

# A Hybrid of the Newton-GMRES and Electromagnetic Meta-Heuristic Methods for Solving Systems of Nonlinear Equations

F. Toutounian · J. Saberi-Nadjafi · S. H. Taheri

Received: 13 October 2008 / Accepted: 22 July 2009 / Published online: 24 September 2009  
© Springer Science + Business Media B.V. 2009

**Abstract** Solving systems of nonlinear equations is perhaps one of the most difficult problems in all numerical computation. Although numerous methods have been developed to attack this class of numerical problems, one of the simplest and oldest methods, Newton's method is arguably the most commonly used. As is well known, the convergence and performance characteristics of Newton's method can be highly sensitive to the initial guess of the solution supplied to the method. In this paper a hybrid scheme is proposed, in which the Electromagnetic Meta-Heuristic method (EM) is used to supply a good initial guess of the solution to the finite difference version of the Newton-GMRES method (NG) for solving a system of nonlinear equations. Numerical examples are given in order to compare the performance of the hybrid of the EM and NG methods. Empirical results show that the proposed method is an efficient approach for solving systems of nonlinear equations.

**Keywords** Systems of nonlinear equations · Electromagnetism Meta-Heuristic method · Newton-GMRES method

**AMS Subject Classifications** 34A34 · 58C15 · 65H10 · 90C59

---

F. Toutounian · J. Saberi-Nadjafi · S. H. Taheri (✉)  
School of Mathematical Sciences, Ferdowsi University of Mashhad,  
P.O. Box. 1159-91775, Mashhad, Iran  
e-mail: taheri@math.um.ac.ir

F. Toutounian  
e-mail: toutouni@math.um.ac.ir

J. Saberi-Nadjafi  
e-mail: saberinadjafi@math.um.ac.ir

S. H. Taheri  
Department of Mathematics, University of Khayyam, P.O. Box. 9189-747178, Mashhad, Iran

### 1 Introduction

A nonlinear system of equations is defined as:

$$F(u) = 0, \tag{1}$$

where  $F = (f_1, f_2, \dots, f_n)^T$  is a nonlinear map from a domain in  $\mathfrak{R}^n$ , that contains the solution  $u^*$ , into  $\mathfrak{R}^n$ , and  $u \in \mathfrak{R}^n$ .

Such systems often arise in applied areas of physics, biology, engineering, geophysics, chemistry and industry. Numerous examples from all branches of the sciences are given in [1–3].

Research on systems of nonlinear equations has widely expanded over the last few decades, and reviews can be found in Broyden [4], Martinez [5], and Hribar [6]. As is well known, Newton’s method and its variations [7, 8] coupled with some direct solution technique such as Gaussian elimination are powerful solvers for these nonlinear systems in case one has a sufficiently good initial guess  $u_0$  and  $n$  is not too large. When the Jacobian is large and sparse, inexact Newton methods [9–13] or some kind of nonlinear block-iterative methods [14–16] may be used.

An Inexact Newton method is actually a two stage iterative method which has the following general form:

---

**Algorithm 1** Inexact Newton method

---

- I. Choose an initial approximation  $u_0$ .
- II. **for**  $k = 1, 2, \dots$  until convergence **do**:  
     find  $x_k$  satisfying

$$F'(u_k)x_k = -F(u_k) - r_k \tag{2}$$

$$\|r_k\|_2 \leq \eta_k \|F(u_k)\|_2; \tag{3}$$

    update  $u_{k+1} = u_k + x_k$ .

---

In Eq. 2,  $r_k = -F(u_k) - F'(u_k)x_k$  is the residual vector associated to  $x_k$ . In Eq. 3, the sequence  $[\eta_k]$  is used to control the level of accuracy needed on the computation of the approximate solution  $x_k$ .

The inner iteration is an iterative method for solving the Newton equations  $F'(u_k)x_k = -F(u_k)$  approximately with the residual  $r_k$ . The stopping relative residual control  $\|r_k\|_2 \leq \eta_k \|F(u_k)\|_2$  guarantees the local convergence of the method under the usual assumptions for Newton’s method [9].

Recently with the development of Krylov subspace projection methods, this class of methods such as Arnoli’s method [17] and the generalized minimum residual method (GMRES) [18] is widely used as the inner iteration for inexact Newton methods [10, 11]. This combined method is called inexact Newton-Krylov methods or nonlinear Krylov subspace projection methods. The Krylov methods have the virtue of requiring almost no matrix storage, resulting in a distinct advantage over direct methods for solving the large Newton equations. In particular, the product of a Jacobian and some fixed vector  $(F'(u)x)$  is only utilized in a Krylov method for solving  $F'(u_k)x_k = -F(u_k)$ , and the product can be approximated by the quotient

$$F'(u_k)x \approx \frac{F(u_k + \sigma x) - F(u_k)}{\sigma}, \tag{4}$$

where  $\sigma$  is a scalar. So, the Jacobian need not be computed explicitly. In [10] Peter N. Brown gave the local convergence results for inexact Newton-Krylov methods with the difference approximations of the Jacobian.

For solving nonlinear system (1), we consider the finite difference version of the Newton-GMRES method, which was given in [19], and can be described as follows:

---

**Algorithm 2** Newton-GMRES method(NG)

---

- I. Choose an initial approximation solution  $u_0$  of the nonlinear system; set  $k = 0$ ; choose a tolerance  $\varepsilon_0$ .
  - II. Solve the linear system  $Ax = b$ , where  $A = F'(u_k)$  and  $b = -F(u_k)$ .
    - (1) Initialization:
      - choose an initial approximation solution  $\hat{x}_0$ ;
      - compute  $q_0 = (F(u_k + \sigma_0 \hat{x}_0) - F(u_k))/\sigma_0$ ;  $\hat{r}_0 = b - q_0$ ;
      - $\hat{\beta} = \|\hat{r}_0\|_2$ ;  $\hat{b}_1 = \hat{r}_0/\hat{\beta}$ ;  $q_1 = \hat{b}_1$ .
    - (2) Arnoldi process:
      - for**  $j = 1$  to  $m$  **do**:
      - (a)  $q_{j+1} = (F(u_k + \sigma_j \hat{b}_j) - F(u_k))/\sigma_j$ ;  $\hat{\omega} = q_{j+1}$ ;
      - for**  $i = 1$  to  $j$  **do**:
      - $\hat{h}_{ij} = (\hat{b}_i, \hat{\omega})$ ;  $\hat{\omega} = \hat{\omega} - \hat{h}_{ij} \hat{b}_i$ ;
      - end for**;
      - $\hat{h}_{i+1j} = \|\hat{\omega}\|_2$ ;  $\hat{b}_{j+1} = \hat{\omega}/\hat{h}_{i+1j}$ ;
      - (b) compute an estimation of  $\rho_j = \|b - (F(u_k + \sigma \hat{x}_j) - F(u_k))/\sigma\|_2$ ;
      - If**  $\rho_j \leq \varepsilon_k$  set  $m = j$  and go to (3);
      - end for**.
    - (3) Update the solution  $\hat{x}_m$ :
      - compute  $\hat{d}_m$  as the solution of  $\min_{d \in \mathbb{R}^m} \|\hat{\beta} e_1^{(m+1)} - \tilde{H}_m d\|_2$ ;
      - compute  $\hat{x}_m = \hat{x}_0 + \hat{B}_m \hat{d}_m$ .
    - (4) GMRES restart:
      - compute an estimation of  $\rho_m = \|b - (F(u_k + \sigma \hat{x}_m) - F(u_k))/\sigma\|_2$ ;
      - If**  $\rho_m > \varepsilon_k$  set  $\hat{x}_0 = \hat{x}_m$  and go to (1);
  - III. Compute  $u_{k+1} = u_k + \hat{x}_m$ .
  - IV. **If**  $\|F(u_{k+1})\|_2$  is small enough or  $k \geq k_{\max}$  then stop;
  - else** set  $k = k + 1$ , choose a new tolerance  $\varepsilon_k$  and go to II.
- 

The local convergence of this algorithm has been studied in [19]. The performance and convergence characteristics of this algorithm are highly dependant on the initial guesses with which it begins and it is very important to have a good starting value  $u_0$ . Several techniques exist to remedy the difficulties associated with choice of the initial guess when solving nonlinear systems of equations. Most of these techniques fall into two categories, Line search methods and trust region methods. For these two categories, there are several theoretical results when combined with Newton-type methods that make them robust and hence, attractive [20–22].

In this paper, we show how, by using the Electromagnetic Meta-Heuristic (EM) method [23], one can obtain the sufficiently good initial guesses  $u_0$ . The results of

comparative study of the hybrid of the EM algorithm and NG method (called EM-NG method) and trust region method based on smooth CGS algorithm (called QCGS algorithm) [26] show that EM-NG method is effective and represents an efficient approach for solving nonlinear systems of equations.

This paper is organized as follows. In Section 2 we give a brief description of Electromagnetic Meta-Heuristic method [23]. In Section 3, we present a hybrid of Newton-GMRES method and Electromagnetic Meta-Heuristic method. Numerical experiments are given in Section 4. Finally, we give some concluding remarks in Section 5.

## 2 Electromagnetic Meta-Heuristic Method

In this section we will briefly review the EM method of Birbil and Fang [23] and discuss its main properties. Consider a special class of problems with bounded variables in the form of

$$\begin{aligned} \text{Min} \quad & f(x) \\ \text{s.t. } x \in & [\tilde{l}, \tilde{u}], \end{aligned} \quad (5)$$

where  $[\tilde{l}, \tilde{u}] = \{x \in \mathbb{R}^n \mid \tilde{l}_k \leq x_k \leq \tilde{u}_k, \quad k = 1, 2, \dots, n\}$ .

In a multi-dimensional solution space where each point represents a solution, a charge is associated with each point. This charge is related to the objective function value associated with the solution point. As in evolutionary search algorithms, a population, or set of solutions of size  $NS$ , is created, in which each solution point will exert attraction or repulsion on other points, the magnitude of which is proportional to the product of the charges and inversely proportional to the distance between the points. The charge of the point  $i$  is calculated according to the relative efficiency of the objective function values in the current population, i.e.,

$$q_i = \exp\left(-n \frac{f(x_i) - f(x_{best})}{\sum_{i=1}^{NS} (f(x_i) - f(x_{best}))}\right), \quad i = 1, \dots, NS, \quad (6)$$

where  $n$  is the dimension of the problem and  $x_{best}$  represents the point that has the best objective function value among all the points at the current iteration. In this way, the points that have better objective function values possess higher charges. Note that, unlike electrical charges, no signs are attached to the charge of an individual point in the Eq. 6; instead, the direction of a particular force between two points will be determined after comparing their objective function values. The principle behind of this algorithm is that inferior solution points will prevent a move in their direction by repelling other points in the population, and that attractive points will facilitate moves in their direction. This can be seen as a form of local search in Euclidian space in a population-based framework. The main difference of these existing methods is that the moves are governed by forces that obey the rules

of electromagnetism. Birbil and Fang provide a generic pseudo-code for the EM algorithm:

---

**Algorithm 3** Electromagnetic Meta-Heuristic method (EM)

---

```

I. Initialize()
II. While termination criteria are not satisfied do
    Local()
    CalcF()
    Move()
end while
    
```

---

The first procedure, Initialize, is used for sampling  $NS$  points randomly from the feasible region and assigning them their initial function values. Each coordinate of a point is assumed to be uniformly distributed between the corresponding upper bound and lower bound. Local is a neighborhood search procedure, which can be applied to one or many points for local refinements at each iteration. As mentioned in [23], the selections of these two procedures do not affect the convergence result of the EM method. The total force vector exerted on each point by all other points is calculated in the CalcF procedure and total force  $F$  exerted on point  $i$  is computed by the following equation:

$$F_i = \sum_{\substack{j=1 \\ j \neq i}}^{NS} \begin{cases} (x_j - x_i) \frac{q_i q_j}{\|x_j - x_i\|^2}, & \text{if } f(x_j) < f(x_i) \text{ (Attraction)} \\ (x_i - x_j) \frac{q_i q_j}{\|x_j - x_i\|^2}, & \text{if } f(x_i) \leq f(x_j) \text{ (Repulsion)} \end{cases}, \quad i = 1, 2, \dots, NS. \tag{7}$$

As explained in [23], between two points, the point that has a better objective function value attracts the other one. Contrarily, the point with a worse objective function value repels the other. So,  $x_{best}$  which has the minimum objective function value, attracts all other points in the population. After evaluating the total force vector  $F_i$  in CalcF procedure, the point  $i$  is moved in the direction of the force by a random step in the Move procedure.

Finally, Birbil and Fang showed that when the number of iterations is large enough, one of the points in the current population moves into the  $\varepsilon$ -neighborhood of the global optimum. More details of EM algorithm can be found in [23].

In Section 3, we propose an efficient algorithm for solving the systems of nonlinear equations in which the EM method is used to supply the good initial guesses to the NG method.

### 3 A Hybrid Method of Newton-GMRES and EM Methods

In this section, we present a hybrid method for solving the systems of nonlinear equations. The idea of the method is to transform the system of nonlinear equations (1) to an unconstrained minimization problem and at each iteration of the EM method to use the current best point as the initial guess for the NG Algorithm.

For solving the system of the nonlinear equations (1), we consider the minimization problem (5) with the objective function  $f(x) = \|F(x)\|_2$  and use the following hybrid method which is named EM-NG method.

---

**Algorithm 4** EM\_NG method
 

---

- I. Initialize()
  - II. **While** termination criteria are not satisfied **do**
    - Local()
    - CalcF()
    - Move()
  - III. Apply the NG method to the nonlinear system (1) using the current best solution as an initial guess. **If** the computed solution is not better than the current best solution, apply the NG method to the nonlinear system (1) using the second best solution as an initial guess and set  $\text{Length} = \text{Length} \times \alpha$  with  $\alpha > 1$ .
- end while**
- 

In the EM-NG Algorithm, the procedures CalcF and Move are the same as those of EM Algorithm [23]. The Local procedure which is a neighborhood search procedure and its selection does not affect the convergence result of EM method, is defined as follows:

---

**Algorithm 5** Local(Length, $\alpha$ )
 

---

- I.  $\text{Length} = \text{Length} \times \alpha$
  - II. **for**  $i = 1$  to  $NS$  **do**
    - for**  $l = 1$  to LSITER **do**
      - $y = x_i$
      - for**  $k = 1$  to  $n$  **do**
        - $z = y(k)$
        - $\lambda_1 = U(0, 1)$
        - $\lambda_2 = U(0, 1)$
        - if**  $\lambda_1 > 0.5$  **then**
          - $y_k = y_k + \lambda_2(\text{Length})$
        - else**
          - $y_k = y_k - \lambda_2(\text{Length})$
        - end if**
        - if**  $\text{abs}(y(k)) > \text{abs}(z)$  **then**
          - $y(k) = z$
        - end if**
      - end for**
      - if**  $f(y) < f(x_i)$  **then**
        - $x_i = y$
      - end if**
    - end for**
-

In Local procedure, in the neighborhood of each point solution  $x_i$  we generate a point solution with a norm smaller than that of  $x_i$ , and replace  $x_i$  with this new point solution if it is better than  $x_i$ . In this manner, the optimum solution with a minimum norm will be obtained in the given interval. Here, the parameters, Length,  $\alpha$ , and LSITER that are passed to this procedure, represent the maximum feasible step length, the multiplier and number of iterations for the neighborhood search, respectively.

In step III of Algorithm 4, we apply the NG method to the nonlinear system (1) using the current best solution as an initial guess. Using the fact that the NG algorithm is locally convergent [19], we replace  $x_{best}$  with the solution obtained by the NG algorithm if it is better than  $x_{best}$ . For the other case, when the NG method does not converge, and the solution obtained by this algorithm is not better than  $x_{best}$ , we conclude that  $x_{best}$  is not close enough to the exact solution and another initial guess must be chosen. In this case, we use the second best solution as an initial guess and set  $\text{Length} = \text{Length} \times \alpha$ , where the multiplier  $\alpha > 1$  will be defined by the user (for example  $\alpha = 10$ ), in order to change the very small step length and to furnish the situation in which the Local procedure can give a substantial reduction of  $f(x) = \|F(x)\|_2$ . Our experiments show that, in many problems, this choice and this change prevent the norm of residuals from oscillation and stagnation (see Example 2).

In Section 4, the numerical results show that with the EM-NG algorithm, it is possible, in a given interval to obtain the solution of the nonlinear systems with desired accuracy and the cost of computation is comparable with those of QCGS algorithm [26] and NG method when the latter converges.

### 4 Computational Results

In this section, we compare the performance of EM-NG method with that of Newton, NG, Evolutionary Method for NSEs (called EMO method) [24], Effati [25] and QCGS [26] methods. The algorithms were written in MATLAB and were tested for the examples given in [26] and [27–32]. All the problems were run on a PC with Pentium IV processor with 512 MB of RAM, and CPU 2.80 GHz. The algorithm used for random-number generation is an implementation of the Mersenne Twister algorithm described in [33]. In the NG Algorithm, as [19], we used the tolerance  $\varepsilon_k = \eta_k \|F(u_k)\|_2$ , with  $\eta_k = (0.5)^k$  for stopping the GMRES method. A maximal value of  $m(m_{\max})$  is used. If  $m = m_{\max}$  but  $\rho_m > \varepsilon_k$ , we restart GMRES once. A maximum number of 60 iterations of stage II ( $k_{\max} = 60$ ) was allowed. In EM-NG Algorithm, we used Algorithm 2 (the NG method) as a procedure (in step III) with the above parameters and  $k_{\max} = 15$ . In addition, the parameters  $NS = 3, 6, 12$ ,  $\text{Length} = \max(\tilde{u}_k - \tilde{l}_k)/2$ ,  $\delta = 0.5$ ,  $\text{LSITER} = 2$ , and  $\alpha = 10$  are used. A maximum number of 15 iterations was allowed in EM-NG Algorithm. As a stopping criterion to determine whether  $u_k$  solves the nonlinear problem  $F(u) = 0$ , we used  $\|F(u_k)\|_2 < \varepsilon \|F(u_0)\|_2$ , where  $\varepsilon$  will be defined for each problem.

*Example 1* We consider the nonlinear system of equations

$$\begin{cases} f_1(u_1, u_2) = e^{u_1} + u_1 u_2 - 1 = 0, \\ f_2(u_1, u_2) = \sin(u_1 u_2) + u_1 + u_2 - 1 = 0. \end{cases}$$

**Table 1** The results obtained for Example 1

Method	Solution	Function value
Newton’s method	(0.026, 0.862)	(0.0256, 0.0565)
Effati’s method	(0.0096, 0.9976)	(0.019223, 0.016776)
EMO method	(0.00138, 1.0027)	(0.00276, 0.0000637)
NG method	(−0.000000000000195, 0.999999999999899)	(−0.389-12, 0.490-12)
EM-NG method	(0.000000000000000, 1.000000000000000)	(0.0, 0.0)

which was described in [24]. In order to compare our results with those of other methods, we collect in Table 1, the results presented in reference [24] (which were obtained by the methods presented in Table 1 and the initial guess  $u_0 = (0.09, 0.09)^T$ ), the results of NG method with this initial guess, and the results of EM-NG method with  $m_{max} = 2, NS = 3, \varepsilon=10E-10$ . For the latter method, the points of the population were chosen randomly in the interval [0, 1]. As we observe, the best result which is the exact solution is obtained by EM-NG method.

*Example 2* (Generalized function of Rosenbrock) This example is given in [19]:

$$\begin{cases} f_1(u_1, u_2, \dots, u_n) = -4\zeta (u_2 - u_1^2) u_1 - 2(1 - u_1) = 0, \\ f_i(u_1, u_2, \dots, u_n) = 2\zeta (u_i - u_{i-1}) - 4 (u_{i+1} - u_i^2) u_i \\ \quad - 2(1 - u_{i-1}) = 0, & i = 2, 3, \dots, n - 1, \\ f_n(u_1, u_2, \dots, u_n) = 2\zeta (u_n - u_{n-1}^2) = 0. \end{cases}$$

The Jacobian matrix  $F'(u)$  is tridiagonal. The nonlinear system  $F(u) = 0$  has the unique solution  $u^* = (1, 1, \dots, 1)^T$ . We consider a system of size  $n = 5000$  and we take  $\zeta = 10$ . We have considered two intervals  $[-4, 4]$  and  $[-8, 8]$  and three initial approximate solutions  $u_0$ , denoted by  $rand_1, rand_2$  and  $rand_3$ . For the NG method, each component of the initial guess  $u_0$  was chosen randomly in these intervals. For the EM-NG method, we considered the initial guess  $u_0$  as the first point of the population and the other points of the population are also chosen randomly in these intervals. The results obtained with  $m_{max} = 10$  and  $\varepsilon=10E-8$ , are presented in Table 2. For each initial guess  $u_0$ , the final iteration number of EM-NG method  $It_{EM}$ , the

**Table 2** Results obtained for Example 2 with NG and EM-NG methods and different initial intervals

	Algorithm	$[\tilde{l}_j, \tilde{u}_j] = [-4, 4]$				$[\tilde{l}_j, \tilde{u}_j] = [-8, 8]$			
		$It_{EM}$	NFE	$\ F(u)\ _2$	CPU	$It_{EM}$	NFE	$\ F(u)\ _2$	CPU
$u_0 = rand_1$	NG	–	125	1.02E-4	2.94	–	1244	2.09	28.75
	EM-NG(3)	3	126	2.02E-6	3.36	3	144	4.22E-4	3.69
	EM-NG(6)	3	151	2.42E-5	4.10	3	146	5.34E-4	4.01
	EM-NG(12)	3	225	6.99E-5	6.69	3	214	6.15E-4	6.52
$u_0 = rand_2$	NG	–	171	4.18E-7	3.88	–	1237	2.02	28.88
	EM-NG(3)	3	141	2.71E-5	3.80	7	678	3.43E-4	15.70
	EM-NG(6)	3	156	1.00E-4	4.28	3	146	2.35E-4	3.99
	EM-NG(12)	5	603	3.82E-5	16.00	5	503	5.81E-4	14.00
$u_0 = rand_3$	NG	–	1260	2.09	27.91	–	1242	2.01	27.93
	EM-NG(3)	3	134	3.56E-5	3.64	5	353	7.59E-5	8.71
	EM-NG(6)	3	152	6.92E-5	4.16	5	400	5.58E-4	9.98
	EM-NG(12)	5	588	5.87E-5	15.60	4	472	4.71E-4	12.65



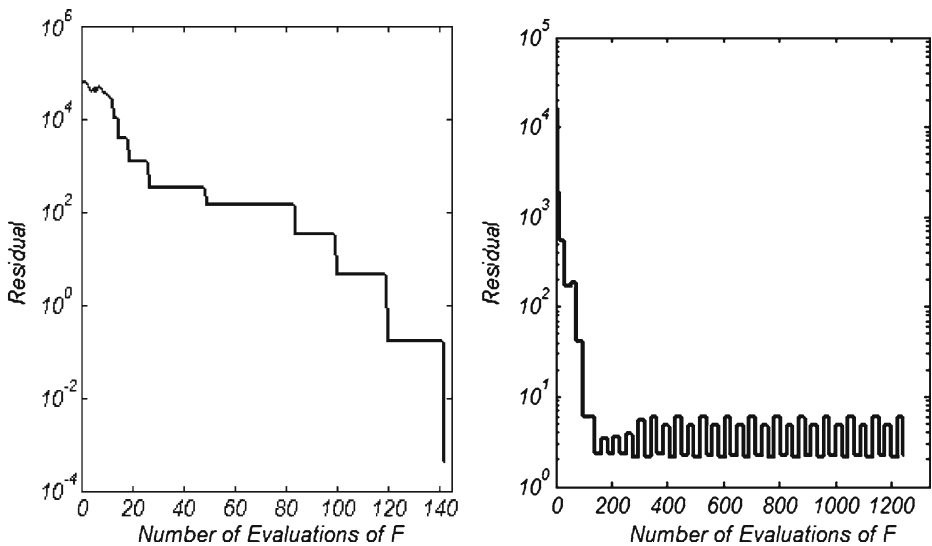
number of function evaluations NFE, the final norm  $\|F(u)\|_2$ , and CPU time needed to obtain the solution are given for different values of  $NS = 3, 6, 12$ .

In this table EM-NG(s) denotes the EM-NG algorithm with  $NS = s$ . The results show that, in all the cases, the EM-NG method could obtain the solution with desired accuracy, but there are the cases in which the NG method did not converge and the norm of residual after 60 iterations is more than 2. When the two methods converge (the cases  $u_0 = rand_1, u_0 = rand_2$ , and  $[\tilde{l}_j, \tilde{u}_j] = [-4, 4]$ ), the convergence behavior of NG method and EM-NG(3) method are similar with respect to the number of function evaluations NFE and CPU time. In addition, we observe that, in all the cases, except  $u_0 = rand_2$  and  $[\tilde{l}_j, \tilde{u}_j] = [-8, 8]$ , the results of EM-NG method with  $NS = 3$  (EM-NG(3)) is better than those of the others. So we can conclude that, for this problem a population of size  $NS = 3$  is sufficient for obtaining good results.

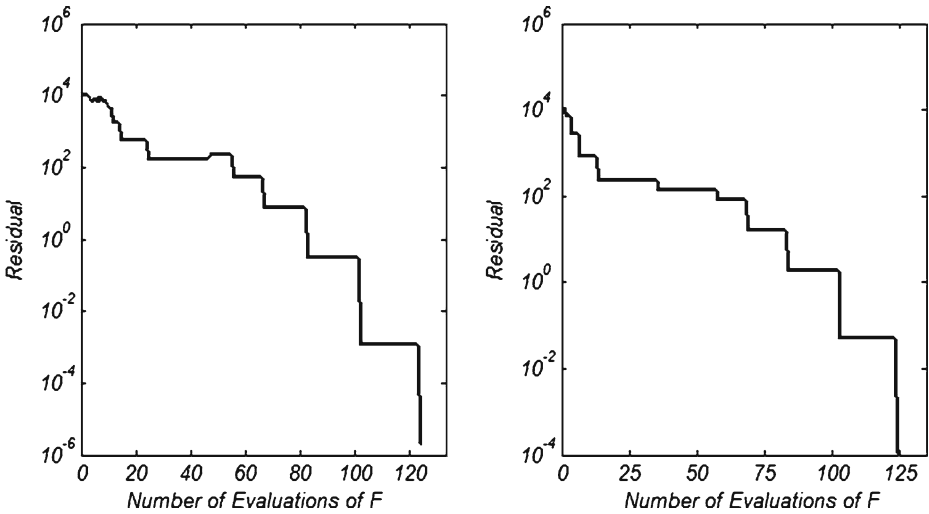
We also plot the values of  $\|F(u)\|_2$  as a function of the number of updating  $u_k$ . Figure 1 (right) shows the case in which the NG method is not able to reduce the residuals and the norm of residuals oscillates. Figure 1 (left) shows that when the procedure NG is not able to improve the solution and the norm of residuals oscillates, the EM-NG method furnishes another initial guess (the second best solution of population) for procedure NG and prevents the norm of residuals from oscillation. Finally, the convergence cases are plotted in Fig. 2.

**Example 3** (Bratu test [19]) The nonlinear problem is obtained after discretization (by 5-point finite differencing) of the following nonlinear partial equation over the unit square of  $\mathfrak{R}^2$  with Dirichlet boundary conditions:

$$-\Delta u + \alpha u_x + \lambda e^u = f.$$



**Fig. 1** Plots for Example 2 with EM-NG method (left) and NG method (right) for  $[\tilde{l}_j, \tilde{u}_j] = [-8, 8]$ ,  $u_0 = rand_1$ , and  $NS = 3$



**Fig. 2** Plots for Example 2 with EM-NG method (left) and NG method (right) when  $[\tilde{l}_j, \tilde{u}_j] = [-4, 4]$ ,  $u_0 = rand_1$ , and  $NS = 3$

The size of the nonlinear system is  $n = n_x n_y$ , where  $n_x + 2, n_y + 2$  are the numbers of mesh points in each direction, including mesh points on the boundary. The function  $f$  is chosen so that the solution of the discretized problem is known to be the constant unity  $e = (1, 1, \dots, 1)^T$ . For this problem, it is known that for  $\lambda \geq 0$ , there is always a unique solution, while this is not the case when  $\lambda < 0$ . As [19], in our experiments, we took  $n_x = n_y = 50$  ( $n = 2500$ ),  $\alpha = 100$  and  $\lambda = -10$ . We have considered two intervals  $[-2, 2]$  and  $[-6, 6]$  and each component of  $u_0$  and other points of population are chosen randomly in these intervals. The results obtained with  $\varepsilon = 10E-11, m_{max} = 5, 10, 15, NS = 3$ , and three initial guesses  $rand_1, rand_2, rand_3$  are presented in Table 3. In this table EM-NG(s) denotes the EM-NG algorithm with  $m_{max} = s$ . The results show that, the two methods converge and the solution was obtained with desired accuracy. The convergence rate of the NG method and EM-NG method with

**Table 3** Results obtained for example 3 with NG and EM-NG methods and different initial intervals

	Algorithm	$[\tilde{l}_j, \tilde{u}_j] = [-2, 2]$				$[\tilde{l}_j, \tilde{u}_j] = [-6, 6]$			
		It_EM	NFE	$\ F(u)\ _2$	CPU	It_EM	NFE	$\ F(u)\ _2$	CPU
$u_0 = rand_1$	NG	–	251	3.50E-7	7.12	–	229	4.09E-6	6.39
	EM-NG(5)	3	267	1.11E-6	7.65	3	259	8.45E-7	7.57
	EM-NG(10)	2	259	1.67E-6	7.29	2	249	2.54E-6	7.01
	EM-NG(15)	1	259	4.98E-7	7.24	1	237	5.38E-6	6.61
$u_0 = rand_2$	NG	–	273	1.19E-6	7.63	–	277	9.14E-6	7.74
	EM-NG(5)	3	264	9.87E-7	7.72	3	270	6.84E-6	7.66
	EM-NG(10)	2	244	3.09E-6	6.85	2	249	9.15E-6	7.01
	EM-NG(15)	1	235	3.68E-6	6.59	1	239	1.50E-5	6.70
$u_0 = rand_3$	NG	–	251	4.71E-6	7.10	–	273	4.26E-6	7.06
	EM-NG(5)	3	268	3.74E-7	7.73	3	266	5.49E-6	7.57
	EM-NG(10)	2	248	3.04E-6	7.01	2	274	1.75E-6	7.72
	EM-NG(15)	1	242	2.59E-6	6.83	1	259	2.29E-6	7.31

different values of  $m_{\max}$ , in term of CPU time and the number of function evaluations are close. The results of this example and Example 2 show that if the NG method converges, then the EM-NG method converges too, and its convergence rate is close to that of NG method. In addition, we observe that, by increasing the parameter  $m_{\max}$ , the number of function evaluations and CPU time decrease a little, except in the case of  $u_0 = rand_3, [\tilde{l}_j, \tilde{u}_j] = [-6, 6]$ , and  $m_{\max} = 10$ . Our experiments showed that the choice  $m_{\max} = 10$  furnishes the good results.

### 4.1 A Set of Problems

In this section, we present the results of a comparative study of three methods, NG, EM-NG methods, and trust region method based on smooth CGS algorithm (called QCGS algorithm) [26], for large sparse systems of nonlinear equations. All test results were obtained by means of 20 problems given in Appendix. All problems were considered with 100 variables, except problem 19 which has 99 variables and problem 20 with 10 variables. The QCGS algorithm contains several parameters. As [26], we have used the values

$$\begin{aligned} \bar{\beta}_1 = 0.05 \quad \bar{\beta}_2 = 0.75, \quad \bar{\gamma}_1 = 2 \quad \bar{\gamma}_2 = 10^6, \quad \bar{\rho}_1 = 0.1, \quad \bar{\rho}_2 = 0.9 \\ \bar{\tau}_0 = 10^{-3}, \quad \bar{\omega}_0 = 0.4, \quad \bar{\Delta} = 10^3, \quad \bar{\varepsilon} = 10^{-16} \quad , \quad \bar{k} = 1000 \quad \bar{l} = 20 \end{aligned}$$

in all numerical experiments. The elements of Jacobian matrix are computed by the formula

$$J_{ji} = \frac{[f_j(x + \delta e_i) - f_j(x)]}{\delta}, \tag{8}$$

where  $e_i$  is the  $i$ th column of the unit matrix and  $\delta = 10E-8$ . If the Jacobian matrix is sparse, only the nonzero elements are computed by the formula (8).

In the EM-NG and NG methods, for computing the vector  $q_j$  in stage II (step (2), (a)), we also used the Jacobian matrix. The parameter  $m_{\max} = 10$  is used for NG and EM-NG methods. The stopping criterion  $\|F(u_k)\|_2 < \varepsilon \|F(u_0)\|_2$  with  $\varepsilon = 10E-8$  was used for three methods. The parameter  $k_{\max} = 100$  and  $k_{\max} = 15$  were used for NG and EM-NG methods, respectively. Finally, a maximum number of 50 iterations was allowed in the EM-NG Algorithm.

First, we have applied the three methods to these problems with an initial guess  $x_0$  given for each problem in the Appendix and the initial guess  $u_0$ , for which each component is chosen randomly in the interval  $[-2, 2]$ . The results obtained are presented in Table 4. The rows of this table correspond to the individual problems; the columns contain the number of function evaluations (NFE) of EM-NG, QCGS, and NG methods. The symbol ‘ \* ’ indicates that the method did not converge after allowable iterations. In the last line,  $N_{suc}$  denotes the number of successfully solved problems by each method. As expected, in many cases, the results of the three methods obtained with the good initial guess  $x_0$  is better, in term of the number of function evaluations, than those obtained with random initial guess  $u_0$ . The last line shows that the EM-NG method is better, measured in the number of successfully solved problems, than the QCGS and NG methods, but in the case of convergence it does not give an advantage to the other methods.

**Table 4** The results of three methods for 20 test problems with initial guesses  $x_0$  and  $u_0$

Prob	Given $x_0$			$u_0 \in [-2, 2]$		
	NFE			NFE		
	EM-NG	QCGS	NG	EM-NG	QCGS	NG
$P_1$	141	67	*	*	*	*
$P_2$	53	314	43	*	*	*
$P_3$	29	19	19	78	73	151
$P_4$	39	34	29	44	58	37
$P_5$	71	65	61	904	*	*
$P_6$	115	178	93	140	57	473
$P_7$	199	99	151	362	91	265
$P_8$	123	194	265	148	298	1137
$P_9$	74	82	64	1233	*	*
$P_{10}$	35	123	25	738	*	*
$P_{11}$	109	46	46	116	49	100
$P_{12}$	79	89	58	*	*	*
$P_{13}$	43	37	33	*	*	*
$P_{14}$	59	37	43	60	67	49
$P_{15}$	735	65	805	320	33	537
$P_{16}$	39	25	29	*	*	*
$P_{17}$	32	312	34	36	149	28
$P_{18}$	35	*	25	576	*	*
$P_{19}$	907	*	117	1492	*	465
$P_{20}$	334	*	883	390	123	*
$N_{suc}$	20	17	19	15	10	10

Next, we have tested each problem with 100 initial guesses. Each component of each initial guess is chosen randomly in the interval  $[-2, 2]$  and  $NS = 3$  has been taken for all tests in EM-NG method. The results are given in Table 5. For each

**Table 5** The results of three methods for 20 test problems with 100 random initial guesses  $u_0$

Prob	EM-NG		QCGS		NG	
	mNFE	perc	mNFE	perc	mNFE	perc
$P_1$	671	33	254	9	826	1
$P_2$	1499	1	3240	0	598	0
$P_3$	78	100	61	98	85	89
$P_4$	40	100	41	100	33	98
$P_5$	852	15	4296	0	793	0
$P_6$	80	100	57	97	85	62
$P_7$	102	100	85	94	115	87
$P_8$	132	100	137	100	161	69
$P_9$	596	30	82	5	73	3
$P_{10}$	287	29	3129	0	601	0
$P_{11}$	93	100	46	100	4	100
$P_{12}$	75	45	3065	0	520	0
$P_{13}$	44	40	37	16	801	0
$P_{14}$	60	100	55	100	55	100
$P_{15}$	280	100	33	100	457	100
$P_{16}$	656	13	4305	0	797	0
$P_{17}$	33	100	82	100	28	28
$P_{18}$	87	96	3179	0	262	28
$P_{19}$	172	90	4314	0	185	99
$P_{20}$	386	100	111	35	111	85

**Table 6** EM-NG method with larger NS

Prob	NS = 25		NS = 50		NS = 100	
	NFE	perc	NFE	perc	NFE	perc
$P_1$	773	41	978	77	1388	79
$P_2$	1473	26	1763	45	4123	80
$P_5$	616	60	665	100	1003	100
$P_9$	773	43	978	93	1388	95
$P_{10}$	336	87	568	95	565	98
$P_{12}$	157	63	251	73	460	81
$P_{13}$	132	46	232	61	436	73
$P_{16}$	326	36	617	100	460	100
$P_{18}$	172	99	257	100	448	100
$P_{19}$	318	93	505	95	885	97

method we reported the minimum number of function evaluations needed for solving the problem and obtained in 100 randomly generated guesses (mNFE) and the number of successes of the method in these tests (perc). Here, a “success” means that the convergence to the exact solution of the system of equations. Table 5 shows that, for all problems, EM-NG method is much better, in term of the number of successes, than QCGS and NG methods.

Finally, we have again applied the EM-NG method with larger  $NS$ ,  $NS = 25, 50, 100$ , to the problems for which the number of successes of EM-NG method is less than 100 (Problems 1, 2, 5, 9, 10, 12, 13, 16, 18, 19). The results are given in Table 6. As we observe, by increasing  $NS$  (the size of population in the EM-NG method), the number of successes of EM-NG method increases. Table 6 shows that with  $NS = 100$  (which is equal to the dimension of problems), for 16 problems, the percentage of successes of EM-NG method is over 95%. From these results, we can conclude that the EM-NG method is an efficient approach for solving the systems of nonlinear equations.

### 5 Conclusion

We have proposed a hybrid of Newton-GMRES and Electromagnetism meta-heuristic method for solving a system of nonlinear equations. In the proposed method, the Electromagnetic Meta-Heuristic method (EM) is used to supply the good initial guesses of the solution to the finite difference version of the Newton-GMRES method (NG) for solving the system of nonlinear equations. We observed that the EM-NG method is able to obtain the solution with desired accuracy in a reasonable number of iterations. The experiments showed that in the case of convergence of the NG method, the convergence behavior of the EM-NG method and NG method are similar with respect to the number of function evaluations NFE and CPU time. The advantage of the EM-NG algorithm is that when the procedure NG is not able to improve the solution and the norm of residuals oscillates, step III of the algorithm furnishes another initial guess (the second best solution of population) for procedure NG and prevents the norm of residuals from oscillation

and stagnation. The experiments show that, the EM-NG method is much better, measured in number of successfully solved problems, than the QCGS and NG methods. In addition, we observe that, the percentage of success of EM-NG methods increases when NS (the size of population in the EM-NG method) increases. Consequently, the EM-NG method is an efficient approach for solving systems of nonlinear equations.

**Acknowledgements** The authors are grateful to the anonymous referees and editor Professor J. MacGregor Smith for their comments which substantially improved the quality of this paper.

### Appendix

Our test problems consist of searching for a solution to the system of nonlinear equations

$$f_k(x) = 0, \quad 1 \leq k \leq n.$$

For each problem an initial guess  $\bar{x}_l, 1 \leq l \leq n$ , is given. We use the functions div (integer division) and mod (remainder after integer division). These problems are given in [26–32].

**Problem 1** Countercurrent Reactor Problem 1 [27]:

$$\begin{aligned} \alpha &= 0.5 \\ f_k(x) &= \alpha - (1 - \alpha)x_{k+2} - x_k(1 + 4x_{k+1}) & k = 1, \\ f_k(x) &= -(2 - \alpha)x_{k+2} - x_k(1 + 4x_{k-1}) & k = 2, \\ f_k(x) &= \alpha x_{k-2} - (1 - \alpha)x_{k+2} - x_k(1 + 4x_{k+1}) & \text{mod}(k, 2) = 1, \quad 2 < k < n - 1 \\ f_k(x) &= \alpha x_{k-2} - (2 - \alpha)x_{k+2} - x_k(1 + 4x_{k-1}) & \text{mod}(k, 2) = 1, \quad 2 < k < n - 1 \\ f_k(x) &= \alpha x_{k-2} - x_k(1 + 4x_{k+1}) & k = n - 1, \\ f_k(x) &= \alpha x_{k-2} - (2 - \alpha) - x_k(1 + 4x_{k-1}) & k = n, \\ \bar{x}_l &= 0.1, \quad \text{mod}(l, 8) = 1, \\ \bar{x}_l &= 0.2, \quad \text{mod}(l, 8) = 2, \text{ or } \text{mod}(l, 8) = 0, \\ \bar{x}_l &= 0.3, \quad \text{mod}(l, 8) = 3, \text{ or } \text{mod}(l, 8) = 7, \\ \bar{x}_l &= 0.4, \quad \text{mod}(l, 8) = 4, \text{ or } \text{mod}(l, 8) = 6, \\ \bar{x}_l &= 0.5, \quad \text{mod}(l, 8) = 5. \end{aligned}$$

**Problem 2** Extended Powell Badly Scaled Function [28]:

$$\begin{aligned} f_k(x) &= 10000 x_k x_{k+1} - 1, & \text{mod}(k, 2) = 1, \\ f_k(x) &= \exp(-x_{k-1}) + \exp(-x_k) - 1.0001, & \text{mod}(k, 2) = 0, \\ \bar{x}_l &= 0, \quad \text{mod}(l, 2) = 1, \\ \bar{x}_l &= 1, \quad \text{mod}(l, 2) = 0. \end{aligned}$$

**Problem 3** Trigonometric System [29]:

$$\begin{aligned}
 i &= \text{div}(k - 1, 5), \\
 f_k(x) &= 5 - (i + 1)(1 - \cos x_k) - \sin x_k - \sum_{j=5i+1}^{5i+5} \cos x_j, \\
 \bar{x}_l &= 1/n, \quad l \geq 1.
 \end{aligned}$$

**Problem 4** Trigonometric-Exponential System, Trigexp 1 [29]:

$$\begin{aligned}
 f_k(x) &= 3x_k^3 + 2x_{k+1} - 5 + \sin(x_k - x_{k+1}) \sin(x_k + x_{k+1}), & k = 1, \\
 f_k(x) &= 3x_k^2 + 2x_{k+1} - 5 + \sin(x_k - x_{k+1}) \sin(x_k + x_{k+1}) + 4x_k \\
 &\quad - x_{k-1} \exp(x_{k-1} - x_k) - 3, & 1 < k < n, \\
 f_k(x) &= 4x_k - x_{k-1} \exp(x_{k-1} - x_k) - 3, & k = n, \\
 \bar{x}_l &= 0, \quad l \geq 1.
 \end{aligned}$$

**Problem 5** Singular Broyden Problem [30]:

$$\begin{aligned}
 f_k(x) &= ((3 - 2x_k)x_k - 2x_{k+1} + 1)^2, & k = 1, \\
 f_k(x) &= ((3 - 2x_k)x_k - x_{k-1} - 2x_{k+1} + 1)^2, & 1 < k < n, \\
 f_k(x) &= ((3 - 2x_k)x_k - x_{k-1} + 1)^2, & k = n, \\
 \bar{x}_l &= -1, \quad l \geq 1.
 \end{aligned}$$

**Problem 6** Tridiagonal System [31]:

$$\begin{aligned}
 f_k(x) &= 4(x_k - x_{k+1}^2), & k = 1, \\
 f_k(x) &= 8x_k(x_k^2 - x_{k-1}) - 2(1 - x_k) + 4(x_k - x_{k+1}^2), & 1 < k < n, \\
 f_k(x) &= 8x_k(x_k^2 - x_{k-1}) - 2(1 - x_k), & k = n, \\
 \bar{x}_l &= 12 \quad l \geq 1.
 \end{aligned}$$

**Problem 7** Five-Diagonal System [31]:

$$\begin{aligned}
 f_k(x) &= 4(x_k - x_{k+1}^2) + x_{k+1} - x_{k+2}^2, & k = 1, \\
 f_k(x) &= 8x_k(x_k^2 - x_{k-1}) - 2(1 - x_k) + 4(x_k - x_{k+1}^2) \\
 &\quad + x_{k+1} - x_{k+2}^2, & k = 2, \\
 f_k(x) &= 8x_k(x_k^2 - x_{k-1}) - 2(1 - x_k) + 4(x_k - x_{k+1}^2) \\
 &\quad + x_{k-1}^2 - x_{k-2} + x_{k+1} - x_{k+2}^2, & 2 < k < n - 1, \\
 f_k(x) &= 8x_k(x_k^2 - x_{k-1}) - 2(1 - x_k) + 4(x_k - x_{k+1}^2) \\
 &\quad + x_{k-1}^2 - x_{k-2}, & k = n - 1, \\
 f_k(x) &= 8x_k(x_k^2 - x_{k-1}) - 2(1 - x_k) + x_{k-1}^2 - x_{k-2}, & k = n, \\
 \bar{x}_l &= -2, \quad l \geq 1.
 \end{aligned}$$

**Problem 8** Seven-Diagonal System [31]:

$$\begin{aligned}
 f_k(x) &= 4(x_k - x_{k+1}^2) + x_{k+1} - x_{k+2}^2 + x_{k+2} - x_{k+3}^2, & k=1, \\
 f_k(x) &= 8x_k(x_k^2 - x_{k-1}) - 2(1-x_k) + 4(x_k - x_{k+1}^2) + x_{k-1}^2 + x_{k+1} - x_{k+2}^2 \\
 &\quad + x_{k+2} - x_{k+3}^2, & k=2, \\
 f_k(x) &= 8x_k(x_k^2 - x_{k-1}) - 2(1-x_k) + 4(x_k - x_{k+1}^2) + x_{k-1}^2 - x_{k-2} + x_{k+1} \\
 &\quad - x_{k+2}^2 + x_{k-2}^2 + x_{k+2} - x_{k+3}^2, & k=3, \\
 f_k(x) &= 8x_k(x_k^2 - x_{k-1}) - 2(1-x_k) + 4(x_k - x_{k+1}^2) + x_{k-1}^2 - x_{k-2} + x_{k+1} \\
 &\quad - x_{k+2}^2 + x_{k-2}^2 + x_{k+2} - x_{k-3} - x_{k+3}^2, & 3 < k < n-2, \\
 f_k(x) &= 8x_k(x_k^2 - x_{k-1}) - 2(1-x_k) + 4(x_k - x_{k+1}^2) + x_{k-1}^2 - x_{k-2} + x_{k+1} \\
 &\quad - x_{k+2}^2 + x_{k-2}^2 + x_{k+2} - x_{k-3}, & k=n-2 \\
 f_k(x) &= 8x_k(x_k^2 - x_{k-1}) - 2(1-x_k) + 4(x_k - x_{k+1}^2) + x_{k-1}^2 - x_{k-2} + x_{k+1} \\
 &\quad + x_{k-2}^2 - x_{k-3}, & k=n-1, \\
 f_k(x) &= 8x_k(x_k^2 - x_{k-1}) - 2(1-x_k) + x_{k-1}^2 - x_{k-2} + x_{k-2}^2 - x_{k-3}, & k=n, \\
 \bar{x}_l &= -3, \quad l \geq 1.
 \end{aligned}$$

**Problem 9** Structured Jacobian Problem [30]:

$$\begin{aligned}
 f_k(x) &= -2x_k^2 + 3x_k - 2x_{k+1} + 3x_{n-4} - x_{n-3} \\
 &\quad - x_{n-2} + 0.5x_{n-1} - x_n + 1, & k=1, \\
 f_k(x) &= -2x_k^2 + 3x_k - x_{k-1} - 2x_{k+1} + 3x_{n-4} \\
 &\quad - x_{n-3} - x_{n-2} + 0.5x_{n-1} - x_n + 1, & 1 < k < n, \\
 f_k(x) &= -2x_k^2 + 3x_k - x_{k-1} + 3x_{n-4} - x_{n-3} \\
 &\quad - x_{n-2} + 0.5x_{n-1} - x_n + 1, & k=n, \\
 \bar{x}_l &= -1, \quad l \geq 1.
 \end{aligned}$$

**Problem 10** Extended Rosenbrock Function [26]:

$$\begin{aligned}
 f_k(x) &= 10(x_{k+1} - x_k^2), & \text{mod}(l, 2) &= 1, \\
 f_k(x) &= 1 - x_{k-1}, & \text{mod}(l, 2) &= 0, \\
 \bar{x}_l &= -1.2, \quad \text{mod}(l, 2) = 1, \\
 \bar{x}_l &= 1, \quad \text{mod}(l, 2) = 0,
 \end{aligned}$$

**Problem 11** Extended Powell Singular Function [26]:

$$\begin{aligned}
 f_k(x) &= x_k + 10x_{k+1}, & \text{mod}(l, 4) &= 1, \\
 f_k(x) &= \sqrt{5}(x_{k+1} - x_{k+2}), & \text{mod}(l, 4) &= 2, \\
 f_k(x) &= (x_{k-1} - 2x_k)^2, & \text{mod}(l, 4) &= 3, \\
 f_k(x) &= \sqrt{10}(x_{k-3} - x_k)^2, & \text{mod}(l, 4) &= 0, \\
 \bar{x}_l &= 3, \quad \text{mod}(l, 4) = 1, \\
 \bar{x}_l &= -1, \quad \text{mod}(l, 4) = 2, \\
 \bar{x}_l &= 0, \quad \text{mod}(l, 4) = 3, \\
 \bar{x}_l &= 1, \quad \text{mod}(l, 4) = 0,
 \end{aligned}$$



**Problem 12** Extended Cragg and Levy Function [26]:

$$\begin{aligned}
 f_k(x) &= (\exp(x_k) - x_{k+1})^2, & \text{mod}(l, 4) &= 1, \\
 f_k(x) &= 10(x_k - x_{k+1})^3, & \text{mod}(l, 4) &= 2, \\
 f_k(x) &= \tan^2(x_k - x_{k+1}), & \text{mod}(l, 4) &= 3, \\
 f_k(x) &= x_k - 1, & \text{mod}(l, 4) &= 0, \\
 \bar{x}_l &= 1, & \text{mod}(l, 4) &= 1, \\
 \bar{x}_l &= 2, & \text{mod}(l, 4) &\neq 1,
 \end{aligned}$$

**Problem 13** Broyden Tridiagonal Function [26]:

$$\begin{aligned}
 f_k(x) &= x_k(0.5x_k - 3) + 2x_{k+1} - 1, & k &= 1, \\
 f_k(x) &= x_k(0.5x_k - 3) + x_{k-1} + 2x_{k+1} - 1, & 1 < k < n, \\
 f_k(x) &= x_k(0.5x_k - 3) - 1 + x_{k-1}, & k &= n, \\
 \bar{x}_l &= -1, & l &\geq 1.
 \end{aligned}$$

**Problem 14** Broyden Banded Problem [28]:

$$\begin{aligned}
 k_1 &= \max(1, k - 5), & k_2 &= \min(n, k + 1), \\
 f_k(x) &= (2 + 5x_k^2)x_k + 1 + \sum_{i=k_1}^{k_2} x_i(1 + x_i), \\
 \bar{x}_l &= -1, & l &\geq 1.
 \end{aligned}$$

**Problem 15** Discrete Boundary-Value Problem [28]:

$$\begin{aligned}
 h &= 1/(n + 1), \\
 f_k(x) &= 2x_k + h^2(x_k + 1 + hk)^3/2 - x_{k+1}, & k &= 1, \\
 f_k(x) &= 2x_k + h^2(x_k + 1 + hk)^3/2 - x_{k-1} - x_{k+1}, & 1 < k < n, \\
 f_k(x) &= 2x_k + h^2(x_k + 1 + hk)^3/2 - x_{k-1}, & k &= n, \\
 \bar{x}_l &= lh(lh - 1), & l &\geq 1.
 \end{aligned}$$

**Problem 16** Broyden Tridiagonal Problem [28]:

$$\begin{aligned}
 f_k(x) &= (3 - 2x_k)x_k - 2x_{k+1} + 1, & k &= 1, \\
 f_k(x) &= (3 - 2x_k)x_k - x_{k-1} - 2x_{k+1} + 1, & 1 < k < n, \\
 f_k(x) &= (3 - 2x_k)x_k - x_{k-1} + 1, & k &= n, \\
 \bar{x}_l &= -1, & l &\geq 1.
 \end{aligned}$$

**Problem 17** Modified Rosenbrock [32]:

$$\begin{aligned}
 f_k(x) &= \frac{1}{1 + \exp(-x_k)} - 0.73, & \text{mod}(k, 2) &= 1, \\
 f_k(x) &= 10(x_k - x_{k-1}^2), & \text{mod}(k, 2) &= 0, \\
 \bar{x}_l &= -1.8, & \text{mod}(l, 2) &= 1, \\
 \bar{x}_l &= -1, & \text{mod}(l, 2) &= 0,
 \end{aligned}$$

**Problem 18** Augmented Rosenbrock [32]:

$$\begin{aligned} f_k(x) &= 10(x_{k+1} - x_k^2), & \text{mod}(l, 4) &= 1, \\ f_k(x) &= 1 - x_{k-1}, & \text{mod}(l, 4) &= 2, \\ f_k(x) &= 1.25x_k - 0.25x_k^3, & \text{mod}(l, 4) &= 3, \\ f_k(x) &= x_k, & \text{mod}(l, 4) &= 0, \\ \bar{x}_l &= 3, & \text{mod}(l, 4) &= 1, \\ \bar{x}_l &= -1, & \text{mod}(l, 4) &= 2, \\ \bar{x}_l &= 0, & \text{mod}(l, 4) &= 3, \\ \bar{x}_l &= 1, & \text{mod}(l, 4) &= 0, \end{aligned}$$

**Problem 19** Diagonal of three variables premultiplied by a quasi-orthogonal matrix [32]:

$$\begin{aligned} f_k(x) &= 0.6x_k + 1.6x_{k+1}^3 - 7.2x_{k+1}^2 + 9.6x_k - 4.8, & \text{mod}(l, 3) &= 1, \\ f_k(x) &= 0.48x_{k-1} - 0.72x_k^3 + 3.24x_k^2 - 4.32x_k \\ &\quad - x_{k+1} + 0.2x_{k+1}^3 + 2.16 & \text{mod}(l, 3) &= 2, \\ f_k(x) &= 1.25x_k - 0.25x_k^3, & \text{mod}(l, 3) &= 0, \\ \bar{x}_l &= 50, & \text{mod}(l, 3) &= 1, \\ \bar{x}_l &= .5, & \text{mod}(l, 3) &= 2, \\ \bar{x}_l &= -1, & \text{mod}(l, 3) &= 0. \end{aligned}$$

**Problem 20** Quadratics and nonlinear equation [32]:

$Q_k \in \mathfrak{N}^{(n-1) \times (n-1)}$ ,  $k = 1, 2, 3, \dots, n-1$ , and  $b_k \in \mathfrak{N}^{n-1}$ ,  $k = 1, 2, 3, \dots, n-1$

have random elements between  $-1$  and  $1$ . We define

$$f_k(x) = \frac{1}{2}x^T Q_k x + b_k^T x, \text{ for } k = 1, 2, 3, \dots, n-1,$$

and  $f_n(x) = \text{atan}(x_1 + x_2 + \dots + x_n)$

$$\begin{aligned} \bar{x}_l &= 1, & \text{mod}(l, 4) &= 1, \\ \bar{x}_l &= 10, & \text{mod}(l, 4) &= 2, \\ \bar{x}_l &= 100, & \text{mod}(l, 4) &= 3, \\ \bar{x}_l &= 1000, & \text{mod}(l, 4) &= 0. \end{aligned}$$

## References

1. Allgower, E.L., Georg, K. (eds.): Computational Solution of Nonlinear Systems of Equations. American Mathematical Society, Providence (1990)
2. Merlet, J.P.: The COPRIN examples. <http://www-sop.inria.fr/coprin/logiciels/ALIAS/Benches/benches.html> (2006)
3. Pinter, J.D.: Computational Global Optimization in Nonlinear Systems: An Interactive Tutorial. Lionhart, Atlanta (2001)
4. Broyden, C.G., Luss, D.: A class of methods for solving nonlinear simultaneous equation. Math. Comput. **19**, 577–593 (1965)

5. Martínez, J.M.: Algorithms for Solving Nonlinear Systems of Equations, pp. 81–108. Continuous Optimization (1994)
6. Hribar, M.B.: Methods for large-scale nonlinear programming and nonlinear systems of equations. Ph.D. dissertation, EECS Department, Northwestern University (1997)
7. Ortega, J.M., Rheinboldt, W.C.: Iterative Solution of Nonlinear Equations in Several Variables. Academic Press, New York (1970)
8. Rheinboldt, W.C.: Methods for Solving Systems of Nonlinear Equations, 2nd edn. SIAM, Philadelphia (1998)
9. Dembo, R.S., Eisenstat, S.C., Steihaug, T.: Inexact Newton methods. *SIAM J. Numer. Anal.* **19**(2), 400–408 (1982)
10. Brown, P.N.: A local convergence theory for combined inexact-Newton/finite-difference projection methods. *SIAM J. Numer. Anal.* **24**(2), 407–434 (1987)
11. Brown, P.N., Saad, Y.: Hybrid Krylov methods for nonlinear systems of equations. *SIAM J. Sci. Statist. Comput.* **11**(3), 450–481 (1990)
12. Morini, B.: Convergence behaviour of inexact Newton methods. *Math. Comp.* **68**, 1605–1613 (1999)
13. Argyros, I.K.: Convergence rates for inexact Newton-like methods at singular points and applications. *Appl. Math. Comput.* **102**, 185–201 (1999)
14. Zecevic, A.I., Siljak, D.D.: A block-parallel Newton method via overlapping epsilon decompositions. *SIAM J. Matrix Anal. Appl.* **15**(3), 824–844 (1994)
15. Yang, G., Dutto, L.C., Fortin, M.: Inexact block Jacobi-Broyden methods for solving nonlinear systems of equations. *SIAM J. Sci. Comput.* **18**(5), 1367–1392 (1997)
16. Xu, J.J.: Convergence of partially asynchronous block Quasi-Newton methods for nonlinear systems of equations. *J. Appl. Math. Comput.* **103**, 307–321 (1999)
17. Saad, Y.: Krylov subspace methods for solving unsymmetric linear systems. *Math. Comp.* **37**, 105–126 (1981)
18. Saad, Y., Schultz, M.H.: GMRES: a generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM J. Sci. Statist. Comput.* **7**, 856–869 (1986)
19. Heyouni, M.: Newton generalized Hessemberg method for solving nonlinear systems of equations. *Numer. Algorithms* **21**, 225–246 (1999)
20. Conn, A.R., Gould, N.I.M., Toint, P.L.: Trust-Region Methods. SIAM, Philadelphia (2000)
21. Dennis, J.E., Schnabel, R.: Numerical Methods for Unconstrained Optimization and Nonlinear Equations. SIAM, Philadelphia (1996)
22. Nocedal, J., Wright, S.: Numerical Optimization. Springer, New York (2000)
23. Birbil, S.I., Fang, S.C.: An electromagnetism-like mechanism for global optimization. *J. Glob. Optim.* **25**, 263–282 (2003)
24. Grosan, C., Abraham, A., Gelbukh, A.: Evolutionary Method for Nonlinear Systems of Equations. MICAI 2006: Advances in Artificial Intelligence, pp. 283–293. Springer, Berlin (2006)
25. Effati, S., Nazemi, A.R.: A new method for solving a system of the nonlinear equations. *Appl. Math. Comput.* **168**, 877–894 (2005)
26. Luksan, L.: Inexact trust region method for large sparse systems of nonlinear equations. *J. Optim. Theory Appl.* **81**(3), 569–590 (1994)
27. Bogle, I.D.L., Perkins, J.D.: A new sparsity-preserving Quasi-Newton update for solving nonlinear equations. *SIAM J. Sci. Statist. Comput.* **11**, 621–630 (1990)
28. Moré, J.J., Garbow, B.S., Hillstöm, K.E.: Testing unconstrained optimization software. *ACM Trans. Math. Softw.* **7**, 17–41 (1981)
29. Toint, P.L.: Numerical solution of large sets of algebraic equations. *Math. Comput.* **46**, 175–189 (1986)
30. Gomez-Ruggiero, M.A., Martínez, J.M., Moretti, A.C.: Comparing algorithms for solving sparse nonlinear systems of equations. *SIAM J. Sci. Statist. Comput.* **13**, 459–483 (1992)
31. Li, G.: Successive column correction algorithms for solving sparse nonlinear systems of equations. *Math. Program.* **43**, 187–207 (1989)
32. Friedlander, A., Gomes-Ruggiero, M.A., Kozakevich, D.N., Martinez, J.M., Santos, S.A.: Solving nonlinear systems of equations by means of Quasi-Newton methods with nonmonotone strategy. *Optim. Methods Softw.* **87**, 25–51 (1997)
33. Matsumoto, M., Nishimura, T.: Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Trans. Model. Comput. Simul.* **8**(1), 3–30 (1998)