4. Bisection Routine

S. Gorn
University of Pennsylvania Computer Center

This procedure evaluates a function at the end-points of a real interval, switching to an error exit (fools exit) FLST if there is no change of sign. Otherwise it finds a root by iterated bisection and evaluation at the midpoint, halting if either the value of the function is less than the free variable \(\varepsilon\) or two successive approximations of the root differ by less than \(\varepsilon l\). \(\varepsilon\) should be chosen of the order of error in evaluating the function (otherwise time would be wasted), and \(l\) of the order of desired accuracy. \(l\) must not be less than two units in the last place carried by the machine or else indefinite cycling will occur due to round-off on bisection. Although this method is of 0 order, and therefore among the slowest, it is applicable to any continuous function. The fact that no differentiability conditions have to be checked makes it, therefore, an 'old work-horse' among routines for finding real roots which have already been isolated. The free variables \(y_1\) and \(y_2\) are (presumably) the end-points of an interval within which there is an odd number of roots of the real function \(F\). \(\alpha\) is the temporary exit for the evaluation of \(F\).

\[
\begin{align*}
\text{procedure} & \quad \text{Bisect}(y_1, y_2, \varepsilon, l, F(), \text{FLST}) = : (x) \\
\text{begin} & \quad \text{Bisect:} \\
& \quad i := 1 ; \ j := 1 ; \ k := 1 ; \ x := y_2 \\
& \quad f := F(x) ; \ \text{if} (\text{abs}(f) \leq \varepsilon) ; \ \text{return} \\
& \quad \text{go to } \gamma_1 \\
& \quad \text{First val:} \\
& \quad i := 2 ; \ f := f ; \ x := y_1 ; \ \text{go to } \alpha \\
& \quad \text{Succ val:} \\
& \quad \text{if} (\text{sign}(f) = \text{sign}(f_1)) ; \ \text{go to } b ; \ \text{go to } \gamma_2 \\
& \quad \text{Succ val:} \\
& \quad j := 2 ; \ k := 2 \\
& \quad \text{Midpoint:} \\
& \quad x := y_1/2 + y_2/2 ; \ \text{go to } a \\
& \quad \text{Reg } \delta : \\
& \quad y_2 := x \\
& \quad \text{Precision:} \\
& \quad \text{if} (\text{abs}(y_1 - y_2) \geq \varepsilon l) ; \ \text{go to } \text{Midpoint} \\
& \quad \text{return} \\
& \quad \text{Reg } \gamma : \\
& \quad y_1 := x ; \ \text{go to } \text{Precision} \\
& \quad \text{integer } (i, j, k) \\
& \quad \text{switch } \gamma := (\text{First val, Succ val}) \\
& \quad \text{switch } \delta := (\text{FLST, Reg } \delta) \\
& \quad \text{switch } \gamma := (\text{Succ val, Reg } \gamma) \\
\text{end Bisect}
\end{align*}
\]

\(\alpha\): go to \(\gamma_1\) should be go to \(\gamma_1\)

* Work supported by the U.S. Atomic Energy Commission.

After this correction was made, the program ran smoothly for \(F(x) = \cos x\), using the following parameters:

<table>
<thead>
<tr>
<th>(y_1)</th>
<th>(y_2)</th>
<th>(\varepsilon)</th>
<th>(\alpha)</th>
<th>Results</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>.001</td>
<td>.001</td>
<td>FLSTT</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>.001</td>
<td>.001</td>
<td>1.5703</td>
</tr>
<tr>
<td>1.5</td>
<td>2</td>
<td>.001</td>
<td>.001</td>
<td>1.5703</td>
</tr>
<tr>
<td>1.55</td>
<td>2</td>
<td>.1</td>
<td>.1</td>
<td>1.5500</td>
</tr>
<tr>
<td>1.5</td>
<td>2</td>
<td>.001</td>
<td>.1</td>
<td>1.5625</td>
</tr>
</tbody>
</table>

These combinations test all loops of the program.

* Work supported by the U.S. Atomic Energy Commission.

CERTIFICATION OF ALGORITHM 4

BISECTION ROUTINE (S. Gorn, Comm. ACM, March 1960)

Patty Jane Rader,* Argonne National Laboratory, Argonne, Illinois

Bisect was coded for the Royal-Precision LGP-30 computer, using an interpretive floating point system (24.2) with 28 bits of significance.

The following minor correction was found necessary.