Computational Research Progress in Applied Science & Engineering

ISSN 2423-4591

CRPASE Vol. 06(01), 19-24, March 2020

On Ill-conditioned Linear System Due to the Levenberg-Marquardt Algorithm in Solving Nonlinear Equations

Seyyed Shahabeddin Azimi*, Mansour Kalbasi

Department of Chemical Engineering, Amirkabir University of Technology, Hafez Avenue, Tehran, Iran

Keywords

Damping Factor,
Ill-conditioning,
Levenberg-Marquardt Algorithm,
Linear System,
Nonlinear Equations.

Abstract

In this paper, as an important problem, the nonlinear equations resulting from the finite volume approximation of the differential equations describing the nanofluid convective heat transfer in a tube are considered to obtain the temperature distribution. One of the numerical algorithms for solving nonlinear equations is the Levenberg-Marquardt in which a system of linear equations must be solved per iteration. In order to prevent the linear system from being ill-conditioned, the damping factor (λ) of this algorithm should not be decreased unconditionally. Indeed, it was proposed that a minimum value for λ is set per iteration for obtaining the search direction, where the time complexity per iteration is O(n3) in which n is the number of unknowns (or equations) of the nonlinear system. However, in this case study, this linear system is ill-conditioned. It is shown that the ill conditioning of the linear system due to this case study can be prevented in all iterations by setting an approximate minimum value for λ in the first iteration. This can considerably reduce the computational cost, in other words, this paper introduces an algorithm with a low computational complexity.

1. Introduction

Nanofluids are liquids in which nanometer-sized solid particles, known as nanoparticles, are dispersed. Nanofluids improve the thermal properties of host fluids [1]. Therefore, there is great industrial interest in nanofluids.

Heat transfer analysis is required to determine the temperature distribution in the domain (see, for example, Sari et al. [2]).Nowadays, more and more researchers choose to investigate the academic research problem by using numerical simulation [3]. To obtain the temperature distribution, the mathematical model may be solved by numerical methods.

Numerical analysis of nanofluids heat transfer is of great interest to engineers. In this study, enhancing the laminar forced convective heat transfer of nanofluid over horizontal tube is considered where the mathematical model is solved by the Levenberg-Marquardt method in which a system of linear equations must be solved per iteration. However, in this case study, it is ill-conditioned.

Let Ax=b be a system of linear equations, where A is the coefficient matrix, x is the vector containing the unknowns and b is the right-hand-side vector. A linear system which its solution is very sensitive to perturbations in its matrix A and vector b (input data) is called ill-conditioned, for example, two linear systems are considered as Eqs. (1.a) and (1.b) [4]

$$A_1x=b_1 \rightarrow \begin{bmatrix} 8 & -5 \\ 4 & 10 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 3 \\ 14 \end{bmatrix} \quad (1.a)$$

$$A_2\bar{x}=b_2 \rightarrow \begin{bmatrix} 0.66 & 3.34 \\ 1.99 & 10.01 \end{bmatrix} \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} = \begin{bmatrix} 4 \\ 12 \end{bmatrix} \quad (1.b)$$

where the solution of both systems is $(1,1)^T$. Now, if $b_1$is changed to $b_1 - (0.04, 0.06)^T$, the new solution to the first system is $(0.993, 0.997)^T$, however, if b_2 is changed by the same vector, the new solution to the second system is $(6, 0)^T$. Therefore, it is clear that the second system is very sensitive to perturbations in its input data.

The sensitivity of the linear system to perturbations (noises) in its input data is measured by the condition number (the larger the condition number is, the more ill-conditioned the system is), which will be defined in the next section. Detailed information for solving ill-conditioned linear systems can be found in Tikhonov and Arsenin [5] and the references therein, also see [6-13].

Nonlinear equations arise in many physical problems where finding their roots, or zeros, is a common problem [14].

Consider a system of nonlinear equations as Eq. (2.a)

* Corresponding Author:
E-mail address: s.azimi@aut.ac.ir – Tel, (+98) 2164543178 – Fax, (+98) 2166405847

Received: 04 January 2020; Accepted: 16 February 2020

$$r(x) = 0 \qquad (2.a)$$

where $r: R^n \rightarrow R^n$is a vector function [15]:

$$r(x) = \begin{bmatrix} r_1(x) \\ \vdots \\ r_n(x) \end{bmatrix}, \quad x = \begin{bmatrix} x_1 \\ \vdots \\ x_n \end{bmatrix} \qquad (2.b)$$

where each $r_i: R^n \rightarrow R^n$, i=1, 2, …,n is smooth. To solve nonlinear system (2.a) by iterative algorithms, the Levenberg-Marquardt algorithm can be used [16,17] in which the search direction S (in the line search strategy) is obtained by solving the linear system as Eq. (3)

$$(J^TJ + \lambda D)S = -J^Tr, \quad J = \begin{bmatrix} \frac{\partial r_1}{\partial x_1} & \cdots & \frac{\partial r_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial r_n}{\partial x_1} & \cdots & \frac{\partial r_n}{\partial x_n} \end{bmatrix} \qquad (3)$$

where D is a diagonal matrix with positive elements which repairs the poor scaling [18], J is the Jacobian matrix of r, J^T is the transpose of J, λ is a positive damping factor which is updated from iteration to iteration. When λ takes on a very small value, S approaches to the Gauss–Newton direction and when λ takes on a very large value, S approaches to the steepest descent direction:

$$(J^TJ)S = -J^Tr \xleftarrow{\lambda \rightarrow 0} (J^TJ + \lambda D)S = -J^Tr \xrightarrow{\lambda \rightarrow \infty} S = -\nabla f \qquad (4)$$

Each iteration of a line search method computes a search direction S and then decides how far to move along that direction [15]:

$$x_{k+1} = x_k + \alpha S \qquad (5)$$

where α>0 is the step length.

When the linear system Ax=b is solved on a computer, due to computations performed in finite-precision arithmetic (the computer version of real arithmetic), the solution obtained is the exact solution to the perturbed system (A+ΔA)x=(b+Δb) [4].

On the other hand, for calculating the elements of the Jacobian matrix, the first partial derivatives of the functions (ri) are needed. In some cases, the functions are too complicated to be calculated in practice, so these derivatives must be approximated by use of the finite difference method. Therefore, when the numerical solution (the Levenberg-Marquardt algorithm) is programmed on a computer, the perturbations (noises) arise from finite-precision arithmetic or approximation of the Jacobian matrix.

In this paper, the set of differential equations (the mathematical model) is approximated by a system of algebraic equations for variables at some set of discrete locations in space and time (discretization) [19]. To solve this nonlinear system by the Levenberg-Marquardt algorithm, the matrix J^TJ and possibly (J^TJ+ λD) are ill-conditioned. It was proposed that a minimum value for λ is set per iteration for obtaining the search direction, where the time complexity per iteration is $O(n^3)$, which is expensive in time. Main objective of this work is to present an efficient computational procedure to compute a suitable search direction at each iteration of the Levenberg-Marquardt algorithm, in other words, this study introduces an algorithm with a low computational complexity.

2.Condition Number

The condition number of the matrix A used to recognize ill-conditioned linear systems is defined asEq. (6)[4]

$$Ax = b \quad \rightarrow \quad \text{cond}(A) = k_p(A) = \|A\|_p\|A^{-1}\|_p \qquad (6)$$

where A^{-1} is the inverse of A, $\|\cdot\|_p$is the matrix norm and subscript p denotes the type of the norm. If the condition number is greater than (machine epsilon)$^{-0.5}$then solution of the linear system may not be trustworthy [20]:

$$\text{cond}(A) < (\text{machine epsilon})^{-0.5} \rightarrow \text{solution is acceptable} \qquad (7)$$

In fact, if the above condition is satisfied, the solution of the linear system is reliable. The machine epsilon is the difference between 1.0 and the next smallest number that can be represented by the machine (computer). The main difficulty that may arises, however, is how to calculate $\|A^{-1}\|$ where it is computationally expensive for large-scale problems, therefore, the condition number is estimated according to the following method given in [4]. The condition number of the matrix A can be estimated by using the QR factorization (also called the QR decomposition in which Q is an orthogonal matrix and R is an upper triangular matrix**)** of the matrix A

$$A = QR \quad \rightarrow \quad \frac{1}{n}k_1(A) \le k_1(R) \le nk_1(A) \ , \quad \|A\|_1 = \max_{1 \le j \le n}\{\|a_j\|_1\} \qquad (8)$$

where n is the dimension of the matrix A and $a_j$denotes the jth column of the matrix A. Also, $\|\cdot\|_1$ as l_1 norm (one-norm) of a matrix is the maximum absolute column sum of the matrix. For $k_1(R)$, l_1 norm of R($\|R\|_1$) is easy to compute and then for estimation of $\|R^{-1}\|_1$the following inequality is used:

$$\frac{\|R^{-1}z\|_1}{\|z\|_1} \le \|R^{-1}\|_1 (z \text{ is nonzero}) \qquad (9)$$

where z is obtained by solving R^Tz=e in which e is a vector of ± 1's with the signs chosen to maximize the magnitude of z_1 ..., z_n in succession, then Ry=z is solved:

$$k_1(R) = \|R\|_1 \frac{\|y\|_1}{\|z\|_1} \ \rightarrow \frac{1}{n}k_1(R) \le k_1(A) \qquad (10)$$

Although $\frac{\|y\|_1}{\|z\|_1}$gives a lower bound on $\|R^{-1}\|_1$, in practice it is quite close.

To obtain QR factorization of the matrix A, the Gram-Schmidt method can be used:

$$A = [a_1 \,|\, a_2 \,|\, \cdots \,|\, a_n] \qquad (11)$$

$$b_1 = a_1 \ , \quad c_1 = \frac{b_1}{\|b_1\|_2} \ , \quad \|b_1\|_2 (l_2 \text{ norm}) = \sqrt{\sum b_i^2} \ ,$$
$$b_2 = a_2 - (a_2.c_1)c_1 \ , \quad c_2 = \frac{b_2}{\|b_2\|_2} \qquad (12)$$

$b_{k+1} = a_{k+1} - (a_{k+1}.c_1)c_1 - \cdots - (a_{k+1}.c_k)c_k \ , \quad c_{k+1} = \frac{b_{k+1}}{\|b_{k+1}\|_2}$(13)

$$A = [a_1 \mid a_2 \mid \cdots \mid a_n] = \overbrace{[c_1 \mid c_2 \mid \cdots \mid c_n]}^{Q} \underbrace{\begin{bmatrix} a_1.c_1 & \cdots & a_n.c_1 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & a_n.c_n \end{bmatrix}}_{R} = QR \qquad (14)$$

where vectors of a_1, a_2, …, a_n are the columns of the matrix A.Therefore, for estimation of the condition number of the matrix A, the computational tasks can be summarized as

a) obtaining the QR factorization of the matrix A (the matrix R):

a.1) calculation of vectors of c_1, c_2, …, c_n:

$n^2(n-1)+ 2n^2$ multiplications (or divisions), $O(n^3)$,

$0.5n(n-1)^2+n(0.5n(n+1)-1)+n(n-1)$ additions (or subtracts**),** $O(n^3)$

a.2) calculation of the matrix R:

$0.5n^2 (n+1)$ multiplications, $O(n^3)$

$0.5n(n-1)(n+1)$ additions, $O(n^3)$

b) calculation of $\|R\|_1$:

n(n-1) additions, $O(n^2)$

n-1 comparisons, O(n)

c) estimation of $\|R^{-1}\|_1$:

c.1) solving $R^Tz=e$:

$0.5n(n-1)+n$ multiplications (or divisions), $O(n^2)$

$0.5(n-1(n-2)+(n-1)$ additions (or subtracts**)**, $O(n^2)$

c.2) solving Ry=z:

$0.5n(n-1)+n$ multiplications (or divisions), $O(n^2)$

$0.5(n-1(n-2)+(n-1)$ additions (or subtracts**)**, $O(n^2)$

c.3) calculation of $\frac{\|y\|_1}{\|z\|_1}$:

2(n-1) additions, O(n)

1 division, O(1)

where O stands for the order of magnitude. The results of the above analysis show that the computational complexity of the method for estimation of the condition number of the matrix A is $O(n^3)$.

3. Defenition of the Nonlinear System

As a classical flow pattern, a nanofluid is flowing in laminar flow through a horizontal tube as the result of a pressure difference where the nanofluid enters at a uniform temperature T_i and the tube surface is maintained at a uniform temperature T_s ($T_s>T_i$).

To obtain the steady state temperature distribution in the tube, the conservation laws, which are expressed as differential equations [21, 22], must be solved. The differential equations are transformed to the algebraic equations (discretization) by the finite volume method [19, 23]. The discretized equations are as Eqs. (15.a) to (15.e)

$$(\rho_{nf}u_zA)_e - (\rho_{nf}u_zA)_w + (\rho_{nf}u_rA)_n - (\rho_{nf}u_rA)_s = 0 \qquad (15.a)$$

$$(\phi\rho_p u_zA)_e - (\phi\rho_p u_zA)_w + (\phi\rho_p u_rA)_n - (\phi\rho_p u_rA)_s = 0 \qquad (15.b)$$

$$(\rho_{nf}u_ru_zA)_e - (\rho_{nf}u_ru_zA)_w + (\rho_{nf}u_ru_rA)_n - (\rho_{nf}u_ru_rA)_s + \frac{P_{node} - P_S}{dr}V + ((\tau_{zr}A)_e - (\tau_{zr}A)_w + (\tau_{rr}A)_n - (\tau_{rr}A)_e)) - (\overline{\rho}_{nf}(-g))V = 0 \qquad (15.c)$$

$$(\rho_{nf}u_zu_zA)_e - (\rho_{nf}u_zu_zA)_w + (\rho_{nf}u_zu_rA)_n - (\rho_{nf}u_zu_rA)_s + \frac{P_{node} - P_W}{dz}V + ((\tau_{zz}A)_e - (\tau_{zz}A)_w + (\tau_{rz}A)_n - (\tau_{rz}A)_e)) = 0 \qquad (15.d)$$

$$(u_z(\rho c_p)_{nf}TA)_e - (u_z(\rho c_p)_{nf}TA)_w + (u_r(\rho c_p)_{nf}TA)_n - (u_r(\rho c_p)_{nf}TA)_s - (K_{nf}\frac{\partial T}{\partial z}A)_e + (K_{nf}\frac{\partial T}{\partial z}A)_w - (K_{nf}\frac{\partial T}{\partial r}A)_n + (K_{nf}\frac{\partial T}{\partial r}A)_s = 0 \qquad (15.e)$$

where subscripts e, w, n and s denote the face directions, see Figure 1 , *A* is the cross-sectional area of the control volume face, $\rho_{nf}$is the nanofluid density, u_z is the axial velocity, $u_r$is the radial velocity, $\phi$is the nanoparticle volume fraction, ρ_p is the nanoparticle density, V is the volume of the control volume, dr is the node spacing in the radial direction (*r*), dz is the node spacing in the axial direction (*z*), P is the pressure, τ is the shear stress, T is the temperature, $(c_p)_{nf}$ is the heat capacity of the nanofluid and K_{nf} is the nanofluid thermal conductivity.

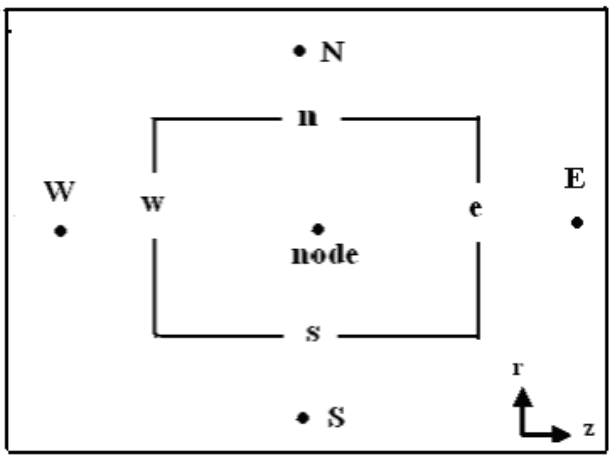

Figure 1. Two-dimensional illustration of a control volume in the tube in which the bold points denote the computational nodes (the capital letters) and the lowercase letters show the surfaces.

Eqs. (15.a) to (15.e) for each control volume, construct a system of nonlinear equations which its size depends on the number of partitions in the tube (unknowns in the nonlinear equations are ϕ , u_z, u_r, TandP).

4. Results and Discussion

Here, the nanofluid (water+Al_2O_3) is considered and the simulation is performed under these conditions:

the nanofluid temperature at the tube entrance =20 ℃ ,
the tube radius= 0.5 cm,
the tube length=1 cm,
the tube surface temperature= 40 ℃ ,
the pressure at the tube entrance =100000 pa,
the Reynolds numbers=800,
the nanoparticle volume fraction at the tube entrance =0.02
(the simulation code was written in C++).

We consider two cases:

Case (1)
The number of control volumes (partitions) in the tube=50,
The number of nonlinear equations=200

Case (2)
The number of control volumes in the tube=90,
The number of nonlinear equations=360.

However, base on the condition number estimated by the method explained in section 2.1, the matrix J^TJ is ill-conditioned for two cases.

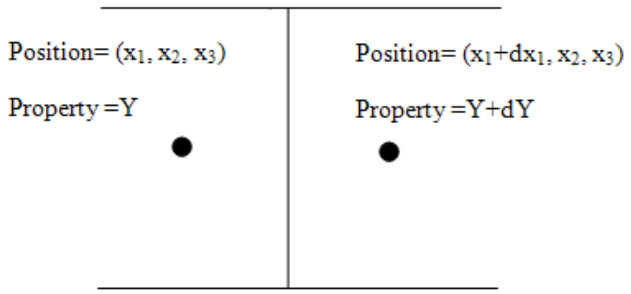

Figure 2 . A schematic illustration of the computational nodes (control volumes) in the domain (tube).

Now, we want to discuss why ill-conditioning occurs. First, a system with two linear equations is considered. Geometrically, two equations in two unknowns represent two straight lines and the point of intersection is the solution for the system; an ill-conditioned system represents two straight lines that are almost parallel where if one of the lines is tilted only slightly, the point of intersection (i.e. the solution for the system) is drastically altered [24]. Considering Ax=b, it seems, where there are some rows in the matrix A in which corresponding elements (in the matrix A and the vector b) have the stated relationship (similar to two straight lines that are almost parallel), the linear system Ax=b may be ill-conditioned. Each discretized location (control volume) is very close to its neighborhoods, as shown schematically in Figure 2, where the difference in property Y (such as ϕ, T and *P*) between two successive control volumes is small. Therefore, the resulting equations, Eqs. (15.a) to (15.e), for each discretized location and its neighborhoods can be quite similar, which may lead to linear relationships among the rows of the Jacobian matrix (J), see Eq. (3), and this affects the matrix's nearness to singularity. On the other hand, the matrix J^TJ and possibly (J^TJ+λD) will be more ill-conditioned, if J is ill-conditioned [23].

The pseudocode of the Levenberg-Marquardt algorithm can be presented as

$\lambda=0.9\times\|r(x)\|^2;\ \ (\|r(x)\|=\sqrt{[r_1(x)]^2+\cdots+[r_n(x)]^2})$

While $(\|r(x)\|>\varepsilon)\qquad \varepsilon\in(0,1)$

{

Solve Eq. (3) for obtaining S;

If (the step improves x_k *) Then* $\lambda=0.1\times\lambda$

Else $\lambda=10\times\lambda$*;*

}

where the damping factor (λ), initially can be set to $0.9\times\|r(x)\|^2$ [26]. As mentioned before, in this work, the line search strategy is used and an iteration step is acceptable when the Armijo condition is satisfied [15]:

$$f(x+\alpha S)\le f(x)+c_1\alpha\nabla f^TS\quad c_1\in(0,1), \tag{16}$$

$$f(x)=0.5\|r(x)\|^2=0.5\sum_{i=1}^{n}r_i^2 \tag{17}$$

where c_1 is a constant and, in practice, is chosen to be quite small ($c_1=10^{-4}$).

To compute the search direction S at each iteration of the Levenberg-Marquardt algorithm, Eq. (3) must be solved. When the matrix J^TJ is ill-conditioned, addition of the matrix J^TJ and the matrix λD may prevent ill-conditioning of the coefficient matrix (i.e., the condition number decreases with damping factor λ), indeed, λD may play role of a regularization term that turns the ill-posed system into a well-posed system.

As the pseudocode describes, when S is acceptable λ will be decreased by multiplying by 0.1. However, if λ is decreased unconditionally for such problems, the condition number may be greater than $(\text{machine epsilon})^{-0.5}$ leading to ill-conditioning. In other words, if J^TJ is ill-conditioned, λ must be chosen so that the condition number of (J^TJ+ λD) satisfies inequality (7), therefore, it is reasonable to set an approximate minimum value for λ (λ_{min}) per iteration, by a trial-and-error procedure, so that for obtaining the search direction $\lambda\ge\lambda_{min}$. Hence, the pseudocodeof the Levenberg-Marquardt algorithm for such problems can be modified:

$\lambda=0.9\times\|r(x)\|^2$*;*
Set λ_{min}*;*
If $(\lambda_{min}>0.9\times\|r(x)\|^2$ *Then* $\lambda=\lambda_{min}$ *;*
While $(\|r(x)\|>\varepsilon)\qquad \varepsilon\in(0,1)$
{
Solve Eq. (3) for obtaining S;
If (the step improves x_k *) Then*
{
Set λ_{min} *;*
If $(0.1\times\lambda>\lambda_{min})$ *Then* $\lambda=0.1\times\lambda$ *;*
Else If $(\lambda_{min}>\lambda)$ *Then* $\lambda=\lambda_{min}$ *;*
}
Else $\lambda=10\times\lambda$*;*

}

Considering inequality (7), we define the reduced condition number as Eq. (18)

$$\text{Reduced condition number} = \frac{\text{cond(A)}}{(\text{machine epsilon})^{-0.5}} \quad (18)$$

then, inequality (7) can be presented:

$$\text{Reduced condition number} < 1 \rightarrow \textit{solution is acceptable} \quad (19)$$

Figure 3 shows the process of obtaining a suitable damping factor (an approximate value for λ_{min}) for iteration 1 due to case (1) by a trial and error procedure in which the dashed line represents the maximum permissible reduced condition number. It can be seen that the reduced condition number due to λ=0.005 lies below the dashed line labeled with Try (3), therefore, λ=0.005 can be considered as λ_{min} for iteration 1 due to case (1).

Here, instead of using the modified algorithm (the above pseudocode) and finding λ_{min} for iteration 2 and so on, we want to test using λmin due to iteration 1 in the next iterations. In Figure 4, λ_{min} obtained for iteration 1 due to case (1) is used in the next iterations (the dashed line represents the maximum permissible reduced condition number), where the temperature distribution is obtained in iteration 4. It is clear that λ_{min} obtained for iteration 1, λ=0.005, prevents the linear system from being ill-conditioned in iterations 2 to 4, as shown in Figure 4, where their reduced condition numbers lie below the dashed line.

Finding λ_{min} for iteration 1 due to Case (2) is similar to Case (1), hence, we ignore discussion about it.

Similar to Figure 4, Figure 5 shows that λ_{min} obtained for iteration 1 due to case (2) , λ=0.01, prevents the linear system from being ill-conditioned in iterations 2 to 3, where the temperature distribution is obtained in iteration 3.

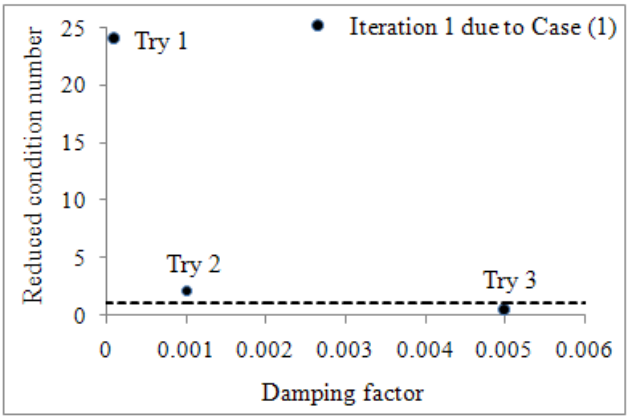

Figure 3 . Finding a suitable damping factor (an approximate value for λ_{min}) in iteration 1 due to case (1) by a trial and error procedure.

Although the results are limited to one case study, but the observed trend may be applicable to other nonlinear system obtained by the discretization methods.

Indeed, the results suggest that setting a suitable value for λ (an approximate value for λ_{min}) in the first iteration of the Levenberg-Marquardt algorithm for such problems may prevent ill conditioning in the next iterations, this provides considerable computational savings. Since, as explained before, the computational complexity of the method for estimation of the condition number of the matrix A is $O(n^3)$ where n, as the matrix dimension, is equal to the number of unknowns (or equations) of the nonlinear system.

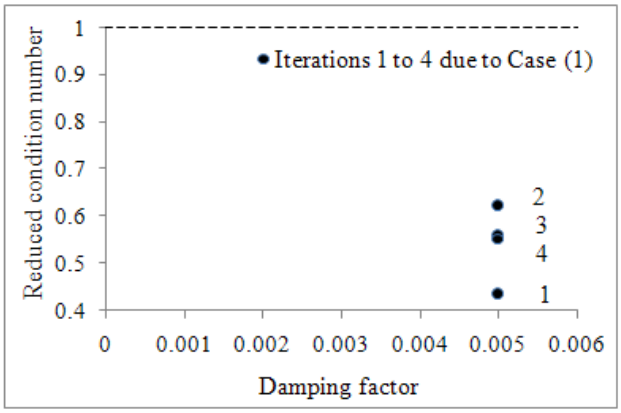

Figure 4 . Testing the suitable damping factor (an approximate value for λ_{min}) of iteration 1 due to case (1) in the next iterations; the temperature distribution is obtained in iteration 4.

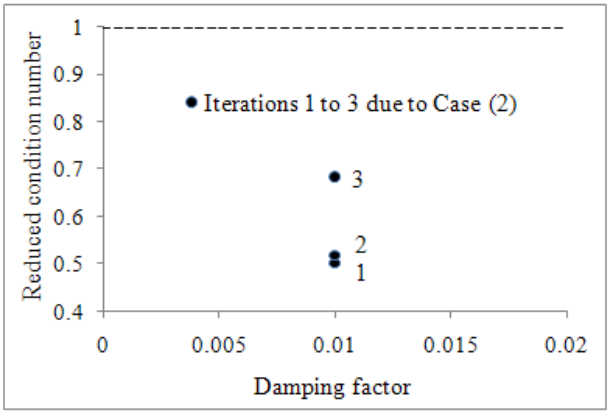

Figure 5 . Testing the suitable damping factor (an approximate value for λ_{min}) of iteration 1 due to case (2) in the next iterations.**;** the temperature distribution is obtained in iteration 3.

5. Conclusion

In this paper, we have considered the Levenberg-Marquardt algorithm from a different point of view where its linear system is ill-conditioned. To handle ill conditioning in solving this linear system, we have modified the Levenberg-Marquardt algorithm, however, the computational complexity of the modified algorithm is considerable where the computational complexity of estimation of the condition number of the coefficient matrix due to the linear system per iteration is $O(n^3)$. Considering one case study, the nanofluid convective heat transfer in a tube, we have found that setting an approximate minimum value for λ in the first iteration prevents ill conditioning in the next iterations leading to considerable computational savings. Indeed, the observed trend may be applicable to other nonlinear system; further analysis on the behavior of matrix $(J^TJ+\lambda D)$ in successive iterations may be an interesting topic.

References

[1] S. K. Das, S. U. S. Choi, W. Yu, T. Pradeep, Nanofluids: Science and Technology. Wiley, 2007.

[2] A. Sari, H. Fatahian, E. Fatahian, Numerical Investigation of Heat Transfer of Flow Over the Cylinder with High Conductivity Fins, Computational Research Progress in Applied Science & Engineering 3 (2017) 85–95.
[3] A. Terfous, Y. Lin, A. Gheaim, Numerical Simulation of Sediment Transport In Tanks with the Influence of Cavity, Computational Research Progress in Applied Science & Engineering 5 (2019) 122–126.
[4] J.E. Dennis, R.B. Schnabel, Numerical Methods for Unconstrained Optimization and Nonlinear Equations (Classics in Applied Mathematics). SIAM, 1987.
[5] A.N. Tikhonov, V.Y. Arsenin, Solutions of Ill Posed Problems. Wiley, 1977.
[6] A. klinger, Approximate pseudoinverse solutions to ill-conditioned linear systems, Journal of Optimization Theory and Applications 2(1968) 117–-124.
[7] I.H. Mufti, C.K. Chow, F.T. Stock, Solution of Ill-Conditioned linear Two-Point Boundary Value Problems by the Riccati Transformation, SIAM Review 11 (1969) 616–619.
[8] J.M. Varah, On the Numerical Solution of Ill-Conditioned Linear Systems with Applications to Ill-Posed Problems, SIAM Journal on Numerical Analysis 10 (1973) 257–267.
[9] C. Brezinski, M. Redivo-Zaglia, G. Rodriguez, S. Seatzu, Extrapolation techniques for ill-conditioned linear systems. Numerische Mathematik 81 (1998) 1–29.
[10] H.J. Kim, K. Choi, H.B. Lee, H.K. Jung, S.Y. Hahn, A new algorithm for solving ill conditioned linear systems, IEEE Transactions on Magnetics 32 (1998) 1373–1376.
[11] C. Brezinski, M. Redivo-Zaglia, G. Rodriguez, S. Seatzu, Multi-parameter regularization techniques for ill-conditioned linear systems, Numerische Mathematik 94 (2003) 203–228.
[12] G. Rodriguez, S. Seatzu, D. Theis, A New Technique for Ill-Conditioned Linear Systems, Numerical Algorithms 33 (2003) 433–442.
[13] N.S. Hoang, A.G. Ramm, Solving ill-conditioned linear algebraic systems by the dynamical systems method, Inverse problems in science and Engineering 16 (2008) 617–630.
[14] J.D. Hoffman, S. Frankel, Numerical Methods for Engineers and Scientists. CRC Press, 2001.
[15] J. Nocedal , S.J. Wright, Numerical Optimization. Springer, 1999.
[16] K. Levenberg, A Method for the Solution of Certain Non-linear Problems in Least Squares, Quarterly of Applied Mathematics 2 (1944) 164–168.
[17] D.W. Marquardt, An Algorithm for the Least-Squares Estimation of Nonlinear Parameters, Journal of the Society for Industrial and Applied Mathematics 11 (1963) 431–441.
[18] S.S. Azimi, M. Kalbasi, H. Sadeghifar, Sensitivity Analysis of Merit Function in Solving Nonlinear Equations by Optimization, Journal of Optimization Theory and Applications 162 (2014) 191–201.
[19] H. Ferziger, M. Peric, Computational methods for fluid dynamics. Springer, 2002.
[20] J. H. Wilkinson, The Algebraic Eigenvalue Problem. Claredo, 1965.
[21] R.B. Bird, W.E. Stewart, E.N. Lightfoot, Transport Phenomena. Wiley, 2007.
[22] W. M. Kays, M. E. Crawford, Convective heat and mass transfer. McGraw-Hill, 1993.
[23] H. K. Versteeg, W. Malalasekera, An introduction to computational fluid dynamics The finite volume method. Longman Scientific & Technical, 1995.
[24] C.D. Meyer, Matrix Analysis and Applied Linear Algebra. SIAM, 2000.
[25] B.N. Datta, Numerical Methods for Linear Control Systems:Design and Analysis. Elsevier Academic Press, 2004.
[26] C. Kanzow, N. Yamashita, M. Fukushima, Levenberg-Marquardt methods for constrained nonlinear equations with strong local convergence properties, Journal of computational and applied mathematics 172 (2004) 375–397.