SoftwareX 7 (2018) 59–62

Contents lists available at ScienceDirect

SoftwareX

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

Original software publication

SIR—An efficient solver for systems of equations

Jan Scheffel, Kristoffer Lindvall *

Department of Fusion Plasma Physics, School of Electrical Engineering and Computer Science, KTH Royal Institute of Technology, Stockholm, Sweden

ARTICLE INFO

Article history:
Received 17 May 2017
Received in revised form 15 January 2018
Accepted 15 January 2018

Keywords:
Newton method
Root solver
Equation solver
MATLAB

ABSTRACT

The Semi-Implicit Root solver (SIR) is an iterative method for globally convergent solution of systems of nonlinear equations. We here present MATLAB and MAPLE codes for SIR, that can be easily implemented in any application where linear or nonlinear systems of equations need be solved efficiently. The codes employ recently developed efficient sparse matrix algorithms and improved numerical differentiation. SIR convergence is quasi-monotonous and approaches second order in the proximity of the real roots. Global convergence is usually superior to that of Newton's method, being a special case of the method. Furthermore the algorithm cannot land on local minima, as may be the case for Newton's method with line search.

© 2018 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).

Code metadata

Current software version	v1.0
Permanent link to code/repository used for this code version	https://github.com/kfli/SIR
Legal Code License	BSD 3-Clause, see http://opensource.org/licenses/BSD-3-Clause
Software code languages, tools, and services used	MATLAB® and MAPLE™
Support email for questions	kfli@kth.se

1. Motivation and significance

Systems of algebraic equations are generally solved computationally, whether it be with direct methods or iterative methods. The Semi-Implicit Root solver (SIR [1]), reported on here, was developed in order to improve on the global convergence characteristics of the widely used Newton method. Line search [2] is often combined with Newton's method to improve convergence but, unlike SIR, it may lead to local extrema rather than to the roots of the equations. SIR development was initially inspired by semi-implicit partial differential equation (PDE) algorithms [3–5] and evolved as a robust equation solver for the time-spectral method GWRM [6] for systems of PDEs; it is, however, generally applicable to systems of equations.

Recently the SIR algorithm has been made more efficient by using current software (MATLAB and MAPLE) versions, along with optimized programming to fit both software platforms. This includes correct matrix handling and the use of numeric instead of symbolic arithmetic.

First, a brief overview of SIR is provided in Section 2. For a thorough explanation of the SIR algorithm, the reader is advised to consult [1]. An example application is given in Section 3. Pseudocode, describing the algorithm in detail, can be found in Section 4.

2. Software description

The roots of the *single equation* $f(x) = 0$, with $f(x) \equiv x - \varphi(x)$, are found by SIR after a reformulation in iterative form as

$$x^{i+1} + \beta x^{i+1} = \beta x^i + \varphi(x^i), \tag{1}$$

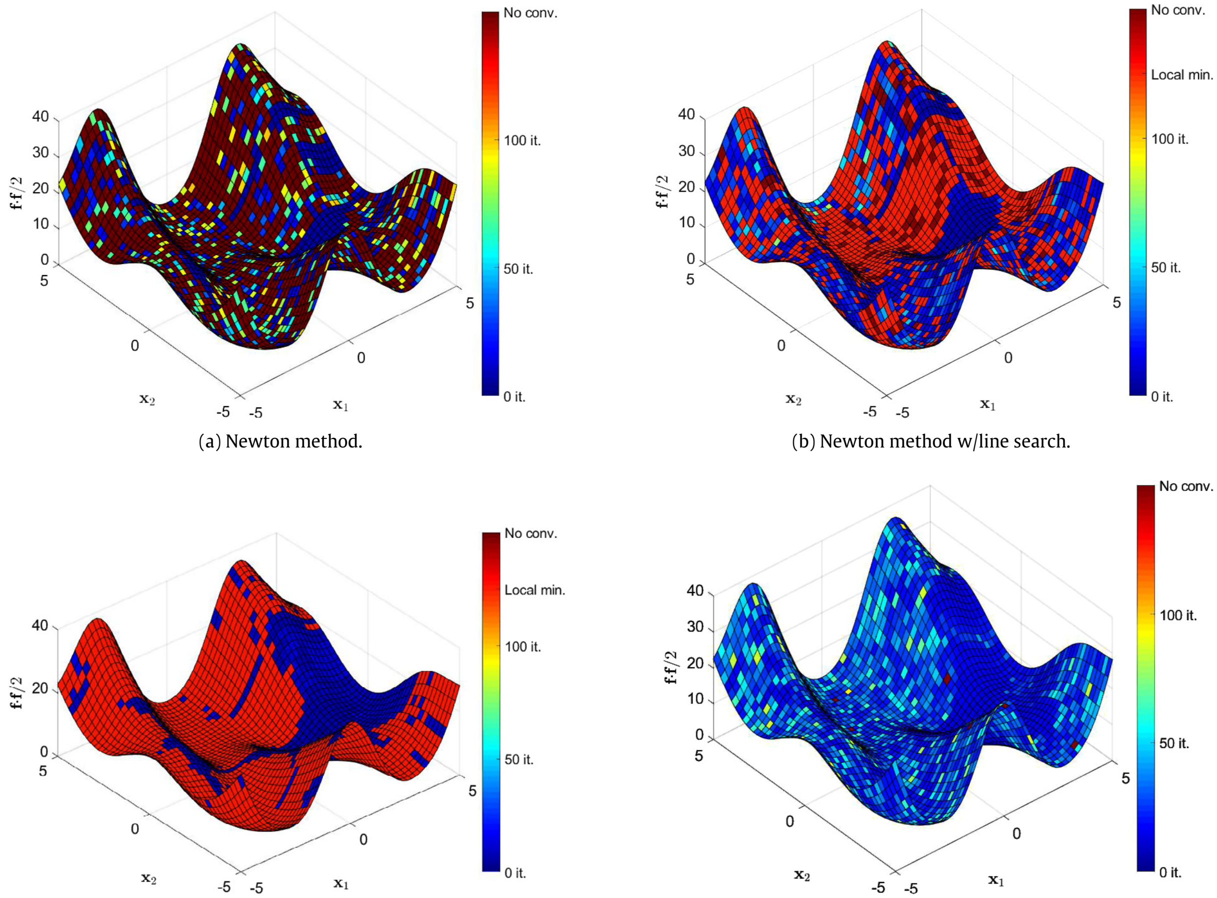
where β is a real parameter. Eq. (1) will have the same roots as the original equation. We cast it into the form

$$x^{i+1} = \alpha(x^i - \varphi(x^i)) + \varphi(x^i), \tag{2}$$

where $\alpha = \beta/(1+\beta)$ is a parameter for optimizing global and local convergence. Introducing $\Phi(x;\alpha) \equiv \alpha(x - \varphi(x)) + \varphi(x)$ it may be shown that convergence requires $|\partial\Phi/\partial x| < 1$ to hold for the iterates x^i in the neighbourhood of the root [1]. Newton's method (for a single equation also known as Newton–Raphson's method) assumes $|\partial\Phi/\partial x| = 0$ everywhere, thus potentially achieving maximized, second order convergence near the root.

* Corresponding author.
E-mail address: kfli@kth.se (K. Lindvall).

https://doi.org/10.1016/j.softx.2018.01.003
2352-7110/© 2018 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY license (http://creativecommons.org/licenses/by/4.0/).



(a) Newton method.

(b) Newton method w/line search.

(c) *fsolve* w/Levenberg–Marquardt method.

(d) SIR-s.

Fig. 1. Convergence diagrams of the (a) Newton method, (b) Newton method with line search, (c) MATLAB in-built solver *fsolve*, and (d) SIR with subiterations. A set of 51 by 51 uniformly distributed $x^0 \in$ [-5,5] is colour-mapped according to the amount of root solver iterations. The two equations solved are $f_1 = x_1 - \cos(x_2)$ and $f_2 = x_2 - 3\cos(x_1)$, where $\|\mathbf{f}\|^2/2 = (f_1^2 + f_2^2)/2$. The solution is $x^* = [-0.6843, 2.3245]$.

Newton's method is, however, not globally convergent because of the breakdown of the linear approximation for initial iterates x^0 positioned too distant from the root. This problem is remedied by SIR through enforcing monotonic convergence. The trick is to choose appropriate values of $|\partial\Phi/\partial x| \equiv R$ at each iteration i.

Thus SIR iterates the equation

$$x^{i+1} = \alpha^i (x^i - \varphi(x^i)) + \varphi(x^i), \tag{3}$$

using

$$\alpha^i = \frac{R^i - \varphi'(x^i)}{1 - \varphi'(x^i)}. \tag{4}$$

where $\varphi'(x) = d\varphi/dx$. Typically R^i is initially given a value in the interval $[0.5, 0.99]$ whereafter SIR automatically reduces it towards zero to achieve second order convergence in the vicinity of the root. Since monotonic convergence is guaranteed, SIR will consecutively find all real roots x to the equation.

Generalizing to *systems of equations*, SIR now solves

$$\mathbf{x}^{i+1} = \mathbf{A}^i(\mathbf{x}^i - \varphi(\mathbf{x}^i)) + \varphi(\mathbf{x}^i), \tag{5}$$

where

$$\mathbf{A} = \mathbf{I} + (\mathbf{R} - \mathbf{I})\mathbf{J}^{-1}, \tag{6}$$

$$(\mathbf{R})_{mn} = \delta_{mn} R_m. \tag{7}$$

Here the Jacobian matrix $\mathbf{J}$ has components $J_{mn} = \partial(x_m - \varphi_m(\mathbf{x}))/\partial x_n$, δ_{mn} is the Kronecker delta and $\mathbf{I}$ is the identity matrix. It is usually most economic to obtain $\mathbf{A}$ by solving the linear matrix relation

$$\Phi' = (\mathbf{A} - \mathbf{I})\mathbf{J} + \mathbf{I}, \tag{8}$$

using $\partial\Phi_m/\partial x_n = R_{mn}$ (diagonal matrix) and sparse matrix methods.

When solving multidimensional equations, strict monotonous convergence cannot be guaranteed, since there is no known effective procedure (like root bracketing in the 1D case) to safely maintain the direction from a starting point towards a root. A procedure for "quasi-monotonicity" is thus employed in SIR [1]. SIR is also safeguarded against certain pitfalls. An evident problem can be seen in Eq. (4); there is a risk that the $\mathbf{A}$ matrix may become singular. The cure for this is subiteration, where new values of R_m towards unity are chosen. See [1] for further discussion of measures that enhance convergence.

Summarizing, at each iteration the SIR algorithm reduces R_m values towards zero to approach second order convergence. If non-monotonous convergence becomes pronounced in any dimension, local subiteration by increasing R_m values towards unity is used.

3. Illustrative example

SIR has been compared to Newton's method with line search (NL) for a large set of standard problems [1]. In several cases it

Download English Version:

https://daneshyari.com/en/article/6964838

Download Persian Version:

https://daneshyari.com/article/6964838

Daneshyari.com