



PEGASE

PAN EUROPEAN GRID ADVANCED SIMULATION AND STATE ESTIMATION

D4.2-part-2 simplified simulation prototype

Associated report

Proprietary Rights Statement

This document contains information, which is proprietary to the "PEGASE" Consortium. Neither this document nor the information contained herein shall be used, duplicated or communicated by any means to any third party, in whole or in parts, except with prior written consent of the "PEGASE" Consortium.

Grant Agreement Number: 211407 implemented as Large-scale Integrating Project

Coordinator: GDF SUEZ

Project Website: <http://www.fp7-pegase.eu>

Document Information

Document Name:	D4.2-part 2-simplified simulation
ID:	DEL_TE_WP4.4_D4.2-part2-simplified simulation
WP:	4
Task:	4
Revision:	3
Revision Date:	29/08/2011
Author:	TE, RTE

Diffusion list

STB, SSB

Approvals

	Name	Company	Date	Visa
Author	B. Haut	TE		
	A. Chieh	RTE		
Task Leader	T. Van Cutsem	ULg		
WP Leader	B. Haut	TE		

Documents history

Revision	Date	Modification	Author
V0	25/07/2011	Initial version	B. Haut
V1	08/08/2011	Revision + added paragraphs	A Chieh
V2	22/08/2011	Comments from STB	
V3	29/08/2011	Typographical error corrections	B. Haut



Table of content

D4.2-part-2 simplified simulation prototype.....	1
Associated report.....	1
1. Introduction	4
2. Algorithm.....	4
2.1. Time simplification.....	4
2.2. Space simplification	5
2.2.1. Decomposition.....	5
2.2.2. Latency exploitation.....	6
3. Differences of implementation with respect to the ULg mockup.....	7
4. Alpha-tests and first performances estimation	9
4.1. Small testcase.....	9
4.2. PEGASE testcase	11
5. User guide.....	13
5.1. General use.....	13
5.2. New parameters.....	14
5.3. Limitations	14
6. Conclusions.....	15
7. References.....	15

1. Introduction

This document is the “report” associated to the simplified simulation prototype developed in WP4. The prototype itself is the main deliverable and can be found on the PEGASE intranet.

This deliverable has been renamed from “*D4.2 part 2 : QSS engine*” to “*D4.2 part 2 : simplified simulation prototype*” in order to reflect more accurately the work carried out by the prototype : *developing a calculation engine providing simplified simulation suitable for security analysis.*

The prototype under consideration has been obtained by re-integrating the algorithms developed and validated in T4.4 in an existing industrial software named EUROSTAG [5].

This document is structured as follow:

1. The main principles of the implemented algorithm are described.
2. The differences between the implementations in the final prototype (based on EUROSTAG) and in the first mock-up developed by ULg are described.
3. The behaviour of the prototype on two important alpha-tests is described and the associated performances are presented.
4. The user guide of the prototype.

It is assumed in this document that the reader is already familiar with the simplified simulation algorithms briefly reminded in Section 2 and the EUROSTAG software.

2. Algorithm

The main principles of the algorithms are only briefly summarized in this section. Full descriptions of the algorithms developed for simplified simulation can be found in [1,2, 3, 4].

The algorithms developed for simplified simulation can be decomposed in two parts:

- Time simplification: use of a stiff-decay method with large time step.
- Space simplification: decomposition of the Jacobian and exploitation of the latency

2.1. Time simplification

The following principles are used in order to reduce the computation effort:

- Use of the Backward Euler Method which has the stiff-decay property;
- Use, most of the time, of a large constant stepsize (around 0.5 sec). If a convergence problem is detected during the Newton iteration then the stepsize is reduced to achieve convergence and re-increased as fast as possible in order to reach back the constant stepsize.
- Use of an “order-0” predictor.
- If an event leading to a discontinuity (such as the opening of a branch) at time t happens then the corresponding algebraic variables are not updated for time t^+ . The discontinuity will be taken into account during the integration step from t to $t+h$
- For the handling of discrete variables, the main idea is:
 1. Freeze the discrete variables and perform an integration step from t to $t+h$.

2. Evaluate the transition conditions associated to the discrete variables. If one condition indicates that a discrete variable must be switched then modify it and go back to step 1 otherwise proceed to the next integration step.

2.2. Space simplification

The space simplification depends on two new ideas: the *decomposition* and the *latency exploitation*.

2.2.1. Decomposition

Usually to solve the linear systems arising in the Newton iterations, the global Jacobian is factorized under an LU form. The decomposition is an alternate way where the global Jacobian is split into different components: one component for each injector and one component for the network. Each component is factorized independently and the interactions between the injectors and the network are managed explicitly.

This technique, which does not introduce any simplification, is summarized in Figure 1.

The two main advantages are the following:

- The injector jacobians can now be updated and re-factorized independently.
- The structure is well suited for the latency exploitation described in the next section.

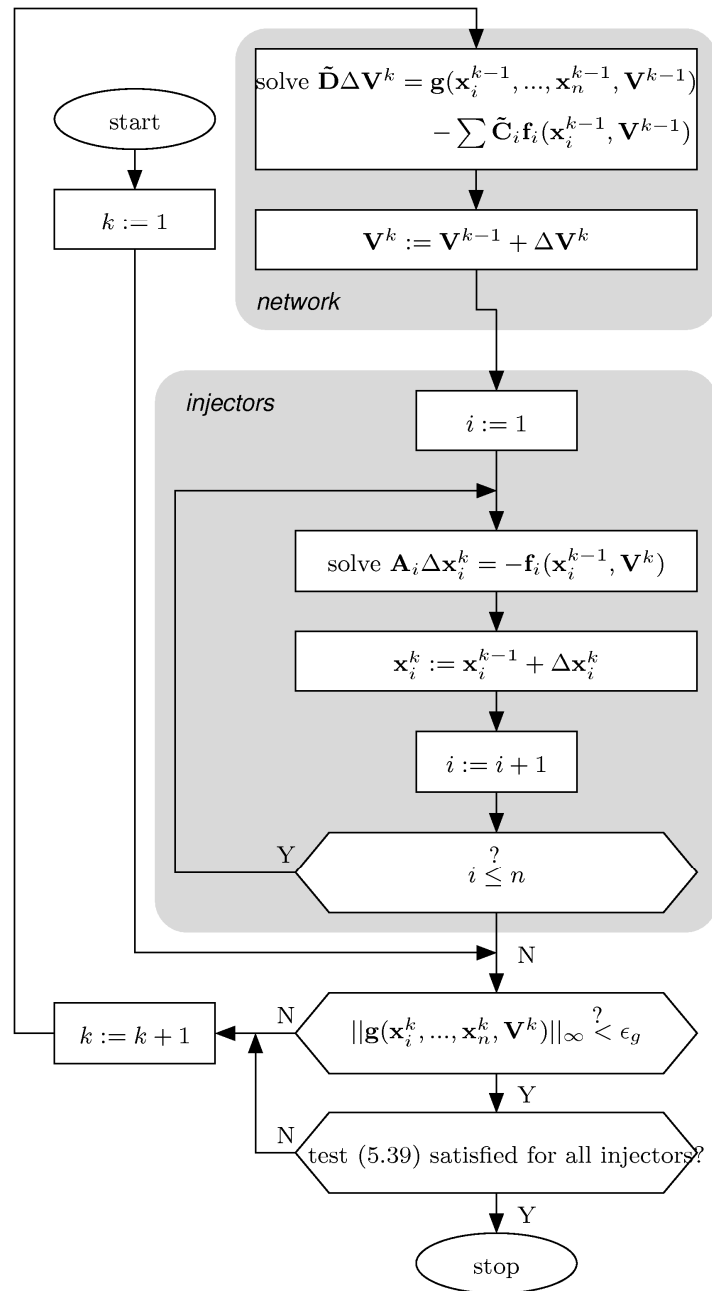


Figure 1: Decomposed Newton scheme

2.2.2. Latency exploitation

The idea behind latency exploitation is to solve only a subset of injectors, identified as active, and skip solution of the other ones, identified as latent. These injectors are replaced by a linear approximation between currents and voltages.

Which injectors are latent and which ones are active cannot be decided a priori because it depends on the simulated disturbance. The following rules are used to classify an injector as latent or active:

1. At the beginning of the simulation or after a user-imposed event: all injectors are classified active.
2. At the end of each time step:
 - a. an injector which has undergone a discrete change is classified as active for the whole remainder of the simulation.
 - b. an injector switches from active to latent if its current components have not changed by more than a threshold since the beginning of the simulation or since the last linearization time.
3. Inside one time step, a latent injector switches from latent to active if
 - a. at least one of its current components exhibits a variation above the specified threshold.
 - b. the mismatches associated to the equations corresponding to the connexion nodes are larger than a given threshold.

3. Differences of implementation with respect to the ULg mockup

The algorithms presented in Section 2 have been first tested and validated in a mockup developed by ULg. The results presented in the reports of task 4.4 and the papers published in the various conferences are based on simulations performed with this mockup.

There are some differences between the mockup implementation and the implementation performed in the final prototype based on EUROSTAG. These differences are:

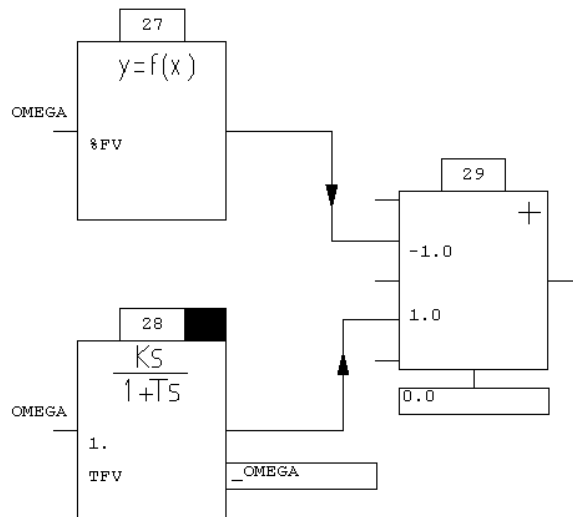
- The state variables are not the same. For example the currents are state variable in the ULg mockup and not in the prototype. It implies that
 - The matrices C_i are not only 0 and 1's in the prototype. However all the formula presented in the various reports/publications are still valid.
 - It could impact the convergence speed (positively or negatively).
- The coupled injectors are managed in the prototype. It implies that a matrix A_i can represent more than one injector and that the corrections applied to the network matrix are not systematically 2x2 but of any size.
- The infinite norm is not used to control the convergence of the Newton iterations. A weighted 2-norm is used instead.
- The mismatches on the network equations are not used to control the convergence of the Newton iterations. The amplitude of the correction is used instead.
- The criteria used to update the Jacobian are not exactly the same even if the global idea is the same (update the Jacobian if the convergence is not fast enough or if a discrete change has been detected).
- The LU libraries used to factorize the injector Jacobian are different: BLAS in the mockup implementation and KLU in the prototype.
- In the mockup, the maximum stepsize is the constant stepsize. In the prototype, if no discontinuities are observed, an error control strategy similar to that in classical EUROSTAG is used and the stepsize can become larger than the constant stepsize. One significant difference between the mockup and the prototype concerns the detection of discrete change in order to disable latency for a particular injector.
 - In the mockup the latency is disabled for each discrete change.
 - In the prototype the discrete changes are classified in two categories. Assuming that the system is written under the following form:

$$x' = f(x).$$

If f' is discontinuous at a point x^* then it is called a level-1 discontinuity. If f itself is discontinuous at a point x^* then it is called a level-2 discontinuity.

The level-1 discontinuities are very common in a simulation and disabling latency for each injector which undergoes such discontinuity will result in a very important loss of performances.

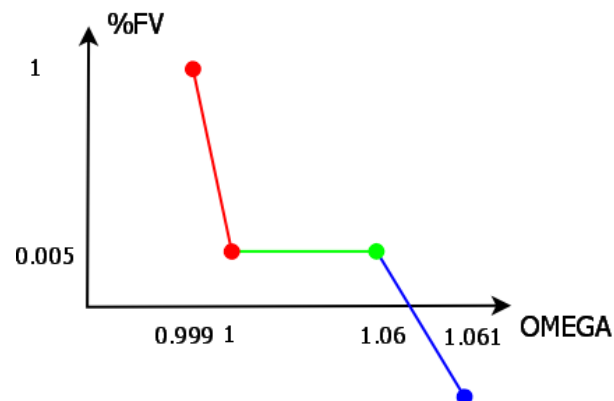
For example let consider the following part of the GOVCLASM macroblock used on most of the machines in the PEGASE testcase.



Fast Valving

Figure 2: a part of the GOVCLASM macroblock

The %FV block is defined as the following piece-wise linear function:



At the beginning of the simulation OMEGA is perfectly equal to 1 (and therefore associated to the red discrete state). If after an event the speeds slightly increase from 1 to $1+1e-6$ for all machines on the network then each machine will undergo a discrete change (from red to green) even if the output of the %FV block does not change. If the latency is disabled for such discrete change nearly all machines on the network will have never been latent.

In the simplified simulation prototype the latency is disabled for a machine only if a **level-2** discontinuity is detected.

4. Alpha-tests and first performances estimation

The prototype has been validated on two alpha-tests:

1. A small testcase named test14_44. This is a very small testcase where small speedup can be expected but it presents some interesting features to validate the development:
 - a. There are some coupled injectors whose variables are non-consecutive in the Jacobian. It implies that a re-ordering must be performed from the global Jacobian to the “local” jacobians associated to each machine.
 - b. Some coupling devices are present in the network.
 - c. A complex scenario including starting/stopping of machines.
2. The PEGASE testcase with a branch opening.

4.1. Small testcase

The prototype was first tested on a small system composed of:

- 13 machines (some of them are coupled)
- 25 buses
- 2171 state variables
- Coupling devices in the network

This system was simulated with the following scenario:

- Start of simulation at steady state at $t = 0$
- At $t = 1$ s, start-up of two machines (M1 and M2)
- At $t = 2$ s, modifications of steps of two capacitor banks
- At $t = 3$ s, shutdown of two machines (M6 and C2)
- At $t = 4$ s, start-up of machine M7
- At $t = 5$ s, modifications of steps of two capacitor banks
- At $t = 6$ s, shutdown of machine M2
- End of simulation at $t = 100$ s.

Different simulations were performed:

- Test14_44_e: standard EUROSTAG with the classical error control. It provides the “reference solution”.
- Simplified simulation with different parameters of latency:
 - Test14_44_lat0: LAT_EPS set to 0.0 (latency disabled)
 - Test14_44_lat1: LAT_EPS set to 0.001
 - Test14_44_lat3: LAT_EPS set to 0.02

Figure 3 shows the voltage at two buses of the system, M1 and N101. The first one is the connection bus of machine M1, and the second one is the connection bus of one of the capacitor banks.

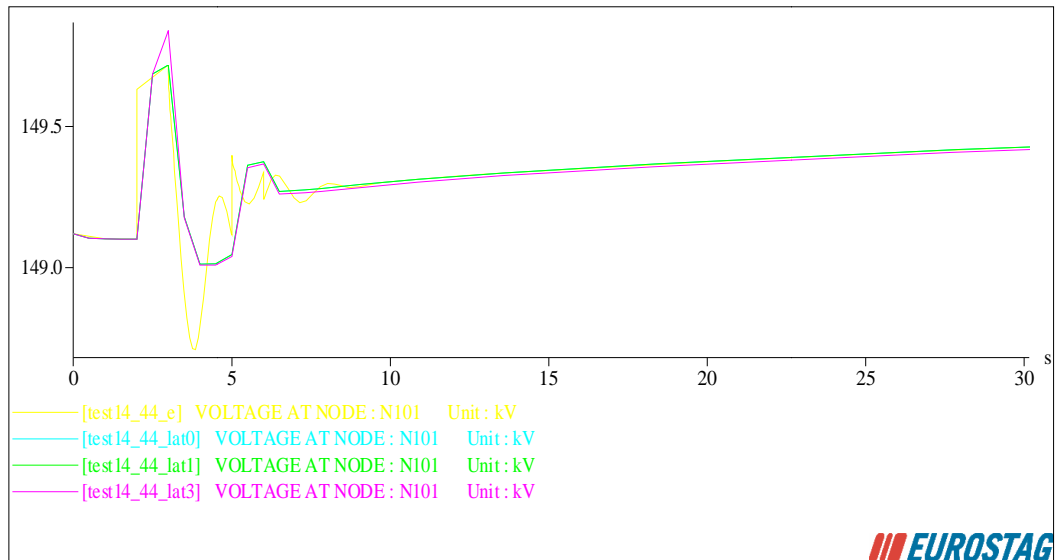
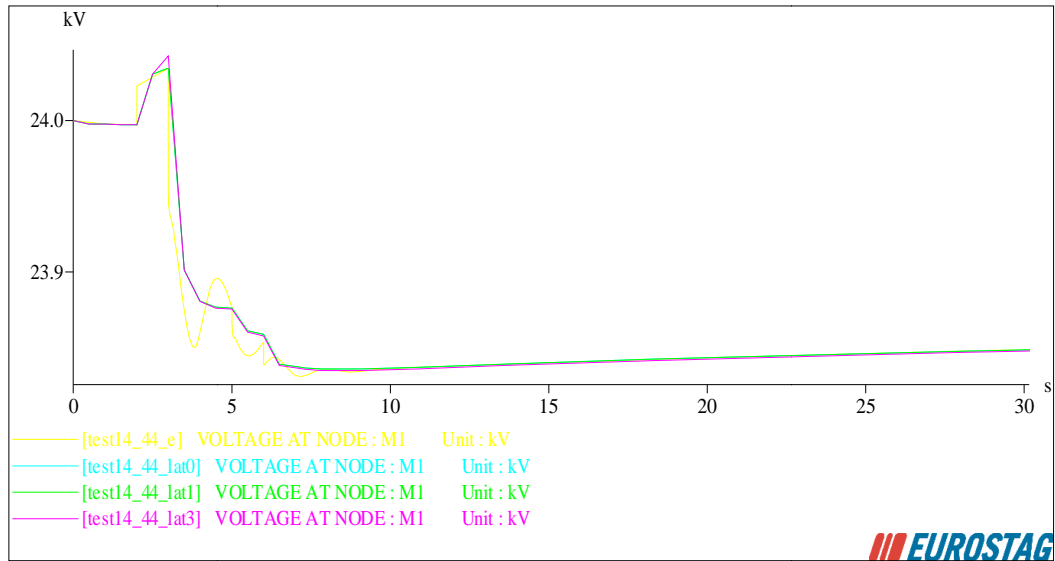


Figure 3 : voltages at nodes M1 and N101

By comparing the results computed by the simplified simulation prototype (shown in blue, green, and pink in Figure 3) to the reference EUROSTAG solution (shown in yellow in Figure 3), we observe that:

- The fast oscillations of the reference solution were filtered in the simplified simulation results.
- In the simplified simulation results, discrete events are delayed of approximately 0.5 s. This delay is due to the switching of discrete states uniquely at the end of a time step.
- After the disturbances, the results coincide.

Table 1 shows the computation costs of the simulations performed. Because the system is small, there is no difference of cost between the different latency thresholds. Using a latency threshold of 0 (no latency), 1313 machine function computations are done, whereas using a latency threshold of 0.02, this number is reduced only to 1151.

The gain in computation speed of the simplified simulation in relation to the full accuracy simulation is of 2.

	EUROSTAG full accuracy	Simplified simulation prototype		
		0	0.001	0.02
Latency threshold	---	0	0.001	0.02
Integration steps	98	24	24	24
Newton steps	156	63	63	63
Total time [s]	2.0	1.0	1.0	1.0

Table 1: performances on the small testcase

4.2. PEGASE testcase

The following testcase is considered:

- PEGASE system (v25)
 - 3.844 machines
 - 15.350 buses
 - 123.714 state variables
 - Impedant loads on the whole network
- Scenario:
 - Start of time $t = 0$ at steady-state
 - Opening of branch F2M6CJ00-F0W6M300-1 at time $t = 10$ seconds
 - End of simulation at time $t = 100$ seconds

Different runs have been performed:

- Line-op-fs: standard EUROSTAG with the classical error control. It provides the “reference solution”.
- Simplified simulation with different parameters of latency:
 - Line-op-s : LAT_EPS set to 0.0 (latency disabled)
 - Line-op-s1 : LAT_EPS set to 0.001
 - Line-op-s2 : LAT_EPS set to 0.005
 - Line-op-s3 : LAT_EPS set to 0.02

In Figure 4 two measurements are represented: the voltage very close to the line outage (F2M6CJ00) and the voltage far away from the line outage (I2M8V300). The following observation can be made:

- Near the outage where there is a strong voltage variation:
 - The simplified solution is close to the reference solution. The maximal voltage deviation is slightly less in the simplified simulation than in the reference simulation. This is a typical behaviour of simplified simulation (over-damping in the solution).
 - The exploitation of latency does not appear to have an impact on the solution around bus F2M6CJ00. Probably that the machines very close to the outage remain active during the transient.
- Far away from the outage where the voltage variation is very small (maximal deviation of $2E-3$ pu):

- The differences between the reference solution and the simplified simulation are more obvious:
 - strong damping of the simplified solution;
 - the resulting steady state is slightly different.
- If the latency threshold LAT_EPS increase then the differences increase.

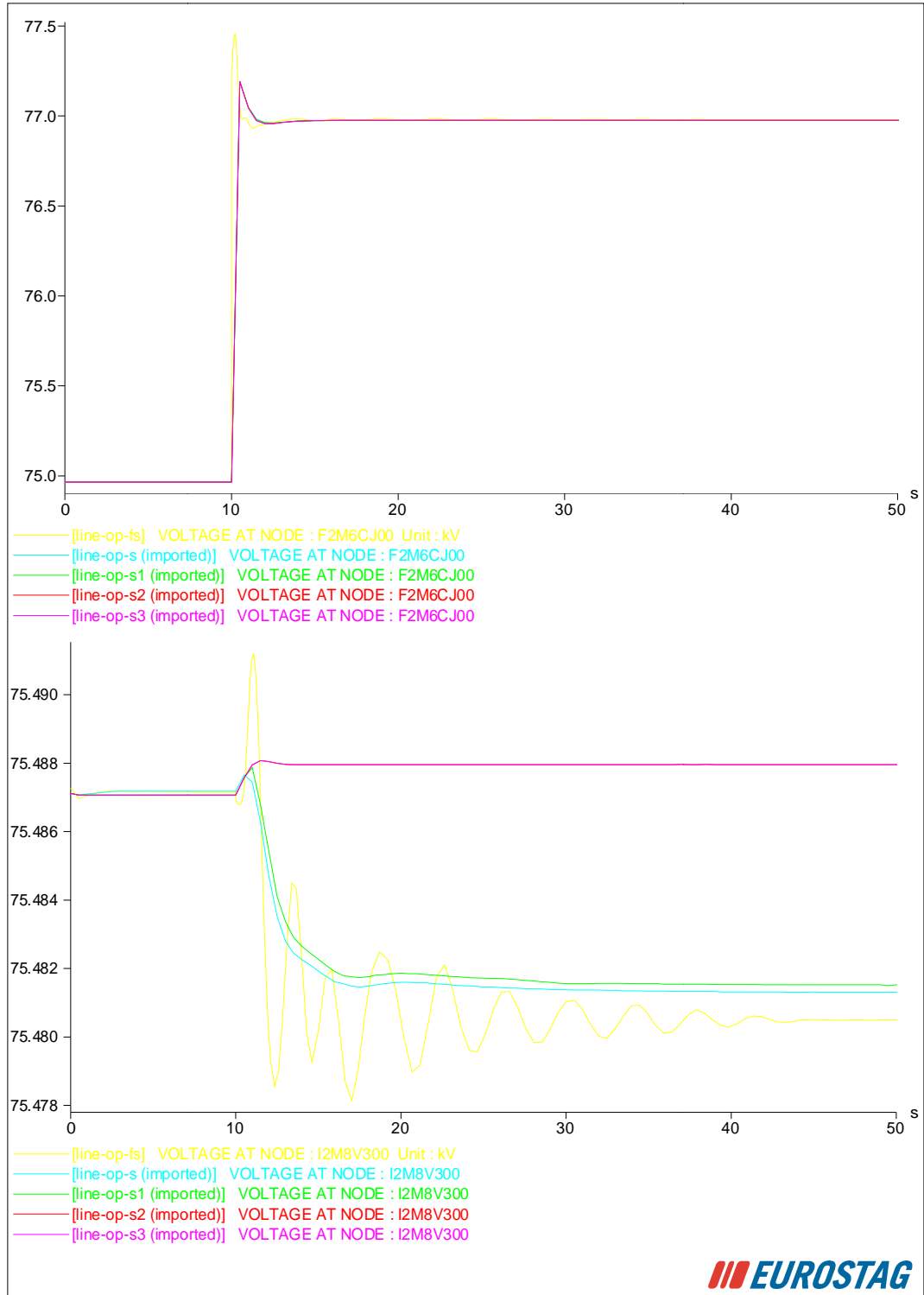


Figure 4 : voltages at two buses (reference and simplified simulations)

The computational costs of the different simulations are represented in Table 2. The following observations can be made:

- The standard version of EUROSTAG is very efficient. In this particular scenario, the outage has a very limited effect on the system and EUROSTAG is quickly able to use, after the outage, a stepsize larger than 0.5 s (fixed stepsize used by the simplified prototype).
- Without latency the speedup obtained by simplified simulation is very limited due to two counter-balancing factors:
 - The first one is negative. Since the stepsize is limited to 0.5 s more integration steps are performed and therefore more Newton steps. It leads to a larger time for function evaluations.
 - The second one is positive. Due to the decomposition, the Jacobian must not be evaluated as a whole. If a discontinuity occurs inside a machine then only the corresponding Jacobian part must be factorized. The time spent for the Jacobian evaluation/factorization is smaller.
- The number of integration steps of the simplified simulation prototype is 201. Since the fixed stepsize specified was 0.5 s and since the total time of simulation is 100 s it indicates that the fixed stepsize was used during the whole simulation. It has not been needed to reduce this stepsize to achieve convergence of some Newton iterations.
- When the latency threshold becomes significant ($LAT_EPS \geq 0.005$) then there is an important gain in term of function evaluation. This gain does not increase more for larger thresholds (most of the machines are already latent for $LAT_EPS = 0.005$).
- The total speedup appears limited (only 2.4 for $LAT_EPS = 0.005$) but it should be reminded that this testcase was particularly favourable for the EUROSTAG full accuracy. It is expected to have better speedup with a more complex scenario.

	EUROSTAG full accuracy	Simplified simulation prototype			
Latency threshold	---	0	0.001	0.005	0.02
Integration steps	86	201	201	201	201
Newton steps	186	213	224	207	207
Time for function evaluation [s]	17.5	19.01	17.42	4.97	5.28
Time for Jacobian evaluation/factorization [s]	9.3	1.31	1.53	1.17	1.55
Total time [s]	38.67	35.0	32.0	16.0	17.0

Table 2: performances on the PEGASE testcase

5. User guide

5.1. General use

The simplified simulation prototype is based on EUROSTAG 4.5. Except if stated differently in this section, everything included in the EUROSTAG 4.5 manuals is valid for this prototype (including the data format).

The prototype uses usual EUROSTAG .ech, .dta and .seq files.

5.2. New parameters

When the simplified simulation prototype starts the working directory is scanned for a file named “simplified_simulation.txt”. This file is used to specify the following new parameters.

Parameter	Default value	Description
YHFIX	0.5	Value of the fixed stepsize.
NEFF	3	Number of Newton iterations to be reached when the stepsize is reduced following a non-convergence.
NHB	0	Tolerance on the number of Newton iterations (used when the stepsize is reduced following a non-convergence).
NMRED	100	Maximal number of successive stepsize reductions.
NCYMAX	3	Maximal number of discontinuity cycles.
NEWTMA	10	Maximal number of Newton iterations.
YKRED	0.1	Stepsize reductions factor (used in case of non-convergence of the Newton iterations).
ITRAST	0	If equal to 1, a debug file is written.
YJTOLMACDIR	1/6	Threshold on the correction amplitude used to trigger the re-evaluation of the machine Jacobian.
YJTOLNETDIR	1/6	Threshold on the correction amplitude used to trigger the re-evaluation of the machine Jacobian.
YJTOLMACERR	3	Threshold on a “pseudo-truncation error” used to trigger the re-evaluation of the machine Jacobian. Decreasing should improve performances and degrade quality.
YJTOLNETERR	3	Threshold on a “pseudo-truncation error” used to trigger the re-evaluation of the network Jacobian. Decreasing should improve performances and degrade quality.
LAT_EPS	0.005	Threshold related to the injected current used to switch an injector from active to latent. Increasing should improve performances and degrade quality.

If the “simplified_simulation.txt” file is not present in the working directory, the default values are used.

5.3. Limitations

This section lists the limitations with respect to EUROSTAG 4.5:

- The distribution network subloads are not well handled in presence of latency ($LAT_EPS > 0$). It is recommended to use impedant loads.
- The binary “.sac” files are not compatible between the simplified prototype and the standard EUROSTAG 4.5.

- The following events are not supported by the simplified prototype: eigenvalue computation, system linearization.
- Even if short-circuits are supported, the prototype is not designed for their simulation. The simulation of short-circuits may force the prototype to reduce its stepsize.

6. Conclusions

The algorithms for simplified simulation developed in the framework of the Task 4.4 have been successfully integrated in EUROSTAG leading to the final simplified simulation prototype of WP4.

The first alpha tests show good results both in terms of accuracy (the solutions found are not too far away from the exact solutions) and performance (speedup of 2 reached on a scenario favourable to the commercial EUROSTAG implementation). More tests still need to be carried out to have a better idea of the average speedups. It will be done in the framework of “WP6-Demonstration”.

7. References

1. The PEGASE report R4.4.2 “*Algorithmic and modelling improvements for simplified simulation; validation on industrial cases*”.
2. D. Fabozzi, T. Van Cutsem, “Simplified time-domain simulation of detailed longterm 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>
3. D. Fabozzi, T. Van Cutsem, “Localization and latency concepts applied to time simulation of large power systems”, Proc. 2010 IREP Symposium, Buzios (Brazil), August 2010. Available at <http://ieeexplore.ieee.org> (DOI 10.1109/IREP.2010.5563287) and <http://hdl.handle.net/2268/65833>
4. D. Fabozzi, A.S. Chieh, P. Panciatici, T. Van Cutsem, “On simplified handling of state events in time-domain simulation”, accepted for presentation at the 17th Power System Computation Conference (PSCC), Stockholm, Sweden, Aug. 2011, pp. 1-9. Available at <http://hdl.handle.net/2268/91650>
5. EUROSTAG, a software developed for accurate and reliable simulations of power systems dynamics, www.eurostag.be