Radial basis functions and FDM for solving fractional diffusion-wave equation

Z. Avazzadeh*, V. R. Hosseini and W. Chen

State Key Laboratory of Hydrology-Water Resources and Hydraulic Engineering, College of Mechanics and Materials, Hohai University, Nanjing 210098, China
E-mail: z.avazzadeh@yahoo.com

Abstract

In this work, we apply the radial basis functions for solving the time fractional diffusion-wave equation defined by Caputo sense for \( 1 < \alpha \leq 2 \). The problem is discretized in the time direction based on finite difference scheme and is continuously approximated by using the radial basis functions in the space direction which achieves the semi-discrete solution. Numerical results show the accuracy and efficiency of the presented method.

Keywords: Diffusion-wave equation; fractional derivative; radial basis functions; finite difference scheme

1. Introduction

In recent years, fractional calculus has been implemented to express some phenomena in physics and engineering. Also, fractional integral and derivative have been successful to describe many events in fluid mechanics, viscoealsticity, chemical physics, electricity, finance, control theory, biomedical engineering, heat conduction, diffusion problems and other sciences (Kilbas et al., 2006; Podlubny, 1999). Fractional partial differential equations (FPDEs) particularly space- and time-fractional equations, have been widely studied to construe the existence of solution and validity of these problems (Li and Xu, 2009; Zhao et al., Zhuang et al. 2011). In addition, finding the reliable and powerful numerical and analytical methods for solving FPDEs have been the focus in two last decades. According to the mathematical literature, fractional partial differential equations have been developed in many various problems in science and engineering as the Schröinger, telegraph, diffusion and diffusion-wave fractional equation (Li and Xu, 2009; Chen et al., 2010a; Li et al., 2011; Liu et al., 2006; Mohebbi et al., 2013; Zhoa and Li, 2012).

In 2009, Wen et al. were pioneers in using the Kansa method for solving the fractional diffusion equation (Chen et al., 2010b) After that the method was expanded for solving the other fractional equations (Mohebbi et al, 2013; Piret and Hanert, 2013; Hosseini et al., 2014; Gu et al., 2010).

*Corresponding author
Received: 13 June 2013 / Accepted: 16 April 2014
obtain the continuous solution with respect to $x$ is investigated. In Section 4, some numerical examples are demonstrated which confirm the accuracy and applicability of the method. The last section includes some other features of the presented method, conclusion and further ideas for future work.

2. Basic Definitions

2.1. Fractional derivative

Definition: The Caputo fractional derivative operator of order $0 < \alpha \leq k$, of a function $F(x)$ is defined as

$$D^\alpha F(x) = \frac{1}{\Gamma(k-\alpha)} \int_0^x (x-\xi)^{k-\alpha-1} F^{(k)}(\xi) d\xi, \quad k-1 < \alpha < k, \quad \alpha = k. \tag{1}$$

More properties of the fractional Caputo derivative can be found in (Kilbas et al., 2006; Podlubny, 1999). Also, the further information about fractional calculus and another definitions of fractional derivatives, one can consult the mentioned references.

2.2. Radial basis functions

Considering a finite set of interpolation points $\mathcal{X} = \{x_1, x_2, \ldots, x_N\}$. So the interpolant of $u$ is constructed in the following form

$$(Su)(x) = \sum_{i=1}^N \lambda_i \phi(||x - x_i||) + p(x), \quad x \in \mathbb{R}^d, \tag{2}$$

where $d$ is the dimension, $|| \cdot ||$ is the Euclidean norm and $\phi(|| \cdot ||)$ is a radial function (Buhmann, 2003; Cheney and Light, 1999). Also, $p(x)$ is a linear combination of polynomials on $\mathbb{R}^d$ of total degree at most $m - 1$ as follows

$$p(x) = \sum_{j=-N+1}^{N+l} \lambda_j q_j(x), \quad l = \left\lfloor \frac{m + d - 1}{d} \right\rfloor. \tag{3}$$

Moreover, the interpolant $Su$ and additional conditions must be determined to satisfy the system

$$\begin{cases} (Su)(x_i) = u(x_i), & i = 1, 2, \ldots, N, \\ \sum_{i=1}^N \lambda_i q_j(x_i) = 0, & \text{for all } q_j \in \Pi_{m-1}^d, \end{cases} \tag{4}$$

where $\Pi_{m-1}^d$ denotes the space of all polynomials on $\mathbb{R}^d$ of total degree at most $m - 1$. The generalized thin plate splines (GTPS) are defined as follows:

$$\phi(||x - x_i||) = \phi(r_i) = r_i^{2m} \log(r_i), \quad i = 1, 2, 3, \ldots, \quad m = 1, 2, 3, \ldots, \tag{5}$$

2.3. Description of the Method

Consider the following time-fractional diffusion-wave equation of order $\alpha (1 < \alpha \leq 2)$

$$\frac{\partial^\alpha u(x,t)}{\partial t^\alpha} + \beta u(x,t) = \gamma \frac{\partial^2 u(x,t)}{\partial x^2} + f(x,t), \quad a \leq x \leq b, \quad 0 \leq t \leq T, \tag{6}$$

with the initial conditions

$$u(x,0) = g_1(x), \quad u_t(x,0) = g_2(x), \quad a \leq x \leq b, \tag{7}$$

and the boundary conditions

$$u(a,t) = h_1(t), \quad u(b,t) = h_2(t), \quad t \geq 0, \tag{8}$$

where $a, b, \alpha, g_1(x), g_2(x), h_1(t)$ and $h_2(t)$ are given and $\frac{\partial^\alpha u(x,t)}{\partial t^\alpha}$ represents the Caputo fractional derivative and $\gamma$ and $\beta$ are the given constants. According to Eq. (1), $\frac{\partial^\alpha u(x,t)}{\partial t^\alpha}$ can be written as follows:
\[
\frac{\partial^n u(x,t)}{\partial t^n} = \frac{1}{\Gamma(3 - \alpha)} \int_{0}^{t} \frac{\partial^{n-1} u(x,\xi)}{\partial \xi^{n-1}} (t - \xi)^{\alpha-1} d\xi, \quad 1 < \alpha < 2, \quad (9)
\]

In order to discretize the problem for \((1 < \alpha < 2)\) in time direction, we substitute \(I^{n+1}\) into Eq. (9), then the integrals can be reformed as

\[
\frac{\partial^n u(x,t^{n+1})}{\partial t^n} = \frac{1}{\Gamma(3 - \alpha)} \int_{0}^{t^{n+1}} \frac{\partial^{n-1} u(x,\xi)}{\partial \xi^{n-1}} (t^{n+1} - \xi)^{\alpha-1} d\xi,
\]

\[
= \frac{1}{\Gamma(3 - \alpha)} \left[ \sum_{k=0}^{n} \frac{\partial^{n+k} u(x,r)}{\partial r^{n+k}} \right] (t^{n+1} - r)^{\alpha-1} d\xi,
\]

where \(t^{0} = 0\), \(t^{n+1} = t^{n} + \Delta t\), \(n = 0,1,2,...,N\).

Approximation of the second order derivative due to the forward finite difference formula is defined as

\[
\frac{\partial^2 u(x,\xi)}{\partial \xi^2} \approx u(x,t^{n+1}) - 2u(x,t^{n}) + u(x,t^{n-1}) \frac{\partial}{\partial \xi}, \quad (11)
\]

Replacement of Eq. (11) into Eq. (10) gives,

\[
\frac{\partial^n u(x,t^{n+1})}{\partial t^n} = \frac{1}{\Gamma(2 - \alpha)} \int_{0}^{t^{n+1}} \frac{\partial^{n+k} u(x,\xi)}{\partial \xi^{n+k}} (t^{n+1} - \xi)^{\alpha-1} d\xi,
\]

\[
= \frac{1}{\Gamma(2 - \alpha)} \left[ \sum_{k=0}^{n} \frac{\partial^{n+k} u(x,r)}{\partial r^{n+k}} \right] (t^{n+1} - r)^{\alpha-1} d\xi, \quad (12)
\]

where \(u^{k} = u(x,t^{k})\), \(k = 0,1,...,M\). By considering \(t^{n+1} - \xi = r\), the integral can be obtained as follows:

\[
\int_{0}^{t^{n+1}} (t^{n+1} - \xi)^{\alpha-1} d\xi = \frac{-1}{(2 - \alpha)} \left[ \left( \frac{r^{2 - \alpha}}{2 - \alpha} \right)^{n + k} \right], \quad (13)
\]

Rearrangement of Eq. (12) and assumption \(b_{k} = (k + 1)^{2 - \alpha} - (k)^{2 - \alpha}\) lead to

\[
\frac{\partial^n u(x,t^{n+1})}{\partial t^n} = \frac{\partial}{\partial t^{n+1}} \left[ \sum_{k=0}^{n} b_{k} (u^{n+k} - 2u^{n+k} + u^{n}) \right] (t^{n+1} - r)^{\alpha-1} d\xi
\]

\[
= a_{0} \left[ u^{n+k} - 2u + u^{n+k} + \sum_{k=0}^{n} b_{k} (u^{n+k} - 2u^{n+k} + u^{n}) \right] \frac{\partial}{\partial t^{n+1}}, \quad (14)
\]

where \(a_{0} = \frac{\partial t^{n+1}}{\Gamma(3 - \alpha)}\) and \(n = 0,1,2,...,M\).

We note that the Eq. (6) in \(t = t^{n+1}\) due to \(\theta\)-weighted finite difference formulation is as follows:

\[
\frac{\partial^n u(x,t^{n+1})}{\partial t^n} + \beta u^{n+1} = \gamma \theta \mathcal{N}_{u}^{2} u^{n+1} + (1 - \theta) \nabla^{2} u^{n} + f^{n+1}, \quad (15)
\]

where \(0 \leq \theta \leq 1\) is a constant, \(\nabla^{2} u^{n} = \nabla^{2} u(x,t^{n})\) and \(f(x,t^{n+1}) = f^{n+1}\).

Now, we substitute Eq. (14) into Eq. (15) and obtain

\[
\{a_{0} + \beta - \gamma \theta \mathcal{N}_{u}^{2}\} u^{n+1} = \left\{2a_{0} - \gamma(1 - \theta) \nabla^{2}\right\} u^{n} - a_{0} u^{n+1} + a_{0} \sum_{k=0}^{n} b_{k} (u^{n+k} - 2u^{n+k} + u^{n+k-1}) + f^{n+1}, \quad (16)
\]

where \(n = 0,1,2,...,M\). Note that \(u^{n+1}\) will be observed when \(n = 0\) or \(k = n\). So, we use the initial conditions to approximate \(u^{n+1}\) as follows

\[
u_{n+1} = \frac{u^{n+1} - u^{n}}{2\Delta t}, \quad (16)
\]

which concludes \(u^{n+1} = u^{n+1} - 2\Delta t \nu_{0}^{n+1}\) or \(u^{n+1} = u^{n+1} - 2\Delta t \nu_{1}^{n+1}\). Therefore, the obtained recursive equation can be rewritten as

\[
\left\{2a_{0} + \beta - \gamma \theta \mathcal{N}_{u}^{2}\right\} u^{n+1} = \left\{2a_{0} - \gamma(1 - \theta) \nabla^{2}\right\} u^{n} - a_{0} u^{n+1} + a_{0} \sum_{k=0}^{n} b_{k} (u^{n+k} - 2u^{n+k} + u^{n+k-1}) + f^{n+1}, \quad (17)
\]

and

\[
\{a_{0} + \beta - \gamma \theta \mathcal{N}_{u}^{2}\} u^{n+1} = \left\{2a_{0} - \gamma(1 - \theta) \nabla^{2}\right\} u^{n} - a_{0} u^{n+1} + a_{0} \sum_{k=0}^{n} b_{k} (u^{n+k} - 2u^{n+k} + u^{n+k-1}) + a_{0} \nu_{n}^{n+1} (u^{n+1} - u^{n} - \Delta t \nu_{0}^{n} + f^{n+1}), \quad (18)
\]

at \(n = 0\) and \(n \geq 1\), respectively. Now we approximate the \(u^{n}(x)\) by radial basis functions as follows:

\[
u^{n}(x) = \sum_{j=1}^{N} \lambda_{j} \phi(r_{j}) + \lambda_{nx+1} x + \lambda_{nx+2}, \quad (19)
\]

where \(\lambda_{1},...,\lambda_{N},\lambda_{nx+1},\lambda_{nx+2}\) are unknowns. So, we consider \(N\) collocation points to obtain the
values of coefficients $\lambda_k$, \( k = 1, 2, ..., N+2 \) in the interpolant of $u^n(x)$ as

\[
u^{n+1}_i = u^{n+1}(x_i) = \sum_{j=1}^{N} \lambda_i^{n+1} \phi_j(r_j) + \lambda_{N+2}^{n+1} x_j \quad \text{for} \quad i = 1, 2, ..., N, \tag{20}\]

where $r_j = \|x_i - x_j\|$ when $\| . \|$ is the Euclidean norm, and $\phi(||.||)$ is a radial function. The additional conditions can be described as

\[
\sum_{j=1}^{N} \lambda_i^{n+1} = \sum_{j=1}^{N} \lambda_j^{n+1} x_j = 0. \tag{21}\]

By considering Eq. (20) together with Eq. (21) in a matrix form, we obtain

\[
[u]^{n+1} = A[\lambda]^{n+1}, \tag{22}\]

where $[u]^{n+1} = [u_1^{n+1}, u_2^{n+1}, ..., u_N^{n+1}]^T$ and $[\lambda]^{n+1} = [\lambda_1^{n+1}, \lambda_2^{n+1}, ..., \lambda_N^{n+1}]^T$. $A$ is an $(N+2) \times (N+2)$ matrix given by

\[
A = [a_{ij}] = \begin{bmatrix}
\phi_1 & \cdots & \phi_j & \cdots & \phi_N & x_1 & 1 \\
\vdots & \ddots & \vdots & \ddots & \vdots & \vdots & \vdots \\
\phi_1 & \cdots & \phi_j & \cdots & \phi_N & x_N & 1 \\
x_1 & \cdots & x_j & \cdots & x_N & 0 & 0 \\
1 & \cdots & 1 & \cdots & 1 & 0 & 0
\end{bmatrix}. \tag{23}\]

In addition, discretization of $\nabla^2 u$ is as follows:

\[
\nabla^2 u(x) = \sum_{j=1}^{N} \frac{\partial^2 \phi_j(r_j)}{\partial x^2} = \sum_{j=1}^{N} \psi_j(r_j), \tag{24}\]

where $r_j = \|x - x_j\|$. Thus, substituting the collocation points gives

\[
\nabla^2 u(x_i) = \sum_{j=1}^{N} \psi_j(r_j), \quad i = 2, ..., N-1. \tag{25}\]

Reconstruction of Eq. (17) in the matrix form can be illustrated as follows

\[
c_{n+1} = B[\lambda]^n, \tag{26}\]

where $L$ is an operator defined by Eq. (17) as

\[
L_i^(*) = \begin{cases}
(2a_0 + \beta - \gamma N^2)\phi_i, & 1 < i < N, \\
(\gamma), & i = 1 \text{ or } i = N,
\end{cases} \quad 1 < i < N, \tag{28}\]

and $[c]^n = [c_1^n, c_2^n, ..., c_N^n, 0, 0]^T$, where

\[
c_i^n = \begin{cases}
\frac{\partial c_i}{\partial t} + 2a_0 - \gamma(1-\Theta)\nabla^2 u_i^n + 2a_0\delta t u_i^n f_i^n, & 1 < i < N, \\
\frac{\partial c_i}{\partial t}, & i = 1,
\end{cases} \quad 1 < i < N, \tag{29}\]

Also, for $n \geq 1$,

\[
[c]^{n+1} = B[\lambda]^{n+1}, \tag{29}\]

where $L$ and $[c]^{n+1} = [c_1^{n+1}, c_2^{n+1}, ..., c_N^{n+1}, 0, 0]^T$ are obtained by Eq. (18) as

\[
L_i^(*) = \begin{cases}
(2a_0 + \beta - \gamma N^2)\phi_i, & 1 < i < N, \\
(\gamma), & i = 1 \text{ or } i = N,
\end{cases} \quad 1 < i < N, \tag{30}\]

and

\[
c_i^{n+1} = \begin{cases}
2a_0 - \gamma(1-\Theta)\nabla^2 u_i^n - 2\delta t u_i^n - 2a_0\delta t \sum_{k=1}^{n-1} \{u_i^{n-k} - 2u_i^{n-k-1} + u_i^{n-k-1} \} \\
+ 2a_0\delta t \sum_{k=1}^{n-1} \{u_i^{n-k} - u_i^{n-k-1} - \delta t u_i^n\}, & 1 < i < N,
\end{cases} \quad 1 < i < N, \tag{31}\]

and finally considering boundary conditions $c_1^{n+1} = h_1^{n+1}$ and $c_N^{n+1} = h_2^{n+1}$. Obviously the method for $\alpha = 2$ coincides with the coupled method of FDM and radial basis function for integer orders (Avazzadeh et al., 2011; Dehghan and Shokri, 2009; Dehghan and Shokri, 2008).

### 4. Numerical Results

In this section, we investigate practically the applicability and efficiency of the presented method. Thus we implement the method for solving some examples with different parameters. Clearly, increasing the number of collocation points, $N$, decreasing the length of time step, $\delta t$, and growth of $m$ which is order of generalized thin plate splines (GTPS) could improve the results. It is
necessary to emphasize the increase of $m$ does not lead to the increasing of computations or complexity of mathematical operations, but it leads to the ill-conditioning (Buhmann, 2003; Powell, 1994; Schaback, 1996; Wu and Schaback, 1993). Hence, we could increase $m$ restrictively to avoid the ill-conditioning of the obtained system of linear equations. The reported results are obtained by using $\theta = \frac{1}{2}$, $m = 4$ and $N = 50$ for different $n$ and $\alpha$. In this study, the infinity norm of error function is the main criteria for evaluation of accuracy and efficiency of method.

**Example 1.** Consider the fractional diffusion-wave equation Eq. (6) in $[0,1]$, $\gamma = \pi$, $\beta = 1$ and $f(x,t)$ which is compatible with the exact solution as follows

$$u(x,t) = t^2 \sinh(x),$$

The semi-discrete approximated solution and error functions are shown in Fig. 1. Also, the root mean square of error (RMSE) for random points is reported in Table 1 for different value of $n$ and $\alpha$. Furthermore, the approximated value at $t = 1$, where we have the most accumulated error, for different value of $n$ and $\alpha$ are reported in Table 2. Note that the differences between $L_\infty$-error and RMS-error shown in Table 1 confirm the error accumulates when $n$ is increasing. Also, the absolute error functions illustrated in Fig. 1 are increase with respect to $t$ which construe accumulation of error with respect to $n$.

![Fig. 1](image-url)
Table 1. The root mean square (RMS) and infinity norm of error function with different \(n\) and \(\alpha\) for Example 1.

<table>
<thead>
<tr>
<th>(n)</th>
<th>(\alpha = 1.25)</th>
<th>(\alpha = 1.5)</th>
<th>(\alpha = 1.75)</th>
<th>(\alpha = 1.95)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(L_\infty)-error</td>
<td>RMS-error</td>
<td>(L_\infty)-error</td>
<td>RMS-error</td>
</tr>
<tr>
<td>10</td>
<td>1.5565 E-04</td>
<td>5.8833 E-07</td>
<td>1.5557 E-04</td>
<td>5.8291 E-07</td>
</tr>
<tr>
<td>20</td>
<td>1.5564 E-04</td>
<td>5.5403 E-07</td>
<td>1.5565 E-04</td>
<td>5.4850 E-07</td>
</tr>
<tr>
<td>50</td>
<td>1.5563 E-04</td>
<td>5.3333 E-07</td>
<td>1.5569 E-04</td>
<td>5.2782 E-07</td>
</tr>
<tr>
<td>100</td>
<td>1.5563 E-04</td>
<td>5.2639 E-07</td>
<td>1.5570 E-04</td>
<td>5.2092 E-07</td>
</tr>
</tbody>
</table>

Table 2. Numerical results for Example 1 with different \(n\) and \(\alpha\) at \(t = 1\).

<table>
<thead>
<tr>
<th>(x)</th>
<th>(n = 10)</th>
<th>(n = 20)</th>
<th>(n = 50)</th>
<th>(n = 10)</th>
<th>(n = 20)</th>
<th>(n = 50)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.9950933</td>
<td>0.9951043</td>
<td>0.9951107</td>
<td>0.9950226</td>
<td>0.9950368</td>
<td>0.9951844</td>
</tr>
<tr>
<td>0.2</td>
<td>0.20079972</td>
<td>0.20080174</td>
<td>0.20080291</td>
<td>0.20078691</td>
<td>0.20078954</td>
<td>0.20081683</td>
</tr>
<tr>
<td>0.3</td>
<td>0.30402685</td>
<td>0.30402960</td>
<td>0.30403116</td>
<td>0.30409686</td>
<td>0.30401323</td>
<td>0.30405039</td>
</tr>
<tr>
<td>0.4</td>
<td>0.41029494</td>
<td>0.41030268</td>
<td>0.41030448</td>
<td>0.4107962</td>
<td>0.41028737</td>
<td>0.41032718</td>
</tr>
<tr>
<td>0.5</td>
<td>0.52065615</td>
<td>0.52065950</td>
<td>0.52066138</td>
<td>0.52065340</td>
<td>0.52063970</td>
<td>0.52068530</td>
</tr>
<tr>
<td>0.6</td>
<td>0.63621617</td>
<td>0.63621936</td>
<td>0.63622115</td>
<td>0.63619639</td>
<td>0.63620048</td>
<td>0.63624390</td>
</tr>
<tr>
<td>0.7</td>
<td>0.75821284</td>
<td>0.75821517</td>
<td>0.75821672</td>
<td>0.75810541</td>
<td>0.75810893</td>
<td>0.75814606</td>
</tr>
<tr>
<td>0.8</td>
<td>0.88761746</td>
<td>0.88761947</td>
<td>0.88762062</td>
<td>0.88760481</td>
<td>0.88760740</td>
<td>0.88763465</td>
</tr>
<tr>
<td>0.9</td>
<td>1.02593218</td>
<td>1.02593326</td>
<td>1.02593389</td>
<td>1.02592522</td>
<td>1.02592662</td>
<td>1.02594135</td>
</tr>
<tr>
<td>1.0</td>
<td>1.17520119</td>
<td>1.17520119</td>
<td>1.17520119</td>
<td>1.17520119</td>
<td>1.17520119</td>
<td>1.17520119</td>
</tr>
</tbody>
</table>

Example 2. Consider the fractional diffusion-wave equation Eq.(6) in \([0,1]\), \(\gamma = 1\), \(\beta = 1\) and \(f(x,t)\) which is compatible with the exact solution as follows

\[u(x,t) = \frac{t+x}{1+\sin(x)} \exp(x-t),\]

The approximated solution and error functions are shown in Fig. 1. Also, the error functions at \(t = 0.5\) and \(t = 1\) are illustrated for different \(\alpha\) in Fig. 2 for both Examples 1 and 2. Observably the error value is often increasing when \(\alpha\) grows up. Also, the approximated value at \(t = 1\) for different \(n\) and \(\alpha\) are reported in Table 3.

5. Conclusion

In this study, we implemented the RBF method for solving the fractional diffusion-wave equation and numerical results show the validity and accuracy of the method. According to the method, the solution is approximated continuously with respect to space direction by using the radial basis function. Obviously, the results could be improved by refinement of meshsize in the both of time and space directions. Moreover, description of the proposed method shows the method is flexible for different boundary conditions. However it is necessary to investigate the initial conditions correspondingly to Eq. (16) in the description of method. In fact, we can modify the first and \(N\) th row of the matrix in Eq. (27) with regard to the non-classic conditions. Therefore, the method is applicable to solve the large class of different type of fractional diffusion-wave equations. At last, implementation of the proposed method leads to the only linear equations systems through the recursive equation Eq. (18). Hence, the method can be considered fast, simple and efficient.
Fig. 2. Error functions for $n = 30$ and different value of $\alpha$ at $t = 0.5$ and $t = 1$ for Examples 1 and 2

Table 3. Numerical results for Example 2 with different $n$ and $\alpha$ at $t = 1$

<table>
<thead>
<tr>
<th>$x$</th>
<th>$n = 10$</th>
<th>$n = 20$</th>
<th>$n = 50$</th>
<th>$\alpha = 1.5$</th>
<th>$n = 10$</th>
<th>$n = 50$</th>
<th>$\alpha = 1.25$</th>
<th>$n = 10$</th>
<th>$n = 50$</th>
<th>$\alpha = 1.75$</th>
<th>$u(x, 1)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.1</td>
<td>0.408958</td>
<td>0.407205</td>
<td>0.406141</td>
<td>0.40864415</td>
<td>0.40628597</td>
<td>0.40999119</td>
<td>0.40614032</td>
<td>0.40663123</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.2</td>
<td>0.454853</td>
<td>0.451773</td>
<td>0.449912</td>
<td>0.45424043</td>
<td>0.45306669</td>
<td>0.45680611</td>
<td>0.44995190</td>
<td>0.44982777</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.3</td>
<td>0.505167</td>
<td>0.501146</td>
<td>0.498723</td>
<td>0.50430596</td>
<td>0.49886598</td>
<td>0.50786205</td>
<td>0.49878619</td>
<td>0.49830245</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.4</td>
<td>0.560962</td>
<td>0.556386</td>
<td>0.553632</td>
<td>0.55993940</td>
<td>0.55376386</td>
<td>0.56412999</td>
<td>0.55370517</td>
<td>0.55299132</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.5</td>
<td>0.622783</td>
<td>0.618531</td>
<td>0.615673</td>
<td>0.62220364</td>
<td>0.61580062</td>
<td>0.62660031</td>
<td>0.61574779</td>
<td>0.61496571</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.6</td>
<td>0.693386</td>
<td>0.688473</td>
<td>0.686114</td>
<td>0.69237620</td>
<td>0.68624375</td>
<td>0.69652791</td>
<td>0.68618364</td>
<td>0.68546782</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.7</td>
<td>0.772704</td>
<td>0.768728</td>
<td>0.766387</td>
<td>0.77188962</td>
<td>0.76652521</td>
<td>0.77537889</td>
<td>0.76644488</td>
<td>0.76595148</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.8</td>
<td>0.863066</td>
<td>0.860060</td>
<td>0.858244</td>
<td>0.86247729</td>
<td>0.85838943</td>
<td>0.86496883</td>
<td>0.85827832</td>
<td>0.85813033</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0.9</td>
<td>0.966335</td>
<td>0.964381</td>
<td>0.963699</td>
<td>0.96603881</td>
<td>0.96374173</td>
<td>0.96733276</td>
<td>0.96360589</td>
<td>0.96403586</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.0</td>
<td>1.086088</td>
<td>1.086088</td>
<td>1.086088</td>
<td>1.08608825</td>
<td>1.08608825</td>
<td>1.08608825</td>
<td>1.08608825</td>
<td>1.08608825</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Acknowledgement

The work described in this paper was supported by the National Basic Research Program of China (973 Project No. 2010CB832702), the National Science Funds for Distinguished Young Scholars of China (11125208), NSFC Funds (No. 11125208), the 111 project under Grant B12032.

References


