Séminaire LaTeX[a]

Introduction à la composition de documents[b]

Solal Nathan

Lundi 03 Février 2020

CR@NS - Cachan Réseau @ Normal Sup'

[a] ou : « How I Learned to Stop Worrying and Love LaTeX »

[b] Ou comment ne pas utiliser Pandoc

Sommaire

Introduction

TEX et LATEX, historique

TEX et LATEX, historique

- TEX par Donald E. Knuth (1977) ;
- LATEX par Leslie Lamport (1983).

TEX et LATEX, historique

Votre utilisation

- CRANS : Factures ;
- BDE : Factures ;
- ENS : Enseignement, recherche, publications scientifiques (ou autre), thèses.

Exemples

Cachan Réseau @ Normale Sup'

Adresse : 61 Avenue du Président Wilson
94230 Cachan
Siret : 452 661 531 00015

Téléphone : 0000000000
Mail : contact@crans.org

Pour : Solal Nathan
Adresse : A409
Facture n° : 28173

Date : 29 août 2018 13:11

Désignation	Prix Unit. €	Quantité	Prix total €
Adhésion et Connexion 1 an	50,00 €	1	50,00 €

Total	50,00 €
Votre règlement	50,00 €
À PAYER	00,00 €

Exemples

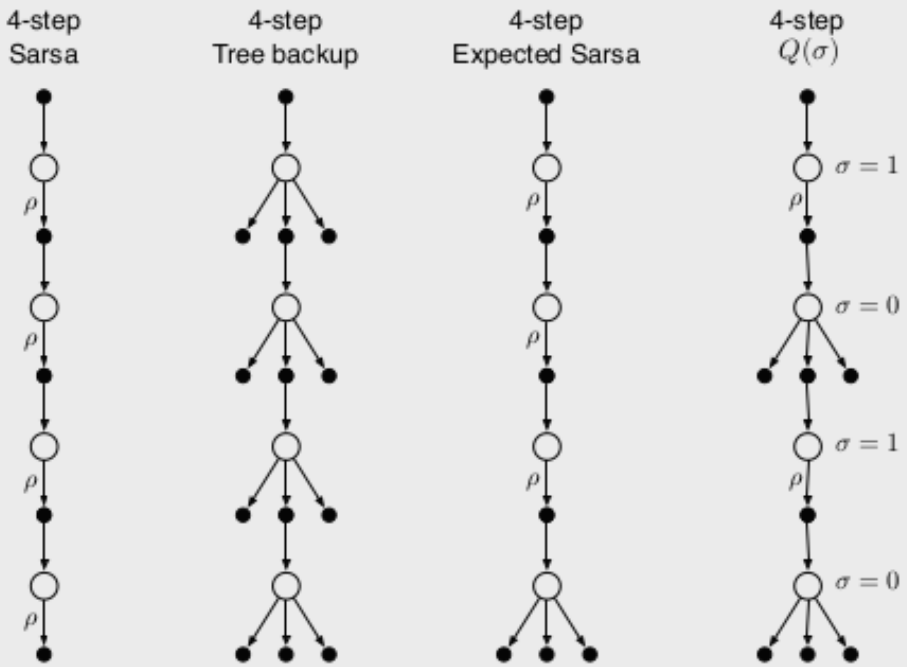


Figure 7.5: The backup diagrams of the three kinds of n-step action-value updates considered so far in this chapter (4-step case) plus the backup diagram of a fourth kind of update that unifies them all. The 'ρ's indicate half transitions on which importance sampling is required in the off-policy case. The fourth kind of update unifies all the others by choosing on a state-by-state basis whether to sample ($\sigma_t = 1$) or not ($\sigma_t = 0$).

And of course there would be many other possibilities, as suggested by the last diagram in the figure. To increase the possibilities even further we can consider a continuous

Installation

Installation

```
sudo apt install texlive-full # tout
# ou
sudo apt install texlive-base # minimal
```

Sinon vous pouvez utilisé un éditeur/compilateur en ligne : ShareLaTeX, CodiMd (Markdown), Overleaf...

https://overleaf.com

https://codimd.auro.re

Concepts clés

Concepts clés

LaTeX est un langage permettant la conception de documents numériques de haute qualité. Il s'agit d'un langage **balisé** un comme le *HTML* ou le *XML*.

Il s'agit d'un ensemble de macros pour TeX.

Philosophie

Il est à la fois **simple** et **puissant**. Il est **agnostique** de tout éditeur de texte ou logiciel, bien que ces derniers peuvent en faciliter l'utilisation. Facilité de partage et de collaboration : c'est juste un fichier texte !

Philosophie

La différence majeure avec d'autres moyen de mise en forme de document est qu'il se base sur une philosophie : **WYSIWYM** *(What You See Is What You Mean)* plutôt que l'habituel **WYSIWYG** *(What You See Is What You Get)*.

C'est à dire que l'utilisateur se concentre sur le **fond**, tandis que le l'ordinateur s'occupe de la **mise en page**.

Structure

Bon, c'est bien joli tout ça mais pour l'instant on a pas vu une seule ligne de code.

Structure

Structure de base :

```
\documentclass[a4paper, 12pt]{article}

\begin{document}
Hello world!
\end{document}
```

Strucutre

Mots clés :

```
\keyword[option1, option2, option3=3px]{argument1}{argument2}
```

Structure

Exemple :

```
\frac{x^2}{3 \alpha}
```

donne

$$\frac{x^2}{3\alpha}$$

Structure

Structure de base d'un document texte : Titre, Author, Date, Section, Subsection

```
\documentclass[12pt, a4paper, french]{article}

\title{Document d'exemple}
\author{Solal Nathan}
\date{hier}

\begin{document}

\maketitle

\section{Présentation}

\section{Exemples}
\subsection{Des choses}
bla bla

\subsection{plop toto}
balais

\end{document}
```

Structure

Document d'exemple

Solal Nathan

hier

1 Présentation

2 Exemples

2.1 Des choses

bla bla

2.2 plop toto

balais

Environnements

Dans LaTeX, tous les environnements sont de la forme :

```
\begin{envi}[options]
    plop
    toto
\end{envi}
```

On peut alors appliquer des changement à l'ensemble d'un environnent : taille de la police, centrage, etc...

Envrionnements - exemples

```
% Défintion de l'environnement
\newenvironment{king}
{ \rule{1ex}{1ex}\hspace{\stretch{1}} }
{ \hspace{\stretch{1}}\rule{1ex}{1ex} }

% Appel de ce dernier
\begin{king}
My humble subjects \ldots
\end{king}
```

Donne alors

■ My humble subjects … ■

Environnements - exemples

Environnements mathématiques : $, $$ et equations.

Macro

C’est quoi une macro ?

On peut voir ça comme une **fonction**. Latex est un ensemble de macros pour Tex.

Macro - Exemples

```
\newcommand{\partialderivative}[3][]{
\frac{\partial^{#1} #2}{\partial #3^{#1}}}
```

partialderivative{f}{x} donne $\frac{\partial f}{\partial x}$

partialderivative[2]{f}{x} donne $\frac{\partial^2 f}{\partial x^2}$

partialderivative[n]{F}{y} donne $\frac{\partial^n F}{\partial y^n}$

Macro - Exemples

On peut également redéfinir des commandes déjà existantes à l'aide de `renewcommand`.

Cela permet, par exemple, de transformer directement les ϵ (epsilon) en ε (varesilon).

Comment utiliser un packet

```
\usepackage[utf8]{inputenc}
\usepackage[french]{babel}
\usepackage{graphicx}
\usepackage{float}
\usepackage{minted}
```

Quelques packages

Beamer : pour faire des jolis diapo

exemple : celui-ci :)

Quelques packages

minted : pour faire du syntax highligh de code

exemple :

```
import numpy as np

t = np.linspace(0,2,100)

print("A + B")
```

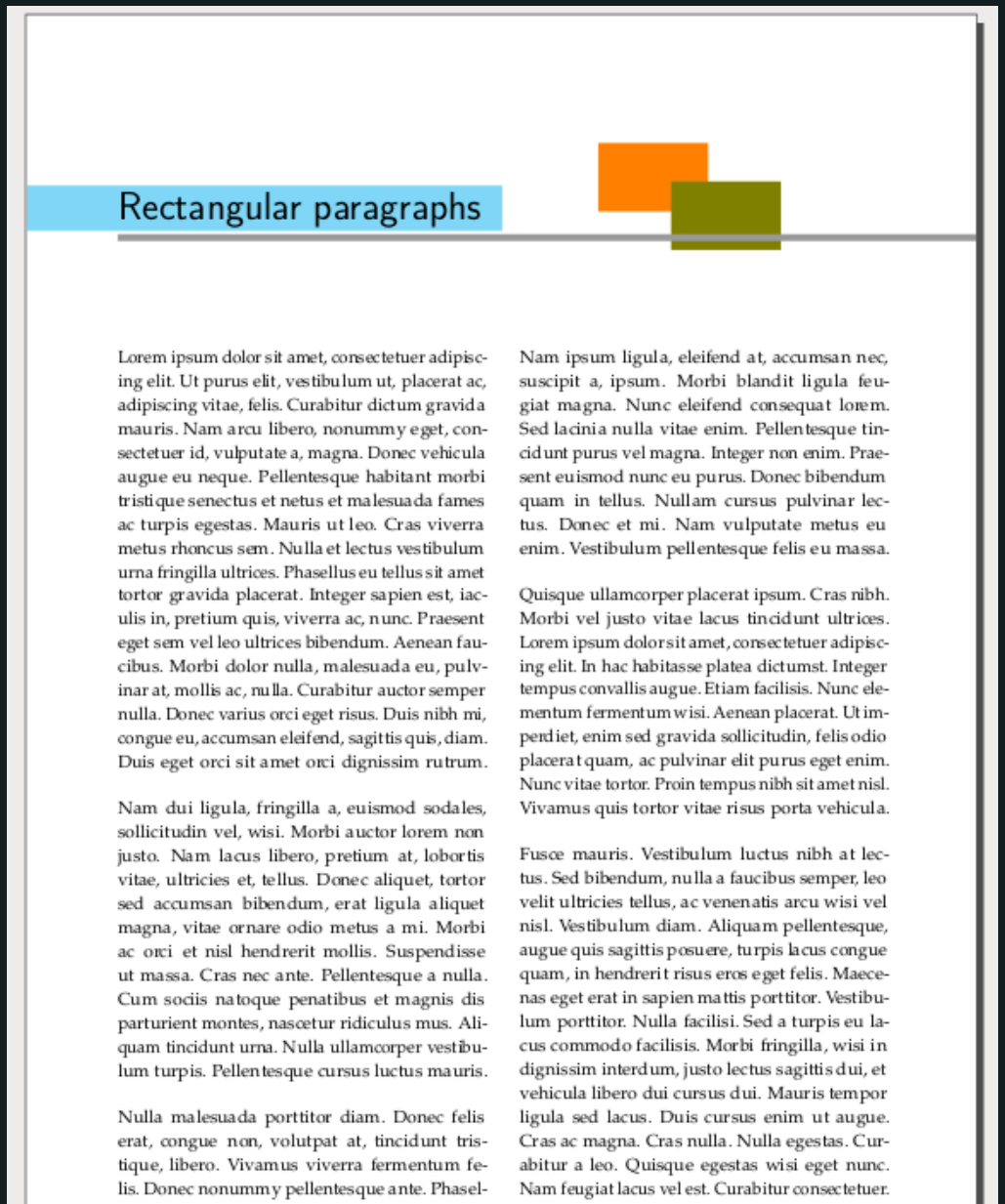
Quelques packages

Et pleins d'autres :

- Tikz, pour faire de jolis graphiques
- graphics, pour faire des figures
- float, pour positionner ses figures
- xcolor, pour ajouter des couleurs custom
- Bibtex, pour faire de la bibliographie (compatible avec des outils tiers tel que Zotero)
- ... *and much more*

Démonstration de force

Démonstration de force



Rectangular paragraphs

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Nam dui ligula, fringilla a, euismod sodales, sollicitudin vel, wisi. Morbi auctor lorem non justo. Nam lacus libero, pretium at, lobortis vitae, ultricies et, tellus. Donec aliquet, tortor sed accumsan bibendum, erat ligula aliquet magna, vitae ornare odio metus a mi. Morbi ac orci et nisl hendrerit mollis. Suspendisse ut massa. Cras nec ante. Pellentesque a nulla. Cum sociis natoque penatibus et magnis dis parturient montes, nascetur ridiculus mus. Aliquam tincidunt urna. Nulla ullamcorper vestibulum turpis. Pellentesque cursus luctus mauris.

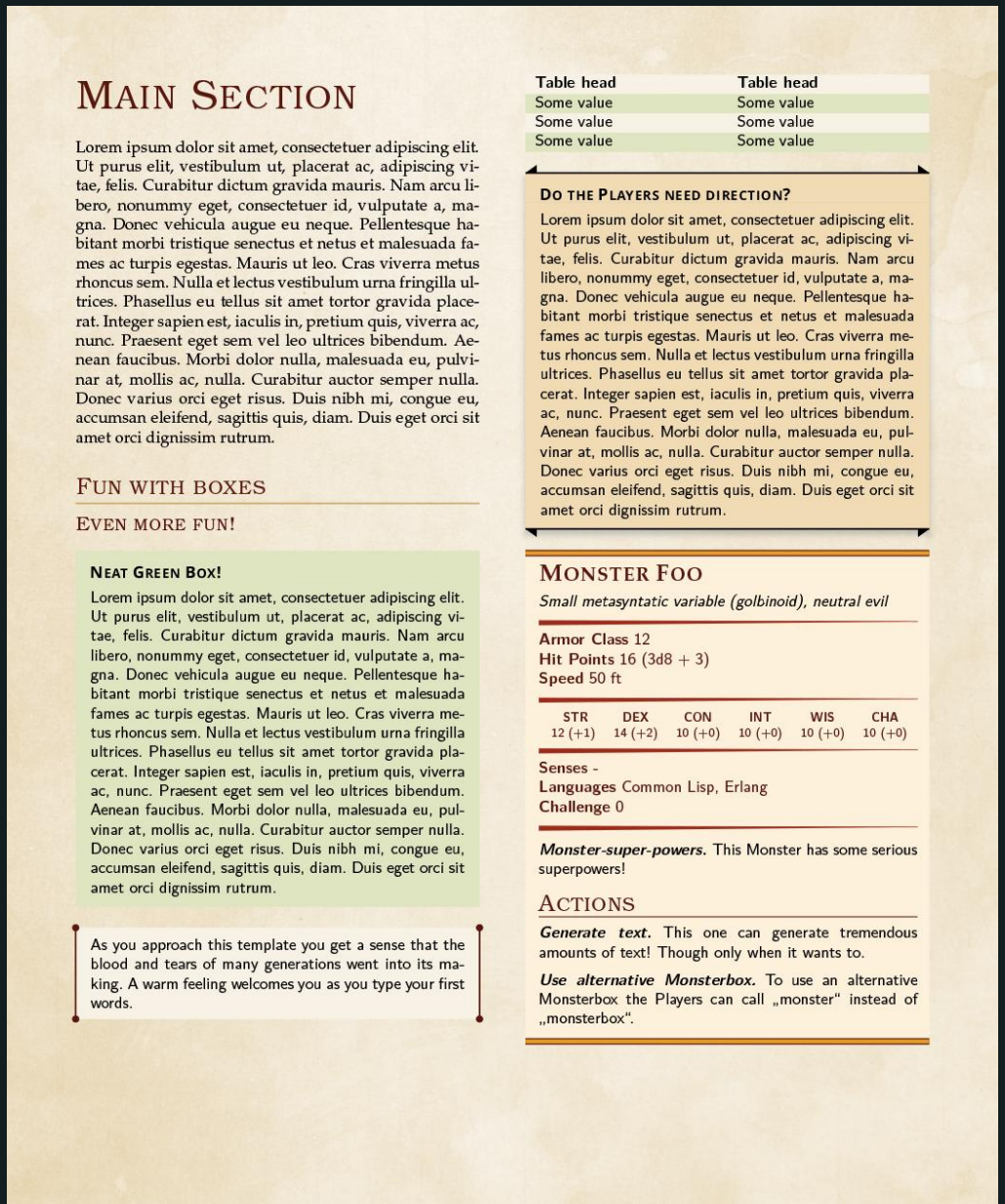
Nulla malesuada porttitor diam. Donec felis erat, congue non, volutpat at, tincidunt tristique, libero. Vivamus viverra fermentum felis. Donec nonummy pellentesque ante. Phasel-

Nam ipsum ligula, eleifend at, accumsan nec, suscipit a, ipsum. Morbi blandit ligula feugiat magna. Nunc eleifend consequat lorem. Sed lacinia nulla vitae enim. Pellentesque tincidunt purus vel magna. Integer non enim. Praesent euismod nunc eu purus. Donec bibendum quam in tellus. Nullam cursus pulvinar lectus. Donec et mi. Nam vulputate metus eu enim. Vestibulum pellentesque felis eu massa.

Quisque ullamcorper placerat ipsum. Cras nibh. Morbi vel justo vitae lacus tincidunt ultrices. Lorem ipsum dolor sit amet, consectetuer adipiscing elit. In hac habitasse platea dictumst. Integer tempus convallis augue. Etiam facilisis. Nunc elementum fermentum wisi. Aenean placerat. Ut imperdiet, enim sed gravida sollicitudin, felis odio placerat quam, ac pulvinar elit purus eget enim. Nunc vitae tortor. Proin tempus nibh sit amet nisl. Vivamus quis tortor vitae risus porta vehicula.

Fusce mauris. Vestibulum luctus nibh at lectus. Sed bibendum, nulla a faucibus semper, leo velit ultricies tellus, ac venenatis arcu wisi vel nisl. Vestibulum diam. Aliquam pellentesque, augue quis sagittis posuere, turpis lacus congue quam, in hendrerit risus eros eget felis. Maecenas eget erat in sapien mattis porttitor. Vestibulum porttitor. Nulla facilisi. Sed a turpis eu lacus commodo facilisis. Morbi fringilla, wisi in dignissim interdum, justo lectus sagittis dui, et vehicula libero dui cursus dui. Mauris tempor ligula sed lacus. Duis cursus enim ut augue. Cras ac magna. Cras nulla. Nulla egestas. Curabitur a leo. Quisque egestas wisi eget nunc. Nam feugiat lacus vel est. Curabitur consectetuer.

Démonstration de force



Main Section

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Fun with boxes

Even more fun!

Neat Green Box!

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

As you approach this template you get a sense that the blood and tears of many generations went into its making. A warm feeling welcomes you as you type your first words.

Table head	Table head
Some value	Some value
Some value	Some value
Some value	Some value

Do the Players need direction?

Lorem ipsum dolor sit amet, consectetuer adipiscing elit. Ut purus elit, vestibulum ut, placerat ac, adipiscing vitae, felis. Curabitur dictum gravida mauris. Nam arcu libero, nonummy eget, consectetuer id, vulputate a, magna. Donec vehicula augue eu neque. Pellentesque habitant morbi tristique senectus et netus et malesuada fames ac turpis egestas. Mauris ut leo. Cras viverra metus rhoncus sem. Nulla et lectus vestibulum urna fringilla ultrices. Phasellus eu tellus sit amet tortor gravida placerat. Integer sapien est, iaculis in, pretium quis, viverra ac, nunc. Praesent eget sem vel leo ultrices bibendum. Aenean faucibus. Morbi dolor nulla, malesuada eu, pulvinar at, mollis ac, nulla. Curabitur auctor semper nulla. Donec varius orci eget risus. Duis nibh mi, congue eu, accumsan eleifend, sagittis quis, diam. Duis eget orci sit amet orci dignissim rutrum.

Monster Foo

Small metasyntatic variable (golbinoid), neutral evil

Armor Class 12
Hit Points 16 (3d8 + 3)
Speed 50 ft

STR	DEX	CON	INT	WIS	CHA
12 (+1)	14 (+2)	10 (+0)	10 (+0)	10 (+0)	10 (+0)

Senses -
Languages Common Lisp, Erlang
Challenge 0

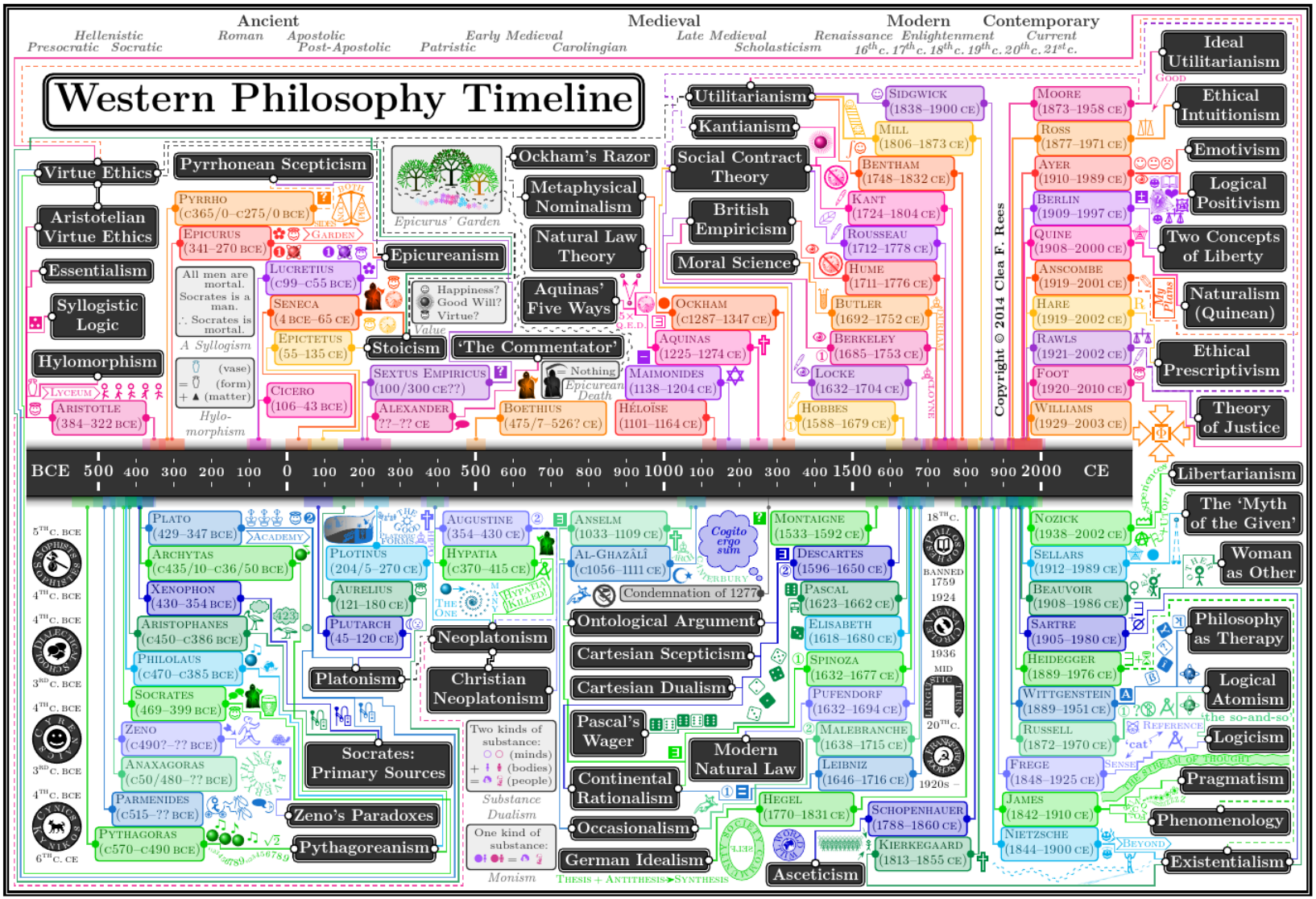
Monster-super-powers. This Monster has some serious superpowers!

Actions

Generate text. This one can generate tremendous amounts of text! Though only when it wants to.

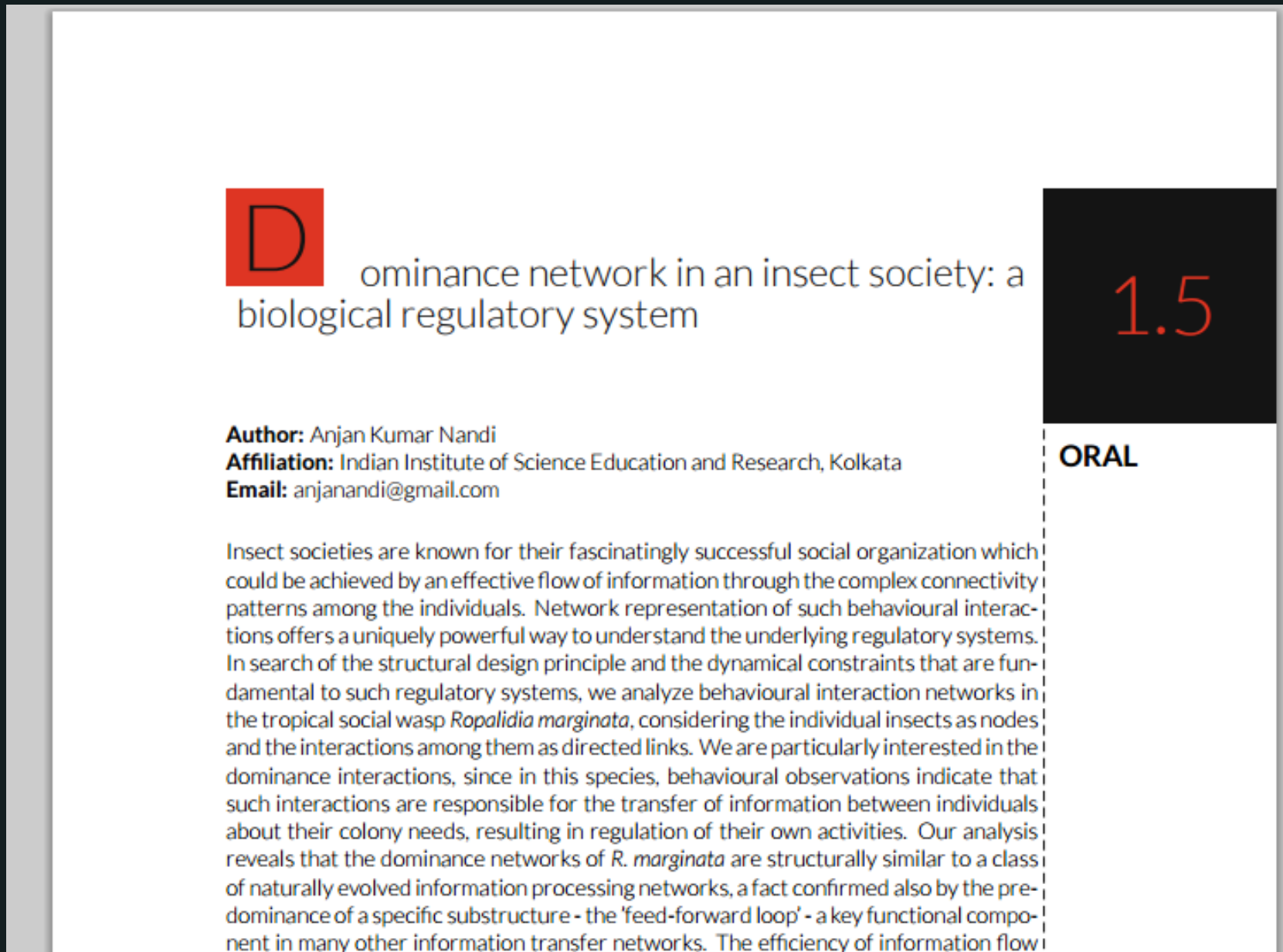
Use alternative Monsterbox. To use an alternative Monsterbox the Players can call „monster“ instead of „monsterbox“.

Démonstration de force



Démonstration de force

Démonstration de force



Dominance network in an insect society: a biological regulatory system

1.5

ORAL

Author: Anjan Kumar Nandi
Affiliation: Indian Institute of Science Education and Research, Kolkata
Email: anjanandi@gmail.com

Insect societies are known for their fascinatingly successful social organization which could be achieved by an effective flow of information through the complex connectivity patterns among the individuals. Network representation of such behavioural interactions offers a uniquely powerful way to understand the underlying regulatory systems. In search of the structural design principle and the dynamical constraints that are fundamental to such regulatory systems, we analyze behavioural interaction networks in the tropical social wasp *Ropalidia marginata*, considering the individual insects as nodes and the interactions among them as directed links. We are particularly interested in the dominance interactions, since in this species, behavioural observations indicate that such interactions are responsible for the transfer of information between individuals about their colony needs, resulting in regulation of their own activities. Our analysis reveals that the dominance networks of *R. marginata* are structurally similar to a class of naturally evolved information processing networks, a fact confirmed also by the predominance of a specific substructure - the 'feed-forward loop' - a key functional component in many other information transfer networks. The efficiency of information flow

Démonstration de force

ma con assi costantemente orientati secondo le tre direzioni geografiche standard: Nord, Est e centro della Terra (*Down*).

1.7 Assi vento (*Wind Axes*, W)

La terna *assi vento* $\mathcal{T}_W$ è una terna trirettangola levogira con origine nel baricentro del velivolo (punto $G \equiv C$) ed avente l'asse vento longitudinale x_W diretto secondo la direzione della velocità $\boldsymbol{V} \equiv \boldsymbol{V}_G$ del velivolo, con verso positivo nel senso del moto (figura 1.13). L'asse vento z_W è definito dall'intersezione del piano verticale π_V contenente $\boldsymbol{V}$ e G con il piano π_n normale alla traiettoria in G, con verso positivo verso il basso. L'asse trasversale y_W è tale da completare la terna $\{G, x_W, y_W, z_W\}$.

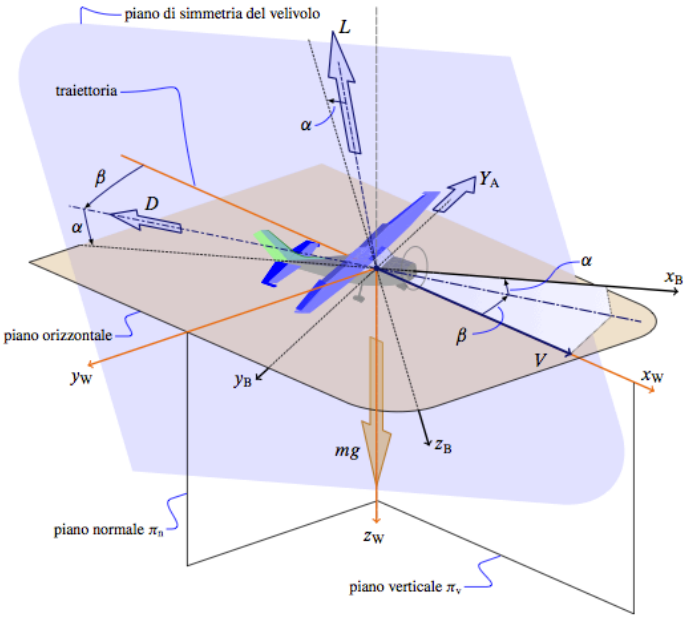


Figura 1.13 Terna di assi vento $\mathcal{T}_W = \{G, x_W, y_W, z_W\}$ (o assi traiettoria). In questa particolare circostanza la traiettoria del baricentro è orizzontale e l'orientamento del velivolo non è simmetrico rispetto al piano verticale $x_W z_W$.

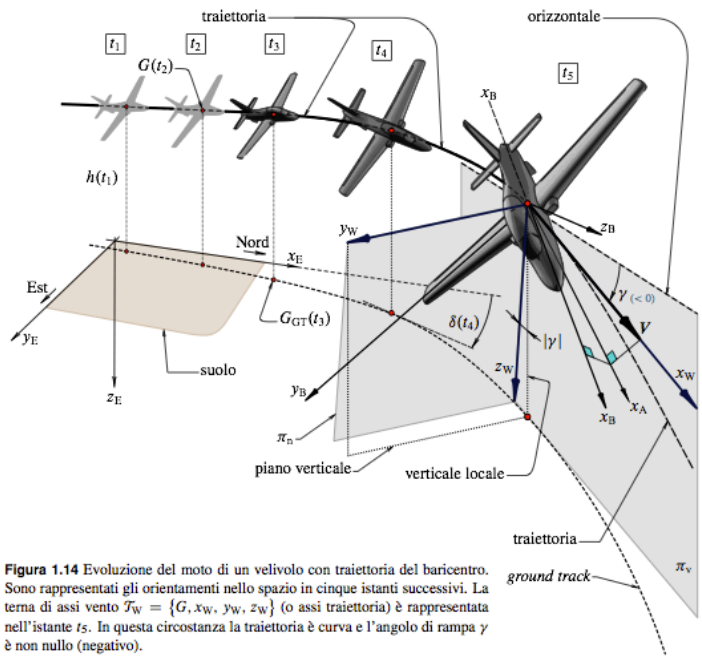


Figura 1.14 Evoluzione del moto di un velivolo con traiettoria del baricentro. Sono rappresentati gli orientamenti nello spazio in cinque istanti successivi. La terna di assi vento $\mathcal{T}_W = \{G, x_W, y_W, z_W\}$ (o assi traiettoria) è rappresentata nell'istante t_5. In questa circostanza la traiettoria è curva e l'angolo di rampa γ è non nullo (negativo).

Si noti che, secondo la definizione data sopra, l'asse vento trasversale y_W è sempre orizzontale. Esso è la normale al piano $\{G, x_W, z_W\}$ che è, per definizione, costantemente verticale come si osserva dalle figure 1.13, 1.14 e 1.15.

Nella figura 1.13, è rappresentato il caso particolare in cui la traiettoria del baricentro è orizzontale. Si osserva che, anche per un orientamento del velivolo non simmetrico rispetto al piano verticale $x_W z_W$, in questa circostanza gli assi vento presentano un asse z_W verticale (allineato con la forza peso mg).

Nella figura 1.14, è rappresentata un'evoluzione in cui la traiettoria del baricentro si incurva e allo stesso tempo l'orientamento del velivolo è non simmetrico rispetto al piano verticale π_v. L'asse vento x_W, per definizione tangente alla traiettoria, è non orizzontale e l'asse z_W è non verticale. Naturalmente — come da definizione — anche in questo caso, in cui il piano π_n non è verticale, l'asse vento y_W è orizzontale.

Tale particolarità evidenzia un'importante differenza tra la definizione data in questa sede e la definizione di assi vento che si ritrova in diversi libri di testo americani e anglosassoni. Secondo questi ultimi [13, 9, 50] l'asse longitudinale coincide con x_W; il terzo

Démonstration de force

However, the advantage of deep models is that they can be synthesized to VHDL or GraphML code (or whatever backend is available).

2.4 MODELING IN SYSTEMC

The theory behind application modeling in the SystemC flavor of ForSyDe is the same as in Haskell. The main difference is, obviously, the language constructs – Haskell is a purely functional language while SystemC is a template library implemented in C++, an object-oriented language. In SystemC, process constructors are implemented as classes, and processes are created by deriving the appropriate process constructor class and providing implementations for the pure virtual functions. Once declared, the processes are instantiated as objects which are used to create modules, which in turn are attached to form a network using channels. Channels are the communication elements of SystemC and can be either something as simple as a wire or as complex as a FIFO queue or a bus. Listing 2.3 shows an equivalent implementation of the *adder* process which was declared in Haskell in Listing 2.1.

```
#include "forsyde.h"

using namespace ForSyDe::SY;

class adderSY : public comb2<int, int, int> {
  public:
    adderSY(sc_module_name _name)
        : comb2<int, int, int>(_name) {}
  protected:
    int _func(int a, int b) {
      return a + b;
    }
};
```

LISTING 2.3 – Declaration of a SystemC equivalent of the Haskell *adder* process.

CHAPTER 3

ALTERNATIVE DSLS

This chapter introduces the reader to two alternative domain-specific programming languages – Obsidian and SkePU – designed either completely or partly for writing applications which targets execution on GPGPUs. Both languages provide backends for synthesizing their models into optimized CUDA C code and the goal of this chapter is to analyze how they compare with ForSyDe and whether the same synthesis solutions and methods can be applied for the ForSyDe component.

3.1 OBSIDIAN

The text in this section is primarily based on the material found in [46, 47].

OBSIDIAN IS A domain-specific programming language for programming data parallel applications for execution on GPGPUS (specifically CUDA platforms). It is a work in progress developed by the Department of Computer Science and Engineering at Chalmers University of Technology and Göteborg University with the goal to offer a tool which "encourages experimentation" by "rais[ing] the level of abstraction of GPU programming" [46].

3.1.1 *Implementation and usage*

Implemented as an embedded language in Haskell[1], Obsidian consists of:

- a collection of scalar type operations;
- arrays together with library functions to operate on these arrays; and
- a collection of operations called combinators which construct the GPGPU kernels.

[1] Obsidian has been implemented twice, the first approach based on monads and the second based on arrows.

Démonstration de force

NIPAJIN

v1.8.1 ; Traduction française : Alain Curato

Un système de jeu de rôle simple et universel, par LUDUS LEONIS.

NIP'AJIN Conçu en vue d'aventures one-shot et de campagnes courtes, ce jeu se prononce ['nɪpaˌʃɪn] *– acronyme de « personne n'est parfoit, mais chacun peut contribuer ». Il vise à permettre aux personnages de se partager le premier rôle, sans les enfermer dans un carcon de règles.*

ludus-leonis.com/fr/

@ludusleonis

RÈGLES POUR LES JOUEURS

Chaque **personnage** est au départ une feuille A4 vierge. On trace sur cette **feuille de personnage** une ligne afin de la partager en deux feuilles A5, et la partie de droite est encore scindée en deux A6.

Dans la moitié de gauche, les joueurs notent une brève description de leur héros : nom, origine, apparence générale, suivie de son **historique**, écrit de façon télégraphique ou en phrases complètes. Cette description doit montrer ce que le personnage a accompli, plutôt que ses talents – c'est le meneur, au cours du jeu, qui décidera de ces derniers. Ainsi, on écrira p. ex. « *a été déménageur de force* » plutôt que « *est fort* ». Le meneur et les joueurs se mettent d'accord sur les **Possessions** (◈ 2) et les **Pouvoirs** (◈ 3) qui leur semblent convenir.

Le joueur dispose ensuite dans la section de droite, en haut, une suite de D4, D6, D8, D10 et D12. Il en choisit un comme **Dé de résistance** (♥) et le pose, sa face la plus forte visible, dans la partie de gauche de sa feuille. Si le ♥ descend en-dessous de 1 au cours du jeu, le personnage est éliminé.

dans la partie supérieure droite de sa feuille. Sur un jet de 1, l'action est un **échec automatique**. Sinon, le meneur de jeu cherche dans l'historique du personnage un éventuel **modificateur** à ajouter au dé.

L'historique indique…	+/-
claire défaillance	-4
novice, maladroit	-2
quelque peu rouillé	-1
moyen sans plus	0
amateur, assez doué	+1
professionnel, routinier, expérimenté	+2
vétéran, expérience mesurée en décennies	+4

Si le résultat final égale ou dépasse le **seuil** (◎) de l'action, c'est un succès. Un total final de 1 n'est pas un échec automatique, mais ça ne suffira que rarement.

Difficulté	◎	Exemple
facile	2	–
bonnes conditions	3	bons outils
moyen	4	–
mauvaises conditions	5	manque de lumière
difficile	6	saut à moto
très difficile	8	corde raide
légendaire	12	–

Démonstration de force

The first step in modeling $\hat{g}_j(st)$ is the estimation of the probability density function (Equation (5.2)) for the nominal executions with parameters $Fb_{0j} = \{5, 7, 15, 20, 40\}$ and $bw = 6.328$ (Figure 5.1). Note that for each value in the horizontal axis, the *KDE* value corresponds to the summation of all underlying kernels at the same point. From Equation (5.2) we can see that $st' \in Fb_{ej}$ determines each kernel's offset and bw the dispersion of the density. In particular, when using the Gaussian kernel, st' corresponds to its mean and bw to its standard deviation.

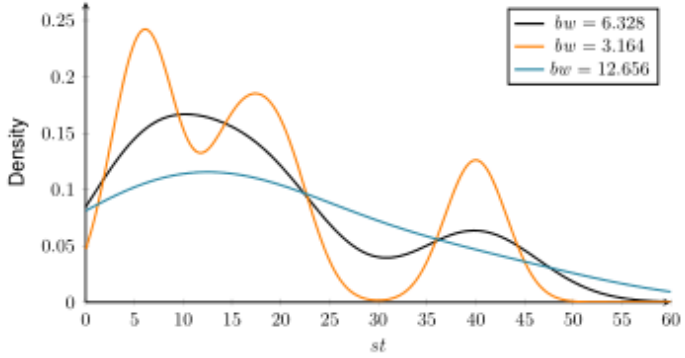
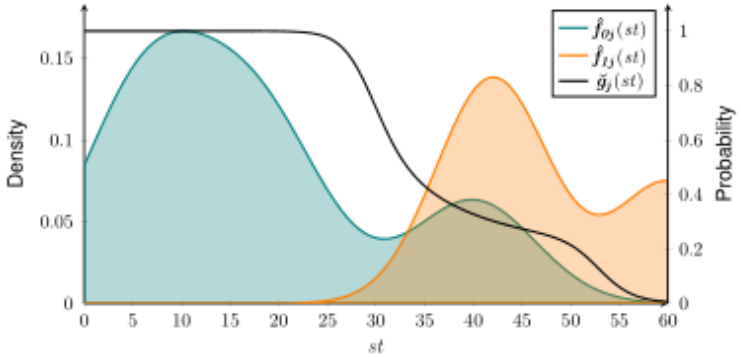
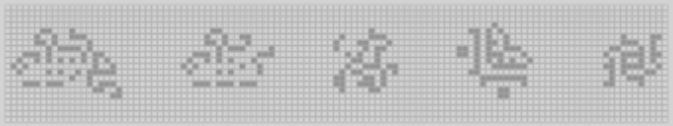


Figure 5.2: Impact of bw value

Figure 5.2 provides a visual intuition on the effect of the parameter bw in the estimate. A sensible selection of bw is crucial in order to yield good results as using a small bw value will reflect sampling artifacts whereas a large bw value will smooth some behavioral trends.



Démonstration de force



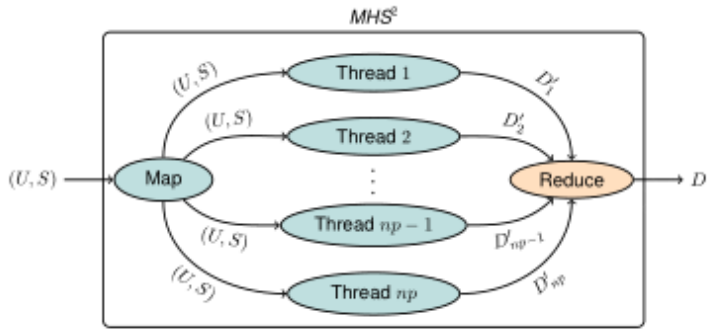
3 *MHS*^{2} – Parallelization

In this section we present the parallelized version of *MHS*^{2}. As explained in Section 1.3.1.2 (page 20), our goal in this chapter is to enable the computation of *MHSs* in parallel or even distributed environments.

This chapter is divided as follows. First, we introduce our *MHS* generation algorithm. Second, we evaluate the performance of our algorithm.

3.1 Approach

The parallel algorithm can be seen as a Map-Reduce algorithm [Dean and Ghemawat, 2004] (Figure 3.1). The map task (Algorithm 3.1) is a modified version of Algorithm 2.1 (page 28).



Démonstration de force

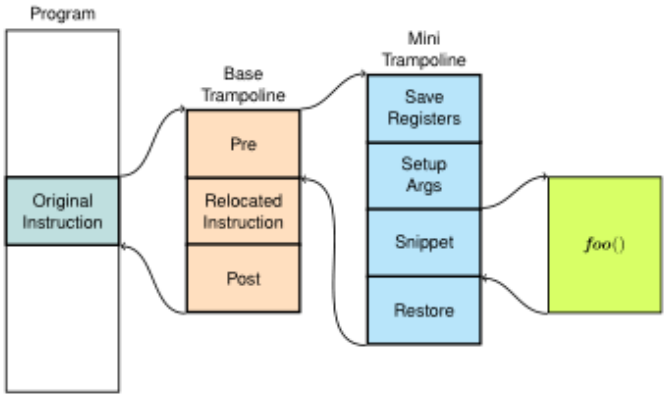


Figure 6.1: Instrumentation code insertion (adapted from [Tikir and Hollingsworth, 2002])

Several instrumentation techniques have been proposed. These techniques include:

- Manually placing instrumentation in a program before it executes.
- Static binary modification [Bus et al., 2004, Larus and Schnarr, 1995, Srivastava and Eustace, 1994].
- Dynamic/Live instrumentation in executing programs [Arnold and Ryder, 2001, Hollingsworth et al., 1997, Luk et al., 2005a,b, Lattner, 2002, Nethercote and Seward, 2003].

Code Coverage

Code coverage is an analysis method that determines which parts of a particular system have been executed (covered) during a test run [Miller and Maloney, 1963, Graham et al., 2006]. Well known examples of code coverage tools are *EMMA*[5] and *Cobertura*[6] (see [Yang et al., 2006] for more examples).

In the scope of *SFL*, by using a code coverage tool in conjunction with a test suite, it is possible to create spectra and thereby identify which components are more likely to be involved in the test suite failure. To solve the potential scalability issues that *SFL* techniques may have when instrumenting large software programs, a dynamic instrumentation approach, called **Dynamic Code Coverage (*DCC*)**, was proposed [Perez, 2012]. This technique automatically adjusts the instrumentation granularity of the system under analysis. First, the system is instrumented using a coarse granularity (*e.g.*, package level in *Java*). Then, *SFL* is executed and, based on the diagnostic results, *DCC* decides which components should be re-instrumented with a finer granularity level (*e.g.*, in *Java* one should instrument classes, then methods, and finally statements). After

Conclusion

Où trouver de la documentation

- Latex Wiki Book : https://en.wikibooks.org/wiki/LaTeX
- Latex Cookbook : http://latex-cookbook.net/
- Awesome : https://github.com/egeerardyn/awesome-LaTeX
- CTAN et TUG : https://ctan.org
- Stackexchange : https://tex.stackexchange.com/
- Venez sur `#latex` sur IRC

Travaux Pratiques

Je vous donne des exemples, vous essayez de les faire, puis je vous donne la solution

Exercice 1

Ceci est du texte, $x = 2$.

Exercice 1 - Corrigé

```
Ceci est du text $x = 2$.
```

Exercice 2

$$\alpha = \sum_{i=0}^{+\infty} x_i$$

Exercice 2 - Corrigé

```
$$\alpha = \sum_{i=0}^{+\infty} x_i$$
```

Exercice 3

Aujourd'hui il fait beau, il fait 12°

Exercice 3 - Corrigé

```
Aujourd'hui il fait beau, il
fait $12^{\circ}$
% En vrai c'est pas la bonne
% manière de faire
% Utiliser le package siunitx
% est mieux !
```

Exercice 4

Je peux même **écrire** des choses { entre } crochets.

Exercice 4 - Corrigé

```
Je peux même \textbf{écrire} des
choses \{ entre \} crochets.
```

Exercice 5

Le $\mathrm{C}^{r}_{an}s^{\mathrm{TM}}$

Exercice 5 - Corrigé

```
Le \smallcaps{C_a^r_ns}
$^{\text{\smallcaps{TM}}}$
```

Exercice 6

$$\ddot{x} + a\dot{x} = b, \quad \forall (a, b) \in \mathbb{R}$$

Exercice 6 - Corrigé

```
$$\ddot{x} + a \dot{x} = b,
\quad \forall (a,b) \in
\mathbb{R}$$
```

Exercice 7

$$\frac{n!}{k!(n-k)!} = \binom{n}{k}$$

Exercice 7 - Corrigé

```
$$\frac{n!}{k!(n-k)!}
= \binom{n}{k}$$
```