Examples and Counterexamples xxx (xxxx) xxx

Contents lists available at ScienceDirect

Examples and Counterexamples

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

Halving interval guaranteed for Dekker and Brent root finding methods

Vilmar Steffen [a,e,*], Carlos Catusso Della Pasqua [b], Maiquiel Schmidt de Oliveira [c], Edson Antonio da Silva [d,e]

[a] Academic Departments of Engineering (DAENG), Federal University of Technology - Parana (UTFPR), Rua Gelindo Joao Folador, 2000, Francisco Beltrao, 85602-863, Parana, Brazil
[b] Chemical Engineering Corse, Federal University of Technology - Parana (UTFPR), Rua Gelindo Joao Folador, 2000, Francisco Beltrao, 85602-863, Parana, Brazil
[c] Academic Department of Physics, Statistics and Mathematics (DAFEM), Federal University of Technology - Parana (UTFPR), Rua Gelindo Joao Folador, 2000, Francisco Beltrao, 85602-863, Parana, Brazil
[d] Center for Engineering and Exact Sciences (CECE), State University of Paraná (Unioeste), Rua da Faculdade, 645, Toledo, 85903-000, Parana, Brazil
[e] Postgraduate Program in Chemical Engineering, State University of Paraná (Unioeste), Rua da Faculdade, 645, Toledo, 85903-000, Parana, Brazil

ARTICLE INFO

Communicated by A. Heinlein

Keywords:
Root finding
Interval halving
Dekker method
Brent method
Hybrid method

ABSTRACT

Hybrid methods are widely used in many areas of applied mathematics. One of the simplest and most common problems in this field is root finding, for which various methods exist. Some of the most efficient approaches combine two or more techniques into hybrid methods. Among these are the Dekker and Brent methods, for which we propose a modification to ensure that the search interval is halved in each iteration. We apply this modification to two examples: a transcendental equation and a cubic equation of state. The results demonstrate that the proposed modifications guarantee at least interval halving and offer a slight improvement in the efficiency of the root-finding process.

1. Introduction

The use of numerical methods is common in scientific studies [1–7], particularly for solving the critical problem of locating the real root of a nonlinear algebraic equation ($f(x) = 0$) [8,9]. The literature presents a wide array of root-finding methods, both analytical and numerical [10], each with its own advantages and disadvantages. While analytical solutions for algebraic equations are limited – primarily applicable to quadratic, cubic, and quartic equations [11–15] – most equations require numerical methods for their resolution.

No single numerical root-finding method excels for all functions [13]. Consequently, hybrid methods have been proposed to leverage the strengths of various approaches [16]. Among these, the Dekker method stands out for combining the convergence speed of the secant method with the reliability of the bisection method, as well as incorporating inverse quadratic interpolation to accelerate convergence. However, in certain iterations and for specific functions, this method may exhibit poorer interval reduction than the bisection method.

This work aims to propose simple modifications to the original Dekker and Brent methods [17,18], ensuring that the interval containing the function's root is at least halved in each iteration. Section 2 provides a detailed description of how the original methods are adapted to guarantee this interval reduction. Section 3 compares the performance of the original and modified methods through two examples, and finally, Section 4 presents the conclusions and future directions regarding the proposed improvements.

2. Methods

This section presents the proposed modifications to the Dekker and Brent methods.

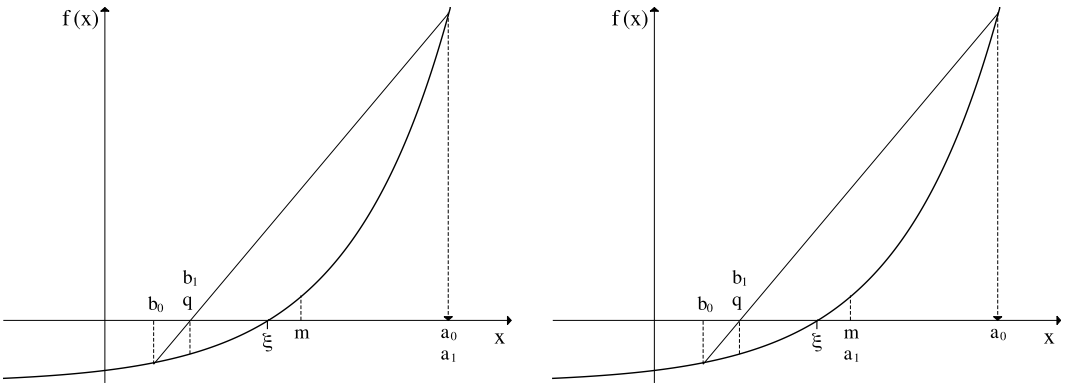
2.1. Dekker method

The Dekker method combines the guaranteed convergence of the bisection method with the efficiency of the secant method. In each iteration, the secant method is used unless the bisection method is deemed more appropriate.

Considering a continuous function $f(x)$ over the initial interval $[a_0, b_0]$, such that there is at least one root (ξ) within this interval, *i.e.*, $f(a_0)\, f(b_0) < 0$. In each iteration, two approximations of the root are

* Corresponding author at: Academic Departments of Engineering (DAENG), Federal University of Technology - Parana (UTFPR), Rua Gelindo Joao Folador, 2000, Francisco Beltrao, 85602-863, Parana, Brazil.
E-mail address: vilmars@utfpr.edu.br (V. Steffen).

https://doi.org/10.1016/j.exco.2024.100173
Received 22 January 2024; Received in revised form 25 September 2024; Accepted 16 December 2024
Available online 24 December 2024



(a) Dekker method (b) Modified Dekker method

Fig. 1. Graphical illustration of Dekker method.

calculated, namely, q using the secant method formula with Eq. (1) and m using the bisection method formula with Eq. (2).

$$q = b_k - f\left(b_k\right)\frac{b_{k-1} - b_k}{f\left(b_{k-1}\right) - f\left(b_k\right)}. \tag{1}$$

$$m = \frac{a_k + b_k}{2}. \tag{2}$$

In the first iteration $b_k = b_0$ and $b_{k-1} = a_0$ for the secant method. In the subsequent iteration, b_{k+1} is set to q if q falls between m and b_k, otherwise, b_{k+1} is set to m. The value of a_{k+1} can be chosen as either a_k or b_k, with the selection made to ensure that $f\left(a_{k+1}\right) f\left(b_{k+1}\right) < 0$. After this, the values of a_{k+1} and b_{k+1} may be swapped to guarantee that $\|f\left(b_{k+1}\right)\| \leq \|f\left(a_{k+1}\right)\|$ ensuring that b_{k+1} provides a better approximation of the function root. The iterative process continues until the search interval becomes smaller than the specified tolerance (ϵ), *i.e.*, $\|b_k - a_k\| \leq \epsilon$, at which point the approximated root is taken as b_k.

As shown in Fig. 1(a), the interval reduction can be less efficient than in the bisection method, where the interval is consistently halved. To overcome this situation, we propose a small modification of the selection of a_{k+1}. If $b_{k+1} = q$ and $f(m)\, f\left(b_{k+1}\right) < 0$, then the selection is $a_{k+1} = m$ as shown in Fig. 1(b), so the interval reduction is always greater than 50%. The pseudocode of this modification is presented in the Algorithm 1.

2.2. Brent method

The Brent method improves upon the Dekker method by using a three-point approach that combines the bisection, secant, and inverse quadratic interpolation methods. The main idea is to apply inverse quadratic interpolation whenever possible, as it offers faster convergence. However, if the new approximation falls outside the current search interval, the secant or bisection method is used instead. Additionally, quadratic interpolation cannot be applied if repeated approximations occur, as this would lead to division by zero.

Considering an interval $\left[a_k, b_k\right]$ where b_k is a better approximation of the function root, *i.e.*, $\|f\left(b_k\right)\| \leq \|f\left(a_k\right)\|$, a new search interval $\left[a_{k+1}, b_{k+1}\right]$ is generated at each iteration such that $f\left(a_{k+1}\right) f\left(b_{k+1}\right) < 0$ and $\|f\left(b_{k+1}\right)\| \leq \|f\left(a_{k+1}\right)\|$. The points a_k, b_k and b_{k-1} (with $b_{k-1} = a_0$ for the initial iteration) are used to calculate the point s using an interpolation method. If the three points are different it uses the quadratic interpolation formula with Eq. (3) otherwise, it uses the linear interpolation formula with Eq. (4) (see Eqs. (3) and (4) in Box I).

The value of s is accepted if it lies between b_k and $\left(3\,a_k + b_k\right)/4$. If s does not fall within this range, it must be calculated using the bisection method: formula with Eq. (5).

$$s = \frac{a_k + b_k}{2}. \tag{5}$$

Algorithm 1 Modified Dekker algorithm

Input: Function $f(x)$, search interval a and b and tolerances tol_1 and tol_2
Initialize the iteration counter, $k \leftarrow 0$
if $|f(a)| < |f(b)|$ **then**
  a, b ← b, a
end if
For the first iteration $b_{k-1} \leftarrow a$
while $|b - a| > tol_1$ and $|f(b)| > tol_2$ **do**
  $k \leftarrow k + 1$
  $q \leftarrow b_k - f\left(b_k\right)\dfrac{b_{k-1} - b_k}{f\left(b_{k-1}\right) - f\left(b_k\right)}$
  $m \leftarrow \dfrac{a_k + b_k}{2}$
  if $q \geq \min(b, m)$ and $q \leq \max(b, m)$ **then**
    $b_{k+1} \leftarrow q$
  else
    $b_{k+1} \leftarrow m$
  end if
  if $f\left(a_k\right) f\left(b_{k+1}\right) > 0$ **then**
    $a_{k+1} \leftarrow b_k$
  else if $q \geq \min(b, m)$ and $q \leq \max(b, m)$ and $f(m)\, f\left(b_{k+1}\right) < 0$ **then**
    $a_{k+1} \leftarrow m$
  end if
  if $\left|f\left(a_{k+1}\right)\right| < \left|f\left(b_{k+1}\right)\right|$ **then**
    $a_{k+1}, b_{k+1} \leftarrow b_{k+1}, a_{k+1}$
  end if
end while
Output: $x_{\text{approx}} \leftarrow b_k$, $f\left(x_{\text{approx}}\right) \leftarrow f\left(b_k\right)$ and $n_{\text{iter}} \leftarrow k$

To prevent the method from becoming excessively slow, the bisection method must be applied if $\|s - b_k\| < \|b_k - b_{k-1}\|/2$ when the bisection method was used in the previous iteration. Alternatively, if an interpolation method was applied in the last iteration, the bisection method should be used if $\|s - b_k\| < \|b_{k-1} - b_{k-2}\|/2$.

For defining of the new search interval, we set $b_{k+1} = s$. If $f\left(a_k\right) f\left(b_{k+1}\right) < 0$, then $a_{k+1} = a_k$ otherwise $a_{k+1} = b_k$.

The modification proposed in this work involves two key changes. First, we adjust the condition for accepting the value of s generated by the interpolation method, this value will only be accepted if it lies between b_k and $\left(a_k + b_k\right)/2$. Second, we modify the criterion for selecting a_{k+1}. If $f\left(a_k\right) f\left(b_{k+1}\right) < 0$ an additional test is made evaluating if $f\left(\left(a_k + b_k\right)/2\right) f\left(b_{k+1}\right) < 0$ so $a_{k+1} = \left(a_k + b_k\right)/2$. The

$$s = b_k + \frac{\frac{f(b_k)}{f(a_k)}\left[\left(1 - \frac{f(b_k)}{f(b_{k-1})}\right)(a_k - b_k) + \frac{f(a_k)}{f(b_{k-1})}\left(\frac{f(b_k)}{f(b_{k-1})} - \frac{f(a_k)}{f(b_{k-1})}\right)(b_{k-1} - b_k)\right]}{\left(\frac{f(b_k)}{f(b_{k-1})} - 1\right)\left(\frac{f(b_k)}{f(a_k)} - 1\right)\left(\frac{f(a_k)}{f(b_{k-1})} - 1\right)}. \tag{3}$$

$$s = b_k - f(b_k)\,\frac{b_{k-1} - b_k}{f(b_{k-1}) - f(b_k)}. \tag{4}$$

Box I.

Algorithm 2 Modified Brent algorithm

Input: Function $f(x)$, search interval a and b and tolerances tol_1 and tol_2
Initialize the iteration counter, $k \leftarrow 0$
if $|f(a)| < |f(b)|$ **then**
  a, b ← b, a
end if
For the first iteration $b_{k-1} \leftarrow a$
while $|b - a| > tol_1$ and $|f(b)| > tol_2$ **do**
  $k \leftarrow k + 1$
  if $b_k \neq b_{k-1}$ and $a_k \neq b_{k-1}$ **then**

$$s \leftarrow b_k + \frac{\frac{f(b_k)}{f(a_k)}\left[\left(1 - \frac{f(b_k)}{f(b_{k-1})}\right)(a_k - b_k) + \frac{f(a_k)}{f(b_{k-1})}\left(\frac{f(b_k)}{f(b_{k-1})} - \frac{f(a_k)}{f(b_{k-1})}\right)(b_{k-1} - b_k)\right]}{\left(\frac{f(b_k)}{f(b_{k-1})} - 1\right)\left(\frac{f(b_k)}{f(a_k)} - 1\right)\left(\frac{f(a_k)}{f(b_{k-1})} - 1\right)}$$

  else

$$s \leftarrow b_k - f(b_k)\,\frac{b_{k-1} - b_k}{f(b_{k-1}) - f(b_k)}$$

  end if
  if $s \leq \min(b, (3a + b)/4)$ or $s \geq \max(b, (3a + b)/4)$ **then**

$$s \leftarrow \frac{a_k + b_k}{2}$$

  end if
  $b_{k+1} \leftarrow s$
  if $f(a_k)\, f(b_{k+1}) > 0$ **then**
    $a_{k+1} \leftarrow b_k$
  else if $f((a_k + b_k)/2)\, f(b_{k+1}) < 0$ **then**
    $a_{k+1} \leftarrow (a_k + b_k)/2$
  end if
  if $\left|f(a_{k+1})\right| < \left|f(b_{k+1})\right|$ **then**
    $a_{k+1}, b_{k+1} \leftarrow b_{k+1}, a_{k+1}$
  end if
end while
Output: $x_{\text{approx}} \leftarrow b_k$, $f(x_{\text{approx}}) \leftarrow f(b_k)$ and $n_{\text{iter}} \leftarrow k$

pseudocode for the modified Brent algorithm is presented in Algorithm 2.

The Brent method already includes conditions to prevent slow convergence, however, the proposed modification ensures that the search interval is at least halved in each iteration.

The methods presented are tested using two examples, which are discussed in the following section. The implementations of these methods were carried out in Python 3.

3. Results and discussion

As a first case study to verify the effects of the proposed modifications, we will use the following function formula with Eq. (6). This function has three root (namely, $\xi_1 \approx 2.1584$, $\xi_2 \approx 4.6196$ and $\xi_3 \approx 7.2550$) as illustrated in Fig. 2.

$$f(x) = \exp\left(-\frac{x^2}{4}\right) - 2\cos(x) + \frac{x}{2} - \frac{5}{2}. \tag{6}$$

To analyze the numerical methods, we focus on finding the first root ($\xi_1 \approx 2.1584$). The initial interval was set with $a = 1.0$ and $b = 3.0$. Since the value of b_k represents the best approximation of the solution in each iteration, Fig. 3 illustrates the reduction of function value for the study case. The results for Dekker and modified Dekker methods in Fig. 3(a), while 3(b) presents the results for Brent and modified Brent methods.

Examining the data presented in these figures, we observe that the modification to the Dekker method yielded improved results from the very first iteration, while the modification to the Brent method demonstrated clear advantages primarily in the later iterations. Additionally, as illustrated in Figs. 4(a) and 4(b), the original methods exhibit some iterations where the reduction of the search interval is less than half, whereas the proposed methods consistently achieved a minimum reduction of 50% of the search interval for each iteration. Notably, in certain iterations the interval reduction for the original methods is very small.

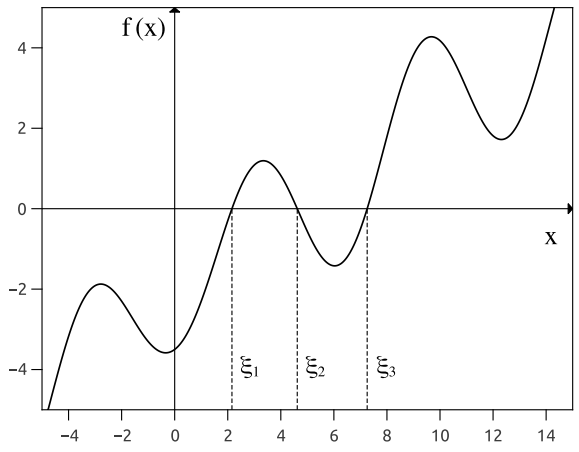


Fig. 2. Study case function 1.

A practical application of root finding methods can be found in solving cubic equation of state. Steffen and Silva [16] introduced a generic equation of state: formula with Eq. (7). They also expressed this equation in a polynomial form based on a dimensionless property V/b, as formula with Eq. (8).

$$P_r = \frac{P}{P_C} = \frac{T_r}{\Gamma}\frac{1}{\left(\frac{V}{b}\right)-1} - \frac{\Lambda\,\alpha\left(\omega, T_r\right)}{\Gamma^2}\frac{1}{\left(\frac{V}{b}\right)^2+\lambda\left(\frac{V}{b}\right)+\sigma}, \tag{7}$$

$$\left(\frac{V}{b}\right)^3 - \left(1-\lambda+\frac{T_r}{\Gamma P_r}\right)\left(\frac{V}{b}\right)^2 + \left(\sigma-\lambda-\lambda\frac{T_r}{\Gamma P_r}+\frac{\Lambda\,\alpha\left(\omega, T_r\right)}{\Gamma^2 P_r}\right)\left(\frac{V}{b}\right) - \left(\sigma+\sigma\frac{T_r}{\Gamma P_r}+\frac{\Lambda\,\alpha\left(\omega, T_r\right)}{\Gamma^2 P_r}\right) = 0, \tag{8}$$

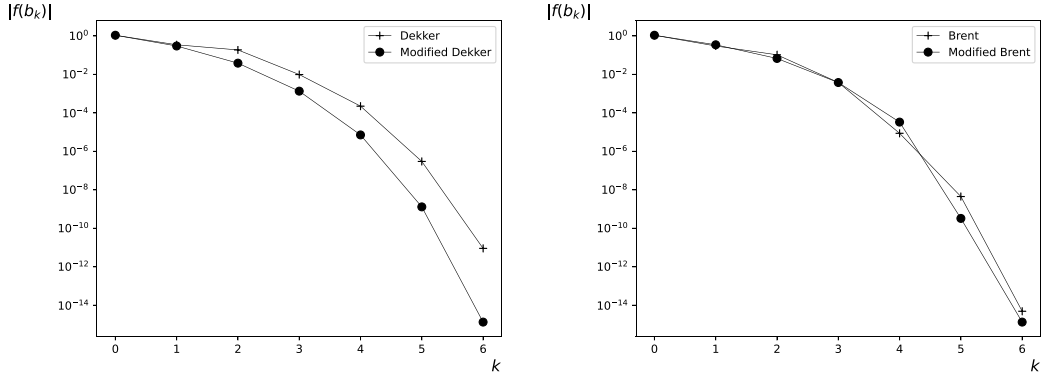
where the values of λ and σ are substance-independent, and each set of these parameters represents a specific equation of state, Λ and Γ are positive parameters whose values depend on the equation of state. T_r and P_r are the reduced temperature and reduced pressure, respectively, while ω is the acentric factor, V is the volume and b the covolume parameter.

For the Peng–Robinson Equation of State, the parameter values are [19]: $\lambda = 2$, $\sigma = -1$, $\Lambda = 0.45724$, $\Gamma = 0.07780$ and

$$\alpha\left(\omega, T_r\right) = \left(1+\left(0.37464+1.54226\,\omega-0.26992\,\omega^2\right)\left(1-\sqrt{T_r}\right)\right)^2. \tag{9}$$

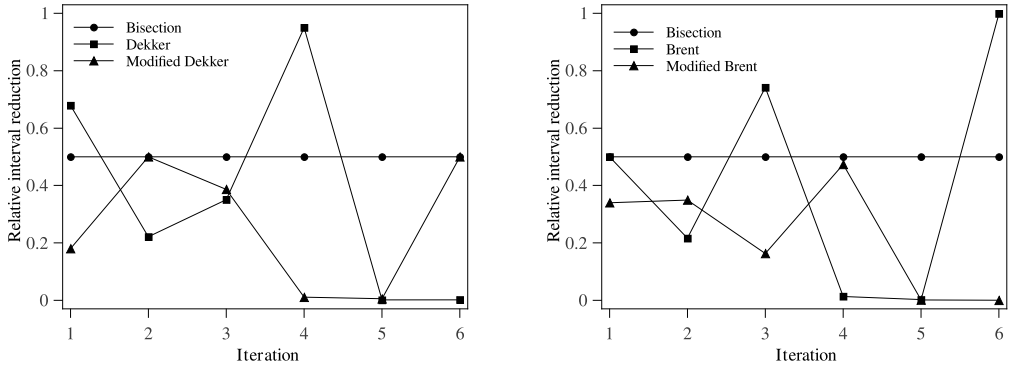
As a second case study, we consider finding the root finding of V/b (a dimensionless value) for the Peng–Robinson equation of state using the parameters $\omega = 0.200$, $T_r = 0.85$ and $P_r = 0.45$. The function profile represented by Eq. (8) is shown in Fig. 5, where it can be observed that there are three roots, with one located in the interval $14 \le V/b \le 17$, representing the volume for the vapor phase. By applying this interval and a tolerance of 10^{-10}, the approximate root is found to be 15.0676609061.

The results presented in Fig. 6 demonstrate better performance for the modified methods. In both modifications, the values of $f(b)$ are



(a) Dekker and modified Dekker methods (b) Brent and modified Brent methods

Fig. 3. Convergence for study case 1.



(a) Dekker and modified Dekker methods (b) Brent and modified Brent methods

Fig. 4. Relative interval reduction for study case 1.

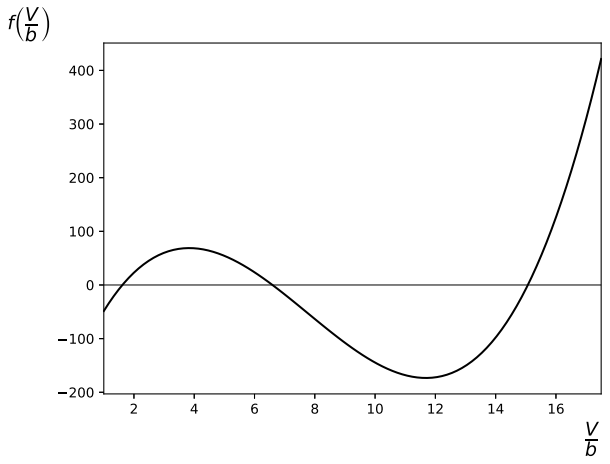


Fig. 5. Study case function 2.

lower than the original methods in almost every iteration, and the modified Dekker method required one fewer iteration than the original Dekker. As shown in Fig. 7, in this second case study, the modified methods also ensured that the search interval was at least halved in each iteration.

Table 1 presents the average execution time for each case evaluated in this study. The mean time was calculated by running each case one hundred times. Only in case study 1 did the modified Dekker method show a slightly worse execution time. However, in all other comparisons, the modified methods demonstrated better performance, emphasizing the improvements proposed in this work.

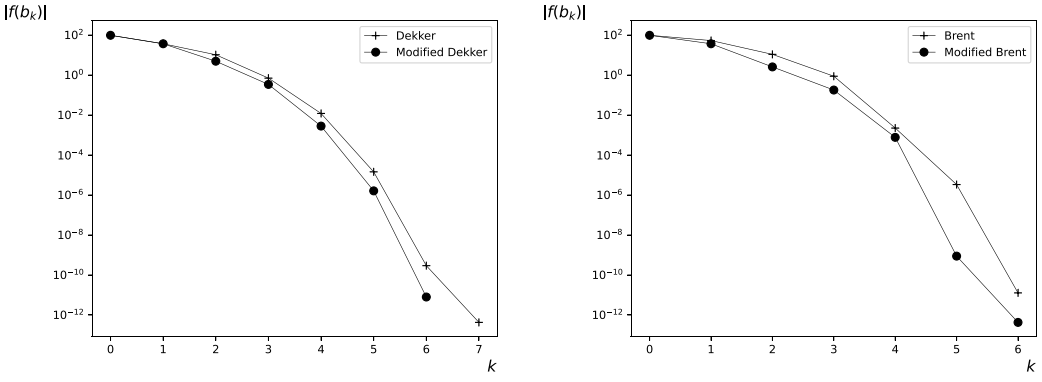
Table 1
Mean execution time.

Method	Time × 10^5 (s)	
	Case study 1	Case study 2
Dekker	1.5535	1.7261
Modified Dekker	2.1913	0.6063
Brent	1.7781	1.9433
Modified Brent	0.9747	1.4467

4. Conclusion

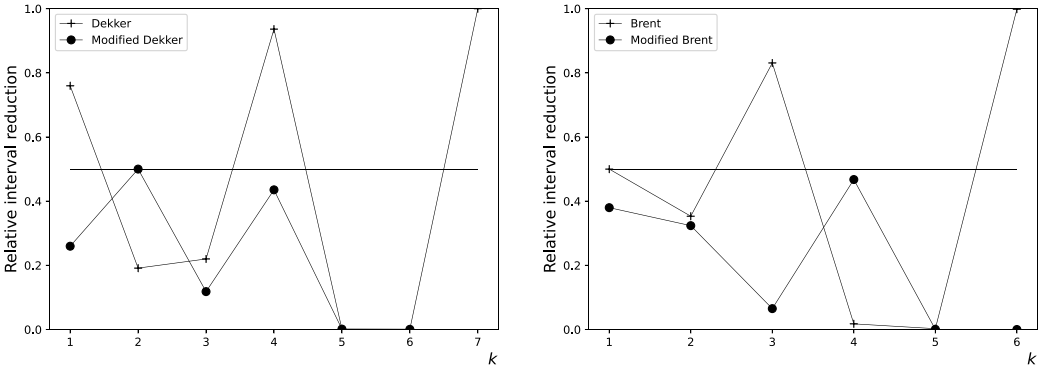
In this work, we proposed modifications to the Dekker and Brent numerical root-finding methods to ensure that the search interval is halved in each iteration. These modifications can lead to a reduced number of iterations with a simple adjustment to the computational code implementation. Our proposed modifications successfully achieved this goal, resulting in significant improvements in relative interval reduction and a modest enhancement in function value reduction, particularly for the modified Dekker method.

As a direction for future research, we suggest applying these methods across a broader range of examples and exploring the incorporation



(a) Dekker and modified Dekker methods (b) Brent and modified Brent methods

Fig. 6. Convergence for study case 2.



(a) Dekker and modified Dekker methods (b) Brent and modified Brent methods

Fig. 7. Relative interval reduction for study case 2.

of characteristics from other methods to develop hybrid approaches that ensure fast and reliable convergence.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

[1] V. Steffen, E.A. da Silva, Steady-state modeling of reactive distillation columns, Acta Sci. Technol. 34 (1) (2012) 61–69, http://dx.doi.org/10.4025/actascitechnol.v34i1.9535.

[2] V. Steffen, E.A. da Silva, Steady-state modeling of equilibrium distillation, in: Distillation - Innovative Applications and Modeling, InTech, 2017, http://dx.doi.org/10.5772/66833.

[3] V. Steffen, E.A. Silva, Numerical methods and initial estimates for the simulation of steady-state reactive distillation columns with an algorithm based on tearing equations methodology, Therm. Sci. Eng. Prog. 6 (2018) 1–13, http://dx.doi.org/10.1016/j.tsep.2018.02.014.

[4] L. Alzaben, F. Bertrand, D. Boffi, On the spectrum of the finite element approximation of a three field formulation for linear elasticity, Ex. Counterex. 2 (2022) 100076, http://dx.doi.org/10.1016/j.exco.2022.100076.

[5] N. Jha, Numerical treatment of fractal boundary value problems for heat conduction in polar bear with spatial variation of thermal conductivity, Ex. Counterex. 2 (2022) 100088, http://dx.doi.org/10.1016/j.exco.2022.100088.

[6] F. Bertrand, M. Brodbeck, T. Ricken, On robust discretization methods for poroelastic problems: Numerical examples and counter-examples, Ex. Counterex. 2 (2022) 100087, http://dx.doi.org/10.1016/j.exco.2022.100087.

[7] F. Bertrand, K. Mang, Editorial - recent fails and findings of numerical methods in mechanics, Ex. Counterex. 3 (2023) 100098, http://dx.doi.org/10.1016/j.exco.2022.100098.

[8] J. Sharma, A composite third order Newton–Steffensen method for solving nonlinear equations, Appl. Math. Comput. 169 (1) (2005) 242–246.

[9] N. Flocke, Algorithm 954: An accurate and efficient cubic and quartic equation solver for physical applications, ACM Trans. Math. Softw. 41 (4) (2015) 1–24.

[10] A. Cordero, J.L. Hueso, E. Martínez, J.R. Torregrosa, Steffensen type methods for solving nonlinear equations, J. Comput. Appl. Math. 236 (12) (2012) 3058–3064.

[11] U.K. Deiters, Calculation of densities from cubic equations of state, AIChE J. 48 (4) (2002) 882–886.

[12] R. Monroy-Loperena, A note on the analytical solution of cubic equations of state in process simulation, Ind. Eng. Chem. Res. 51 (19) (2012) 6972–6976.

[13] C.L. Sabharwal, Hybrid algorithm improving bisection, regula falsi, dekker, brent algorithms for roots of non-linear equations, Int. J. Latest Res. Eng. Technol. (IJLRET) 5 (2019) 01–15.

[14] V. Steffen, E.A. da Silva, An analysis about analytical calculation of volume roots from cubic equations of state, AIChE J. 67 (7) (2021) e17273.

[15] R.A. Fernández Molina, L.D.G. Sigalotti, O. Rendón, A.J. Mejias, A rapidly convergent method for solving third-order polynomials, AIP Adv. 12 (4) (2022).

[16] V. Steffen, Particle swarm optimization with a simplex strategy to avoid getting stuck on local optimum, AI Comput. Sci. Robot. Technol. (2022) http://dx.doi.org/10.5772/acrt.11.

[17] T.J. Dekker, Finding a zero by means of successive linear interpolation, Constr. Aspects Fundam. Theorem Algebra 1 (1969).

[18] R.P. Brent, An algorithm with guaranteed convergence for finding a zero of a function, Comput. J. 14 (4) (1971) 422–425.

[19] D.-Y. Peng, D.B. Robinson, A new two-constant equation of state, Ind. Eng. Chem. Fundam. 15 (1) (1976) 59–64, http://dx.doi.org/10.1021/i160057a011.