

A waveform relaxation Newmark method for structural dynamics problems

Marco Pasetto¹ · Haim Waisman² · J. S. Chen¹

Received: 1 May 2018 / Accepted: 9 October 2018
© Springer-Verlag GmbH Germany, part of Springer Nature 2018

Abstract

In the conventional Newmark family for time integration of hyperbolic problems, both explicit and implicit methods are inherently sequential in the time domain and not well suited for parallel implementations due to unavoidable processor communication at every time step. In this work we propose a Waveform Relaxation Newmark (WRN_β) algorithm for the solution of linear second-order hyperbolic systems of ODEs in time, which retains the unconditional stability of the implicit Newmark scheme with the advantage of the lower computational cost of explicit time integration schemes. This method is unstructured in the time domain and is well suited for parallel implementation. We consider a Jacobi and Gauss–Seidel type splitting and study their convergence and stability. The performance of the WRN_β algorithm is compared to a standard implicit Newmark method and the obtained results confirm the effectiveness of the Waveform Relaxation Newmark algorithm as a new class of more efficient time integrators, which is applicable, as shown in the numerical examples, to both the finite element method and meshfree methods (e.g. the reproducing kernel particle method).

Keywords Structural dynamics · Waveform relaxation · Newmark method · Implicit integration · Explicit integration

1 Introduction

Structural dynamics problems are governed by a second-order hyperbolic system of ordinary differential equations. One of the most well known and widely used family of direct integration methods is the Newmark family of methods [1]. Its implicit implementation is unconditionally stable but requires the solution of a linear system, which makes it computationally expensive; its explicit form, on the other hand, has a low computational cost as it doesn't require the solution of systems but is conditionally stable, thus limiting the allowed time increment.

The Waveform Relaxation method (WR), also called dynamic iteration method, is an iterative technique for solving space–time problems and can be used for solving the related systems of ODEs. It offers so-called parallelization

of time and due to its implicit nature provides improved stability compared to explicit methods [2].

As highlighted in [3], time parallel methods can be grouped in different categories: methods based on multiple shooting [4–6], direct time parallel methods [7–10] (mainly based on predictor-corrector type of approaches), space–time multigrid techniques [11,12] and waveform relaxation methods [14–16]. The multiple shooting method was developed in its discrete form in [5] and then expanded in continuous form in [6]. The idea behind this approach is to partition the time domain into windows and, after a first coarse serial solution used to determine approximate initial conditions for each time window, solve in parallel a refined solution over each window. The solution at the final time step for a window might not coincide with the assumed initial condition of the next one (thus the term *shooting*): the solution is therefore performed iteratively until agreement is achieved. Different from the multiple shooting methods, multigrid methods in space and time are not naturally parallel [3], but they can be used to solve systems simultaneously over the entire space–time domain [3,17]. Following a similar idea where space–time is one large domain, a common finite element discretization of both the space and the time domains was proposed in [18].

✉ Haim Waisman
waisman@civil.columbia.edu

¹ Department of Structural Engineering, University of California San Diego, La Jolla, CA 92093, USA

² Department of Civil Engineering and Engineering Mechanics, Columbia University, New York, NY 10027, USA

In this work we focus on the class of time-parallel Waveform Relaxation methods [13,19]. The WR method was mostly applied to parabolic initial value problems with applications in electrical network systems and the analysis of circuits [13] and has rarely been applied to hyperbolic problems that arise in structural dynamics.

This method differs from most standard iterative techniques in that it is a continuous-time method, iterating with functions in time, called waveforms. In other words, the space domain in this algorithm is partitioned into smaller subsystems while time is not discretized but rather kept continuous. However, in practical implementations of the method, discretization of time is also desired, especially when non-linearities are considered. One of the advantages of the WR schemes is its being unstructured in the time domain. Hence, if in a system there are several variables changing at different rates, they can be uncoupled and each integrated with the appropriate time step size, with a pre-defined interpolating operator to exchange the information between subsystems [2]. Several ways for accelerating the convergence, thus improving the efficiency, of the waveform relaxation schemes have been investigated, such as successive over relaxation (SOR), Polynomial acceleration (Chebyshev iteration, Krylov subspace acceleration) and Multigrid [20–24].

The SOR waveform relaxation and its convergence were first studied by Miekkala and Nevanlinna [25,26]. Reichelt et al. [27] improved the results by proposing a convolution SOR acceleration of the waveform relaxation scheme in which, instead of multiplying by a fixed SOR parameter as in the traditional SOR approach, a time-dependent SOR kernel was used. The effectiveness of this approach was confirmed in [28], where the performance of the convolution SOR was compared to SOR methods based on matrix splitting and extrapolation. It was found that, while the latter techniques led only to a minor acceleration, the method based on convolution achieved convergence acceleration similar to the one obtained for optimal stationary SOR methods. Lastly, a parareal Schwarz waveform relaxation algorithm was recently proposed by combining the aforementioned multiple shooting methods and the Schwarz waveform relaxation method [3,15,29,30]. In this scheme, similar to the space–time multigrid methods, simultaneous iteration over a set of unknowns in the space–time domain is performed [3]. The focus of these studies and procedures has been mainly on parabolic first order differential equations; as of today, to the authors' knowledge, limited studies have been conducted on hyperbolic and second-order systems. In [31] and [32] the application of waveform relaxation methods to second-order differential equations and their convergence have been studied; however this was performed by converting the system to first order. None of these studies considered the Newmark

family of integration methods for second order ODEs that is common in structural dynamics.

To this end, we consider an extension of the Waveform Relaxation method to the Newmark family of methods for the solution of linear second-order systems such as those common in structural dynamics and solid mechanics. We show that this approach allows us to retain the unconditional stability of the implicit Newmark scheme with the advantage of the lower computational cost typical of explicit time integration schemes. This method is also unstructured in the time domain, allows for multi-rate integration of subcomponents and is well suited for parallel implementation.

The paper is organized as follows. In Sect. 2 we review the general WR Gauss–Seidel and Jacobi relaxation schemes, along with a one-dimensional illustrative example; the Newmark Waveform Relaxation algorithm is also introduced. In Sect. 3 stability analysis and convergence studies are performed, followed by discussion on the necessary conditions needed for stability and convergence of the method. The performance of the method in terms of computational time and iteration counts is assessed in Sect. 4. The performance of the method is illustrated on the semi-discrete equations of structural dynamics discretized by the finite element method as well as the reproducing kernel particle method [33,34].

2 The waveform relaxation scheme

The Waveform Relaxation scheme is an iterative method used to solve systems of time dependent differential equations. To briefly review the key ideas of the WR method we start by considering a dynamical system that can be described as a set of p second-order ordinary differential equations, with associated initial conditions, of the form:

$$\begin{aligned}\ddot{\mathbf{d}} &= \mathbf{f}(t, \mathbf{d}) \quad t \in [0, T], \\ \mathbf{d}(0) &= \mathbf{d}_0, \\ \dot{\mathbf{d}}(0) &= \mathbf{v}_0,\end{aligned}\tag{1}$$

where $T > 0$ is the final simulation time, $\mathbf{f} : [0, T] \times \mathbb{R}^p \rightarrow \mathbb{R}^p$ is a generalized load vector, $\mathbf{d}(t) = [d_1(t) d_2(t) \dots d_p(t)]^t \in \mathbb{R}^p$ is the displacement vector, and \mathbb{R}^p denotes a real valued vector space of dimension p . $\mathbf{d}_0 = [d_{1,0} d_{2,0} \dots d_{p,0}]^t \in \mathbb{R}^p$ and $\mathbf{v}_0 = [v_{1,0} v_{2,0} \dots v_{p,0}]^t \in \mathbb{R}^p$ are vectors which contain the initial displacements $d_{i,0}$ and initial velocity values $v_{i,0}$, respectively.

The key idea of the WR algorithm is to (i) uncouple the set of equations, (ii) integrate efficiently the uncoupled set of equations in time, (iii) repeat the process with the additional information generated from the previous integration step until convergence is reached. Equation decoupling in step (i) is achieved through an assignment-partitioning process,

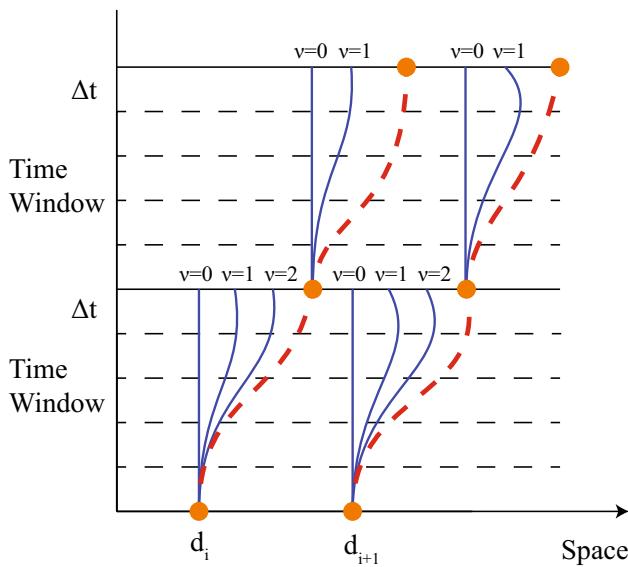


Fig. 1 Space decomposition and iterations over the time windows. The continuous lines represent the solutions obtained for each iteration v while the dashed lines are the converged solutions; the dot points represent the values at the end of each time window, taken as initial conditions for the subsequent one

while steps (ii)–(iii) consist of an **iterative relaxation procedure**. In the assignment-partitioning process, each unknown variable is assigned to one equation of the original system (1), which is then partitioned into disjoint subsystems consisting of one (point-wise decomposition) or more (block-wise decomposition) equations [13]. Therefore, within each subsystem there are the assigned internal variables to be solved for and external variables, which are internal variables of the other subsystems. Each subsystem is then solved iteratively over the time domain during the relaxation process. In order to accelerate convergence and reduce storage, the time domain can be divided into intervals called windows. The solution of each subsystem can be performed independently, even on different processors, and the information is exchanged once the solution over the considered time window has taken place [2].

Figure 1 illustrates the underlying WR concept for a 1D space–time domain. The space domain is first discretized into independent function variables (indicated by the dots) and assigned an arbitrary value over the time window at iteration $v = 0$; each subsystem is then solved iteratively by using the information coming from the other subsystems (external variables) until convergence (dashed line) for its assigned internal variable is achieved. The values of the external variables get updated using the information from the current or previous iterations, depending on the chosen decoupling WR scheme [13]. As shown in Fig. 1, once convergence is reached in one time window the obtained result is then used as a starting point for the successive window.

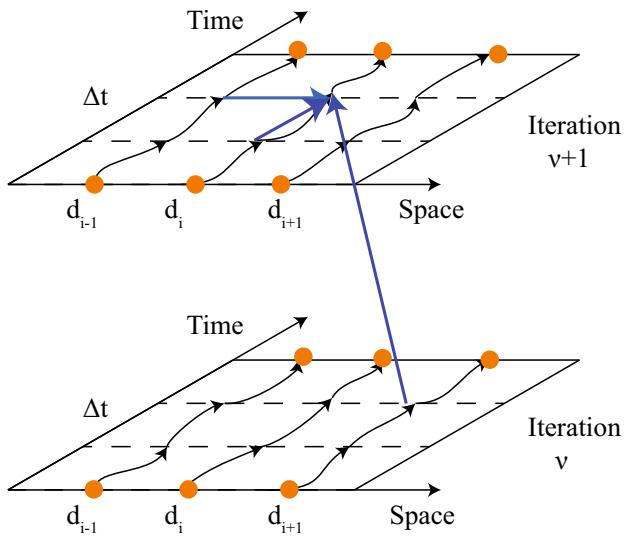


Fig. 2 Information flow for the Gauss–Seidel WR scheme

In this paper we will limit ourselves to the two most commonly used types of relaxation schemes: the Gauss–Seidel (GS) WR and the Jacobi WR. When the Gauss–Seidel relaxation is used, the uncoupled equations are solved sequentially and the waveform solution of one decomposed subsystem is used immediately as input to update the approximate waveforms of the other subsystems. The point-wise GS iteration scheme can therefore be expressed as

$$\begin{cases} \ddot{d}_i^{(v+1)} = f_i(t, d_1^{(v+1)}, \dots, d_{i-1}^{(v+1)}, d_i^{(v+1)}, d_{i+1}^{(v)}, \dots, d_p^{(v)}) \\ d_i^{(v+1)}(0) = d_0 \\ \dot{d}_i^{(v+1)}(0) = v_0 \end{cases}, \quad (2)$$

where the superscripts v and $v + 1$ denote the iteration count and p as before is the number of unknown variables. Figure 2 shows the flow of information if a Gauss–Seidel WR scheme is used.

Instead, if the Jacobi WR is used, all the uncoupled equations could be solved simultaneously using the information on the external variables available from the previous iteration v , updated at the beginning of each iteration [13]. The point-wise Jacobi WR scheme can be mathematically expressed as

$$\begin{cases} \ddot{d}_i^{(v+1)} = f_i(t, d_1^{(v)}, \dots, d_{i-1}^{(v)}, d_i^{(v+1)}, d_{i+1}^{(v)}, \dots, d_p^{(v)}) \\ d_i^{(v+1)}(0) = d_0 \\ \dot{d}_i^{(v+1)}(0) = v_0 \end{cases}. \quad (3)$$

The flow of information for the Jacobi WR is illustrated in Fig. 3.

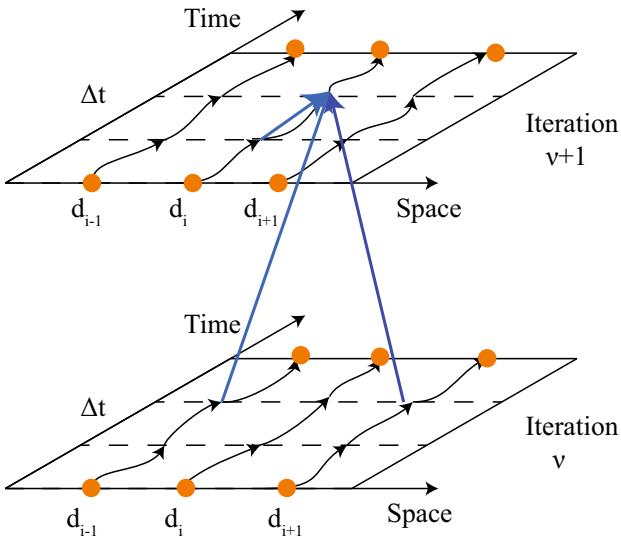


Fig. 3 Information flow for the Jacobi WR scheme

In addition, block-wise schemes can also be defined in a similar way by considering multiple equations for each subsystem. For example, a two block Jacobi scheme can be expressed as [23]

$$\begin{cases} \ddot{d}_{2i-1}^{(v+1)} = f_i(t, d_1^{(v)}, \dots, d_{2i-2}^{(v)}, d_{2i-1}^{(v+1)}, d_{2i}^{(v+1)}, \dots, d_p^{(v)}) \\ \ddot{d}_{2i}^{(v+1)} = f_i(t, d_1^{(v)}, \dots, d_{2i-2}^{(v)}, d_{2i-1}^{(v+1)}, d_{2i}^{(v+1)}, \dots, d_p^{(v)}) \\ d_{2i-1}^{(v+1)}(0) = d_{0,2i-1} \\ \dot{d}_{2i-1}^{(v+1)}(0) = v_{0,2i-1} \\ d_{2i}^{(v+1)}(0) = d_{0,2i} \\ \dot{d}_{2i}^{(v+1)}(0) = v_{0,2i} \end{cases} . \quad (4)$$

Both Jacobi and Gauss–Seidel relaxation schemes are carried out until satisfactory convergence is achieved and are started with an initial approximation $d^{(0)}(t)$ defined over the considered time interval. For known initial conditions, this initial approximation is usually taken to be constant over time and equal to the values specified by the initial conditions [23] so that

$$d_i^{(0)}(t) = d_{0,i}, \quad \forall t \in [0, T], \quad i = 1, \dots, p. \quad (5)$$

The Jacobi type approach allows more parallelization at the expense of slower convergence rates with respect to the Gauss–Seidel WR scheme [2].

As mentioned in Sect. 1, the WR method was first developed as a continuous-time scheme. In practical applications though, the time domain is usually discretized and the differential equations composing the WR subsystems are solved by using conventional numerical integration methods [13,23]; in this study the solution is performed by means of a standard

Newmark- β integration scheme, resulting in the proposed WR-Newmark method (WRN $_\beta$).

In the system defined by Eq. (1), the equation coupling was present through the d_i variables while the \ddot{d}_i terms were already uncoupled. In the remaining of this section we present the approach of the WR method for linear systems describing damped structural dynamics problems. General linear structural dynamics problems are governed by a linear second order hyperbolic system of ODEs, which is written in the well-known standard semi-discrete form as

$$\begin{cases} \mathbf{M}\ddot{\mathbf{d}} + \mathbf{C}\dot{\mathbf{d}} + \mathbf{K}\mathbf{d} = \mathbf{f} \\ \mathbf{d}(0) = \mathbf{d}_0 \\ \dot{\mathbf{d}}(0) = \mathbf{v}_0 \end{cases}, \quad (6)$$

where \mathbf{d} is the displacement vector, \mathbf{f} is the vector of applied forces, \mathbf{M} , \mathbf{C} and \mathbf{K} are respectively the mass, damping and stiffness matrices, and a dot denotes a time-derivative.

The key idea is to split the mass matrix \mathbf{M} into matrices \mathbf{M}_+ and \mathbf{M}_- , the damping matrix into \mathbf{C}_+ and \mathbf{C}_- and the stiffness matrix \mathbf{K} into \mathbf{K}_+ and \mathbf{K}_- so that $\mathbf{M} = \mathbf{M}_+ - \mathbf{M}_-$, $\mathbf{C} = \mathbf{C}_+ - \mathbf{C}_-$ and $\mathbf{K} = \mathbf{K}_+ - \mathbf{K}_-$ and to then consider the following iteration form:

$$\begin{cases} \mathbf{M}_+\ddot{\mathbf{d}}^{(v+1)} + \mathbf{C}_+\dot{\mathbf{d}}^{(v+1)} + \mathbf{K}_+\mathbf{d}^{(v+1)} \\ = \mathbf{M}_-\ddot{\mathbf{d}}^{(v)} + \mathbf{C}_-\dot{\mathbf{d}}^{(v)} + \mathbf{K}_-\mathbf{d}^{(v)} + \mathbf{f} \\ \mathbf{d}^{(v+1)}(0) = \mathbf{d}_0 \\ \dot{\mathbf{d}}^{(v+1)}(0) = \mathbf{v}_0 \end{cases}. \quad (7)$$

The chosen splitting for the matrices influences the convergence and the computational complexity of the solution [23]; based on the splittings proposed in the literature for systems of first-order linear differential equations, we assume \mathbf{M} , \mathbf{C} and \mathbf{K} to be decomposed as $\mathbf{L} + \mathbf{D} + \mathbf{U}$, where \mathbf{D} is a diagonal matrix (block diagonal if a block-wise partitioning is used) and \mathbf{L} and \mathbf{U} are respectively strictly lower and strictly upper triangular matrices [23]. The splitting corresponding to the WR Jacobi and WR Gauss–Seidel schemes are as follows:

Jacobi:	$\mathbf{M}_+ = \mathbf{D}_\mathbf{M}$	$\mathbf{M}_- = -(\mathbf{L}_\mathbf{M} + \mathbf{U}_\mathbf{M})$
	$\mathbf{C}_+ = \mathbf{D}_\mathbf{C}$	$\mathbf{C}_- = -(\mathbf{L}_\mathbf{C} + \mathbf{U}_\mathbf{C})$
	$\mathbf{K}_+ = \mathbf{D}_\mathbf{K}$	$\mathbf{K}_- = -(\mathbf{L}_\mathbf{K} + \mathbf{U}_\mathbf{K})$

Gauss – Seidel :	$\mathbf{M}_+ = \mathbf{L}_\mathbf{M} + \mathbf{D}_\mathbf{M}$	$\mathbf{M}_- = -\mathbf{U}_\mathbf{M}$
	$\mathbf{C}_+ = \mathbf{L}_\mathbf{C} + \mathbf{D}_\mathbf{C}$	$\mathbf{C}_- = -\mathbf{U}_\mathbf{C}$
	$\mathbf{K}_+ = \mathbf{L}_\mathbf{K} + \mathbf{D}_\mathbf{K}$	$\mathbf{K}_- = -\mathbf{U}_\mathbf{K}$

It has to be noted that if lumped mass matrices are considered no splitting of \mathbf{M} is required. Once the desired splitting has been performed, the system in Eq. (7) is integrated over the chosen time interval $t \in [t_0, t_n]$ using the Newmark algorithm. The WR iteration is then terminated when the error between two successive iterations is smaller than a specified tolerance

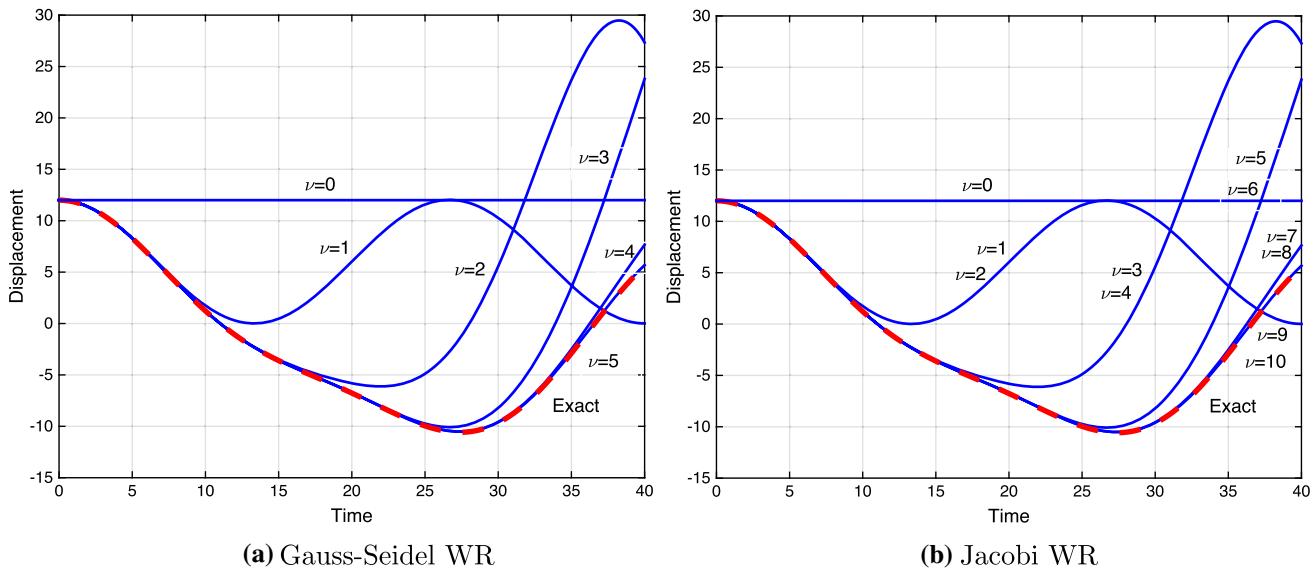


Fig. 4 Space–time convergence of Gauss–Seidel and Jacobi waveform relaxation method; the dashed line represents the analytical solutions while the continuous lines represent the WR solution at each iteration ν

$$\max \left\{ \left\| \mathbf{d}^{\nu+1}(t) - \mathbf{d}^\nu(t) \right\| \right\} \leq \epsilon, \quad (8)$$

for some small positive constant ϵ . Convergence of the WR method to the free-vibration displacement of a two mass system is shown in Fig. 4, for illustration purposes.

The procedure of the WR-Newmark method is summarized in tables corresponding to Algorithms 1 and 2. As for the rest of this paper, bold capital letters indicate matrices while bold lower case letters indicate vectors. Bold capital letters followed by (t) (as in $\mathbf{X}_d(t)$) indicate a column in a matrix corresponding to discrete time t . Waveforms are stored in \mathbf{X}_d , \mathbf{X}_v and \mathbf{X}_a , which denote respectively the space–time displacement, velocity and acceleration matrices. Δt is the time integration time step while β and γ are the parameters of the Newmark method. \mathbf{d} , \mathbf{v} and \mathbf{a} indicate the displacement, velocity and acceleration vectors, respectively. The *Splitting* function corresponds to matrix splitting as described earlier in this section. It can be noted that, if the Jacobi WR scheme is used, the solution step in line 10 of Algorithm 2 does not require any matrix solve and is thus performed at the cost of an explicit scheme even when the overall scheme is unconditionally stable (as traditional Newmark implicit schemes).

Lastly, Algorithms 1 and 2 describe the procedure for a chosen time step Δt . However, in the WRN_β method each subsystem or groups of subsystems in the WRN_β method can be solved using a different time step (hence Algorithm 2 would be called separately for each subsystem). As mentioned in the introduction and in Sect. 4.3, though, this requires the introduction of an interpolatory scheme.

Remark 1 Note that the WR-Newmark method is well suited for parallel implementation. The key issues with parallelization correspond to two main aspects: (i) frequency of processors communication and (ii) solver parallelization algorithms. Below we compare these two aspects for standard Newmark and WR-Newmark algorithms.

- A standard Newmark method is said to be an inherently sequential algorithm, since communication between processors must be done after every time step and the algorithm cannot proceed in time without exchange of information between processors. On the other hand, a WR-Newmark algorithm is said to be time-parallel, since each subsystem coming from the WR assignment-partitioning process can be solved independently on different processors as it is decoupled from all the other subsystems. Hence, communication between processors does not need to be performed after each time step. This can speed up the solution and be especially advantageous when different time steps are used for different subsystems. However, the optimal time-window size needs to be determined by considering the cost of each WR iteration versus the cost for processor communication. This corresponds to the time loop, lines 5–16 in Algorithm 2.
- In case of a standard Newmark method a parallel solver is required to solve the set of equations at every time step. While the systems that arise in linear structural dynamics are sparse and symmetric, scalable parallel solvers are needed to solve these systems efficiently. On the other hand, the WR-Newmark method with Jacobi type splitting does not require any solvers and the solution of the

system is obtained automatically. This corresponds to the solver phase, line 10 in Algorithm 2.

Algorithm 1 Waveform Relaxation Newmark scheme

```

1:  $[\mathbf{K}, \mathbf{C}, \mathbf{M}, \mathbf{f}, \mathbf{d}_0, \mathbf{v}_0, \mathbf{a}_0, t_0, t_n, \Delta t, \beta, \gamma] = setup()$ 
2:  $\mathbf{FX} = [t_0, t_n, \beta, \gamma]$ 
3:  $\mathbf{X}_d \leftarrow [\mathbf{d}_0, \dots, \mathbf{d}_n]$   $\quad \quad \quad \{ \% \} \text{ initialize on space-time}$ 
4:  $\mathbf{X}_v \leftarrow [\mathbf{v}_0, \dots, \mathbf{v}_n]$ 
5:  $\mathbf{X}_a \leftarrow [\mathbf{a}_0, \dots, \mathbf{a}_n]$ 
6:  $[\mathbf{M}_+, \mathbf{M}_-, \mathbf{C}_+, \mathbf{C}_-, \mathbf{K}_+, \mathbf{K}_-] = Splitting(\mathbf{M}, \mathbf{C}, \mathbf{K})$ 
7:  $\mathbf{S}_+ = [\mathbf{M}_+, \mathbf{C}_+, \mathbf{K}_+], \mathbf{S}_- = [\mathbf{M}_-, \mathbf{C}_-, \mathbf{K}_-]$ 
8: while  $NormE \geq \epsilon$  do
9:    $\mathbf{X}_d^0 = \mathbf{X}_d$ 
10:   $\mathbf{X}_v^0 = \mathbf{X}_v$ 
11:   $\mathbf{X}_a^0 = \mathbf{X}_a$ 
12:   $[\mathbf{X}_d, \mathbf{X}_v, \mathbf{X}_a] = NK_\beta(\mathbf{S}_+, \mathbf{S}_-, \mathbf{f}, \mathbf{X}_d, \mathbf{X}_v, \mathbf{X}_a, \mathbf{FX}, \Delta t)$ 
13:   $NormE \leftarrow \max_t \{\|\mathbf{X}_d - \mathbf{X}_d^0\| \}$   $\quad \quad \quad \{ \% \} \text{ compute residual}$ 
14: end while

```

Algorithm 2 Newmark method (a-form)

```

1: function  $[\mathbf{X}_d, \mathbf{X}_v, \mathbf{X}_a] = NK_\beta(\mathbf{S}_+, \mathbf{S}_-, \mathbf{f}, \mathbf{X}_d, \mathbf{X}_v, \mathbf{X}_a, \mathbf{FX}, \Delta t)$ 
2:  $\mathbf{d} \leftarrow \mathbf{d}_0$   $\quad \quad \quad \{ \% \} \text{ initial conditions}$ 
3:  $\mathbf{v} \leftarrow \mathbf{v}_0$ 
4:  $\mathbf{a} \leftarrow \mathbf{a}_0$ 
5: for  $t = t_1 : \Delta t : t_n$  do
6:    $\hat{\mathbf{d}} = \mathbf{d} + \Delta t \mathbf{v} + \frac{\Delta t^2}{2} (1 - 2\beta) \mathbf{a}$   $\quad \quad \quad \{ \% \} \text{ predictor phase}$ 
7:    $\hat{\mathbf{v}} = \mathbf{v} + \Delta t (1 - \gamma) \mathbf{a}$ 
8:    $\mathbf{f}_{WR} = \mathbf{M}_- \mathbf{X}_a(t) + \mathbf{C}_- \mathbf{X}_v(t) + \mathbf{K}_- \mathbf{X}_d(t)$   $\quad \quad \quad \{ \% \} \text{ solution phase}$ 
9:    $\tilde{\mathbf{f}} = \mathbf{f} + \mathbf{f}_{WR} - \mathbf{K}_+ \hat{\mathbf{d}} - \mathbf{C}_+ \hat{\mathbf{v}}$ 
10:   $(\mathbf{M}_+ + \gamma \Delta t \mathbf{C}_+ + \beta \Delta t^2 \mathbf{K}_+) \mathbf{a} = \tilde{\mathbf{f}}$   $\quad \quad \quad \{ \% \} \text{ solve for } \mathbf{a}$ 
11:   $\mathbf{d} = \hat{\mathbf{d}} + \beta \Delta t^2 \mathbf{a}$   $\quad \quad \quad \{ \% \} \text{ corrector phase}$ 
12:   $\mathbf{v} = \hat{\mathbf{v}} + \gamma \Delta t \mathbf{a}$ 
13:   $\mathbf{X}_d(t) = \mathbf{d}$   $\quad \quad \quad \{ \% \} \text{ update waveforms}$ 
14:   $\mathbf{X}_v(t) = \mathbf{v}$ 
15:   $\mathbf{X}_a(t) = \mathbf{a}$ 
16: end for
17: end function

```

在下面的小节中，将介绍一个简单的无阻尼两质量自由振动问题以及使用时间连续 WR 方案获得的结果。

In the following subsection a simple undamped two-mass free-vibration problem is presented along with the results obtained with the time-continuous WR scheme.

2.1 Example problem: free vibration of spring-mass system with two degrees of freedom

To illustrate the key ideas of the Waveform Relaxation method, we begin by considering a system with two discrete masses and springs. The governing system of equations for this problem is

$$\begin{cases} m_1 \ddot{d}_1 + (k_1 + k_2)d_1 - k_2 d_2 = 0 \\ m_2 \ddot{d}_2 - k_2 d_1 + k_2 d_2 = 0 \end{cases}, \quad (9)$$

with $d_1(0) = d_{1,0}$, $d_2(0) = d_{2,0}$, $\dot{d}_1(0) = v_{1,0}$ and $\dot{d}_2(0) = v_{2,0}$. The exact analytical solution to Eq. (9) can be found in “Appendix A”. Applying the Jacobi and the Gauss–Seidel WR schemes we obtain the following iterative systems:

- Jacobi:

$$\begin{cases} m_1 \ddot{d}_1^{(v+1)} + (k_1 + k_2)d_1^{(v+1)} = k_2 d_2^{(v)} \\ m_2 \ddot{d}_2^{(v+1)} + k_2 d_1^{(v+1)} = k_2 d_1^{(v)} \end{cases} \quad (10)$$

- Gauss–Seidel:

$$\begin{cases} m_1 \ddot{d}_1^{(v+1)} + (k_1 + k_2)d_1^{(v+1)} = k_2 d_2^{(v)} \\ m_2 \ddot{d}_2^{(v+1)} + k_2 d_2^{(v+1)} = k_2 d_1^{(v)} \end{cases} \quad (11)$$

with the initial conditions

$$\begin{cases} d_i^{(v+1)}(0) = d_{0,i} \\ \dot{d}_i^{(v+1)}(0) = v_{0,i} \end{cases}.$$

In this example, the initial waveforms corresponding to iteration $v = 0$ are taken to be constant over time and equal to the values specified by the initial conditions. Note that in this illustrative example the waveform relaxation method is solved analytically with continuous time functions reported for the first three iterations in “Appendix B”. Figures 4a, b show the first five iterations for the continuous Gauss–Seidel WR solution and the first ten iterations for the continuous Jacobi WR solution, respectively, along with the exact analytical solution for the system in Eq. (9). It can be observed that convergence is slower for the Jacobi WR scheme with respect to the Gauss–Seidel WR. As noted in [19], for systems with strong coupling each iteration lengthens the time for which the WR solution is close to the exact analytical solution.

3 Stability and convergence analysis

3.1 Stability

The stability of the WR iteration is analyzed following the procedure outlined in [1] since each iteration employs a Newmark scheme. Let us start by recalling the split matrix system for the undamped case defined in Eq. (7):

$$\mathbf{M}_+ \ddot{\mathbf{d}}^{(v+1)} + \mathbf{K}_+ \mathbf{d}^{(v+1)} = \mathbf{F}, \quad (12)$$

where $\mathbf{F} = \mathbf{M}_- \ddot{\mathbf{d}}^{(v)} + \mathbf{K}_- \mathbf{d}^{(v)} + \mathbf{f}$.

Next, the system described in Eq. (7) is reduced to a single degree of freedom SDOF form through modal decomposition. To this end, we solve the eigenvalue problem associated with Eq. (12) for the $v + 1$ iteration. Since the matrices \mathbf{M}_+

and \mathbf{K}_+ do not change for the various iterations, the eigenvalue problem will be the same for every iteration and if one iteration $n + 1$ is stable, so will the others. Hence, stability of the method is analyzed for a generic iteration $n + 1$ and the associated superscript will be omitted. The eigenvalue problem to be considered is therefore:

$$(\mathbf{K}_+ - (\omega_l^{h+})^2 \mathbf{M}_+) \Psi_l = 0, \quad l \in \{1, 2, \dots, n_{eq}\}, \quad (13)$$

with

$$0 \leq (\omega_1^{h+})^2 \leq (\omega_2^{h+})^2 \leq \dots \leq (\omega_{n_{eq}}^{h+})^2, \quad (14)$$

where ω_l^{h+} and n_{eq} represent the l -th system natural frequency and the number of system degrees of freedom, respectively. Note that the eigenvalues and eigenvectors in Eq. (13) are associated with the split WRN $_\beta$ system and not with the original system, which would be considered if the Newmark's method was directly implemented.

In order to analyze the stability of the proposed WR-Newmark method, we follow the procedure in [1] and rewrite the system as a set of first order ODE system as follows:

$$\mathbf{y}_{n+1} = \mathbf{A}\mathbf{y}_n + \mathbf{L}_n, \quad y = \begin{Bmatrix} d \\ \dot{d} \end{Bmatrix}, \quad (15)$$

where \mathbf{A} is the amplification factor, given as

$$\mathbf{A} = \mathbf{A}_1^{-1} \mathbf{A}_2, \quad (16)$$

$$\mathbf{L}_n = \mathbf{A}_1^{-1} \left\{ \frac{\Delta t^2}{2} [(1 - 2\beta)F_n + 2\beta F_{n+1}] \right\}, \quad (17)$$

$$\begin{aligned} \mathbf{A}_1 &= \begin{bmatrix} 1 + \Delta t^2 \beta (\omega^{h+})^2 & 0 \\ (\omega^{h+})^2 \gamma \Delta t & 1 \end{bmatrix}, \\ \mathbf{A}_2 &= \begin{bmatrix} 1 - \frac{\Delta t^2}{2} (\omega^{h+})^2 (1 - 2\beta) & \Delta t \\ -\Delta t (1 - \gamma) (\omega^{h+})^2 & 1 \end{bmatrix}, \end{aligned} \quad (18)$$

where F_n and F_{n+1} are the forcing terms of the modal equations defined at time n and $n + 1$ respectively, Δt is the discretized time step and the parameters β and γ determine accuracy, stability and type of the Newmark algorithm. In order for \mathbf{A} to be spectrally stable the following conditions are required:

- a) $\rho(\mathbf{A}) \leq 1$, where $\rho(\mathbf{A})$ is the spectral radius of \mathbf{A} , defined as

$$\rho(\mathbf{A}) = \max_i |\lambda_i(\mathbf{A})|$$
- b) Eigenvalues of \mathbf{A} of multiplicity greater than one are strictly less than one in modulus

The aforementioned conditions on \mathbf{A} lead to the same stability conditions as for the standard Newmark scheme for β , γ and time step Δt , which are given by [1]:

- Unconditional stability:

$$2\beta \geq \gamma \geq \frac{1}{2}$$

- Conditional stability:

$$\gamma \geq \frac{1}{2}, 2\beta < \frac{\gamma}{2} \quad ; \quad \omega^{h+} \Delta t \leq \Omega_{crit}$$

where the critical sampling frequency Ω_{crit} for an undamped system is $\Omega_{crit} = (\frac{\gamma}{2} - \beta)^{-\frac{1}{2}}$.

Several conclusions can be drawn from this analysis: (i) the stability analysis of WRN $_\beta$ approach is similar to that of traditional Newmark scheme but with the natural frequencies computed using Eq. (13) rather than the original mass and stiffness matrices, (ii) the traditional implicit Newmark method requires the solution of systems of equations while the proposed implicit WRN $_\beta$ does not, i.e. it is an implicit method at the cost of an explicit scheme with several repeated time sweeps (iii) the critical time step in case of traditional conditionally stable Newmark method is more restrictive than the similar (same β and γ parameters) conditionally stable WRN $_\beta$. In other words, a standard Newmark method will require smaller time steps than WR-Newmark to maintain stability. The proof for this last statement follows.

Let us start by considering the eigenvalue problem associated with the use of the traditional Newmark scheme:

$$(\mathbf{K} - \lambda \mathbf{M}) \hat{\Phi} = \mathbf{0}, \quad (19)$$

where λ and $\hat{\Phi}$ are an eigenvalue and an eigenvector of the system.

Recalling the Rayleigh quotient [1], for any arbitrary vector Φ the following inequality holds

$$\frac{\Phi^T \mathbf{K} \Phi}{\Phi^T \mathbf{M} \Phi} \leq \lambda_{max}, \quad (20)$$

where λ_{max} is the largest eigenvalue associated with Eq. (19). The equality holds true only when Φ is the eigenvector of Eq. (19) associated with the largest eigenvalue.

By considering the Jacobi splitting ($\mathbf{M}_+ = \mathbf{D}_M$, $\mathbf{K}_+ = \mathbf{D}_K$):

$$\frac{\Phi^T (\mathbf{K}_+ + \mathbf{K}_-) \Phi}{\Phi^T (\mathbf{M}_+ + \mathbf{M}_-) \Phi} = \frac{\Phi^T \mathbf{K}_+ \Phi + \Phi^T \mathbf{K}_- \Phi}{\Phi^T \mathbf{M}_+ \Phi + \Phi^T \mathbf{M}_- \Phi} \leq \lambda_{max}. \quad (21)$$

Now, let us assume an eigenvector of the WR-Jacobi scheme given in Eq. (13), Ψ that is associated with the largest eigenvalue of the system such that

$$\frac{\Psi^T \mathbf{K}_+ \Psi}{\Psi^T \mathbf{M}_+ \Psi} = \lambda_{max}^+, \quad (22)$$

where λ_{max}^+ is the largest eigenvalue. Since \mathbf{M}_+ and \mathbf{K}_+ are diagonal matrices, i.e.,

$$\mathbf{M}_+ = \begin{bmatrix} m_{11} & 0 & 0 & 0 \\ 0 & m_{22} & 0 & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & m_{n_{eq}n_{eq}} \end{bmatrix},$$

$$\mathbf{K}_+ = \begin{bmatrix} k_{11} & 0 & 0 & 0 \\ 0 & k_{22} & 0 & 0 \\ 0 & 0 & \dots & 0 \\ 0 & 0 & 0 & k_{n_{eq}n_{eq}} \end{bmatrix}, \quad (23)$$

Ψ is such that it is only non-zero in the i -th position corresponding to $\max_i \left(\frac{k_i}{m_i} \right)$. Specifically, $\Psi^T = [0 \dots 1 \dots 0]$, where 1 is in the i -th position, which corresponds to $\max_i \left(\frac{k_i}{m_i} \right) = \lambda_{max}^+$. Since both \mathbf{M}_- and \mathbf{K}_- are matrices with zero diagonals,

$$\Psi^T \mathbf{K}_- \Psi = 0, \quad (24)$$

$$\Psi^T \mathbf{M}_- \Psi = 0. \quad (25)$$

Therefore, by choosing $\Phi = \Psi$ in Eq. (21) we get

$$\frac{\Psi^T \mathbf{K}_+ \Psi + \Psi^T \mathbf{K}_- \Psi}{\Psi^T \mathbf{M}_+ \Psi + \Psi^T \mathbf{M}_- \Psi} = \frac{\Psi^T \mathbf{K}_+ \Psi}{\Psi^T \mathbf{M}_+ \Psi} = \lambda_{max}^+ \leq \lambda_{max}. \quad (26)$$

Furthermore, given that $\hat{\Phi} \neq \Psi$, the inequality in Eq. (26) becomes strict, meaning that

$$\lambda_{max}^+ < \lambda_{max}. \quad (27)$$

Since the largest eigenvalue associated with the WRN $_\beta$ scheme is smaller than the maximum eigenvalue associated with the traditional Newmark problem, the critical time step for stability of the WRN $_\beta$ method is larger than the one for standard Newmark.

Additionally, if instead of a Jacobi partitioning a Gauss–Seidel one is used, both \mathbf{M}_+ and \mathbf{K}_+ are lower triangular matrices. The largest eigenvalue associated with Eq. (13) is still the maximum ratio between the diagonal values of \mathbf{K}_+ and \mathbf{M}_+ . The same discussion and conclusions therefore hold. Lastly, it can be noted that the derivation above provides an inequality between the maximum eigenvalue and the largest diagonal element of any matrix that can be expressed as $(\mathbf{M}_+)^{-1} \mathbf{K}$, with \mathbf{M}_+ being diagonal with positive diagonal entries.

3.2 Convergence

Next, we analyze the convergence of the WR-Newmark method by considering a fixed-point in time with successive iterations v , assuming a stable method for time integration

is chosen, as discussed in the previous section. Consider the semi-discrete dynamic equation at some time $n+1$, neglecting the damping of the system:

$$\mathbf{M} \ddot{\mathbf{d}}_{n+1} + \mathbf{K} \dot{\mathbf{d}}_{n+1} = \mathbf{f}_{n+1}. \quad (28)$$

Application of the WR splitting leads to

$$(\mathbf{M}_+ - \mathbf{M}_-) \ddot{\mathbf{d}}_{n+1} - (\mathbf{K}_+ + \mathbf{K}_-) \dot{\mathbf{d}}_{n+1} = \mathbf{f}_{n+1}, \quad (29)$$

and by rewriting as an iterative scheme, one gets

$$\mathbf{M}_+ \ddot{\mathbf{d}}_{n+1}^{v+1} + \mathbf{K}_+ \dot{\mathbf{d}}_{n+1}^{v+1} = \mathbf{f}_{n+1} + \mathbf{M}_- \ddot{\mathbf{d}}_n^v + \mathbf{K}_- \dot{\mathbf{d}}_{n+1}^v. \quad (30)$$

Considering the Newmark's predictor-corrector scheme, the predictor phase is

$$\hat{\mathbf{d}}_{n+1}^{v+1} = \mathbf{d}_n^{v+1} + \Delta t \dot{\mathbf{d}}_n^{v+1} + \frac{\Delta t^2}{2} (1 - 2\beta) \ddot{\mathbf{d}}_n^{v+1}, \quad (31)$$

$$\hat{\mathbf{v}}_{n+1}^{v+1} = \dot{\mathbf{d}}_n^{v+1} + \Delta t (1 - \gamma) \ddot{\mathbf{d}}_n^{v+1}, \quad (32)$$

$$(\mathbf{M}_+ + \beta \Delta t^2 \mathbf{K}_+) \ddot{\mathbf{d}}_{n+1}^{v+1} = \mathbf{f}_{n+1} + \mathbf{K}_- \dot{\mathbf{d}}_{n+1}^v + \mathbf{M}_- \ddot{\mathbf{d}}_{n+1}^v - \mathbf{K}_+ \hat{\mathbf{d}}_{n+1}^{v+1}, \quad (33)$$

and the corrector phase updated from the previous iteration v is

$$\mathbf{d}_{n+1}^v = \hat{\mathbf{d}}_{n+1}^v + \beta \Delta t^2 \ddot{\mathbf{d}}_{n+1}^v, \quad (34)$$

leading to

$$(\mathbf{M}_+ + \beta \Delta t^2 \mathbf{K}_+) \ddot{\mathbf{d}}_{n+1}^{v+1} = \mathbf{f}_{n+1} + (\mathbf{M}_- + \beta \Delta t^2 \mathbf{K}_-) \ddot{\mathbf{d}}_{n+1}^v + \mathbf{K}_- \hat{\mathbf{d}}_{n+1}^v - \mathbf{K}_+ \hat{\mathbf{d}}_{n+1}^{v+1}. \quad (35)$$

Since we are considering the convergence of the displacement iteration at time $n+1$ between successive iterations v and $v+1$ we rewrite Eq. (33) as

$$\mathbf{P} \ddot{\mathbf{d}}^{v+1} = \mathbf{Q} \ddot{\mathbf{d}}^v + \tilde{\mathbf{f}}, \quad (36)$$

where the subscript $(n+1)$ has been dropped and where $\mathbf{P} = \mathbf{M}_+ + \beta \Delta t^2 \mathbf{K}_+$, $\mathbf{Q} = \mathbf{M}_- + \beta \Delta t^2 \mathbf{K}_-$ and $\tilde{\mathbf{f}} = \mathbf{f} + \mathbf{K}_- \hat{\mathbf{d}}^v - \mathbf{K}_+ \hat{\mathbf{d}}^{v+1}$, which leads to

$$\ddot{\mathbf{d}}^{v+1} = \mathbf{P}^{-1} \mathbf{Q} \ddot{\mathbf{d}}^v + \mathbf{P}^{-1} \tilde{\mathbf{f}}. \quad (37)$$

Equation (37) can further be rewritten in the following form:

$$\ddot{\mathbf{d}}^{v+1} = [\mathbf{I} - \mathbf{P}^{-1} (\mathbf{P} - \mathbf{Q})] \ddot{\mathbf{d}}^v + \mathbf{P}^{-1} \tilde{\mathbf{f}}, \quad (38)$$

which leads to

$$\ddot{\mathbf{d}}^{v+1} = \ddot{\mathbf{d}}^v + \mathbf{P}^{-1}[\tilde{\mathbf{f}} - (\mathbf{P} - \mathbf{Q})\ddot{\mathbf{d}}^v], \quad (39)$$

$$\ddot{\mathbf{d}}^{v+1} = \ddot{\mathbf{d}}^v + \mathbf{P}^{-1}\mathbf{r}^v, \quad (40)$$

where $\mathbf{r}^v = \tilde{\mathbf{f}} - (\mathbf{P} - \mathbf{Q})\ddot{\mathbf{d}}^v$ is the residual at v with respect to the system described by Eq. (36) at the considered time $n + 1$. By denoting with $\ddot{\mathbf{d}}^*$ the exact solution to Eq. (33), we define $\mathbf{e}^{v+1} = \ddot{\mathbf{d}}^{v+1} - \ddot{\mathbf{d}}^*$ as the error at $v + 1$ at time $n + 1$; Rewriting Eq. (40) in terms of the errors, one gets

$$\mathbf{e}^{v+1} = \mathbf{e}^v + \mathbf{P}^{-1}\mathbf{r}^v, \quad (41)$$

$$\mathbf{e}^{v+1} = \mathbf{e}^v + \mathbf{P}^{-1}[-(\mathbf{P} - \mathbf{Q})\mathbf{e}^v], \quad (42)$$

$$\mathbf{e}^{v+1} = [\mathbf{I} - \mathbf{P}^{-1}(\mathbf{P} - \mathbf{Q})]\mathbf{e}^v, \quad (43)$$

$$\mathbf{e}^{v+1} = \mathbf{P}^{-1}\mathbf{Q}\mathbf{e}^v = \mathbf{R}\mathbf{e}^v, \quad (44)$$

where $\mathbf{R} = \mathbf{P}^{-1}\mathbf{Q}$ is the error propagation matrix, which relates the errors in two successive iteration steps and therefore relates the error at every iteration $v + 1$ to the initial one: $\mathbf{e}^{v+1} = \mathbf{R}^{v+1}\mathbf{e}^0$. Iteration (36) converges to the true solution if $\lim_{v \rightarrow \infty} \mathbf{e}^{v+1} = 0$. The error converges to zero and therefore (36) converges to the true solution if and only if all the eigenvalues of \mathbf{R} have magnitudes less than unity [13]:

$$\rho(\mathbf{R}) < 1, \quad (45)$$

$$\rho(\mathbf{P}^{-1}\mathbf{Q}) < 1, \quad (46)$$

$$\rho[(\mathbf{M}_+ + \beta\Delta t^2\mathbf{K}_+)^{-1}(\mathbf{M}_- + \beta\Delta t^2\mathbf{K}_-)] < 1. \quad (47)$$

Furthermore, if matrix $\mathbf{P} - \mathbf{Q}$ is diagonally dominant, then the convergence condition will be satisfied [13]. As presented in Sect. 2, depending on the type of chosen WR iteration scheme, different splittings will be performed, resulting in different error propagation matrices \mathbf{R} . The above derivation is for the a-form of the WR-Newmark method; a similar derivation for the d-form is presented in “Appendix C”.

我们分析 WR-Newmark 方法在考虑自由度数、时间积分窗口和分裂方案的影响的自由振动问题上的性能。考虑三个示例问题：

4 Numerical results

In this section we analyze the performance of the WR-Newmark method on free vibration problems considering the effect of number of degrees of freedom, time integration window and splitting schemes. Three example problems are considered: a two mass-spring system both undamped and damped, and a two dimensional plate subjected to a prescribed initial displacement. In order to show the effectiveness of the proposed algorithm for different structures of mass and stiffness matrices, this last problem is solved using a finite element discretization and an RKPM meshfree method. Finally, we consider a bimaterial plate problem and illustrate the applicability of the WRN $_{\beta}$ method to multi-rate

time integration in which each material subdomain is integrated with a different time step. To assess the accuracy of the algorithm we track the displacements of the nodal points over time and compare them with the analytical solution in the two mass-spring system and with those obtained through the use of the traditional Newmark method for the 2D plate problems. The parameters $\beta = \frac{1}{4}$ and $\gamma = \frac{1}{2}$ are used for the time integration and a tolerance $\epsilon = 10^{-14}$ is used as the iteration convergence criterion.

All results in this section are obtained for a serial implementation in MATLAB on a macOS machine with a single double-core processor (3.1 GHz Intel Core i7). An LU decomposition is used for the solution step of the standard Newmark method.

4.1 Free vibration of a two mass-spring system

In this section we consider the free vibration of a two-mass-dashpot-spring system. We begin by the numerical solution of the free vibration of the undamped case that was presented as a model problem in Sect. 2.1, for which the analytical solution is known. To assess the accuracy of the WRN $_{\beta}$ Jacobi and WRN $_{\beta}$ Gauss–Seidel algorithms, we track the displacement/velocity/acceleration of mass m_2 over time and compare it with its analytical solution and the traditional Newmark method, as shown in Fig. 5. The mass matrix \mathbf{M} for this problem is diagonal as in the case of lumped mass matrices and $\mathbf{C} = \mathbf{0}$.

An initial displacement $\mathbf{d}_0(t) = (6, 12)^T$ [m] is imposed and the ratios between k_1 and k_2 and m_1 and m_2 are taken respectively as $\frac{k_1}{k_2} = 1$ and $\frac{m_1}{m_2} = 2$ with chosen values: $k_2 = \frac{1}{6} \left[\frac{\text{N}}{\text{m}} \right]$, $m_2 = 3$ [kg]. A time step size of $\Delta t = 0.01$ [s] is taken for all integration schemes.

Absolute errors in the displacement, velocity and accelerations solutions of the considered numerical schemes are also reported in Fig. 5. It can be seen that the error remains quite small on the order of 10^{-5} but increases with the simulation time.

Figure 6 shows the L_{∞} norm of the errors for the different considered schemes for various values of Δt ; a convergence rate of 2 with respect to the time step size is observed for all methods.

Inspection of Figs. 5 and 6 shows that, as expected, the presented WR-Newmark algorithm is comparable to the traditional Newmark method and converges to the exact solution.

Table 1 shows the spectral radii of the error propagation matrix \mathbf{R} for the numerical schemes used. It can be noted that the spectral radius of \mathbf{R} for the Gauss–Seidel-WRN $_{\beta}$ is lower with respect to the Jacobi-WRN $_{\beta}$, indicating a higher convergence rate of the former. This is confirmed by the number of iterations needed for convergence for the two methods

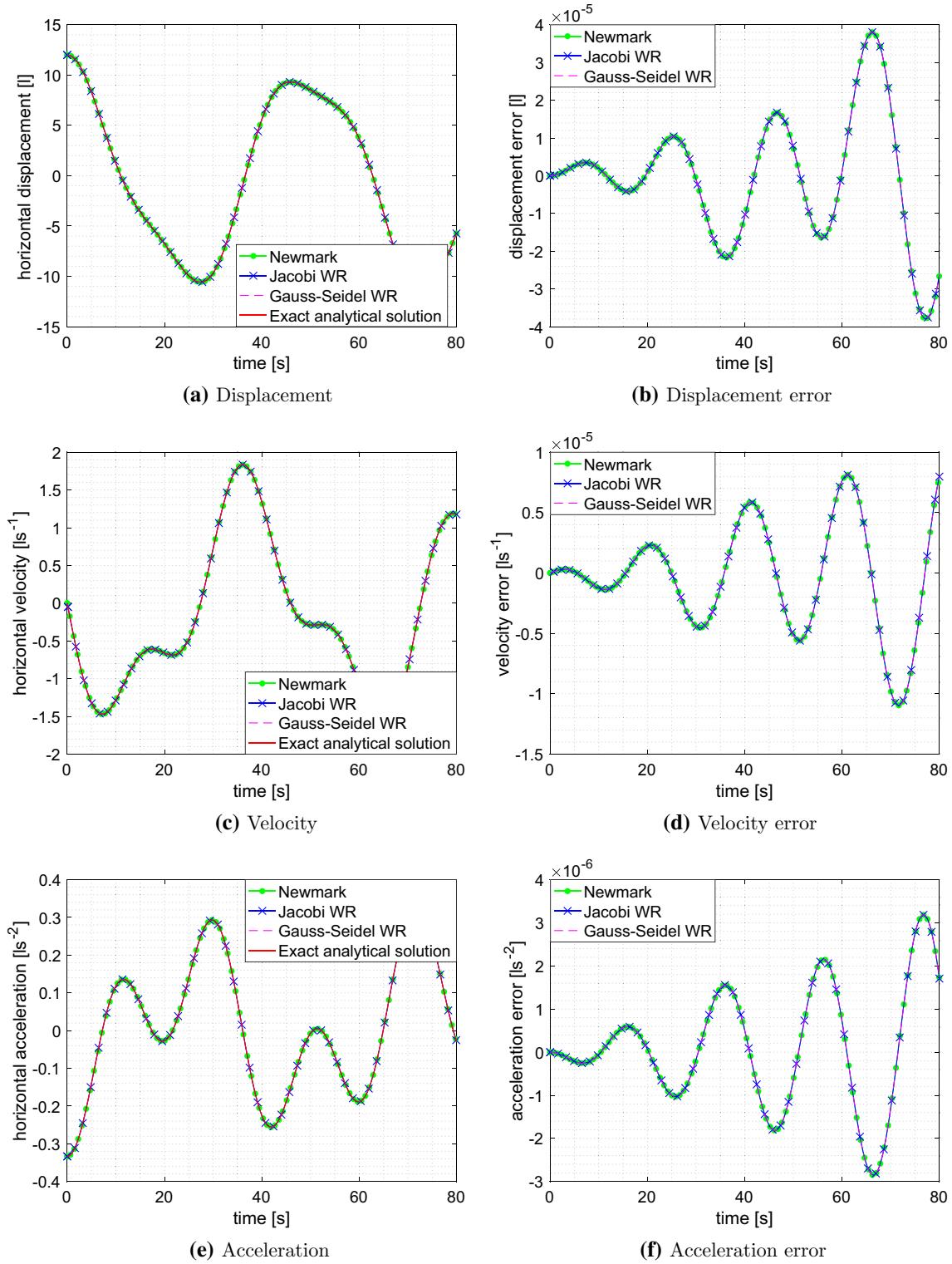


Fig. 5 Undamped two-mass system dynamic results reported for the m_2 degree of freedom. (Left) Comparison of displacement/velocity/acceleration of Newmark, WRN $_{\beta}$ Jacobi and WRN $_{\beta}$ GS methods with the analytical solution, $\Delta t = 0.01$ [s]; and (right) corresponding absolute error

报告了 m_2 自由度的无阻尼二质量系统动态结果

Fig. 6 L_∞ norm of the error for different values of Δt . The continuous lines represent the error in the displacement solution while the dashed and dotted ones represent the error in the velocity and acceleration, respectively. Note that the rate of convergence for all methods/fields is 2

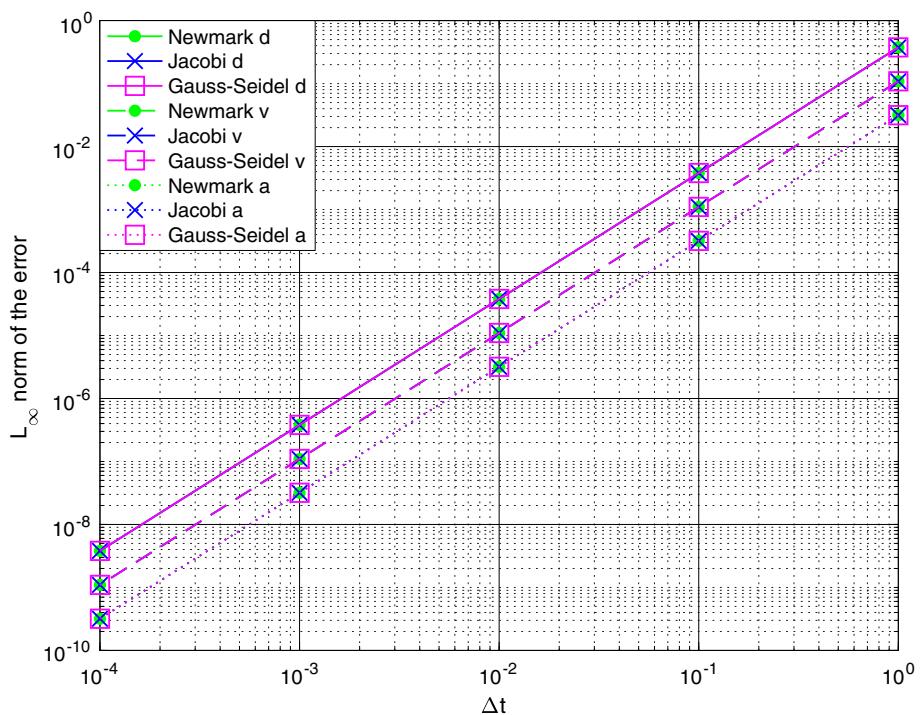


Table 1 Spectral radii of the error propagation matrix \mathbf{R} and number of iterations for for Jacobi WRN $_{\beta}$ and Gauss–Seidel WRN $_{\beta}$ and maximum natural frequency for the different methods. $\Delta t = 0.01$ [s]

Method	$\rho(\mathbf{R})$	Number of iterations	ω_{max}^h (Hz)
Standard Newmark	—	—	0.31
WRN $_{\beta}$ Jacobi	0.9820×10^{-6}	35	0.24
WRN $_{\beta}$ Gauss–Seidel	0.9645×10^{-12}	18	0.24

for the chosen $\Delta t = 0.01$ [s], reported in the same table. The maximum natural frequency for each of the considered methods is also reported in the Table. It can be seen that the frequency of the standard Newmark method is greater than those of the WR systems.

In order to demonstrate the applicability of the proposed method to damped systems, a case with $\mathbf{C} \neq \mathbf{0}$ is also considered. As for the undamped example, the accuracy of the proposed algorithm is assessed through comparison of the numerically obtained displacement of mass m_2 and its analytical solution. An initial displacement $\mathbf{d}_0(t) = (1, 2)^T$ [m] is imposed and the ratios between k_1 and k_2 and m_1 and m_2 are taken respectively as $\frac{k_1}{k_2} = 2$ and $\frac{m_1}{m_2} = 2$ with chosen values: $k_2 = 3 \left[\frac{\text{N}}{\text{m}} \right]$, $m_2 = 1$ [kg]. The damping matrix is taken as

$$\mathbf{C} = \begin{bmatrix} 0.2 & -0.1 \\ -0.1 & 0.1 \end{bmatrix} \begin{bmatrix} Ns \\ m \end{bmatrix}$$

The displacement solutions and absolute errors obtained by using the WR-Newmark algorithm with Jacobi and Gauss–

Seidel splittings with a time step $\Delta t = 0.01$ [s] are presented in Fig. 7 together with the traditional Newmark solution and the analytical solution. We note that even for damped systems, the results of the proposed WRN $_{\beta}$ method are comparable with those obtained from the traditional Newmark scheme.

Finally, we study the optimal window size in terms of computational cost for all WRN $_{\beta}$ methods. That is, convergence is achieved in each time window before moving on to the next one. In order to find the optimal time window size, the CPU time necessary to solve the monodimensional two mass problem with the Jacobi and Gauss–Seidel WR-Newmark methods is studied with increasing time window sizes. A time step $\Delta t = 0.01$ [s] and an overall time domain of 120 [s] are used. The obtained results shown in Table 2 suggest that a single time step is the least computationally expensive time-window for WR methods. Hence in subsequent examples we employ a single time step as our choice for time window length.

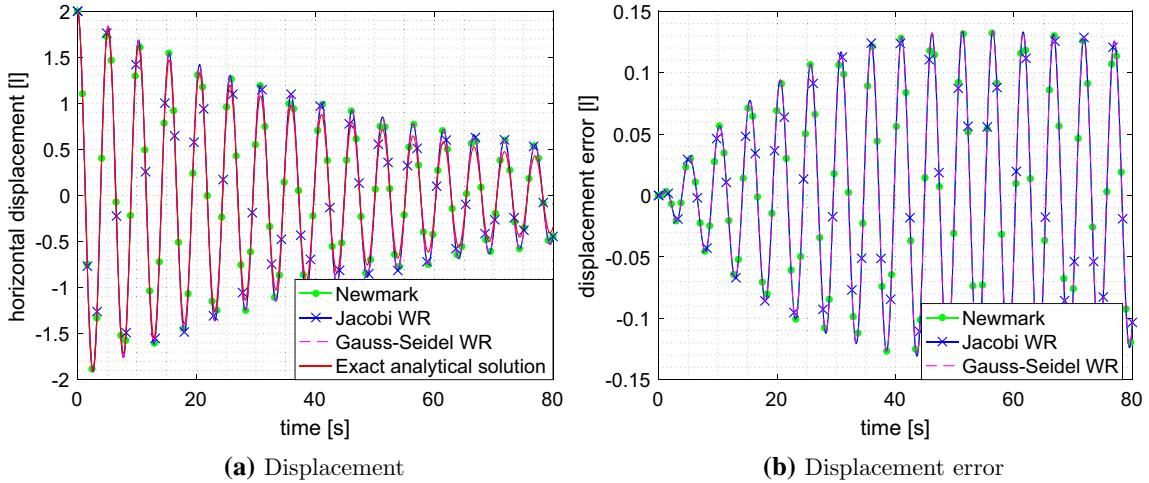


Fig. 7 Damped two-mass system dynamic results reported for the m_2 degree of freedom. (Left) Comparison of displacement of Newmark, WRN_β , Jacobi and WRN_β GS methods with the analytical solution, $\Delta t = 0.01$ [s]; and (right) corresponding absolute error

Table 2 CPU time for different time windows dimensions

Time window	WRN_β Jacobi CPU (s)	WRN_β GS CPU (s)
$1\Delta t$	0.0008	0.0008
$10\Delta t$	0.0012	0.0012
$50\Delta t$	0.021	0.023
$100\Delta t$	0.038	0.04

4.2 Two-dimensional plate (single material)

This example considers the free vibration of a two-dimensional plate subjected to an initial horizontal displacement. Herein the motion is described by Eq. (6) without damping, i.e. $\mathbf{C} = 0$. The problem is solved by means of two different discretization methods: a standard finite element discretization and an RKPM meshfree discretization. The plate has a total length of $L = 4$ [m] and a height of $H = 1.25$ [m]. Initial horizontal displacement with value equal to three times the nodal abscissa coordinate is applied and then immediately released. A fixed boundary displacement condition is present at the leading edge of the plate. Its Young's modulus, density and Poisson's ratio are chosen to be respectively: $E = 1[\frac{\text{N}}{\text{m}^2}]$, $\rho = 1[\frac{\text{kg}}{\text{m}^3}]$ and $\nu = 0.3$. Plane stress conditions are considered. Because of the complexity of the problem, an analytical solution is not available; therefore the solution obtained with the Newmark's method, using a refined time step ($\Delta t = 0.0001$ [s]) with respect to the one utilized for the WRN_β schemes ($\Delta t = 0.01$ [s]), is taken as the reference solution.

4.2.1 Finite element solution

In this subsection a finite element discretization is considered (Fig. 8).

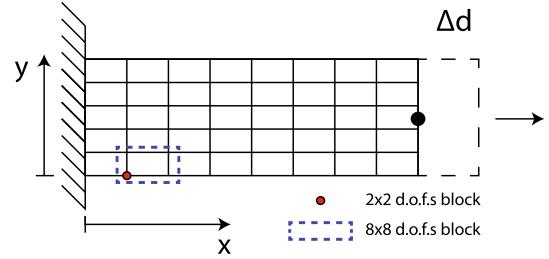


Fig. 8 Two dimensional plate discretized with finite elements. The nodes on the edge at $x = 0$ are fixed. The degrees of freedom associated with 2×2 and 8×8 partitioning block sizes are shown

\mathbf{M} and \mathbf{K} are therefore obtained by assembling the mass and stiffness matrices, \mathbf{m}_{el} and \mathbf{k}_{el} , respectively, of the bilinear quad elements used for the discretization. The element node numbering has been chosen so as to obtain a largely sparse coefficient matrix. The problem is solved by employing point-wise Jacobi, point-wise Gauss–Seidel and block Jacobi Waveform Relaxation Newmark algorithms. Unless otherwise specified, a point-wise scheme is inferred. In addition, we also consider a block Jacobi WRN_β scheme with blocks containing two and eight degrees of freedom; these, as shown in Fig. 8, refer to the degrees of freedom associated respectively with single nodes and individual elements.

Figure 9a shows the displacement of the free-end tip midpoint computed with the standard implicit Newmark method and with the WR-Newmark algorithms for 40 elements, while the corresponding error of the WRN_β schemes is presented in Fig. 9b. We note that for all the considered methods the L_2 norm of the absolute error is equal to 0.1743. The absolute errors were computed by comparing the solutions of the different methods to the reference solution over the total time domain. It can be observed that all the WRN_β methods attain the same solution accuracy of the standard Newmark scheme.

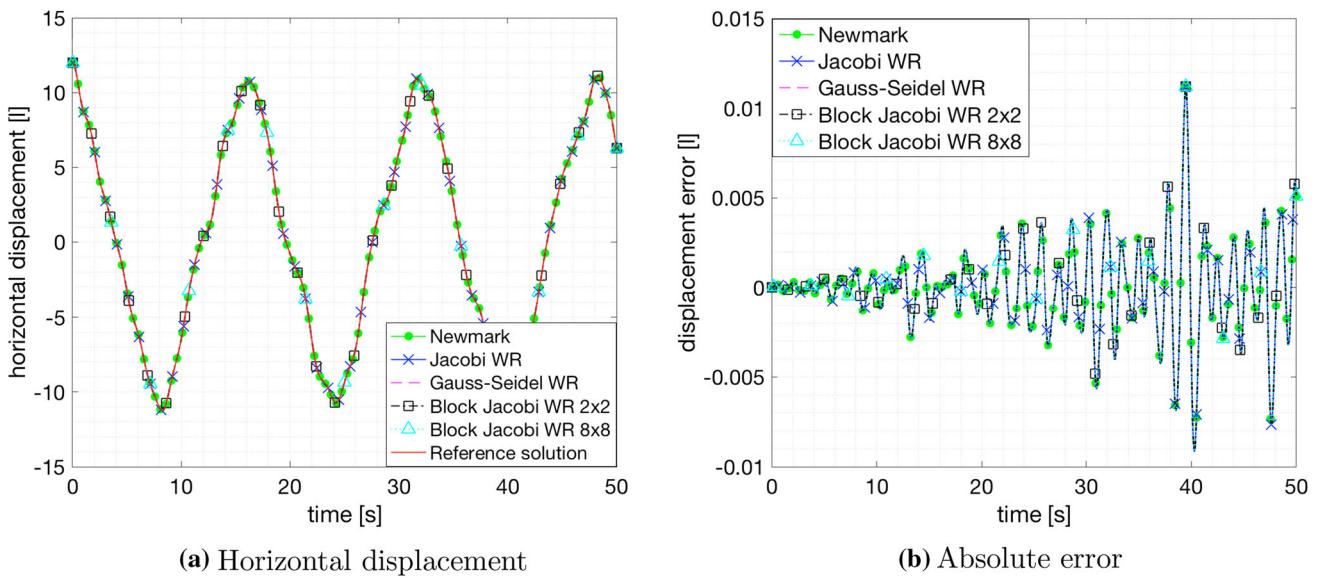


Fig. 9 Finite Element solution; **a** Horizontal displacement of the tip central point over time for the WRN_β methods, and **b** the error compared with a refined Newmark method

Table 3 Maximum natural frequency for different methods

Partitioning scheme	ω_{max}^h (Hz)
Standard Newmark	8.36
Jacobi	5.05
Gauss–Seidel	5.05

The maximum natural frequencies for the traditional system used in the Newmark method and for the point-wise Jacobi and Gauss–Seidel ones are presented in Table 3. The frequency of the traditional system is greater than those of the partitioned systems.

Figure 10 presents the eigenvalues of the error propagation matrix \mathbf{R} for the point-wise Jacobi-WRN $_\beta$ and the point-wise Gauss–Seidel-WRN $_\beta$ methods for the considered 40 element discretization. Note that the eigenvalues are all within the unit circle and that the Gauss–Seidel-WRN $_\beta$ eigenvalues are more clustered and closer to zero than the Jacobi ones, which indicates faster convergence of the Gauss–Seidel iterative scheme. Furthermore, Table 4 shows the number of iterations and the spectral radius of the error propagation matrix \mathbf{R} of the point-wise Jacobi and point-wise Gauss–Seidel numerical schemes for different number of elements. The results show that there is almost no change in their spectral radius, and therefore no change in the convergence behaviour in terms of WR iterations, with increase of system size.

Finally, we study the computational efficiency of the proposed WRN $_\beta$ algorithms in the solution of system of linear second order hyperbolic differential equations for an increasing number of elements as compared with the traditional

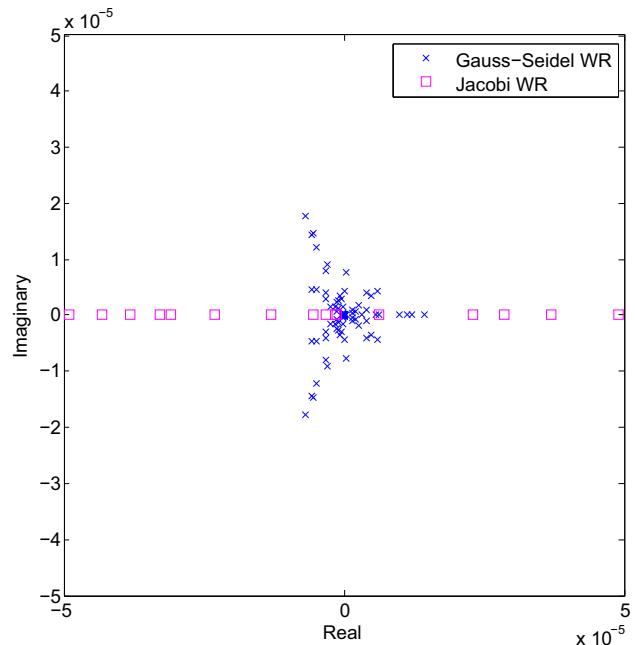


Fig. 10 Eigenvalues of the error propagation matrix \mathbf{R} for the 2D plate problem

Newmark's method. Figure 11 shows the cpu timings for a time window of one time step. One can clearly see a trend reversal in terms of CPU time required to solve the problem. Initially, for a small number of elements, the standard Newmark method performs well and requires the minimum time. However, as the number of elements increases, the trend is reversed and all WRN $_\beta$ methods outperform the standard Newmark method, confirming the effectiveness of the Wave-

Table 4 spectral radii of \mathbf{R} and number of iterations for increasing number of elements

Number of elements	$\rho(\mathbf{R})$		Number of iterations	
	Jacobi	GS	Jacobi	GS
500	0.5158	0.1223	11	7
1000	0.5310	0.1267	11	7
1500	0.5338	0.1275	11	7
2000	0.5347	0.1277	11	7
2500	0.5350	0.1278	11	7

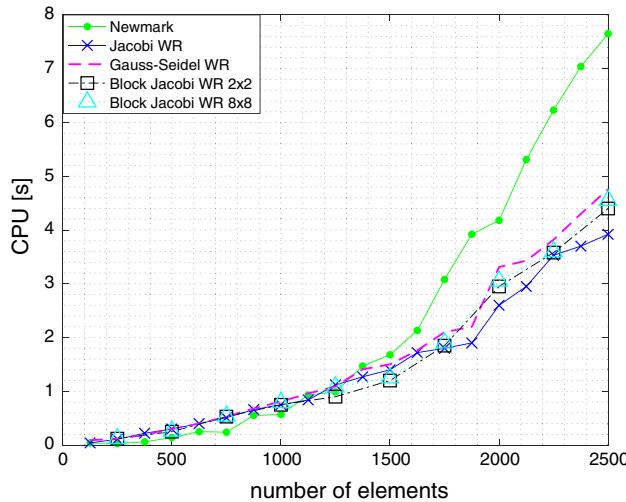


Fig. 11 CPU time vs number of elements. Newmark and point-wise WRN $_{\beta}$ methods. Notice that as the system increases, WRN $_{\beta}$ methods converge faster than the Newmark method

form Relaxation Newmark algorithm as a new class of more efficient time integrators.

The main reason is due to the matrix solve time in the Newmark method which increases dramatically with the size of the system, while in the WR Jacobi method there is no need to solve any system (each relaxation iteration is an explicit Newmark scheme) and the WR Gauss–Seidel method requires only to solve forward substitutions. Therefore even if the solution has to be computed as many times as required to obtain convergence of the iterations, its cost is still lower than the one to invert the original dense matrix for complex problems.

4.2.2 RKPM meshfree solution

We repeat the same problem using the RKPM meshfree method. Different from finite elements, where the shape functions and the related approximation space are strictly linked to element connectivity, in RKPM the approximate solutions are constructed over a node-based meshfree discretization of the considered domain [33,34] (Fig. 12), hence the system

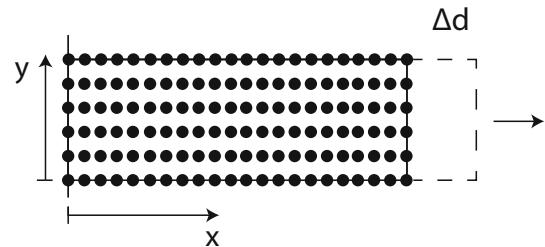


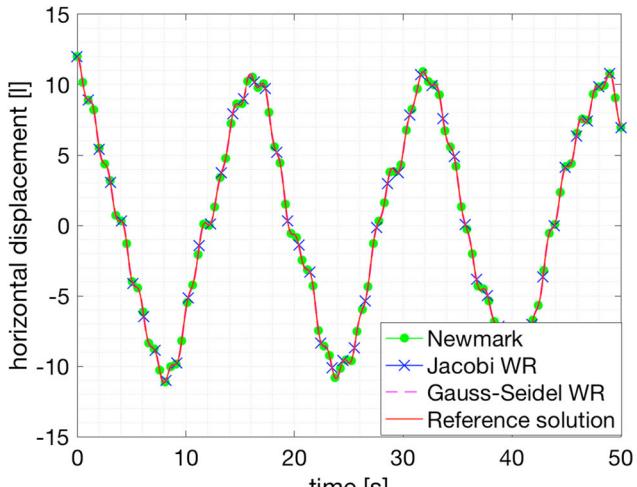
Fig. 12 Two dimensional plate discretized with discrete points

is denser and not compactly supported. An overview of the construction of the RKPM shape functions is provided in “Appendix D”.

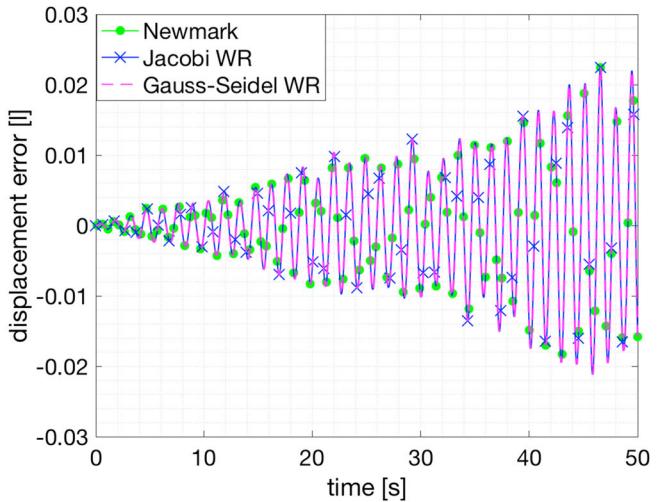
To assess the accuracy of the WRN $_{\beta}$ method, the plate was discretized with 54 nodes and a second order basis, a cubic B-spline kernel and a normalized support size of 2.001 in both directions were used. Domain integration was performed by a background mesh with 2×2 Gauss integration over each nodal integration cell. In this work, boundary conditions were imposed through a penalty approach. Figure 13a shows the horizontal displacement of the free-end midpoint tip computed with the traditional implicit Newmark method and with the point-wise Jacobi and Gauss–Seidel WR-Newmark algorithms, while the corresponding errors of the WRN $_{\beta}$ schemes are presented in Fig. 13b. The L_2 norm values of the absolute error accumulated over time are equal to 0.5651 for all the different considered methods, meaning that the obtained results are comparable.

It has to be noted that the RKPM method recovers the solution and accuracy of a linear finite element solution as the normalized support size approaches 1 if a tent function is chosen as kernel. Given the nonlocal characteristic of the Meshfree approximation, different partitioning choices and integration schemes specifically designed for an RKPM approximation [35,36] might be more appropriate; however, the aim of this example is to show the flexibility of the proposed scheme and its possibility of being applied to systems coming from both Finite Element and Meshfree discretizations.

The effectiveness of the proposed WRN $_{\beta}$ algorithms is shown in Fig. 14 where the required CPU time for an increasing number of nodes and a time window of one time step was compared to the one required by the traditional Newmark’s method. Similar to the FEM results, a clear trend reversal is also observed in the RKPM meshfree time integration, with the WRN $_{\beta}$ outperforming the traditional Newmark method as the size of the system increases. This, together with the previous example, confirms that the Waveform Relaxation Newmark algorithm is effective for different structures of the mass and stiffness matrices, regardless whether they arise from a finite element or a meshfree discretization.



(a) Horizontal displacement



(b) Absolute error

Fig. 13 RKPM meshfree solution; **a** Horizontal displacement of the tip central point over time for the WRN_β methods, and **b** the error compared with a refined Newmark method

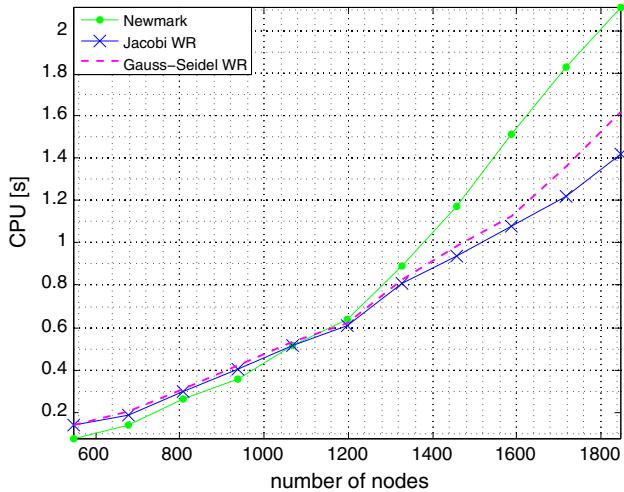


Fig. 14 CPU time vs number of nodes. Newmark and WRN_β methods. Notice that as the system increases, WRN_β methods converge faster than the Newmark method

4.3 Two-dimensional plate (two materials with multi-rate integration)

In the final example we illustrate the concept of multi-rate, or multiple time stepping of different subsystems using the WRN_β schemes as applied to a bimaterial plate.

As in Sect. 4.2, we consider the free vibration of a two-dimensional plate subjected to an initial horizontal displacement. The dimensions, density, Poisson's ratio of the plate, β and γ values, and initial prescribed displacement are the same as those chosen in Sect. 4.2. The plate is however composed of two different materials with Young's moduli

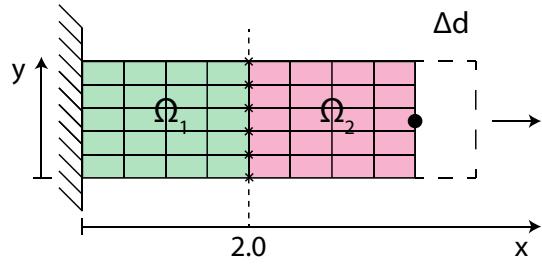


Fig. 15 Two dimensional plate composed of two materials, as indicated by the different colors, and discretized with finite elements. The nodes on the edge at $x = 0$ are fixed. Nodes at the connecting boundary between the two materials are indicated by crosses

equal to 1 and 10 when $x < 2.0$ and $x \geq 2.0$, respectively (see Fig. 15).

The problem is solved using the traditional Newmark scheme with $\Delta t = 0.01$ [s] and the Jacobi WRN_β algorithm. Exploiting the unstructuredness in time domain of the WR schemes, the degrees of freedom in the portion of the domain for which $x < 2.0$, hereon referred as Ω_1 (see Fig. 15), are integrated using a time step $\Delta t = 0.01$ [s], while the degrees of freedom in Ω_2 ($x \geq 2.0$), are solved using time steps equal to Δt , $2\Delta t$ and $5\Delta t$. Given that the subsystems of the WRN_β scheme are solved using different time steps sizes, an interpolating technique to exchange the information between subsystems needs to be defined [2]. Figure 16 illustrates this concept for the Jacobi WRN_β scheme. The degrees of freedom belonging to nodes at the interface between the two materials (indicated as d_2 in Fig. 16) use as external variables the values of the solution from degrees of freedom belonging to Ω_1 (called d_1), as well as from other degrees of freedom in Ω_2 (called d_3). However, given that a larger time step is

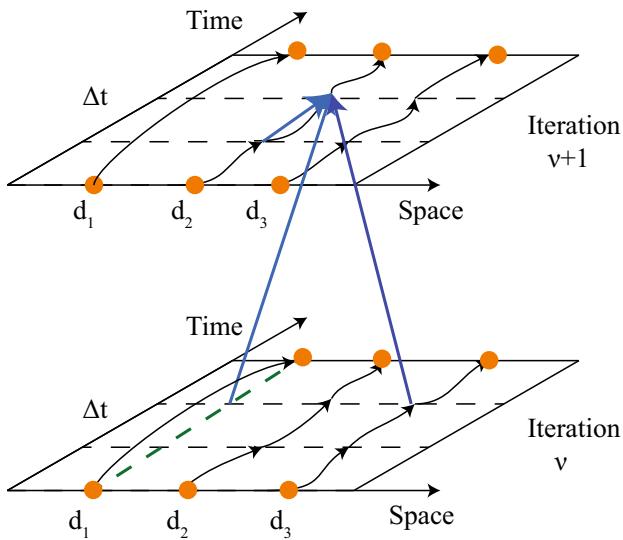


Fig. 16 Exchange of information between degrees of freedom at the boundary between the zones solved with different time steps, specifically Δt and $3\Delta t$

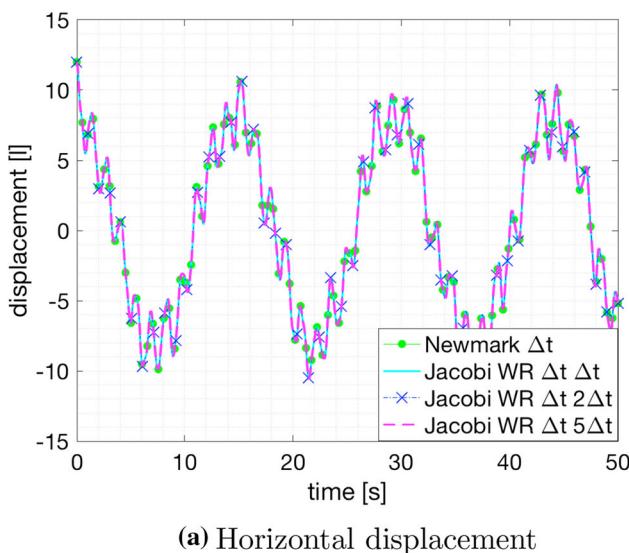
used for the unknowns in Ω_1 , the information regarding d_1 needed to solve for d_2 will be unknown at certain intermediate times. For this reason an interpolating procedure needs to be employed to approximate the values of the solution of the degrees of freedom belonging to domain 1 at those times. In this work a linear interpolation, represented by the dashed line in Fig. 16, is used.

Figure 17a shows the displacement over time of the plate tip central point obtained by using the aforementioned

schemes: traditional Newmark with Δt for all the degrees of freedom, and three cases of Jacobi WRN $_{\beta}$ with time step equal to Δt for domain 1 and Δt , $2\Delta t$ and $5\Delta t$ for domain 2, respectively. The absolute difference between the solution obtained with the traditional Newmark scheme and the considered Jacobi WRN $_{\beta}$ cases is presented in Fig. 17b. As for the previously considered numerical problems, when the same time step is used everywhere for both the Newmark and the WRN $_{\beta}$ schemes, the same level of accuracy is achieved. When are employed different time steps are employed, and hence an interpolating scheme is introduced, we can observe that the WRN $_{\beta}$ solution has a relatively small deviation from the traditional Newmark one. As expected, this difference becomes larger with the time step size used for the first material since the accuracy of the interpolation scheme is degraded.

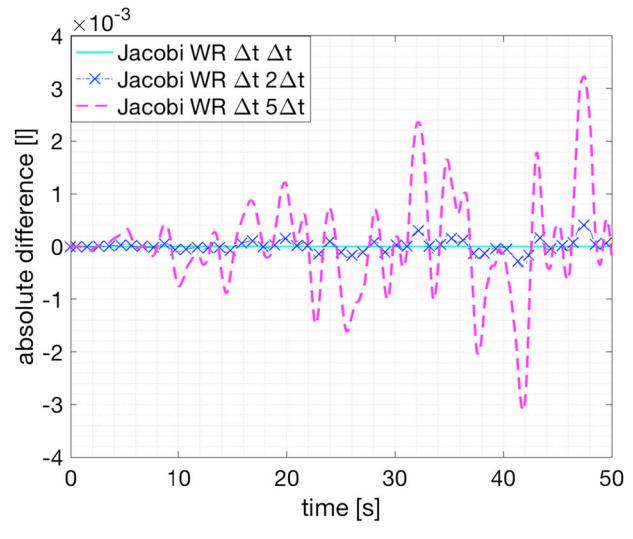
5 Conclusions

A Waveform Relaxation Newmark algorithm for the solution of linear structural dynamics hyperbolic systems was developed by pairing the implicit form of the Newmark's method and the iterative waveform relaxation schemes. This method has several notable features as compared with traditional Newmark methods: it requires iterations in the time domain, however, the convergence can be obtained quickly depending on the integration time window size; it retains unconditional stability but requires significantly lower computational costs which is emphasized for larger problem sizes; it is unstruc-



(a) Horizontal displacement

Fig. 17 Finite element solution of a bimaterial plate with multi-rate integration scheme. **a** Horizontal displacement of the tip central point over time for the WRN $_{\beta}$ Jacobi method with multiple time steps com-



(b) Absolute error

pared with the traditional Newmark method with a single (smallest) time step integration, and **b** the absolute error compared with the Newmark method

tured in time domain and well suited for time parallelization compared with the inherently sequential Newmark method; and it is well suited for multi-rate integration of subcomponents (such as in bimaterial problems).

While in the current work we focused on linear structural dynamics problems, the WR-Newmark algorithm is not limited to linear systems, and in future work we plan to apply the algorithm to general nonlinear structural problems. Furthermore, we plan to implement the method in parallel and study optimal splitting schemes for denser systems like the ones coming from RKPM and Isogeometric discretizations.

Appendix A

The exact analytical solution to (9) can be expressed as:

$$\begin{aligned} d_1(t) &= \hat{X}_{11} \left[g_{10} \cos(\omega_1 t) + \frac{\dot{g}_{10}}{\omega_1} \sin(\omega_1 t) \right] + \hat{X}_{12} \left[g_{20} \cos(\omega_2 t) + \frac{\dot{g}_{20}}{\omega_2} \sin(\omega_2 t) \right] \\ d_2(t) &= \hat{X}_{21} \left[g_{10} \cos(\omega_1 t) + \frac{\dot{g}_{10}}{\omega_1} \sin(\omega_1 t) \right] + \hat{X}_{22} \left[g_{20} \cos(\omega_2 t) + \frac{\dot{g}_{20}}{\omega_2} \sin(\omega_2 t) \right] \end{aligned}$$

where the analytical expressions for \hat{X}_{11} , \hat{X}_{12} , \hat{X}_{21} , \hat{X}_{22} , g_{10} , \dot{g}_{10} , g_{20} , \dot{g}_{20} , ω_1 and ω_2 are as follows:

$$\begin{aligned} \hat{X}_{11} &= \frac{k_1 m_1 - k_1 m_2 - k_2 m_2 + \hat{a}}{2k_2 m_1 \sqrt{m_2 + \frac{(k_1 m_1 - k_1 m_2 - k_2 m_2 + \hat{a})^2}{4k_2^2 m_1}}} \\ \hat{X}_{12} &= \frac{k_1 m_1 - k_1 m_2 - k_2 m_2 - \hat{a}}{2k_2 m_1 \sqrt{m_2 + \frac{(-k_1 m_1 + k_1 m_2 + k_2 m_2 + \hat{a})^2}{4k_2^2 m_1}}} \\ \hat{X}_{21} &= \frac{1}{\sqrt{m_2 + \frac{(k_1 m_1 - k_1 m_2 - k_2 m_2 + \hat{a})^2}{4k_2^2 m_1}}} \\ \hat{X}_{22} &= \frac{1}{\sqrt{m_2 + \frac{(-k_1 m_1 + k_1 m_2 + k_2 m_2 + \hat{a})^2}{4k_2^2 m_1}}} \\ g_{10} &= \frac{d_{1,0} \hat{a} + d_{1,0} k_1 m_1 - d_{1,0} k_1 m_2 - d_{1,0} k_2 m_2 + 2d_{2,0} k_2 m_2}{2k_2 \sqrt{m_2 + \frac{(k_1 m_1 - k_1 m_2 - k_2 m_2 + \hat{a})^2}{4k_2^2 m_1}}} \\ g_{20} &= \frac{-d_{1,0} \hat{a} + d_{1,0} k_1 m_1 - d_{1,0} k_1 m_2 - d_{1,0} k_2 m_2 + 2d_{2,0} k_2 m_2}{2k_2 \sqrt{m_2 + \frac{(-k_1 m_1 + k_1 m_2 + k_2 m_2 + \hat{a})^2}{4k_2^2 m_1}}} \\ \dot{g}_{10} &= \frac{v_{1,0} \hat{a} + v_{1,0} k_1 m_1 - v_{1,0} k_1 m_2 - v_{1,0} k_2 m_2 + 2v_{2,0} k_2 m_2}{2k_2 \sqrt{m_2 + \frac{(k_1 m_1 - k_1 m_2 - k_2 m_2 + \hat{a})^2}{4k_2^2 m_1}}} \end{aligned}$$

$$\begin{aligned} \dot{g}_{20} &= \frac{-v_{1,0} \hat{a} + v_{1,0} k_1 m_1 - v_{1,0} k_1 m_2 - v_{1,0} k_2 m_2 + 2v_{2,0} k_2 m_2}{2k_2 \sqrt{m_2 + \frac{(-k_1 m_1 + k_1 m_2 + k_2 m_2 + \hat{a})^2}{4k_2^2 m_1}}} \\ \omega_1 &= \sqrt{\frac{k_1 m_1 + k_1 m_2 + k_2 m_2 - \hat{a}}{2m_1 m_2}} \\ \omega_2 &= \sqrt{\frac{k_1 m_1 + k_1 m_2 + k_2 m_2 + \hat{a}}{2m_1 m_2}} \end{aligned}$$

where

$$\hat{a} = \sqrt{k_1^2 m_1^2 - 2k_1^2 m_1 m_2 + k_1^2 m_2^2 - 2k_1 k_2 m_1 m_2 + 2k_1 k_2 m_2^2 + 4k_2^2 m_1 m_2 + k_2^2 m_2^2}$$

The results shown in Fig. 4a, b were obtained for $m_1 = 6$ [kg], $m_2 = 3$ [kg], $k_1 = \frac{1}{6} [\frac{N}{m}]$, $k_2 = \frac{1}{6} [\frac{N}{m}]$, $d_{1,0} = 6$ [m], $d_{2,0} = 12$ [m], $v_{1,0} = 0 [\frac{m}{s}]$ and $v_{2,0} = 0 [\frac{m}{s}]$.

Appendix B

Initialization $v = 0$:

$$\begin{cases} d_1^{(0)}(t) = 6 \\ d_2^{(0)}(t) = 12 \end{cases}$$

Jacobi WR:

- Iteration $v = 1$:

$$\begin{cases} 6\ddot{d}_1^{(1)}(t) + \frac{1}{3}d_1^{(1)}(t) = 2 \\ 3\ddot{d}_2^{(1)}(t) + \frac{1}{6}d_2^{(1)}(t) = 1 \end{cases}$$

Solution:

$$\begin{cases} d_1^{(1)}(t) = 6 \\ d_2^{(1)}(t) = 6 \left[1 + \cos \left(\frac{t}{3\sqrt{2}} \right) \right] \end{cases}$$

- Iteration $v = 2$:

$$\begin{cases} 6\ddot{d}_1^{(2)}(t) + \frac{1}{3}d_1^{(2)}(t) = 1 + \cos \left(\frac{t}{3\sqrt{2}} \right) \\ 3\ddot{d}_2^{(2)}(t) + \frac{1}{6}d_2^{(2)}(t) = 1 \end{cases}$$

Solution:

$$\begin{cases} d_1^{(2)}(t) = 3 + 3 \cos \left(\frac{t}{3\sqrt{2}} \right) + \frac{t}{2\sqrt{2}} \sin \left(\frac{t}{3\sqrt{2}} \right) \\ d_2^{(2)}(t) = 6 \left[1 + \cos \left(\frac{t}{3\sqrt{2}} \right) \right] \end{cases}$$

- Iteration $\nu = 3$:

$$\begin{cases} 6\ddot{d}_1^{(3)}(t) + \frac{1}{3}d_3^{(2)}(t) = 1 + \cos\left(\frac{t}{3\sqrt{2}}\right) \\ 3\ddot{d}_2^{(3)}(t) + \frac{1}{6}d_3^{(2)}(t) = \frac{1}{2} + \frac{1}{2}\cos\left(\frac{t}{3\sqrt{2}}\right) \\ \quad + \frac{t}{12\sqrt{2}}\sin\left(\frac{t}{3\sqrt{2}}\right) \end{cases}$$

Solution:

$$\begin{cases} d_1^{(3)}(t) = 3 + 3\cos\left(\frac{t}{3\sqrt{2}}\right) + \frac{t}{2\sqrt{2}}\sin\left(\frac{t}{3\sqrt{2}}\right) \\ d_2^{(3)}(t) = 3 + \left(9 - \frac{t^2}{48}\right)\cos\left(\frac{t}{3\sqrt{2}}\right) + \frac{5t}{8\sqrt{2}}\sin\left(\frac{t}{3\sqrt{2}}\right) \end{cases}$$

Gauss-Seidel WR:

- Iteration $\nu = 1$:

$$\begin{cases} 6\ddot{d}_1^{(1)}(t) + \frac{1}{3}d_1^{(1)}(t) = 2 \\ 3\ddot{d}_2^{(1)}(t) + \frac{1}{6}d_2^{(1)}(t) = 1 \end{cases}$$

Solution:

$$\begin{cases} d_1^{(1)}(t) = 6 \\ d_2^{(1)}(t) = 6\left[1 + \cos\left(\frac{t}{3\sqrt{2}}\right)\right] \end{cases}$$

- Iteration $\nu = 2$:

$$\begin{cases} 6\ddot{d}_1^{(2)}(t) + \frac{1}{3}d_1^{(2)}(t) = 1 + \cos\left(\frac{t}{3\sqrt{2}}\right) \\ 3\ddot{d}_2^{(2)}(t) + \frac{1}{6}d_2^{(2)}(t) = \frac{1}{2} + \frac{1}{2}\cos\left(\frac{t}{3\sqrt{2}}\right) \\ \quad + \frac{t}{12\sqrt{2}}\sin\left(\frac{t}{3\sqrt{2}}\right) \end{cases}$$

Solution:

$$\begin{cases} d_1^{(2)}(t) = 3 + 3\cos\left(\frac{t}{3\sqrt{2}}\right) + \frac{t}{2\sqrt{2}}\sin\left(\frac{t}{3\sqrt{2}}\right) \\ d_2^{(2)}(t) = 3 + \left(9 - \frac{t^2}{48}\right)\cos\left(\frac{t}{3\sqrt{2}}\right) + \frac{5t}{8\sqrt{2}}\sin\left(\frac{t}{3\sqrt{2}}\right) \end{cases}$$

- Iteration $\nu = 3$:

$$\begin{cases} 6\ddot{d}_1^{(3)}(t) + \frac{1}{3}d_3^{(2)}(t) \\ = \frac{1}{6}\left[3 + (9 - \frac{t^2}{48})\cos\left(\frac{t}{3\sqrt{2}}\right) + \frac{5t}{8\sqrt{2}}\sin\left(\frac{t}{3\sqrt{2}}\right)\right] \\ 3\ddot{d}_2^{(3)}(t) + \frac{1}{6}d_3^{(2)}(t) \\ = \frac{1}{6}\left[\frac{3}{2} - \frac{1}{64}(-288 + t^2)\cos\left(\frac{t}{3\sqrt{2}}\right) + \right. \\ \left. - \frac{\sqrt{2}t}{3456}(-1458 + t^2)\sin\left(\frac{t}{3\sqrt{2}}\right)\right] \end{cases}$$

Solution:

$$\begin{cases} d_1^{(3)}(t) = \frac{3}{2} - \frac{1}{64}(-288 + t^2)\cos\left(\frac{t}{3\sqrt{2}}\right) \\ - \frac{\sqrt{2}t}{3456}(-1458 + t^2)\sin\left(\frac{t}{3\sqrt{2}}\right) \\ d_2^{(3)}(t) = \frac{3}{2} + \frac{1}{82944}(870912 - 3294t^2 + t^4)\cos\left(\frac{t}{3\sqrt{2}}\right) + \\ + \frac{\sqrt{2}t}{13824}(6831 - 7t^2)\sin\left(\frac{t}{3\sqrt{2}}\right) \end{cases}$$

Appendix C

$$\begin{aligned} \mathbf{M}\ddot{\mathbf{d}}_{n+1} + \mathbf{Kd}_{n+1} &= \mathbf{f}_{n+1} \\ (\mathbf{M}_+ - \mathbf{M}_-)\ddot{\mathbf{d}}_{n+1} - (\mathbf{K}_+ + \mathbf{K}_-)\mathbf{d}_{n+1} &= \mathbf{f}_{n+1} \\ \mathbf{M}_+\ddot{\mathbf{d}}_{n+1}^{\nu+1} + \mathbf{K}_+\mathbf{d}_{n+1}^{\nu+1} &= \mathbf{f}_{n+1} + \mathbf{M}_-\ddot{\mathbf{d}}_{n+1}^\nu + \mathbf{K}_-\mathbf{d}_{n+1}^\nu \end{aligned}$$

Now considering the Newmark's scheme with $\Delta t \neq 0$ and predictor:

$$\hat{\mathbf{a}}_{n+1}^{\nu+1} = \frac{1}{\beta\Delta t^2}\mathbf{d}_n^{\nu+1} + \frac{1}{\beta\Delta t}\dot{\mathbf{d}}_n^{\nu+1} + \left(\frac{1}{2\beta} - 1\right)\ddot{\mathbf{d}}_n^{\nu+1}$$

The solution phase then becomes:

$$\begin{aligned} \left(\frac{\mathbf{M}_+}{\beta\Delta t^2} + \mathbf{K}_+\right)\mathbf{d}_{n+1}^{\nu+1} &= \mathbf{f}_{n+1} + \mathbf{K}_-\mathbf{d}_{n+1}^\nu \\ &+ \mathbf{M}_-\ddot{\mathbf{d}}_{n+1}^\nu - \mathbf{M}_+\hat{\mathbf{a}}_{n+1}^{\nu+1} \end{aligned}$$

From the corrector step coming from the previous iteration ν , $\ddot{\mathbf{d}}_{n+1}^\nu = \frac{1}{\beta\Delta t^2}\mathbf{d}_{n+1}^\nu - \hat{\mathbf{a}}_{n+1}^\nu$, leading to:

$$\begin{aligned} \left(\frac{\mathbf{M}_+}{\beta \Delta t^2} + \mathbf{K}_+ \right) \mathbf{d}_{n+1}^{v+1} &= \mathbf{f}_{n+1} \\ &+ (\mathbf{K}_- + \frac{\mathbf{M}_-}{\beta \Delta t^2}) \mathbf{d}_{n+1}^v - \mathbf{M}_- \hat{\mathbf{a}}_{n+1}^v - \mathbf{M}_+ \hat{\mathbf{a}}_{n+1}^{v+1} \end{aligned}$$

Rewrite as:

$$\mathbf{P}\mathbf{d}^{v+1} = \mathbf{Q}\mathbf{d}^v + \tilde{\mathbf{f}}$$

where the subscript $(n+1)$ has been dropped and where $\mathbf{P} = \frac{\mathbf{M}_+}{\beta \Delta t^2} + \mathbf{K}_+$, $\mathbf{Q} = \mathbf{K}_- + \frac{\mathbf{M}_-}{\beta \Delta t^2}$ and $\tilde{\mathbf{f}} = \mathbf{f} - \mathbf{M}_- \hat{\mathbf{a}}^v - \mathbf{M}_+ \hat{\mathbf{a}}^{v+1}$, which leads to

$$\mathbf{d}^{v+1} = \mathbf{P}^{-1} \mathbf{Q} \mathbf{d}^v + \mathbf{P}^{-1} \tilde{\mathbf{f}}$$

The relationship between the error of two successive iterations can be written as:

$$\mathbf{e}^{v+1} = \mathbf{P}^{-1} \mathbf{Q} \mathbf{e}^v = \mathbf{R} \mathbf{e}^v$$

where $\mathbf{R} = \mathbf{P}^{-1} \mathbf{Q}$ is the error propagation matrix. For convergence:

$$\begin{aligned} \rho(\mathbf{R}) &< 1 \\ \rho(\mathbf{P}^{-1} \mathbf{Q}) &< 1 \\ \rho \left[\left(\frac{\mathbf{M}_+}{\beta \Delta t^2} + \mathbf{K}_+ \right)^{-1} \left(\mathbf{K}_- + \frac{\mathbf{M}_-}{\beta \Delta t^2} \right) \right] &< 1 \\ \rho[(\mathbf{M}_+ + \beta \Delta t^2 \mathbf{K}_+)^{-1} (\mathbf{M}_- + \beta \Delta t^2 \mathbf{K}_-)] &< 1 \end{aligned}$$

Appendix D

In the RKPM meshfree approach, the closed domain $\bar{\Omega} \subset \mathbb{R}^3$ is discretized by a set of NP nodes $\{\mathbf{x}_I | \mathbf{x}_I \in \bar{\Omega}\}_{I=1}^{NP}$ and the reproducing kernel (RK) approximation $u^h(\mathbf{x})$ of a generic function $u(\mathbf{x})$ in $\bar{\Omega}$ is defined as:

$$u^h(\mathbf{x}) = \sum_{I=1}^{NP} \chi_I(\mathbf{x}) u_I, \quad (\text{E-1})$$

where $\{\chi_I(\mathbf{x})\}_{I=1}^{NP}$ is the set of RK shape functions and $\{u_I\}_{I=1}^{NP}$ is the set of nodal coefficients of the approximation [33,34]. RK shape functions are constructed as the product of a weight kernel function $\Phi_a(\mathbf{x} - \mathbf{x}_I)$ with compact support measure a , such that $\Phi_I = \{\mathbf{x} | \Phi_a(\mathbf{x} - \mathbf{x}_I) \neq 0\}$, and of a correction function $C(\mathbf{x}; \mathbf{x} - \mathbf{x}_I)$:

$$\chi_I(\mathbf{x}) = \Phi_a(\mathbf{x} - \mathbf{x}_I) C(\mathbf{x}; \mathbf{x} - \mathbf{x}_I). \quad (\text{E-2})$$

Depending on the support size a , different sets of nodes interact with each other, leading to different structures of the mass \mathbf{M} and stiffness matrix \mathbf{K} .

The correction function $C(\mathbf{x}; \mathbf{x} - \mathbf{x}_I)$ is defined as a linear combination of monomial basis functions:

$$\begin{aligned} C(\mathbf{x}; \mathbf{x} - \mathbf{x}_I) &= \sum_{i+j+k=0}^n (x_1 - x_{I1})^i (x_2 - x_{I2})^j (x_3 - x_{I3})^k b_{ijk} \\ &\equiv \mathbf{h}^T(\mathbf{x} - \mathbf{x}_I) \mathbf{b}(\mathbf{x}), \end{aligned} \quad (\text{E-3})$$

where $\mathbf{b}(\mathbf{x})$ is the column vector of the monomial coefficients b_{ijk} and $\mathbf{h}^T(\mathbf{x} - \mathbf{x}_I)$ is the row vector of the monomial bases. The basis order n defines the order of the shape functions and the reproducing completeness while the choice of the kernel function $\Phi_a(\mathbf{x} - \mathbf{x}_I)$ determines the smoothness of the approximation functions. The coefficients $\mathbf{b}(\mathbf{x})$ are obtained by substituting (E-2) in the following polynomial reproducing conditions:

$$\sum_{I=1}^{NP} \chi_I(\mathbf{x}) x_{I1}^i x_{I2}^j x_{I3}^k = x_1^i x_2^j x_3^k, \quad 0 \leq i + j + k \leq n. \quad (\text{E-4})$$

or equivalently,

$$\sum_{I=1}^{NP} \chi_I(\mathbf{x}) (x_1 - x_{I1})^i (x_2 - x_{I2})^j (x_3 - x_{I3})^k = \delta_{ijk0} \quad (\text{E-5})$$

Once $\mathbf{b}(\mathbf{x})$ is obtained from (E-5), the RK shape functions are constructed as

$$\chi_I(\mathbf{x}) = \mathbf{h}^T(\mathbf{0}) \mathbf{M}^{-1}(\mathbf{x}) \mathbf{h}(\mathbf{x} - \mathbf{x}_I) \Phi_a(\mathbf{x} - \mathbf{x}_I), \quad (\text{E-6})$$

where $\mathbf{M}(\mathbf{x})$ is called the moment matrix and is defined as follows:

$$\mathbf{M}(\mathbf{x}) = \sum_{I=1}^{NP} \mathbf{h}(\mathbf{x} - \mathbf{x}_I) \mathbf{h}^T(\mathbf{x} - \mathbf{x}_I) \Phi_a(\mathbf{x} - \mathbf{x}_I). \quad (\text{E-7})$$

In general, the RK nodal coefficients (usually called generalized displacements) are not equivalent to the physical displacements at the nodes ($u_I \neq u^h(\mathbf{x}_I)$) [33], meaning that the RK shape functions, differently from the finite element ones, lack the Kronecker delta property. Because of this, exact imposition of essential boundary conditions in a RKPM framework is not trivial. Several methods have been proposed for the enforcement of Dirichlet boundary conditions in the RK approximation framework [37–39]. Such methods include the transformation method [33,37], the Reproducing Kernel Interpolation functions [39], the boundary singular kernel method [40], the Lagrange multiplier method [41], the penalty method and the Nitsche's method [42,43].

References

1. Hughes TJR (2012) The finite element method—linear static and dynamic finite element analysis. Courier Dover Publications, New York
2. Waismann H, Fish J, Fish J (1996) A space-time multilevel method for molecular dynamics simulations. *Comput Methods Appl Mech Eng* 195(44–47):6542–6559
3. Gander MJ (2015) 50 years of time parallel time integration. *Contrib Math Comput Sci* 9:69–113
4. Nievergelt J (1964) Parallel methods for integrating ordinary differential equations. *Commun ACM* 7(12):731–733
5. Bellen A, Zennaro M (1989) Parallel algorithms for initial-value problems for difference and differential equations. *J Comput Appl Math* 25:341–350
6. Chartier P, Philippe B (1993) A parallel shooting technique for solving dissipative odes. *Computing* 51:209–236
7. Miranker WL, Liniger W (1967) Parallel methods for the numerical integration of ordinary differential equations. *Math Comput* 21(99):303–320
8. Sheen D, Sloan IH, Thomée V (1999) A parallel method for time-discretization of parabolic problems based on contour integral representation and quadrature. *Math Comput* 69(299):177–195
9. Christlieb AJ, Macdonald CB, Ong BW (2010) Parallel high-order integrators. *SIAM J Sci Comput* 32(2):818–835
10. Gütteil S (2013) A parallel overlapping time-domain decomposition method for odes. *Domain Decompos Methods Sci Eng* XX:459–466
11. Hackbusch W (1984) Parabolic multi-grid methods. *Comput Methods Appl Sci Eng* VI:189–197
12. Horton G, Vandewalle S (1995) A space-time multigrid method for parabolic partial differential equations. *SIAM J Sci Comput* 16(4):848–864
13. Lelarasmee E (1982) The waveform relaxation method for time domain analysis of large scale integrated circuits: theory and applications. PhD thesis, EECS Department, University of California, Berkeley
14. Lelarasmee E, Ruehli AE, Sangiovanni-Vincentelli AL (1982) The waveform relaxation method for time-domain analysis of large scale integrated circuits. *IEEE Trans Comput Aided Des Integr Circuits Syst* 1(3):131–145
15. Gander MJ, Jiang YL, Li RJ (2013) Parareal schwarz waveform relaxation methods. *Domain Decomposition Methods in Science and Engineering*, vol 60. Lecture Notes in Computational Science and Engineering, pp 45–56
16. Gander MJ (1998) Overlapping Schwarz for parabolic problems. In: Proceedings of the ninth international conference on domain decomposition methods
17. Horton G, Vandewalle S (1995) A space–time multigrid method for parabolic partial differential equations. *SIAM J Sci Comput* 16:848–864
18. Hulbert GM, Hughes TJR (1990) Space-time finite element methods for second order hyperbolic equations. *Comput Methods Appl Mech Eng* 84:327–348
19. White J, Sangiovanni-Vincentelli A (1985) Waveform relaxation: theory and practice. *Trans Soc Comput Simul* 2:95–133
20. Janssen J, Vandewalle S (1996) Multigrid waveform relaxation on spatial finite element meshes: the discrete-time case. *SIAM J Sci Comput* 17(1):133–155
21. Janssen J, Vandewalle S (1996) Multigrid waveform relaxation on spatial finite element meshes: the continuous-time case. *SIAM J Numer Anal* 33(2):456–474
22. Ta’asan S, Zhang H (1995) On the multigrid waveform relaxation method. *SIAM J Sci Comput* 16(5):1092–1104
23. Vandewalle S (1992) The parallel solution of parabolic partial differential equations by multigrid waveform relaxation methods. PhD thesis, Department of Computer Science, Katholieke Universiteit Leuven
24. Vandewalle S, Horton G (1996) Fourier mode analysis of the multi-grid waveform relaxation and time-parallel multigrid methods. *Computing* 54:317–330
25. Miekkala U, Nevanlinna O (1987) Convergence of dynamic iteration methods for initial value problems. *SIAM J Sci Stat Comput* 8–4:459–482
26. Miekkala U, Nevanlinna O (1987) Sets of convergence and stability regions. *BIT* 27(4):554–584
27. Reichelt MW (1995) Optimal convolution sor acceleration of waveform relaxation with application to parallel simulation of semiconductor devices. *SIAM J Sci Stat Comput* 16(5):1137–1158
28. Janssen J, Vandewalle S (1997) On sor waveform relaxation methods. *SIAM J Numer Anal* 34:2456–2481
29. Liu J, Jiang Y-L (2012) A parareal algorithm based on waveform relaxation. *Math Comput Simul* 82:2167–2181
30. Maday Y, Turinici G (2005) The parareal in time iterative solver: a further direction to parallel implementation. *Domain Decomposition Methods in Science and Engineering*, vol 40. Lecture Notes in Computational Science and Engineering, pp 441–448
31. Jiang YL (2001) Waveform relaxation of nonlinear second-order differential equations. *Circuits Syst I Fundam Theory Appl* 48(11):1344–1347
32. Osawa K, Yamada S (1998) Waveform relaxation for second order differential equation $y''=f(x, y)$. *Lect Notes Comput Sci* 1470:780–787
33. Chen JS, Pan C, Wu C-T, Liu WK (1996) Reproducing kernel particle methods for large deformation analysis of non-linear structures. *Comput Methods Appl Mech Eng* 139:195–227
34. Liu WK, Jun S, Zhang YF (1995) Reproducing kernel particle methods. *Int J Numer Methods Fluids* 20:1081–1106
35. Chen JS, Wu C-T, Yoon S, You Y (2001) A stabilized conformal nodal integration for Galerkin mesh-free methods. *Int J Numer Methods Eng* 50:435–466
36. Hillman M, Chen JS (2015) An accelerated, convergent and stable nodal integration in Galerkin meshfree methods for linear and nonlinear mechanics. *Int J Numer Methods Eng* 107:603–630
37. Günter FC, Liu WK (1998) Implementation of boundary conditions for meshless methods. *Comput Methods Appl Mech Eng* 163:205–230
38. Chen JS, Hillman M, Chi SW (2017) Meshfree methods: progress made after 20 years. *J Eng Mech* 143(4):04017001
39. Chen JS, Han W, You Y, Meng X (2003) A reproducing kernel method with nodal interpolation property. *Int J Numer Methods Eng* 56(7):935–960
40. Chen JS, Wang H-P (2000) New boundary condition treatments in meshfree computation of contact problems. *Comput Methods Appl Mech Eng* 187:441–468
41. Belytschko T, Lu YY, Gu L (1994) Element-free Galerkin methods. *Int J Numer Methods Eng* 37:229–256
42. Nitsche JA (1970–1971) Über ein Variationsprinzip zur Lösung von Dirichlet-Problemen bei Verwendung von Teilräumen, die keinen Randbedingungen unterworfen sind. *Abhandlungen aus dem mathematischen Seminar der Universität Hamburg* 36:9–15
43. Fernandez-Mendez S, Huerta A (2004) Imposing essential boundary conditions in mesh-free methods. *Comput Methods Appl Mech Eng* 193:1257–1275

Publisher’s Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.