01 Dec 1990

The Waveform Relaxation Method for Systems of Differential/ Algebraic Equations

Marija D. Ilić

Mariesa Crow

Missouri University of Science and Technology, crow@mst.edu

Proceedings of the 29th Conference
on Decision and Control
Honolulu, Hawaii • December 1990

The Waveform Relaxation Method for Systems of Differential/Algebraic Equations

by

Mariesa L. Crow[1]

Department of Electrical and Computer Engineering
Arizona State University
Tempe, AZ 85287

Marija D. Ilić

Laboratory for Electromagnetic and Electronic Systems
Massachusetts Institute of Technology
Cambridge, MA 02139

1 Introduction

A differential/algebraic system may be modeled in the form

$$0 = F(\dot{y}, y, t) \tag{1}$$

where $\frac{\partial F}{\partial \dot{y}}$ is singular and $\frac{\partial F}{\partial y}$ may or may not be singular. Systems of differential/algebraic equations (DAEs) of this type arise in connection with power systems [1], singular perturbation theory [2], control theory [3], circuit simulations [4], robot dynamics [5], and many other applications in the fields of mechanical and chemical engineering, economics, and physics. Only recently has concerted effort been put forth to find methods to numerically solve these systems [3] [4] [6] – [15]. Previously, systems of DAEs were frequently restated as ODEs, often with considerable difficulty or by destroying the structure of the problem (i.e., the resulting variables often no longer represent physical quantities, or the inherent sparsity of the system is destroyed) [12], but as DAE systems arise more and more frequently, it has become necessary to develop numerical methods for solving these systems distinct from the traditional methods for ODEs. If $\frac{\partial F}{\partial \dot{y}}$ is non-invertible, the system is said to be of *higher* index [13]. Difficulties in using ODE methods for solving DAE systems occur when the systems have index greater than or equal to two.

Standard circuit simulators use direct methods to discretize the system by standard stable implicit integration methods. The direct method may become inefficient for very large, dynamic systems. This is because the matrix solution time of the linear algebraic equations grows super linearly with the size of the problem, thus swamping all other steps of the integration method. The direct methods are also inefficient for systems with states which are varying with considerably different rates. Direct application of a discretizing integration method forces all of the states to be discretized identically and with sufficient fineness such that the fastest changing state can be accurately reproduced. If it were possible to divide the system into several subsystems, each of which were changing at individual rates, then it would be possible to integrate each subsystem with the largest possible time step which would accurately reflect the behavior of the subsystem.

In addition, if it were possible to divide a higher index system into subsystems, many of the subsystems could have a lower index, making it possible to solve these subsystems with the usual numerical methods, with little or no additional precautionary measures, while those of higher index may be integrated using methods specifically tailored for high index systems.

One method which overcomes all of the above drawbacks is the *waveform relaxation* (WR) algorithm. The WR algorithm was introduced as an iterative method for the numerical integration of the system of ordinary differential equations over a finite time interval [16]. It is based on the Gauss–Seidel and Gauss–Jacobi relaxation methods [17] used for solving large systems of algebraic equations. In the WR approach, the system is broken into subsystems which are solved independently, with each subsystem using the previous iterate waveforms as "guesses" about the behavior of the state variables in other subsystems. Waveforms are then exchanged between subsystems, and the subsystems are then resolved with improved information about the other subsystems. This process is repeated until convergence is achieved. The WR algorithm was first applied to index one DAEs in [18] for the simulation of VLSI circuits. This paper endeavors to generalize the WR algorithm to encompass a broader spectrum of DAE systems, namely those of higher–index.

Because the WR algorithm is inherently parallel in nature, it is well suited for implementation on various parallel processors.

2 The Solvability and Index of a DAE System

Although general DAE systems look similar to the standard ordinary differential equation systems of the form

$$\dot{y} = f(y, t) \quad y(0) = y_0 \tag{2}$$

they are in many ways quite different. Some DAE systems may be solved successfully using numerical techniques which are commonly used for solving stiff systems of ODEs, but many present difficulties peculiar to DAE systems. The classical theory of explicit ordinary differential equations assures that state space models of dynamic systems have unique solutions for continuous inputs and arbitrary initial values [19]. When DAEs are used to model dynamic systems, the question of the existence and uniqueness of solutions is more complex. Depending on the particular choice of inputs and initial values, the DAE system may have no solution, a unique solution, or an infinite number of solutions. A *solvable* DAE system is one which has a unique solution for sufficiently differentiable inputs and initial values which are consistent with the inputs [13].

An important subclass of nonlinear DAEs have the form

$$\dot{x}(t) = f(x, y, t) \quad x(0) = x_0 \tag{3}$$

$$0 = g(x, y, t) \tag{4}$$

where $x \in R^n, y \in R^m, f : R^{n+m+1} \rightarrow R^n$, and $g : R^{n+m+1} \rightarrow R^m$. For a nonlinear system of this type, a local and global

[1] The work of this author is an extension of part of her doctoral dissertation at the University of Illinois, Urbana, IL 61801.

index can be defined. The local index is the index of the linear constant coefficient system that results from linearizing a nonlinear system at a given fixed time. If $\left[\frac{\partial g}{\partial y}\right]$ of (3) is nonsingular, the index of (3) and (4) is defined to be one. An index two or higher system results when $\left[\frac{\partial g}{\partial y}\right]$ is noninvertible. The global index is the number of times the nonlinear DAE system must be differentiated to obtain a system of ODEs [13]. In summary, the index might be considered to be a measure of the singularity of the system. In this work, only systems where the local and global indices are the same are considered. For a more general discussion, see [8].

3 The WR Algorithm for Systems of DAEs

The WR algorithm is a means of solving a large system of nonlinear DAEs. The basic method proposed here is to partition the system into subsystems in which tightly coupled state and non-state variables are grouped together. In particular, the system is decomposed into r subsystems as

$$\begin{aligned}
\dot{x}_1(t) &= F_1(x_1, x_2, \ldots, x_r, y_1, y_2, \ldots, y_r, t) \quad x_1(0) = x_{10} \quad (5)\\
0 &= G_1(x_1, x_2, \ldots, x_r, y_1, y_2, \ldots, y_r, t) \quad y_1(0) = y_{10} \quad (6)\\
&\vdots\\
\dot{x}_r(t) &= F_r(x_1, x_2, \ldots, x_r, y_1, y_2, \ldots, y_r, t) \quad x_r(0) = x_{r0} \quad (7)\\
0 &= G_r(x_1, x_2, \ldots, x_r, y_1, y_2, \ldots, y_r, t) \quad y_r(0) = y_{r0} \quad (8)
\end{aligned}$$

where $x_i \in R^{n_i}, y_i \in R^{m_i}, \sum_{i=1}^{r} n_i = n, \sum_{i=1}^{r} m_i = m, F_i : R^n \times R^m \times R \rightarrow R^{n_i}$, and $G_i : R^n \times R^m \times R \rightarrow R^{m_i}$. The Gauss–Jacobi WR algorithm for solving (5) through (8) is given in Algorithm 4.1.

Algorithm 4.1 – The Gauss–Jacobi WR Algorithm.

$k \leftarrow 0$
Guess some $x_i^0(t)$ such that $x_i^{k+1}(0) = x_i(0)$. $t \in [0, T]$
Guess some $y_i^0(t)$ such that $y_i^{k+1}(0) = y_i(0)$. $t \in [0, T]$
repeat { $k \leftarrow k + 1$.
for each $(i \in \{i, \ldots, r\})$ solve on $[0, T]$

$$\begin{aligned}
\dot{x}_i^{k+1} &= f_i(x_1^k, ..., x_i^{k+1}, ..., x_r^k, y_1^k, ..., y_i^{k+1}, ..., y_r^k)\\
0 &= g_i(x_1^k, ..., x_i^{k+1}, ..., x_r^k, y_1^k, ..., y_i^{k+1}, ..., y_r^k)
\end{aligned}$$

} **until** $(\|x^{k+1} - x^k\| \leq \epsilon_x$ and $\|y^{k+1} - y^k\| \leq \epsilon_y)$

The WR algorithm is iterative in nature, with the previous iterate waveforms of both differential and algebraic variables acting as inputs to the subsystem currently being solved.

3.1 The Solvability of the WR Algorithm for Systems of DAEs

The notion of solvability for DAE systems can be extended to DAE systems with waveform relaxation applied.

Theorem 3.1 *If the system*

$$E\dot{x}(t) = Ax(t) + BU \quad (9)$$

is solvable, then the waveform relaxation formulation is solvable if and only if all the subsystems are solvable.[2]

There are two important observations to make from this theorem. The first observation is that this theorem does not imply convergence of the WR method. It only states that for a solvable system, given any continuous input vector x^k, then there exists a unique output vector x^{k+1}. It does not guarantee anything about the relationship of x^k to x^{k+1}. The second observation is that the solvability of the WR method formulation depends on the subsystems chosen. If two possible ways of partitioning the total system into subsystems are chosen, one may be solvable while the other is not.

3.2 Convergence of the WR Algorithm for Index Two Systems

Several problems in engineering and physics result in problems which have an index of two. Applying the WR algorithm to index two systems yields the canonical form:

$$\begin{aligned}
\dot{w}^{k+1} &= f(w^{k+1}, w^k, \dot{w}^k, z^k, \dot{z}^k) \quad (10)\\
z^{k+1} &= g(w^{k+1}, w^k, \dot{w}^k, z^k, \dot{z}^k) \quad (11)
\end{aligned}$$

where w is possibly different from x. In the remainder of this section sufficient conditions to guarantee convergence of a WR algorithm are derived. The conditions are stated for the canonical form of the WR algorithm, but the WR is not necessarily implemented in its canonical form, i.e. it is not required to find f and g explicitly.

Theorem 3.2 *Consider a WR algorithm whose iterated equations can be transformed into the following canonical form:*

$$\begin{aligned}
\dot{w}^{k+1} &= f(w^{k+1}, w^k, \dot{w}^k, z^k, \dot{z}^k) \quad w^{k+1}(0) = w_0 \quad (12)\\
z^{k+1} &= g(w^{k+1}, w^k, \dot{w}^k, z^k, \dot{z}^k) \quad (13)
\end{aligned}$$

where $w \in R^n, z \in R^m$, and $w_0 \in R^n$. Assume that

1. *the canonical* differential *variables can be expressed in the form*
$$\dot{w}^{k+1} = \tilde{f}(w^{k+1}, w^k, w^{k-1}, \dot{w}^k, \dot{w}^{k-1})$$
2. *there exist norms in $R^n \times R^n, \lambda_1 \geq 0, \lambda_2 \geq 0, \lambda_3 \geq 0$, and $\gamma_1 \in [0, 1), \gamma_2 \in [0, 1)$ where $\gamma_1 + \gamma_2 < 1$, such that for any $a, b, c, d, e, \tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}, \tilde{e} \in R^n$*
$$\begin{aligned}
\left\|\tilde{f}(a, b, c, d, e) - \tilde{f}(\tilde{a}, \tilde{b}, \tilde{c}, \tilde{d}, \tilde{e})\right\| \leq\ & \lambda_1 \|a - \tilde{a}\| + \lambda_2 \left\|b - \tilde{b}\right\|\\
& + \lambda_3 \|c - \tilde{c}\| +\\
& \gamma_1 \left\|d - \tilde{d}\right\| + \gamma_2 \|e - \tilde{e}\|
\end{aligned}$$

Then for any initial guesses $(w^0(t), z^0(t); t \in [0, T]), (w^1(t), z^1(t); t \in [0, T]), w^0(t) \not\equiv w^1(t), z^0(t) \not\equiv z^1(t)$ the sequence $\left\{(\dot{w}^{k+1}(t), w^{k+1}(t), z^{k+1}(t); t \in [0, T])\right\}_{k=1}^{\infty}$ generated by the WR algorithm converges uniformly to $(\dot{w}(t), \dot{w}(t), \dot{z}(t); t \in [0, T])$ which satisfies

$$\begin{aligned}
\dot{\dot{w}} &= f(\dot{w}, \dot{w}, \dot{\dot{w}}, \dot{z}, \dot{\dot{z}}) \quad \dot{w}(0) = w_0 \quad (14)\\
\dot{z} &= g(\dot{w}, \dot{w}, \dot{\dot{w}}, \dot{z}, \dot{\dot{z}}) \quad (15)
\end{aligned}$$

The above theorem[3] may be generalized to other index systems as well. For higher index systems, the convergence theorem follows that of the index two case, except the canonical differential functions must be m–point contractive with respect to the time derivative iterates where m is the index of the system.

As an example of a linear index two circuit, consider the simple two–node example in Figure 1, which is made up of two capacitors, two resistors, and an independent voltage source. It is possible to construct the system of differential/algebraic equations that describes the circuit by using nodal analysis. This gives rise to the following DAE system:

$$\begin{bmatrix} C_1 & -C_1 & 0 \\ -C_1 & C_1 + C_2 & 0 \\ 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \dot{V}_1 \\ \dot{V}_2 \\ \dot{I} \end{bmatrix} = \begin{bmatrix} -g_1 & 0 & 1 \\ 0 & -g_2 & 0 \\ 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \\ I \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ -1 \end{bmatrix} \nu(t) \quad (16)$$

[2] The proof of this theorem is straightforward, but lengthy and is omitted for brevity. The proof may be found in [22].

[3] The proof of this theorem may be found in [23]

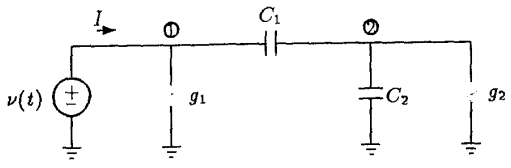

Figure 1: Index Two Floating Capacitor Circuit Example

where I is the current through the independent voltage source. By partitioning the equations such that the variables (V_1, I) and (V_2) are grouped together, the WRDAE algorithm generates the following system of equations:

$$\dot{V}_1^{k+1} = \frac{-g_1}{C_1} V_1^{k+1} + \frac{1}{C_1} I^{k+1} + \dot{V}_2^k \quad (17)$$

$$0 = V_1^{k+1} - \nu(t) \quad (18)$$

$$\dot{V}_2^{k+1} = \frac{-g_2}{C_1 + C_2} V_2^{k+1} + \frac{C_1}{C_1 + C_2} \dot{V}_1^k \quad (19)$$

By defining $\tilde{I} = \frac{1}{C_1} I$, the above equations are equivalent to the following system of equations:

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} \dot{V}_2^{k+1} \\ \dot{V}_1^{k+1} \\ \tilde{I}^{k+1} \end{bmatrix} = \begin{bmatrix} \frac{-g_2}{C_1+C_2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} V_2^{k+1} \\ V_1^{k+1} \\ \tilde{I}^{k+1} \end{bmatrix} + \begin{bmatrix} \frac{C_1}{C_1+C_2} \dot{V}_1^k \\ -\dot{\nu}(t) \\ \frac{g_1}{C_1} \nu(t) + \dot{V}_2^k \end{bmatrix} \quad (20)$$

which can be transformed into the following canonical form:

$$\dot{V}_2^{k+1} = \frac{-g_2}{C_1 + C_2} V_2^{k+1} + \frac{C_1}{C_1 + C_2} \dot{V}_1^k \quad (21)$$

$$V_1^{k+1} = \nu(t) \quad (22)$$

$$\tilde{I}^{k+1} = \frac{-g_1}{C_1 + C_2} \nu(t) - \dot{V}_2^k - \dot{\nu}(t) \quad (23)$$

By substituting the derivative of (22) into (21), the reduced canonical system becomes

$$\dot{V}_2^{k+1} = \frac{-g_2}{C_1 + C_2} V_2^{k+1} + \frac{C_1}{C_1 + C_2} \dot{\nu}(t) \quad (24)$$

This reduced canonical system satisfies the assumptions of Theorem 3.2, and the WRDAE algorithm applied to the system will converge.

At this point, one final note to this section on the index two WRDAE method is pointed out. Whether or not the WRDAE will converge depends very heavily on the manner in which the original problem is partitioned. A partitioning may result in a nonconvergent system, or may not be solvable. As an example of this latter problem, consider again the previous linear index two example of the floating capacitor. If the system variables and equations would be partitioned as (V_1) and (I, V_2), then the resultant equations from the WRDAE algorithm are

$$\dot{V}_1^{k+1} = \frac{-g_1}{C_1} V_1^{k+1} + \frac{1}{C_1} I^k + \dot{V}_2^k \quad (25)$$

$$0 = V_1^k - \nu(t) \quad (26)$$

$$\dot{V}_2^{k+1} = \frac{-g_2}{C_1 + C_2} V_2^{k+1} + \frac{C_1}{C_1 + C_2} \dot{V}_1^k \quad (27)$$

This system is no longer solvable. There is no possible way to update I^{k+1}, and $I^{k+1} = I^0$ for all k. From Theorem 3.1, the problem is solvable, if and only if each of the subsystems yields a solvable system of equations. Clearly, the second subsystem is nonsolvable.

This concludes the presentation of the waveform relaxation algorithm for systems of differential/algebraic equations. The theorems presented give conditions under which the WR algorithm generates a set of solvable equations, and the properties these equations must have to converge to the solution of the original system of equations.

4 The Discretized WR Method for DAEs

Interconnected dynamic systems are often modeled in DAE form. One traditional approach to solving large scale systems has been to replace the full system model with a reduced order state space model. These reduction processes destroy the natural physical structure and sparsity of the full order system. Thus numerical solution algorithms, which make effective use of structure and sparsity for efficiency, perform poorly on the reduced order system even though this system is still quite large. Thus it is desirable to be able to solve the systems in their original DAE form. The most promising approach to solving a DAE system numerically is the direct application of implicit ordinary differential equation methods to the DAE system.

The approach of applying ODE methods to discretize and numerically integrate a DAE system was first introduced by Gear [8], and consists of replacing $\dot{x}(t)$ by a k–step backwards difference formula (BDF) approximation,

$$\dot{x}(t) \approx \frac{\rho_n x_n}{h_n} = \frac{1}{h_n} \sum_{i=0}^{k} \alpha_i x_{n-i} \quad (28)$$

and then solving the resulting equations for approximations to x_n and y_n.

A major advantage of the WR algorithm is that the differential/algebraic equations are solved in a decomposed manner. This implies that if discretization methods are used to solve the independent sets of equations, the time steps for each subsystem can be selected relatively independently. This leads naturally to three questions: Does the waveform relaxation process still converge? If it does converge, does the resulting multirate integration method possess the stability properties of the integration method used for the decomposed systems? Can the various time steps be chosen independently?

4.1 The Uniform Time Step Case

As an example of a situation where the WR convergence depends on discretization, consider WR applied to the following linear index two test system with partitioning $\{x_1, y\}, \{x_2\}$:

$$\dot{x}_1 = -5x_1 + y + 0.1x_2 \quad (29)$$

$$0 = x_1 + \lambda_1 x_2 \quad (30)$$

$$\dot{x}_2 = x_1 - \lambda_2 x_2 \quad (31)$$

This system is solvable and stable for all $\lambda_1, \lambda_2 \geq 0$. Therefore, if each equation in the system is discretized identically (this is known as a *uniform* time step), the discretized equations under backward-Euler become

$$(1 + 5h)x_{1,n+1}^{k+1} - hy_{n+1}^{k+1} = 0.1hx_{2,n+1}^k + x_{1,n}^{k+1} \quad (32)$$

$$x_{1,n+1}^{k+1} = -\lambda_1 x_{2,n+1}^k \quad (33)$$

$$(1 + \lambda_2 h)x_{2,n+1}^{k+1} = hx_{1,n+1}^k + x_{2,n}^{k+1} \quad (34)$$

	k=1		k=2		k=3		k=4	
t	x_1	x_2	x_1	x_2	x_1	x_2	x_1	x_2
0.00	-1.000	0.100	-1.000	0.100	-1.000	0.100	-1.000	0.100
0.01	-1.000	0.089	-0.896	0.089	-0.896	0.091	-0.906	0.091
0.02	-1.000	0.079	-0.792	0.079	-0.792	0.082	-0.823	0.082
0.03	-1.000	0.069	-0.688	0.068	-0.688	0.075	-0.750	0.075
0.04	-1.000	0.058	-0.585	0.058	-0.585	0.069	-0.688	0.069
0.05	-1.000	0.048	-0.483	0.048	-0.483	0.064	-0.637	0.064

Table 1: Successive Iterates for the Linear Index Two Test System

where $h = t_{n+1} - t_n$.

The waveforms for the Gauss–Jacobi WR algorithm exhibit a strange behavior when discretization is applied to DAE systems of index greater than or equal to two. The iterates for all variables do not change at every iteration as they might in the WR applied to ODEs, but for an index two case, they only change every other iteration. The first few iterations of x_1 and x_2 for $h = 0.01$ of the previous example are given in Table 1, with a "flat start" initial condition.[4] From Table 1, note that x_1 is updated in iterations 2 and 4, whereas x_2 is updated in iterations 1 and 3.

An examination of the discretized equations for the example clarifies why this phenomenon occurs. After discretization by backward–Euler, the discrete variables may be rearranged to yield

$$x_{1,n+1}^{k+1} = -\lambda_1 x_{2,n+1}^{k} \quad (35)$$

$$y_{n+1}^{k+1} = -\left(0.1 + \lambda_1\left(\frac{1+5h}{h}\right)\right) x_{2,n+1}^{k} - \frac{1}{h} x_{1,n}^{k+1} \quad (36)$$

$$x_{2,n+1}^{k+1} = \frac{h}{1+\lambda_2 h} x_{1,n+1}^{k} + \frac{1}{1+\lambda_2 h} x_{2,n}^{k+1} \quad (37)$$

Thus $x_{1,n+1}^{k+1}$ can be equivalently expressed as

$$x_{1,n+1}^{k+1} = -\lambda_1\left(\frac{h}{1+\lambda_2 h} x_{1,n+1}^{k-1} + \frac{1}{1+\lambda_2 h} x_{2,n}^{k}\right)$$

and similarly

$$x_{2,n+1}^{k+1} = \frac{h}{1+\lambda_2 h}\left(-\lambda_1 x_{2,n+1}^{k-1}\right) + \frac{1}{1+\lambda_2 h} x_{2,n}^{k+1}$$

Note the dependence of the $(k+1)^{st}$ iterate on the $(k-1)^{st}$ iterate. Thus each of these variables is only updated every other iteration, and the "middle" iteration is redundant. This dependency is not unexpected in light of Convergence Theorem 3.2 which states that an index two system may give rise to this type of "nested" iterations.

If the system were discretized and relaxation methods were used to solve for x_{n+1} and y_{n+1}, that is, if the system were solved by "algebraic relaxation," then this system will converge if the eigenvalues of the relaxation matrix lie within the unit circle in the complex plane. For the algebraic relaxation this leads to the following allowable values of h:

$$h < \frac{1}{\lambda_1 - \lambda_2} \qquad \lambda_1 > \lambda_2 \geq 0$$

The WR iterates given above will converge to the discretized solution if

$$\frac{\lambda_1 h}{1+\lambda_2 h} < 1$$

or equivalently

$$h < \frac{1}{\lambda_1 - \lambda_2}$$

This is identical to the time step bounds given by the algebraic relaxation method. The upper bounds on the time steps for which the uniform time step discretized WR algorithm will converge are very similar to the constraints on the time steps for which the algebraic relaxation algorithm applied to a discretized numerical integration scheme will converge. In fact, in the linear case they are closely related as was shown in the previous example. A comparison theorem between the WR and the algebraic relaxation method is presented below.

Theorem 4.1 *Let a k–step backwards difference formula be applied to the linear DAE system of the form*

$$E\dot{y}(t) = Ay(t) \quad y(0) = y_0$$

where $E, A \in R^{m\times m}$, and E has the form

$$E = diag\{E_i\} \quad i = 1, \ldots, r$$

where

$$E_i = \begin{bmatrix} I_{n_i} & 0 \\ 0 & 0 \end{bmatrix} \qquad \sum_{i}^{r} n_i = n\ ; \quad n \leq m \quad (38)$$

and $y(t) \in R^m$. Assume that the Gauss–Jacobi (or Gauss–Seidel) algebraic relaxation algorithm is used to solve the linear algebraic equations generated by the BDF. Given a sequence of time steps $\{h_m\}$ where each h_m is chosen such that the discretized DAE is h_m solvable, the Gauss–Jacobi (Gauss–Seidel) relaxation algorithm will converge at every step, for any initial guess, if and only if the WR algorithm, discretized with the same sequence of time steps and with the same BDF, converges for any initial guess.

This theorem may be generalized to the nonlinear WR algorithm if it is assumed that the initial guess may by chosen arbitrarily close to the exact solution. This requirement is necessary because of the constraints imposed by the Newton's method on the initial guess.

4.2 The Multirate WRDAE Convergence Theorem

When different time steps are chosen by the individual subsystems, the computation of values at a time step of one subsystem may require a value of another subsystem which was not explicitly computed, due to differing time steps. This value must then be interpolated. If this interpolation is not performed carefully, the convergence of the algorithm may be destroyed.

The following DAE system

$$\dot{x}(t) = f(x, y, t) \quad x(0) = x_0$$
$$0 = g(x, y, t)$$

becomes

$$\dot{x}_i^{j+1} = f_i(x_1^j, \ldots, x_i^{j+1}, \ldots, x_r^j, y_1^j, \ldots, y_i^{j+1}, \ldots, y_r^j, t) \quad (39)$$

$$0 = g_i(x_1^j, \ldots, x_i^{j+1}, \ldots, x_r^j, y_1^j, \ldots, y_i^{j+1}, \ldots, y_r^j, t) \quad (40)$$

for $i = 1, \ldots, r$, when the Gauss–Jacobi WRDAE algorithm is applied. If a k–step BDF is used to numerically solve (39)–(40), the iteration equations for the i_{th} subsystem become

$$0 = \hat{x}_i^{j+1}(\tau_{m_i}) - \sum_{\ell=1}^{k-1} a_\ell \hat{x}_i^{j+1}(\tau_{m_i-\ell}) - h_{m_i} b_{-1} f_i(\hat{x}_1^j(\tau_{m_i}), \ldots, \hat{x}_i^{j+1}(\tau_{m_i}), \ldots, \hat{x}_r^j(\tau_{m_i}), \hat{y}_1^j(\tau_{m_i}), \ldots, \hat{y}_i^{j+1}(\tau_{m_i}), \ldots, \hat{y}_r^j(\tau_{m_i})) \quad (41)$$

$$0 = g_i(\hat{x}_1^j(\tau_{m_i}), \ldots, \hat{x}_i^{j+1}(\tau_{m_i}), \ldots, \hat{x}_r^j(\tau_{m_i}), \hat{y}_1^j(\tau_{m_i}), \ldots, \hat{y}_i^{j+1}(\tau_{m_i}), \ldots, \hat{y}_r^j(\tau_{m_i})) \quad (42)$$

If the r subsystems are solved independently, then h_{m_i} may not be the same as h_{m_1}, h_{m_2}, or any of the other $(r-1)$ possible

[4] A "flat start" initial condition is one in which: $x(t) = x_0\ \forall\ t \in [0, T]$.

time steps for the remaining subsystems. Thus, the values of $\hat{x}_1^j(\tau_{m_i}), \hat{x}_2^j(\tau_{m_i}), \ldots$, may not be available and have to be *interpolated* from the existing values.

The most common type of interpolation is linear interpolation, where the unknown value is taken to be on a line connecting the immediately greater and lesser values. For example, if $\hat{x}(\tau_{n-1})$ and $\hat{x}(\tau_n)$ are known, then $\hat{x}(\hat{\tau})$ is approximated by

$$\hat{x}(\hat{\tau}) = \left(\frac{\hat{\tau} - \tau_{n-1}}{\tau_n - \tau_{n-1}}\right)(\hat{x}(\tau_n) - \hat{x}(\tau_{n-1})) + \hat{x}(\tau_{n-1})$$

where $\tau_{n-1} \leq \hat{\tau} \leq \tau_n$. If linear interpolation is used to approximate the unknown values, the multirate discretized WRDAE algorithm will converge. This statement is formalized in the following theorem.

Theorem 4.2 *If linear interpolation is used to approximate the unknown variable values for nonconcurrent time steps, then there exists a collection of time steps $h_{i0} > 0, i = \{1, \ldots, n\}$, such that if $0 < h_i \leq h_{i0}$ for all i, then the multirate fixed–time step discretized WRDAE algorithm converges with respect to the interpolated sequences.*

These theorems imply that the discretized WRDAE algorithms will converge only if the "underlying" discretized equations will converge. When a DAE system of index m is divided into subsystems, not all of the subsystems may have index m; some may have index less than m. This implies that different restrictions may apply in discretizing the various subsystems. A subsystem having an index of three must be discretized by a constant step size BDF of order ≤ 6 [13] whereas a subsystem of index two or less may be discretized by a variable–step variable–order BDF [9]. This is an additional advantage of the WRDAE method. Those subsystems which are of low index may be integrated with traditional numerical methods with little or no additional precautionary measures, while those of higher index may be integrated using methods tailored for high–index systems.

5 Conclusions

This paper presents an extension of the waveform relaxation algorithm to systems of differential/algebraic equations. Although this type of application has been explored earlier in relation to VLSI circuits, the algorithm has not been generalized to include the vast array of DAE system structures. This paper establishes the solvability and convergence requirements of the waveform relaxation algorithm for higher–index systems.

Many systems in robotics and control applications are modeled with DAE systems having an index greater than two. Computer simulation of these systems has been hampered by numerical integration methods which perform poorly and must be explicitly tailored to the system. The WR algorithm presents a means by which these systems may be more efficiently simulated by breaking them into weakly coupled subsystems, many of which will no longer retain the limiting high–index properties.

Appendix – Proof of Theorems

Proof of Theorem 4.1 Applying a k–step BDF to the linear time invariant DAE system

$$E\dot{x}(t) = Ax(t) \tag{43}$$

yields

$$E\sum_{i=0}^{k} \alpha_i \tilde{x}_{n-i} = h_m A \tilde{x}_n \tag{44}$$

which may be written

$$E\left(\alpha_0 \tilde{x}_n + \sum_{i=1}^{k} \alpha_i \tilde{x}_{n-i}\right) = h_m A \tilde{x}_n \tag{45}$$

By the assumption of consistency, $\alpha_0 = 1$, thus

$$E\tilde{x}_n = h_m A \tilde{x}_n - E\sum_{i=1}^{k} \alpha_i \tilde{x}_{n-i} \tag{46}$$

Note that by assumption E is a block diagonal matrix where $E = \text{diag}\,\{E_i\}$ and E_i is as defined in (38). Let $A = M_A - N_A$ where $M_A = \text{diag}\{A_i\}$ such that A_i is of the same dimension as E_i and $N_A = -(A - M_A)$ is an off–block diagonal matrix. Using this notation, the Gauss–Jacobi relaxation iteration applied to solving (46) for $\tilde{x}_n$ is

$$E\tilde{x}_n^{j+1} = h_m M_A \tilde{x}_n^{j+1} - h_m N_A \tilde{x}_n^j - E\sum_{i=1}^{k} \alpha_i \tilde{x}_{n-i} \tag{47}$$

Subtracting the j^{th} iterate from the $(j+1)^{st}$ iterate yields

$$E\delta^{j+1} = h_m M_A \delta^{j+1} - h_m N_A \delta^j \tag{48}$$

where $\delta^{j+1} = \tilde{x}_n^{j+1} - \tilde{x}_n^j$. This is equivalently

$$(E - h_m M_A)\,\delta^{j+1} = -h_m N_A \delta^j \tag{49}$$

or

$$\delta^{j+1} = -h_m (E - h_m M_A)^{-1} N_A \delta^j$$

where $(E - h_m M_A)$ is invertible by the assumption of solvability for h_m. This will converge at the m^{th} time point for any initial guess if and only if all of the eigenvalues of $h_m (E - h_m M_A)^{-1} N_A$ lie within the unit circle in the the complex plane. Using the Gauss–Jacobi WR to solve (43) yields

$$E\dot{x}^{j+1}(t) = M_A x^{j+1}(t) - N_A x^j(t) \tag{50}$$

Applying a BDF to discretize (50) gives

$$E\left(\tilde{x}_n^{j+1} + \sum_{i=1}^{k} \alpha_i \tilde{x}_{n-i}^{j+1}\right) = h_m M_A \tilde{x}_n^{j+1} - h_m N_A \tilde{x}_n^j \tag{51}$$

which by consistency is equivalently

$$(E - h_m M_A)\,\tilde{x}_n^{j+1} = -h_m N_A \tilde{x}_n^j - E\sum_{i=1}^{k} \alpha_i \tilde{x}_{n-i}^{j+1} \tag{52}$$

Subtracting iterates:

$$(E - h_m M_A)\,\delta^{j+1} = -h_m N_A \delta^j - E\sum_{i=1}^{k} \alpha_i \delta_{n-i}^{j+1} \tag{53}$$

To show that the discretized WR algorithm will converge only if the algebraic relaxation converges, let h_l be a time step for which $E - h_l M_A$ is invertible, but for which at least one eigenvalue of

$$h_l (E - h_l M_A)^{-1} N_A \tag{54}$$

does not lie within the unit circle in the complex plane. Assume an initial guess such that the first $l-1$ time points are the exact solution to the discretized problem. Then

$$E\sum_{i=1}^{k} \alpha_i \delta_{n-i}^{j+1} = 0 \tag{55}$$

and (53) is equivalent to (49) and is not convergent. This proves that if the discretized WR iterates converge for any

initial guess then the algebraic relaxation iterates must also converge for any initial guess as well.

To prove that if the algebraic relaxation algorithm converges, then the WR algorithm must converge for any initial guess, assume that the theorem holds for all $m < l$. Then $\delta_{l-1}^{j} \rightarrow 0$ as $j \rightarrow \infty$, and (53) approaches (49). Since it is assumed that the WR iterates converge, this implies that the eigenvalues of (54) are less than one, and the algebraic relaxation iterates converge at the l^{th} time step. Note that (49) is identical to (53) for $m = 1$ from which the proof follows by induction. □

Proof of Theorem 4.2 The proof of this theorem parallels a similar proof in [16] for ODEs. For an index two DAE system, the multirate discretized WRDAE differential canonical system is a two–point contraction in a β–norm. This result is used to prove convergence, and may be generalized to other index systems provided the contraction assumptions are generalized as well. □

References

[1] N. A. Tsolas, A. Arapostathis, P. P. Varaiya, " A structure preserving energy function for power system transient stability analysis," *IEEE Transactions on Circuits and Systems*, vol. CAS-32, no. 10, pp. 1041-1049, October 1985.

[2] P. V. Kokotovic, R. E. O'Malley, Jr., P. Sannuti, " Singular perturbations and order reduction in control theory – An overview," *Automatica*, vol. 12, pp. 123-132, March 1976.

[3] S. L. Campbell, " The numerical solution of singular systems arising in control problems," *Proceedings of the 1989 American Control Conference*, Pittsburgh, Pennsylvania, June 1989.

[4] R. F. Sincovec, B. Dembart, M. A. Epton, A. M. Erisman, J. W. Manke, E. L. Yip, " Solvability of large–scale descriptor systems," *Report*, Boeing Computer Services Company, Seattle, Washington, 1979.

[5] M. W. Spong, M. Vidyasagar, *Robot Dynamics and Control*, John Wiley and Sons, New York, New York, 1989.

[6] K. E. Brenan, S. L. Campbell, L. R. Petzold, *Numerical Solution of Initial–Value Problems in Differential–Algebraic Equations*, Elsevier Science Publishing Co., Inc., New York, New York, 1989.

[7] S. L. Campbell, " The numerical solution of higher index linear time varying singular systems of differential equations," *SIAM Journal of Scientific and Statistical Computing*, vol. 6, no. 2, pp. 334-348, April 1985.

[8] C. W. Gear, L. R. Petzold, " ODE methods for the solution of differential/algebraic systems," *SIAM Journal of Numerical Analysis*, vol. 21, no. 4, pp. 716-728, August 1984.

[9] C. W. Gear, G. K. Gupta, B. Leimkuhler, " Automatic integration of Euler–Lagrange equations with constraints," *Journal of Computational and Applied Mathematics*, no. 12-13, pp. 77-90, 1985.

[10] G. K. Gupta, C. W. Gear, and B. Leimkuhler, " Implementing linear multistep formulas for solving DAEs," *Report*, no. UIUCDCS-R-85-1205, University of Illinois, Urbana, Illinois, April 1985.

[11] B. J. Leimkuhler, *Error estimates for differential–algebraic equations*, Master's Thesis, University of Illinois, Urbana, Illinois, May 1986.

[12] B. J. Leimkuhler, " Approximation methods for the consistent initialization of differential–algebraic equations," *Report*, no. UIUCDCS-R-88-1450, University of Illinois, Urbana, Illinois, August 1988.

[13] P. Lötstedt, L. Petzold, " Numerical solution of nonlinear differential equations with algebraic constraints I: Convergence results for backward differentiation formulas," *Mathematics of Computation*, vol. 46, no. 174, pp. 491-516, April 1986.

[14] I. M. Mack, *Block implicit one–step methods for solving smooth and discontinuous systems of differential/algebraic equation*, Ph.D. dissertation, Harvard University, Cambridge, Massachusetts, 1986.

[15] L. Petzold and P. Lötstedt, " Numerical solution of nonlinear differential equations with algebraic constraints II: Practical implications," *SIAM Journal of Scientific Statistical Computation*, vol. 7, no. 3, pp. 720-733, July 1986.

[16] J. K. White, A. Sangiovanni–Vincentelli, *Relaxation Techniques for the Simulation of VLSI Circuits*, Kluwer Academic Publishers, Boston, MA, 1987.

[17] J. M. Ortega, W. C. Rheinboldt, *Iterative Solution of Nonlinear Equations in Several Variables*, Academic Press, Inc., New York, New York, 1970.

[18] E. Lelarasmee, A. E. Ruehli and A. L. Sangiovanni–Vincentelli, " The waveform relaxation method for time–domain analysis of large scale integrated circuits," *IEEE Transactions on Computer–Aided Design of Integrated Circuits and Systems*, vol. CAD-1, no. 3, pp. 131-144, July, 1982.

[19] J. K. Hale, *Ordinary Differential Equations*, Wiley(Interscience), New York, New York, 1969.

[20] D. G. Feingold, R. S. Varga, " Block diagonally dominant matrices and generalizations of the Gerschgorin circle theorem," *Pacific Journal of Mathematics*, vol. 12, pp. 1241-1250, 1962.

[21] M. Ilić–Spong, I. N. Katz, H. Dai, J. Zaborsky, " Block diagonal dominance for systems of nonlinear equations with application to load flow calculations in power systems," *Mathematical Modeling*, vol. 5, pp. 275-297, 1984.

[22] M. L. Crow, *Waveform relaxation methods for the simulation of systems of differential/algebraic equations with applications to electric power systems*, Ph.D. dissertation, University of Illinois, Urbana, IL, January 1990.

[23] M. L. Crow and M. Ili'c, "The waveform relaxation algorithm for systems of differential/algebraic equations with power system applications," *Proceedings of the 1989 American Control Conference*, Pittsburgh, Pennsylvania, June 1989.