A quadrature method for the numerical solution of two real nonlinear equations in two unknowns

Nikolaos I. Ioakimidis

Division of Applied Mathematics and Mechanics, Department of Engineering Sciences, School of Engineering, University of Patras, GR-265 04 Patras, Greece
e-mail: n.ioakimidis@upatras.gr

Abstract A quadrature method for the numerical solution of a system of two real nonlinear equations in two unknowns in a region of the $xy$-plane, based on the use of two appropriate numerical integration rules for ordinary integrals in this region, is proposed. The main advantage of the method, beyond its originality and peculiarity, is the fact that no initial approximation to the sought solution is required. Three numerical applications, where the Gauss– and Lobatto–Chebyshev quadrature rules have been used, show the efficiency and the convergence of the proposed method. A generalization to the case of four real nonlinear equations in four unknowns (by using quaternions) is also suggested in brief.

Keywords Systems of nonlinear equations · Systems of real algebraic equations · Solution of systems of equations · Noniterative methods · Approximate solutions · Numerical solutions · Numerical integration · Quadrature rules · Gauss–Chebyshev rule · Lobatto–Chebyshev rule · Quaternions

Copyright © 1994 by the author

1. Introduction

Systems of two real nonlinear algebraic equations in two unknowns appear very frequently in physical and engineering applications. For the determination of a solution of such a system of equations many numerical methods are available [8, 10, 11]. These methods fall into two main categories: the iterative methods and the minimization methods [8]. A very good description of these methods and an extensive bibliography can be found in the classical relevant book by Ortega and Rheinboldt [8]. Unfortunately, the difficulties faced when these methods are used are well known [10, pp. 349–350], [11, pp. 174–177]. In general, if the classical iterative methods, like the Newton–Raphson method, are given a good initial approximation to the sought solution, then they converge rapidly to this solution. But, generally, such an approximation is not available.

1Both the internal and the external links (all appearing in blue) were added by the author on December 31, 2017 for the online publication of this technical report.
Here we present a quadrature method for the numerical solution of a system of two (or four) real nonlinear equations. This method is irrelevant to the classical methods of solution of systems of nonlinear equations, it is noniterative and it can be considered as one more (and probably somewhat strange) practical application of numerical integration beyond the already known related applications, some of which can be found in the excellent recent book on practical numerical integration [3]. Moreover, the present approach is a nontrivial extension of the corresponding method for the determination of a root \( c \) of one nonlinear equation \( f(x) = 0 \) in an interval \([a, b]\) [5], based on the double numerical evaluation of the ordinary integral

\[
    J = \int_a^b \frac{1}{\sqrt{(b-x)(x-a)}} \frac{x-c}{f(x)} \, dx
\]

by using the Gauss– and Lobatto–Chebyshev quadrature rules and, next, solving the resulting linear equation for \( c \). (Another quadrature method for one nonlinear equation, based on Cauchy-type principal value integrals, is described in [6].) The present method can be used either as a self-sufficient method or as a method for the provision of good approximations to the solutions of systems of two (or four) nonlinear equations for the Newton–Raphson and related rapidly convergent methods.

### 2. The quadrature method

We consider a system of two real nonlinear equations in two unknowns:

\[
    u(x, y) = 0, \quad v(x, y) = 0
\]

and we assume that this system has a simple solution \((x_0, y_0)\) in a region \( D \) of the \( xy \)-plane. This can be ascertained either by physical considerations or by the graphical method [11] or, more accurately, by the Picard method [9], which was reconsidered and tested numerically by Hoenders and Slump [4]. We do not feel that it is necessary to repeat this method here. We will determine an approximation to the solution \((x_0, y_0)\) of Eqs. (2) in \( D \).

To this end, we consider the double integral

\[
    I = \iint_D w \frac{(u_x + iv_x)(x-x_0) + (u_y + iv_y)(y-y_0)}{u + iv} \, dD,
\]

where \( w = w(x,y) \) is a positive weight function in \( D \) and \( u_x, u_y, v_x \) and \( v_y \) are the first partial derivatives of \( u \) and \( v \), respectively, at the point \((x, y)\). Since the solution \((x_0, y_0)\) of Eqs. (2) in \( D \) was assumed simple, it is clear from Eq. (3) that its integrand remains bounded at \((x_0, y_0)\). Moreover, we can put some continuity requirements on \( u \) and \( v \), e.g. to have continuous second partial derivatives in \( D \), in order to have a well-behaved integrand in Eq. (3). Clearly, only if \((x_0, y_0)\) is the sought solution of Eqs. (2), we can approximate the integral \( I \) by using a quadrature rule [1, 3] for the region \( D \). Otherwise, this is not permissible.

Although it is possible to work directly with Eq. (3) using complex variables, we can also, alternatively, split it into two real integrals:

\[
    I_1 = \iint_D w \frac{(u_x + iv_x)(x-x_0) + (u_y + iv_y)(y-y_0)}{u^2 + v^2} \, dD,
\]

\[
    I_2 = \iint_D w \frac{(uv_x - vu_x)(x-x_0) + (uv_y - vu_y)(y-y_0)}{u^2 + v^2} \, dD.
\]
Then $I = I_1 + iI_2$. Now we can use two quadrature rules, Q1 and Q2, for the region $D$ corresponding to the essentially arbitrary weight function $w$:

\[ Q1: \quad \int_D w f(x,y) \, dD = \sum_{i=1}^{n_1} w_{i1} f(x_{i1}, y_{i1}) + E_1, \quad (6) \]

\[ Q2: \quad \int_D w f(x,y) \, dD = \sum_{i=1}^{n_2} w_{i2} f(x_{i2}, y_{i2}) + E_2, \quad (7) \]

where $(x_{i1}, y_{i1})$ and $(x_{i2}, y_{i2})$ are the nodes, $w_{i1}$ and $w_{i2}$ are the weights, and $E_1$ and $E_2$ are the error terms. The dependence of these quantities on the numbers of nodes ($n_1$ and $n_2$, respectively) was omitted in the notation used in Eqs. (6) and (7).

By applying both quadrature rules Q1 and Q2 to Eqs. (4) and (5), ignoring the error terms $E_1$ and $E_2$ and equating the numerical results, we obtain

\[ \sum_{i=1}^{n_1} w_{i1} \left[ g_1(x_{i1}, y_{i1})(x_{i1} - \tilde{x}_0) + g_2(x_{i1}, y_{i1})(y_{i1} - \tilde{y}_0) \right] = \sum_{i=1}^{n_2} w_{i2} \left[ g_1(x_{i2}, y_{i2})(x_{i2} - \tilde{x}_0) + g_2(x_{i2}, y_{i2})(y_{i2} - \tilde{y}_0) \right], \quad (8) \]

\[ \sum_{i=1}^{n_1} w_{i1} \left[ g_3(x_{i1}, y_{i1})(x_{i1} - \tilde{x}_0) + g_4(x_{i1}, y_{i1})(y_{i1} - \tilde{y}_0) \right] = \sum_{i=1}^{n_2} w_{i2} \left[ g_3(x_{i2}, y_{i2})(x_{i2} - \tilde{x}_0) + g_4(x_{i2}, y_{i2})(y_{i2} - \tilde{y}_0) \right], \quad (9) \]

where

\[ g_1 = \frac{uu_x + vv_x}{u^2 + v^2}, \quad g_2 = \frac{uu_y + vv_y}{u^2 + v^2}, \quad g_3 = \frac{uv_x - vu_x}{u^2 + v^2}, \quad g_4 = \frac{uv_y - vu_y}{u^2 + v^2} \quad (10) \]

and $\tilde{x}_0, \tilde{y}_0$ denote approximations to $x_0, y_0$ due to the omission of the error terms $E_1$ and $E_2$. Eqs. (8) and (9) constitute a system of two linear algebraic equations in two unknowns: $\tilde{x}_0$ and $\tilde{y}_0$. The solution of this system is directly found to be

\[ \tilde{x}_0 = \frac{a_4 b_1 - a_2 b_2}{\delta}, \quad \tilde{y}_0 = \frac{a_1 b_2 - a_3 b_1}{\delta} \quad \text{with} \quad \delta = a_1 a_4 - a_2 a_3, \quad (11) \]

where

\[ a_k = \sum_{i=1}^{n_1} w_{i1} g_k(x_{i1}, y_{i1}) - \sum_{i=1}^{n_2} w_{i2} g_k(x_{i2}, y_{i2}), \quad k = 1, 2, 3, 4, \quad (12) \]

\[ b_k = \sum_{i=1}^{n_1} w_{i1} [x_{i1} g_{2k-1}(x_{i1}, y_{i1}) + y_{i1} g_{2k}(x_{i1}, y_{i1})] \]

\[ - \sum_{i=1}^{n_2} w_{i2} [x_{i2} g_{2k-1}(x_{i2}, y_{i2}) + y_{i2} g_{2k}(x_{i2}, y_{i2})], \quad k = 1, 2. \quad (13) \]

As is clear from Eqs. (8) to (13), the implementation of the present approach requires the numerical evaluation of the functions $u$ and $v$ and their first partial derivatives at $n_1 + n_2$ points, the nodes of the quadrature rules (6) and (7). This is a kind of sampling of the values of $u$ and $v$ and their first partial derivatives at fixed points in $D$ which permits the approximate determination of the
The square 

the aforementioned error terms

3. Numerical applications

unchanged.

like 2

general, better

x

the classical one-dimensional Gauss– and Lobatto–Chebyshev quadrature rules on

and the quadrature rules Q1 and Q2 in Eqs. (6) and (7) respectively are the product rules \[1, 3\] of

n

of nodes

E

solution \((x_0, y_0)\) of the system of Eqs. (2). On the other hand, it is also clear from the above
equations that as the error terms \(E_1\) and \(E_2\) become smaller (for increasing values of the numbers
of nodes \(n_1\) and \(n_2\)), then the approximate numerical results \(\hat{x}_0\) and \(\hat{y}_0\) for \(x_0\) and \(y_0\) approximate, in
general, better \(x_0\) and \(y_0\) (since the only source of error in Eqs. (8), (9) and (11) is the omission of
the aforementioned error terms \(E_1\) and \(E_2\)). In practice, \(n_1 \approx n_2\) and we leave \(n_{1,2}\) increase (usually
like \(2^k, k = 1, 2, 3, \ldots\)) until the significant digits in \(\hat{x}_0\) and \(\hat{y}_0\), in which we are interested, remain
unchanged.

3. Numerical applications

We apply the method described in the previous section to the special case where the region \(D\) is
the square \([-1, 1] \times [-1, 1]\), the weight function \(w = w(x, y)\) is the function

\[
w = w(x, y) = \frac{1}{\sqrt{(1-x^2)(1-y^2)}}
\]  

and the quadrature rules Q1 and Q2 in Eqs. (6) and (7) respectively are the product rules \[1, 3\] of
the classical one-dimensional Gauss– and Lobatto–Chebyshev quadrature rules on \([-1, 1]\) \[7\]

\[
\int_{-1}^{1} \frac{f(t)}{\sqrt{1-t^2}} \, dt \approx \frac{\pi}{n} \sum_{i=1}^{n} f(t_i), \quad t_i = \cos \left(\frac{2(i-1)\pi}{2n}\right), \quad i = 1, 2, \ldots, n,
\]  

(15)

\[
\int_{-1}^{1} \frac{f(t)}{\sqrt{1-t^2}} \, dt \approx \frac{\pi}{n} \sum_{i=0}^{n} f(t_i^*), \quad t_i^* = \cos \left(\frac{i\pi}{n}\right), \quad i = 0, 1, \ldots, n,
\]  

(16)

respectively. (In the sum in Eq. (16) the first and last terms should be halved.) Clearly, the selection
of the weight function (14) aimed just at the determination of the nodes and the weights from
elementary closed-form formulae as is clear from Eqs. (15) and (16). Moreover, as it is evident
from Eqs. (6), (7), (15) and (16), \(n_1 = n^2\) and \(n_2 = (n+1)^2\), that is, \(2n^2 + 2n + 1\) evaluations of our
functions \(u\) and \(v\) and their first partial derivatives are required for the application of the proposed
method. But if we use \(n^* = 2n\) after \(n\), then the functional evaluations for \(n\) are useful for \(n^*\) as well
and need not be repeated. This is clear from the formulae for the nodes in Eqs. (15) and (16).

<table>
<thead>
<tr>
<th>(n)</th>
<th>(x_0)</th>
<th>(y_0)</th>
<th>((u^2 + v^2)(x_0, y_0))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.519380</td>
<td>0.868217</td>
<td>1.505 \times 10^{-1}</td>
</tr>
<tr>
<td>2</td>
<td>-0.091970</td>
<td>0.601042</td>
<td>1.288 \times 10^{-1}</td>
</tr>
<tr>
<td>4</td>
<td>-0.298171</td>
<td>0.756013</td>
<td>7.047 \times 10^{-3}</td>
</tr>
<tr>
<td>8</td>
<td>-0.296171</td>
<td>0.708883</td>
<td>2.999 \times 10^{-5}</td>
</tr>
<tr>
<td>16</td>
<td>-0.293636</td>
<td>0.705983</td>
<td>7.106 \times 10^{-6}</td>
</tr>
<tr>
<td>32</td>
<td>-0.292899</td>
<td>0.707108</td>
<td>1.017 \times 10^{-10}</td>
</tr>
</tbody>
</table>

“Exact” values: -0.29289322 0.70710678 —
Table 2: Analogous results to those of Table 1 but for the system of nonlinear equations (18)

<table>
<thead>
<tr>
<th>( n )</th>
<th>( x_0 )</th>
<th>( y_0 )</th>
<th>((u^2 + v^2)(x_0, y_0))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>-0.287929</td>
<td>-1.478627</td>
<td>3.852 \times 10^0</td>
</tr>
<tr>
<td>2</td>
<td>-1.406241</td>
<td>-1.736595</td>
<td>1.045 \times 10^1</td>
</tr>
<tr>
<td>4</td>
<td>-0.228592</td>
<td>-0.893874</td>
<td>7.278 \times 10^{-2}</td>
</tr>
<tr>
<td>8</td>
<td>-0.172876</td>
<td>-0.765620</td>
<td>2.941 \times 10^{-3}</td>
</tr>
<tr>
<td>16</td>
<td>-0.188321</td>
<td>-0.744983</td>
<td>2.766 \times 10^{-4}</td>
</tr>
<tr>
<td>32</td>
<td>-0.189898</td>
<td>-0.754352</td>
<td>9.106 \times 10^{-7}</td>
</tr>
</tbody>
</table>

“Exact” values: -0.18946662 -0.75392163 —

In Tables 1, 2 and 3 we present the numerical results that we obtained for the following three systems of nonlinear equations:

\[
\begin{align*}
    u(x,y) &= x - y + 1, & v(x,y) &= x^2 + 2x + y^2, \\ 
    u(x,y) &= e^x + \sinh y, & v(x,y) &= (x + 1)^4 + y^2 - 1, \\ 
    u(x,y) &= x^2 - y, & v(x,y) &= (x + 1)^2 + y^2 - 4,
\end{align*}
\]

respectively for \( n = 2^k, k = 0, 1, \ldots, 5 \), in Tables 1 and 2, and for \( n = 1, 2, \ldots, 10 \) in Table 3. The existence of one and only one solution \((x_0, y_0)\) of the systems of nonlinear Eqs. (17), (18) and (19) in \([-1, 1] \times [-1, 1]\) is assured by elementary considerations. The “exact” values of \( x_0 \) and \( y_0 \) are also displayed in Tables 1, 2 and 3 together with the “error” \( u^2(x_0, y_0) + v^2(x_0, y_0) \). It is clear from the numerical results of these tables that the present method for the approximate numerical solution of systems of two real nonlinear equations gives sufficiently accurate results even for small values of \( n \) and, moreover, that for increasing values of \( n \) the approximate values \( \tilde{x}_0 \) and \( \tilde{y}_0 \) of \( x_0 \) and \( y_0 \) really tend to \( x_0 \) and \( y_0 \), respectively. Finally, clearly, the computer program for the implementation of the present method is both very elementary and very short.

4. A generalization

A generalization of the above approach for the numerical solution of systems of two real nonlinear equations is also possible. In fact, this approach was based on Eq. (3), which led to Eqs. (4) and (5) by multiplying the numerator and the denominator of the integrand in Eq. (3) by the complex conjugate \( u - iv \) of \( u + iv \). In this way, the square of the absolute value of \( u + iv \), i.e. \( u^2 + v^2 \), appears in the denominator of the integrands in Eqs. (4) and (5).

In a quite analogous way, we can proceed to systems of four nonlinear equations in four unknowns \((x, y, z, t)\). In this case, we can use the most elementary results of the quaternion algebra (see, e.g., [2]). Then the fundamental integral (3) will take the modified form

\[
I = \iiint_D w \frac{q_x(x - x_0) + q_y(y - y_0) + q_z(z - z_0) + q_t(t - t_0)}{q} \, dD, \quad dD = dx \, dy \, dz \, dt,
\]

where \( w \) and \( q \) are functions of \((x, y, z, t)\).
Table 3: Analogous results to those of Table 1 but for the system of nonlinear equations (19) and for \( n = 1, 2, \ldots, 10 \)

<table>
<thead>
<tr>
<th>( n )</th>
<th>( x_0 )</th>
<th>( y_0 )</th>
<th>((u^2 + v^2)(x_0, y_0))</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.574194</td>
<td>3.761290</td>
<td>1.712 \times 10^2</td>
</tr>
<tr>
<td>2</td>
<td>0.941953</td>
<td>0.595796</td>
<td>1.009 \times 10^{-1}</td>
</tr>
<tr>
<td>3</td>
<td>0.845818</td>
<td>0.798368</td>
<td>8.857 \times 10^{-3}</td>
</tr>
<tr>
<td>4</td>
<td>0.861643</td>
<td>0.729196</td>
<td>1.817 \times 10^{-4}</td>
</tr>
<tr>
<td>5</td>
<td>0.858870</td>
<td>0.747034</td>
<td>2.690 \times 10^{-4}</td>
</tr>
<tr>
<td>6</td>
<td>0.843458</td>
<td>0.755521</td>
<td>2.896 \times 10^{-3}</td>
</tr>
<tr>
<td>7</td>
<td>0.859632</td>
<td>0.750401</td>
<td>5.858 \times 10^{-4}</td>
</tr>
<tr>
<td>8</td>
<td>0.858967</td>
<td>0.742702</td>
<td>7.806 \times 10^{-5}</td>
</tr>
<tr>
<td>9</td>
<td>0.861615</td>
<td>0.744855</td>
<td>4.230 \times 10^{-4}</td>
</tr>
<tr>
<td>10</td>
<td>0.857682</td>
<td>0.742639</td>
<td>5.552 \times 10^{-5}</td>
</tr>
</tbody>
</table>

“Exact” values: 0.85894413 0.73778502 —

where \( w = w(x, y, z, t) \) denotes again the related weight function and \( q_m (m = x, y, z, t) \) denote the first partial derivatives of the quaternion function \( q(x, y, z, t) \) defined by

\[
q = q(x, y, z, t) = \mathbf{1} q_1(x, y, z, t) + \mathbf{i} q_2(x, y, z, t) + \mathbf{j} q_3(x, y, z, t) + \mathbf{k} q_4(x, y, z, t),
\]

where

\[
q_k(x, y, z, t) = 0, \quad k = 1, 2, 3, 4,
\]

are the actual nonlinear equations to be solved and \( \mathbf{1}, \mathbf{i}, \mathbf{j} \) and \( \mathbf{k} \) are the four “units” in the quaternion algebra (like only 1 and \( i \) in the ordinary complex algebra).

Next, we can multiply the numerator and the denominator of the integrand in Eq. (20) by the conjugate quaternion function \( \bar{q} \), defined by [2]

\[
\bar{q} = \mathbf{1} q_1 - \mathbf{i} q_2 - \mathbf{j} q_3 - \mathbf{k} q_4.
\]

Then we get again the square of the absolute value of \( q \), that is [2]

\[
|q|^2 = q \bar{q} = q_1^2 + q_2^2 + q_3^2 + q_4^2
\]

in the denominator of the integrand in Eq. (20). Applying the product quadrature rules analogous to Q1 and Q2 in Eqs. (6) and (7) respectively (but now in four dimensions, that is with respect to the four variables \( x, y, z \) and \( t \)) to this modified form of Eq. (20) and getting the parts corresponding to \( \mathbf{1}, \mathbf{i}, \mathbf{j} \) and \( \mathbf{k} \), we can work exactly as in the quadrature method of Section 2 for two variables, \( x \) and \( y \), and reach analogous results.

Of course, we can further proceed to the case of eight nonlinear equations and so on. In all cases, our final equations will be purely real ones (not complex or hypercomplex ones) exactly as has been the case in Section 2, where only real variables have been used. We will not proceed to further details on these possible generalizations.
References


