Fast Transient Simulation of System-Level Power Delivery Networks via Parallel Waveform Relaxation

Fast Transient Simulation of System-Level Power Delivery Networks via Parallel Waveform Relaxation

Alessandro Moglia, Antonio Carlucci, *Graduate Student Member, IEEE*, Stefano Grivet-Talocia, *Fellow, IEEE*, Siddharth Kulasekaran, Kaladhar Radhakrishnan, *Senior Member, IEEE*

***Abstract*—This application paper addresses the problem of transient simulation of system-level Power Distribution Networks (PDN) of multicore processing systems. In particular, we consider a post-layout Power Integrity verification problem where all system parts are finalized and a highly accurate transient verification is performed to ensure that voltage supply signals remain within prescribed bounds when the PDN is loaded by realistic current stimuli. Systems with tens of even hundreds of cores are considered, equipped with per-core local voltage stabilization, attained through Integrated Voltage Regulators (IVR) suitably controlled by sensing and feedback loops. Transient simulation of such system-level PDNs becomes particularly challenging when interconnect models or macromodels computed by electromagnetic solvers are embedded. In order to break system complexity, we propose a set of algorithms based on an ad-hoc system partitioning strategy, combined with multi-level Waveform Relaxation (WR) schemes. The main advantage of this approach is a straightforward parallelization, aimed at solving concurrently by parallel computing threads only small and well-defined circuit partitions. Several partitioning and associated WR schemes are discussed and tested, showing excellent scalability with up to 60 computing threads, with significant speedup in runtime with respect to a standard SPICE-based approach.**

I. Introduction

Numerical transient simulation of the large-scale circuits arising in Signal and Power Integrity applications is a longstanding challenge. The main difficulty arises from the interaction of two factors: i) the large size of the system equations, which is a direct consequence of embedding circuit representations of the electromagnetic behavior of interconnects with multiple scales, complex geometry, and overall size comparable with or even larger than the operation wavelength; ii) nonlinear behavior of components and subsystems, such as drivers/receivers in Signal Integrity (SI) or voltage regulation circuitry in Power Integrity (PI). It is well-known that the concurrent presence of such factors makes traditional circuit simulation particularly inefficient [1].

Several attempts have been proposed to tackle such difficulties. One approach is geared towards embedding few localized nonlinearities in full-wave electromagnetic solver codes, see e.g. [2]–[6]. This solution is promising for special cases but seems to be unable to consider a complete system-level scenario by solving at the same time multiscale interconnect routed through Printed Circuit Boards (PCB), packages, and chips. A complementary approach brings to the circuit domain accurate electromagnetic characterizations of interconnects, which are individually evaluated from frequency-domain field solvers in terms of scattering responses. The latter are converted to behavioral circuits through passive rational fitting macromodeling algorithms and tools. The resulting models are fully compatible with a circuit simulation environment such as SPICE, where nonlinear components are naturally embedded. This second approach is a standard in PI analysis [7], and most Computer Aided Design (CAD) tools provide semi-automated flows in this framework to support PI designers.

Modern developments in packaging and heterogeneous integration, driven by the everincreasing request for performance of microprocessor systems in High-Performance Computing HPC) and Artificial Itelligence (AI) applications, are pushing to their limit currently available system-level transient simulation approaches. Focusing on PI verification, various fundamental factors pose additional challenges, namely the larger and larger number of computing cores in microprocessors, associated to the fine-grained voltage regulation that is required at least on a per-core level [8]. These factors involve tens or hundreds of Integrated Voltage Regulators (IVR), each equipped with its own sensing and feedback loops to provide voltage stabilization [9]. Efficient numerical simulation at the SPICE level of such scenarios, even including state-of-the art reduced-order models of the Power Distribution Network (PDN), still remains an open issue.

In this paper, we propose a Waveform Relaxation (WR) framework to boost efficiency and reduce runtime in transient PI verification. WR approaches are well-known and well documented in the scientific literature. Starting from the early formulations and developments [10]–[15], various improvements and application fields have been addressed [16]–[18], and WR is still an active research field [19]–[21], including application to on-chip power grids [22], [23]. Waveform Relaxation first breaks the system into separate parts by a suitable decoupling strategy. Individual parts are then solved independently. Finally, the correct solution is attained through

A. Moglia, A. Carlucci, and S. Grivet-Talocia are with the Dept. of Electronics and Telecommunications, Politecnico di Torino, C. Duca degli Abruzzi 24, 10129 Torino, Italy (email: alessandro.moglia@polito.it, antonio.carlucci@polito.it, stefano.grivet@polito.it).

S. Mongrain, S. Kulasekaran, and K. Radhakrishnan are with Intel Corporation, Chandler, AZ, USA (email: siddharth.kulasekaran@intel.com, kaladhar.radhakrishnan@intel.com).

The work of Alessandro Moglia, Antonio Carlucci, and Stefano Grivet-Talocia was supported by Intel Corporation under the 2022-24 Intel Strategic Research Segment (SRS) Grant titled "API-S: Accelerated system-level transient Power Integrity Solvers."

iterations, where the neglected couplings are reintroduced as additional sources, usually denoted as *relaxation sources*. This approach is valid when the neglected couplings are small, so that convergence takes place in few iterations.

This manuscript should be regarded as an application paper, where existing WR approaches are tailored to the specific problem at hand. We do not propose a new WR scheme, rather we propose a WR application framework for transient simulation of system-level PDNs equipped by multiple voltage regulation feedback loops. In particular, we deploy various WR implementations corresponding to different system partitioning strategies, namely Longitudinal Partitioning (LP), Transverse Partitioning (TP) and their combination (LPTP), discussing the model requirements that make these approaches competitive for different PDN structures. All WR schemes are here implemented in a high-performance parallel (multithreaded) C code. We document major speedup with respect to brute-force SPICE solutions, exceeding three orders of magnitude when applied to real PDN models of commercial multicore systems (both mobile and enterprise server level). As an example, a full-system ramp-up transient analysis, which fails in SPICE due to convergence problems, is solved in few seconds using proposed WR framework.

This paper is organized as follows. Section II states the numerical simulation problem and sets notation for later developments. Section III provides a high-level description of the proposed WR schemes as applied to the application at hand. Section IV digs into some relevant details that enable an efficient parallel implementation. Section V presents and discusses numerical results on two reference PDN benchmarks, namely a 4-core mobile and a 60-core enterprise server. Finally, Section VI draws conclusions.

II. Formulation and Problem Statement

Let us consider the general system topology depicted in Fig. II. This topology is the same already addressed in previous publications on this subject, see e.g. [24]–[26]. In this section, we introduce the equations for all individual blocks to set up notation and to enable the developments of Sec. III. We anticipate the complete set of equations that we solve in this work in (1) below, referring to the list of relevant variables and parameters listed in Table I. These equations are illustrated and discussed in detail in this section.

$$\mathbf{E}_1\dot{\boldsymbol{x}}_1 = \mathbf{A}_1\boldsymbol{x}_1 + \textstyle\sum_{k'}\mathbf{B}_{1;k'}\boldsymbol{i}_{1;k'} + \mathbf{B}_{\mathrm{dc}}V_{\mathrm{dc}} \tag{1a}$$

$$\boldsymbol{v}_{1;k} = \mathbf{C}_{1;k}\boldsymbol{x}_1 + \textstyle\sum_{k'}\mathbf{D}_{1;kk'}\boldsymbol{i}_{1;k'} + \mathbf{D}_{\mathrm{dc};k}V_{\mathrm{dc}} \tag{1b}$$

$$\mathbf{E}_{2;k}\dot{\boldsymbol{x}}_{2;k} = \mathbf{A}_{2;k}\boldsymbol{x}_{2;k} + \mathbf{B}_{2;k}\boldsymbol{v}_{2;k} + \mathbf{B}_{o;k}\boldsymbol{i}_{o;k} \tag{1c}$$

$$\boldsymbol{i}_{2;k} = \mathbf{C}_{2;k}\boldsymbol{x}_{2;k} + \mathbf{D}_{22;k}\boldsymbol{v}_{2;k} + \mathbf{D}_{2o;k}\boldsymbol{i}_{o;k} \tag{1d}$$

$$\boldsymbol{v}_{o;k} = \mathbf{C}_{o;k}\boldsymbol{x}_{2;k} + \mathbf{D}_{o2;k}\boldsymbol{v}_{2;k} + \mathbf{D}_{oo;k}\boldsymbol{i}_{o;k} \tag{1e}$$

$$e_k = \mathbf{N}_k\boldsymbol{v}_{o;k} - V_{\mathrm{ref}} \tag{1f}$$

$$\dot{\boldsymbol{x}}_{\mathcal{K},k} = \mathbf{A}_{\mathcal{K},k}\boldsymbol{x}_{\mathcal{K},k} + \mathbf{B}_{\mathcal{K},k}e_k \tag{1g}$$

$$d_k = \sigma\left(\mathbf{C}_{\mathcal{K},k}\boldsymbol{x}_{\mathcal{K},k}; T_k\right) \tag{1h}$$

$$\boldsymbol{v}_{2;k} = d_k\boldsymbol{v}_{1;k} \tag{1i}$$

$$\boldsymbol{i}_{1;k} = -d_k\boldsymbol{i}_{2;k} \tag{1j}$$

$$\boldsymbol{i}_{o;k} = -\boldsymbol{i}_{s;k} \quad \text{(sources)} \tag{1k}$$

$$\text{for} \quad k = 1, \dots, N_c$$

The reference structure under analysis includes an *input network* $\mathcal{G}_1$ that embeds models of the PDN components connecting the reference platform voltage (V_{dc}, here modeled as an ideal voltage source) to the input stage of the FIVR switches. The input network includes power bus models at the board and package level, typically computed by 2.5D or 3D electromagnetic solvers in form of tabulated scattering responses, then converted to state-space form by passive rational fitting algorithms [27]–[29]. Suitable decoupling capacitor models are also embedded as terminations of the corresponding ports. The remaining ports of the overall input network are connected to V_{dc} and to the FIVR switches. Considering a system with N_c cores, each being regulated by an N_p-phase DC-DC buck converter, the total number of interface ports of the input network is $1 + N_cN_p$. We collect all voltages and current signals of the N_p phases for each k-th core in vectors $\boldsymbol{v}_{1;k}$ and $\boldsymbol{i}_{1;k}$. The entire input network can be considered as a Linear Time-Invariant (LTI) system (1a)-(1b), derived by assembling all component and interconnect models in descriptor form [25].

To the output of the FIVR switches we find the *output network*, which collects all components that provide the output filter of the buck converters (integrated inductors and MIM capacitors), as well as appropriate models for the on-chip power grid including on-chip decoupling capacitance. We assume that the entire output network can be partitioned into independent blocks $\mathcal{G}_{2;k}$, one for each core, labeled with the index $k = 1, \dots, N_c$. Also each per-core output network can be represented as an LTI subsystem, whose equations can be assembled in descriptor form (1c)–(1e). Note that the k-th output network has two block inputs, i.e., the voltages on the switch side $\boldsymbol{v}_{2;k}$ and the load currents $\boldsymbol{i}_{o;k}$, with the corresponding dual variables $\boldsymbol{i}_{2;k}$ and $\boldsymbol{v}_{o;k}$ considered as outputs. A total of N_c sets of independent descriptor equations (1c)–(1e) are defined for all cores.

Per-core voltage regulation is realized by sensing one load voltage through a sampling matrix $\mathbf{N}_k$, comparing to a reference V_{ref} and returning an error signal e_k for each core (1f). This signal is the input to the controller circuitry $\mathcal{K}_k$ that is in charge of synthesizing a duty cycle signal d_k that drives the FIVR switches through a standard Pulse Width Modulation (PWM). Each per-core controller is realized through a difference amplifier (op-amp based), whose constitutive equations can be written in state-space form (1g)-(1h). Since the duty cycle signals d_k must be limited between a valid range $0 < d_{\min} \leq d_k \leq d_{\max} < 1$, we introduce in the output equation (1h) the clipping operator $\sigma(\cdot)$ which saturates its argument to $[d_{\min}, d_{\max}]$. The second argument T_k in (1h) intends to model a delay in the PWM control of the switches. The FIVR switch models $\mathcal{S}_k$ adopted for current system-level simulation are low-frequency averaged models, which can be expressed through equivalent ideal transformers (1i)-(1j).

Finally, load variation is represented by ideal current stimuli located at the output ports of the output network (total N_o ports for each core k). Such stimuli are represented as in (1k) as ideal current sources $\boldsymbol{i}_{s;k}(t)$ for $k = 1, \dots, N_c$, with

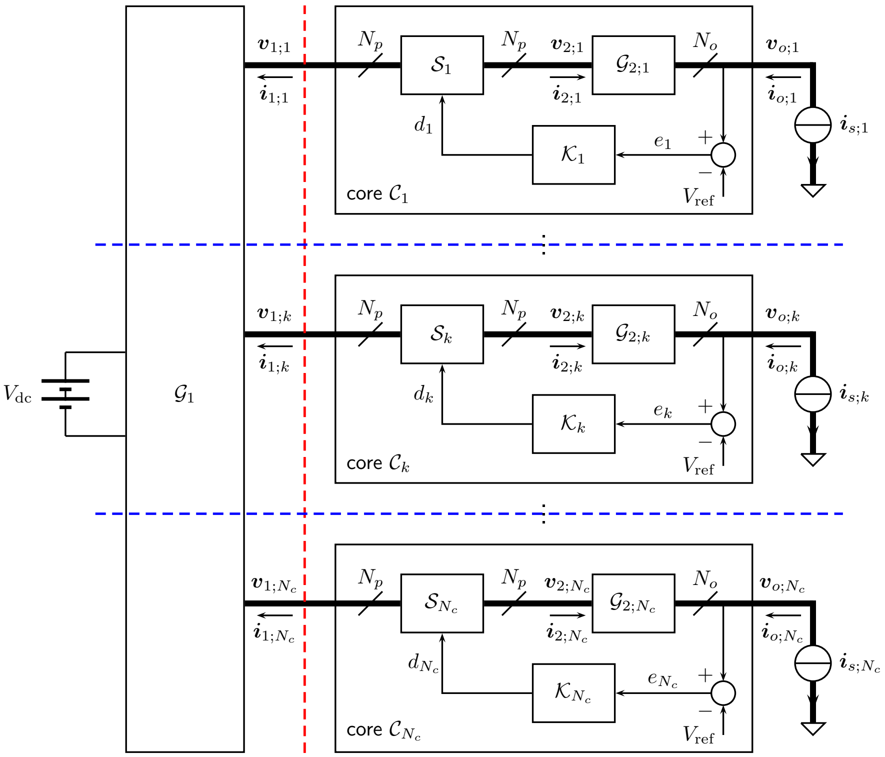


Fig. 1. Schematic description of the power delivery network topology addressed in this work, see Sec. II.

TABLE I
LIST OF RELEVANT PDN VARIABLES

Symbol	**Definition**	Size/range
k	Core index	$1 \leq k \leq N_c$
j	FIVR phase index (for each core)	$1 \leq j \leq N_p$
n	Output port index (for each core)	$1 \leq n \leq N_o$
$\boldsymbol{v}_{1;k}$	voltages at FIVR input, k-th core	N_p
$\boldsymbol{i}_{1;k}$	currents at FIVR input, k-th core	N_p
$\boldsymbol{v}_{2;k}$	voltages at FIVR output, k-th core	N_p
$\boldsymbol{i}_{2;k}$	currents at FIVR output, k-th core	N_p
$\boldsymbol{v}_{o;k}$	load voltages, k-th core	N_o
$\boldsymbol{i}_{o;k}$	load currents, k-th core	N_o
$\boldsymbol{i}_{s;k}$	source currents, k-th core	N_o
$V_{\rm ref}$	Reference voltage (same for all cores)	1
e_k	Error signal from the $k-$th core	1
d_k	Duty cycle signal, k-th core	1

components $i_{s;k,n}(t)$ for $n = 1, \ldots, N_o$. It is assumed that for $t < 0$ these current stimuli are constant

$$\boldsymbol{i}_{s;k}(t) = \boldsymbol{I}^{\rm dc}_{s;k} \quad \forall t < 0 \tag{2}$$

where $\boldsymbol{I}^{\rm dc}_{s;k}$ collects the N_o reference (nominal) load currents for core k.

A. Direct transient simulation

Given a set of current stimuli $\boldsymbol{i}_{s;k}(t)$, two reference solutions will be used to validate both accuracy and performance of proposed WR approaches. One fundamental reference will be the industry-standard HSPICE solver, as applied to a netlist description of the overall PDN structure. This is in fact the native description that is available to PI engineers from the various design teams that are responsible for the various components.

The second reference solution is obtained from a direct time discretization of (1). In order to ensure robustness and unconditional stability, we adopt the basic implicit Euler scheme with fixed time step δt and computed time samples $t_q = q\,\delta t$, so that all derivative terms in (1) are approximated with the backward difference

$$\left.\frac{dx}{dt}\right|_{t_q} \approx \frac{x(t_q) - x(t_{q-1})}{\delta t}. \tag{3}$$

Further, we exploit the PWM delay T_k in the discretization of (1i)-(1j), assuming that this delay is larger than the time step, $T_k \geq \delta t$. This implies that at any time step t_q, the nonlinearities in (1i)-(1j) become a multiplication of the variables $\boldsymbol{v}_{1;k}(t_q)$ or $\boldsymbol{i}_{2;k}(t_q)$ times a constant $d_k(t_q - T_k) \approx d_k(t_{q-Q_k})$, where $1 \leq Q_k \approx T_k/\delta t$, which is known from previous time steps. The resulting update equations become therefore explicit in the duty cycle variable d_k, and the time discretization of (1) results in a linear system to be solved to update all state variables at t_q from the previous time step t_{q-1}. Efficient update is attained by a pre-computed LU factorization of the system matrix to be inverted at each time step, which is in fact invariant. This procedure is standard, and the corresponding implementation details are omitted.

III. Waveform Relaxation for Transient Power Integrity Simulations

In this section, we present the three WR schemes that we apply to accelerate transient PDN simulation. The three schemes are based on a Longitudinal Partitioning (LP, corresponding to the partition induced by the red dashed line in Fig. II), Transverse Partitioning (TP, blue dashed line in Fig. II), and Longitudinal-Transverse Partitioning (LPTP, which is a combination of the above). These WR schemes are not new and are in fact inspired by [13], [16], [30], where they were introduced for fast Signal Integrity (SI) simulations based on partition of multiconductor transmission lines or general coupled channels from their terminations (LP) and transverse decoupling (TP). The three schemes are here modified and customized to the present PI scenario, which is characterized by a more complex topology and by a different structure of system-wise couplings.

The three WR schemes are presented in dedicated sections below. In these sections, we will refer to system equations (1) through a more compact notation aimed at representing the input-output behavior of relevant blocks. In particular, we represent the input network $\mathcal{G}_1$ through the operator

$$\{\boldsymbol{v}_{1;1},\ldots,\boldsymbol{v}_{1;N_c}\} = \mathcal{G}_1(\boldsymbol{i}_{1;1},\ldots,\boldsymbol{i}_{1;N_c}) \tag{4}$$

which corresponds to (1a)-(1b) viewed as an impedance system subject to inputs $\{\boldsymbol{i}_{1;k}, k=1,\ldots,N_c\}$ and returning the outputs $\{\boldsymbol{v}_{1;k}, k=1,\ldots,N_c\}$. The contribution of the constant source $V_{\rm dc}$ is implied. This source drives the system to the nominal operating point and will be used to initialize all signals in all algorithms and simulation results documented in this work. This term can be considered as a fixed parameter. Similarly, we will represent each individual core subsystem $\{\mathcal{C}_k,\ k=1,\ldots,N_c\}$ by collecting the contribution of all equations (1c)-(1j) pertaining to all enclosed subsystems through the operator

$$\{\boldsymbol{i}_{1;k},\boldsymbol{v}_{o;k}\} = \mathcal{C}_k(\boldsymbol{v}_{1;k},\boldsymbol{i}_{o;k}),\quad k=1,\ldots,N_c. \tag{5}$$

This operator describes the dynamics of each core subsystem $\mathcal{C}_k$ as driven by the two inputs $\boldsymbol{v}_{1;k}$, $\boldsymbol{i}_{o;k}$ and returning the two outputs $\boldsymbol{i}_{1;k}$, $\boldsymbol{v}_{o;k}$.

A. Longitudinal Partitioning

The red line in Fig. II decouples the input network $\mathcal{G}_1$ from the set of regulated core subsystems $\{\mathcal{C}_k,\ k=1,\ldots,N_c\}$. The WR-LP scheme solves each of these blocks represented by (4) and (5) separately, while setting up a fixed point iteration with index ν that uses the result of one block at iteration ν to evaluate the solution of the next block at iteration $\nu+1$. This framework corresponds to the two update equations

$$\{\boldsymbol{i}^{\nu}_{1;k},\boldsymbol{v}^{\nu}_{o;k}\} = \mathcal{C}_k(\boldsymbol{v}^{\nu}_{1;k},-\boldsymbol{i}_{s;k}),\quad k=1,\ldots,N_c \tag{6a}$$

$$\{\boldsymbol{v}^{\nu+1}_{1;1},\ldots,\boldsymbol{v}^{\nu+1}_{1;N_c}\} = \mathcal{G}_1(\boldsymbol{i}^{\nu}_{1;1},\ldots,\boldsymbol{i}^{\nu}_{1;N_c}) \tag{6b}$$

to be iterated for $\nu=1,2,\ldots$ until all signals stabilize. Initialization is performed by setting

$$\boldsymbol{v}^{\nu=1}_{1;k} = \boldsymbol{V}^{\rm dc}_{1;k},\quad k=1,\ldots,N_c \tag{7}$$

where $\boldsymbol{V}^{\rm dc}_{1;k}$ is the nominal operating point resulting from the DC solution of the system equations (1) for $t<0$, which is computed as a preprocessing step to initialize all signals, by including the input bias $V_{\rm dc}$, the reference voltages $V_{\rm ref}$, and setting all load currents to their nominal value $\boldsymbol{I}^{\rm dc}_{s;k}$ defined in (2). The circuit representation of (6) is depicted in Fig. 2, where the so-called *relaxation sources* that are used to establish equivalence of decoupled and original systems are highlighted in red color. These are represented as ideal (independent) sources since the numerical solution of individual blocks assumes that the corresponding signals are fully determined (from previous iterations).

We emphasize that the partitioning discussed herein use interface currents and voltages as relaxation variables: no attempt is carried out to optimize the relaxation process by introducing matching conditions through a suitably designed decoupling impedance, as in [31], [32]. In fact, in present PDN application, the input network $\mathcal{G}_1$ offers a very low impedance (about 1 mΩ) at its output ports (by design), whereas each core subnetwork $\mathcal{C}_k$ offers a high impedance (about 1 kΩ) at its input ports (being an interconnect loaded by a current source, as a first-order approximation). Therefore, the adopted longitudinal decoupling (Fig. 2) based on ideal voltage sources (zero-impedance) connected to the core subnetworks and current sources (zero-admittance) connected to the input network are deemed to be quasi-optimal. The numerical results of Sec. V will in fact confirm this statement.

B. Transverse Partitioning

The blue lines in Fig. II decouple the different core subsystems by including also a portion of the input network that is directly connected to the corresponding interface ports. This operation requires a slicing operation of the input network, since it is assumed that the input PDN subsystem provides a fully coupled input-output map between all interface ports. In fact, the input network provides an unwanted coupling path between different core subsystems, which share the same global PDN interconnect on board and package. Global system resonances, although damped by suitable decoupling capacitors, may potentially emphasize such inter-core coupling. A precise assessment of this coupling is in fact one of the main motivations for solving the global PDN equations (1) concurrently.

Since the input network is an LTI subsystem, we represent the corresponding response from (4) as

$$\boldsymbol{v}_{1;k} = \boldsymbol{V}^{\rm dc}_{1;k} + \sum_{k'=1}^{N_c} \mathbf{z}_{1;kk'} \star \boldsymbol{\delta i}_{1;k'},\quad k=1,\ldots,N_c \tag{8}$$

where $\mathbf{z}_{1;kk'} \in \mathbb{R}^{N_p\times N_p}$ represent blocks of the impedance impulse response matrix of the input network,

$$\boldsymbol{\delta i}_{1;k} = \boldsymbol{i}_{1;k} - \boldsymbol{I}^{\rm dc}_{1;k} \tag{9}$$

is the difference between port currents and their nominal (reference) value $\boldsymbol{I}^{\rm dc}_{1;k}$, and $\star$ denotes time-domain convolution. Note that the nominal port voltages and currents $\boldsymbol{V}^{\rm dc}_{1;k}$, $\boldsymbol{I}^{\rm dc}_{1;k}$ are known constants. Transverse partitioning is achieved by

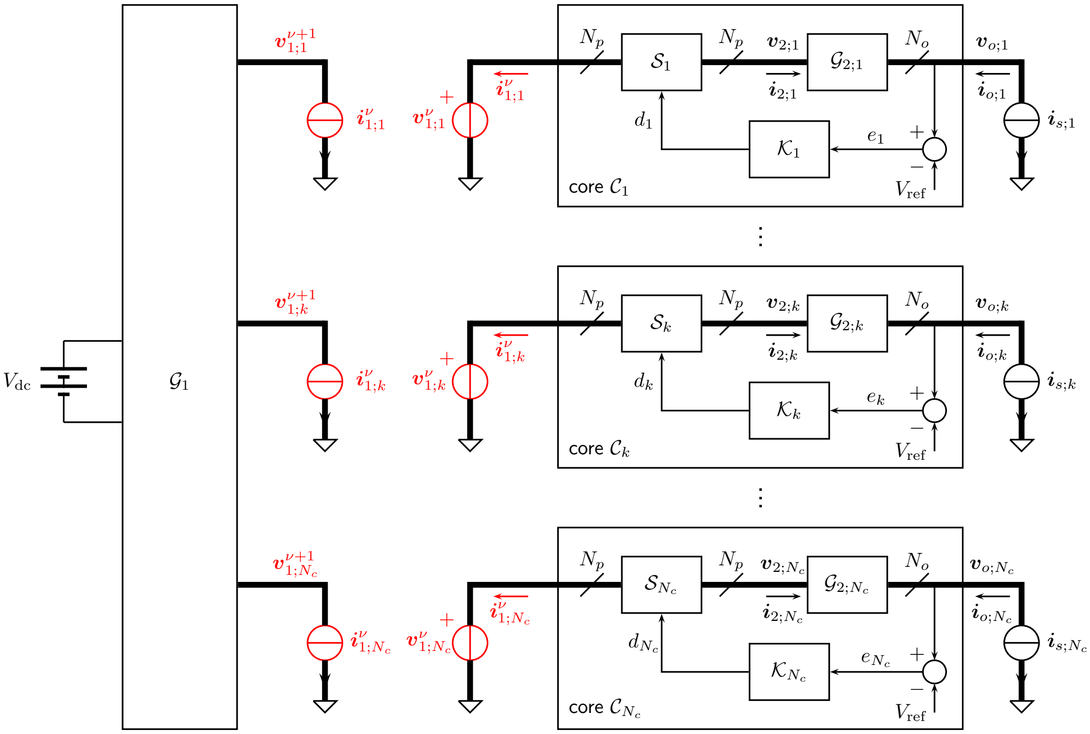


Fig. 2. Schematic illustration of WR-LP as applied to the PDN structure of Fig. II.

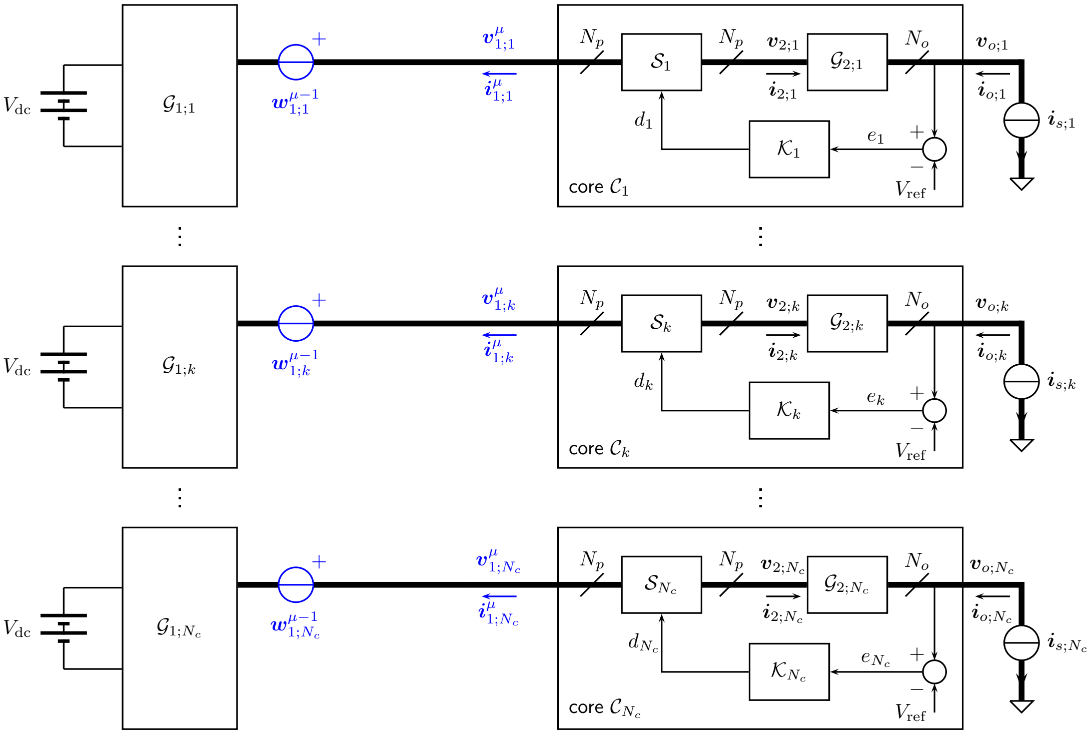


Fig. 3. Schematic illustration of WR-TP as applied to the PDN structure of Fig. II.

isolating the individual input and output signals $\boldsymbol{i}_{1;k}$, $\boldsymbol{v}_{1;k}$ at the interface between input network and k-th core subsystem, while regarding all other contributions as couplings

$$\boldsymbol{v}_{1;k} = \boldsymbol{V}^{\mathrm{dc}}_{1;k} + \mathbf{z}_{1;kk} \star \boldsymbol{\delta i}_{1;k} + \boldsymbol{w}_{1;k} \tag{10a}$$

$$\boldsymbol{w}_{1;k} = \sum_{k' \neq k} \mathbf{z}_{1;kk'} \star \boldsymbol{\delta i}_{1;k'}, \quad k = 1, \ldots, N_c \tag{10b}$$

A relaxation scheme can now be applied by introducing a transverse iteration index μ and solving (10a) for $\mu = 1, 2, \ldots$ coupled only to the associated core subsystem $\mathcal{C}_k$, assuming that the coupling sources $\boldsymbol{w}_{1;k}$ are known from previous iteration $\mu - 1$. The equations that are iteratively solved for $k = 1, \ldots, N_c$ are

$$\boldsymbol{w}^{\mu-1}_{1;k} = \sum_{k' \neq k} \mathbf{z}_{1;kk'} \star \boldsymbol{\delta i}^{\mu-1}_{1;k'} \tag{11a}$$

$$\boldsymbol{v}^{\mu}_{1;k} = \boldsymbol{V}^{\mathrm{dc}}_{1;k} + \mathbf{z}_{1;kk} \star \boldsymbol{\delta i}^{\mu}_{1;k} + \boldsymbol{w}^{\mu-1}_{1;k} \tag{11b}$$

$$\{\boldsymbol{i}^{\mu}_{1;k}, \boldsymbol{v}^{\mu}_{o;k}\} = \mathcal{C}_k(\boldsymbol{v}^{\mu}_{1;k}, -\boldsymbol{i}_{s;k}) \tag{11c}$$

where (11a) updates relaxation sources by collecting the solution known from previous iteration $\mu - 1$, and the two coupled equations (11b)-(11c) are solved independently for each k-th subsystem at each iteration μ. Figure 3 depicts a circuit interpretation of the above WR-TP scheme, where the relaxation sources $\boldsymbol{w}^{\mu-1}_{1;k}$ are highlighted in blue color.

C. Longitudinal-Transverse Partitioning

Instead of solving the coupled equations (11b)-(11c) at each WR-TP iteration μ, it is possible to set up a nested WR-LP iteration that finds the solution of this system by successive evaluations. The resulting scheme is a two-level WR-LPTP iteration with an outer TP iteration (index μ) that exploits relaxation on input PDN couplings, and an inner LP iteration with index ν that solves each decoupled input-core subsystem. More precisely, (11b)-(11c) are replaced by

$$\{\boldsymbol{i}^{\mu,\nu}_{1;k}, \boldsymbol{v}^{\mu,\nu}_{o;k}\} = \mathcal{C}_k(\boldsymbol{v}^{\mu,\nu}_{1;k}, -\boldsymbol{i}_{s;k}) \tag{12a}$$

$$\boldsymbol{v}^{\mu,\nu+1}_{1;k} = \boldsymbol{V}^{\mathrm{dc}}_{1;k} + \mathbf{z}_{1;kk} \star \boldsymbol{\delta i}^{\mu,\nu}_{1;k} + \boldsymbol{w}^{\mu-1}_{1;k} \tag{12b}$$

to be solved for $\nu = 1, 2, \ldots$ until convergence, before updating outer TP relaxation sources through (11a) and starting the next outer iteration $\mu + 1$. Figure 4 depicts the circuit interpretation of this two-level WR-LPTP scheme, with relaxation sources associated to the TP and LP iteration highlighted in blue and red color, respectively.

D. Convergence

All three WR schemes were preliminarly tested for convergence using the formulation in [33]. In particular, the iteration operators associated to LP, TP, and LPTP schemes were computed in the frequency domain by linearizing the core subsystem operators in the neighborhood of the nominal operating point. The spectral radius of all iteration operators resulted less than one, thus granting unconditional convergence for all schemes.

In WR-type schemes, convergence is often attained for earlier time instants first, so that the sequence of iterations provide an increasingly refined solution at a certain simulation time only after convergence for the preceding interval has been reached. Therefore, optimized schemes (see e.g. [34] and references therein) include *windowing*, i.e. the entire simulation time is split into several shorter sub-intervals. This optimization is not used in our examples because the time horizons considered are not long enough to warrant the use of such techniques. However, it would be advisable to combine the partitioning here presented with windowing in case of longer simulations (e.g., PI verification with real workloads).

IV. Parallelization and implementation details

A. General considerations

All three WR schemes to be demonstrated in this work will be set up to compute, for any given iteration of the LP (ν) or TP (μ) scheme, the complete set variables at all time steps $\{t_q,\, q = 1, \ldots, Q_{\max}\}$ up to the desired maximum simulation time $T_{\max} = Q_{\max}\,\delta t$. The WR scheme is thus seen as a fixed point iteration on (discretized) waveforms rather than on individual samples at a given time step (as SPICE solvers do). Approximations of the complete solution at all time steps are successively refined through WR iterations, as opposed to SPICE-based time-stepping methods which compute only a single solution estimate at each time step, before passing to the next time step.

We remark that, given that our primary objective is to parallelize as efficiently as possible the overall PI simulation, the Gauss-Jacobi (GJ) variant of WR has been preferred over the Gauss-Seidel (GS) iteration [12]. Both approaches have been demonstrated to lead to effective relaxations. The GS approach may provide a faster convergence rate, but it does not allow full parallelization since all partitions are not independent and must be solved sequentially. Therefore, this manuscript only considers GJ relaxation.

Algorithm parallelization is here performed using standard multithread implementations for shared-memory single workstations or servers, without considering Graphical Processing Units (GPUs), and with no support for networked parallel workers. For this architecture, the OpenMP paradigm is used for all parallelization tasks, whereas basic linear algebra operations are implemented via Intel MKL [35], in particular `cblas_dgemm` (matrix-matrix multiplication routine), `cblas_dgemv` (matrix-vector multiplication), `LAPACKE_dgetrf` (LU decomposition routine), `LAPACKE_dgbtrs` (back-substitution routine), each of which running in a single-thread environment. Parallelization is achieved by manual allocation of partitioned high-level matrix operations to independent computing threads, as discussed below.

B. Longitudinal Partitioning

The parallelization induced by Longitudinal Partitioning (6) and depicted in Fig. 2 is straightforward. After computing the initial operating point for all variables, which is performed outside the parallel section of the solver, the two equations (6a)-(6b) are solved iteratively. In particular,

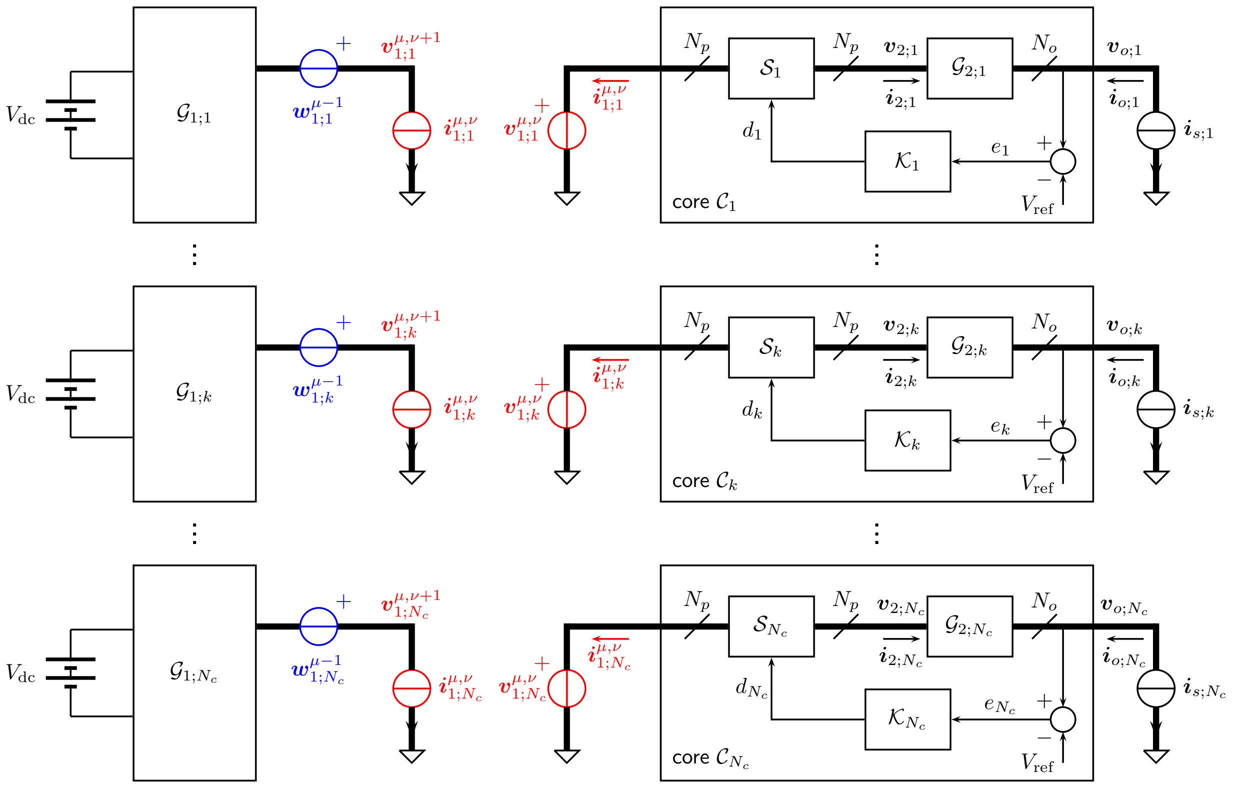


Fig. 4. Schematic illustration of WR-LPTP as applied to the PDN structure of Fig. II.

1) Separate N_c instances of (6a) are allocated to N_T computing threads and solved in parallel.
2) After completing step 1 above, interface signals $\boldsymbol{i}_{1;k}$ are collected and fed to the input model to update voltages $\boldsymbol{v}_{1;k}$ through (6b). This portion is solved in a single computing thread.

The above is repeated for successive LP iterations, as outlined by the pseudocode in Algorithm 1. The parallel sections allocated to multiple concurrent threads are marked with the symbol $\{\|\}$.

Algorithm 1 WR-LP iteration scheme

```
Find initial conditions (nominal DC solution)
Partition circuit and initialize all $\boldsymbol{v}^1_{1;k}$ waveforms to initial conditions
for $\nu = 1$ to $\nu_{\max}$ do
    for $k = 1$ to $N_c$ do
        Solve core $\mathcal{C}_k$ for interface variables $\boldsymbol{i}^\nu_{1;k}$    $\{\|\}$
    end for
    if $||\boldsymbol{v}^\nu_{o;k} - \boldsymbol{v}^{\nu-1}_{o;k}||_\infty < \epsilon$, $\forall k$ then
        Break
    end if
    Solve input model $\mathcal{G}_1$ for all $\boldsymbol{v}^{\nu+1}_{1;k}$
end for
```

The WR-LP scheme as applied to the discussed PDN structure is expected to be effective only when the input model $\mathcal{G}_1$ is characterized by low or moderate complexity with respect to core models $\mathcal{C}_k$. In fact, the main speedup resulting from parallelization is achieved by breaking the complexity of solving the output model equations (6b) in parallel, so that the ideal reduction in execution time that can be expected with N_T computing threads is

$$\rho_{LP} = \frac{\texttt{CPU}\{\mathcal{C}_k\}\lceil N_c/N_T\rceil + \texttt{CPU}\{\mathcal{G}_1\}}{\texttt{CPU}\{\mathcal{C}_k\}N_c + \texttt{CPU}\{\mathcal{G}_1\}}, \tag{13}$$

assuming that all core models $\mathcal{C}_k$ are identical. The notation $\texttt{CPU}\{\chi\}$ denotes the runtime required to evaluate model χ using a single computing thread, and operator $\lceil\cdot\rceil$ rounds its argument to the smallest larger integer. When the number of cores N_c that are modeled is an integer multiple of the number of computing threads N_T, load balancing for the parallel evaluation of the core models $\mathcal{C}_k$ is optimal since all threads complete their work concurrently. Otherwise, some threads may remain inactive by waiting for the active threads to finish their work. Under such optimal load balancing, we see that

$$\rho_{LP} \approx 1/N_T \quad \text{if} \quad \texttt{CPU}\{\mathcal{G}_1\} \ll \texttt{CPU}\{\mathcal{C}_k\} \tag{14}$$

and conversely

$$\rho_{LP} \approx 1 \quad \text{if} \quad \texttt{CPU}\{\mathcal{G}_1\} \gg \texttt{CPU}\{\mathcal{C}_k\}. \tag{15}$$

It is therefore expected that the LP scheme will be mostly effective when the input model has low complexity when compared to the output model, unless the evaluation of $\mathcal{G}_1$ can also be parallelized efficiently. This requires a specific model format that is introduced in Sec. IV-C. A fully-parallel

LP implementation that is enabled by such format is discussed in Sec. IV-D.

C. Transverse Partitioning

More care needs to be taken when parallelizing the WR-TP scheme of Fig. 3 and represented by the update equations (11). Despite the apparent block-partitioning and ideal decoupling of the impedance impulse response matrix $\mathbf{z}(t)$ in (11a)-(11b), the actual efficiency of this partitioning strongly depends on the particular state-space or descriptor realization of the input model $\mathcal{G}_1$.

Let us take a closer loop at the coupled equations (1a)-(1b), and let us assume for simplicity that $\mathbf{E}_1 = \mathbb{I}$ and $\mathbf{D}_1 = \mathbf{0}$. The blocks of the impedance impulse response matrix are available in closed-form as

$$\mathbf{z}_{1;kk'}(t) = \mathbf{C}_{1;k}\, e^{\mathbf{A}_1 t}\, \mathbf{B}_{1;k'}, \quad \forall t > 0 \tag{16}$$

This expression is to be compared to the full impedance impulse response matrix collecting all blocks, which reads

$$\mathbf{z}_1(t) = \mathbf{C}_1\, e^{\mathbf{A}_1 t}\, \mathbf{B}_1, \quad \forall t > 0 \tag{17}$$

where $\mathbf{B}_1$ stacks $\mathbf{B}_{1;k'}$ as block-columns and $\mathbf{C}_1$ stacks $\mathbf{C}_{1;k}$ as block-rows. The cost that is required for the evaluation of (16) is not significantly smaller than the cost required for (17), since both are dominated by the cost for the matrix exponential, which is identical in both cases. Note that this holds true both in the closed-form expressions (16)-(17), but also for the time-domain discretization of the corresponding equations based on the adopted implicit Euler method. The size of the matrix to be inverted at any time step is dominated by the state-space matrix $\mathbf{A}_1$, whose size is invariant even after Transverse Partitioning. Therefore, we do not expect any gain in execution speed until we reduce the complexity for the evaluation of the individual blocks $\mathbf{z}_{1;kk'}$.

Two directions will be investigated to attain this goal. One is to perform a structured model order reduction of the input network, following the procedure that is well documented in [25]. This approach will result in a smaller state-space size, with computational cost reduction both for the fully-coupled (reduced) system and for the WR-TP. Yet, the latter will not be advantageous with respect to the direct simulation of the fully-coupled (reduced) system, since the state-space matrix $\mathbf{A}_1$ will still dominate the cost.

A second direction aims at modifying the state-space realization of the input model, by enforcing a structure for which each individual (block) input $\boldsymbol{i}_{1;k}$ excites only a subset of input network states, henceforth denoted as $\boldsymbol{x}_{1;k}$, instead of the full set of states $\boldsymbol{x}_1$. In order to achieve this goal, both $\mathbf{A}_1$ and $\mathbf{B}_1$ must have a block-diagonal structure, as depicted in Fig. 5 (right panel), as opposed to a standard unstructured realization in the left panel, where the state matrix $\mathbf{A}_1$ is possibly sparse but without particular input-induced structure, and $\mathbf{B}_1$ is full. Fortunately, off-the-shelf tools are available to compute such a state-space realization. In this work we follow the standard procedure of computing frequency samples of the associated impedance matrix

$$\mathbf{Z}_1(s) = \mathbf{C}_1(s\mathbf{E}_1 - \mathbf{A}_1)^{-1}\mathbf{B}_1 + \mathbf{D}_1 \tag{18}$$

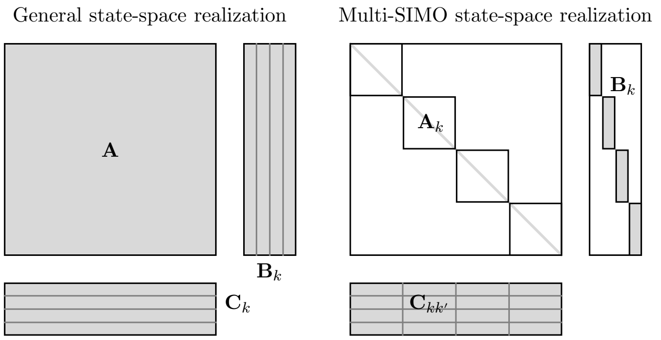


Fig. 5. Standard (unstructured) state-space realization of the input network (left) and multi-SIMO structured realization (right).

through a direct AC sweep over the bandwidth of interest, and we fit these samples with the Fast Vector Fitting (VF) algorithm [27], [28], [36], [37] with passivity enforcement [29], [38]–[40] as implemented in a commercial tool [41]. Finally, we convert the pole-residue form obtained by VF to a state-space realization in the multi-SIMO format, as detailed in [29, Chapter 8]. The result is the structured form depicted in Fig. 5 (right panel) where additionally all diagonal blocks $\mathbf{A}_{1,k}$ are diagonal (or at most with 2×2 diagonal blocks in case of complex pole pairs).

Adopting the above multi-SIMO realization allows to rewrite (1a)-(1b) as

$$\begin{aligned} \dot{\boldsymbol{x}}_{1;k'} &= \mathbf{A}_{1;k'}\boldsymbol{x}_{1;k'} + \mathbf{B}_{1;k'}\boldsymbol{i}_{1;k'} + \mathbf{B}_{\mathrm{dc};k'}V_{\mathrm{dc}} \\ \boldsymbol{v}_{1;k} &= \sum_{k'}\mathbf{C}_{1;kk'}\boldsymbol{x}_{1;k'} + \sum_{k'}\mathbf{D}_{1;kk'}\boldsymbol{i}_{1;k'} + \mathbf{D}_{\mathrm{dc};k}V_{\mathrm{dc}} \end{aligned} \tag{19}$$

so that (16) becomes

$$\mathbf{z}_{1;kk'}(t) = \mathbf{C}_{1;kk'}\, e^{\mathbf{A}_{1;k'} t}\, \mathbf{B}_{1;k'}, \quad \forall t > 0. \tag{20}$$

The latter expression requires a significantly reduced cost for its evaluation. The actual reduction depends on the size of the blocks $\mathbf{A}_{1;k}$ with respect to the size of $\mathbf{A}_1$, which ultimately depends on the number of VF poles used in the rational approximation of the input impedance model with respect to its original dynamic order. The numerical examples discussed in Sec. V show that in practical applications this reduction is quite significant.

Adopting the above multi-SIMO realization of the input impedance model, the WR-TP scheme becomes competitive since the partitioning of Fig. 3 splits the input network into decoupled submodels $\mathcal{G}_{1;k}$ represented by the diagonal blocks $\mathbf{z}_{1;kk}$, whose evaluation requires only a fraction of the overall input network states. Parallelization is then applied at each WR-TP iteration to solve all N_c decoupled blocks through totally independent N_T computing threads. The set of equations that are actually solved at WR-TP iteration μ are (1) where (1a)-(1b) are replaced $\forall k$ with (19) restated as

$$\dot{\boldsymbol{x}}^{\mu}_{1;k} = \mathbf{A}_{1;k}\boldsymbol{x}^{\mu}_{1;k} + \mathbf{B}_{1;k}\boldsymbol{i}^{\mu}_{1;k} + \mathbf{B}_{\mathrm{dc};k}V_{\mathrm{dc}} \tag{21a}$$

$$\boldsymbol{v}^{\mu}_{1;k} = \mathbf{C}_{1;kk}\boldsymbol{x}^{\mu}_{1;k} + \mathbf{D}_{1;kk}\boldsymbol{i}^{\mu}_{1;k} + \mathbf{D}_{\mathrm{dc};k}V_{\mathrm{dc}} + \boldsymbol{w}^{\mu-1}_{1;k} \tag{21b}$$

The WR-TP relaxation sources are computed after each iteration as

$$\boldsymbol{w}^{\mu}_{1;k} = \sum_{k'\neq k} \mathbf{C}_{1;kk'}\boldsymbol{x}^{\mu}_{1;k'} + \sum_{k'\neq k} \mathbf{D}_{1;kk'}\boldsymbol{i}^{\mu}_{1;k'} \tag{22}$$

in order to set up the next iteration. In our implementation, also the evaluation of the relaxation sources (22) is performed in the parallel section of the code, where the contribution of inputs and states pertaining to block k' are evaluated in a dedicated computing thread as

$$\boldsymbol{w}^{\mu}_{1;kk'} = \mathbf{C}_{1;kk'}\boldsymbol{x}^{\mu}_{1;k'} + \mathbf{D}_{1;kk'}\boldsymbol{i}^{\mu}_{1;k'} \tag{23}$$

before their accumulation in a synchronization point through

$$\boldsymbol{w}^{\mu}_{1;k} = \sum_{k'\neq k} \boldsymbol{w}^{\mu}_{1;kk'} \tag{24}$$

Based on this implementation, the CPU time reduction of the TP scheme that is attainable by using N_T parallel threads reads

$$\rho_{TP} = \frac{\texttt{CPU}\{\mathcal{C}_k\mathcal{G}_{1;k}\}\lceil N_c/N_T\rceil + \texttt{CPU}\{\mathcal{W}_1\}}{\texttt{CPU}\{\mathcal{C}_k\mathcal{G}_{1;k}\}N_c + \texttt{CPU}\{\mathcal{W}_1\}}, \tag{25}$$

where $\texttt{CPU}\{\mathcal{C}_k\mathcal{G}_{1;k}\}$ is the cost for solving the coupled input and output networks of the k-th transverse partition including (23), and $\texttt{CPU}\{\mathcal{W}_1\}$ is the cost for relaxation source accumulation (24). A pseudocode description of proposed WR-TP scheme is reported in Algorithm 2.

Algorithm 2 WR-TP iteration scheme

```
Find initial conditions (nominal DC solution)
Partition circuit and initialize all relaxation sources $\boldsymbol{w}^0_{1;k}$
  to initial conditions
for $\mu = 1$ to $\mu_{\max}$ do
    for $k = 1$ to $N_c$ do
        Solve coupled system $(\mathcal{G}_{1;k}, \mathcal{C}_k)$ for $\boldsymbol{i}^{\mu}_{1;k}$        {||}
        Update relaxation sources $\boldsymbol{w}^{\mu}_{1;k'k}$ for $k' \neq k$      {||}
    end for
    Update relaxation sources $\boldsymbol{w}^{\mu}_{1;k}$, $\forall k$ via (24)
    if $||\boldsymbol{v}^{\mu}_{o;k} - \boldsymbol{v}^{\mu-1}_{o;k}||_\infty < \epsilon$, $\forall k$ then
        Break
    end if
end for
```

D. Optimizing LP iterations

The availability of a block-diagonal input model $\mathcal{G}_1$ with multi-SIMO realization as introduced in Sec. IV-C enables a significant improvement in the WR-LP scheme. The second LP update equation in (6b) can be expressed for a multi-SIMO realization of $\mathcal{G}_1$ as

$$\dot{\boldsymbol{x}}^{\nu+1}_{1;k} = \mathbf{A}_{1;k}\boldsymbol{x}^{\nu+1}_{1;k} + \mathbf{B}_{1;k}\boldsymbol{i}^{\nu}_{1;k} + \mathbf{B}_{\mathrm{dc};k}V_{\mathrm{dc}} \tag{26a}$$

$$\boldsymbol{v}^{\nu+1}_{1;kk'} = \mathbf{C}_{1;kk'}\boldsymbol{x}^{\nu+1}_{1;k'} + \mathbf{D}_{1;kk'}\boldsymbol{i}^{\nu}_{1;k'} \tag{26b}$$

$$\boldsymbol{v}^{\nu+1}_{1;k} = \sum_{k'} \boldsymbol{v}^{\nu+1}_{1;kk'} + \mathbf{D}_{\mathrm{dc};k}V_{\mathrm{dc}} \tag{26c}$$

where

- the state update equation (26a) is performed independently on each block partition of the input model for $k = 1, \ldots, N_c$, so that individual instances $\forall k$ can be allocated to separate computing threads;
- also the state-output map (26b) leads to a set of separate contributions from each block of states $\boldsymbol{x}^{\nu+1}_{1;k'}$, so that these terms can be computed $\forall k'$ by independent computing threads;
- evaluation of the output voltages $\boldsymbol{v}^{\nu+1}_{1;k}$ through (26c) constitutes a synchronization point for all partial contribution from all block-states, remaining the only operation that needs to be performed outside of the parallel code section.

As a result, the CPU time reduction of the basic LP scheme in (13) improves for this Block-LP (BLP) scheme as

$$\rho_{BLP} = \frac{(\texttt{CPU}\{\mathcal{C}_k\} + \texttt{CPU}\{\mathcal{G}'_{1;k}\})\lceil N_c/N_T\rceil + \texttt{CPU}\{\mathcal{G}''_{1;k}\}}{\texttt{CPU}\{\mathcal{C}_k\}N_c + \texttt{CPU}\{\mathcal{G}_1\}}, \tag{27}$$

where $\texttt{CPU}\{\mathcal{G}'_{1;k}\}$ and $\texttt{CPU}\{\mathcal{G}''_{1;k}\}$ refer, respectively, to (26a)-(26b) and (26c).

E. Two-level Longitudinal- Transverse Partitioning

The parallelization of the WR-LPTP scheme combines the above WR-LP and WR-TP, as discussed in Sec. III-C and depicted in Fig. 4. Based on the multi-SIMO realization discussed in Sec. IV-C, at each nested iteration indexed by (μ, ν) each of the N_T computing thread processes the decoupled core subsystems $\mathcal{C}_k$ as in (12a) in a first pass, followed by one of the block-partitioned input subsystems $\mathcal{G}_{1;k}$ expressed by (12b) and solved through (26). Then, TP relaxation sources are collected after syncrhonization of all threads through (22), and the iterations continue.

A pseudocode description of the WR-LPTP scheme is provided in Algorithm 3, where the inner LP iterations are performed only up to a maximum number of passes denoted as m. In this implementation, we avoid waiting for LP iterations to converge to a solution that still needs to be updated through the outer TP iterations. Rather, we perform a limited number of inner LP iterations ($m = 1$ or 2), just to allow propagation of the information between the decoupled blocks. Numerical results will show that $m = 2$ is more than sufficient for achieving a very good convergence date of the overall LPTP scheme, whereas $m = 1$ may be too small and could slow down overall convergence. Correspondingly, we will label as WR-LPTP(m) the iteration scheme based on the number of inner LP iterations. The CPU time reduction of this scheme is practically identical to (27).

V. Results

The proposed parallel WR-based transient solvers are demonstrated using two benchmark PDNs, already documented in earlier works [24], [25]. The first is a small-scale test case, namely a mobile computing system equipped with a 4-cores Intel® Core™ microprocessor. The corresponding PDN includes four FIVRs with $N_p = 4$ phases each, $N_o = 36$ output ports per core and 144 output ports overall. The second example can be considered as a large-scale benchmark consisting of a PDN of an enterprise server based on an Intel® Xeon® microprocessor with $N_c = 60$ modeled cores

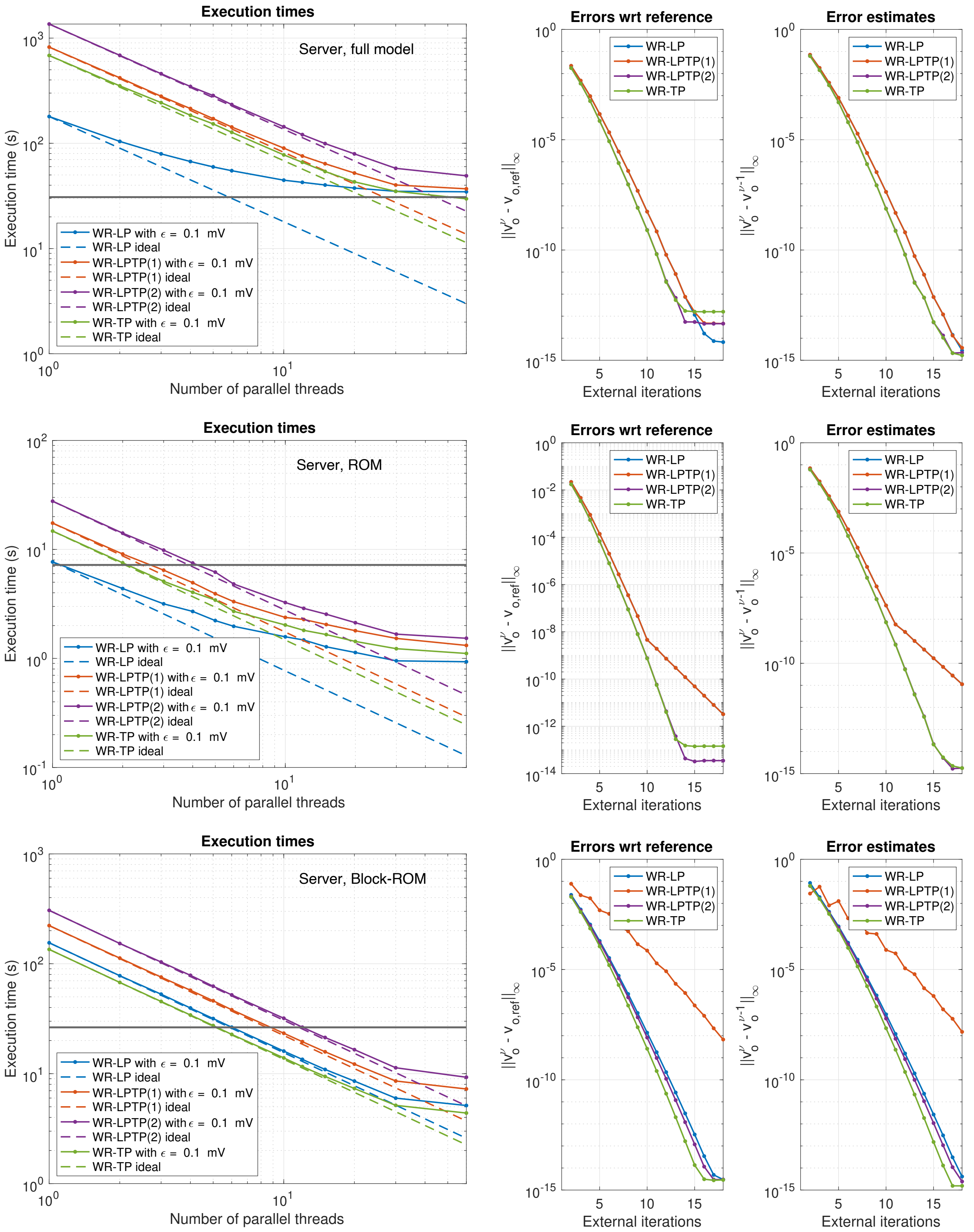


Fig. 6. Server benchmark. Left panels: scalability results for different PDN models (top: full model; middle: standard ROM; bottom: block-ROM of input model in multi-SIMO format), obtained by running proposed LP, TP and LPTP(n) Waveform Relaxation schemes on N_T computing threads. Dashed lines provide a reference ideal scaling law proportional to N_T^{-1}. Solid lines indicate the execution time with $\epsilon = 0.1\,\text{mV}$. Center and right panels: evolution of worst-case absolute error between different WR results and reference solution (obtained by direct numerical integration of (1)) through WR iterations (center panels) and error estimates obtained as the worst-case deviation with respect to previous WR iteration (right panels).

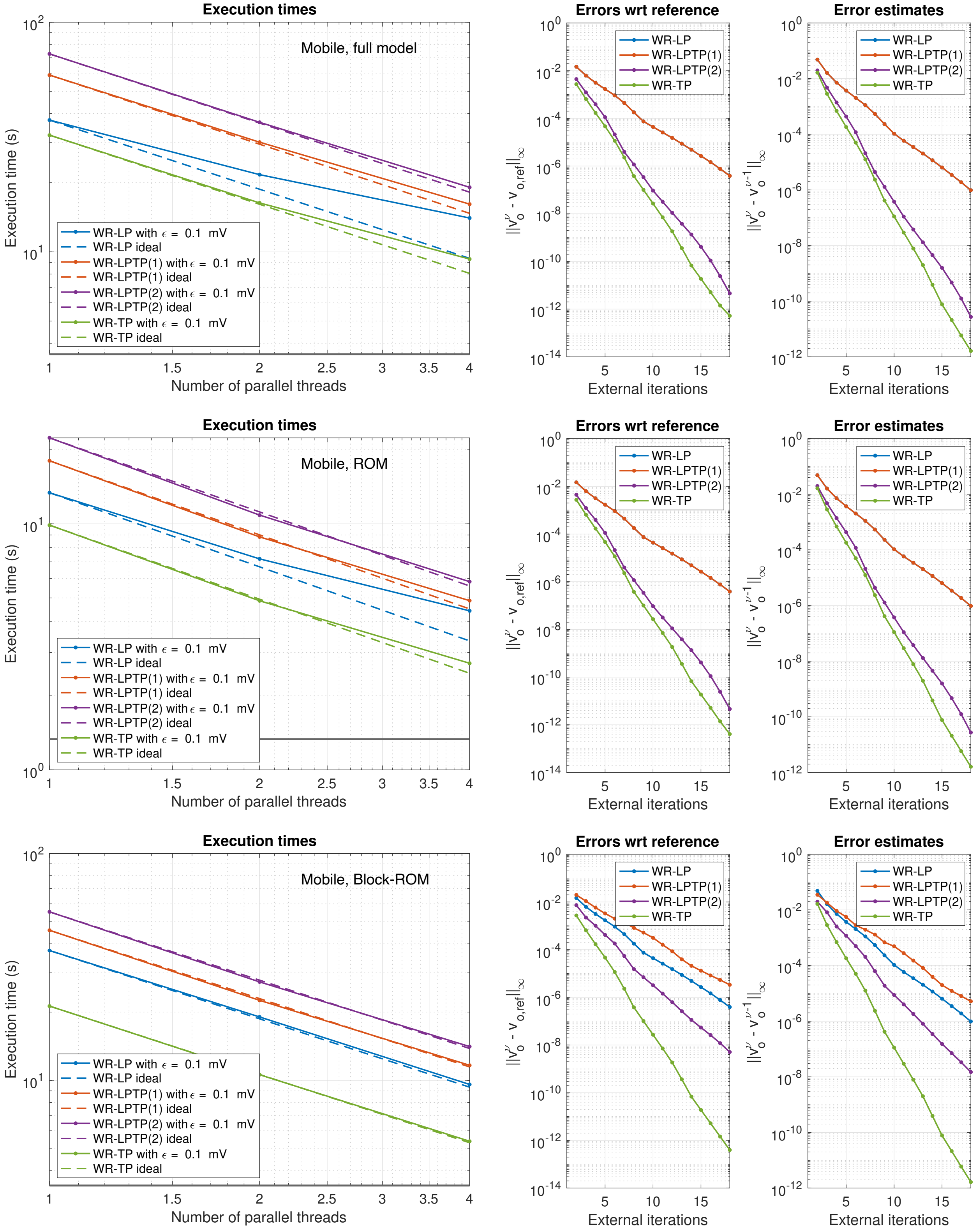


Fig. 7. As in Fig. 6, but for the mobile PDN benchmarks.

Algorithm 3 WR-LPTP(m) iteration scheme

```
Find initial conditions (nominal DC solution)
Partition circuit and initialize relaxation sources $w^0_{1;k}$ to initial condition
for $\mu = 1$ to $\mu_{\max}$ do
    for $k = 1$ to $N_C$ do
        for $\nu = 1$ to $m$ do
            Solve input model $\mathcal{G}_{1;k}$ for $v^{\mu,\nu}_{1;k}$          $\{||\}$
            Solve core $\mathcal{C}_k$ for $i^{\mu,\nu}_{1;k}$                $\{||\}$
        end for
        Update relaxation sources $w^{\mu}_{1;k'k}$ for $k' \neq k$      $\{||\}$
    end for
    Update relaxation sources $w^{\mu}_{1;k}$, $\forall k$ via (24)
    if $||v^{\mu}_{o;k} - v^{\mu-1}_{o;k}||_{\infty} < \epsilon$, $\forall k$ then
        Break
    end if
end for
```

TABLE II
OVERVIEW OF PDN BENCHMARKS

	Server platform			Mobile platform		
	Full	ROM	Block-ROM	Full	ROM	Block-ROM
N_c	60	60	60	4	4	4
N_p	3	3	3	4	4	4
N_o	57	57	57	36	36	36
N_1	6170	68	1086	450	91	357
N_2	744	3	744	420	144	420
N_{tot}	51170	608	46086	2142	679	2049

and $N_p = 3$ FIVR phases. The load ports for each core are $N_o = 57$, leading to 3420 output ports overall where voltage needs to be stabilized and monitored.

For each of the two test cases, three different models are derived and tested for both input and output networks, as detailed in Table II.

- "full" models obtained from a conversion of the native SPICE description to a Modified Nodal Analysis (MNA) form via direct stamping. Both input and output network LTI models are preprocessed and converted to a regular state-space form with $\mathbf{E}_1 = \mathbb{I}$, $\mathbf{E}_{2;k} = \mathbb{I}$, and diagonal $\mathbf{A}_1$ and $\mathbf{A}_{2;k}$.
- unstructured Reduced-Order Models (ROMs) obtained through a classical structured projection framework, as discussed in [25], [26].
- for the input network, block-diagonal multi-SIMO models as discussed in Sec. IV-C, obtained through the software [41].

For each of these models, Table II reports the associated structure and sizes, in order to enable a sound interpretation of the WR results. In this table, N_1 and N_2 denote the state-space size of input network $\mathcal{G}_1$ and cores $\mathcal{C}_k$, whereas N_{tot} is the global state-space size including also all controller states.

The main results of an extensive campaign of numerical simulations are reported in Fig. 6 for the server benchmark and in Fig. 7 for the mobile benchmark. For both examples, a sequence of current steps exciting blocks of cores at successive times were used as excitation (server: 20 A per core with 3 ns rise time; mobile: 10 A per core with 5 ns rise time), as in [24].

All numerical results have been computed using a dual-socket server equipped with two 24-core (48-thread) CPUs running at 2.65 GHz and 1024 GB RAM. This machine allowed us to run all numerical tests using an increasing number of computing threads N_T up to the maximum required to allocate a single core submodel to a single computing thread for the most complex example (the server example, with $N_T = N_c = 60$). All numerical tests were executed by selecting N_T as an integer divisor of N_c in order to provide ideal load balancing among all threads and avoid idle waiting time for some threads. This resulted in $N_T = \{1, 2, 4\}$ for the mobile benchmark and $N_T = \{1, 2, 3, 4, 5, 6, 10, 12, 15, 20, 30, 60\}$ for the server benchmark. Note that N_T is limited to 4 in the mobile benchmark because there are only four core submodels to be dispatched to different computing threads for parallel solution.

In both Fig. 6 and Fig. 7 results are reported for the three different model structures, namely the full-size models in the top rows, the standard ROMs in the middle rows, and the block-structured ROMs in the bottom rows. In each row, the CPU time required to run a full transient simulation ($Q_{\max} = 11000$ and 50000 time steps for server and mobile benchmarks, respectively) is reported in the leftmost panels as a function of N_T for the WR-LP, WR-TP and two WR-LPTP(m) executed using $m = 1$ and $m = 2$ inner LP iterations. Runtime of each scheme is compared to a reference ideal scaling law (dashed lines) obtained by dividing the runtime of a single-threaded execution by the number of threads N_T. The horizontal solid line in each panel represents the reference CPU time required for a direct solution of (1) through the implicit Euler scheme with the same time step δt. This solution provides also the reference for assessing accuracy and convergence through WR iterations, see below. The middle panels in each row report, for each of the four schemes, the worst-case error (maximum deviation among all output ports and all time steps) between the solution at the current WR iteration and the reference solution. The rightmost panels report the error estimates used to stop WR iterations when a convergence threshold ϵ is attained. Such estimates are simply derived by using as reference the solution at the previous iteration. Table III provides all runtime in seconds for all models and all schemes, including reference runtime for the direct solution of (1) using MATLAB and C implementations. As an additional reference, the runtime required by HSPICE for running the same transient simulation of the mobile benchmark was 1792 seconds, with a maximum deviation on all output voltages for all three adopted models of about 3.3 mV with respec to HSPICE. The server benchmark netlist failed to converge in HSPICE [24], [25].

A. Server benchmark

We start by analyzing the server benchmark in Fig. 6, for which we can draw the following observations. The full model (Fig. 6, top-left) results in a poor parallel efficiency for all schemes. This is due to structure of the input network model, which provides full coupling between all core inputs. Parallelization provides some speedup with a limited number of computing threads, especially for the TP and LPTP schemes, but runtime saturates to a plateau which is even larger than the serial reference runtime. We conclude that, without a dedicated preprocessing, the direct application of WR schemes to such models is impractical.

The situation improves dramatically using the standard ROM (Fig. 6, middle-left). Although not in block-diagonal form, so that parallel efficiency saturates as for the full model, the reduced number of states for both input and output network make the total runtime significantly faster than the serial time, especially for the LP scheme. The fastest runtime among all models is below one second for a massively parallel execution of this ROM with the LP scheme.

As expected, the best parallel efficiency for the server is achieved with the multi-SIMO input model structure (Fig. 6, bottom-left). Almost ideal speedup is achieved with up to $N_T = 30$ computing threads, with saturation that appears only with $N_T = 60$ due to the residual non-parallelized sections of the algorithms. Total runtime is larger than for the standard ROM, mainly due to the model sizes: the number of states of the multi-SIMO models is in fact significantly larger than for the corresponding standard ROMs.

For all server models and all algorithms, convergence speed is excellent, requiring 5-7 WR iterations to achieve a worst-case accuracy at all time steps and for all output voltages less than 0.1 mV. The only scheme that offers worst convergence properties is the WR-LPTP(1) scheme, for both standard ROM (although this is visible only below very aggressive accuracy thresholds) and especially for the block-diagonal ROM. For the latter case, one inner LP iteration is not sufficient to update relaxation sources to an accuracy level that guarantees fast overall convergence.

Figure 8 reports the evolution of the output voltage signal estimates through WR iterations, by comparing to the reference solution the results of the first three WR-LP and WR-TP iterations. We see that after three iterations the WR solutions are practically indistinguishable from the reference, as a confirmation of the fast convergence of the WR schemes for this particular application.

B. Mobile benchmark

We now analyze the mobile benchmark results in Fig. 7. For this testcase, the WR parallelization makes sense only up to a very limited number of threads $N_T = N_c = 4$. Similar observations apply as for the server testcase, with suboptimal parallel efficiency for all WR schemes as applied to the full and standard ROM models (top-left and middle-left panels). Ideal speedup is instead granted by the block-diagonal model (bottom-left), for all WR schemes. Due to the model size and the very simple benchmark, the serial direct solver still remains the best option, given the fact that all WR schemes need to repeat the simulation of the decoupled blocks over several iterations. It is therefore obvious that WR is not appropriate when the overall model complexity is low.

Also for the mobile benchmark the WR-LPTP(1) scheme provides worst convergence properties for all models (center and right panels, all rows of Fig. 7). The WR-LPTP(2) scheme instead converges very fast by gaining almost one order of magnitude in accuracy per iteration. Similar performance is provided by the WR-TP scheme. It is notable that the structure of the block-diagonal model (bottom row) causes a reduced performance in convergence for all schemes that involve LP decoupling, whereas the TP partitioning converges extremely fast. This is confirmed by all panels on the left in Fig. 7, from which we desume the best performance of TP among all tested WR implementations.

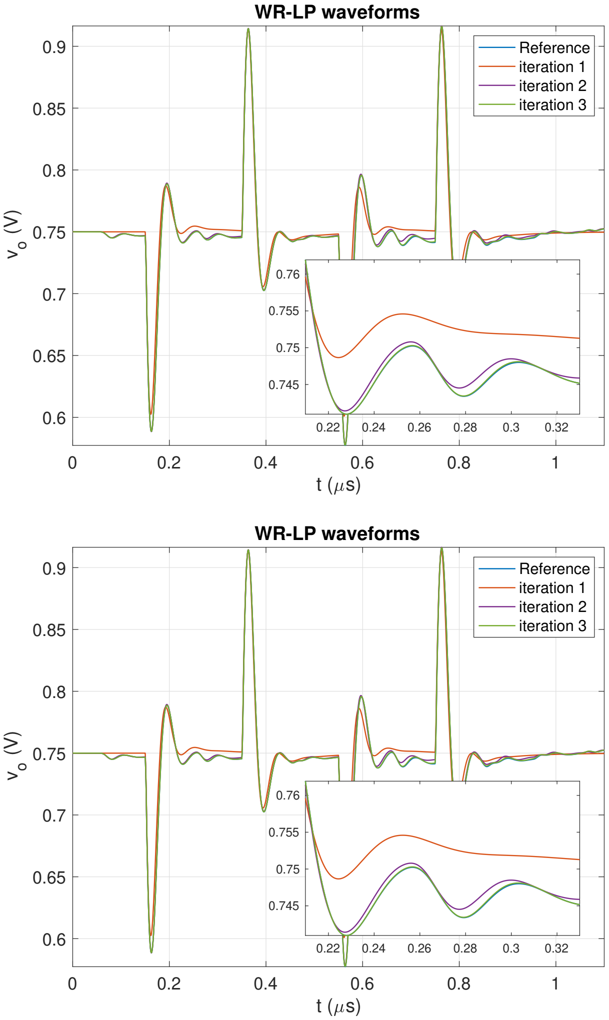


Fig. 8. Server platform: partial solution for one selected output voltage waveform at few initial LP (top panel) and TP (bottom panels) WR iterations.

VI. Conclusions

Three Waveform Relaxation schemes based on longitudinal and transverse partitioning were presented, customized, and applied to the parallel transient simulation of system-level Power Distribution Network (PDN) models of multi-core processing systems. The results obtained by applying proposed methods to PDN models of real products show that almost ideal parallel efficiency can be achieved only when the adopted interconnect models are characterized by a particular block-partitioned structure of their state-space matrices. Overall speedup with respect to reference HSPICE

TABLE III
DETAILED TIMING RESULTS (RUNTIME IN SECONDS) FOR ALL WR SCHEMES AND ALL PDN BENCHMARKS ($\epsilon = 10^{-4}$ V)

	Server platform			Mobile platform		
	Full	ROM	Block-ROM	Full	ROM	Block-ROM
matlab	760	690	120	30	10.5	26.9
C	30.8	7.20	26.4	3.58	1.33	3.44
N_T			WR-LP			
1	180	7.68	155	37.5	13.4	37.3
2	104	4.38	77.8	21.7	7.20	19.0
4	67.1	2.70	39.6	14.0	4.43	9.60
10	44.6	1.58	16.0	—	—	—
15	40.2	1.28	10.9	—	—	—
30	35.0	0.95	5.99	—	—	—
60	34.6	0.93	5.13	—	—	—
last ν	7	7	7	11	11	11
N_T			WR-TP			
1	683	14.7	135	32.2	9.88	21.2
2	351	7.54	67.6	16.3	4.87	10.6
4	184	4.05	34.2	9.30	2.71	5.40
10	77.6	2.02	13.9	—	—	—
15	54.0	1.66	9.43	—	—	—
30	35.0	1.23	5.15	—	—	—
60	29.7	1.11	3.48	—	—	—
last μ	6	6	6	6	6	6
N_T			WR-LPTP(1)			
1	821	17.4	222	58.8	18.0	45.9
2	418	9.07	112	30.0	8.85	22.5
4	214	4.95	57.4	16.1	4.87	11.7
10	90.0	2.38	23.3	—	—	—
15	64.0	2.05	15.8	—	—	—
30	40.2	1.53	8.58	—	—	—
60	36.9	1.32	7.26	—	—	—
last μ	7	7	10	11	11	13
N_T			WR-LPTP(2)			
1	1358	27.7	306	72.8	22.4	55.3
2	685	14.1	153	36.6	10.8	27.2
4	346	7.50	78.5	19.1	5.82	14.1
10	144	3.26	31.9	—	—	—
15	99.4	2.54	21.3	—	—	—
30	57.8	1.67	11.3	—	—	—
60	49.1	1.53	9.28	—	—	—
last μ	6	6	7	7	7	8

simulations exceed three orders of magnitude, thanks to an optimal combination of Model Order Reduction strategies with parallel WR simulation.

REFERENCES

[1] R. Achar and M. S. Nakhla, "Simulation of high-speed interconnects," *Proceedings of the IEEE*, vol. 89, no. 5, pp. 693–728, May 2001.

[2] K. Aygun, B. Fischer, and J. Meng, "A fast hybrid field-circuit simulator for transient analysis of microwave circuits," *IEEE Trans. Microw. Theory Tech.*, vol. 52, no. 2, pp. 573–583, 2004.

[3] H. Xie, J. Wang, R. Fan, and Y. Liu, "A hybrid FDTD-SPICE method for transmission lines excited by a nonuniform incident wave," *IEEE Trans. Electromagn. Compat.*, vol. 51, no. 3 PART 2, pp. 811–817, 2009.

[4] R. Wang and J. M. Jin, "Incorporation of multiport lumped networks into the hybrid time-domain finite-element analysis," *IEEE Trans. Microw. Theory Tech.*, vol. 57, no. 8, pp. 2030–2037, 2009.

[5] S. Safavi and J. Ekman, "A hybrid PEEC–SPICE method for time-domain simulation of mixed nonlinear circuits and electromagnetic problems," *IEEE Trans. Electromagn. Compat.*, vol. 56, no. 4, pp. 912–922, Aug. 2014.

[6] S. Grivet-Talocia, I. S. Stievano, and F. Canavero, "Hybridization of FDTD and device behavioral-modeling techniques," *IEEE Trans. Electromagnetic Compatibility*, vol. 45, no. 1, pp. 31–42, February 2003.

[7] M. Swaminathan, D. Chung, S. Grivet-Talocia, K. Bharath, V. Laddha, and J. Xie, "Designing and modeling for power integrity," *IEEE Transactions on Electromagnetic Compatibility*, vol. 52, no. 2, pp. 288–310, May 2010.

[8] K. Radhakrishnan, M. Swaminathan, and B. K. Bhattacharyya, "Power delivery for high-performance microprocessors—challenges, solutions, and future trends," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 11, no. 4, pp. 655–671, 2021.

[9] E. A. Burton, G. Schrom, F. Paillet, J. Douglas, W. J. Lambert, K. Radhakrishnan, and M. J. Hill, "FIVR — Fully integrated voltage regulators on 4th generation Intel® Core™ SoCs," in *2014 IEEE Applied Power Electronics Conference and Exposition - APEC 2014*, Mar. 2014, pp. 432–439.

[10] M. J. Gander, M. Al-Khaleel, and A. E. Ruehli, "Corrections to optimized waveform relaxation methods for longitudinal partitioning of transmission lines [aug 09 1732-1743]," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 57, no. 1, pp. 312–312, 2010.

[11] J. K. White and A. L. Sangiovanni-Vincentelli, *Relaxation techniques for the simulation of VLSI circuits*. Springer New York, NY, 1987.

[12] E. Lelarasmee, A. Ruehli, and A. 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. 1, no. 3, pp. 131–145, 1982.

[13] N. Nakhla, A. Ruehli, M. Nakhla, and R. Achar, "Simulation of coupled interconnects using waveform relaxation and transverse partitioning," *IEEE Transactions on Advanced Packaging*, vol. 29, no. 1, pp. 78–87, 2006.

[14] F.-Y. Chang, "The generalized method of characteristics for waveform relaxation analysis of lossy coupled transmission lines," *IEEE Transactions on Microwave Theory and Techniques*, vol. 37, no. 12, pp. 2028–2038, 1989.

[15] ——, "Transient simulation of nonuniform coupled lossy transmission lines characterized with frequency-dependent parameters. i. waveform relaxation analysis," *IEEE Transactions on Circuits and Systems I*, vol. 39, no. 8, pp. 585–603, 1992.

[16] V. Loggia, S. Grivet-Talocia, and H. Hu, "Transient simulation of complex high-speed channels via waveform relaxation," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 1, no. 11, pp. 1823–1838, 2011.

[17] M. J. Gander, M. Al-Khaleel, and A. E. Ruchli, "Optimized waveform relaxation methods for longitudinal partitioning of transmission lines," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 56, no. 8, pp. 1732–1743, 2009.

[18] M. Gander and A. Ruehli, "Optimized waveform relaxation methods for rc type circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 51, no. 4, pp. 755–768, 2004.

[19] M. De Stefano, T. Wendt, C. Yang, S. Grivet-Talocia, and C. Schuster, "A waveform relaxation solver for transient simulation of large-scale nonlinearly loaded shielding structures," *IEEE Transactions on Electromagnetic Compatibility*, vol. 64, no. 6, pp. 2042–2054, Dec 2022.

[20] T. Menkad and A. Dounavis, "Convergence of the resistive coupling-based waveform relaxation method for chains of identical and symmetric circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 12, pp. 5120–5133, 2021.

[21] ——, "Using strictly dissipative impedance coupling in the waveform relaxation method for the analysis of interconnect circuits," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 3, pp. 1283–1296, 2021.

[22] R. Achar, M. S. Nakhla, H. S. Dhindsa, A. R. Sridhar, D. Paul, and N. M. Nakhla, "Parallel and Scalable Transient Simulator for Power Grids via Waveform Relaxation (PTS-PWR)," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 19, no. 2, pp. 319–332, feb 2011. [Online]. Available: http://ieeexplore.ieee.org/document/5325667/

[23] A. Moglia, A. Carlucci, S. Grivet-Talocia, S. Mongrain, S. Kulasekaran, and K. Radhakrishnan, "A two-level waveform relaxation approach for system-level power delivery verification," in *2023 IEEE Electrical Design of Advanced Packaging and Systems (EDAPS)*, 2023, pp. 1–3.

[24] A. Carlucci, T. Bradde, S. Grivet-Talocia, S. Mongrain, S. Kulasekaran, and K. Radhakrishnan, "A compressed multivariate macromodeling framework for fast transient verification of system-level power delivery networks," *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 13, no. 10, pp. 1553–1566, 2023.

[25] A. Carlucci, S. Grivet-Talocia, S. Kulasekaran, and K. Radhakrishnan, "Structured model order reduction of system-level power delivery networks," *IEEE Access*, 2024, *Early access*. [Online]. Available: https://doi.org/10.1109/ACCESS.2024.3359853

[26] A. Carlucci, S. Grivet-Talocia, S. Mongrain, S. Kulasekaran, and K. Radhakrishnan, “Balancing-based model reduction for fast power integrity verification,” in *2023 IEEE 32nd Conference on Electrical Performance of Electronic Packaging and Systems (EPEPS)*, 2023, pp. 1–3.

[27] B. Gustavsen and A. Semlyen, “Rational approximation of frequency domain responses by vector fitting,” *Power Delivery, IEEE Transactions on*, vol. 14, no. 3, pp. 1052–1061, jul 1999.

[28] D. Deschrijver, M. Mrozowski, T. Dhaene, and D. De Zutter, “Macromodeling of multiport systems using a fast implementation of the vector fitting method,” *Microwave and Wireless Components Letters, IEEE*, vol. 18, no. 6, pp. 383–385, june 2008.

[29] S. Grivet-Talocia and B. Gustavsen, *Passive macromodeling: Theory and applications*. John Wiley & Sons, 2015.

[30] M. A. Farhan, N. M. Nakhla, M. S. Nakhla, and R. Achar, “Fast transient analysis of tightly coupled interconnects via overlapping partitioning and model-order reduction,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 4, no. 10, pp. 1648–1656, 2014.

[31] T. Menkad and A. Dounavis, “Using strictly dissipative impedance coupling in the waveform relaxation method for the analysis of interconnect circuits,” *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 3, pp. 1283–1296, 2021.

[32] R. Wang and O. Wing, “Analysis of vlsi multiconductor systems by bilevel waveform relaxation,” in *1990 IEEE International Conference on Computer-Aided Design. Digest of Technical Papers*, 1990, pp. 166–169.

[33] V. Loggia, S. Grivet-Talocia, and H. Hu, “Transient simulation of complex high-speed channels via waveform relaxation,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 1, no. 11, pp. 1823–1838, November 2011.

[34] H. Zhang, “A note on windowing for the waveform relaxation method,” *Applied Mathematics and Computation*, vol. 76, no. 1, pp. 49–63, 1996.

[35] “Intel®oneAPI Math Kernel Library (oneMKL).” [Online]. Available: https://www.intel.com/content/www/us/en/docs/onemkl/developer-reference-c/2023-0/overview.html

[36] S. Grivet-Talocia and M. Bandinu, “Improving the convergence of vector fitting for equivalent circuit extraction from noisy frequency responses,” *IEEE Trans. Electromagnetic Compatibility*, vol. 48, no. 1, pp. 104–120, February 2006.

[37] A. Chinea and S. Grivet-Talocia, “On the parallelization of vector fitting algorithms,” *IEEE Transactions on Components, Packaging and Manufacturing Technology*, vol. 1, no. 11, pp. 1761–1773, November 2011.

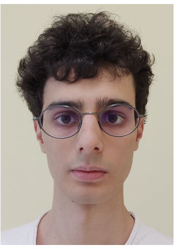
[38] S. Grivet-Talocia, “Passivity enforcement via perturbation of Hamiltonian matrices,” *IEEE Trans. Circuits and Systems I: Fundamental Theory and Applications*, vol. 51, no. 9, pp. 1755–1769, September 2004.

[39] S. Grivet-Talocia and A. Ubolli, “On the generation of large passive macromodels for complex interconnect structures,” *IEEE Trans. Advanced Packaging*, vol. 29, no. 1, pp. 39–54, February 2006.

[40] ——, “A comparative study of passivity enforcement schemes for linear lumped macromodels,” *IEEE Trans. Advanced Packaging*, vol. 31, no. 4, pp. 673–683, Nov 2008.

[41] “IdEM, Dassault Systèmes.” [Online]. Available: www.3ds.com/products-services/simulia/products/idem/

Alessandro Moglia received the B.Sc. in Electronic Engineering in 2023 from Politecnico di Torino, Turin, Italy, where he is currently pursuing a M.Sc. degree in Quantun Engineering. His research focuses on large-scale simulation of electronic systems using parallel computing and relaxation methods and is currently conducted through a research internship at the Department of Electronics and Telecommunications, Politecnico di Torino.



Antonio Carlucci (Graduate Student Member, IEEE) received the B.Sc. and M.Sc. degrees in Electronic Engineering respectively in 2019 and 2021, both from Politecnico di Torino, Turin, Italy, where he is currently pursuing a Ph.D. degree within the EMC group. His research focuses on large-scale simulation of electronic systems using macromodeling methods. He received the Best Student Paper award of SPI 2023, the 27th IEEE Workshop on Signal and Power Integrity.

Stefano Grivet-Talocia (M’98–SM’07–F’18) received the Laurea and Ph.D. degrees in electronic engineering from the Politecnico di Torino, Turin, Italy in 1994 and 1998, respectively. From 1994 to 1996, he was with the NASA/Goddard Space Flight Center, Greenbelt, MD, USA. He is currently a Full Professor of electrical engineering with the Politecnico di Torino. He co-founded the academic spinoff company IdemWorks (Turin, Italy) in 2007, serving as the President until its acquisition by CST in 2016. He has authored about 200 journal and conference papers. His current research interests include passive macromodeling of lumped and distributed interconnect structures, model-order reduction, modeling and simulation of fields, circuits, and their interaction, wavelets, time-frequency transforms, and their applications. Dr. Grivet-Talocia was a co-recipient of the 2007 Best Paper Award of the IEEE TRANSACTIONS ON ADVANCED PACKAGING. He received the IBM Shared University Research Award in 2007, 2008, and 2009 and an Intel Strategic Research Segment Grant in 2022, 2023 and 2024. He was an Associate Editor of the IEEE TRANSACTIONS ON ELECTROMAGNETIC COMPATIBILITY from 1999 to 2001 and He is currently serving as Associate Editor for the IEEE TRANSACTIONS ON COMPONENTS, PACKAGING AND MANUFACTURING TECHNOLOGY. He was the General Chair of the 20th and 21st IEEE Workshops on Signal and Power Integrity (SPI2016 and SPI2017), the co-Chair of SPI2023 and the Program co-Chair of SPI2024.

Siddharth Kulasekaran (Member, IEEE) received the B.Tech degree in Electrical Engineering from NIT, Trichy, India in 2010 and M.Sc. degree and the Ph.D. degree in electrical, electronic and communications engineering from Arizona State University, Tempe, USA in 2012 and 2017 respectively. He is currently working at Intel, Arizona as a Senior Analog Engineer at the Power Delivery Core Competency team since 2017. His areas of expertise are in integrated voltage regulators, advanced packaging and passives technologies. At Intel, he focuses on developing modelling methodologies and measurement metrologies for characterizing power delivery networks. His work involves closing the gap between modeling and measurements through accurate 3D EM extraction and loss estimation.

Kaladhar Radhakrishnan (Senior Member, IEEE) is an Intel Fellow and a Power Delivery Architect with the Technology Development group at Intel. He has played a significant role in shaping and driving power delivery technologies for Intel microprocessors. His areas of expertise are in integrated voltage regulators, advanced packaging and passives technologies. Kaladhar is a two-time recipient of the Intel Achievement Award. He has authored four book chapters, over 50 technical papers in peer reviewed journals, and has been awarded 40 US patents. Kaladhar joined Intel in 2000 after he received his Ph.D. in Electrical Engineering from the University of Illinois at Urbana-Champaign.