Scientific
Research

Solving Large Scale Unconstrained Minimization Problems by a New ODE Numerical Integration Method

Tianmin Han[1], Xinlong Luo[2], Yuhuan Han[3]
[1]*China Electric Power Research Institute, Beijing, China*
[2]*School of Information and Communication Engineering, Beijing University of Posts and Telecommunications, Beijing, China*
[3]*Hedge Fund of America, San Jose, USA*
E-mail: han_tianmin@yahoo.com.cn, luoxinlong@gmail.com, ibmer.ibm@gmail.com

Abstract

In reference [1], for large scale nonlinear equations $F(X) = 0$, a new ODE solving method was given. This paper is a continuous work. Here $F(X)$ has gradient structure *i.e.* $F(X) = -\nabla f(X)$, $f(X)$ is a scalar function. The eigenvalues of the Jacobian of $F(X)$, or the Hessian of $f(X)$, are all real number. So the new method is very suitable for this structure. For quadratic function the convergence was proved and the spectral radius of iteration matrix was given and compared with traditional method. Examples show for large scale problems (dimension $N = 100, 1000, 10000$) the new method is very efficient.

1. Introduction

This work is a continuation of [1]. In [1] we solved a general nonlinear equations by a new ODE method. The numerical results were very encouraging. For instance, a very tough problem: Brown's equation with the dimension $N = 100$ can be solved easily by the new method.

In this paper we turn our attention to a special function $F(X) = -\nabla f(X)$, the ODE

$$\dot{X} = -\nabla f(X) \tag{1}$$

is said to have a gradient structure. This structure comes from seeking a local minimizer in the optimization area: to seek a point X^* such that $f(X^*) \leq f(X)$ for all X in some neighbourhood of X^*, here $X = (x_1, x_2, \cdots, x_N)^{\mathrm{T}}$ is a vector, $f(X)$ is a scalar function, $\nabla f(X) = \left(\dfrac{\partial f}{\partial x_1}, \dfrac{\partial f}{\partial x_2}, \cdots, \dfrac{\partial f}{\partial x_N} \right)$.

It is well known that the conditions $\nabla f(X^*) = 0$ and $\nabla^2 f(X^*)$ positive semidefinite are necessary for X^* to be a local minimizer, while the conditions $\nabla f(X^*) = 0$ and $\nabla^2 f(X^*)$ positive definite are sufficient.

$\nabla^2 f(X)$ is symmetric and it is called Hessian matrix (or Hessian, for short). In term of ODE numerical integration $-\nabla^2 f(X)$ is called Jacobian of the right function $-\nabla f(X)$.

For a symmetric matrix, the eigenvalues are all real numbers. If the Hessian is positive definite, that means the eigenvalues of $-\nabla^2 f(X)$ are all negative real numbers. That is to say the Jacobian of differential Equation (1) possesses negative real eigenvalues, so the new ODE method in [1] is very suitable for this case. (see the stability region Figure 1 and Figure 2 in [1]).

2. The Method

In [1] for initial problem:

$$\begin{cases} \dot{X} = F(X) \\ X(0) = X_0 \end{cases} \tag{2}$$

a new ODE integration method is as follows:

$$\begin{cases} X_{n+1}^{(0)} = X_n + Z_n \\ Z_{n+1} = \omega \left(\varepsilon F(X_{n+1}^{(0)}) + Z_n \right) \\ X_{n+1} = X_n + Z_{n+1} \end{cases} \tag{3}$$

here $\omega = h/(h+\varepsilon)$. $\varepsilon > 0$ is a parameter, h is the step size. From the initial value X_0, calculating $Z_0 = hF(X_0)$, then for $n = 1, 2, \cdots$, we can get sequence $X_1, X_2, \cdots$. This method is based on the implicit Euler method, however, it is entirely different from linearized implicit Euler method:

$$X_{n+1} = X_n + h\left(I - hF'(X_n)\right)^{-1} F(X_n) \qquad (4)$$

([2] p. 300). The above method is a linear combination of Newton method and fixed iteration. In optimization area the adaptive linearized implicit Euler method is identical to a updating trust region algorithm ([3] p. 205).

Despite Hessian does not appear in the method (3), however in the process to determine the parameters ε, h, it plays an important role.

3. The Choice of Parameters and the Rate of Convergence

Because in the neighbourhood of minimal point, the ordinary nonlinear functions approximately emerge quadratic function properties, so our discussion is carried out for a quadratic function:

$$q_n(\delta) = f(X_n) + \nabla f(X_n)^{\mathrm{T}} \delta + \frac{1}{2}\delta^T \nabla^2 f(X_n)\delta \quad (5)$$

minimization problem.

This point of view leads to the following topic:

What is the good choice for parameters ε and h to solve the linear equations:

$$-\nabla^2 f(X_n)\delta = \nabla f(X_n) \qquad (6)$$

In order to simplify the writing symbol, we put $A = \nabla^2 f(X_n)$, $b = -\nabla f(X_n)$, $\delta = X$. The Equation (6) now is turned into (7):

$$-AX + b = 0 \qquad (7)$$

here A is a symmetric positive definite matrix and b is a vector.

For $2N$ order vector $\left(X_n^{\mathrm{T}}, Z_n^{\mathrm{T}}\right)^{\mathrm{T}}$, the $2N \times 2N$ iteration matrix M of the method (3) can be expressed by

$$M = \begin{bmatrix} I - \omega\varepsilon A & \omega I - \omega\varepsilon A \\ -\omega\varepsilon A & \omega I - \omega\varepsilon A \end{bmatrix} \qquad (8)$$

Let the eigenvalue of M be μ and the eigenvector be $w = \left(u^{\mathrm{T}}, v^{\mathrm{T}}\right)^{\mathrm{T}}$. We have

$$\begin{bmatrix} I - \omega\varepsilon A & \omega I - \omega\varepsilon A \\ -\omega\varepsilon A & \omega I - \omega\varepsilon A \end{bmatrix}\begin{bmatrix} u \\ v \end{bmatrix} = \mu\begin{bmatrix} u \\ v \end{bmatrix} \qquad (9)$$

From (9) we can easily get

$$v = \frac{\mu-1}{\mu}u$$

and

$$\omega\varepsilon\left(1 + \frac{\mu-1}{\mu}\right)Au = \left(1 - \mu + \omega\frac{\mu-1}{\mu}\right)u \qquad (10)$$

Equality (10) shows u is the eigenvector of matrix A, let the corresponding eigenvalues be λ, *i.e.*

$$Au = \lambda u$$

then λ satisfies

$$\omega\varepsilon(2\mu-1)\lambda = \mu - \mu^2 + \omega\mu - \omega \qquad (11)$$

or

$$\mu^2 - (1 + \omega - 2\omega\varepsilon\lambda)\mu + \omega(1 - \varepsilon\lambda) = 0 \qquad (12)$$

From (12) we can see that for every eigenvalue λ of A there are two eigenvalues $\mu_{1,2}$ of M. Convergence means $|\mu_{1,2}| < 1$. We use the following lemma ([4] p. 799) to prove the convergence:

If b and c are real, then the norm of the roots for the quadratic equation

$$\lambda^2 - b\lambda + c = 0$$

is less than unity if and only if

$$|c| < 1, \quad |b| < 1 + c$$

In our case $c = \omega(1 - \varepsilon\lambda), 0 < \omega < 1, \varepsilon\lambda > 0, b = 1 + \omega - 2\omega\varepsilon\lambda$. If $1 + \omega - 2\omega\varepsilon\lambda > 0$, we have the further result $1 + \omega - \omega\varepsilon\lambda > 0$ and $|b| = 1 + \omega - 2\omega\varepsilon\lambda < 1 + \omega - \omega\varepsilon\lambda = 1 + \omega(1 - \varepsilon\lambda) = 1 + c$.

If $1 + \omega - 2\omega\varepsilon\lambda < 0$, choosing ε such that $0 < \varepsilon\lambda < \frac{4}{3}$, we have

$$-2 - 2\omega < -4\omega < -3\omega\varepsilon\lambda$$

Rewriting the above inequality in the form

$$-1 - \omega + 2\omega\varepsilon\lambda < 1 + \omega - \omega\varepsilon\lambda$$

then

$$|1 + \omega - 2\omega\varepsilon\lambda| < 1 + \omega - \omega\varepsilon\lambda = 1 + c$$

We also get $|b| < 1 + c$.

Thus we complete the proof of the convergence, for any $h, 0 < h < \infty$, if only $\varepsilon\lambda < 4/3$. $\left(|c| < 1\right)$.

The rate of convergence for the iteration scheme (3) is determined by $S(M)$, the spectral radius of matrix M which is defined by

$$S(M) = \max_{1 \le i \le N}|\mu_{1,2}(\lambda_i)|$$

From (12)

$$\mu_{1,2}(\lambda_i) = \left(\alpha_i \pm \sqrt{\Delta_i}\right)\big/2$$

here

$$\alpha_i = 1 + \omega - 2\omega\varepsilon\lambda_i, \quad \Delta_i = \alpha_i^2 - 4\omega(1 - \varepsilon\lambda_i)$$
$$i = 1, 2, \cdots, N$$

　　　　　　　　　　　　　　　　　　　　　　　AM

We take a guess that

$$S(M) = |\mu_1(\lambda_1)|$$

$\left(\mu_1(\lambda_1) = \dfrac{1}{2}\left(\alpha_1 + \sqrt{\Delta_1}\right)\right)$. Though we cannot prove it, we consider this guess to be very reasonable.

In fact the differential equation

$$\dot{X} = -AX + b$$

has the solution

$$X(t) = e^{-At}\left(X(0) - X^*\right) + X^*$$

$X^* = A^{-1}b$. X^* is the exact solution of the linear equations:

$$AX = b$$

The term $e^{-At}\left(X(0) - X^*\right)$ is just a linear combination of the terms $e^{-\lambda_i t}$. As $t \to \infty$ all $e^{-\lambda_i t} \to 0$, but $e^{-\lambda_1 t} \geq e^{-\lambda_i t}$ $(i = 2, 3, \cdots, N)$. This means the rate of $e^{-\lambda_1 t}$ approaching zero is the slowest among all the $e^{-\lambda_i t}$. Since μ_1 is determined by λ_1, it must be the largest one (by modulus) for all the $\mu_{1,2}(\lambda_i)$.

In the following we discuss how to choose $\omega(or, h)$ making μ_1 reach its minimum. Reviewing the process of proof for the convergence, we take $\varepsilon < \dfrac{4}{3}\dfrac{1}{\lambda_N}$, provided λ_1 and λ_N satisfy $\dfrac{\lambda_1}{\lambda_N} < \dfrac{3}{4}$, we have $(1 - \varepsilon\lambda_1) > 0$, If $\dfrac{\lambda_1}{\lambda_N} < \dfrac{3}{4}$ is not satisfied, we can choose an even smaller ε to make sure that $(1 - \varepsilon\lambda_1) > 0$.

The problem we are now investigating is, for a fixed $\varepsilon > 0$ which satisfies $(1 - \varepsilon\lambda_1) > 0$, how to choose $\omega(or, h)$ making the $S(M)$ to be the minimum.

From

$$\Delta_1 = \alpha_1^2 - 4\omega(1 - \varepsilon\lambda_1) = (1 - \omega)^2 + 4\omega^2\varepsilon^2\lambda_1^2 - 4\omega^2\varepsilon\lambda_1$$

if $\omega \ll 1$ (or $h \ll 1$), we have $\Delta_1 > 0$, then $\mu_1(\lambda_1)$, $\mu_2(\lambda_1)$ are real and $S(M) = \mu_1 > 0$.

In the Equation (12), we consider μ_1 to be a function of ω and differentiate it:

$$\frac{d\mu_1}{d\omega} = \frac{(1 - 2\varepsilon\lambda_1)\mu_1 - (1 - \varepsilon\lambda_1)}{2\mu_1 - (1 + \omega - 2\omega\varepsilon\lambda_1)} = \frac{(1 - 2\varepsilon\lambda_1)\mu_1 - (1 - \varepsilon\lambda_1)}{\sqrt{\Delta_1}} < 0 \tag{13}$$

(notice: $\mu_1 = \dfrac{1}{2}\left(\alpha_1 + \sqrt{\Delta_1}\right)$), $0 < \mu_1 < 1, (1 - 2\varepsilon\lambda_1) < (1 - \varepsilon\lambda_1)$ (13) shows that μ_1 is a decreasing function of the ω.

We now observe the situation in which Δ_1 varies with ω. $\Delta_1(\omega)$ is a 2nd degree polynomial in ω:

$$\Delta_1(\omega) = \left(1 - 4\varepsilon\lambda_1 + 4\varepsilon^2\lambda_1^2\right)\omega^2 - 2\omega + 1 \tag{14}$$

From $\Delta_1(0) = 1 > 0$, $\Delta_1(1) = -4\varepsilon\lambda_1(1 - \varepsilon\lambda_1) < 0$, we know $\Delta_1(\omega)$ has one and only one root in the interval $(0,1)$.

We solve Equation (14) $\Delta_1(\omega) = 0$ and get the roots:

$$\begin{aligned}\omega^* &= \frac{2 \pm \sqrt{4 - 4\left(1 - 4\varepsilon\lambda_1 + 4\varepsilon^2\lambda_1^2\right)}}{2\left(1 - 4\varepsilon\lambda_1 + 4\varepsilon^2\lambda_1^2\right)} \\ &= \frac{1 \pm \sqrt{4\varepsilon\lambda_1 - 4\varepsilon^2\lambda_1^2}}{1 - \left(\sqrt{4\varepsilon\lambda_1 - 4\varepsilon^2\lambda_1^2}\right)^2} = \frac{1}{1 + \sqrt{4\varepsilon\lambda_1 - 4\varepsilon^2\lambda_1^2}}\end{aligned} \tag{15}$$

(In the numerator we take the sign "$-$", otherwise ω^* will be greater than 1)

After $\Delta_1(\omega^*) = 0$, we consider the situation of $\Delta_1 < 0$. The Equation (12) has a pair of conjugate complex roots with modulus $|\mu_1(\lambda_1)| = |\mu_2(\lambda_1)| = \sqrt{\omega(1 - \varepsilon\lambda_1)}$, which increase with increasing ω. Therefore $|\mu_1|$ reaches its minimum when $\omega = \omega^*$, the corresponding value of $S(M)$ is denoted by S^* and we have

$$S^* = \sqrt{w^*(1 - \varepsilon\lambda_1)} \tag{16}$$

If the system is ill-conditioned, *i.e.* $\lambda_N \gg \lambda_1$, in order to compare our method with traditional methods, omit the factor $(1 - \varepsilon\lambda_1)$ in (16) and $-4\varepsilon^2\lambda_1^2$ " in (15), then we have approximately

$$\begin{aligned}S^{*2} &= \frac{1}{1 + \sqrt{4\varepsilon\lambda_1}} = \frac{1}{1 + \sqrt{\dfrac{16}{3}\dfrac{\lambda_1}{\lambda_N}}} \approx \frac{1}{1 + 2.3\sqrt{\dfrac{\lambda_1}{\lambda_N}}} \\ &= \frac{\sqrt{\lambda_N}}{\sqrt{\lambda_N} + 2.3\sqrt{\lambda_1}} \approx \left(\frac{\sqrt{\lambda_N}}{\sqrt{\lambda_N} + 1.15\sqrt{\lambda_1}}\right)^2 \\ &\approx \left(\frac{\sqrt{\lambda_N} - 0.575\sqrt{\lambda_1}}{\sqrt{\lambda_N} + 0.575\sqrt{\lambda_1}}\right)^2\end{aligned} \tag{17}$$

(notice: if $b \ll a$ then $\dfrac{a}{a + 2b} \approx \dfrac{a - b}{a + b}$).

For $Ax = b$ the Richardson iteration ([5], p. 107)

$$X_{n+1} = X_n + \alpha(b - AX_n)$$

If taking optimal $\alpha = 2/(\lambda_N + \lambda_1)$ then

$$S^* = \frac{\lambda_N - \lambda_1}{\lambda_N + \lambda_1} \tag{18}$$

From the definition of ω and (15)

$$\omega^* = \frac{h}{h + \varepsilon} = \frac{1}{1 + \varepsilon/h} = \frac{1}{1 + \sqrt{4\varepsilon\lambda_1 - 4\varepsilon^2\lambda_1^2}} \tag{19}$$

We have

$$h = \frac{\varepsilon}{\sqrt{4\varepsilon\lambda_1 - 4\varepsilon^2\lambda_1^2}} \qquad (20)$$

If the system is ill-conditioned, we can simplify (20) as

$$h = \frac{\varepsilon}{\sqrt{4\varepsilon\lambda_1}} = \frac{\sqrt{\varepsilon}}{2\sqrt{\lambda_1}} \qquad (21)$$

In order to check the analysis in this paragraph, we construct a 2-dimension linear equation

$$AX = b \qquad (22)$$

The entries:

$$a_{11} = 1.5c - 0.5d, \quad a_{12} = -0.6c + 0.6d$$

$$a_{21} = 1.25c - 1.25d, \quad a_{22} = -0.5c + 1.5d$$

The exact solution $X^* = (1,1)^{\mathrm{T}}$, so $b(1) = a_{11} + a_{12}$, $b(2) = a_{21} + a_{22}$. We take $X(0) = (0.5, 0.5)^{\mathrm{T}}$.

The matrix A has eigenvalues c and d. In our test we put $c = \lambda_2 = 1.0$, $d = \lambda_1 = 10^{-\beta}$, $\beta = 3, 4, 5, 6$. *i.e.* the systems have condition number 10^3, 10^4, 10^5, 10^6.

In our test problem we known the exact solution X^* so we use $\left|X^*(1) - X(1)_n\right| < 10^{-10}$ and $\left|X^*(2) - X(2)_n\right| < 10^{-10}$ as a stopping condition.

The methods to compare are Richardson's iteration and our method, it was named EPS. S_1 and S_2 denote the spectral radius of iteration matrix for each method respectively. As usual we use NFE stands for the Number of Function Evaluation. The expected value of the ratio of NFE for these two methods should be $\ln S2 / \ln S1$. The Richardson iteration is entirely equivalent to the Euler method, so we replace Richardson parameter α by step size h in **Table 1**.

From **Table 1** we can see that the ratio of NFE for theoretical expected values are basically consistent with the real calculation results and the higher the condition number is, the more efficient for our method EPS.

There is another thing need to mention. For $\lambda_1 = 10^{-6}$, EPS method taking $h = 570.0877$, is it possible for so large step size? In [1] Figure 1 and Figure 2 give the ab-

solute stability region for $\alpha = 0.01$ and $\alpha = 0.1$ $(\varepsilon = \alpha h)$. Their left end points are -134 and -14. In our present case $\varepsilon = 1.3$, $h = 570.0877$, $\alpha = 0.00228$, using the same method we can get the left end point of the stability region at -585.5. The Max $\left|\lambda_i\right| = 1.0$, $i = 1, 2$, even though we take so large step size, $h\lambda_i$ are still located at the stability region.

4. Numerical Experiment

The outline of our algorithm EPS is the same as described in [1]. The differential equations to solve are

$$\dot{X} = -\nabla f(X) = -G(X) \qquad (23)$$

$$\text{or } \dot{X} = D^{-1}\left(-\nabla f(X)\right) = -D^{-1}G(X) \qquad (24)$$

Usually we like using (24), especially when $\nabla^2 f(X)$ is a diagonal dominant matrix. In this case we simply take $\varepsilon = 0.5$

For ODE (1), it is said to have a gradient structure. By the chain rule, we have ([3], p. 194)

$$\frac{\mathrm{d}}{\mathrm{d}t} f(X(t)) = \sum_{i=1}^{N} \frac{\partial f}{\partial x_i} \frac{\mathrm{d}x_i}{\mathrm{d}t} = -\sum_{i=1}^{N} \left(\frac{\partial f}{\partial x}\right)^2 = -\left\|\nabla f(x(t))\right\|^2 \qquad (25)$$

From (25) we see that along any analytic solution of the ODE, the quantity $f(X(t))$ decreases in Euclidean norm as t increases.

Different case happens with the present method. Because in our method, we take very large step size, it will produce large local error, the numerical solution X_n may go far from the analytic solution $X(t_n)$, so we cannot keep the condition $f(X_{n+1}) < f(X_n)$, especially at the beginning of the calculation.

In some earlier literatures, for example, [6]; using $\left|f(X^*) - f(X_n)\right| < TOL$ as convergence criteria, but this rule just applies to the test problems which the X^* was known already. For real problems this criterion is not suitable, so we take $\left\|G(X_n)\right\| < TOL$ as our stopping criteria.

Example 1. The Generalized Rosenbrock Function (GENROSE) ([7], p. 421)

Table 1. 2-Dimension test problem results $(\lambda_2 = \lambda_N = 1.0)$, $\varepsilon = 1.3$.

λ_1	Step size h		NFE		Ratio of NFE	
	Richardson	EPS	Richardson	EPS	Expect	Test
10^{-3}	1.998002	18.0278	11057	667	18.1851	16.5722
10^{-4}	1.999800	57.0088	110517	2071	57.5003	53.3641
10^{-5}	1.999980	180.2776	1102675	6433	181.8312	171.4091
$^*10^{-6}$	1.999998	570.0877	5144987	9094	575.0001	565.7562

*Note: for $\lambda_1 = 10^{-6}$ the Richardson iteration cannot get any result, so we replace 10^{-10} by 10^{-5} for this case.

$$f(X) = 1 + f_0(X)$$

$$f_0(X) = \sum_{i=2}^{N} \left[100\left(x_i - x_{i-1}^2\right)^2 + \left(1 - x_{i-1}\right)^2 \right]$$

$$X(0) = (-1.2, 1, -1.2, 1, 1, 1, 1, \cdots, 1)$$

$X_U^* = (1, 1, 1, \cdots, 1)$, X_U^* stands for the optimal solution of unconstrained problem.

The Gradient function:

$$G(1) = -400x_1\left(x_2 - x_1^2\right) - 2(1 - x_1)$$

$$G(i) = 200\left(x_i - x_{i-1}^2\right) - 400x_i\left(x_{i+1} - x_i^2\right) - 2(1 - x_i),$$
$$i = 2, 3, \cdots, N-1$$

$$G(N) = 200\left(x_N - x_{N-1}^2\right)$$

The Diagonal of Gradient function

$$D(1) = 1200x_1^2 - 400x_2 + 2$$

$$D(i) = 1200x_i^2 - 400x_{i+1} + 202 \quad i = 2, 3, \cdots, N-1.$$

$$D(N) = 200$$

As we did in [1], we divided the calculation process into three stages and took $\varepsilon = 0.5, ToL1 = 1.0, ToL2 = 10^{-3}, ToL3 = 10^{-5}$; $h_1 = 1.0$, $h_2 = 2.5$, $h_3 = 5.0$. For $N = 100, 1000, 10000$, we have the same results:

$$NFE = 228, \quad f = 1 + f_0$$

$$f_0 = 0.1755 \times 10^{-12}, \|G\| = 0.8771 \times 10^{-5}$$

$$\underset{1 \le i \le N}{\text{Max}} \left| x_i^* - \tilde{x}_i \right| = 0.1174 \times 10^{-7}.$$

For large scale problem ([8], pp. 14-15) proposed a sub-space trust region method STIR. For example 1 the method gave two results, one was STIR with exact Newton steps, another was STIR with inexact Newton steps. The number of iterations were 25, 21 respectively. These results showed that STIR need to solve 25 or 21 large scale linear equations [8]. Compare with our method EPS, we just need 228 times gradient function evaluation, and no linear equations were needed to solve.

It is well known that ordinary Newton method is very sensitive for the initial value. For this example $(N = 100, 1000, 10000)$ all the $\|G(X(0))\| = 0.1054 \times 10^4$. Ordinary Newton method to be converged needs 375 iterations and get the $\|G(\tilde{x}^*)\| = 0.7389 \times 10^{-6}$.

In order to improve the initial value, we use EPS method making $\|G(\tilde{X})\| < 1$ (after 33 function evaluations), then turns to Newton method. For the present initial value $\tilde{X}$ to get the convergent result needs 229 Newton iterations. If we further improve the initial value, making $\|G(\tilde{X})\| < 0.5$ (after 128 function evaluations) only two

Newton steps we can get $\|G(\tilde{X}^*)\| = 0.3878 \times 10^{-8}$ $|f_0| = 0.2996 \times 10^{-19}$.

Example 2. The Chained Woof function (CHAIN-WOOD) ([7], p. 423)

$$f(X) = 1 + f_0(X)$$

$$f_0(X) = \sum_{i \in J} \left[100\left(x_{i+1} - x_i^2\right)^2 + \left(1 - x_i\right)^2 + 90\left(x_{i+3} - x_{i+2}^2\right)^2 \right.$$
$$\left. + \left(1 - x_{i+2}\right)^2 + 10\left(x_{i+1} + x_{i+3} - 2\right)^2 + 0.1\left(x_{i+1} - x_{i+3}\right)^2 \right],$$

Where N is a multiple of 4 and $J = \{1, 3, 5, \cdots, N-3\}$.

The Gradient function:

$$G(1) = -400x_1\left(x_2 - x_1^2\right) - 2(1 - x_1)$$

$$G(2) = 200\left(x_2 - x_1^2\right) + 20\left(x_2 + x_4 - 2\right) + 0.2\left(x_2 - x_4\right)$$

$$G(i) = -760x_i\left(x_{i+1} - x_i^2\right) - 4(1 - x_i)$$

$$G(i+1) = 380\left(x_{i+1} - x_i^2\right) + 20\left(x_{i-1} + x_{i+1} - 2\right)$$
$$-0.2\left(x_{i-1} - x_{i+1}\right) + 0.2\left(x_{i+1} - x_{i+3}\right) + 20\left(x_{i+1} + x_{i+3} - 2\right),$$
$$i = 3, 5, \cdots, N-3$$

$$G(N-1) = -360x_{N-1}\left(x_N - x_{N-1}^2\right) - 2(1 - x_{N-1})$$

$$G(N) = 180\left(x_N - x_{N-1}^2\right) + 20\left(x_{N-2} + x_N - 2\right)$$
$$-0.2\left(x_{N-2} - x_N\right)$$

The diagonal of Gradient function

$$D(1) = 1200x_1^2 - 400x_2 + 2$$

$$D(2) = 220.2$$

$$D(i) = 2280x_i^2 - 760x_{i+1} + 4$$

$$D(i+1) = 420.4, \quad i = 3, 5, \cdots, N-3$$

$$D(N-1) = 1080x_{N-1}^2 - 360x_N + 2$$

$$D(N) = 200.2$$

As ([7], p. 423) did, for $N = 8$, $X(0) = (-3, -1, -3, -1, -2, 0, -2, 0)$ the unconstrained problem have the solution $X_U^* = (1, 1, 1, 1, 1, 1, 1, 1)$.

Use EPS method, take $\varepsilon = 0.5, ToL1 = 1, ToL2 = 10^{-3}$, $ToL3 = 10^{-5}$, $h_1 = 5$, $h_2 = 10, h_3 = 15$; after 572 function evaluations we get $\|G(\tilde{X})\| = 0.3395 \times 10^{-5}$, $f_0 = 0.5253 \times 10^{-12}$. Despite we get the approximate solution $\tilde{X}_U^*$, but for such a small dimension problem EPS method does not show the superiority expect the simplicity. But we are interested in large scale problems.

For unconstrained (ChainWood) problem, $N = 100, 1000, 10000$, with three different start points $X(0)$, we get three different results:

1) $X(0) = (-3, -1, -3, -1, -2, 0, \cdots, -2, 0)$ (the original)

Take $\varepsilon = 0.5$, $ToL1 = 1.0$, $ToL2 = 10^{-3}$, $ToL3 = 10^{-5}$, $h_1 = 5.0$, $h_2 = 10.0$, $h_3 = 15.0$, after 811 function evaluations, for $N = 100, 1000, 10000$, we get same approximation of the minimal value of function $f = 1 + f_0 = 1 + 13.808 = 14.808$.

2) $X(0) = (-3, -1, -3, -1, 0, 0, \cdots, 0, 0)$

The parameters are the same as (1), for $N = 100$, after 857 function evaluations, we have $f = 1 + f_0 = 1 + 0.18824 \times 10^{-10}$. For $N = 1000, 10000$ after 856 function evaluations, we get the same result $f = 1 + f_0 = 1 + 0.19438 \times 10^{-10}$.

3) $X(0) = (0, -1, 0, -1, 0, 0, \cdots, 0, 0)$, this $X(0)$ is the same as [8].

The parameters are $ToL1 = 5.0$, $ToL2 = 10^{-3}$, $ToL3 = 10^{-5}$, $h_1 = 2.0$, $h_2 = 10.0$, $h_3 = 15.0$, $\varepsilon = 0.5$. For $N = 100$, after 628 function evaluations, we have $f = 1 + f_0 = 1 + 3.5743 = 4.5743$, it is exactly consistent with STIR. However, SBMIM which STIR wanted to compare with, converged to 46.713. For $N = 1000$, 10000, [8] did not give the results because SBMIM did not converge. Our method EPS, after 652, 659 function evaluations converged to the same result 4.5743.

7. References

[1]   T. M. Han and Y. H. Han, "Solving Large Scale Nonlinear Equations by a New ODE Numerical Integration Method," *Applied Mathematics*, Vol. 1, No. 3, 2010, pp. 222-229. doi: 10.4236/am.2010.13027

[2]   P. Deuflhard, "Newton Methods for Nonlinear Problems Affine Invariance and Adaptive Algorithms," Springer-Verlag, Berlin, 2004.

[3]   D. J. Higham, "Trust Region Algorithms and Timestep Selection," *SIAM Journal on Numerical Analysis*, Vol. 37, No. 1, 1999, pp. 194-210. doi:10.1137/S0036142998335972

[4]   D. M. Young, "Convergence Properties of the Symmetric and Unsymmetric Successive Overrelaxation Methods and Related Methods," *Mathematics of Computation*, Vol. 24, No. 112, 1970, pp. 793-807. doi:10.1090/S0025-5718-1970-0281331-4

[5]   Y. Saad, "Iterative Methods for Sparse Linear Systems," PWS Publishing Company, Boston, 1996.

[6]   A. Miele and J. W. Cantrell, "Study on a Memory Gradient Method for the Minimization of Functions," *Journal of Optimization Theory and Applications*, Vol. 3, No. 6, 1969, pp. 459-470. doi:10.1007/BF00929359

[7]   A. R. Conn, N. I. M. Gould and P. L. Toint, "Testing a Class of Methods for Solving Minimization Problems with Simple Bounds on the Variables," *Mathematics of Computation*, Vol. 50, No. 182, 1988, pp. 399-430. doi:10.1090/S0025-5718-1988-0929544-3

[8]   M. A. Branch, T. F. Coleman and Y. Y. Li, "A Subspace, Interior, and Conjugate Gradient Method for Large-Scale Bound-Constrained Minimization Problems," *SIAM Journal on Scientific Computing*, Vol. 21, No. 1, 1999, pp. 1-23. doi:10.1137/S1064827595289108