Performance analysis of parallelized extrapolation method for the orbital motions

Takashi Ito [*] *Toshio Fukushima* [†]

Abstract

We developed a parallelized version of the extrapolation method to integrate ordinary differential equations. A simple technique made its load balance among processors equal. In case of a computer with 4 processors, our method runs roughly 3 to 3.5 times as fast as the original method without any loss of accuracy. This method is suitable to study the orbital dynamics by personal computers or workstations with 2–4 processors which will be widely available in the near future.

1. Introduction

The extrapolation method is one of the most powerful methods to integrate numerically the ordinary differential equations [1]. It is often used in the orbital and rotational dynamics where an extremely high precision is required [2]. We note that calculation for different stepsizes can be executed by different processors independently. Namely, the extrapolation method can be well parallelized. In this paper we present a parallelized extrapolation method and show its application to the orbital dynamics of the solar system.

Extrapolation method [3] solves the ordinary differential equations of first order in two stages: (1): Determine the order p and the basic stepsize H. Set the sub stepsizes h_i $(i = 1, 2, \dots, p)$ as $h_i = H/(2n_i)$. Integrate the equation using h_i by an appropriate integration method, and obtain the corresponding solutions Y_i. (2): Using Y_i and h_i $(i = 1, 2, \dots, p)$, extrapolate the true solution Y_∞ where $h_i \to 0$ by a certain extrapolation technique (we used the polynomial extrapolation of Aitken-Neville [4] here). As for the integration method in the stage (1), the modified mid-point rule is often utilized since it has a character of ${h_i}^2$-convergence. As for the selection of n_i, the harmonic sequence $n_i = 1, 2, 3, 4, \dots,$ is generally recommended [5]. In addition, we can reduce round-off errors as much as a few digits by rewriting the formula of increment of y_j [6].

2. Implementation

The implementation of the extrapolation method for parallel computers is simple. Each Y_i can be calculated in parallel because all the calculations to obtain Y_i is completely independent with each other. The computational load of Y_i is roughly proportional to i when the harmonic sequence of n_i is selected. To achieve the equal load balance among processors, we use *folding* of sequence of tasks and distribute load equally to each processor. For example in the case of 4-processor machine, Y_1 and Y_8 are for processor 1, Y_2 and Y_7 are for processor 2, and so on. For the numerical integration of planetary motions in double precision, p is usually set as 8. We performed all the calculations as $p = 8$ below.

[*]Computer Center, National Astronomical Observatory, Mitaka, Tokyo 181, Japan (E-mail: tito@cc.nao.ac.jp)
[†]Division of Astrometry and Celestial Mechanics, National Astronomical Observatory, Mitaka, Tokyo 181, Japan

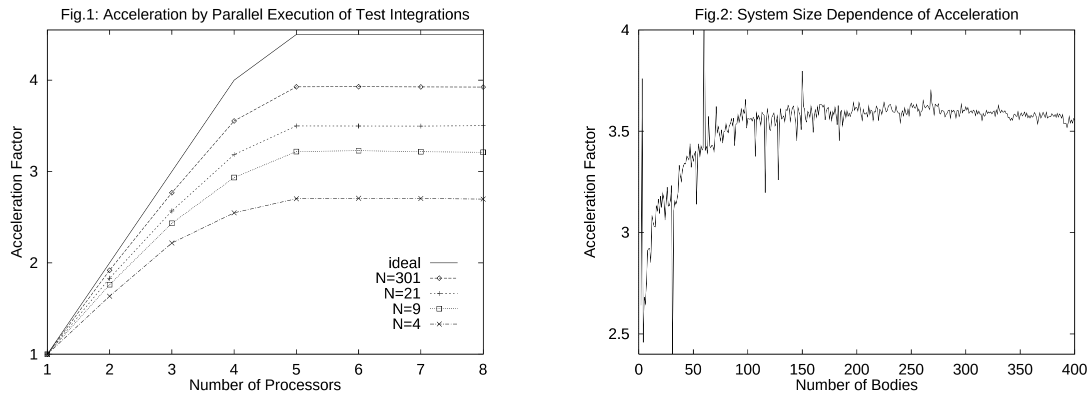

Figure 1. Acceleration by parallel execution of test integrations. Effect of the number of processors (left) and system size dependence of acceleration (right).

We can easily estimate an ideal acceleration factor by the parallelism. Let n_p be the number of processors used. Suppose that the total load is $1 + 2 + \cdots + 8 = 36$. When $n_p < 4$, the ideal acceleration is just n_p owing to the equal load balance. When $n_p \geq 5$, the bottleneck to assign a processor to Y_8 keeps the acceleration factor $36/8 = 4.5$ independently of n_p. Our numerical results shows a similar trend as this estimation.

3. Numerical results

We applied the parallelized extrapolation method to the numerical integration of celestial bodies in the solar system by using the vector/parallel supercomputer FUJITSU VPP300/16R in National Astronomical Observatory, which has 16 processors. We performed the numerical experiments using 2–8 processors, and compared the total computational time in the sense of elapsed time with that of single processor. Typical results are shown in Figure 1 left together with the ideal acceleration factors mentioned above. The system we considered consists of Sun, Jupiter, and asteroids in the main belt. We adjusted the total number of celestial bodies N by changing the number of asteroids. In Figure 1 left, the acceleration factor, which we define by the ratio of the computational time of parallel execution to that of sequential one, increases almost proportionally to the number of processors n_p when $n_p = 2, 3$, and 4. In these cases, the load balance becomes equal. However in the case of $n_p \geq 5$, the load balance becomes worse. Since the heaviest load Y_8 is the bottleneck, the acceleration factors for the case of $n_p = 5, 6, 7$ and 8 are nearly the same; they do not increase even if n_p is increased. In short, there is no need to prepare 8-processor machines to calculate the 8-th order extrapolation method. A 4-processor machine is sufficient for the 8-th order method.

Dependence on system size: The size of system is another important factor on the efficiency of the parallel algorithm. To inspect it, we changed the number of celestial bodies N and compared the computational time between the sequential calculation and the parallel calculation using 4-processors. The dependence is clear in Figure 1 right. The larger the system size (N) is, the higher the efficiency of parallelism is. This is quite a natural result, because the mostly time-consuming part in the extrapolation method is which takes $O(N^2)$ amount of calculation, while the other parts take only $O(N)$ amount of calculation.

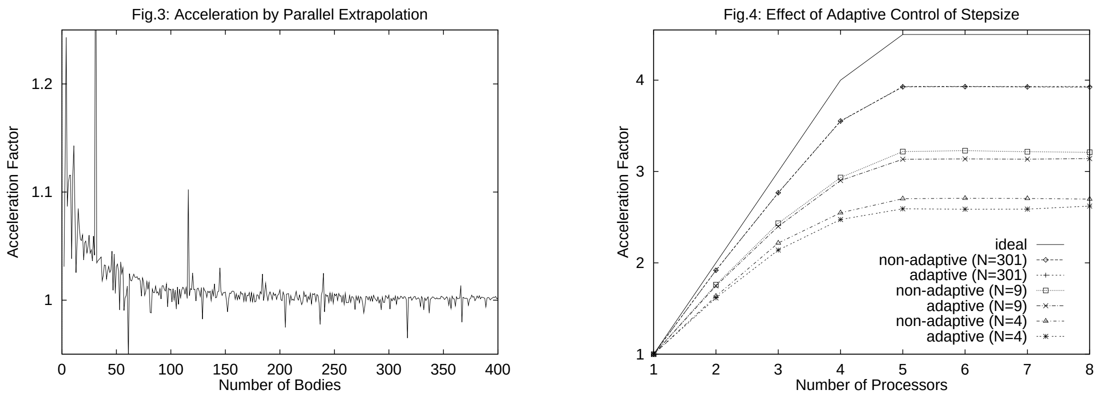

Figure 2. Acceleration by parallel extrapolation (left) and the effect of adaptive control of stepsize (right).

Effect of parallelism in the extrapolation stage: In addition to parallelize the tasks of force evaluation (stage (1)), we can also parallelized the extrapolation stage (2) with respect to the number of celestial bodies N, instead of the sub stepsizes h_i. Figure 2 left shows the ratio of the acceleration factor using the parallel extrapolation to that using the sequential extrapolation in the case of 4-processor calculation. The amount of speed-up is not so large, but apparently the effect is significant in cases of small N.

Effect of adaptive control of stepsize: In the case of the highly eccentric orbits such as those of the long-periodic comets, we need adaptive methods where H and/or p vary in the midst of integration. To inspect the efficiency of parallelization in the adaptive (variable H) cases, we performed numerical experiments including a celestial body with a highly eccentric orbit. Their results are shown in Figures 2 right. The efficiency of parallelism is hardly dependent on whether the extrapolation method is adaptive or not. This is quite reasonable because the computational amount of the procedure to determine a new H is only of $O(N)$. As for the determination of new H, we have adopted the method of Hairer et al. (1993).

4. Conclusion

By means of the technique described here, the extrapolation method is well parallelized even for small systems. Even in the small scale scalar-multiprocessor machines as 4-processor without vector units, the method described here can well be applicable. In addition, accuracy of the method does not change at all by the parallelism. The prominent character of extrapolation method — extremely high accuracy — is kept unaltered.

References

[1] Montenbruck, O. *Celes. Mech. and Dyn. Astron.,* **53**, 59, 1992.
[2] Kinoshita, H., and Nakai, H. *Celes. Mech.,* **45**, 231, 1989.
[3] Gragg, W.B. *SIAM J. Numer. Anal.,* **2**, 384, 1965.
[4] Hairer, E., Nørsett, S.P., and Wanner, G. Solving Ordinary Differential Equation I –Nonstiff Problems (second revised edition), Springer-Verlag, New York, 1993.
[5] Deuflhard, P. *Numer. Math.,* **41**, 399, 1983.
[6] Fukushima, T. *Astron. J.,* **112**, 1298, 1996.