Available online at www.sciencedirect.com

SCIENCE DIRECT®

Nuclear Instruments and Methods in Physics Research A 558 (2006) 1–10

NUCLEAR INSTRUMENTS & METHODS IN PHYSICS RESEARCH Section A

www.elsevier.com/locate/nima

Long-term stability of the Tevatron by verified global optimization ☆

Martin Berz*, Kyoko Makino, Youn-Kyung Kim

Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, USA

Available online 28 November 2005

Abstract

The tools used to compute high-order transfer maps based on differential algebraic (DA) methods have recently been augmented by methods that also allow a rigorous computation of an interval bound for the remainder. In this paper we will show how such methods can also be used to determine rigorous bounds for the global extrema of functions in an efficient way. The method is used for the bounding of normal form defect functions, which allows rigorous stability estimates for repetitive particle accelerator. However, the method is also applicable to general lattice design problems and can enhance the commonly used local optimization with heuristic successive starting point modification. The global optimization approach studied rests on the ability of the method to suppress the so-called dependency problem common to validated computations, as well as effective polynomial bounding techniques. We review the linear dominated bounder (LDB) and the quadratic fast bounder (QFB) and study their performance for various example problems in global optimization. We observe that the method is superior to other global optimization approaches and can prove stability times similar to what is desired, without any need for expensive long-term tracking and in a fully rigorous way.

PACS: 29.20.Dh; 29.27.Bd; 02.60.Pn; 06.20.Dk

Keywords: Maps with remainder; Taylor model; Global optimization; Linear dominated bounder LDB; Quadratic fast bounder QFB; COSY INFINITY; COSY-GO; Symbolic-numeric; Interval; Taylor model

1. Introduction

In this paper we describe a practical method to rigorously assess the long-term stability of storage rings and other repetitive systems. The approach is based on methods of rigorous global optimization and will be applied to the study of the dynamics of the Tevatron at Fermilab. The theoretical foundation of the stability estimate is the method of normal forms [1,2] that allows to determine a family of three approximate invariants I of the motion. The defect of these invariants I, i.e. the quantity

$$d = \max(I(\mathscr{M}) - I)$$

where $\mathscr{M}$ is the transfer map describing one revolution, can be used to provide bounds for minimum stability. In fact, if the distance of the outermost approximate invariant to the disallowed region is a, then the number of iterations necessary to reach this region is apparently at least a/d. Thus, bounding d from above allows to assert stability, and the sharpness of the bound directly determines the quality of the stability estimate.

High-order map methods relating final and initial conditions of phase space variables for one revolution of the accelerator by the transfer map $\mathscr{M}$ have grown into a very widely used tool since their inception [3]; many details can be found in [2]. Recently, methods have been developed that allow not only the computation of the maps themselves, but also rigorous bounds for their remainders [4–6]. As we shall illustrate, these methods can also be used very beneficially for the problem of rigorously finding the global maximum or minimum of a function, and thus for the rigorous estimate of stability time of the Tevatron.

☆This work was supported by the US Department of Energy, the National Science Foundation, the German National Merit Foundation, and an Alfred P. Sloan Fellowship. We thank Ray Moore for many fruitful discussions.

*Corresponding author.

E-mail addresses: berz@msu.edu (M. Berz), makino@msu.edu (K. Makino), kimyounk@msu.edu (Y.-K. Kim).

0168-9002/$ - see front matter © 2005 Elsevier B.V. All rights reserved.
doi:10.1016/j.nima.2005.11.035

In the following sections we will briefly outline general ideas about rigorous global optimization and how they can be enhanced using the tools to compute remainder bounds for maps. We will illustrate the performance of these methods with various examples, and then apply them to the study of the Tevatron normal form defect problem. We will arrive at a rigorous stability estimate for the Tevatron that is practically relevant.

2. Rigorous global optimizers

Rigorous, validated, or verified global optimization characterizes the attempt of finding bounds of the global maximum or minimum of a sufficiently smooth function over a given domain. This task is distinct from the widely known and applied method of local optimization in that not only a nearby local extremum is sought. Furthermore, enclosures for the extrema and the points where they are assumed are obtained in a fully mathematically rigorous way, including the accounting of all effects of numerical inaccuracy.

2.1. Interval-based methods

The simplest method for validated global minimization is the interval branch-and-bound approach that successively studies sub-boxes of the original domain and attempts to prove that they cannot contain the minimum, upon which they are discarded. Usually these algorithms are based on a stack of boxes to be studied, which is initialized with the original full domain, as well as a rigorous upper bound for the minimum, the so-called cutoff, which is initialized with the result of a function evaluation in interval arithmetic at a suitably chosen point.

There are many variations used in practice, but the most elementary approach is based on picking the oldest box from the stack, evaluating the function on the center of the box to possibly obtain an update for the cutoff, and determining a lower bound of the function over the box by interval evaluation. If this lower bound exceeds the current cutoff, the box can be discarded; if it does not, then the box will be split along its longest direction and the two resulting boxes will be added to the stack for further study. By executing all operations in outward rounded interval arithmetic, full rigor of the argument is retained. This is not the place to do justice to the wide field of interval global optimization, but we rather refer to some of the common references on the topic [7–10].

Different from local optimization, the complexity of which often scales polynomially in dimension, that of global optimization can frequently grow exponentially with dimension. This is exemplified by the function f_n on $[-1,1]^n$ in n variables x_i given by $f(x_1,\ldots,x_n)=\sum_{i=1}^{n}\cos(x_i/2\pi)$; apparently, each term in the sum assumes a minimum for $x_i=\pm\frac{1}{2}$, so that there are 2^n local minima. While local optimizers will settle into one of these depending on the starting point chosen, global optimizers at the very least have to probe all of them.

Performing global optimization using interval tools can introduce further complications. The two most important ones are that

(1) Interval methods are known for the fact that the resulting range bounds contain potentially significant overestimation, depending on the complexity of the function; this is frequently referred to as the *dependency problem*.
(2) It is usually observed that as the global optimization algorithm progresses, the number of boxes in the stack that lie in the vicinity of a local minimum remains nearly constant for a long time and thus all these neighboring boxes have to be split in each step, slowing down the process of elimination [11]; this is frequently referred to as the *cluster effect*.

2.2. Control of the dependency problem

It was recognized in Ref. [12] that the computation of a map with remainder bound of a functional dependency can significantly reduce the dependency problem. The reason for this simple yet very powerful observation is that the dominating part of the description of the function is caught by the Taylor coefficients. Thus, any operations on representations of functions merely require floating point arithmetic for the treatment of coefficients, and thus no overestimation occurs (with the exception of small effects due to the limited mantissa length and the resulting floating point errors, which are rigorously accounted for [13]). Overestimation does occur, as with all interval arithmetic, within the arithmetic of the remainder bound; but the influence of this remainder bound is many orders of magnitude smaller than that of the coefficients and thus its harm is reduced accordingly.

To illustrate this phenomenon, let us consider the following simple one-dimensional function:

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

which is recognized to be the power series representation of $\cos(x)$ up to order 60; over the domain $[0,4\pi]$, it represents $\cos(x)$ to an accuracy of better than 10^{-15}. In Fig. 1 we show the overestimation of the true range of the function (which is assumed to be that of $\cos(x)$) for the naive interval method, as well as the more advanced approaches of mean value form and centered form (see for example Ref. [9]) for domains of width 2^{-j} for the values $j=1,\ldots,8$ around $x=\pi/4$. The results are compared with what is obtained by first determining the map representation of the function, and then evaluating the polynomial part of this map representation in interval arithmetic. Apparently all methods become sharper for larger j, but the sharpness of

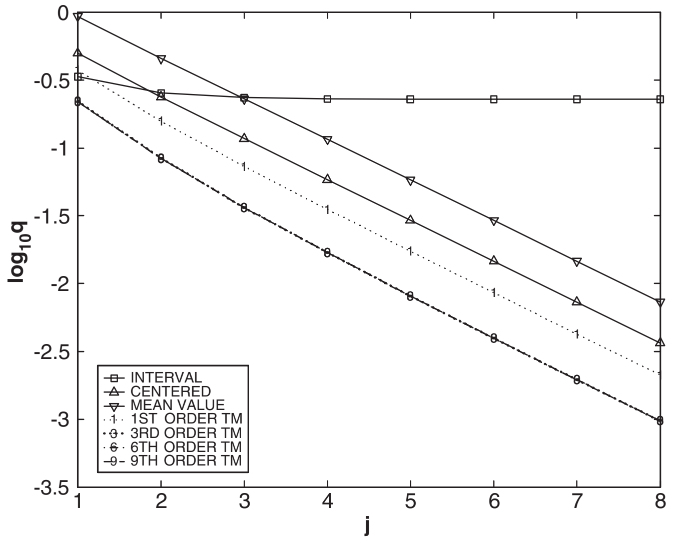

Fig. 1. Relative overestimation for the range bounding by the interval-, mean value-, and centered form methods, as well as with maps with remainders of orders 1, 2, 6, and 9 at $x = \pi/4$.

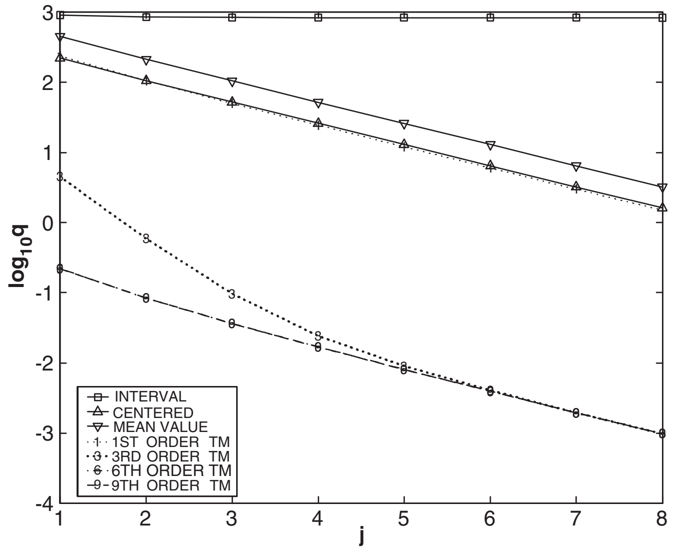

Fig. 2. Relative overestimation for the range bounding by the interval-, mean value-, and centered form methods, as well as with maps with remainders of orders 1, 2, 6, and 9 at $x = 2\pi + \pi/4$.

the map approach is superior because the map representation has less dependency than the original problem.

Performing the same comparison again for $x = 2\pi + \pi/4$, where the function has the same form except that the dependency problem is more significant, we see that the map approach provides a significant improvement, as seen in Fig. 2. Indeed, the higher order map methods achieve the same amount of overestimation as in the previous case, while the interval-, centered form- and mean value form methods suffer from overestimation due to the dependency problem.

A substantially more detailed study of the behavior of the method for many other cases can be found in Ref. [14].

2.3. *The linear dominated bounder, LDB*

As we saw, the map representation with remainder bound can significantly reduce the dependency problem. However, it is even possible to obtain sharper bounds yet by replacing the interval evaluation of the polynomial part with more sophisticated methods. In particular, we will in the following study the so-called linear dominated bounder (LDB) first introduced in Ref. [15]. It is based on using the map representation to determine a linear function that is a lower bound to the original function, and use this linearization and simple linear constraint methods to successively reduce the domain that can contain the extremum.

Within the framework of validated global optimization, after reducing the dependency problem with map methods as shown above, this will now lead to a tool to effectively reduce or eliminate boxes. While there is nothing that can help the inherent exponential complexity of high-dimensional problems, it will help improve practical performance significantly. Given a domain D, we consider the representation of the function f by a Taylor polynomial P and a rigorous bound for the remainder I, which can be obtained by the above-mentioned methods [4–6] such that

$$f(x) \in P(x) + I \quad \text{for all } x \in D.$$

In case we are away from a stationary point, the linear part of P will dominate the behavior of the representation. The linear dominated bounder utilizes the linear part as a guideline for iterative domain reduction to bound P. Specifically, the algorithm is as follows.

LDB Algorithm

(1) Re-expand P at the mid-point c of D, call the resulting polynomial P_m and the centered domain D_1.
(2) Make the linear coefficients L_i of P_m all positive by flipping coordinate directions as necessary; call the resulting polynomial P_+.
(3) In step n, compute interval bounds of the linear (I_1) and nonlinear (I_h) parts of P_+ in D_n. The minimum is then bounded by $[M, M_{\text{in}}] := \underline{I}_1 + I_h$. If applicable, lower M_{in} by the current cutoff, the actual function value at the lower endpoints, and that at the midpoint.
(4) Let $d = \text{width}([M, M_{\text{in}}])$. If d lies below a termination value, stop. Otherwise, if $L_i \neq 0$, the domain containing the minimum can be restricted to $\overline{D}_{n+1,i} := \min(\underline{D}_{n,i} + d/L_i, \bar{D}_{n,i})$. Re-expand P_+ at the mid-point c of D_{n+1}. Prepare the new coefficients L_i, and continue with step 3.

Any errors associated with re-expansion and estimating point values are included in the remainder error bound interval. If f is monotonic, the exact bound is often enclosed with high accuracy. If only a threshold cutoff test is needed, the resulting domain reduction or elimination is often very effective.

We now consider the performance of the LDB approach for the problem discussed in the previous sections. Figs. 3 and 4 contain the behavior of Figs. 1 and 2 on the left, as well as the results of using the LDB bounder on the right. Observe that even in a high-dependency case, the LDB bounder significantly outperforms the other methods, achieving accuracies that exceed those of conventional interval-based tools by 10 orders of magnitude.

2.4. The quadratic fast bounder, QFB

The natural next idea of the bounding of the polynomial is to explicitly bound the quadratic part of P. This will help in cases where the linear part alone is not dominating, for example, in the proximity of a local minimizer. Exact range bounding of a general quadratic polynomial has a complexity that scales exponentially with dimension and can thus be expensive. A preliminary scheme of a quadratic dominated bounder (QDB) is discussed in Ref. [15], and a more advanced method as well as the bounder QFB that will be discussed in the following is introduced in Ref. [16].

However, obtaining a lower bound of the quadratic polynomial near a local minimizer, which is the most important problem in global optimization, turns out to be much simpler. Indeed, in sufficient proximity of an isolated interior minimizer of the function f, the Hessian of f is positive definite, and so the purely quadratic part of a representation $P+I$ that locally encloses f also has a positive definite Hessian matrix H. The actual definiteness can be tested in a validated way using the common LDL or extended Cholesky decomposition [16]. The quadratic fast bounder (QFB) provides a lower bound of $P+I$ cheaply when the purely quadratic part is positive definite. It is based on the following observation.

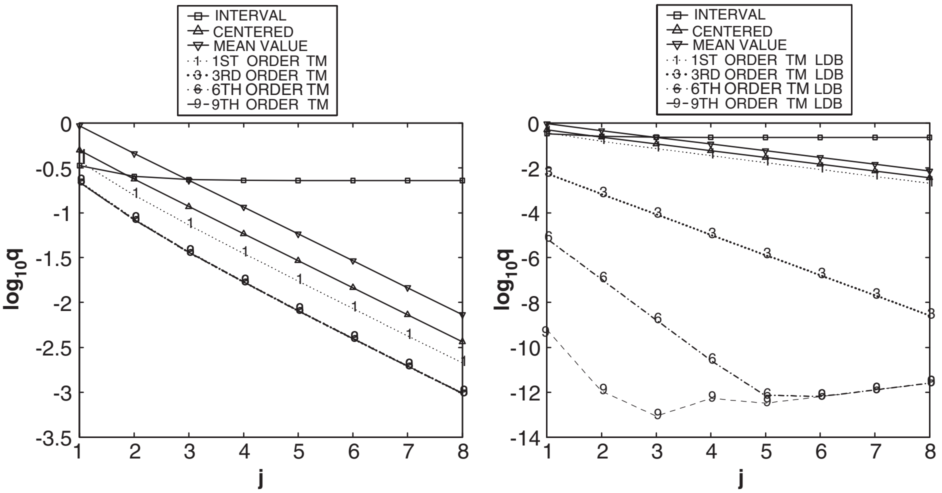

Fig. 3. Overestimation for the range bounding by the interval-, mean value-, centered form methods, as well as with maps with remainders of orders 1, 2, 6, and 9 at $x=\pi/4$ for naive interval evaluation of P (left) and using the LDB bounder (right).

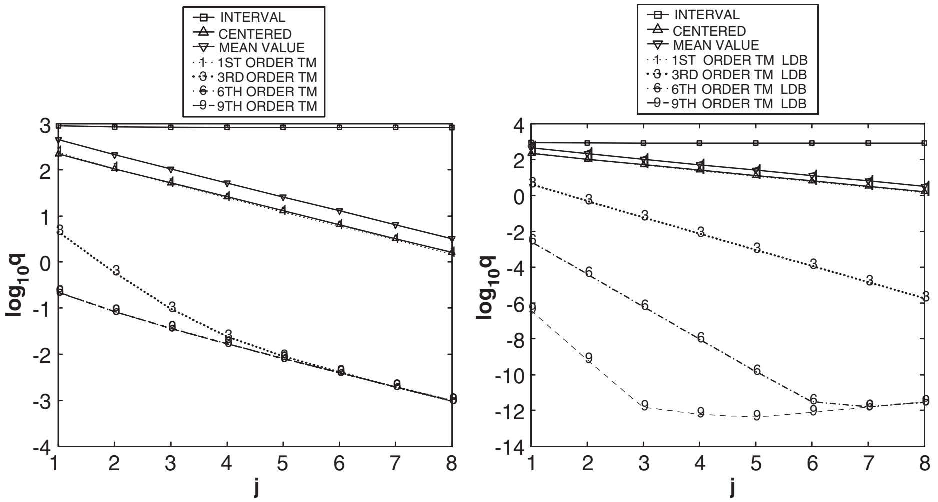

Fig. 4. Overestimation for the range bounding by the interval-, mean value-, centered form methods, as well as with maps with remainders of orders 1, 2, 6, and 9 at $x=2\pi+\pi/4$ for naive interval evaluation of P (left) and using the LDB bounder (right).

Given the quadratic form

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

where H is symmetric, we determine the so-called Ordered LDL (OLDL) Decomposition (L: lower diagonal with unit diagonal, D: diagonal) of H as follows:

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

Similar to what is done in conventional Cholesky or LDL decomposition, it is useful to give careful consideration to the case in which $0 \in D(i,i)$, or $D(i,i)$ is too small. In this case, it is useful to apply a small correction C to H, i.e. to study $H + C$ instead of H, such that all elements of D are clearly positive or negative. As is typical also in non-validated LDL decomposition, C is usually chosen to be diagonal, and $|C|$ is lumped into the remainder bound of the original problem. The resulting LDL decomposition has the important property that sufficiently near a local minimizer, D will contain only positive elements that are sufficiently away from zero.

In the wider vicinity of a local minimum, the method may still possibly determine that all diagonal terms of D can be proven positive, in which case positive definiteness is asserted. If this is not the case, at least the upper part where $D(i,i) > 0$ will describe a large positive semi-definite subspace.

In the following we study the case in the near proximity of a minimizer where the OLDL decomposition succeeds to assert that H is positive definite. We will now use that knowledge to determine a sharp lower bound for f. Let $f \in P + I$ over D. We decompose the polynomial P into two parts and write

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

Then a lower bound for $P + I$ is obtained as

$$l(P + I) = l(P - Q) + l(Q) + l(I). \tag{1}$$

For the purpose of the QFB algorithm, we choose

$$Q = Q_{x_0} = \tfrac{1}{2}(x - x_0)^t H (x - x_0) \tag{2}$$

with some suitable $x_0 \in D$.

Since H is positive definite, $l(Q_{x_0}) = 0$, and the value 0 is attained (at $x = x_0$). The remaining $P - Q_{x_0}$ does not contain pure quadratic terms anymore, but consists of linear as well as third and higher order terms $P_{>2}$. If x_0 is chosen to be the minimizer of the quadratic part P_2 of P in D, then x_0 is also a minimizer of the remaining linear part (a consequence of the Kuhn–Tucker conditions), and so the lower bound estimate (1) is optimally sharp. Thus, by choosing x_0 sufficiently close to the minimizer in D of P_2, the contribution of $P_2 - Q_{x_0}$ to the lower bound can become arbitrarily small.

A simple and efficient way to determine a sequence $x_0^{(n)}$ of candidates for x_0 is based on determining the “feasible descent direction”

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

and to move in the direction of $g^{(n)}$ until we hit the bounding box or a one-dimensional quadratic minimum along the line. The method is very fast, can cover large ground per step, and in the terminology of constrained optimization, can change the set of active constraints very quickly. As a result, we obtain an inexpensive third order cutoff test that requires very few, if any, iterations to determine a useful x_0.

2.5. The validated global optimizer, COSY-GO

For the example problems of validated global optimization in the next sections, we apply three branch-and-bound methods available in the code COSY-GO [17]. The first one is the optimizer utilizing the LDB and QFB algorithms. We compare the performance with two other optimizers; one based on mere interval bounding (IN) and one based on bounding with centered form (CF). The subdomain box list management is performed in the same way for all three optimizers. At each step in which a subdomain is being studied for possible elimination, the following tasks are performed:

(1) A lower bound l of the function is obtained using various bounding schemes in a hierarchical manner. If the lower bound is above the cutoff value, the box is eliminated; if not, the box is bisected.
   (a) Interval bounding of the polynomial P is utilized for all optimizers; if it fails to eliminate the box, then additional tests are performed.
   (b) For the CF optimizer, centered form bounding is performed. For the COSY-GO optimizer, as a first test the polynomial part is evaluated in interval arithmetic. When it fails to eliminate the box, the LDB bounding and possible domain reduction is executed. If it also fails to eliminate the box, and if the quadratic part of the polynomial P is positive definite, QFB bounding is performed.
(2) The cutoff value is updated using various schemes.
   (a) The conventional midpoint test is performed for all optimizers.

(b) For the COSY-GO optimizer, the linear and quadratic parts of P are utilized to obtain a potential cutoff update. In particular, if the quadratic part of the polynomial is positive definite, the minimizer of the quadratic polynomial is tested. If the quadratic part is not positive definite, the minimizer of the quadratic part in the direction of the negative gradient is tested.

3. Validated global optimization—illustrative examples

In the following, we study the performance of the various versions of validated optimizers discussed above for some illustrative examples. We also compare with the performance of non-validated local optimization.

3.1. The Rosenbrock function

As a first example, we consider a relatively benign-looking function of two variables that contains a single local minimum; but the minimum occurs along a long and very shallow parabolic valley. This function originally proposed by Rosenbrock has the form

$$f_R(x, y) = 100(y - x^2)^2 + (1 - x)^2$$

and it apparently assumes its minimum of 0 at $(x, y) = (1, 1)$.

We study the function over the domain $[-1.5, 1.5]^2$, where its values range from the minimum 0 to more than 1400 near the point $(-1.5, -1.5)$. Fig. 5 shows a three-dimensional rendering of the function and the seemingly innocent parabolic valley generated by the first term, as well as a set of logarithmic contour lines for function values $0.1 \times 10^{i/2}$ for $i = 1, \ldots, 13$, revealing the substructure of the parabolic valley generated by the second term.

This function causes difficulties even for powerful conventional local minimizers because as soon as the optimizer is probing not exactly inside the valley, the direction of steepest descent is nearly perpendicular to the valley and hence in the almost completely wrong direction. Table 1 shows the performance of various tried-and-true optimizers in COSY [18,19], the LMDIF algorithm based on steepest descent with various enhancements, the Simplex algorithm which is particularly powerful for non-smooth problems, as well as the Anneal approach based on stochastic search of the global minimum by simulated annealing. The starting point was chosen at $(-1.2, 1.0)$, which is near the valley but on the opposite side of the true minimum. For LMDIF and Simplex, an accuracy tolerance of 10^{-12} and a maximum number of steps of 100,000 was specified, and Anneal was given a total number of 100,000 annealing steps. Table 1 shows the performance of the three non-validated optimizers. It can be seen that the usually quite powerful LMDIF algorithm performs rather poorly and is in this example significantly outperformed by the Simplex algorithm.

Now we utilize various validated global optimization tools to study the problem. In particular, we use the common Moore–Skelboe algorithm and determine ranges of the function by the mere interval evaluation method, and also the usually more accurate centered form method. From the perspective of interval methods, the function has a rather benign form with little dependency, and the squares that appear in the two terms can even be handled exactly. We compare these methods with the COSY-GO optimizer. Table 2 shows the performance of these three methods. It is apparent that for COSY-GO, the number of processing steps, i.e. the number of boxes considered, compares very favorably even with the performance of the best non-validated optimizer, aside from the fact that it of course determines a rigorous global optimum. On the other hand, the Interval and CF methods each require approximately one order of magnitude more steps, but still compare rather favorably to LMDIF and Anneal. It is also interesting to note that in a significant number of cases, LDB could reduce the size of a box under consideration without actually fully eliminating it.

Table 1
Performance of various local optimizers for finding the minimum of the Rosenbrock function in $[-1.5, 1.5]^2$ from the starting point $(-1.2, 1.0)$

	LMDIF	Simplex	Anneal
Number of steps	100,000	225	100,000
Error in $f(x, y)$	1×10^{-10}	2×10^{-13}	3×10^{-4}
Error in (x, y)	2×10^{-5}	4×10^{-7}	6×10^{-3}

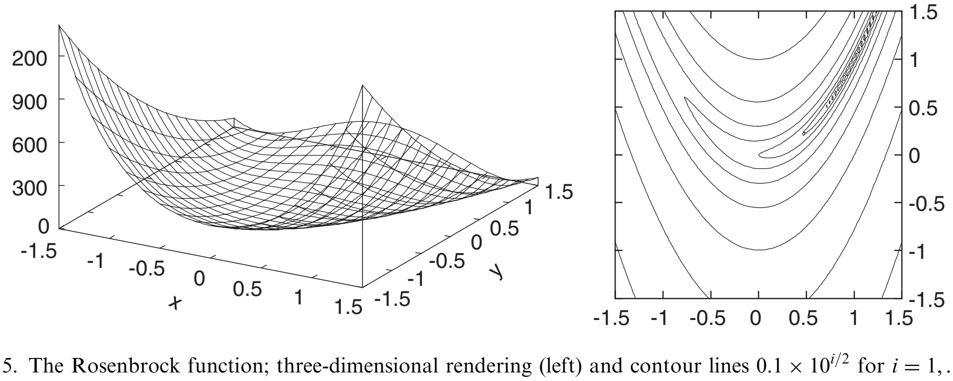

Fig. 5. The Rosenbrock function; three-dimensional rendering (left) and contour lines $0.1 \times 10^{i/2}$ for $i = 1, \ldots, 9$.

Table 2
Performance of various validated global optimizers for finding the minimum of the Rosenbrock function in $[-1.5, 1.5]^2$

	IN	CF	COSY-GO
Total box processing steps	1325	1325	143
Max number of active boxes	47	47	9
Retained small boxes ($<10^{-6}$)	15	15	1
LDB domain reduction steps	–	–	43

We now study in more detail the performance of the validated global optimizers. To this end we show all boxes that were studied in the process within the original domain box. Fig. 6 shows these for both the interval- and COSY-GO methods. Apparently, both methods successfully eliminate relatively large boxes away from the minimizer, while the boxes tend to get smaller and smaller as the minimizer is approached; however, the size of the boxes rejected by COSY-GO is significantly larger.

On the right of Fig. 6 we show the number of currently active boxes, as well as the value of the current upper bound of the minimum, the so-called cutoff value. For a

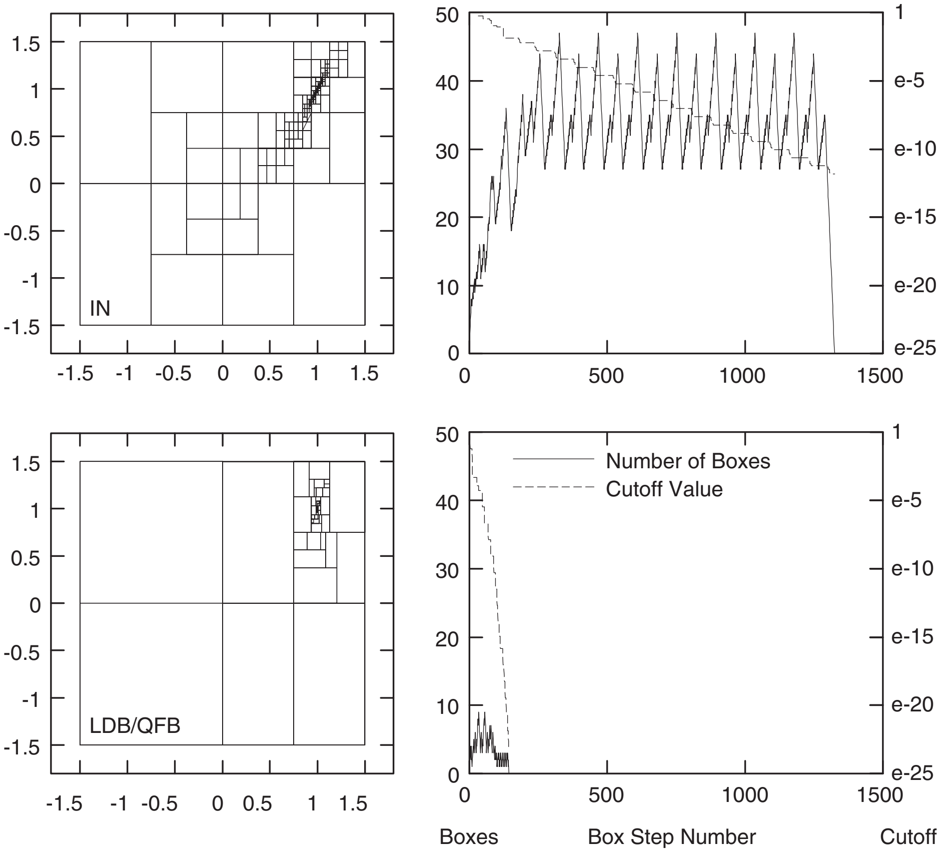

Fig. 6. The minimum search for the Rosenbrock function in $[-1.5, 1.5]^2$ by the interval and the LDB/QFB optimizers. Left: subdomain boxes studied. Right: number of active boxes and cutoff value as function of step number.

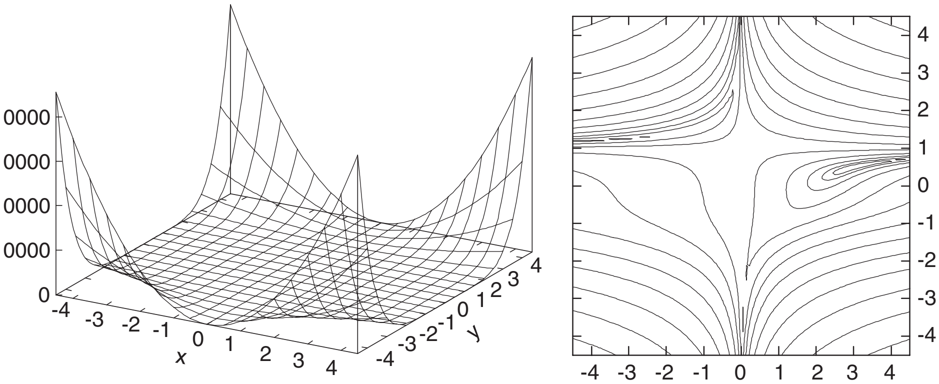

Fig. 7. The Beale function; three-dimensional rendering (left) and contour lines $0.1 \times 10^{i/2}$ for $i = 1, \ldots, 13$.

long time, the interval method maintains a nearly constant number of around 35 active boxes, while COSY-GO never require more than 10 active boxes, and soon keeps only one or two.

3.2. The Beale function

The next example is the Beale function [20]

$$f_B(x_1,x_2) = (1.5 - x_1(1-x_2))^2 + (2.25 - x_1(1-x_2^2))^2 + (2.625 - x_1(1-x_2^3))^2.$$

The problem is to find the minimum in the initial domain $[-4.5,4.5] \times [-4.5,4.5]$; one easily verifies that the function assumes the minimum 0 at $(3,0.5)$. Fig. 7 shows the behavior of the function, exhibiting a nearly flat extended valley containing the minimum. However, logarithmic contour lines of levels $0.1 \times 10^{i/2}$ for $i = 1,\ldots 13$ show a subtle substructure of the valley, having a trench-like shape near the minimizer $(3,0.5)$, but another trench of small function values in the upper left quadrant.

In particular this additional "trench" of small function values makes optimization difficult. We begin the study by using the three default optimizers in COSY, with an accuracy tolerance of 10^{-12} and a maximum of 100,000 iterations. Starting at $(x,y) = (4,-4)$ yields reasonable convergence in under 1000 steps for both LMDIF and Simplex, as shown in Table 3. On the other hand, the situation changes drastically when using as a starting value the point $(x,y) = (-4,4)$. In this case, the local optimizers get caught by the wrong trench of the objective function, and fail to get near the global minimum. The simulated annealing tool still succeeds in finding a reasonable approximation of the global minimum; the details of the performance are summarized in Table 4.

On the other hand, all validated global optimizers have no difficulty finding the global minimum accurately. Significant differences exist, however, in the speed with which this is achieved; the results are summarized in Fig. 8 and Table 5. Square expressions in f_B are executed as multiplications.

Table 3
Performance of various local optimizers for finding the minimum of the Rosenbrock function from the starting point $(4,-4)$ in $[-4.5,4.5]^2$

	LMDIF	Simplex	Anneal
Number of steps	825	123	100,000
Error in $f(x,y)$	7×10^{-13}	3×10^{-13}	3×10^{-2}
Error in (x,y)	2×10^{-6}	4×10^{-7}	8×10^{-2}

Table 4
Performance of various local optimizers for finding the minimum of the Beale function from the starting point $(-4,4)$ in $[-4.5,4.5]^2$

	LMDIF	Simplex	Anneal
Number of steps	100,000	100,000	100,000
Error in $f(x,y)$	6×10^{-1}	4×10^{0}	1×10^{-4}
Error in (x,y)	2×10^{1}	5×10^{0}	3×10^{-2}

Table 5
Performance of various validated global optimizers for finding the minimum of the Beale function in $[-4.5,4.5]^2$

	IN	CF	COSY-GO
Total steps	3407	3285	353
Max boxes	236	234	52
Retained boxes	25	25	3
LDB reductions	–	–	108

Table 6
Performance of the validated global optimizer COSY-GO for a generic normal form defect function

Dimension	CPU time (s)	Max list	Total boxes
2	5.747	11	31
3	38.49	44	172
4	346.9	357	989
5	3970	2248	6641
6	57,842	17,241	49,821

We observe no significant advantage in the CF optimizer compared to the interval optimizer, both of which maintain a list of about 60 active boxes for an extended time. On the other hand, the COSY-GO optimizer significantly outperformed both others because of more efficient box rejection and LDB domain size reduction, and requires a number of boxes comparable to the number of steps needed for the local optimizers in the case of a favorable initial condition.

4. Long-term stability of the Tevatron

We now return to the study of long-term stability of the Tevatron storage ring at collision. We utilize the four-dimensional map of lattice with parameters as described and optimized by Snopok et al. [21]. To illustrate the behavior of the normal form defect function, Fig. 9 shows a two-dimensional projection in which the two normal form radii are frozen at 5×10^{-4}, and the two normal form angles moving from 0 to 2π. Already in these low-dimensional projections it becomes apparent that the functions have a large number of local minima and maxima, which makes finding their global extrema difficult.

In order to assess the performance of the COSY-GO global optimizer, we attempt a comparison with the validated global optimizer GLOBSOL [7]. As it turned out, the very high demands on the sharpness of the upper bound of the maximum of the normal form defect function did not allow the use of GLOBSOL for the specific problem at hand. So for the purpose of comparison, we chose a different, less demanding normal form defect bounding problem based on the polynomials available at bt.pa.msu.edu. In Tables 6 and 7 we show some parameters describing the performance of COSY-GO and GLOBSOL for subspaces of different dimensionality. GLOBSOL succeeds to complete only the two-dimensional case in a reasonable time, while COSY-GO succeeds to complete even the six-dimensional case in similar time. For the two-dimensional case, COSY requires much less than 1% of the CPU time of GLOBSOL. The maximum list length and total number of boxes studied are rather manageable (Table 8).

Table 7
Performance of the validated global optimizer GLOBSOL for a generic normal form defect function

Dimension	CPU time (s)	Max list	Total Boxes
2	18,810		4723
3	$>562,896$		–

We next apply COSY-GO to the realistic Tevatron problem. The one sigma emittance of the beam translates

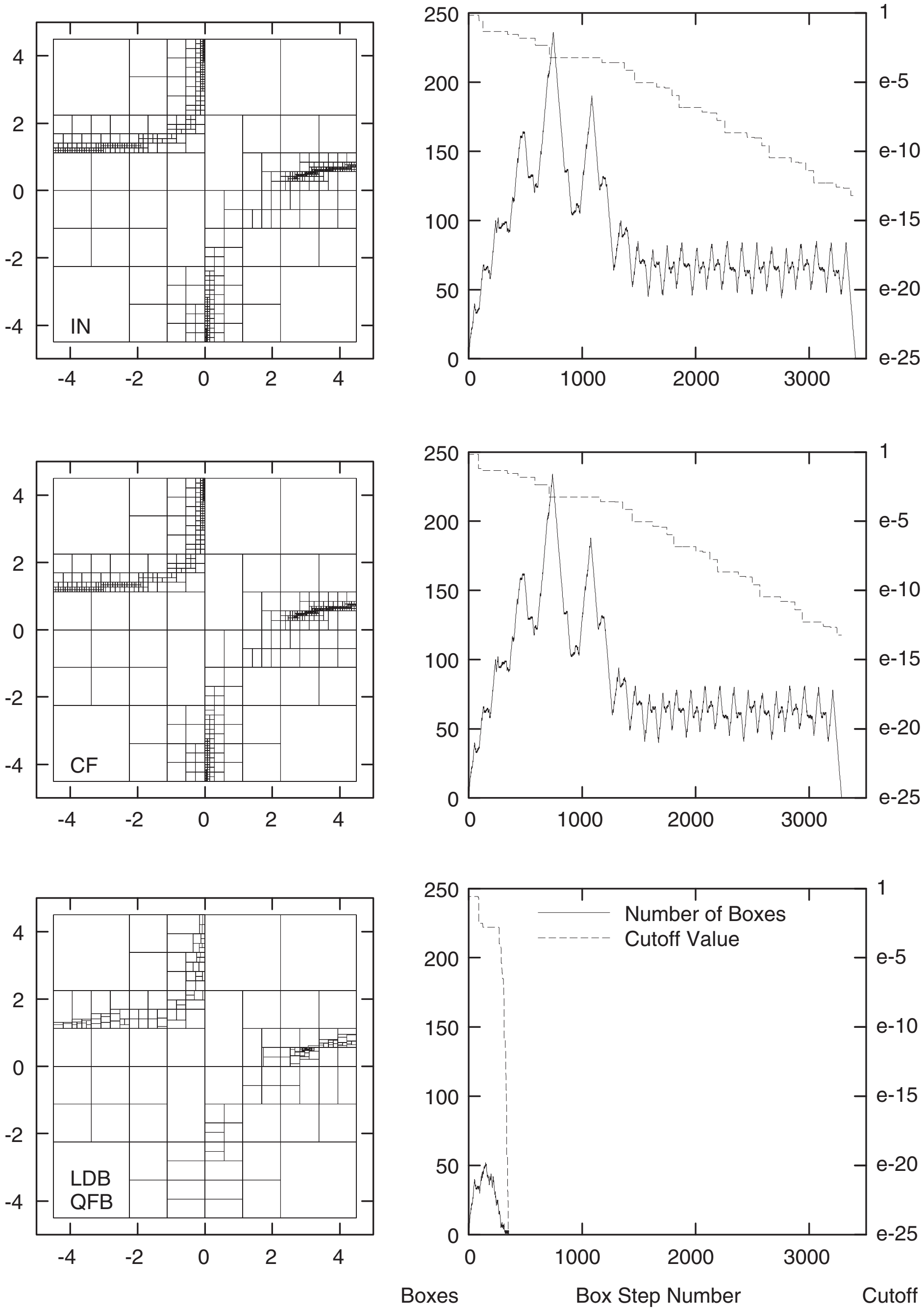

Fig. 8. Minimum search for the Beale function in $[-4.5, 4.5]^2$ by the interval, the centered form, and the LDB/QFB optimizers. Left: subdomain boxes studied. Right: number of active boxes and cutoff value as a function of step number.

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

Region	Boxes studied	CPU time (s)	Bound	Transversal iterations
$[0.2, 0.4] \times 10^{-4}$	82,930	30,603	0.859×10^{-13}	2.3283×10^{8}
$[0.4, 0.6] \times 10^{-4}$	82,626	30,603	0.587×10^{-12}	3.4072×10^{7}
$[0.6, 0.9] \times 10^{-4}$	64,131	14,441	0.616×10^{-11}	4.8701×10^{6}
$[0.9, 1.2] \times 10^{-4}$	73,701	13,501	0.372×10^{-10}	8.0645×10^{5}
$[1.2, 1.5] \times 10^{-4}$	106,929	24,304	0.144×10^{-9}	2.0833×10^{5}
$[1.5, 1.8] \times 10^{-4}$	111,391	26,103	0.314×10^{-9}	0.95541×10^{5}

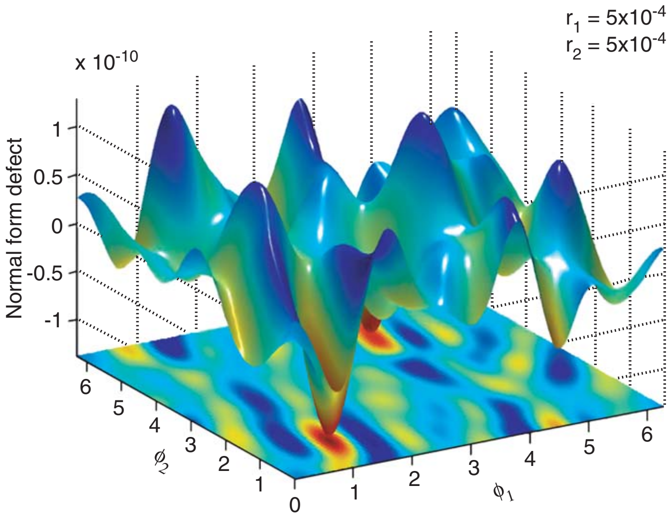

Fig. 9. Projection of the normal form defect function. Dependence on two angle variables for the fixed radii $r_1 = r_2 = 5 \times 10^{-4}$.

into a normal form radius of approximately 0.12×10^{-4}. We study the magnitude of the normal form defect bound for various different radius bands, and determine the ratio of width of band to normal form defect bound, which gives the minimum transversal iterations for the band. Fig. 9 shows the result of the calculations; it becomes apparent that it requires more than 2.7×10^{8} revolutions for a particle starting with a normal form radius below 0.2×10^{-4} to be lost.

References

[1] M. Berz, Differential algebraic formulation of normal form theory, in: M. Berz, S. Martin, K. Ziegler (Eds.), Proceedings of Nonlinear Effects in Accelerators, IOP Publishing, London, 1992, p. 77.

[2] M. Berz, Modern Map Methods in Particle Beam Physics, Academic Press, San Diego, 1999 Also available at ⟨http://bt.pa.msu.edu/pub⟩.

[3] M. Berz, Particle Accelerators 24 (1989) 109.

[4] M. Berz, K. Makino, New approaches for the validation of transfer maps using remainder-enhanced differential algebra, Nucl. Instr. and Meth. A 519 (2004) 53.

[5] M. Berz, K. Makino, Reliable Comput. 4 (4) (1998) 361.

[6] K. Makino, M. Berz, Suppression of the wrapping effect by Taylor model based validated integrators, submitted for publication, also MSUHEP-040910, available at ⟨http://bt.pa.msu.edu/pub⟩.

[7] R.B. Kearfott, Rigorous Global Search: Continuous Problems, Kluwer, Dordrecht, 1996.

[8] E. Hansen, Global Optimization using Interval Analysis, Marcel Dekker, New York, 1992.

[9] R.E. Moore, Interval Analysis, Prentice-Hall, Englewood Cliffs, NJ, 1966.

[10] R.E. Moore, Methods and Applications of Interval Analysis, SIAM, Philadelphia, 1979.

[11] R.B. Kearfott, K. Du, J. Global Optim. 5 (1994) 253.

[12] K. Makino, M. Berz, Reliable Comput. 5 (1) (1999) 3.

[13] N. Revol, K. Makino, M. Berz, J Logic and Algebraic Programming 64 (1) (2004) 135 University of Lyon LIP Report RR 2003-11, MSU Department of Physics Report MSUHEP-030212, ⟨http://bt.pa.msu.edu/pub⟩.

[14] K. Makino, M. Berz, Int. J. Pure and Appl. Math. 6 (3) (2003) 239 available at ⟨http://bt.pa.msu.edu/pub⟩.

[15] K. Makino, Rigorous Analysis of Nonlinear Motion in Particle Accelerators. Ph.D. Thesis, Michigan State University, East Lansing, Michigan, USA, 1998, also MSUCL-1093.

[16] K. Makino, M. Berz, The LDB, QDB, and QFB bounders, Technical Report MSUHEP-40617, Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, June 2004.

[17] K. Makino, M. Berz, Nucl. Instr. Meth., in print.

[18] M. Berz, J. Hoefkens, K. Makino, COSY INFINITY Version 8.1—programming manual, Technical Report MSUHEP-20703, Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, 2002, see also ⟨http://cosy.pa.msu.edu⟩.

[19] M. Berz, K. Makino, COSY INFINITY Version 8.1—user's guide and reference manual, Technical Report MSUHEP-20704, Department of Physics and Astronomy, Michigan State University, East Lansing, MI 48824, 2002, see also ⟨http://cosy.pa.msu.edu⟩.

[20] More, Garbow, Hillstrom, ACM Trans. Math. Software 7 (1981) 17.

[21] P.V. Snopok, C.J. Johnstone, M. Berz, D.A. Ovsyannikov, A.D. Ovsyannikov, Nucl. Instr. and Meth., in print.