BIT 11 (1971), 168–174

A MODIFIED REGULA FALSI METHOD FOR COMPUTING THE ROOT OF AN EQUATION

M. DOWELL and P. JARRATT

Abstract.

The Illinois method is briefly described and the asymptotic convergence of the method investigated. Numerical examples are also given including comparisons with other similar robust methods.

1. Introduction.

In practical root-finding problems where good initial estimates of the roots are known, there is a wide range of computationally efficient algorithms available which can be programmed for use on a digital computer. For difficult problems, however, in which the prior information on the location of the root is poor, such methods often fail to converge, and this has led to a search for techniques which are relatively intensitive to the choice of starting values. One of the oldest and best known of these methods is the Regula Falsi which we briefly describe.

We consider the equation

$$f(x) = 0 \tag{1}$$

and suppose that at the start of the solution process two approximations to a root, x_{i-1} and x_i, are available for which $f_{i-1}f_i < 0$. A new value, x_{i+1}, is now computed from the rule

$$x_{i+1} = x_i - \frac{f_i(x_i - x_{i-1})}{f_i - f_{i-1}}, \tag{2}$$

and f_{i+1} is evaluated. The estimates to be used for the next iteration are x_{i+1} and whichever of x_i and x_{i-1} give a function values of opposite sign to f_{i+1}. The process is continued until some suitable criterion has been satisfied, for example $|(x_{i+1} - x_i)/x_i| < \delta$, where δ is a nominated tolerance. The method is attractive since convergence to a root of (1) is guaranteed. It suffers from the drawback, however, that once an interval is reached on which the function is convex or concave, thereafter one of the end-points of this interval is always retained, and this feature slows

Received November 5, 1970.

down the asymptotic convergence to first order. We now describe a modification of the Regula Falsi which leads to a considerable enhancement in speeed of convergence without the guarantee of convergence being lost. This refinement, which has now entered the folklore of computing, is thought by some to have been due originally to the staff of the computer centre at the University of Illinois in the early 1950's. The present paper gives a theoretical analysis of the behaviour of this refinement and also reports some numerical experience with the process.

2. The Illinois algorithm.

The method follows the Regula Falsi except that the estimates chosen for the next iteration are selected according to the following rules:

i) if $f_{i+1}f_i < 0$, then (x_{i-1},f_{i-1}) is replaced by (x_i,f_i)
ii) if $f_{i+1}f_i > 0$, then (x_{i-1},f_{i-1}) is replaced by $(x_{i-1},f_{i-1}/2)$.

As before, (x_{i+1},f_{i+1}) replaces (x_i,f_i). The function values used at each iteration will again always have opposite signs and the introduction of the value $f_{i-1}/2$ for f_{i-1} is a modification designed to speed convergence by preventing the retention of an end point.

We now analyse the asymptotic convergence of the method. For initial estimates x_0 and x_1 which are sufficiently close to a root θ of (1), asymptotic error theory can be used to examine formally the behaviour of the algorithm. We begin by defining the error in the ith approximation by $\varepsilon_i = x_i - \theta$, and using the Taylor expansion of f_i about θ we find

$$f_i = \sum_{r=1}^{\infty} c_r \varepsilon_i^r, \text{ where } c_r = f^{(r)}(\theta)/r!$$

and $c_0 = f(\theta) = 0$.

By substituting in (2) it is easy to show that for a simple root

$$\varepsilon_{i+1} \sim \frac{c_2}{c_1}\varepsilon_i\varepsilon_{i-1}\,. \tag{3}$$

The behaviour at the next iteration depends on whether $f_{i+1}f_i$ is less than or greater than zero. In the former unmodified case, by straightforwardly applying (3) we shall have

$$\varepsilon_{i+2} \sim \frac{c_2}{c_1}\varepsilon_{i+1}\varepsilon_i\,,$$

while for the latter modified case analysis gives

$$\varepsilon_{i+2} \sim -\varepsilon_{i+1}\,. \tag{4}$$

By using (3) and (4) it is now possible to examine the asymptotic iterating pattern of the Illinois method. Assuming that ε_{i-1} is negative and ε_i positive, and taking first the case c_2/c_1 positive, we find a sequence of values, calculated from modified (M) and unmodified (U) iterations in the order $UUM, UUM, UUM, \ldots$. For c_2/c_1 negative, the pattern is $UM, UUM, UUM, \ldots$, and in both cases the basic pattern is one of two unmodified iterations, followed by a modified one. From (3) and (4) it can be shown that

$$\varepsilon_{i+2} \sim \left(\frac{c_2}{c_1}\right)^2 \varepsilon_{i-1}{}^3 ,$$

and if we designate the three iterations UUM as a single iteration with error μ, we find

$$\mu_{i+1} \sim \left(\frac{c_2}{c_1}\right)^2 \mu_i{}^3 .$$

Hence we have a process which is third order at a cost of three evaluations of f per step in the sequence.

Accordingly, using Traub's (1, Appendix C) Efficiency Index, we find that the computational efficiency of the Illinois method is $3^{\frac{1}{3}} = 1.442\ldots$, compared with an efficiency of 1 for the unmodified Regula Falsi.

3. Numerical Illustrations.

We show first in Table 1 the behaviour of the method for the equation $\sin x - 0.5 = 0$.

The root is $\theta = \pi/6$ for which $c_2/c_1 = -\tan\theta = -1/\sqrt{3}$, and we start with initial estimates $x_0 = 0.0$, $x_1 = 1.5$.

Table 1.

i	ε_i	Iteration
2	0.228	U
3	-0.895×10^{-1}	M
4	0.666×10^{-2}	U
5	0.160×10^{-3}	U
6	-0.152×10^{-3}	M
7	0.702×10^{-8}	U
8	0.308×10^{-12}	U
9	-0.308×10^{-12}	M
10	$< \frac{1}{2} \times 10^{-18}$	U

From the table it is clear that $\varepsilon_6 \sim -\varepsilon_5$, $\varepsilon_9 \sim -\varepsilon_8$, illustrating the effect of the modified iteration.

We now undertake a numerical comparison of the performance of the Illinois method with those of a number of competitive robust methods. The algorithms chosen for comparison were the Regula Falsi, already described, the well-known Bisection algorithm (1), and a hybrid method frequently used and formed from a combination of the Regula Falsi and the Bisection algorithm. Here a similar approach to the Illinois method is adopted except that for the case $f_{i+1}f_i > 0$, the next value is taken as $(x_{i-1}+x_{i+1})/2$, again preventing the retention of an end point.

In an attempt to carry out the comparison over as representative a class of problems as possible, a number of equations were constructed which possess characteristics commonly encountered in root-finding problems. In each case the function used possesses a parameter n which can be varied to give a family of curves of the same type. The calculations were performed in double precision arithmetic and terminated when $|f(x)| < 0.5 \times 10^{-19}$. All the roots lie in the range $[0,1]$ and apart from Table 7 the starting values $x_0 = 0$, $x_1 = 1$ were used in each case.

Table 2. $f(x) = 2xe^{-n} + 1 - 2e^{-nx}$. *No turning points or inflexions on* $[0,1]$.

n	Number of Iterations			
	Bisection	Regula Falsi	Regula Falsi + Bisection	Illinois
1	64	23	19	9
5	64	40	21	10
15	67	41	23	11
20	67	42	23	11

Table 3. $f(x) = (1+(1-n)^2)x - (1-nx)^2$. *One turning point on* $[0,1]$.

n	Number of Iterations			
	Bisection	Regula Falsi	Regula Falsi + Bisection	Illinois
2	64	25	21	9
5	62	16	17	9
15	71	11	15	7
20	71	10	15	7

Table 4. $f(x)=x^2-(1-x)^n$. *One inflexion on* $[0,1]$.

n	Number of Iterations			
	Bisection	Regula Falsi	Regula Falsi + Bisection	Illinois
2	1	1	1	1
5	64	54	19	8
15	61	179	23	11
20	62	245	23	12

Table 5. $f(x)=(1+(1-n)^4)x-(1-nx)^4$. *One turning point and one inflexion on* $[0,1]$.

n	Number of Iterations			
	Bisection	Regula Falsi	Regula Falsi + Bisection	Illinois
2	64	40	21	10
5	71	9	13	7
15	76	6	9	6
20	77	5	9	6

Table 6. $f(x)=e^{-nx}(x-1)+x^n$. *A family of curves which lie increasingly close to the x-axis for large n.*

n	Number of Iterations			
	Bisection	Regula Falsi	Regula Falsi + Bisection	Illinois
1	64	26	19	9
5	63	114	21	9
10	57	1286	23	13
15	55	$>10^4$	21	16

Table 7. $f(x) = (nx-1)/((n-1)x)$. *A family of curves with the y-axis asymptotic.*

n	Number of Iterations			
	Bisection	Regula Falsi	Regula Falsi + Bisection	Illinois
2	59	2008	2	14
5	56	809	25	14
15	61	262	25	14
20	58	192	25	15

The starting values used here were $x_0 = 0.01$, $x_1 = 1.0$.

From these tables it is clear that the Illinois method is consistently superior, in many cases completing the calculation in fewer than half the number of iterations required by the next best method.

4. Algol procedure.

We give finally an ALGOL procedure for implementing the Illinois algorithm.

```
real procedure Illinois(f, x, x0, x1, delta);
real f, x, x0, x1, delta;
value x0, x1, delta;
comment This procedure carries out the Illinois iterative process to find a
  root of f(x) = 0, given two approximations x0 and x1 which must be
  chosen such that f(x0) and f(x1) are of opposite sign;
begin real f0, f1, fx;
  x := x1; f1 := f;
  x := x0; f0 := f;
  for x := x1 - f1 × (x1 - x0)/(f1 - f0) while abs(x - x1) > delta do
  begin
    fx := f;
    if fx × f1 < 0.0 then begin
                                x0 := x1; f0 := f1
                            end
      else f0 := f0 × 0.5;
```

```
    x1 := x; f1 := fx
  end of iterative loop;
  Illinois := x
end of procedure Illinois;
```

REFERENCE

J. F. Traub, *Iterative Methods for the Solution of Equations*, Prentice–Hall, Englewood Cliffs, N.J., 1964.

COMPUTING LABORATORY
UNIVERSITY OF BRADFORD
ENGLAND