters to the cost functions by introducing non-diagonal constraint matrix in order to improove the LM method and the LMM method.

Acknowledgements

The research is supported by Grant-in-Aid for Scientific Research (C) from Japan Society for the Promotion of Science. The research and development of interfacial tension measurement system is also supported by Regional Science Promotion Program from Fukuoka Industry, Science and Technology Foundation.

References

[1] W.H.Press, et.al. : *Numerical Recipes in C*, Cambridge Univ. Press, 1988.

[2] Fujitsu Inc. : *SSL II users manual* (in Japanease), 1986.

[3] Mathworks Inc.:*Optimization Toolbox User's Guide*,2000.

[4] J.J.Morè : "The Levenberg-Marquardt Algorithm : Implementation and Theory," in *Lecture Notes in Mathematics 630 Numerical Analysis*, Springer-Verlag, 1978.

[5] H.Sakamoto, et.al. : "Real-time Measurement of Interfacial Tension by Image Processing," to be published in J. ITE Japan, vol.56, no.10, 2002.