A NEW SUBDIVISION STRATEGY FOR RANGE COMPUTATIONS

P. S. V. NATARAJ AND S. SHEELA

Systems and Control Engineering, Department of Electrical Engineering,
Indian Institute of Technology, Bombay, 400 076 India.
Email: nataraj@ee.iitb.ernet.in

Abstract. We present a new subdivision strategy in interval analysis for computing the ranges of functions. We show through several real-world examples that the proposed subdivision strategy is more efficient than the widely used uniform and adaptive subdivision strategies of Moore [11].

1. Introduction

A fundamental problem in numerical analysis is computing the range of a function of several variables in an n-dimensional rectangle. Interval analysis [5], [11] provides several techniques to solve this problem.

Let f be a real function of n variables given by an expression $f(x_1, x_2, \ldots, x_n)$, where $x_1, x_2, \ldots, x_n$ are real numbers in the n-dimensional box $X = (X_1, X_2, \ldots, X_n)$ and $X_1, X_2, \ldots, X_n$ are closed bounded intervals on the real line. If the arithmetic operations and elementary functions in $f(x_1, x_2, \ldots, x_n)$ are replaced by the corresponding interval arithmetic operations and interval elementary functions, and the real variables x_i are replaced by the corresponding interval variables X_i, then we obtain the natural interval extension, denoted as $F(X_1, \ldots, X_n)$, of f. We shall assume that F is defined on X.

We wish to compute the range of values of f on X

$$\bar{f}(X) = \{f(x) : x \in X\}$$

Since it is in general not possible to compute the exact range $\bar{f}(X)$, we therefore consider the problem of finding an interval enclosure of $\bar{f}(X)$ with a desired degree of accuracy ε.

We can obtain with a single interval arithmetic evaluation, an interval $F(X)$ that encloses the exact range. However, this interval $FX)$ usually overestimates the range considerably. Moore [10], [11] therefore proposed the tool of subdivision to compute $\bar{f}(X)$ to a desired accuracy. Moore actually suggested two different strategies for subdivision. Both these strategies are well-known and are extensively used in various applications of interval analysis, see, for example, [9], [14].

In the first subdivision strategy called as uniform subdivision, we find a uniform subdivision factor N for all X_i, making use of Lipschitz constant and inequality relation in [11, equation 4.5]. Then, we subdivide each X_i into N equal subintervals with this subdivision factor, and create a so-called *uniform* subdivision partition. Lastly, we evaluate F over the boxes of the uniform partition, in a *parallel* manner, using vectorized operations, and take the union of all these evaluations of F. Moore [11, sec. 4.1] showed that the obtained union indeed encloses the range with the desired accuracy ε.

The second kind of subdivision strategy is adaptive subdivision. In this strategy, we first evaluate F over the current box X, and check the width of the resulting interval against the specified maximum width ε. If the specified width is exceeded, then we bisect the box into two boxes in the coordinate direction in which the box is longest, and discard the original box. We pick any one of the two boxes, and put the other box in a processing list. We then successively bisect the current box till the width of the resulting evaluation of F is at most ε. Then, we write the information to a solution list, and discard the current box. We repeat the above for all boxes created in this process. Finally, we take the union of all the items (i.e., the evaluations of F) present in the solution list to get an enclosure of the exact range with accuracy ε.

We briefly compare uniform and adaptive subdivisions. The merit of uniform subdivision is that it is essentially a single step one - a range enclosure of desired accuracy can be generated with a single interval evaluation of F for each box of the partition, and moreover, this can be done in a *parallel* manner for all the boxes using vectorized interval arithmetic operations. The difficulty with this strategy is that N is usually considerably overestimated. Consequently, a much larger number of boxes than required may be generated, with correspondingly greater computational effort.

The merit of adaptive subdivision is that it generates a considerably much smaller number of boxes, because of its "adaptiveness" - by which we mean that a box is successively subdivided into smaller boxes only as long as the resulting width of the evaluation of F is unacceptable. However, as each box is processed sequentially, the overall process generally takes considerably more time than uniform subdivision.

In short, the attractive feature of uniform subdivision is parallel evaluation of F over all boxes of a partition, while that of adaptive subdivision is "adaptiveness". In this note, we propose a new kind of subdivision strategy that combines these two advantageous features. We therefore call this new subdivision strategy as parallel-adaptive subdivision, and present it next.

2. A new subdivision strategy

Parallel-Adaptive Subdivision

- Inputs : An expression for the function $f(x)$, the initial box X, and the specified accuracy ε.
- Output: An enclosure of the exact range $\bar{f}(X)$, having the specified accuracy ε.

BEGIN Procedure

1. Construct a natural interval extension $F(X)$ of f.
2. (Adaptive subdivision and parallel evaluation)
   (a) Bisect in the longest direction all boxes present, in a parallel fashion, and discard all boxes just used for bisection.
   (b) Using vectorized operations, perform *parallel evaluation* of F over all boxes present.
   (c) Deposit all evaluations of F whose widths are less than ε in the solution list L^{sol}, and discard the corresponding boxes from further processing [1].
   (d) If there are no more boxes left for processing, go to the following step. Else, go back to the beginning of this step (of adaptive subdivision and parallel evaluation), and repeat.
3. Take the union of all the intervals (i.e., the evaluations of F having acceptable widths) present in the solution list L^{sol}, and print it out as the desired enclosure of the exact range of values of f on X of accuracy ε.

END Procedure

Remark 2.1. *Parallel-adaptive subdivision generates the same number of boxes as adaptive subdivision, as the subdivision procedure in both is based on the same concept of "adaptiveness". However, the former executes faster since it involves bisection and evaluation of F in parallel over all boxes, as opposed to sequential processing in the latter. Therefore, it is advantageous to use parallel-adaptive over adaptive subdivision strategy in all problems.*

Remark 2.2. *Uniform subdivision generally generates a much larger number of boxes than parallel adaptive subdivision, and needs correspondingly greater effort. This is because the subdivision factor N is generally heavily overestimated, due to overestimation in the Lipschitz constant calculation* [15] *and due to the inequality nature of Moore's relation* [11, equation 4.5]. *Therefore, it is generally much more advantageous to use parallel-adaptive subdivision over uniform subdivision.*

[1] The corresponding boxes are no longer needed, as these have produced evaluations of F with desired widths and these evaluations have just been stored.

Remark 2.3. *In this work, by* parallel *we mean* simultaneous *processing of all boxes present in a subdivision partition. This usage of the term* parallel *is not to be confused with the one associated with parallel processing as done on parallel computers.*

Remark 2.4. *To illustrate how a function can be evaluated in* parallel *even on an ordinary (sequential) computer, consider, for example, the* 1- *dimensional manifold*

$$F(x_1, x_2) = x_1^3 - x_1 x_2^2 + x_1^2 - x_1 x_2 - x_2^2$$

and suppose we want to evaluate F *over a* set *of boxes. Then, using the notation of INTLAB* [17] *(which is based on MATLAB), we can do this using the single program statement*

$$F = power\,(x(:,1),3) - x(:,1).*sqr\,(x(:,2)) + sqr\,(x(:,1)) - x(:,1).*x(:,2) - sqr\,(x(:,2))$$

where, $x(:,1)$ *denotes* x_1 *for all boxes and* $x(:,2)$ *denotes* x_2 *for all boxes. The function evaluation over all the boxes is done in a* parallel *manner with this statement, because the operations* $+, -, *$, *sqr, and power are performed element-wise between vectors (cf.* [1]*) and INTLAB overloads these ordinary arithmetic operations with the corresponding interval arithmetic operations* [17].

Remark 2.5. *The parallel-adaptive strategy does not require a parallel computer. It can be run on any computer that has an interval arithmetic compiler supporting vectorized interval arithmetic operations, such as INTLAB* [17] *or Forte Fortran 95* [2]. *The main contribution of this work is to show that* parallelization *is efficient even on* serial *architectures where good vectorization brings advantages.*

All of the preceding, of course, applies to the case of vector-valued functions $f : R^n \to R^m$.

3. Computational results

We next test and compare the performances of the various subdivision strategies on some examples. For all our computations, we use a single processor PC/Pentium III 550 MHz machine with 384 MB RAM, and the interval arithmetic toolbox INTLAB of Rump [17]. We emphasize that no parallel processors are used in our tests.

In all examples, $f = (f_1, f_2)$ where f_1 represents the function for the phase angle and f_2 represents the function for the magnitude of a system. The set $\mathcal{F} := \{f(x) : x \in X\}$ defines a region in the angle-magnitude plane, called as the template or value set of the system. We wish to generate a collection of (angle-magnitude) rectangles covering template $\mathcal{F}$, where each rectangle has a width at most equal to ε. Clearly, this problem is a more involved version of the range computation problem, and is known as the template generation or value set computation problem in robust control, see for instance, [3], [4], [8].

We test the various subdivision strategies to generate the system templates on a suite of several real-world examples taken from the engineering literature. The examples are listed in Appendix A. We choose the units for the magnitude as decibels (dB), and for phase angle as degrees. In these units, we specify the width of each generated rectangle as $\varepsilon = 1$. That is, we wish to generate the template in each example to an accuracy of 1 deg and 1 dB.

The performances of the various subdivision rules are compared, in terms of number of template rectangles (also called as solution boxes), the execution time in seconds taken to generate these boxes, and number of floating operations required (*flops*). The results are given in Table 1.

In this Table, an entry marked with star denotes the number of solution boxes (estimated using Lipschitz constant and Moore's inequality relation referred above), but these boxes could not be actually generated as the computer runs out of memory.

We see from Table 1 that parallel-adaptive subdivision generates the same number of solution boxes but in much less time and *flops* than adaptive subdivision. On the other hand, uniform subdivision could provide a

TABLE 1. Performance comparison of the various subdivision strategies on a suite of real-world examples.

	Example	n	Solution	Uniform Subdivision	Adaptive Subdivision	Proposed Subdivision
1	Underdamped	2	boxes	$360,000$	$27,594$	$27,594$
			time(s)	25	$1,274$	3.2
			flops	$36,283,082$	$8,687,967$	$5,957,375$
2	DC Motor	2	boxes	791	45	45
			time(s)	0.1	0.87	0.1
			flops	$73,613$	10161	8606
3	Simple poles	3	boxes	$1,311,025^*$	$7,712$	$7,712$
			time(s)	–	339	1
			flops	–	$2,860,983$	$2,298,795$
4	Non-minimum phase	3	boxes	$23,716$	512	512
			time(s)	2	18	0.25
			flops	$3,370,306$	$181,599$	$152,830$
5	Non-rational	3	boxes	$1,822,500^*$	$6,759$	$6,759$
			time(s)	–	376	3.0
			flops	–	$6,402,673$	$6,166,936$
6	Vehicle clutch	3	boxes	$1.2e12^*$	$30,424$	$30,424$
			time(s)	–	$1.97e3$	3.5
			flops	–	$14,053,919$	$10,284,745$
7	Multiple lags	4	boxes	$5.653e13^*$	$2,35,139$	$2,35,139$
			time(s)	–	$83,754$	850
			flops	–	$1.1541e9$	$1.1216e9$
8	Mechanical	5	boxes	$1,310,720$	$17,320$	$17,320$
			time(s)	102	934	2.5
			flops	$165,151,882$	$5,641,491$	$4,469,337$
9	Aircraft	5	boxes	$9.695e10^*$	$40,002$	$40,002$
			time(s)	–	$3.17e3$	9.5
			flops	–	$22,760,872$	$17,841,946$
10	Inverted pendulum	7	boxes	$1.1964e14^*$	$12,191$	$12,191$
			time(s)	–	738	3.1
			flops	–	$8,289,554$	$6,670,737$

solution in only four of the ten examples studied. Even in these four examples, parallel-adaptive subdivision generates much less number of solution boxes, in much less time and *flops* than uniform subdivision.

Summarizing the results of the numerical tests given in Table 1, we find parallel-adaptive subdivision to be clearly superior to the uniform and adaptive subdivision in every example. These findings strongly suggest the proposed subdivision strategy as a preferred one in other applications.

REFERENCES

[1] *MATLAB user guide, version 5.3.* The MathWorks Inc., MA, USA, 2000.

[2] *Forte Fortran 95 user manual.* Sun Microsystems, Palo Alto, CA, USA, 2001.
[3] J. Ackermann. *Robust control: systems with uncertain physical parameters.* Springer-Verlag, 1975.
[4] J. Ackermann and W. Sienel. On the computation of value sets for robust stability analysis. In *Proc. 1st European Control Conf.*, pages 1345–1350, Grenoble, France, 1991.
[5] G. Alefeld and J. Herzberger. *Introduction to interval computations.* Academic Press, New York, 1983.
[6] C. Borghesani, Y. Chait, and O. Yaniv. The quantitative feedback theory toolbox for MATLAB. 1995.
[7] W. Chen and D. J. Ballance. Plant template generation for uncertain plants in QFT. *Trans. of the ASME Journal of Dynamic Systems, Measurement and Control*, 121:359–364, 1999.
[8] I. M. Horowitz. *Quantitative feedback design theory (QFT).* QFT Publications, Boulder, Colorado, 1993.
[9] R. B. Kearfott. Some tests of generalized bisection. *ACM Transactions on Mathematical Software*, 13(3):197–220, 1987.
[10] R. E. Moore. *Interval analysis.* Prentice-Hall, Englewood Cliffs, New Jersey, 1966.
[11] R. E. Moore. *Methods and applications of interval analysis.* SIAM, Philadelphia, 1979.
[12] P. S. V. Nataraj and G. Sardar. Computation of QFT bounds for robust sensitivity and gain-phase margin specifications. *Trans. of the ASME Journal of Dynamic Systems, Measurement and Control*, 122:528–534, September 2000.
[13] P. S. V. Nataraj and G. Sardar. Template generation for continuous transfer functions using interval analysis. *Automatica*, 36:111–119, 2000.
[14] A. Neumaier. The enclosure of solutions of parameter dependent systems of equations. In R. E. Moore, editor, *Reliability in computing: the role of interval methods in scientific computations.* Academic Press, 1988.
[15] L. B. Rall. *Automatic differentiation, techniques and applications.* Number 120 in Lecture Notes in Computer Science. Springer-Verlag, Berlin, 1981.
[16] J. M. Rodrigues, Y. Chait, and C. V. Hollot. An efficient algorithm for computing QFT bounds. *Trans. of the ASME Journal of Dynamic Systems, Measurement and Control*, 119(3):548–552, 1997.
[17] S. M. Rump. INTLAB - interval laboratory. In T. Csendes, editor, *Developments in reliable computing.* Kluwer Academic Publishers, 1999.
[18] M. Sidi. Feedback synthesis with plant ignorance, nonminimum phase, and time-domain tolerances. *Automatica*, 12, 1976.
[19] D. F. Thomspon and O. D. I. Nwokah. Analytical loop shaping methods in quantitative feedback theory. *Trans. of the ASME Journal of Dynamic Systems, Measurement and Control*, 116:169–177, 1994.

Appendix A. List of Examples

Example 1 *Active noise and vibration control system* [16]: The magnitude and phase angle functions for a system occurring in active noise and vibration control with highly underdamped resonances are

$$\begin{aligned} f_1(x) &= -\frac{180}{\pi}\arctan\left\{\frac{2x_2\dfrac{\omega}{x_1}}{1-\left(\dfrac{\omega}{x_1}\right)^2}\right\} \\ f_2(x) &= -10\log_{10}\left\{\left(\left(\frac{\omega}{x_1}\right)^2+2x_2^2-1\right)^2+1-\left(2x_2^2-1\right)^2\right\} \\ x_1 &\in [0.75, 1.25],\ x_2\in[0.02, 0.06] \end{aligned}$$

The frequency is $\omega = 1$.

Example 2 *DC Motor* [8]: The magnitude and phase angle functions for a DC Motor are

$$\begin{aligned} f_1(x) &= -\frac{180}{\pi}\left(\frac{\pi}{2}+\arctan\left(\frac{\omega}{x_1}\right)\right) \\ f_2(x) &= 20\log_{10}\left\{\frac{x_2}{\omega\sqrt{(\omega^2+x_1^2)}}\right\} \\ x_1 &\in [1,4], x_2\in[1,10] \end{aligned}$$

The frequency is $\omega = 1$.

Example 3 *Simple poles* [12]: The magnitude and phase angle functions for a stable second order system with real poles are

$$
\begin{aligned}
f_1(x) &= -\frac{180}{\pi}\left\{\arctan\left(\frac{\omega}{x_1}\right)+\arctan\left(\frac{\omega}{x_2}\right)\right\}\\
f_2(x) &= -10\log_{10}\left\{(x_1^2+x_2^2+\omega^2)\omega^2+(x_1x_2)^2\right\}+20\ \log_{10}(x_3)\\
x_1 &\in [1,5],\ x_2\in[20,30],x_3\in[1,10]
\end{aligned}
$$

The frequency is $\omega = 1$.

Example 4 *Non-minimum phase* [18]: The magnitude and phase angle functions for a non-minimum phase (NMP) system with real poles and zeros are

$$
\begin{aligned}
f_1(x) &= \frac{180}{\pi}\left(\arctan(-\omega x_2)-\arctan(\omega x_1)-\frac{\pi}{2}\right)\\
f_2(x) &= 10\log_{10}\left\{\frac{1+(x_2\omega)^2}{1+(x_1\omega)^2}\right\}+20\log_{10}\left(\frac{x_3}{\omega}\right)\\
x_1 &\in [0.3,1],\ x_2\in[0.05,0.1],\ x_3\in[1,3]
\end{aligned}
$$

The frequency is $\omega = 1$.

Example 5 *Non-rational* [8, pp.129]: The magnitude and phase angle functions for a non-rational system are

$$
\begin{aligned}
f_1(x) &= -\frac{180}{\pi}\left(\arctan\left\{\frac{-x_2\sin(x_1\omega)}{x_2\cos(x_1\omega)+1}\right\}+\omega x_3\right)\\
f_2(x) &= -10\log_{10}\left\{1+x_2\left(x_2+2\cos(x_1\omega)\right)\right\}\\
x_1 &\in [1,2],\ x_2\in[0.4,0.6],\ x_3\in[0.01,0.02]
\end{aligned}
$$

The frequency is $\omega = 2$.

Example 6 *Vehicle clutch system* [7]: The magnitude and phase angle functions between the input clutch position to the output transmission speed of a vehicle clutch system are

$$
\begin{aligned}
f_1(x) &= \frac{180}{\pi}\left(\arctan\left(\frac{24734.97\omega}{65.61(x_1-x_2\omega^2)}\right)-\arctan\left\{\frac{3.07}{-1157.39\omega}\left(x_1-\frac{\omega^2}{\left(\frac{1}{65.61}+\frac{1}{x_2}\right)}\right)\right\}\right)\\
f_2(x) &= 10\log_{10}\left\{\frac{\left(65.61\left(x_1-x_2\omega^2\right)\right)^2+(24734.97\omega)^2}{(-1157.39\omega)^2+\left(3.07\left(x_1-\frac{\omega^2}{\left(\frac{1}{65.61}+\frac{1}{x_2}\right)}\right)\right)^2}\right\}+20\ \log_{10}\left(\frac{x_3}{(65.61+x_2)\,\omega}\right)\\
x_1 &\in [5800,115000],x_2\in[1400,11000],x_3\in[100,800]
\end{aligned}
$$

The frequency is $\omega = 10$.

Example 7 *Multiple transport lags* [13]: The magnitude and phase angle functions for a system with multiple transport lags are

$$
\begin{aligned}
f_1(x) &= \frac{180}{\pi}\left(\arctan(\frac{n_i}{n_r}) - \arctan\left\{\frac{\log_{10}(x_1)\,\omega\cos(x_4)+\sin(\omega x_3)}{\log_{10}(x_1)-\cos(\omega x_3)}\right\} - \pi/2\right) \\
f_2(x) &= 10\log_{10}\left(\frac{n_r^2+n_i^2}{\left(\log_{10}(x_1)-\cos(\omega x_3)\right)^2+\left(\omega\log_{10}(x_1)\cos(x_4)+\sin(\omega x_3)\right)^2}\right) - \\
&\quad 20\log_{10}\left(\omega x_2\left(1-\left(\frac{\omega x_2}{4\pi}\right)^2\right)\right) \\
x_1 &\in [3,5],\ x_2\in[0.5,0.7],\ x_3\in[9.25,9.35], x_4\in[0.49,0.5]
\end{aligned}
$$

where n_r and n_i are defined as

$$
\begin{aligned}
n_r &= \left(1-\cos(\omega x_2)\right)\left(\log_{10}(x_1)-\cos(\omega x_3)\right)-\sin(\omega x_2)\left(\omega\log_{10}(x_1)\cos(x_4)+\sin(\omega x_3)\right)+ \\
&\quad \omega x_2(x_3+x_4)\left(1-\left(\frac{\omega x_2}{4\pi}\right)^2\right)\left(\sin(\omega x_1)-\omega\cos(x_4)\cos(\omega x_1)\right) \\
n_i &= \omega x_2(x_3+x_4)\left(1-\left(\frac{\omega x_2}{4\pi}\right)^2\right)\left(\omega\sin(\omega x_1)\cos(x_4)+\cos(\omega x_1)\right)+ \\
&\quad \left(1-\cos(\omega x_2)\right)\left(\omega\log_{10}(x_1)\cos(x_4)+\sin(\omega x_3)\right)+\sin(\omega x_2)\left(\log_{10}(x_1)-\cos(\omega x_3)\right)
\end{aligned}
$$

The frequency is $\omega = 0.5$.

Example 8 *Mechanical system* [8, pp. 222]*:* The magnitude and phase angle functions for a mechanical system are

$$
\begin{aligned}
f_1(x) &= -\frac{180}{\pi}\left(\arctan\left\{\frac{x_3}{\left(\frac{x_4}{\omega x_2}-x_1\omega x_2\right)}\right\}+\frac{\pi}{2}\right) \\
f_2(x) &= -20\ \log_{10}\left\{\omega^2\sqrt{x_3^2+\omega x_2\left(\frac{x_4}{(\omega x_2)^2}-x_1\right)^2}\right\}+20\ \log_{10}(x_5) \\
x_1 &\in [1,2],\ x_2\in[1,\sqrt{10}],\ x_3\in[0.5,1],\ x_4\in[2,3],\ x_5\in[0.5,2]
\end{aligned}
$$

The frequency is $\omega = 8$.

Example 9 *Aircraft, longitudinal Motion* [19]: The magnitude and phase angle functions for the longitudinal motion of an aircraft are

$$
\begin{aligned}
f_1(x) &= \frac{180}{\pi}\left(\arctan\left(\frac{\omega}{x_1}\right)-\left\{\arctan\left(\frac{\omega}{x_2}\right)+\frac{\pi}{2}+\arctan\left(\frac{2x_3\frac{\omega}{x_4}}{1-\left(\frac{\omega}{x_4}\right)^2}\right)\right\}\right) \\
f_2(x) &= 10\ \log_{10}\left\{\frac{1+\left(\frac{\omega}{x_1}\right)^2}{\omega^2\left(1+\left(\frac{\omega}{x_2}\right)^2\right)\left(\left(1-\left(\frac{\omega}{x_4}\right)^2\right)^2+\left(2x_3\frac{\omega}{x_4}\right)^2\right)}\right\}+20\ \log_{10}(x_5) \\
x_1 &\in [0.5,0.75],\ x_2\in[1,10],\ x_3\in[0.8,0.9],\ x_4\in[5,6], x_5\in[0.2,2]
\end{aligned}
$$

The frequency is $\omega = 0.1$.

Example 10 *Inverted pendulum* [6]: The magnitude and phase angle functions between pendulum angle to the cart's motor current are

$$\begin{aligned}
f_1(x) &= -\frac{180}{\pi}\left(\arctan\left\{\frac{\omega}{-\left(x_2 x_1 + \frac{\omega^2}{x_4}\right)}\right\} + \arctan\left\{\frac{2x_3\frac{\omega}{x_5}}{1-\frac{\omega}{x_5^2}}\right\} + \omega x_7\right) \\
f_2(x) &= 20\ \log_{10}\left\{\frac{\omega^2}{x_6\omega^2 + 9.81}\frac{x_1}{\sqrt{\omega^2 + \left(x_2 x_1 + \frac{\omega^2}{x_4}\right)^2}}\frac{1}{\sqrt{\left(\frac{\omega}{x_5^2} + 2x_3^2 - 1\right)^2 + 1 - (2x_3^2 - 1)^2}}\right\} \\
x_1 &\in [1.5, 1.7], x_2 \in [0.05, 0.15], x_3 \in [0.01, 0.02] \\
x_4 &\in [15, 17], x_5 \in [50, 60], x_6 \in [0.3, 0.45], x_7 \in [0.014, 0.015]
\end{aligned}$$

The frequency is $\omega = 10$.