High Order Range Bounding Using Taylor Models

- LDB and QDB -

Kyoko Makino and Martin Berz

Department of Physics
University of Illinois at Urbana-Champaign

Department of Physics and Astronomy
Michigan State University

One Dimensional TM Range Bounders

(Work with Youn-Kyung Kim at Michigan State Univ.)

It is relatively easy to produce efficient range bounders of up to sixth order

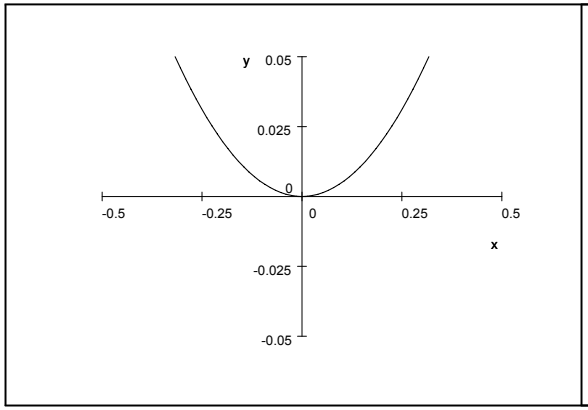
- There are well-known formulas for zeros of polynomials up to order 4
- Apply these to the derivatives and find all real roots
- Yields all critical points of polynomials up to order 5
- Evaluating polynomial at these and boundary points, and take min, max

Care has to be taken about the following aspects:

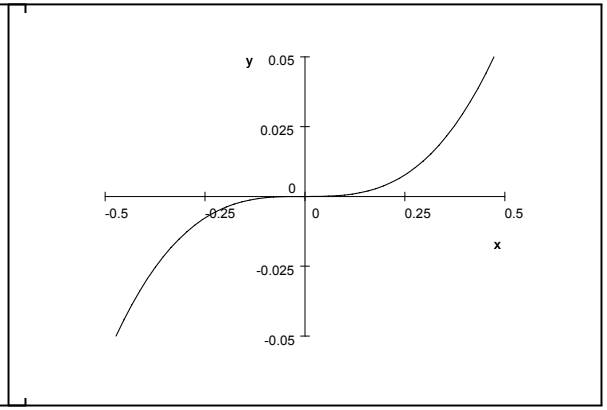
- Obviously, Evaluate formulas by interval arithmetic
- Branches in the code because of different sub-cases:
  - follow each one separately, or
  - slightly perturb the original polynomial so that branches disappear

  $P^*(x) = P(x) + \sum_{i=1}^{5} \varepsilon_i \; x^i$, then $B(P) \subset B(P^*) - B\left(\sum_{i=1}^{5} \varepsilon_i x^i\right)$
- Only interested in real roots: re-write expressions to avoid complex roots
- Cleverly write formulas to minimize width of enclosures of critical points (cancellation problem)

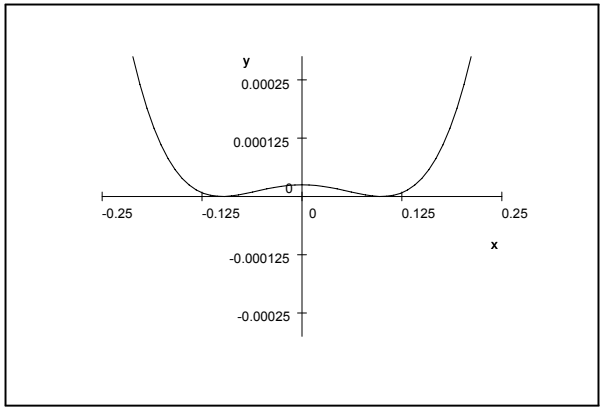
Examples



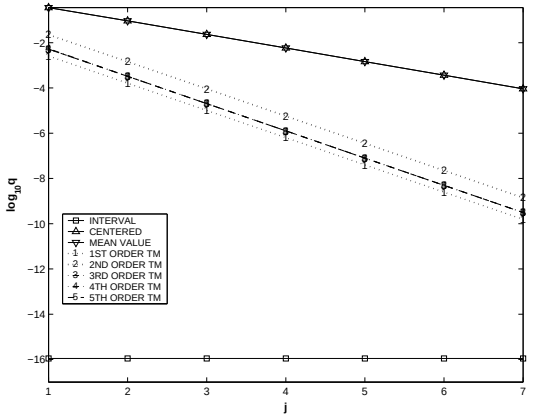
$1 - \cos(x)$

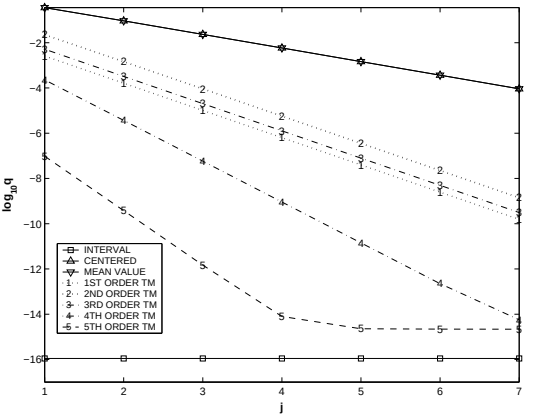


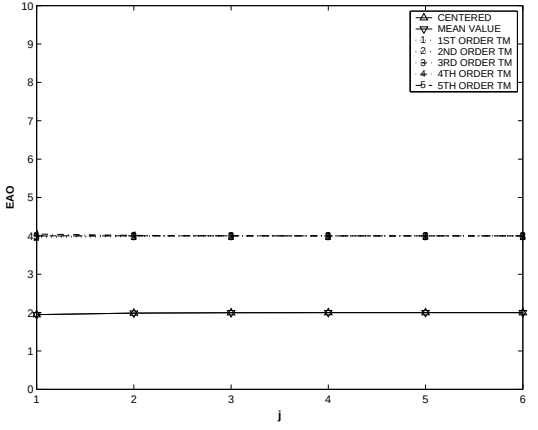
$(1 - \cos(x)) \sin(x)$

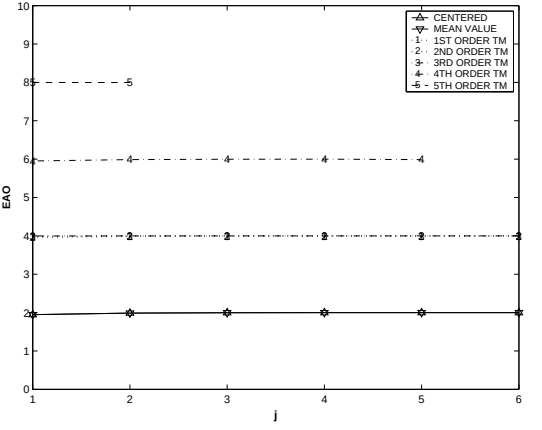


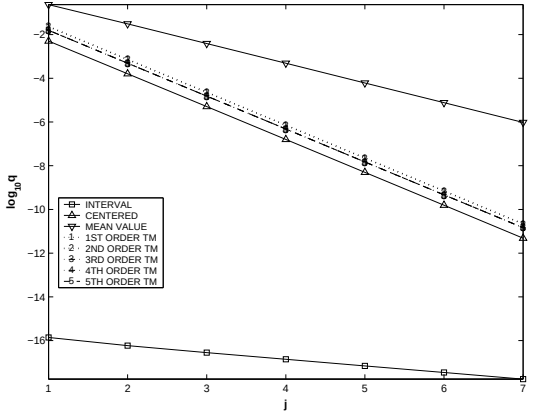
$(1 - \cos(x + 0.1)) \cdot (1 - \cos(x - 0.1))$

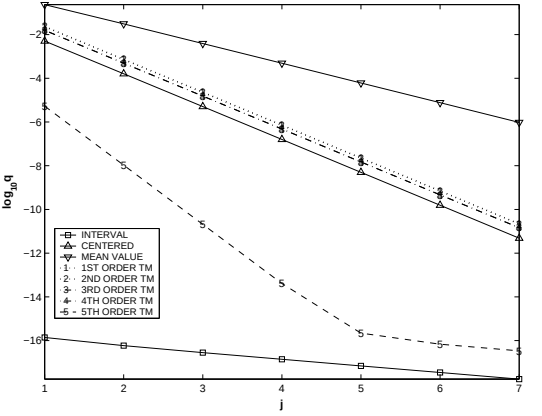

INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
2ND ORDER TM
3RD ORDER TM
4TH ORDER TM
5TH ORDER TM
log10 q
j

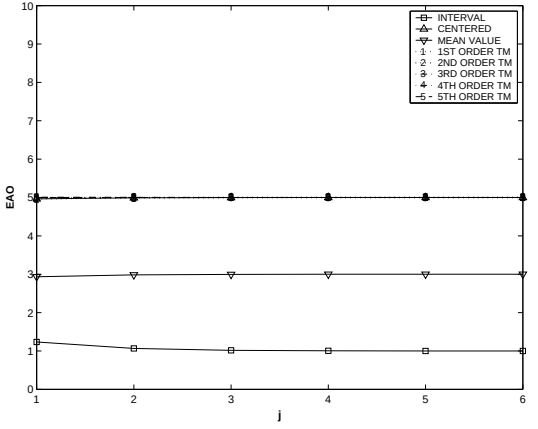

INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
2ND ORDER TM
3RD ORDER TM
4TH ORDER TM
5TH ORDER TM
log10 q
j

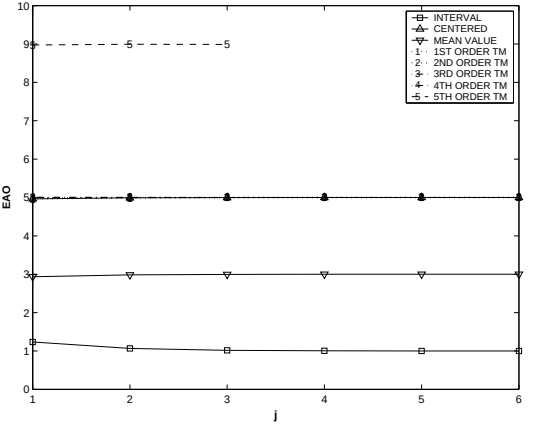

CENTERED
MEAN VALUE
1ST ORDER TM
2ND ORDER TM
3RD ORDER TM
4TH ORDER TM
5TH ORDER TM
EAO
j

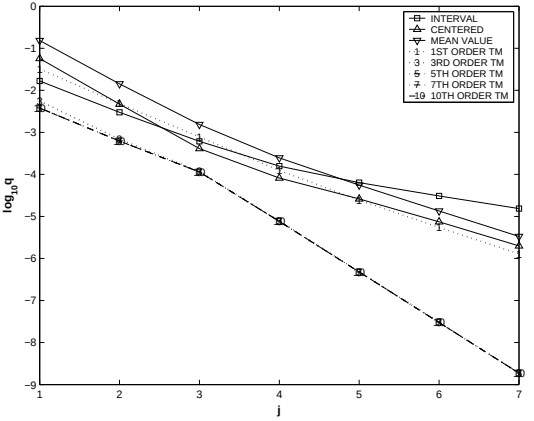

CENTERED
MEAN VALUE
1ST ORDER TM
2ND ORDER TM
3RD ORDER TM
4TH ORDER TM
5TH ORDER TM
EAO
j

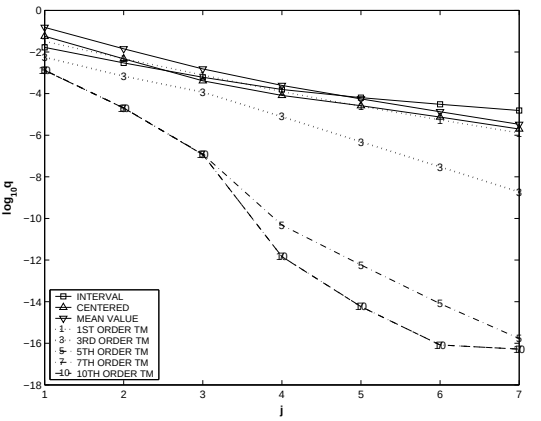

INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
2ND ORDER TM
3RD ORDER TM
4TH ORDER TM
5TH ORDER TM
log10 q
j

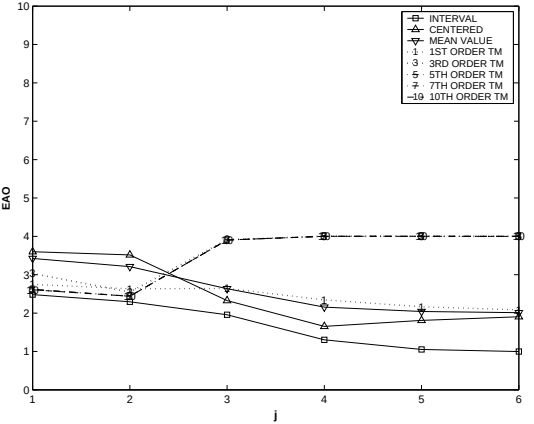

INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
2ND ORDER TM
3RD ORDER TM
4TH ORDER TM
5TH ORDER TM
log10 q
j

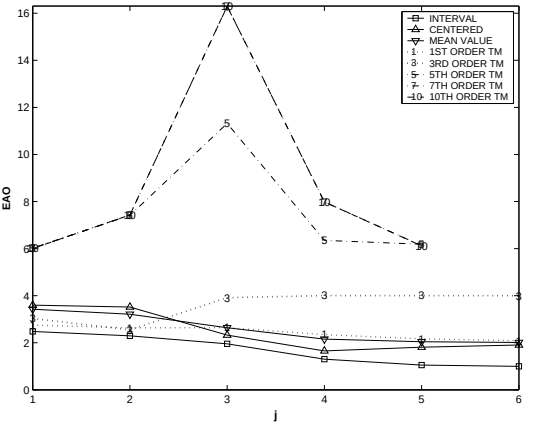

INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
2ND ORDER TM
3RD ORDER TM
4TH ORDER TM
5TH ORDER TM
EAO
j


INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
2ND ORDER TM
3RD ORDER TM
4TH ORDER TM
5TH ORDER TM
EAO
j


INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
5TH ORDER TM
7TH ORDER TM
10TH ORDER TM
log10q
j

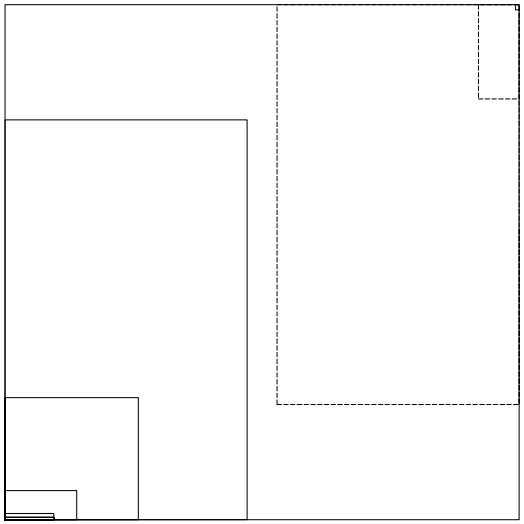

INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
5TH ORDER TM
7TH ORDER TM
10TH ORDER TM
log10q
j


INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
5TH ORDER TM
7TH ORDER TM
10TH ORDER TM
EAO
j


INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
5TH ORDER TM
7TH ORDER TM
10TH ORDER TM
EAO
j

The Linear Dominated Bounder (LDB)

- The linear part of TM polynomial is the leading part, so is it for range bounding.
- The idea is easily extended to multi-dimensional.
- Use the linear part as a guideline for domain splitting and elimination.
- The reduction of the size of interested box works multi-dimensionally and automatically. Thus, the reduction rate is fast.
- Even there is no linear part in the original TM, by shifting the expansion point, normally the linear part is introduced.

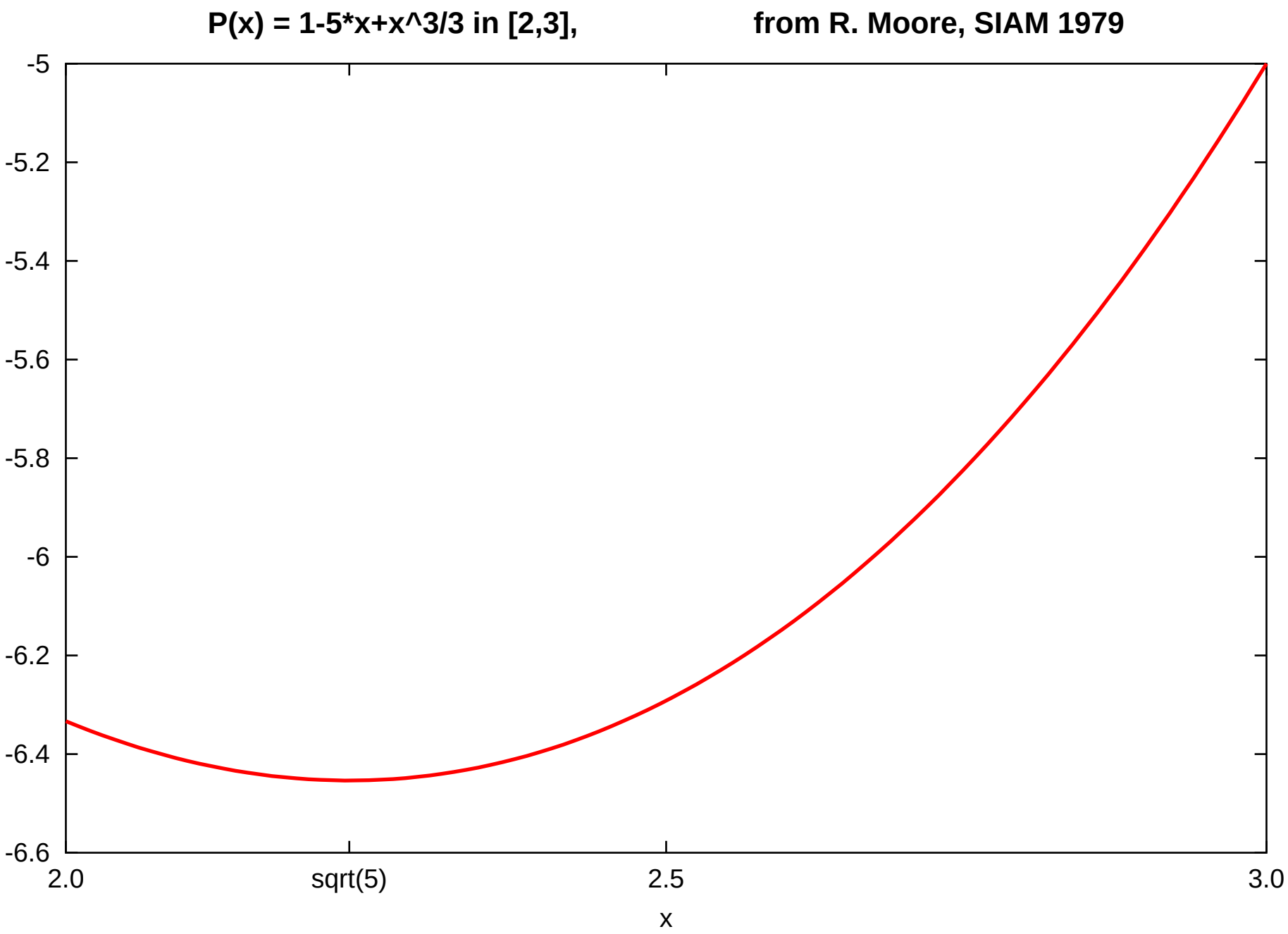


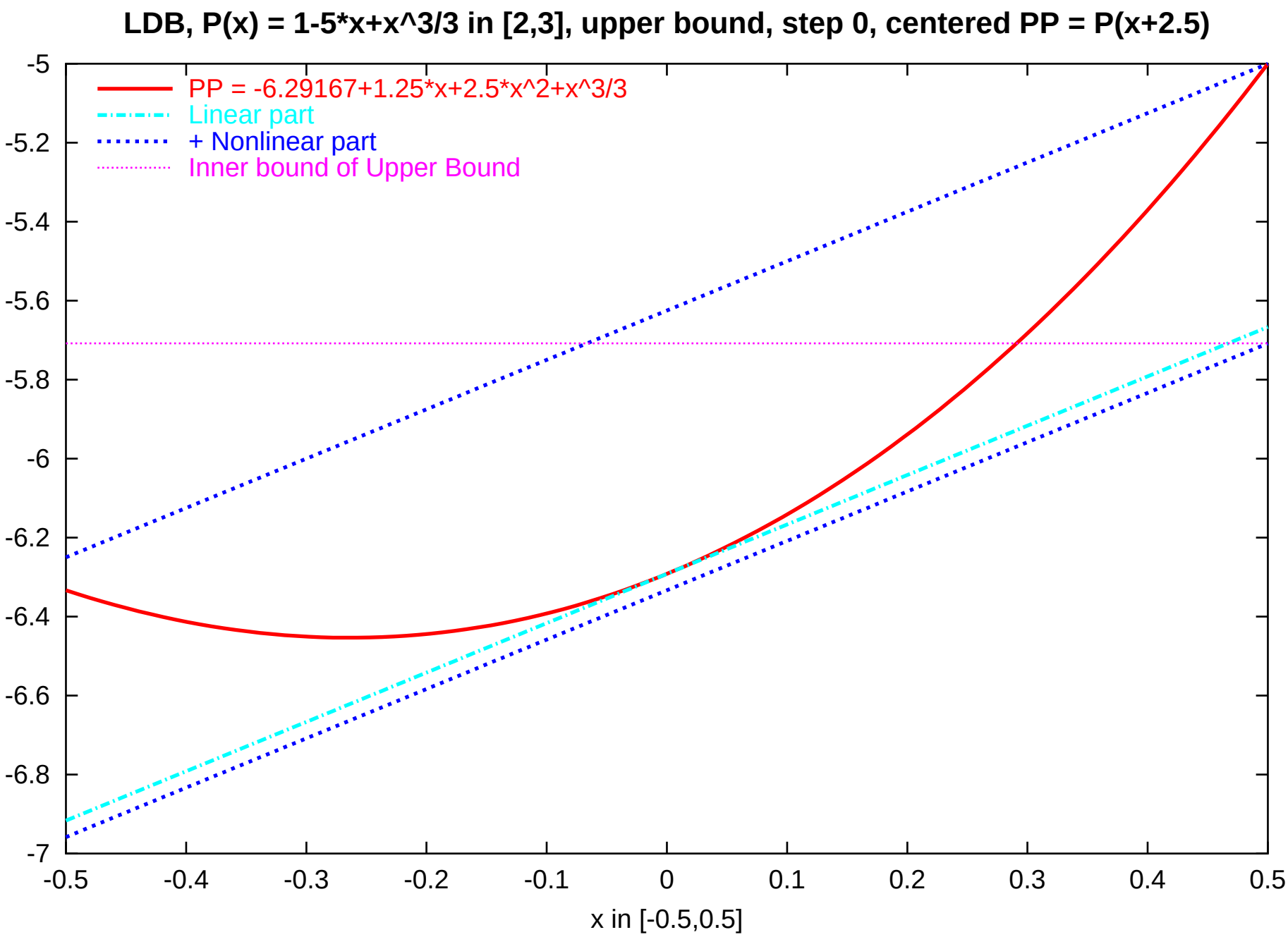
LDB Algorithm

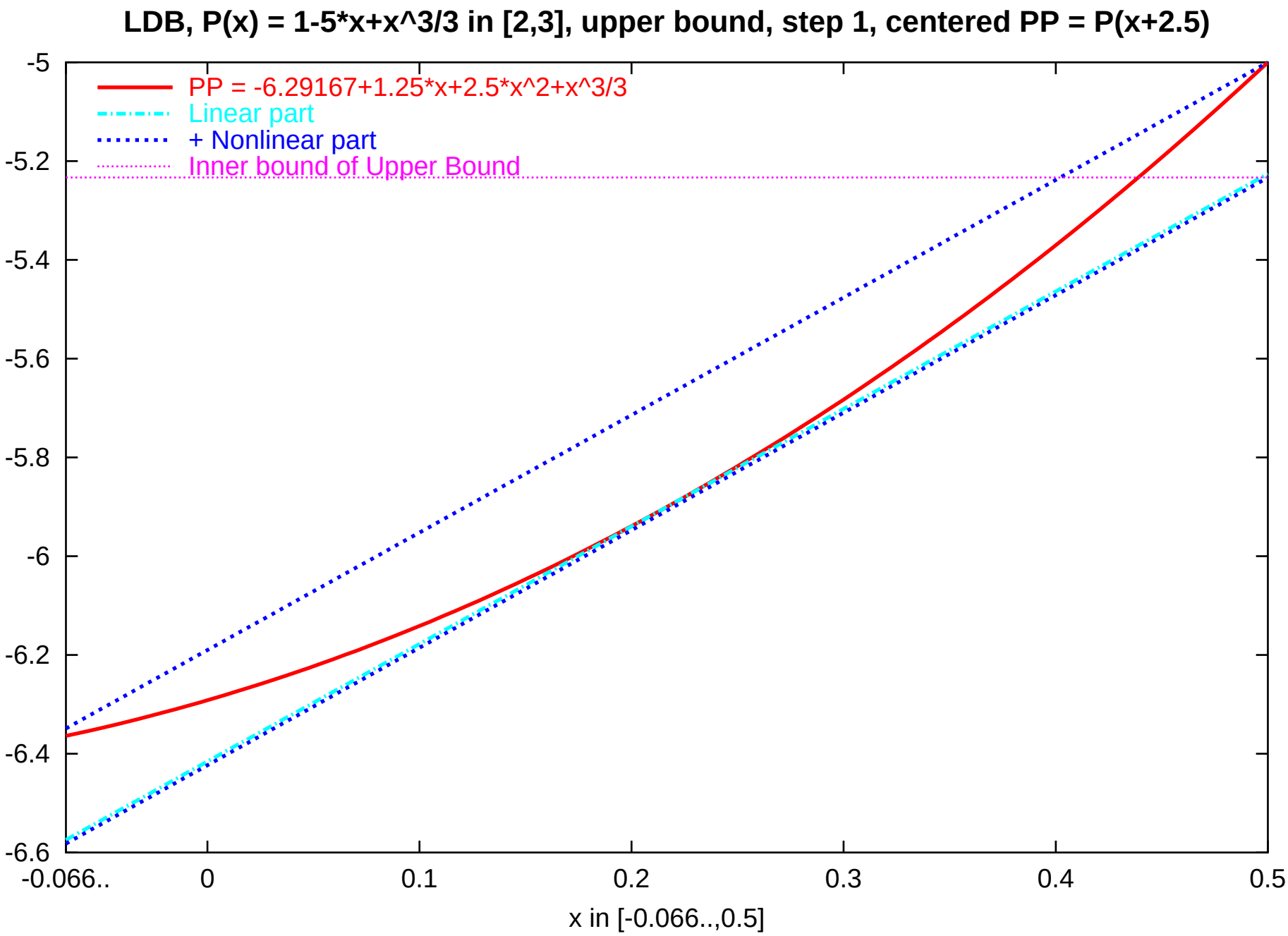
Wlog, find the upper bound of maximum of a polynomial P in B.

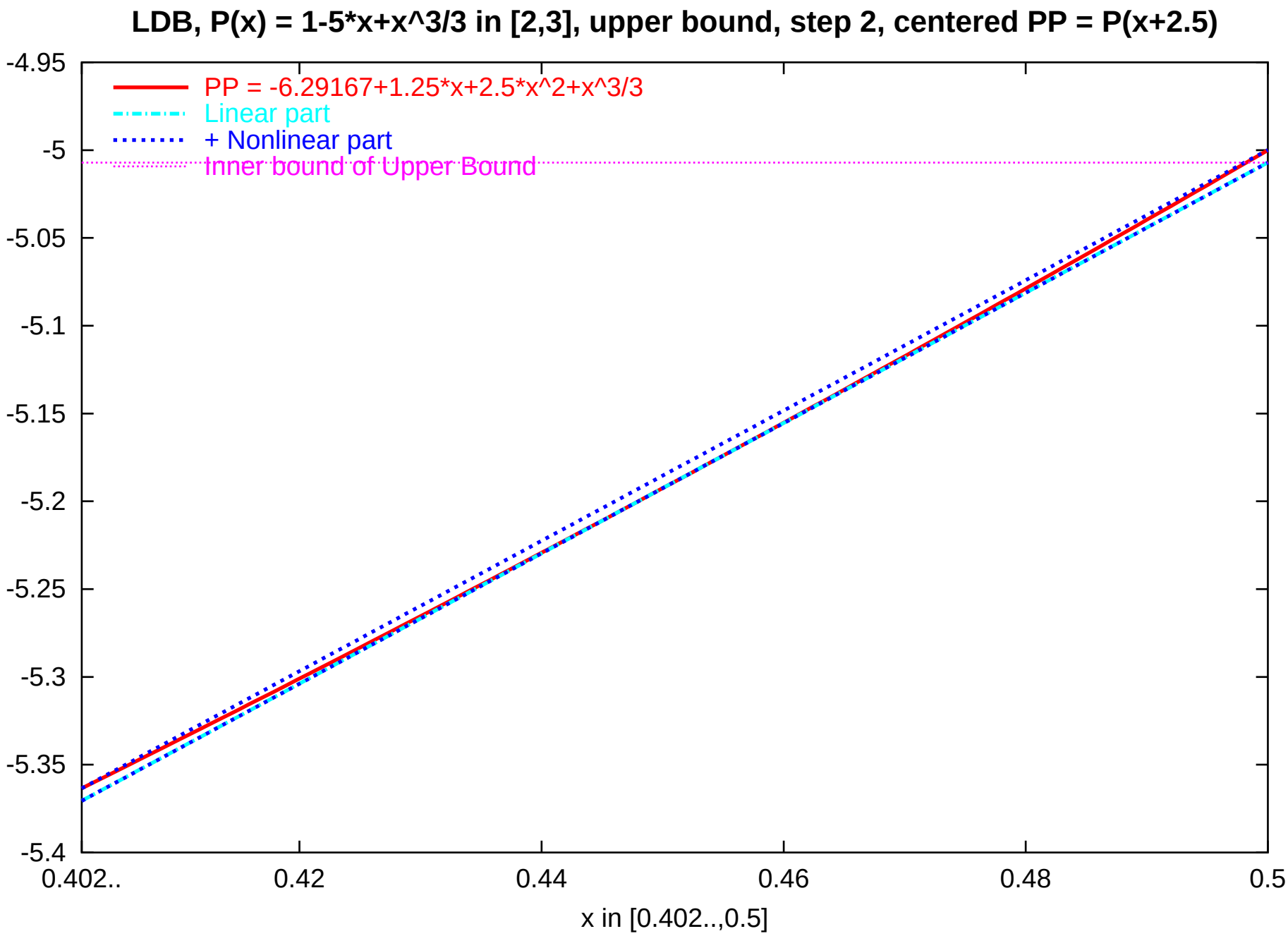
1. Re-expand P at the mid-point m of B to P_m. Center the domain as B_m.
2. Turn the linear coefficients c of P_m all positive by a transformation D, with D_{ii} =sign(c_i), $D_{ij} = 0$ for $i \neq j$.
   The polynomial is P_+ in the domain $B_w = B_m$.
3. Compute the bound of the linear (I_1) and nonlinear (I_h) parts of the polynomial P_+ in B_w. The maximum is bounded by $[M_{in}, M] := \overline{I}_1 + I^h$.
4. The refinement iteration
   (a) If $M - M_{in} > \varepsilon$, set $B_w : \forall i$, if $c_i > 0$ and width(B_{wi})>width(I^h)/c_i, then
   - $\underline{B}_{wi} := \overline{B}_{wi}$−width($I^h$)/$c_i$.
   - Re-expand P_+ at the mid-point of B_w. c_i's are the new coefficients.
   - Go to **3**.

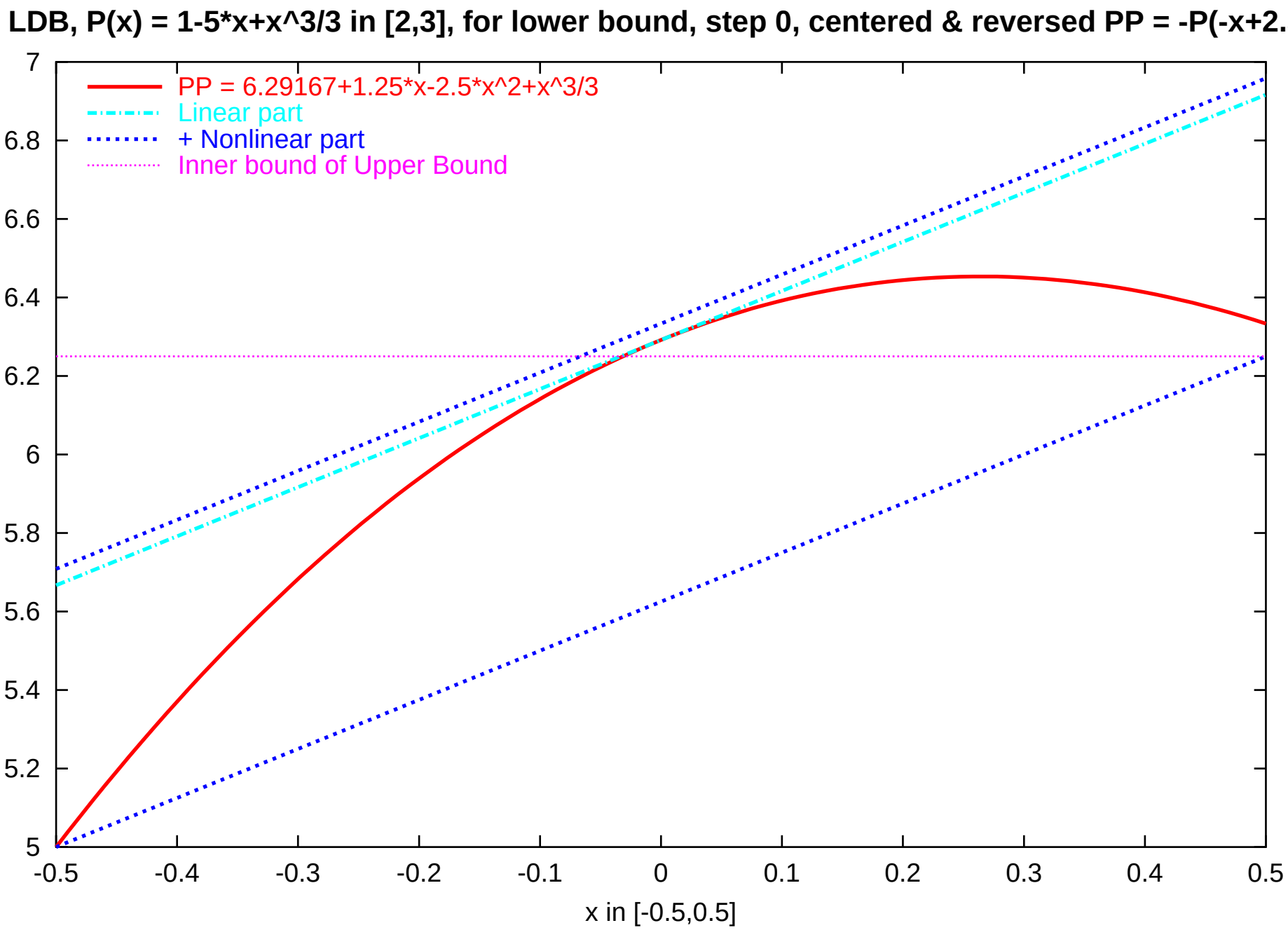
   (b) Else, M is the upper bound of maximum.

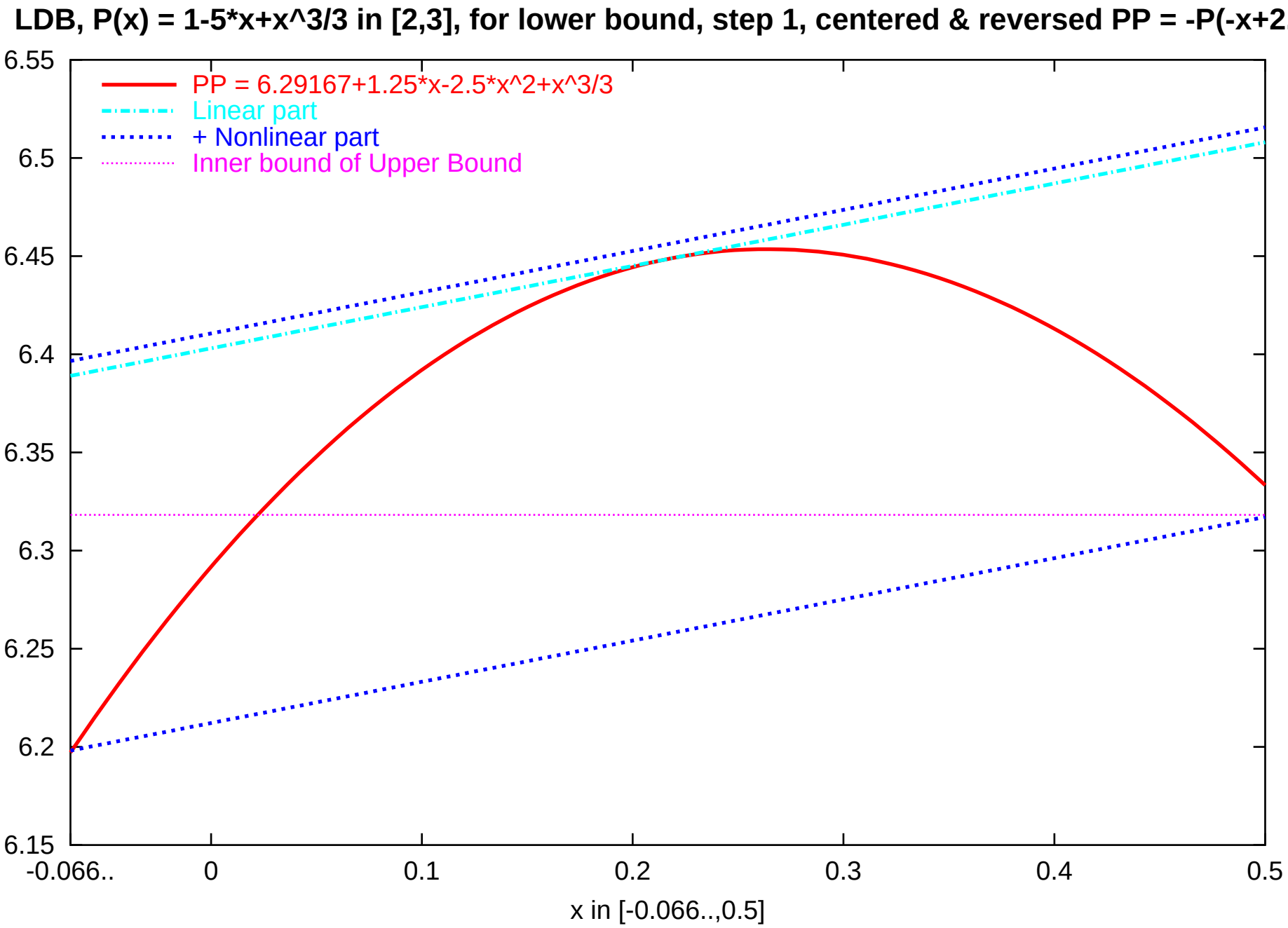

P(x) = 1-5*x+x^3/3 in [2,3],
from R. Moore, SIAM 1979
-5
-5.2
-5.4
-5.6
-5.8
-6
-6.2
-6.4
-6.6
2.0
sqrt(5)
2.5
3.0
x


LDB, P(x) = 1-5*x+x^3/3 in [2,3], upper bound, step 0, centered PP = P(x+2.5)
PP = -6.29167+1.25*x+2.5*x^2+x^3/3
Linear part
+ Nonlinear part
Inner bound of Upper Bound
-5
-5.2
-5.4
-5.6
-5.8
-6
-6.2
-6.4
-6.6
-6.8
-7
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5
x in [-0.5,0.5]


LDB, P(x) = 1-5*x+x^3/3 in [2,3], upper bound, step 1, centered PP = P(x+2.5)
PP = -6.29167+1.25*x+2.5*x^2+x^3/3
Linear part
+ Nonlinear part
Inner bound of Upper Bound
-5
-5.2
-5.4
-5.6
-5.8
-6
-6.2
-6.4
-6.6
-0.066..
0
0.1
0.2
0.3
0.4
0.5
x in [-0.066..,0.5]


LDB, P(x) = 1-5*x+x^3/3 in [2,3], upper bound, step 2, centered PP = P(x+2.5)
PP = -6.29167+1.25*x+2.5*x^2+x^3/3
Linear part
+ Nonlinear part
Inner bound of Upper Bound
-4.95
-5
-5.05
-5.1
-5.15
-5.2
-5.25
-5.3
-5.35
-5.4
0.402..
0.42
0.44
0.46
0.48
0.5
x in [0.402..,0.5]

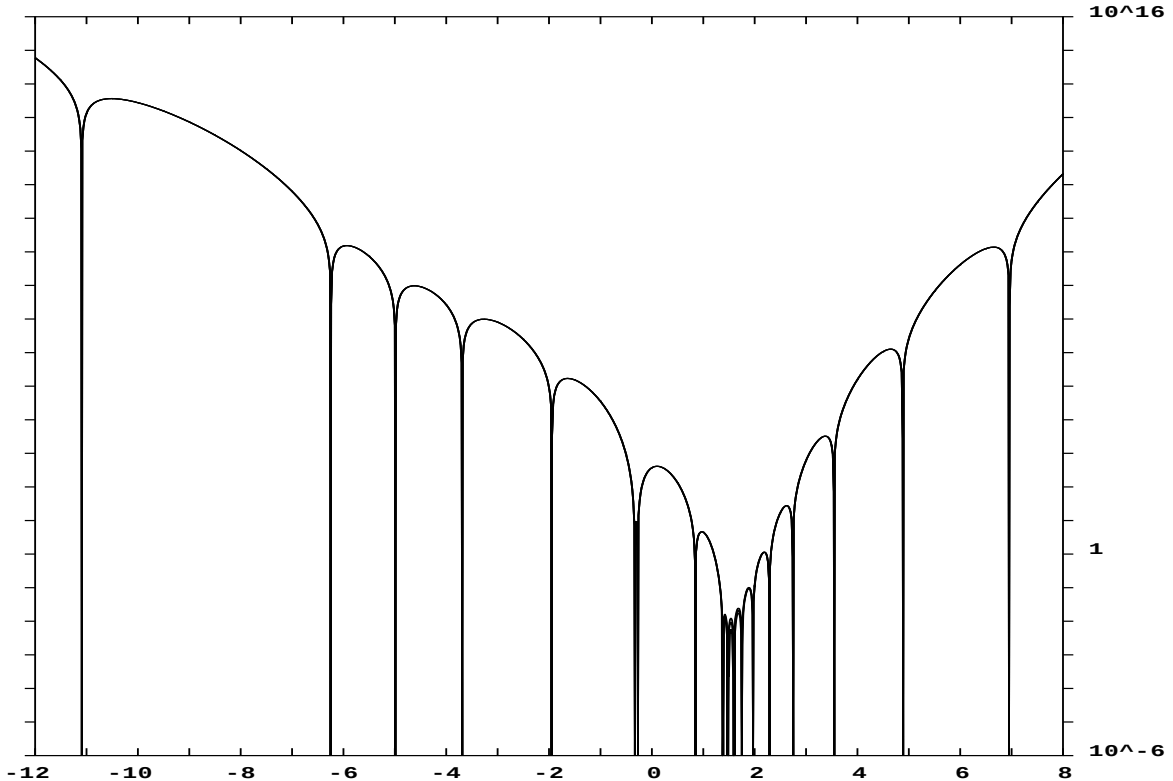

LDB, P(x) = 1-5*x+x^3/3 in [2,3], for lower bound, step 0, centered & reversed PP = -P(-x+2.5)
PP = 6.29167+1.25*x-2.5*x^2+x^3/3
Linear part
+ Nonlinear part
Inner bound of Upper Bound
7
6.8
6.6
6.4
6.2
6
5.8
5.6
5.4
5.2
5
-0.5
-0.4
-0.3
-0.2
-0.1
0
0.1
0.2
0.3
0.4
0.5
x in [-0.5,0.5]

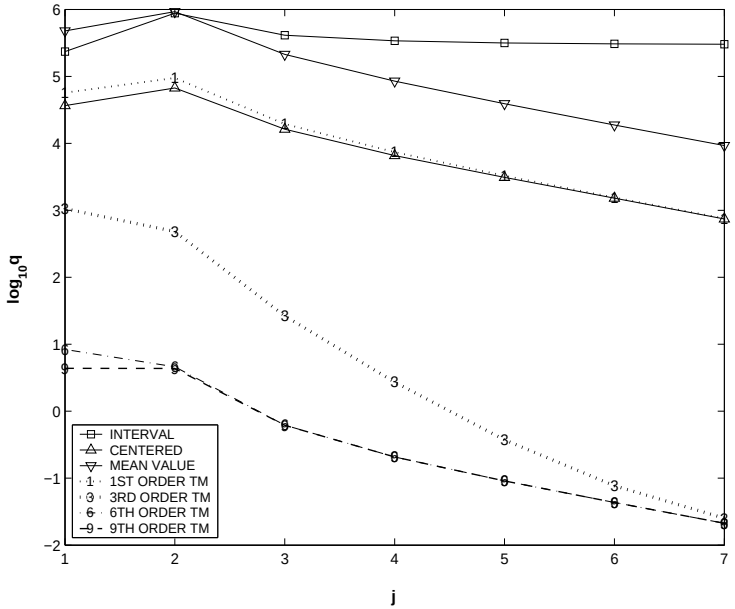

LDB, P(x) = 1-5*x+x^3/3 in [2,3], for lower bound, step 1, centered & reversed PP = -P(-x+2.5)
PP = 6.29167+1.25*x-2.5*x^2+x^3/3
Linear part
+ Nonlinear part
Inner bound of Upper Bound
6.55
6.5
6.45
6.4
6.35
6.3
6.25
6.2
6.15
-0.066..
0
0.1
0.2
0.3
0.4
0.5
x in [-0.066..,0.5]

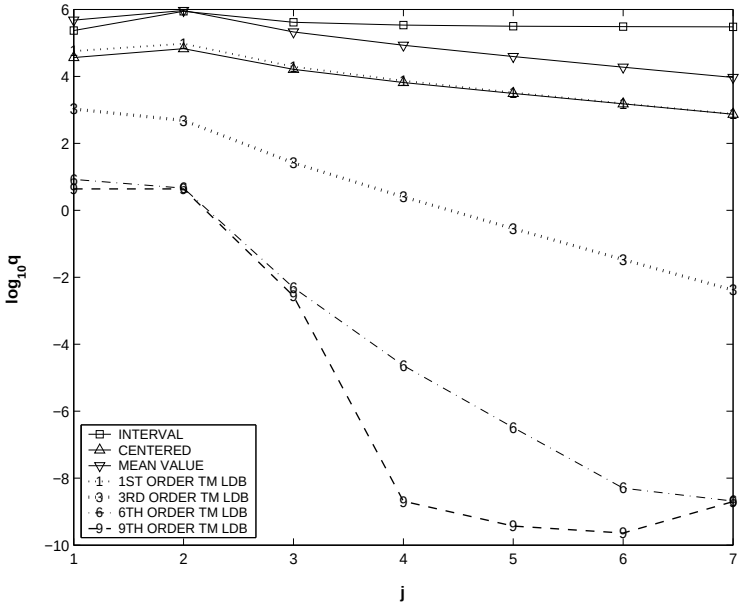
Comparison LDB Bounder and Bernstein Bounder

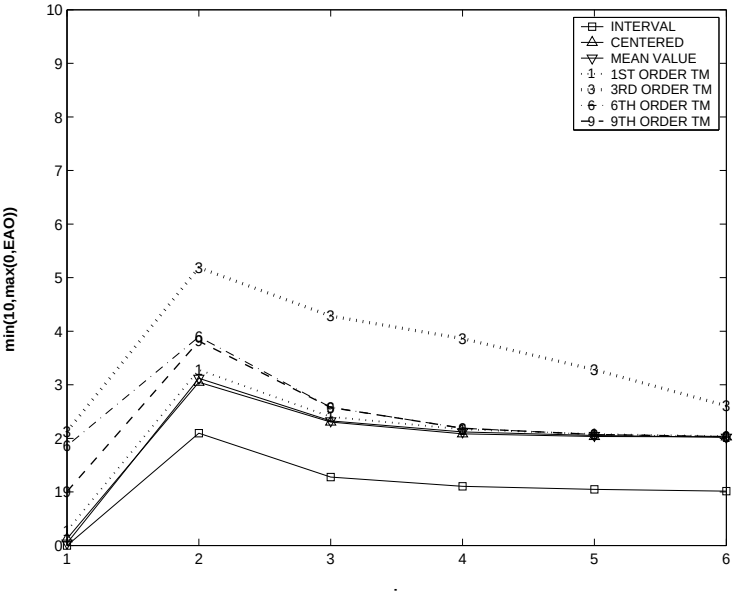
	LDB Bounder	**Bernstein Bounder**
Domain Splitting	yes	yes
Reduction per step	varies, but usually $\gg 1/2$	$1/2$, in one dimension
Convergence Order	$2, ..., (n+1)$	$1, ..., (n+1)$
Ops per step (dense)	$(n+v)!/n!/v!$	$>(n+v)!/n!/v! + k \cdot (n+1)^v$
ops for $n=4$, $v=2$	15	$15 + k \cdot 25$
ops for $n=6$, $v=4$	210	$210 + k \cdot 2401$
ops for $n=8$, $v=6$	3003	$3003 + k \cdot 531,441$

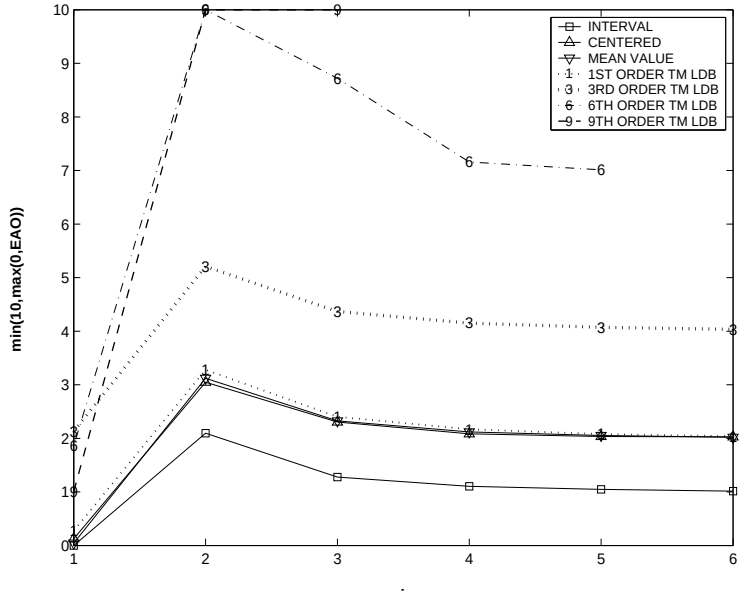
k : number of operations on each matrix element for filling, re-shaping, etc of matrix, and bounding process.


10^16
1
10^-6
-12
-10
-8
-6
-4
-2
0
2
4
6
8


log10 q
j
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM


log10 q
j
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB


min(10,max(0,EAO))
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM
3RD ORDER TM
6TH ORDER TM
9TH ORDER TM


min(10,max(0,EAO))
INTERVAL
CENTERED
MEAN VALUE
1ST ORDER TM LDB
3RD ORDER TM LDB
6TH ORDER TM LDB
9TH ORDER TM LDB

Statistical Analysis of LDB Bounder

Question: How frequently can the LDB bounder succeed to determine the bounds without subdivision?

- Use 1000 random polynomials of order 4
- Vary the dimensionality NV over $1, ..., 10$
- Vary the domain size as $[-2^{-j}, 2^{-j}]$ for $j = 1, ..., 5$
- Record the frequency of success
- Display Mode 1: Freeze j, show results for different NV
- Display Mode 2: Freeze NV, show results for different j

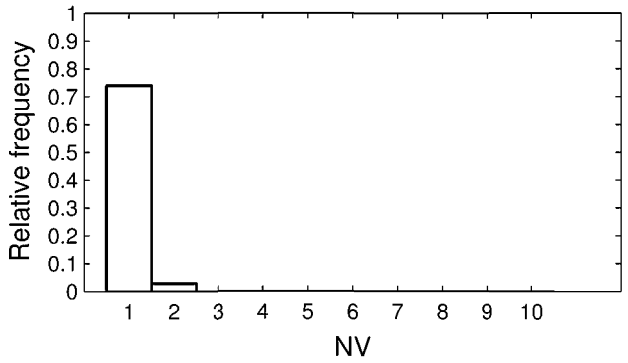
Range bounds of random polynomials
Domain: NV-dimensional interval vector with 0 + [-2^(-j), 2^(-j)] for each component
Tested value(s) of j: 1, 2, ..., 5

DA vectors generated by DARAN DAn n
with the following parameters
n: 1, 2, ..., 1000
NO: 4
NV: 1, 2, ..., 10

Order of Taylor Model approximation: 4
DEFAULT: TM with default tightening
LDB: TM with LDB
k: first value of j for which DEFAULT and LDB
   yield different bounds

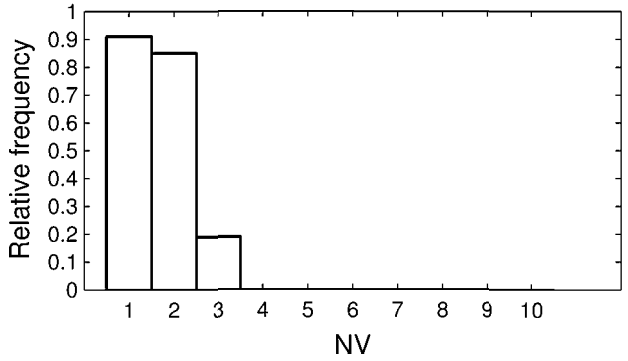
Frequency table for NV (j:1)

NV	frequency	relative frequency
1	740	0.740
2	29	0.029
3	0	0.000
4	0	0.000
5	0	0.000
6	0	0.000
7	0	0.000
8	0	0.000
9	0	0.000
10	0	0.000



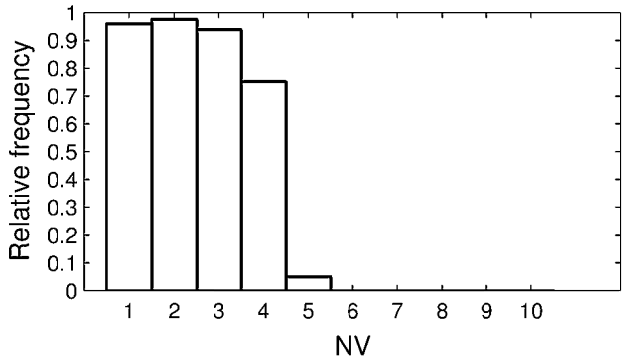
Frequency table for NV (j:2)

NV	frequency	relative frequency
1	910	0.910
2	850	0.850
3	190	0.190
4	0	0.000
5	0	0.000
6	0	0.000
7	0	0.000
8	0	0.000
9	0	0.000
10	0	0.000



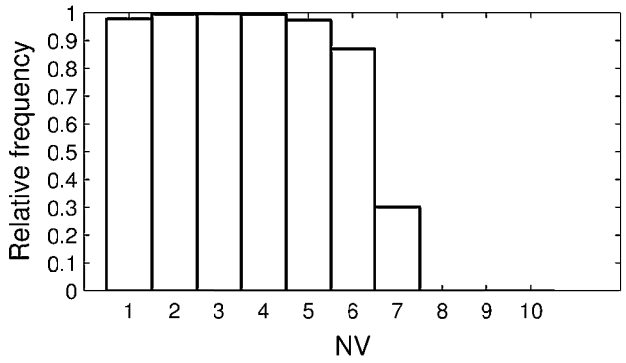
Frequency table for NV (j:3)

NV	frequency	relative frequency
1	960	0.960
2	976	0.976
3	939	0.939
4	752	0.752
5	50	0.050
6	0	0.000
7	0	0.000
8	0	0.000
9	0	0.000
10	0	0.000



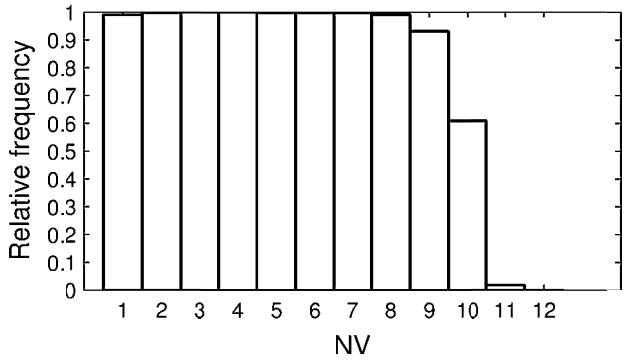
Frequency table for NV (j:4)

NV	frequency	relative frequency
1	977	0.977
2	995	0.995
3	996	0.996
4	993	0.993
5	973	0.973
6	870	0.870
7	301	0.301
8	0	0.000
9	0	0.000
10	0	0.000



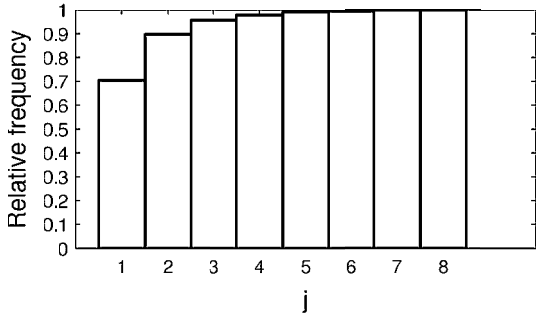
Frequency table for NV (j:5)

NV	frequency	relative frequency
1	990	0.990
2	999	0.999
3	1000	1.000
4	1000	1.000
5	999	0.999
6	999	0.999
7	997	0.997
8	991	0.991
9	931	0.931
10	609	0.609
11	18	0.018
12	0	0.000



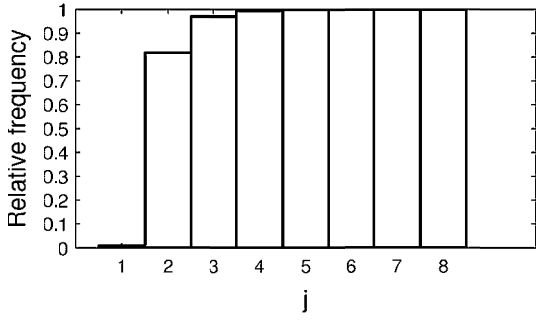
Frequency table for j (NV:1)

j	frequency	relative frequency
1	705	0.705
2	897	0.897
3	957	0.957
4	978	0.978
5	991	0.991
6	995	0.995
7	998	0.998
8	999	0.999



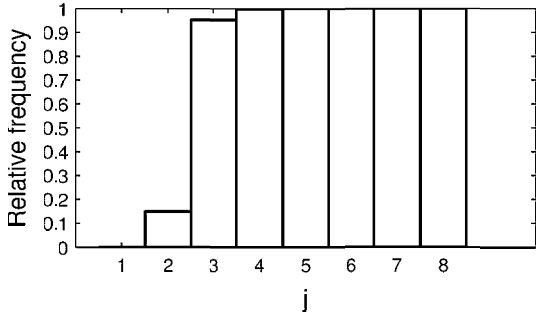
Frequency table for j (NV:2)

j	frequency	relative frequency
1	9	0.009
2	818	0.818
3	970	0.970
4	995	0.995
5	1000	1.000
6	1000	1.000
7	1000	1.000
8	1000	1.000



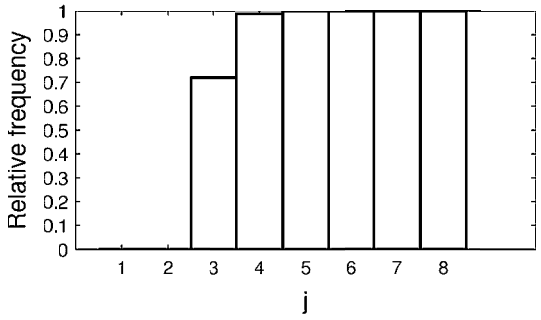
Frequency table for j (NV:3)

j	frequency	relative frequency
1	0	0.000
2	149	0.149
3	953	0.953
4	998	0.998
5	1000	1.000
6	1000	1.000
7	1000	1.000
8	1000	1.000



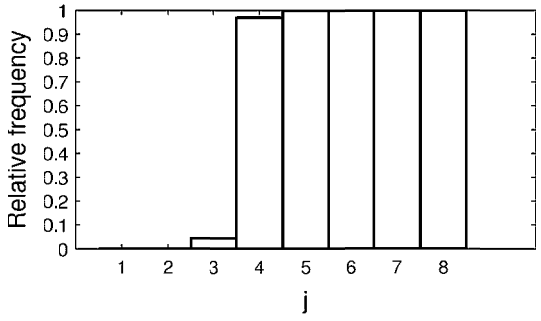
Frequency table for j (NV:4)

j	frequency	relative frequency
1	0	0.000
2	0	0.000
3	720	0.720
4	987	0.987
5	1000	1.000
6	1000	1.000
7	1000	1.000
8	1000	1.000



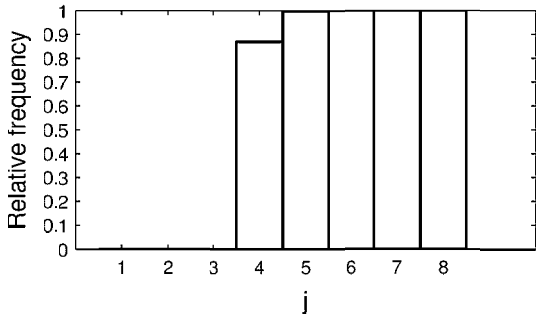
Frequency table for j (NV:5)

j	frequency	relative frequency
1	0	0.000
2	0	0.000
3	44	0.044
4	969	0.969
5	999	0.999
6	1000	1.000
7	1000	1.000
8	1000	1.000



Frequency table for j (NV:6)

j	frequency	relative frequency
1	0	0.000
2	0	0.000
3	0	0.000
4	870	0.870
5	999	0.999
6	1000	1.000
7	1000	1.000
8	1000	1.000



The QDB-LDB Bounder I

Observe that bounding polynomial of a TM up to order 2 exactly yields a third order bounder. Let this polynomial be

$$P(x) = P_o + P_1(x) + P_2(x) = P_0 + cx + \frac{1}{2}x^t H x$$

We use a **peeling strategy** enhanced by LDB bounder and positive definiteness check.

Wlog assume we are interested in minimum, the domain is $[-1, 1]^v$, and all variables do appear in the second order part (if not, reduce dimensionality). Initialize stack to be studied by all 3^v hypersurfaces consisting of $[-1, 1]^v$, all its boundaries, all their boundaries, etc etc. and all corner points. (These are characterized by a vector of length v where each component assumes the values $-1, 0$, or $+1$; the number of 0s is the dimension of the hypersurface)

Evaluate P at the corner point that lies closest to the direction $-\vec{c}$ of the gradient at the center, call result M. Initialize stack of boxes to be studied to $[-1, 1]^v$.

Example for "-1,0,+1" Notation

Consider a 3D cube. There are 27 surfaces

0	0	0	3D center volume
-1	0	0	2D left face
+1	0	0	2D right face
0	-1	0	2D front face
0	+1	0	2D back face
0	0	-1	2D bottom face
0	0	1	2D top face
-1	-1	0	1D edge left back edge
			... ...
+1	+1	+1	0D top right back corner

The QDB-LDB Bounder II

1. If there is no box on the stack, we are done. Otherwise, pick a box B of maximum remaining dimension from the stack. Let P on B be $P = c_0 + cx + \frac{1}{2}x^t\ Hx$.
2. Evaluate P on B with LDB; call the lower bound M_B. If $M_B \geq M$, discard B and all its boundaries and their boundaries ... and go to 1. The discarding is greatly simplified by "$-1, 0, +1$" vector notation.
3. If box B is not discarded, perform a non-positive definiteness test of P_2 on B. If non-positive definiteness is shown, discard B and go to 1. P_2 is shown to not be positive definite if any one of the following holds

   * Any diagonal element H_{ii} is negative or zero

   * $\sum_{i,j} P_{ij} \leq 0$

   * Any 2×2 principal minor m_{ij} is negative or zero

   * (Any higher principal minor is negative or zero; may be too much to check)

The QDB-LDB Bounder III

4. If P_2 survives this check, try to solve $Hx + c = 0$. Use Moore-Krawcyk algorithm with preconditioning.

   a) If unique solution can be found, check

      - If $x \notin B$, discard box B and go to 1.
      - If $x \in B$, set $M = \min(M, P(x))$.

        Discard B and all its boundaries and their boundaries ...

        Go to 1.

   b) If unique solution can not be found, H has an eigenvalue close to 0.

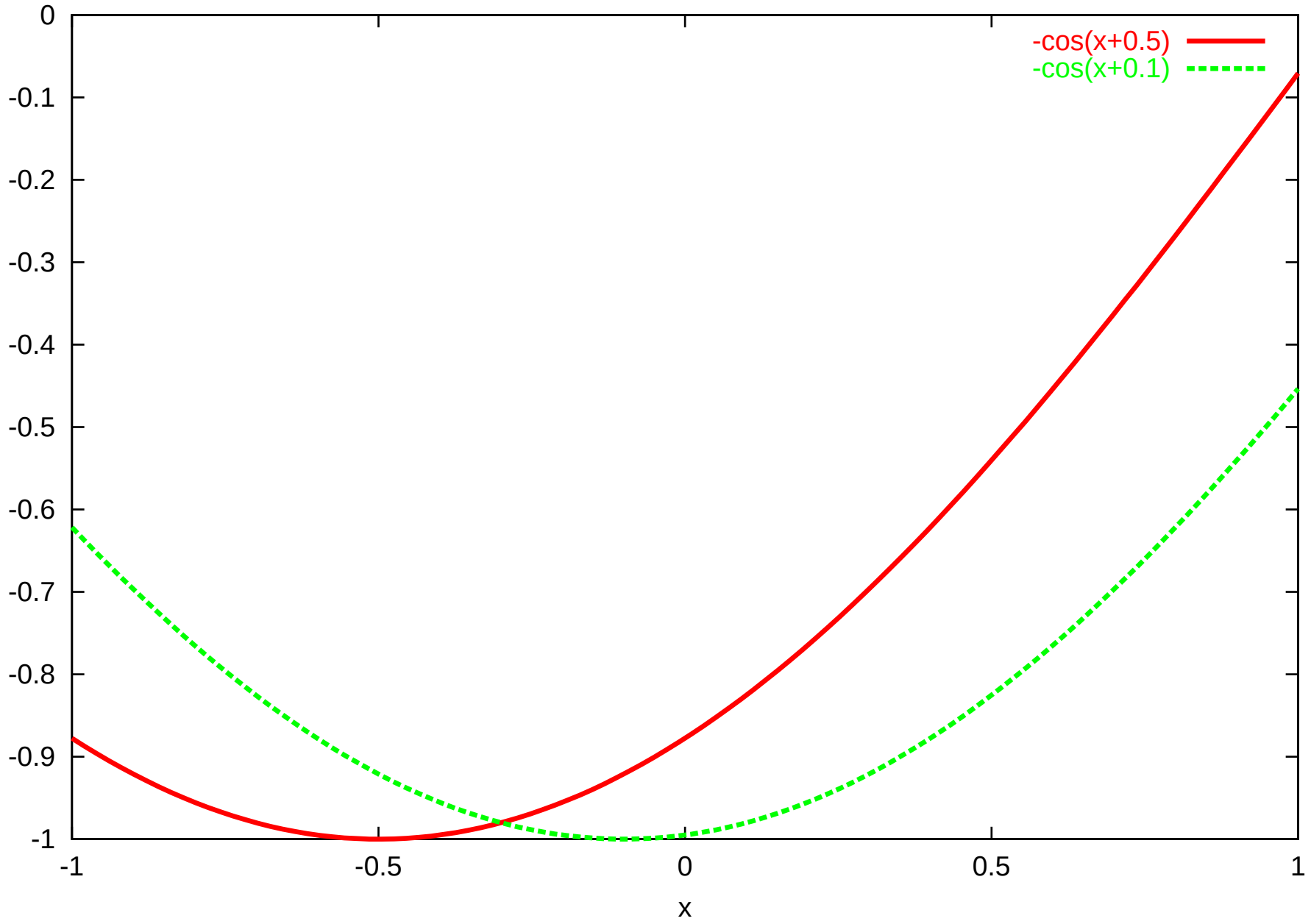
      Modify $H \rightarrow H + \varepsilon I$, where I is the identity,

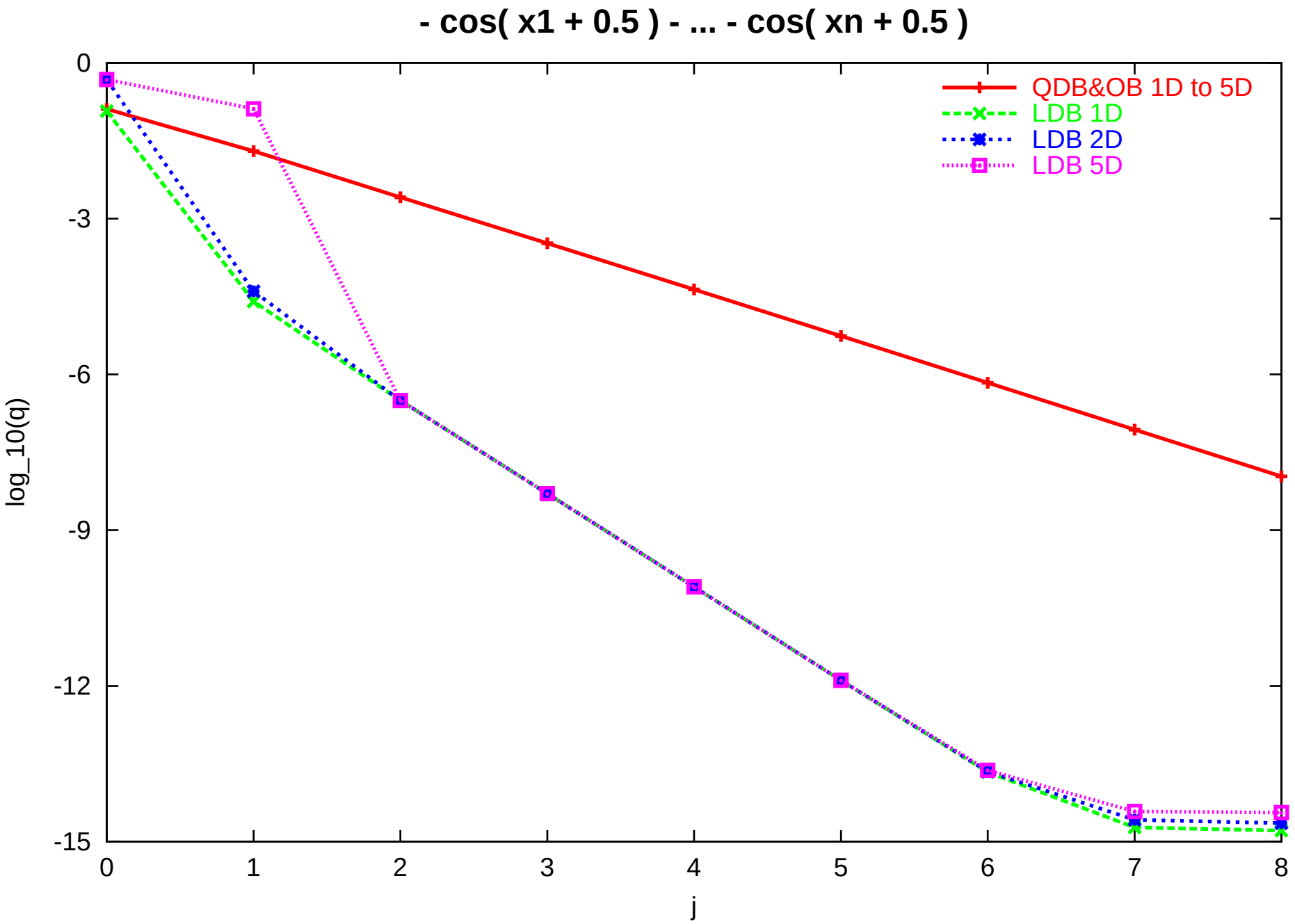
      until unique solution can be found, go to a).

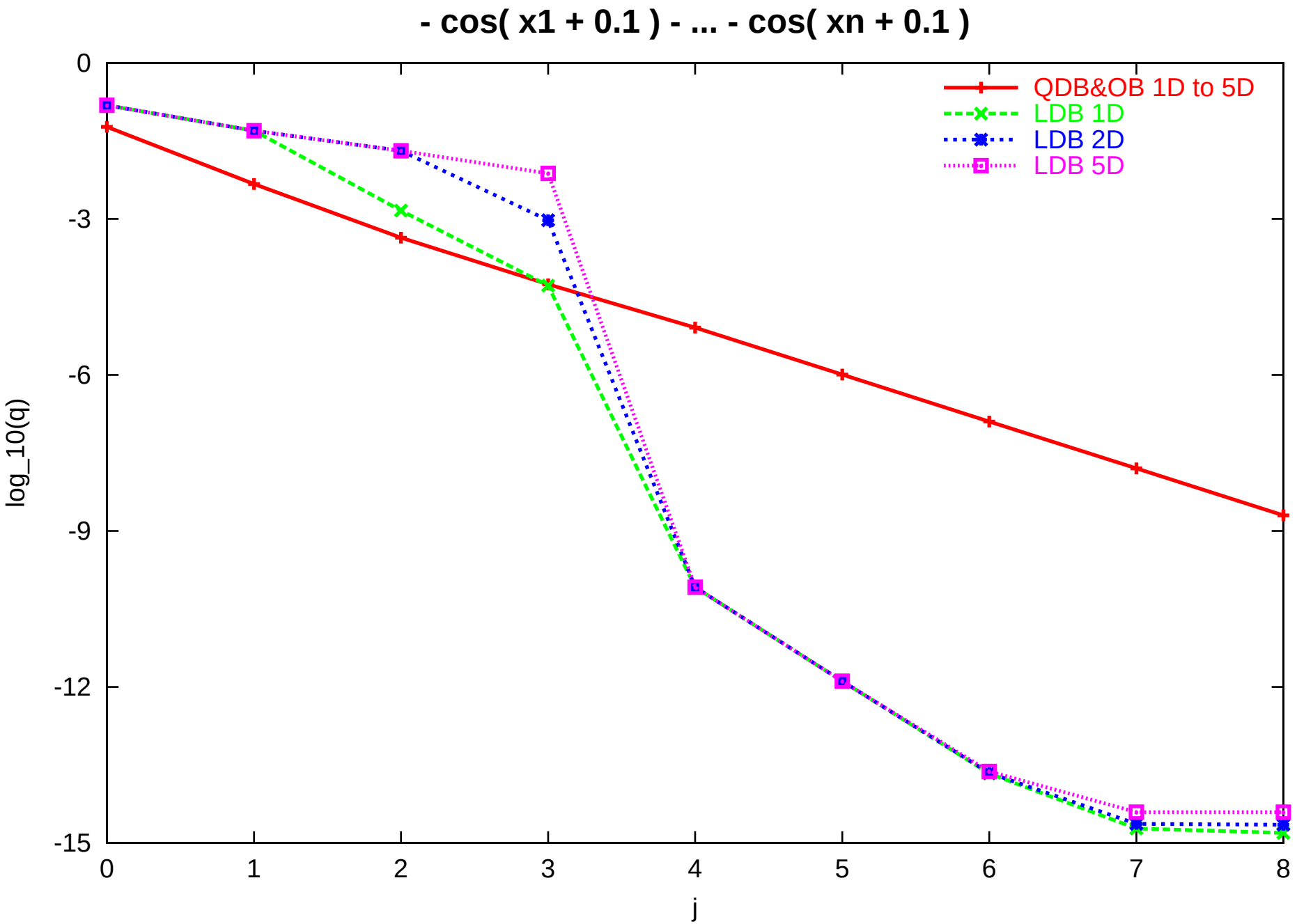
The QDB-LDB Bounder - Properties

A few observations about the practical performance are in order

- A successful use of LDB on the first box immediately terminates the algorithm. This is very likely to occur if the function is monotonic, and in particular if the nonlinearities are small, as is the case for results of most TM computations.
- A successful use of LDB on only one of the primary boundaries decreases the workload by about 1/3 in typical cases. This is likely to occur once or twice if the function is monotonic in one direction.
- A successful use of LDB on one of the remaining lower boundaries has less impact, but is still very useful.
- Since LDB is relatively cheap, the combination with LDB can have a big influence on practical performance
- Algorithm requires as a worst case $3^v - 2^v$ solutions of linear systems of various dimensions; the 2^v corner points do not require linear algebra.


- cos(x + 0.5) and - cos(x + 0.1)
-cos(x+0.5)
-cos(x+0.1)
0
-0.1
-0.2
-0.3
-0.4
-0.5
-0.6
-0.7
-0.8
-0.9
-1
-1
-0.5
0
0.5
1
x


- cos(x1 + 0.5) - ... - cos(xn + 0.5)
QDB&OB 1D to 5D
LDB 1D
LDB 2D
LDB 5D
log_10(q)
j
0
-3
-6
-9
-12
-15
0
1
2
3
4
5
6
7
8


- cos(x1 + 0.1) - ... - cos(xn + 0.1)
QDB&OB 1D to 5D
LDB 1D
LDB 2D
LDB 5D
log_10(q)
j
0
-3
-6
-9
-12
-15
0
1
2
3
4
5
6
7
8

Example: Polynomial from Moore's 1979 Book

$$P(x) = 1 - 5x + \tfrac{1}{3}x^3, \quad x \in [2, 3]$$

Method	Range	Width
Exact	[−6.453559.. , −5]	1.46
Naive Interval	[−11.333333.. , 0]	11.34
Centered Form	[−7.583333.. , −5]	2.59
by Order Bounds	[−6.958333.. , −5]	1.96
by QDB+hi OB	[−6.489583.. , −5]	1.49
LDB	[−6.516634.. , −5]	1.52
Mean Value Form	[−8.291.. , −4.291..]	4.01
Monotonicity	local minimum at $\sqrt{5}$	
POLBND	[−6.453560.. , −5]	1.46
Rastering, 1000pnt (inner)	[−6.453559.. , −5]	1.46

POLBND is a **global optimizer** implemented in COSY by Jens Hoefkens (1999) based on the algorithm in the dissertation of Ratz. Combined with nth order TM, this is our first $(n+1)$st order bounder.

Gritton's Second Problem, Domain 1.4+[-1,1]·2^{-5}

The polynomial of degree 18 from Chemical Engineering. There are 18 roots in the range $[-12, 8]$, and it is particularly difficulty to bound around 1.4.

Method	Width	Accur.
Rastering (1000)	2.69e-2	
Naive Interval	1.04e4	1.04e4
Centered Form	1.25e3	1.25e3
Mean V. Form	1.43e3	1.43e3
LDB	2.86e-2	1.74e-3
Global Opt.	3.20e-2	5.10e-3

4th TM ε	3.90e-4	
+ QDB + hi OB	2.90e-2	2.14e-3
+ LDB	2.90e-2	2.14e-3
+ GO	2.73e-2	3.99e-4
8th TM ε	4.89e-10	
+ QDB + hi OB	2.86e-2	1.74e-3
+ LDB	2.86e-2	1.74e-3
+ GO	2.69e-2	9.58e-8
Monotonicity	No	

Conclusion

- Taylor models provide enclosures of functional dependencies by polynomials and a remainder that scales with $n+1$st order.
- Range bounding of polynomial is often easier than range bounding of the original function. Thus, the TM range bounding algorithms can lead to a high order method.
- For one dimensional systems, there are bounders up to the sixth order.
- The LDB bounding is cheap, and can be used to assist various other methods. (QDB, Bernstein)
- The QDB bounder provides the third order method. By combining with LDB, the QDB bounder can be quite efficient.