INTERVAL MATHEMATICS AND INTERVAL CONSTRAINT PROGRAMMING
ICCM-2004 WORKSHOPS P. 285–288

Example of Babuška, Práger and Vitásek in interval computations

Gregory G. Men'shikov*

Abstract. Taking the well-known example by Babuška, Práger and Vitásek ("BPV example"), we show that the interval computations themselves (even their validating version) do not guarantee high quality of the computational work yet, and the stability of the "core" algorithm plays a crucial role. If such an algorithm is not suitable, then the whole procedure is fallable.

As for "BPV example", we, first, use it to show troubles emerging in traditional computations apart from the absence of the guaranteed information about the error. Then, we apply validated interval computations to the "BPV example", but the accuracy of results still remains low.

Trying to procede in the interval manner without taking any care of the validation, shows that the quality of the results is not improved. On the other hand, we may lose the information about the accuracy, the latter may prove really interesting in the results.

Finally we make an decisive break by using the intersection of separate enclosures of the result under computation or a family of such results. At the same time, we can see how the validation achieved by analytical menas may make the results of calculations more precise.

1. The point execution of the example

Let us consider the example of recurrent computation

$$I_n = \frac{1}{e}\int_0^1 x^n e^x\,dx, \qquad n = 0,\ 1, \ldots \tag{1}$$

from the book [1]. It is clear that $I_0 = 1 - e^{-1}$. To express I_n from I_{n-1} [2], we, integrating by parts, obtain the following recursive equality:

$$I_n = 1 - nI_{n-1}. \tag{2}$$

Computing $I_1, I_2, \ldots$ in accordance to the above formula on a real computer (when the computation are subject to rounding, etc.) produces after $n \approx 10 \div 15$ the result which is evedently wrong (negative), since the values I_n must be nonnegative. The analysis of of the phenomenon was fulfilled in the book [2], and its main reason turns out the subtraction of neighbouring compute values.

2. The interval (validated) execution of the example

Let us rerun the same example, in interval-validating manner at this time. Making use of the Second Theorem on compositions [2], we rewrite the right part of the recursive relation in enclosures, which leads to the inclusion:

*St. Petersburg State University, miksha@pobox.spbu.ru.

$$I_n \in 1 - n[I_{n-1}], \qquad n = 0, 1, \dots \tag{3}$$

In the above, $[I_{n-1}]$ means an interval enclosure for I_{n-1} found by computer.

Hence, it is possible to use the right-hand part as $[I_n]$. Then, taking in account the majorization [1] (i. e., the auxiliary extention of the enclosure), we obtain the futher modification of the relation (3):

$$[I_n] = 1 - n[I_{n-1}], \qquad n = 0, 1, \dots \tag{4}$$

Its execution has been performed an interval assembler described in [2, § 17–18] while the corresponding code of the algorithm is presented in the Table 212.1 from the book [3].

The results of several steps of computations are in columns 1, 2, 3 of Table 1.

If n grows, the width of the enclosure increases even in an accelerating manner. Finally, at $n = 9$ it may be stated that further computation is senseless since drastical decreasing of the accuracy: the ratio of the width $[I_n]$ (denoted as $w(I_n)$) to I_n has the value near 0.1. Due to this, the value of $w([I_{15}])$ is large.

Table 1

n	$[I_n]$	$w([I_n])$	$[I_n]_{\text{int modif}}$	$w([I_n])_{\text{int modif}}$
0	.6321203 .6321208	4.77E−07	.6321203 .6321208	4.77E−07
1	.3678789 .36788	1.01E−06	.3678789 .36788	1.01E−06
2	.2642398 .2642424	2.56E−06 .2642424	.2642398	2.56E−06
…	…	…	…	…
9	−.1697655 .3596304	5.29E−01	9.090908E−02 .1	9.09E−03
…	…	…	…	…
15	−941 903 965 832.4	1.91E+06	5.822351E−02 6.250001E−02	3.62E−03

Thus, interval-validated computation allows one to check the accuracy permanently. This property is absent in the usual, traditional computations, but the interval validated result may have abnormally large width, the enclosure being thus of low practical value.

3. The interval (non-validated) execution of the example

Table 2

n	$[I_n]$
0	.6321205
1	.3678795
2	.2642411
…	…
14	−797.5973
15	−11964.96

It may appear that the successful interval execution of the example does not give an essential narrowing of the width of the enclosure. This execution can also have a low accuracy. The results of interval execution of the same example are shown in Table 2. As before, the same program is used from Table 1. To make the results non-validated ones, we remove a majorization procedure from the computation. Specifically, we set the constant C of the majorization equal to zero [2]. Therefore, we obtain degenerated intervals. Naturally, the same effect (in a quality but without fail in a quantity) is produced by the point execution.

4. The interval-validated modification of the example

By interval analogue of the recursive formula (4), we have computed enclosures and confirmed computational instability of the process.

On the other hand, theoretical reasoning in the book [2] proves the boundness of $\{I_n\}$: the inequality is obtained:

$$\frac{1}{n+1} < I_{n-1} < \frac{1}{n}. \tag{5}$$

The corresponding close intervals $\left[\dfrac{1}{n+2}, \frac{1}{n+1}\right]$ present one more system of enclosures. Using them, it is possible to stop cathastrophic raise of the width. Indeed, the non-empty intersection of the interval enclosures is the interval enclosure too. Thanks to this fact, we denote

$$[I_n] = \big(1 - n[I_{n-1}]\big) \cap \left[\frac{1}{n+2},\ \frac{1}{n+1}\right] \tag{6}$$

instead of (3a).

It seems (in the ideal model of interval computing [3]),

$$w([I_n]) \leq w\left(\left[\frac{1}{n+2}, \frac{1}{n+1}\right]\right) = \frac{1}{n+1} - \frac{1}{n+2} = \frac{1}{(n+1)(n+2)}. \tag{7}$$

Hence, we arrive at $w(I_n) \to 0$.

Moreover, the relative width is infinitely small too:

$$\frac{w([I_n])}{|I_n|} \leq \frac{w([I_n])}{(n+1)} \leq \frac{1}{(n+2)} \to 0. \tag{8}$$

Making use of the modified interval algorithmics, let us carry out one more series of numerical experiments. Their program is formed on the basis of the above-mentioned one. Also, we run the computations for the same n as in the preceding section.

The results are contained in the columns 1, 4, 5 of Table 1, and this time the width does not increase. Moreover, it tends to relatively small values.

Thus, modifying the interval approach enables us not only to inquire into the accuracy issues, but to control the accuracy as well.

Probably, it is possible to obtain even more precise two-sided inequality for I_n. The relations of this type are suitable as a basis for further modifications of the algorithm for finding I_n.

5. The BPV example as the diverging iterative process

Sometimes, it may prove that checking out has confirmed that the point is of bad quality of the outer algorithm by point or interval execution, either validated or not.

Let us consider the BPV example as an iterative process, which correlates to the subjects of the book [1]. Here, a divergence takes place. Note that from the numerical stability viewpoint the other way is impossible.

Conclusion

Our text is primarily intended for those who are interested in the scientific computations, specifically, in their accuracy, and who teaches the interval computation in high schools.

Acknowledgements. I am very grateful to Dr. S. Shary (Novosibirsk) for his attention to this manuscript. I am also very grateful to my daughter Mrs. O. Men'shikova (St. Petersburg) for her efforts and helpful comments that facilitated the improvement of the manuscript.

References

[1] Babuška I., Práger M., Vitásek E. Numerical Processes in Differential Equations. – Praha, Czechoslovakia: SNTL-publishers of Technical Literature, 1966. – 368 p.

[2] Men'shikov G.G. Localizing Computing: Lectures Summary. Issue 1. Introduction to Interval-localizing Organization of Computations. – St. Petersburg: Department of Operative Poligraphy of Scientifical Research Institute of Chemistry of SPb State University, 2003. – 89 p. (in Russian).

[3] Men'shikov G.G. Localizing Computing: Lectures Summary. Issue 2. Tasks of Compositional Computation and Problem of Rouphness their Interval-localizing Solution. – St. Petersburg: Department of Operative Poligraphy of Scientifical Research Institute of Chemistry of SPb State University, 2003. – 59 p. (in Russian).