Parareal in time: a new approach for Kansa's methods

Zeng Y., Duan Y., Zhang P., and Ye W. J.

Department of Computational Mathematics and Center for Applied Mathematics, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, P. R. China

Keywords: Parareal; Kansa's method; Radial basis functions Partial differential equations

Abstract: The parareal algorithm is a method to solve time dependent problems parallel in time. The focus of this paper is analyzing to the results of the solution of partial differential equations (PDEs) by the time parallel coupling Kansa's method using radial basis functions. Numerical examples demonstrate the advantages of the proposed methods for computational effort and the accuracy in the numerical solution.

1. Introduction

Kansa's method [3] (also namely unsymmertic collocation method) to solve PDEs is very attractive due to the fact that it is a true meshless method and spatial dimension independent, which can easily be extended to solve high dimensional problems. Stability and convergence analysis was proposed by Schaback [7, 8].

The parareal algorithm was presented by Lions, Maday, and Turinici in [1] as a numerical method to solve evolution problems in parallel and was extensively analyzed in [5]. The algorithm has received a lot of attention over the past few years, especially in the domain decomposition literature [4]. The idea of time parareal algorithm is that we can solve a time-dependent problem, then, compute the numerical solution on each sub-domain in time with finer time step independently. Finally, a correction is designed to match the exact solution. The same is true for spatial parallelism.

What's more, there are many scholars or researchers who have done a lot of research on the parallel time coupling with other methods. The domain decomposition algorithm [2] combine with Kansa's method was proposed by Y. Duan, P. F. Tang and T. Z. Huang, which show that this method can reduce the number of conditions of the collocation matrix in some extent, and improve the solving speed. At the same time, Fasshauer G. E. [6] reduced the number of matrix conditions by preconditioning technology.

This paper is devoted to analyzing the results of the solution of PDEs by the time parallel coupling Kansa's method. One of its advantages is that the solution of every moment can be obtained. Another advantage is that it is suitable for computing large-scale real-time problems, which can be miniaturized to achieve the purpose of reducing the number of conditions. This paper is organized as follow: section II is in detail devoted to introduce Kansa's method using RBFs. In section III we introduce correlation algorithm. Numerical examples are given in section IV. A conclusion is made at the end of this paper.

2. Kansa's method using RBFs.

In this part, we give a theoretical introduction to Kansa's method with the general elliptic boundary value problems

$$\begin{cases} Lu = f \quad in \ \ \Omega \in R^n, \\ Bu = g \quad on \ \ \partial\Omega \ . \end{cases} \tag{1}$$

Where Ω is a n-dimensional domain, with a boundary $\partial\Omega$. f, g is a given function. The

operators L, B are any elliptic operator and some boundary operator. For a given set of distinct centers $\{x_1,\cdots, x_n\}\subset\Omega, \{x_{n+1},\cdots, x_N\}\subset\partial\Omega$, and a radial basis functions $\phi(r)$. Let the approximated solution of equation (1) to be $u(x)=\sum_k^N \lambda_k\phi(\|x-x_k\|)$, here $\|\cdot\|$ is the Euclidean norm. Where λ_i are undetermined coefficients. Substitute $u(x)=\sum_k^N \lambda_k\phi(\|x-x_k\|)$ into (1), then get:

$$\begin{cases} \sum_{k=1}^{N} \lambda_k L\phi(\|x-x_k\|) = f(x) \;\; x\in\Omega, \\ \sum_{k=1}^{N} \lambda_k B\phi(\|x-x_k\|) = g(x) \;\; x\in\partial\Omega\,. \end{cases} \tag{2}$$

Substitute n points in Ω and $N-n$ points in $\partial\Omega$ into the first formula and the second formula of (2) respectively, and obtain

$$\begin{cases} \sum_{k=1}^{N} \lambda_k L\phi(\|x_i-x_k\|) = f(x_i) \;\; 1\le i\le N_n, \\ \sum_{k=1}^{N} \lambda_k B\phi(\|x_j-x_k\|) = g(x_j) \;\; N_{n+1}\le j\le N. \end{cases} \tag{3}$$

Then , we get matrix form system

$$\mathrm{A}\lambda = b. \tag{4}$$

Where $\lambda=[\lambda_1,\lambda_2,\cdots,\lambda_N]^T$, $b=[f(x_1),\cdots,\ g(x_N)]^T$,

$$\mathbf{A}=\left[L\phi(\|x_i-x_j\|),\ \ B\phi(\|x_i-x_j\|)\right]^T \; i,j=1,2,\cdots N.$$

The linear system induced by Kansa's method using RBFs just have to solve for the coefficients λ. Once the coefficient λ_i are solved, the approximate solution at any Ω is given by the interpolation formula $u(x)$. The most widely used RBFs are show in Table I. But there are other radial basis functions that are not listed due to space issues. In our numerical examples we have used the Multiquadric (MQ), which have the coefficient c called the shape parameter.

Table I. Radial Basis Functions

RBF	Definition
Multiquadric (MQ)	$\phi(r,c)=\sqrt{r^2+c^2}$
Inverse Multiquadric (IMQ)	$\phi(r,c)=1/\sqrt{r^2+c^2}$
Gaussian (GA)	$\phi(r,c)=e^{-(cr^2)}$
Thin- Plate Splines (TPS)	$r^n\log(r), n=2,4,\cdots$
Smooth Splines (SS)	$r^n, n=1,3,5,\cdots$

3. Numerical algorithm

The time domain decomposition method is described below as the model:

$$\begin{cases} au_t - \nabla^2 u = f \ \ in \ \ \Omega \quad at \ 0 < t < T, \\ Bu = g \qquad\quad on \ \ \partial\Omega \ \ at \ 0 \le t < T, \\ u(x, y, 0) = h \ \ in \ \ \Omega \quad at \ t = 0. \end{cases} \tag{5}$$

Algorithm:

Step 1: For coarse time step ΔT , we use implict scheme to solve the equation (5):

$$\begin{cases} a\dfrac{U_0^{n+1} - U_0^n}{\Delta T} - \nabla^2 U_0^{n+1} = f^{n+1} \quad in \ \ \Omega \\ BU_0^{n+1} = g^{n+1} = g(x, y, T^{n+1}) \quad on \ \ \partial\Omega \\ U_0^0 = h \qquad\qquad\qquad\qquad in \ \ \Omega \end{cases} \tag{6}$$

(6) is solved by the Kansa's method using RBFs.

Step 2: Solve N independent problems in each time interval $[T^n, T^{n+1}]$, with the following form:

$$\begin{cases} a\dfrac{\partial u_k^n}{\partial t} - \nabla^2 u_k^n = f \quad in\,\Omega \quad at\ T^n < t < T^{n+1}, \\ Bu_k^n = g \qquad\quad on\,\partial\Omega \ \ at\ T^n \le t < T^{n+1}, \\ u_k^n(x, y, T^n) = U_k^n \quad in \ \ \Omega \quad at\ t = T^n. \end{cases} \tag{7}$$

by fine step and meshless method with the same as step 1.

Step 3: Defined the jump on T^n by $S_k^n = u_k^{n-1}(T^n) - U_k^n$

and solve the following equation

$$\begin{cases} a\dfrac{\delta_k^{n+1} - \delta_k^n}{\Delta T} - \nabla^2 \delta_k^{n+1} = a\dfrac{S_k^n}{\Delta T} \ \ in \ \ \Omega, \\ B\delta_k^{n+1} = 0 \qquad\qquad\qquad on \ \ \partial\Omega, \\ \delta_k^0 = 0 \qquad\qquad\qquad\quad in \ \ \Omega. \end{cases} \tag{8}$$

Step 4: Let $U_{k+1}^n = u_k^{n-1}(T^n) + \delta_k^n$, then solve

$$\begin{cases} a\dfrac{\partial u_{k+1}^n}{\partial t} - \nabla^2 u_{k+1}^n = f, \ \ in \ \ \Omega \ \ at\ T^n < t < T^{n+1}, \\ Bu_{k+1}^n = g\ , \qquad\quad on\,\partial\Omega \ \ at\ T^n \le t < T^{n+1}, \\ u_{k+1}^n(x, y, T^n) = U_{k+1}^n, \ in \ \ \Omega \quad at\ t = T^n. \end{cases} \tag{9}$$

On each time sub-interval $[T^n, T^{n+1}]$, by the same method with step 2: Once a tolerance is designed, we can obtain the numerical solution of the original problem after several iterations.

4. Numerical examplems

In this section our goal is to use the above algorithm to solve heat conduction partial differential equations by using the above algorithm, get the numerical solution, and analyze their calculation quantity and accuracy .

4.1 Example 1 one dimensional case:

Consider the following linear heat conduction equations in one-dimension :

$$\frac{\partial u(x,t)}{\partial t} - \nabla^2 u = 0,\ 0 < x < 1,\ 0 < t < 0.2, \tag{10}$$

with boundary and initial conditions

$$\begin{aligned} &u(0,t) = u(1,t) = 0,\ 0 < t < 0.2, \\ &u(x,t) = \sin(\pi x),\ t=0,\ 0 < x < 1. \end{aligned} \tag{11}$$

Where ∇^2 is usually the Laplace operator and let coarse time step to be ΔT, the interval $[0,0.2]$ is divide N subdomain , the fine time step in the subdomain is taken as Δt. 200 nodes are uniformly distributed in [0,1] and taking $\phi(r,c) = \sqrt{r^2+c^2}$ as the radial basis function. Here $c = 0.1266$. Analytical solution is $u(x,t) = e^{-\pi^2 t}\sin(\pi x)$. The considered errors are

$$\mathrm{MSR} = \sqrt{\sum_{i=1}^{N} (u\_exact(x_i,\ t) - u\_num(x_i,t))^2 \Big/ \mathrm{N}},$$

$$\mathrm{MAX} = \max \left| u\_exact(x_i,\ t) - u\_num(x_i,t) \right|.$$

Table II. Time $t = 0.05$ and Error with the change of N.

N	MSR	MAX	Cond
10	0.0042	0.0059	1.3136E+14
20	0.0020	0.0029	2.6397E+14
40	0.0010	0.0015	5.2685E+14
80	0.0005	0.0007	1.0385E+15
160	0.0003	0.0004	2.0337E+15

Table III. Time t= $t = 0.1$ and Error with the change of N.

N	MSR	MAX	Cond
10	0.0050	0.0071	1.3136E+14
20	0.0025	0.0036	2.6397E+14
40	0.0013	0.0018	5.2685E+14
80	0.0006	0.0009	1.0385E+15
160	0.0003	0.0005	2.0337E+15

Table IV. Time $t = 0.15$ and Error with the change of N.

N	MSR	MAX	Cond
10	0.0015	0.0021	1.3136E+14
20	0.0023	0.0033	2.6397E+14
40	0.0012	0.0017	5.2685E+14
80	0.0006	0.0008	1.0385E+15
160	0.0003	0.0004	2.0337E+15

Table V. $t = 0.2$ and Error with the change of N.

N	MSR	MAX	Cond
10	0.0038	0.0054	1.3136E+14
20	0.0019	0.0027	2.6397E+14
40	0.0009	0.0014	5.2685E+14
80	0.00048	0.00068	1.0385E+15
160	0.00023	0.00033	2.0337E+15

Table II-V is the numerical result error analysis at time $t=0.05$, $t=0.1$, $t=0.15$ and $t=0.2$ respectively, which indicates that one of the advantages of time domain decomposition is that the solution at each time can be obtained.

4.2 Example 2 two dimensional case:

Consider the following linear heat conduction equations in two-dimension:

$$\begin{cases} 16\dfrac{\partial u(x,y,t)}{\partial t}-\nabla^2 u=0, \ (x,y)\in\Omega, \ t>0, \\ u(0,y,t)=u(1,y,t)=0, \ 0<y<1, t>0, \\ u(x,y,0)=\sin(\pi x)\cos(\pi y), \ t=0. \end{cases} \tag{12}$$

Where Ω=[0,1]*[0,1], ∇^2 denote Laplace operator. In the same time, following the method of example 1 to solve the formula (12). Let coarse time step to be ΔT=T/N, the interval [0,8] is divide N subdomain and the fine time step in the subdomain is taken as $\Delta t=T/N*Nn$. Where T is the total time, N is the number of partitions, Nn is the number of centers. 400 nodes are uniformly distributed in [0,1]*[0,1] and taking $\phi(r,c)=\sqrt{r^2+c^2}$ as the radial basis function . Here $c=0.1$. Analytical solution of this question is $u=e^{(-\pi^2/8)\,t}\sin(\pi x)\cos(\pi y)$. The error representation is the same as the example one.

Table VI. $N=80$ and Error with the change of Time

Time(t/s)	MSR	MAX	I	Cond
0.1	0.0033	0.0097	5	2.8738E+06
1	0.0028	0.0055	5	2.8738E+06
2	0.0014	0.0025	5	2.8738E+06
4	0.0002	0.0003	5	2.8738E+06
6	2.8797E-05	4.4194E-05	5	2.8738E+06
8	3.2624E-06	4.9132E-06	5	2.8738E+06

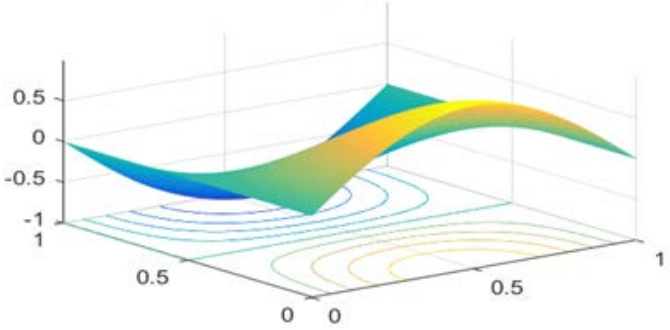

Figure 1. $t=0.001$ The exact solution distribution.

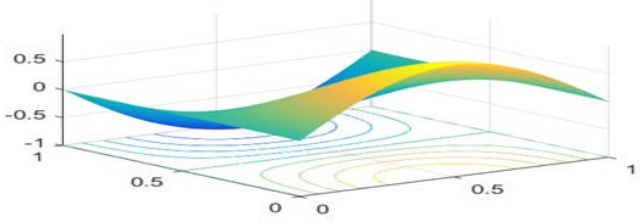

Figure 2. $t=0.001$ Numerical solution distribution.

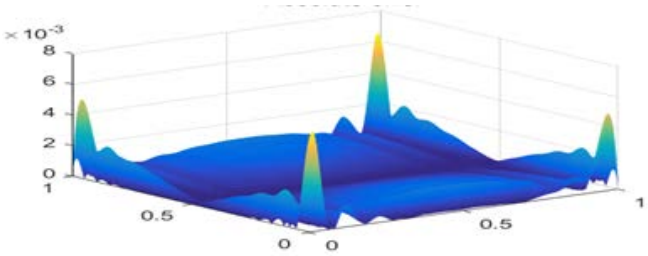

Figure 3. $t = 0.001$ Absolute error distribution.

From Fig. 1 to Fig. 3 we know that the time domain decomposition method mentioned above is used to solve the partial differential equation, and the numerical solution is very close and the error is relatively small. At the same time, this method can know the numerical solution at each moment. The second advantage is that it is suitable for computing large-scale real-time problems, which can be miniaturized to achieve the purpose of reducing the number of conditions.

4.3 Example 3 two dimensional case:

$$\begin{cases} 10000\dfrac{\partial u}{\partial t} - \nabla^2 u = 0 & (x,y) \in \Omega \quad 0 < t < 18, \\ u = 100 & x = D_1 \text{ or } y = D_2 \quad 0 \le t < 18, \\ \dfrac{\partial u}{\partial x} = 0 & x = 0 \quad 0 \le t < 18, \\ \dfrac{\partial u}{\partial y} = 0 & y = 0 \quad 0 \le t < 18, \\ u(x,y,0) = 80\cos\left(\dfrac{3\pi}{2D_1}x\right)\cos\left(\dfrac{5\pi}{2D_2}y\right) + 100 & (x,y) \in \Omega \quad t = 0. \end{cases}$$

Where $\Omega = \{(x,y) \mid 0 < x < D_1, 0 < y < D_2\}$, $D_1 = 0.4$, $D_2 = 0.2$, Analytical solution of this problem is

$u = 80[-(\frac{15}{4}\pi x)^2 + (\frac{25}{2}\pi y)^2)(\frac{t}{10000})]\cos(\frac{15}{4}x)\cos(\frac{25}{2}y) + 100.$ Will $[0,12]$ be divided into 12 districts, namely, the step size $\Delta T = 1$, then solve the problem in each district, which means that each district is divided into 50 equal parts. Taking 100 evenly distributed collocation points in Ω. A_err and R_err stands for absolute error and relative error, respectively.

Table VII show that the maximum relative error of the numerical solution is not exceeded1.3% when the time parallel algorithm is applied to solve example 3, which indicates that the time domain decomposition algorithm to solve the initial boundary value problem of the heat conduction equation is effective and feasible.

Table VII. The error of the numerical solution at the 800 points.

Time	1	2	3	4	5	6
A_err	0.5773	0.6830	0.7683	0.8262	0.8599	0.9290
R_err	0.0091	0.0113	0.0121	0.0124	0.0123	0.0124
Time	7	8	9	10	11	12
A_err	0.9737	1.0082	1.0304	1.0410	1.0533	1.0571
R_err	0.0122	0.0119	0.0117	0.0115	0.0115	0.0116

5. Conclusion

In this paper, we use the parallel algorithm to couple Kansa's method to solve PDE with boundary condition, initial condition and Mixed boundary condition. It can be clearly seen from the numerical

examples that this method can improve the precision of Kansa's method and can know the numerical solution at each moment. Secondly, it is suitable for computing large-scale real-time problems, which can be miniaturized to achieve the purpose of reducing the number of conditions and the advantages of the improved algorithm are proved by numerical examples.

Acknowledgment

This work was partly supported by NSFC 11171054 and Science Strength Promotion Programme of UESTC. Corresponding author: zengyanuestc@163.com (Zeng Y.) duanyong@uestc.edu.cn (Duan Y.)

References

[1] J. L. Lions, Y. Maday and G. Turinici, "RWsolution d.EDP par un schema temps5pararWel6," Compt. Rend. De Lacade. De. Sci., vol. 332, pp. 661-668, 2001.

[2] Y. Duan, P. F. Tang and T. Z. Huang, "A novel domain decomposition method for highly oscillating partial differential equations,'' Eng. Anal. Bound. Elem., vol. 33, pp. 1284-1288, 2009.

[3] E. J. Kansa, "Multiquadrics - A scattered data approximation scheme with applications to computional fluid-dynamics - II: Solutions to parabolic, hyperbolic and elliptic partial differential equations," Comput. Math. Appl., vol. 19, pp. 147-161, 1990.

[4] R. Kornhuber, R. H. W. Hoppe, J. P´eeriaux, O. Pironneau, O. B. Widlund, and J. Xu, eds., Proceedings of the 15th International Domain Decomposition Conference, Lect. Notes Comput. Sci. Eng. 40, Springer, Berlin, 2003.

[5] M. J. Gander and S. Vandewalle, "Analysis of parareal of time-parallel time-integration methods," SIAM. J. Sci. Comput., vol. 29 pp. 556-578, 2007.

[6] G. E. Fasshauer, "Solving differential equations with radial basis functions: multilevel methods and smoothing," Adv. Comput. Math., vol. 11, pp. 139-159, 1999.

[7] R. Schaback, "Convergence of unsymmertic kernel-based meshless collocation methods," SIAM. J. Numer. Anal., vol. 45 pp. 333-351, 2007.

[8] K. C. Cheung, L. Ling and R. Schaback, "H^2 convergence of least-squares kernel collocation methods," SIAM J. Numer. Anal., vol. 56 , pp. 614-633, 2018.