J Math Chem (2008) 43:1533–1548
DOI 10.1007/s10910-007-9282-2

ORIGINAL PAPER

Wavelet approach incorporated with optimization for solving stiff systems

Tonghua Zhang · Moses O. Tadé · Yu-Chu Tian · Yanduo Zhang · Johan Utomo

Received: 1 March 2007 / Accepted: 10 July 2007 / Published online: 25 August 2007

Abstract Wavelet-based methods open a door for numerical solution of differential equations. Stiff systems, a special type of differential equation systems, have the solutions with the components that exhibit complex dynamic behaviours such as singularities and abrupt transitions, which are hard to be captured by the typical numerical method or incur the computing complexity. This paper proposed to use the Wavelet-Galerkin scheme for solving stiff systems. Daubechies wavelet based connection coefficients, required in the Wavelet-Galerkin scheme, were computed using an algorithm that we recently rectified. The Lagrange multiplier method was incorporated into the wavelet approach in order to optimise the fitting of the initial conditions. Comparative studies were also carried out between the proposed approach and the Haar wavelet approach.

Keywords Wavelet-based method · Wavelet-Galerkin method · Connection coefficients · Stiff system · Numerical solution · Daubechies wavelet

1 Introduction

Many systems in science and engineering are governed by differential equations. Obtaining the solutions of these differential equation systems is essential for system

T. Zhang (✉) · M. O. Tadé · J. Utomo
Department of Chemical Engineering, Curtin University of Technology, GPO Box U1987, Perth, WA 6845, Australia
e-mail: t.zhang@exchange.curtin.edu.au

Y.-C. Tian
School of Software Engineering and Data Communications, Faculty of Information Technology, Queensland University of Technology, GPO Box 2434, Brisbane, QLD 4001, Australia

Y. Zhang
School of Computer Science and Engineering, Wuhan Institute of Technology, Wuhan 430073, P.R. China

analysis, design, optimization, control, etc. While analytical solutions can be derived for many such systems; there are much more systems that cannot be solved analytically. This has been motivating the development of various numerical methods for differential equations.

There have been several well-developed numerical methods for differential equation systems. Typical ones are the finite difference, segmentation and Runge–Kutta methods. All these methods work well when system solutions are regular.

However, some systems such as stiff systems have the solutions with the components that display complex dynamic behaviours like singularities and abrupt transitions. At the stiff part, generally, traditional numerical methods cannot give satisfactory solutions of these systems because either the round-off errors may cause instability of the numerical method by using of a small step size or the pay to adding solution complexity. This motivates the development of innovative methods for numerical computing of the systems more accurately and effectively. Wavelet based methods are a good candidate.

Hsiao proposed a wavelet method, which is based on Haar wavelet, to linear stiff systems [5]. Similar idea was used by Liu and Tad*é* for development of the so-called wavelet-collocation method [6] based on the Daubechies wavelet, which was constructed by Daubechies [4]. This paper will develop a wavelet-based approach to stiff systems. We propose to use the Wavelet-Galerkin scheme with the connection coefficients being computed using an algorithm that we recently rectified [7], which is a significant improvement and correction to the original work by Chen et al. [3]. Realising that previous methods, traditional and wavelet based, have significant numerical computing errors around the initial conditions, we also propose to incorporate the Lagarange multiplier based optimization into the wavelet approach to significantly reduce the computing errors. Moreover, we will compare the results from our approach and those from the Haar wavelet method.

The paper is organised as follows. To make the paper self-contained, we discuss some wavelet properties and the algorithm for computing the connection coefficient in Sect. 2. In Sect. 3, we apply the Wavelet-Galerkin scheme based on the Daubechies wavelet to a linear stiff system. In order to properly fit the initial conditions, Sect. 4 introduces the Lagrange multiplier to optimise the numerical solutions. Comparative studies between the proposed approach and the Haar approach are carried out in Sect. 5. Section 6 concludes the paper.

2 Wavelet-based scheme for solving linear differential equations

2.1 Daubechies orthonormal wavelet

In 1992, Daubechies [4] constructed a family of compactly supported orthonormal wavelets, which include members from highly localized to highly smooth. Each wavelet number is governed by a set of L coefficients $\{p_k : k = 0, \ldots, L-1\}$ through the following two-scale relations:

$$\phi(x) = \sum_{k=0}^{L-1} p_k \phi(2x - k) \tag{1}$$

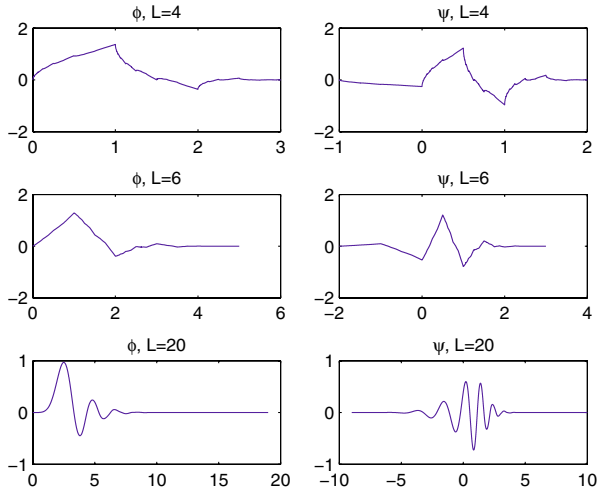


Fig. 1 Basic scaling functions and wavelets for $L = 4$, 6 and 20, respectively

and

$$\psi(x) = \sum_{k=0}^{L-1} (-1)^k p_{1-k} \phi(2x - k) \tag{2}$$

where $\phi(x)$ and $\psi(x)$ are called the scaling function and mother wavelet, respectively, with $\phi(0) = \phi(L-1) = 0$. The translations and dilations of level J for $\phi(x)$ and $\psi(x)$ are respectively defined as

$$\phi_{J,k}(x) = 2^{J/2}\phi(2^J x - k), \quad \psi_{J,k}(x) = 2^{J/2}\psi(2^J x - k). \tag{3}$$

The fundamental supports of $\phi(x)$ and $\psi(x)$ are the finite intervals $[0, L-1]$ and $[1 - L/2, L/2]$, respectively. See Fig. 1 for instance.

By [2], the scaling function $\phi(x)$ has the following property

$$\sum_{l=-\infty}^{\infty} l^n \phi(x - l) = \sum_{j=0}^{n} (-1)^j \binom{n}{j} M_j^{\phi} x^{n-j}, \quad n = 0, 1, \ldots, L/2 - 1, \tag{4}$$

where M_j^{ϕ} is the jth moment of $\phi(x)$ and is defined by the following equation

$$M_j^{\phi} = \int_{-\infty}^{\infty} x^j \phi(x) dx$$

with the initial condition $M_0^{\phi} = 1$. If denoting by $\phi^{(n)}(x)$ the nth derivative of the

scaling function $\phi(x)$, then we have

$$\phi^{(n)}(x) = \frac{d^n\phi(x)}{dx^n} = \frac{d}{dx}\phi^{(n-1)}, \quad \phi^{(0)} = \phi(x). \tag{5}$$

It is easy to know that the compact support of $\phi^{(n)}(x)$ is $[0, L-1]$. Applying (1) to (5) gives

$$\phi^{(n)} = 2^n \sum_{k=0}^{L-1} p_k \phi^{(n)}(2x-k), \quad n = 0, 1, \ldots, L/2-1. \tag{6}$$

2.2 Computation of the connection coefficients

Now let us briefly review the computation of the connection coefficients for

$$\Gamma_k^n(x) = \int_0^x \phi^{(n)}(y-k)\phi(y)dy, \tag{7}$$

which is the integral of the product of the scaling function $\phi(x)$ and its nth derivative $\phi^{(n)}(x-k)$. We will use the algorithm that we recently rectified in [7], which is a significant improvement and correction to the original work by Chen et al. [3]

By the properties of the scaling function $\phi(x)$ listed in Sect. 2.1 and straightforward computation, it is easy to verify the following relationships for $n = 0, 1, \ldots, L/2-1$ and all integers k:

$$\Gamma_k^n(x) = \Gamma_k^n(L-1) \quad \text{for } x \geq L-1, \tag{8}$$

$$\Gamma_k^n(x) = 0 \quad \text{for } |k| \geq L-1, \quad \text{or} \quad x \leq 0 \text{ or } x \leq k, \tag{9}$$

$$\Gamma_{-k}^n(L-1) = (-1)^n \Gamma_k^n(L-1), \tag{10}$$

$$\Gamma_{-k}^n(x) = (-1)^n \Gamma_k^n(L-1) \quad \text{for } x+k \geq \mathrm{L}-1. \tag{11}$$

Equations 1, 6 and 7 give

$$\Gamma_k^n(x) = 2^{n-1} \sum_{i,j=0}^{L-1} p_i p_j \Gamma_{2k+i-j}^n(2x-j). \tag{12}$$

Let

$$\Gamma^n(L-1) = [\Gamma_0^n(L-1), \quad \Gamma_1^n(L-1), \ldots, \Gamma_{L-2}^n(L-1)]^T$$

and take $x = L-1$ in (12), then from [3], we can easily obtain the values of $\Gamma_k^n(L-1)$ through the following algorithm:

$$\Gamma^n(L-1) = D\Gamma^n(L-1)$$

with normalization condition

$$\sum_{k=0}^{L-2} k^n \Gamma_k^n(L-1) = \frac{n!}{2},$$

where $D = (d_{l,m})$ for $l, m = 1, 2, \ldots, L-1$,

$$d_{l,m} = 2^{n-1}\left(\sum_{\mu_1(l,\,m)} p_i p_j + (-1)^n \sum_{\mu_2(l,\,m)} p_i p_j\right),$$

and

$$\mu_\lambda(l,m) = \{(i,j) : 0 \le i, j \le L-1 \,\&\, 2(1-1) + \mathrm{i} - \mathrm{j} = (-1)^{\lambda+1}(\mathrm{m}-1)\},\ \lambda = 1, 2.$$

After getting the values of $\Gamma_k^n(x)$ for $x = L-1$, we can compute the values of $\Gamma_k^n(x)$ for $x = 0, 1, \ldots, L-2$ and $k = 2-L, 3-L, \ldots, L-2$ using the methods described in [3,7]. Let

$$\Gamma^n = [\Gamma^n(1), \ldots, \Gamma^n(L-2)]^T \tag{13}$$

where

$$\Gamma^n(i) = [\Gamma_{i-L+2}^n(i), \ldots, \Gamma_{i-1}^n(i)]^T, \quad i = 1, 2, \ldots, L-2. \tag{14}$$

Then we have the following system for $\Gamma_k^n(x)$ with $x = 1, \ldots, L-2$ and $k = x-L+2, \ldots, x-1$.

$$\widetilde{Q}\Gamma^n = (2^{1-n}\widetilde{I} - Q)\Gamma^n = d, \tag{15}$$

where $\widetilde{I}$ is a square unit matrix of order $(L-2)^2$, $Q = (Q_{i,j})$ is a square matrix of order $(L-2)^2$ with $Q_{i,j} = (q_{i,j,k,m})$ being a $(L-2) \times (L-2)$ matrix and $q_{i,j,k,m} = p_{2i-j}\, p_{L-1-2k+m}$, and

$$d = [d^1, d^2, \ldots, d^{L-2}]^T,$$

with

$$d^i = [d((i-1)(L-2)+1), \ldots, d((i-1)(L-2)+k), \ldots, d(i(L-2)]^T, \quad (16)$$

$$d((i-1)(L-2)+k) = \sum_{\mu_2(i,k,L)} p_{i_1} p_{j_1} \Gamma^n_{2(i-(L-2)+(k-1))+i_1-j_1}(L-1), \quad (17)$$

$$\mu_2(i,k,L) = \{(i_1, j_1) \in \mu(i,k,L) : 2i - j_1 \geq L-1 \quad \text{or} \quad 2k + i_1 \leq \mathrm{L}-1\}. \quad (18)$$

It is worth mentioning that Q has eigenvalues $2^{-\lambda}(\lambda = 0, 1, \ldots, L-2)$, but 2, we can easily obtain the value of Γ^n for $n = 0$ from Eq. 15. To end the study for $n > 0$, we need the following relation

$$\sum_{l=x-L+2}^{x-1} l^n \Gamma^n_l(x) = n!\theta_1(x) - \sum_{l=L-1-x}^{L-2} l^n \Gamma^n_l(L-1), \quad (19)$$

which can be rewritten in the following vector equation form

$$[(x-L+2)^n, \ldots, (x-1)^n]\Gamma^n(x) = n!\theta_1(x) - \sum_{l=L-1-x}^{L-2} l^n \Gamma^n_l(L-1). \quad (20)$$

Combining Eq. 15 and 20 gives the value of Γ^n for $n > 0$. More precisely, for $i = 1, 2, \ldots, n$,

(1) replace the ith row of $\widetilde{Q}_{i,i} = 2^{1-n} I - Q_{i,i}$ and $\widetilde{Q}_{i,j} = -Q_{i,j}$ by $[(i-L+2)^n, \ldots, (i-1)^n]$ and a zero row vector of order $L-2$, respectively;
(2) replace $d((i-1)(L-2)+i)$, the ith element of d^i, by $n!\theta_1(i) - \sum_{l=L-1-i}^{L-2} l^n \Gamma^n_l(L-1)$.

2.3 Wavelet-based scheme for solving linear systems

Now we are ready to develop a wavelet-based scheme for solving linear systems. The scheme will be applied to linear stiff systems later in this paper.
Consider the following linear system

$$\dot{\mathbf{x}}(t) = \mathbf{x}(t)\mathbf{A}, \quad \mathbf{x}(0) = \mathbf{x}_0, \quad \mathbf{x} = (x_1, \ldots, x_n) \in \mathbf{R}^n, \quad \mathbf{A} = (a_{i,j})_{n\times n}, \quad t \in [0, N]. \quad (21)$$

The wavelet approximation of level J for the unknown $\mathbf{x}$ is given by

$$\mathbf{x}(t) = \mathbf{\Phi}_J(t)\mathbf{B}, \quad (22)$$

where $\mathbf{\Phi}_J(t) = (\phi_{J,k}(t))$ is a row vector and $\mathbf{B} = (b_{k,i})$, $k = 2-L, \ldots, 2^J - 1$, $i = 1, \ldots, n$. Substituting Eq. 22 into Eq. 21 and multiplying both sides of (21) by

$(\mathbf{\Phi}_J(t))^T$ from the left, and then integrating for t from $t = 0$ to $t = N$, we have

$$\mathbf{C}_J^1\mathbf{B} = \mathbf{C}_J^0\mathbf{B}\mathbf{A}, \tag{23}$$

where

$$\mathbf{C}_J^i = (c_{J,l,k}^i) = \left(\int_0^N \phi_{J,k}^{(i)}(t)\phi_{J,l}(t)dt\right), \quad l,k = 2-L,\ldots,2^J-1,\ i = 0,1.$$

From Eq. 7, we have

$$\begin{aligned}\mathbf{c}_{J,l,k}^1 &= 2^J[\phi(2^JN-k)\phi(2^JN-l) - \phi(-l)\phi(-k) - \Gamma_{l-k}^1(2^JN-k) + \Gamma_{l-k}^1(-k)]\\ &\equiv \widetilde{\mathbf{c}}_{J,l,k}^1 - 2^J\phi(-l)\phi(-k),\end{aligned} \tag{24}$$

$$\mathbf{c}_{J,l,k}^0 = \Gamma_{l-k}^0(2^JN-k) - \Gamma_{l-k}^0(-k), \tag{25}$$

which implies that

$$\widetilde{\mathbf{C}}_J^1\mathbf{B} - \mathbf{C}_J^0\mathbf{B}\mathbf{A} - 2^{J/2}(\Phi_J(0))^T\mathbf{x}_0 = 0 \tag{26}$$

or

$$(\mathbf{C}_J^0)^{-1}\widetilde{\mathbf{C}}_J^1\mathbf{B} - \mathbf{B}\mathbf{A} - 2^{J/2}(\mathbf{C}_J^0)^{-1}(\Phi_J(0))^T\mathbf{x}_0 = 0, \tag{27}$$

where $\widetilde{\mathbf{C}}_J^1 = (\widetilde{\mathbf{c}}_{J,l,k}^1)$. $MATLAB$ function $LYAP(\cdot)$ gives the solution of unknown matrix $\mathbf{B}$. Then from Eq. 22, we get the numerical solution for system (21).

3 Wavelet-based method for solving linear stiff systems

In this section, we switch to a case study for linear stiff systems.
Consider the following linear stiff system, which was also studied in [1,5]

$$\begin{pmatrix}\dot{x}(t)\\ \dot{y}(t)\end{pmatrix} = \begin{pmatrix}-1 & 95\\ -1 & -97\end{pmatrix}\begin{pmatrix}x(t)\\ y(t)\end{pmatrix}, \quad \begin{pmatrix}x(0)\\ y(0)\end{pmatrix} = \begin{pmatrix}1\\ 1\end{pmatrix}, \quad t \in [0,1], \tag{28}$$

Approximate the solutions of $x(t)$ and $y(t)$ by wavelet series of level J as we did in Sect. 2.3

$$x(t) = \sum_{k=2-L}^{2^J-1} x_{J,k}\phi_{J,k}(t), \quad y(t) = \sum_{k=2-L}^{2^J-1} y_{J,k}\phi_{J,k}(t), \tag{29}$$

where $x_{J,k}$ and $y_{J,k}$ are the wavelet coefficients to be determined. Then we have

$$\begin{aligned}\dot{x}(t) &= \sum_{k=2-L}^{2^J-1} x_{J,k}\frac{d\phi_{J,k}(t)}{dt},\\ \dot{y}(t) &= \sum_{k=2-L}^{2^J-1} y_{J,k}\frac{d\phi_{J,k}(t)}{dt}.\end{aligned} \tag{30}$$

Now substitute Eqs. 29 and 30 into (28). The Galerkin discretization scheme implies that

$$\begin{aligned}\sum_{k=2-L}^{2^J-1} x_{J,k}\int_0^1 \phi_{J,l}(t)\frac{d\phi_{J,k}(t)}{dt}dt = &-\sum_{k=2-L}^{2^J-1} x_{J,k}\int_0^1 \phi_{J,l}(t)\phi_{J,k}(t)dt\\ &+ 95\sum_{k=2-L}^{2^J-1} y_{J,k}\int_0^1 \phi_{J,l}(t)\phi_{J,k}(t)dt, \quad (31)\\ \sum_{k=2-L}^{2^J-1} y_{J,k}\int_0^1 \phi_{J,l}(t)\frac{d\phi_{J,k}(t)}{dt}dt = &-\sum_{k=2-L}^{2^J-1} x_{J,k}\int_0^1 \phi_{J,l}(t)\phi_{J,k}(t)dt\\ &- 97\sum_{k=2-L}^{2^J-1} y_{J,k}\int_0^1 \phi_{J,l}(t)\phi_{J,k}(t)dt, \quad (32)\end{aligned}$$

$$c^1_{J,l,k} = \int_0^1 \phi_{J,l}(t)\frac{d\phi_{J,k}(t)}{dt}dt, \quad c^0_{J,l,k} = \int_0^1 \phi_{J,l}(t)\phi_{J,k}(t)dt. \tag{33}$$

and

$$c^1_{J,l,k} = 2^J[\phi(2^J-l)\phi(2^J-k) - \phi(-l)\phi(-k) + \Gamma^1_{l-k}(-k) - \Gamma^1_{l-k}(2^J-k)], \tag{34}$$

$$c^0_{J,l,k} = \Gamma^0_{l-k}(2^J-k) - \Gamma^0_{l-k}(-k). \tag{35}$$

Substituting Eqs. 34 and 35 into 31 and 32 leads to

$$\sum_{k=2-L}^{2^J-1} x_{J,k}X^1_{J,k} + \sum_{k=2-L}^{2^J-1} y_{J,k}Y^1_{J,k} = 2^{J/2}\phi(-l), \tag{36}$$

$$\sum_{k=2-L}^{2^J-1} x_{J,k}X^2_{J,k} + \sum_{k=2-L}^{2^J-1} y_{J,k}Y^2_{J,k} = 2^{J/2}\phi(-l), \tag{37}$$

Table 1 The values of x from STHW and DAUS methods for L = 4 and J = 3

Method	EXACTS	STHW	DAUS
t = 0.0	1.000000000000000e+000	1.000000000000000e+000	8.002033059382285e−001
0.125	1.574165520629585e+000	1.170043138928742e+000	1.592634154268434e+000
0.25	1.225966227040173e+000	1.124362384204138e+000	1.221290493826090e+000
0.375	9.547834576680082e−001	9.287259243596476e−001	9.514515082537287e−001
0.5	7.435861044954686e−001	7.363958370395243e−001	7.412328383885984e−001
0.625	5.791054404620863e−001	5.766533886324167e−001	5.775516262729533e−001
0.75	4.510077705127837e−001	4.497853077422582e−001	4.501220978634202e−001
0.875	3.512452048466444e−001	3.503872997086948e−001	3.509296510848822e−001
1.0	2.735500405846427e−001	2.728449871207873e−001	2.735440290699423e−001

Table 2 The values of y from STHW and DAUS methods for L = 4 and J = 3

Method	EXACTS	STHW	DAUS
t = 0.0	1.000000000000000e+000	1.000000000000000e+000	1.197998596464597e+000
0.125	−1.656395448677543e−002	3.872369404066052e−001	−4.092022002223580e−002
0.25	−1.290490761490572e−002	8.805290247563069e−002	−1.271886319519464e−002
0.375	−1.005035218597880e−002	1.519601163671173e−002	−1.001279383485378e−002
0.5	−7.827222152583880e−003	−1.508516627133777e−003	−7.802514120888475e−003
0.625	−6.095846741706172e−003	−4.509281044915388e−003	−6.079490318693153e−003
0.75	−4.747450215924039e−003	−4.344393530535735e−003	−4.738127341331357e−003
0.875	−3.697317945754152e−003	−3.590740201298463e−003	−3.693996327369718e−003
1.0	−2.879474111417292e−003	−2.847665704994232e−003	−2.879410832313992e−003

for $l = 2 - L, \ldots, 2^J - 1$, where

$$
\begin{aligned}
X^1_{J,k} &= 2^J(\phi(2^J - l)\phi(2^J - k) + \Gamma^1_{l-k}(-k) - \Gamma^1_{l-k}(2J - k)) \\
&\quad + \Gamma^0_{l-k}(2^J - k) - \Gamma^0_{l-k}(-k), \\
Y^1_{J,k} &= -95(\Gamma^0_{l-k}(2^J - k) - \Gamma^0_{l-k}(-k)), \\
X^2_{J,k} &= \Gamma^0_{l-k}(2^J - k) - \Gamma^0_{l-k}(-k), \\
Y^2_{J,k} &= 2^J(\phi(2^J - l)\phi(2^J - k) + \Gamma^1_{l-k}(-k) - \Gamma^1_{l-k}(2J - k)) \\
&\quad + 97(\Gamma^0_{l-k}(2^J - k) - \Gamma^0_{l-k}(-k)).
\end{aligned}
$$

Then, we can get the values of $x_{J,k}$ and $y_{J,k}$ from Eqs. (36) and (37). After that, we get the numerical solution of system (28) through (29). Comparisons between the exact solution and the numerical solution are shown in Tables 1 and 2, and Figs. 2 and 3.

4 Optimizing solutions via the Lagrange multiplier

It is easy to find that the numerical solutions of system (28) are good in the entire range of time t except for the initial time $t = 0$. This is because $x_{J,k}$ and $y_{J,k}$ determined by Eqs. 36 and 37 generally do not satisfy the initial condition and also because

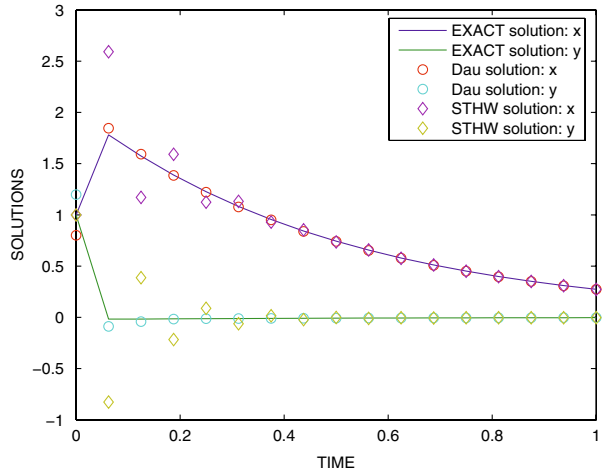


Fig. 2 Comparison of the exact solutions with the results obtained by Haar wavelet and Daubechies wavelet with $J = N = 4$, $L = 4$

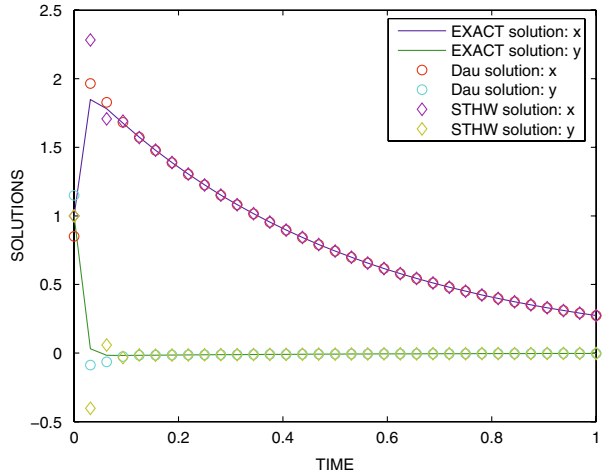


Fig. 3 Comparison of the exact solutions with the results obtained by Daubechies wavelet with $J = N = 5$, $L = 4$

the system is very stiff near the initial time. In the following, we use the constrained optimization method via the Lagrange multipliers to reduce the computing errors significantly. Constrained optimisation is a powerful and versatile technique for solving cost minimization problem.

Let us start from Eqs. 31–33 and rearrange them into the matrix equation form as

$$\mathbb{A}\mathbb{X} = \mathbf{0} \tag{38}$$

where

$$\mathbb{A} = \begin{pmatrix} \mathbf{a}_{11} & \mathbf{a}_{12} \\ \mathbf{a}_{21} & \mathbf{a}_{22} \end{pmatrix}$$

and

$$\begin{aligned} &\mathbf{a}_{11} = (c^1_{J,l,k} + c^0_{J,l,k})_{l,k}, \quad \mathbf{a}_{12} = (-95c^0_{J,l,k})_{l,k}, \\ &\mathbf{a}_{J,l,k} = (c^0_{J,l,k})_{l,k}, \quad \mathbf{a}_{22} = (c^1_{J,l,k} + 97c^0_{J,l,k})_{l,k}, \quad l,k = 2-L,\ldots,2^J-1, \\ &\mathbb{X} = [\mathbf{X}_1, \mathbf{X}_2]^T, \quad \mathbf{X}_1 = (x_{J,k})_k, \quad \mathbf{X}_2 = (y_{J,k})_k. \end{aligned}$$

Then the cost function to be minimized is defined by

$$G(\mathbb{X}) = (\mathbb{A}\mathbb{X})^T \mathbb{A}\mathbb{X}, \tag{39}$$

the constraints have the following form

$$\mathbf{X}_1^T \mathbf{Z}_1 = 1, \quad \mathbf{X}_2^T \mathbf{Z}_2 = 1, \tag{40}$$

and the objective function is given by

$$F(\mathbb{X}, \lambda_1, \lambda_2) = (\mathbb{A}\mathbb{X})^T \mathbb{A}\mathbb{X} + \lambda_1(\mathbf{X}_1^T \mathbf{Z}_1 - 1) + \lambda_2(\mathbf{X}_2^T \mathbf{Z}_2 - 1), \tag{41}$$

where $\lambda_i, i = 1, 2$ are the Lagrange multipliers. Appliying optimization theory, we know that Eq. 41 implies

$$\begin{pmatrix} 2\mathbb{A}^T\mathbb{A} & \mathbf{Z}_1 & \mathbf{Z}_2 \\ \mathbf{Z}_1^T & 0 & 0 \\ \mathbf{Z}_2^T & 0 & 0 \end{pmatrix} \begin{pmatrix} \mathbb{X} \\ \lambda_1 \\ \lambda_2 \end{pmatrix} = \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix}. \tag{42}$$

From Eq. 42, we can get the optimized values of $x_{J,k}$ and $y_{J,k}$. Then, we get the optimized numerical solution for system (28) through (29). Because of the use of optimization, the solutions obtained here are better than those obtained by using the Daubechies wavelet method without optimization near the initial time. Refer to the results shown in Tables 3 and 4 and Figs. 4 and 5.

5 Discussions

The results (such as Figs. 2, 3 and 5) reveal that the numerical solution can be improved by increasing either the scale parameter J or the order parameter L. We also make the following observations.

- Figure 2 shows that the method based on Daubechies wavelet (Dau) is much better than the single-term Haar wavelet method (STHW) in the entire interval [0, 1], except at the initial time instant $t = 0$. This has been explained at the beginning of

Table 3 The values of x from OPDAUS and DAUS methods for L=4 and J = 3

Method	EXACTS	OPDAUS	DAUS
t = 0.0	1.000000000000000e+000	9.999999999999992e−001	8.002033059382285e−001
0.125	1.574165520629585e+000	1.575024208288943e+000	1.592634154268434e+000
0.25	1.225966227040173e+000	1.203632754000326e+000	1.221290493826090e+000
0.375	9.547834576680082e−001	9.373647247549073e−001	9.514515082537287e−001
0.5	7.435861044954686e−001	7.302991587337776e−001	7.412328383885984e−001
0.625	5.791054404620863e−001	5.695124986482306e−001	5.775516262729533e−001
0.75	4.510077705127837e−001	4.447861731175914e−001	4.501220978634202e−001
0.875	3.512452048466444e−001	3.481719437454351e−001	3.509296510848822e−001
1.0	2.735500405846427e−001	2.733716955045099e−001	2.735440290699423e−001

Table 4 The values of y from OPDAUS and DAUS methods for L=4 and J = 3

Method	EXACTS	OPDAUS	DAUS
t = 0.0	1.000000000000000e+000	9.999999999999999e-001	1.197998596464597e+000
0.125	−1.656395448677543e−002	−4.460043847567377e−002	−4.092022002223580e−002
0.25	−1.290490761490572e−002	−1.286821405778022e−002	−1.271886319519464e−002
0.375	−1.005035218597880e−002	−1.023043961806455e−002	−1.001279383485378e−002
0.5	−7.827222152583880e−003	−8.001947295498926e−003	−7.802514120888475e−003
0.625	−6.095846741706172e−003	−6.265708358594185e−003	−6.079490318693153e−003
0.75	−4.747450215924039e−003	−4.915156542873738e−003	−4.738127341331357e−003
0.875	−3.697317945754152e−003	−3.863690295246785e−003	−3.693996327369718e−003
1.0	−2.879474111417292e−003	−2.982567085698608e−003	−2.879410832313992e−003

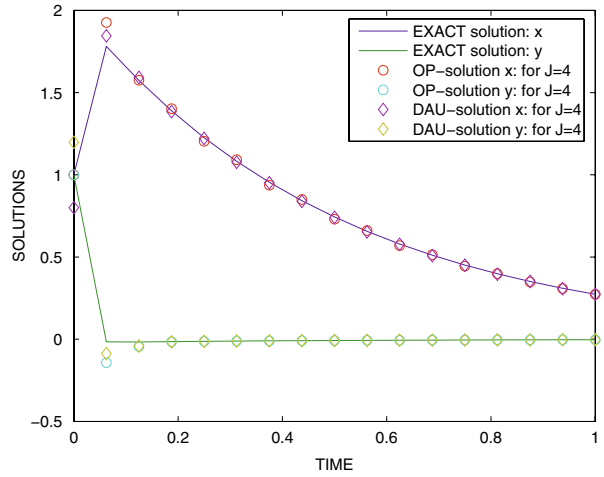


Fig. 4 Comparison of the exact solutions with the results obtained by Daubechies wavelet and OP-Dau with $J = 4$, $L = 4$

Sect. 4 for the first method, and is also because we have fixed $x(0) = 1$, $y(0) = 1$ for the STHW method.

- Figures 4 and 5 show that the Daubechies wavelet method (DAU) does not work well at $t = 0$ since the considered system is very stiff near the initial time. However,

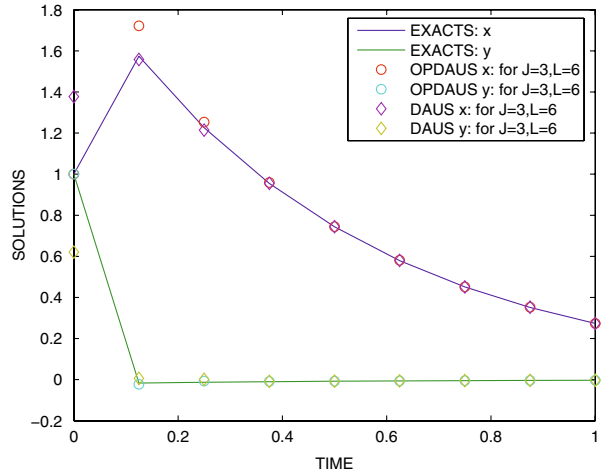


Fig. 5 Comparison of the exact solutions with the results obtained by Daubechies wavelet and OP-Dau with $J = 3$, $L = 6$

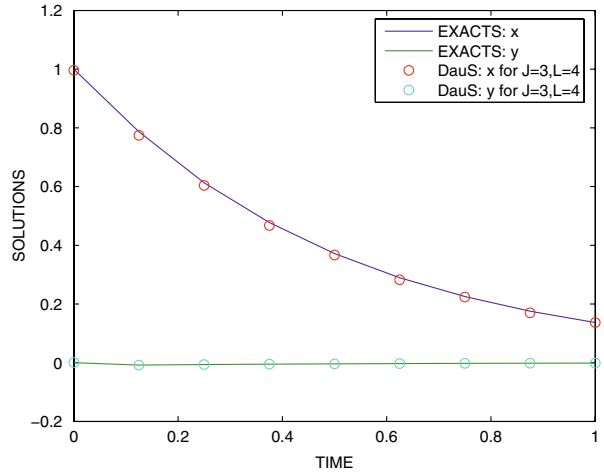


Fig. 6 Comparison of the exact solutions with the results obtained by Daubechies wavelet $J = 3$, $L = 4$ and initial condition $x(0) = 1$, $y(0) = 0$

wavelet-based method incorporated with optimization via the so-called Lagrange multiplier (OPDAU) can significantly improve the numerical results at the stiff part.

- It is also seen from Figs. 4 and 5 that for times far away from the initial time instant,
  - the DAU method outperforms other methods; and
  - the solutions from the Lagrange multiplier method (OPDAU) are also quite acceptable, justifying the applicability of the proposed approach to general linear stiff systems.

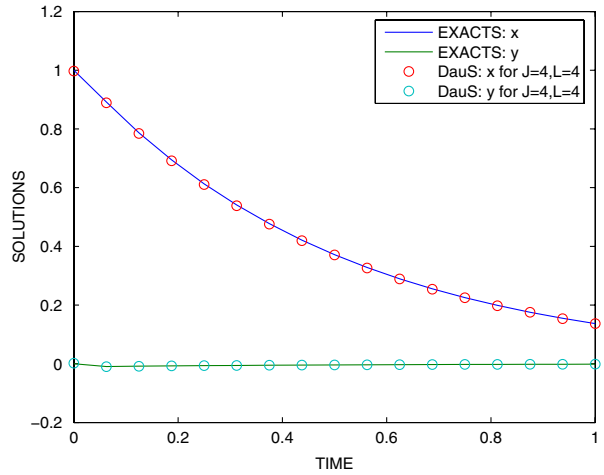
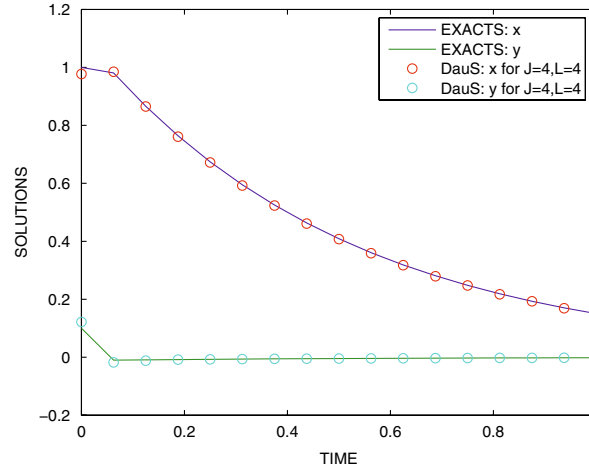


Fig. 7 Comparison of the exact solutions with the results obtained by Daubechies wavelet $J = 4$, $L = 4$ and initial condition $x(0) = 1$, $y(0) = 0$

Fig. 8 Comparison of the exact solutions with the results obtained by Daubechies wavelet $J = 4$, $L = 4$ and initial condition $x(0) = 1$, $y(0) = 1/10$



- Figures 6–9 show that the nearer the initial value $y(0)$ approaches to 0, the better the numerical results of the system. And if $y(0) = 0$, the system is no longer a stiff system.
- For the single-term Haar wavelet method, one has to redo the whole computing procedure whenever N changes. In contrast, the computation of the connection coefficients in this work is determined only by the order parameter L, implying that we can compute the connection coefficients $\Gamma_k^n(x)$ off-line to build a library which can be used later in on-line computation. This means that the Wavelet-Galerkin scheme adopted in this work can significantly reduce the on-line computing time for numerical solutions of differential equations/systems.

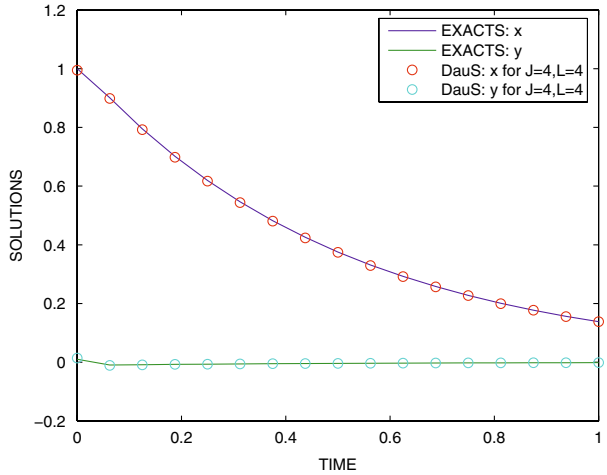


Fig. 9 Comparison of the exact solutions with the results obtained by Daubechies wavelet $J = 4$, $L = 4$ and initial condition $x(0) = 1$, $y(0) = 1/100$

6 Conclusion

In this work, the Daubechies wavelet scheme has been adopted to solve linear stiff systems and satisfactory approximations have been obtained. We have shown that incorporation of the Lagrange multiplier based optimization into the wavelet approach can improve the numerical solutions significantly at or near the initial conditions. With the proposed approach, the solutions for times far away from the initial time instant are also quite acceptable.

Acknowledgements The authors Zhang, Tadé, and Tian would like to acknowledge the support from Australian Research Council (ARC) under Discovery Projects grants DP0559111 and DP0770420.

Appendix: Values of $\theta_1(x)$ for R $L = 4$ and $L = 6$

Table 5 Values of $\theta_1(x)$ for L = 4 and L = 6

x	L = 4	L = 6
0.0	0.000000000000000e+000	0.000000000000000e+000
0.5	2.901709006307399e−001	1.413146044791601e−001
1.0	8.496793685588863e−001	6.007415698311157e−001
1.5	1.077350269189626e+000	1.052908231945552e+000
2.0	1.016346035225553e+000	1.096711447148340e+000
2.5	9.985042339640733e−001	9.850661449026523e−001
3.0	1.000000000000000e+000	9.854867262024150e−001
3.5		1.003318342528402e+000
4.0		9.996590890072497e−001
4.5		9.999915083333543e−001
5.0		1.000000000000000e+000

References

1. R. Alt, J. Comp. Appl. Math. **4**, 29–35 (1978)
2. G. Beylkin, SIAM J. Numer. Anal. **29**, 1716 (1992)
3. M. Chen, C. Hwang, Y. Shih, Int. J. Numer. Methods Eng. **39**, 2921 (1996)
4. I. Daubechies, *Ten Lectures on Wavelets*, (SIAM, Philadelphia, 1992)
5. C.H. Hsiao, Math. Comp. Simu. **64**, 561 (2004).
6. Y. Liu, M.O. Tad*é*, Powder Technol. **139**, 61 (2004)
7. T. Zhang, Y.-C. Tian, M.O. Tadé, J. Utomo, Comments on "The Computation of Wavelet-Galerkin Approximation on a Bounded Interval" Int. J. Num. Methods Eng. (2007). (Accepted)