T. E. Simos, G. Psihoyios, Ch. Tsitouras (Eds.)

ICNAAM 2006

International Conference on Numerical Analysis and Applied Mathematics 2006

Official Conference of the European Society of Computational Methods in Sciences and Engineering (ESCMSE)

Celebration in honour of the 65th birthday of Professor Peter Graves-Morris

WILEY-VCH

WILEY-VCH Verlag GmbH & Co. KGaA

Editors

Prof. T. E. Simos
University of Peloponnese
tsimos@mail.ariadne-t.gr

Dr. G. Psihoyios
Anglia Polytechnic University
g.psihoyios@ntlworld.com

Dr. Ch. Tsitouras
TEI of Chalkis
tsitoura@teihal.gr

Printed in the Federal Republic of Germany
Printed on acid-free paper

Composition, printing and bookbinding:
Druckhaus „Thomas Müntzer“, Bad Langensalza

Extending Interval Newton Method for Nonlinear Parameterized Equations

I.A. Nikas*[1], D.G. Sotiropoulos[2], and T.N. Grapsa[1]

[1] University of Patras, Department of Mathematics,GR–265 04 Patras, Greece,

[2] Ionian University, Computer Science Department, Plateia Tsirigoti 7, GR–491 00 Corfu, Greece

Received 8 June 2006, revised 23 June 2006, accepted 30 June 2006

Key words Interval Newton, zeros of nonlinear equations, interval parameters.
Subject classification 65H05, 65G20, 65G40

In this work, we consider the problem of solving nonlinear parameterized equations, where the problem parameters can be described by closed intervals. The proposed method can be assumed as an extension of the interval Newton's method and aims to find all interval zeros of a nonlinear parameterized equation. The method has been validated in various difficult problems and always returns sharp bounds in contrast to the classical interval Newton which fails to give guaranteed enclosures within a reasonable time.

1 Introduction

We address the problem of finding all zeros of the parametric equation

$$f(x;[p]) = 0, \tag{1}$$

for a continuously differentiable function $f : \mathbb{R} \rightarrow \mathbb{IR}$ and $x \in [x]$, while $[p] \in \mathbb{IR}^m$ is an interval parameter vector. An interval $[x]^* \in \mathbb{IR}$ will be called **interval root** or **interval zero** of $f(x;[p])$ if $\forall x \in [x]^*$, $\exists p^* \in [p]$ such that $f(x;p^*) = 0$.

According to our knowledge the only approach for the problem (1) is interval Newton's method [1, 2], which iterates according to the following scheme

$$N([x]^{(k)}) = m([x]^{(k)}) - \frac{f(m([x]^{(k)}))}{F'([x]^{(k)})}. \tag{2}$$

$$[x]^{(k+1)} = [x]^{(k)} \cap N([x]^{(k)}), \quad k = 0, 1, 2, \ldots \tag{3}$$

Interval Newton method is the best approach for solving nonlinear equations when the parameter vector is degenerate, although the method fails in the general case, when $0 \in f(m([x]))$ and $0 \in f'([x])$, and reduces to utilize the bisection method. In addition, for this pathological case, is difficult to define a stopping criterion, see [2, pp. 73]. In this work, we introduce an extension of the interval Newton method for the problem (1). The method is based on the simultaneous application of interval Newton method both on upper and lower bound of the function. Our approach always returns guaranteed bounds for *all* the interval zeros $[x]^*$ of $f(x;[p]) = 0$ with arbitrarily accuracy.

2 The proposed method

Since, the parametric function $f(x;[p])$ is an interval valued function, i.e. a family of functions, it can be expressed as

$$f(x;[p]) = [\underline{f}(x), \overline{f}(x)],$$

* Corresponding author: e-mail: nikas@math.upatras.gr, Phone: +30 2610 996744, Fax: +30 2610 992965

where the nonlinear functions $\underline{f} : \mathbb{R} \to \mathbb{R}$ and $\overline{f} : \mathbb{R} \to \mathbb{R}$ bound the set of functions $f(x;p)$, for every $p \in [p]$. However, in general, analytical expressions for $\underline{f}(x)$ and $\overline{f}(x)$ are not available, thus, let us define

$$f_L(x) = \inf f(x;[p]) \quad \text{and} \quad f_U(x) = \sup f(x;[p]) \tag{4}$$

then

$$f_L(x) \le \underline{f}(x) \quad \text{and} \quad \overline{f}(x) \le f_U(x),$$

for every $x \in [x]$. We note that the functions $f_L : \mathbb{R} \to \mathbb{R}$ and $f_U : \mathbb{R} \to \mathbb{R}$, as defined above, can be evaluated implicitly at any point $x \in [x]$ through $f(x;[p])$, while the corresponding interval derivatives F'_L and F'_U are contained in the interval $F'([x];[p])$. Thus, we can define their corresponding interval Newton operators:

$$N_L([x]) = m([x]) - \frac{f_L(m([x]))}{F'([x];[p])}, \quad \text{and} \tag{5}$$

$$N_U([x]) = m([x]) - \frac{f_U(m([x]))}{F'([x];[p])}. \tag{6}$$

Utilizing (5) and (6) we define a new operator H, which is constituted as the hull of N_L and N_U operators over the interval $[x]$, i.e.,

$$H([x]) = N_L([x]) \sqcup N_U([x]) \tag{7}$$

where the symbol $\sqcup$ denotes the hull of two intervals. The following theorem summarizes the most important properties of operator $H([x])$:

Theorem 2.1 *Let $F : \mathbb{R} \to \mathbb{IR}$ be a continuously differentiable m-parametric interval function, and let $[x] \in \mathbb{IR}$ be an interval. Moreover, let $f_L : \mathbb{R} \to \mathbb{R}$ and $f_U : \mathbb{R} \to \mathbb{R}$ be functions as defined in* (4)*, while $N_L([x])$ and $N_U([x])$ be the corresponding Newton operators of f_L and f_U, respectively. Then, the operator the $H([x])$:*

$$H([x]) := N_L([x]) \sqcup N_U([x])$$

has the following properties:

1. *Every interval zero $[x]^* \subseteq [x]$ of $F(x;[p])$ satisfies $[x]^* \subseteq H([x])$.*
2. *If $H([x]) \cap [x] = \emptyset$, then there exists no interval zero of $F(x;[p])$ in $[x]$.*
3. *If $H([x]) \overset{\circ}{\subset} [x]$, then there exists a unique interval zero of $F(x;[p])$ in $[x]$ and hence in $H([x])$.*

Remark 2.2 It is worth to notice that the operator $H([x])$ can also be applied to non-parametric nonlinear equations. In this case our method is reduced to the classical interval Newton method as given in (2) and (3).

3 Algorithmic formulation of the method

We present now the algorithmic formulation of the method which briefly described above. Algorithm 1 takes as input the one-dimensional m-parametric function $f(x;[p])$, the search interval $[x]$, the tolerance ε, and returns the result list $\mathcal{L}^*$ of the interval zeros $[x]^*$. Most of the steps are self explanatory, except from Steps 15 and 17 where two other algorithms are called. The procedure `ext_decomp` in Step 15 handles the case where the function is non-monotone, i.e., $0 \in F'([x],[p])$, while in different case, the routine `decomp` is called. Both of them have as an output at most three non-empty intervals, i.e., $[x]_1$, $[x]_2$ and $[r]$. The interval $[r]$ contains a continuum of points that are zeros of $f(x;p)$ for some $p \in [p]$. Therefore, $[r]$ is stored in the result list $\mathcal{L}^*$, while the remaining parts $[x]_1$, $[x]_2$ of $H([x])$ are stored in the stack $\mathcal{S}$ for further consideration.

Algorithm 1 Hull Interval Newton

HIN($F, [x], \varepsilon, \mathcal{L}^*$)

$\mathcal{L}^* := \emptyset$; $\mathcal{S} := \{ [x] \}$
while $\mathcal{S} \neq \emptyset$ **do**
  $[x] :=$ Pop($\mathcal{S}$); $\mathcal{S} := \mathcal{S} - \{[x]\}$
  if $(0 \in F([x]; [p]))$ **then**
    if $(w([x]) \leq \varepsilon)$ **then**
      $\mathcal{L}^* := \mathcal{L}^* \uplus [x]$
    else
      $H([x]) := (N_L([x]) \sqcup N_U([x])) \cap [x]$
      if $(H([x]) = [x])$ **then**
        Bisect($[x], [x]_L, [x]_U$)
        $\mathcal{S} := \mathcal{S} \uplus [x]_U$; $\mathcal{S} := \mathcal{S} \uplus [x]_L$;
      else
        $[x]_L := N_L([x]) \cap [x]$; $[x]_U := N_U([x]) \cap [x]$;
        if $(0 \in F'([x]; [p]))$ **then**
          ext_decomp($H([x]), [x], f(m([x])), [x]_L, [x]_U, [r]$)
        else
          decomp($H([x]), [x]_L, [x]_R, [x]_L, [x]_U, [r]$)
        $\mathcal{S} := \mathcal{S} \uplus [x]_U$; $\mathcal{S} := \mathcal{S} \uplus [x]_L$; $\mathcal{L}^* := \mathcal{L}^* \uplus [r]$;
return $\mathcal{L}^*$;

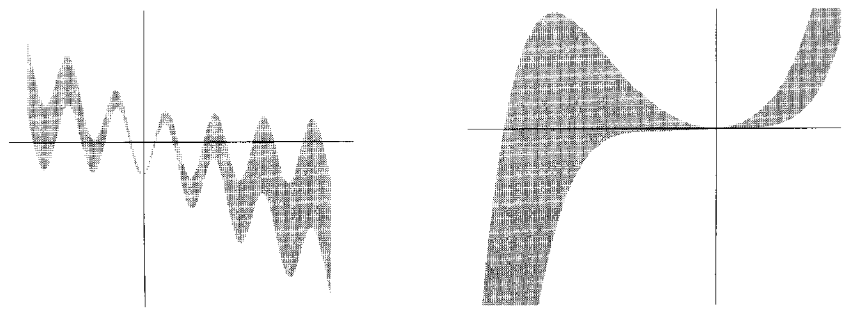

Fig. 1 The graphical representation of the reported functions f and g.

4 Numerical Results

The proposed method has been successfully implemented and tested in various difficult problems using INTLAB toolbox for self-validating algorithms [3]. In the sequel we report numerical results from two representative examples.

Example 1. We consider the following one-parametric function

$$f(x; [p]) = \sum_{j=1}^{5} j \left[\sin\left((j+1)x + j\right) + x * \sin\left((j+1)[p] + j\right)\right], \qquad [p] = [-0.1, 0.2]$$

on the interval $[x]^0 = [-2.5, 4]$. The graph of $f(x; [p])$ is shown in Figure 1, while the interval zeros along with their corresponding widths are reported in Table 1. The bounds of each interval zero has been calculated with accuracy $\varepsilon = 10^{-12}$. The computational effort for this problem was 295 function and 139 derivative evaluations.

Example 2. We consider the following 7-parametric function

$$g(x; [p]) = x(0.1x + 1)(x + p_2)\left(p_3 + p_4 x + x^2\right) + p_1\left(p_5 + p_6 x + p_7 x^2\right)$$

Table 1 Interval zeros and the corresponding widths of Example 1, $[x]^0$=[-2.5,4], $\varepsilon = 10^{-12}$

No	*Interval Root*	$w([x]^*)$
$[x]_1^*$	$[-2.371078208169876, -1.907051968931622]$	0.464026
$[x]_2^*$	$[-1.338359776592643, -0.845045355949249]$	0.493314
$[x]_3^*$	$[-0.305503420467276, -0.265028907451554]$	0.0404745
$[x]_4^*$	$[0.216867613832021, 0.257221935058413]$	0.0403543
$[x]_5^*$	$[0.624951439020535, 0.727383869146695]$	0.102432
$[x]_6^*$	$[1.278804445668570, 1.760197175644943]$	0.481393
$[x]_7^*$	$[2.340884391622359, 2.792826579953032]$	0.451942
$[x]_8^*$	$[3.403237512131658, 3.825149104834054]$	0.421912

on the interval $[x]^0 = [-30, 15]$. The components of the parameter vector $[p] = ([p]_1, \ldots, [p]_7)^\top$ have the following values: $[p]_1 = [20, 60]$, $[p]_2 = [0, 1]$, $[p]_3 = [p]_4 = [p]_5 = [p]_6 = [p]_7 = [0.2, 10]$. The nonlinear equation $g(x; [p]) = 0$ has only one interval zero

$$[x]^* = [-25.6146318439, -0.0190738795], \quad w([x]^*) = 25.5956$$

where the bounds are calculated with the same accuracy $\varepsilon = 10^{-12}$. The computational effort in this case was 168 function and 81 derivative evaluations. The graph of $g(x; [p])$ is also given in Figure 1.

References

[1] R.E. Moore, Interval Analysis. (Prentice-Hall, Inc., Englewood Cliffs, N.J., 1966).
[2] E. Hansen, Global Optimization Using Interval Analysis. (M. Dekker, 1992).
[3] S.M. Rump, INTLAB – Interval Laboratory, the Matlab toolbox for verified computations, Version 5.3, 2006. Available at: http://www.ti3.tu-harburg.de/rump/intlab/index.html