Research Article

Mathematical Methods in the Applied Sciences

Received 24 March 2014 Published online 4 September 2015 in Wiley Online Library

(wileyonlinelibrary.com) DOI: 10.1002/mma.3610
MOS subject classification: 47J25; 49M15; 65E05

Some notes on split Newton iterative algorithm

Dongfang Li[a], Hongyu Qin[b*†], Xiujun Cheng[a] and Fengyan Wu[a]

Communicated by S. Wise

In this study, a parameterized split Newton method is derived by using the accelerating technique. Convergence and error estimates of the method are obtained. In practical application, the proposed method can give a better result in view of computational CPU time. Numerical examples on several partial differential equations are shown to illustrate our findings.

Keywords: error estimate; convergence; split Newton iterative algorithm; computational time

1. Introduction

This study is concerned with the numerical approximations of the following nonlinear equations

$$H(x) = 0, \quad x \in D \subset \mathbb{R}^n, \tag{1.1}$$

where H is Fréchet-differentiable operator defined on a convex subset D of a Banach space. These systems typically arise from the full discretization of time-dependent partial differential equations (PDEs), such as Burger–Fisher equations, Burger-Huxley equations and Korteweg–de Vries (KdV) equations. In order to approximate the locally unique solution of the nonlinear Equation (1.1) effectively, it is important to choose suitable iterative methods.

It is known that a classical algorithm to solve (1.1) is Newton iterative method, which generates a sequence of iterates x_k satisfying

$$x_{k+1} = x_k - [H'(x_k)]^{-1} H(x_k), \quad k = 0, 1, 2, \ldots, \tag{1.2}$$

where $H'(x_k)$ denotes the Jacobian matrix of $H(x)$ at x_k; for a more detailed description of Newton iterative method as well as its implementation and application, we refer the readers to the survey papers [1–4] and the references therein. When the classical Newton iterative method (4.2) is applied to solve the nonlinear problems, we have to update the Jacobian matrix at every computational step. This disadvantage may lead to a great computational cost.

A reasonable improvement for this problem is to solve the Equation (4.2) in an approximate way. This is the approach of the so-called inexact Newton iterative method or Newton-like method, which yields a sequence of iterates x_k satisfying

$$\Delta x_k = -[B_k]^{-1} H(x_k) + r_k, \quad x_{k+1} = x_k + \Delta x_k, \ k = 0, 1, 2, \ldots, \tag{1.3}$$

where $\|r_k\| \leq \eta_k \|H(x_k)\|$ with $\eta_k \in [0, 1)$, B_k is equivalent or approximate to the Jacobian matrix $H'(x_k)$. The local convergence rate of the scheme (1.3) can be controlled by the forcing sequence η_k (see e.g. [5–11]). Especially in [12], Jay further proposed an inexact simplified Newton method. The sequence of the iterates is determined as follows:

$$x_{k+1} = x_k - [H'(x_0)]^{-1} H(x_k), \quad k = 0, 1, 2, \ldots, \tag{1.4}$$

[a] School of Mathematics and Statistics, Huazhong University of Science and Technology, Wuhan 430074, China
[b] Wenhua College,Wuhan 430074, China
* Correspondence to: Hongyu Qin, Wenhua College and School of Mathematics and Statistics, Huazhong University of Science and Technology, Wuhan 430074, China.
† E-mail: qinhongyuhust@sina.com

where $H'(x_0)$ can be viewed as an approximation to the Jocabian matrix $H'(x_k)$ if x_0 is close to x_k. When solving nonlinear equations that are derived from the discretization of time-dependent ODEs, we can change the time stepsize to let a starting value x_0 sufficiently close to the exact solution x^*. This can ensure the local convergence of the method.

In practical application, it is noted that space discretization of some PDEs yields some nonlinear equations containing some terms with different stiff properties, which can be better taken advantage of by using split numerical methods (see e.g. [13–15]). Inspired by the split idea, Li and Zhang developed a kind of split Newton iterative method for solving reaction diffusion equations in [16], where the nonlinear equation $H(x)$ is split into a linear stiff term and a nonstiff term. The Jacobian matrix is only determined by the linear stiff term. Then, the method was successfully applied to solve the generalized Burger–Huxley equation [16], some general coupled matrix equation [17], the KdV equation [18] and some biological models [19, 20]. The method saves considerable computational time by reducing the computational cost of the Jacobian matrix. We remark that the split iterative methods are also discussed in recent studies (e.g. [21–27]), where one part is assumed to be continuously Fréchet differentiable and the other is not necessarily differentiable. Therefore, the split idea is different. Our split Newton method is also different from the method proposed in [28, 29], where the waveform relaxation method is applied.

In this study, we investigate the error estimates of the split Newton method, which is studied in [16]. Meanwhile, an improved split Newton method is derived by using the accelerating technique. To exhibit the effectiveness and advantage of the method, we compare it with the previously mentioned classical Newton method, inexact simplified Newton method and the split Newton method. Numerical experiments indicate that our method is convergent and saves a considerable amount of CPU time.

The rest of the paper is organized as follows. In Section 2, we describe the split Newton iterative algorithm and investigate its error analysis. Section 3 is devoted to discussing the improved split Newton method. Section 4 contains an application of the new method to some models. These numerical examples illustrate the good performance of the algorithm. Finally, in Section 5, conclusions and discussions for this paper are summarized.

2. Error estimates

Considering the following nonlinear equations:

$$H(x) := F(x) + G(x) = 0, \quad x \in \mathbb{R}^n, \tag{2.1}$$

where F, G denote the stiff and non-stiff terms, respectively. In [16], Li and Zhang once presented a split Newton iterative algorithm, which generates a sequence of iterates x_k satisfying:

$$x_{k+1} = x_k - [F'(x_k)]^{-1} H(x_k), \;\; k = 0, 1, 2, \ldots \tag{2.2}$$

where the Jacobian matrix $F'(\bar{x}_k)$ is determined by one part of the function $H(x)$.

The convergence criteria of the split Newton iterative method are stated as follows, where $e_k = x_k - x^*$, x^* is the exact solution of Equation (2.1) and $B(x^*; \|e_0\|)$ is an open ball around x^* with radius $\|e_0\|$. Here and in the following, $\|\cdot\|$ denotes the L_2-norm.

Lemma 2.1 (cf. [16])
Assume that $x \in D \subset \mathbb{R}^n$, function $F(x)$ is Fréchet differentiable, its Jacobian matrix $F'(x)$ is invertible for any $x \in D$ and there exist a vector norm $\|\cdot\|$ and constants $\alpha, \beta > 0$, such that the following affine covariant Lipschitz conditions hold:

$$\begin{cases} \|[F'(x)]^{-1}[G(x) - G(\tilde{x})]\| \le \alpha \|x - \tilde{x}\|, \;\; x, \tilde{x} \in D, \\ \|[F'(x)]^{-1}(F'(x) - F'(\tilde{x}))(x - \tilde{x})\| \le \beta \|x - \tilde{x}\|, \;\; x, \tilde{x} \in D, \\ \delta_0 := \; \alpha + \beta\|e_0\|/2 < 1, \;\; B(x^*, \|e_0\|) \subseteq D. \end{cases} \tag{2.3}$$

Then the approximate solution sequence $x_k \in B(x^*, \|e_0\|)$ and converges to x^*.

Based on this lemma, we have the following error estimate for the iterative method (2.2).

Theorem 2.1
Suppose the assumptions in Lemma 2.1 hold. Then, we have the following error estimates,

$$\|x_k - x_{k-1}\| \le \delta_0 \|x_{k-1} - x_{k-2}\|,$$

$$\|x_k - x^*\| \le \frac{1}{1 - \delta_0} \|x_k - x_{k-1}\|,$$

where δ_0 is defined in (2.3).

Proof
Setting $\Delta x_k = x_{k+1} - x_k$, an application of the split Newton iteration for k and $k-1$ yields:

$$\begin{aligned}
\|\Delta x_k\| &= \left\|[F'(x_k)]^{-1} H(x_k)\right\| \\
&= \left\|[F'(x_k)]^{-1}\left(H(x_k) - \left(H(x_{k-1}) + F'(x_{k-1})\Delta x_{k-1}\right)\right)\right\| \\
&= \left\|[F'(x_k)]^{-1}\left(G(x_k) - G(x_{k-1})\right) + [F'(x_k)]^{-1}\left(F(x_k) - \left(F(x_{k-1}) + F'(x_{k-1})\Delta x_{k-1}\right)\right)\right\| \\
&= \left\| [F'(x_k)]^{-1}[G(x_k) - G(x_{k-1})] + [F'(x_k)]^{-1}\int_0^1 \left(F'(x_{k-1} + \Delta x_{k-1} t) - F'(x_{k-1})\right)\Delta x_{k-1} dt \right\|
\end{aligned}$$

Thanks to the conditions (2.3), we arrive at

$$\|\Delta x_k\| \le \alpha\|\Delta x_{k-1}\| + \frac{1}{2}\beta\|\Delta x_{k-1}\|^2 = \delta_{k-1}\|\Delta x_{k-1}\|, \ \ k \ge 1, \tag{2.4}$$

where

$$\delta_{k-1} := \alpha + \frac{1}{2}\beta\|\Delta x_{k-1}\|.$$

With the help of the known condition $0 < \delta_0 < 1$ and (2.4), it can be proved by mathematical induction that the sequence $\|\Delta x_k\|$ and δ_k are contractive.

Hence,

$$\|\Delta x_k\| \le \delta_0\|\Delta x_{k-1}\|, \tag{2.5}$$

and

$$\begin{aligned}
\|x_{k+p} - x_k\| &\le \sum_{i=0}^{p-1}\|x_{k+p-i} - x_{k+p-i-1}\| \le \sum_{i=0}^{p-1}\delta_0^{p-i}\|x_k - x_{k-1}\| \\
&= \frac{1-\delta_0^{p+1}}{1-\delta_0}\|x_k - x_{k-1}\|.
\end{aligned} \tag{2.6}$$

Letting $p \to +\infty$ in (2.6) gives the error estimate. $\square$

3. The improved split Newton iterative algorithm

We firstly recall the accelerating technique in iterative algorithms. Consider the following sequences of iterates x_k in R^1

$$x_{k+1} = \varphi(x_k), \ \ k = 0, 1, 2, \ldots \tag{3.1}$$

where φ is Fréchet-differentiable operator defined on a convex subset D_1 of a Banach space. Let φ' denote the Fréchet derivative of φ evaluated at $x \in D_1$. Suppose that the function $\varphi'(x)$ slightly changes near the root x^* and that $\bar{x}_{k+1} = \varphi(x_k)$. Then, we have

$$x^* - \bar{x}_{k+1} = \varphi(x^*) - \varphi(x_k) = \varphi'(\xi)(x^* - x_k) \approx L(x^* - x_k), \qquad \xi \in (x^*, \bar{x}_{k+1})$$

where we set $\varphi'(\xi) \approx L$. Hence,

$$x^* = \frac{1}{1-L}\bar{x}_{k+1} - \frac{L}{1-L}x_k = \frac{1}{1-L}\varphi(x_k) - \frac{L}{1-L}x_k.$$

As a result, the sequences of iterates x_k can be derived as follows,

$$x_{k+1} = \frac{1}{1-L}\varphi(x_k) - \frac{L}{1-L}x_k, \ \ k = 0, 1, 2, \ldots \tag{3.2}$$

The improved split Newton method is derived from the accelerating technique in iterative algorithms. Applying the accelerating technique to split Newton iterative algorithms gives

$$\begin{aligned} x_{k+1} &= (1-L)^{-1}\left(x_k - [F'(x_k)]^{-1}H(x_k)\right) - L\cdot(1-L)^{-1}x_k \\ &= x_k - \omega[F'(x_k)]^{-1}H(x_k), \end{aligned} \tag{3.3}$$

where $\omega = (1-L)^{-1}$.

When $x \in R^n$, the parameter L can be derived approximately. For example, assume that the stiff term $F(x) = Ax$ is a linear one, where A is a constant matrix, then

$$\varphi'(x) = \left(x - A^{-1}H(x)\right)' = \left(x - A^{-1}(Ax + G(x))\right)' = -A^{-1}G'(x).$$

In practical application, we set $L \approx \| -A^{-1}G'(x_0)\|$ when applying the method to solving some nonlinear equations.

The convergence criteria of the method (3.3) are stated as follows. Where $e_k = x_k - x^*$, x^* is the exact solution of Equation (2.1) and $B(x^*, \|e_0\|)$ is an open ball around x^* with radius $\|e_0\|$.

Theorem 3.1

Assume that $x \in D \subset \mathbb{R}^n$, function $F(x)$ is Fréchet differentiable, its Jacobian matrix $F'(x)$ is invertible for any $x \in D$ and there exist constants α, β and a vector norm $\|\cdot\|$, such that the following affine covariant Lipschitz conditions hold:

$$\begin{cases} \|[F'(x)]^{-1}[G(x) - G(\tilde{x})]\| \le \alpha\|x - \tilde{x}\|, \ \ x, \tilde{x} \in D, \\ \|[F'(x)]^{-1}(F'(x) - F'(\tilde{x}))(x - \tilde{x})\| \le \beta\|x - \tilde{x}\|, \ \ x, \tilde{x} \in D, \\ \sigma_0 := \alpha\omega + \beta\omega\|e_0\|/2 + |1-\omega| < 1, \ \ B(x^*, \|e_0\|) \subseteq D, \end{cases} \tag{3.4}$$

where $\omega \in R^1$ is a parameter. Then the approximate solution sequence $\{x_k\}$ generated by (3.3) is well defined, remains in $B(x^*, \|e_0\|)$ and converges to x^*.

Proof

According to the improved Newton algorithm (3.3) and Equation (2.1), we have

$$\begin{aligned} \|e_{k+1}\| &= \|e_k + \triangle x_k\| \\ &= \left\|e_k - \omega[F'(x_k)]^{-1}[H(x_k) - H(x^*)]\right\| \\ &= \left\|e_k - \omega[F'(x_k)]^{-1}[F(x_k) + G(x_k) - F(x^*) - G(x^*)]\right\| \\ &= \left\| \omega[F'(x_k)]^{-1}[G(x^*) - G(x_k)] + \omega[F'(x_k)]^{-1}[F'(x_k)e_k - F(x_k) + F(x^*)] + (I-\omega)e_k \right\| \\ &= \left\| \omega[F'(x_k)]^{-1}[G(x^*) - G(x_k)] + \omega[F'(x_k)]^{-1}\int_0^1 [F'(x_k) - F'(x_k - te_k)]e_k dt + (I-\omega)e_k \right\| \\ &\le \omega \left\| [F'(x_k)]^{-1}[G(x^*) - G(x_k)] \right\| + \omega \left\| [F'(x_k)]^{-1}\int_0^1 [F'(x_k) - F'(x_k - te_k)]e_k dt \right\| + |(1-\omega)| \left\| e_k \right\|. \end{aligned} \tag{3.5}$$

Substituting (3.4) into (3.5) yields

$$\|e_{k+1}\| \le \left(\alpha\omega + \beta\omega\|e_k\|\int_0^1 t\,dt + |1-\omega|\right)\|e_k\| = \sigma_k\|e_k\|, \quad k \ge 0, \tag{3.6}$$

where

$$\sigma_k = \alpha\|\omega\| + \beta\omega\|e_k\|/2 + |1-\omega|. \tag{3.7}$$

With (3.6), (3.7) and the known condition $0 < \sigma_0 < 1$, it can be proved by mathematical induction that the sequence $\|e_k\|$ and σ_k are mono-decreasing, and thus, $0 < \sigma_k < 1$ for all k. An induction to (3.7) yields that

$$\|e_k\| \le \left(\prod_{i=0}^{k-1}\sigma_i\right)\|e_0\| \le \sigma_0^k\|e_0\|, \quad k \ge 1,$$

which, as well as $0 < \sigma_0 < 1$, implies

$$\lim_{n\to\infty} \|e_k\| = 0,$$

and the sequence x_k belongs to the open ball $B(x^*, \|e_0\|)$. Therefore, the conclusion is proven. □

Specially, if $F(x)$ is a linear function, the Jacobian matrix $F'(x)$ becomes a constant matrix. In this case, we can arrive at the following conclusion.

Theorem 3.2
Assume that $x \in D \subset \mathbb{R}^n$, $F(x) = \tilde{A}x$, where $\tilde{A} \in \mathbb{R}^{n\times n}$ is an invertible constant matrix and that there exist a vector norm $\|\cdot\|$ and constants $\alpha, \omega > 0$, such that

$$\begin{cases} \|\tilde{A}^{-1}[G(x) - G(\tilde{x})]\| \le \alpha\|x - \tilde{x}\|, \ \ x, \tilde{x} \in D \\ \sigma := \alpha\omega + |1-\omega| < 1, \ \ B(x^*, \|e_0\|) \subseteq D. \end{cases} \tag{3.8}$$

Then the approximate solution sequence x_k converges to x^* and satisfies the following error estimate

$$\|x_k - x^*\| \ \le \ \frac{1}{1-\sigma}\|x_k - x_{k-1}\|, \quad k = 1, 2, \ldots. \tag{3.9}$$

Proof
When $F(x)$ is a linear function, the inequality (3.5) can be reduced to

$$\|e_{k+1}\| \le \omega\|\tilde{A}^{-1}[G(x^*) - G(x_k)]\| + |(1-\omega)|\|e_k\|, \ \ k \ge 0. \tag{3.10}$$

Applying condition (3.8) to (3.10) yields

$$\|e_k\| \le \sigma\|e_{k-1}\| \le \sigma^k\|e_0\|, \ \ k \ge 1. \tag{3.11}$$

This, together with $0 < \sigma < 1$, gives

$$\lim_{n\to\infty} \|e_k\| = 0.$$

Hence, the approximate solution sequence x_k converges to x^*.

Also, according to the algorithm 3.3, we have

$$\begin{aligned} \|x_{k+1} - x_k\| &= \left\|x_k - x_{k-1} - \omega\tilde{A}^{-1}[H(x_k) - H(x_{k-1})]\right\| \\ &= \left\|\omega\tilde{A}^{-1}[G(x_k) - G(x_{k-1})] + (1-\omega)(x_k - x_{k-1})\right\| \\ &\le \sigma\|x_k - x_{k-1}\|, \end{aligned} \tag{3.12}$$

which leads to

$$\begin{aligned} \|x_{k+p} - x_k\| &\le \sum_{i=0}^{p-1}\|x_{k+p-i} - x_{k+p-i-1}\| \le \sum_{i=0}^{p-1}\sigma^{p-i}\|x_k - x_{k-1}\| \\ &= \frac{1-\sigma^{p+1}}{1-\sigma}\|x_k - x_{k-1}\|. \end{aligned} \tag{3.13}$$

Letting $p \to +\infty$ in (3.13) yields

$$\|x_k - x^*\| \ \le \ \frac{1}{1-\sigma}\|x_k - x_{k-1}\|,$$

where we have noted that $0 < \sigma < 1$. This completes the proof. □

4. Application

In this section, we will illustrate the effectiveness of the split algorithm. All the computations are performed by using MATLAB (MathWorks, Natick, MA, USA). The convergence tolerance for different Newton methods is 10^{-8}.

Example 1
As a first example, we show some improved numerical results of the improved split Newton method. Consider the following nonlinear equation with function $H : R^1 \longrightarrow R^1$ defined by

$$H(x) = 10x + \sin^2(x) - 3\sin(x) + 5 = 0. \tag{4.1}$$

We set $F(x) = 10x$, $G(x) = \sin^2(x) - 3\sin(x) + 5$ and the initial Gauss $x_0 = 0$. The parameter $L \approx -0.1G'(x_0) \approx 0.3$ is used. The improved split Newton method generates a sequence of iterates x_k satisfying

$$x_{k+1} = x_k - \omega[F'(x_k)]^{-1}H(x_k), \quad k = 0, 1, 2, \ldots, \tag{4.2}$$

where $\omega = \frac{10}{7}$ and $F'(x_k) = 10$. An advantage of our split method is that we do not have to compute $H'(x)$, which is a little complicated.

We compare the improved split Newton iterative algorithm (improved) with the split Newton iterative method (split), the classical Newton method (classical) and the inexact simplified Newton method (simplified) in Table I, where some statistics of the number of iterations with different initial values x_0 and methods are presented. Clearly, the improved method is more effective than the split Newton iterative method. Meanwhile, the improved method is more easy to be implemented (we do not have to compute $H'(x)$), although it may need more number of iterations than the inexact simplified and the classical Newton method.

Example 2
In the second example, we show that the improved split Newton method saves considerable computational time in practical application. Consider the following weakly damped, gKdV equation

$$u_t + u_{yyy} + u^3 u_y + \gamma u = f(y),\ 0 \le t \le t, 0 \le y \le 2\pi, \tag{4.3}$$

with a 2π-periodic boundary condition and the initial condition

$$u(y, 0) = \sin(2y),\ 0 \le y \le 2\pi, \tag{4.4}$$

In [18], Wang *et al.* investigate the change of the attractor according to the change of f and the coefficient γ of the damping term. Here we set $\gamma = 1$, $f(y) = 5\cos(y)$ and use the following finite difference method to discrete gKdV Equation (4.3),

$$\begin{aligned} &\frac{u_j^{n+1} - u_j^n}{k} + \frac{1}{2}\left(\frac{u_{j+2}^{n+1} - 2u_{j+1}^{n+1} + 2u_{j-1}^{n+1} - u_{j-2}^{n+1}}{2h^3} + \frac{u_{j+2}^n - 2u_{j+1}^n + 2u_{j-1}^n - u_{j-2}^n}{2h^3}\right) \\ &\quad + \frac{1}{2}\left(\left(u_j^{n+1}\right)^3 \frac{u_{j+1}^{n+1} - u_{j-1}^{n+1}}{2h} + \left(u_j^n\right)^3 \frac{u_{j+1}^n - u_{j-1}^n}{2h}\right) + \frac{1}{2}\gamma\left(u_j^{n+1} + u_j^n\right) = f(y_j), \end{aligned} \tag{4.5}$$

where $h = 2\pi/N$ is space stepsize, $k = 6/M$ is time stepsize and u_j^n is a numerical approximation to $u(y_j, t_n)$ at $(y_j, t_n) = (jh, nk)$. Writing Equation (4.5) in matrix form, we have

$$\begin{aligned} H(U_{n+1}) = &(I + 0.5kA + 0.5k\gamma I)U_{n+1} + 0.5k\left(diag\left(U_{n+1}^3\right)\right)BU_{n+1} + (0.5kA + 0.5k\gamma - I)U_n \\ &+ 0.5k\left(diag\left(U_n^3\right)\right)BU_n - kf(\tilde{y}) = 0 \end{aligned} \tag{4.6}$$

Table I. Average iterative number for Equation (4.1).

x_0	Split	Improved	Simplified	Classical
0	17	7	7	4
0.5	18	6	12	5
−0.5	16	6	7	4
1	18	7	15	5
−1	16	6	8	4

where I is a unit matrix, $U_n = [u_1^n, u_2^n, \cdots, u_{N-1}^n]'$, $\tilde{y} = [y_1, y_2, \cdots, y_{N-1}]'$, the matrix

$$A = \frac{1}{2h^3}\begin{bmatrix} 0 & -2 & 1 & \cdots & 1 & -2 \\ -2 & 0 & -2 & \cdots & 0 & 1 \\ 1 & -2 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots & \vdots \\ 1 & 0 & 0 & \cdots & 0 & -1 \\ -2 & 1 & 0 & \cdots & -2 & 0 \end{bmatrix} \in \mathbb{R}^{(N-1)\times(N-1)}$$

and

$$B = \frac{1}{2h}\begin{bmatrix} 0 & -1 & \cdots & 0 & 1 \\ 1 & 0 & \cdots & 0 & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & 0 & \cdots & 0 & -1 \\ -1 & 0 & \cdots & 1 & 0 \end{bmatrix} \in \mathbb{R}^{(N-1)\times(N-1)}$$

Now, we let the split functions

$$F(x) = (I + 0.5kA + 0.5k\gamma I)x$$

and

$$G(x) = 0.5k(diag(x^3))Bx + (0.5kA + 0.5k\gamma - I)U_n + 0.5k\left(diag\left(U_n^3\right)\right)BU_n - kf(\tilde{y}).$$

Noting that at every time level $t = t_n$, our split Newton iterative algorithm yields the same Jacobian matrix $(I + 0.5kA + 0.5k\gamma)$. There is no need to update the Jacobian matrix to solve the nonlinear equations all the time. Moreover, we can accelerate our inner linear problems solvers using the following method. Let

$$temp := (I + 0.5kA + 0.5k\gamma I)^{-1} = (I + 0.5kA + 0.5k\gamma I)\backslash I,$$

which produces the solution using Gaussian elimination in MATLAB. Then the improved split Newton iterative method can be shown as follows,

$$x_{k+1} = x_k - \omega \cdot tmep \cdot H(x_k), \;\; k = 0, 1, 2, \ldots. \tag{4.7}$$

In the numerical experiment, we set $L \approx \|temp \cdot G'(U_1)\| \approx 0.0148$ and $\omega = \frac{1}{1-L}$.

In the present paper, we mainly compare the efficiency of the split Newton algorithm with that of the other iterative methods. We show the total computational time (time) from the time interval 0 to 6 and the average iterative number of every timestep (number) in Table II. Although the inner iterative number of our split method may be a little more than that of some classical Newton methods, our split method saves a considerable amount of CPU time. The advantage become more obvious when we choose a smaller stepsize. Specially, the improved split Newton method needs a minimum amount of computational time.

Example 3

In the third example, we apply our split Newton iterative method to the Bursselator system in one spatial variable.

$$\begin{cases} u_t = c + u^2v - (b+1)u + au_{xx} \\ v_t = bu - u^2v + av_{xx} \end{cases}$$

where a, b and c are constant parameters.

Table II. Total computational time and average iterative number for the generalized Korteweg–de Vries equation.

	Method	Split	Improved	Simplified	Classical
N=100, M=600	Time (/s)	2.53	2.49	3.56	8.06
	Number	16.4	16.2	4.92	4.37
N=200, M=1200	Time (/s)	24.6	22.6	51.5	117
	Number	11.3	11.1	4.07	3.83
N=300, M=1800	Time (/s)	87.9	82.3	455	201
	Number	9.68	9.56	3.61	3.73

We let a=0.5, b=1, c=0 and solve the problems on the regions $\{(x,t) : x \in [0,1],\ t \in [0,6]\}$. The initial and boundary conditions are defined using the following exact solution

$$\begin{cases} u(x,t) = \exp(-0.5t - x) \\ v(x,t) = \ \exp(0.5t + x) \end{cases}$$

Space discretization of the systems by finite differences give rise to the following ODEs

$$\begin{cases} (u_i)_t = u_i^2 v_i - 2u_i + \frac{1}{2h^2}(u_{i-1} - 2u_i + u_{i+1}) \\ (v_i)_t = u_i - u_i^2 v_i + \frac{1}{2h^2}(v_{i-1} - 2v_i + v_{i+1}) \end{cases}$$

where h=1/N. Now, we applied an implicit Euler method with the stepsize k=6/M and only one Richardson iteration per split Newton iterative to solve the nonlinear ODEs. The Jacobian matrix of our split Newton iterative is of the form $I - kD$, where

$$D = \frac{1}{h^2}\begin{bmatrix} -2 & 1 & \cdots & 0 & 0 \\ 1 & -2 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & \cdots & -2 & 1 \\ 0 & 0 & \cdots & 1 & -2 \end{bmatrix}_{(N-1)\times(N-1)}$$

Similar to the second example, our split Newton iterative method is effective and convergent. We also give some statistics for the Brussselator system using different iterative methods in Table III. Note the fact that different iterative methods own nearly the same iterative number, our split algorithm need a half or less time. This is especially true for our new method with the parameter $\omega = 1.356$.

Example 4
In the last example, we are interested in applying the split Newton iterative method to the following ODE system

$$Y' = AY + F(Y,t), \qquad 0 \le t \le 6. \tag{4.8}$$

with zero initial values, where $Y = [y_1(t), y_2(t)]'$,

$$A = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix}, \qquad F(Y,t) = \begin{pmatrix} y_2(t) \\ \sin(y_1(t)) \end{pmatrix} + \begin{pmatrix} t\sin(5t) \\ -20t^2\cos(100t) \end{pmatrix}.$$

Noting that matrix A is skew-symmetric (all the eigenvalues of A are pure imaginary numbers), there will be oscillating solution components.

Now, applying the trapezoidal rule to discrete Equation (4.8), we have

$$H(Y_{n+1}) = \left(I - \frac{h}{2}A\right)Y_{n+1} - \frac{h}{2}F(Y_{n+1}, t_{n+1}) - \frac{h}{2}AY_n - \frac{h}{2}F(Y_n, t_n) - Y_n = 0, \tag{4.9}$$

where I is a unit matrix, h is the stepsize and Y_n is the numerical approximation to $Y(t_n)$. The Jacobian matrix of our split Newton iterative is of the form $I - \frac{h}{2}A$. The numerical solutions, which are derived at different times by the method, can be seen in Figure 1. Clearly, there are two oscillating components, and our split iterative method is effective. In order to illustrate the advantage of the proposed method, we present the total computational time and the average iterative number of different iterative methods in Table IV. As shown, the split Newton iterative method need the least computational time. The advantage becomes obvious when we chose a smaller stepsize.

Table III. Total CPU time and average iterative number.

	method	$\omega = 1$	$\omega = 1.356$	Inexact	Classical
N=100, M=200	Time (/s)	1.89	8.91E-1	3.30	6.50
	Number	24.1	15.7	24.2	24.2
N=200, M=400	Time (/s)	5.72	3.61	13.3	16.1
	Number	23.3	15.1	23.3	23.3
N=400, M=800	Time (/s)	166	108	285	516
	Number	22.4	14.7	22.4	22.4

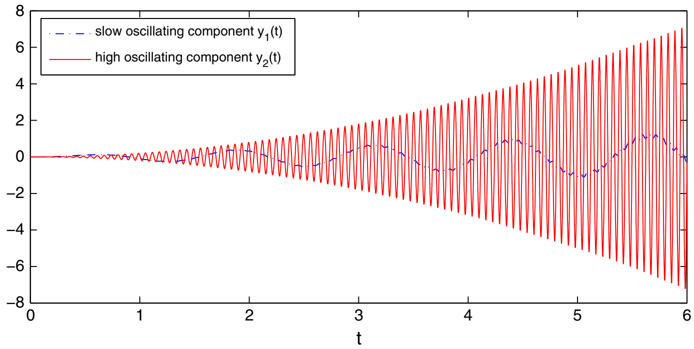

Figure 1. The numerical solutions to Problem (4.8) with stepsize $h = 0.001$.

Table IV. Total CPU time and average iterative number.

H	Method	Split	Inexact	Classical
0.01	Time (/s)	1.56E-1	1.25E-1	1.29E-1
	Number	4.80	3.71	3.94
0.001	Time (/s)	6.56E-1	8.44E-1	8.44E-1
	Number	2.92	2.90	2.90
0.0001	Time (/s)	4.88	8.08	8.13
	Number	2.92	2.90	2.90
0.00001	Time (/s)	39.8	68.0	68.6
	Number	2.30	2.38	2.38

5. Conclusions

In this study, a kind of parameterized split Newton method is derived by using the accelerating technique. The key point is to split the nonlinear equation into a linear stiff term and a nonlinear nonstiff term. The advantage of the split Newton iterative is that we do not have to update the Jacobian matrix at different time intervals. In view of computational CPU time in practical application, the proposed method can give a better result. Meanwhile, convergence and error estimates of the method are derived. Therefore, the method is a good candidate to solve a class of problems effectively.

References

1. Deuflhard P. *Newton Method for Nonlinear Problems: Affine Invariance and Adaptive Algorithms*. Springer Press: Berlin, 2004.
2. Knoll DA, Keyes DE. Jacobian-free Newton–Krylov methods: a survey of approaches and applications. *Journal of Computational Physics* 2004; **193**:357–397.
3. Yamamoto T. A unified derivation of several error bounds for Newton's process. *Journal of Computational and Applied Mathematics* 1985; **12-13**: 179–191.
4. Ypma TJ. Historical development of the Newton–Raphson method. *SIAM Review* 1995; **37**:521–551.
5. Dembo RS, Eisenstat SC, Steihaug T. Inexact Newton methods. *SIAM Journal on Numerical Analysis* 1982; **19**:400–408.
6. Morini B. Convergence behaviour of inexact Newton methods. *Mathematics of Computation* 1999; **68**:1605–1613.
7. Pawlowski RP, Simonis JP, Walker HF, Shadid JN. Inexact Newton dogleg methods. *SIAM Journal on Numerical Analysis* 2008; **46**:2112–2132.
8. Ypma TJ. Local convergence of inexact Newton methods. *SIAM Journal on Numerical Analysis* 1984; **21**:583–590.
9. Zhou G, Qi L. On the convergence of an inexact Newton-type method. *Operations Research Letters* 2006; **34**:647–652.
10. Shadid JN, Tuminaro RS, Walker HF. An inexact Newton method for fully coupled solution of the Navier–Stokes equations with heat and mass transport. *Journal of Computational Physics* 1997; **137**:155–185.
11. Śmietański MJ. Convergence of a generalized Newton and an inexact generalized Newton algorithms for solving nonlinear equations with nondifferentiable terms. *Numerical Algorithms* 2009; **50**:401–415.
12. Jay LO. Inexact simplified Newton iterations for implicit Runge–Kutta methods. *SIAM Journal on Numerical Analysis* 2000; **38**:1369–1388.
13. Ascher UM, Ruuth SJ, Wetton B. Implicit-explicit methods for time dependent partial differential equations. *SIAM Journal on Numerical Analysis* 1995; **32**:797–823.
14. Li D, Zhang C, Wang W, Zhang Y. Implicit-explicit predictor-corrector schemes for nonlinear parabolic differential equations. *Applied Mathematical Modelling* 2011; **35**:2711–2722.
15. Hundsdorfer W, Ruuth SJ. IMEX extensions of linear multistep methods with general monotonicity and boundedness properties. *Journal of Computational Physics* 2007; **225**:2016–2042.
16. Li D, Zhang C, Split Newton iterative algorithm and application its. *Applied Mathematics and Computation* 2010; **217**:2260–2265.
17. Xie L, Yang H, Liu Y, Ding F. Iterative solutions for general coupled matrix equations with real coefficients. *American Control Conference* 2011:669–674.

18. Wang M, Li D, Zhang C, Tang Y. Long time behavior of solutions of mKdV equations. *Journal of Mathematical Analysis and Applications* 2012; **390**: 136–150.
19. Li D, Zhang C, Wang W. Long time behavior of non-Fickian delay reaction–diffusion equations. *Nonlinear Analysis: RWA* 2012; **13**:1401–1405.
20. Li D, Tong C, Wen J. Stability of exact and discrete energy for non-Fickian reaction–diffusion equations with a variable delay. *Abstract and Applied Analysis* 2014,Article ID 840573.
21. Vijesh VA, Subrahmanyam PV. A Newton-type method and its applications. *International Journal of Mathematics and Mathematical Sciences* 2006, Article ID 23674.
22. Vijesh VA, Subrahmanyam PV. A Newton-like method and its application. *Journal of Mathematical Analysis and Applications* 2008; **339**:1231–1242.
23. Argyros IK, Hilout S. On the convergence of Newton-type methods under mild differentiability conditions. *Numerical Algorithms* 2009; **52**:701–726.
24. Argyros IK, Hilout S. Newton–Kantorovich approximations under weak continuity conditions. *Journal of Applied Mathematics and Computing* 2011; **37**:361–375.
25. Argyros IK, Hilout S. Majorizing sequences for iterative methods. *Journal of Computational and Applied Mathematics* 2012; **236**:1947–1960.
26. Shakhno S, Yarmola H. Two-step method for solving nonlinear equations with nondifferentiable operators. *J. Numer. Appl. Math* 2012; **3**:105–115.
27. Sahu DR, Singh KK, Singh VK. A Newton-like method for generalized operator equations in Banach spaces. *Numerical Algorithms* 2014; **3**:289–303.
28. Wu S, Huang C, Liu Y. Newton waveform relaxation method for solving algebraic nonlinear equations. *Applied Mathematics and Computation* 2008; **201**:553–560.
29. Wu S, Huang C. Two-step relaxation newton method for nonsymmetric algebraic Riccati equations arising from transport theory. *Mathematical Problems in Engineering* 2009,Article ID 783920.