EPSC Abstracts,
Vol. 4, EPSC2009-275, 2009
European Planetary Science Congress,

Once again on inaccuracies of trigonometric functions in computer mathematical libraries

N.A. Shupikova, **G.O. Ryabova**

Tomsk State University, Tomsk, Russia (shups@sibmail.com; rgo@rambler.ru)

The reason for the study

In 2005 Ito & Cojima published a paper [1], where a case of accuracy loss was described. More specifically, they found that the total angular momentum of gravitational N-body systems was no longer conserved in a certain type of numerical integration. The reason was the numerical inaccuracy in sine and cosine functions. As they noted: "…sometimes … mathematical functions do not possess sufficient accuracy in terms of scientific research. Even if the numerical error in a single arithmetic operation is very small, it could pile up and lead to a devastating result, producing a totally different solution in a computer simulation" [1].

We use other computer platforms, and being inspired by work of Ito & Kojima, we decided to examine our compilers and some others available. They are Turbo Pascal 7.0, Borland Delphi 7 and C++ Builder (Windows XP), and also FreePascal 1.0.8 2007/02/02 (Kubuntu Gutsy Gibbon 7.10 with Linux 2.6.22-14 kernel).

Method

We used the method described in [1]. Here its main points are repeated briefly. We calculated the value $\cos^2 x + \sin^2 x - 1$ in the x range of $[-\pi/128, \pi/128]$ using the step size of 1.0×10^{-5} radian with extended-precision arithmetic. The results of the tests are summarized in Figures 1 – 4, where $y = (\cos^2 x + \sin^2 x - 1)/\varepsilon$, and ε is the machine epsilon; $\varepsilon = 1.08420217248550443\times10^{-19}$ for extended-precision arithmetic. If values of y are distributed asymmetrically around the $y = 0$ line, it means that rounding error is not balanced, and the secular error in total angular momentum is possible.

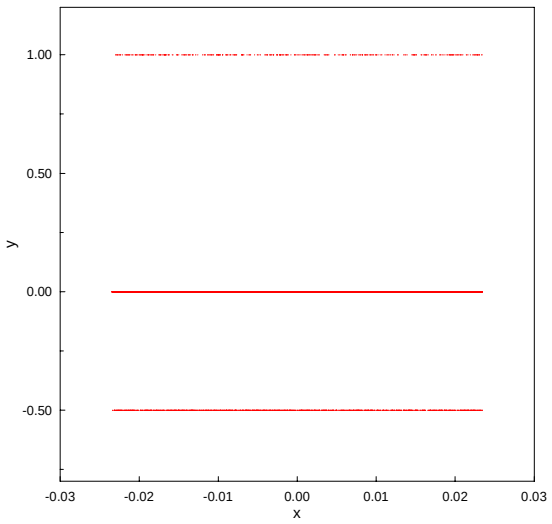

Figure 1: Residual of $\cos^2 x + \sin^2 x - 1$ normalized by machine epsilon. C++ Builder.

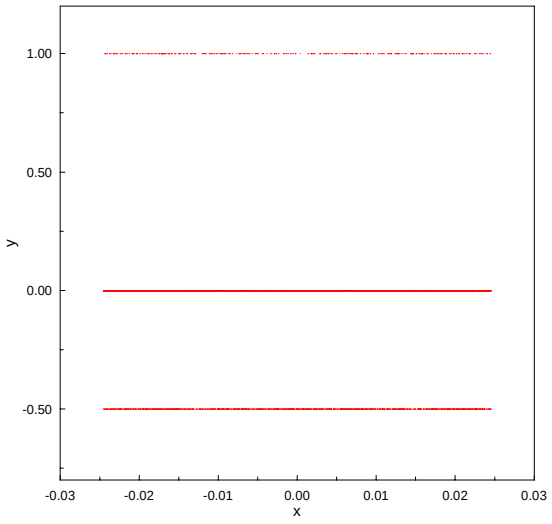

Figure 2: The same, but for FreePascal.

EPSC Abstracts,
Vol. 4, EPSC2009-275, 2009
European Planetary Science Congress,

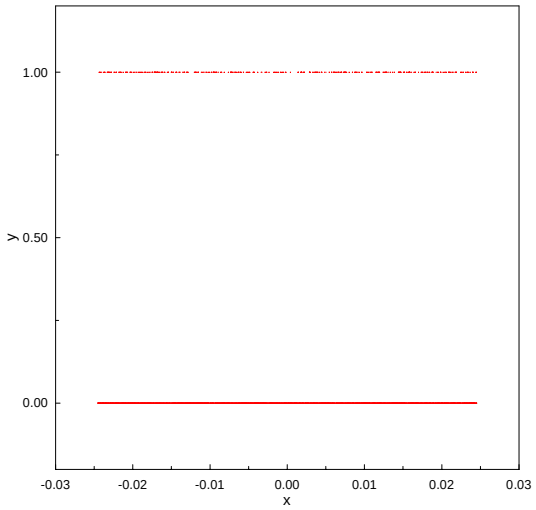

Figure 3: The same, but for Borland Delphi 7.

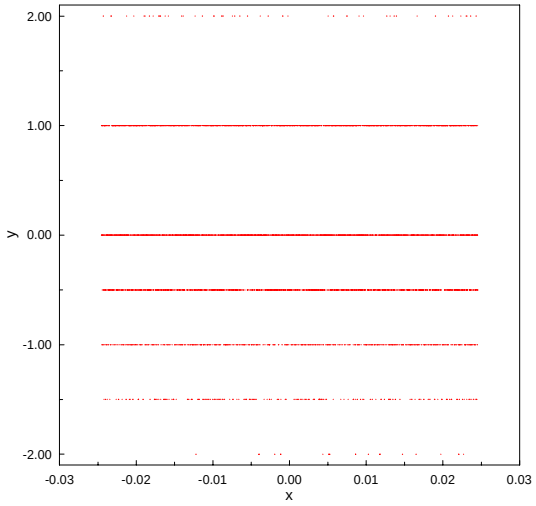

Figure 4: The same, but for Turbo Pascal 7.0.

Results

In all Figures we see asymmetrical distribution of the normalized error. It is difficult to predict without special investigation what will be the average value of the error, positive or negative, for Figures 1 and 2, but it is obvious that the errors are not balanced. For Borland Delphi 7 (Figure 3) it is clear that the average error will be positive. For Turbo Pascal 7.0 (Figure 4) the average error will be negative.

Turbo Pascal 7.0 is an old compiler, which is not supported since 1995. In fact, Borland Delphi 7 is its descendant. So its mathematical libraries should be improved, and Figure 3 seems to confirm that.

The work was supported by Ministry of Education and Science of Russian Federation (Program AVZP № 2.1.1/2629).

References

[1] Ito, T. and Kojima, S. (2005) *Publ. Natl. Astron. Obs. Japan*, 8, 17–31.