Computers and Mathematics with Applications 71 (2016) 1464–1478

Contents lists available at ScienceDirect

Computers and Mathematics with Applications

journal homepage: www.elsevier.com/locate/camwa

Solving systems of nonlinear equations when the nonlinearity is expensive

Sabahat Qasim[a], Zulifqar Ali[a], Fayyaz Ahmad[b,c,*], S. Serra-Capizzano[b,d], Malik Zaka Ullah[e,b], Arshad Mahmood[e]

[a] Department of Mathematics, Riphah International University, Main Satyana Road, Faisalabad 44000, Pakistan
[b] Dipartimento di Scienza e Alta Tecnologia, Università dell'Insubria, Via Valleggio 11, Como 22100, Italy
[c] Departament de Física i Enginyeria Nuclear, Universitat Politècnica de Catalunya, Comte d'Urgell 187, 08036 Barcelona, Spain
[d] Department of Information Technology, Uppsala University, Box 337, SE-751 05 Uppsala, Sweden
[e] Department of Mathematics, King Abdulaziz University, Jeddah 21589, Saudi Arabia

ARTICLE INFO

Article history:
Received 10 October 2015
Received in revised form 21 January 2016
Accepted 13 February 2016
Available online 22 March 2016

Keywords:
Multi-step iterative methods
Systems of nonlinear equations
Ordinary differential equations
Partial differential equations
Computationally expensive nonlinear functions

ABSTRACT

Construction of multi-step iterative method for solving system of nonlinear equations is considered, when the nonlinearity is expensive. The proposed method is divided into a base method and multi-step part. The convergence order of the base method is five, and each step of multi-step part adds additive-factor of five in the convergence order of the base method. The general formula of convergence order is $5(m-2)$ where $m(\geq 3)$ is the step number. For a single instance of the iterative method we only compute two Jacobian and inversion of one Jacobian is required. The direct inversion of Jacobian is avoided by computing LU factors. The computed LU factors are used in the multi-step part for solving five systems of linear equations that make the method computational efficient. The distinctive feature of the underlying multi-step iterative method is the single call to the computationally expensive nonlinear function and thus offers an increment of additive-factor of five in the convergence order per single call. The numerical simulations reveal that our proposed iterative method clearly shows better performance, where the computational cost of the involved nonlinear function is higher than the computational cost for solving five lower and upper triangular systems.

1. Introduction

Numerical iterative methods are attractive for solving nonlinear problems in the diverse area of science and technology. It is hard to find closed form solutions for nonlinear problems, but numerical solvers provide the way to approximate them. Even in the case of linear problems, when the size becomes large, direct methods become inefficient and often do not produce accurate solutions. Thus iterative numerical methods represent a valid alternative to the use of direct numerical procedures. Of course, we do not need just an iterative method, but we are looking for efficient numerical iterative methods. Most of the iterative methods are proposed and tested for academic problems, but problems originated from real life are

* Corresponding author at: Departament de Física i Enginyeria Nuclear, Universitat Politècnica de Catalunya, Comte d'Urgell 187, 08036 Barcelona, Spain.
E-mail addresses: sabahat.math@riphahfsd.edu.pk (S. Qasim), zulifqar.ali@nu.edu.pk (Z. Ali), fayyaz.ahmad@upc.edu (F. Ahmad), s.serracapizzano@uninsubria.it (S. Serra-Capizzano), malik.zakaullah@uninsubria.it (M.Z. Ullah), mahmood.arshad35@yahoo.com (A. Mahmood).

http://dx.doi.org/10.1016/j.camwa.2016.02.018
0898-1221/

computationally expensive. A wide class of iterative methods when are seen in the real context are not of interest because they are inefficient. Multi-step iterative methods are good candidates to qualify as efficient solvers. The multi-step iterative methods consist of two parts namely base method that contains the Jacobian and its LU-factors information and the multi-step part where we use LU-factor information to solve the system of linear equations. The efficiency of multi-step iterative methods is hidden in the reuse of Jacobian information in the multi-step part. For each multi-step, we get an enhancement in the convergence order at the cost of single evaluation of nonlinear function along with the solution of systems of linear equations and matrix–vector multiplications. The development of iterative solvers for solving nonlinear ordinary differential equations (ODEs) and partial differential equations (PDEs) is always of interest. The discretization of nonlinear ODEs and PDEs provides the system of nonlinear equations that are known as weakly nonlinear systems. So the solutions of weakly nonlinear systems of equations are also the topic of research in this article.

When we talk about iterative solvers for solving systems of nonlinear equations the most famous and classical method come into mind is the Newton method [1,2] and its various variants. Let $\mathbf{F}(\mathbf{y}) = [f_1(\mathbf{y}), f_2(\mathbf{y}), \ldots, f_n(\mathbf{y})]^T = \mathbf{0}$ be system of nonlinear equations and Newton method can be stated as

$$\text{NR} = \begin{cases} \mathbf{y}_0 = \text{initial guess} \\ \mathbf{y}_{n+1} = \mathbf{y}_n - \mathbf{F}'(\mathbf{y}_n)^{-1}\mathbf{F}(\mathbf{y}_n), \end{cases}$$

where $\det\left(\mathbf{F}'(\mathbf{y}_n)\right) \neq 0$. The convergence order of Newton method is two. The Newton method is computationally expensive because we compute Jacobian and its inverse in each iteration. The multi-step version of Newton method [3] can be written as

$$\text{MNR} = \begin{cases} \text{Number of steps} & = m \geq 1 \\ \text{Convergence order} & = m+1 \\ \text{Function evaluations} & = m \\ \text{Jacobian evaluations} & = 1 \\ \text{Number of LU-factorization} & = 1 \\ \text{Number of solutions of lower} & \\ \text{and upper triangular systems} & = m \end{cases} \begin{cases} \text{Base method} \rightarrow & \begin{cases} \mathbf{y}_0 = \text{initial guess} \\ \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_1 = \mathbf{F}(\mathbf{y}_0) \\ \mathbf{y}_1 = \mathbf{y}_0 - \boldsymbol{\phi}_1 \end{cases} \\ \text{Multi-step part} \rightarrow & \begin{cases} \text{for } s = 1, m-1 \\ \quad \mathbf{F}'(\mathbf{y}_0)\,\boldsymbol{\phi}_{s+1} = \mathbf{F}(\mathbf{y}_s) \\ \quad \mathbf{y}_{s+1} = \mathbf{y}_s - \boldsymbol{\phi}_{s+1} \\ \text{end} \\ \mathbf{y}_0 = \mathbf{y}_m, \end{cases} \end{cases}$$

where $\det\left(\mathbf{F}'(\mathbf{y}_n)\right) \neq 0$. MNR method is more efficient than NR method because its single instance requires one Jacobian evaluation and its inversion in the form of LU factors. The convergence order of MNR is $m+1$ which is low if we compare it with number of function evaluations, so we have enough room for improvement in the convergence order compared to function evaluations. Other classical iterative methods are Halley [4,5] and Chebyshev [6] methods which are respectively

$$\text{Halley Method} = \begin{cases} \mathbf{y}_n = \text{initial guess} \\ \mathbf{F}'(\mathbf{y}_n)\,\boldsymbol{\phi} = \mathbf{F}(\mathbf{y}_n) \\ \mathbf{F}'(\mathbf{y}_n)\,\mathbf{L}(\mathbf{y}_n) = \mathbf{F}''(\mathbf{y}_n)\boldsymbol{\phi} \\ \mathbf{y}_{n+1} = \mathbf{y}_n - \left[I - \frac{1}{2}\mathbf{L}(\mathbf{y}_n)\right]^{-1}\boldsymbol{\phi} \end{cases}$$

and

$$\text{Chebyshev Method} = \begin{cases} \mathbf{y}_n = \text{initial guess} \\ \mathbf{F}'(\mathbf{y}_n)\,\boldsymbol{\phi} = \mathbf{F}(\mathbf{y}_n) \\ \mathbf{F}'(\mathbf{y}_n)\,\mathbf{L}(\mathbf{y}_n) = \mathbf{F}''(\mathbf{y}_n)\boldsymbol{\phi} \\ \mathbf{y}_{n+1} = \mathbf{y}_n - \left[I + \frac{1}{2}\mathbf{L}(\mathbf{y}_n)\right]\boldsymbol{\phi}. \end{cases}$$

The order of convergence of both iterative methods is three. The Halley method requires two inversions so it is an expensive method. The Chebyshev method requires one Jacobian inversion, but the computation cost of second order Fréchet derivative is high for a general system of nonlinear equations. There are also further classical iterative methods for solving system of nonlinear equations, for instance the super-Halley method and the Euler–Chebyshev method. These methods are less efficient, since they are characterized by low convergence order combined with a high computational cost. Another set of one-point iterative methods for solving a system of nonlinear equations is described below:

$$\begin{array}{ll} \mathbf{y}_{n+1} = \mathbf{y}_n - \mathbf{t}_1 & \text{2nd order Newton–Raphson method} \\ \mathbf{y}_{n+1} = \mathbf{y}_n - \mathbf{t}_1 - \mathbf{t}_2 & \text{3rd order Chebyshev method} \\ \mathbf{y}_{n+1} = \mathbf{y}_n - \mathbf{t}_1 - \mathbf{t}_2 - \mathbf{t}_3 & \text{4th order} \\ \mathbf{y}_{n+1} = \mathbf{y}_n - \mathbf{t}_1 - \mathbf{t}_2 - \mathbf{t}_3 - \mathbf{t}_4 & \text{5th order} \\ \mathbf{y}_{n+1} = \mathbf{y}_n - \mathbf{t}_1 - \mathbf{t}_2 - \mathbf{t}_3 - \mathbf{t}_4 - \mathbf{t}_5 & \text{6th order,} \end{array} \tag{1}$$

where $\mathbf{C}_j = \frac{1}{j!}\mathbf{F}'(\mathbf{y}_n)^{-1}\mathbf{F}^{(j)}(\mathbf{y}_n)$ $\mathbf{t}_1 = \mathbf{F}'(\mathbf{y}_n)^{-1}\mathbf{F}(\mathbf{y}_n)$, $\mathbf{t}_2 = \mathbf{C}_2\mathbf{t}_1^2$, $\mathbf{t}_3 = \left(-\mathbf{C}_3 + 2\mathbf{C}_2^2\right)\mathbf{t}_1^3$, $\mathbf{t}_4 = \left(-5\mathbf{C}_3\mathbf{C}_2 + 5\mathbf{C}_2^3 + \mathbf{C}_4\right)\mathbf{t}_1^4$, $\mathbf{t}_5 = \left(-\mathbf{C}_5 + 3\mathbf{C}_3^2 + 14\mathbf{C}_2^4 - 21\mathbf{C}_3\mathbf{C}_2^2 + 6\mathbf{C}_4\mathbf{C}_2\right)\mathbf{t}_1^5$. The high computational cost of the iterative methods displayed in (1) makes