High-Order Method for Rigorous Lower Bounds of Smooth Functions near Minimizers

Youn-Kyoung Kim, Kyoko Makino

Department of Physics and Astronomy
Michigan State University
A Simple 1D Example

Approximate the cos function by its power series to order 60:

\[ f(x) = \sum_{i=0}^{30} (-1)^i \frac{x^{2i}}{(2i)!}. \]

Several nice properties:

1. Properties of the function are well known
2. Dependency increases with \( x \) from very small to very large
3. Periodicity allows the study of the same functional behavior with varying amounts of dependency
4. Study at points with both non-stationary and stationary points is possible

Study results for expansion points \( x_0 = n \cdot \pi/4 \) for

\[ n = 1, 5, 9, 13 \text{ and } n = 0, 4, 8, 12. \]

For each of these points, domains are \( x_0 + [-2^{-j}, 2^{-j}] \) for \( j = 1, \ldots, 8. \)
<table>
<thead>
<tr>
<th>log₁₀ q</th>
<th>INTERVAL CENTERED MEAN VALUE</th>
<th>1ST ORDER TM</th>
<th>3RD ORDER TM</th>
<th>6TH ORDER TM</th>
<th>9TH ORDER TM</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

**Method:** AEAO

<table>
<thead>
<tr>
<th>log₁₀ q</th>
<th>INTERVAL CENTERED MEAN VALUE</th>
<th>1ST ORDER TM</th>
<th>3RD ORDER TM</th>
<th>6TH ORDER TM</th>
<th>9TH ORDER TM</th>
</tr>
</thead>
<tbody>
<tr>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>-4</td>
<td>-3</td>
<td>-2</td>
<td>-1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>
INTERVAL CENTERED MEAN VALUE 1ST ORDER TM 3RD ORDER TM 6TH ORDER TM 9TH ORDER TM

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

INTERVAL CENTERED MEAN VALUE TM 1 TM 3 TM 6 TM 9

INTERVAL CENTERED MEAN VALUE TM LDB 1 TM LDB 3 TM LDB 6 TM LDB 9

METHOD
Implementation of TM Arithmetic

Validated Implementation of TM Arithmetic exists. The following points are important

- Strict requirements for **underlying FP arithmetic**
- Taylor models require cutoff threshold (**garbage collection**)
- Coefficients remain FP, not intervals
- Package quite **extensively tested** by Corliss et al.

For practical considerations, the following is important:

- Need **sparsity** support
- Need efficient coefficient **addressing** scheme
- About 50,000 lines of code
- **Language Independent** Platform, coexistence in F77, C, F90, C++
Ordered LDL (Extended Cholesky) Decomposition

Given Quadratic Form with symmetric $H$

$$Q(x) = \frac{1}{2} x^t \cdot H \cdot x + a \cdot x + b$$

We determine Ordered LDL Decomposition (L: lower diagonal with unit diagonal, D: diagonal) as follows

1. Pre-sort rows and columns by the size of their diagonal elements
2. Successively execute conventional $LtDL$ decomposition step in interval arithmetic, beginning by representing every element of $H$ by a thin interval; in step $i$:
   
   (a) If $\underline{D}(i, i) > 0$ proceed to the next row and column.
   
   (b) If $\bar{D}(i, i) < 0$ exchange row and column $i$ with row and column $i + 1$, $i + 2$, ... If a positive element is found, increment $i$ and repeat. If none is found, stop.

**Note:** Correction Matrix In case $0 \in D(i, i)$, apply small correction $C$ to $H$, i.e. study $H + C$ instead of $H$, such that all elements of $D$ are clearly positive or negative. $|C|$ is lumped into the remainder bound of the original problem.
Ordered LDL Decomposition - Result

Have obtained representation of $H$ as LDL composition

$$P^t HP = L^t DL$$

- First $p$ elements of $D$ satisfy $l(D(i, i)) > 0$
- Remaining $(n - p)$ elements of $D$ will satisfy $u(D(i, i)) < 0$

**Proposition:** Sufficiently near a local minimizer, $D$ will contain only positive elements. Furthermore, in the wider vicinity of the local minimizer, the number of negative elements in $D$ will decrease as the minimizer is approached.

Simply follows from continuity of the matrix $D$ as a function of position
The QDB (Quadratic Dominated Bounder) Algorithm

1. Let $u$ be an external cutoff. Initialize $u = \min(u, Q(C'))$. Initialize list with all $3^n$ surfaces for study.
2. If no boxes are remaining, terminate. Otherwise select one surface $S$ of highest dimension.
3. On $S$, apply LDB. If a complete rejection is possible, strike $S$ from the list and proceed to step 2. If a partial rejection is possible, strike the respective surfaces of $S$ from the list and proceed to step 2.
4. Determine the definiteness of the Hessian of $Q$ when restricted to $S$
5. If the Hessian is not p.d. strike $S$ from the list and proceed to step 2.
6. If the Hessian is p.d., determine the corresponding critical point $c$.
7. If $c$ is fully inside $S$, strike $S$ and all surfaces of $S$ from the list, update $u = \min(u, Q(c))$, and proceed to step 2
8. If $c$ not inside $S$, strike $S$. If certain components of $c$ lie between $-1$ and $+1$, strike the corresponding surfaces and proceed to step 2
The QDB Algorithm - Properties

The QDB algorithm has the following properties.

1. The quadratic bounder QDB has the third order approximation property.
2. The effort of finding the minimum requires the study of at most $3^n$ surfaces.
3. In the p.d. case, the computational effort requires at most the study of $2^n$ surfaces.
4. Because of extensive box striking, in practice, the numbers of boxes to study is usually much much less.
The QDB Algorithm - Properties

The QDB algorithm has the following properties.

1. The quadratic bounder QDB has the third order approximation property.
2. The effort of finding the minimum requires the study of at most $3^n$ surfaces.
3. In the p.d. case, the computational effort requires at most the study of $2^n$ surfaces.
4. Because of extensive box striking, in practice, the numbers of boxes to study is usually much much much less.

But still, it is desirable to have something FASTER.
The QFB (Quadratic Fast Bounder) Algorithm

Let $P + I$ be a given Taylor model. Idea. Decompose into two parts

$$P + I = (P - Q) + I + Q$$

and observe

$$l(P + I) = l(P - Q) + l(Q) + l(I)$$

Choose $Q$ such that

1. $Q$ can be easily bounded from below
2. $P - Q$ is sufficiently simplified to allow bounding above given cutoff.

First possibility: Let $H$ be p.d. part of $P$, set

$$Q = x^t H x$$

Then $l(Q) = 0$. Removes all second order parts of $P$ (!) Better yet:

$$Q_{x_0} = (x - x_0)^t H (x - x_0)$$

Allows to manipulate linear part. Works for ANY $x_0$ in domain. Still $l(Q_{x_0}) = 0$.

Which choices for $x_0$ are good?
The QFB Algorithm - Properties

Most critical case: near local minimizer, so $H$ is the entire purely quadratic part of $P$.

**Theorem:** If $x_0$ is the (unique) minimizer of quadratic part of $P$ on the domain of $P + I$, then the lower bound of the linear part of $(P - Qx_0)$ is zero. Furthermore, the lower bound of $(P - Qx_0)$, when evaluated with plain interval evaluation, is accurate to order 3 of the original domain box.

**Proof:** Follows readily from Kuhn-Tucker conditions. If $x_0$ inside, linear part vanishes completely. Otherwise, wlog if $i$-th component of $x_0$ is at left end, $i$-th partial there must be non-negative, so that we get non-negative contribution.

**Remark:** The closer $x_0$ is to the minimizer, the closer we are to order 3 cutoff.

**Algorithm: (Third Order Cutoff Test).** Let $x^{(n)}$ be a sequence of points that converges to the minimum $x_0$ of the convex quadratic part $P_2$ In step $n$, determine a bound of $(P - Qx_n)$ by interval evaluation, and assess whether the bound exceeds the cutoff threshold. If it does, reject the box and terminate; if it does not, proceed to the next point $x_{n+1}$. 
The QMLoc Algorithm

Tool to generate efficient sequence $x^{(n)}$. Determine ”feasible descent direction”

$$g_i^{(n)} = \begin{cases} 
-\frac{\partial Q}{\partial x_i} & \text{if } x_i^{(n)} \text{ inside} \\
\min \left( -\frac{\partial Q}{\partial x_i}, 0 \right) & \text{if } x_i^{(n)} \text{ on right} \\
\max \left( -\frac{\partial Q}{\partial x_i}, 0 \right) & \text{if } x_i^{(n)} \text{ on left}
\end{cases}$$

Now move in direction of $g^{(n)}$ until we hit box or quadratic minimum along line. Very fast to do, can change set of active constraints very quickly.

**Result:** Cheap iterative third order cutoff.
Use of QFB - Example

Let $f_1(x) = \frac{1}{2}x^t \cdot A_v \cdot x - A_v \cdot (a \cdot x) + \frac{1}{2}a^t \cdot A_v \cdot a$ with

$$A_v = \begin{pmatrix} 2 & 3 & \ldots & 3 \\ -1 & 2 & \ldots & 3 \\ \vdots & \vdots & \ddots & \vdots \\ -1 & -1 & \ldots & 2 \end{pmatrix}$$

known to be p.d. with minimum $a$. Choose a random vector $a$, and 5$^v$ boxes around it. Check box rejection with Interval evaluation, Centered Form, QFB. Output average number of QFB iterations.
Use of QFB - Example

Let \( f_1(x) = \frac{1}{2} x^t \cdot A_v \cdot x - A_v \cdot (a \cdot x) + \frac{1}{2} a^t \cdot A_v \cdot a \) with

\[
A_v = \begin{pmatrix}
2 & 3 & \ldots & 3 \\
-1 & 2 & \ldots & 3 \\
\vdots & \vdots & \ddots & \vdots \\
-1 & -1 & \ldots & 2
\end{pmatrix}
\]

known to be p.d. with minimum \( a \). Choose a random vector \( a \), and \( 5^v \) boxes around it. Check box rejection with Interval evaluation, Centered Form, QFB. Output average number of QFB iterations.

<table>
<thead>
<tr>
<th>( v )</th>
<th>( N=5^v )</th>
<th>( NI )</th>
<th>( NC )</th>
<th>( NQFB )</th>
<th>Avg. Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>25</td>
<td>25</td>
<td>8</td>
<td>1</td>
<td>1.1</td>
</tr>
<tr>
<td>4</td>
<td>625</td>
<td>625</td>
<td>308</td>
<td>1</td>
<td>0.31</td>
</tr>
</tbody>
</table>
Use of QFB - Example

Let \( f_1(x) = \frac{1}{2}x^t \cdot A_v \cdot x - A_v \cdot (a \cdot x) + \frac{1}{2}a^t \cdot A_v \cdot a \) with

\[
A_v = \begin{pmatrix}
2 & 3 & \cdots & 3 \\
-1 & 2 & \cdots & 3 \\
\vdots & \vdots & \ddots & \vdots \\
-1 & -1 & \cdots & 2
\end{pmatrix}
\]

known to be p.d. with minimum \( a \). Choose a random vector \( a \), and \( 5^v \) boxes around it. Check box rejection with Interval evaluation, Centered Form, QFB. Output average number of QFB iterations.

<table>
<thead>
<tr>
<th>( v )</th>
<th>( N=5^v )</th>
<th>( NI )</th>
<th>( NC )</th>
<th>NQFB</th>
<th>Avg. Iter</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>25</td>
<td>25</td>
<td>8</td>
<td>1</td>
<td>1.1</td>
</tr>
<tr>
<td>4</td>
<td>625</td>
<td>625</td>
<td>308</td>
<td>1</td>
<td>0.31</td>
</tr>
<tr>
<td>6</td>
<td>15,625</td>
<td>15,625</td>
<td>12,434</td>
<td>1</td>
<td>0.31</td>
</tr>
<tr>
<td>8</td>
<td>390,625</td>
<td>390,625</td>
<td>372,376</td>
<td>1</td>
<td>0.43</td>
</tr>
<tr>
<td>10</td>
<td>9,765,625</td>
<td>9,765,625</td>
<td>9,622,750</td>
<td>1</td>
<td>0.55</td>
</tr>
</tbody>
</table>
Moore’s Simple 1D Function

\[ f(x) = 1 + x^5 - x^4. \]

Study on [0, 1]. Trivial-looking, but dependency and high order. Assumes shallow min at 0.8.
COSY-GO with LDB-QFB (TM 5th). 1D. \( f = x^5 - x^4 + 1 \)

- Studied box
- Box size reduction by LDB
COSY-GO with naive IN with mid point test. 1D. $f = x^5 - x^4 + 1$
COSY-GO with IN. 1D. $f=x^5-x^4+1$. -- Up to the 160th box
COSY-GO with Centered Form with mid point test. 1D. \( f = x^5 - x^4 + 1 \)
Beale’s 2D and 4D Function

\[ f(x_1, x_2) = (1.5 - x_1(1 - x_2))^2 + (2.25 - x_1(1 - x_2^2))^2 + (2.625 - x_1(1 - x_2^3))^2 \]

Domain \([-4.5, 4.5]^2\). Minimum value 0 at \((3, 0.5)\).

Little dependency, but tricky very shallow behavior.

Generalization to 4D:

\[ f(x_1, x_2, x_3, x_4) = (1.5 - x_1(1 - x_2))^2 + (2.25 - x_1(1 - x_2^2))^2 + (2.625 - x_1(1 - x_2^3))^2 \]
\[ + (1 + x_3(1 - x_4))^2 + (3 + x_3(1 - x_4^2))^2 + (7 + x_3(1 - x_4^3))^2 \]
\[ + (3 + x_1(1 - x_4))^2 + (9 + x_1(1 - x_4^2))^2 + (21 + x_1(1 - x_4^3))^2 \]
\[ + (0.5 - x_3(1 - x_2))^2 + (0.75 - x_3(1 - x_2^2))^2 + (0.875 - x_3(1 - x_2^3))^2 \]

Domain \([0, 4]^4\). Minimum value 0 at \((3, 0.5, 1, 2)\)
The Beale function. 
\[ f = [1.5-x(1-y)]^2 + [2.25-x(1-y^2)]^2 + [2.625-x(1-y^3)]^2 \]
COSY-GO with IN. The Beale function
COSY-GO with CF. The Beale function
COSY-GO with LDB/QFB. The Beale function
COSY-GO. The Beale function. Remaining Boxes ( < 1e-6 ) around (3,0.5)
COSY-GO The Beale Function: Number of Boxes -- CF

To Be Studied
Small Boxes ( < 1e-6 )
COSY-GO The Beale Function: Number of Boxes -- LDB/QFB

To Be Studied
Small Boxes (< 1e-6)
COSY-GO Beale 4D: Number of Boxes -- IN

Number

Step (Number of Boxes Studied)

To Be Studied

Small Boxes (< 1e-6)
COSY-GO Beale 4D: Number of Boxes -- CF

To Be Studied
Small Boxes (< 1e-6)

Number

Step (Number of Boxes Studied)
COSY-GO Beale 4D: Number of Boxes -- LDB/QFB

To Be Studied
Small Boxes ( < 1e-6 )
Lennard-Jones Potentials

Ensemble of \( n \) particles interacting pointwise with potentials

\[
V_{LJ}(r) = \frac{1}{r^{12}} - 2 \cdot \frac{1}{r^6}
\]

Has very shallow minimum of \(-1\) at \( r = 0 \). Very hard to Taylor expand. Extremely wide range of function values: \( V_{LJ}(0.5) \approx 4000, V_{LJ}(2) \approx 0.03 \)

\[
V = \sum_{i<j}^{n} V_{LJ}(r_i - r_j)
\]

Study \( n = 3, 4, 5 \). Pop quiz: What do resulting molecules look like?
COSY-GO Lennard-Jones potential for 4 molecules: Cutoff Value -- LDB/QFB
COSY-GO Lennard-Jones potential for 5 molecules: Number of Boxes -- LDB/QFB

To Be Studied

Small Boxes ( < 1e-6 )
COSY-GO Lennard-Jones potential for 5 molecules: Cutoff Value -- LDB/QFB
Lennard-Jones Potentials - Results

Find minimum with COSY-GO and Globsol.
Use TMs of Order 5, QFB&LFB.
Use Globsol in default mode.

<table>
<thead>
<tr>
<th>Problem</th>
<th>CPU-time needed</th>
<th>Max list</th>
<th>Total # of Boxes</th>
</tr>
</thead>
<tbody>
<tr>
<td>n=4, COSY</td>
<td>89 sec</td>
<td>2,866</td>
<td>15,655</td>
</tr>
<tr>
<td>n=5, COSY</td>
<td>1,550 sec</td>
<td>6,321</td>
<td>69,001</td>
</tr>
</tbody>
</table>
COSY-GO Lennard-Jones potential for 4 molecules: Number of Boxes -- LDB/QFB

To Be Studied
Small Boxes (< 1e-6)
## Lennard-Jones Potentials - Results

Find minimum with COSY-GO and Globsol.
Use TMs of Order 5, QFB&LFB.
Use Globsol in default mode.

<table>
<thead>
<tr>
<th>Problem</th>
<th>CPU-time needed</th>
<th>Max list</th>
<th>Total # of Boxes</th>
</tr>
</thead>
<tbody>
<tr>
<td>n=4, COSY</td>
<td>89 sec</td>
<td>2,866</td>
<td>15,655</td>
</tr>
<tr>
<td>n=5, COSY</td>
<td>1,550 sec</td>
<td>6,321</td>
<td>69,001</td>
</tr>
<tr>
<td>n=4, Globsol</td>
<td>5,833 sec</td>
<td></td>
<td>243,911</td>
</tr>
<tr>
<td>n=5, Globsol</td>
<td>&gt;60,530 sec</td>
<td></td>
<td>(not finished yet)</td>
</tr>
</tbody>
</table>
The Higher Order Bounder

After removing first and second order part of polynomial, we have

\[ P (\vec{x} - \vec{x}_0) = \tilde{P} (\vec{x} - \vec{x}_c) \]
\[ = b + \frac{1}{2} (\vec{x} - \vec{x}_c)^T H (\vec{x} - \vec{x}_c) + \tilde{P}_{>2} (\vec{x} - \vec{x}_c), \]

Goal: want to find nonlinear polynomial \( \vec{T} : \mathbb{R}^v \to \mathbb{R}^v \) such that with \( \vec{y} = (\vec{x} - \vec{x}_0) \), we have

\[ \frac{1}{2} \vec{T} (\vec{y})^T H \vec{T} (\vec{y}) = \frac{1}{2} \vec{y}^T H \vec{y} + \tilde{P}_{>2} (\vec{y}), \]
The Higher Order Bounder Algorithm

Will do this to arbitrary order, in an order-by-order fashion. Let $\tilde{T}_m(y)$ denote the part of $\tilde{T}(y)$ consisting of the terms of the $m$-th order, so that

$$\tilde{T}(y) = \sum_{m=0}^{n-1} \tilde{T}_m(y).$$

Let $\tilde{T}_{\leq m}(y) = \sum_{l=0}^{m} \tilde{T}_l(y)$.

Note $\tilde{T}_1(y) = y$. Let us now define a sequence of real-valued polynomial functions $S_m(y)$ by

$$S_m(y) = \tilde{P}_{\geq 2}(y) - \frac{1}{2} \tilde{T}_{\leq m-1}(y)^T H \tilde{T}_{\leq m-1}(y)$$

for $m = 1, 2, \ldots, n$. 
The Higher Order Bounder II

Assume we have determined $\tilde{T}_{\leq m-1}$. We want to determine $\tilde{T}_m$. Note that then, $S_m(\vec{y})$ has only terms of order $m + 1$ and higher. We demand

$$0 =_{m+1} \tilde{P}_{\geq 2}(\vec{y}) - \frac{1}{2} \left( \tilde{T}_{\leq m-1}(\vec{y}) + \tilde{T}_m(\vec{y}) \right)^T H \left( \tilde{T}_{\leq m-1}(\vec{y}) + \tilde{T}_m(\vec{y}) \right)$$

$$=_{m+1} \tilde{P}_{\geq 2}(\vec{y}) - \frac{1}{2} \tilde{T}_{\leq m-1}(\vec{y})^T H \tilde{T}_{\leq m-1}(\vec{y})$$

$$- \tilde{T}_{\leq m-1}(\vec{y})^T H \tilde{T}_m(\vec{y}) - \frac{1}{2} \tilde{T}_m(\vec{y})^T H \tilde{T}_m(\vec{y})$$

$$=_{m+1} \tilde{S}_{m-1}(\vec{y}) - \tilde{T}_{\leq m-1}(\vec{y})^T H \tilde{T}_m(\vec{y})$$

$$=_{m+1} \tilde{S}_{m-1}(\vec{y}) - \vec{y}^T H \tilde{T}_m(\vec{y}) .$$

This establishes a requirement for the sought $\tilde{T}_m(\vec{y})$. Now note that each term in $S_{m-1}$ contains at least one of the variables $y_1, \ldots, y_n$ comprising $\vec{y} = (y_1, \ldots, y_n)$. 
The Higher Order Bounder III

Now factor out one such term in term in $S_{m-1}$, and write

$$S_{m-1} = \bar{y}^t \cdot I \cdot \tilde{S}_{m-1}$$

Then we can satisfy condition on $\tilde{T}_m (\bar{y})$ by picking

$$\tilde{T}_m (\bar{y}) = H^{-1} \cdot \tilde{S}_{m-1}$$
Example: Smooth Function in 6 Dimensions

Let
\[ f (\bar{x}) = - \exp \left( -\frac{1}{2} g (\bar{x}) \right) + \frac{1}{4} \exp (-g (\bar{x})) \text{ for } \bar{x} \in B_j, \text{ where} \]
\[ g (\bar{x}) = \left( \sum_{i=1}^{v} (R\bar{x}_i)^2 \right) + \left( \exp \left( \frac{1}{2} \sum_{i=1}^{v} (R\bar{x}_i) \right) - 1 \right)^2 \]
with a \( v \times v \) rotation matrix \( R \). Has resemblance to a linear combination of two Gaussian functions.

Choose boxes
\[ B_j = a + 2^{-j-1} \cdot [-1, 1] \]
Figure 1: Logarithmic plot of the measurements of an upper bound $q$ of the overestimation in $l(f)$ with different orders $n = 3, \ldots, 9$ of Taylor models.

Figure 2: Plot of the empirical approximation order (EAO) for different orders $n = 3, \ldots, 9$ of Taylor model representations.
Figure 3: Logarithmic plot of the size $w(I)$ of the remainder bounds of Taylor models of different orders $n = 3, \ldots, 9$.

Figure 4: Plot of the empirical approximation order (EAO) of $w(I)$ for different orders $n = 3, \ldots, 9$ of Taylor model representations.
Figure 5: Logarithmic plot of an upper bound $q - w(I)$ of the overestimation in $l(P)$ of Taylor models of orders $n = 3, \ldots, 9$.

Figure 6: Logarithmic plot of the ratio of $q - w(I)$ to the size $w(I)$ of the remainder bounds of Taylor models of orders $n = 3, \ldots, 9$. 

30