

Research Article

An Efficient Iteration Method for Toeplitz-Plus-Band Triangular Systems Generated from Fractional Ordinary Differential Equation

Chunye Gong^{1,2,3} Weimin Bao,^{1,2} Guojian Tang,¹ Changwan Min,² and Jie Liu³

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

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

Received 2 January 2014; Accepted 27 May 2014; Published 12 June 2014

Academic Editor: Riccardo Caponetto

Copyright © 2014 Chunye Gong et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

It is time consuming to numerically solve fractional differential equations. The fractional ordinary differential equations may produce Toeplitz-plus-band triangular systems. An efficient iteration method for Toeplitz-plus-band triangular systems is presented with $O(M \log(M))$ computational complexity and $O(M)$ memory complexity in this paper, compared with the regular solution with $O(M^2)$ computational complexity and $O(M^2)$ memory complexity. M is the discrete grid points. Some methods such as matrix splitting, FFT, compress memory storage and adjustable matrix bandwidth are used in the presented solution. The experimental results show that the presented method compares well with the exact solution and is 4.25 times faster than the regular solution.

1. Introduction

Fractional differential equation (FDE) plays an important role in dynamical systems [1] and has more than 300 years of research history [2]. Many analytical solutions and numerical solutions [3–6] have been proposed for FDE, such as finite difference method [7, 8], finite element method [9], and spectral method [10, 11]. In recent times, interest in fractional ordinary differential equations (FODE) has increased [12–15]. The derivatives in the FODE are approximated by linear combinations of function values at the discrete grid points. Compared with integer ordinary differential equations, the FODE has nonlocal effect, which means a grid point may rely on the grid points far away from its position. And a grid point of the classical integer equations may only rely on its several neighboring grid points.

For integer order equations, the coefficient matrices are often sparse. Because of the nonlocal property of fractional differential operators, the numerical methods for fractional diffusion equations often generate dense or even full coefficient matrices [16]. This nonlocal property makes the computation of FODE and FDE much heavier than that of

the traditional integer equations. The short memory principle [17], parallel computing [18–21], fast Fourier transformation (FFT) [22, 23], multigrid method [24], and preconditioner technologies [25, 26] are used to overcome this heavy computation. Gong et al. presented many parallel algorithms for different FDEs on both traditional and heterogeneous parallel platforms [16, 18]. Diethelm [19] proposed a parallel second-order Adams-Bashforth-Moulton method for a FODE. Wang and Du [26] proposed a superfast-preconditioned iterative method for steady-state two-side space-fractional diffusion equations.

The fractional ordinary differential equations may produce Toeplitz-plus-band triangular systems. Toeplitz-plus-band systems were studied by professors Chan and Ng [27]. They considered the solutions of Hermitian Toeplitz-plus-band systems $(A_n + B_n)x = b$, where A_n are n -by- n Toeplitz matrices and B_n are n -by- n band matrices with bandwidth independent of n . A_n and B_n are both Hermitian matrix. The authors proved that if A_n is generated by a nonnegative piecewise continuous function and B_n is positive semidefinite, then there exists a band matrix C_n , with bandwidth

independent of n , such that the spectra of $C_n^{-1}(A_n + B_n)$ are uniformly bounded by a constant independent of n . The band preconditioner was developed for Hermitian Toeplitz systems [28]. The recursive blocked algorithms were proposed for triangular systems and the recursive algorithms lead to an automatic variable blocking that has the potential of matching the memory hierarchies of today's HPC (high performance computing) systems [29, 30].

This paper focuses on the fractional ordinary differential equation [13]:

$$u'(t) + a(t)D_t^\alpha u(t) + b(t)u(t) = f(t), \quad u(0) = 0, \quad (1)$$

where $0 < \alpha < 1$, $0 < t < T < +\infty$, $a(t) > 0$, and $b(t) > 0$. The fractional derivative is in the Caputo form [31].

Define $t_i = i\tau$ for $0 \leq i \leq M$, where M is a positive integer, and $\tau = T/M$ are step size. Assume u_i to be the numerical approximation to $u(t_i)$ and f_i the numerical approximation to $f(t_i)$. Using the Grünwald approximation, the finite difference scheme for (1) is shown as follows:

$$\frac{u_i - u_{i-1}}{\tau} + a_i \tau^{-\alpha} \sum_{k=0}^i w_k u_{i-k} + b_i u_i = f_i, \quad u_0 = 0, \quad (2)$$

where the normalized Grünwald weight w is defined by

$$w_0 = 1, \quad w_i = (-1)^i \frac{\alpha(\alpha-1)\cdots(\alpha-i+1)}{i!}, \quad (3)$$

$$i = 1, 2, 3, \dots$$

Equation (2) results in a linear system of equations

$$AU = F, \quad (4)$$

where $U = (u_1, u_2, \dots, u_M)^T$ and $F = (f_1, f_2, \dots, f_M)^T$. If $u_0 \neq 0$, the term should be included in F . $A = (a_{ij})_{M \times M}$ is the coefficient matrix. A is defined by

$$a_{i,j} = \begin{cases} 0 & \text{for } i < j \\ \frac{1}{\tau} + \frac{a_i w_0}{\tau^\alpha} + b_i & \text{for } i = j \\ -\frac{1}{\tau} + \frac{a_i w_1}{\tau^\alpha} & \text{for } i = j + 1 \\ \frac{a_i w_{i-j}}{\tau^\alpha} & \text{for } i > j + 1. \end{cases} \quad (5)$$

2. Method

2.1. Analysis. In a more explicit format, matrix A can be represented as

$$A = \begin{pmatrix} \frac{1}{\tau} + \frac{a_1 w_0}{\tau^\alpha} + b_1 & 0 & \cdots & 0 \\ -\frac{1}{\tau} + \frac{a_2 w_1}{\tau^\alpha} & \frac{1}{\tau} + \frac{a_2 w_0}{\tau^\alpha} + b_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ \frac{a_M w_{M-1}}{\tau^\alpha} & \frac{a_M w_{M-2}}{\tau^\alpha} & \cdots & \frac{1}{\tau} + \frac{a_M w_0}{\tau^\alpha} + b_M \end{pmatrix}. \quad (6)$$

```

input:  $M, F, A$ 
output:  $F$ 
(1) for  $i = 1$  to  $M$  by  $1$  do
(2)    $f_i \leftarrow f_i/a_{i,i}$ 
(3)   for  $j = i + 1$  to  $M$  by  $1$  do
(4)      $f_j \leftarrow f_j - a_{i,j}f_j$ 

```

ALGORITHM 1: Forward substitution for lower triangular matrix.

The linear system (4) can be solved with computational complexity $O(M^2)$, shown in Algorithm 1. The output F equals U .

From (6), we can see that A has some properties.

- (1) A is a low triangular, diagonal dominant matrix.
- (2) One has $|a_{i,j+1}| > |a_{i,j}|$ for $1 \leq i \leq M$, $j < i$. This property is determined by the normalized Grünwald weight w_i and is the mathematical background of short memory principle. This property means that for grid point p , if the distance of grid point p_1 is smaller than that of grid point p_2 , p_1 has more impact on p than p_2 .
- (3) If A is split into two matrices B and C , $A = B - C$. B is a banded matrix and the bandwidth (number of diagonals) $\eta > 2$. Matrix C can be factorized into a product $C = DT$. D is a diagonal matrix $\text{diag}\{a_1, a_2, \dots, a_M\}$. T is a Toeplitz matrix with $M - \eta$ nonzero diagonals on its left-bottom part.
- (4) The Toeplitz matrix T can be stored with $M - \eta$ memory space compared with $(M - \eta)(M - \eta + 1)/2$ for a general low triangular matrix with order $M - \eta$.

2.2. Efficient Iteration Method. Equation (4) evolves as follows:

$$(B - C)U = F \quad (7)$$

$$BU^{n+1} = CU^n + F \quad (8)$$

$$BU^{n+1} = DTU^n + F \quad (9)$$

$$BU^{n+1} = D(TU^n) + F. \quad (10)$$

So the linear algebra can be solved iteratively, shown in (10). Because D is a diagonal matrix, D keeps associative law and commutative law for matrix-matrix multiplication. The rate of convergence associated with (10) depends on the eigenvalues of the iteration matrix [32]:

$$H = B^{-1}DT = B^{-1}C. \quad (11)$$

Assuming error $e^{n+1} = U^{n+1} - U$ with U satisfies $AU = F$, then

$$e^{n+1} = B^{-1}Ce^n = He^n = H^n e^1 \quad (12)$$

```

input:  $M, \eta, F, A$ 
output:  $U$ 
(1)  $\epsilon \leftarrow 10^{-6}, \delta = 1.0$ 
(2) while  $\delta > \epsilon$  do
(3)   set  $vb_{1 \rightarrow M-\eta}$  with  $ua_{M-\eta \rightarrow 1}$ 
(4)    $V1 \leftarrow \text{FFT}(vb)$ 
(5)    $V2 \leftarrow \text{FFT}(va)$ 
(6)    $v \leftarrow V1 \odot V2$ 
(7)    $u_{1 \rightarrow M} \leftarrow f_{1 \rightarrow M}$ 
(8)   for  $i = 1$  to  $M - \eta$  by 1 do
(9)      $u_{\eta+i} \leftarrow u_{\eta+i} + v_{2(M-\eta)-i}$ 
(10)  for  $j = 1$  to  $M$  by 1 do
(11)     $u_j \leftarrow u_j d_j$ 
(12)    for  $i = j + 1$  to  $\min\{j + \eta - 1, M\}$  by 1 do
(13)       $u_i \leftarrow u_i - b_{i-j+1} u_j$ 
(14)     $\delta \leftarrow \max(|u_{1 \rightarrow M} - ua_{1 \rightarrow M}|)$ 
(15)   $ua_{1 \rightarrow M} \leftarrow ua_{1 \rightarrow M}$ 

```

ALGORITHM 2: The efficient iteration method for FODE.

with norm $\| * \|$ [32]:

$$\|e^{n+1}\| = \|G^n e^1\| = \|G^n\| \times \|e^1\|. \quad (13)$$

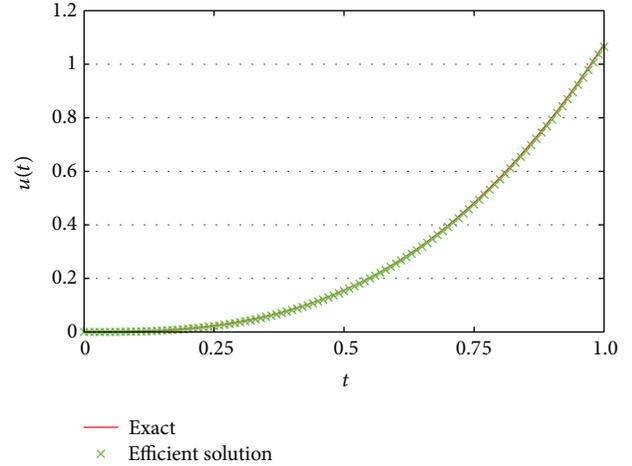
So the spectral radius of H ($\rho(H)$) determines the asymptotic behavior of H^n . From Theorem 11.2.1 of [32], we can conclude that if and only if $\rho(H) < 1$, (10) will converge to $A^{-1}F$. Generally speaking, the iteration is expected to work well with small $\rho(H)$.

Assume the bandwidth of matrix B is η and $V = D(TU^n) + F$. Solving $BU^{n+1} = V$ needs about $M\eta$ arithmetical operations. If η is near $\log_2 M$, there are about $M \log_2 M$ arithmetical operations with forward substitution. So the computational complexity of $BU^{n+1} = V$ is $O(M \log_2 M)$.

Assume $E1 = TU^n$, $E2 = DE_1$, and $V = E2 + F$. The computation of $E2$ and V needs M multiplications and M additions, respectively. Because T only has nonzero $M - \eta$ diagonals on its left-bottom part, only the front $M - \eta$ elements of U^n are effective for the multiplication TU^n . The back η elements of $E1$ are zero. So $E1 = TU^n$ can be regarded as a Toeplitz matrix vector multiplication $T_1 U_1^n$ with order $M - \eta$. It is well known that Toeplitz matrix vector multiplication with order $M - \eta$ can be finished with $O(M \log_2 M) = O((M - \eta) \log_2(M - \eta))$ operations [33]. The Toeplitz matrix vector multiplication $T_1 U_1^n$ can be computed by FFTs by first embedding T_1 into a $2(M - \eta)$ -by- $2(M - \eta)$ circulant matrix. The cost of circulant matrix vector multiplication is $O(2(M - \eta) \log_2(2(M - \eta)))$ by using FFTs of length $2(M - \eta)$.

So the cost of each iteration of (10) is $O(M \log_2 M)$. If A is a diagonal dominant matrix, we can expect (10) can converge with not too many iterations. The efficient iteration method is shown in Algorithm 2.

In Algorithm 2, $ua_{1 \rightarrow M}$ stands for the value of previous iteration and $u_{1 \rightarrow M}$ stands for the current iteration. $V1 \odot V2$ stands for $v_1 v_2$ with $1 \leq i \leq 2(M - \eta) - 1$. d_i equals the reciprocal of $a_{i,i}$ with $1 \leq i \leq M$. b_i stands for $a_{i,1}$ with $1 \leq i \leq \eta$. va and vb are $2(M - \eta) - 1$ arrays. $va_{M-\eta-i+1}$ equals $-a_{\eta+i,1}$.

FIGURE 1: Comparison of exact solution to the solution of the fast solution at time $t = 1.0$.

The value of η can affect the performance of Algorithm 2 shown in Table 1.

Algorithm 2 has five features/advantages compared to Algorithm 1.

- (1) Split the coefficient matrix and solve the triangular system iteratively.
- (2) Use FFT to compute matrix vector multiplication.
- (3) Precompute d_i .
- (4) Compress storage.
- (5) Adjust parameter η .

3. Numerical Example

The experiment platform is a laptop with Intel(R) Core (TM) i3-3110M CPU, 2 GB main memory, and Windows 7 operating system. The CPU clock frequency is 2.40 GHz. The code is developed with MATLAB R2012a and runs on default double precision floating point operations.

The following fractional ($\alpha = 0.8$) ordinary differential equation [13] was considered:

$$u'(t) + {}_0D_t^\alpha u(t) + (1+t)u(t) = f(t), \quad t > 0, \quad u(0) = 0, \quad (14)$$

where $f(t) = (14/\Gamma(3.8))t^{1.8} + (5/2)t^2 + (5/\Gamma(3.8))(1+t)t^{2.8}$.

The exact solution of (14) is

$$u(t) = \frac{5}{\Gamma(3.8)} t^{2.8}. \quad (15)$$

The efficient iteration method of Algorithm 2 compares well with the exact solution to the FODE in the test case of (14), shown in Figure 1. The τ is 1.0/100. The maximum absolute error is 9.78×10^{-3} . The difference between the efficient iteration method and the forward substitution Algorithm 1 is only 2.37×10^{-10} . The efficient iteration method and

TABLE 1: Impact of η .

Procedure	Big η	Small η
Iterations	Less	More
$BU^{n+1} = V$	Slow	Fast
$V = D(TU^n) + F$	Fast	Slow

TABLE 2: Performance comparison between regular solution and the presented efficient iteration method.

M	Presented method	Regular solution	Speedup
5×10^3	0.14	0.27	1.96
1×10^4	0.41	1.10	2.67
2×10^4	1.37	4.39	3.20
4×10^4	4.51	17.74	3.93
8×10^4	14.30	60.71	4.25

TABLE 3: Impact of η for $M = 4 \times 10^4$.

η	Number of iterations	Runtime
$1(\lceil \log_2 M \rceil + 1)$	93	7.95
$2(\lceil \log_2 M \rceil + 1)$	57	4.62
$3(\lceil \log_2 M \rceil + 1)$	43	5.61
$4(\lceil \log_2 M \rceil + 1)$	36	4.51
$5(\lceil \log_2 M \rceil + 1)$	31	4.93
$6(\lceil \log_2 M \rceil + 1)$	28	5.01

the regular forward substitution solution have no noticeable artifacts.

The performance comparison between regular forward substitution solution of Algorithm 1 and efficient iteration method of Algorithm 2 is shown in Table 2. Columns 2 and 3 of Table 2 are the runtime and the runtime is recorded in seconds. With $M = 8 \times 10^4$, the maximum speedup is 4.25. Because the speedup increases with M , the bigger M is, the higher the speedup that can be expected is. Because of the 2 GB memory limitation, the compress memory usage is also used in Algorithm 1.

The impact of η on the performance of Algorithm 2 is shown in Table 3. The runtime of the presented method varies with η . So η is a key parameter for the performance of Algorithm 2. In real fractional ordinary applications, the proper η should be chosen.

The presented iteration method should be regarded as an iteration method to solve not only the system generated from FODE but also the more general Toeplitz-plus-band triangular systems. The technology of parallel computing is very useful, but with less mathematical background. Parallel computing is attractive for fractional differential equations [34]. As a part of future work, first, we would like to parallelize the presented solution on shared memory or distributed memory systems. Second, accelerating the presented efficient iteration method on heterogeneous architecture [35–38] should also be interesting.

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 in part by the National Natural Science Foundation of China under Grant nos. 11175253 and 61170083, in part by the Specialized Research Fund for the Doctoral Program of Higher Education under Grant no. 20114307110001, and in part by the 973 Program of China under Grant no. 61312701001. The authors would like to thank the anonymous reviewers for their helpful comments as well.

References

- [1] R. Caponetto, G. Dongola, F. Pappalardo, and V. Tomasello, "Auto-tuning and fractional order controller implementation on hardware in the loop system," *Journal of Optimization Theory and Applications*, vol. 156, no. 1, pp. 141–152, 2013.
- [2] J. A. Tenreiro Machado, V. Kiryakova, and F. Mainardi, "A poster about the recent history of fractional calculus," *Fractional Calculus & Applied Analysis*, vol. 13, no. 3, pp. 329–334, 2010.
- [3] A. Atangana and A. Kilicman, "Analytical solutions of the space-time fractional derivative of advection dispersion equation," *Mathematical Problems in Engineering*, vol. 2013, Article ID 853127, 9 pages, 2013.
- [4] S. Yu. Lukashchuk, "An approximate solution method for ordinary fractional differential equations with the Riemann-Liouville fractional derivatives," *Communications in Nonlinear Science and Numerical Simulation*, vol. 19, no. 2, pp. 390–400, 2014.
- [5] D. Baleanu, K. Diethelm, E. Scalas, and J. J. Trujillo, *Fractional Calculus: Models and Numerical Methods*, vol. 3, World Scientific Publishing, 2012.
- [6] E. Gao, S. Song, and X. Zhang, "A numerical algorithm for solving a four-point nonlinear fractional integro-differential equations," *Mathematical Problems in Engineering*, vol. 2012, Article ID 329575, 11 pages, 2012.
- [7] 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.
- [8] J. Huang, Y. Tang, L. Vázquez, and J. Yang, "Two finite difference schemes for time fractional diffusion-wave equation," *Numerical Algorithms*, vol. 64, no. 4, pp. 707–720, 2013.
- [9] N. J. Ford, J. Xiao, and Y. Yan, "A finite element method for time fractional partial differential equations," *Fractional Calculus and Applied Analysis*, vol. 14, no. 3, pp. 454–474, 2011.
- [10] M. Zayernouri and G. E. Karniadakis, "Exponentially accurate spectral and spectral element methods for fractional ODEs," *Journal of Computational Physics*, vol. 257, pp. 460–480, 2014.
- [11] J. Huang, N. Nie, and Y. Tang, "A second order finite difference-spectral method for space fractional diffusion equations," *Science China: Mathematics*, vol. 57, no. 6, pp. 1303–1317, 2014.
- [12] J. Leszczynski and M. Ciesielski, "A numerical method for solution of ordinary differential equations of fractional order," in *Parallel Processing and Applied Mathematics*, pp. 695–702, Springer, 2006.

- [13] X. Cai and J. H. Chen, "Numerical method for fractional order ordinary differential equation," *Journal of Jimei University(Natural Science)*, vol. 12, no. 4, pp. 367–370, 2007.
- [14] R. Lin and F. Liu, "Fractional high order methods for the nonlinear fractional ordinary differential equation," *Nonlinear Analysis: Theory, Methods & Applications*, vol. 66, no. 4, pp. 856–869, 2007.
- [15] J. Cao and C. Xu, "A high order schema for the numerical solution of the fractional ordinary differential equations," *Journal of Computational Physics*, vol. 238, pp. 154–168, 2013.
- [16] 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.
- [17] W. Deng, "Short memory principle and a predictor-corrector approach for fractional differential equations," *Journal of Computational and Applied Mathematics*, vol. 206, no. 1, pp. 174–188, 2007.
- [18] J. Liu, C. Gong, W. Bao, G. Tang, and Y. Jiang, "Solving the caputo fractional reaction-diffusion equation on GPU," *Discrete Dynamics in Nature and Society*. In press.
- [19] K. Diethelm, "An efficient parallel algorithm for the numerical solution of fractional differential equations," *Fractional Calculus and Applied Analysis*, vol. 14, no. 3, pp. 475–490, 2011.
- [20] 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.
- [21] 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*, vol. 2014, Article ID 681707, 5 pages, 2014.
- [22] H. Wang and K. Wang, "An $O(N\log^2 N)$ alternating-direction finite difference method for two-dimensional fractional diffusion equations," *Journal of Computational Physics*, vol. 230, no. 21, pp. 7830–7839, 2011.
- [23] H. Wang, K. Wang, and T. Sircar, "A direct $O(N\log^2 N)$ finite difference method for fractional diffusion equations," *Journal of Computational Physics*, vol. 229, no. 21, pp. 8095–8104, 2010.
- [24] H.-K. Pang and H.-W. Sun, "Multigrid method for fractional diffusion equations," *Journal of Computational Physics*, vol. 231, no. 2, pp. 693–703, 2012.
- [25] S.-L. Lei and H.-W. Sun, "A circulant preconditioner for fractional diffusion equations," *Journal of Computational Physics*, vol. 242, pp. 715–725, 2013.
- [26] H. Wang and N. Du, "A superfast-preconditioned iterative method for steady-state space-fractional diffusion equations," *Journal of Computational Physics*, vol. 240, pp. 49–57, 2013.
- [27] R. H. Chan and K.-P. Ng, "Fast iterative solvers for Toeplitz-plus-band systems," *SIAM Journal on Scientific Computing*, vol. 14, no. 5, pp. 1013–1019, 1993.
- [28] R. H. Chan and P. T. P. Tang, "Fast band-Toeplitz preconditioners for Hermitian Toeplitz systems," *SIAM Journal on Scientific Computing*, vol. 15, no. 1, pp. 164–171, 1994.
- [29] I. Jonsson and B. Kågström, "Recursive blocked algorithm for solving triangular systems. I. One-sided and coupled Sylvester-type matrix equations," *ACM Transactions on Mathematical Software*, vol. 28, no. 4, pp. 392–415, 2002.
- [30] I. Jonsson and B. Kågström, "Recursive blocked algorithm for solving triangular systems. II. Two-sided and generalized Sylvester and Lyapunov matrix equations," *ACM Transactions on Mathematical Software*, vol. 28, no. 4, pp. 416–435, 2002.
- [31] I. Podlubny, *Fractional Differential Equations*, vol. 198, Academic Press, San Diego, Calif, USA, 1999.
- [32] G. H. Golub and C. F. Van Loan, *Matrix Computations*, vol. 3, Johns Hopkins University Press, Baltimore, Md, USA, 4th edition, 2013.
- [33] R. H.-F. Chan and X.-Q. Jin, *An introduction to iterative Toeplitz solvers*, vol. 5, Society for Industrial and Applied Mathematics (SIAM), Philadelphia, Pa, USA, 2007.
- [34] 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.
- [35] 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.
- [36] Q. Wu, C. Yang, T. Tang, and L. Xiao, "Exploiting hierarchy parallelism for molecular dynamics on a petascale heterogeneous system," *Journal of Parallel and Distributed Computing*, vol. 73, no. 12, pp. 1592–1604, 2013.
- [37] 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.
- [38] 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.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

