Journal of Universal Computer Science, vol. 4, no. 6 (1998), 561-573
submitted: 25/9/97, accepted: 22/4/98, appeared: 28/6/98 © Springer Pub. Co.

Perturbation simulations of rounding errors in the evaluation of Chebyshev series

Roberto Barrio
(GME, Dep. Matemática Aplicada, Centro Politécnico Superior
Universidad de Zaragoza, E–50015 Zaragoza, Spain
rabarrio@posta.unizar.es)

Jean–Claude Berges
(Dep. DGA/T/TI/MS/MN, CNES, 18 Avenue Edouard Belin
F–31401 Toulouse Cedex 4, France
jean-claude.berges@cst.cnes.fr)

Abstract: This paper presents some numerical simulations of rounding errors produced during evaluation of Chebyshev series. The simulations are based on perturbation theory and use recent software called AQUARELS. They give more precise results than the theoretical bounds (the difference is of some orders of magnitude). The paper concludes by confirming theoretical results on the increment of the error at the end of the interval $[-1, 1]$ and the increased performance achieved by some modifications to Clenshaw's algorithm near those points.

Key Words: Rounding errors, perturbation methods, Chebyshev polynomials, polynomial evaluation

Category: G.1.0, F.2.1

1 Introduction

Today, numerical simulation is the most widely used method for evaluating physical systems. In order to obtain descriptive results of a physical phenomenon, a scientific software program must implement a mathematical representation of an actual problem using numerical resolution methods. To validate the simulation results, one has to evaluate the propagation of roundoff errors dependent on computer arithmetic units, as well as the impact of erroneous input data on the final result. This process guarantees that results can be trusted.

Chebyshev polynomial series appear in several fields of scientific research and mathematics, as, for instance, in the resolution of partial differential equations by means of the Chebyshev collocation method [Canuto *et al.* 1988], in the approximation of functions [Schonfelder 1978] and in the compression of satellite ephemerides [Coffey *et al.* 1996].

Let $p_n(x)$ be one of such series with a finite number of terms, i.e.,

$$p_n(x) = \sum_{i=0}^{n} c_i \, T_i(x), \quad \text{with} \quad x \in [-1, 1], \tag{1}$$

where $T_i(x)$ are the Chebyshev polynomials of the first kind. A typical operation is the evaluation of (1) at one point x for which several algorithms have been proposed [Bakhvalov 1971, Clenshaw 1955, Forsythe 1957, Gentleman 1969].

The numerical stability of those algorithms has been studied [Bakhvalov 1971, Oliver 1979], giving some theoretical bounds of the rounding errors. Following these studies Clenshaw's algorithm is recommended except for x near -1 and 1, where the Reinsch modifications to Clenshaw's algorithm [Gentleman 1969] are preferred.

These theoretical bounds are very conservative, as they are of several orders of magnitude bigger than the real errors, and consequently, of limited use in practical situations. For this reason, we chose to study "realistic" rounding errors for some Chebyshev series by using each of the proposed algorithms in turn in order to confirm the theoretical analysis (for more details see [Barrio and Berges 1997]). These analyses are of practical interest due to the need to choose one or several algorithms for accurate evaluation of any specific series.

2 Perturbation analysis

We thus used modern software called AQUARELS [Berges 1995, Erhel *et al.* 1991, Francois 1989] to do perturbation analysis in order to obtain estimations of the rounding errors. This program provides a problem solving environment for controlling and improving the numerical quality of scientific software. The simulation is based on random perturbations of the last bit of the result in each elementary operation. This kind of methods has proven its usefulness in scientific computing [Chesneaux 1994].

The algorithms analyzed are those of: *Forsythe*, *Clenshaw*, and *Bakhvalov* and also the Reinsch modifications near $x = -1$ (*Reinsch*(-1)) and near $x = 1$ (*Reinsch*(1)). In the tests, four kinds of Chebyshev series were used; the first (**P1**) with 6000 random coefficients normally distributed with mean 0 and variance 1, the second (**P2**) with monotonically decreasing coefficients, $c_i = 1/i^2$ ($i = 1, \ldots, 6000$), and the third (**P3**) and the fourth (**P4**) consisted of the 3000 first coefficients of the development of two functions in Chebyshev polynomials:

$$f_3(x) = \frac{\sin 8\,(x+1) + \sin 400\,(x+1)}{(x+1.1)^{3/2}}, \quad f_4(x) = \frac{\sin 8\,(x+1) + \sin 2000\,(x+1)}{(x+1.1)^{3/2}}.$$

2.1 Theoretical bounds

The theoretical bounds [Bakhvalov 1971, Oliver 1979] of the rounding errors are usually given up to the first order on the unit roundoff (u) of the computer used, i. e.,

$$|\widetilde{p}_n(x) - p(x)| < k \cdot u + \mathcal{O}(u^2),$$

where $\widetilde{p}_n(x)$ is the calculated value of $p_n(x)$.

Figure 1 shows the theoretical error bounds (factor k). The left-hand side shows the bounds for $x \in (-1+\varepsilon, 1-\varepsilon)$, whereas the right-hand side shows the bounds for $x \in [-1, -1+\varepsilon)$ (the bounds near $x = 1$ are symmetric, and changing *Reinsch*(-1) by *Reinsch*(1)). As a result, we can hypothesize that random problems are more difficult to evaluate due to the similar size of all the coefficients, which gives very pessimistic bounds. For monotonically decreasing coefficients (problem **P2**) the bound of the errors is small. For "real" series (coming from approximations of functions) the bounds increase with the complexity of the function. Also, it may be observed that, *a priori*, Reinsch's algorithm adapted for values near $x = -1$ works better in that interval, while the performance is worse near $x = 1$. Something similar occurs with the other algorithm of Reinsch for the case $x = 1$. At these values ($\simeq -1, \simeq 1$) the theoretical bounds increase by several orders of magnitude, being in some cases of an order of 10^{10}. In the middle range, Clenshaw's algorithm is preferred. The bounds of Bakhvalov's algorithm show a lower dependence on the problem but as the bound involves an unknown constant, it is not possible to make any hypotheses.

Taking into account the figures of the theoretical error bounds leads to the same recommendations as Oliver, i. e. using the algorithm of *Reinsch*(-1) for $x \in [-1, -0.6)$, Clenshaw for $x \in [-0.6, 0.6]$ and *Reinsch*(1) for $x \in (-0.6, 1]$ (Oliver only uses very small series in his simulations). Cox (1993) recommends the intervals $[-1, -0.5)$, $[-0.5, 0.5]$ and $(0.5, 1]$, respectively (the subroutines `E02AEF` and `E02AKF` of NAG (1993) use these intervals, whereas the subroutine `C06DBF` uses only Clenshaw's algorithm).

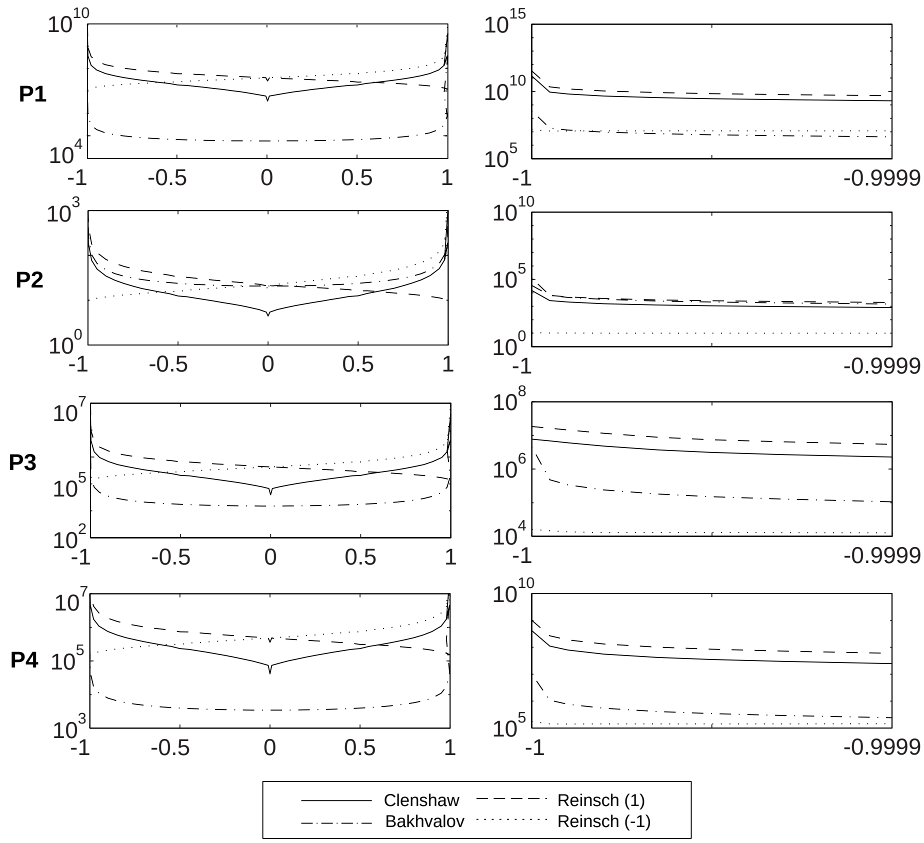

Figure 1: Theoretical error bounds.

2.2 Numerical tests

The previous sub-section dealt with the theoretical bounds, but the question remains as to what happens in a real problem? How accurate are the theoretical bounds?

In some algorithms the theoretical bounds are quite accurate and yield good estimations of the rounding errors, but in others there are some "pathological" problems with a bound of several orders of magnitude bigger than average. Therefore, in such algorithms the theoretical bound would be of little use because it accounts for the worst-case propagation of errors.

Due to the big size of the theoretical bounds of some algorithms, specially near $x = -1$ and $x = 1$, it appeared to be worthwhile to do some numerical analysis. As mentioned above, AQUARELS software with double precision was used. AQUARELS applies a perturbation method in order to estimate "realistic" error bounds for some problems of interest.

For each evaluation algorithm, each series, and each point of evaluation $x \in [-1, 1]$, 100 simulations were performed, which was sufficient to obtain statistical results of the rounding errors. First, by means of Shapiro and Wilk's normality test [Royston 1982], the distribution of errors was checked to see whether it followed a normal law. Table 1 shows the results of the test for some simulations for the **P1** problem using the Forsythe algorithm. It may be assumed that, in general, the normal law can be used except for $x = 0.6$, where the hypothesis may be rejected.

Table 1: Results of the normality test of Shapiro and Wilk. The significance level is $\alpha = 0.05$. The values of the statistic W of the simulations are bigger than the corresponding tabulated values of W for $\alpha = 0.05$ and $N = 100$ (the p–value $\geq \alpha$) except for the case $x = 0.6$.

x	$-1.$	-0.8	-0.6	-0.4	-0.2	-0.003	$0.$	0.003	0.2	0.4	0.6^*	0.8	$1.$
W	0.98	0.986	0.975	0.989	0.989	0.972	0.98	0.975	0.986	0.971	0.962^*	0.984	0.984
p	0.52	0.82	0.29	0.93	0.94	0.20	0.53	0.29	0.86	0.18	0.03^*	0.73	0.77

Let $x_i \in [-1, 1]$ and let $\{y_{ij} \,|\, 1 \leq j \leq N\}$ be the N simulations (here $N = 100$) of the evaluation of the series at the point x_i. The following notation will be used:

$$\overline{y}_i = \tfrac{1}{N} \sum_{j=1}^{N} y_{ij},$$

$$r_{ij}^A = y_{ij} - \overline{y}_i, \qquad r_{ij}^R = \frac{y_{ij} - \overline{y}_i}{|\overline{y}_i|},$$

$$l_{ij}^A = -\log_{10}(|r_{ij}^A|), \qquad l_{ij}^R = -\log_{10}(|r_{ij}^R|).$$

In order to represent all the perturbation tests we used what we have called the *logarithmic polar representation*. In 2D, this representation of the absolute rounding errors consists of representing on the plane

$$\begin{cases} P_x = l_{ij}^A \cos\theta_j, \\ P_y = l_{ij}^A \sin\theta_j. \end{cases} \qquad \text{with} \quad \theta_j = (j-1)\,\frac{\pi}{N},$$

For the relative rounding errors we used l_{ij}^R instead of l_{ij}^A. In 3D, the representation is similar, but with x as the third coordinate. Moreover, the data r_{ij} was sorted from bigger to smaller $\{r_{i1} \geq r_{i2} \geq \ldots \geq r_{iN}\}$ in order to smooth the curve. Also, with the re-ordering, the negative errors are plotted on the lower part of the graph, while the positive are on the upper part. These representations of the rounding errors enabled us to better visualize their behavior.

Figures 2, 3, 4 and 5 show the *logarithmic polar representation* of the simulation errors and Figures 6, 7, 8 and 9 show the standard deviation, in a logarithmic scale. The increment at the ends of the interval appears in every algorithm and problem, but without the symmetry of the theoretical bounds. In each problem (except **P2**) the worst performance is presented for the Bakhvalov algorithm.

For the **P1** problem the behavior of the algorithms is quite similar, except for the Bakhvalov one. In general, the best performance was achieved using the Clenshaw algorithm, but near the ends, a slight improvement was found for the Reinsch algorithms. For **P2** the Forsythe algorithm worked "badly", due to the particular behavior of the coefficients, monotonically decreasing in size. Thus, it is better to begin the evaluation from the last coefficients (the Forsythe algorithm is the only one that begins from the first ones). The "good" behavior of this analysis is shown in 2D in the joined curves and in 3D in the almost cylindrical figure. On **P3** and **P4** we observed that the rounding errors increase with the complexity of the function to be approximated.

It has been demonstrated that the theoretical bounds are some orders of magnitude higher than the simulations. Besides, for the Bakhvalov algorithm these bounds are of little use, due to the unknown coefficient in the theoretical bound. Therefore, our recommendations on the choice of one algorithm, taking into account only the rounding errors, are similar to those resulting from the theoretical analysis, i. e.: in general, the Clenshaw algorithm should be used and near the ends the Reinsch algorithms. Table 2 shows the "best" algorithm depending on the interval of the variable x.

Table 2: "Recommended" algorithms.

Problem	*Reinsch*(-1)	*Clenshaw*	*Reinsch*(1)
P1	$[-1, -0.6)$	$[-0.6, 0.6]$	$(0.6, 1]$
P2	$[-1, -1+\varepsilon)$	$[-1+\varepsilon, 0]$	$(0, 1]$
P3	$[-1, -0.6)$	$[-0.6, 0.7]$	$(0.7, 1]$
P4	$[-1, -0.7)$	$[-0.7, 0.7]$	$(0.7, 1]$

3 Conclusions

The numerical tests confirmed the theoretical results on the increment of the error at the ends of the interval $[-1, 1]$ and the better performance of some modifications of the Clenshaw algorithm at those points.

It should be pointed out that, as the differences on the rounding errors observed in the simulations for the different algorithms are in general very small, it is not worth using other algorithms than that of Clenshaw due to its good performance in the tests and its cheaper evaluation cost.

It may be concluded that it is worthwhile controlling rounding errors via perturbation analysis as a complement to theoretical studies of the error bounds since this control serves as a numerical validation and analysis of the theoretical bounds. Also, it can be useful in a preliminary study of the stability of new algorithms, when no bounds are known or when the determination of bounds is a difficult task.

4 Acknowledgments

The first author has received partial support from both the *Département de Mathématiques Spatiales* at the *Centre National d'Etudes Spatiales* (Toulouse, France) and from project PB95–0807 of the *Ministerio de Educación y Ciencia* (Spain).

References

[Bakhvalov 1971] Bakhvalov, N. S: "The stable calculation of polynomial values"; J. Comp. Math. and Math. Phys., 11 (1971), 1568–1574.
[Barrio and Berges 1997] Barrio, R. and Berges, J. C.: "Analysis of Rounding Errors in the Evaluation of Chebyshev Series"; CNES Report DGA/T/TI/MS/MN 97-222, Toulouse, France (1997).
[Berges 1995] Berges, J. C.: "Improvement of the accuracy of Space Dynamics Software: The Toolbox AQUARELS"; proc. SCAN'95 (1995).
[Canuto *et al.* 1988] Canuto, C., Quarteroni, A., Hussaini, M.Y. and Zang, T.: "Spectral Methods in Fluid Mechanics"; Ed. Springer-Verlag, New York (1988).
[Chesneaux 1994] Chesneaux, J. M.: "The equality relations in scientific computing"; Numerical Algorithms, 7 (1994), 129–143.
[Clenshaw 1955] Clenshaw, C. W.: "A note on the summation of Chebyshev series"; MTAC, 9 (1955), 118–120.
[Coffey *et al.* 1996] Coffey, S., Kelm, B. and B. Eckstein: "Compression of Satellite Orbits"; AAS 96–125 (1996).
[Cox 1993] Cox, M.G.: "Reliable determination of interpolating polynomials", Numerical Algorithms, 5 (1993), 113–154.
[Erhel *et al.* 1991] Erhel, J. and Philippe, B.: "Design of a toolbox to control arithmetic reliability"; proc. SCAN'91 (1991).
[Forsythe 1957] Forsythe, G. E.: "Generation and use of orthogonal polynomials for data fitting with a digital computer"; J. SIAM, 5 (1957), 74–88.
[Francois 1989] Francois, P.: "Contribution à l'étude de méthodes de contrôle automatique de l'erreur d'arroundi: la methodologie SCALP"; Ph.D. Dissertation (in French), University of Grenoble, France (1989).

[Gentleman 1969] Gentleman, W. M.: "An error analysis of Goertzel's (Watt's) method for computing Fourier coefficients"; Comput. J., 12 (1969), 160–165.

[NAG 1993] NAG: "NAG Fortran Library Manual, Mark 16"; Volume 1 (1993).

[Oliver 1979] Oliver, J.: "Rounding error propagation in polynomial evaluation schemes"; Journal of Computation and Applied Mathematics, 5 (1979), 85–97.

[Royston 1982] Royston, J. P.: "An extension of Shapiro and Wilk's W test for normality to large samples"; Appl. Statist., 31 (1982), 115–124.

[Schonfelder 1978] Schonfelder, J. L.: "Chebyshev expansions for the error and related functions"; Maths. Comput., 32 (1978), 1232–1240.

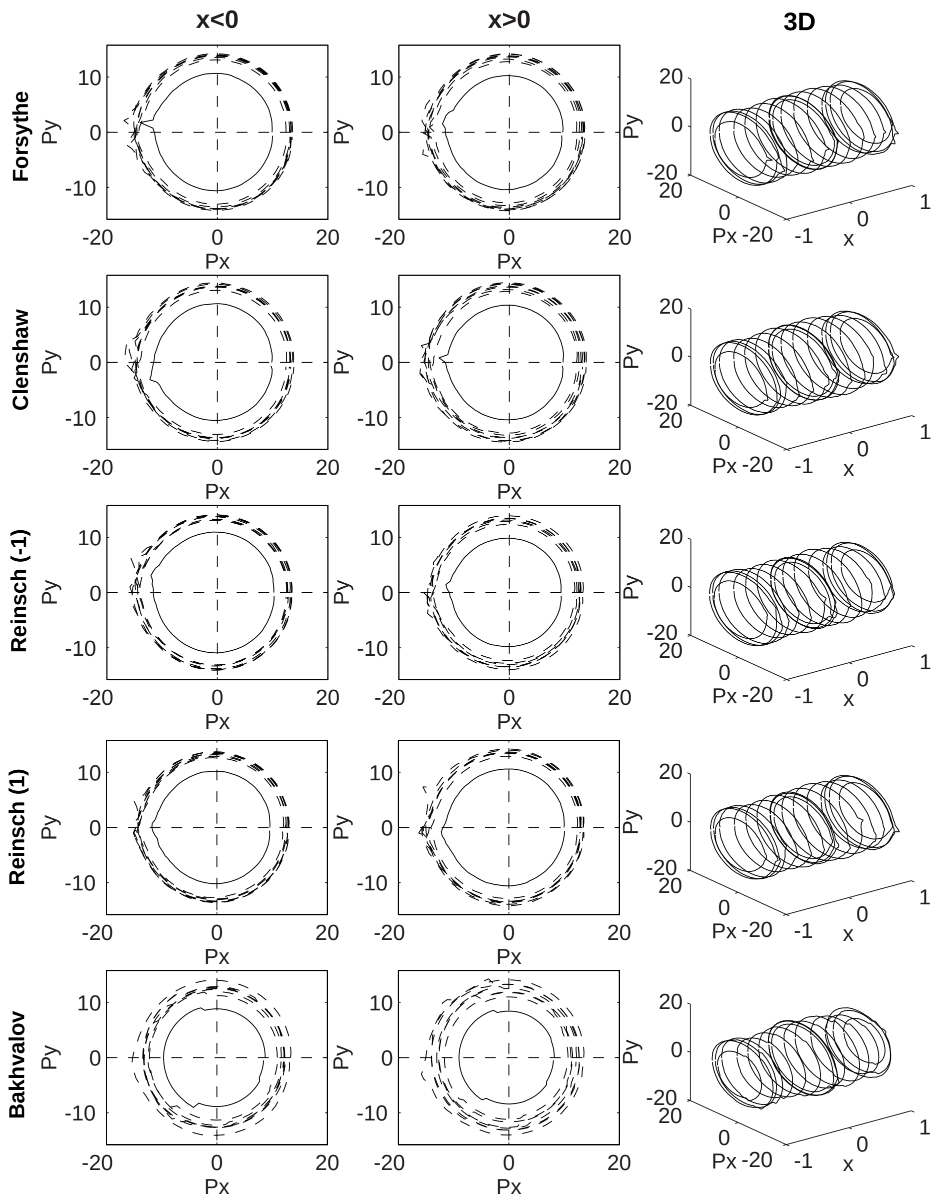

Figure 2: Logarithmic polar representation of the error for the problem **P1**.

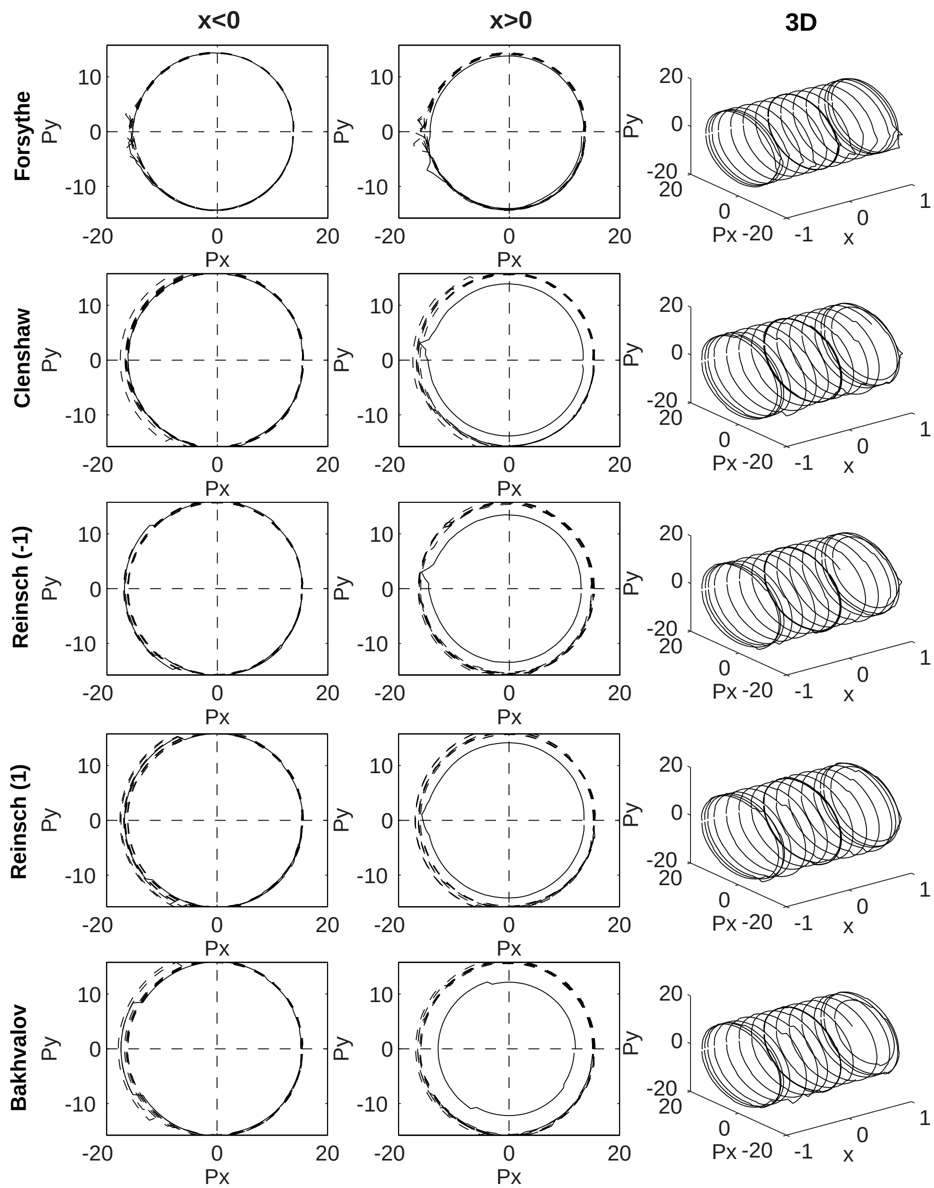

Figure 3: Logarithmic polar representation of the error for the problem **P2**.

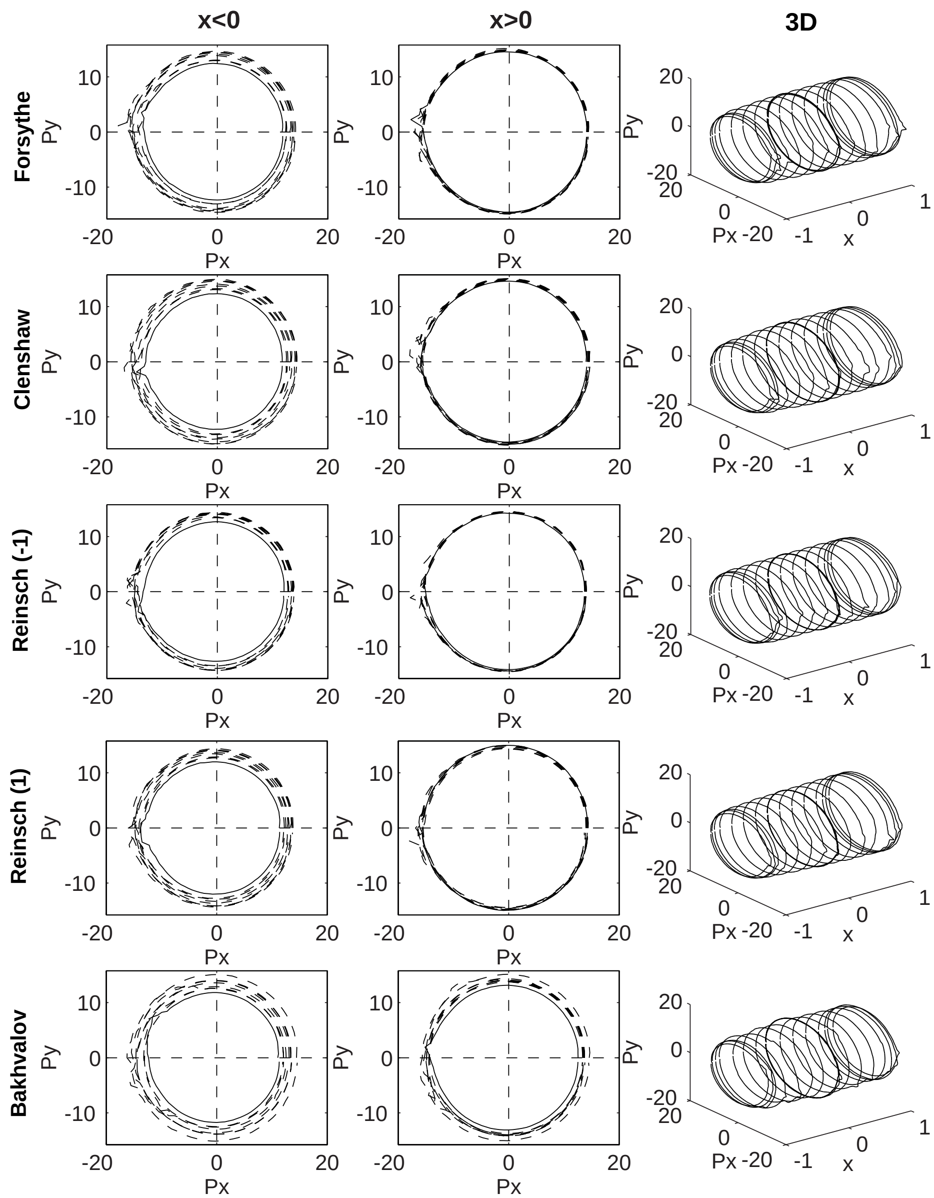

Figure 4: Logarithmic polar representation of the error for the problem **P3**.

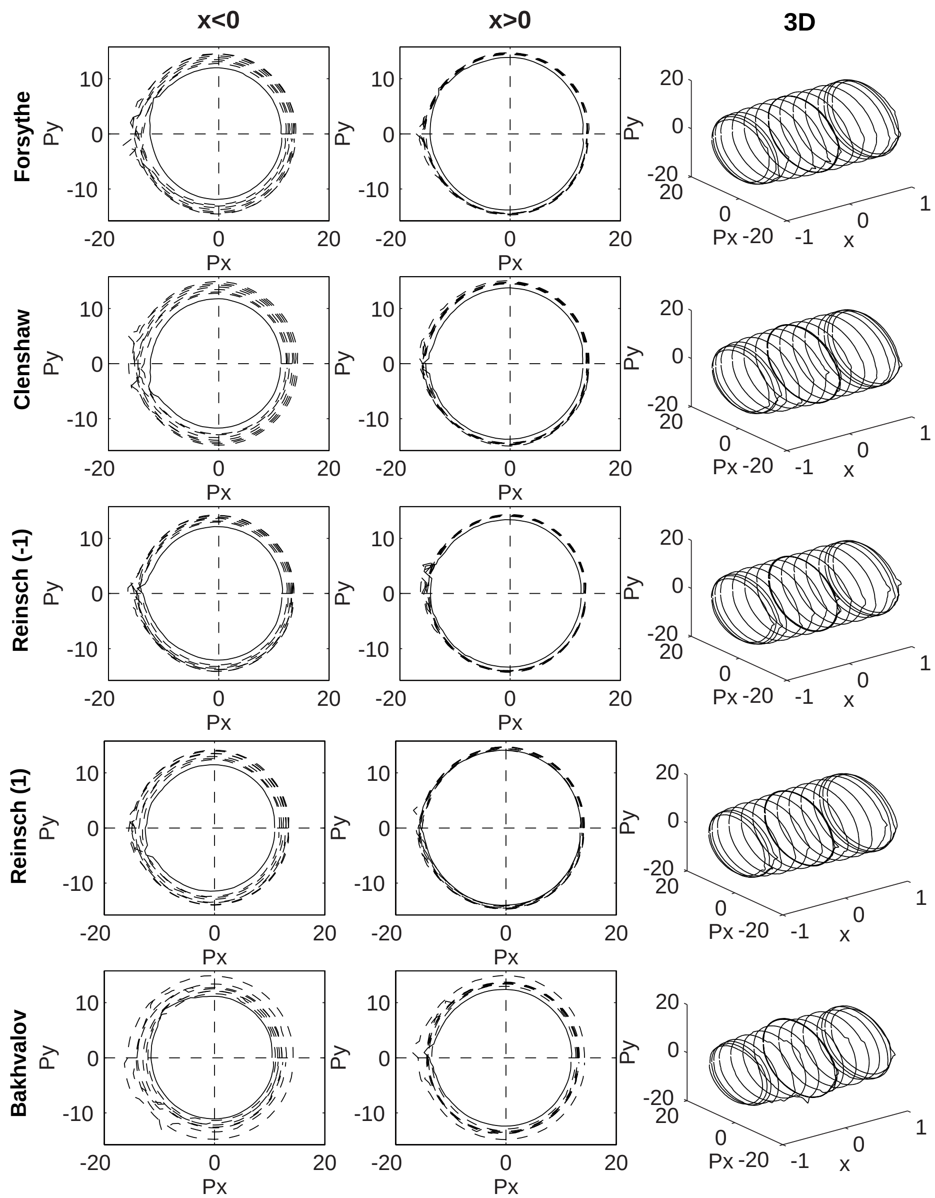

Figure 5: Logarithmic polar representation of the error for the problem **P4**.

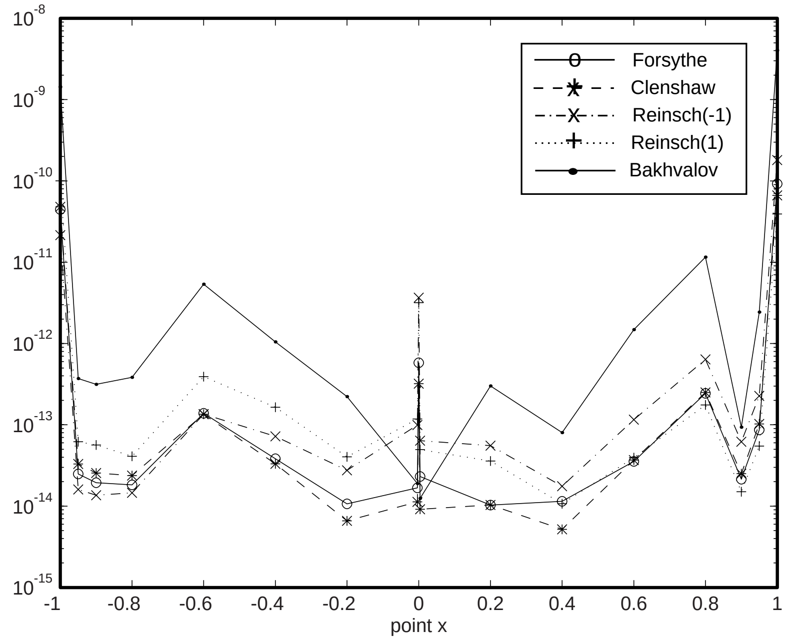

Figure 6: Standard deviation of the rounding errors in the **P1** problem.

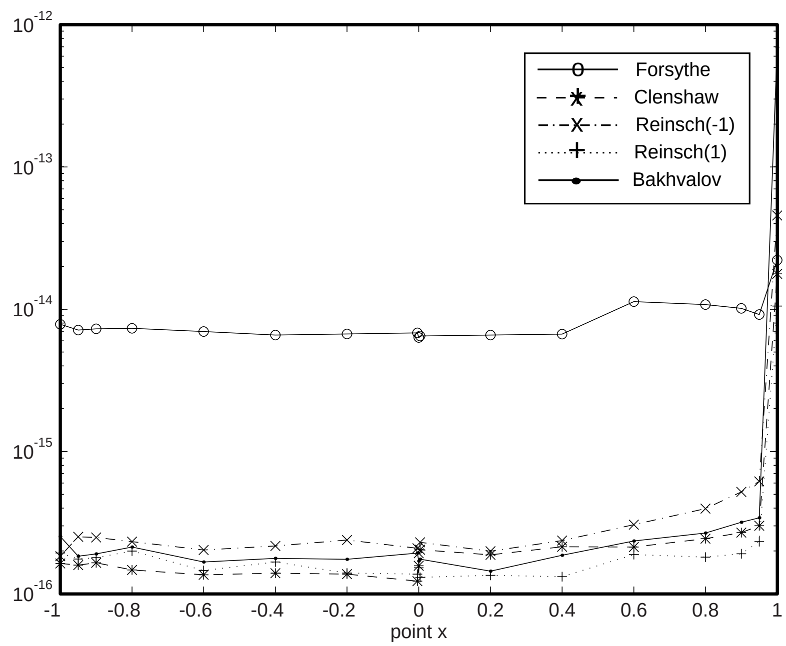

Figure 7: Standard deviation of the rounding errors in the **P2** problem.

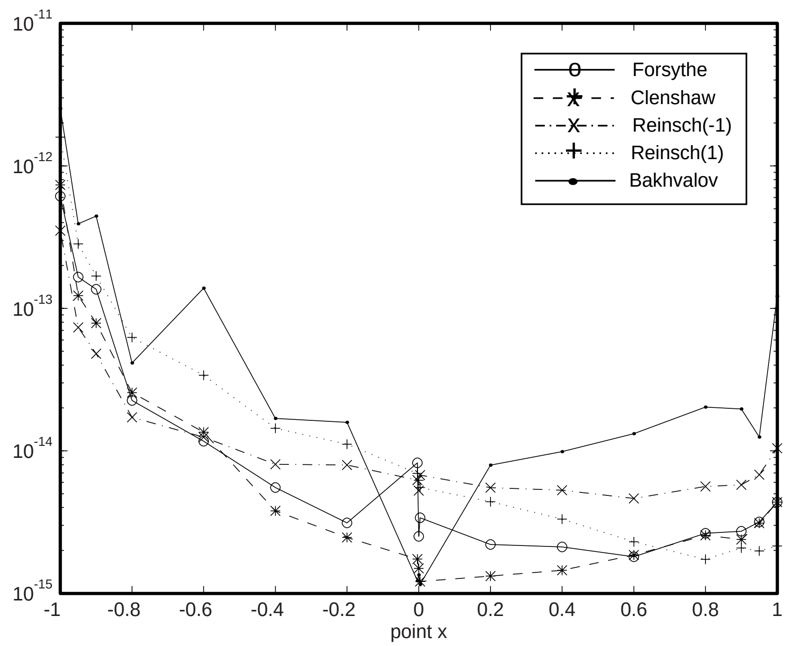

Figure 8: Standard deviation of the rounding errors in the **P3** problem.

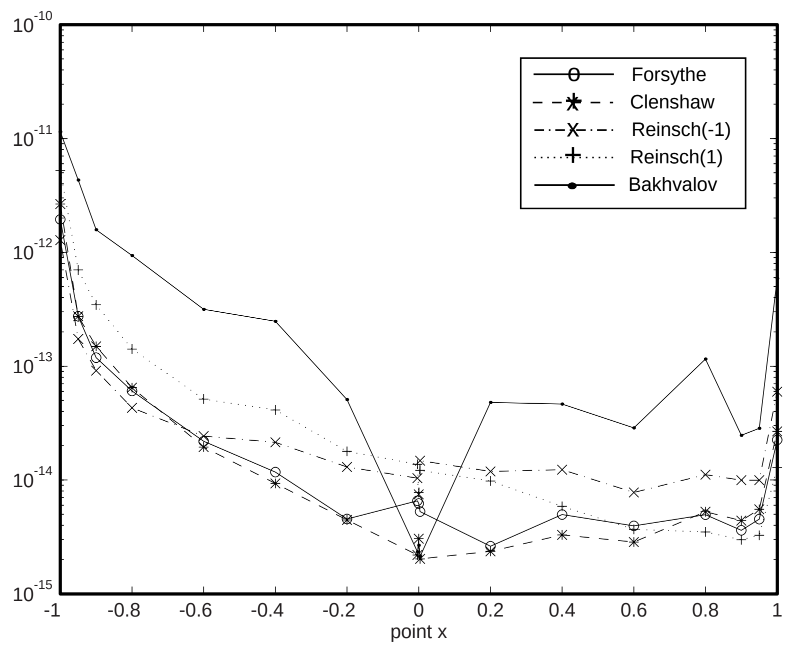

Figure 9: Standard deviation of the rounding errors in the **P4** problem.