Efficiently Computing Many Roots of a Function

D. J. Kavvadias, F. S. Makri, and M.N. Vrahatis

September 23, 2009
Finding Zeros of Functions

- Importance in practical applications
  Example: Finding local extrema (the zeros of a differential functions)

- Importance in theoretical problems
  Example: Riemann’s hypothesis (open mathematical problem)
Strengths of Algorithm

- Simplicity - easy to implement.
- Highly efficient to reduce the problem - takes advantage of large number of roots.
- Bisection method only requires the sign of the function at a point.
- Backed by robust analysis of algorithm’s expected behavior.
**Algorithm**

**Input \( \lambda \).**
Divide initial interval into two.

A - set of intervals with even number of roots.
B - intervals with odd number of roots.

Find one root in each interval in A using bisection.
Add intervals with even roots to B.

Estimate total number of roots

**Found required number of roots?**

No → Replace each interval in B by its two halves.

Yes → Output the roots.
End program
Algorithm

Input $\lambda$.
Divide initial interval into two.

A - set of intervals with even number of roots.
B - intervals with odd number of roots.

Find one root in each interval in A using bisection.
Add intervals with even roots to B.

Estimate total number of roots

Found required number of roots?

Yes
Output the roots.
End program

No
Replace each interval in B by its two halves.
Input $\lambda$

- $\lambda$ is the fraction of roots to be found.
- It is shown that

$$\lambda = \frac{2^{iN} - (2^i - 2)^N}{2^{i(N-1)+1}},$$

where $N$ is the total number of roots in the problem, $i$ is the iteration where $\lambda$ is achieved.

<table>
<thead>
<tr>
<th>$\lambda$</th>
<th>N=100</th>
<th>N=500</th>
<th>N=1000</th>
<th>N=5000</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>$i$</td>
<td>$w$</td>
<td>$i$</td>
<td>$w$</td>
</tr>
<tr>
<td>0.5</td>
<td>7</td>
<td>776</td>
<td>10</td>
<td>3508</td>
</tr>
<tr>
<td>0.7</td>
<td>8</td>
<td>1093</td>
<td>11</td>
<td>5175</td>
</tr>
<tr>
<td>0.9</td>
<td>10</td>
<td>1919</td>
<td>13</td>
<td>11313</td>
</tr>
<tr>
<td>0.95</td>
<td>11</td>
<td>2898</td>
<td>14</td>
<td>19203</td>
</tr>
</tbody>
</table>
Algorithm

Input $\lambda$.
Divide initial interval into two.

A - set of intervals with even number of roots.
B - intervals with odd number of roots.

Find one root in each interval in A using bisection.
Add intervals with even roots to B.

Estimate total number of roots

Found required number of roots?

Replace each interval in B by its two halves.

Yes

Output the roots.
End program

No
Sets of Intervals

- A is the set of intervals with an odd number of roots.
- B is the set of intervals with an even number of roots.
- Theorem giving the expected number of odd subintervals, hence the expected number of discovered roots in iteration \( i \).

\[
E_d = \frac{m^N - (m - 2)^N}{2m^{(N-1)}},
\]

where \( m \) is the number of equal subintervals.
Algorithm

Input $\lambda$.
Divide initial interval into two.

A - set of intervals with even number of roots.
B - intervals with odd number of roots.

Find one root in each interval in A using bisection.
Add intervals with even roots to B.

Estimate total number of roots

Found required number of roots?

No
Replace each interval in B by its two halves.

Yes
Output the roots.
End program
Find roots using Bisection Method

- Bisection Method always converges in interval \((a, b)\) and is globally convergent.
- Asymptotically the best possible rate of convergence in the worst case.
- Use simplified version of Bisection Method

\[ x_{i+1} = x_i + c \text{sgn} f(x_i)/2^{i+1}, \]

where \(c = \text{sgn} f(a)(b - a)\)
Input $\lambda$. Divide initial interval into two.

A - set of intervals with even number of roots. B - intervals with odd number of roots.

Find one root in each interval in A using bisection. Add intervals with even roots to B.

Estimate total number of roots

Found required number of roots?

Yes

Output the roots. End program

No

Replace each interval in B by its two halves.
Estimate total number of roots

The total number roots $N$ is estimated by

$$N = \frac{N_{\text{lower}} + N_{\text{upper}}}{2}$$

where $N_{\text{lower}}$ and $N_{\text{upper}}$ are the solutions to

$$p_{\text{lower}} = \frac{1 - (1 - 2^{1-i})N_{\text{lower}}}{2} \quad \text{and} \quad p_{\text{upper}} = \frac{1 - (1 - 2^{1-i})N_{\text{upper}}}{2}$$

where

$$p_{\text{lower}} = \frac{k - z_{\alpha/2} \sqrt{\frac{k(m-k)}{m}}}{m} \quad \text{and} \quad p_{\text{upper}} = \frac{k + z_{\alpha/2} \sqrt{\frac{k(m-k)}{m}}}{m},$$

where $k = \text{car}(A)$ and $z_{\alpha/2}$ is the standard normal value.
Algorithm

1. Input $\lambda$. Divide initial interval into two.

2. A - set of intervals with even number of roots. B - intervals with odd number of roots.

3. Find one root in each interval in A using bisection. Add intervals with even roots to B.

4. Estimate total number of roots

5. Found required number of roots?
   - Yes: Output the roots. End program
   - No: Replace each interval in B by its two halves.
Stopping Criterion

Algorithm stops if either

- $N_{lower}$ or $N_{upper}$ is infinite, or
- the number of roots found is at least $\lambda N$
Algorithm

Input $\lambda$.
Divide initial interval into two.

A - set of intervals with even number of roots.
B - intervals with odd number of roots.

Find one root in each interval in A using bisection.
Add intervals with even roots to B.

Estimate total number of roots

Found required number of roots?
No
Replace each interval in B by its two halves.
Yes
Output the roots.
End program
Conclusion and Further Research

- Algorithm discovers roots effectively until the problem has been greatly reduced.
- Its simplicity results in fairly easy programming.
- It may be possible to extend method to higher dimensions.
- Consider an arbitrary distribution of roots.
- Apply algorithm on natural problem.