2010 IREP Symposium - Bulk Power System Dynamics and Control - VIII (IREP), August 1-6, 2010, Buzios, RJ, Brazil

Localization and latency concepts applied to time simulation of large power systems

Davide Fabozzi Thierry Van Cutsem*
Dept. of Electrical Engineering and Computer Science (Montefiore Institute),
University of Liège, Sart Tilman B37, B-4000 Liège, Belgium.
* Fund for Scientific Research - FNRS.

Abstract

This paper reports on investigations to speed up time-domain simulations of large electric power systems. First, a decomposed scheme is considered which exploits the bordered block diagonal structure of the original Jacobian involved in the Newton iterations, and keeps the resulting "local" Jacobians constant over multiple iterations. Next, a localization technique is considered, allowing to perform less iterations on the system components with lower activity. Finally, a third scheme consists of visiting only a subset of components, identified as active, and skipping the other ones, identified as latent. The first two techniques solve the whole set of equations with the required accuracy, while the third one involves an adjustable degree of approximation. The methods are illustrated on a small system, while preliminary checks of computational savings from a large test system are reported. Additional results deal with the application of the localization and latency techniques to simplified simulation of the detailed model.

Introduction

Time-domain simulation of power system long-term dynamics involves the solution of large sparse systems of nonlinear stiff differential-algebraic equations. The simulation of large interconnections with tens of thousands of variables requires a significant computational effort, which may be prohibitive for dynamic security assessment (where numerous contingencies have to be handled in real-time) or for dispatch training simulator (where simulation must be faster than real-time). This is even more true since the above applications require long-term simulations, typically ten minutes of the system post-disturbance evolution.

To tackle with this complexity, a number of practical approaches have been proposed and/or are used by industry to simulate the above models in an approximate but much faster way. Let us quote non exhaustively:

- *Model simplification*. Based on engineering knowledge, simplified models are set up to to study a particular class of power system dynamics. A typical example is the quasi steady-state approximation of long-term dynamics [1, 2]. This time-scale decomposition approach consists of replacing the short-term dynamics by a compacted equilibrium (i.e. algebraic) in the simulation of long-term dynamics.
- *Simplified simulation of the detailed model*. When modelling simplification is not easy (for instance because the above mentioned time-scale separation is not clear [2]) it may be attractive to simulate the detailed model but with a simplified solver, for instance a stiff-decay method with large time steps to filter out the fast dynamics [3]. This approach, however, cannot detect oscillatory instability of the fast dynamics.
- *Equivalents*. When the portion of interest of a given system is known beforehand, remote parts can be substituted by some equivalent representation. This common practice by Transmission System Operators (TSOs) suffers from two drawbacks: (i) the partitioning of the system into studied and equivalenced parts has to be decided a priori and may not be valid for any disturbance; (ii) the equivalent has to be updated with the system topology, operating point, etc. which places burden on TSOs when each of them is in charge of updating the equivalent of its system.

To achieve faster simulations parallel processing attracted many researchers. Indeed, power system time-domain simulation presents some characteristics that may make it suitable for parallelization *in space* and *in time* [4]. Parallelization in space takes advantage of the power system structure in which components such as generators, loads, compensators, etc. interact with each other through the network, but are not connected directly to each other. Parallelization in time allows computing in parallel the evolution of sub-systems over windows of time. Parallelization in space and in time have been exploited to various degrees in the waveform relaxation method [5], in the methods described in [6, 7], and in the multi-rate method [8]. All these approaches involve some form of relaxation which, on one hand, allows full parallelization but, on the other hand, may require to perform several iterations over the same window of time, until convergence of solution is reached.

Another track of research exploited the fact that most of

978-1-4244-7467-7/10/$26.00

disturbances give rise to rather localized effects, i.e. some parts of the system are nearly unaffected. At least, they do not respond most of the time during the simulation. *Localization concepts* have been used in static security analysis [9] and in the already mentioned multi-rate method [8].

They have been used successfully in the simulation of VLSI circuits [10]. In VLSI simulation, the system components which are nearly unaffected by a disturbance are called *latent*, and the objective is to skip useless numerical operations relative to a hopefully large proportion of latent blocks. To this purpose, the network is torn into one main circuit and many other blocks in order to obtain a *bordered block diagonal* structure favorable to latency exploitation. The sub-networks who have already converged are not solved, while some others may converge faster and be skipped in the last iterations. Special attention is paid to minimizing the links between the main circuit and the blocks and between the various blocks. Latency concept is apparent in digital circuits, while application to analog circuits requires some insight into the underlying dynamics.

The objective of this paper is to present investigations being carried out by the authors to exploit the bordered block diagonal structure of the Jacobian involved in Newton method, the localized nature of the system response and the latent behavior of components little affected by the simulated disturbance. A small system is used for illustration purposes, while preliminary checks of computational savings from a large test system are reported. Additional results deal with the application of the localization and latency techniques to simplified simulation of the detailed model.

Notation. Lowercase bold letters indicate column vectors. Uppercase bold letters refer to matrices. ^{T} denotes transposition.

Deriving and solving power system dynamic models

We view the power system as a set of n *injectors* connected through network. The injector term is to be understood in a wide sense, since it relates to components that can either produce or consume active power.

We work under the phasor (or quasi-sinusoidal regime) assumption [11]. We refer all phasors to orthogonal axes denoted x and y, rotating at a conveniently chosen speed [12]. Let N be the number of buses. We denote by V_{xi} and V_{yi} the rectangular components of the complex voltage at the i-th bus ($i = 1, \ldots, N$), and we group all components into:

$$\mathbf{V} = [V_{x1} V_{y1} \ldots V_{xN} V_{yN}]^T \tag{1}$$

Injector model

The i-th injector ($i = 1, \ldots, n$) is described by n_i differential and/or algebraic equations according to:

$$\mathbf{\Gamma}_i \dot{\mathbf{x}}_i = \boldsymbol{\phi}_i(\mathbf{x}_i, V_{xj}, V_{yj}) \tag{2}$$

in which $\mathbf{x}_i$ is the state vector of the injector, V_{xj} and V_{yj} are the rectangular components of its terminal voltage, and $\mathbf{\Gamma}_i$ is a diagonal matrix of dimension n_i with:
$(\mathbf{\Gamma}_i)_{\ell\ell} = 0$ if the ℓ-th equation (2) is algebraic, and
$(\mathbf{\Gamma}_i)_{\ell\ell} = 1$ if the ℓ-th equation (2) is differential.

Without loss of generality, we assume that the first two components of $\mathbf{x}_i$ are I_{xi} and I_{yi}, the rectangular components of the complex current injected into the network:

$$\mathbf{x}_i = [I_{xi}\ I_{yi}\ \ldots]^T$$

Network model

The network equations are obtained from the bus admittance matrix, replacing voltage and current phasors in terms of their rectangular components. The resulting equations take on the form:

$$\mathbf{D}\,\mathbf{V} - \sum_{i=1}^{n} \mathbf{C}_i \mathbf{x}_i = \mathbf{0} \tag{3}$$

where the sparse and structurally symmetric $(2N \times 2N)$ matrix $\mathbf{D}$ involves conductances and susceptances, and the $(2N \times n_i)$ matrix $\mathbf{C}_i$ aims at extracting the I_{xi} and I_{yi} components of $\mathbf{x}_i$ and adding them to the appropriate current mismatch equation. Assuming that the i-th injector is connected to the j-th bus ($j = 1, \ldots, N$):

$$\mathbf{C}_i = [\mathbf{e}_{2j-1}\ \mathbf{e}_{2j}\ \mathbf{0} \ldots \mathbf{0}\ ]^T \tag{4}$$

where $\mathbf{e}_{2j-1}$ denotes a unit vector of dimension n_i with the unit component in position $2j-1$ and similarly for $\mathbf{e}_{2j}$.

Static load models do not involve $\mathbf{x}$ states and can be embedded in (3) without resorting to injectors. The resulting network equations are written in compact form as:

$$\mathbf{g}(\mathbf{x}_1, \ldots, \mathbf{x}_n, \mathbf{V}) = \mathbf{0} \tag{5}$$

Algebraization

The differential equations in (2) are algebraized using one of the well-known integration formulae. For instance, the

results presented in this paper have been obtained with Backward Differentiation Formulae (BDF), known to have the stiff decay property [13], allowing the step size to be increased [14, 3]. The BDF of order o applied to the i-th injector is:

$$\mathbf{x}_i(t_j) = \sum_{\ell=1}^{o} \gamma_\ell \, \mathbf{x}_i(t_{j-\ell}) + h \, \beta \, \dot{\mathbf{x}}(t_j) \tag{6}$$

where t_j is the discrete time and h the time step size. The above formula is easily rewritten as:

$$\frac{1}{h \, \beta} \sum_{\ell=1}^{o} \gamma_\ell \, \mathbf{x}_i(t_{j-\ell}) + \dot{\mathbf{x}}(t_j) - \frac{1}{h \, \beta} \mathbf{x}_i(t_j) = \mathbf{0}$$

and extended to incorporate the algebraic equations:

$$\begin{aligned} & \frac{1}{h \, \beta} \mathbf{\Gamma}_i \sum_{\ell=1}^{o} \gamma_\ell \, \mathbf{x}_i(t_{j-\ell}) \\ + \quad & \boldsymbol{\phi}_i(\mathbf{x}_i(t_j), V_{xj}, V_{yj}) - \frac{1}{h \, \beta} \mathbf{\Gamma}_i \mathbf{x}_i(t_j) = \mathbf{0} \end{aligned}$$

This last equation is written in compact form as:

$$\mathbf{f}_i(\mathbf{x}_i, \mathbf{V}) = \mathbf{0} \tag{7}$$

where the dependency on time (t_j) has been omitted for simplicity.

Standard Newton scheme

At a given time step the Newton method is used to solve equations (5) and (7) with respect to $\mathbf{V}$ and the various state vectors $\mathbf{x}_i$. This requires solving a sequence of linear systems ($k = 1, 2, \ldots$):

$$\mathbf{J} \begin{bmatrix} \Delta\mathbf{x}_1 \\ \Delta\mathbf{x}_2 \\ \vdots \\ \Delta\mathbf{x}_n \\ \Delta\mathbf{V} \end{bmatrix} = - \begin{bmatrix} \mathbf{f}_1(\mathbf{x}_1^{k-1}, \mathbf{V}^{k-1}) \\ \mathbf{f}_2(\mathbf{x}_2^{k-1}, \mathbf{V}^{k-1}) \\ \vdots \\ \mathbf{f}_n(\mathbf{x}_n^{k-1}, \mathbf{V}^{k-1}) \\ \mathbf{g}(\mathbf{x}_1^{k-1}, \ldots, \mathbf{x}_n^{k-1}, \mathbf{V}^{k-1}) \end{bmatrix} \tag{8}$$

with

$$\mathbf{J} = \begin{bmatrix} \mathbf{A}_1 & & & & \mathbf{B}_1 \\ & \mathbf{A}_2 & & & \mathbf{B}_2 \\ & & \ddots & & \vdots \\ & & & \mathbf{A}_n & \mathbf{B}_n \\ -\mathbf{C}_1 & -\mathbf{C}_2 & \ldots & -\mathbf{C}_n & \mathbf{D} \end{bmatrix} \tag{9}$$

and incrementing the variables according to:

$$\begin{aligned} \mathbf{x}_i^k &= \mathbf{x}_i^{k-1} + \Delta\mathbf{x}_i \qquad i = 1, \ldots, n \\ \mathbf{V}^k &= \mathbf{V}^{k-1} + \Delta\mathbf{V} \end{aligned}$$

where $\mathbf{A}_i$ is the Jacobian of $\mathbf{f}_i$ with respect to $\mathbf{x}_i$, $\mathbf{B}_i$ the Jacobian of $\mathbf{f}_i$ with respect to $\mathbf{V}$ and the empty entries in (9) are zero sub-matrices. Because the i-th injector involves only the V_{xj} and V_{yj} components of $\mathbf{V}$, $\mathbf{B}_i$ is also a very sparse matrix with two nonzero columns only.

Furthermore, we use the so-called *Dishonest Newton scheme* [15] in which the Jacobian $\mathbf{J}$ is kept constant (in factorized form) over several iterations and possibly several time steps. The Jacobian is updated (and factorized) when convergence deteriorates, which is detected through:

- too large the number of iterations at a given time step
- insufficiently decreasing components in the right-hand side of (8).

When applied to very large systems, the above standard Newton scheme suffers from three drawbacks:

- when any injector undergoes a change in its equations, for instance due to a switching or a limit on a state variable, $\mathbf{A}_i$ and $\mathbf{B}_i$ change and the whole Jacobian $\mathbf{J}$ has to be updated and factorized;
- the same holds true when a few injectors (or the network) undergo large changes and trigger an update of the whole Jacobian $\mathbf{J}$ as indicated above;
- the fact that many injectors having "little activity" yield very small components in the right hand side of (8) is not exploited.

To deal with the above issues, it is appropriate to exploit the bordered block diagonal structure of $\mathbf{J}$, easily seen in (9), and decompose the system of equations (8,9) over the injectors and the network, respectively. This is detailed in the next section.

Introducing decomposition and localization in Newton iterations

Newton method: the decomposed scheme

The decomposition of concern can be seen as a Gaussian elimination with pivoting on the block of equations and unknowns relative to injectors.

One easily obtains from (8, 9):

$$\mathbf{A}_i \Delta\mathbf{x}_i = -\mathbf{f}_i(\mathbf{x}_i^{k-1}, \mathbf{V}^{k-1}) - \mathbf{B}_i \Delta\mathbf{V} \tag{10}$$

Assuming that $\mathbf{A}_i$ is nonsingular, $\Delta\mathbf{x}_i$ can be obtained from this equation and replaced into the last row of (8),

which yields:

$$\sum_{i=1}^{n} \mathbf{C}_i \mathbf{A}_i^{-1} \left(\mathbf{f}_i(\mathbf{x}_i^{k-1}, \mathbf{V}^{k-1}) + \mathbf{B}_i \Delta \mathbf{V} \right) + \mathbf{D} \Delta \mathbf{V} = \mathbf{g}(\mathbf{x}_1^{k-1}, \ldots, \mathbf{x}_n^{k-1}, \mathbf{V}^{k-1}) \quad (11)$$

Reorganizing the terms in (11) yields:

$$\tilde{\mathbf{D}} \Delta \mathbf{V} = \mathbf{g}(\mathbf{x}_1^{k-1}, \ldots, \mathbf{V}^{k-1}) - \sum_{i=1}^{n} \mathbf{C}_i \mathbf{A}_i^{-1} \mathbf{f}_i(\mathbf{x}_i^{k-1}, \mathbf{V}^{k-1}) \quad (12)$$

where:

$$\tilde{\mathbf{D}} = \mathbf{D} + \sum_{i=1}^{n} \mathbf{C}_i \mathbf{A}_i^{-1} \mathbf{B}_i \quad (13)$$

At this point, it is convenient to define:

$$\tilde{\mathbf{C}}_i = \mathbf{C}_i \mathbf{A}_i^{-1} \quad (14)$$

and rewrite equations (12) and (13) respectively as:

$$\tilde{\mathbf{D}}\, \Delta \mathbf{V} = \mathbf{g}(\mathbf{x}_1^{k-1}, \ldots, \mathbf{V}^{k-1}) - \sum_{i=1}^{n} \tilde{\mathbf{C}}_i \mathbf{f}_i(\mathbf{x}_i^{k-1}, \mathbf{V}^{k-1}) \quad (15)$$

$$\tilde{\mathbf{D}} = \mathbf{D} + \sum_{i=1}^{n} \tilde{\mathbf{C}}_i \mathbf{B}_i \quad (16)$$

The correction $\tilde{\mathbf{C}}_i \mathbf{B}_i$ brought by the i-th injector involves a (2×2) sub-matrix centered on the diagonal of $\mathbf{D}$. Hence $\tilde{\mathbf{D}}$ inherits the structural symmetry of $\mathbf{D}$.

Assuming that the $\mathbf{A}_i$ and $\tilde{\mathbf{D}}$ matrices are available in factorized form, and the corresponding $\tilde{\mathbf{C}}_i$ matrices have been computed, the procedure to solve one Newton iteration in a decomposed manner consists of:

1. solving (15) with respect to $\Delta \mathbf{V}$ to obtain $\mathbf{V}^k$
2. solving (10) with respect to $\Delta \mathbf{x}_i$ to obtain $\mathbf{x}_i^k$ for each injector.

Let us emphasize that these two steps are mathematically equivalent to solving the original system (8).

A variant consists of replacing the computation of the right-hand side of (10) by a single $\mathbf{f}_i$ evaluation. Indeed:

$$\begin{aligned} & \mathbf{f}_i(\mathbf{x}_i^{k-1}, \mathbf{V}^{k-1}) + \mathbf{B}_i \Delta \mathbf{V} \\ = \; & \mathbf{f}_i(\mathbf{x}_i^{k-1}, \mathbf{V}^{k-1}) + \frac{\partial \mathbf{f}_i}{\partial \mathbf{V}} (\mathbf{V}^k - \mathbf{V}^{k-1}) \\ \simeq \; & \mathbf{f}_i(\mathbf{x}_i^{k-1}, \mathbf{V}^k) \end{aligned} \quad (17)$$

where the last but one expression has been considered as a Taylor series expansion of the last one, limited to the first order. In principle, the latter approximation will result in $\Delta \mathbf{x}_i$ values a bit different from the ones obtained by solving the original system (8). On the other hand, one may expect that evaluating $\mathbf{f}_i$ with the newly updated $\mathbf{V}^k$ as in (17) yields an even more satisfactory iterative scheme.

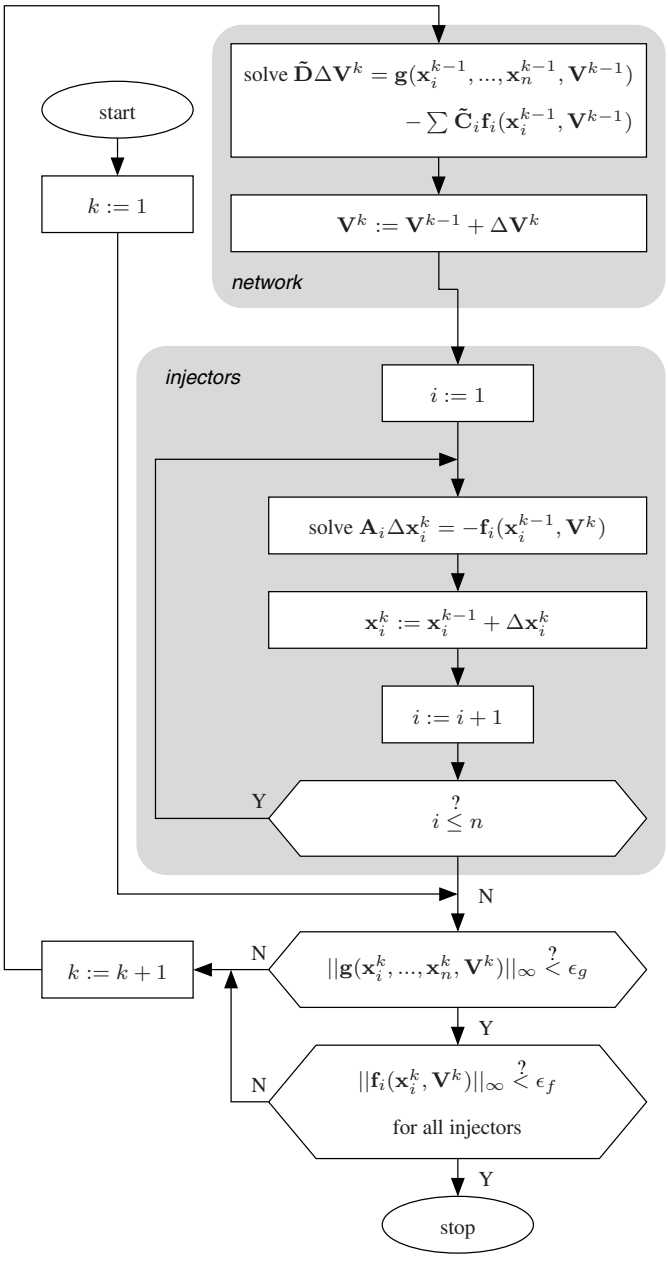

Figure 1: Decomposed Newton scheme

A flowchart of the decomposed Newton scheme, using the above variant, is given in Fig. 1. The criterion to stop iterations is on the largest component of the right-hand side vector in (8), i.e.

$$||\mathbf{f}_i(\mathbf{x}_i^k, \mathbf{V}^k)||_\infty < \epsilon_f \qquad i = 1, \ldots, n \quad (18)$$

$$||\mathbf{g}(\mathbf{x}_1^k, \ldots, \mathbf{x}_n^k, \mathbf{V}^k)||_\infty < \epsilon_g \tag{19}$$

We finally consider the computation of $\tilde{\mathbf{C}}_i$. Equation (14) can be rewritten as:

$$\mathbf{A}_i^T \, \tilde{\mathbf{C}}_i^T = \mathbf{C}_i^T$$

Since $\mathbf{C}_i^T$ has only two nonzero columns, so has $\tilde{\mathbf{C}}_i^T$. Those columns are the solutions of:

$$\mathbf{A}_i^T \mathbf{z} = \mathbf{e}_1 \qquad \mathbf{A}_i^T \mathbf{z} = \mathbf{e}_2$$

where $\mathbf{e}_1$ (resp. $\mathbf{e}_2$) is a unit vector of dimension n_i with the unit component in first (resp. second) position.

Newton method: the localized scheme

The smaller the activity of an injector, the smaller the components of the right-hand side vector in (10) (or its variant (17)) and the faster all components of this vector satisfy the convergence criterion (18).

The idea behind the localized scheme is to avoid updating the state vector of an injector when condition (18) is satisfied. Thus, if a given time step requires K Newton iterations (i.e. K executions of the outer loop in Fig. 1), for the injectors with little activity, Eq. (10) will be solved once, at most, while injectors with the highest activity will have (10) solved K times. If the system is large, significant savings in computing times are expected from the skipped operations.

A similar treatment is applied to the network: if condition (19) is satisfied, Eq. (15) is not solved. Note that no attempt is made to apply localization to the network itself. Indeed, this would require restructuring matrix $\tilde{\mathbf{D}}$ during the simulation, while it is very convenient to perform optimal ordering of this matrix once for all before the simulation starts.

The flowchart of the localized scheme is shown in Fig. 2, which should be compared to Fig. 1.

Another important aspect of localization has to do with the Jacobian updates. According to our experience, when applying the dishonest scheme, it is possible to update the $\tilde{\mathbf{D}}$ and $\mathbf{A}_i$ matrices independently of each other. More precisely:

- when an injector calls for a Jacobian update (owing to a change in its equations or a degradation of convergence), its matrix $\mathbf{A}_i$ is updated. So is the $\tilde{\mathbf{C}}_i$ matrix, subsequently used in the right-hand side of (15). However, the $\tilde{\mathbf{D}}$ matrix is not updated;

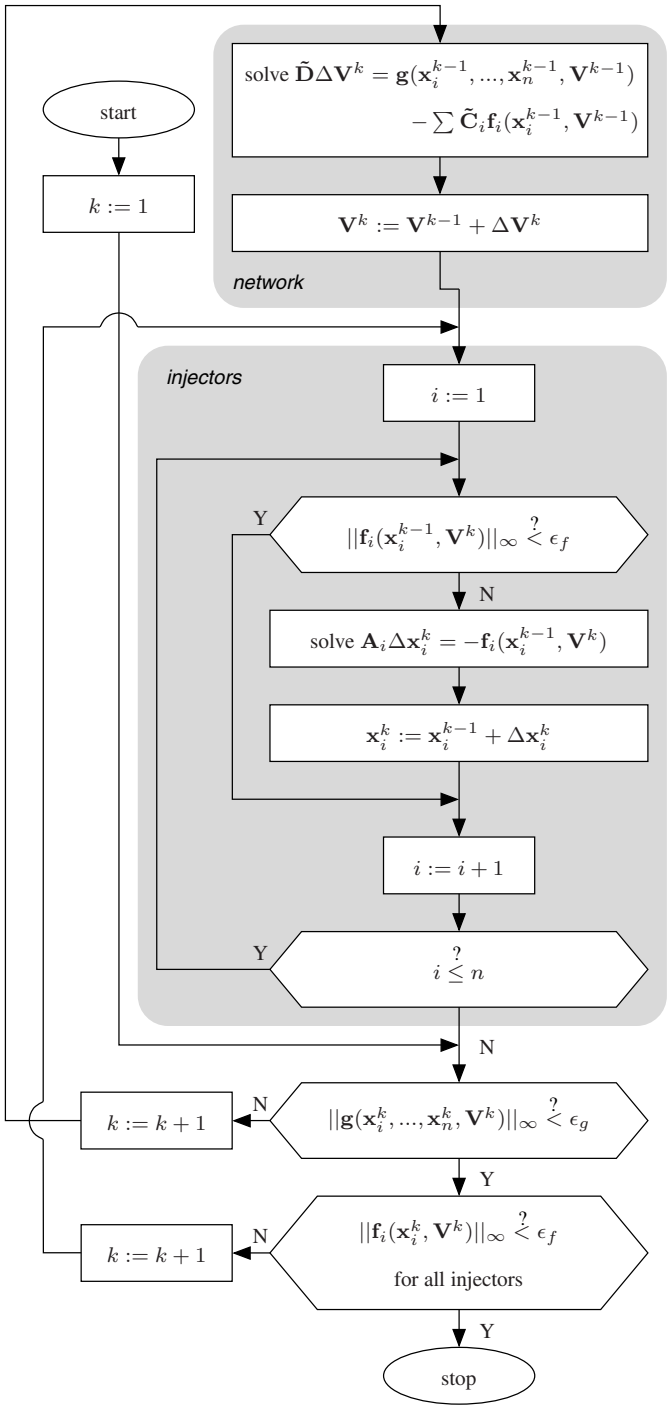

Figure 2: Localized Newton scheme

- when the network calls for a Jacobian update (for similar reasons), the $\mathbf{B}_i$ matrices of all injectors are updated and $\tilde{\mathbf{D}}$ is computed from (16) using the available $\tilde{\mathbf{C}}_i$ matrices. However, neither the $\mathbf{A}_i$ nor the $\tilde{\mathbf{C}}_i$ matrices are updated.

Exploiting latency in the localized Newton scheme

The localized Newton scheme solves the whole set of equations with the required accuracy. Hence, the system responses it provides are indiscernable from those obtained with the standard Newton scheme. On the contrary, the approach considered in this section involves some degree of approximation.

The idea behind *latency exploitation* is to visit only a subset of injectors, identified as *active*, and skip the other ones, identified as *latent*. For the active injectors, Eqs. (7) are solved with the same accuracy as for a detailed simulation. Furthermore, we keep on applying the localized scheme, which results in Eqs. (10) being solved more times for some active injectors than for others. On the other hand, for the latent blocks, Eqs. (10) are not solved; in fact, their functions $\mathbf{f}_i$ are even not computed. Note that the whole network is still solved, as in the localized approach, using matrix $\tilde{\mathbf{D}}$.

Several aspects related to the implementation of such as scheme are discussed hereafter.

We found appropriate to replace a latent synchronous machine (and its regulators) by a Thévenin equivalent whose internal voltage phasor rotates at the system angular frequency determined from the center of inertia [12] and to replace a latent (dynamic) load by an equivalent shunt admittance. In both cases, the equivalent is computed so that it exchanges with the network the same active and reactive powers as the injector it replaces, at the time the latter switches from active to latent. Both equivalents have (intentionally) no internal state and do not modify the structure of the $\tilde{\mathbf{D}}$ matrix.

Which injectors are latent and which ones are active cannot be decided *a priori* because it depends on the simulated disturbance. Furthermore, it may happen that a block changes from active to latent after some transients have died out, then switch back to active under the effect of the system evolving. So far, the following procedure was found the most satisfactory:

1. an injector switches from active to latent if both its active and reactive powers have varied by less than some amount Δ over the last τ seconds;
2. an injector switches back from latent to active if either its active or its reactive power have varied by more than Δ over the last τ seconds.

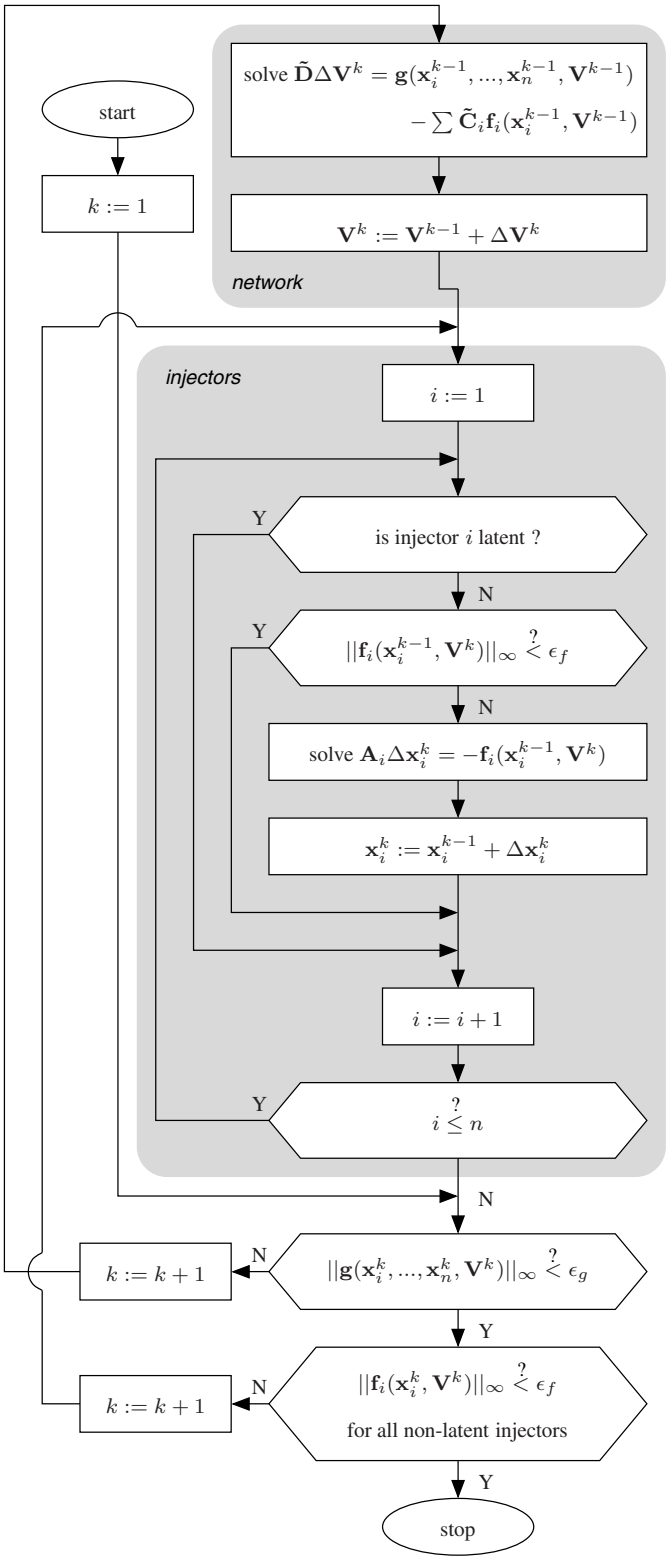

Figure 3: Localized Newton scheme with latency exploitation

Note that in case 1 the powers are computed from the detailed injector model while in case 2 they are computed from its equivalent. In the latter case, detecting a change of powers is equivalent to detecting a change of the ter-

minal voltage. Additional safeguards are put to prevent switching from active to latent:

- blocks with internal dynamics evolving but not impacting the active and reactive powers
- blocks whose active or reactive powers have significantly varied with respect to their initial values.

The flowchart of the localized scheme with latency exploitation is shown in Fig. 3, which should be compared to Fig. 2.

On parallelization and computational efficiency

The power system models used in time-domain simulation have some features that allow a certain degree of parallelization of the involved computations. Although this topic is outside the scope of this paper, it is of interest to comment on the above algorithms from the perspective of distributing tasks over multiple processors.

It is clear that the treatment of injectors can be distributed over a set of processors since Eq. (10), or its variant (17), is solved for each injector independently of the others. This is easily seen in the flowcharts of Figs. 1, 2 and 3, where the computations included in the shaded block labelled "injectors" can be performed in parallel.

However, this parallel processing is followed by a sequential step of udpating the network voltages according to (15). Clearly, the network, which couples all system components, does not allow a complete algorithm parallelization. To deal with this bottleneck, several approaches in the past resorted to relaxation, as outlined in the Introduction [4]. In this work, it was chosen not to do so, in order to preserve the convergence properties of a direct Newton method and avoid performing several solutions over the same time window. Instead, the focus of this paper is on the gain given by decomposition, localization and latency by themselves.

Whether the algorithm is implemented in a sequential or a parallel way, attention must be paid to solving the sparse system (15) very efficiently. To this purpose, the structural symmetry of matrix $\tilde{\mathbf{D}}$ must be exploited. Excellent sparse solvers are available to this purpose. Furthermore, in the detailed model of a very large power system, one can expect the network equations (5) to make up a modest subset of the whole set of equations, which contributes to reducing the above mentioned bottleneck effect.

Illustrative examples on a small test system

In this section the performance of the above algorithms are illustrated on a small system. Needless to say that computing times are not of concern here. On the other hand, in spite of its small size, the system already allows visualizing some localization effects.

The following abbreviations are used to refer to the algorithms presented in the previous sections:

- Decomposed Newton scheme (see Fig. 1): Dec.
- Localized Newton scheme (see Fig. 2): Loc.
- Localized Newton scheme with Latency Exploitation (see Fig. 3): LLE

Test system

We consider a variant of the Nordic32 test system originally proposed in [16] and previously used in [17, 18]. The system includes 74 buses and 20 synchronous machines. Its one-line diagram is shown in Fig. 4. The model includes simple representations of speed governors (for generators in the North and Equiv. areas), hydro or steam turbines, Automatic Voltage Regulators (AVRs) and OverExcitation Limiters (OELs).

The 22 loads are connected to the low-voltage side of transformers, and their voltages are controlled by Load Tap Changers (LTCs) reacting with delays in the range [30 39] s for the first tap change, and [8 12] s for subsequent changes. Each load behaves instantaneously as constant admittance. Then its active power quickly restores (with a time constant of 0.05 s) to constant current, while the reactive power remains constant admittance.

Algorithm settings

The BDF of order 2 (corresponding to $\gamma_1 = 4/3, \gamma_2 = -1/3, \beta = 2/3$ in (6)), was used with a time step h of 0.05 s in all simulations. However, BDF of order 1 (or Backward Euler method, corresponding to $\gamma_1 = 1$ and $\beta = 1$ in (6)) was used for the very first step (with $h = 0.05$ s) and for the first step after the initiating disturbance (with $h = 0.001$ s).

For latency exploitation:

- the maximum deviation Δ has been set to $0.001 \times S_{nom}$, where S_{nom} is the nominal apparent power of

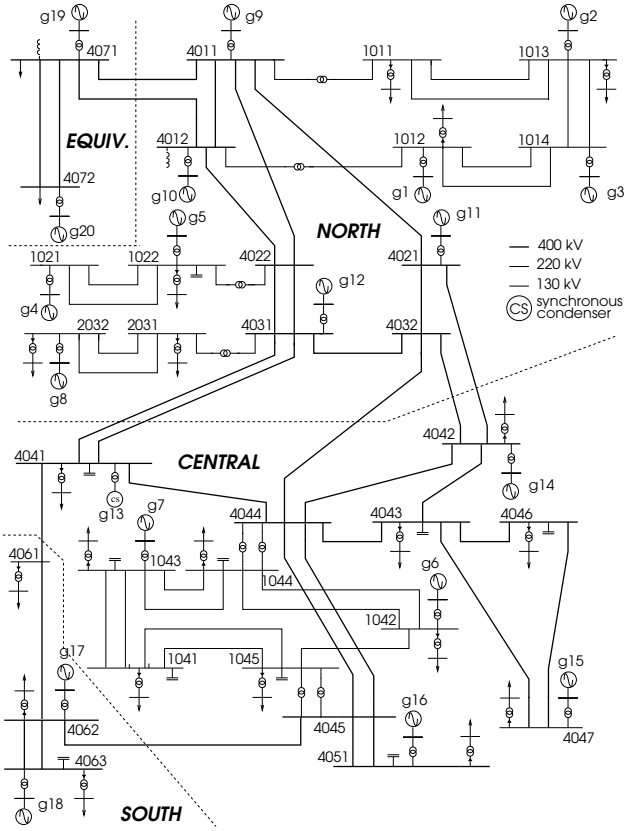

Figure 4: One-line diagram of Nordic32 test system

the generator, or is given a default value of 100 MVA for a load;

- the observation period τ has been set to 5 s.

Case N1

We consider the response to a mild contingency not causing LTCs nor OELs to act. Thus the system regains a new operating point in the short term, after electromechanical oscillations have damped out.

At $t = 1\ s$ one circuit of the double-circuit line 1013-1014 is tripped. Figure 5 shows the current flowing in the remaining circuit. Note that the outputs provided by the three schemes are indiscernable in this scale.

Figure 6 shows the number of linear systems of the type (10) solved at each time step up to convergence.

The Dec. simulation, after the transients caused by the line tripping, shows a constant value of 42 systems solved per step, which corresponds to one Newton iteration for each of the 20 machines and 22 dynamic loads.

The Loc. algorithm runs with a few iterations less than

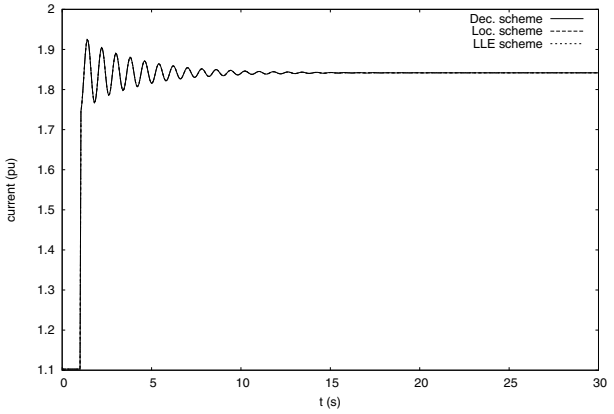

Figure 5: Case N1: current flowing in one circuit of line 1013-1014

the Dec. algorithm for the first 13 s, which indicates that already in this time interval the equations corresponding to some injectors are satisfied without performing any iteration. Then, from $t \simeq 13$ s on, there are more injectors which do not need any iteration until the system reaches steady-state at $t \simeq 17\ s$.

In the same figure, it can be seen that the LLE algorithm generally requires solving less systems of the type (10) than its Loc. counterpart, with the exception of a few peaks (the highest at $t \simeq 7$ s) and reaches steady state a little earlier. Most likely, the peaks are due to the poor quality of the Jacobian matrices, more precisely the choice of not updating any of them when an injector switches from latent to active or conversely. It should be noted though that, despite those peaks, the LLE scheme requires less computational effort than the others.

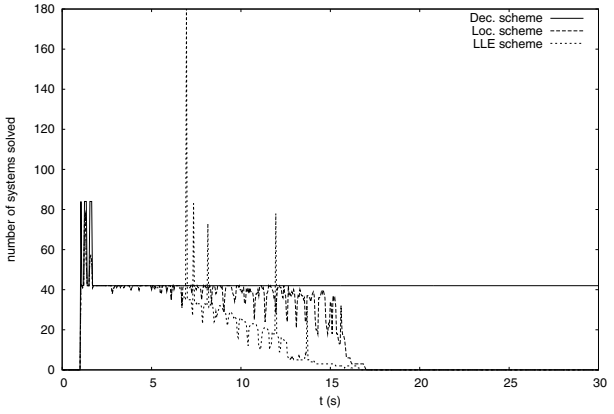

Figure 6: Case N1: number of linear systems of the type (10) solved at each time step

Figure 7 shows the field voltage of generator g15 located "far" from the tripped line (note the narrow range of values on the ordinate). The output obtained with the Dec. scheme exhibits oscillations until the end of the display interval, while with the Loc. scheme the oscillation dies out

at $t \simeq 16\ s$. This trend of the Loc. scheme to damp the last, small oscillations has been observed in other cases as well; it is attributed to the fact that progressively more and more injectors equations are satisfied up to the desired tolerance. Hence, their variables $\mathbf{x}_i$ are no longer updated, and they do no longer contribute to the oscillation. In the Dec. simulation, on the other hand, all equations are solved (and the corresponding variables $\mathbf{x}_i$ are updated) until all of them are satisfied. This is why the solid line in Fig. 6 shows that 42 systems (10) are solved up to the end of the simulation.

The same figure shows that the output of the LLE scheme stops being updated at $t \simeq 12.5$ s when the machine of concern is replaced by its Thévenin equivalent and its variables (among which the field voltage) frozen. The difference between the Loc. and LLE curves confirms the expected gain in computational efficiency brought when allowing some degree of approximation. This is of course a trade-off between accuracy and efficiency.

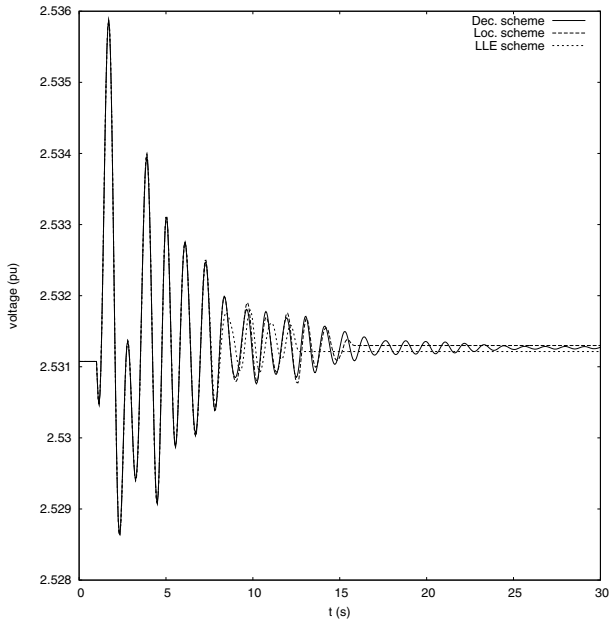

Figure 7: Case N1: field voltage of generator g15

Case N2

In this second case, longer-term dynamics is involved in the form of LTCs responding to the voltage drop caused by the initial disturbance, which is the tripping of line 4043-4047 at $t = 1$ s.

Figure 8 shows the current flowing in the neighboring line 4046-4047. Again, the outputs provided by the three schemes are indiscernable in this scale.

Figure 9 shows the voltage at the distribution bus fed by

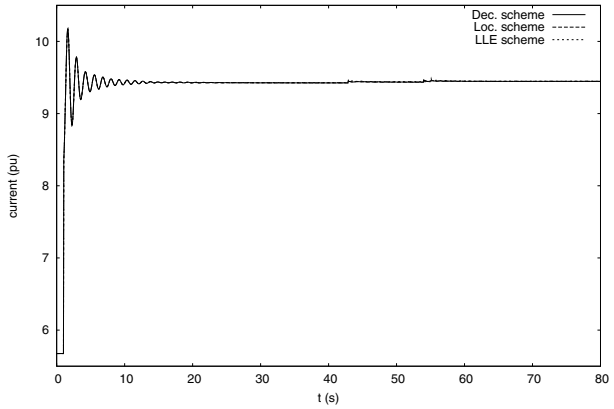

Figure 8: Case N2: current flowing in line 4046-4047

bus 4043. The ratio of the corresponding transformer is modified by the LTC at $t \simeq 43\ s$. Prior to this discrete event, all transients had already died out and, hence, all injectors were latent. The next, opposite change in voltage takes place at $t \simeq 55\ s$ when the tap of the transformer connected to the neighboring bus 4046 also moves by one step.

While the Dec. and Loc. output are almost indiscernable in this scale, the LLE output shows minor differences, the most evident being the $1\ s$ delay of the second LTC move, which is quite acceptable in practice.

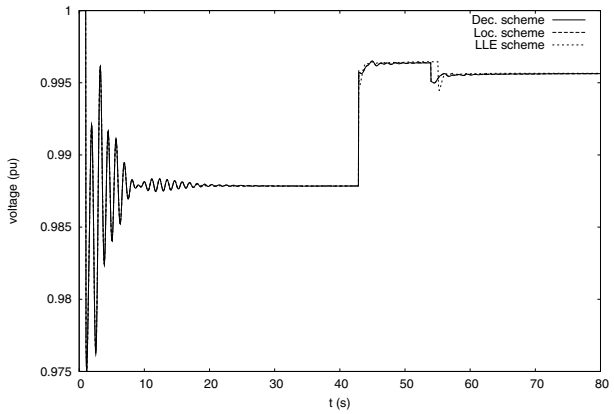

Figure 9: Case N2: voltage at distribution bus fed from bus 4043

Figures 10 and 11 aim at illustrating how the rules involving active and reactive powers made injectors switch from active to latent and back. They show respectively the active and reactive powers produced by generator g15, very close to the disturbance location. In both plots the rectangles have the width τ and the height Δ. The crosses indicate when the machine of concern became latent while the circles indicate when it became active again. For instance, at $t \simeq 43$ s and $t \simeq 55$ s, the Thévenin equivalent which replaces the latent synchronous machine (and its controllers) responds to the tap changes with a variation of reactive

power going out of the (τ, Δ) envelop. This causes the generator to switch back from latent to active.

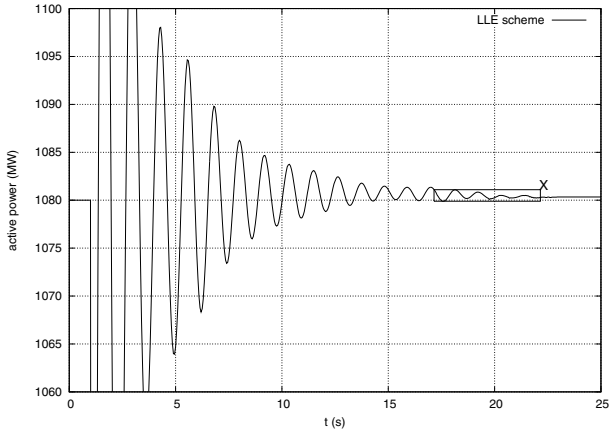

Figure 10: Case N2: active power produced by generator g15

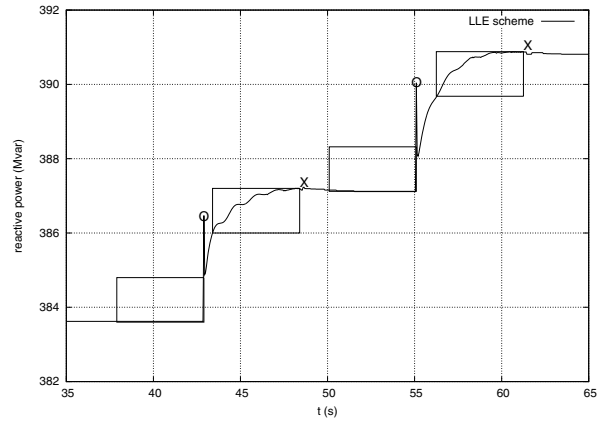

Figure 11: Case N2: reactive power produced by generator g15

Figure 12 shows the field voltage of the same generator. A zoom on the part within the dotted rectangle is provided in Fig. 13. Two minor discrepancies can be observed between the LLE and Dec. schemes: a more pronounced response after each LTC move and the already mentioned time delay on the second move. Intuitively, the stronger response is due to the fact that not all injectors switch to active, and those which do so have to somewhat compensate for those which do not. The delay of the second LTC move is more difficult to trace, but such small differences are to be expected from a scheme that solves only a subset of equations. Clearly, there is a trade-off between accuracy and efficiency, which can be adjusted through the choice of Δ and τ.

Results from a large test system

This section reports on results obtained with a large test system set up in the context of the FP7 European project PEGASE [19] and loosely inspired of the European transmission grid. The emphasis will be on computational effort.

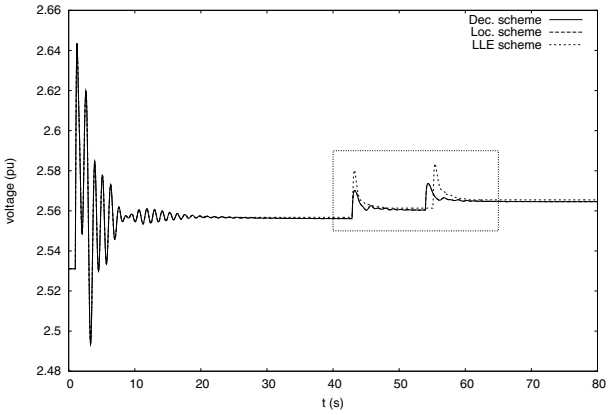

Figure 12: Case N2: field voltage of generator g15

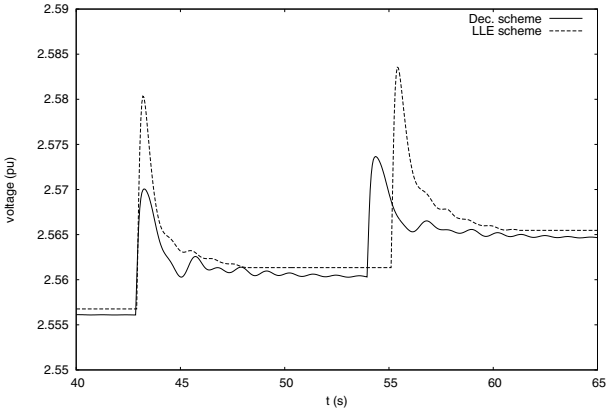

Figure 13: Zoom on the part inside the dotted rectangle in Fig. 12

System characteristics and simulation settings

The main feature of the PEGASE test system are:

- $15,350$ buses
- $3,824$ synchronous machines with generic models of AVRs, speed governors and turbines
- $2,168$ dynamic loads. Some of them include an equivalent of the distribution transformer and medium-voltage feeder.
- $2,918$ LTCs represented as discrete devices

This leads to a model with:

- $55,569$ differential variables
- $58,781$ algebraic variables, out of which $30,700$ are V_x, V_y voltage components
- a total of 114,350 states

which yields an average of 17 states per power plant and 7 states per dynamic load.

Unless otherwise stated, the simulation settings were the same as for the Nordic32 system. The simulations were performed until the system returned to full steady state.

Case P1

We simulate the outage of a double-circuit line over 100 seconds. Between $t \simeq 24\ s$ and $t \simeq 44\ s$, a few LTCs act to restore voltages.

Table 1 shows the various computational efforts. Note the last column (labeled "S-LLE") will be used in a later section. Comparing the figures of the Loc. and LLE schemes to those of the Dec. method clearly shows that the largest savings are at the injector level (factorizing $\mathbf{A}_i$, solving Eq. (17) and evaluating $\mathbf{f}_i$) as well as on the factorization of $\tilde{\mathbf{D}}$. On the other hand, there is some increase in the number of times Eq. (15) is solved and function $\mathbf{g}$ is evaluated.

Table 1: Case P1: computational effort comparison

nb. of times	Dec.	Loc.	LLE	S-LLE
$\tilde{\mathbf{D}}$ factorized	64	8	21	11
(15) solved	2587	2255	2788	143
$\mathbf{g}$ evaluated	7187	7390	9419	544
$\mathbf{A}_i$ factorized	377.0 10^3	20.8 10^3	17.5 10^3	22.0 10^3
(17) solved	15.5 10^6	8.0 10^6	1.6 10^6	0.2 10^6
$\mathbf{f}_i$ evaluated	43.1 10^6	49.5 10^6	8.5 10^6	0.7 10^6

Case P2

We simulate the outage of another double-circuit line over 320 s. The case is more severe, and between $t \simeq 21$ s and $t \simeq 279$ s several LTCs move repeatedly to finally restore the controlled voltages within their deadbands.

The results in Table 2 confirm that the Loc. and LLE schemes offer the largest savings at the injector level, while some saving is also obtained at the network level, especially in terms of number of $\tilde{\mathbf{D}}$ factorizations.

Table 2: Case P2: computational effort comparison

nb. of times	Dec.	Loc.	LLE	S-LLE
$\tilde{\mathbf{D}}$ factorized	119	7	23	10
(15) solved	8338	3272	3154	868
$\mathbf{g}$ evaluated	23.1 10^3	17.7 10^3	18.2 10^3	2.7 10^3
$\mathbf{A}_i$ factorized	707.0 10^3	20.7 10^3	19.9 10^3	25.1 10^3
(17) solved	50.0 10^6	17.3 10^6	2.3 10^6	0.2 10^6
$\mathbf{f}_i$ evaluated	138.0 10^6	134.0 10^6	10.7 10^6	0.8 10^6

The larger savings obtained in Case P2 can be explained as follows. A significant part of the computational effort is spent in the first transients, where the Loc. and LLE schemes performances are comparable to those of the Dec. scheme, especially if the event has widespread effects. Hence, the gain induced by the Loc. and LLE schemes is smaller in short-lasting cases than in long-lasting ones. Precisely, the system was simulated over a longer period in Case P2 than in Case P1. Furthermore, the Loc. and LLE schemes take better advantage from events with localized effects such as LTC adjustments, which take place in greater number in Case P2.

Figures 14 and 15 show the number of linear systems of the type (10) solved at each time step, up to convergence (note the logarithmic scale used to display the wide range of values).

After the initial transients caused by the line tripping, the Dec. scheme solves $5,922$, $11,984$ or occasionally $17,976$ linear systems (10), which corresponds to respectively 1, 2 or 3 Newton iterations per time step[1].

In Fig.14, during the first transients, the Loc. scheme shows a little lower effort compared to the Dec. scheme, which indicates that already in this period the equations relative to some injectors are satisfied without performing any iteration. Then, from $t \simeq 100$ s on, the number of injectors not requiring an update of their states increases more significantly until the system reaches steady state at $t \simeq 280$ s.

A comparison of Figs. 14 and 15 shows the much lower number of injector systems solved in the LLE scheme. Notice the increase in computational effort around $t = 60$ s, when most of the LTCs start acting (nevertheless the number of visited injectors remains lower than with the other schemes). A last tap movement at $t \simeq 290$ s causes the system to "wake up" from its apparent steady state and 44 injectors to oscillate for some more time. Those oscillations, caused by the approximations in the LLE scheme, are of tiny magnitude and involve very few injectors (more than 99 % of them are latent).

Application of LLE to simplified simulation

Since the LLE scheme basically exploits the block bordered structure of the Jacobian involved in Newton iterations, it applies equally well to the simplified simulation technique mentioned in the introduction and presented in [3]. In this paper, this simplified simulation is obtained by resorting to a BDF of order 2 with large steps to filter out

[1] 3,824 generators + 2,168 dynamic loads = 5922 injectors; 5922 injectors $\times$ 2 iterations = 11,984 systems to solve, etc.

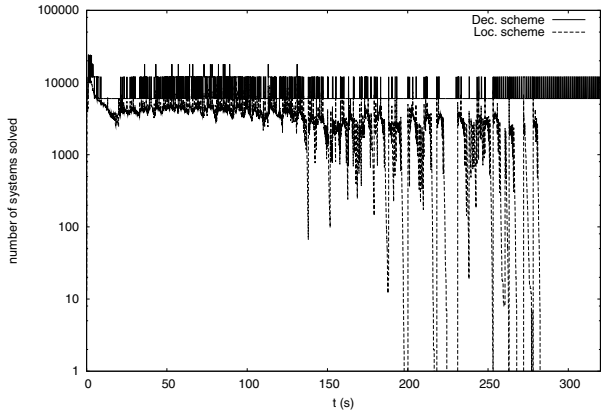

Figure 14: Case P2: number of injector Newton iterations

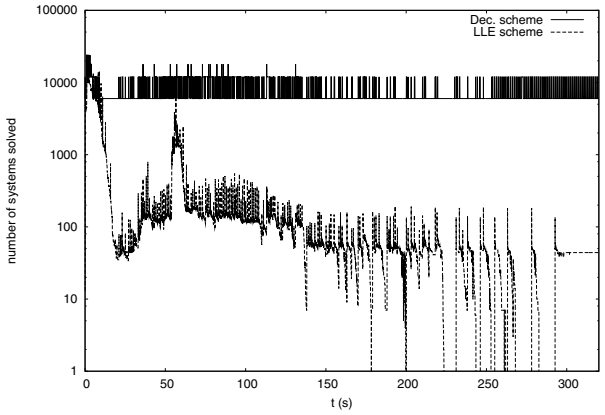

Figure 15: Case P2: number of injector Newton iterations

the oscillatory short-term dynamics.

Note that the model remains unchanged. Precisely, the main attractiveness of this simplified simulation is to avoid maintaining several models of the same system.

Case P3

A new scenario in the PEGASE test system was considered in order to illustrate how the simplified simulation filters out the short-term oscillatory dynamics (the effect could not be easily seen in Cases P1 and P2, where the simplified simulation produces almost the same results). At $t = 1$ s a fault was imposed at one end of the double-circuit line considered in Case P2. The fault was cleared at $t = 1.1$ s by opening that line. Once the transients induced by the fault have died out, the system evolves as in Case P2.

The time step sizes used in the detailed and simplified simulations, respectively, are shown in Table 3. Although most of the simplified simulation was performed with a large step of 0.5 s, a smaller value was adopted during and immediately after the fault, to capture the very large system variations.

Table 3: Case P3: time step sizes

time interval	detailed simulation	simplified simulation
[0.0 1.0] s	0.05 s	0.50 s
[1.0 1.1] s	0.01 s	0.05 s
[1.1 1.4] s	0.05 s	0.15 s
[1.4 320.0] s	0.05 s	0.50 s

Figure 16 shows the evolution of the field voltage of a synchronous machine located close to the fault, over a short period including the fault, as provided by detailed and simplified simulations, respectively. Note that the LLE scheme has been used in both cases. As can be seen, the

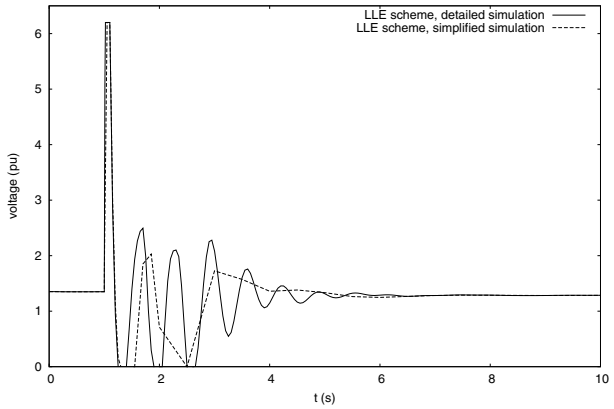

Figure 16: Case P3: field voltage of a synchronous machine

simplified simulation overlooks the oscillations but its output rejoins the detailed one, when the system reaches (temporary) equilibrium.

Computational effort of simplified simulation

The columns labeled "S-LLE" in Tables 1 and 2 show the computational effort of simplified simulation in Cases P1 and P2, while for Case P3 the figures of the four schemes are given in Table 4.

Table 4: Case P3: computational effort comparison

nb. of times	Dec.	Loc.	LLE	S-LLE
$\tilde{\mathbf{D}}$ factorized	191	30	53	31
(15) solved	$13.2\ 10^3$	$7.4\ 10^3$	$8.5\ 10^3$	$1.0\ 10^3$
$\mathbf{g}$ evaluated	$32.9\ 10^3$	$27.6\ 10^3$	$31.6\ 10^3$	$2.9\ 10^3$
$\mathbf{A}_i$ factorized	$1150\ 10^3$	$61.4\ 10^3$	$59.5\ 10^3$	$75.6\ 10^3$
(17) solved	$79.3\ 10^6$	$28.8\ 10^6$	$4.1\ 10^6$	$0.2\ 10^6$
$\mathbf{f}_i$ evaluated	$197\ 10^6$	$202\ 10^6$	$20.2\ 10^6$	$1.0\ 10^6$

Combining simplified simulation and LLE yields dramatic savings. In all three cases:

- the number of systems solved and functions evaluated is much lower at both injector and network levels;
- slightly more factorizations of $\mathbf{A}_i$ Jacobians are required, compared to the Loc. and LLE schemes, but much less than with the Dec. scheme;
- the network Jacobian is factorized as infrequently as with the Loc. scheme, and significantly less than with the Dec. scheme.

A slightly more difficult convergence is the cause for the higher number of $\mathbf{A}_i$ factorizations, while the much larger step size explains the much lower number of equations solved and function evaluated.

Computing times

Although computing times remain the bottom line of such investigations, they may be misleading or premature results. They depend on the final implementation in a production-grade software. For instance, in some simulation packages, the $\mathbf{f}_i$ function evaluations from a user-friendly modelling interface can be an expensive item, requiring to give more weight to the figures in the last row of Tables 1, 2 and 4.

Tentative computing times measured on the P2 scenario (which is representative of dynamic security assessment studies targeted by the tools presented in this paper) are as follows:

- Dec. scheme: 291.3 s
- Loc. scheme: 215.9 s
- LLE scheme: 123.1 s
- LLE scheme combined with simplified simulation: 13.1 s.

These times have been observed using a standard PC (Intel Core 2 Duo E8400 CPU, 3 GHz, 3.24 Gb RAM, running under Windows XP).

The gain of the LLE scheme with respect to the Dec. one is probably lower than what could be expected from the figures in Tables 1, 2 and 4. This is attributable to the still large effort spent on solving the network equations (15). We expect, however, to somewhat decrease this item by resorting to a more recent sparse solver. Furthermore, in the PEGASE test system, the proportion of network equations among the whole set of equations is higher than what could be expected in a real-life detailed model. This ratio is $\frac{30,000}{114,350} = 0.26$. Increasing the average number of states per power plant from 17 to - say - 30 would decrease this ratio to $\frac{30,000}{114,350+3,824\times(30-17)} = 0.18$ and, hence, the efficient treatment of injectors by the proposed method would yield a larger speed-up ratio.

Summary and concluding remarks

In this paper a novel approach for faster power system time-domain simulation has been presented. It exploits the bordered block diagonal structure of the Jacobian involved in Newton method, the localized nature of the system response and the latent behavior of components little affected by the simulated disturbance. The proposed approach can be used either in detailed simulation, where accuracy is preserved, or combined with simplified simulation, to obtain substantial computational savings.

Some issues that could improve the overall efficiency while preserving accuracy are presently under investigation. Among them, let us quote:

- safeguards to avoid essential injectors from being switched to latent mode, for instance when subject to an internal dynamics not observable from the active or reactive power outputs;
- conversely, criteria for earlier switching of injectors to latent mode without having to simulate their initial response to the disturbance (until τ seconds of history are collected);
- alternative equivalents to replace the latent blocks, for instance equilibrium models as in the quasi steady-state approximation;
- possible ways to reduce the bottleneck caused by solving the network equations in sequence with the injector ones.

Although the tests were performed in the context of sequential computing, further advantages could be obtained from the parallel processing of the injectors.

Acknowledgement

This work was performed in the context of the PEGASE project funded by European Community's 7th Framework Programme (grant agreement No. 211407).

REFERENCES

[1] T. Van Cutsem, C. Vournas, *Voltage stability of electric power systems*, Springer (previously Kluwer Academic Publishers), Boston (USA), 1998, ISBN 0-7923-8139-4

[2] M.-E. Grenier, D. Lefebvre, T. Van Cutsem, "Quasi steady-state models for long-term voltage and frequency dynamics simulation", Proc. IEEE Power Tech conference, St Petersburg (Russia), June 2005. Available at http://ieeexplore.ieee.org (DOI 10.1109/PTC.2005.4524400) and http://hdl.handle.net/2268/5661

[3] D. Fabozzi, T. Van Cutsem, "Simplified time-domain simulation of detailed long-term dynamic models", Proc. IEEE PES General Meeting, Calgary (Canada), July 2009. Available at http://ieeexplore.ieee.org (DOI 10.1109/PES.2009.5275463) and http://hdl.handle.net/2268/9524

[4] A. Bose, "Parallel processing in dynamic simulation of power systems", Sadhana, Academy Proceedings in Engineering Sciences, Indian Academy of Sciences, Vol. 18, Part 5, Sept. 1993, pp.815-827

[5] M. Ilic'-Spong, M.L. Crow, M.A. Pai, "Transient stability simulation by waveform relaxation methods", IEEE Trans. on Power Systems, Vol. PWRS-2, No. 4, Nov. 1987, pp. 943-952

[6] M. La Scala, M. Brucoli, F. Torelli, M. Trovato, "A Gauss-Jacobi-block-Newton method for parallel transient stability analysis [of power systems]", IEEE Trans. on Power Systems, Vol. 5, No. 4, Nov. 1990, pp. 1168-1177

[7] M. La Scala, A. Bose, D.J. Tylavsky, J.S. Chai, "A highly parallel method for transient stability analysis", IEEE Trans. on Power Systems, Vol. 5, No. 4, Nov. 1990, pp. 1439-1446

[8] M.L. Crow, J.G. Chen, "The Multirate Method for Simulation of Power System Dynamics", IEEE Trans. on Power Systems, Vol. 9, No. 3, Aug. 1994, pp. 1684-1690

[9] V. Brandwajn, "Localization Concepts in (In)-Security Analysis", Proc. IEEE Power Tech conference, Athens (Greece), Sept. 1993

[10] J. Ogrodzki, *Circuit Simulation Methods and Algorithms*, CRC Press, Inc., Boca Raton, FL, 1994

[11] P. Kundur, *Power System Stability and Control*, Mc Graw Hill, EPRI Power System Engineering Series, 1994

[12] D. Fabozzi, T. Van Cutsem, "On angle references in long-term time-domain simulations", to appear in IEEE Trans. on Power Systems. Available at http://ieeexplore.ieee.org (DOI 10.1109/TPWRS.2010.2042652) and http://hdl.handle.net/2268/30371

[13] U.M. Ascher, L.R. Petzold, *Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations*, Society for Industrial and Applied Mathematics (SIAM), 1998

[14] J.Y. Astic, A. Bihain, M. Jerosolimski, "The mixed Adams - BDF variable step size algorithm to simulate transient and long term phenomena in power systems", IEEE Trans. on Power Systems, Vol. 9, No. 2, May 1994

[15] B. Dembart, A.M. Eirsman, E.G. Cate, M.A. Epton and H. Dommel, "Power System Dynamic Analysis - Phase I. Final Report EPRI EL-484", July 1977

[16] M. Stubbe (Convener), *Long-Term Dynamics - Phase II*, Report of CIGRE Task Force 38.02.08, Jan. 1995.

[17] C. Vournas, T. Van Cutsem, "Local Identification of Voltage Emergency Situations", IEEE Trans. on Power Systems, Vol. 23, No 3, 2008, pp. 1239-1248. Available at http://ieeexplore.ieee.org (DOI 10.1109/TPWRS.2008.926425) and http://hdl.handle.net/2268/3194

[18] M. Glavic, T. Van Cutsem, "Wide-area Detection of Voltage Instability from Synchronized Phasor Measurements. Part I: Principle. Part II: Simulation Results", IEEE Trans. on Power Systems, Vol. 24, 2009, pp. 1408-1425. Available at http://ieeexplore.ieee.org (DOI 10.1109/TPWRS.2009.2023272) and http://hdl.handle.net/2268/7885

[19] [Online], Available at http://www.fp7-pegase.eu/