Hindawi Publishing Corporation
The Scientific World Journal
Volume 2014, Article ID 681707, 5 pages
http://dx.doi.org/10.1155/2014/681707

Hindawi

Research Article

A Domain Decomposition Method for Time Fractional Reaction-Diffusion Equation

Chunye Gong,[1,2,3] Weimin Bao,[1,2] Guojian Tang,[1] Yuewen Jiang,[4] and Jie Liu[3]

[1] *College of Aerospace Science and Engineering, National University of Defense Technology, Changsha 410073, China*
[2] *Science and Technology on Space Physics Laboratory, Beijing 100076, China*
[3] *School of Computer Science, National University of Defense Technology, Changsha 410073, China*
[4] *Department of Engineering Science, University of Oxford, Oxford OX2 0ES, UK*

Correspondence should be addressed to Chunye Gong; gongchunye@gmail.com

Received 21 December 2013; Accepted 20 February 2014; Published 19 March 2014

Academic Editors: A. Atangana, S. C. O. Noutchie, and A. Secer

The computational complexity of one-dimensional time fractional reaction-diffusion equation is $O(N^2M)$ compared with $O(NM)$ for classical integer reaction-diffusion equation. Parallel computing is used to overcome this challenge. Domain decomposition method (DDM) embodies large potential for parallelization of the numerical solution for fractional equations and serves as a basis for distributed, parallel computations. A domain decomposition algorithm for time fractional reaction-diffusion equation with implicit finite difference method is proposed. The domain decomposition algorithm keeps the same parallelism but needs much fewer iterations, compared with Jacobi iteration in each time step. Numerical experiments are used to verify the efficiency of the obtained algorithm.

1. Introduction

Fractional equations can be used to describe some physical phenomenon more accurately than the classical integer order differential equation. The reaction-diffusion equations play an important role in dynamical systems of mathematics, physics, chemistry, bioinformatics, finance, and other research areas. There has been a wide variety of analytical and numerical methods proposed for fractional equations [1–7], for example, finite difference method [8], finite element method [9], Adomian decomposition method [10], and spectral technique [11]. Interest in fractional reaction-diffusion equations has increased [12].

Domain decomposition methods (DDM) solve a boundary value problem by splitting it into smaller boundary value problems on subdomains and iterating it to coordinate the solution between adjacent subdomains [13]. A coarse problem with one or few unknowns per subdomain is used to further coordinate the solution between the subdomains globally. The DDM can be divided into two categories: the overlapping and nonoverlapping [14]. Chan and Mathew [15] gave a survey on iterative domain decomposition techniques that had been developed for solving several kinds of partial differential equations, including elliptic, parabolic, and differential systems such as the Stokes problem and mixed formulations of elliptic problems. The problems on the subdomains are almost independent, which makes domain decomposition methods suitable for parallel computing. Parallel computing is used to solve intensive computation applications simultaneously [16], such as particle transport [17, 18] and fast multipole methods [19]. It is time consuming to numerically solve fractional differential equations for long time tail. Parallel computing [20–22] can be used to overcome the computational challenge of fractional approximation. DDM will embody large potential for a parallelization of the numerical solution for fractional equations. Until today the power of DDM for approximating fractional derivatives and solving fractional differential equations has not been recognized.

This paper focuses on the Caputo fractional reaction-diffusion equation:

$$\begin{aligned} &{}_0^C D_t^\alpha u(x,t) + \mu u(x,t) \\ &\quad = \frac{\partial^2 u(x,t)}{\partial x^2} + Kf(x,t), \quad (0 < \alpha < 1), \\ &u(x,0) = g(x), \quad x \in [0, x_R], \\ &u(0,t) = u(x_R,t) = 0, \quad t \in [0,T] \end{aligned} \tag{1}$$

on a finite domain $0 \le x \le x_R$ and $0 \le t \le T$. The $\mu > 0$ and K are constants. If α equals 1, (1) is the classical reaction-diffusion equation. The fractional derivative is in the Caputo form.

2. Background

2.1. Numerical Solution. The fractional derivative of $f(t)$ in the Caputo sense is defined as [23]

$${}_0^C D_t^\alpha f(t) = \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{f'(\xi)}{(t-\xi)^\alpha} d\xi, \quad (0 < \alpha < 1). \tag{2}$$

If $f'(t)$ is continuous bounded derivatives in $[0,T]$ for every $T > a$, we can get

$$\begin{aligned} {}_0 D_t^{\alpha f(t)} &= \lim_{\xi \to 0, n\xi = t} \xi^\alpha \sum_{i=0}^{n} (-1)^i \binom{\alpha}{i} \\ &= \frac{f(0)\, t^{-\alpha}}{\Gamma(1-\alpha)} + \frac{1}{\Gamma(1-\alpha)} \int_0^t \frac{f'(\xi)}{(t-\xi)^\alpha} d\xi. \end{aligned} \tag{3}$$

Define $\tau = T/N$, $h = x_R/(M+1)$, $t_n = n\tau$, and $x_i = 0 + ih$ for $0 \le n \le N, 0 \le i \le M+1$. Define u_i^n, f_i^n, and g_i as the numerical approximation to $u(x_i, t_n)$, $f(x_i, t_n)$, and $g(x_i)$. We can get [12]

$$\begin{aligned} {}_0^C D_t^\alpha u(x,t)\big|_{x_i}^{t_n} &= \frac{1}{\tau\Gamma(1-\alpha)} \\ &\quad \times \left[b_0 u_i^n - \sum_{k=1}^{n-1} \left(b_{n-k-1} - b_{n-k} \right) u_i^k - b_{n-1} u_i^0 \right] \\ &\quad + \bigcirc\left(\tau^{2-\alpha} \right), \end{aligned} \tag{4}$$

where $1 \le i \le M$, $n \ge 1$, and

$$b_l = \frac{\tau^{1-\alpha}}{1-\alpha} \left[(l+1)^{1-\alpha} - l^{1-\alpha} \right], \quad l \ge 0. \tag{5}$$

By using center difference scheme for $\partial^2 u(x,t)/\partial x^2$, we can get

$$\left. \frac{\partial^2 u(x,t)}{\partial x^2} \right|_{x_i}^{t_n} = \frac{1}{h^2} \left(u_{i+1}^n - 2u_i^n + u_{i-1}^n \right) + \bigcirc\left(h^2 \right). \tag{6}$$

The implicit finite difference approximation for (1) is

$$\begin{aligned} &\frac{1}{\tau\Gamma(1-\alpha)} \left[b_0 u_i^n - \sum_{k=1}^{n-1} \left(b_{n-k-1} - b_{n-k} \right) u_i^k - b_{n-1} u_i^0 \right] + \mu u_i^n \\ &\quad = \frac{u_{i+1}^n - 2u_i^n + u_{i-1}^n}{h^2} + Kf_i^n. \end{aligned} \tag{7}$$

Define $s = 2/h^2 + b_0 \tau^{-1}/\Gamma(1-\alpha) + \mu$, $U^n = (u_1^n, u_2^n, \ldots, u_M^n)^T$, $F^n = (f_1^n, f_2^n, \ldots, f_M^n)^T$, and r_l as

$$r_l = \frac{b_l - b_{l+1}}{s}. \tag{8}$$

Equation (7) evolves as

$$AU^n = \sum_{k=1}^{n-1} r_{n-1-k} U^k + b_{n-1} U^0 + KF^n, \tag{9}$$

where matrix A is a tridiagonal matrix, defined by

$$A_{M\times M} = \begin{pmatrix} s & -\frac{1}{h^2} & & & \\ -\frac{1}{h^2} & s & -\frac{1}{h^2} & & \\ & \cdot & \cdot & \cdot & \\ & & \cdot & \cdot & -\frac{1}{h^2} \\ & & & -\frac{1}{h^2} & s \end{pmatrix}. \tag{10}$$

Because $\mu > 0$, $b_0 > 0$, the elements of matrix A satisfy $|s| > |-1/h^2| + |-1/h^2|$. This means that matrix A is strictly diagonally dominant.

2.2. Computational Challenge. In order to get U^n, the right-sided computation of (9) should be performed and tridiagonal linear system should be solved. There are mainly many constant vector multiplications and many vector vector additions in the right-sided computation.

(1) The constant vector multiplications are $V' = b_{n-1}U^0$, $V^k = r_{n-1-k}U^k$, and $V'' = KF^n$.

(2) The vector vector additions are $V = V' + \sum_{k=1}^{n-1} V^k + V''$.

(3) After solving tridiagonal linear system $AU^{n+1} = V$, we get U^{n+1}.

The Thomas algorithm for tridiagonal systems needs $5M$ multiplications and $3M$ additions. The computational complexity of $AU^n = V$ is $O(M)$. The total computation of (9) is determined by $\sum_{k=1}^{n-1} r_{n-1-k}U^k$, which means $(n-1)M$ multiplications and $(n-2)M$ additions for each time step;

$$\sum_{n=1}^{N} (2nM - 3M) = O\left(N^2 M \right). \tag{11}$$

The computational complexity of (1) is $O(N^2M)$, while the computational complexity of classical one-dimensional reaction-diffusion equation is only $O(NM)$. The computational cost of (11) varies linearly along the number of grid points but squares with the number of time steps.

3. Domain Decomposition Method

3.1. DDM with Two Subdomains. Similar to the classical alternating Schwarz method [13, 24], the domain $\Omega = [0, x_R] = p_0, p_1, \ldots, p_M$ can be divided into two subdomains Ω_a and Ω_b. There are $M + 1$ grid points for Ω. $\Omega_a = \{p_0, p_1, \ldots, p_m, p_{m+1}\}$ and $\Omega_b = \{p_m, p_{m+1}, \ldots, p_M\}$, where $0 < m < M$. The global physical boundary is defined in (1). $\partial\Omega_a$ (the right boundary of Ω_a) and $\partial\Omega_b$ (the left boundary of Ω_b) are called artificial internal boundary.

In order to approximate the time fractional equation on the two subdomains separately, the following iterative procedure can be performed. For each time step, the right hand side of (9) is calculated at first and the $U^n_{\Omega_i}$ is given as initial guess $U^{n-1}_{\Omega_i}$, where $i = a, b$. The better approximation of $U^n_{\Omega_i}$ can be obtained iteratively. During each iteration, which is inside of a time step, the time fractional equation is solved in the subdomain Ω_a, using the approximation of the previous iteration from Ω_b on $\partial\Omega_a$ as follows:

$$\begin{aligned} &{}_0^C D_t^\alpha u(x,t) + \mu u(x,t) \\ &\qquad = \frac{\partial^2 u(x,t)}{\partial x^2} + Kf(x,t), \quad (0 < \alpha < 1), \\ &u(x,0) = g(x), \quad x \in [0, x_R], \\ &u(x,0) = u^n_{b,m+1,\text{previous}}, \quad x \text{ on } \partial\Omega_a, \\ &u(0,t_n) = 0, \end{aligned} \tag{12}$$

where $u^n_{b,m+1,\text{previous}}$ stands for the previous solution of grid point p_{m+1} in subdomain Ω_b. The better approximation $U^n_{a,\text{new}}$ is obtained. $U^n_{a,\text{new}}$ is defined as $\{u^n_{a,1,\text{new}}, \ldots, u^n_{a,m,\text{new}}\}$. $U^n_{a,\text{previous}}$ is defined as $\{u^n_{a,1,\text{previous}}, \ldots, u^n_{a,m,\text{previous}}\}$. The definitions for $U^n_{b,\text{previous}}$ and $U^n_{a,\text{new}}$ are similar.

Then, we solve the time fractional equation within the subdomain of Ω_b, using the approximation of the previous iteration from Ω_a on $\partial\Omega_b$ as follows:

$$\begin{aligned} &{}_0^C D_t^\alpha u(x,t) + \mu u(x,t) \\ &\qquad = \frac{\partial^2 u(x,t)}{\partial x^2} + Kf(x,t), \quad (0 < \alpha < 1), \\ &u(x,0) = g(\mathrm{x}), \quad x = x_R \\ &u(x,0) = u^n_{a,m,\text{previous}}, \quad x \text{ on } \partial\Omega_b \\ &u(x_R,t_n) = 0, \end{aligned} \tag{13}$$

where $u^n_{a,m,\text{previous}}$ stands for the previous solution of grid point p_m in subdomain Ω_a.

The two local time fractional equations in Ω_a and Ω_b are connected by the artificial boundary condition. The artificial boundary condition on the internal boundary $\partial\Omega_a$ of subdomain Ω_a is provided by $u^n_{b,m+1,\text{previous}}$ from subdomain Ω_b, and vice versa. The approximation $u^n_{a,m,\text{previous}}$ and $u^n_{b,m+1,\text{previous}}$ may change until converged to the true solution. So, in an inner iteration of each time step, the two time fractional equations need to exchange two sets of data (send one and receive one) to update the artificial boundary conditions.

3.2. A Domain Decomposition Algorithm. Section 3.1 shows the procedure of DDM for time fractional equation with two subdomains. It is not hard to extend the method of Section 3.1 to more than two subdomains. The domain Ω can be decomposed into a set of P subdomains $\{\Omega_p\}_{p=1}^P$ with $\Omega = \cup_{p=1}^P \Omega_p$. For time step n, Ω_1 has one global boundary $x = 0$ and one artificial inner boundary $\partial\Omega_{1,b}$. Ω_P has one global boundary $x = x_R$ and one artificial inner boundary $\partial\Omega_{P,a}$. The Ω_p $(1 < p < P)$ has two artificial inner boundaries $\partial\Omega_{P,a}$ and $\partial\Omega_{p,b}$. $\Omega_p \cap \Omega_{p+1} \neq \Phi$ means that the neighboring subdomains have explicit overlap.

The iterative procedure for the time step $n + 1$ is similar to Section 3.1. The current iteration of Ω_p uses the data of previous iteration of its neighboring subdomains. Assuming M is divisible with P, the domain decomposition algorithm is shown in Algorithm 1.

In Algorithm 1, there are some fast algorithms to solve the tridiagonal matrix $A_{1\to I,3,p} V^3_{1\to I,p} = V^2_{1\to I,p}$, such as Thomas algorithm. ϵ is a threshold, such as 10^{-6}. The signal *localIteration* is used to count how many iterations are needed in each time step. The data exchange between neighboring iterations is shown in lines 19–22. From the view of computer science, lines 2–6, lines 7–30, and lines 31-32 are preprocessing procedure, numerical solver, and postprocessing procedure.

3.3. Analysis. The presented DD algorithm updates the artificial boundary condition in a Jacobi fashion, using approximation from all the relevant neighboring subdomains from the previous iteration for each time step. A subdomain only exchanges two sets of data for one artificial boundary with its neighbor. Therefore, the subdomain solved in Algorithm 1 can be carried out almost completely independently, thus making the method inherently as parallel as the Jacobi iteration. The DD algorithm keeps the good parallelism of Jacobi iteration but needs fewer inner iterations in each time step; see Section 4. Equation (9) can be regarded as approximation of a special integer order reaction-diffusion equation. The stability and convergence analysis of integer order reaction-diffusion equation can refer to Mathew's book [25].

4. Numerical Example

The following Caputo fractional reaction-diffusion equation [12] was considered, as shown in (14):

$$\begin{aligned} &{}_0^C D_t^\alpha u(x,t) + \mu u(x,t) \\ &\qquad = \frac{\partial^2 u(x,t)}{\partial x^2} + Kf(x,t), \quad (0 < \alpha < 1), \end{aligned}$$

(1) [h!] **input:** $M, P, x_R, T, \mu, K, \epsilon$ et al.
Output: U
(2) $I \leftarrow M/P$
(3) Allocate memory space $U^N_{I,P}, A_{I,3,P}, F^N_{I,P}, V^3_{I,P}$ et al.
(4) Init matrices U, A, F, b, r et al.
(5) Declare local variables δ_P, δ
(6) $V^{1\to 3}_{1\to I,1\to P} \leftarrow 0$
(7) Get $U^0_{1\to I,1\to P}$ with initial boundary.
(8) $totalIteration \leftarrow 0$
(9) **for** $n = 0$ **to** $N - 1$ *step by 1* **do**
(10) **for** $p = 1$ **to** P *step by 1* **do**
(11) $U^{n+1}_{1\to I,p} \leftarrow b_n U^0_{1\to I,p} + KF^{n+1}_{1\to I,p}$
(12) **for** $k = 1$ **to** n *step by 1* **do**
(13) $U^{n+1}_{1\to I,p} \leftarrow U^n_{1\to I,p} + U^k_{1\to I,p} r_{n-k}$
(14) $\delta \leftarrow 1.0$
(15) $localIteration \leftarrow 0$
(16) **while** $\delta > \epsilon$ **do**
(17) **for** $p = 1$ **to** P *step by 1* **do**
(18) $V^2_{1\to I,p} \leftarrow U^{n+1}_{1\to I,p}$
(19) **if** $p > 1$ **then**
(20) $V^2_{1,p} \leftarrow V^2_{1,p} + V^1_{I,p-1}/h^2$
(21) **if** $p < P$ **then**
(22) $V^2_{I,p} \leftarrow V^2_{I,p} + V^1_{I,p+1}/h^2$
(23) **for** $p = 1$ **to** P *step by 1* **do**
(24) solve $A_{1\to I,3,p} V^3_{1\to I,p} = V^2_{1\to I,p}$
(25) $\delta_p = \max\left\{\left|V^1_{i,p} - V^3_{i,p}\right|^I_{i=1}\right\}$
(26) $\delta = \max\left\{\delta, \delta_p\right\}$
(27) $localIteration \leftarrow localIteration + 1$
(28) $totalIteration \leftarrow totalIteration + count$
(29) **for** $p = 1$ **to** P *step by 1* **do**
(30) $U^{n+1}_{1\to I,p} \leftarrow V^3_{1\to I,p}$
(31) Output the information
(32) Free memory space

Algorithm 1: Domain decomposition algorithm for time fractional reaction-diffusion equation.

$$u(x,0) = 0, \quad x \in (0,2),$$
$$u(0,t) = u(2,t) = 0 \tag{14}$$

with $\mu = 1$, $K = 1$, and

$$f(x,t) = \frac{2}{\Gamma(2.3)} x(2-x)t^{1.3} + x(2-x)t^2 + 2t^2. \tag{15}$$

The exact solution of (14) is

$$u(x,t) = x(2-x)t^2. \tag{16}$$

With $\epsilon = 10^{-6}$, $T = 1.0$, $x_R = 2.0$, and $P = 3$, the comparison between exact solution and the presented DD algorithm is shown in Table 1. We can find that the DD algorithm compares well with the exact solution.

We can replace the DDM (lines 16–27 of Algorithm 1) with Jacobi method. The Jacobi method for a time step has the same parallelism with the DD algorithm. But the Jacobi method needs more iterations. With $\epsilon = 10^{-6}$ and $P = 3$, the comparison between Jacobi method and the presented DD algorithm is shown in Table 2. The sum of "count" (total

Table 1: Comparing exact solution and DD algorithm.

h	τ	Δ
2/10	1/10	8.36×10^{-3}
2/10	1/20	3.44×10^{-3}
2/61	1/61	7.84×10^{-4}
2/61	1/100	4.02×10^{-4}
2/100	1/300	6.10×10^{-5}

Table 2: Comparing Jacobi method and DDM.

h	τ	Jacobi method	DDM
2/10	1/10	741	250
2/10	1/20	1147	378
2/61	1/61	52423	3155
2/61	1/100	67164	4138
2/100	1/300	276243	11373

iterations) for all time steps is recorded. We can see that the DDM needs much less iterations than Jacobi method.

As a part of the future work, we would like to implement an efficient DDM for time fractional equations on parallel computer systems, for example, Tianhe-1A supercomputer [26].

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgments

This research work is supported by the National Natural Science Foundation of China under Grant no. 11175253. The authors would like to thank the anonymous reviewers for their helpful comments as well.

References

[1] A. Kilicman and Z. A. A. A. Zhour, "Kronecker operational matrices for fractional calculus and some applications," *Applied Mathematics and Computation*, vol. 187, no. 1, pp. 250–265, 2007.

[2] S. Zhai, X. Feng, and Z. Weng, "New high-order compact adi algorithms for 3D nonlinear time-fractional convection-diffusion equation," *Mathematical Problems in Engineering*, vol. 2013, Article ID 246025, 11 pages, 2013.

[3] A. Atangana and E. Alabaraoye, "Solving a system of fractional partial differential equations arising in the model of HIV infection of $CD4^+$ cells and attractor one-dimensional Keller-Segel equations," *Advances in Difference Equations*, vol. 2013, no. 1, article 94, pp. 1–14, 2013.

[4] Q. Liu, F. Liu, I. Turner, and V. Anh, "Numerical simulation for the 3D seep age flow with fractional derivatives in porous media," *IMA Journal of Applied Mathematics*, vol. 74, no. 2, pp. 201–229, 2009.

[5] P. Zhuang, F. Liu, V. Anh, and I. Turner, "Stability and convergence of an implicit numerical method for the non-linear fractional reaction-subdiffusion process," *IMA Journal of Applied Mathematics*, vol. 74, no. 5, pp. 645–667, 2009.

[6] A. Secer, "Approximate analytic solution of fractional heat-like and wave-like equations with variable coefficients using the differential transforms method," *Advances in Difference Equations*, vol. 2012, no. 1, article 198, pp. 1–10, 2012.

[7] A. Atangana and A. Secer, "The time-fractional coupled-Korteweg-de-Vries equations," *Abstract and Applied Analysis*, vol. 2013, Article ID 947986, 8 pages, 2013.

[8] J. Chen, F. Liu, I. Turner, and V. Anh, "The fundamental and numerical solutions of the Riesz space-fractional reaction-dispersion equation," *The ANZIAM Journal*, vol. 50, no. 1, pp. 45–57, 2008.

[9] X. Zhang, J. Liu, L. Wei, and C. Ma, "Finite element method for Grwünwald-Letnikov time-fractional partial differential equation," *Applicable Analysis*, vol. 92, no. 10, pp. 1–12, 2013.

[10] S. S. Ray, "A new approach for the application of Adomian decomposition method for the solution of fractional space diffusion equation with insulated ends," *Applied Mathematics and Computation*, vol. 202, no. 2, pp. 544–549, 2008.

[11] C. Li, F. Zeng, and F. Liu, "Spectral approximations to the fractional integral and derivative," *Fractional Calculus and Applied Analysis*, vol. 15, no. 3, pp. 383–406, 2012.

[12] J. Chen, "An implicit approximation for the Caputo fractional reaction dispersion equation," *Journal of Xiamen University*, vol. 46, no. 5, pp. 616–619, 2007 (Chinese).

[13] H. A. Schwarz, *Gesammelte Mathematische Abhandlungen*, AMS Bookstore, 1972.

[14] J. Xu and J. Zou, "Some nonoverlapping domain decomposition methods," *SIAM Review*, vol. 40, no. 4, pp. 857–914, 1998.

[15] T. F. Chan and T. P. Mathew, "Domain decomposition algorithms," *Acta Numerica*, vol. 3, pp. 61–143, 1994.

[16] C. Gong, W. Bao, G. Tang, B. Yang, and J. Liu, "An efficient parallel solution for Caputo fractional reaction-diffusion equation," *The Journal of Supercomputing*, 2014.

[17] C. Gong, J. Liu, L. Chi, H. Huang, J. Fang, and Z. Gong, "GPU accelerated simulations of 3D deterministic particle transport using discrete ordinates method," *Journal of Computational Physics*, vol. 230, no. 15, pp. 6010–6022, 2011.

[18] C. Gong, J. Liu, H. Huang, and Z. Gong, "Particle transport with unstructured grid on GPU," *Computer Physics Communications*, vol. 183, no. 3, pp. 588–593, 2012.

[19] X. Cao, Z. Mo, X. Liu, X. Xu, and A. Zhang, "Parallel implementation of fast multipole method based on JASMIN," *Science China Information Sciences*, vol. 54, no. 4, pp. 757–766, 2011.

[20] C. Gong, W. Bao, and G. Tang, "A parallel algorithm for the Riesz fractional reaction-diffusion equation with explicit finite difference method," *Fractional Calculus and Applied Analysis*, vol. 16, no. 3, pp. 654–669, 2013.

[21] C. Gong, W. Bao, G. Tang, Y. Jiang, and J. Liu, "A parallel algorithm for the two-dimensional time fractional diffusion equation with implicit difference method," *The Scientific World Journal*, vol. 2014, Article ID 219580, 8 pages, 2014.

[22] C. Gong, W. Bao, G. Tang, Y. Jiang, and J. Liu, "A domain decomposition method for time fractional reaction-diffusion equation," *The Scientific World Journal*. In press.

[23] I. Podlubny, *Fractional Differential Equations*, Academic Press, San Diego, Calif, USA, 1999.

[24] X. Cai, "Overlapping domain decomposition methods," in *Advanced Topics in Computational Partial Differential Equations*, vol. 33 of *Lecture Notes in Computational Science and Engineering*, pp. 57–95, Springer, Berlin, Germany, 2003.

[25] T. P. A. Mathew, *Domain Decomposition Methods for the Numerical Solution of Partial Differential Equations*, vol. 61 of *Lecture Notes in Computational Science and Engineering*, Springer, Berlin, Germany, 2008.

[26] X.-J. Yang, X.-K. Liao, K. Lu, Q.-F. Hu, J.-Q. Song, and J.-S. Su, "The TianHe-1A supercomputer: its hardware and software," *Journal of Computer Science and Technology*, vol. 26, no. 3, pp. 344–351, 2011.