Applied Mathematics, 2010, 1, 222-229
doi:10.4236/am.2010.13027 Published Online September 2010 (http://www.SciRP.org/journal/am)

Solving Large Scale Nonlinear Equations by a New ODE Numerical Integration Method

Tianmin Han[1], **Yuhuan Han**[2]
[1]*China Electric Power Research Institute, Beijing, China*
[2]*Hedge Fund of America, San Jose, USA*
E-mail: *han_tianmin@yahoo.com.cn*, *ibmer.ibm@gmail.com*, *william.han@eMallGuide.com*

Received June 11, 2010; *revised July* 23, 2010; *accepted July* 26, 2010

Abstract

In this paper a new ODE numerical integration method was successfully applied to solving nonlinear equations. The method is of same simplicity as fixed point iteration, but the efficiency has been significantly improved, so it is especially suitable for large scale systems. For Brown's equations, an existing article reported that when the dimension of the equation N = 40, the subroutines they used could not give a solution, as compared with our method, we can easily solve this equation even when N = 100. Other two large equations have the dimension of N = 1000, all the existing available methods have great difficulties to handle them, however, our method proposed in this paper can deal with those tough equations without any difficulties. The sigularity and choosing initial values problems were also mentioned in this paper.

Keywords: Nonlinear Equations, Ordinary Differential Equations, Numerical Integration, Fixed Point Iteration, Newton's Method, Stiff, Ill-Conditioned

1. Introduction

The classic methods for solving nonlinear equations $F(X) = 0$ mainly have the following two types:

1) Fixed Point Iteration:

$$X_{n+1} = G(X_n) \quad \text{here } G(X) = F(X) + X \tag{1}$$

2) Newton Iteration:

$$X_{n+1} = X_n - J(X_n)^{-1} F(X_n) \quad \text{here } J(X) \text{ is the Jacobian of } F(X) \tag{2}$$

As the book [1] Pg. 17 described, the solution of the nonlinear system $F(X) = 0$ can be interpreted as steady states or equilibrium point of the dynamic system $\dot{X} = F(X)$ In fact, those two iterations are all equivalent to explicit Euler method in the field of ODE numerical integration.

For the differential equation:

$$\dot{X} = F(X) \tag{3}$$

The Euler method:

$$X_{n+1} = X_n + hF(X_n) \tag{4}$$

is a general expression of fixed point iteration [1] Pg.299

If we take $h = 1$, we can get (1)

As for Newton iteration, for the differential equation:

$$\dot{X} = -J(X)^{-1} F(X) \tag{5}$$

using explicit Euler method:

$$X_{n+1} = X_n - hJ(X)^{-1} F(X) \tag{6}$$

and taking $h = 1$ we get (2)

These relations can also be found in [2] Pg.768 or [3] §7.5.

We developed a set of numerical integration method in [4]. They have accuracy 1st-5th order. Among them, the simplest one is the 1st order PEC scheme. This scheme has very large stable region, so we can take it as a tool to integrate the differential equation and get fast convergence speed to solve $F(X) = 0$.

For the sake of completeness, we rederive the algorithm in the next section.

2. Algorithm

Consider the problem:

$$\begin{cases} \dot{X} = F(X) \\ X(0) = X_0 \end{cases} \tag{7}$$

Using implicit Euler method:

$$X_{n+1} = X_n + hF\left(X_{n+1}\right) \tag{8}$$

introducing variable

$$Z_{n+1} = hF\left(X_{n+1}\right) \tag{9}$$

we have

$$Z_{n+1} = X_{n+1} - X_n \tag{10}$$

Multiplying both sides of (9) by ε (ε > 0), we obtain

$$\varepsilon Z_{n+1} = \varepsilon hF\left(X_{n+1}\right) \tag{11}$$

Equation (11) can be reformulated as follows:

$$\left(\varepsilon/h\right) Z_{n+1} = \varepsilon F\left(X_{n+1}\right) \tag{12}$$

Equation (12) plus Equation (10), we obtain

$$\left(1+\varepsilon/h\right) Z_{n+1} = \varepsilon F\left(X_{n+1}\right) + X_{n+1} - X_n \tag{13}$$

Let

$$\omega = h/\left(h+\varepsilon\right) \tag{14}$$

Equation (13) can be rewritten as

$$Z_{n+1} = \omega\left(\varepsilon F\left(X_{n+1}\right) + X_{n+1} - X_n\right) \tag{15}$$

From (8) and (9), we have

$$X_{n+1} = X_n + Z_{n+1} \tag{16}$$

Combining (15) and (16), we obtain a new implicit integration method, which is fully equivalent to (8).

We use the simple iteration method to solve the implicit system (15) and (16), and choose the initial iteration value $X_{n+1}^{(0)} = X_n + Z_n$. Only one iteration applies to the implicit system (15) and (16), then we obtain an explicit integration scheme as follows:

$$\begin{cases} X_{n+1}^{(0)} = X_n + Z_n \\ Z_{n+1} = \omega\left(\varepsilon F\left(X_{n+1}^{(0)}\right) + Z_n\right) \\ X_{n+1} = X_n + Z_{n+1} \end{cases} \tag{17}$$

(17) is named as the EPS method in this article.

In order to investigate the stability of the EPS method (17), we consider the model equation

$$\dot{x} = \lambda x \tag{18}$$

where λ is a complex number. Then, we have

$$\begin{cases} x_{n+1} = x_n + z_{n+1} \\ z_{n+1} = \omega\varepsilon\lambda\left(x_n + z_n\right) + \omega z_n \end{cases} \tag{19}$$

or the matrix form

$$\begin{bmatrix} x_{n+1} \\ z_{n+1} \end{bmatrix} = \begin{bmatrix} 1+\omega\varepsilon\lambda & \omega+\omega\varepsilon\lambda \\ \omega\varepsilon\lambda & \omega+\omega\varepsilon\lambda \end{bmatrix} \begin{bmatrix} x_n \\ z_n \end{bmatrix} \tag{20}$$

The characteristic equation of (20) is given by

$$\mu^2 - \left(1+\omega+2\omega\varepsilon\lambda\right)\mu + \omega\left(1+\varepsilon\lambda\right) = 0 \tag{21}$$

Let

$$\varepsilon = \alpha h \tag{22}$$

From (21), we obtain

$$h\lambda = \frac{\mu^2 - \left(1+\omega\right)\mu + \omega}{\left(2\mu-1\right)\omega\alpha} \tag{23}$$

Giving α a special value, let μ vary and keep $|\mu| =$ 1, then we obtain an enclosed curve, which is just the boundary of the absolute stability region in $h\lambda$-plane. Set $\mu = \cos\theta + j\sin\theta$, $j = \sqrt{-1}$, $0 \le \theta \le 2\pi$, then we rewrite (23) as follows:

$$\mathrm{Re}\left(h\lambda\right) = -\Big(\left(1+\omega-2\sin^2\theta-\cos\theta-\omega\cos\theta\right) \left(1-2\cos\theta\right) - 2\left(2\cos\theta-1-\omega\right)\sin^2\theta\Big)\Big/\Big(\left(5-4\cos\theta\right)\alpha\omega\Big) \tag{24}$$

$$\mathrm{Im}\left(h\lambda\right) = -\Big(2\left(1+\omega-2\sin^2\theta-\cos\theta-\omega\cos\theta\right)\sin\theta + \left(2\cos\theta-1-\omega\right)\sin\theta\left(1-2\cos\theta\right)\Big)\Big/\Big(\left(5-4\cos\theta\right)\alpha\omega\Big) \tag{25}$$

The curve of the boundary of the absolute stability region is obtained when θ varies from 0 to 2π. If α is a small number, the stability region will be close to real axis and spreads far away towards the left-half plane. For example, when α = 0.01, as it is shown in **Figure 1**, the left end point of the stability region can reach −134, so the integration step size can be increased significantly.

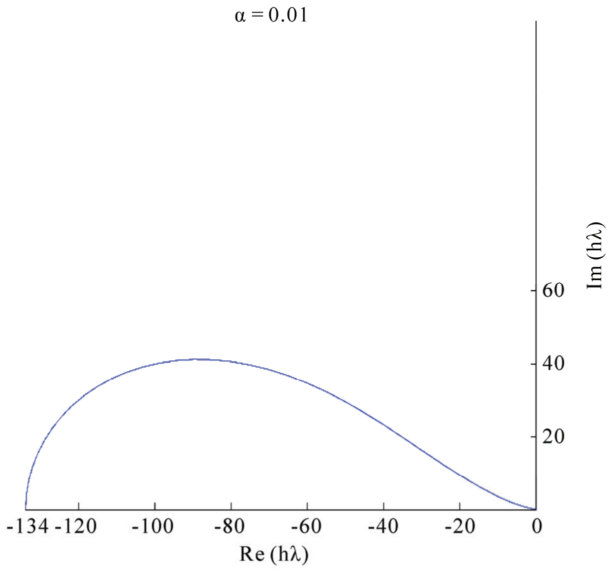

Figure 1. The absolute stability region of the EPS method for *α* = 0.01.

In the model Equation (18), if λ is very close to the imaginary axis, *i.e.*, $Im(\lambda) \gg Re(\lambda)$, α should be taken a bigger value. For $\alpha = 100$, the stability region is shown by **Figure 4**. We can find that the region includes a section of the imaginary axis. This property is unusual for an explicit method.

When $\alpha = 1$, *i.e.*, $h = \varepsilon$, then the stability region of the EPS method is all the same as the explicit Euler method. It is enclosed by a circle with center at (−1, 0) and its radius is 1. In fact, in (24) and (25) taking $\alpha = 1$, then $\omega = 0.5$, we have

$$\begin{aligned}\mathrm{Re}(h\lambda) &= -\Big(\big(1.5-2\sin^2\theta-1.5\cos\theta\big)\big(1-2\cos\theta\big) \\ &\quad -4\cos\theta\sin^2\theta+3\sin^2\theta\Big)\Big/\big(2.5-2\cos\theta\big) \quad (26)\\ &= \cos\theta-1\end{aligned}$$

$$\begin{aligned}\mathrm{Im}(h\lambda) &= -\Big(\big(3-4\sin^2\theta-3\cos\theta\big)\sin\theta \\ &\quad +\big(5\cos\theta-1.5-4\cos^2\theta\big)\sin\theta\Big)\Big/\big(2.5-2\cos\theta\big) \\ &= \sin\theta\end{aligned}$$

(27)

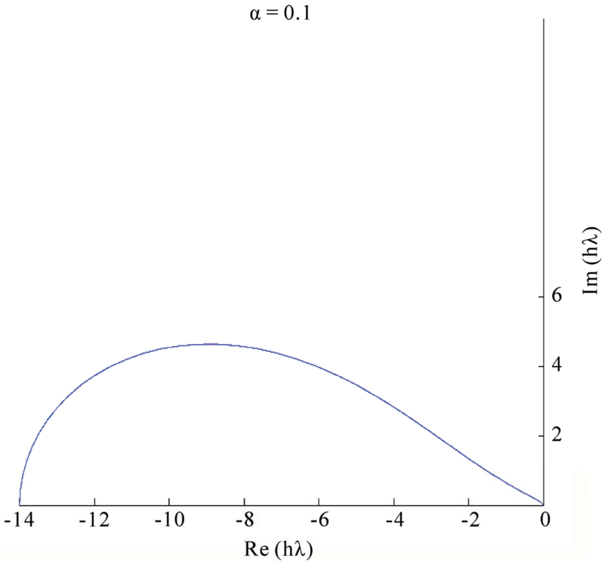

Figure 2. The absolute stability region of the EPS method for *α* = 0.1.

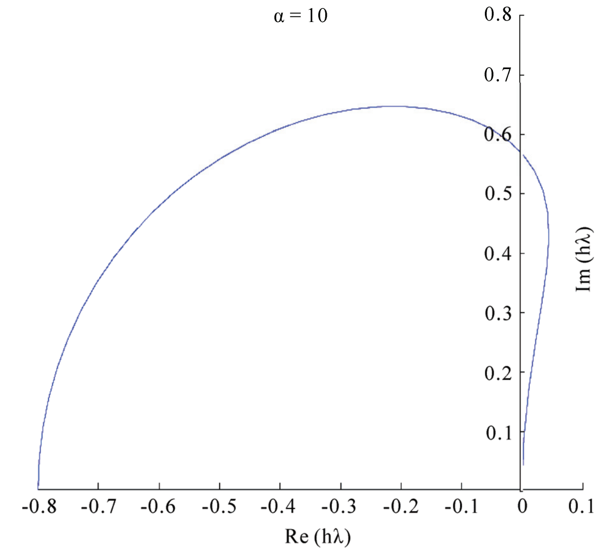

Figure 3. The absolute stability region of the EPS method for *α* = 10.

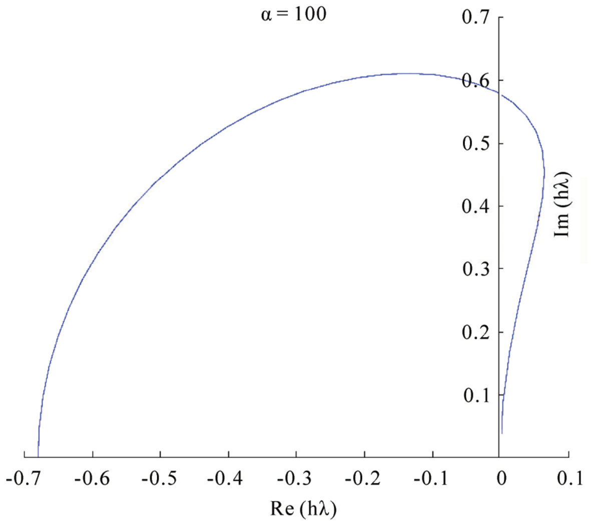

Figure 4. The absolute stability region of the EPS method for *α* = 100.

3. Implementation of the Algorithm

In this article, we merely discuss how to use the EPS method to integrate the differential equation $\dot{X} = F(X)$. Usually ODE integration methods require the condition $\partial F/\partial X < 0$ holds. That is to say the eigenvalues of the Jacobian distribute in the left-half part of the complex plane. For our purpose, to solve $F(x) = 0$ and to solve $-F(x) = 0$ are equivalent. In other field the "half plane condition" is always said to be "positive definite", *i.e.*, the eigenvalues are in the right plane. This fact reminds us the differential equation to deal with is $\dot{X} = -F(X)$.

The EPS method can also be applied to the differential equation

$$\dot{X} = -J(X)^{-1} F(X) \qquad (28)$$

In this case, if $F(X)$ is replaced with $-F(X)$ in (28), it does not change the form of (28). So the sign in front of $F(X)$ is meaningless at all. By the way, choosing $\varepsilon = h = 1$, according to many numerical experiments have done by us, the numerical results of EPS are almost the same as the numerical results of the Newton's method (the details are not given in this article).

Despite the EPS method is a Jacobian-Free method, if it is not difficult to obtain the diagonal matrix *D*(*X*) of the Jacobian *J*(*X*), then we can integrate differential equation

$$\dot{X} = -D(X)^{-1} F(X) = -G(X)$$

we can get even much better results, especially, when *J*(*X*) is a diagonal dominant matrix. However, it needs to consider a strategy to avoid overflow when some elements of the matrix *D*(*X*) are very small.

At present we have not developed a adaptive program which can automatically choose parameter ε and the step size h, but we give a strategy roughly as follows.

For non-stiff system, we pick up the parameter ε on [0.5, 1.0] and determine h by the size of $\|F(X)\|$. For stiff system, we need to estimate the spectral radius ρ of the Jacobian matrix *J*(*X*) such that $\varepsilon\rho < 1$ is satisfied. In fact, if λ is a positive real number, for $\dot{x} = -\lambda x$, when $\varepsilon\lambda \le 4/3$, we can prove that the scheme (19) is stable for all h ($0 < h < \infty$). Small value ε can strengthen stability but will reduce the efficiency.

For some easy problems we can take fixed step size in the whole calculating process. Usually we divide the calculating process into three stages, in each stage, different step size will be taken.

To do this, we set three parameters TOL_1, TOL_2, TOL_3. At first, we choose step size h_1 to start the calculation till $\|F\| < TOL_1$ is satisfied, the first stage is completed. Taking current value of *X* as initial value, we start the second stage calculation with step size h_2 till $\|F\| < TOL_2$. Do the same as we have done till finally $\|F\| < TOL_3$, then we end our calculation. In this paper, the $\|\,.\,\|$ means Euclidean norm.

Outline of the Algorithm

Step 1. Give an initial value X_0. Set ε, h and compute $\omega = h/(h+\varepsilon)$.

Step 2. Compute $F(X)$ and $D(X)$ if it is needed.

Step 3. Compute $G(X) = D(X)^{-1} F(X)$ or $G(X) = F(X)$. If an element $d_i(X)$ of matrix $D(X)$ is less than one, the division is omitted and we have $g_i(X) = f_i(X)$.

Step 4. Compute $Z_0 = hG(X_0)$.

Step 5. $X := X + Z$.

Step 6. Compute $F(X)$, $D(X)$, $G(X)$ by the way of Step 2 and Step 3.

Step 7. If $F(X) < TOL$, then stop, else do

$$X := X - Z$$

$$Z := \omega(\varepsilon G(X) + Z)$$

$$X := X + Z$$

Go to Step 5.

4. Numerical Experiments

We now present numerical results for five examples. Some of them have already had results in literature. So we can compare our results with theirs. We also compare the results of fixed point iteration (explicit Euler method) with ours as well. This is because we identify our method as an improvement for the fixed point iteration and the explicit Euler method was well represented in all explicit methods.

4.1. Example 1 [2]

$$f_1(X) = x_1^2 - x_2 + 1$$

$$f_2(X) = x_1 - \cos\left(\frac{\pi}{2}x_2\right)$$

The initial value x_0 = (1, 0). The solution we want to seek is x^* = (0, 1). The Jacobian of the system is:

$$J(X) = \begin{bmatrix} 2x_1 & -1 \\ 1 & \frac{\pi}{2}\sin\left(\frac{\pi}{2}x_2\right) \end{bmatrix}$$

and the determinant of the Jacobian is given by

$$\det(J(X)) = x_1\pi\sin\left(\frac{\pi}{2}x_2\right) + 1$$

So at the line $\sin\left(\frac{\pi}{2}x_2\right) = -\frac{1}{\pi x_1}$, the singularity occurs.

Newton method does not converge to x^* but rather, it crosses the singularity line and converges in eight iterations to $x^* = \left(-\frac{1}{2}\sqrt{2}, \frac{3}{2}\right)$.

The damped Newton method was also applied to this problem and it converged to x^* in 107 iterations. The total number of function evaluations is as many as 321.

In [2], there are 12 algorithms, all of them are based on trapezoid formula, have been tested for this example. Among them the PE_BCE_B is the best, here the E_B means using Broyden method to approximate J^{-1}. The iteration is 17 times and the evaluation is 36 times.

There are four algorithms, each of them needs iterate more than 100 times. The rest seven algorithms need to iterate 23~47 times and evaluate 68~282 times respectively. All those calculations use double precision.

This example was considered as a difficult problem, because the differential equation to deal with is $\dot{X} = -J(X)^{-1} F(X)$ and the Jacobian is singular.

If the differential equation to be handled is $\dot{X} = -F(X)$, all the trouble will disappear. In fact it is a non-stiff equation, we can reach the equilibrium point easily. In our calculation the single precision was used. For the sake of comparing with [2], we take $|f_1| < 10^{-5}$, $|f_2| < 10^{-5}$ as convergence criteria.

Explicit Euler method and EPS method were tested for this example. For Euler method, taking step size $h = 0.24$, 0.25, 0.26, 0.27, 0.28, the results show $h = 0.28$ overflow happened. The numbers of function evaluation for other step size were 74,72,72,84 respectively. The best result was given by $h = 0.25$: $x_1 = .1879 \times 10^{-4}$, $x_2 = .1000 \times 10^1$, $f_1 = -.6020 \times 10^{-5}$, $f_2 = -.9568 \times 10^{-5}$. EPS method: take $\varepsilon = 1$ and $h = 0.4$, 0.5, 0.6, overflow happened at $h = 0.6$. For $h = 0.4$, 37 times evaluation was needed. The best result was give by $h = 0.5$: the numbers of evaluation is 31, $x_1 = .2418 \times 10^{-5}$, $x^2 = .1000 \times 10^1$, $f_1 = -.5305 \times 10^{-5}$, $f_2 = .7182 \times 10^{-5}$.

4.2. Example 2

We construct a large scale mild stiff system to test our method. For $F(X) = A(X) - b = 0$, the differential equation is $\dot{X} = -F(X) = -(A(X) - b)$. Here b is a constant vector and $A(X) = UDUC(X)$, $C(X) = (x_1^3, x_2^3, \ldots, x_N^3)^T$, $U = \left(I - \frac{2}{u^T u} uu^T\right)\left(U = U^T = U^{-1}\right)$, $u = (1, 1, \ldots, 1)^T$, $N = 1000$. D is block diagonal matrix: $D = diag(D_i)$ and $D_i = \begin{bmatrix} a_i & b_i \\ c_i & d_i \end{bmatrix}$. The solution of $F(X) = 0$ is $X^* = (1, 1, \ldots, 1)^T$. For this X^*, when the value of D_i was given, the value of b can be calculated, and the equation can be entirely determined.

The characteristic of the system depends on the choice of D_i. In the following three types of the D are given, the initial values for all of them are $X_0 = (0, 0, \ldots, 0)^T$. In this problem, using the special form of U and D the function $F(X)$ can be easily computed, but to compute the Jacobian is no longer an easy task, we must compute every element of the matrix. Furthermore, the Jacobian is a dense matrix and the Newton method will lose all superiority for this large scale system.

Example 2.1 Take D as a diagonal matrix, *i.e.*, in D_i we put $b_i = c_i = 0$, $a_i = 2_i - 1$, $d_i = 2_i$, $i = 1, 2, \ldots, N/2$. The results of EPS method and explicit Euler method are listed in the **Tables 1**, **2**. Here the NFE is the abbreviation of Number of Function Evaluations and the "Step Size" means the best step size, the highest efficiency was reached by this step size.

Compare **Tables 1** and **2**, for EPS Method in three stages the Step Size h_1, h_2, h_3 have the relation $h_3 = 2h_2 = 2(2h_1) = 4h_1$, but for Euler Method h_1, h_2, h_3 almost keep a constant. The ratio of NFE is $1244/12003 \approx 0.1$

Example 2.2 The subblocks of D have the following form:

$$D_i = \begin{bmatrix} 2i & i \\ -i & 2i \end{bmatrix}, \ i = 1, 2, \ldots, N/2$$

i.e., the eigenvalues of matrix D distribute in a wedge region. The results of both methods are listed in **Table 3** and **Table 4**.

The ratio of NFE is $2219/8014 \approx 0.28$

Example 2.3 $D_i = \begin{bmatrix} 1 & i/100 \\ -i/100 & 1 \end{bmatrix}, \ i = 1, 2, \ldots, N/2$.

The eigenvalues of D distribute in a line. The line is parallel with imaginary axis. The maximum ratio of imaginary part and real part is 5:1. The results of both methods are listed in the **Tables 5** and **6**.

The ratio of NFE is $499/1379 \approx 0.36$. We did not give the Jacobian of F, but according to the situation of matrix

Table 1. EPS method ε = 0.0004; $\|F(X_0)\|$ = .1827 × 10^5.

Tolrance	Step Size	NFE	$\|F\|$
TOL1 = 1D - 0	0.0025	119	0.99×10^0
TOL2 = 1D - 5	0.005	669	0.9810×10^{-5}
TOL3 = 1D - 10	0.01	1244	0.9610×10^{-10}

Table 2. Euler method $\|F(X_0)\|$ = .1827 × 10^5.

Tolrance	Step Size	NFE	$\|F\|$
TOL1 = 1D - 0	0.00055	597	0.9980×10^0
TOL2 = 1D - 5	0.00066	6099	0.9985×10^{-5}
TOL3 = 1D - 10	0.0066	12003	0.9685×10^{-10}

Table 3. EPS method ε = 0.00025, $\|F(X_0)\|$ = .2044 × 10^5.

Tolrance	Step Size	NFE	$\|F\|$
TOL1 = 1D - 0	0.001	273	0.9967×10^0
TOL2 = 1D - 5	0.002	1165	0.9890×10^{-5}
TOL3 = 1D - 10	0.004	2219	0.9977×10^{-10}

Table 4. Euler method $\|F(X_0)\|$ = .2044 × 10^5.

Tolrance	Step Size	NFE	$\|F\|$
TOL1 = 1D - 0	0.00044	636	0.9987×10^0
TOL2 = 1D - 5	0.000528	4223	0.9971×10^{-5}
TOL3 = 1D - 10	0.000528	8014	0.9860×10^{-10}

D, we can get a general conception for the distribution of eigenvalues of the Jacobian. Compare three cases above, we can conclude that if the eigenvalues are close to real axis the EPS method will be more efficient.

4.3. Example 3

Brown's Almost Linear Function

$$f_i(X) = X_i + \sum_{j=1}^{N} X_j - (N+1), i = 1, 2, \ldots, N-1$$

$$f_N(X) = \prod_{j=1}^{N} X_j - 1$$

the initial values are $X_i(0) = 0.5$. The solution to be searched is $X^* = (1, 1, \ldots, 1)^T$. This is a difficult problem. Brown in [5] reported his research work. For $N = 5$ Newton method converged to the root given approximately by (−.579, −.579, −.579, −.579, 8.90); however, for $N = 10, 30$ Newton method diverged quite rapidly.

Brown's method did an excellent work, for $N = 5, 10, 30$, after 6, 7, 9 times iteration they all converged to X^* For $N = 10, 30, 40$ the authors of [6] tested their elaborate subroutines *NEQ*1 and *NEQ*2 for this tough problem, unfortunately the test failed for $N = 40$. Let us take a look at the differential equation:

$$\dot{X} = -F(X)$$

The last row of th Jacobian is $\partial f_N(X)/\partial X_i = \prod_{j=1, j\neq i}^{N} X_j$.

When *N* is large enough, at the neighborhood of the initial point this row almost equals zero vector, so the equation is considered a very stiff or ill-conditioned system for large *N*.

The differential equation virtually to deal with is:

$$\dot{X} = -D(X)^{-1} F(X)$$

The diagonal matrix $D(X)$ has elements $d_i(X) = 2.0$, $i = 1, 2, \ldots, N-1$ and $d_N(X) = \prod_{j=1}^{N-1} X_j$. If the value of $d_N(X)$ is very small, the measures must be taken to avoid overflow (for the details see paragraph 3). As we mentioned before, we divided the calculation into three stages and took different step size for each stage. For $N = 10, 30, 40, 100$, the results of EPS method were listed in **Tables 7-10**.

Explicit Euler method (fixed point iteration) can also get the results, but the expense was very expensive. The change in step size is very small in different stage. We use:

$$N(h_1, h_2, h_3)M$$

to express the dimension of the equation, three different

Table 5. EPS method $\varepsilon = 0.1$, $\|F(X_0)\| = .8396 \times 10^2$.

Tolrance	Step Size	NFE	$\|F\|$
TOL1 = 1D - 0	0.01	217	0.9762×10^{0}
TOL2 = 1D - 5	0.02	401	0.9580×10^{-5}
TOL3 = 1D - 10	0.04	499	0.9789×10^{-10}

Table 6. Euler method $\|F(X_0)\| = :8396 \times 10^2$.

Tolrance	Step Size	NFE	$\|F\|$
TOL1 = 1D - 0	0.011	255	0.9894×10^{0}
TOL2 = 1D - 5	0.0132	790	0.9935×10^{-5}
TOL3 = 1D - 10	0.0132	1379	0.9835×10^{-10}

Table 7. $N = 10$, $\varepsilon = \frac{2}{10}$, $\|F(X_0)\| = .1653 \times 10^2$.

Tolrance	Step Size	NFE	$\|F\|$
TOL1 = 1D - 0	0.65	5	0.4991×10^{0}
TOL2 = 1D - 5	1.0	35	0.2714×10^{-5}
TOL3 = 1D - 10	1.2	119	0.8864×10^{-10}

Table 8. $N = 30$, $\varepsilon = \frac{2}{30}$, $\|F(X_0)\| = .8348 \times 10^2$.

Tolrance	Step Size	NFE	$\|F\|$
TOL1 = 1D - 0	0.3	6	0.3737×10^{0}
TOL2 = 1D - 5	0.9	61	0.4548×10^{-5}
TOL3 = 1D - 10	1.2	277	0.5100×10^{-10}

Table 9. $N = 40$, $\varepsilon = \frac{2}{40}$, $\|F(X_0)\| = .1280 \times 10^3$.

Tolrance	Step Size	NFE	$\|F\|$
TOL1 = 1D - 0	0.2	6	0.6212×10^{0}
TOL2 = 1D - 5	0.6	41	0.5133×10^{-5}
TOL3 = 1D - 10	1.2	293	0.1592×10^{-10}

Table 10. $N = 100$, $\varepsilon = \frac{2}{100}$, $\|F(X_0)\| = .5025 \times 10^3$.

Tolrance	Step Size	NFE	$\|F\|$
TOL1 = 1D - 0	0.1	7	0.2985×10^{0}
TOL2 = 1D - 5	0.3	57	0.5787×10^{-5}
TOL3 = 1D - 10	1.2	640	0.4534×10^{-10}

step size and the NFE. The results are as follows: 10 (0.2, 0.25, 0,3) 788, 30 (0.11, 0.11, 0.112) 4586, 40 (0.09, 0.09, 0.09) 7540, 100 (0.035, 0.035, 0.035) 42183.

The ratio of NFE for both methods are: 119/788 ≈ 0.15, 277/4586 ≈ 0.06, 293/7540 ≈ 0.04, 640/42183 ≈ 0.015. From the data above we can see that along with N increasing the ill-conditioned extent is becoming more severe and the superiority of EPS method compared with Euler method is even more obvious.

The evaluation of functions is main calculation in both methods. Despite EPS method needs some extra expenses, this part is relatively very small. For $N = 100$, as we listed above, the ratio of NFE for both methods is approximately 0.015. Even if the extra expense is added, as a conservative estimate, the work amount of EPS method does not reach 2% of Euler's.

We have no intention for $N = 10$, 30 to compare the NFE with NEQ1 and NEQ2 in [6]. It is because that the main expense in those two subroutines is solving linear equations, the expense for evaluation of functions only takes small part of the total.

4.4. Example 4

Two-Point Boundary Value Problem [7] P.80 For two-point boundary value problem

$$u^{n}\left(t\right)=\frac{1}{2}\left(u\left(t\right)+t+1\right)^{3},0<t<1,u\left(0\right)=u\left(1\right)=0$$

we apply the standard O (h^2) discretization then we can get the following nonlinear equations:

$$f_i\left(X\right)=-x_{i-1}+2x_i-x_{i+1}+\frac{1}{2}h^2\left(x_i+t_i+1\right)^3$$

$1\le i\le n$, taking $n = 10$, $x_0 = x_{n+1} = 0$, $t_i = ih$, $h=\dfrac{1}{n+1}$

It is well known that the initial values play an important role in the procedure of solving a nonlinear equations. As in [6,7] did, set standard starting vector x_s, which regarded as being close to the solution, then using x_s, $10x_s$, $100x_s$ as initial values to test the algorithm. Usually for most algorithm when $x_0 = x_s$ the test got success, when $x_0 = 100x_s$ the test failed.

Four algorithms with three initials x_s, $10x_s$, $100x_s$ were tested in [7], here $x_s=\left(\xi_1,\xi_2,\dots,\xi_n\right)^T$ and $\xi_j=t_j\left(t_j-1\right)$, $1\le j\le n$.

Relatively speaking, this is a simple problem, every algorithm with any initial value had no trouble to get the solution. Same thing happened for EPS method, for each case mentioned above we get the solution without any trouble. For the sake of comparison, we take $\max\limits_{1\le i\le n}\left|f_i\right|<10^{-15}$ as convergence criteria. We integrate differential equation:

$$\dot{X}=-D\left(X\right)^{-1}F\left(X\right)$$

and take $D = diag\ (2, 2, \dots, 2)$, $\varepsilon = 0.5$, $h = 2.0$ For initial value $X_0 = X_s$, $X_0 = 10X_s$, $X_0 = 100X_s$, the NEFs are 197, 237, 259 respectively. As compared with Newton method the corresponding figures are 34, 45, 100 [7]. Our goal is not to compare those two sets of figure, as we said before, the main work for Newton method is to solve the linear equations. What we want to do is to compare the following figures: 237/197 ≈ 1.2, 259/197 ≈ 1.3, 45/34 ≈ 1.3, 100/34 ≈ 2.9 Those figures mean that when initial value varied from $X_0 = X_s$ to $X_0 = 100X_s$ EPS method only increases work amount 30%, but for Newton method the work amount will increase 190%.

Another thing is worth mentioning here. For $X_0 = 100X_s$, using tridiagonal solver, after 10 times iteration Newton method got the result max $|f_i| < .2776 \times 10^{-16}$, ($1 \le i \le 10$). However, if at the starting stage using EPS method with $\varepsilon = 0.5$, $h = 1.6$, to make max $|f_i| < 1.0$, only 11 times function evaluation is needed. At the moment, taking current *X's* values as initial value and using Newton method merely 4 times iteration the almost same result was obtained. This fact shows that if the initial value is regarded as being far away from the solution, then EPS method can be chosen as a tool to improve it.

By the way, for Euler method the best step size is 0.9, the numbers of function evaluation for $X_0 = X_sX_0 = 10X_s$, $X_0 = 100X_s$ are 609, 685, 705, respectively.

4.5. Example 5

Broyden tridiagonal function [6] p.28

$$f_i\left(X\right)=-x_{i-1}+\left(3-2x_i\right)x_i-2x_{i+1}+1$$

where $x_0 = x_{n+1} = 0$ *and* $i = 1, 2, \dots, N$; $N = 1000$ the diagonal elements of Jacobian are $3 - 4x_i$ $X_s = (-1, -1, \dots, -1)^T$ In [6] $N = 10$, the numbers of function evaluation for NEQ1 and NEQ2 are 23 and 25.

We use Euler method and EPS method to integrate differential equation

$$\dot{X}=-D\left(X\right)^{-1}F\left(X\right)$$

The initial values are $X_0 = X_sX_0 = 10X_s$, $X_0 = 100X_s$. The results are almost the same for both methods. It can be found in **Table 11**.

From the results above, when the initial value $x_i(0)$ taking "negative" values, despite $X_0 = 100X_s$ is regarded as being far away form the solution, however, every method carried out smoothly. But when $x_i(0)$ taking "positive" values the situation would be totally different.

We tested Newton method, taking $\|F\| < 10^{-10}$ as convergence criteria. The results are as follows: for $x_i(0) =$

Table 11. The results of Euler and EPS methods.

Method	Tolrance	Step Size	NFE	$\|F\|$
Euler	$X_0 = X_s$	1.0	41	$.7468 \times 10^{-10}$
EPS	$X_0 = X_s$	1.0 ($\varepsilon = h$)	41	$.7468 \times 10^{-10}$
Euler	$X_0 = 10X_s$	0.5	108	$.9335 \times 10^{-10}$
EPS	$X_0 = 10X_s$	0.5 ($\varepsilon = h$)	108	$.9334 \times 10^{-10}$
Euler	$X_0 = 100X_s$	0.5	117	$.7573 \times 10^{-10}$
EPS	$X_0 = 100X_s$	0.5 ($\varepsilon = h$)	117	$.7572 \times 10^{-10}$

0.0, 0.001, 0.002, 0.003, 0.004, 0.005, 0.006, 0.007, 0.008, 0.009, 0.01, 0.015, 0.016, 0.017, 0.018, the numbers of iteration are 16, 19, 24, 29, 35, 41, 47, 53, 59, 65, 72, 105, 112, 120, 127. When x_i (0) = 0.019 overflow happened.

For EPS method, taking the same convergence criteria, choosing $h = \varepsilon = 1.0$ and x_i (0) = 0.0, 0.5, 0.7, the numbers of function evaluation are merely 42, 43, 45. Overflow happened if x_i (0) = 0.8. This result may be expected because at the very beginning if x_i (0) = 0.75 the diagonal elements of the Jacobian are all equal to zero.

5. References

[1] P. Deuflhard, “Newton Methods for Nonlinear Problems Affine Invariance and Adaptive Algorithms,” Springer-Verlag, Berlin, Heidelberg, 2004.

[2] P. T. Boggs, “The Solution of Nonlinear Systems of Equations by A-Stable Integration Technique,” *SIAM Journal on Numerical Analysis*, Vol. 8, No. 4, 1971, pp. 767-785.

[3] J. M. Ortega and W. C. Rheinboldt, “Iterative Solution of Nonlinear Equations in Several Variables,” Academic Press, New York, 1970.

[4] T. M. Han and Y. H. Han, “Solving Implicit Equations Arising from Adams-Moulton Methods,” *BIT*, Vol. 42, No. 2, 2002, pp. 336-350.

[5] K. M. Brown, “Computer Oriented Algorithms for Solving Systems of Simultaneous Nonlinear Algebraic Equations,” In: G. D. Byrne and C. A. Hall, Eds., *Numerical Solution of Systems of Nonlinear Algebraic Equations*, Academic Press, New York, 1973, pp. 281-348.

[6] J. J. Moré, B. S. Garbow and K. E. Hillstrom, “Testing Unconstrained Optimization Software,” *ACM Transactions on Mathematical Software*, Vol. 7, No. 1, 1981, pp. 17-41.

[7] J. J. Moré and M. Y. Cosnard, “Numerical Solution of Nonlinear Equations,” *ACM Transactions on Mathematical Software*, Vol. 5, No. 1, 1979, pp. 64-85.