Rosenbrock’s “Banana” Function

\[ f(x, y) = 100 \cdot (y - x^2)^2 - (1 - x)^2 \]

Study on \([-2, 2] \times [-2, 2]\).
Assumes min 0 at \((1, 1)\), but it is very difficult for gradient methods.

Picture from http://www.math.wm.edu/~buckaroo/classes/csci638/homework/project2.html
Nonvalidated Results of Rosenbrock’s Function

Using COSY’s true and tested default optimizers:

Starting point \((X,Y)\)\((-1.2,1.0)\)

Optimizer #> 1 : Simplex
Number of steps> 251
\(F>0.1711168421282399E-16\)
\((X,Y)\)\((1.000000004115731,1.000000008272989)\)

Optimizer #> 2 : LMDIF
Number of steps> 70424
\(F>0.9124815296170133E-10\)
\((X,Y)\)\((0.9999904485839599,0.99999809108990141)\)

Optimizer #> 3 : Simulated Annealing
Number of steps> 100003
\(F>0.5106520406572324E-05\)
\((X,Y)\)\((0.9977499955081044,0.9954840773134492)\)
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 $L^t DL$ decomposition step in interval arithmetic, beginning by representing every element of $H$ by a thin interval; in step $i$:

   (a) If $\mathbb{I}(D(i, i)) > \emptyset$ proceed to the next row and column.

   (b) If $\mathbb{I}(D(i, i)) < \emptyset$ exchange row and column $i$ with row and column $i + 1, i - 2, \ldots$ 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 -- IN

Number

Step (Number of Boxes Studied)

To Be Studied

Small Boxes (< 1e-6)
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 The Beale Function: Cutoff Value -- LDB/QFB
COSY-GO Beale 4D: Number of Boxes -- IN

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

Number

Step (Number of Boxes Studied)
COSY-GO Beale 4D: Number of Boxes -- CF

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

To Be Studied
Small Boxes ( < 1e-6 )
COSY-GO Beale 4D: Cutoff Value -- LDB/QFB
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></td>
</tr>
<tr>
<td></td>
<td>(not finished yet)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
The Higher Order Bounder

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

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

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

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

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

\[
\tilde{T} (\vec{y}) = \sum_{m=0}^{n-1} \tilde{T}_m (\vec{y}) . \quad \text{Let} \quad \tilde{T}_{\leq m} (\vec{y}) = \sum_{l=0}^{m} \tilde{T}_l (\vec{y}) .
\]

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

\[
S_m (\vec{y}) = \tilde{P}_{\geq 2} (\vec{y}) - \frac{1}{2} \tilde{T}_{\leq m-1} (\vec{y})^T \bar{H} \tilde{T}_{\leq m-1} (\vec{y}) \quad \text{for} \quad m = 1, 2, \ldots, n.
\]
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

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

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

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

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

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

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 $y_1, ..., y_n$ comprising $\vec{y} = (y_1, ..., y_n)$. 
The Higher Order Bounder III

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

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

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

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

Let

$$f (\vec{x}) = -\exp \left( -\frac{1}{2} g (\vec{x}) \right) + \frac{1}{4} \exp (-g (\vec{x})) \quad \text{for } \vec{x} \in B_j,$$

where

$$g (\vec{x}) = \left( \sum_{i=1}^{v} (R\vec{x})_{i}^{2} \right) + \left( \exp \left( \frac{1}{2} \sum_{i=1}^{v} (R\vec{x})_{i} \right) - 1 \right)^{2}$$

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 \( n = 3, \ldots, 9 \).
Quadratic Pruning - The Idea

Extract a convex quadratic part \( P_2 \) of Taylor model, write
\[
f(x) \in P_2(x) + R(x) + I \text{ where } P_2(x) = \frac{1}{2} x^t \cdot H \cdot x
\]

Want to confine the region \( P_2(x) \leq a \) with \( a > 0 \), by an interval box \([-x_m, x_m]\) with \( x_m > 0 \).

Because of positive definiteness and convexity, this region is inside a closed ellipsoidal contour surface \( P_2(x) = a \). The optimal confining interval box touches such a region at each box side surface tangentially, so the condition to find \( x_m \) is, for each dimensional direction, \( \nabla f \) is normal to the corresponding box surface, namely for determining \( x_{mk} \), the \( k \)-th component of \( x_m \),
\[
(\nabla P)_i = 0 \text{ for } \forall i \neq k.
\]
Quadratic Pruning - The Algorithm I

WLOG, choose $k = n$, where $n$ is the dimensionality of $x$. The condition for $i \neq n$ is

$$(H \cdot x)_i = 0 \text{ for } \forall i \neq n.$$  

Denote the $(n - 1)$ dimensional system of $H$ and $x$, obtained by removing the $n$-th components from $H$ and $x$, by $\tilde{H}$ and $\tilde{x}$. Using these, (can be expressed as

$$(H \cdot x)_i = (\tilde{H} \cdot \tilde{x})_i + h_{i,n}x_n = 0.$$  

where

$$\tilde{H} = \begin{pmatrix} h_{1,1} & h_{1,2} & \cdots & h_{1,n-1} \\ h_{1,2} & h_{2,2} & \cdots & h_{2,n-1} \\ \vdots & \vdots & \ddots & \vdots \\ h_{1,n-1} & h_{2,n-1} & \cdots & h_{n-1,n-1} \end{pmatrix}, \quad \tilde{x} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_{n-1} \end{pmatrix}, \quad \tilde{h}_n = \begin{pmatrix} h_{1,n} \\ h_{2,n} \\ \vdots \\ h_{n-1,n} \end{pmatrix}$$

Combining all the components, we have

$$\tilde{H} \cdot \tilde{x} + x_n\tilde{h}_n = 0.$$  

Thus

$$\tilde{x} = -\tilde{H}^{-1} \cdot \tilde{h}_n x_n.$$
Quadratic Pruning - The Algorithm II

Now, under this condition, the function $P_2$ is simplified as

$$P_2 = \frac{1}{2} x^t \cdot H \cdot x = \frac{1}{2} x_n (H \cdot x)_n$$

$$= \frac{1}{2} x_n^2 \left[ h_{n,n} - \tilde{h}_n^t \cdot \tilde{H}^{-1} \cdot \tilde{h}_n \right],$$

which contains only $x_n$ among all the components of $x$. Here, the last expression is derived as

$$(H \cdot x)_n = \tilde{h}_n^t \cdot \tilde{x} + h_{n,n} x_n = \tilde{h}_n^t \cdot \left( -\tilde{H}^{-1} \cdot \tilde{h}_n x_n \right) + h_{n,n} x_n.$$  

Demand the function value to be $a$ when $x_n = x_{mn}$. From above, such $x_{mn}$ can be determined as

$$x_{mn} = \sqrt{\frac{2a}{h_{n,n} - \tilde{h}_n^t \cdot \tilde{H}^{-1} \cdot \tilde{h}_n}}.$$
Quadratic Pruning - Example I

Consider

\[ f(x, y) = 2x^2 + y^2 = \frac{1}{2}(x, y) \cdot \begin{pmatrix} 4 & 0 \\ 0 & 2 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \end{pmatrix} \]

Let us demand the pruning value \( a = 1 \). We have

\[
\begin{align*}
x_m &= \sqrt{\frac{2 \cdot 1}{4 - 0 \cdot \frac{1}{2} \cdot 0}} = \sqrt{\frac{1}{2}}, \\
y_m &= \sqrt{\frac{2 \cdot 1}{2 - 0 \cdot \frac{1}{4} \cdot 0}} = 1.
\end{align*}
\]
Quadratic Pruning - Example II

The next example function is created by rotating the above \( f(x, y) \) by 30°. The function is now

\[
f(x, y) = \frac{7}{4}x^2 - \frac{\sqrt{3}}{2}xy + \frac{5}{4}y^2 = \frac{1}{2}(x, y) \cdot \left( \begin{array}{c}
\frac{7}{2} \\
-\frac{\sqrt{3}}{2}
\end{array} \right) \cdot \left( \begin{array}{c}
x \\
y
\end{array} \right),
\]

Again, we demand \( a = 1 \). Using the formula, at this time, we obtain \( x_m \) and \( y_m \) as

\[
x_m = \sqrt{\frac{2 \cdot 1}{\frac{7}{2} - \left( -\frac{\sqrt{3}}{2} \right) \cdot \frac{2}{5} \cdot \left( -\frac{\sqrt{3}}{2} \right)}} = \sqrt{\frac{5}{8}},
\]

\[
y_m = \sqrt{\frac{2 \cdot 1}{\frac{5}{2} - \left( -\frac{\sqrt{3}}{2} \right) \cdot \frac{2}{7} \cdot \left( -\frac{\sqrt{3}}{2} \right)}} = \sqrt{\frac{7}{8}}.
\]

As expected, \( x_m \) is larger and \( y_m \) is smaller, and the area size of the interval box is larger.
Rosenbrock Banana Function: TM LDB/QFB

- Number of Boxes
- Cutoff Value
Rosenbrock Banana Function: TM LDB/QFB
Some General Thoughts about Rigorous Parallel Optimization

1. Performance gains in modern computing are gained through multi-processor architectures, not increased clock speed and more efficient microcode.

2. While the global optimization task does not parallelize trivially, with due care it is manageable

Caveats:

1. Communication mode, in particular for large numbers of processors - point to point, master - slave, common meeting?

2. Load balancing, in particular with many processors and slow connections
Key Features and Algorithms of COSY-GO

- List management of boxes not yet determined to not contain the global minimizer. Loading a new box. Discarding a box with range above the current threshold value. Splitting a box with range not above the threshold value for further analysis. Storing a box smaller than the specified size.

- Application of a series of bounding schemes, starting from mere interval arithmetic to naive Taylor model bounding, LDB, then QFB. A higher bounding scheme is executed only if all the lower schemes fail.

- Update of the threshold cutoff value via various schemes. It includes upper bound estimates of the local minimum by corresponding bounding schemes, the mid point estimate, global estimates based on local behavior of function using gradient line search and convex quadratic form.

- Box size reduction using LDB QPB.

- Resulting data is available in various levels including graphics output.
COSY-GO in Parallel Environment

Design aspects of COSY-GO-P

1. Utilize MPI and be standard. This is done with a COSY language construct called **PLOOP**, a parallel loop with various types inter-processor updates upon conclusion. Can be nested.

2. Should scale from for **different numbers of processors**
   (a) multiple cores in a chip
   (b) large clusters with thousands of processors

3. Should scale for **different connection speeds**
   (a) extremely fast interconnect (multiple cores in one chip)
   (b) very fast (a few cores in a "node" with a nearly bus-like interconnect)
   (c) fast (specialized network for parallel use, at least Gigabit)
   (d) slow (grid-based systems - geographicly dispersed, relying on standard Internet)
Basic Ideas of the COSY-GO Parallel Environment

1. List Management: Each processor has two lists:
   (a) **Short List of large boxes**, shared with other processors
   (b) A section of Short List is pre-allocated to each processor.
   (c) **Long List of regular boxes** owned by each processor.
   (d) Long List is kept in moderately strict order of difficulty. Achieved by selection strategy favoring newer boxes

2. Communication Concept
   (a) Processors communicate in **scheduled meeting mode** after pre-determined fixed time interval $T_m$.
   (b) Time interval $T_m$ is determined by trial and error for each environment under consideration. Single node: fraction of second, Berkeley NERSC cluster (~6000 processors): 1-2 minutes, Grid systems: fractions of hours.
What Happens in a Meeting

1. **Assess status.** Gather short data from each processor, scatter this information to all others. Cutoff updates, number of remaining large boxes and small boxes

2. **Processing of results.** Global cutoff is updated; it is determined if we can stop code

3. **Processing of status.** Each processor simultaneously identifies
   
   (a) how many boxes $N_r$ are needed to replenish **Short List**
   
   (b) Let $N_p = N_r / N_{proc}$

4. **Load balancing.**
   
   (a) Each processor uploads its $N_p$ largest boxes, if available, to the Short List
   
   (b) The Short List is randomized, so that the sections allocated to each processor are roughly of similar complexity
What Happens Between Meetings

1. Each processor splits its time between
   (a) working on its Long List of boxes. For each box, perform a sequence of tests: interval evaluation rejection test; Taylor model evaluation: LDB, QFB bounders, Gradient-based box rejection with Gradient Taylor models
   (b) performing non-rigorous global search (currently via genetic algorithm) in its assigned search space of global boxes, as well as neighboring global boxes
2. If Long List of boxes is exhausted, retrieve a box from the processor’s section on the Short List
3. If processor’s section on Short List is exhausted, continue to perform non-rigorous global search as in 1b.
4. After appropriate time, join next meeting.
Normal Form Methods

Iterative order-by-order coordinate transformation to simplify dynamics around a fixed point.

Result: Except for resonances, up to order $n$,

- Elliptic case $\lambda_{i+1} = \bar{\lambda}_i$: spiral motion in $(\lambda_i, \lambda_{i+1})$ plane
- Elliptic unity case $\lambda_{i+1} = \bar{\lambda}_i$ and $|\bar{\lambda}_i| = 1$: circular motion, radius-dependent rotation frequency
- Hyperbolic case ($\lambda_i$ real) motion along $\vec{e}_i$ axis, expanded or contracted by $\lambda_i$

Practical use:

- Can be performed rigorously in Taylor model arithmetic
- Implemented to arbitrary order in arbitrarily many variables in COSY INFINITY
The Normal Form Defect Function

• **Extreme cancellation**: one of the reasons TM methods were invented
• Six-dimensional problem from dynamical systems theory
• Describes invariance defects of a particle accelerator
• Essentially composition of three tenth order polynomials
• The function vanishes identically to order ten
• Study for $a \cdot (1, 1, 1, 1, 1, 1)$ for $a = .1$ and $a = .2$

• **Interesting Speed observation**: on same machine,
  * one CF in INTLAB takes 45 minutes
  * one TM of order 7 takes 10 seconds

$$f_4(x_1, \ldots, x_6) = \sum_{i=1}^{3} \left( \sqrt{y_{2i-1}^2 + y_{2i}^2} - \sqrt{x_{2i-1}^2 + x_{2i}^2} \right)^2$$

where $\vec{y} = \vec{P}_1 \left( \vec{P}_2 \left( \vec{P}_3(\vec{x}) \right) \right)$
GlobSol Results

For the computations, GlobSol’s maximum list size was changed to $10^6$, and the CPU limit was set to 10 days. All other parameters affecting the performance of GlobSol were left at their default values.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>CPU-time needed</th>
<th>Max list</th>
<th>Total # of Boxes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>18810 sec</td>
<td></td>
<td>4733</td>
</tr>
<tr>
<td>3</td>
<td>&gt;562896 sec (not finished yet)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>&gt;259200 sec (could not finish)</td>
<td>63446 (remaining)</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>&gt; 86400 sec (could not finish)</td>
<td>21306 (remaining)</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>not attempted</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

We observe that in this example, COSY outperforms GlobSol by many orders of magnitude. However, we are not completely sure if a different choice of parameters for GlobSol could result in better performance.
COSY-GO Results

Tolerance on the sharpness of the resulting minimum is $10^{-10}$. For the evaluation of the objective function, Taylor models of order 5 were used. For the range bounding of the Taylor models, Makino’s LDB with domain reduction was being used.

<table>
<thead>
<tr>
<th>Dimension</th>
<th>CPU-time needed</th>
<th>Max list</th>
<th>Total # of Boxes</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>5.747071 sec</td>
<td>11</td>
<td>31</td>
</tr>
<tr>
<td>3</td>
<td>38.48828 sec</td>
<td>44</td>
<td>172</td>
</tr>
<tr>
<td>4</td>
<td>346.8604 sec</td>
<td>357</td>
<td>989</td>
</tr>
<tr>
<td>5</td>
<td>3970.746 sec</td>
<td>2248</td>
<td>6641</td>
</tr>
<tr>
<td>6</td>
<td>57841.94 sec</td>
<td>17241</td>
<td>49821</td>
</tr>
</tbody>
</table>
Fig. 9. Projection of the normal form defect function. Dependence on two angle variables for the fixed radii $r_1 = r_2 = 5 \cdot 10^{-4}$

<table>
<thead>
<tr>
<th>Region</th>
<th>Boxes studied</th>
<th>CPU-time</th>
<th>Bound</th>
<th>Transversal Iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>$[0.2, 0.4] \cdot 10^{-4}$</td>
<td>82,930</td>
<td>30,603 sec</td>
<td>$0.859 \cdot 10^{-13}$</td>
<td>$2.3283 \cdot 10^8$</td>
</tr>
<tr>
<td>$[0.4, 0.6] \cdot 10^{-4}$</td>
<td>82,626</td>
<td>30,603 sec</td>
<td>$0.587 \cdot 10^{-12}$</td>
<td>$3.4072 \cdot 10^7$</td>
</tr>
<tr>
<td>$[0.6, 0.9] \cdot 10^{-4}$</td>
<td>64,131</td>
<td>14,441 sec</td>
<td>$0.616 \cdot 10^{-11}$</td>
<td>$4.8701 \cdot 10^6$</td>
</tr>
<tr>
<td>$[0.9, 1.2] \cdot 10^{-4}$</td>
<td>73,701</td>
<td>13,501 sec</td>
<td>$0.372 \cdot 10^{-10}$</td>
<td>$8.0645 \cdot 10^5$</td>
</tr>
<tr>
<td>$[1.2, 1.5] \cdot 10^{-4}$</td>
<td>106,929</td>
<td>24,304 sec</td>
<td>$0.144 \cdot 10^{-9}$</td>
<td>$2.0833 \cdot 10^5$</td>
</tr>
<tr>
<td>$[1.5, 1.8] \cdot 10^{-4}$</td>
<td>111,391</td>
<td>26,103 sec</td>
<td>$0.314 \cdot 10^{-9}$</td>
<td>$0.95541 \cdot 10^5$</td>
</tr>
</tbody>
</table>

Table 8
Global bounds obtained for six radial regions in normal form space for the Tevatron. Also computed are the guaranteed minimum transversal iterations.


