MODRIO

.

WP6 Modeling and simulation services

D 6.2.7 State-of-the-art analysis of complex systems with varying parameters

Document describing techniques for the decomposition of large scale hybrid systems in the context of parallel and distributed simulation. It aims to provide rules of decomposition that will constitute an input to the tasks related to parallel compilation, execution and sensitivity analysis

Version 1.0
Date July 10, 2014

Authors:

Thierry SORIANO	SUPMECA	Toulon	France
Manel ZERELLI	SUPMECA	Toulon	France
Jean BRAC	IFPEN	Rueil	France
Thibaut-Hugues GALLOIS	IFPEN	Rueil	France
Mongi BEN GAÏD	IFPEN	Rueil	France

Accessibility : Public

Contents

Table of figures

Introduction

This document is a state of the art analysis of complex systems with varying parameters. Hybrid systems are systems whose behavior can be expressed as an interconnection of discrete event models, for example state machines, with time continuous models represented for instance as ordinary differential equations. Complexity emerges firstly from the size of the graphs and/or or the ODEs and secondly from the coupling level of equations and/or events. The analysis of such nonlinear systems is quite impossible today with formal approaches and only the simulation is possible. Then in order to multiply the number of scenarios tested, it is necessary to increase the speed of executions and use particularly the parallelization approach. The document provides a description of parallel computation for large scale systems and of parallel simulation of large hybrid systems. It presents especially the Waveform Relaxation algorithm (WFR) as a parallel algorithm using the windowing method, parallelization and partitioning. The last part of this document is focused on the tools which are devoted to the analysis of the behavior when the hybrid system contains parameters in the continuous model with varying values; the sensibility analysis (SA) is studied. A definition of SA is given and some tools are presented for this type of analysis like stochastic methods, deterministic methods and an industrial tool using Modelica.

I- Hybrid system

1- Definition

Hybrid systems are dynamical systems that involve with interaction of different types of dynamics: continuous state dynamics and discrete state dynamics. Recall that discrete variable takes only finite number of values and continuous variable takes value in Euclidean space R^n. Discrete states can change value only through a discrete "jump", while continuous states change value in a "jump" or " in continuous time according to a differential equation. In hybrid system it is an interaction of discrete jumps and continuous evolution. The analysis and design of hybrid systems is in general more difficult than that of purely discrete or purely continuous systems, because the discrete dynamics may affect the continuous evolution and vice versa [**Lygeros, 2003**], [**Lygeros, 2004**].

• In mechanical systems continuous motion may be interrupted by collisions.

• In electrical circuits continuous phenomena such as the charging of capacitors, etc are interrupted by switches opening and closing.

• In embedded computation systems a digital computer interacts with a mostly analogue environment.

2- Hybrid automata

A hybrid automaton is a framework that describes the evolution in time of the values of a set of discrete and continuous state variables.
A hybrid automaton is a collection H= (Q, X, Σ, A, Inv, F, q0, x0) with:

- Q : a set of discrete states, q_0 initial situation;
- X, a set of continuous states, $X \subset \mathrm{R}^n$; x_0 the initial value of the continuous state
- Σ, a set of events;
- A, a set of transitions defined by (q, Guard, σ , Jump, q') with :
    - $q \in Q, q' \in Q,$
    - Guard, a set of state space.
    - Jump, represents the continuous state transformation in the situation change
    - $\sigma \in \Sigma$, event associated to the transition
- Inv: invariant of the state
- F, defines for each state the continuous evolution when the state is active. [**Henzinger, 1995**].

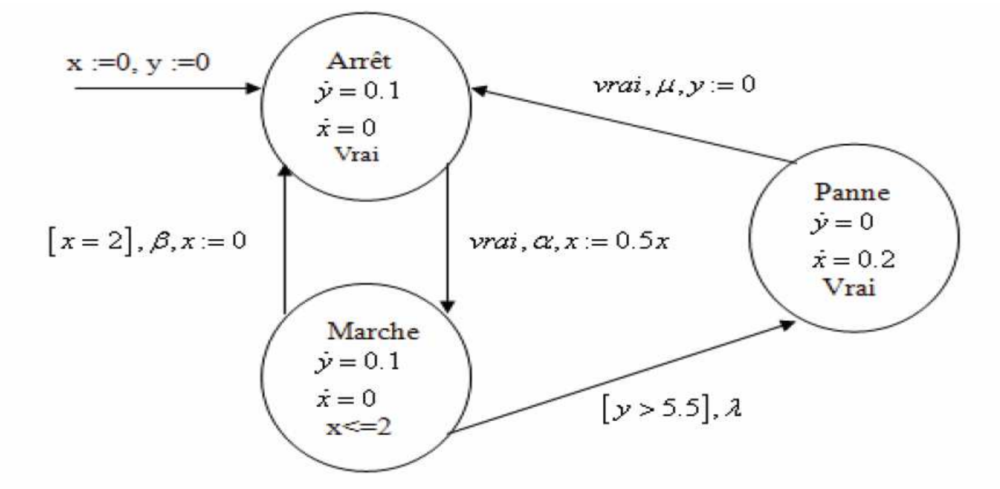

Figure 1 Example of a hybrid automaton for a simple case

II- Parallel computation

A parallelization strategy consists to support both cache-based and vector architectures. It includes both

- A Message Passing Interface-based parallel implementation to communicate between groups of partitioned sub domains

- An Open-MP set of directives to communicate between various sub domains contained within each MPI process.

Thus, the code can run in a purely MPI mode suitable for distributed memory architecture, a purely Open-MP mode suitable for shared memory architecture or an hybrid two level MPI-Open-MP mode suitable for cluster of shared memory processors as it currently occurs on large parallel supercomputer architectures.

1. Parallel Discrete Event Simulation

Parallel discrete event simulation is a simulation of the discrete event on parallel computers**.** The advantage of parallel discrete event simulation is reducing the time and memory more than in the serial execution. It is used to solve large-scale composite models.

Simulation system is modeling systems behavior using mathematical and logical relationships. In simulation, computers are used to evaluate the models numerically, where data is gathered and used to estimate the behavior of target systems. In simulation, we represent the state evolution over time. The target system can be viewed as a continuous system, discrete system or as a hybrid system.

- In continuous systems, state changes continuously with respect to time.
- In discrete system the state changes only at specific period of time.
- The hybrid system is a combination of both continuous system and discrete jump.

There are two simulation methods available:

- Time driven simulation: (or time-stepping) time is measured at small intervals giving the impression that the system evolves continuously over time. As such, it is naturally more appropriate for simulating continuous systems.
- Event driven simulation. (or discrete-event) simulation, time jumps through distinct points in time, which we call events. Consequently, event-driven simulation is more suitable for simulating discrete systems.

Note that one can combine both types of simulations, for example, in a computer network simulation, using discrete events to represent detail network transactions, such as sending and receiving packets, and using continuous simulation to capture the fluid dynamics of overall network traffic.

A discrete-event simulation maintains a data structure called the event-list, in which events are classified in a priority queue according to the time at which they are scheduled to happen in the simulated future. A clock variable T is used to points out the current time in simulation. The program is based on a loop; the simulator repeatedly removes an event with the smallest timestamp from the event list, sets the clock variable T to the timestamp of this event, and processes the event. Processing an event typically changes the state of the model and may generate more future events to be inserted into the event-list. The loop continues until the simulation termination condition is met, for example, when the event-list becomes empty or when the simulation clock has reached a designated simulation completion time.

Parallel Discrete Event Simulation (PDES) is a research area for parallel simulation and high performance computing. PDES is the execution of a single event simulation on parallel systems like clusters, shared memory multiprocessors or a combination of both. That's why, PDES can bring substantial benefit to time-critical simulations, and simulations of large-scale systems that demand an immense amount of computing resources.

The simplest form of parallel simulation is called **replicated trials**, which executes multiple instances of a sequential simulation program concurrently on parallel computers. The advantage of this approach is that is simple and it can expedite the exploration of a large parameter space. The disadvantage is that each replicated trial does not provide any speed up and cannot overcome the memory limit due to sequential execution. To address the latter problem, Hybinette and Fujimoto introduced in [Hybinette, 2001] a cloning method as an efficient parallel computation technique that enables simultaneous exploration of different simulation branches resulted from alternative decisions made in simulation.

Another form of parallel simulation is to assign different functions of a simulation program, such as random number generation and event handling, to separate processors. This method is called **functional decomposition**. The main problem is the lack of ample parallelism. Also, the tight coupling of the simulation functions creates an excessive demand for communication and synchronization among the parallel components, which can easily defeat the parallelization effort.

More generally, one can view simulation as a set of state variables that evolve over time. In space-time view of simulation each event can be characterized by

- a temporal coordinate, indicated by the timestamp of the event,
- a spatial coordinate, indicated by the location of the state variables affected by the event.

According to the space-time view of simulation, the state space of a discrete-event simulation can be consisting of a continuous time axis and a discrete space axis; the objective of the simulation is therefore to compute the value at each point in the space-time continuum. This space-time view provides a high-level unifying concept for parallel simulation, where one can divide the space-time graph into regions of arbitrary shape and assign them to separate processors for parallel processing.

The time-parallel approach is a special case of the space-time view. We proceed to a temporal decomposition of the time-space continuum.

Time- parallel simulation divides the space-time graph along the time axis into non-overlapping time intervals, and assigns them to different processors for parallel processing. Due to the obvious dependency issue, that is, the initial state of a time interval must match the final state of the preceding time interval, the efficiency of this approach relies heavily on the model's ability of either rapidly computing the initial state or achieving fast convergence under relaxation. For this reason, only a limited number of cases using time-parallel simulation exist in the literature. Successful examples include trace-driven cache simulations, queuing network, Petri net simulations, and road traffic simulations.

Orthogonal to the time-parallel approach, **space-parallel** simulation is based on data decomposition, where the target system is divided into a collection of subsystems, each simulated by a logical process (LP). Each LP maintains its own simulation clock and event-list, and is only capable of processing events pertaining to the subsystem to which it is assigned. These LPs can be assigned to different processors and executed concurrently. Communications between the LPs take place exclusively by exchanging time stamped events. Space-parallel simulation is in general more robust than the other parallelization approaches, mainly because data decomposition is naturally applicable to most models.

2. Performance metric

A performance metric is a tool to determine systems performance. It focuses on measuring a certain aspect of the system and allows comparison of various types of systems. The criteria for evaluating performance in parallel computing can include: speedup, efficiency and scalability.

2.1 Speedup

In general, for multiprocessing systems we want to know who is faster parallel algorithm or sequential one. For that we use to show how much a parallel algorithm is faster than a sequential one. It is defined as follows:

$$S_p = \frac{T_1}{T_p} \qquad (1)$$

where Sp is the speedup, T_1 is the execution time for a sequential algorithm, T_p is the execution time for a parallel algorithm and p is the number of processors.

There are three possibilities for speedup: linear, sublinear and super-linear, shown in figure 2.

- The speedup is called linear when it is equal to the number of processors Sp = p. In such a case, doubling the number of processors, will double the speedup.
- The speedup is called sub-linear when its evolution is inversely proportional to the number of processors, i.e increasing the number of processors, decreases the speedup. Most algorithms are sub-linear, because of various overheads associated with multiple processors like communication. This can occur because of the increasing parallel overhead from such areas as: interprocessor communication, load imbalance, synchronization, and extra computation. An interesting case occurs in super-linear speedup, which can mainly be due to cache size increase.

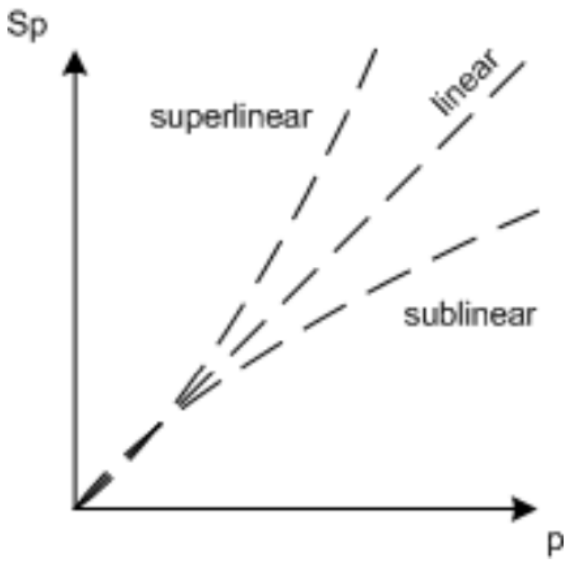

Figure 2 Speedup vs. number of processor **[Borisenko**, 2010]

2.2 Efficiency

Efficiency is an important performance metric in parallel computing. It is defined as the achieved fraction of total potential parallel processing gain. It estimates how well the processors are used in solving the problem**.**

$$E_p = \frac{S_p}{p} = \frac{T_1}{pT_p} \qquad (2)$$

where Ep is the efficiency.

2.3 Scalability

Usually, when the number of processors increases in a system, speedup and efficiency will drop because of the associated overheads. Scalability measures the capacity to effectively utilize an increasing number of processors.
We can predict the scalability of the system by modeling systems. One such example is the SCALA system [**Sun, 2002**]. SCALA (SCALability Analyser) is a framework for developing performance modeling and prediction systems.

3. Benchmarks

For assessing the performance of a system we can use benchmarks. Three types of benchmarks exist:
1- Synthetic benchmarks - artificial programs, which copy the functionality of real programs. Examples include Hyper Pi, Sis oft Sandra, wPrime.

2- Kernel benchmarks - code fragments extracted from real programs. Examples: LINPACK, PARKBENCH

3- Real application benchmarks actual real-life applications. Examples: gcc, modern video games, AutoCad.

The disadvantages of synthetic benchmarks are that they are easy to optimize for. A system architect might optimize his/her system to provide good results in a synthetic benchmark. Kernel and real application benchmarks tend to give better evaluations compared to synthetic.

Challenges associated in creating benchmarks are the variety of sources needed to be tested, i.e. processor, memory, etc. Finally, selecting a suitable workload is a challenge in benchmark design

Some practical examples of benchmarks include:

- SPEC suite - includes a wide range of benchmarks for various aspects, including: testing integer arithmetic, php/jsp performance, smtp/pop performance. A collection of real application benchmarks.
- DMIPS - a synthetic benchmark program. Used to evaluate soft-core CPUs.
- PARSEC - a suite of real application benchmarks composed of multithreaded programs.

Some developments on benchmarks can be found also in ***[Gallois, 2013] [Gallois, 2014]***

III- MPI message Passing Interface

The Message Passing Interface (MPI) is a library of functions for exchanging data between compute nodes of distributed memory architecture (or shared but the interest in this case is minimal). Use these functions in C or FORTRAN is very often in calculations on the clusters. Calculations nodes are generally processors interconnected by a bus.

When running parallel code the desired number of processors is specified by the user. The program then assigns a number to each processor which is preserved throughout the execution of the program. The two main functions to exchange data are MPI_Send and MPI_Recv functions. To understand how these functions work, you can make an analogy with the mails

The MPI_Send function allows the processor that executes send a message (mail) , that contains data (the text of the email), the size of the data (the size of the mall) , the type of data (HTML or plain text for e-mail) and the recipient of this message (email address) processors. When this function is executed, the message is sent but the recipient processor must receive this message and open (as in mail) .

For this, the processor must receive the message performs the MPI_Recv function specifying: the data they wish to store (the text of the email), the size of the data (the size of the email) , the type of data (HTML or plain text) and the sender of the message (which is the sender of the email he wants to open)

The use of MPI_Send and MPI_Recv functions is blocking for processors that use them. As sending or receiving a message is not completed, the processor does not execute the rest of the program. To avoid this, MPI_ISend MPI_Irecv are used, and that they are not blocking functions. However, the use of these functions can be misleading if they are not all performed before handling the data they contain.

Different processors generally read the same program. To specify which task should execute a particular processor conditions "if" type conditions are used in specifying how the number of processor should execute the block.

One of the difficulties still remaining in the use of MPI is the management of inputs / outputs. In particular read in parallel from different parts of a file .Txt or .Bin is not yet possible. However, the MPI community is very active and offers regular changes in the library.

IV- Parallel algorithms

Complex systems are made of connected sub-systems including various dynamics and behaviors. The multirate methods [**Gear&Well, 1984**], (**Skelboe, 1989**) take into account the dynamics diversity of these sub-systems. Therefore, to simulate correctly the behavior of complex systems, the sub-systems the components of which vary quickly are integrated with a small time step and however, the slower components are integrated with larger time steps. If the smallest time step is used for all the rates, a too long execution time is needed and could be avoided. Moreover, it is necessary to interpolate the values of the components integrated with large time steps to have the values of all components at each smaller time steps in order to have a global solution regularly sampled.

In [**Bartel, 2012**], some multirating considerations can be found. He considers several levels of latency exploitation, multi rate step size, multi order and multi method implementation of schemes. This method needs also to gather the sub-systems with quick dynamics in an identified sub set and to separate them off the slower dynamics sub-systems gathered inside another sub set. This splitting is not an obvious task.

Such methods seem not to be still used in the industrial applications but can be implemented in platform as Xmod when the modeler knows what sub-systems are quick and what sub-systems are slow.

The most famous algorithm for parallel strategy is the Wave Form Relaxation algorithm. The WFR methods were originally introduced by [**Lelarasmee, 1992**]. [**Ruehli-Sangiovanni-Vincentelli, 1982**] made the first studies of time domain analysis of large scale problems arising from the modeling of the integrated circuits.

The idea to make partitions is also the basis of the Wave Form Relaxation Method [**Lelarasmee, Ruehli, & Sangiovanni-Vincentelli, 1982**], [**White, Sangiovanni-Vincentelli, Odeh, Ruehli, 1985**]. A waveform is the set of the values of the state variables of a system or a sub-system on a given time laps.

Méthodology : The Waveform Relaxation method is an iterative process. The whole system is split

into several sub-systems more or less coupled and integrated separately on a given and common time laps. The time steps can be different for the sub-systems, mainly if they are very different rates. Each sub-system is featured by its own wave form.

Let be a system made of 2 sub-systems 1 and 2, of waveforms x_1 and x_2 satisfying to the following conditions:

$$\frac{dx_1}{dt} = f_1(x_1, x_2, t), x_1(0) = x_{10} \tag{3}$$

$$\frac{dx_2}{dt} = f_2(x_1, x_2, t), x_2(0) = x_{20} \tag{4}$$

The purpose is to find the waveform of a sub-system, solution of the corresponding equation; the wave form of a subsystem is computed by integrating on the state vector components considered in the current wave and by considering the other components as frozen. The frozen components are updated at each relaxation and the iterations are managed until the difference between two waves is less than a given threshold.

Two methods exist:

- Gauss Seidel method: solve (3) and then (4) in setting the waveform x_2 and then x_1 equal to the found solutions at the previous iteration.
- Gauss Jacobi method: solve (3) and (4) in setting simultaneously the waveforms x_1 and x_2 equal to the found solutions at the previous iteration.

[**Vachoux, 1984**] were the first to notice the slow convergence of this method in case of strong couplings between sub-systems. Elegant theories for linear problems have been developed by Miekkala and Nevanlinna in [**Miekkala, 1987**]. **[White, 1985]** study the contraction properties of various classes of numerical methods used in WFR implementation.

Recall of the Contraction Mapping Theorem:

Let Y be a Banach space and $F : Y \to Y$. If F is such that $\|F(y) - F(x)\| \leq \gamma \|y - x\|$ for $\forall x, \ \forall y \in Y$ for some $\gamma \in [0,1]$, then F has a unique fixed point $\tilde{y}$ such that $F(\tilde{y}) = \tilde{y}$. Furthermore, for any initial guess $\forall y^0 \in Y$, the sequence $(y^k \in Y)$ generated by the fixed point algorithm $y^k = F(y^{k-1})$ converges uniformly to $\tilde{y}$.

In other words, the mapping F must be lipschitzian with a Lipschitz constant positive and smaller than 1. If this condition is not fulfilled, the fixed point algorithm does not converge and another method as a Newton-Raphson method has to be implemented. Indeed when $\gamma \leq 1$, it is clear that F mapping realizes a contraction of the interval $\frac{\|F(y) - F(x)\|}{\|y - x\|} \leq \gamma$.

[**Crow, 1994**] formulates an explicit problem with Ordinary Differential Equation/Differential Algebraic Equation. This problem is said *Cauchy problem* where DAE have been added

$$\begin{cases} X(t=t_0)=X_0 & X\in R^n, \quad \mathrm{I}=[\mathrm{t}_0,\mathrm{t}_{\mathrm{end}}]\subset \mathrm{R} \\ \mathrm{Y}(\mathrm{t}=\mathrm{t}_0)=\mathrm{Y}_0 & X\in R^m \\ \dot{X}(t)=F(X(t),Y(t),t) & F:R^n\times R^m\times I\to R^n \\ 0=G(X(t),Y(t),t) & \mathrm{Y}\in \mathrm{R}^{\mathrm{m}},\ G:R^n\times R^m\times R\times I\to \mathrm{R}^{\mathrm{m}} \end{cases} \tag{5}$$

The system of dimension n is split into r sub-systems such as $n=\sum_{i=1}^{r} n_i$

$$\begin{aligned} &\begin{cases} \dot{x}_1(t)=F_1(,) & x_1(t=0)=x_{10} \\ 0=G_1(,) & y_1(t=0)=y_{10} \end{cases} \\ &\qquad\qquad \dots \\ &\begin{cases} \dot{x}_r(t)=F_r(,) & x_r(t=0)=x_{r0} \\ 0=G_r(,) & y_r(t=0)=y_{r0} \end{cases} \end{aligned} \tag{6}$$

The generalized Cauchy-Lipschitz or Picard-Lindelöf theorem states the existence and the unicity of the solution of this problem and then shows that this problem is well-posed.

Theorem: *let* $I=[t_1,t_2]$ *a compact interval on R with t1<t2, f a continuous mapping of* $I\times R^n\times R^n \to R^n$ *lipschitzian with respect to the second variable, that is to say*

$$\exists L\in R^+,\forall t\in[t_1,t_2],\forall(v,w)\in R^n\times R^n,\left|f(t,v)-f(t,w)\right|\leq L\left|v-w\right|$$

Then, $\forall t_0\in[t_1,t_2],\ \forall v_0\in R^n,\ \exists u\in C^1(I,R^n)\, solution\ of\ Cauchy\ problem.$

*Definition of a discrete or continuous wave [***Chapon, 2010***]:* set of values or continuous functions representing the state components of a system or of a sub-system on a given time interval.

The Wave Form Relaxation algorithm is an iterative process starting from initial conditions; it can include a numerical scheme or an analytical process for ODE time integration and DAE zero crossing solving:

```
$k=0$
$initial\ \ guess \Rightarrow \begin{cases} x^{k+1}(t=0)=x_0^{k+1} \\ y^{k+1}(t=0)=y_0^{k+1} \end{cases}$
$repeat\ \{k++$
 $foreach\,(i++,i=1,r)\quad integrate\ and\ solve\ \ on\ [0,T]$
$\dot{x}_i^{k+1}(t)=F_i\left(x_1^k,..,x_i^k,..,x_r^k,y_1^k,..,y_i^k,..,y_r^k,t\right);\quad x_i^{k+1}(t=0)=x_i(0)$
$0=G_i\left(x_1^k,..,x_i^k,..,x_r^k,y_1^k,..,y_i^k,..,y_r^k,t\right);\qquad y_i^{k+1}(t=0)=y_0(0)$
   $\}\ until\ \ \left(\left\|x^{k+1}-x^k\right\|\leq\varepsilon_x\quad \left\|y^{k+1}-y^k\right\|\leq\varepsilon_y\right)$
```

This Gauss-Seidel WFR algorithm has to integrate the Cauchy problem on [0, T] and solved simultaneously the algebraic equations for all the components as long as convergence is not reached. It is well known that this algorithm converges slowly because of the couplings between the r subsystems. Naturally, if the r subsystems are fully decoupled, WFR algorithm converges within only one iteration.

It can be found interesting references in [**Crow,1994**] but many papers are focused on this algorithm.

The conditions of convergence are discussed in [**Crow, 1994**] and they are illustrated by means of small sensitive examples. They can be very coarsely resumed by the following properties: if F and G are Lipschitzian, the ODE/DAE problem is well posed.

A major advantage of the WFR method is to allow a separated solving of the sub-systems. Therefore, in case of time step adapted according to the stiffness of each sub-system, it appears different time steps in the sub-system. After the exchange of the wave emitted by each sub-system, it becomes obvious the discretization points have to be interpolated. This topic has to be solved with care to avoid a divergence of the algorithm. The theorem 3.2 of [**Crow, 1994**] states that WFR converges if linear interpolation is carried out; this result needs to be discussed with respect to the index of the DAE.

In his thesis, Jan Janssen **[Janssen, 1997]** examines several accelerations of the WFR method.

- Firstly, he examines the method of **Successive Over-Relaxation** which is a variant of the Gauss-Seidel method for solving a linear system of equations, resulting in faster convergence. A similar method can be used for a slowly converging iterative process as WFR. This method consists of splitting the matrix in diagonal matrix and strictly inferior and superior triangular matrices. Studied by [**Miekkala, 1987**], the convergence of this improving technique turned out to be disappointing.

- The polynomial acceleration and preconditioning do not provide good results as it could be hoped, said [**Janssen, 1997**]. This method consists of taking into account, not only the waves themselves, but a linear combination of the waveform iterates –see [**Luk, 1996**] and [**Lumsdaine, 1992**].

- Successive Over Relaxation Wave which [**Miekkala, 1987**] said rather disappointing, is a natural extension of the standard SOR procedure. Convolution SOR Wave Relaxation seems to have more potential [**Riechelt, 1995**]. In this approach, the linear combination of the wave iterates is replaced by a time convolution of theses iterates.

- Multi grid WFR approach consists of working with a fine grid smoothing and a coarse grid correction approach. The initial conditions are projected on the fine grid where some relaxation iteration are managed and the problem is continued on the coarse grid before to be projected back and ended on the fine grid

1. Illustration of the WFR algorithm in continuous formulation by means of a simple example

Let us consider the following system of ODE and the initial conditions with $x_i \in \Re,\ i=1,2$ considered on $t \in [0,T] \subset \Re$

$$\begin{cases} \dot{x}_1 = x_2 & \mathrm{x}_1(t=0)=0 \\ \dot{x}_2 = -x_1 & \mathrm{x}_2(t=0)=1 \end{cases} \tag{7}$$

The analytical solution is $\begin{cases} x_1 = \sin(t) \\ x_2 = \cos(t) \end{cases}$

The first step of the WFR algorithm consists to take the initial values as initial and constant solution Then, the calculations are performed on x1 and x2 separately, mimicking 2 processors with distributed memories; each of the two components is calculated on interval [0, T] independently of the other one. Thirdly, the solution is exchanged with the other component seen as another processor. This calculation is iteratively repeated until the wave update is less than a threshold given at the beginning. The analytical successive solutions of each iterate of the WFR algorithm are summed up in the following table:

iter	analytical solution $x_1(t)$		iter	analytical solution $x_2(t)$
0	$x_1(t)=0$		0, 1	$x_2(t)=1$
1, 2	$x_1(t)=t$		2, 3	$x_2(t)=-\frac{t^2}{2}+1$
3, 4	$x_1(t)=-\frac{t^3}{6}+t$		4, 5	$x_2(t)=\frac{t^4}{24}-\frac{t^2}{2}+1$
5, 6	$x_1(t)=\frac{t^5}{120}-\frac{t^3}{6}+t$		6, 7	$x_2(t)=-\frac{t^6}{720}+\frac{t^4}{24}-\frac{t^2}{2}+1$
7, 8	$x_1(t)=-\frac{t^7}{5040}+\frac{t^5}{120}-\frac{t^3}{6}+t$		8, 9	$x_2(t)=\frac{t^8}{40320}-\frac{t^6}{720}+\frac{t^4}{24}-\frac{t^2}{2}+1$
9, 10	$x_1=\frac{t^9}{362880}-\frac{t^7}{5040}+\frac{t^5}{120}-\frac{t^3}{6}+t$		10, 11	$x_2=-\frac{t^{10}}{3628800}+\frac{t^8}{40320}-\frac{t^6}{720}+\frac{t^4}{24}-\frac{t^2}{2}+1$

figure 3 displays the time variations of the successive solutions on interval $t \in [0,2\pi]$. On the figure 3, the black solid curve represents the exact sinus and the chopped curves the WFR iterates; it is clear that from one iteration to the following, the WFR solution approaches better the exact solution. At the 18th iteration, the convergence is reached; it is clear that the convergence is difficult to get and the WFR algorithm is high cost. The research topic of interest is: how to accelerate the WFR algorithm?

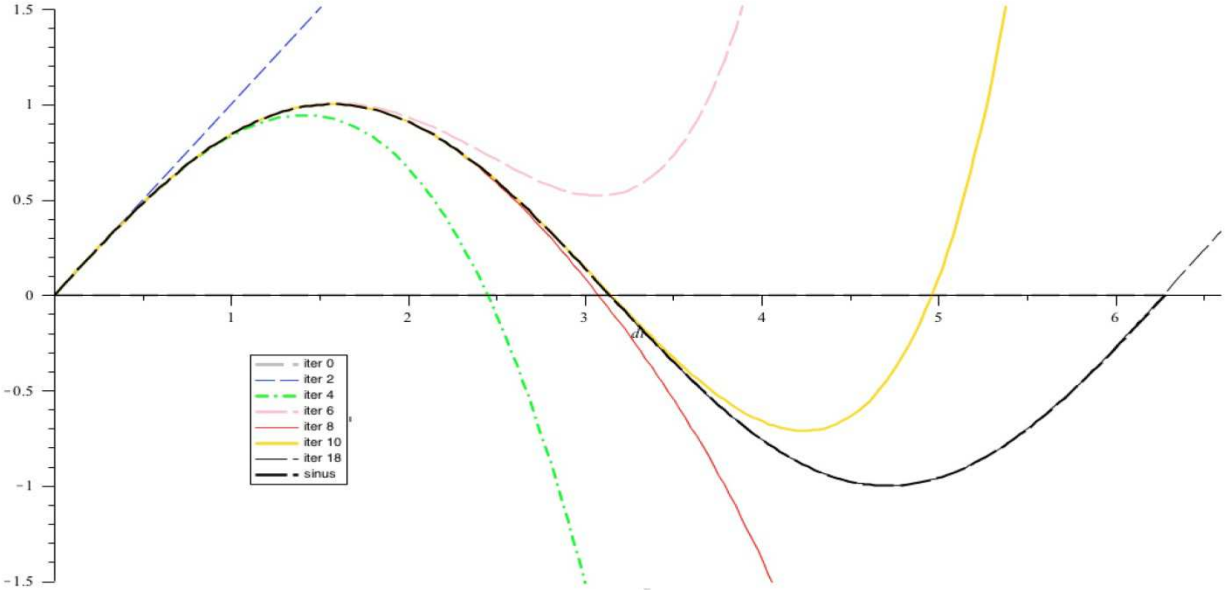

Figure 3 Successive iterates of the WFR algorithm inside the interval [0, 2π].

The 18th iterate follows exactly the exact solution of sinus With this simple example, it has been possible to describe the WFR algorithm behavior which is iterative and slowly converging.

2. Improvement of the WFR algorithm by means of windowing.

2.1 Case of continuous systems.

It is possible to state that the convergence $c = \frac{X^{n+1}}{X^{n}}$ of a WFR method in case of linear problem (Picard problem) is proportional to the length of the interval such that $c \cong e^{-T}$. Therefore, longer the time interval is, slower the convergence is. This topic has been pointed out by means of the previous simple example in continuous formulation. A similar behavior has been observed for nonlinear examples. Then, the idea of windowing springs very clearly. This is a way to improve the convergence in splitting into a series of windows the whole time interval and in iterating the process on each window. The change of window is done by setting new initial condition taken equal to solution at the end of the previous window.

On the following figure 4, the time windowing is illustrated. The whole interval $I = [0,2\pi]$ is split into 2 sub intervals $I_1 = [0,\pi]$ and $I_2 = [\pi,2\pi]$. A WFR iterative algorithm is managed into the interval I1 and the last values of I1 are set as initial conditions for the interval I2. The WFR algorithm is launched a second time in this last interval I2. An excellent result is got after 10 iterations (instead of 18 on the whole interval) on each interval with the junction point corresponds to $t = \pi$. We remark that $e^{-T} < e^{-\frac{T}{2}}$ proving the better convergence when splitting into 2 windows. This result with a continuous formulation of the WFR method can be translated for discretized formulations when a time integration scheme has been chosen. But some questions stays until now without general response: what is the optimal windowing? Optimal means the window which ensures the quickest convergence. It seems the optimal window is model dependent. Thus, a search of the optimal window on such model is still to do.

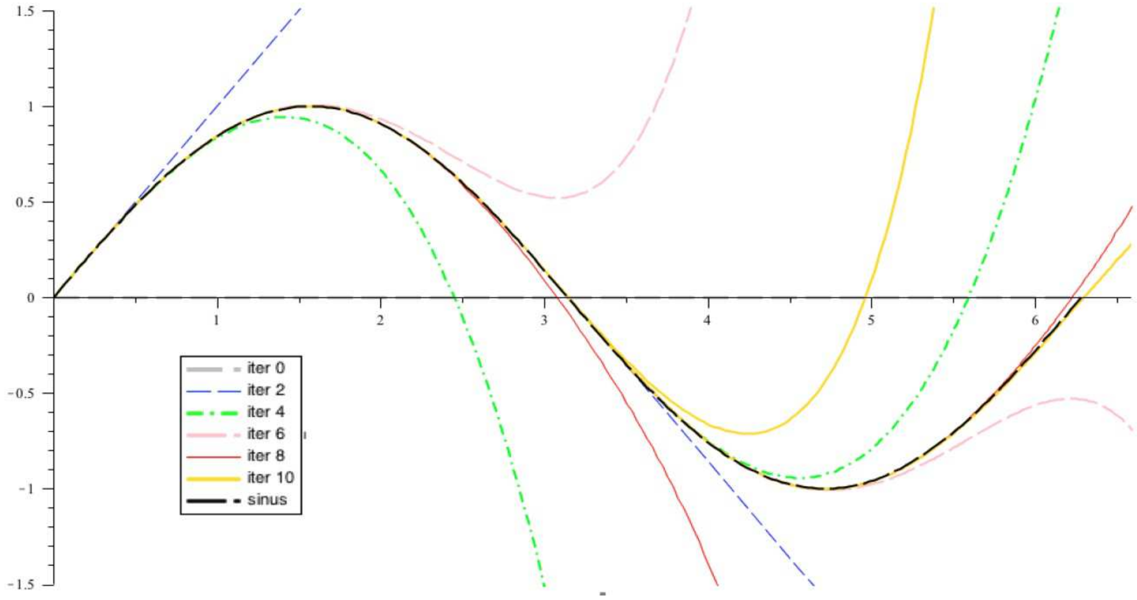

Figure 4 Successive iterates of the WFR algorithm inside 2 windows: [0, π], [π, 2π].

The whole interval [0,2p] has been cut into 2 windows [0, π], [π, 2 π].

The iterates of WFR within each window are represented by curves of same color. The convergence is reached after 10 iterates

2.2 Case of hybrid systems.

Although the WFR method has been originally created for continuous time, it can be also used in case of system where continuous and discrete times are blended. The events are a hard difficulty for the WFR method which can be used only between two events. If the events take place at known time, it is easy to adapt the windowing according to these recurrent events. However, in case of unpredictable event, it is necessary to detect them inside the WFR run and to adapt the current window according to the event detection. Such difficulty is not obvious and needs care since the event detection changes from WFR iteration to the following to become more accurate. An accurate detection of event is very important for the global precision of the solution since the state at the detected event location provides the initial conditions for the following window.

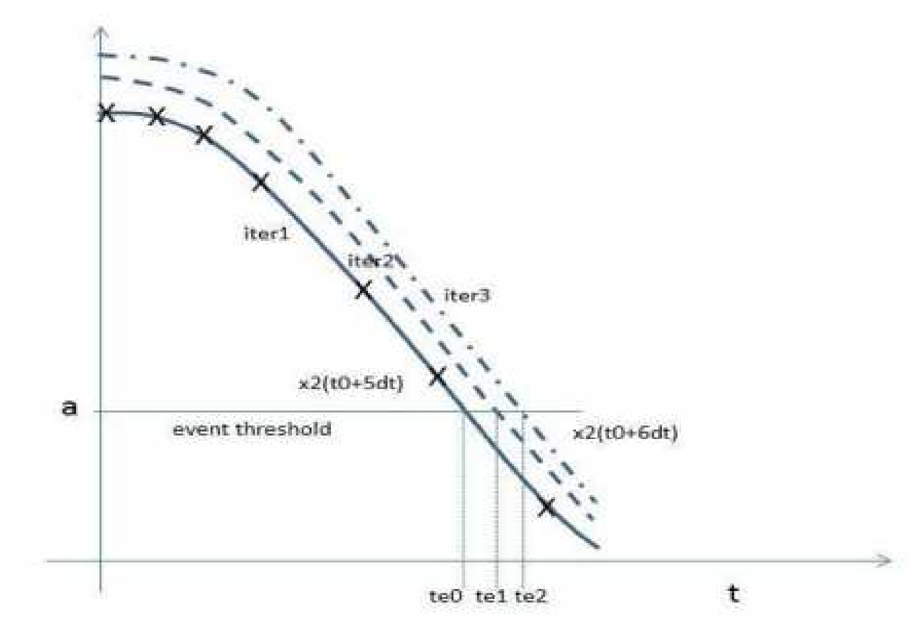

Figure 5 Estimation of time te of an event during the iterations.

The considering event occurs when the trajectory crosses the threshold a.
The event computed at the first iteration provides a time te1 ranging between x2(t0+5dt) and x2(t0+6dt). This first estimation of x2 is get on a not converged trajectory. That is why we compute again this trajectory after one wave exchange and detect again the same event at time te2 and its corresponding location. Since the estimations are contracting, the time te* is finally obtained after a sufficient number of iterations. Therefore, it is clear that the event has to be recomputed at each wave exchange.

2.3 Convergence of the WFR method

WFR method is subject to convergence condition depending on the solving method used. For instance, when using a fix point method, there is a convergence condition based on the contraction mapping as described in [**Burrage 1996**]. T is the time horizon, k le number of iteration and C the Lipschitz condition, it becomes that the convergence rate K has to be such that

$$K^k = \frac{X^{n+1}}{X^n} \leq \frac{|CT|^k}{k!} \qquad (8)$$

It is clear, in particular, that the criterium depends on the length of the time horizon. In splitting the whole horizon in smaller pieces, convergence can get within each piece although the convergence on the whole interval is not possible. [**Bartel, 2013**] also analyses the convergence in terms of Lipschitz constant of decoupled sub-systems.

[**Jansen, 1994**] searches for convergence condition of WFR applied to first and second order ODE equations; the solver is based on Euler backward scheme. He applies the WFR method to heat conduction in 1D and 2D.

In [**Jiang, 2004**], the results are obtained by the contraction mapping principle on Banach spaces with weighted norms and by the use of the Perron–Frobenius theory of nonnegative and non reducible matrices. It is demonstrated that waveform relaxation methods are convergent faster than the classical Picard iterations. In [**Jiang-Chen, 2001**] a Krylov method is used to split the system into decoupled sub-systems and then accelerate the convergence of the WFR algorithm.

[**Geiser, 2011**] proposes a splitting of the matrix (only linear case) based on the LU decomposition. He analyses the error due to the splitting method; a 2 equation example and an experiment on 2D advection-diffusion model illustrates the implementation of the method. For the last experiment, the analytical solution is known.

3. Parallelization of the WFR method and partition.

The parallelization consists of splitting a large problem into pieces and to dial the pieces of work to cores and/or processors. First of all, the problem has to be large enough to make necessary the parallelization. Indeed, hidden inside cores and processors, there are cache memories and the implementation of these cache is complex and developed at several levels. Moreover, the use of several cores or processors generates communication which consumes time. Therefore, the code must generates more calculation than communication and not too much saturated the caches to be well parallelized in the cluster.

As the global problem is split into pieces, it is obvious that the better splitting is the splitting which ensures that the pieces are uncoupled as much as possible. The communications are made of messages including the wave as defined for the WFR method. The wave becomes 0 if the sub-systems are fully uncoupled. The following figure 6 displays a splitting in case of 2 processors fit out three cores each. The array is then split between 6x6 sub blocks, 6 sub blocks on the diagonal and 30 outside the diagonal. In general, this array is almost sparse but the convergence speed is tied up with the extra diagonal sub block fulfilling.

The array is split into 4 sub blocks, 2 on diagonal, 2 out of diagonal. The out diagonal sub blocks are frozen during the update of state vector since they are not in shared memories between the 2 processors. Each of diagonal sub blocks work in shared memory and has the knowledge of the half component of the state vector at each time step.

The core number 2 of the processor 1 updates only the corresponding components of the state vector (1/6 of the components) but however needs the values of all other components as indicated by the red rectangle. This core knows at each time step the value of the first part of the state vector; the second part is frozen and updated by means of wave exchanges between processors inside a time window.

Therefore, the core working with shared memory, the couplings existing inside the sub blocks of their own processor are taken into account easily whereas the other couplings of the outside diagonal sub blocks penalize heavily the convergence.

3.1 DSM method

It exists many methods to split a system of ODE into sub-systems. To explicit the main idea of the ideal splitting, we consider a linear system represented by the matrix A such that $\dot{X} = AX$. It is possible to link a dependencies array of $\dot{X}$ depending of X . The purpose is to transform the dependencies array in order to have a number of diagonal blocks equal to the number of cores (or processors) and to have terms equal zero in the non-diagonal blocks. The only way to do that is to change the basis of the state vector by transformation and reordering of the components.

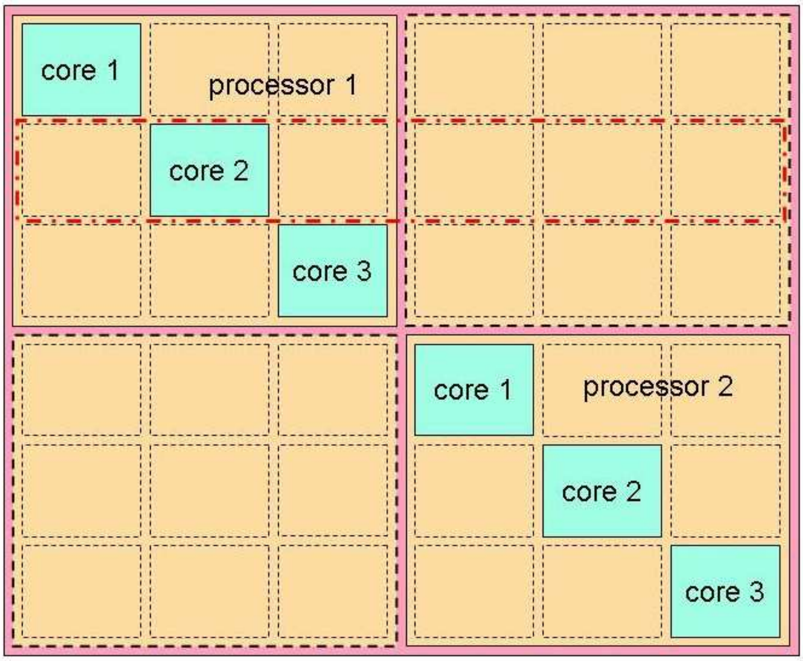

Figure 6 Dependencies array of $\dot{X}$ depending of X .

The array has been split into 4 sub blocks corresponding to 2 processors and each sub blocks is still split into 9 sub blocks corresponding to 3 cores per processor.

The Design Structure Matrix Method is well known and applied in a large spectrum of applications. It consists of starting with a square array of dependencies of the state vector components between them, that-is-to-say, $X(X)$; this problem is different of the previous one where $\dot{X}(X)$. Obviously, the diagonal of this DSM array is fulfilled. The interest is focused on production diagram, organization architecture, process architecture. It is also classical to associate a graph to the dependencies array. The transformation of the array takes advantage of the graph methods. For stability care, it is important that the graph does not contain algebraic loops which can be easily point out on the graph and the methods of opening a loop become useful.

An example about "Automobile climate control system" is given by [**Tyson, 2001**] and illustrated in the following figure 7. The raw and initial dependencies array is on the left and shows the couplings between the different sub-systems of the global climate system (components listed per lines on the left). On the right part of the figure, the new order of the sub-systems (listed on the left part of the right figure) allows to gather all the couplings inside 3 diagonal blocks with overlap. Thus, all state variables specific to each subsystem can be integrated separately of the integration process of the other sub-systems.

Generally speaking, such a beautiful result is not reachable and it remains some couplings after all the efforts performed to attain this ideal target. The general result is only a re-organization which minimizes the couplings of the blocks without a whole uncoupling and facilitate as much as possible the convergence of methods such the WFR algorithm.

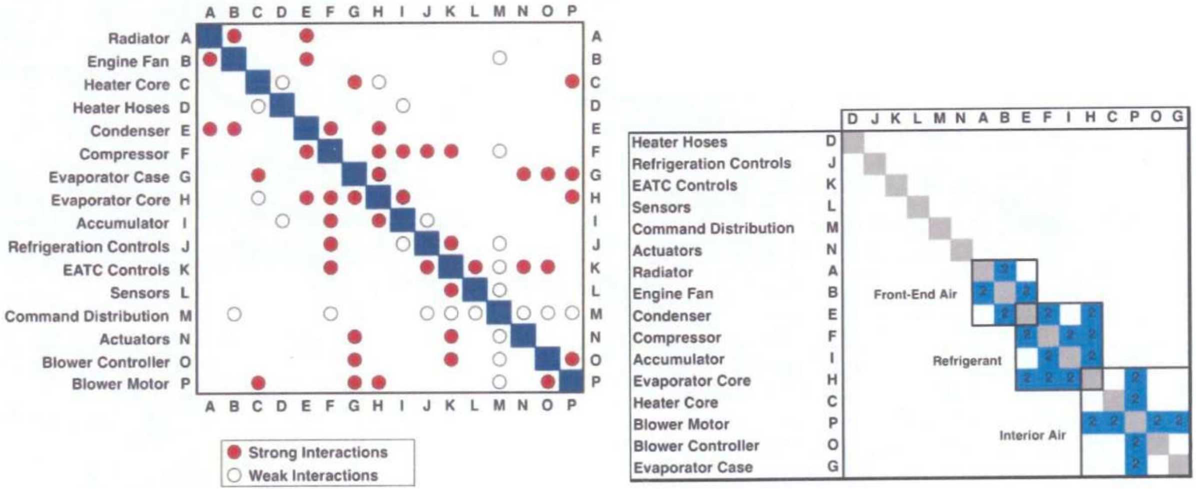

Figure 7 DSM example about automobile climate control system

That is why many works aim to transform the system in order to have r sub-systems as uncoupled as possible to accelerate the convergence of WFR method.

Sparse matrix

In the linear algebra context, matrices are involved and very often, they are partially fulfilled with non-zero terms. The matrices can be symmetric; we recall the definition of a defined and positive matrix:

Definition: a Matrix is said defined and positive if and only if all its eigen values are strictly positive.

Moreover, we also recall a **theorem**:

If a matrix A is defined and positive, then it admits a Cholesky factorization , that is to say, it exists a lower triangular matrix L the diagonal terms of which are positive and such that A=LL^t

We consider two sparse matrix formats well fitted to represent matrix including less than 30% of non zero terms. Therefore, all the elementary floating point operation including zero are avoided

- Compressed sparse row: 3 arrays are used to represent the sparse matrix. **Array element** contains the non zero terms, row per row. The array size is equal to the number of none zero terms. **Array_index_col** includes the column number of each non zero elements. Its size is equal to the Array_element size. **Array_nze_row** indicates the cumulated number of none zero terms for each row. Its size is equal to the matrix dimension.
- Compressed sparse column: same format as previously with row replaced by column. This format is described by column whereas the previous one is described by row.

These three arrays are currently organized on structure shape in C language. Consequently, some arithmetical basic operation as addition or multiplication has to be implemented. Obviously, a vector is represented as a matrix of dimension 1. Finally, it can be remarked that the transposition of the matrix consists of translate from a row format to column format and conversely.

In the Brice Boyer thesis [**Boyer, 2012**], there is an interesting topic about the product of a sparse matrix with a dense vector. The number of operations is equal to the number of non zero terms of the sparse matrix. However, the accesses are random and it is difficult to manage efficiently the cache. Some links with the LINBOX are indicated in this thesis.

IF the matrix is dense, the sparse matrix format requires (2*n*n+n) memory cells whereas the requirement in matrix format is only (n*n). Therefore, as soon as the sparsity reachs 50%, the sparse format costs more than the matrix format. It is currently admitted that the upper limit of the efficiency of the sparse format is around 30% of sparsity in terms of storage.

3.2 Partitioning

When solving a problem on parallel context, the first point necessary to look after, is to decide how to subdivide and map data into processors and how to implement a good mapping. There are different ways to partition a computational domain. The underlying goal of partitioner is to achieve a good load balance of the computational load among the processors as well as to ensure that the ratio of communications to computation is small for the given task.
METIS [**Karypis, 1999**] and SCOTCH [**Chevalier, 2008**] are two popular and widely used packages; they present very good performance keeping good quality of partition (mostly in terms of data balance). Both of them implement the recursive-bisection and k-way algorithms.

Let suppose a general sparse linear system whose adjacency graph is G= (V,E). The ***k*-way graph** partitioning can be defined as follows: for the given a graph G with $|V| = n$, partitions V into *k* subsets, $V_1, V_2, \ldots, V_k$ such that for $i \neq j, V_i \cap V_j = 0;\ |V_i| = n/k$ and $uV_i = V$,
and the number of edges of E whose incident vertices belong to different subset is minimized. A ***k*-way partition** of V is commonly represented by a partition vector P of length *n*, such that for every vertex $v \in V$, $P(v)$ is an integer ranging between 1 and *k* and indicating the partition at which vertex v belongs.

4. MUMPS: a parallel sparse direct solver for sparse matrix graph

MUMPS is a long term project whose objectives developed by universities of Toulouse, Bordeaux and Lyon with the aim to provide theoretical and algorithmic approaches in order to find solutions of large linear systems (Ax = b) with symmetric positive definite matrices. The matrix A involved can be general symmetric matrix or unsymmetrical matrix.

A version of the code for complex arithmetic is also available.

It involves a process of parallel factorization of matrix and solves phases. Partial factorization and Schur complement matrix (centralized or 2D block-cyclic) are taken into account

In order to control the accuracy of the process, an iterative refinement and backward error analysis is implemented. In practice, various matrix input formats are admitted

- Assembled format;
- Distributed assembled format;
- Elemental format.

The system provides interfaces for languages as FORTRAN, C, Matlab and Scilab. It has been built to be also several orderings interfaced : METIS as an example etc.... [**Agulo, 2008**] [**Lexecellent, 2012**]

Conclusion

The WFR methods replace the integration of a complex system by several integrations of sub-systems opportunely chosen of smaller size. The division of the whole system into several sub-systems is a delicate operation; it can be seen as a partition but it is also possible to have the pieces in overlapping. The time integration can be done in one piece on the whole time interval or divided into time sub intervals. The optimal division seems to be problem dependent. When the problem is hybrid, it is valuable to fit the window on the detected events by window resizing.

The WFR method parallelization is currently managed by means of Open MP directives or MPI instructions. The cores and processors are used simultaneously to provide memory and computation capacity and hybrid implementation using both Open MP and MPI instructions are currently in use.

VI- Distributed simulation

A general overview of real time distributed simulation can be found in [**Ben Khaled , 2014**]. One the important approach cited is HLA.

The HLA was adopted as service for the distributed systems 1,0 of simulation by the group of management of object (OMG) in November, 1998 and updated in 2001 to reflect the changes resulting from the commercial calibration of the specifications under the IEEE.HLA was approved as standard opened by the institute of the electric engineers and Electronics engineers (IEEE) - the standard 1516 of IEEE - in September, 2000. In November, 2000 the services and the common staff signed the identifying memorandum HLA as architecture preferred for the interoperability of simulation in the DOD.

These two specifications (HLA 1.3 and IEEE on 1516) which coexist are incompatible, because of the important brought modifications. The initial principles of HLA are kept, certain services are added, and others are simplified. A synthesis of the differences between HLA 1.3 and HLA IEEE 1516 is presented in **[Lightner, 2000]**

General architecture of HLA

HLA is based on the hypothesis that no simulation can satisfy all the needs. The objective of this architecture is triple:

- Facilitate the re-use of elementary simulators
- Facilitate the interoperability between distributed simulators
- Reduce the costs of modeling and simulation

A global simulation HLA is called federation, and a simulator participating in it is called federal. A federal can also receive outside data corresponding to a real object or a human being. Another essential constituent of a simulation HLA which is added is the RTI (Run-time Infrastructure). It constitutes an IT implementation of the specifications of the interface of programming.

It is thus a question of a set of software offering common services in a set of federated, possibly distributed, and participating in the same federation. This set behaves as a distributed operating system.

Every exchange of data between federal has to pass by the RTI.

Some definitions are necessary for the continuation of the presentation of HLA: -

Federation: A federation is a group having a federated object model common; it is the representation of a set of inter-operating simulators.

Federated: A federated is a member of an HLA federation; it is the representation each elementary simulator

RTI (Run-Time Infrastructure) is a computer implementation of HLA interface specifications. This is a process ensuring communication between federated of same federation, providing services through a HLA API14

The communication is divided into two parts: the RTIambassador and FederateAmbassador. The RTIambassador is an interface allowing the federated to send messages (RTI requests or information transmission).

Its API is fully defined in different languages and a specification describes how implementation should behave when a message is received. The FederateAmbassador has a defined and standardized interface, specific for the application and its implementation is the responsibility of the developer of the Federated.

HLA specifications

The HLA architecture is described by specifications : [IEEE 1516] composed by a set of rules defining the responsibilities of federal and federation, a federated object-oriented modeling (OMT) and interface specifications of programming interface (API) (Application Program Interface).

Rules

The U.S. Department has identified ten rules that govern all simulations HLA. Compliance is essential to the success of a simulation. five first concerns the overall functioning of federation

Rule 1: The federations must have a Federation Object Model (FOM: Federation Object Model) compliant with the HLA OMT. The FOM must describe all data exchanged during the execution of a federation, and must indicate the conditions of trade.

Rule 2: The representations of objects associated with the simulation must be in Federated and not in the RTI.

Rule 3: During the execution of the federation, the data described in the FOM cannot be traded between Federated that through the RTI

Rule 4: During the execution of the federation, Federated must interact with the RTI according to the specification of the HLA interface.

Rule 5: During the execution of the federation, an instance attribute of an object may be the property of a single federated at any moment.

The last five rules concerned with Federated.

Rule 6: The Federated must have a simulation object model (SOM: Simulation Object Model) according to the HLA OMT. The SOM corresponds to information which might be made public during the execution of the federation.

Rule 7: Federated must update and take into account changes attributes of objects, as well as send and/or receive interactions accordance with the specification of their SOM

Rule 8: Federated must be able to accept the transfer or ownership of attributes dynamically during the execution of a federation, in accordance requirements specified in their SOM.

Rule 9: The Federated must be able to vary the conditions under which they provide updates of attributes of objects, in accordance with SOM their specifications

Rule 10: The Federated need to be able to check the local time of manner that will enable them to coordinate data exchange with other members of the federation

At this point we will take a quick overview to get an idea of the scope of the rules. Federation rules establish basic rules to create a federation, including documentation requirements (Rule 1), object representation (Rule 2), data exchange (Rule 3), interface conditions (Rule 4) property attributes (Rule 5)

The rules cover the Federated documentation (Rule 6) control and transfer of appropriate object attributes (rules 7, 8 and 9), and time management (Rule 10).

VII Sensitivity analysis

1- Definitions

Sensitivity Analysis (SA) is the study of how the uncertainty in the output of a mathematical model or system (numerical or otherwise) can be apportioned to different sources of uncertainty in its inputs.[1] A related practice is uncertainty analysis, which has a greater focus on uncertainty quantification and propagation of uncertainty. Ideally, uncertainty and sensitivity analysis should be run in tandem.

Sensitivity analysis can be useful for a range of purposes including:

- Testing the robustness of the results of a model or system in the presence of uncertainty.
- Increased understanding of the relationships between input and output variables in a system or model.
- Uncertainty reduction: identifying model inputs that cause significant uncertainty in the output and should therefore is the focus of attention if the robustness is to be increased (perhaps by further research).
- Searching for errors in the model (by encountering unexpected relationships between inputs and outputs).
- Model simplification – fixing model inputs that have no effect on the output, or identifying and removing redundant parts of the model structure.
- Enhancing communication from modelers to decision makers (e.g. by making recommendations more credible, understandable, compelling or persuasive).
- Finding regions in the space of input factors for which the model output is either maximum or minimum or meets some optimum criterion.

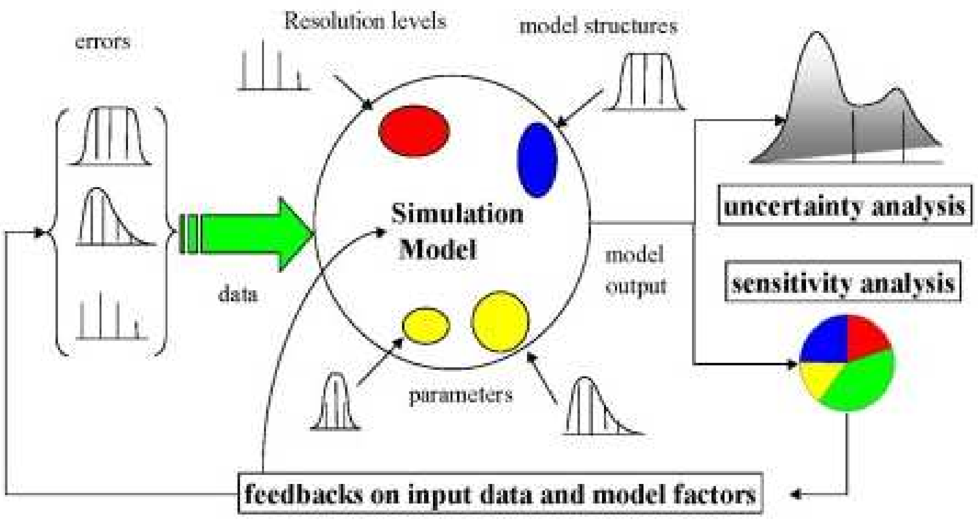

Figure 8 The use of uncertainty analysis and sensitivity analysis in simulation

Structuring the different uses can lead to these groups;

Decision Making or Development of Recommendations for Decision Makers

Testing the robustness of an optimal solution.

Identifying critical values, thresholds or break-even values where the optimal strategy changes.

Identifying sensitive or important variables.

Investigating sub-optimal solutions.

Developing flexible recommendations which depend on circumstances.

Comparing the values of simple and complex decision strategies.

Assessing the "riskiness" of a strategy or scenario.

Communication

Making recommendations more credible, understandable, compelling or persuasive.

Allowing decision makers to select assumptions.

Conveying lack of commitment to any single strategy.

Increased Understanding or Quantification of the System

Estimating relationships between input and output variables.

Understanding relationships between input and output variables.

Developing hypotheses for testing

Model Development

Testing the model for validity or accuracy.

Searching for errors in the model.

Simplifying the model.

Calibrating the model.

Coping with poor or missing data.

Prioritizing acquisition of information.

[Saltelli, 1997] gives below a condensed list of reasons why and (and instances where) SA should be considered.

1. In control problems, SA may help to identify critical regions in the space of the input parameters.

2. In screening exercises, SA may help to locate a few influential parameters in systems with hundreds of uncertain inputs.

3. Variance-based SA techniques are useful to ascertain whether a subset of input parameters may account for (most of) the output variance.

4. Point (3) above may be used for mechanism reduction (dropping or fixing non-relevant parts of the model) and for model lumping (building/extracting a model from a more complex one). The problem of model ‘relevance' is important: are the parameters in the model input set relevant to the task of the model?

5. Point (3) above may also be used for model identification, by pinpointing the experimental conditions for which your ability to discriminate among models is maximum.

6. As in (5) above, SA may be used for model calibration, to ascertain whether the experiments with their related uncertainties will allow parameter estimation. Useful especially for ill-conditioned problems.

7. SA may be coupled to optimization/search algorithms; by identifying the most important parameters, SA may allow the dimensionality the space where the search is made to be reduced.

8. As a quality assurance tool, to make sure that the dependence of the output from the input parameters in the model has a physical resemblance and explanation.

9. To solve an inverse problem, i.e., as a tool to extract parameters embedded into models, whose output does not correlate easily with the unknown input (e.g., in chemical kinetics, to extract kinetic constants of complex systems from measured yield rate of components); see also functional sensitivities

10. To ascertain on a quantitative basis what fraction of my prediction uncertainty is due to parametric uncertainty and how much to structural uncertainty.

11. More specific applications exist, such as SA based training of neural networks (instead of the back-propagation algorithm), SA-based system to couple atmospheric transport of gaseous species with their chemical reaction Uses of sensitivity analysis

2- Stochastic methods

2.1 Design of Experiments DoE

Statically variation of the response) and identify the interactions of factors that have a significant influence on the response .The goal is to know why the answer varies.

- Studies of response surface: calculus of variations depending on the factors and influential interactions (determined by screening or screening). The goal is to know how the answer varies.

This is a parameter that evolves independently of the time but not a law of continuous or hybrid behavior. This method requires a large number of actual tests which is not applicable in our case study because in fact, the variable factors are components of the mechatronic system is not available experimental designs but an evolution operation of the system under its variations.

The method of design experiment is a static method; it does not cover dynamic models.

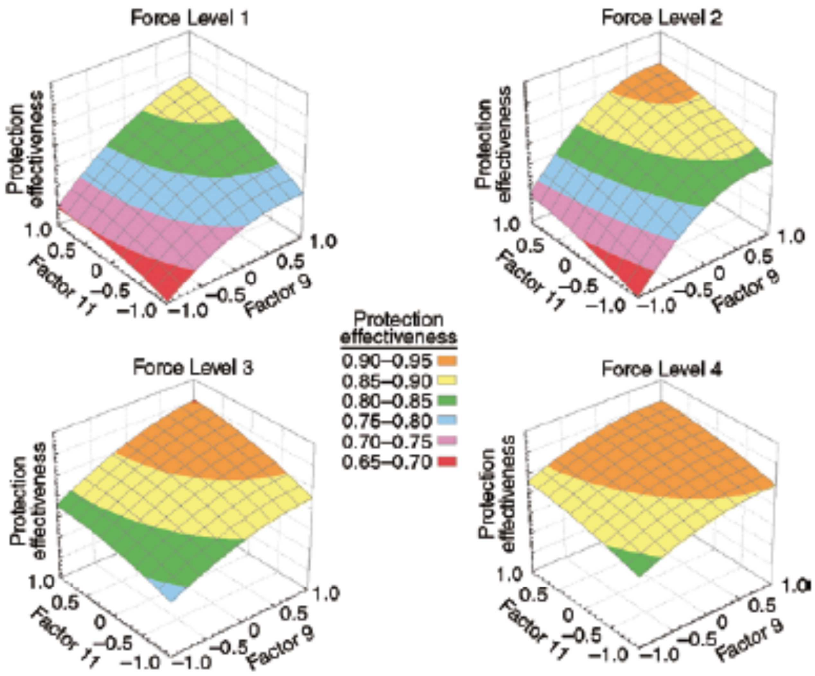

Figure 9 Example of continuous response surface obtained through a DOE analysis

3- Deterministic methods

3.1 Interval computations

The calculation interval is a special case calculation sets and set theory provides us the basics of interval analysis [**Le Bars 2011**].

Interval definition:

A real interval [x] is a subset of connected R. In the case of open intervals, one must also keep the notation [x].

The lower limit of an interval [x] is defined as denoted by:

$$\underline{x} \overset{\Delta}{=} \sup\{a \in R \cup \{-\infty,\infty\} \mid \forall x \in [x], a \le x\} \qquad (9)$$

The upper limit of the range noted is defined by:

$$\overline{x} \overset{\Delta}{=} \inf\{b \in R \cup \{-\infty,\infty\} \mid \forall x \in [x], x \le b\} \qquad (10)$$

The center or middle of a non-empty interval [x] is defined by:

$$mid(x)=\frac{\underline{x}+\overline{x}}{2} \tag{11}$$

Definition of a tube

Tube [x] (t) with a time step δ> 0 is a constant vector function intervals [kδ, kδ + δ]. The box [kδ, kδ + δ] [x] is called the kth slice of the tube [x] (t) and will be denoted by [x] (k). We can see a tube as a union of slices.

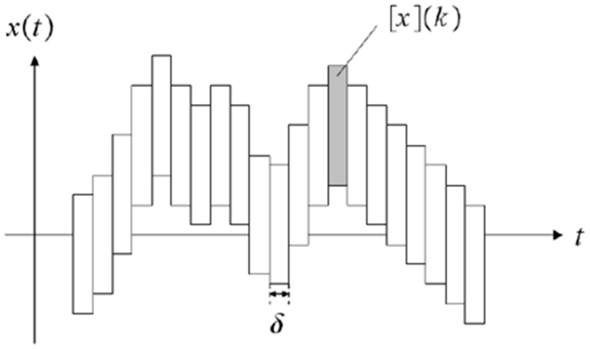

Figure 10 A tube interval [x] (t) with no time δ (t) [**Le Bars 2011**]

Contraction and spread

The propagation of constraints intervals (ICP: Interval Constraint Propagation) is a technique that uses a combination of two more general concepts: constraint propagation and interval analysis to solve equations containers intervals.

The purpose of the propagation method is to minimize the areas containing the desired while keeping the solution variables. This can be done using the concept of contractor defined in [**Jaulin and Chabaret, 2008**] and [**Chabaret and Jaulin, 2009**].

The contraction is the action of contractor to reduce the size of a field containing a variable. The contractor is an algorithm constructed from a constraint to contract the variable domains involved in this constraint.

Bissection Algorithm

The bisection algorithm is to perform a search tree where each node is a range of variables representing the current field. A test over this interval is performed via interval arithmetic. If the test is false, then it will be for all points of the interval, and can be therefore suppressed. If it is true then it is also true on all points of the interval and can be stored in the list of solutions [**Chabert, 2007**] and [**Le Bars, 2011**]. In the case where the test information for points of the interval is false and it is true for other points of the interval then it is cut into two sub- intervals. We talk about bisection (branch) for cutting sub intervals and evaluation (bound) for the test. The result of this algorithm is in the form of 2D drawings sub- flooring solution, sub- flooring solution and not sub- flooring undetermined.

There are various algorithms such as bisection SIVIA (for the basic algorithm) and other optimized and used in combination with existing or contractors as 3BCID STRANGLE . Among the software used, one find PROJ2D QSOLVE .

The interval analysis method has several advantages. You can have polynomial algorithms to compute precise intervals containing all possible values of the random variables through the process of propagation. The propagation methods do not require that the equations are linear. They exhibit robustness with respect to outliers, inconsistencies. When the model is no longer valid (for example, if the data error is Gaussian unbounded) propagation methods intervals return an empty set .

Methods set intervals are used in different areas in the automation and robotics , as state estimation and parameter [**Raissi et al. 2004**] the control [**Lydoire and Wrist , 2003**], [**Vinas , et al. 2006**] , the location of robots [**Meizel et al. 1996**], the SLAM (Simultaneous Localization And Mapping) in the case of rolling land robots [**Drocourt et al. , 2005**] , loop detection in a path of submarine **[Aubry et al robot . 2011**] , topological configuration space **[Delaoune et al analysis. 2006**] .

However, the methods intervals also have disadvantages: the overestimation (or conservatism) errors which have a major drawback which corresponds to having very large intervals that give inaccurate information to be usable [**Jaulin et al, 2001**]. Another disadvantage is that the computation time can be occasionally exponential relative to the dimension of the blocks (interval vectors) if bisection methods are used.

In our research we focus on parametric variations and their influence on the system behavior. The interval analysis method performs a tessellation of the state space. It allows the characterization of all solutions in the form of a sub- paving. Parts of the space incompatible with research measures are excluded. And a sub- paving solution obtained guarantees include all the solutions of the problem.

This approach frames the solution, while the methods of differential inclusion and impulse differential inclusion give exactly the reachable area that encompasses all possible solutions of the system.

3.2 Differential inclusions

The operation of a mechatronic system depends on the behavior of its components. A common way to model these systems is to use differential equations. But it should be noted that infinitesimal change in one component can significantly affect the overall system performance. Because if a parameter of the physical part of the system then has an uncertainty factor then the corresponding coefficient in the differential equation also varies in an interval. A well- appropriate mathematical tools for modeling such systems is the differential inclusion [**Aubin, 1984**] .

Differential Inclusions (DI) represent a generalization of differential equations with variable parameters. The solution given by the differential inclusions is «reachable area «instead of a single path. This concept is well known in the field of control and more specifically in the optimal control but is not yet used in the modeling and simulation [**Raczynski , 2006**].

The general form of ID is as follows:

$$\frac{dx}{dt} \in F(t, x(t)), \qquad x(0) \in X_0 \tag{12}$$

With $x \in R^n$ and F an application R x R^n and in R^n portions of and X_0 the initial set.

To show the relationship between differential inclusions and control domain it suffices to consider the following dynamic system:

$$\frac{dx}{dt} = f(t, x(t), u(t)), \quad x(0) = x_0, \quad u(t) \in C(t, x(t)) \tag{13}$$

With u (t) control variable, and the application F is defined as follows:

$$F(t, x(t)) = \left\{ z = f(t, x, v) \,\middle|\, v \in C(t, x) \right\} \tag{14}$$

v is a variable belonging to a range of variation C. This variable can be either a control or a parameter of the system state.

Raczynski used this theoretical concept for the simulation with v corresponds to a control variable of the system. In the general case, the area C is the variation range of variation of the control. This concept is known in the field of control and more specifically in the field of optimal control.

One might think that the achievable field solution by the method of ID could be obtained by simple extension of the algorithm for solving differential equations, but this is not the case.

Several applications can be solved using differential inclusions, such as the problems of uncertainty and differential games. For modeling this type of problem we need to have the density function in the state area for a given time interval and the results will be achieved by having some probabilistic properties of the input function. While in the real case we have only imprecise or wrong input data, which makes simulation impossible or gives totally wrong results.

In "primitive" methods, time is discretized and points are selected in the field F at each step time with a uniform density distribution, this method makes it difficult to estimate the shape of the attainable field. While the method used by Raczynski for the simulation of dynamic systems using differential inclusions, is to generate the contour of the achievable field.

Indeed, the solution of the ID is a field in space - time state. This is a set that includes all the possible trajectories of the system. However, finding the limits of the achievable field is not obvious.

3.3 Impulse differential inclusions

Impulse differential inclusions represent a hybrid model that contains a continuous evolution and discrete evolution [**Aubin , 2002**].

Definition 1: The hybrid time trajectory is a finite or infinite sequence of intervals of real lines as

- For i < N $I_i = [\tau_i, \tau_i^{'}]$

- If N < ∞ then $I_N = [\tau_N, \tau_N^{'}]$

- For all i, $\tau_i \leq \tau_i^{'} = \tau_{i+1}$

τ$_i$ is the time of the realization of the discrete transition .

Definition 2 : The differential impulse inclusion is a collection H (X , M, R , J) with :

- X: space vector of finite dimension

- F: X ----> 2x differential inclusion

- R: X ----> 2^x

- $J \subseteq X$ All forced transitions.

Definition 3 : The operation of a differential inclusion : H (X , F, R , J) is a pair (τ , x) which consists of a hybrid time trajectory and a mapping x : τ ---> X which satisfies :

- A discrete evolution for all x (τi +1) ∈ R (x (τ'i))

- A continuous evolution. If $\tau_i < \tau'_i$ x (.) is a solution of the impulse differential inclusion $\dot{x} \in F(x)$ on the interval $[\tau_i, \tau_i']$starting with $x(\tau_i)$ $x(t) \notin J \;\; \forall \;\; t \in [\tau_i, \tau'_i[$ T

Under sets hypothesis not developed here, continuous evolution is possible because either x is in J, or it is forced to enter J throughout solutions differential inclusions. Therefore, the transition is possible.

Impulse differential inclusions are an extension of differential inclusions and discrete time systems in a Finite Dimension Space Vector Gone.

A differential inclusion $\dot{x} \in F(x)$ can be considered as a differential impulse inclusion = H (X , M, R , J) with $R(x) = \Phi \ \forall x \in X \quad et \quad J = \Phi$.

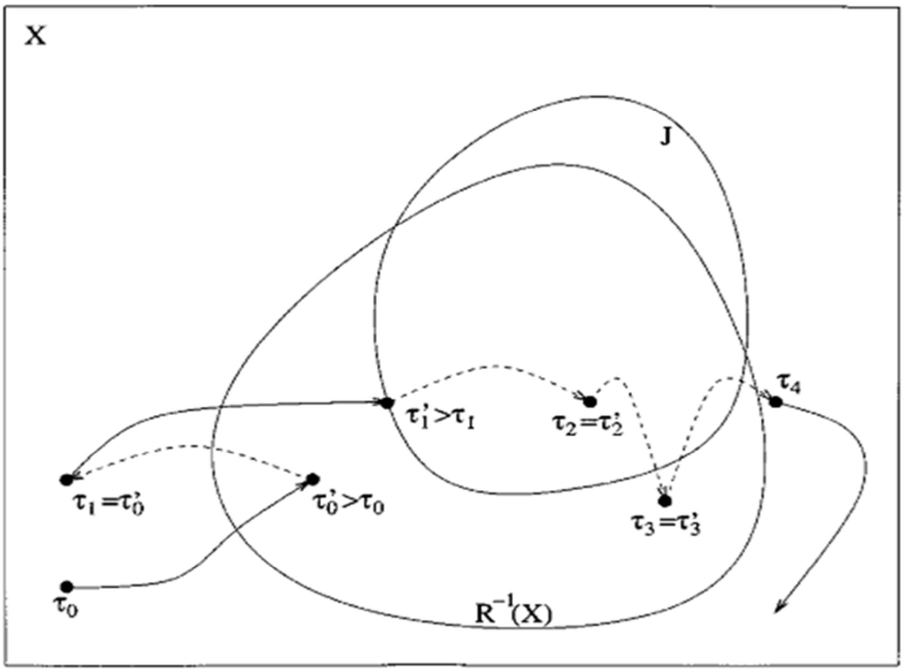

Figure 11 Example of a run of an impulse differential inclusion

Some developments and algorithms can be also found in **[Zerelli 2014]** .

4- Modelica Implementation specificities

Openturns

In industrial applications can be found uncertain data (supplied by the sensors, for example) . These data are not measures and parameters. They can significantly influence the overall functioning of the system and affect its performance, accuracy. For this study these uncertainties is important. EDF is one of those manufacturers who deal with the problems of uncertainty to improve its existing plants (increase lifespan at 60 years, for example) and build new plants.

EDF and IFPEN (French Petroleum Institute and New Energies) are working on the development of a library for modeling plants and systems in Modelica [Website Modelica] in order to create an environment for modeling and study uncertainties. In [**Bouskella D , and all 2011**] we present different techniques for the treatment of uncertainties :

- Reconciliation of data: what is to minimize the uncertainty interval variable (a variable find conciliation physical condition that comes closest to the actual value of the physical condition) .

- The propagation of uncertainties: the source propagate uncertainties through the model of the system. EDF R & D and have presented a method consists of four steps detailed in [Global uncertainty methodology of study]

- Step A: Identification of sources of uncertainty and specification of case study y = h (x, d) with :

• y: vector of variables

• h: the model

• x: vector of input variables which we will study the uncertainty

• d: vector of input variables of some

By asking a deterministic test (determining the area of possible currently by determining the minimum and maximum variables y) and a probability criterion (random vector y is) .

- Step B: quantification of sources of uncertainty, the deterministic criterion is that the potential area of each element of x must be determined. The probabilistic criterion is to find the dependence between the different elements of X.

- Step C : the propagation of uncertainty . Deterministic criterion is to determine the range of variation of y specifying its minimum and maximum , if h is monotonic in x then it is easy if you must resort to optimization methods. The probabilistic criterion is to propose methods of approximation and discretization methods for estimating (eg robust sampling) .

- Step D: to classify the sources of uncertainty in a deterministic and probabilistic criterion . The tool used for simulation is OPEN TURN.

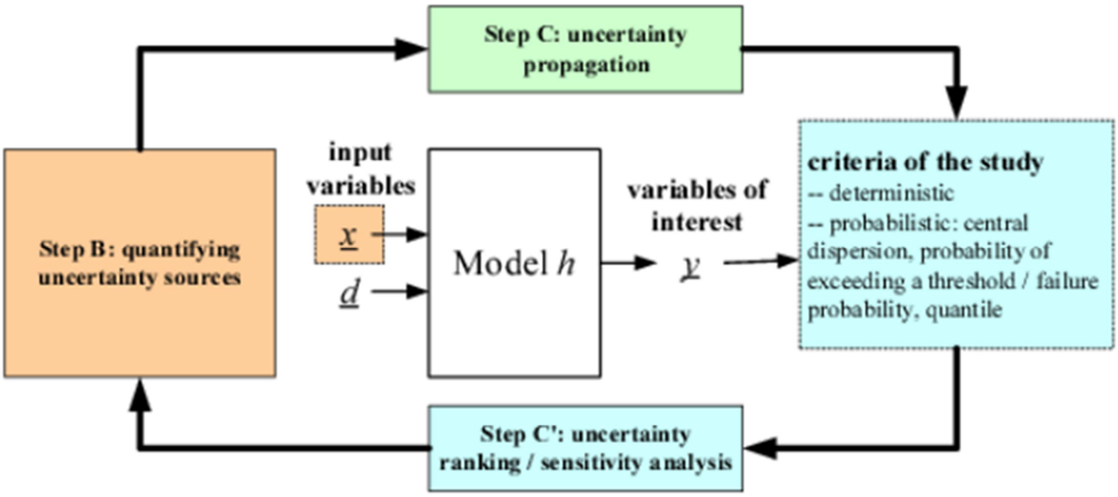
Step C: uncertainty propagation
Step B: quantifying uncertainty sources
input variables
x
d
Model h
variables of interest
y
criteria of the study
-- deterministic
-- probabilistic: central dispersion, probability of exceeding a threshold / failure probability, quantile
Step C': uncertainty ranking / sensitivity analysis

Figure 12 The different steps of the method of propagation of uncertainties and developed by EDF

For the study of complex systems IFPEN uncertainty (French Institute of Petroleum and Energy News) uses methods such as kriging . This spatial interpolation method for a linear estimation based on mathematical expectation and variance of the data. Indeed, the kriging is based on the calculation, interpretation and modeling of the variogram , which is an assessment of the variance depending on the distance between data. This method eliminates implausible values (e.g. in the case of weather temperatures "absurd" is removed) [**Gratton , 2002**].

In [Bouskella D , and all 2011] there is also a proposal by EDF to extend the Modelica language for studying uncertainties.

Introducing different methods of probabilistic distribution.

This method presented by EDF is a statistical method that involves using the laws of probability. It has the advantage of creating a block extension of the Modelica language to model a complex system while taking into account variations. The disadvantage of this method is the lack of development on the model h as it does not allow to enter the parametric variations. Another disadvantage is that this method is not deterministic and can be neglected values have a significant influence on the system behavior. It is for these reasons that we will not develop this tool benefit in this thesis.

Conclusion

The aim of this work is a state of the art for the study and analysis of complex systems with varying parameters, in order to finally simulate them in a parallel or distributed environment.

The notion of hybrid system is first presented.

Then in order to give a frame of evaluation for parallel computation, some important definitions are given in relationship with metrics and benchmarks.

For parallel simulation, different approaches have been found and especially the Waveform Relaxation algorithm WFR and it is showed how to use it for hybrid large scale systems using windowing and partitioning.

For distributed simulation High Level architecture (HLA) is one of the solutions. Its specifications are presented; the set of rules that define the responsibilities of federal and federation, a federated object-oriented modeling (OMT) and also an interface specifications of programming interface (API) (Application Program Interface) are detailed.

It can be seen that there are still research activities at three levels for the acceleration of parallelization of simulation;

- The mathematical model (graph, matrix, hybrid) organization

- The numerical schema

- The efficient use of different cores or distributed environments

The sensibility analysis is finally introduced and some tools to deal with it for as stochastic method, differential inclusions for continuous system, and impulse differential inclusion for hybrid systems with varying parameters. This part is ended by introducing an industrial tool using OpenTurns and Modelica.

References

[Agulo 2008] Méthodes directes hors-mémoire (out-of-core) pour la résolution de systèmes linéaires creux de grande taille (MUMPS) PhD thesis de l'Université de Lyon - École Normale Supérieure de Lyon France Emmanuel Agullo

[Aubin, 1984] J.P Aubin, A. Cellina, . "*Differential Inclusions, Set-Valued Maps And Viability Theory"*. Grundl.der Math. Wiss.. 264**. Berlin: Springer, 1984**

[Aubin, 2002] J. P Aubin., J. Lygeros ,M. Quincampoix, S. Sastry S. & N. Seube « Impulse Differential Inclusions : A Viability Approach to Hybrid Systems", IEEE Transactions on Automatic Control, 47, 2-20,2002

[Aubry et al., 2011] Aubry, C., Desmare, R., and Jaulin, L. (2011). "*Loop detection in a mobile robot trajectory using interval analysis"*. In SWIM 2011, Bourges, France.

[Bartel, 2012] Andreas Bartel, **Michael Günther**, **Developments in multirating for coupled systems, ZAMM, Editor: Holm Altenbach , ISSN: 1521-4001, section9-25, S739, 2012**

[Bartel, 2013] Andreas Bartel, Markus Brunk, Michael Günther, and Sebastian Schöps, **Dynamic Iteration for Coupled Problems of Electric Circuits and Distributed Devices. SIAM Journal on Scientific Computing 35:2, B315-B335, 2013**

[Ben Khaled, 2014] Ben khaled el Feki Abir "Distributed real time simulation of numerical models: application to powertrain " PhD thesis , University of Grenoble France 2014

[Borisenko, 2010] Alexey Borisenko "Performance Evaluation in Parallel Systems"

[Bouskela , and al 2011] Bouskela D, Jardin A, Benjelloun-Touimi Z, Aronsson P, and Fritzson P. (2011) « Modelling of uncertainties with Modelica ». proceedings of the 8th Modelica conference, Drsden, Germany

[Boyer, 2012] Brice Boyer, Multiplication matricielle efficace et conception logicielle pour la bibliothèque de calcul exact LinBox, thèse Université de Grenoble, 21/06/2012

[Burrage, 1996] K. Burrage, C. Dyke, and B. Pohl. On the performance of parallel waveform relaxations for differential systems. J. Appl. Numer. Math., 20:39–55, 1996. (cited on pages 49 and 50)

[Chabert, 2007] Chabert, G. (2007), « *Techniques d'intervalles pour la résolution de systèmes d'équations ».* PhD dissertation, Université de Nice-Sophia Antipolis, Nice-Sophia Antipolis, France. Available at : www.emn.fr/z-info/gchabe08/.

[Chabert and Jaulin, 2009] Chabert, G. and Jaulin, L. (2009). "*Contractor Programming. Artificial Intelligence »*, 173 :1079.1100. WOS.

[Chapon, 2010] Chapon D, Etude de l'apport de l'ingénierie des modèles et des langages de modélisation équationnels orienté objets pour la définition de processus interopérable de conception système, thèse, Univ. Paris Sud XI, nov. 2010.

[Chevalier, 2008] Chevalier C, Pallegrini F, PT-SCOTCH F: a tool for efficient parallel graph ordering, Parallel Computing 6-8, 34 (2008), 318–331.

[Crow, 1994] Crow M L. «the wave relaxation for systems of differential/ algebric equation.» *Math, Comput, Modeling, Vol.19, n°12*, 1994: 67-84.

[Delanoue et al., 2006] Delanoue, N., Jaulin, L., and Cottenceau, B. (2006).*"Using interval arithmetic to prove that a set is path-connected.* Theoretical Computer Science, Special issue : Real Numbers and Computers," 351(1) :119.128.

[Drocourt et al., 2005] Drocourt, C., Delahoche, L., E. Brassart, B. M., and Clerentin, A. (2005) "*Incremental construction of the robots environmental map using interval analysis*". Global Optimization and Constraint Satisfaction : Second International Workshop, COCOS 2003, 3478 :127.141.

[Gear & Wells, 1984]Gear C, Wells D. «Multirate linear multistep methods.» *Bit Numerical Mathematics*, 1984: 484 - 502.

***[*Geiser, 2011] Junger Geiser, An iterative splitting method via waveform relaxation, International Journal of Computer Mathematics , Vol. 88, Issue 17, 2011**

[Gallois, 2013] A new method of event handling for ODE solvers and application to a benchmark TH. Gallois, J. Brac, Th. Soriano 27 th European Simulation and Modelling Conference ESM2013 23-25 October Lancaster UK, ed. EUROSIS , 2013

[Gallois, 2014] Event handling solver compared to Modelica DASSL under XMOD in a bouncing ball benchmark TH. Gallois, J. Brac, Th. Soriano Workshop OpenModelica 2014 3-5 February Linkôping Sweden, 2014

[Gallois T., 2014] A new method to build a stiff, flexible, large scale model as benchmark for ODE solvers. TH. Gallois, J. Brac, Th. Soriano 10TH AIMS (American Institute of Mathematical Science) Conference Madrid July 2014

[Gratton , 2002] Gratton Y., « *Krigeage : la méthode optimale d'interpolation spatiale* » (2002) les articles de l'Institut d'Analyse Géographique, ,**www.iag.asso**

[Hybinette, 2001] Hybinette M, Fujimoto R M "Cloning Parallel Simulation" ACM Transactions on Modeling and Computer Simulation, Vol. 11, No. 4, October 2001.

[Jansen, 1994] J. Jansen, R. Mattheij, M. Penders, W. Schilders, Stability and efficiency of waveform relaxation method, Computers Math. Applic., Vol. 28, n°. 1-3, pp. 153-166,1994

[**Janssen, 1997] Acceleration of waveform relaxation methods for linear and partial differential equations, thesis, Université Catholique de Louvain, dec 1997.**

[Jaulin and Chabert, 2008] Jaulin, L. and Chabert, G. (2008).QUIMPER : « *un langage de programmation pour le calcul ensembliste* » ; Application à l.automatique. In CIFA 2008, Bucharest, Romania.

[Jaulin L, et al, 2001] Jaulin, L., Kieffer, M., Didrit, O., and Walter, E. (2001) .*"Applied Interval Analysis, with Examples in Parameter and State Estimation, Robust Control and Robotics"*. Springer-Verlag, London, *2004*

[Jiang, 2004] YL. Jiang, A General Approach to Waveform Relaxation Solutions of Nonlinear Differential-Algebraic Equations: The Continuous-Time and Discrete-Time Cases. IEEE Transactions on Circuits and Systems I: Regular Papers 51:9, 1770-1780, 2004

[Jiang-Chen, 2001] Jiang, YL; Chen, RMM; Wing, O, Improving convergence performance of relaxation-based transient analysis by matrix splitting in circuit simulation. IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications 48:6, 769-780, 2001.

[Karpis, 1999] **Karypis G, Kumar V, A fast and highly quality multilevel scheme for partitioning irregular graphs. SIAM Journal on Scientific Computing 20, 1 (1999), 359–392.**

[Le Bars, 2011] Le Bars, F, (2011), « *Analyse par intervalles pour la localisation et la cartographie simultanées; Application à la robotique sous-marine* », université de Bretagne occidentale, France, thèse.

[**Lelarasmee, 1992**]**Lelarasmee E. «The waveform relaxation method for the time domain analysis of large non linear dynamical system.» *Ph .D Thesis Univ. Berkeley.* 1992**.

[Lelarasmee, 1982]Lelarasmee E, A Ruehli, A Sangiovanni-Vincentelli. «The waveform relaxation method for time-domain analysis of large scale integrated circuits. .» *IEEE transactions on computer aided design*, 1982. : 131-154

[Lexcellent 2012] Jean Yves Lexcellent Multifrontal Methods: Parallelism,(MUMPS) Memory Usage and Numerical Aspects Habitation à diriger de recherches Ecole Normale Supérieure de Lyon, France

[Lightner, 2000] G. M. Lightner, Robert Lutz, and Reed Little. "The IEEE HLA standards : The evolution from U.S. DoD HLA v1.3.", In Proceedings of the 2000 Fall Simulation Interoperability Workshop, 2000.

[**Luk, 1996] Luk W, Krylov's methods in function space for waveform relaxation, thesis, Division computer science and engeneering, chinese university of HongKong, june 1996.**

[Lumsdaine, 1992] Lumsdaine A, Theorical and practical aspects of parallel numerical algorithms for initial values problems with applications, thesis, Department of electronical engineering and computer science, Massachusets Institute of Technology, Cambridge, USA, Jan. 1992.

[Lydoire and Poignet, 2003] Lydoire, F. and Poignet, P. (2003) "*Nonlinear predictive control using constraint satisfaction*". In 2nd International Workshop on Global Constrained Optimization and Constraint Satisfaction (COCOS), pages 179.188.

[Meizel et al., 1996] Meizel, D., Preciado-Ruiz, A., and Halbwachs, E. (1996). "*Estimation of mobile robot localization: geometric approaches*". In Milanese, M., Norton, J., Piet-Lahanier, H., and Walter, E., editors, Bounding Approaches to System Identification, pages 463.489. Plenum Press, New York, NY.

[Miekkala, 1987] Miekkala .U, Nevanlinna .O. «convergence of dynamic iteration methods for initial value problems.» *Sian J Sci. Statist. Comput 8(4)*, 1987: 459-482.

[Raczynski , 2006] Raczynski S., « *Modeling and simulation: The Computer Science of Illusion"*, Wiley, England, p115-116.2006

[Raissi et al., 2004] Raissi, T., Ramdani, N., and Candau, Y. (2004) "*Set membership state and parameter estimation for systems described by nonlinear differential equations*." Automatica, 40 :1771.1777.

[Riechelt, 1995] Riechelt M, White J, Allen J. «optimal convolution SOR acceleration of waveform relaxation with application to parallel simulation of semiconductor devices.» *Siam J . Sci. Comput 16(5)*, 1995: 1137-1158.

[SALTELLI 1997] Guest editorial: The role of sensitivity analysis in the corroboration of models and its link to model structural and parametric uncertainty A. Saltelli, M. Scott Reliability Engineering and System Safety 57 (1997) 1-4) 1997 Elsevier Science Limited 0951-8320/97

[Skelboe, 1989]Skelboe, S. «stability propreties of backward differentiation multirate formulas.» *Applied Numerical Mathematics*, 1989: 151 - 160.

[Sun, 2002] Sun X, Fahringer T, Pantano M. «A performance system for scalable computing.» *international journal of high performance computing applications vol16 no4*, 2002: 357-370.

[Tyson, 2001] Tyson Browning, Applying the design structure matrix to system decomposition and integration problems: a review and new directions, Engineering Management, IEEE Transactions on, Volume: 48, Issue 3, Aug. 2001.

[Vachoux, 1984] Vachoux.C, Carlin .C. «On partionning for waveform relaxation time-domain analysis of VLSI circuit.» *Proc of int.Conf. on Circ. and Syst. Montreal*, 1984.

[Vinas et al., 2006] Vinas, P. H., Sainz, M. A., Vehi, J., and Jaulin, L. (2006)."*Quantified set inversion algorithm with applications to control"*. Reliable computing, 11(5) :369.382.

[White, 1985] White J, Sangiovanni-Vincentelli A, Odeh A,Ruehli A. «Waveform relaxation : theory and practice.» *transactions of the society for compute simulation*, 1985: 95- 133.

[Zerelli 2014] Manel Zerelli Systèmes mécatroniques à paramètres variables. Analyse du comportement et approche du tolérancement. PhD thesis Ecole Centrale de Paris / Supmeca

Complementary bibliography

[Agulo 2008] Méthodes directes hors-mémoire (out-of-core) pour la résolution de systèmes linéaires creux de grande taille (MUMPS) PhD thesis de l'Université de Lyon - École Normale Supérieure de Lyon France Emmanuel Agullo

[Arnold] M. Arnold. Stability of sequential modular time integration methods for coupled multibody system models. *J. Comput. Nonlin. Dyn.*, 5(3), 2010. (cited on pages 51, 122, and 155).

[Aubin, 2001] J. P Aubin, J. Lygeros, M. Quincampoix, S. Sastry S. & N. Seube "*Viability and invariance kernels for impulse differential inclusions*" Proceedings of the 40th IEEE conference on decision and control, 2001

[Aubin, 1991] J.P.Aubin."Viability Theory". Birkh¨auser, Boston, 1991.

[Aubin, 1990] J.-P. Aubin , H. Frankowska. "Set Valued Analysis". Birkh¨auser, Boston, 1990.

[Aubin, 2001 a] J P. Aubin, G. Haddad *"Detectability through Measurements under Impulse Differential Inclusions"* Cahier de la MES vol. a.2001, n.30, 2001 .

[Aubin, 2006] J.P Aubin, "*Boundary-Value Problems for Systems of Hamilton-Jacobi-Bellman Inclusions with Constraints*" SIAM Journal on Control and Optimization , SIAM J. Control Optim., 41(2), 425–456. (32 pages), 2006

[Aubib,1997] AUBIN J.-P., *Dynamic Economic Theory : a Viability Approach*, Springer-Verlag,1997.

[Aubin,1999] AUBIN J.-P., *Mutational and Morphological Analysis : Tools for Shape Regulationand Morphogenesis*, Birkhäuser, 1999.

[Alur, 2000] Alur, R .Henzinger. T A, Lafferriere. G, and .Pappas.G J, (2000) "*Discrete abstractions of hybrid systems*," *Proc. IEEE*, vol. 88, pp. 971–984, July 2000.

[Artstein, 1994] Artstein Z, "*First order approximations for differential inclusions*". Set-Valued Anal 2:2 17, 1994

[Asarina, 2007] Eugene Asarina, Gerardo Schneiderb, Sergio Yovinec "*Algorithmic analysis of polygonal hybrid systems, part I: Reachability"*, Theoretical Computer Science 379, pp 231–265, (2007)

[Asarin, 2008] Eugene Asarin,Gordon Pace, Gerardo Schneiderb, Sergio Yovinec "*Algorithmic analysis of polygonal hybrid systems, part II: Phase portrat and tools"*, Theoretical Computer Science 390, pp 1–26, (2008)

[Asarin, 2001] E. Asarin, G. Schneider, and S. Yovine. "*On decidability of the reachability problem of planar differential inclusions*".In HSCC'01.LNCS 2034, Springer, 2001.

[Asarin,2003] E. Asarin, G. Schneider, and S. Yovine "*Towards computing phase portraits of polygonal differential inclusions*" , Technical Report 2003-042, Department of Information Technology, Uppsala University, August 2003

[Alur,1999] Rajeev Alur, Sampath Kannan, Salvatore La Torre "Polyhedral flows in hybrid automata", *Hybrid Systems: Computation and Control, Proceedings of the Second International Conference*, (HSCC'99), Lecture Notes in Computer Science 1569, pages 5--18, Springer-Verlag, 1999.

[Barrios, 2012], **Jorge Barrios**, **Alain Piétrus**, **Gonzalo Joya**, **Aymée Marrero**, **Héctor de Arazoza**"A differential inclusion approach for modeling and analysis of dynamical systems under uncertainty", Soft Comput, DOI 10.1007/s00500-012-0889-2, Springer-Verlag , 2012

[Baniardalani, 2012] Sobhi Baniardalani, Javad Askari "A Geometric Method for Generating Discrete Trace Transition System of a Polyhedral Invariant Hybrid Automaton", *Intelligent Control and Automation*, vol 3, pp: 197-206 **,** 2012

[Bemporad, 2000] Alberto Bemporad, Giancarlo Ferrari-Trecate, Manfred Morari "Observability and Controllability of Piecewise Affine and Hybrid Systems", IEEE Transaction on Automatic Control, vol. 45, no. 10, october 2000

[Ben Abdessalem, 2011] Ben Abdessalem M A; (2011), « *Optimisation avec prise en compte des incertitudes dans la mise en forme par hydroformage ».* thèse*,* école doctorale SPMII, Rouen ,France, vol. 35, p. 261-293.1997

[Cardaliaguet, 1998] Cardaliaguet P. , Quincampix M ., Saint-Pierre P., « Set-Valued Numerical Analysis for Optimal Control and Differential Games », *Stochastic and Differential Games*, Birkhäuser, 1998.

[Chutinan, 2003] Alongkrit Chutinan , Bruce H. Krogh, Fellow, IEEE, "*Computational techniques for hybrid system verification"* IEEE Transaction on automatic control, vol. 48, no. 1, 2003

[Colin, 2008] « Chaotic dynamics in hybrid systems », Nonlinear Dynamics and Systems Theory, 8 (2) , 169–194, 2008

[Devictor, 2004] N. DEVICTOR "Advances in methods for uncertainity and sensitivity analysis" Proceedings of the workshop "Level 2 PSA and Severe Accident Management", OCDE/AEN/CSNI/WGRISK, Köln, Mars 2004.

[Dontchev, 1989] Dontchev A, Farkhi EM "*Error estimates for discretized differential inclusions*". Computing 41:349–358, 1989

[Dontchev, 1994] Dontchev A, Hager W "*Euler approximation to the feasible set*". Numer Funct Anal Optim 15:245–262, 1994

[Donchev, 2006] Tzanko Donchev « *Impulsive differential inclusion with constraints* » , Electronic Journal of Differential Equations, Vol. 2006, No. 66, pp. 1–12. (2006)

[Donchev, 2007] "*Fixed time impulsive differential inclusions*", Surveys in Mathematics and its Applications, Volume 2,pp: 1 – 9, 2007

[Doyen, 19 96] Doyen L., Gabay D., « Economie des ressources renouvellables et viabilité »,

[Duc, 1999] Duc .G, Font, S (1999) « *commande H∞ et μ-analyse des outils pour la robustesse* » Hermes Science publiacation , Paris.

[Es Sadek, 2009] M.Z Es Sadek *« Contribution à l'optimisation globale. Approche déterministe , stochastique et application »,* INSA de Rouen, 2009

[Fritzon, 2003] P. Fritzon., *"Principles of Object-Oriented Modeling and Simulation with Modelica 2.1"*,IEEE press, Wiley-interscience, 2003.

[Fritzon, 2007]P. Fritzon, al, "*OpenModelica System Documentation, Preliminary Draft, 2007-06-20 for OpenModelica 1.4.3"*, june 2007.

[Fritzon, 1998] P.Fritzon, V. Engelson, J. Gunnarsson *'An Integrated Modelica Environment for Modeling, Documentation and Simulation'* The 1998 Summer Computer Simulation Conference (SCSC '98) ; 1998

[Global methodology of an uncertainty study] site web : http://doc.openturns.org/openturns-latest/html/ReferenceGuide/cid2.xhtml

[Gorre, 19 96] Gorre A., « The "Montagnes Russes" Algorithm in mutational spaces », *Parametric Optimization IV*, , 1996, Springer-Verlag.

George Karypis, Vipin Kumar, Multilevel k-way partitioning scheme for irregular graph, Journal of parallel and distributed computing, Vol. 48, pp 96-129 article no PC971404, 1998.

[Gordon, 2006] Gordon J. Pace, Gerardo Schneider , "*Static analysis of SPDIs for state-space reduction*", Technical Report 336, Department of Informatics, University of Oslo, PO Box 1080 Blindern, N-0316 Oslo, Norway, April 2006

Grander, M. «, A waveform relaxation algorithm with overlapping splitting for reaction diffusion equations,.» *numerical linear algebra with applications*, 1993: 1-7.

[Hackl, 1993] Hackl G "*Numerical approximation of reachable sets and control sets*". Rand Comput Dyn 1(4):371–394, 1993

[Hadj amor, 2008] H J. Hadj-Amor « Contribution au prototypage virtuel de systèmesmécatroniques basé sur une architecture distribuée HLA. Expérimentation sous les environnements OpenModelica-OpenMASK », Thèse , Supméca Toulon, 2008

[Hadj amor, 2012] H J. Hadj-Amor, T. Soriano "A contribution for virtual prototyping virtuel of mechatronic systems based on HLA real time distributed high level architecture" *Journal of Computing and Information Science in Engineering JCISE* , vol. 12, no. 1, ed. ASME American Society of Mechanical Engineer, 2012

[Henzinger, and al 1995] T.A. Henzinger, P.W. Kopke, A. Puri, P. Varaiya, (1995) "*What's decidable about hybrid automata ?*",27th Annual ACM Symp. On Theory and Computing (STOCS),

[Henzinger, 1997] T.A. Henzinger, P.-H.Ho, and H. Wong-Toi. *"HyTech: A Model Checker for Hybrid Systems."*, Lecture Notes In Computer Science; Vol. 1254, Proceedings of the 9th International Conference on Computer Aided Verification, pp 460 – 463, 1997, ISBN:3-540-63166-6.

[Henzinger, 1997 a] Thomas A. Henzinger, Pei-Hsin Ho, Howard Wong-Toi "HyTech: A Model Checker for Hybrid Systems" software Tools Technol. Transfer, vol 1, numéro 1/2 , pp 110-122, 1997

[Hospital, 2012] F. Hospital '*Conception préliminaire des actionneurs électromécaniques basée sur les modèles : lois d'estimations et règles de conception pour la transmission de puissance mécanique'* Institut National des Sciences Appliquées de Toulouse (INSA de Toulouse), 2012.

[Ibn Taarit, 2010] Kaouther Ibn Taarit, « *Contribution à l'identification des systèmes à retards et d'une classe de systèmes hybrides* »,Thèse , LAGIS de l'Ecole Centrale de Lille , LACS de l'Ecole Nationale d'Ingénieurs de Tunis, 2010

[Jantschi, 2009] **L. Jantschi**, **S. D. Bolboaca** « *Distribution Fitting 2. Pearson-Fisher, Kolmogorov-Smirnov, Anderson-Darling, Wilks Shapiro, Cramer-von-Misses and Jarque-Bera statistics », 2009*

Jean-Yves L'Excellent, Habilitation thesis Ecole Normale Supérieure de Lyon, France, 2012

[Juarez and all, 2008] Z . JUÁREZ, B. DENIS, J-J. LESAGE « Réseaux d'automates hybrides à synchronisations typées pour la modélisation des Systèmes dynamiques hybrides » "Conférence Internationale Francophone d'Automatique, CIFA 2008, Bucarest : Roumanie (2008)"

[Kirkpatrick,1983] S. Kirkpatrick, C.D. Gelatt and M.P. Vecchi. "*Optimization by Simulated Annealing*". Science, May 1983

[Kofman] E. Kofman. Discrete event simulation of hybrid systems. *SIAM J. Sci. Comput.*, 25:1771–1797, 2004. (cited on page 41)

[Kucharavy, 2007] Kucharavy,D (2007) « *TRIZ instruments for forcasting : past, present and future»;* LGECO - design engineering laboratory INSA Starsbourg Graduate school of science and technology.

[Kurzhanski, 1992] Kurzhanski A, Valyi I "*Ellipsoidal techniques for dynamic systems: control synthesis for uncertain systems*." Dyn Control 2:87–111, 1992

[Koutsoukos, 2008] Xenofon D. Koutsoukos, *Senior Member, IEEE*, and Derek Riley"Computational methods for verification of stochastic hybrid systems",IEEE Transaction on systems, man, and cybernetics- Part A: system and human, vol. 38, no. 2, march 2008

[Kryszewski, 2006] Wojciech Kryszewski, Sławomir Plaskacz "Periodic solutions to impulsive differential inclusions with constraints", Nonlinear Analysis Vol, 65, pp: 1794–1804, 2006

[Lelarasmee] E. Lelarasmee, A. E. Ruehli, and A. L. Sangiovanni-Vincentelli. The waveform relaxation method for time-domain analysis of large scale integrated circuits. *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst*, 1(3):131–145, Jul 1982. (cited on page 49)

[Lexcellent 2012] Jean Yves Lexcellent Multifrontal Methods: Parallelism,(MUMPS) Memory Usage and Numerical Aspects Habitation à diriger de recherches Ecole Normale Supérieure de Lyon, France

[Lygeros, 2003] John Lygeros, Karl Henrik Johansson, Slobodan N. Simié, Jun Zhang, S.Shankar Sastry "Dynamical properties of hybrid automata", IEEE transaction on automatic control, Vol 48, No1, 2003

[Lygeros, 2004] J. Lygeros, "Lecture Notes on Hybrid Systems", *Notes for an ENSIETA short course*, 54-58, 2004

Lygeros 2010, Stochastic Hybrid Systems: A powerful framework for complex large scale applications, John Lygeros, Maria Prandini, European Journal of Control 583-594 vol 16, 2010

[Lynch and all, 1996] N. Lynch, R. Segala, F. Vaandrager, and H. B. Weinberg, "Hybrid I/O automata," in *Hybrid Systems III*. New York: Springer-Verlag, 1996, number 1066 in LNCS, pp. 496–510.

[Liu, 2008] Bin Liu, David J. Hill "*Stability of Discrete Impulsive Hybrid Systems via Comparison Principle*", Proceedings of the 17th World Congress The International Federation of Automatic Control Seoul, Korea, July 6-11, 2008

[Mattioli, 2003], Mattiolli, J et Artiouchine, K, « Noyau de viabilité : une contrainte globale pour la modélisation de systèmes dynamiques »*JFPLC'2003 Douzièmes Journées Francophones de Programmation Logique et Programmation par Contraintes*, Amiens, 2003

[Mattiolli, 96] Mattiolli J., Doyen L., NajmanL., « Lattice operators underlying dynamic systems », *Set-Valued Analysis*, vol. 4, p. 119-134.1996

Magoulès 2013 Calcul scientifique parallèle, F. Magoulès, F.X. Roux, DUNOD

MESLEM 2008 Atteignabilité hybride des systèmes dynamiques continus par analyse par intervalles . Application à estimation ensembliste Nacim MESLEM PhD thesis, Université Paris Est, 2008

METIS 2011 Serial Graph >partitioning and Fill reducing Matrix Ordering G; Karypis University of Minnesota

[Mignoni] G. Migoni, E. Kofman, and F. E. Cellier. Quantization-based new integration methods for stiff ODEs. *Simulation: Trans. Soc. Model. Simul. Int.*, 88(4):387–407, 2012. (cited on page 44)

[MODRIO 2013] Eurêka ITEA2 MODRIO deliverable D 6.2.7 « analysis of complex systems with varying parameters » Supmeca-IFPEN France 2013 .

[Morton, 1985] Monrton, B. G "*new application of μ to real parameter variation problem* " IEEE conference on decision and control, volume 24, page: 233- 238, 1985

MPI 1999 Using MPI portable Parallel Programming with the Message Passing Interface MIT Press 1999

MUMPS 2005 MUltifrontal Massively Parallel sparse direct solver CEC ESPRIT n° 20160

MUMPS 2012 Multifrontal Methods: Parallelism, Memory Usage and Numerical Aspects

OpenMP 2001 Shared memory Parallel Programming Lecture notes in Computer science WOMPAT 2001 Workshop Springer 2001

[Pace, 2004] Gordon J. Pace, Gerardo Schneider“*Model checking polygonal differential inclusions using invariance kernels”* VMCAI’ 04, numero 2937 in LNCS, 2004

[Petzold1] L. Petzold. Automatic selection of methods for solving stiff and nonstiff systems of Ordinary Differential Equations. *SIAM J. on Sci. Stat. Comput.*, 4(1):136–148, 1983. (cited on page 38)

[Petzold2] L. R. Petzold. A description of DASSL: A differential/algebraic system solver. In *International Mathematics and Computers Simulation congress on Systems Simulation and Scientific computing*, Montreal, Canada, Aug 1982. (cited on page 38)

[Quincampoix, 19 91] Quincampoix M., « Problèmes de cibles en théorie du contrôle et des jeux différentiels », PhD thesis, Paris Dauphine, 1991.

[Raczynski, 2004] Raczynski, S. (2004) ‘Continuous simulation, differential inclusions, uncertainty, and traveling in time’, *Simulation*, Vol. 80, No. 2, pp.87–100

[Raczynski, 2007] Raczynski S., (2007) *’Determination of attainable regions in ship maneuvers : an application of differential inclusions to the modeling of ship dynamics’*. **Journal of Marine Science and Technology**.

[Raczynski, 2011] Raczynski, S (2011) ‘uncertainty, dualism and inverse reachable sets’ International journal of simulation and modelling vol. 10, no1, pp. 38-45 .

[Rubino, 2007] G . Rubino, B. Tuffin ‘*Simulations et méthodes de Monte Carlo*’ techniques de l’ingénieur, Référence AF600, 10 oct. 2007

[Rabier, 2007] Rabier, F (2007). « *Modélisation par la méthode des plans d’expériences du comportement dynamique d’un module IGBT utilisé en traction ferroviaire* » , Thèse, ENI de Tarbes.

[Rüdiger, 1998] Rüdiger H, Harald M, Willy S.,(1998). “An approach to a general view on tolerances in mechanical engineering”. 2nd International Workshop on Integrated Product Development IPD 98. Magdeburg. 65 - 76

[Saint-Pierre, 1994] Saint-Pierre P., « Approximation of Viability Kernel », *Applied Mathematics and Optimisation*, vol. 29, 1994, p. 187-209.

[Saint-Pierre, 1998] Saint-Pierre P., « Dynamical System with State Constraints, Theory and Applications », *IFIP Conference on Systems Modelling and Optimization*, 1998.

[Schimmerling et al , 1998] Schimmerling P, Sisson J.C, Zaidi A (1998) « pratique des plans d’expérience » techniques & documentation.

[Schneider, 2003], *Gerardo Schneider* “Invariance Kernels of Polygonal Differential Inclusions” , Technical report 2003-042, ISSN 1404-, Department of Information Technology Uppsala University Box 337, SE-751 05 Uppsala, Sweden, 2003

[Schneider, 2002] Gerardo Schneider, « *Analyse Algorithmique de Systèmes Hybrides polygonaux* », Université de Grenoble I_ Joseph Fourier, Sciences et Géographie VERIMAG, 2002.

[Schneider, 2002 a] Gerardo Schneider, Eugene Asarin, Godon Pace et Sergio Yovine, "SPeeDi a verification tool for polygonal hybrid system" , Proceedings of the 14th International Conference on Computer Aided Verification page, 354-358, Springer- Verlag London UK , 2002

[Site Web Mathmodelica]**http://www.mathcore.com/products/mathmodelica/**

[Site Web Modelica] **https://www.modelica.org/**

[Skelboe, 2006] Skelboe, S. «Adaptative partitionning techniques for ordinary differential equations .» *BIT Numerical Mathematics 46 (3)*, 2006: 617-629.

[Tian, 2000] Tian M.F., Shi R., 2000. Worst case tolerance analysis of linear Analog Circuits using sensitivity*bands* IEEE transactions on circuits and systems vol47, N°8

[Turki, 2008] Turki S., *"Ingénierie système guidée par les modèles: Application du standard IEEE15288, de l'architecture MDA et du langage SysML à la conception des systèmes mécatroniques",* Thèses outenue le 02/10/2008, Université du Sud ToulonVar.

[zaytoon, 2001] (sous la direction de) J. Zaytoon, « Systèmes dynamiques hybrides », HERMES science publication , 157-161, 2001.

Bibliography & tools DSM

- S. Eppinger, T. Browning, Design structure matrix method and applications, MIT Press, 2012.

- T. Yu and al, Dependency Structure matrix, genetic algorithms and effective recombination, evolutionary computation, 17 (4), 594-626, 2009.

- M.E. Sosa, T. Browning, J Mihm, Studying the dynamics of the architecture of software products, Proceedings of the ASME 2007 International Design Eng. Technical Conferences & Computers and Info – IDETC/CIE,sept. 4-7, Las Vegas, NV, 2007.

- S.K. Fixson, Product architecture assessment. A tool to link product, process and supply chain design decisions, Journal of operations management, 23: 345-369, 2005.

www.dsmweb.org : macros available for DSM construction.

www.latix.com , **www.plexusplanning.com** , **www.problematics.com** , **www.projectdsm.com**

Acknowledgements

We wish to thank all our partners involved in this project, the French ministry of industry and the European community through the ITEA2 Esprit program