How to construct efficient solvers for nonlinear equations?

Topics in Computational Mathematics

F. Soleymani

Presentation at the IASBS, Zanjan, Iran, October 2024

Contents

Stating the problem and classifications

- Following the classifications of Traub in [11], iterations to calculate the solution of nonlinear equations the form

$$f(x) = 0,$$

  are categorized into two major portions of '**methods without memory**' and '**methods with memory**'.

A hypothesis

- On the other hand, in 1974, Kung and Traub in [7] proposed a **hypothesis** indicating that an iteration method without memory consisting of m functional evaluations could get at most 2^{m-1} order of convergence to find simple zeros for general functions.
- Taking this concept into account many iteration methods have been developed and proposed to the literature, [1].

Steffensen's solver

- For solving nonlinear scalar equations of the form $f(x) = 0$, Steffensen in [10] proposed an iteration method (SM), which does not consist of any derivative evaluations per cycle to proceed as follows:

$$x_{k+1} = x_k - \frac{f(x_k)}{f[x_k, w_k]}, \quad \beta \in \mathbb{R}\backslash\{0\}, \; k = 0, 1, 2, \cdots, \qquad (1)$$

  wherein $w_k = x_k + \beta f(x_k)$.

- Though originally in [10], (1) does not contain a nonzero free parameter β, its incorporation as in (1) preserves the rate of convergence and gives us a family of iterations useful for nonlinear problems, at which the computation of derivatives is hard.
- The iteration (1) satisfies Kung-Traub conjecture.

Efficiency index

- One of the ways in order to distinguish the best iteration schemes for solving equations is to rely on the **efficiency index** given by [8]:

$$EI = p^{\frac{1}{\eta}}, \tag{2}$$

  where p and η stand for the convergence speed and the evaluations in terms of functions.

- Furthermore, **attraction basins** give another criterion for choosing the best iteration when the main concentration is to have better convergence regions with less sensitive areas for selecting the choices of the **initial guesses**.

Another recent solver

- A recently well–developed Steffensen–type scheme (HM) is given by [5]:

$$\begin{cases} w_k = x_k - \beta f(x_k), \qquad \beta \in \mathbb{R}\backslash\{0\}, \qquad k \geq 0, \\ x_{k+1} = x_k - \dfrac{f(x_k)}{f[x_k, w_k]} \left(1 + \xi \frac{f(w_k)}{f[x_k, w_k]}\right), \ \xi \in \mathbb{R}. \end{cases} \tag{3}$$

The method without memory (3) reaches the second order convergence just like (1) but with one more free parameter.

With memorization

- The concept of methods with memory is to keep the number of functional evaluations **unchanged**, but to also save the computed values and apply them in an interpolation–based process to **accelerate** the rate of convergence.
- Thus, this acceleration is attained *without any additional functional evaluations*, which make them not only interesting in terms of the convergence speed but also in terms of the computational *efficiency index*, [2].

Other state-of-the-art solvers

- In [3], the author discussed a two–step method with memory without the use of any functional derivatives as follows:

$$\begin{cases} v_k = x_k - \beta_k f(x_k), \qquad \beta_k = -\dfrac{1}{N'(x_k)}, \\ y_k = x_k - \dfrac{f(x_k)}{f[x_k, v_k]}, \\ x_{k+1} = y_k - \dfrac{f(y_k)}{f[x_k, v_k]}. \end{cases} \tag{4}$$

Here the function $N'(x_k)$ is a suitable function. This methods has two steps and include three functional evaluations per iteration with 3.37 R–order of convergence.

A pioneer solver

Speaking of methods with memory, Džunić in [4] proposed a one–step iteration expression with memory (DM) including two parameters and reaching $\frac{1}{2}(3+\sqrt{17})$ R–order of convergence as comes next:

$$\begin{cases} w_k = x_k + \beta_k f(x_k), \\ \beta_k = -\dfrac{1}{N_2'(x_k)}, \qquad p_k = -\dfrac{N_3''(w_k)}{2N_3'(w_k)}, \qquad k \geq 1, \\ x_{k+1} = x_k - \dfrac{f(x_k)}{f[x_k, w_k] + p_k f(w_k)}, \qquad k \geq 0, \end{cases} \tag{5}$$

wherein $N_j(l)$ is the Newton's interpolation function of j-th order going through $j+1$ nodes at the point l. As an illustration, we may consider $N_3(t)$ as the Newton's interpolation polynomial of 3rd degree, setting through 4 existing estimates $x_k, w_k, w_{k-1}, x_{k-1}$.

Another state-of-the-art solver

- The author in [6] presented the following iterative scheme with memory (KM):

$$\begin{cases} \beta_k = -\frac{1}{N_4'(x_k)}, \qquad p_k = -\frac{N_5''(w_k)}{2N_5'(w_k)}, \qquad k \geq 2, \\ w_k = x_k + \beta_k f(x_k), \\ x_{k+1} = x_k - \dfrac{f(x_k)}{f[x_k, w_k] + p_k f(w_k)}, \end{cases} \tag{6}$$

  which hits the convergence R–order 3.90057 and is a one–step method.

- To get the highest rate of convergence and efficiency index, without any additional functional evaluation, we rely on introducing approximation for the involved parameters. Hence, the increasing R–order is related to one or more accelerator parameters in the error equation of the method.

How to enhance the solvers?

We approximate the accelerator parameters in each cycle by Newton's interpolating polynomials passing through the *best* saved points. Here, the rate of convergence is increased from 2 to roughly 4 and the efficiency index is significantly optimized from 1.41421 to roughly 2, which is highest possible index of computational efficiency.

How to enhance the solvers?

We investigate a modification of (1) so as to get the quadratical convergence rate but with having two free nonzero parameters as follows:

$$\begin{cases} w_k = x_k + \beta f(x_k), \qquad \beta \in \mathbb{R}\backslash\{0\}, \qquad k \geq 0, \\ x_{k+1} = x_k - \dfrac{f(x_k)}{f[x_k, w_k]} \left(1 - p\frac{f(w_k)}{f[x_k, w_k]}\right), \;\; p \in \mathbb{R}. \end{cases} \tag{7}$$

A theorem

Theorem

If $f(x)$ is sufficiently smooth in a neighborhood of its simple root α and a starting guess x_0 is close enough to α. Therefore, the convergence speed of (7) without memory is quadratic.

$$\begin{aligned} e_{k+1} = &(c_2 + p)\,(\beta f'(\alpha) + 1)e_k^2 \\ &+ O(e_k^3). \end{aligned} \tag{8}$$

The error equation (8) reveals that the convergence speed may pass the quadratic level and even more than cubic, if we had the ability to replace $p = -c_2$ and $\beta = -1/f'(\alpha)$.

Self-accelerators

- Although the value of the zero is not obvious/given, but through the already computed values, we can approximate these parameters as if the R–order of convergence increases. In fact, we now develop a method with memory that uses the information not only from the last step, but also from the previous iterations (as much as needed).
- This technique enables us to achieve the highest efficiency both theoretically and practically. Accordingly, let us impose the following approximations recursively for the free parameters:

$$\beta = \beta_k, \quad p = p_k, \tag{9}$$

as the iterative scheme goes on via $\beta_k = -\frac{1}{\bar{f}'(\alpha)}$ and $p_k = -\bar{c}_2$, wherein $\bar{f}'(\alpha)$ and $\bar{c}_2$ are estimates to $f(\alpha)$ and c_2, respectively.

A method with memory

Now, we can propose an iteration schemes (PM1) with memory as follows:

$$\begin{cases} \beta_k = -\frac{1}{N_2'(x_k)}, \qquad p_k = -\frac{N_3''(w_k)}{2N_3'(w_k)}, \qquad k \geq 1, \\ w_k = x_k + \beta_k f(x_k), \qquad k \geq 0, \\ x_{k+1} = x_k - \dfrac{f(x_k)}{f[x_k, w_k]} \left(1 + p_k \frac{f(w_k)}{f[x_k, w_k]}\right), \qquad k \geq 0. \end{cases} \tag{10}$$

Other variants

Similarly the following ones with better interpolation degrees:

$$\begin{cases} \beta_k = -\frac{1}{N_4'(x_k)}, \qquad p_k = -\frac{N_5''(w_k)}{2N_5'(w_k)}, \qquad k \geq 2, \\ w_k = x_k + \beta_k f(x_k), \qquad k \geq 0, \\ x_{k+1} = x_k - \dfrac{f(x_k)}{f[x_k, w_k]} \left(1 + p_k \frac{f(w_k)}{f[x_k, w_k]}\right), \qquad k \geq 0, \end{cases} \tag{11}$$

and

$$\begin{cases} \beta_k = -\frac{1}{N_6'(x_k)}, \qquad p_k = -\frac{N_7''(w_k)}{2N_7'(w_k)}, \qquad k \geq 3, \\ w_k = x_k + \beta_k f(x_k), \qquad k \geq 0, \\ x_{k+1} = x_k - \dfrac{f(x_k)}{f[x_k, w_k]} \left(1 + p_k \frac{f(w_k)}{f[x_k, w_k]}\right), \qquad k \geq 0. \end{cases} \tag{12}$$

Interpolation functions of different degrees

As an illustration, here we also can define:

- $N_4(t) = \mathcal{N}_4(t; x_k, x_{k-1}, w_{k-1}, x_{k-2}, w_{k-2})$, as an interpolation polynomial of fourth degree, passing through the best five saved points x_k, x_{k-1}, w_{k-1}, w_{k-2}, x_{k-2}, for any $k \geq 2$.
- $N_5(t) = \mathcal{N}_5(t; w_k, x_k, x_{k-1}, w_{k-1}, x_{k-2}, w_{k-2})$, as an interpolation polynomial of fifth degree, passing through the best six saved points w_k, x_k, x_{k-1}, w_{k-1}, w_{k-2}, x_{k-2}, for any $k \geq 2$.

The convergence order in with memorization scenario

Theorem

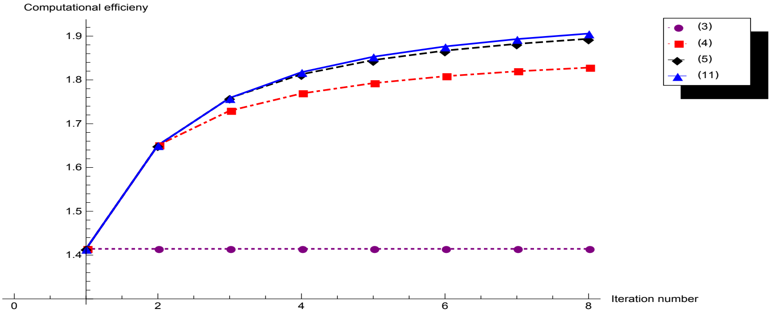
Consider the same assumptions as in Theorem 1. Then, convergence R-order of the improved Steffensen's method with memory (12) is 3.97609*.*

1. This shows that the one step scheme could reach the highest possible R–order four using only two functional evaluations per cycle by applying the approach of with memorization. This means 100% improvement over the methods without memory.

Observation for the enhancement

1. To manifest the improvement of (10)–(12) in terms of computational efficiency index, in Figure 1, a comparison among various iterative schemes is provided. The improvement in R–order and the efficiency indices for (10)–(12) is obvious.

Results: parallelization



A comparison of various iteration methods in terms of the computational efficiency indices after performing several cycles.

An inquiry

A question may arise that how this procedure can be generalized to produce a family of iteration methods with memory. To respond this, we emphasize that the degree of the interpolation polynomial can be increased two units if we make the updating process after one more iterations. Hence, by increasing the interpolation degree, the R–order will increase but never passes the quadratic speed of convergence:

$$\begin{cases} \beta_k = -\frac{1}{N'_{2l}(x_k)}, \qquad p_k = -\frac{N''_{2l+1}(w_k)}{2N'_{2l+1}(w_k)}, \qquad k \geq l, \\ w_k = x_k + \beta_k f(x_k), \qquad k \geq 0, \\ x_{k+1} = x_k - \dfrac{f(x_k)}{f[x_k, w_k]} \left(1 + p_k \frac{f(w_k)}{f[x_k,w_k]}\right), \qquad k \geq 0. \end{cases} \tag{13}$$

Basins of attractions

- When analyzing particular methods the structure has to be taken into account, and the efficiency of iterations cannot be reduced solely to convergence speed and informational volume, [9]. Hence, it is necessary to study some other aspects of iteration methods for such a task. An important criterion which worth investigating is to show how the schemes are useful in terms of the **freedom in the choice of the starting guess**.
- To check the stability and usefulness of different iterative methods, investigating the dynamical behavior of such iterations is necessary.

Considerations 1

- We used $\beta_0 = 10^{-6}$ whenever required unless stated clearly. The attraction basins for several polynomials

$$f(z) = z^n - 1, \;\; n = 2, 3, 4, 5,$$

having complex roots are furnished. The other involved parameters are set to zero.

Considerations 2

- To calculate and plot the attraction basins for the roots of a polynomial applying an iteration scheme, we consider a grid of points in the rectangle

$$D = [-4.0, 4.0] \times [-4.0, 4.0] \subset \mathbb{C},$$

and we use these points as z_0. If the iterates produced via the iteration scheme converges a zero α of the polynomial applying the condition

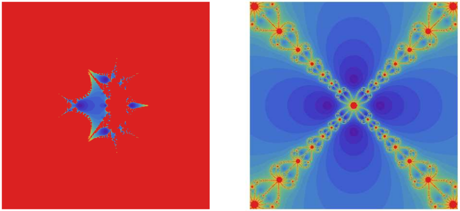
$$|f(z_k)| < 10^{-2},$$

and a maximum of 30 iterations, we decide that z_0 is in the attraction basin of the zero and we paint this point.

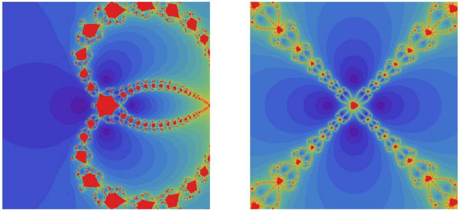
Notes

- Here, wherever fewer number of iterates are used to converge, darker colors are applied. It means darker areas show convergence in fewer iterates while lighter area show that for those starting points, one need more number of iterates to converge.
- Red color denotes lack of convergence to any of the roots (with the maximum of iterations established) or convergence to the infinity.
- The dynamical results given here show that the convergence radius of the proposed variants of the Steffensen's method can be improved by updating the self–accelerating parameters.

Results and fractals 1

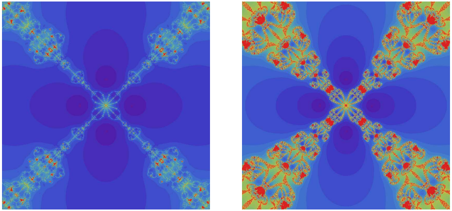


Attraction basins for (1) with $\beta = 1$ (left) and $\beta = 10^{-6}$ (right) for a quartic polynomial.

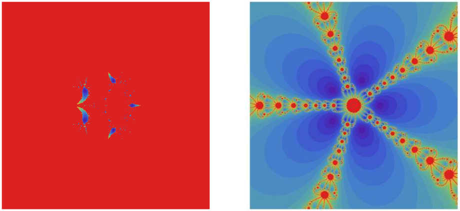


Attraction basins for (7) with $\beta_0 = 10^{-6}$, $p = 1$ (left) and (7) with $\beta_0 = 10^{-6}$, $p = 0.01$ (right) for a quartic polynomial.

Results and fractals 2

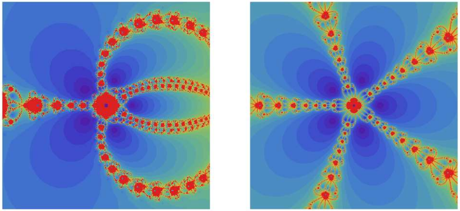


Attraction basins for (6) with $\beta_0 = 10^{-6}$, $p_0 = 0$ (left) and (10) with $\beta_0 = 10^{-6}$, $p_0 = 0$ (right) for a quartic polynomial.

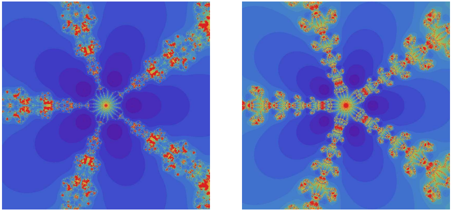


Attraction basins for (1) with $\beta = 1$ (left) and $\beta = 10^{-6}$ (right) for a quintic polynomial.

Results and fractals 3



Attraction basins for (7) with $\beta_0 = 10^{-6}$, $p = 1$ (left) and (7) with $\beta_0 = 10^{-6}$, $p = 0.01$ (right) for a quintic polynomial.



Attraction basins for (6) with $\beta_0 = 10^{-6}$, $p_0 = 0$ (left) and (10) with $\beta_0 = 10^{-6}$, $p_0 = 0$ (right) for a quintic polynomial.

Results

Experiment 4. *The last test problem is taken into consideration as follows:*

$$f_4(x) = \tan^{-1}(\exp(x+2)+1) + \tanh(\exp(-x\cos(x))) - \sin(\pi x),$$

where $\alpha \approx -3.6323572\cdots$ *and* $x_0 = -4.1$.

$$coc = \frac{\ln|(f(x_k)/f(x_{k-1})|}{\ln|(f(x_{k-1})/f(x_{k-2})|}$$

Results

Table 4. Result of comparisons for the function f_4.

Methods	$\lvert f_4(x_3) \rvert$	$\lvert f_4(x_4) \rvert$	$\lvert f_4(x_5) \rvert$	$\lvert f_4(x_6) \rvert$	coc
SM	0.00001166	3.7123×10^{-10}	3.7616×10^{-19}	3.8622×10^{-37}	2.00
DZ	1.6×10^{-13}	6.9981×10^{-47}	1.0583×10^{-164}	7.0664×10^{-585}	3.57
PM	3.0531×10^{-11}	3.2196×10^{-38}	3.7357×10^{-134}	6.5771×10^{-476}	3.56
M4	2.5268×10^{-13}	1.5972×10^{-49}	2.8738×10^{-191}	1.6018×10^{-744}	3.90

Conclusions and discussions

- The condition $f'(x) \neq 0$ in a neighborhood of the required root is sometimes severe for convergence of Newton–type methods, which accordingly restricts their implementations in some practical problems.
- The improvement of the R–order for the proposed variant is 100%, which is the highest possible convergence acceleration for iterative methods with memory.
- The attraction basins showed not only the acceleration of the convergence speed but also improved convergence radii.

References I

- F. Ahmad, E. Tohidi, M. Zaka Ullah, J.A. Carrasco, Higher order multi–step Jarratt–like method for solving systems of nonlinear equations: Application to PDEs and ODEs, Comput. Math. Appl. 70 (2015), 624–636.
- H. Arora, J.R. Torregrosa, A. Cordero, Modified Potra–Pták multi-step schemes with accelerated order of convergence for solving systems of nonlinear equations, Math. Comput. Appl. 24 (2019), 1–15.
- F.I. Chicharro, A. Cordero, N. Garrido, J.R. Torregrosa, Stability and applicability of iterative methods with memory, J. Math. Chem., (2018), 1–19.
- J. Džunić, M.S. Petković, On generalized biparametric multipoint root finding methods with memory, J. Comput. Appl. Math., 255 (2014), 362–375.
- F. Khaksar Haghani, A modified Steffensen's method with memory for nonlinear equations, Int. J. Math. Model. Comput., 5 (2015), 41–48.
- F. Kiyoumarsi, On the construction of fast Steffensen–type iterative methods for nonlinear equations, Int. J. Comput. Meth., 15 (2018), Art. ID: 1850002.
- H.T. Kung, J.F. Traub, Optimal order of one–point and multi–point iteration, J. ACM, 21 (1974), 643–651.
- A.M. Ostrowski, Solution of equations and systems of equations, Academic Press, New York, 1966.
- J.R. Sharma, I.K. Argyros, S. Kumar, Ball convergence of an efficient eighth order iterative method under weak conditions, Mathematics, 6 (2018), Article ID: 260, 1–8.
- J.F. Steffensen, Remarks on iteration, Skand. Aktuarietidskr, 16 (1933), 64–72.
- J.F. Traub, Iterative Methods for the Solution of Equations, Prentice Hall, New York, 1964.


THANK YOU
THAT'S IT
FOR NOW...
Why don't you?