Book review for
Numerica: A Modeling Language for Global Optimization
by
Pascal van Hentenryck, Laurent Michel, and Yves Deville

The overall subject area encompassing this book is the numerical solution of nonlinear systems of equations and constrained and unconstrained optimization. More precisely, the book describes certain techniques for finding *all* solutions to nonlinear systems of equations and to finding *global* optima. Until recently unrecognised by many researchers in the field, such computational methods both provide mathematical rigor and are applicable to many practical problems. The authors have a commercial implementation, ILOG Numerica, that embodies both their own variants of these methods and a modeling language to interface well with the methods and to be user-friendly. The book introduces the underlying methods and the modeling language, presents computational results obtained with Numerica, and guides the reader in use of Numerica. Some details are given below.

1 Overall Subject Area

Part of the basis of the computational procedures in Numerica is *interval arithmetic*, in which operations are performed on intervals, rather than numbers. The crucial fact about implementatations of interval arithmetic with floating point arithmetic is *rigorous enclosure of ranges.* That is, if $\boldsymbol{x} = [\underline{x}, \overline{x}]$ is an interval (or, more generally, an interval vector), and $\boldsymbol{f}(\boldsymbol{x})$ is an interval value obtained by evaluating an expression f with interval arithmetic, then

$$\left\{ f(x) \mid x \in \boldsymbol{x} \right\} \subseteq \boldsymbol{f}(\boldsymbol{x}),$$

i.e. the interval value contains the range. Since, with proper rounding control, this computational result contains the range with *mathematical rigor*, interval arithmetic is used in various automatic theorem proving contexts. Interval arithmetic is thus particularly elegant and powerful in handling systems of inequality constraints and in global optimization, where bounds on the range of an objective function are valuable.

For a simple illustrative example, suppose it is to be determined whether the constraint $x_1^2 - x_2 \geq 0$ is valid in the rectangle, i.e. in the interval vector $([1,2],[3,4])^T$. An interval evaluation of the the left side of the constraint yields

$$[1,2]^2 - [3,4] = [1,4] - [9,16] = [-15,-5] < 0.$$

Thus, every point in $([1,2],[3,4])^T$ is infeasible with respect to the constraint $x_1^2 - x_2 \geq 0$.

Until recently, knowledge of interval arithmetic was either limited or eschewed. Many numerical analysts had not recognised that it is possible to compute rigorous bounds on ranges. Among those who had heard of interval arithmetic, a common objection was that the "bounds on the range are so pessimistic that they are useless." This is true for naive, inappropriate use of interval arithmetic, and overly optimistic early claims about interval arithmetic led to disbelief. However, research in the past twenty years has led to clearer views of the possibilities, more practical algorithms, and solution of significant application problems with interval methods. Numerica is a manifestation of such research.

The elements of interval arithmetic necessary for understanding the other subject matter are introduced clearly in *Numerica: A Modeling Language for Global Optimizaton*.

On the application level, the book *Numerica* addresses two main problems of nonlinear programming:

- the numerical solution of inequality-constrained nonlinear systems of equations,

$$\boxed{\begin{array}{lccc} \text{Find } \textit{all} \text{ solutions to} & f(x) & = & 0 \\ \text{subject to} & g(x) & \le & 0, \end{array}} \tag{1}$$

  where $f : \boldsymbol{x} \subset \mathbf{R}^n \to \mathbf{R}^n$ and $g : \boldsymbol{x} \to \mathbf{R}^{m_2}$, $m_2 \ge 0$, where $\boldsymbol{x}$ is an interval vector

$$\boldsymbol{x} = ([\underline{x}_1, \overline{x}_1], \ldots, [\underline{x}_n, \overline{x}_n])^T,$$

  and

- Constrained nonlinear optimization:

$$\boxed{\begin{array}{ll} \text{minimize} & \phi(x) \\ \text{subject to} & \left\{ \begin{array}{cclc} c(x) & = & 0 & \text{and} \\ g(x) & \le & 0, & \end{array} \right\}, \end{array}} \tag{2}$$

  where $\phi : \boldsymbol{x} \subset \mathbf{R}^n \to \mathbf{R}$, $c : \boldsymbol{x} \to \mathbf{R}^{m_1}$, and $g : \boldsymbol{x} \to \mathbf{R}^{m_2}$, where $\boldsymbol{x}$ is an interval vector, and $m_1 \ge 0$ and $m_2 \ge 0$. Here, a *global* optimum, that is, the lowest possible value of ϕ over the feasible set, is sought.

In problem 2, it is possible to consider all or some of the interval bounds $\underline{x}_i$ and $\overline{x}_i$ as constraints. In *Numerica*, the concept "min-stable," basically the assumption that the minimum of ϕ occurs in the interior of the compact search region $\boldsymbol{x}$, is introduced. If this assumption is violated, some or all of the bounds $x_i \ge \underline{x}_i$ and $x_i \le \overline{x}_i$ can be introduced as explicit inequality constraints.

2 Main Research Contributions of the Authors

The book *Numerica* is not a research monograph, but it does contain polished results of original work of the authors. These include

- A unique implementation of constraint-solving, and
- A modeling language to fit both the users' thought processes and the solution process.

Constraint-solving or constraint propagation is a fairly well-known technique in artificial intelligence in the form of *constraint logic programming.* As a simplified example of the underlying idea for interval constraint propagation, suppose we are interested in the portion of the box $\boldsymbol{x} = ([-1, 1], [0, 2])^T$ that is feasible with respect to the constraint system

$$\begin{array}{rcccl} c(x) & = & x_1^2 + x_2^2 - 1 & = & 0, \\ g(x) & = & x_1 + x_2 & \le & 0. \end{array}$$

Then we can solve for one variable in terms of the others in each of the constraints, to obtain sharper bounds. For example, we can solve for x_2 in $c(x) = 0$ to obtain

$$\begin{array}{rcl} x_2 & \in & -\sqrt{1 - \boldsymbol{x}_1^2} \cup \sqrt{1 - \boldsymbol{x}_1^2} = [-1, 1], \quad \text{whence} \\ x_2 & \in & [-1, 1] \cap [0, 2] = [0, 1]. \end{array}$$

We may then solve for x_1 in $g(x) \le 0$ to obtain

$$\begin{array}{rcl} x_1 & \le & -x_2 \in [-1, 0], \quad \text{whence} \\ x_1 & \in & [-1, 0]. \end{array}$$

We have thus reduced the initial set $([-1, 1], [0, 2])^T$ to the smaller set $([-1, 0], [0, 1])^T$. In some cases, it is possible for this technique alone to reduce an initial region $\boldsymbol{x}$ to a point, or to prove that no feasible point exists in $\boldsymbol{x}$.

This basic idea can be used in various ways. For example, each variable can be solved in each equation, as above. Alternately, as in [1] or [5, Ch. 7], the constraints can be parsed, and the relationships among the intermediate results produced during evaluation of the expressions defining the constraints can be used. In *Numerica*, the authors have developed a special univariate bisection method to work with the original constraints. This method appears to be effective on many problems.

Some experts have implemented interval constraints in the logic programming (artificial intelligence) language PROLOG; see [2], [6], or [7]. A disadvantage of PROLOG implementations of algorithms for problems 1 or 2 is that execution can be extremely slow. One idea behind the authors' modeling language Numerica is increased efficiency.

The book *Numerica* itself needs to be read to grasp the full scope of the authors' new language. However, as an example, take Problem 4.3 from [3]:

$$\text{minimize} \quad x_1^{0.6} + \big(3(x_1+u_1)\big)^{0.6} - 6x_1 - 4u_1 + 3u_2$$

$$\text{subject to} \quad \left\{ \begin{array}{rcll} x_1 + 2u_1 & \leq & 4 & \text{and} \\ 3x_1 + 3u_1 + 2u_2 & \leq & 4 & \end{array} \right\}.$$

A corresponding Numerica file is:

```
Variable:
      x1 in [1e-2..3];
      u1 in [0..2];
      u2 in [0..1];
Body:
   minimize
      x1^0.6 + (3*(x1+u1))^0.6 - 6*x1 - 4*u1 + 3*u2
   subject to
      x1 + 2*u1 <= 4;
      3*x1 + 3*u1 + 2*u2 <= 4;
Display:
   dynamic;
```

Others, as in [4] or [1], have developed special languages or systems for constraint techniques. However, [4] is a general system without specific regard to global optimization, while the algorithms underlying [1] are different.

3 Contents of the Book

Chapter 1 contains an explanation of nonlinear programming and constrained global optimization problems, along with a guide to the rest of the book. This guide includes recommendations of which chapters are important for different classes of readers, such as engineers applying the package, researchers in the underlying methods, etc. In Chapter 2, a clear, abbreviated explanation is given of the main features of the Numerica modeling language. Chapter 3 deals with (i) the elements of interval analysis, (ii) a clear theory of what a solution is (to aid in the interpretation of results given by Numerica), and (iii) an explanation of constraint solving. In Chapter 4, examples are given of problems for which Numerica fails, such as problems with an infinite number of solutions. Chapter 4 then goes on to indicate how problems can be re-posed to improve Numerica's performance. (This includes restating the objective and constraints in various ways; considerations are based mainly on properties of interval arithmetic.) Chapter 5, good for reference purposes, gives a complete description of Numerica's syntax. Chapter 6, entitled "The Semantics of Numerica," includes formal definitions of interval extensions of functions, as well as formal definitions of the relationship between the results Numerica gives and the exact solution to the original problem. The purpose of the formality in Chapter 6 to aid intuitive understanding. Chapter 7 describes some details of the underlying algorithms used in the implementation of Numerica. Chapter 8 presents a sizable set of test problems, as well as numerous tables of performance results of running Numerica on these test problems. Two appendices give (i) a concise, diagramatic depiction of Numerica's syntax, and (ii) Numerica input files for the test problems.

4 Overall Assessment and Recommendations

The book, overall, is clearly written, and should be accessible to non-experts. It can serve both as a user guide to the Numerica package and as an introduction to various techniques of interval global optimization. Additionally, the implementation details contain some ideas of possible interest to researchers in the field. The test results give an indication of the practicality and scope of applicability of the Numerica package; additionally, the completely specified test problem set can serve for benchmarking.

In summary, the book is *not* a comprehensive overview or a comprehensive reference to research in interval techniques for global optimization. The book *is* both a good introduction to interval global optimization and constraint solving and a good user guide to the Numerica package. Although not comprehensive, the book *does* contain items of interest to researchers in the techniques for global optimization and nonlinear systems.

References

[1] A. B. Babichev, O. B. Kadyrova, T. P. Kashevarova, A. S. Leshchenko, and A. L. Semenov. UniCalc, a novel approach to solving systems of algebraic equations. *Interval Computations*, 1993(2):29–47, 1993.

[2] F. Benhamou. Interval constraint logic programming. In A. Podelski, editor, *Constraint Programming: Basics and Trends*, pages 1–21, Berlin, 1995. Springer-Verlag.

[3] C. A. Floudas and P. M. Pardalos. *A Collection of Test Problems for Constrained Global Optimization Algorithms*. Lecture Notes in Computer Science no. 455. Springer-Verlag, New York, 1990.

[4] E. Hyvönen and S. De Pascale. Interval computations on the spreadsheet. In R. B. Kearfott and V. Kreinovich, editors, *Applications of Interval Computations*, Applied Optimization, pages 169–209, Dordrecht, Netherlands, 1996. Kluwer.

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

[6] H. Koch, A. Schenkel, and P. Wittwer. Computer–assisted proofs in analysis and programming in logic: A case study. *SIAM Rev.*, 38(4):565–604, December 1996.

[7] P. Van Hentenryck. *Constraint Satisfaction in Logic Programming.* MIT Press, Cambridge, MA, 1989.