Eur. Phys. J. Special Topics **222**, 1987–1998 (2013)
© EDP Sciences, Springer-Verlag 2013
DOI: 10.1140/epjst/e2013-01979-7

THE EUROPEAN
PHYSICAL JOURNAL
SPECIAL TOPICS

Regular Article

A finite difference method with non-uniform timesteps for fractional diffusion and diffusion-wave equations

J. Quintana-Murillo and S.B. Yuste[a]

Departamento de Física, Universidad de Extremadura, 06071 Badajoz, Spain

Received 3 June 2013 / Received in final form 6 August 2013
Published online 7 October 2013

Abstract. An implicit finite difference method with non-uniform timesteps for solving fractional diffusion and diffusion-wave equations in the Caputo form is presented. The non-uniformity of the timesteps allows one to adapt their size to the behaviour of the solution, which leads to large reductions in the computational time required to obtain the numerical solution without loss of accuracy. The stability of the method has been proved recently for the case of diffusion equations; for diffusion-wave equations its stability, although not proven, has been checked through extensive numerical calculations.

1 Introduction

Fractional derivatives and fractional integrals are the basis of the fractional calculus, for long a relatively obscure branch of mathematics whose origins go back to Leibniz [1]. However, in recent years, the usefulness of these integro-differential operators to analyze and solve problems in, e.g., Physics, Economics, Biology, and Chemistry [2–12], has been increasingly recognized. A field of especial interest is that of fractional differential equations (FDE) [3,4]. For example, some anomalous diffusion processes accounted for by the Continuos Time Random Walk model are well described by fractional diffusion equations [5,6,13].

Different analytical methods have been considered to solve FDE (method of images, separation of variables, integral transform method, . . .) [4,5,14–17] but, as for non-fractional problems, no solution is available in many cases and one has to resort to numerical methods. The development and analysis of numerical methods for FDE is a very active field of investigation today [18–35]. In particular, finite difference methods have been intensively studied, but, with the exception of a few works (e.g., a recent paper by Skovranek et al. [28], some works by Mustapha et al. [29,30], or a previous paper by the authors [31]), the size of the timesteps is always taken as constant. However, to have at one's disposal methods with variable timesteps would be very convenient because they would allow the implementation of adaptive methods where the size of the timesteps is tailored in accordance with the behaviour of the solution: small timesteps are employed when the solution changes rapidly in order to

[a] e-mail: santos@unex.es

maintain the accuracy of the method, but large timesteps can be used wherever the function's variation is smooth, which usually leads to much faster methods than those with fixed timesteps. In particular, the length of the timesteps can be chosen so that the error is smaller than a prefixed value. This idea is developed in this paper.

We present a finite difference method with non-uniform timesteps to solve both subdiffusion and diffusion-wave equations. The method was already considered in [31] for subdiffusion equations. However, the adaptive algorithm we discuss in this article is different. Regarding the stability of the method for diffusion-wave equations, it is remarkable that the Von Neumann (or Fourier) stability analysis, so successful for the subdiffusive case [19,20,32], seems inappropriate to assert the stability of the difference algorithm for diffusion-wave equation when variable timesteps are considered. Nonetheless, in practical terms, we have always found that the algorithm behaves stably despite the extremely different timestep distributions we have used to test its stability.

The structure of the present paper is as follows. In Sect. 2 we present the finite difference algorithm with non-uniform timesteps for both subdiffusion and diffusion-wave equations. In Sect. 3 we describe the adaptive procedure, i.e., the criterion we have employed to choose the size of the timesteps. We present in Sect. 4 some numerical results obtained by means of this method for a subdiffusion problem and a diffusion-wave problem both with known exact solution. We end with some conclusions in Sect. 5.

2 The subdiffusion and diffusion-wave implicit algorithms

We shall consider the fractional subdiffusion and diffusion-wave equations in the Caputo form

$$\partial u = F \tag{1}$$

with

$$\partial \equiv \frac{\partial^\gamma}{\partial t^\gamma} - K\frac{\partial^2}{\partial x^2}, \tag{2}$$

where

$$\frac{\partial^\gamma}{\partial t^\gamma} y(t) \equiv \frac{1}{\Gamma(1-\gamma)}\int_0^t d\tau \frac{1}{(t-\tau)^\gamma}\frac{dy(\tau)}{d\tau}, \quad 0<\gamma<1, \tag{3}$$

is the Caputo fractional derivative for the subdiffusion equation and

$$\frac{\partial^\gamma}{\partial t^\gamma} y(t) \equiv \frac{1}{\Gamma(2-\gamma)}\int_0^t d\tau \frac{1}{(t-\tau)^{\gamma-1}}\frac{d^2y(\tau)}{d\tau^2}, \quad 1<\gamma<2, \tag{4}$$

is the Caputo fractional derivative for the diffusion-wave equation. The function $F(x,t)$ is a given source term.

The two difference schemes we develop in this paper are obtained by discretizing (i) the Laplacian with the three-point centred formula, and (ii) the Caputo derivatives (3) and (4) by means of a generalization of the L1 and L2 formulae, respectively, to non-uniform meshes. The resulting algorithm is second-order accurate in the spatial mesh size and first-order accurate in the timestep size. For a uniform temporal mesh, the algorithm corresponding to the subdiffusion equation becomes the one discussed by Liu et al. [33] and Murio [32], whereas it becomes the algorithm discussed by the authors in [35] in the case of diffusion-wave equations.

In general, in order to build a numerical difference scheme, one starts by considering a mesh in the space-time region where one wants to obtain the numerical estimate $U_j^{(m)}$ of the exact solution $u(x_j,t_m)=u_j^{(m)}$, with (x_j,t_m) being the coordinates of the (j, m) node of the mesh. Next, one replaces the continuous operator ∂ of the equation $\partial u = F$ one has to solve by a difference operator δ and a truncation error $R(x, t)$: $\partial u = \delta u + R$. For example, the continuous operators that define ∂ in Eq. (2) are replaced by

$$\frac{\partial^\gamma}{\partial t^\gamma} u(x,t) = \delta_t^\gamma u(x,t) + R_t(x,t), \tag{5}$$

$$\frac{\partial^2}{\partial x^2} u(x,t) = \delta_x^2 u(x,t) + R_x(x,t), \tag{6}$$

with δ_t^γ and δ_x^2 being the corresponding difference operators. This way one gets

$$\left[\delta_t^\gamma - K\delta_x^2\right] u(x,t) = R(x,t) + F(x,t) \tag{7}$$

where $R(x,t) = KR_x(x,t) - R_t(x,t)$. In this case $\delta = \delta_t^\gamma - K\delta_x^2$. Neglecting the truncation term, $R(x,t)$, one gets a difference equation $\delta U = F$ whose solution leads to the finite difference estimate of the exact solution $u(x,t)$ at the mesh points. For a given operator ∂ one can consider many different difference operators δ, and hence many different finite difference methods for solving $\partial u = F$. In this paper we assume that the spatial size of the mesh $x_{j+1} - x_j = \Delta x$ is constant, and use the three-point centred formula to approximate $\partial^2/\partial x^2$ by

$$\delta_x^2 u(x_j, t) = \frac{u(x_{j+1},t) - 2u(x_j,t) + u(x_{j-1},t)}{(\Delta x)^2}. \tag{8}$$

In this case $R_x(x_j, t) = O(\Delta x)^2$.

2.1 The subdiffusion algorithm

For the fractional Caputo derivative we choose a discretized operator δ_t^γ that is a generalization to non-uniform meshes of the L1 formula [1]:

$$\delta_t^\gamma u(x,t_n) = \frac{1}{\Gamma(2-\gamma)} \sum_{m=0}^{n-1} T_{m,n}^{(\gamma)} \left[u(x,t_{m+1}) - u(x,t_m)\right], \tag{9}$$

where [31]

$$T_{m,n}^{(\gamma)} = \frac{(t_n - t_m)^{1-\gamma} - (t_n - t_{m+1})^{1-\gamma}}{t_{m+1} - t_m}, \quad m \le n-1, \tag{10}$$

and $R_t(x,t_n)$ is of order $t_n^{1-\gamma} \max_{0\le m\le n-1} (t_{m+1} - t_m)$. Therefore the truncation error is

$$R(x_j,t_n) = O(\Delta x)^2 + t_n^{1-\gamma} O\left[\max_{0\le m\le n-1} (t_{m+1} - t_m)\right], \tag{11}$$

which goes to zero when the spacing of the time-spatial mesh goes to zero. This means that the method is consistent [36]. Hence, introducing (8) and (9) into $\delta = \delta_t^\gamma - K\delta_x^2$, the finite difference equation $\delta U = F$ we have to solve becomes, after multiplying it

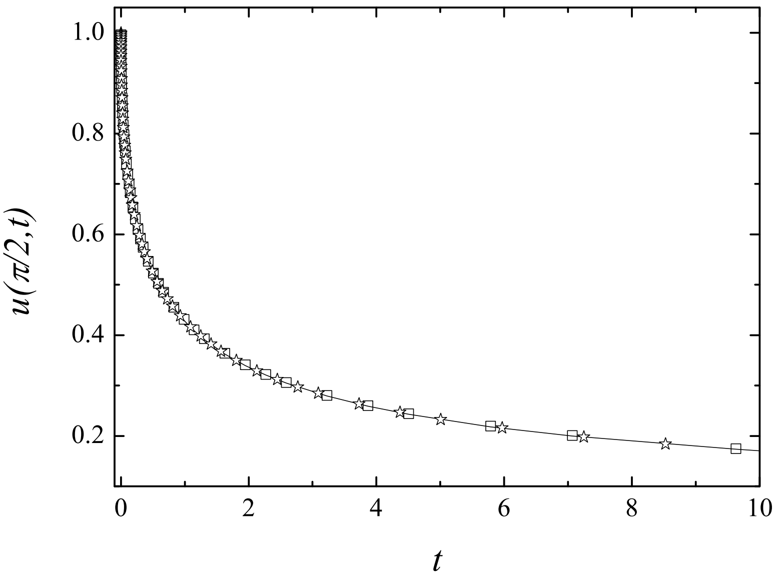

Fig. 1. Solution at the mid-point, $u(x_k = \pi/2, t)$, of the subdiffusion problem described in the main text when $\gamma = 1/2$. Solid line: exact solution; squares: adaptive numerical solution $U_k^{(n)}$ up to $n = 45$ ($t_{45} = 9.6268$) and $\bar{\tau} = 5 \times 10^{-4}$; stars: $U_k^{(n)}$ up to $n = 112$ ($t_{112} = 9.1690$) and $\bar{\tau} = 10^{-4}$. In both cases $\Delta_0 = 0.01$ and $\Delta x = \pi/40$.

by $(t_n - t_{n-1})^\gamma$,

$$\sum_{m=0}^{n-1} \tilde{T}_{m,n}^{(\gamma)} \left[U_j^{(m+1)} - U_j^{(m)}\right] - S_n[U_{j+1}^{(n)} - 2U_j^{(n)} + U_{j-1}^{(n)}] = G_j^{(n)} \tag{12}$$

where

$$S_n = \Gamma(2-\gamma) K \frac{(t_n - t_{n-1})^\gamma}{(\Delta x)^2}, \tag{13}$$

$$\tilde{T}_{m,n}^{(\gamma)} = (t_n - t_{n-1})^\gamma T_{m,n}^{(\gamma)}, \tag{14}$$

$$G_j^{(n)} = \Gamma(2-\gamma)(t_n - t_{n-1})^\gamma F(x_n, t_n). \tag{15}$$

Reordering (12) and taking into account that $\tilde{T}_{n-1,n}^{(\gamma)} = 1$, we get the (implicit) finite difference scheme we were looking for:

$$-S_n\, U_{j+1}^{(n)} + (1+2S_n) U_j^{(n)} - S_n\, U_{j-1}^{(n)} = U_j^{(n-1)} - \sum_{m=0}^{n-2} \tilde{T}_{m,n}^{(\gamma)} \left[U_j^{(m+1)} - U_j^{(m)}\right] + G_j^{(n)}. \tag{16}$$

It is interesting to note that, if one discretizes the Caputo derivative at time t_{n+1} and the Laplacian at time t_n, then one straightforwardly gets an *explicit* finite difference scheme. We will not explore this explicit method here because it is unstable if, for a given set of parameters γ, Δx, and K, the timesteps are not small enough.

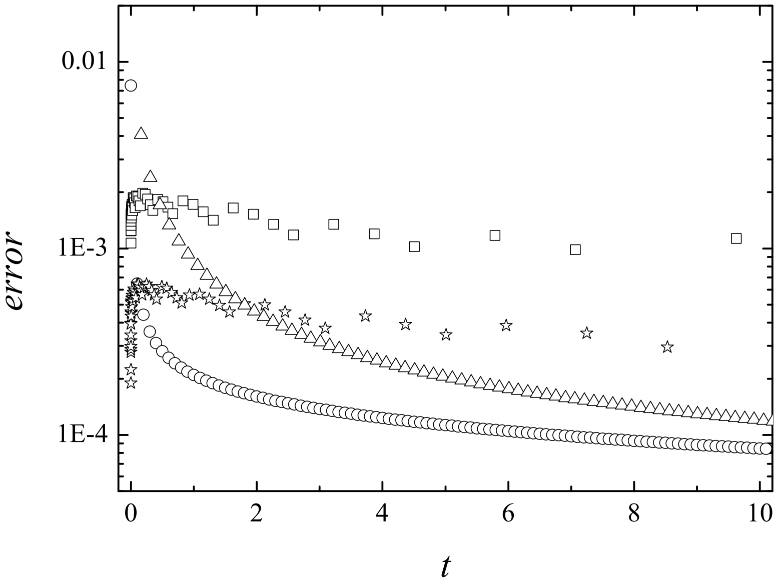

Fig. 2. Logarithmic plot of the numerical errors at the mid-point, $|u(\pi/2, t_n) - U_k^{(n)}|$ of (i) the adaptive method with $\bar{\tau} = 5 \times 10^{-4}$, $\Delta_0 = 0.01$ (squares, 45 timesteps, CPU time $\approx$ 0.15s); (ii) the adaptive method with $\bar{\tau} = 10^{-4}$ and $\Delta_0 = 0.01$ (stars, 112 timesteps, CPU time $\approx$ 0.75s); (iii) the method with constant timesteps of size $\Delta_n = 0.01$ (triangles, 1010 timesteps, CPU time $\approx$ 440s); and (iv) the method with uniform timesteps of size $\Delta_n = 0.001$ (circles, 10100 timesteps, CPU time $\approx$ 43800s). In all cases $\gamma = 1/2$ and $\Delta x = \pi/40$.

2.2 The diffusion-wave algorithm

For the fractional Caputo derivative we choose a discretized operator δ_t^γ that is a generalization of the L2 formula [1] for non-uniform meshes:

$$\begin{aligned} \delta_t^\gamma u(x, t_n) = \frac{1}{\Gamma(3-\gamma)} \sum_{m=0}^{n-1} \Big\{ & A_{m,n}^{(\gamma)} \left[u(x, t_{m+1}) - u(x, t_m) \right] \\ & - B_{m,n}^{(\gamma)} \left[u(x, t_m) - u(x, t_{m-1}) \right] \Big\} \end{aligned} \tag{17}$$

where

$$A_{m,n} = 2 \frac{(t_n - t_m)^{2-\gamma} - (t_n - t_{m+1})^{2-\gamma}}{(t_{m+1} - t_m)(t_{m+1} - t_{m-1})}, \tag{18}$$

$$B_{m,n} = 2 \frac{(t_n - t_m)^{2-\gamma} - (t_n - t_{m+1})^{2-\gamma}}{(t_m - t_{m-1})(t_{m+1} - t_{m-1})}, \tag{19}$$

and $R_t(x, t_n)$ is of order $t_n^{2-\gamma} \max_{0 \le m \le n-1} (t_{m+1} - t_m)$. Therefore the truncation error is

$$R(x_j, t_n) = O(\Delta x)^2 + t_n^{2-\gamma} O \left[\max_{0 \le m \le n-1} (t_{m+1} - t_m) \right]. \tag{20}$$

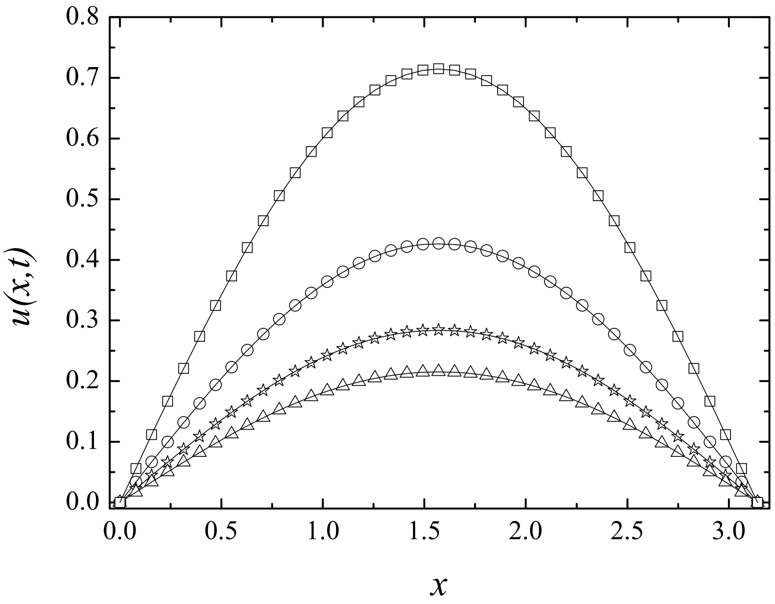

Fig. 3. Exact solution (lines) and adaptive numerical solution (symbols) of the subdiffusion problem described in the main text for $\gamma = 1/2$ and $t_{27} = 0.109$ (squares), $t_{82} = 1.009$ (circles), $t_{99} = 3.089$ (stars), and $t_{107} = 5.969$ (triangles), with $\bar{\tau} = 10^{-4}$, $\Delta_0 = 0.01$, and $\Delta x = \pi/40$.

This quantity goes to zero when the spacing of the time-spatial mesh goes to zero, which implies that the method is consistent [36]. Introducing (8) and (17) into $\delta = \delta_t^\gamma - K\delta_x^2$ one gets the finite difference equation we were looking for:

$$-S_n\, U_{j+1}^{(n)} + (1+2S_n)U_j^{(n)} - S_n\, U_{j-1}^{(n)} = U_j^{(n-1)} + \frac{t_n - t_{n-1}}{t_{n-1} - t_{n-2}} \left[U_j^{(n-1)} - U_j^{(n-2)}\right]$$

$$- \sum_{m=0}^{n-2} \left(\hat{A}_{m,n} \left[U_j^{(m+1)} - U_j^{(m)}\right] - \hat{B}_{m,n} \left[U_j^{(m)} - U_j^{(m-1)}\right]\right) + G_j^{(n)} \tag{21}$$

with the empty sum convention if $n = 1$ and

$$\hat{A}_{m,n} = \frac{t_n - t_{n-2}}{2(t_n - t_{n-1})^{1-\gamma}} A_{m,n}, \tag{22}$$

$$\hat{B}_{m,n} = \frac{t_n - t_{n-2}}{2(t_n - t_{n-1})^{1-\gamma}} B_{m,n}, \tag{23}$$

$$S_n = \frac{K(t_n - t_{n-2})\Gamma(3-\gamma)}{2(t_n - t_{n-1})^{1-\gamma}(\Delta x)^2}\cdot \tag{24}$$

Finally, it is worth mentioning that, as in the case of the subdiffusion equation, if one discretizes the Caputo derivative at time t_{n+1} and the Laplacian at time t_n, then one gets an *explicit* finite difference scheme. Again, we will not explore this explicit method here because it is unstable if, for a given set of parameters γ, Δx, and K, the timesteps are not small enough.

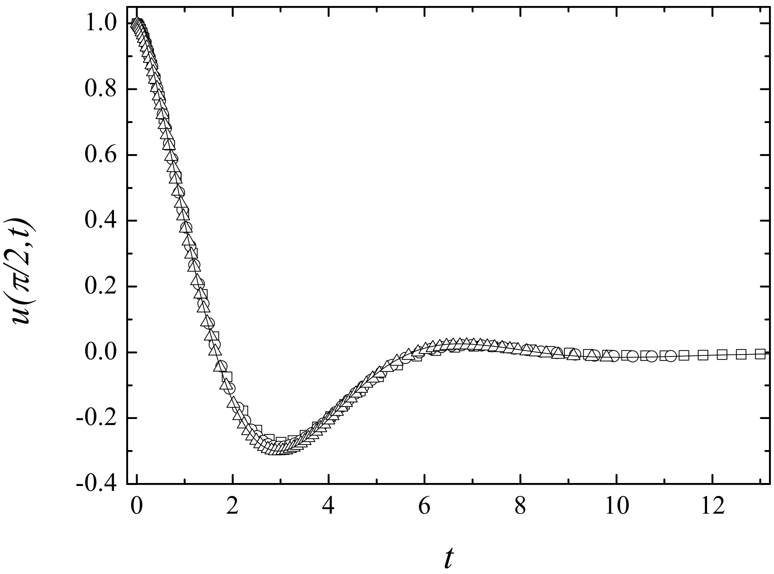

Fig. 4. Solutions at the mid-point $x = \pi/2$ for the fractional diffusion-wave problem described in the main text with $\gamma = 3/2$. Solid line: exact solution; symbols: adaptive method with $\bar{\tau} = 10^{-4}$ up to time $t_{100} = 13.1958$ (squares), $\bar{\tau} = 10^{-5}$ up to time $t_{150} = 11.3309$ (circles), and $\bar{\tau} = 10^{-6}$ up to time $t_{300} = 9.9783$ (triangles). In all cases $\Delta x = \pi/40$ and $\Delta_0 = 0.01$

3 The adaptive method

In order to solve Eq. (1) by means of the scheme (16) we have developed an adaptive method that is as follows. Let us assume we have already evaluated $U_j^{(m)}$ for $m \le n-1$. The time t_n of the n-th step is chosen in the following way. To initialize the adaptive algorithm, we assume the size of the n-th timestep is equal to the size of the previous timestep (i.e., $\Delta_n = \Delta_{n-1}$, where $\Delta_m = t_m - t_{m-1}$), and then we estimate the solution at t_n, which we denote by $\hat{U}_j^{(n)}$. Next we estimate the solution again at time t_n but using now two timesteps of size $\Delta_n/2$. We denote the corresponding solution by $\bar{U}_j^{(n)}$. If the difference between these two values at a given position, say $x = x_k$, $\bar{\Delta} = |\hat{U}_k^{(n)} - \bar{U}_k^{(n)}|$, is greater than a prefixed quantity $\bar{\tau}$ (tolerance), we halve the size of the n-th timestep, $\Delta_n \to \Delta_n/2$, and repeat the procedure until the difference $\bar{\Delta}$ is less than $\bar{\tau}$. In this case, we take the last values of $\hat{U}_j^{(n)}$ as the solution at t_n. On the contrary, if the initial difference $\bar{\Delta}$ is less than $\bar{\tau}$, then we double the value of the timestep, $\Delta_n \to 2\Delta_n$, and repeat the procedure until the difference $\bar{\Delta}$ is greater than $\bar{\tau}$. When this occurs, we take the last values of $\hat{U}_j^{(n)}$ (the values obtained when the difference $\bar{\Delta}$ was *less* than $\bar{\tau}$) as the solution at t_n.

For the subdiffusion equation, the first value of Δ_0, i.e., the first value of t_1, has to be conjectured. Note that this value should be chosen taking into account the value of the tolerance because, for example, it would be inefficient to employ a very large initial value for Δ_0 if the chosen prescribed tolerance is very small. Anyway, unless unrealistic bad initial guesses are made, one expects the method will be able to quickly reach an initial timestep compatible with the prescribed tolerance. For the diffusion-wave equation, one has to choose, besides $\bar{\tau}$ and t_1, the value of t_{-1}. The value of $\hat{U}_j^{(-1)}$ is then fixed so that the initial condition that involves the time

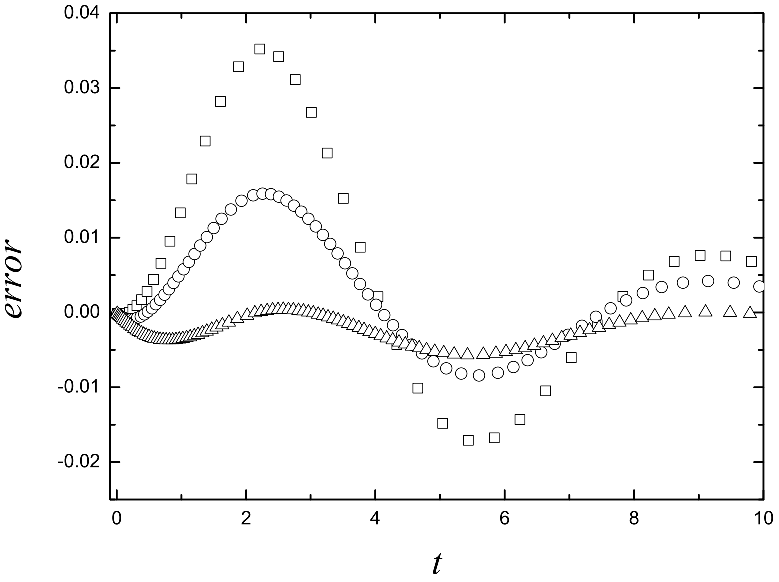

Fig. 5. Numerical errors at the mid-point, $|u(\pi/2,t_n) - U_k^n|$, of the adaptive method for the fractional diffusion-wave equation described in the main text with $\gamma = 3/2$ when $\bar{\tau} = 5 \times 10^{-4}$ (squares), $\bar{\tau} = 10^{-5}$ (circles), and $\bar{\tau} = 10^{-6}$ (triangles). In all cases $\Delta_0 = 0.01$ and $\Delta x = \pi/40$.

derivative holds. This procedure is not different from that usually employed in non-fractional equations. In our numerical calculations we have always taken $\Delta_1 = \Delta_0$. Apart from its simplicity, this choice has another advantage: it turns out that the algorithm for the diffusion-wave equation worsens when the lengths of two successive timesteps, Δ_m and Δ_{m-1}, are too different (although it seems that even in this case the algorithm is stable; see Sect. 4). For this reason we have restricted the ratio between the lengths of two successive timesteps, $|\Delta_m/\Delta_{m-1}|$ or $|\Delta_{m-1}/\Delta_m|$, to be smaller than a given value. In our numerical computations we use the value 1.1.

4 Numerical results

4.1 Subdiffusion problem

The subdiffusion problem we are going to consider is given by the equation

$$\frac{\partial^\gamma u}{\partial t^\gamma} = K\frac{\partial^2 u}{\partial x^2}, \quad 0 < \gamma < 1, \tag{25}$$

defined in the interval $0 \le x \le \pi$, with absorbing boundary conditions, $u(x = 0, t) = u(x = \pi, t) = 0$, initial condition $u(x, 0) = \sin x$ and $K = 1$. Its exact solution is

$$u(x,t) = E_\gamma(-t^\gamma)\sin(x) \tag{26}$$

where E_γ is the Mittag-Leffler function [4,7].

The results we are going to show correspond to $\gamma = 0.5$, which is a representative value; the election of other values for γ leads to similar results. In Fig. 1 the exact values of $u(0,t)$ are compared with their corresponding numerical estimates obtained by means of our present adaptive algorithm. We have used $\Delta_0 = 0.01$ and two different

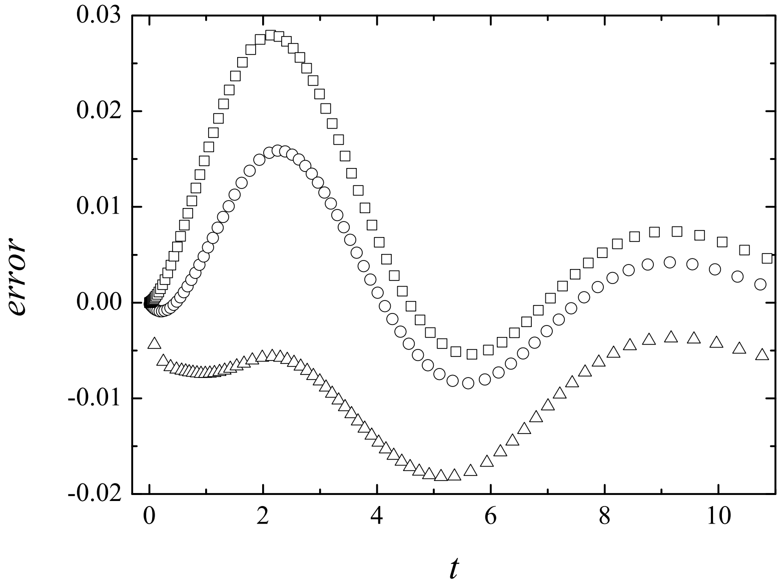

Fig. 6. Numerical errors at the mid-point, $\left|u(\pi/2, t_n) - U_k^{(n)}\right|$, of the adaptive method for the fractional diffusion-wave equation described in the main text with $\gamma = 3/2$ when $\Delta_0 = 0.001$ (squares), $\Delta_0 = 0.01$ (circles), and $\Delta_0 = 0.1$ (triangles). In all cases $\bar{\tau} = 10^{-5}$ and $\Delta x = \pi/40$.

values for the tolerance, $\bar{\tau} = 5 \times 10^{-4}$ and $\bar{\tau} = 10^{-4}$, with $\bar{\Delta} = |\hat{U}_k^{(n+1)} - \bar{U}_k^{(n+1)}|$ being evaluated at the midpoint, i.e., we choose $x_k = \pi/2$. The differences between the exact solution and the numerical estimates are shown in Fig. 2. We see that the results provided by the adaptive method are quite good, improving when the tolerance decreases. Note that the numerical integration is able to span a large time interval with a relatively small number of timesteps while keeping the accuracy of the method around 10^{-3} (or even lower when $\bar{\tau} = 10^{-4}$ is employed). For the sake of comparison, we also show the error of the algorithm when the size of the timesteps is fixed. In this case, the errors of the numerical integration for short times are larger than those found when the adaptive procedure is used, although these errors decrease fast, and they quickly become smaller than those of the adaptive procedure. That is, the algorithm with fixed timesteps leads to quite uneven errors: in order to keep the errors reasonably small for some times one has to assume too much (and in some cases, irrelevant) precision for other time regimes. Besides, it should be noted that the number of steps required by the method with fixed timesteps is much larger than the number required by the adaptive method. This implies the use of much larger quantities of computer memory and much longer computation times. For example, the CPU time we needed to obtain the numerical results shown in Fig. 2 for $\Delta_n = 0.001$ was well above 10^4 times the time required to obtain the solution by means of the adaptive method where $\bar{\tau} = 10^{-4}$. Finally, in Fig. 3 we plot the exact solution $u(x, t)$ versus x for different values of t and the corresponding estimates obtained by means of the adaptive method with $\bar{\tau} = 10^{-4}$ and $\Delta_0 = 0.01$. The agreement is excellent.

4.2 Diffusion-wave problem

The diffusion-wave problem we will deal with now is given by Eq. (25) defined in the interval $0 \leq x \leq \pi$, with $1 < \gamma < 2$, $K = 1$, absorbing boundary conditions,

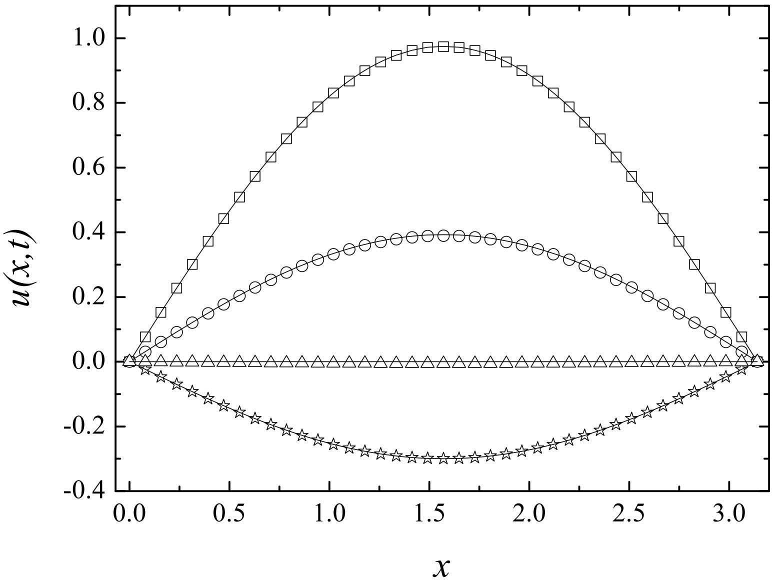

Fig. 7. Exact solution (lines) and adaptive numerical solution (symbols) of the diffusion-wave problem described in the main text for $\gamma = 3/2$ and $t_{16} = 0.105$ (squares), $t_{84} = 1.006$ (circles), $t_{156} = 3.015$ (stars), and $t_{233} = 5.741$ (triangles), with $\tau = 10^{-6}$, $\Delta_0 = 0.01$ and $\Delta x = \pi/40$.

$u(x = 0, t) = u(x = \pi, t) = 0$, and initial conditions $u(x, 0) = \sin x$ and $\partial u(x, t)/\partial t|_{t=0} = 0$. The solution is again given by Eq. (26).

In what follows we always use $\gamma = 3/2$ as a representative value; similar results are obtained for other values of γ belonging to the interval $1 < \gamma < 2$. In Fig. 4 the exact solution at the midpoint $x = \pi/2$, $u(\pi/2, t)$, is compared with the numerical solutions obtained for three different values of $\bar{\tau}$. This comparison is carried out in a more detailed way in Fig. 5 where the errors of the numerical calculations are shown. One clearly sees that the accuracy of the method improves when $\bar{\tau}$ decreases. On the other hand, in Fig. 6 we show the error of the numerical method for three different values of Δ_0 when the tolerance $\bar{\tau}$ is fixed. No pattern is appreciated here as the middle value of Δ_0 seems to be, overall, the best of the three values. Finally, in Fig. 7 we have plotted the exact and numerical solutions versus the position x for several times. As expected from the results shown in Fig. 4, the agreement between the exact values and the numerical estimates is very good.

As we mentioned in Sect. 1, we have not been able to prove that the present finite difference method with non-uniform timesteps for diffusion-wave equations is stable. The von-Neumann stability analysis, so successful for the subdiffusion algorithm, seems useless for the diffusion-wave algorithm as we have not found any rigorous constraint for the growth of the amplitude of the diffusion-wave modes. However, we have carried out very extensive calculations and considered a large variety of timestep functions (including random distributions), and we have always found well-behaved (i.e., stable) numerical solutions.

5 Conluding remarks

We have studied a finite difference method with non-uniform timesteps for fractional diffusion and diffusion-wave equations in which the time derivative is a fractional derivative in the Caputo form. The scheme is obtained by discretizing (i) the Laplacian with the three-point centred formula and (ii) the Caputo derivative with a generalization of the L1 [L2] formula to non-uniform meshes for the diffusion [diffusion-wave] equation. It is second-order accurate in the spatial mesh size and first-order accurate in the timestep size.

The flexibility in the choice of the size of the timesteps allows one to build adaptive methods where these timesteps are determined according to the behaviour of the solution. Typically, one chooses small timesteps when the solution changes fast and large timesteps when the solution changes slowly. This keeps the numerical errors small and reduces the number of steps needed to get the numerical solution up to a given time. This reduction in the number of timesteps is especially relevant for fractional problems because the number of operations required to calculate the numerical solution scales roughly as the square of the number of timesteps, so that this reduction greatly decreases the computer memory and CPU time requirements.

In this paper we have employed an adaptive procedure in which the size of the timesteps is determined so that the difference between the numerical result obtained at time t_{n+1} by means of one timestep of size $t_{n+1} - t_n$ and the numerical result obtained by using two timesteps of half the size, $(t_{n+1} - t_n)/2$, is, in absolute value, smaller than an arbitrary prefixed quantity (the tolerance). We have found that the proposed method is excellent both in terms of accuracy and computational efficiency for subdiffusion equations. However, for diffusion-wave equations we have found that the results, although still reasonable, are not as good as for the subdiffusion equation. Moreover, whereas the stability of the difference method with non-uniform timesteps has been proven for the subdiffusion algorithm, no similar proof has been provided for the diffusion-wave algorithm. Nevertheless, we have never found any sign of instability despite extensive numerical tests with extremely different timestep distributions. Anyway, either improvement of the present method or a proposal of alternative algorithms with non-uniform timesteps for the case of diffusion-wave equations would be desirable.

This work was partially supported by the Ministerio de Ciencia y Tecnología (Spain) through Grant No. FIS2007-60977 and by the Junta de Extremadura (Spain) through Grant No. GRU09038.

References

1. K.B. Oldham, J. Spanier, *The Fractional Calculus* (Academic Press, New York, 1974)
2. R. Hilfer (ed.), *Applications of Fractional Calculus in Physics* (World Scientific, Singapore, 2000)
3. I. Podlubny, *Fractional Differential Equations: An Introduction to Fractional Derivatives, Fractional Differential Equations, to Methods of Their Solution and Some of Their Applications* (Academic Press, San Diego, 1999)
4. A.A. Kilbas, H.M. Srivastava, J.J. Trujillo, *Theory and Applications of Fractional Differential Equations* (Elsevier, Amsterdam, 2006)
5. R. Metzler, J. Klafter, Phys. Rep. **339**, 1 (2000)
6. R. Metzler, J. Klafter, J. Phys. A-Math. Gen. **37**, R161 (2004)
7. R. Klages, G. Radons, I.M. Sokolov (eds.), *Anomalous Transport: Foundations and Applications* (Elsevier, Amsterdam, 2008)
8. R.L. Magin, O. Abdullah, D. Baleanu, X.J. Zhou, J. Mag. Reson. **190**, 255 (2008)
9. I.M. Sokolov, J. Klafter, A. Blumen, Phys. Today **55**, 48 (2002)
10. B.I. Henry, T.A.M. Langlands, S. Wearne, Phys. Rev. Lett. **100**, 128103 (2008)

11. S.B. Yuste, E. Abad, K. Lindenberg, *Reactions in Subdiffusive Media and Associated Fractional Equations*, in *Fractional Dynamics. Recent Advances*, edited by J. Klafter, S.C. Lim R. Metzler (World Scientific, Singapore, 2011)
12. S.B. Yuste, E. Abad, K. Lindenberg, Phys. Rev. E **82**, 061123 (2010)
13. E. Barkai, R. Metzler, J. Klafter, Phys. Rev. E **61**, 132 (2000)
14. S.B. Yuste, L. Acedo, Physica A **336**, 334 (2004)
15. A.M.A. El-Sayed, M. Gaber, Phys. Lett. A **359**, 175 (2006)
16. H. Jafari, S. Momani, Phys. Lett. A **370**, 388 (2007)
17. S.S. Ray, Phys. Scripta **75**, 53 (2007)
18. R. Gorenflo, F. Mainardi, D. Moretti, P. Paradisi, Nonlinear Dynam. **29**, 129 (2002)
19. S.B. Yuste, L. Acedo, SIAM J. Numer. Anal. **42**, 1862 (2005)
20. S.B. Yuste, J. Comput. Phys. **216**, 264 (2006)
21. V.E. Lynch, B.A. Carreras, D. del-Castillo-Negrete, K.M. Ferreira-Mejias, H.R. Hicks, J. Comput. Phys. **192**, 406 (2003)
22. M.M. Meerschaert, C. Tadjeran, J. Comput. Appl. Math. **172**, 65 (2004)
23. C.M. Chen, F. Liu, I. Turner, V. Anh, J. Comput. Phys. **227**, 886 (2007)
24. Z.Z. Sun, X. Wu, Appl. Numer. Math. **56**, 193 (2006)
25. I. Podlubny, A.V. Chechkin, T. Skovranek, Y. Chen, B.M. Vinagre, J. Comput. Phys. **228**, 3137 (2009)
26. M. Cui, J. Comput. Phys. **228**, 7792 (2009)
27. H. Brunner, L. Ling, M. Yamamoto, J. Comput. Phys. **229**, 6613 (2010)
28. T. Skovranek, V.V. Verbickij, Y. Tarte, I. Podlubny, Discretization of fractional-order operators and fractional differential equations on a non-equidistant mesh, Article no. FDA10-062, edited by I. Podlubny, B.M. Vinagre Jara, YQ. Chen, V. Feliu Batlle, I. Tejado Balsera, *Proceedings of FDA10* (The 4th IFAC Workshop Fractional Differentiation and its Applications, Badajoz, 2010), p. 18
29. K. Mustapha, W. McLean, Numer. Algorithms **56**, 159 (2011)
30. K. Mustapha, J. AlMutawa, Numer. Algorithms **61**, 1017 (2012)
31. S.B. Yuste, J. Quintana-Murillo, Comput. Phys. Comm. **182**, 2594 (2012)
32. D.A. Murio, Comput. Math. Appl. **56**, 1138 (2008)
33. F. Liu, P. Zhuang, V. Anh, I. Turner, ANZIAM J. **47**, C48 (2006)
34. S.B. Yuste, J. Quintana-Murillo, Phys. Scripta **T136**, 014025 (2009)
35. J. Quintana-Murillo, S.B. Yuste, J. Comput. Nonlin. Dyn. **6**, 021014 (2011)
36. R.J. LeVeque, *Finite Difference Methods for Ordinary and Partial Differential Equations: Steady-State and Time-Dependent Problems* (SIAM, Philadelfia, 2007)