数理解析研究所講究録 1288 巻 2002 年 102-107

Domain Decomposition Method and Infinite-Precision Numerical Simulation

Toshiki TAKEUCHI (竹内 敏己) and Hitoshi IMAI (今井 仁司)

Faculty of Engineering, University of Tokushima, Tokushima 770-8506, Japan.
(徳島大学工学部)

1 Introduction

DDM(Domain Decomposition Method) has been popular in numerical simulation[2]. It saves CPU time and memory space. Moreover, balanced accuracy realized by suitable resolution in each subdomains makes numerical simulation stable.

On the other hand, IPNS(Infinite-Precision Numerical Simulation) has been developed recently[5]. It attains ultimately high accuracy. From this property IPNS has reclaimed new fields of numerical simulation, e.g. direct simulation of inverse problems[3, 4, 6, 7].

In the paper application of DDM and IPNS is considered. A test problem is solved. Numerical results are investigated from the view point of accuracy.

2 Application of DDM and IPNS

2.1 Infinite-Precision Numerical Simulation

Numerical errors originate from the truncation error in the discretization and the rounding error. Realization of highly accurate numerical simulation needs arbitrary reduction of both errors. For such numerical simulation we proposed a simple method called IPNS(Infinite-Precision Numerical Simulation). IPNS consists of the arbitrary order approximation and the multiple-precision arithmetic. The former is used for the arbitrary reduction of the truncation error. The last is used for the arbitrary reduction of the rounding error. As for the arbitrary order approximation spectral methods are very useful[1]. Especially, the spectral collocation method is most useful. Its application is same in FDM, so it is easily applicable to nonlinear problems, even to free boundary problems[10]. In the spectral collocation method, the order of approximation can be controlled by the number of collocation points. The multiple-precision arithmetic[8] is now easily available. A lot of subroutines about it are already prepared. Some libraries are free and distributed on the net, e.g.

`http://www.lmu.edu/acad/personal/faculty/dmsmith2/FMLIB.html` [9]. IPNS has bee applied to many problems and ultimately high accuracy has been seen in numerical results.

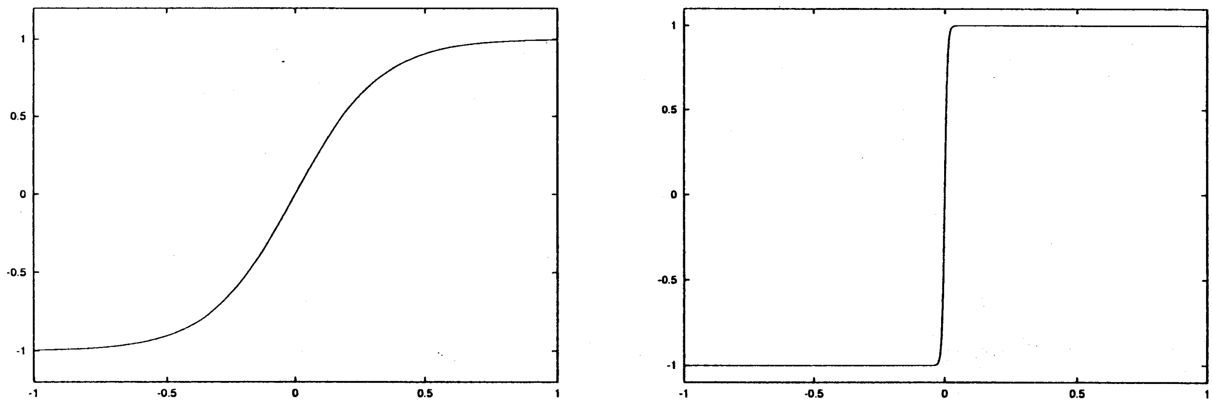
2.2 Test problem

We consider the following simple boundary value peoblem.

Problem 1. For a given a find $u(x)$ s.t.

$$\begin{cases} \dfrac{d^2u}{dx^2} = \dfrac{-8a^2(e^{ax} - e^{-ax})}{(e^{ax} + e^{-ax})^3}, & -1 < x < 1, \\ u(-1) = \dfrac{e^{-a} - e^{a}}{e^{-a} + e^{a}}, \quad u(1) = \dfrac{e^{a} - e^{-a}}{e^{a} + e^{-a}}. \end{cases} \tag{1}$$

Remark 1. *The exact solution to Problem 1 is* $u(x) = \tanh(ax) = \dfrac{e^{ax} - e^{-ax}}{e^{ax} + e^{-ax}}$. *If a is large this problem becomes difficult to be solved numerically. This is because for a large a the solution becomes the step function approximately. This situation can be seen in Fig. 1.*



(a) $\tanh(3x)$ (b) $\tanh(100x)$

Fig. 1. Exact solutions for various values of a.

2.3 Application of DDM and IPNS

The exact solution to Problem 1 is analytic, so IPNS can catch it in arbitrary accuracy. However, if a is large, IPNS cost very much. Thus efficiency of DDM to such a case is our interest. Our interest is rather mathematical, so parallel computing or automatic domain decomposition are not considered. DDM is applied to Problem 1 as follows. The domain is decomposed into three subdomains $[-1,-c],[-c,c]$ and $[c,1]$ where $0 < c < 1$. Then Problem 1 is decomposed into the following three problems.

$$\begin{cases} \dfrac{d^2u}{dx^2} = \dfrac{-8a^2(e^{ax}-e^{-ax})}{(e^{ax}+e^{-ax})^3}, & c < x < 1, \\ u(1) = \dfrac{e^a - e^{-a}}{e^a - e^{-a}}, \end{cases} \tag{2}$$

$$\frac{d^2u}{dx^2} = \frac{-8a^2(e^{ax}-e^{-ax})}{(e^{ax}+e^{-ax})^3}, \quad -c < x < c, \tag{3}$$

$$\begin{cases} \dfrac{d^2u}{dx^2} = \dfrac{-8a^2(e^{ax}-e^{-ax})}{(e^{ax}+e^{-ax})^3}, & -1 < x < -c, \\ u(-1) = \dfrac{e^{-a} - e^{a}}{e^{-a} + e^{a}}. \end{cases} \tag{4}$$

For the application of IPNS with the Chebyshev polynomial in each subdomains, the following mapping functions are indroduced for mapping each subdomains into $[-1,1]$. For $-1 \leqq t \leqq 1$

$$\begin{cases} x_1(t) = \dfrac{1-c}{2}(t+1)+c, \\ x_2(t) = ct, \\ x_3(t) = \dfrac{1-c}{2}(t+1)-1. \end{cases} \tag{5}$$

By using these mapping functions equations in subdomains are transformed as follows, respectively.

$$\begin{cases} \dfrac{d^2\tilde{u}_1}{dt^2} = \dfrac{-2a^2(1-c)^2(e^{ax_1(t)}-e^{-ax_1(t)})}{(e^{ax_1(t)}+e^{-ax_1(t)})^3}, & -1 < t < 1, \\ \tilde{u}_1(1) = \dfrac{e^a - e^{-a}}{e^a + e^{-a}}, \end{cases} \tag{6}$$

$$\frac{d^2\tilde{u}_2}{dt^2} = \frac{-8a^2c^2(e^{act}-e^{-act})}{(e^{act}+e^{-act})^3}, \quad -1 < t < 1, \tag{7}$$

$$\begin{cases} \dfrac{d^2\tilde{u}_3}{dt^2} = \dfrac{-2a^2(1-c)^2(e^{ax_3(t)} - e^{-ax_3(t)})}{(e^{ax_3(t)} + e^{-ax_3(t)})^3}, & -1 < t < 1, \\ \tilde{u}_3(-1) = \dfrac{e^{-a} - e^{a}}{e^{-a} + e^{a}}. \end{cases} \tag{8}$$

Here $\tilde{u}_i(t) = u(x_i(t)),\ i = 1,\ 2,\ 3$. As for patching conditions the followings are introduced :

$$\begin{cases} \tilde{u}_1(-1) = \tilde{u}_2(1), \\ \dfrac{d\tilde{u}_1}{dt}(-1) = \dfrac{1-c}{2c}\dfrac{d\tilde{u}_2}{dt}(1), \end{cases} \tag{9}$$

$$\begin{cases} \tilde{u}_2(-1) = \tilde{u}_3(1), \\ \dfrac{1-c}{2c}\dfrac{d\tilde{u}_2}{dt}(-1) = \dfrac{d\tilde{u}_3}{dt}(1). \end{cases} \tag{10}$$

As mentioned before, our interest is efficiency of DDM in accuracy. So, iteration for parallel computing is not used. Eqs. (6), (7), (8), (9) and (10) are discretized by SCM(Spectral Collocation Method) with the Chebyshev polynomial and C-G-L(Chebyshev-Gauss-Lobatto) collocation points and they are solved simultaneously in high precision. Multiple precision arithmetic is not necessary in numerical computation seen later.

3 Numerical Results

In this section numerical results are shown. Our simple investigation did not require multiple precision and consequently strict IPNS was not carried out. However, results obtained here suggest the role of DDM in IPNS. Of course, IPNS is necessary for detailed investigation.

For the case where DDM is not applied, i.e. Problem 1 is solved by IPNS without DDM, $(N + 1)$ C-G-L points in $[-1,\ 1]$ are used. Then,

$$\text{error} = \max_{0 \leqq j \leqq N} |u^c(x_j) - u(x_j)|, \quad x_j = \cos\frac{j\pi}{N}, \quad j = 0, 1, \cdots, N, \tag{11}$$

where u^c and u denote the numerical solution and the exact solution, respectively. For the case where DDM is applied, $(N_1 + 1)$, $(N_2 + 1)$ and $(N_3 + 1)$ C-G-L points are used in $[c,\ 1]$, $[-c,\ c]$, and $[-1,\ -c]$, respectively. Then, $N = N_1 + N_2 + N_3$. Moreover, $N_1 = N_3 = 10$ for our purpose. Then,

$$\text{error} = \max_{1 \leqq i \leqq 3}\{ \max_{0 \leqq j \leqq N_i} |\tilde{u}_i^c(t_j^i) - u(x_i(t_j^i))| \}, \quad t_j^i = \cos\frac{j\pi}{N_i}, \quad j = 0, 1, \cdots, N_i. \tag{12}$$

$_i^c$ denotes the numerical solution by DDM.

Table 1. Maximum error for Problem 1 with $a = 100$.
(Quadruple precision, DDM : $c = 0.1$, $N_1 = N_3 = 10$)

N	IPNS	IPNS+DDM	N	IPNS	IPNS+DDM
40	9.215×10^{-1}	2.233×10^{-1}	600	5.747×10^{-5}	6.559×10^{-8}
60	9.423×10^{-1}	3.637×10^{-3}	700	7.890×10^{-6}	6.559×10^{-8}
80	9.186×10^{-1}	2.093×10^{-4}	800	1.318×10^{-6}	6.559×10^{-8}
100	8.370×10^{-1}	1.342×10^{-5}	900	2.361×10^{-7}	6.559×10^{-8}
120	7.113×10^{-1}	7.019×10^{-7}	1000	4.713×10^{-8}	6.559×10^{-8}
140	5.691×10^{-1}	2.537×10^{-8}	1200	1.456×10^{-9}	6.559×10^{-8}
160	4.337×10^{-1}	6.350×10^{-8}	1400	5.147×10^{-11}	6.559×10^{-8}
180	3.180×10^{-1}	6.548×10^{-8}	1600	1.905×10^{-12}	6.559×10^{-8}
200	2.263×10^{-1}	6.558×10^{-8}	1800	7.166×10^{-14}	6.559×10^{-8}
300	3.185×10^{-2}	6.559×10^{-8}	2000	2.736×10^{-15}	6.559×10^{-8}
400	3.790×10^{-3}	6.559×10^{-8}	3000	2.629×10^{-22}	6.559×10^{-8}
500	4.519×10^{-4}	6.559×10^{-8}	—	—	—

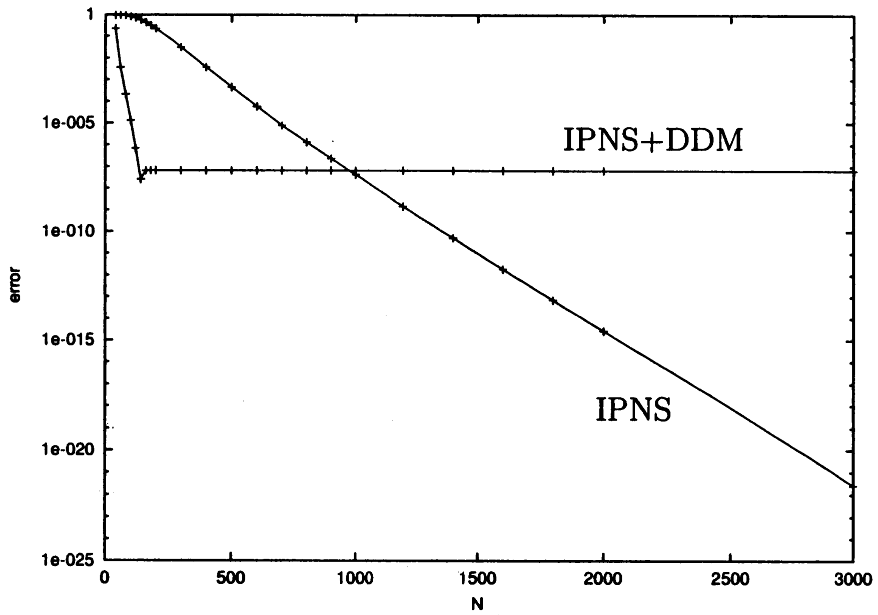


Fig. 2. Behavior of maximum error for Problem 1 with $a = 100$.
(Quadruple precision, DDM : $c = 0.1$, $N_1 = N_3 = 10$)

ı these numerical results it is seen DDM is efficient in IPNS. This means DD
· space (and CPU time) for a degree of accuracy comparing with the case]

not used. At the same time, it should be remarked improper DDM spoils merit of DDM. This means high resolution in the region where solutions change a lot does not always attain high accuracy.

4 Conclusion

In the paper DDM is applied in IPNS. Numerical results show efficiency of DDM on saving memory space (and CPU time). At the same time they also show high accuracy is not attained by improper resolution in each subdomains. Our future work is parallelization and solving more difficult problems in high accuracy.

Acknowledgements

This work is partially supported by Grants-in-Aids for Scientific Research (No. 13640119), from the Japan Society of Promotion of Science.

References

[1] C. Canuto et al., Spectral Methods in Fluid Dynamics, Springer, 1998.

[2] N. Debit et al., Domain Decomposition Methods in Science and Engineering, ddm.org, 2001.

[3] H. Imai and T. Takeuchi, Application of the Infinite-Precision Numerical Simulation to an Inverse Problem, NIFS-PROC-**40**(1999), 38-47.

[4] H. Imai and T. Takeuchi, Some Advanced Applications of the Spectral Collocation Method, GAKUTO Int. Ser. Math. Sci. Appl., **17**(2001), 323-335.

[5] H. Imai, T. Takeuchi and M. Kushida, On Numerical Simulation of Partial Differential Equations in Infinite Precision, Advances in Mathematical Sciences and Applications, **9**(2)(1999), 1007-1016.

[6] H. Imai, T. Takeuchi, M. Nakamura and N. Ishimura, A DIRECT APPROACH TO AN INVERSE PROBLEM, GAKUTO Int. Ser. Math. Sci. Appl., **12**(1999), 223-232.

[7] H. Imai, T. Takeuchi and H. Sakaguchi, Infinite Precision Numerical Simulation for PDE Systems and Its Applications, RIMS Kokyuroku, Kyoto University, **1147**(2000), 42-50.

[8] D. E. Knuth, The Art of Computer Programming, Addison-Wesley, 1981.

[9] D. M. Smith, A FORTRAN Package For Floating-Point Multiple-Precision Arithmetic, Transactions on Mathematical Software, **17**(1991), 273-283.

[10] Tarmizi, T. Takeuchi, H. Imai and M. Kushida, Numerical simulation of one-dimensional free boundary problems in infinite precision, Advances in Mathematical Sciences and Applications, **10**(2)(2000), 661-672.