Parallel Conjugate Gradient Method Based on Spline Difference Method for the One-Dimensional Heat Equation

Aijia Ouyang[1], Wangdong Yang[1], Guangxue Yue[2,⋆], Tao Jiang[2], Xiaoyong Tang[3], and Xu Zhou[2]

[1] School of Information Science and Engineering, Hunan City University, Yiyang, Hunan 413000, China
[2] College of Mathematics, Physics & Information Engineering, Jiaxing University, Jiaxing, Zhejiang 314001, China
guangxueyue@163.com
[3] State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, Jiangsu 210093, China

Abstract. A new parallel conjugate gradient method (PCGM) implemented in GPU is presented to solve the one-dimensional heat equation with nonlocal boundary conditions in the paper. It is found that, at the interior nodal points, the method derived by using quartic spline is equivalent to the classical compact difference scheme which is unconditionally stable and the accuracy order is of $O(k^2 + h^4)$, where k and h are the time step length and the space step length, respectively. Both the accuracies of the new difference schemes at the two endpoints to deal with the nonlocal boundary conditions are of $O(k + h^4)$, which is much better than that of the classical finite difference. Finally, a numerical example is given to illustrate the efficiency of our method. Computational times are compared between high-end GPU and CPU systems with speedup of over 7.27 times when applied to one-dimensional heat equation.

Keywords: Quartic spline, Heat equation, Unconditionally stable, Derivative conditions.

1 Introduction

Graphics Processing Units (GPU) technique has been applied to engineering problems [1], shallow water simulations [2], finite difference scheme [3], finite element operator [4], incompressible flow [5], 3D flow simulations [6,7,8], computational fluid dynamics [9,10,11], et al.

We consider the one-dimensional nonclassical heat equation

$$\frac{\partial u}{\partial t} = \alpha \frac{\partial^2 u}{\partial x^2}, \quad 0 < x < 1, \quad 0 < t \le T, \tag{1}$$

⋆ Corresponding author.

with initial conditions

$$u(x,0) = f(x), \quad 0 \le x \le 1, \tag{2}$$

$$\frac{\partial u}{\partial x} u(1,t) = g(t), \quad 0 < t \le T, \tag{3}$$

and a nonlocal condition

$$\int_0^b u(x,t)dx = m(t), \quad 0 < t \le T, \quad 0 < b < 1. \tag{4}$$

In this problem, $f(x)$, $g(t)$ and $m(t)$ are given functions and α and b are constants. If $b = 1$, Eq. (4) can be differentiated as

$$m'(t) = \int_0^1 u_t dx = \int_0^1 \alpha u_{xx} dx = \alpha u_x(1,t) - \alpha u_x(0,t). \tag{5}$$

The derivation holds only when m and u are differentiable.

The one-dimensional heat equation is a well-known simple second order linear partial differential equation (PDE) [12]-[14]. PDEs like the heat equation very often arise in modeling problems in science and engineering. It is also used in financial mathematics in the modeling of options. For example, the Black-Scholes option pricing model's differential equation can be transformed into the heat equation [15].

Recently nonlocal boundary value problems have been considered in the literature [16]-[20]. Very recently, Dehghan [21] presented a new finite difference technique for solving the one-dimensional heat equation subject to the specification of mass, but the accuracy of their method is only first order. Caglar *et al.* [22] developed a third degree B-spines method to solve the heat equation (1-3) with the accuracy being $O(k^2 + h^2)$, and at the endpoints, the approximation order is second order only.

2 Quartic Splines and Interpolation Errors

Let Π be a uniform partition of $[0,1]$ as follows

$$0 = x_0 < x_1 < ...... < x_n = 1, \tag{6}$$

where $x_i = ih$, $h = 1/n$.

The quartic spline space on $[0,1]$ is defined as

$$S_4^3(\Pi) = \left\{ s \in C^3[0,1] : s \,\big|_{[x_{i-1},x_i]} \in P_4, i = 1(1)n \right\},$$

where P_d is the set of polynomials of degree at most d.

It is easy to know that $\dim S_4^3(\Pi) = n+4$. For any $s \in S_4^3(\Pi)$, the restriction of s in $[x_{i-1}, x_i]$ can be expressed as

$$s(x) = s_{i-1} + h s'_{i-1} t + h^2 s''_{i-1} \frac{t^2}{12}(6 - t^2) + h^2 s''_i \frac{t^4}{12} + h^3 s'''_{i-1} \frac{t^3}{12}(2 - t), \tag{7}$$

where $x = x_{i-1} + th$, $t \in [0,1]$, $s_i = s(x_i)$, $s'_i = s'(x_i)$, $s''_i = s''(x_i)$, $s'''_i = s'''(x_i)$, $i = 0(1)n$.

This leads to

$$s_i = s_{i-1} + hs'_{i-1} + \frac{5}{12}h^2 s''_{i-1} + \frac{1}{12}h^2 s''_i + \frac{1}{12}h^3 s'''_{i-1}, \tag{8}$$

$$s'_i = s'_{i-1} + \frac{2}{3}hs''_{i-1} + \frac{1}{3}hs''_i + \frac{1}{6}h^2 s'''_{i-1}, \tag{9}$$

$$s'''_i = -s'''_{i-1} + \frac{2}{h}\left[s''_i - s''_{i-1}\right], \tag{10}$$

for $i = 1(1)n$. Based on the above three equations, we can have

$$\frac{1}{12}s''_{i+1} + \frac{5}{6}s''_i + \frac{1}{12}s''_{i-1} = \frac{1}{h^2}\left[s_{i+1} - 2s_i + s_{i-1}\right], \quad i = 1(1)\overline{n-1}. \tag{11}$$

3 Quartic Spline Method

We consider the following heat equation (1) with the initial boundary value conditions (2- 3) and derivative boundary condition (5).

The domain $[0,1] \times [0,T]$ is divided into an $n \times m$ mesh with the spatial step size $h = 1/n$ in x direction and the time step size $k = T/m$, respectively.

Grid points (x_i, t_j) are defined by $x_i = ih$ and $t_j = jk$, where $i = 0(1)n$, $j = 0(1)m$, n and m are integers.

Let $s(x_i, t_j)$ and U_i^j be approximations to $u(x_i, t_j)$ and $M_i(t) = \frac{\partial^2 s(x,t)}{\partial x^2}|_{(x_i,t)}$, respectively.

Assume that $u(x,t)$ is the exact solution to Eq.(1). For any fixed t, let $s(x,t) \in S_4^3(\mathit{\Pi})$ be the quartic spline interpolating to $u(x,t)$ such that

$$s(x_i,t) = u(x_i,t), i = 0(1)n, \tag{12}$$

$$\frac{\partial^2 s}{\partial x^2}|(x_n,t) = \frac{\partial^2 u}{\partial x^2}|(x_n,t), \tag{13}$$

$$\frac{\partial s}{\partial x}|(x_0,t) + \frac{1}{12}h^2\frac{\partial^3 s}{\partial x^3}|(x_0,t) = \frac{\partial u}{\partial x}|(x_0,t) + \frac{1}{12}h^2\frac{\partial^3 u}{\partial x^3}|(x_0,t), \tag{14}$$

$$\frac{\partial s}{\partial x}|(x_n,t) - \frac{1}{12}h^2\frac{\partial^3 s}{\partial x^3}|(x_n,t) = \frac{\partial u}{\partial x}|(x_n,t) - \frac{1}{12}h^2\frac{\partial^3 u}{\partial x^3}|(x_n,t). \tag{15}$$

Then it follows from Theorem 1 that

$$\frac{\partial^2 u}{\partial x^2}|(x_i,t) = \frac{\partial^2 s}{\partial x^2}|(x_i,t) + O(h^4), \quad i = 1(1)\overline{n-1}. \tag{16}$$

For any fixed t, by using the Taylor series expansion, Eq.(1) can be discretized at the point (x_i, t_j) into

$$\begin{aligned} &\frac{u(x_i,t_{j+1}) - u(x_i,t_j)}{k} = \\ &\frac{1}{2}\alpha\left[\left(\frac{\partial^2 s(x,t)}{\partial x^2}\right)_i^j + \left(\frac{\partial^2 s(x,t)}{\partial x^2}\right)_i^{j+1}\right] + O(k^2 + h^4), \end{aligned} \tag{17}$$

where $i = 1(1)\overline{n-1},\ j = 0(1)\overline{m-1}$.

Substituting (17) into spline relation (11) and using Eq.(12), we conclude

$$\begin{aligned}&(1-6\alpha r)u(x_{i+1},t_{j+1}) + (10+12\alpha r)u(x_i,t_{j+1}) + (1-6\alpha r)u(x_{i-1},t_{j+1}) = \\ &(1+6\alpha r)u(x_{i+1},t_j) + (10-12\alpha r)u(x_i,t_j) + \\ &(1+6\alpha r)u(x_{i-1},t_j) + O(k^2+h^4),\end{aligned} \tag{18}$$

where $r = \alpha\frac{k}{h^2}$.

Neglecting the error term, we can get the following difference scheme

$$\begin{aligned}&(1-6\alpha r)U_{i+1}^{j+1} + (10+12\alpha r)U_i^{j+1} + (1-6\alpha r)U_{i-1}^{j+1} = \\ &(1+6\alpha r)U_{i+1}^{j} + (10-12\alpha r)U_i^{j} + (1+6\alpha r)U_{i-1}^{j}, \\ &i = 1(1)\overline{n-1},\ j = 0(1)m,\end{aligned} \tag{19}$$

with the accuracy being $O(k^2+h^4)$.

It is clearly that the difference scheme (19) identify to the classical fourth order compact difference scheme in [23], which is unconditionally stable.

4 The Difference Schemes on the Boundary

From the interpolation conditions (13-15) and Theorem 1, for each $t > 0$, we have

$$\begin{aligned}\frac{\partial u(x_0,t)}{\partial x} = &\frac{u(x_1,t)-u(x_0,t)}{h} - \frac{1}{12}h\frac{\partial^2 u(x_1,t)}{\partial x^2} \\ &- \frac{5}{12}h\frac{\partial^2 u(x_0,t)}{\partial x^2} - \frac{1}{12}h^2\frac{\partial^3 u(x_0,t)}{\partial x^3} + O(h^4),\end{aligned} \tag{20}$$

$$\begin{aligned}\frac{\partial u(x_n,t)}{\partial x} = &\frac{u(x_n,t)-u(x_{n-1},t)}{h} + \frac{5}{12}h\frac{\partial^2 u(x_n,t)}{\partial x^2} \\ &+ \frac{1}{12}h\frac{\partial^2 u(x_{n-1},t)}{\partial x^2} + \frac{1}{12}h^2\frac{\partial^3 u(x_n,t)}{\partial x^3} + O(h^4).\end{aligned} \tag{21}$$

Let $t = t_{j+1}$,it follows form Eqs.(1) and (20) that

$$\begin{aligned}u_x(x_0,t_{j+1}) = &\frac{u(x_1,t_{j+1})-u(x_0,t_{j+1})}{h} - \frac{1}{12\alpha}h\frac{u(x_1,t_{j+1})-u(x_1,t_j)}{k} \\ &- \frac{5}{12\alpha}h\frac{u(x_0,t_{j+1})-u(x_0,t_j)}{k} - \frac{1}{12}h^2\frac{\partial^3 u(x_0,t_{j+1})}{\partial x^3} \\ &+ O(k+h^4).\end{aligned} \tag{22}$$

From Eqs.(1),(3) and (5), we can get

$$\frac{\partial^3 u(0,t)}{\partial x^3} = \frac{1}{\alpha}\frac{\partial^2 u(0,t)}{\partial x \partial t} = \frac{1}{\alpha^2}m''(t) - \frac{1}{\alpha}g'(t). \tag{23}$$

Substituting (23) into (22) and neglecting the error term, we get the fourth-order difference scheme at $x = x_0$

$$\begin{aligned}&\left(1-\frac{1}{12r}\right)U_1^{j+1}-\left(1+\frac{5}{12r}\right)U_0^{j+1}+\frac{1}{12r}U_1^j+\frac{5}{12r}U_0^j=\\&hu_x(0,t_{j+1})+\frac{1}{12}h^3\left(\frac{1}{\alpha^2}m''(t_{j+1})-\frac{1}{\alpha}g'(t_{j+1})\right).\end{aligned}\tag{24}$$

Similarly, the difference scheme at the other end $x = x_n$ is

$$\begin{aligned}&\left(1+\frac{5}{12r}\right)U_n^{j+1}+\left(-1+\frac{1}{12r}\right)U_{n-1}^{j+1}-\frac{5}{12r}U_n^j+\frac{1}{12r}U_{n-1}^j=\\&hu_x(1,t_{j+1})-\frac{1}{12\alpha}h^3g'(t_{j+1}).\end{aligned}\tag{25}$$

5 Parallel Conjugate Gradient Method

5.1 GPU and CUDA

This section is dedicated to the description of the hardware architecture and software environment. A series of experiments are carried out on a dual-processor eight-core 2.3 GHz AMD Opteron 6134 machine with 8 GB main memory. The NVIDIA graphics cards GTX 465 is used to check the scalability of our approach.

The GTX 465 has 11 multiprocessors, 352 CUDA cores, 48 KB of shared memory per block, 607MHz processor clock, 1 GB GDDR5 RAM, 102.6GB/s memory bandwidth, and 802MHz memory clock. For the GPU card, the maximum number of resident blocks per multiprocessor is 8, the maximum number of threads per block is 1024, the maximum number of resident threads per multiprocessor is 1536, the total number of registers available per block is 32768. In software, the testing system is built on top of the Linux (Ubuntu 10.10) operating system, the NVIDIA CUDA Driver version 4.2, and GCC version 4.4.5.

5.2 Parallel Conjugate Gradient Algorithm

One of the fast iterative algorithms for a system of linear equations is the conjugate gradient method (CGM) [24]. The introduction of a preconditioner is applied to accelerate the convergence of the iterative process in the CGM. Preconditioning consists in the fact that the initial system of equations $Ax = b$ is replaced by the system

$$C^{-1}Ax = C^{-1}b. \tag{26}$$

for which the iterative method converges essentially faster. The condition of choosing the preconditioner C is as follows:

$$cond(\tilde{A}) << cond(A), \quad cond(\tilde{A}) = \frac{\tilde{\lambda}_{max}}{\tilde{\lambda}_{min}}, \quad cond(A) = \frac{\lambda_{max}}{\lambda_{min}}, \tag{27}$$

where $cond(A)$ and $cond(\tilde{A})$ are the condition numbers of the matrices A and $\tilde{A}$; λ_{min}, $\tilde{\lambda}_{min}$ and λ_{max}, $\tilde{\lambda}_{max}$ are the smallest and largest eigenvalues of the matrices A and $\tilde{A}$, respectively. For system of equations (37), the conjugate gradient method with preconditioner C has the form:

$$r^0 = b - Ax^0, \quad p^0 = C^{-1}r^0, \quad z^0 = p^0, \tag{28}$$

$$x^{k+1} = x^k + \alpha_k p^k, \quad \alpha_k = \frac{r^k, z^k}{Ap^k, p^k}, \quad r^{k+1} = r^k - \alpha_k Ap^k, \tag{29}$$

$$z^{k+1} = C^{-1}r^{k+1}, \quad p^{k+1} = z^{k+1} + \beta_k p^k, \quad \beta_k = \frac{r^{k+1}, z^{k+1}}{r^k, z^k}. \tag{30}$$

Where, the preconditioner is chosen by the incomplete LUdecomposition of the matrix A. The condition for stopping the CGM iterative process with preconditioner is

$$\frac{||Az^k - b||}{||b||} < \varepsilon \tag{31}$$

6 Experimental Results and Discuss

In this section, our new method is tested on the following problems from the literature [21]. Absolute errors of numerical solutions are calculated and compared with those obtained by using the three degree B-spline method [21,22].

In this example, we consider the following heat equation

$$f(x) = \cos\left(\frac{\pi}{2}x\right), \quad 0 < x < 1, \tag{32}$$

$$g(t) = -\exp\left(-\frac{\pi^2}{4}t\right), \quad 0 < x < 1, \tag{33}$$

$$m(t) = \frac{4}{\pi^2}\exp\left(-\frac{\pi^2}{4}t\right), \quad 0 < x < 1, \tag{34}$$

with

$$u(x,t) = \exp\left(-\frac{\pi^2}{4}t\right)\cos\left(\frac{\pi}{2}x\right) \tag{35}$$

as its analytical solution.

At first, the results with $h = k = 0.01,\ 0.05,\ 0.025,\ 0.001$, using the present method discussed in Section 3 and 4, are shown in Table 1. We present the relative error $\text{abs}((U_i^j - u(x_i,t_j))/\text{abs}(u(x_i,t_j))$ for $u(0.5, 0.1)$ using the present method. The numerical results are compared with those results obtained by the methods in [18,21,25,26], see Table 1. It is shown from Table 1 that the numerical

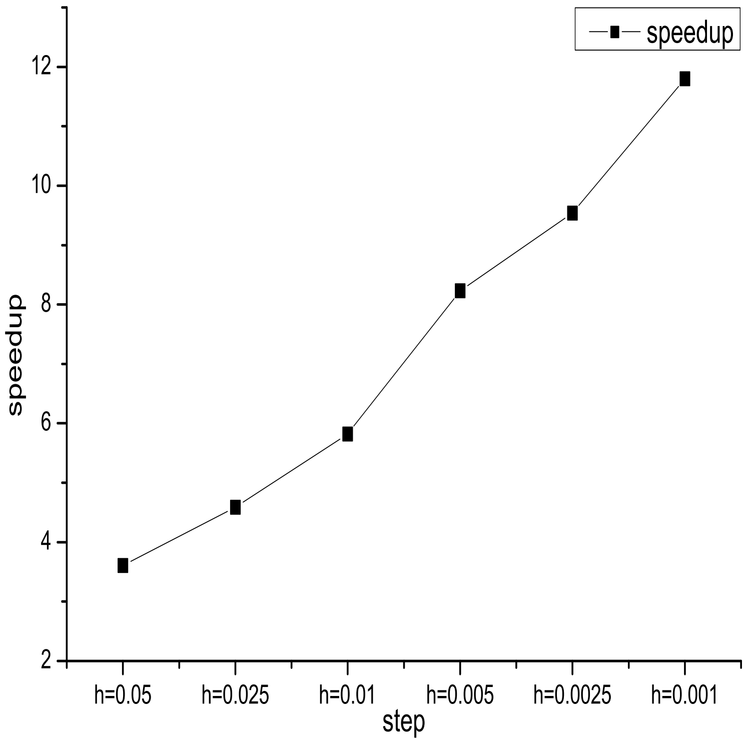

Fig. 1. The speedup of PCGM at various spatial lengths

Table 1. Relative errors at various spatial lengths

Step	Implicit[25]	Galerkin[18]	Keller[26]	RKC[27]	Saulyev[21]	CGM
$h = 0.0500$	9.1×10^{-3}	9.9×10^{-2}	9.4×10^{-2}	9.8×10^{-2}	9.6×10^{-3}	1.2×10^{-4}
$h = 0.0250$	2.3×10^{-3}	3.0×10^{-2}	2.4×10^{-2}	3.7×10^{-2}	2.5×10^{-3}	3.0×10^{-5}
$h = 0.0100$	3.8×10^{-4}	4.9×10^{-3}	4.1×10^{-3}	6.1×10^{-3}	3.9×10^{-4}	4.4×10^{-6}
$h = 0.0050$	9.4×10^{-5}	1.2×10^{-3}	1.0×10^{-3}	1.5×10^{-3}	9.6×10^{-5}	1.1×10^{-6}
$h = 0.0025$	2.3×10^{-5}	3.1×10^{-4}	2.5×10^{-4}	3.5×10^{-4}	2.5×10^{-5}	2.8×10^{-7}
$h = 0.0010$	4.1×10^{-6}	5.0×10^{-5}	4.0×10^{-5}	6.0×10^{-5}	4.3×10^{-6}	4.5×10^{-8}

results of the present method are much better than that in [18,25,26,21]. In [21], Dehghan solved the example by using the Saulyev I's technique with the accuracy being first-order with respect to the space and time variables, and the numerical integration trapezoidal rule is used to approximation the integral condition in (4). However, in this paper, we present a new method to deal with the nonlocal boundary conditions by using quartic splines [28], and the method is very easier than the numerical integration trapezoidal rule[21].

Table 2. The Maximum absolute errors at various step h and k

steps	method[22]	CGM
$h = k = 1/20$	6.1575×10^{-3}	2.7824×10^{-4}
$h = k = 1/40$	1.5751×10^{-3}	6.2456×10^{-5}
$h = 1/40,\ k = 1/60$	9.5842×10^{-4}	3.6223×10^{-5}

Table 3. The speedup of PCGM at various step h

spatial length	C(ms)	C-CUDA(ms)	speedup
$h = 0.0500$	3701.24	1025.27	3.61
$h = 0.0250$	7809.62	1701.44	4.59
$h = 0.0100$	18899.27	3247.30	5.82
$h = 0.0050$	36853.58	4477.96	8.23
$h = 0.0025$	76655.45	8035.16	9.54
$h = 0.0010$	200070.71	16955.15	11.80

7 Conclusion

In this paper, a new parallel method by using quartic splines functions is applied to the one-dimensional heat equation with boundary integral conditions replacing standard boundary conditions. This technique worked well for one-dimensional diffusion with an integral condition which can be transferred to the derivative boundary conditions. Various approaches reported for the numerical solution of diffusion subject to the specification of mass deals with the parabolic partial differential equations whose usual boundary specifications are replaced with integral boundary value conditions. However, no special method to deal with the integral boundary value conditions. In this paper, we construct a special method to approximate the derivative boundary conditions, and the accuracy of this method is fourth order at the end points. The numerical results obtained by using the quartic spline method described in this article give acceptable results and suggests convergence to the exact solution when k and h goes to zero. Computational times are compared between high-end GPU and CPU systems with speedup of over 7.27 times when applied to one-dimensional heat equation.

Acknowledgements. This paper was partially funded by the Key Program of National Natural Science Foundation of China (Grant No.61133005), and the National Natural Science Foundation of China (Grant Nos.90715029, 61070057, 60603053, 61370098 and 61202109), Key Projects in the National Science & Technology Pillar Program the Cultivation Fund of the Key Scientific and Technical Innovation Project, Ministry of Education of China (Grant No.708066), the Ph.D. Programs Foundation of Ministry of Education of China (Grant No.20100161110019), the Program for New Century Excellent Talents in University (Grant No.NCET-08-0177), A Project Supported by the Science and Technology Research Foundation of Hunan Province (Grant No.2013GK3082), and A Project Supported by Scientific Research Fund of Hunan Provincial Education Department (Grant No.12A062 and No.13C333), A Project supported by the Research Foundation of Education Bureau of Hunan Province, China (Grant No.11C0573); Project supported by the Natural Science Foundation of Zhejiang pvovince, China (Grant No.LY12F02019, No.LY12F02047), Project supported by the Science and Technology Foundation of Jiaxing City, China (Grant No. 2012AY1027).

References

1. Kuo, F.-A., Smith, M.R., Hsieh, C.-W., Chou, C.-Y., Wu, J.-S.: GPU acceleration for general conservation equations and its application to several engineering problems. Computers & Fluids 45(1), 147–154 (2011)
2. Brodtkorb, A.R., Stra, M.L., Altinakar, M.: Efficient shallow water simulations on GPUs: Implementation, visualization, verification, and validation. Computers & Fluids 55(15), 1–12 (2012)
3. Tutkun, B., Edis, F.O.: A GPU application for high-order compact finite difference scheme. Computers & Fluids 55(15), 29–35 (2012)
4. Kronbichler, M., Kormann, K.: A generic interface for parallel cell-based finite element operator application. Computers & Fluids 63(30), 135–147 (2012)
5. Storti, M.A., Paz, R.R., Dalcin, L.D., Costarelli, S.D., Idelsohn, S.R.: A FFT preconditioning technique for the solution of incompressible flow on GPUs. Computers & Fluids 74(30), 44–57 (2013)
6. Marrone, S., Bouscasse, B., Colagrossi, A., Antuono, M.: Study of ship wave breaking patterns using 3D parallel SPH simulations. Computers & Fluids 69(30), 54–66 (2012)
7. Borazjani, I., Ge, L., Le, T.: Fotis Sotiropoulos, A parallel overset-curvilinear-immersed boundary framework for simulating complex 3D incompressible flows. Computers & Fluids 1(77), 76–96 (2013)
8. Xiong, Q., Li, B., Xu, J., Wang, X., Wang, L., Ge, W.: Efficient 3D DNS of gas-solid flows on Fermi GPGPU. Computers & Fluids 70(30), 86–94 (2012)
9. Appleyard, J., Drikakis, D.: Higher-order CFD and interface tracking methods on highly-Parallel MPI and GPU systems. Computers & Fluids 46(1), 101–105 (2011)
10. Soni, K., Chandar, D.D.J., Sitaraman, J.: Development of an overset grid computational fluid dynamics solver on graphical processing units. Computers & Fluids 58(15), 1–14 (2012)
11. Lefebvre, M., Guillen, P., Le Gouez, J.-M., Basdevant, C.: Optimizing 2D and 3D structured Euler CFD solvers on Graphical Processing Units. Computers & Fluids 70(30), 136–147 (2012)
12. Carslaw, H.S., Jaeger, J.C.: Conduction of heat in solids. Oxford University Press, Oxford (1959)
13. Widder, D.V.: The heat equation. Academic Press, London (1975)
14. Cannon, J.R.: The one-dimensional heat equation. Cambridge University Press, Cambridge (1984)
15. Wilmott, P., Howison, S., Dewynne, J.: The mathematics of financial derivatives: a student introduction. Cambridge University Press, Cambridge (1995)
16. Ang, W.T.: A method of solution for the one-dimensional heat equation subject to a nonlocal condition. SEA Bull. Math. 26(2), 197–203 (2002)

17. Boutayeb, A., Chetouani, A.: Global extrapolation of numerical methods for solving a parabolic equation with nonlocal boundary conditions. Int. J. Comput. Math. 80, 789–797 (2003)
18. Cannon, J.R., Matheson, A.L.: A numerical procedure for diffusion subject to the specification of mass. Int. J. Eng. Sci. 31(3), 347–355 (1993)
19. Saadatmandi, A., Razzaghi, M.: A Tau method approximation for the diffusion equation with nonlocal boundary conditions. Int. J. Comput. Math. 81(11), 1427–1432 (2004)
20. Wang, S., Lin, Y.: A numerical method for the diffusion equation with nonlocal boundary specifications. Int. J. Eng. Sci. 28(6), 543–546 (1990)
21. Dehghan, M.: The one-dimensional heat equation subject to a boundary intergral specification. Chaos, Solitons Fractals 32(2), 661–675 (2007)
22. Caglar, H., Ozer, M., Caglar, N.: The numerical solution of the one-dimensional heat equation by using third degree B-spline functions. Chaos, Solitons Fractals 38(4), 1197–1201 (2008)
23. Thomas, J.W.: Numericlal Partial Differential Equations: Finite Difference Methods. Springer, New York (1995)
24. Akimova, E.N., Belousov, D.V.: Parallel algorithms for solving linear systems with block-tridiagonal matrices on multi-core CPU with GPU. Journal of Computational Science 3(6), 445–449 (2012)
25. Cannon, J.R., Prez-Esteva, S., van, J.: A Galerkin prodedure for the diffusion equation subject to the specification of mass. SIAM J. Num. Anal. 24, 499–515 (1987)
26. Ewing, R.E., Lin, T.: A class of parameter estimation techniques of fluid flow in porous meida. Adv. Water Resour. 14(2), 89–97 (1991)
27. Makarov, V.L., Kulyev, D.T.: Solution of a boundary value problem for a quai-linear parabolic equation with nonclassical boundary conditions. Different. Equat. 21, 296–305 (1985)
28. Liu, H.-W., Liu, L.-B., Chen, Y.: A semi-discretization method based on quartic splines for solving one-space-dimensional hyperbolic equations. Applied Mathematics and Computation 210(2), 508–514 (2009)