A MANIFOLD-BASED EXPONENTIALLY CONVERGENT ALGORITHM FOR SOLVING NON-LINEAR PARTIAL DIFFERENTIAL EQUATIONS

Chein-Shan Liu
Department of Civil Engineering, National Taiwan University, Taipei, Taiwan, R.O.C., liucs@ntu.edu.tw

Follow this and additional works at: https://jmstt.ntou.edu.tw/journal

Part of the Ocean Engineering Commons

Recommended Citation
DOI: 10.6119/JMST-011-0421-1
Available at: https://jmstt.ntou.edu.tw/journal/vol20/iss4/12

This Research Article is brought to you for free and open access by Journal of Marine Science and Technology. It has been accepted for inclusion in Journal of Marine Science and Technology by an authorized editor of Journal of Marine Science and Technology.
A MANIFOLD-BASED EXPONENTIALLY CONVERGENT ALGORITHM FOR SOLVING NON-LINEAR PARTIAL DIFFERENTIAL EQUATIONS

Acknowledgements
Taiwan's National Science Council project NSC-99-2221- E-002-074-MY3 granted to the author is highly appreciated.

This research article is available in Journal of Marine Science and Technology: https://jmstt.ntou.edu.tw/journal/vol20/iss4/12
A MANIFOLD-BASED EXponentially CONVERGENT ALGORITHM FOR SOLVING NON-LINEAR PARTIAL DIFFERENTIAL EQUATIONS

Chein-Shan Liu

Key words: non-linear algebraic equations (NAEs), non-linear partial differential equations, manifold-based exponentially convergent algorithm (MBECA), fictitious time integration method (FTIM).

ABSTRACT

For solving a non-linear system of algebraic equations of the type:

\[ F_i(x_j) = 0, \quad i, j = 1, \ldots, n, \]

a Newton-like algorithm is still the most popular one; however, it had some drawbacks as being locally convergent, sensitive to initial guess, and time consumption in finding the inversion of the Jacobian matrix \( \frac{\partial F_i}{\partial x_j} \). Based-on a manifold defined in the space of \( (x_i, t) \) we can derive a system of non-linear Ordinary Differential Equations (ODEs) in terms of the fictitious time-like variable \( t \), and the residual error is exponentially decreased to zero along the path of \( x(t) \) by solving the resultant ODEs. We apply it to solve 2D non-linear PDEs, and the vector-form of the matrix-type non-linear algebraic equations (NAEs) is derived. Several numerical examples of non-linear PDEs show the efficiency and accuracy of the present algorithm. A scalar equation is derived to find the adjustive fictitious time stepsize, such that the irregular bursts appeared in the residual error curve can be overcome. We propose a future direction to construct a really exponentially convergent algorithm according to a manifold setting.

I. INTRODUCTION

In many practical non-linear engineering problems governed by partial differential equations, the methods such as the finite element method, boundary element method, finite volume method, the meshless method, etc., eventually led to a system of non-linear algebraic equations (NAEs). Many numerical methods used in the computational mechanics, such as demonstrated by Zhu et al. [31], Atluri and Zhu [6], Atluri [2], Atluri and Shen [5], and Atluri et al. [3] were always led to the solution of a system of linear algebraic equations for a linear problem, and of an NAEs system for a non-linear problem.

Over the past forty years two important contributions have been made towards the numerical solutions of NAEs. One of the methods has been called the “predictor-corrector” or “pseudo-arclength continuation” method. This method has its historical roots in the embedding and incremental loading methods which have been successfully used for several decades by engineers to improve the convergence properties when an adequate starting value for an iterative method is not available. Another is the so-called simplical or piecewise linear method. The monographs by Allgower and Georg [1] and Deuflhard [12] are devoted to the continuation methods for solving NAEs.

Liu and Atluri [23] have employed the technique of Fictitious Time Integration Method (FTIM) to solve a large system of NAEs, and showed that high performance can be achieved by using the FTIM. More recently, Liu [17] has used the FTIM technique to solve the non-linear complementarity problems. Then, Liu [18, 19] has used the FTIM to solve the boundary value problems of elliptic type partial differential equations. Liu and Atluri [24] also employed this technique of FTIM to solve mixed-complementarity problems and optimization problems. Liu and Atluri [25] using the technique of FTIM solved the inverse Sturm-Liouville problem under specified eigenvalues. Liu [20] has utilized a simple FTIM to compute both the forward and backward in time problems of Burgers equation, and has found that the FTIM is robust against the disturbance of noise. Liu and Chang [28] have used the FTIM to solve ill-posed linear problems. For its numerical implementation being quite simple, the FTIM was also used in other problems, like as, Liu [21, 22], Liu and Atluri [26], Chang and Liu [8], Chi et al. [10], and Tsai et al. [30].

In spite of its success, the FTIM is local convergence and needs to determine viscous damping coefficients for different equations in one problem. Liu and Atluri [27] were the first to
make a breakthrough by deriving a residual-norm based algorithm, which can circumvent the drawbacks of FTIM and Newton’s algorithm; also refer Atluri et al. [4].

In this paper we introduce a novel continuation method of Manifold-Based Exponentially Convergent Algorithm (MBECA), which can be easily implemented to solve non-linear partial differential equations, after some suitable discretizations by using the finite difference method or expansion by the radial basis function [14]. To remedy the shortcoming of vector homotopy method as initiated by Davidenko [11], Liu et al. [29] have proposed a scalar homotopy method, and Ku et al. [15] combined this idea with the exponentially decayed scalar homotopy function, developing a Manifold-Based Exponentially Convergent Algorithm (MBECA). In this paper we will point out the limitation of MBECA, and explain why it may stuck and after that the algorithm cannot proceed to find solution.

II. THEORETICAL BASIS

For the following non-linear algebraic equations (NAEs) in a vector form:

\[ \mathbf{F}(\mathbf{x}) = \mathbf{0}, \]  

we will propose a numerical algorithm to solve it based on a space-time manifold.

1. Motivation

To motivate the present approach, let us consider an uncoupled algebraic system:

\[ x - 1 = 0, \quad y - 1 = 0. \]  

The above two equations can be combined into a single one:

\[ \frac{1}{2} \| \mathbf{F} \|^2 = \frac{1}{2} [(x-1)^2 + (y-1)^2] = 0. \]  

We can see that it is a circle in the spatial-plane \((x, y)\) with a center \((1,1)\) but with a zero radius.

Now, we introduce an extra variable \(t\) as a fictitious time-like variable, and let \(x\) and \(y\) be functions of \(t\). We consider the following equation defined in the space-time domain by

\[ \frac{1}{2} e^{2\alpha t} \| \mathbf{F} \|^2 = \frac{1}{2} e^{2\alpha t} [(x(1) - 1)^2 + (y(1) - 1)^2] = C, \]  

where \(\alpha > 0\) and \(C\) is determined by the initial values of \(x(0) = x_0\) and \(y(0) = y_0\) by

\[ C = \frac{1}{2} [(x_0 - 1)^2 + (y_0 - 1)^2]. \]

Eq. (4) can be written as

\[ (x-1)^2 + (y-1)^2 = 2Ce^{-2\alpha t}, \]  

which is a manifold in the space-time domain \((x, y, t)\) at each cross-section of a fixed \(t\) it being a circle with center \((1,1)\) and with a radius \(\sqrt{2Ce^{-2\alpha t}}\). We can construct the following ODEs:

\[ \dot{x} = -\alpha(x-1), \quad \dot{y} = -\alpha(y-1), \]  

such that the path of \((x(t), y(t))\) generated from the above ODEs is located on the manifold defined by Eq. (5).

Solving Eq. (6) we have

\[ x(t) = [x_0 - 1]e^{-\alpha t} + 1, \quad y(t) = [y_0 - 1]e^{-\alpha t} + 1. \]

Inserting them into Eq. (4) we indeed can prove that \((x(t), y(t))\) fast tends to the solution \((1,1)\).

2. A Setting Based on Manifold

From the above idea of space-time manifold, for Eq. (1) we can consider

\[ h(\mathbf{x}, t) = \frac{1}{2} Q(t) \| \mathbf{F}(\mathbf{x}) \|^2 = C, \]  

where \(Q(t) > 0\) is a given function of \(t\), monotonically increasing with \(t\) and with \(Q(0) = 1\), and \(C\) is determined by the initial condition \(\mathbf{x}(0) = \mathbf{x}_0\) with

\[ C = \frac{1}{2} \| \mathbf{F}(\mathbf{x}_0) \|^2. \]

Usually, \(C > 0\), and \(C = 0\) when the initial value \(x_0\) is just the root of Eq. (1). However, it is rare of this lucky case.

When \(C > 0\) and \(Q > 0\), the manifold defined by Eq. (8) is continuous, and thus the differential operation being taken on the manifold makes sense. For the requirement of consistency condition, by taking the differential of Eq. (8) with respect to \(t\) and considering \(\mathbf{x} = \mathbf{x}(t)\), we have

\[ \frac{1}{2} Q(t) \| \mathbf{F}(\mathbf{x}) \|^2 - \dot{Q}(t) (\mathbf{B}^\top \mathbf{F}) \cdot \dot{\mathbf{x}} = 0, \]  

where \(\mathbf{B}\) is the Jacobian matrix with its \(ij\)-component given by \(B_{ij} = \partial F_i / \partial x_j\).

Eq. (10) cannot uniquely determine the governing equation of \(\mathbf{x}\); however, we suppose that

\[ \dot{\mathbf{x}} = -\lambda \frac{\partial h}{\partial \mathbf{x}} = -\lambda Q(t) \mathbf{B}^\top \mathbf{F}, \]  

where \(\lambda > 0\).
where \( \lambda \) is to be determined. Inserting Eq. (11) into Eq. (10) we can solve for \( \lambda \) by

\[
\lambda = \frac{\dot{Q}(t) \|F\|^2}{2Q(t)\|B^T F\|^2}.
\]

Thus we obtain an evolution equation for \( x \) defined by the following ODEs:

\[
\dot{x} = -q(t) \frac{\|F\|^2}{\|B^T F\|^2}B^T F,
\]

where

\[
q(t) = \frac{\dot{Q}(t)}{2Q(t)}.
\]

There are many ways to choose a suitable function of \( Q(t) \); however, we can let

\[
q(t) = \frac{\dot{Q}(t)}{2Q(t)} = \frac{\nu}{2(1+t)^m}, \quad 0 < m \leq 1.
\]

Hence, we have

\[
Q(t) = \exp\left(\frac{\nu}{1-m}[(1+t)^{1-m} - 1]\right).
\]

Therefore we can derive the following equation:

\[
\dot{x} = -\frac{\nu}{2(1+t)^m} \frac{\|F\|^2}{\|B^T F\|^2}B^T F.
\]

It is not difficult to prove that the orbit \( x(t) \) with \( x(0) = x_0 \) solved from Eq. (17) is located on the space-time manifold defined by Eq. (8) with \( Q(t) \) given by Eq. (16). Hence, we have an exponentially convergent property in solving the NAEs in Eq. (1):

\[
\|F(x)\|^2 = \frac{2C}{Q(t)}.
\]

When \( t \) is increasing the above equation enforces the residual error \( \|F(x)\| \) tending to zero exponentially, and meanwhile the solution of Eq. (1) is obtained approximately. However, it is a great challenge by developing a suitable numerical integrator for Eq. (17), such that the orbit of \( x \) can really retain on the manifold.

### III. NUMERICAL METHODS

#### 1. Adjusting the Fictitious Time Step

In order to keep \( x \) on the manifold defined by Eq. (18) we can consider the evolution of \( F \) along the path \( x(t) \) by

\[
\dot{F} = Bx = -q(t) \frac{\|F\|^2}{\|B^T F\|^2}AF,
\]

where

\[
A := BB^T.
\]

Suppose that we simply use the Euler forward scheme to integrate Eq. (19), which yields

\[
F(t + \Delta t) = F(t) - q(t)\Delta t \frac{\|F\|^2}{\|B^T F\|^2}AF.
\]

Taking the square-norms of both the sides and using Eq. (18) we can obtain

\[
\frac{2C}{Q(t + \Delta t)} = \frac{2C}{Q(t)} - 2q(t)\Delta t \frac{2C}{Q(t)} \frac{F(AF)}{\|B^T F\|^2} + (q(t)\Delta t)^2 \frac{2C}{Q(t)} \frac{\|F\|^2}{\|B^T F\|^2} \frac{\|AF\|^2}{\|B^T F\|^2}.
\]

Thus we have the following scalar equation to solve \( \Delta t \):

\[
f(\Delta t) := a(\Delta t)^2 - b\Delta t + 1 - \frac{Q(t)}{Q(t + \Delta t)} = 0,
\]

where

\[
a := q^2(t) \frac{\|F\|^2}{\|B^T F\|^2} \frac{\|AF\|^2}{\|B^T F\|^2},
\]

\[
b := 2q(t).
\]

Obviously, \( \Delta t = 0 \) is a root of Eq. (23); however, we prefer to search another one with \( \Delta t > 0 \). On the other hand, we note that \( \nu \) in Eq. (16) cannot be too large, say \( \nu > 100 \); otherwise, the term \( Q(t)/Q(t + \Delta t) \) in Eq. (23) can be very near to zero, and thus makes Eq. (23) having no real solution, because of

\[
(-b)^2 - 4a = 4q^2 \left[ 1 - \frac{\|F\|^2}{\|B^T F\|^2} \frac{\|AF\|^2}{\|B^T F\|^2} \right] < 0.
\]
which is due to
\[
\|B^2F\| = F \cdot (AF) < \|F\| \|AF\|. \tag{26}
\]

We can apply the FTIM to find the solution of Eq. (23):
\[
\dot{x} = -\frac{\mu}{(1+t)^2} f(x). \tag{27}
\]

When \(\Delta t\) is solved, we can use the following iteration to calculate \(x(t + \Delta t)\):
\[
x(t + \Delta t) = x(t) - q(t)\Delta t \frac{\|F\|}{\|B^2F\|} + \frac{\|F\|}{\|B^2F\|} B^2F. \tag{28}
\]

The above algorithm is indeed a very powerful numerical method with exponentially convergent speed and the orbit of \(x(t)\) being retained on the manifold. Thus, we may call the present algorithm a Manifold-Based Exponentially-Convergent Algorithm (MBECA).

2. A Matrix Type NAEs

Sometimes we may encounter the NAEs generated from the discretization of PDE with a matrix type. In this situation it is not so straightforward to write its counterpart as being a vector-form NAEs. Let us consider
\[
\Delta u(x, y) = F(x, y, u, u_x, u_y, \ldots), (x, y) \in \Omega, \tag{29}
\]
\[
u(x, y) = H(x, y), (x, y) \in \Gamma, \tag{30}
\]
where \(\Delta\) is the Laplacian operator, \(\Gamma\) is the boundary of a problem domain \(\Omega := [a_0, a_1] \times [b_0, b_1]\), and \(F\) and \(H\) are given functions.

By a standard finite difference applied to Eq. (29), one has a system of NAEs of matrix type:
\[
F_{i,j} = \frac{1}{(\Delta x)^2} [u_{i+1,j} - 2u_{i,j} + u_{i-1,j}] + \frac{1}{(\Delta y)^2} [u_{i,j+1} - 2u_{i,j} + u_{i,j-1}]
- F \left( x, y, u_{i-1,j}, \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}, \frac{u_{i+1,j} - u_{i,j-1}}{2\Delta y}, \ldots \right)
= 0, 1 \leq i \leq n_1, 1 \leq j \leq n_2. \tag{31}
\]

Here, we divide the rectangle \(\Omega\) by a uniform grid with \(\Delta x = (a_1 - a_0)/(n_1 + 1)\) and \(\Delta y = (b_1 - b_0)/(n_2 + 1)\) being the uniform spatial grid lengths in the \(x\)- and \(y\)-direction, and \(u_{i,j} := u(x_i, y_j)\) be a numerical value of \(u\) at the grid point \((x_i, y_j) \in \Omega\).

Let \(K = n_2(i - 1) + j\) and with \(i\) running from 1 to \(n_1\) and \(j\) running from 1 to \(n_2\) we can, respectively, set the vectorial variables \(x_K\) and the vectorial algebraic equations \(F_K\) by
\[
\text{Do } i = 1, n_1
\]
\[
\text{Do } j = 1, n_2
\]
\[K = n_2(i - 1) + j \tag{32}\]
\[x_K = u_{i,j}
\]
\[F_K = F_{i,j}, \]
\[\text{At the same time the components of the Jacobian matrix } B \text{ are constructed by}\]
\[
\text{Do } i = 1, n_1
\]
\[
\text{Do } j = 1, n_2
\]
\[K = n_2(i - 1) + j \tag{33}\]
\[
B_{K,i_1} = \frac{1}{(\Delta x)^2} \left( x, y, u_{i-1,j}, \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}, \frac{u_{i+1,j} - u_{i,j-1}}{2\Delta y}, \ldots \right)
+ \frac{1}{2\Delta x} F_u \left( x, y, u_{i,j}, \frac{u_{i+1,j} - u_{i,j-1}}{2\Delta x}, \frac{u_{i+1,j} - u_{i,j-1}}{2\Delta y}, \ldots \right)
\]
\[
B_{K,i_2} = \frac{1}{(\Delta y)^2} \left( x, y, u_{i,j+1}, \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}, \frac{u_{i+1,j} - u_{i,j-1}}{2\Delta y}, \ldots \right)
+ \frac{1}{2\Delta y} F_u \left( x, y, u_{i,j}, \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}, \frac{u_{i+1,j} - u_{i,j-1}}{2\Delta y}, \ldots \right)
\]
\[
B_{K,i_3} = \frac{1}{(\Delta x)^2} \left( x, y, u_{i+1,j}, \frac{u_{i+1,j} - u_{i,j}}{2\Delta x}, \frac{u_{i+1,j} - u_{i,j-1}}{2\Delta y}, \ldots \right)
+ \frac{1}{2\Delta x} F_u \left( x, y, u_{i,j}, \frac{u_{i+1,j} - u_{i,j-1}}{2\Delta x}, \frac{u_{i+1,j} - u_{i,j-1}}{2\Delta y}, \ldots \right)
\]
\[
B_{K,i_4} = \frac{1}{(\Delta y)^2} \left( x, y, u_{i,j}, \frac{u_{i+1,j} - u_{i,j}}{2\Delta x}, \frac{u_{i+1,j} - u_{i,j-1}}{2\Delta y}, \ldots \right)
+ \frac{1}{2\Delta y} F_u \left( x, y, u_{i,j}, \frac{u_{i+1,j} - u_{i,j-1}}{2\Delta x}, \frac{u_{i+1,j} - u_{i,j-1}}{2\Delta y}, \ldots \right)
\]
\[
B_{K,i_5} = \frac{1}{(\Delta x)^2} \left( x, y, u_{i,j}, \frac{u_{i+1,j} - u_{i-1,j}}{2\Delta x}, \frac{u_{i+1,j} - u_{i,j-1}}{2\Delta y}, \ldots \right)
+ \frac{1}{2\Delta x} F_u \left( x, y, u_{i,j}, \frac{u_{i+1,j} - u_{i,j-1}}{2\Delta x}, \frac{u_{i+1,j} - u_{i,j-1}}{2\Delta y}, \ldots \right)
\]
In above, \( F_u, F_w, \) and \( F_y \) denote, respectively, the partial differentials of the function \( F(x, y, u, u_x, u_y, \ldots) \) with respect to \( u, u_x \) and \( u_y \). When the above quantities are available, we can apply the vector-form MBEC to solve the non-linear PDE in Eq. (29).

**IV. NUMERICAL TESTS OF MBEC**

In this section we apply the new method of MBEC to solve some non-linear PDEs.

1. **Example 1**

   We consider a non-linear heat conduction equation:
   \[
   u_t = \alpha(x)u_{xx} + \alpha'(x)u_x + u^2 + h(x,t),
   \]
   (34)
   \[
   \alpha(x) = (x-3)^2, \quad h(x,t) = -7(x-3)^2 e^{-t} - (x-3)^2 e^{-2t},
   \]
   (35)
   with a closed-form solution \( u(x, t) = (x-3)^2 e^{-t} \).

   By applying the MBEC to solve the above equation in the domain of \( 0 \leq x \leq 1 \) and \( 0 \leq t \leq 1 \) we fix \( n_1 = n_2 = 13, \nu = 500 \) and \( m = 1 \). We apply the group-preserving scheme (GPS) developed by Liu [16] to integrate the resultant ODEs with a fictitious time stepsize \( \Delta t = 0.005 \). In Fig. 1(a) we show the residual errors with respect to the number of steps up to 2000. The absolute errors of numerical solution are plotted in Fig. 1(b), which reveal an accurate numerical result with the maximum error being \( 6.288 \times 10^{-3} \).

2. **Example 2**

   One famous mesh-less numerical method to solve the non-linear PDE of elliptic type is the radial basis function (RBF) method, which expands the solution \( u \) by
   \[
   u(x, y) = \sum_{k=1}^n a_k \phi_k,
   \]
   (36)
   where \( a_k \) are the expansion coefficients to be determined and \( \phi_k \) is a set of RBFs, for example,
   \[
   \phi_k = (r_k^2 + c^2)^{N-3/2}, \quad N = 1, 2, \ldots;
   \]
   \[
   \phi_k = r_k^{2N} \ln r_k, \quad N = 1, 2, \ldots;
   \]
   \[
   \phi_k = \exp \left( -\frac{r_k^2}{a^2} \right),
   \]
   \[
   \phi_k = (r_k^2 + c^2)^{N-3/2} \exp \left( -\frac{r_k^2}{a^2} \right), \quad N = 1, 2, \ldots;
   \]
   \[
   \phi_k = (r_k^2 + c^2)^{N-3/2} \exp \left( -\frac{r_k^2}{c^2} \right), \quad N = 1, 2, \ldots;
   \]
   where the radius function \( r_k \) is given by \( r_k = \sqrt{(x-x_k)^2 + (y-y_k)^2} \), while \( (x_k, y_k), k = 1, \ldots, n \) are called source points. The constants \( a \) and \( c \) are shape parameters. In the below we take the first set of \( \phi_k \) as trial functions, which is known as a multi-quadric RBF [9, 13], with \( N = 2 \).

   In this example we apply the multi-quadric radial basis function to solve the following non-linear PDE:
   \[
   \Delta u = 4u^3(x^2 + y^2 + a^2),
   \]
   (38)
   where \( a = 4 \) was fixed. The domain is an irregular domain with
   \[
   \rho(\theta) = (\sin 2\theta)^2 \exp(\sin \theta) + (\cos 2\theta)^2 \exp(\cos \theta).
   \]
   (39)
   The analytic solution is given by
   \[
   u(x, y) = \frac{-1}{x^2 + y^2 - a^2},
   \]
   (40)
   which is singular on the circle with a radius \( a \).
Inserting Eq. (36) into Eq. (38) and placing some field points inside the domain to satisfy the governing equation and some points on the boundary to satisfy the boundary condition we can derive $n$ NAEs to determine the $n$ coefficients $a_i$. The source points $(x_k, y_k)$, $k = 1, \ldots, n$ are uniformly distributed on a contour given by $R_0 + \rho \theta_k$, where $\theta_k = 2k\pi/n$. Under the following parameters $R_0 = 0.5, c = 0.5, \nu = 5, m = 0.01$ and $\Delta t = 0.01$, in Fig. 2(a) we show the residual errors up to 1000 steps. It can be seen that in the residual-error curve there is a little oscillation, and it decays very fast at the first few steps. The absolute error of numerical solution is plotted in Fig. 2(b), which is accurate with the maximum error being $8.57 \times 10^{-3}$.

Now we test the effect by adjusting the stepsize. Under the following parameters $R_0 = 0.5, c = 0.1, \nu = 5, m = 0.01$, in Fig. 3 we compare the residual errors obtained by fixing a stepsize with $\Delta t = 0.01$, which exhibits many irregular bursts as shown by the dashed line, and that obtained by adjusting the stepsize as shown by a solid line, which is smooth and decreased. The adaptive stepsize is also shown by the dashed-dotted line up to 500 steps.

3. Example 3

In this example we apply the MBCEA to solve the following boundary value problem of non-linear elliptic equation [6, 7, 31, 32]:

$$\Delta u(x, y) + \omega^2 u(x, y) + \epsilon u^3(x, y) = p(x, y).$$  \hspace{1cm} (41)

While the exact solution is

$$u(x, y) = \frac{-5}{6}(x^3 + y^3) + 3(x^2y + xy^2),$$  \hspace{1cm} (42)

the exact $p$ can be obtained by inserting the above $u$ into Eq. (41).

By introducing a finite difference discretization of $u$ at the grid points we can obtain

$$F_{ij} = \frac{1}{(\Delta x)^2}(u_{i+1,j} - 2u_{i,j} + u_{i-1,j}) + \frac{1}{(\Delta y)^2}(u_{i,j+1} - 2u_{i,j} + u_{i,j-1})$$

$$+ \omega^2 u_{i,j} + \epsilon u_{i,j}^3 - p_{i,j} = 0.$$  \hspace{1cm} (43)

The boundary conditions can be obtained from the exact solution in Eq. (42).

Under the following parameters $n_1 = n_2 = 13, \Delta t = 0.005, \nu = 500, \omega = 1$ and $\epsilon = 0.001$ we compute the solution of the above system, and compare them with the exact solution. In Fig. 4(a) we show the residual errors up to 2000 steps. The absolute errors of numerical solution are plotted in Fig. 4(b), of which the maximum error is smaller than $1.626 \times 10^{-3}$. 

![Fig. 2. For example 2: (a) showing the convergence speed of residual errors, and (b) displaying the numerical error.](image1)

![Fig. 3. For example 2 comparing the residual errors obtained by a fixed stepsize and an adaptive stepsize.](image2)

![Fig. 4. For example 3 showing the residual errors up to 2000 steps.](image3)
V. LIMITATION

Before Eq. (11) we have mentioned that the governing equation of $x$ is not unique. Indeed, it is not a necessary condition but a sufficient condition to satisfy Eq. (10). Thus the following algorithm:

$$x(t + \Delta t) = x(t) - q(t)\Delta t - \frac{[F]}{[B^T F]} [B^T F]$$  (44)

based-on Eq. (13) has some limitations. Obviously, from the residual error curves as shown in Figs. 1(a), 2(a), 3 and 4(a) there is a common pattern that the curves decrease very fast at the beginning, and then they all tend to be flattened with a very slow convergence, without having an exponential convergence no longer. Another feature as obviously shown in Figs. 1(a) and 4(a) is that the curves exhibit irregular bursts. To solve the latter problem we can employ the technology in Section III.1 by adjusting the time stepsize. As shown in Fig. 5 we plot the profile time stepsize and residual error for example 1. In the curve of residual error the bursts disappear. However, it still has the first problem. More difficultly, the time stepsize adjusting technique spends a lot of computational time to find $\Delta t$.

In order to find the reason for a slow convergence of the residual error curve in its later stage we give the following analysis. Inserting Eqs. (24) and (25) into Eq. (23) we can derive

$$a_0(q\Delta t)^2 \geq 2q\Delta t + 1 - \frac{Q(t)}{Q(t + \Delta t)} = 0,$$  (45)

where

$$a_0 := \frac{\|F\|^2 \|AF\|^2}{\|B^T F\|^2} \geq 1$$  (46)

in view of Eq. (26).

From Eq. (14) and the approximation of

$$Q(t)\Delta t = Q(t + \Delta t) - Q(t),$$

we have

$$q(t)\Delta t = \frac{1}{2} |R(t)| = 1,$$  (47)
where the ratio $R(t)$ is defined by

$$R(t) = \frac{Q(t + \Delta t)}{Q(t)}. \quad (48)$$

As a requirement of $\dot{Q}(t) > 0$, we need $R(t) > 1$.

Thus, through some manipulations, Eq. (45) becomes

$$a_0 R^3(t) - (2a_0 + 4)R^2(t) + (a_0 + 8)R(t) - 4 = 0. \quad (49)$$

It is interesting that the above equation can be written as

$$[R(t) - 1]^2[a_0 R(t) - 4] = 0. \quad (50)$$

Because $R = 1$ is a double roots and it is not the desired one, we take

$$R(t) = \frac{4}{a_0} = \frac{4\|B^T F\|^2}{\|F\|\|AF\|^2}. \quad (51)$$

By using Eq. (47), Eq. (44) can now be written as

$$x(t + \Delta t) = x(t) - \frac{1}{2} [R(t) - 1] \frac{\|F\|^2}{\|B^T F\|} B^T F. \quad (52)$$

When $R(t)$ tends to 1, i.e. $a_0$ tends to 4, the dynamical force on the right-hand side is lost for the above equation. In Fig. 6 we show $a_0$, $R$ and the residual error with respect to the number of steps. It can be seen that the above dynamic equation loses its pushing force, and remains a large residual error due to $a_0$ tending to 4 and thus $R$ tending to 1.

VI. CONCLUSIONS

A manifold-based exponentially convergent algorithm (MBECA) was established in this paper. When we apply it to solve some 2D non-linear PDEs, the vector-form was derived. Several numerical examples of non-linear PDEs showed the efficiency and accuracy of MBECA. The irregular bursts appeared in the residual error curve can be overcome by solving a scalar equation to find the adjustable fictitious time stepsize. However, the governing Eq. (13) has faced a bottle-neck by only considering with the gradient vector as a unique source of dynamic force. An open problem is that how to construct a really exponentially convergent algorithm according to the setting of a manifold defined in Eq. (18).

ACKNOWLEDGMENTS

Taiwan’s National Science Council project NSC-99-2221-E-002-074-MY3 granted to the author is highly appreciated.

REFERENCES