Computation of Interval Extensions Using Berz-Taylor Polynomial Models

Hongkun Liang and Mark A. Stadtherr*

Department of Chemical Engineering
University of Notre Dame
Notre Dame, IN 46556 USA

AIChE Annual Meeting, Los Angeles, CA, Nov. 12–17, 2000
Session 272 (Numerical Analysis): Paper 272f

*Fax: (219)631-8366; E-mail: markst@nd.edu

Outline

- Background: Interval Analysis
- Context: Interval Newton/Generalized Bisection Methods
- Berz-Taylor Polynomial Approach for Interval Extensions
- Examples
- Concluding Remarks

Interval Analysis

- A real interval $X = [a, b] = \{x \in \Re \mid a \leq x \leq b\}$ is a segment on the real number line

- An interval vector $\mathbf{X} = (X_1, X_2, ..., X_n)^T$ is an n-dimensional rectangle or "box"

- Basic interval arithmetic for $X = [a, b]$ and $Y = [c, d]$ is X op $Y = \{x$ op $y \mid x \in X,\ y \in Y\}$

$$X + Y = [a + c, b + d]$$

$$X - Y = [a - d, b - c]$$

$$X \times Y = [min(ac, ad, bc, bd), max(ac, ad, bc, bd)]$$

$$X \div Y = [a, b] \times [1/d, 1/c], \quad 0 \notin Y$$

- For $X \div Y$ when $0 \in Y$, an extended interval arithmetic is available

Interval Analysis (continued)

- Computed endpoints are *rounded out* to guarantee the enclosure

- Interval elementary functions (e.g., $\exp(X)$, $\log(X)$, etc.) are also available

- The **interval extension** $F(\mathbf{X})$ encloses all values of $f(\mathbf{x})$ for $\mathbf{x} \in \mathbf{X}$; that is,

$$F(\mathbf{X}) \supseteq \{f(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}\}$$

- Interval extensions can be computed using interval arithmetic (the "natural" interval extension), or with other techniques (e.g., **Berz-Taylor polynomial models**)

- Context: Nonlinear equation solving and global optimization using interval Newton/generalized bisection (IN/GB) approach

Interval Newton/Generalized Bisection

- Given initial bounds on each variable, IN/GB can:
  - Find (enclose) any and all solutions to a nonlinear equation system to a desired tolerance
  - Determine that there is no solution of a nonlinear equation system
  - Find the global optimum of a nonlinear objective function

- This methodology:
  - Provides a **mathematical guarantee** of reliability
  - Deals automatically with rounding error, and so also provide a **computational guarantee** of reliability
  - Represents a particular type of branch-and-prune algorithm (or branch-and-bound for optimization)

Interval Approach (Cont'd)

Problem: Solve $\mathbf{f}(\mathbf{x}) = \mathbf{0}$ for all roots in initial interval $\mathbf{X}^{(0)}$

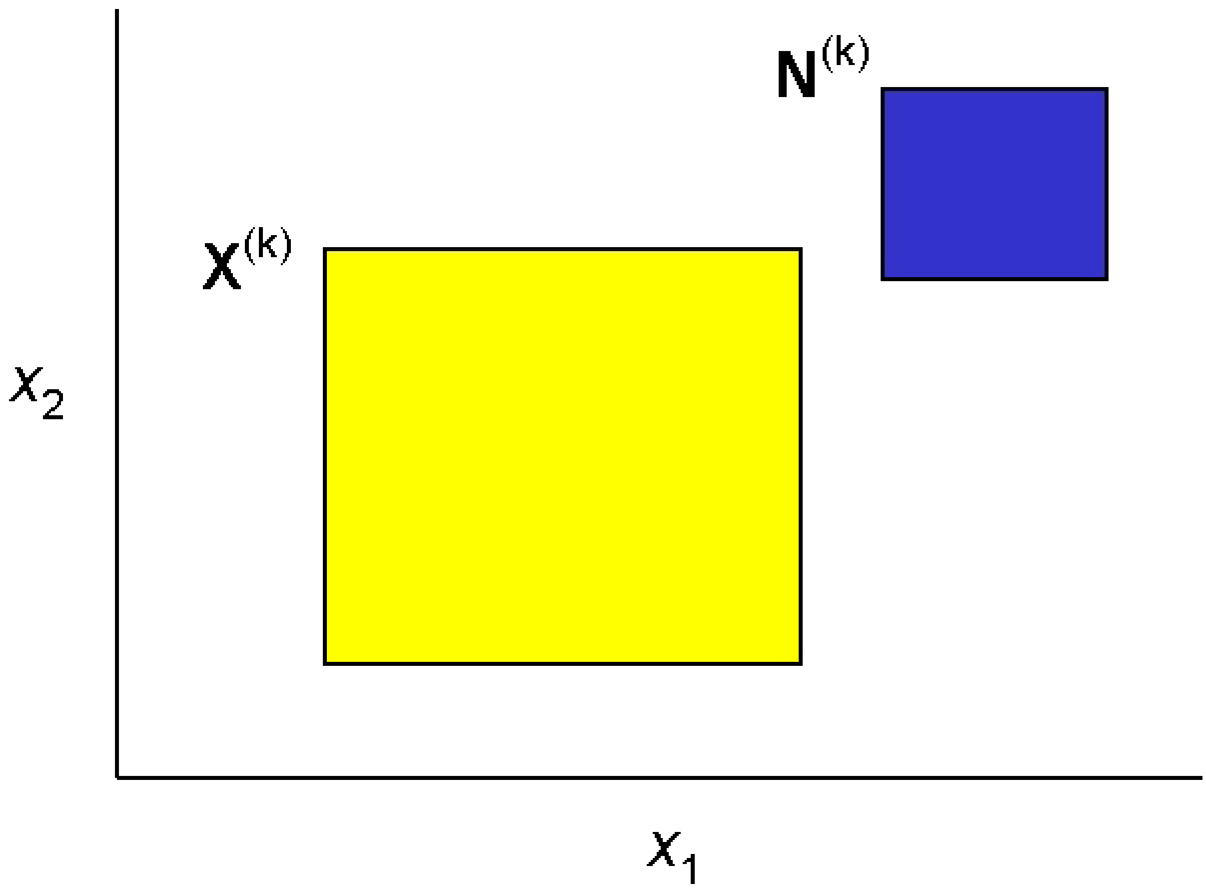
Basic iteration scheme: For a particular subinterval (box), $\mathbf{X}^{(k)}$, arising from some branching (bisection) scheme, perform **root inclusion test**:

- Compute the interval extension (range) of each function in the system
- If there is any range for which 0 is not an element, delete (prune) the box
- If 0 is an element of every range, then compute the *image*, $\mathbf{N}^{(k)}$, of the box by solving the interval Newton equation

$$F'(\mathbf{X}^{(k)})(\mathbf{N}^{(k)} - \mathbf{x}^{(k)}) = -\mathbf{f}(\mathbf{x}^{(k)})$$

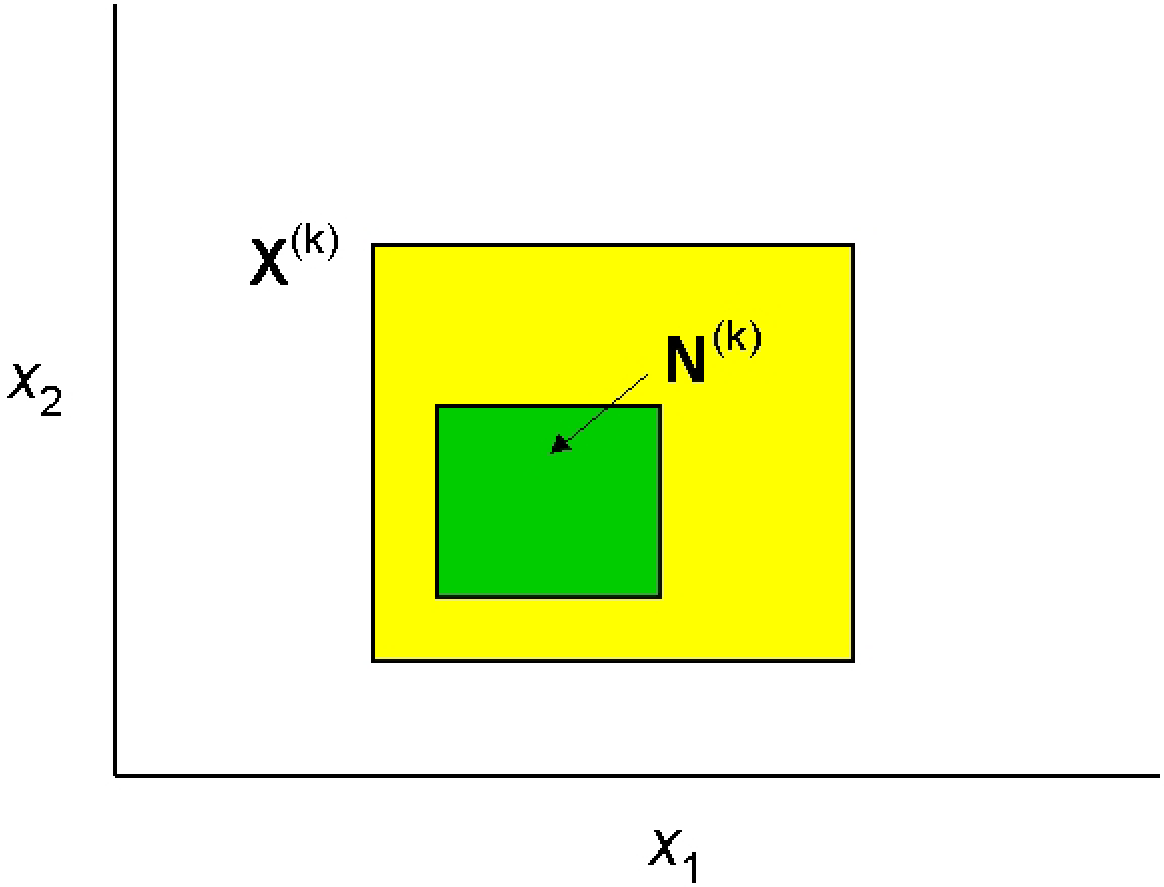
- $\mathbf{x}^{(k)}$ is some point in the interior of $\mathbf{X}^{(k)}$
- $F'\left(\mathbf{X}^{(k)}\right)$ is an interval extension of the Jacobian of $\mathbf{f}(\mathbf{x})$ over the box $\mathbf{X}^{(k)}$

Interval Newton Method



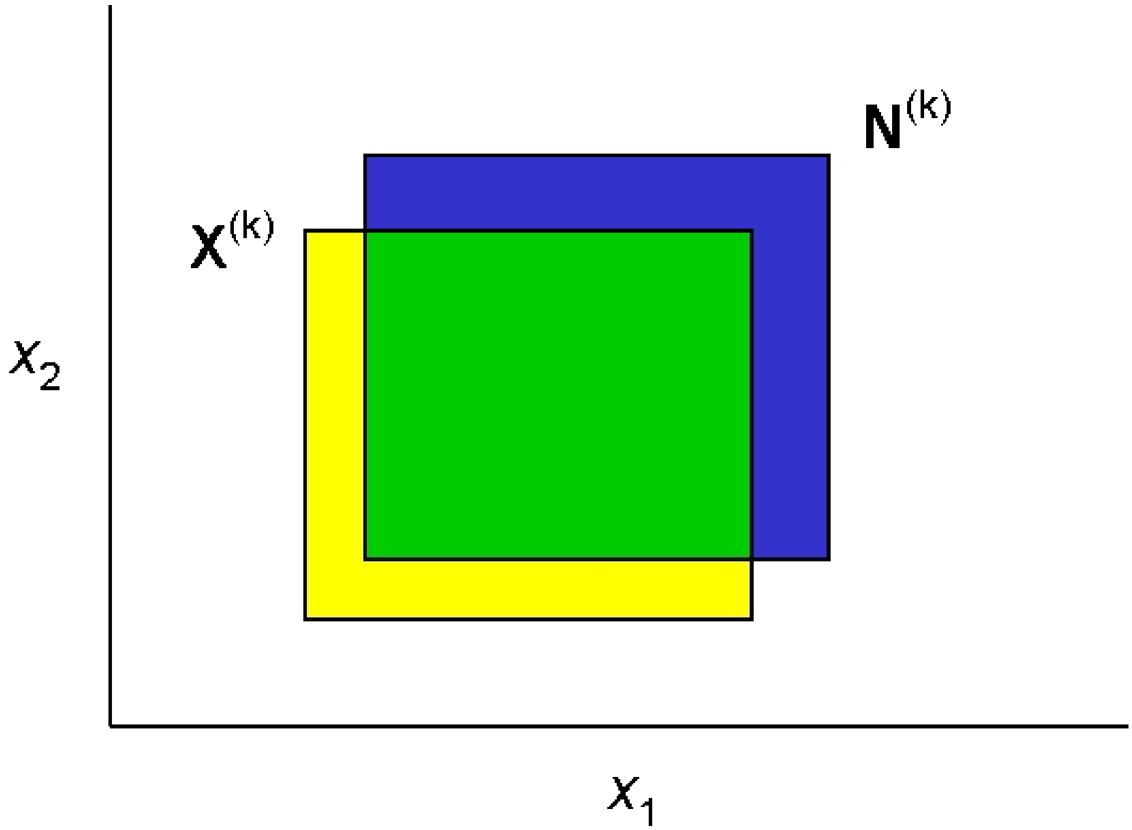
- There is no solution in $\mathbf{X}^{(k)}$

Interval Newton Method



- There is a *unique* solution in $\mathbf{X}^{(k)}$
- This solution is in $\mathbf{N}^{(k)}$
- Point Newton method will converge to solution

Interval Newton Method



- Any solutions in $\mathbf{X}^{(k)}$ are in intersection of $\mathbf{X}^{(k)}$ and $\mathbf{N}^{(k)}$
- If intersection is sufficiently small, repeat root inclusion test
- Otherwise, bisect the intersection and apply root inclusion test to each resulting subinterval

Interval Approach (Cont'd)

- For best efficiency, need to compute interval extensions that tightly bound function ranges

- Some chemical engineering problems solved using IN/GB
  - Fluid phase stability and equilibrium (e.g. Hua et al., 1998)
  - Location of azeotropes (Maier *et al.*, 1998, 1999, 2000)
  - Location of mixture critical points (Stradi *et al.*, 2000)
  - Solid-fluid equilibrium (Xu *et al.*, 2000)
  - Parameter estimation (Gau and Stadtherr, 1999, 2000)
  - Phase behavior in porous materials (Maier *et al.*, 2000)
  - General process modeling problems—up to 163 equations (Schnepper and Stadtherr, 1996)

Computing Interval Extensions

- The **interval extension** $F(\mathbf{X})$ encloses all values of $f(\mathbf{x})$ for $\mathbf{x} \in \mathbf{X}$; that is,

$$F(\mathbf{X}) \supseteq \{f(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}\}$$

- Interval extensions can be computed using interval arithmetic (the “natural” interval extension)

- However, if a variable occurs more than once in an expression, the natural interval extension may not tightly bound the true range; e.g.,

$$f(x) = x - x$$

$$X = [1, 3]$$

$$F(X) = [1, 3] - [1, 3] = [-2, 2]$$

Interval Extensions (cont'd)

- Another example: $f(x) = x/(x-1)$ evaluated for $X = [2, 3]$

- The natural interval extension is

$$
\begin{aligned}
F([2,3]) &= [2,3]/([2,3]-1) \\
&= [2,3]/[1,2] = [1,3]
\end{aligned}
$$

- Rearranged $f(x) = x/(x-1) = 1 + 1/(x-1)$, the natural interval extension is

$$
\begin{aligned}
F([2,3]) &= 1 + 1/([2,3]-1) \\
&= 1 + 1/[1,2] \\
&= 1 + [0.5, 1] = [1.5, 2]
\end{aligned}
$$

  which is the true range.

- This is the "dependency" problem. In the first case, each occurrence of x was treated as a independent interval in performing interval arithmetic.

Some Methods for Computing Tighter Interval Extensions

- Try to rearrange to eliminate dependencies
  - Potential to obtain exact bounds
  - May not be possible in many cases

- Try to identify monotonicity or convexity
  - Potential to obtain exact bounds
  - May be difficult due to overestimation in computing derivative bounds

- Use centered or mean-value forms (e.g., Ratschek and Rokne, 1984)

- ⇒ Use Taylor polynomial models (e.g. Berz and colleagues)

- Etc.

Approach of Berz and Colleagues

- Compute interval extension using Taylor polynomial model plus remainder bound

- Construct models of complex functions from models of simpler functions by operating on Taylor model coefficients and remainder bounds

- Implement using automatic differentiation (AD) and AD-like techniques

- Can provide efficient control of dependency problems and tight enclosures of complicated functions

- Applied successfully to problems in physics and astrophysics (e.g., beam physics, galaxy dynamics, orbital stability): Hoffstätter and Berz (1996, 1998), Makino and Berz (1999)

Berz-Taylor Model

- The Taylor model $T_f(\mathbf{X})$ is an interval extension of $f(\mathbf{x})$ on $\mathbf{X}$: $T_f(\mathbf{X}) \supseteq \{f(\mathbf{x}) \mid \mathbf{x} \in \mathbf{X}\}$

$$
\begin{aligned}
T_f &= P_f \text{ (Taylor polynomial)} + R_f \text{ (remainder)} \\
&= \sum_{i=0}^{n} \frac{1}{i!}[(\mathbf{X}-\mathbf{x}_0)\cdot\nabla]^i f(\mathbf{x}_0) + \frac{1}{(n+1)!}[(\mathbf{X}-\mathbf{x}_0)\cdot\nabla]^{n+1} f\left(\mathbf{x_0} + (\mathbf{X}-\mathbf{x_0})\mathbf{\Theta}\right) \\
\mathbf{x}_0 &= (x_{1_0}, \ldots, x_{m_0})^{\mathrm{T}}, \quad \mathbf{\Theta} \in [0,1] \\
[\mathbf{g}\cdot\nabla]^{\mathbf{k}} &= \sum_{\substack{j_1+\ldots+j_m=k \\ 0\le j_1,\ldots,j_m\le k}} \frac{k!}{j_1!\ldots j_m!} g_1^{j_1}\ldots g_m^{j_m} \frac{\partial^k}{\partial x_1^{j_1}\ldots x_m^{j_m}}
\end{aligned}
$$

Berz-Taylor Model

- Basic operations

$$f \pm g \in (P_f + R_f) \pm (P_g + R_g) = (P_f \pm P_g) + (R_f \pm R_g)$$

$$f * g \in (P_f + R_f) * (P_g + R_g) \subseteq$$
$$P_f * P_g + P_f * R_g + P_g * R_f + R_f * R_g = P_{f.g} + R_{f.g}$$

$$P_{f.g} = P_f * P_g \text{ (terms of order } \leq n)$$
$$R_{f.g} = P_f * R_g + P_g * R_f + R_f * R_g + P_f * P_g \text{ (terms of order } > n)$$

- Division operation and intrinsic functions can also be defined

Example 1

- $f(x) = x - x$

- Compute interval extension $F(X)$ for $X = [1,3]$

- Using interval arithmetic

$$F(X) = [1,3] - [1,3] = [-2,2]$$

- Using Taylor model (first order, $x_0 = 2$)

$$\begin{aligned} F(X) &= \{x_0 + (X - x_0)\} - \{x_0 + (X - x_0)]\} \\ &= (2-2) + (1-1)(X - x_0) \\ &= [0,0] \end{aligned}$$

Example 2

- $f(x) = x \ln x$ for $X = [0.3, 0.4]$
- Using interval arithmetic

$$F(X) = [0.3, 0.4] * [-1.2040, -0.9163] = [-0.482, -0.275]$$

- Using Taylor model (3rd order, $x_0 = 0.35$)

$$\begin{aligned} F(X) &= -0.3674 - 0.04982 * (X - x_0) + 1.4286 * (X - x_0)^2 \\ &\quad -1.3605 * (X - x_0)^3 + [-1.25 * 10^{-4}, 4.86 * 10^{-5}] \\ &= [-0.370, -0.361] \end{aligned}$$

- The exact range is [-0.368,-0.361]

Example 3

- $f = \sin(2x) + \sin(3x) + \cos(4x)$ for $X = [0.2, 0.5]$
- Using interval arithmetic

$$\begin{aligned} F(X) &= [0.3894, 0.8415] + [0.5481, 1.1006] + [-0.4161, 0.6967] \\ &= [0.5379, 2.5357] \end{aligned}$$

- The exact range is [1.4228,1.7094]

Example 3 (cont'd)

- Using Taylor model (3rd order, $x_0 = 0.35$)

$$
\begin{aligned}
F(X) &= \Big(0.6442 + 1.5297(X - x_0) - 1.2884(X - x_0)^2 - 1.0198(X - x_0)^3\Big) \\
&\quad + \Big(0.8674 + 1.4927(X - x_0) - 3.9034(X - x_0)^2 - 2,2391(X - x_0)^3\Big) \\
&\quad + \Big(0.1700 - 3.9418(X - x_0) - 1.3597(X - x_0)^2 + 10.5115(X - x_0)^3\Big) \\
&\quad + \Big([0, 2.83 * 10^{-4}] + [0, 1.704 * 10^{-3}, 0] + [-0.002247, 0.003762]\Big) \\
&= 1.6816 - 0.9194(X - x_0) - 6.5516(X - x_0)^2 + 7.2526(X - x_0)^3 \\
&\quad + [-0.002247, 0.005751]
\end{aligned}
$$

resulting in [1.3696, 1.8497]

- The exact range is [1.4228,1.7094]

Example 4

- Using IN/GB approach, solve

$$f = x \ln x + 0.36787 = 0$$

  for all roots in $X^{(0)} = [0.2, 0.5]$

- Two roots are found

- Using interval arithmetic to get interval extensions, the required number of root inclusion tests (NTest) is 41

- Using 3rd order Taylor models to get interval extensions, the required number of root inclusion tests (NTest) is 18

Example 5

- Use IN/GB to solve $f(x) = 0$ for all roots in $X^{(0)} = [0, 20]$, where

  Power Form

$$f(x) = \frac{481.6282 - 533.2807x + 166.197x^2 - 21.1115x^3 + 1.1679x^4 - 0.023357x^5}{e^x}$$

  Nested (Horner) Form

$$f(x) = \frac{481.6282 - x(533.2807 + x(166.197 - x(21.1115 + x(1.1679 - 0.023357x))))}{e^x}$$

- Five real roots are found in [0,20]

Example 5 (cont'd)

		Power Form	Nested Form
Interval Arithmetic	Ntest CPU (sec.)	748 0.24	353 0.03
3rd-Order Taylor Model	Ntest CPU (sec.)	186 0.71	136 0.38

CPU time on Sun Ultra 30

Example 6

- Use IN/GB to solve $f(x) = 0$ for all roots in $X^{(0)} = [0, 20]$, where

Power Form

$$\begin{aligned} f(x) &= -(7.79082 \times 10^{-16})x^{10} - (2.888 \times 10^{-5})x^9 + 0.0025992x^8 - 0.09836x^7 \\ &\quad +2.0334x^6 - 24.9679x^5 + 185.3593x^4 - 809.8583x^3 + 1925.5244x^2 \\ &\quad -2101.828x + 688.0609 \end{aligned}$$

- Nine real roots are found in [0,20]

Example 6 (cont'd)

		Power Form	Nested Form
Interval Arithmetic	Ntest CPU (sec.)	9956 5.09	3174 1.23
3rd Order Taylor Model	Ntest CPU (sec.)	145 1.11	126 0.36

CPU time on Sun Ultra 30

Example 7

- Use IN/GB to solve Gritton's Second Problem (Gritton, 1992; Kearfott, 1997)
- 19 component flash problem (Shacham and Kehat, 1972)
- Can be reduced to degree-18 polynomial in one variable (Gritton, 1992)
- Find all roots in $X^{(0)} = [-20, 20]$: 18 real roots found
- Find all roots in $X^{(0)} = [0, 1]$: one real root found

Example 7 (cont'd)

Results for [-20,20]

		Power Form	Nested Form
Interval Arithmetic	Ntest CPU (sec.)	52255 54.43	14721 9.28
3rd OrderTaylor Model	Ntest CPU (sec.)	529 11.19	400 1.97

CPU time on Sun Ultra 30

Example 7 (cont'd)

Results for [0,1]

		Power Form	Nested Form
Interval Arithmetic	Ntest CPU (sec.)	217 0.23	89 0.05
3rd OrderTaylor Model	Ntest CPU (sec.)	20 0.54	16 0.09

CPU time on Sun Ultra 30

Issues

- Use of Taylor models may not be effective when applied to large interval domains

- Computational overhead needs to be reduced

- Use of monotonicity or convexity properties may yield tighter bounds

- Need strategy for deciding when to use Taylor models and when to use other means for computing interval extensions

Concluding Remarks

- Berz-Taylor approach provides a methodology for reducing overestimation in functions with a high degree of dependency

- In equation-solving problems using IN/GB, the Berz-Taylor approach can lead to large reductions in root inclusion tests (fewer leaves in binary search tree)

- To better achieve CPU time savings, improved implementation to reduce overhead is needed

Acknowledgments

– ACS PRF 30421-AC9 and 35979-AC9
– NSF EEC97-00537-CRCD
– EPA R826-734-01-0

For more information:

– Contact Prof. Stadtherr at markst@nd.edu
– See also
  http://www.nd.edu/~markst