Available online at www.sciencedirect.com
ScienceDirect
Applied Mathematics and Computation 189 (2007) 208–220
APPLIED MATHEMATICS AND COMPUTATION
www.elsevier.com/locate/amc

An adaptive wavelet method for nonlinear differential-algebraic equations ☆

Jing Gao *, Yao-Lin Jiang

Department of Mathematical Sciences, Xi'an Jiaotong University, Xi'an 710049, PR China

Abstract

A multi-resolution collocation method with specially designed spline wavelet is presented to numerically solve a system of nonlinear differential-algebraic equations of index one in circuit simulation. As the wavelet collocation method is applied to such system, we will have to solve large nonlinear algebraic equations. For the algebraic equations, we give a modified Newton type iterative method which bases on two wavelet adaptive techniques. We prove the convergence, the stability and the complexity on the adaptive method reported here. The new approach could effectively reduce computational costs and storage requirements to large extent. Numerical experiments are given to further illustrate the method's principle.

Keywords: Nonlinear differential-algebraic equations; Newton iteration; Adaptive wavelet techniques

1. Introduction

It is well known that in engineering application fields we often need to find dynamic responses of large systems, such as in nonlinear electronic circuits and so on. Most of these systems may be modelled by differential-algebraic equations (DAEs), see [3].

In the paper we mainly consider nonlinear DAEs given as follows:

$$\begin{cases} \frac{\mathrm{d}x(t)}{\mathrm{d}t} = f(x(t), y(t)), & x(0) = x_0, \\ 0 = g(x(t), y(t)), & t \in [0, L], \end{cases} \tag{1.1}$$

where t is time variable, $\Omega = [0, L]$ is finite, $x_0 = (x_0^1, x_0^2, \ldots, x_0^n)^{\mathrm{T}}$ is a consistent initial value, $f = (f_1, f_2, \ldots, f_n)^{\mathrm{T}}$ and $g = (g_1, g_2, \ldots, g_m)^{\mathrm{T}}$ are two known continuous functions, $x(t) = (x_1(t), x_2(t), \ldots, x_n(t))^{\mathrm{T}} \in \mathbf{R}^n$ and $y(t) = (y_1(t), y_2(t), \ldots, y_m(t))^{\mathrm{T}} \in \mathbf{R}^m$ for all $t \in \Omega$ are to be computed. We assume that (1.1) is index-one, that is, the nonlinear function $g(x, y)$ is solvable with respect to y.

☆ This work was supported in part by the National Natural Science Foundation of China (NSFC) under Grant 10171080 and 60472003, and by the National Key Basic Research Program of China (973 program) under Grant 2005CB321701.
* Corresponding author.
E-mail addresses: jgao@mail.xjtu.edu.cn (J. Gao), yljiang@mail.xjtu.edu.cn (Y.-L. Jiang).

0096-3003/$ - see front matter
doi:10.1016/j.amc.2006.11.102

Many numerical methods, such as Runge–Kutta method [6] and waveform relaxation (WR) [7,8], have been successfully applied to solve DAEs. However, under some situations these methods possibly suffer from the difficulty in effectively handling the singularity, which often occurs in simulations of high-speed circuits. As we know, due to the advantageous properties of localization in both space and frequency domains, wavelet technique has been a great tool for adaptivity and multi-resolution schemes to obtain solutions of systems which vary dynamically both in space and time. Especially, the cubic spline wavelet collocation approach in [4,10] is not only powerful in treating singularity, but also has $O(h^4)$ convergence rate, where h is the step length. Dian Zhou et al. have succeeded in applying the wavelet method to solve ordinary differential equations (ODEs) in circuit simulation [11,12]. In [1,2,5,9], the adaptive wavelet iterative method have arisen for PDEs. However, no research exists in DAEs, and no further analysis appears on the iterations of the resulting nonlinear algebraic equations, to the best of our knowledge.

The main purpose of the paper is to develop a wavelet multi-resolution scheme for nonlinear DAEs in circuit simulation. We propose a modified Newton adaptive iterative method by adopting two adaptive properties of wavelets to solve resulting nonlinear algebraic equations. The convergence and the stability on the proposed iterative process is analyzed theoretically. Comparing with the original Newton method, the proposed Newton adaptive iterative method, which is favorable for the solutions of large nonlinear DAEs, could accelerate iterative process and reduce computation and storage costs. Numerical experiments are provided to further show the efficiency of the method.

The organization of the paper is as follows. Section 2 firstly outlines the general form of multi-resolution wavelet collocation and then extends it to solve nonlinear DAEs. Meanwhile, the modified Newton adaptive iterative process is formulated. The proofs of the convergence and the stability on the relaxation-based method are shown in Section 3. Section 4 demonstrates the proposed process with numerical experiments. Conclusion is given in Section 5.

2. The modified Newton adaptive iteration

Without loss of generality, we assume that a circuit system is described by nonlinear DAEs of index one like (1.1). We are interested in the solution of $x(t)$ and $y(t)$ in a finite interval $\Omega = [0, L]$. The idea of multi-resolution wavelet collocation is that, after replacing the functions $x(t)$, $y(t)$, $f(x(t), y(t))$, $g(x(t), y(t))$ and the differential operator $\frac{d}{dt}$ of (1.1) with their cubic spline wavelet series, we acquire an algebraic equation system, whose unknowns are the corresponding wavelet coefficients of these wavelet series.

2.1. Wavelet collocation discretization of nonlinear DAEs

Define Sobolev space $H^2(I) = \{f(x), x \in I | \|f^{(i)}\|_2 < \infty,\ i = 1, 2, 3\}$. Introduce an approximation subspace $V_J \subset H^2(I)$, where $J \geqslant 0$ and a fixed interval $\Omega = [0, L]$ [4,11,12],

$$V_J = V_0 \oplus W_0 \oplus W_1 \oplus \cdots \oplus W_{J-1}, \tag{2.1}$$

where

$$\begin{aligned} V_0 &= \operatorname{span}\{\eta_1(t), \eta_2(t), \eta_2(L-t), \eta_1(L-t), \phi_{-1,-1}(t), \ldots, \phi_{-1,L-4}(t), \phi_{-1,L-3}(L-t)\}, \\ W_i &= \operatorname{span}\{\psi_{i,-1}(t), \psi_{i,0}(t), \ldots, \psi_{i,n_0-2}(t)\}, \quad 0 \leqslant i \leqslant J-1. \end{aligned} \tag{2.2}$$

The concrete properties of these functions $\eta_1(t)$, $\eta_2(t)$, $\phi(t)$, $\psi(t)$ can be found in [4]. Here, we only give the definitions as follows:

$$\begin{aligned} \eta_1(t) &= (1-t)_+^3, \\ \eta_2(t) &= 2t_+ - 3t_+^2 + \frac{7}{6}t_+^3 - \frac{4}{3}(t-1)_+^3 + \frac{1}{6}(t-2)_+^3, \\ \phi(t) &= N_4(t) = \frac{1}{6}\sum_{l=0}^{4}\binom{4}{1}(-1)^l(t-l)_+^3, \\ \psi(t) &= -\frac{3}{7}\phi(2t) + \frac{12}{7}\phi(2t-1) - \frac{3}{7}\phi(2t-2), \end{aligned} \tag{2.3}$$

where

$$t_+^n = \begin{cases} t^n, & t \geqslant 0, \\ 0, & t < 0. \end{cases} \tag{2.4}$$

When $J \to \infty$, the space $V_J \to H^2(I)$, V_J is the multilevel decomposition of the space $H^2(I)$. Expand the unknown function $x(t) \in H^2(I)$ in V_J. Denote $n_j = 2^j * L + 3$ and $P_J x(t)$ by the interpolation approximation of the function $x(t)$ in V_J, we have

$$\begin{aligned} P_{V_J}x(t) = x_J(t) &= I_{b,J}x(t) + \hat{x}_{-1,-1}\phi_b(t) + \sum_{k=0}^{L-4}\hat{x}_{-1,k}\phi_k(t) + \hat{x}_{-1,L-3}\phi_b(L-t) + \sum_{j=0}^{J-1}\left[\sum_{k=-1}^{n_j-2}\hat{x}_{j,k}\psi_{j,k}(t)\right] \\ &= x_{-1}(t) + \sum_{j=0}^{J-1}x_j(t), \end{aligned} \tag{2.5}$$

where $I_{b,J}x(t)$ is the boundary approximation of $x_J(t)$, $I_{b,J}x(t) = \alpha_1\eta_1(2^Jx) + \alpha_2\eta_2(2^Jx) + \alpha_3\eta_2(2^J(L-x)) + \alpha_4\eta_1(2^J(L-x))$, $\alpha_i(1 \leqslant i \leqslant 4)$ in [4]. The spatial approximation order is $\|x_J(t) - x(t)\|_{H^2(\Omega)} \leqslant 2^{-4J}$. The wavelet coefficients of $x_J(t)$ is

$$\hat{x}_J = (\hat{x}_{-1,-3}, \ldots, \hat{x}_{-1,L-2}, \hat{x}_{0,-1}, \ldots, \hat{x}_{0,n_0-2}, \ldots, \hat{x}_{J,-1}, \ldots, \hat{x}_{J,k}, \ldots, \hat{x}_{J,n_J-2})^{\mathrm{T}}. \tag{2.6}$$

The wavelet series of $x_J(t)$ satisfies

$$\begin{cases} P_{V_J}x(t_k^{(-1)}) = x(t_k^{(-1)}), & 1 \leqslant k \leqslant L+3, \\ P_{V_J}x(t_k^{(j)}) = x(t_k^j), & j \geqslant 0,\ -1 \leqslant k \leqslant n_j - 2,\ 0 \leqslant j \leqslant J-1. \end{cases} \tag{2.7}$$

The corresponding interpolation points t_k^i are

$$\begin{aligned} &t_1^{-1} = 0, \quad t_2^{(-1)} = \frac{1}{2}, \quad t_k^{-1} = k-2 \ (3 \leqslant k \leqslant L+1), \quad t_{L+2}^{(-1)} = L - \frac{1}{2}, \quad t_{L+3}^{(-1)} = L, \\ &t_{-1}^i = \frac{1}{2^{i+2}}, \quad t_k^i = \frac{k+1.5}{2^i} \ \ 0 \leqslant k < n_i - 2, \quad t_{n_i-2}^i = L - \frac{1}{2^{i+2}}. \end{aligned} \tag{2.8}$$

Let $H^2(\Omega)$ be a Sobolev space which basically contains functions with square integrable second derivatives. We introduce an approximation subspace $V_J \subset H^2(\Omega)$ for a given integer $J \geqslant 0$, which denotes the wavelet decomposition level, and a fixed interval $\Omega = [0, L]$, consisting of scaling functions and wavelet functions.

Substituting them into the DAEs, we obtain the nonlinear discrete algebraic system as follows:

$$\begin{cases} \widetilde{A}\hat{x} = F(\hat{x}, \hat{y}), \\ 0 = G(\hat{x}, \hat{y}), \end{cases} \tag{2.9}$$

where $\widetilde{A} = \mathrm{diag}(A, \ldots, A) \in \mathbf{R}^{nN \times nN}$ in which

$$A = \{(a_{m,l})\}_{1 \leqslant m,l \leqslant N} = \begin{pmatrix} \frac{\mathrm{d}\psi_1(t_1)}{\mathrm{d}t} & \cdots & \frac{\mathrm{d}\psi_N(t_1)}{\mathrm{d}t} \\ \vdots & \vdots & \vdots \\ \frac{\mathrm{d}\psi_1(t_N)}{\mathrm{d}t} & \cdots & \frac{\mathrm{d}\psi_N(t_N)}{\mathrm{d}t} \end{pmatrix}_{N \times N} \tag{2.10}$$

is an invertible derivative matrix. The vectors $F(\hat{x}, \hat{y})$ and $G(\hat{x}, \hat{y})$ respectively consist of the values of two vector functions $f(x, y)$ and $g(x, y)$ at all collocation points ordered in the same way. It says $F(\hat{x}, \hat{y}) = (F_1, F_2, \ldots, F_n)^{\mathrm{T}}_{nN \times 1}$ where

$$F_k = \begin{pmatrix} f_k(\sum\hat{x}_i\psi_i(t_1),\ \sum\hat{y}_i\psi_i(t_1)) \\ f_k(\sum\hat{x}_i\psi_i(t_2),\ \sum\hat{y}_i\psi_i(t_2)) \\ \vdots \\ f_k(\sum\hat{x}_i\psi_i(t_N),\ \sum\hat{y}_i\psi_i(t_N)) \end{pmatrix} = \begin{pmatrix} f_{k,1} \\ f_{k,2} \\ \vdots \\ f_{k,N} \end{pmatrix}_{N \times 1}. \tag{2.11}$$

Similarly, the vector $(G(\hat{x}, \hat{y}))_{nN \times 1}$ has the same form.

2.2. The Newton iterative solution of nonlinear algebraic equations

We choose the Newton-type methods to solve the nonlinear equation system. In the paper, in order to distinguish the modified Newton adaptive iterative method from the nonadaptive classic Newton method, the latter now is referred to as the original Newton iterative method.

Let

$$\hat{z}^{(k)} = \left(\hat{x}^{(k)}_{1,1}, \hat{x}^{(k)}_{1,2}, \ldots, \hat{x}^{(k)}_{1,N}, \hat{x}^{(k)}_{2,1}, \hat{x}^{(k)}_{2,2}, \ldots, \hat{x}^{(k)}_{2,N}, \ldots, \hat{x}^{(k)}_{n,N}, \hat{y}^{(k)}_{1,1}, \hat{y}^{(k)}_{1,2}, \ldots, \hat{y}^{(k)}_{m,N}\right)^{\mathrm{T}}_{(n+m)N\times 1}. \tag{2.12}$$

Then, the original Newton iterative form is

$$\hat{z}^{(k+1)} = \hat{z}^{(k)} - \Delta\hat{z}^{(k)}, \tag{2.13}$$

where the $(n+m)N$-dimensional vector $\Delta\hat{z}^{(k)}$ satisfies the relationship as follows:

$$Q_{(n+m)N\times(n+m)N} \times \Delta\hat{z}^{(k)} + H_{(n+m)N\times 1} = 0 \tag{2.14}$$

in which Q is the corresponding Jacobian matrix.

Denote $w = \widetilde{A}\hat{x} - F(\hat{x},\hat{y})$, we have

$$w = \widetilde{A}\begin{pmatrix} \hat{x}_1 \\ \hat{x}_2 \\ \vdots \\ \hat{x}_n \end{pmatrix} - \begin{pmatrix} F_1(\hat{x},\hat{y}) \\ F_2(\hat{x},\hat{y}) \\ \vdots \\ F_n(\hat{x},\hat{y}) \end{pmatrix} = \begin{pmatrix} A\hat{x}_1 - F_1(\hat{x},\hat{y}) \\ A\hat{x}_2 - F_2(\hat{x},\hat{y}) \\ \vdots \\ A\hat{x}_n - F_n(\hat{x},\hat{y}) \end{pmatrix} = \begin{pmatrix} w_1 \\ w_2 \\ \vdots \\ w_{nN} \end{pmatrix}. \tag{2.15}$$

Moreover, we have also

$$H(\hat{x},\hat{y}) = \begin{pmatrix} \widetilde{A}\hat{x} - F(\hat{x},\hat{y}) \\ G(\hat{x},\hat{y}) \end{pmatrix}_{(n+m)N\times 1} \tag{2.16}$$

and

$$\begin{aligned} Q &= \begin{pmatrix} \frac{\partial w_1}{\partial \hat{x}_{1,1}}, \frac{\partial w_1}{\partial \hat{x}_{1,2}}, \ldots, & \frac{\partial w_1}{\partial \hat{x}_{n,N}}, \frac{\partial w_1}{\partial \hat{y}_{1,1}}, & \frac{\partial w_1}{\partial \hat{y}_{1,2}}, \ldots, \frac{\partial w_1}{\partial \hat{y}_{m,N}} \\ \frac{\partial w_2}{\partial \hat{x}_{1,1}}, \frac{\partial w_2}{\partial \hat{x}_{1,2}}, \ldots, & \frac{\partial w_2}{\partial \hat{x}_{n,N}}, \frac{\partial w_2}{\partial \hat{y}_{1,1}}, & \frac{\partial w_2}{\partial \hat{y}_{1,2}}, \ldots, \frac{\partial w_2}{\partial \hat{y}_{m,N}} \\ & \vdots & \\ \frac{\partial w_{nN}}{\partial \hat{x}_{1,1}}, \frac{\partial w_{nN}}{\partial \hat{x}_{1,2}}, \ldots, & \frac{\partial w_{nN}}{\partial \hat{x}_{n,N}}, \frac{\partial w_{nN}}{\partial \hat{y}_{1,1}}, & \frac{\partial w_{nN}}{\partial \hat{y}_{1,2}}, \ldots, \frac{\partial w_{nN}}{\partial \hat{y}_{m,N}} \\ \left(\frac{\partial G}{\partial \hat{x}}\right)_{mN\times nN}, & & \left(\frac{\partial G}{\partial \hat{y}}\right)_{mN\times mN} \end{pmatrix}_{(n+m)N\times(n+m)N} \\ &= \begin{pmatrix} A - \frac{\partial F_1}{\partial \hat{x}_1}, -\frac{\partial F_1}{\partial \hat{x}_2}, \ldots, & -\frac{\partial F_1}{\partial \hat{x}_n}, -\frac{\partial F_1}{\partial \hat{y}_1}, & -\frac{\partial F_1}{\partial \hat{y}_2}, \ldots, -\frac{\partial F_1}{\partial \hat{y}_m} \\ -\frac{\partial F_2}{\partial \hat{x}_1}, A - \frac{\partial F_2}{\partial \hat{x}_2}, \ldots, & -\frac{\partial F_2}{\partial \hat{x}_n}, -\frac{\partial F_2}{\partial \hat{y}_1}, & -\frac{\partial F_2}{\partial \hat{y}_2}, \ldots, -\frac{\partial F_2}{\partial \hat{y}_m} \\ & \vdots & \\ -\frac{\partial F_n}{\partial \hat{x}_1}, -\frac{\partial F_n}{\partial \hat{x}_2}, \ldots, & A - \frac{\partial F_n}{\partial \hat{x}_n}, -\frac{\partial F_n}{\partial \hat{y}_1}, & -\frac{\partial F_n}{\partial \hat{y}_2}, \ldots, -\frac{\partial F_N}{\partial \hat{y}_m} \\ \left(\frac{\partial G}{\partial \hat{x}}\right)_{mN\times nN}, & & \left(\frac{\partial G}{\partial \hat{y}}\right)_{mN\times mN} \end{pmatrix}_{(n+m)N\times(n+m)N} \\ &= \begin{pmatrix} \widetilde{A} - \frac{\partial F}{\partial \hat{x}} & -\frac{\partial F}{\partial \hat{y}} \\ \frac{\partial G}{\partial \hat{x}} & \frac{\partial G}{\partial \hat{y}} \end{pmatrix}_{(n+m)N\times(n+m)N}. \end{aligned} \tag{2.17}$$

For the iteration method, the mainly computational burden comes from solving the equation at each iteration step. If the vector $\hat{z}^{(k)}$ is sparse, namely, it has many zero elements relative to its dimension, then the Jacobian matrix Q is sparse. Similarly, the vector H has also many zero elements. As a result, the computation time and storage cost can be economized at each iteration step. In fact, as the wavelet coefficients of the functions $x(t)$ and $y(t)$, the vectors $\hat{x}$ and $\hat{y}$ have such sparsity which could always improve the iterative process. Taking the properties of the spline wavelets with Newton iteration, we can really construct the so-called adaptive iterative process.

In next section, we will discuss how to exert the wavelet adaptive properties for the design on the modified Newton adaptive process. For the purpose, some wavelet adaptive properties should be stated here.

2.3. Adaptive properties with the cubic spline wavelet bases

Denote $\hat{x}^k_{i,l}$ the wavelet coefficients of sub-element function $x^k(t)$ in vector function $x(t)$ where i and l are the scale and dilation indexes respectively. First, let $v_i > 0$, $N = 2^J * L + 3$, and

$$\hat{x}^{*k}_{i,l} = \begin{cases} \hat{x}^k_{i,l}, & |\hat{x}^k_{i,l}| \geqslant v_i, \\ 0, & |\hat{x}^k_{i,l}| < v_i. \end{cases} \tag{2.18}$$

Define

$$\begin{aligned} x^{*k}_J(t) &= \hat{x}^k_{-1,-3}\eta_1(t) + \hat{x}^k_{-1,-2}\eta_2(t) + \hat{x}^k_{-1,-1}\phi_b(t) + \sum_{l=0}^{L-4}\hat{x}^k_{-1,l}\phi_l(t) + \hat{x}^k_{-1,L-3}\phi_b(L-t) \\ &\quad + \hat{x}^k_{-1,L-2}\eta_2(L-t) + \hat{x}^k_{-1,L-1}\eta_1(L-t) + \sum_{i=0}^{J-1}\sum_{l=-1}^{n_i-2}\hat{x}^{*k}_{i,l}\psi_{i,l-2} \end{aligned} \tag{2.19}$$

and

$$\begin{aligned} x^k_J(t) &= \hat{x}^k_{-1,-3}\eta_1(t) + \hat{x}^k_{-1,-2}\eta_2(t) + \hat{x}^k_{-1,-1}\phi_b(t) + \sum_{l=0}^{L-4}\hat{x}^k_{-1,l}\phi_l(t) + \hat{x}^k_{-1,L-3}\phi_b(L-t) \\ &\quad + \hat{x}^k_{-1,L-2}\eta_2(L-t) + \hat{x}^k_{-1,L-1}\eta_1(L-t) + \sum_{i=0}^{J-1}\sum_{l=-1}^{n_i-2}\hat{x}^k_{i,l}\psi_{i,l-2} \end{aligned} \tag{2.20}$$

Based on the above definitions, we turn to giving the two adaptive properties on the cubic spline wavelet.

2.3.1. Adaptive wavelet sparsity representation for function $x(t)$

Now, we study the characteristics of the function $x(t)$ approximated by the wavelet series $x^J(t) = (x^k_J(t), 1 \leqslant k \leqslant n)^{\mathrm{T}}_{n\times 1}$. Due to time-frequency localization property of wavelet functions, the wavelet coefficients decay quickly to zero in smooth regions, and large only in region where the function gradient is large. From [10], if function $f \in Lip(\alpha)$, the wavelet coefficient $\hat{x}_{i,l} = \mathrm{O}(2^{-i\alpha})$. Hence, we can reduce a quantity of the wavelet coefficients in order to save the operation time and the memory space. According to the properties of the cubic spline wavelets, we know the wavelet coefficients in the corresponding wavelet expansion will be less than the error tolerance v in a large part of the solution domain if the decomposition level J becomes larger. Furthermore, the wavelet coefficients $\hat{x}^k_{i,l}$ can be set zeros if the magnitudes of $x^k(t)$ at collocation points is less than some given error tolerance v. Thus, many terms of wavelet series may be discarded in the wavelet expansion of $x^k(t)$. Namely, the wavelet coefficients vector $\hat{x} = (\hat{x}^k_m,\ 1 \leqslant m \leqslant N,\ 1 \leqslant k \leqslant n)^{\mathrm{T}}_{nN\times 1}$ has more zero elements with the level J increasing, where $N = 2^J L + 3$. This fact will be used to achieve adaptivity for the Newton iterative solution to accelerate convergence and reduce computational cost. For the clarity of the formulation, we use the hard thresholding v in the next section.

2.3.2. Adaptive multilevel representation

Lemma 1. *Let* $x(t) \in H^2(\Omega)$, $x_J(t) = (x^k_J(t), 1 \leqslant k \leqslant n)^{\mathrm{T}}_{n\times 1}$, $x^{*k}_J(t) = (x^{*k}_J(t),\ 1 \leqslant k \leqslant n)^{\mathrm{T}}_{n\times 1}$ *and the hard thresholding* $v = \frac{13\epsilon}{15J}$, *then we have*

$$\|x_J(t) - x^*_J(t)\|_{H^2(\Omega)} \leqslant \epsilon. \tag{2.21}$$

Proof. From [10], we know

$$\|x_J(t)-x_J^*(t)\|_{H^2(\Omega)} = \left\| \begin{matrix} x_J^1(t)-x_J^{*1}(t) \\ x_J^2(t)-x_J^{*2}(t) \\ \vdots \\ x_J^n(t)-x_J^{*n}(t) \end{matrix} \right\|_{H^2(\Omega)} \leqslant C_1 \left\| \begin{matrix} \sum_{m,|\hat{x}_m^1|<v} \hat{x}_m^1 \\ \sum_{m,|\hat{x}_m^2|<v} \hat{x}_m^1 \\ \vdots \\ \sum_{m,|\hat{x}_m^n|<v} \hat{x}_m^n \end{matrix} \right\|_{l^2} \sim \mathrm{O}(\epsilon),$$

where C_1 is some constant. □

Corollary 1. *For $x(t) \in H^2(\Omega)$, the wavelet approximation are $x_J(t)$ and $x_J^*(t)$. Let $v = \frac{13\epsilon}{15J}$, then the truncation error is*

$$\|x(t)-x_J^*(t)\|_{H^2(\Omega)} \sim \mathrm{O}(\epsilon). \tag{2.22}$$

Proof. By Lemma 1 and the property of the cubic spline wavelet interpolation, we know

$$\|x(t)-x_J^*(t)\|_{H^2(\Omega)} \leqslant \|x(t)-x_J(t)\|_{H^2(\Omega)} + \|x_J(t)-x_J^*(t)\|_{H^2(\Omega)} \sim \mathrm{O}(2^{-4J}) + \mathrm{O}(\epsilon). \qquad \square$$

From the wavelet series, we know, when $J \to \infty$ the subspace V_J may approximate the whole space $H^2(\Omega)$ infinitely. It is also clear that the approximated accuracy depends on the wavelet decomposition order J and the threshold v. The higher the space order is, the less the error would be. Due to the characteristic of the wavelet representation, the magnitudes of the wavelet coefficients in W_J will indicate whether a refinement, by increasing the wavelet space order, is needed or not. It may be simply described by mathematical symbol. That is, for given δ_2, if $\max\{|\hat{x}_{i,l}^k|\} > \delta_2$, we will increase the wavelet space order J to J' where $J' > J$.

With the above adaptive properties of the cubic spline wavelets, we present the following modified Newton adaptive iterative process for the DAEs.

2.4. Modified Newton adaptive iterative process

We give the basic steps of the modified Newton iterative process for the nonlinear DAEs. Let $\hat{z} = (\hat{z}_1, \hat{z}_2, \ldots, \hat{z}_{(n+m)N})^{\mathrm{T}}$ and P_v be an adaptive truncation operator such that, for $1 \leqslant k \leqslant (n+m)N$,

$$P_v\hat{z} = \begin{cases} \hat{z}_k, & \text{if } |\hat{z}_k| > v, \\ 0, & \text{otherwise}, \end{cases} \tag{2.23}$$

where $v = \frac{13\epsilon}{15J}$.

Given initial conditions, the value $\hat{z}^{(0)} = (\hat{x}^{(0)}, \hat{y}^{(0)})^{\mathrm{T}}$ and the truncation parameter v.

Step 1. Initially given: the level J, the matrix $\widetilde{A}$; vectors F and G; vector $\hat{z}^{(0)}$ and $\Delta\hat{z}^{(0)} = \hat{z}^{(0)}$.
Step 2. When $\frac{\|\Delta\hat{z}^{(k)}\|}{\|\hat{z}^{(k)}\|} \leqslant \delta_1$, to Step 3; else
1. compute $Q(\hat{z}^{(k)})$ and $H(\hat{z}^{(k)})$, solve the increment $\Delta\hat{z}^{(k)}$;
2. compute $\hat{z}^{(k+1)}$ by $\hat{z}^{(k+1)} = \hat{z}^{(k)} - \Delta\hat{z}^{(k)}$;
3. $\hat{z}^{(k+1)} = P_v\hat{z}^{(k+1)}$;
4. $\hat{z}^{(k)} = \hat{z}^{(k+1)}$; to Step 3.

Step 3. If $\max|\hat{z}_{J,k}| < \delta_2$, to Step 4; else
1. $J = J + 1$;
2. compute $\widetilde{A}$, F, G, and $\hat{z}^{(k)}$; to Step 1.

Step 4. Set $\hat{z}^{(k)} = (\hat{x}^{(k)}, \hat{y}^{(k)})^{\mathrm{T}}$ and use inverse discrete wavelet transform to compute $x_J(t)$ and $y_J(t)$.

3. Theoretical analysis of the modified Newton adaptive iteration

In this section, we provide the theoretical analysis on the convergence and the stability for the modified Newton adaptive iterative process under some given conditions.

3.1. Convergence analysis

We denote the value $\hat{z} = \begin{pmatrix} \hat{x} \\ \hat{y} \end{pmatrix}$ as the true solution. It should satisfy

$$H(\hat{z}) = H(\hat{x},\hat{y}) = \begin{pmatrix} \widetilde{A}\hat{x} - F(\hat{x},\hat{y}) \\ G(\hat{x},\hat{y}) \end{pmatrix}_{(n+m)N\times 1} = 0. \tag{3.1}$$

Let

$$\begin{aligned} \theta_k &= P_\epsilon(\hat{z}^{(k)} - Q^{-1}(\hat{z}^{(k)}) \times H(\hat{x}^{(k)},\hat{y}^{(k)})) - (\hat{z}^{(k)} - Q^{-1}(\hat{z}^{(k)}) \times H(\hat{x}^{(k)},\hat{y}^{(k)})), \\ e_k &= \hat{z}^{(k)} - \hat{z}, \end{aligned} \tag{3.2}$$

we have

$$\hat{z}^{(k+1)} = P_\epsilon(\hat{z}^{(k)} - Q^{-1}(\hat{z}^{(k)}) \times H(\hat{x}^{(k)},\hat{y}^{(k)})) - \hat{z} = P_\epsilon\left(\hat{z}^{(k)} - Q^{-1}(\hat{z}^{(k)}) \times \begin{pmatrix} \widetilde{A}\hat{x}^{(k)} - F(\hat{x}^{(k)},\hat{y}^{(k)}) \\ G(\hat{x}^{(k)},\hat{y}^{(k)}) \end{pmatrix}\right) - \hat{z}. \tag{3.3}$$

Before doing the theoretical analysis, we need some lemmas.

Lemma 2. *Let* $x^{(1)}(t) = (x_1^{(1)}(t), x_2^{(1)}(t), \ldots, x_n^{(1)}(t))^{\mathrm{T}}$, $x^{(2)}(t) = (x_1^{(2)}(t), x_2^{(2)}(t), \ldots, x_n^{(2)}(t))^{\mathrm{T}}$, $y^{(1)}(t) = (y_1^{(1)}(t), y_2^{(1)}(t), \ldots, y_m^{(1)}(t))^{\mathrm{T}}$, $y^{(2)}(t) = (y_1^{(2)}(t), y_2^{(2)}(t), \ldots, y_m^{(2)}(t))^{\mathrm{T}}$, $f = (f_1, f_2, \ldots, f_n)^{\mathrm{T}}$, *and* $g = (g_1, g_2, \ldots, g_m)^{\mathrm{T}}$. *Suppose that for all* $t \in \Omega = [0, L]$ *the vector functions* $f(x(t), y(t))$ *and* $g(x(t), y(t))$ *satisfy*

$$\|f(x^{(1)}(t), y^{(1)}(t)) - f(x^{(2)}(t), y^{(2)}(t))\|_{H^2(\Omega)} \leqslant L_1\|x^{(1)}(t) - x^{(2)}(t)\|_{H^2(\Omega)} + L_1'\|y^{(1)}(t) - y^{(2)}(t)\|_{H^2(\Omega)} \tag{3.4}$$

and

$$\|g(x^{(1)}(t), y^{(1)}(t)) - g(x^{(2)}(t), y^{(2)}(t))\|_{H^2(\Omega)} \leqslant L_2\|x^{(1)}(t) - x^{(2)}(t)\|_{H^2(\Omega)} + L_2'\|y^{(1)}(t) - y^{(2)}(t)\|_{H^2(\Omega)} \tag{3.5}$$

furthermore, under cubic spline wavelet bases the representation forms of $x^{(1)}(t)$ *and* $y^{(1)}(t)$ *are*

$$x^{(1)}(t) = \begin{pmatrix} \sum_{k=1}^{N} \hat{x}_{1,k}^{(1)}\psi_k(t) \\ \sum_{k=1}^{N} \hat{x}_{2,k}^{(1)}\psi_k(t) \\ \vdots \\ \sum_{k=1}^{N} \hat{x}_{n,k}^{(1)}\psi_k(t) \end{pmatrix}_{n\times 1}, \quad y^{(1)}(t) = \begin{pmatrix} \sum_{k=1}^{N} \hat{y}_{1,k}^{(1)}\psi_k(t) \\ \sum_{k=1}^{N} \hat{y}_{2,k}^{(1)}\psi_k(t) \\ \vdots \\ \sum_{k=1}^{N} \hat{y}_{m,k}^{(1)}\psi_k(t) \end{pmatrix}_{m\times 1}, \tag{3.6}$$

similarly, for $x^{(2)}(t)$ *and* $y^{(2)}(t)$. *Then, there are the constants* a_1, a_2, b_1, *and* b_2, *which are related to* L_1, L_1', L_2, *and* L_2', *such that*

$$\|F(\hat{x}^{(1)}, \hat{y}^{(1)}) - F(\hat{x}^{(2)}, \hat{y}^{(2)})\|_{l^2} \leqslant a_1\|\hat{x}^{(1)} - \hat{x}^{(2)}\|_{l^2} + a_2\|\hat{y}^{(1)} - \hat{y}^{(2)}\|_{l^2} \tag{3.7}$$

and

$$\|G(\hat{x}^{(1)}, \hat{y}^{(1)}) - G(\hat{x}^{(2)}, \hat{y}^{(2)})\|_{l^2} \leqslant b_1\|\hat{x}^{(1)} - \hat{x}^{(2)}\|_{l^2} + b_2\|\hat{y}^{(1)} - \hat{y}^{(2)}\|_{l^2}, \tag{3.8}$$

where $\hat{x}^{(1)} = (\sum_{k=1}^{N}\hat{x}_{1,k}^{(1)}, \sum_{k=1}^{N}\hat{x}_{2,k}^{(1)}, \ldots, \sum_{k=1}^{N}\hat{x}_{n,k}^{(1)},)_{n\times 1}^{T}$, *similarly, for* $\hat{x}^{(2)}$, $\hat{y}^{(1)}$, *and* $\hat{y}^{(2)}$.

Proof. By the definitions of F and G, we know

$$\begin{aligned}
&\|F(\hat{x}^{(1)},\hat{y}^{(1)}) - F(\hat{x}^{(2)},\hat{y}^{(2)})\|_{l^2}\\
&= \left\| \begin{matrix} F_1(\hat{x}^{(1)},\hat{y}^{(1)}) - F_1(\hat{x}^{(2)},\hat{y}^{(2)}) \\ F_2(\hat{x}^{(1)},\hat{y}^{(1)}) - F_2(\hat{x}^{(2)},\hat{y}^{(2)}) \\ \vdots \\ F_n(\hat{x}^{(1)},\hat{y}^{(1)}) - F_n(\hat{x}^{(2)},\hat{y}^{(2)}) \end{matrix} \right\|_{l^2}\\
&= \left\| \begin{matrix} f_1\left(\sum_k \hat{x}_k^{(1)}\psi_k(t_1), \sum_k \hat{y}_k^{(1)}\psi_k(t_1)\right) - f_1\left(\sum_k \hat{x}_k^{(2)}\psi_k(t_1), \sum_k \hat{y}_k^{(2)}\psi_k(t_1)\right) \\ f_1\left(\sum_k \hat{x}_k^{(1)}\psi_k(t_2), \sum_k \hat{y}_k^{(1)}\psi_k(t_2)\right) - f_1\left(\sum_k \hat{x}_k^{(2)}\psi_k(t_2), \sum_k \hat{y}_k^{(2)}\psi_k(t_2)\right) \\ \vdots \\ f_n\left(\sum_k \hat{x}_k^{(1)}\psi_k(t_N), \sum_k \hat{y}_k^{(1)}\psi_k(t_N)\right) - f_n\left(\sum_k \hat{x}_k^{(2)}\psi_k(t_N), \sum_k \hat{y}_k^{(2)}\psi_k(t_N)\right) \end{matrix} \right\|_{l^2}\\
&\leqslant L_1 \left\| \begin{matrix} \sum_k \hat{x}_{1,k}^{(1)} - \sum_k \hat{x}_{1,k}^{(2)} \\ \sum_k \hat{x}_{2,k}^{(1)} - \sum_k \hat{x}_{2,k}^{(2)} \\ \vdots \\ \sum_k \hat{x}_{n,k}^{(1)} - \sum_k \hat{x}_{n,k}^{(2)} \end{matrix} \right\|_{l^2} \left\| \begin{matrix} \psi_k(t_1) \\ \psi_k(t_2) \\ \vdots \\ \psi_k(t_N) \end{matrix} \right\|_{l^2} + L_1' \left\| \begin{matrix} \sum_k \hat{y}_{1,k}^{(1)} - \sum_k \hat{y}_{1,k}^{(2)} \\ \sum_k \hat{y}_{2,k}^{(1)} - \sum_k \hat{y}_{2,k}^{(2)} \\ \vdots \\ \sum_k \hat{y}_{n,k}^{(1)} - \sum_k \hat{y}_{n,k}^{(2)} \end{matrix} \right\|_{l^2} \left\| \begin{matrix} \psi_k(t_1) \\ \psi_k(t_2) \\ \vdots \\ \psi_k(t_N) \end{matrix} \right\|_{l^2}\\
&\leqslant a_1\|\hat{x}^{(1)} - \hat{x}^{(2)}\|_{l^2} + a_2\|\hat{y}^{(1)} - \hat{y}^{(2)}\|_{l^2}.
\end{aligned}$$

Similarly, we have

$$\|G(\hat{x}^{(1)},\hat{y}^{(1)}) - G(\hat{x}^{(2)},\hat{y}^{(2)})\|_{l^2} \leqslant b_1\|\hat{x}^{(1)} - \hat{x}^{(2)}\|_{l^2} + b_2\|\hat{y}^{(1)} - \hat{y}^{(2)}\|_{l^2}.$$

It is also clear that a_1, a_2, b_1 and b_2 are induced by the constants L_1, L_1', L_2 and L_2'. □

Lemma 3. *Let* $W = \begin{pmatrix} \widetilde{A} & 0 \\ 0 & 0 \end{pmatrix}$ *be the system matrix, then, we have* $W \in L(\ell^2, \ell^2)$ *and*

$$\|I - \eta W\|_{L(\ell^2,\ell^2)} \leqslant \mu < 1 \tag{3.9}$$

for $0 \leqslant \eta \leqslant \eta_0$, *where* $\eta_0 > 0$ *is some constant.*

Proof. For the representation matrix W of the differential operator in (1.1), it is not difficult to check $W \in L(\ell^2, \ell^2)$. By [5], we have that the conclusion holds. □

Next, we state the property of the Jacobian matrix Q. Since the system of nonlinear DAEs is index one, the matrix $\frac{\partial G}{\partial \hat{y}}$ is nonsingular. If the matrix $\widetilde{A} - \frac{\partial F}{\partial \hat{x}}$ is nonsingular, then the Jacobian matrix Q is nonsingular, too. Furthermore, the standard local convergence theory on Newton iteration implies the following lemma.

Lemma 4. *Let the constant* η *be given in Lemma 3. If the initial value* $\hat{z}^{(0)}$ *is selected appropriately, then, for the iterative solution* $\hat{z}^{(k)}$, *the Jacobian matrix* Q *has the property* $\|Q^{-1}(\hat{z}^{(k)})\|_{l^2} < \eta$.

Now, we give a convergence proof of the above modified Newton iterative process. To do so, we also need an assumption.

Assumption N. Let a_1, a_2, b_1, and b_2 be Lipschitz constants appearing in Lemma 2, we suppose that they satisfy

$$\left\| I - \eta \begin{pmatrix} \widetilde{A} - a_1 I & -a_2 I \\ b_1 I & b_2 I \end{pmatrix} \right\|_{l^2} \leqslant \mu < 1 \tag{3.10}$$

for some constant μ.

Theorem 1. *Given θ_k and e_k. If the truncation parameter is $v = \frac{13\epsilon}{15J}$, then the iterative error of the modified Newton process is*

$$\|e_{k+1}\|_{l^2} \leqslant \frac{\epsilon}{1-\mu} + \mu^{k+1}\|e_0\|_{l^2}. \tag{3.11}$$

Proof. First, we observe that the operator P_v is contractive. From the definitions of θ_k and e_k, we know

$$\begin{aligned}
e_{k+1} &= \hat{z}^{(k+1)} - \hat{z} = \hat{z}^{(k+1)} - (\hat{z}^{(k)} - Q^{-1}(\hat{z}^{(k)}) \times H(\hat{z}^{(k)})) + (\hat{z}^{(k)} - Q^{-1}(\hat{z}^{(k)}) \times H(\hat{z}^{(k)})) - \hat{z} \\
&= P_\epsilon\big(\hat{z}^{(k)} - Q^{-1}(\hat{z}^{(k)}) \times H(\hat{z}^{(k)})\big) - (\hat{z}^{(k)} - Q^{-1}(\hat{z}^{(k)}) \times H(\hat{z}^{(k)})) \\
&\quad + (\hat{z}^{(k)} - Q^{-1}(\hat{z}^{(k)}) \times H(\hat{z}^{(k)})) - \hat{z} \\
&= \theta_k + \hat{z}^{(k)} - \hat{z} - Q^{-1}(\hat{z}^{(k)}) \times H(\hat{z}^{(k)}) \\
&= \theta_k + e_k - Q^{-1}(\hat{z}^{(k)}) \begin{pmatrix} \widetilde{A}\hat{x}^{(k)} - F(\hat{x}^{(k)}, \hat{y}^{(k)}) \\ G(\hat{x}^{(k)}, \hat{y}^{(k)}) \end{pmatrix} \\
&= \theta_k + e_k - Q^{-1}(\hat{z}^{(k)}) \left[\begin{pmatrix} \widetilde{A}\hat{x}^{(k)} - F(\hat{x}^{(k)}, \hat{y}^{(k)}) \\ G(\hat{x}^{(k)}, \hat{y}^{(k)}) \end{pmatrix} - \begin{pmatrix} \widetilde{A}\hat{x} - F(\hat{x}, \hat{y}) \\ G(\hat{x}, \hat{y}) \end{pmatrix} \right] \\
&= \theta_k + e_k - Q^{-1}(\hat{z}^{(k)}) \begin{pmatrix} (\widetilde{A}\hat{x}^{(k)} - \widetilde{A}\hat{x}) + (F(\hat{x}, \hat{y}) - F(\hat{x}^{(k)}, \hat{y}^{(k)})) \\ G(\hat{x}^{(k)}, \hat{y}^{(k)}) - G(\hat{x}, \hat{y}) \end{pmatrix} \\
&= \theta_k + e_k - Q^{-1}(\hat{z}^{(k)}) \begin{pmatrix} \widetilde{A} - a_1 I & -a_2 I \\ b_1 I & b_2 I \end{pmatrix} \begin{pmatrix} \hat{x}^{(k)} - \hat{x} \\ \hat{y}^{(k)} - \hat{y} \end{pmatrix} \\
&= \theta_k + \left(I - Q^{-1}(\hat{z}^{(k)}) \begin{pmatrix} \widetilde{A} - a_1 I & -a_2 I \\ b_1 I & b_2 I \end{pmatrix} \right) e_k.
\end{aligned}$$

Furthermore, under Assumption N, we get

$$\begin{aligned}
\|e_{k+1}\|_{l^2} &\leqslant \|\theta_k\|_{l^2} + \mu\|e_k\|_{l^2} \\
&\leqslant \|\theta_k\|_{l^2} + \mu(\|\theta_{k-1}\|_{l^2} + \mu\|e_{k-1}\|_{l^2}) \\
&\quad \vdots \\
&\leqslant (\|\theta_k\|_{l^2} + \mu\|\theta_{k-1}\|_{l^2} + \mu^2\|\theta_{k-2}\|_{l^2} + \ldots + \mu^k\|\theta_0\|_{l^2}) + \mu^{k+1}\|e_0\|_{l^2} \\
&= \left(\sum_{i=0}^{k} \mu^{k-i}\|\theta_i\|_{l^2} \right) + \mu^{k+1}\|e_0\|_{l^2} \\
&\leqslant \left(\max_{0 \leqslant i \leqslant k} \|\theta_i\|_{l^2} \right) \sum_{m=0}^{k} \mu^m + \mu^{k+1}\|e_0\|_{l^2}.
\end{aligned}$$

Due to

$$\|\theta_i\|_{l^2} = \|P_v(\hat{z}^{(i)} - Q^{-1}(\hat{z}^{(i)})H(\hat{z}^{(i)})) - (\hat{z}^{(i)} - Q^{-1}(\hat{z}^{(i)})H(\hat{z}^{(i)}))\|_{l^2} \leqslant \epsilon,$$

we have the following formula:

$$\|e_{k+1}\|_{l^2} \leqslant \left(\max_{0 \leqslant i \leqslant k} \|\theta_i\|_{l^2} \right) \sum_{m=0}^{k} \mu^m + \mu^{k+1}\|e_0\|_{l^2} \leqslant \frac{\epsilon}{1-\mu} + \mu^{k+1}\|e_0\|_{l^2}.$$

The above iterative error will go to zero as $\epsilon \to 0$ and $k \to +\infty$. The convergence rate depends on the order of ϵ. □

Note that by prescribing the value of ϵ, we can actively control the accuracy of the solution.

3.2. Stability analysis

Let $\widetilde{\hat{z}}^{(k+1)}$ be the small disturbance of $\hat{z}^{(k+1)}$, it has

$$w_{k+1} = \tilde{\hat{z}}^{(k+1)} - \hat{z}^{(k+1)}, \quad \tilde{e}_k = \tilde{\hat{z}}^{(k+1)} - \hat{z}. \tag{3.12}$$

Now, we show that the small disturbance cannot affect the whole stability under some conditions. For convenience, we denote that

$$Q_1^{-1} = Q^{-1}(\hat{z}^{(k)}), \quad Q_2^{-1} = Q^{-1}(\tilde{\hat{z}}^{(k)}) \tag{3.13}$$

and

$$B = \begin{pmatrix} \widetilde{A}\hat{x}^{(k)} - F(\hat{x}^{(k)}, \hat{y}^{(k)}) \\ G(\hat{x}^{(k)}, \hat{y}^{(k)}) \end{pmatrix}, \quad D = \begin{pmatrix} \widetilde{A}\tilde{\hat{x}}^{(k)} - F(\tilde{\hat{x}}^{(k)}, \tilde{\hat{y}}^{(k)}) \\ G(\tilde{\hat{x}}^{(k)}, \tilde{\hat{y}}^{(k)}) \end{pmatrix}. \tag{3.14}$$

Theorem 2. *Suppose*

$$\|(Q_1^{-1} - Q_2^{-1})\|_{l^2} < \tau \tag{3.15}$$

for small enough τ, and given w_{k+1} and $\tilde{e}_k$, we have

$$\|w_{k+1}\|_{l^2} \leqslant \mu^{k+1}\|w_0\|_{l^2} + \frac{C}{1-\mu}\tau, \tag{3.16}$$

where C is constant and $\mu < 1$ in Assumption N.

Proof. From the definitions on w_{k+1} and $\tilde{e}_k$, we have

$$\begin{aligned}
w_{k+1} &= \hat{z}^{(k+1)} - \tilde{\hat{z}}^{(k+1)} = P_\epsilon\big(\tilde{\hat{z}}^{(k)} - Q^{-1}(\tilde{\hat{z}}^{(k)})\big)\begin{pmatrix} \widetilde{A}\tilde{\hat{x}}^{(k)} - F(\tilde{\hat{x}}^{(k)}, \tilde{\hat{y}}^{(k)}) \\ G(\tilde{\hat{x}}^{(k)}, \tilde{\hat{y}}^{(k)}) \end{pmatrix} \\
&\quad - P_\epsilon\big(\hat{z}^{(k)} - Q^{-1}(\hat{z}^{(k)})\big)\begin{pmatrix} \widetilde{A}\hat{x}^{(k)} - F(\hat{x}^{(k)}, \hat{y}^{(k)}) \\ G(\hat{x}^{(k)}, \hat{y}^{(k)}) \end{pmatrix} \\
&= \tilde{\hat{z}}^{(k)} - Q^{-1}(\tilde{\hat{z}}^{(k)})\begin{pmatrix} \widetilde{A}\tilde{\hat{x}}^{(k)} - F(\tilde{\hat{x}}^{(k)}, \tilde{\hat{y}}^{(k)}) \\ G(\tilde{\hat{x}}^{(k)}, \tilde{\hat{y}}^{(k)}) \end{pmatrix} - \hat{z}^{(k)} + Q^{-1}(\hat{z}^{(k)})\begin{pmatrix} \widetilde{A}\hat{x}^{(k)} - F(\hat{x}^{(k)}, \hat{y}^{(k)}) \\ G(\hat{x}^{(k)}, \hat{y}^{(k)}) \end{pmatrix} \\
&= [\tilde{\hat{x}}^{(k)} - \hat{x}^{(k)}] + \left[Q^{-1}(\hat{z}^{(k)})\begin{pmatrix} \widetilde{A}\hat{x}^{(k)} - F(\hat{x}^{(k)}, \hat{y}^{(k)}) \\ G(\hat{x}^{(k)}, \hat{y}^{(k)}) \end{pmatrix} - Q^{-1}(\tilde{\hat{z}}^{(k)})\begin{pmatrix} \widetilde{A}\tilde{\hat{x}}^{(k)} - F(\tilde{\hat{x}}^{(k)}, \tilde{\hat{y}}^{(k)}) \\ G(\tilde{\hat{x}}^{(k)}, \tilde{\hat{y}}^{(k)}) \end{pmatrix}\right] \\
&= w_k + Q_1^{-1}B - Q_2^{-1}D.
\end{aligned}$$

Now, we focus on the analysis of the formula $Q_1^{-1}B - Q_2^{-1}D$. Because

$$Q_1^{-1}B - Q_2^{-1}D = Q_1^{-1}B - Q_1^{-1}D + Q_1^{-1}D - Q_2^{-1}D = Q_1^{-1}(B-D) + (Q_1^{-1} - Q_2^{-1})D,$$

where

$$\begin{aligned}
Q_1^{-1}(B-D) &= Q_1^{-1}\begin{pmatrix} \widetilde{A}\hat{x}^{(k)} - F(\hat{x}^{(k)}, \hat{y}^{(k)}) - \widetilde{A}\tilde{\hat{x}}^{(k)} + F(\tilde{\hat{x}}^{(k)}, \tilde{\hat{y}}^{(k)}) \\ G(\hat{x}^{(k)}, \hat{y}^{(k)}) - G(\tilde{\hat{x}}^{(k)}, \tilde{\hat{y}}^{(k)}) \end{pmatrix} \\
&\leqslant Q_1^{-1}\left[\begin{pmatrix} -\widetilde{A} + a_1 I & a_2 I \\ -b_1 I & -b_2 I \end{pmatrix}\right]\begin{pmatrix} \tilde{\hat{x}}^{(k)} - \hat{x}^{(k)} \\ \tilde{\hat{y}}^{(k)} - \hat{y}^{(k)} \end{pmatrix} \\
&= -Q_1^{-1}\left[\begin{pmatrix} \widetilde{A} - a_1 I & -a_2 I \\ b_1 I & b_2 I \end{pmatrix}\right] w_k
\end{aligned}$$

and

$$\begin{aligned}(Q_1^{-1}-Q_2^{-1})D &= (Q_1^{-1}-Q_2^{-1})\begin{pmatrix} \widetilde{A}\tilde{\hat{x}}^{(k)} - F(\tilde{\hat{x}}^{(k)}, \tilde{\hat{y}}^{(k)}) \\ G(\tilde{\hat{x}}^{(k)}, \tilde{\hat{y}}^{(k)}) \end{pmatrix} \\ &\leqslant (Q_1^{-1}-Q_2^{-1})\begin{pmatrix} \widetilde{A}-a_1I & -a_2I \\ b_1I & b_2I \end{pmatrix}\begin{pmatrix} \tilde{\hat{x}}^{(k)} - \hat{x} \\ \tilde{\hat{y}}^{(k)} - \hat{y} \end{pmatrix} \\ &= (Q_1^{-1}-Q_2^{-1})\begin{pmatrix} \widetilde{A}-a_1I & -a_2I \\ b_1I & b_2I \end{pmatrix}\tilde{e}_k,\end{aligned}$$

we get

$$\begin{aligned}\|w_{k+1}\|_{l^2} &\leqslant \|w_k + Q_1^{-1}B - Q_2^{-1}D\|_{l^2} \\ &= \left\| \left(I - Q_1^{-1}\left[\begin{pmatrix} \widetilde{A}-a_1I & -a_2I \\ b_1I & b_2I \end{pmatrix}\right]\right) w_k + (Q_1^{-1}-Q_2^{-1})\begin{pmatrix} \widetilde{A}-a_1I & -a_2I \\ b_1I & b_2I \end{pmatrix}\tilde{e}_k \right\|_{l^2}.\end{aligned}$$

From Assumption N, the matrix $\begin{pmatrix} \widetilde{A}-a_1I & -a_2I \\ b_1I & b_2I \end{pmatrix}$ is bounded. It follows that

$$\left\| \begin{pmatrix} \widetilde{A}-a_1I & -a_2I \\ b_1I & b_2I \end{pmatrix}\tilde{e}_k \right\|_{l^2} \leqslant C,$$

where C is some positive constant. Thus, we arrive at

$$\|w_{k+1}\|_{l^2} \leqslant \mu\|\omega_k\|_{l^2} + C\tau \leqslant \cdots \leqslant \mu^{k+1}\|\omega_0\|_{l^2} + \frac{C}{1-\mu}\tau. \qquad \square$$

3.3. Complexity analysis

Here, we give the complexity analysis of the step 2 of the adaptive Newton algorithm.

Given the level J, when the iterative time is k, the vectors $\hat{x}^{(k)}$ and $\hat{y}^{(k)}$ have the non-zero element numbers $N_k(\hat{x})$ and $N_k(\hat{y})$, respectively. Let $N_k = \max(N_k(\hat{x}), N_k(\hat{y}))$. The main computational burden of the step 2 depends on the computation of $Q(\hat{z}^{(k)})$ and $H(\hat{z}^{(k)})$, that is, $F(\hat{x}^{(k)}, \hat{y}^{(k)})$, $G(\hat{x}^{(k)}, \hat{y}^{(k)})$, $\frac{\partial F}{\partial x}(\hat{x}^{(k)}, \hat{y}^{(k)})$, $\frac{\partial F}{\partial y}(\hat{x}^{(k)}, \hat{y}^{(k)})$, $\frac{\partial G}{\partial x}(\hat{x}^{(k)}, \hat{y}^{(k)})$ and $\frac{\partial G}{\partial y}(\hat{x}^{(k)}, \hat{y}^{(k)})$. For the level J, the matrix A_J can be computed only once. The complexity of F, G, $\frac{\partial F}{\partial x}$, $\frac{\partial F}{\partial y}$, $\frac{\partial G}{\partial x}$ and $\frac{\partial G}{\partial y}$ are $O(N_k\log(N_k))$, respectively. The operation of the operator P_ϵ equals the sort program operation with $O(N_k\log(N_k))$. Thus, the complexity of the step 2 needs $O(N_k\log(N_k))$ operations at most. By [4], the wavelet transform of the step 4 also needs $O(N_k\log(N_k))$ operations.

From the above complexity analysis, we know the storage and computation of the Newton iterative method depend on the nonzero numbers of the wavelet coefficients vector $\hat{x}$ and $\hat{y}$. In order to demonstrate the tremendous savings of the adaptive algorithm it is illustrative to compare the number of nonzero element numbers used in the adaptive and nonadaptive methods with adequate resolution.

4. Numerical experiments

In this section, we do some elementary experiments on a simple system of nonlinear DAEs of index one to confirm the theoretical analysis discussed above. For this purpose, we apply the proposed adaptive wavelet method to compute its transient responses

$$\begin{cases} \frac{\mathrm{d}x(t)}{\mathrm{d}t} = \frac{18}{5}\tanh(y(t)-x(t)) + \frac{3}{5}\sin(4\pi t), \\ y(t) = \frac{1}{5}\tanh(y(t)) + \frac{3}{5}\tanh(y(t)-x(t)) + 3\sin(\pi t/2), \\ [x(0), y(0)]^t = [0, 0]^t, \quad t \in [0, 5], \end{cases}$$

where $\tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$.

Table 1
Numerical result for the example

Level J	Size $N = 2^J * L + 3$	Error R	Iteration n
1	13	$6.12e{-}2$	18
2	23	$1.24e{-}2$	15
3	43	$2.25e{-}3$	15
4	83	$7.32e{-}4$	10

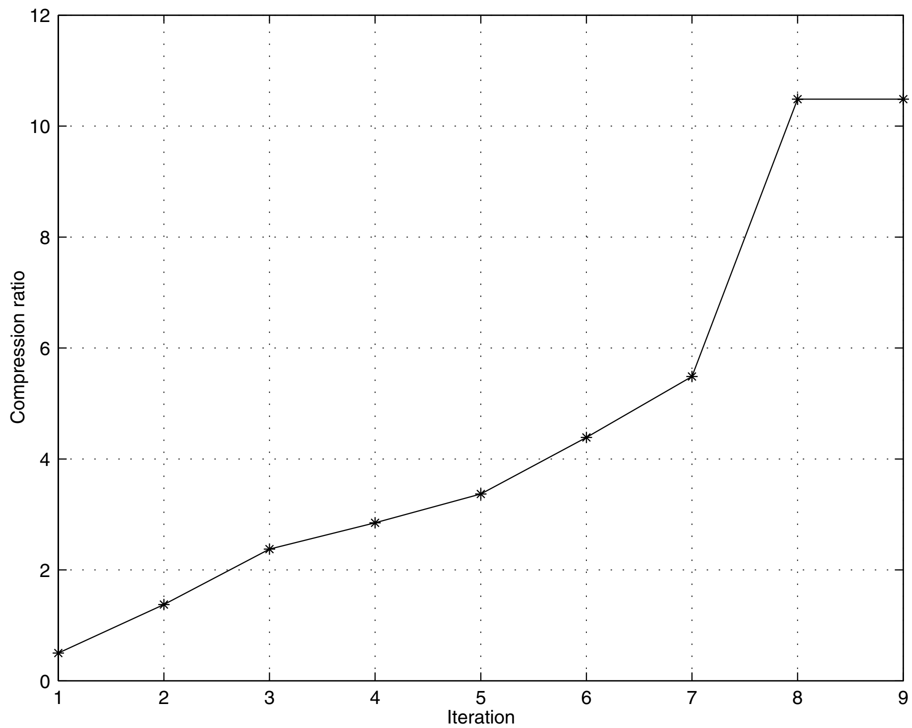

Fig. 1. The compression ratio $\aleph$ as a function of the iteration k for $\epsilon = 10^{-4}$ and $J = 5$.

The computed errors of the modified Newton adaptive iterative method is presented in Table 1. It is obvious that the adaptive algorithm is convergent with the iteration step.

Let the compression ratio $\aleph = \frac{N^J}{N_k^J}$, where N_k^J is the actual non-zero numbers of the vectors $\hat{x}$ and $\hat{y}$ used in the calculations at the kth iterative step and N^J is the number of the vectors $\hat{x}$ and $\hat{y}$, required for the non-adaptive algorithm problem. The compression ratio measures the ratio of the efficiency of this adaptive method. Under the convergence circumstance, the increase compression ratio means the less actual complexity operation of this algorithm. Here, we give the graph of the function compression ratio $\aleph$ as the iteration k (Fig. 1).

5. Conclusion

An adaptive wavelet numerical method is developed for nonlinear differential-algebraic equations (DAEs) of index one. The method combines cubic spline wavelet collocation with the modified Newton adaptive iterative process to find the responses of large dynamic systems. The approach developed here is more appropriate in theory for highly oscillatory analysis of DAEs in circuit simulation. How to select "good" boundary functions for some concrete circuit systems to further improve the efficiency of the method will be studied in the future.

References

[1] A. Barinka, A. Barsch, T. Charton, et al., Adaptive wavelet schemes for elliptic problems implementation and numerical experiments, SIAM J. Sci. Comput. 23 (2001) 910–939.
[2] S. Bertoluzza, M. Verani, Convergence of a nonlinear wavelet algorithm for the solution of PDEs, Appl. Math. Lett. 16 (2003) 113–118.
[3] K.E. Brenan, S.L. Campbell, L.R. Petzold, Numerical Solution of Initial-value Problems in Differential-Algebraic Equations, SIAM, Philadelphia, PA, 1996.
[4] W. Cai, J.Z. Wang, Adaptive multiresolution collocation methods for initial boundary value problems of nonlinear PDEs, SIAM J. Numer. Anal. 33 (1996) 937–970.
[5] A. Cohen, W. Dahmen, R. Devore, Adaptive wavelet for elliptic operator equations: convergence rates, Math. Comput. 70 (2000) 27–75.
[6] E. Hairer, C. Lubich, M. Roche, The Numerical Solution of Differential-Algebraic Systems by Runge–Kutta Methods, Springer Verlag, Berlin, 1989.
[7] Y.L. Jiang, A general approach to waveform relaxation solutions of nonlinear differential-algebraic equations: the continuous-time and discrete-time cases, IEEE Trans. Circ. Syst. I 51 (2004) 1770–1780.
[8] Y.L. Jiang, O. Wing, Monotone waveform relaxation for systems of nonlinear differential-algebraic equations, SIAM J. Numer. Anal. 38 (2000) 170–185.
[9] M. Verani, A wavelet adaptive Newton method for the solution of nonlinear equations, Appl. Math. Lett. 16 (2003) 1301–1306.
[10] J. Wang, Cubic spline wavelet bases of Sobolev spaces and multilevel interpolation, Appl. Comput. Harmon. Anal. 23 (1996) 154–163.
[11] D. Zhou, W. Cai, A fast wavelet collocation method for high-speed circuit simulation, IEEE Trans. Circ. Syst. I 46 (1999) 920–930.
[12] D. Zhou, W. Zhang, An adaptive wavelet method for nonlinear circuit simulation, IEEE Trans. Circ. Syst. I 46 (1999) 931–938.